

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Luiz Ricardo Flôres Maestri

**FERRAMENTA DE SUPORTE AO PROCESSO DE
VOLUNTARIADO DA INICIATIVA COMPUTAÇÃO NA
ESCOLA**

Florianópolis

2018

Luiz Ricardo Flôres Maestri

**FERRAMENTA DE SUPORTE AO PROCESSO DE
VOLUNTARIADO DA INICIATIVA COMPUTAÇÃO NA
ESCOLA**

TCC submetido ao curso de Sistemas
de Informação para a obtenção do Grau
de Bacharel em Sistemas de Informa-
ção.

Orientador: Prof. Dr. Jean Carlo Rossa
Hauck

Florianópolis

2018

Catalogação na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:
<http://www.bu.ufsc.br/design/Catalogacao.html>

Luiz Ricardo Flôres Maestri

**FERRAMENTA DE SUPORTE AO PROCESSO DE
VOLUNTARIADO DA INICIATIVA COMPUTAÇÃO NA
ESCOLA**

Este TCC foi julgado aprovado para a obtenção do Título de “Bacharel em Sistemas de Informação”, e aprovado em sua forma final pelo curso de Sistemas de Informação.

Florianópolis, 27 de Novembro 2018.

Prof. Cristian Koliver, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Dr. Jean Carlo Rossa Hauck
Orientador

Prof. Leandro José Komosinski, Dr.

Nathalia da Cruz Alves, Bsc

Minha mãe que me ensinou o que é fé e
perseverança.

Todas as pessoas deveriam aprender a programar, pois isso ensina a pensar.

Steve Jobs

RESUMO

Atualmente o mercado na área de tecnologia apresenta-se no mundo em grande crescimento. Todavia no Brasil não se tem percebido interesse suficiente na capacitação de jovens para atuarem na área. Iniciativas como a Computação na Escola (CnE), que utilizam esforços voluntários para apoiar o ensino de computação para crianças e jovens, nascem procurando atender esta deficiência na educação básica. A gestão do processo completo de voluntariado se torna algo complexo e trabalhoso, principalmente pelas dificuldades de comunicação devido à troca de documentos por meio físico entre as entidades envolvidas. Nesse contexto, este trabalho propõem o desenvolvimento e implantação de um sistema web a fim de auxiliar a gestão do processo de voluntariado. O sistema foi concebido, modelado, desenvolvido, testado e implantado. Através disto busca-se favorecer a CnE a realizar sua proposta de estimular o ensino de computação a crianças e adolescentes. Tendo este objetivo em mente foi averiguado por meio de pesquisa na literatura e entrevista com os responsáveis, melhorias necessárias no gerenciamento de processos de iniciativas voluntárias para que as mesmas possam ser corrigidas com a implementação de um sistema web.

Palavras-chave: Sistema Web, Computação na Escola, Gestão de Processos, Voluntariado, Ensino de Programação.

ABSTRACT

Currently, the market in technology is in the world in great growth, however, in Brazil, there has not been perceived enough interest in the training of young people to work in the area. Initiatives such as Computação na Escola (CnE), which use voluntary efforts to support computer education for children and young people, are born seeking to address this deficiency in basic education, yet managing the entire process of volunteering becomes complex and cumbersome, especially due to communication difficulties due to the physical exchange of documents between the entities involved. In this context, this work proposes the development and implementation of a web system in order to assist the management of the volunteer process. The system had been designed, modeled, developed, tested and deployed. Through this, it seems to favor the CnE to carry out its proposal to stimulate the teaching of computation to children and adolescents. Having this goal in mind had been ascertained through research in the literature and interview with those responsible, necessary improvements in the management of voluntary initiatives processes so that they can be corrected with the implementation of a web system.

Keywords: Web System, Computação na Escola, Process Management, Voluntary, Programming Teaching.

LISTA DE FIGURAS

Figura 1 As fases do ciclo de gestão do voluntariado (MARCOS; AMADOR, 2014).....	19
Figura 2 Gênero dos voluntários (MENDES, 2015).....	20
Figura 3 Idade dos voluntários (MENDES, 2015).....	20
Figura 4 Profissão dos voluntários (MENDES, 2015).....	21
Figura 5 Renda dos voluntários (MENDES, 2015).....	21
Figura 6 Escolaridade dos voluntários (MENDES, 2015).....	22
Figura 7 Estado Civil dos voluntários (MENDES, 2015).....	22
Figura 8 Logo Amigos da Escola – Todos pela Educação (DIÁRIO CATARINENSE, 2012).....	23
Figura 9 Típica organização cliente-servidor de um web site/sistema (TANENBAUM; STEEN, 2007).....	24
Figura 10 Ecossistema da plataforma Spring IO (DEVMIDIA, 2015).	26
Figura 11 Virtual Dom calculando a diferença após alteração de estado para otimização da nova renderização (EISENMAN, 2015)...	27
Figura 12 Fluxo de dados proposto pela arquitetura Flux (FACEBOOK, 2015).....	28
Figura 13 Tela inicial do sistema Better Impact (2018).....	33
Figura 14 Listagem de Voluntários (TEAM KINECTIC, 2018).....	33
Figura 15 Tela de boas vindas do Volgistics (2018).....	34
Figura 16 Dashboard do voluntário (DO-IT, 2018).....	35
Figura 17 Site da Samaritan Technologies (2018).....	36
Figura 18 Site da Volunteer Hub (2018).....	36
Figura 19 Diagrama do processo de voluntariado (fornecido por um dos coordenadores da CnE).....	40
Figura 20 Diagrama de casos de uso.....	44
Figura 21 Tela inicial do Responsável pela empresa.....	50
Figura 22 Tela de detalhes do funcionário.	51
Figura 23 Tela inicial do Voluntário.	52
Figura 24 Tela de avaliação de treinamento.....	53
Figura 25 Diagrama Entidade-Relacional	55
Figura 26 Código de configuração para funcionamento assíncrono	57
Figura 27 Exemplo de application.properties para configuração do	

projeto.....	57
Figura 28 Classe ActivityResource	58
Figura 29 Exemplo de Classe Java utilizando <i>Annotations</i> do Lombok	60
Figura 30 Código gerado pela <i>Annotation @Value</i>	61
Figura 31 Método gerado pela <i>Annotation @Wither</i>	62
Figura 32 código gerado pela <i>Annotation @Builder</i>	62
Figura 33 código gerado pela <i>Annotation @FieldDefaults</i>	63
Figura 34 Exemplo de PO com as <i>Annotations</i> do Lombok.....	64
Figura 35 Exemplo método gerado pela <i>Annotation @Wither</i> do Lombok.....	65
Figura 36 Exemplo de uso de <i>Builder</i>	65
Figura 37 Exemplo de uso de <i>Builder</i>	66
Figura 38 Classe <i>Service</i>	66
Figura 39 Exemplos de Tela criada com React JS e componentes (EIS, 2016).....	68
Figura 40 Tela desenvolvida para o Sistema Cultivar com alguns componentes identificados.....	69
Figura 41 Exemplo de Componente React.....	71
Figura 42 Construtor do componente.....	72
Figura 43 Método <i>toggle</i>	72
Figura 44 Método <i>render</i>	73
Figura 45 Tela de Login.....	74
Figura 46 Primeiro passo do <i>wizard</i> de registro do de voluntário..	75
Figura 47 Segundo passo do <i>wizard</i> de registro do de voluntário..	76
Figura 48 Último passo do <i>wizard</i> de registro do de voluntário....	76
Figura 49 Primeiro Login do voluntário.....	77
Figura 50 Tela de Espera pela recomendação da empresa.....	78
Figura 51 Tela de envio do termo de responsabilidade	79
Figura 52 <i>Dashboard</i> voluntário.	80
Figura 53 <i>Dashboard</i> voluntário.	81
Figura 54 Detalhes do evento.....	82
Figura 55 Formulário de avaliação de eventos do Voluntário.....	82
Figura 56 Tela de Espera pela recomendação da empresa.....	84
Figura 57 Tela de detalhes do usuário.....	85
Figura 58 Tela de recomendação da empresa.....	85

Figura 59 <i>Dashboard</i> Responsável na CnE.....	87
Figura 60 Artigo.....	101
Figura 61 Diagrama de classes - classes de modelo e DTOs.	110

LISTA DE TABELAS

Tabela 1	Sinônimos e traduções dos termos de busca	30
Tabela 2	Termos de busca	31
Tabela 3	Funcionalidades atendidas	38
Tabela 4	Requisitos Funcionais	41
Tabela 5	Requisitos não-Funcionais	42
Tabela 6	Comparação dos requisitos da fundamentação teórica com os da solução	43
Tabela 7	Atores e responsabilidades	45
Tabela 8	Rastreamento Casos de Uso - Requisitos Funcionais ...	46
Tabela 9	Detalhamento de caso de uso: UC1 - Alocar Voluntário.	47
Tabela 10	Detalhamento de caso de uso: UC 9 - Recomendar Voluntário.	48
Tabela 11	Detalhamento de caso de uso: UC 10 - Avaliar Treinamento.	49
Tabela 12	Spring Java Annotations (PIVOTAL, 2018b).....	59
Tabela 13	Respostas do questionário de utilidade.....	89
Tabela 14	Respostas do questionário de Funcionalidade	89
Tabela 15	Resultado da pesquisa SUS.....	90
Tabela 16	Respostas dos voluntários	90
Tabela 17	Detalhamento de caso de uso: UC 2 - CRUD Evento... .	103
Tabela 18	Detalhamento de caso de uso: UC 3 - CRUD Projeto.. .	104
Tabela 19	Detalhamento de caso de uso: UC 4 -Avaliar recomendação de voluntário.....	105
Tabela 20	Detalhamento de caso de uso: UC 5 - Emitir relatório de desempenho.	106
Tabela 21	Detalhamento de caso de uso: UC 7 - CRUD escola....	106
Tabela 22	Detalhamento de caso de uso: UC 8 - CRUD empresa..	107
Tabela 23	Detalhamento de caso de uso: UC 11 - CRUD Voluntário.	107
Tabela 24	Detalhamento de caso de uso: UC 14 - CRUD Tipo de evento.....	108

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação	14
NAVE	Núcleo Avançado em Educação.....	15
CnE	Computação na escola	15
ONG	Organização Não-Governamental.....	15
UE	União Europeia	18
RH	Recursos Humanos.....	18
FIFA	Fédération Internationale de Football Association	20
UNESCO	Organização das Nações Unidas para a educação	22
FHC	Fernando Henrique Cardoso	22
ORM	Object-relational mapping	26
JPA	Java Persistence API.....	26
JTA	Java Transaction API.....	26
JS	JavaScript	27
DOM	Document Object Model.....	27
JSX	JavaScript XML	27
SEO	Search Engine Optimization.....	27
GQS	Grupo de Qualidade de Software	39
INE	Departamento de Informática e Estatística	39
UFSC	Universidade Federal de Santa Catarina	39
GoF	Gang of Four	63
DAO	Data Access Object	63
DTO	Data transfer object	64
PO	Persistent Object	64
JVM	Java Virtual Machine	67
SUS	System Usability Scale.....	89
PWA	Progressive Web App	92

SUMÁRIO

1 INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO	14
1.2 OBJETIVOS	16
1.2.1 Objetivo Geral	16
1.2.2 Objetivos Específicos	16
1.3 MÉTODO DE PESQUISA	16
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 VOLUNTARIADO	18
2.1.1 Sobre o perfil Voluntário	20
2.1.2 Voluntariado e educação	22
2.2 APLICAÇÕES WEB	23
2.2.1 Tecnologias	24
2.2.1.1 Servidor	24
2.2.1.1.1 Arquitetura Rest	24
2.2.1.1.2 Spring Project	25
2.2.1.1.3 MySQL	26
2.2.1.2 Cliente	27
2.2.1.2.1 Virtual DOM	27
2.2.1.2.2 ReactJS	27
2.2.1.2.3 Arquitetura Flux	28
3 ESTADO DA ARTE	29
3.1 DEFINIÇÃO DO PROTOCOLO DE MAPEAMENTO	29
3.1.1 Critérios de inclusão/exclusão	29
3.1.2 Critério de qualidade	30
3.1.3 Dados a serem extraídos	30
3.1.4 Strings de busca	30
3.2 EXECUÇÃO DE BUSCA	31
3.2.1 Primeira e Segunda Execução	31
3.2.2 Terceira Execução	32
3.2.3 Ferramentas selecionadas	32
3.2.3.1 Better Impact	32
3.2.3.2 Team Kinetic	33
3.2.3.3 Volistics	34
3.2.3.4 Do-it	34
3.2.3.5 Samaritan	35
3.2.3.6 Volunteer Hub	36
3.3 ANÁLISE DOS RESULTADOS	37

4 PROPOSTA DE SOLUÇÃO	39
4.1 REQUISITOS	39
4.2 CASOS DE USO	43
4.3 PROTOTIPAÇÃO DE TELAS	49
4.3.1 UC 9 - Recomendar Voluntário	50
4.3.2 UC 10 - Avaliar Treinamento	52
4.4 MODELAGEM DO BANCO DE DADOS	54
5 DESENVOLVIMENTO	56
5.1 BACKEND	56
5.1.1 Spring	56
5.1.2 Project Lombok	60
5.1.3 Design Patterns	63
5.1.3.1 Immutable	67
5.2 FRONT-END	67
5.2.1 React JS	67
5.2.2 Interfaces	73
5.2.2.1 Login	74
5.2.2.2 Voluntário	75
5.2.2.3 Responsável da Empresa	83
5.2.2.4 Responsável na CnE	86
6 AVALIAÇÃO DO SISTEMA CULTIVAR.....	88
6.1 TESTE COM USUÁRIOS	88
6.1.1 Utilidade	88
6.1.2 Funcionalidade	89
6.1.3 Usabilidade	89
6.1.4 Pontos Fortes	91
6.2 AMEAÇAS A VALIDADE DA AVALIAÇÃO	91
7 CONCLUSÃO	92
REFERÊNCIAS	93
APÊNDICE A - Artigo	102
APÊNDICE B - Detalhamento dos casos de uso	103
APÊNDICE C - Diagramas de Classe	111
APÊNDICE D - Código	112

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A demanda de profissionais nas áreas de Ciências, Tecnologia, Engenharia e Matemática tem a previsão, nos Estados Unidos, de triplicar no período entre 2011 e 2021 (LANGDON et al., 2011), chegando a 1,1 milhão em 2024, porém somente 45% destas mesmas vagas serão preenchidas (DUBOW, 2013).

Mesmo com essa demanda crescente na área da TI , em 2015 um estudo divulgado (ABDUL-ALIM, 2015) aponta que alunos do ensino médio não seriam capacitados em sala de aula com a finalidade de ocupar essas vagas e, dada esta falta de conhecimento prévio, poucos estudantes na graduação de tecnologia acham que aprender a programar seja fácil (JENKINS, 2002).

Jenkins (2002) afirma que o ensino de programação nos cursos de graduação na área tecnológica, normalmente lecionada nas fases iniciais, é assunto fundamental do curso. Jenkins explica que este momento na vida de muitos estudantes é uma transição difícil e que os materiais mais básicos que os mesmos encontrarão podem ser potencialmente desafiadores em meio a essa mudança.

Muitos estudantes deixam de acreditar no seu sucesso, e com isso perdem motivação e sem a mesma não há aprendizado (JENKINS, 2001). Segundo Allende (2012), no Brasil, 38% dos alunos de nível superior abandonam o curso em seu primeiro ano, deste valor 25,93% não tinha informações o suficiente sobre o curso no início (MELLO et al., 2013).

Uma possível solução seria o ensino de programação e de conceitos de computação já desde as séries iniciais. A implantação do ensino de computação na educação depende basicamente de quatro fatores: o computador, o software educativo, o aluno e da disponibilidade de professor capacitado a usar o computador como ferramenta educacional (VALENTE, 1993)

Para Marques et al. (2011) a prática de programação no ensino médio é interessante e o uso de jogos pode servir de fator motivacional forte, tal qual têm servido em cursos de programação introdutórios em universidades.

Alguns programas governamentais pontuais como o “Um Computador por Aluno”, iniciado no ano de 2007, tem como objetivo incentivar o ensino da informática na educação pública básica entretanto muitas vezes o que se encontra nas escolas são laboratórios abandonados, com-

putadores quebrados e ultrapassados. E mesmo que os laboratórios estejam em boas condições, há falta de profissionais para ocupar o cargo de professor de Informática (LACERDA, 2012).

Segundo o censo sobre o ensino superior do INEP (2016) no ano de 2015 somente 1476 do concluintes, maior taxa dos últimos anos, estariam em cursos de Licenciatura em Computação (informática). E contrapartida, o INEP (2017) afirma que “o País conta com 186,1 mil escolas de educação básica”, o que evidencia a carência de profissionais para o ensino de computação nas escolas.

Iniciativas público-privadas como a escola NAVE¹ e voluntárias como code.org² e Computação na escola(CnE)³ surgiram com a intenção de levar o ensino da computação para crianças e jovens a um maior público. Muitas dessas iniciativas tem contado com a participação de profissionais da área tecnológica como voluntários para apoiar o ensino de programação.

O Centro de Voluntariado de São Paulo define o voluntariado como “a doação de tempo, trabalho e talento para causas de interesse social, a fim de melhorar a qualidade de da comunidade” (GOLDSTEIN, 2007).

A Iniciativa Computação na Escola tem utilizado o envolvimento de voluntários no ensino de programação para crianças (CNE, 2017). No entanto, gerenciar o processo de voluntariado é complexo, especialmente sem o apoio de uma ferramenta. Tipicamente o processo de voluntariado na Iniciativa Computação na Escola envolve:

- A CnE, o voluntário de alguma empresa parceira, a empresa e uma escola alvo.
- Envios de diversos documentos e Termos de Responsabilidade por e-mail/fax.
- Avaliações de perfil ao início do processo e avaliações de desempenho ao fim do mesmo.
- Agendamento das datas de treinamento e monitoria.

Para Tenório (2001) ONG (voluntárias) tem um “estilo próprio de gestão” construído ao longo de sua existência e demonstrando uma resistência aos modelos mais estruturados, apresentando dificuldades na execução de tarefas administrativas.

¹<http://www.oifuturo.org.br/educacao/nave/>

²<http://code.org>

³<http://www.computacaonaescola.ufsc.br>

Assim, espera-se que a existência de uma ferramenta web para gerência do processo de voluntariado venha a contribuir com o aumento do ensino de computação nas escolas e o interesse de crianças e adolescentes na área tecnológica.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver e avaliar uma ferramenta web de gestão do processo de voluntariado para o ensino de computação nas escolas, no contexto da Iniciativa Computação na Escola do Departamento de Informática e Estatística da UFSC.

1.2.2 Objetivos Específicos

1. Analisar da fundamentação teórica sobre gestão de voluntariado e desenvolvimento web.
2. Mapeamento sistemático do estado da arte sobre ferramentas de auxílio ao voluntariado.
3. Modelar e implementar um sistema web para auxiliar no processo de voluntariado na iniciativa Computação na Escola.
4. Implantar e avaliar do sistema desenvolvido.

1.3 MÉTODO DE PESQUISA

Este trabalho será desenvolvido em 6 etapas:

1. Análise da fundamentação teórica (E-1):
Será analisada a fundamentação teórica sobre gestão de voluntariado e desenvolvimento web.
2. Mapeamento sistemático do estado da arte (E-2):
Nesta etapa será realizado Mapeamento sistemático do estado da arte sobre ferramentas de auxílio ao voluntariado.
3. Realizar um estudo comparativo de tecnologias de mercado para desenvolvimento web (E-3)

Nesta etapa será realizada uma análise das tecnologias de desenvolvimento web selecionadas, com base nos requisitos do sistema, as ferramentas mais indicadas para o desenvolvimento.

4. Modelar e implementar um sistema web para auxiliar no processo de voluntariado na iniciativa Computação na Escola (E-4):

Nesta etapa será realizada a modelagem do sistema web e do banco de dados, utilizando a UML e diagrama entidade relacional. Esta etapa possui as seguintes atividades:

- (a) Implementar Módulo administrativo para a CnE.
- (b) Implementar Módulo avaliativo para as escolas participantes.
- (c) Implementar Módulo de autorização para as empresas parceiras.
- (d) Implementar Módulo de cadastro para candidatos a voluntários.

5. Implantação e avaliação (E-5):

Nesta etapa o sistema será implantado no servidor fornecido pela Setic/UFSC e avaliado pelos responsáveis da iniciativa Computação na Escola.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos mais importantes utilizados no desenvolvimento deste trabalho como os aspectos relacionados ao voluntariado, área de aplicação deste trabalho, e aspectos técnicos envolvidos no desenvolvimento do sistema proposto.

2.1 VOLUNTARIADO

Segundo a Lei nº 9.608/1998 do Voluntariado (BRASIL, 1998), o serviço voluntário no Brasil é definido por:

- Não ser remunerado.
- Não gerar vínculo empregatício.
- Não pode ser exigido em contrapartida de algum benefício.
- Ser exercido em entidade pública ou privada, sem fins lucrativos e com objetivos sociais.
- Exigir a assinatura do Termo de Adesão descrevendo objetivo e condições de trabalho e reembolso de despesas.

Marcos e Amador (2014) salientam que o voluntariado mesmo não sendo uma prática recente, somente em 2011 entrou nas pautas políticas da UE. A gestão do voluntariado também é nova nos estudos da administração, tornando-se um desafio para o setores de RH, habituados com as relações contratuais típicas da gestão mais clássica (VIDAL et al., 2007).

O ciclo de gestão do voluntariado como apresentado na figura 1 pode ser dividido em 6 fases como objetivos bem específicos (MARCOS; AMADOR, 2014):

1. Preparação - Contém a criação de um plano de voluntariado (explicado abaixo) e de uma ferramenta de gestão que permita a replicação de processos na organização.
2. Definição - Esta fase do ciclo se constitui da criação de perfis sociodemográficos, escolares e profissionais desejáveis para missão da empresa e da seleção de voluntários com tais perfis.

3. Acolhimento - Este é o momento onde há o comprometimento, com os deveres e direitos, do voluntário com a organização e vice-versa. Também é o momento de esclarecimento das dúvidas do voluntário sobre suas tarefas na organização.
4. Desenvolvimento - Este é o momento de formação do voluntário para as tarefas que lhe serão designadas, uma forte comunicação com o líder e demais integrantes do grupo de trabalho. É um acompanhamento para analisar o grau de satisfação das expectativas e identificação de necessidades.
5. Reconhecimento - Esta fatia do ciclo de gestão do voluntariado consiste de ações que visam reconhecer o trabalho do voluntário no dia a dia.
6. Desvinculação - Implica em um processo de orientação ao voluntário de saída e a manutenção da relação de mesmo com a organização após o período de voluntariado.

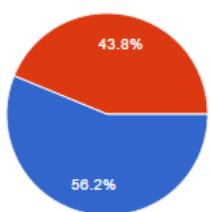


Figura 1 – As fases do ciclo de gestão do voluntariado (MARCOS; AMADOR, 2014).

2.1.1 Sobre o perfil Voluntário

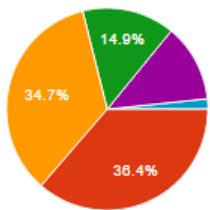
O voluntário é, em senso comum, todo aquele que por valores solidários e sociais, doa seu tempo, seu trabalho e seus talentos espontaneamente e sem remuneração, para causas de interesse comunitário, humanitário e social.(YAZBEK, 2002).

Segundo MENDES (2015) os voluntários no Brasil, durante o período da copa do mundo da FIFA de 2014, em sua maioria são mulheres, entre 18 e 35 anos, com boa estabilidade financeira, possivelmente estudantes de nível superior ou pós-graduação, sem filhos como pode-se inferir pelos gráficos abaixo:



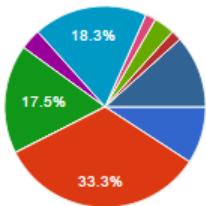
Feminino	68	56.2%
Masculino	53	43.8%
Prefere não responder	0	0%

Figura 2 – Gênero dos voluntários (MENDES, 2015).



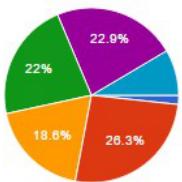
Menor que 18 anos	0	0%
Entre 18 e 25 anos	44	36.4%
Entre 26 e 35 anos	42	34.7%
Entre 36 e 45 anos	18	14.9%
Entre 46 e 60 anos	15	12.4%
Maior que 60 anos	2	1.7%
Prefere não responder	0	0%

Figura 3 – Idade dos voluntários (MENDES, 2015).



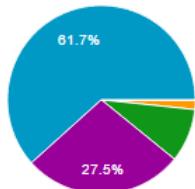
Autônomo	11	9.2%
Estudante	40	33.3%
Empresário	0	0%
Servidor Público	21	17.5%
Profissional Liberal	4	3.3%
Empregado na iniciativa privada	22	18.3%
Empregado em ONG's	2	1.7%
Aposentado	4	3.3%
Prefere não responder	2	1.7%
Outros	14	11.7%

Figura 4 – Profissão dos voluntários (MENDES, 2015).



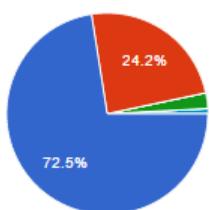
Menos que 1 salário mínimo (- R\$ 788,00)	2	1.7%
1 a 3 salários mínimos (R\$ 788,00 a 2.364,00)	31	25.8%
3 a 5 salários mínimos (R\$ 2.364,00 a 3.940,00)	22	18.3%
5 a 10 salários mínimos (R\$ 3.940,00 a 7.880,00)	26	21.7%
Mais que 10 salários mínimos (+ R\$ 7.880,00)	27	22.5%
Prefere não responder	10	8.3%

Figura 5 – Renda dos voluntários (MENDES, 2015).



Ensino Fundamental Incompleto	0	0%
Ensino Fundamental Completo	0	0%
Ensino Médio Incompleto	2	2.1%
Ensino Médio Completo	11	9.2%
Nível Superior Incompleto	33	27.5%
Nível Superior Completo ou mais	74	61.7%
Não se Aplica	0	0%
Prefere não responder	0	0%

Figura 6 – Escolaridade dos voluntários (MENDES, 2015).



Solteiro	87	72.5%
Casado	29	24.2%
Viúvo	0	0%
Divorciado	3	2.5%
Prefere não responder	0	0%
Outros	1	0.8%

Figura 7 – Estado Civil dos voluntários (MENDES, 2015).

Segundo Centro de Voluntariado de São Paulo (2016), 77% dos voluntários se sentem completamente satisfeitos com a atividade desempenhada, 87% estão motivados a continuar no voluntariado sendo as principais motivações, solidariedade (67%), melhorar o mundo fazendo a diferença (32%) e religiosidade (22%).

2.1.2 Voluntariado e educação

Na área da educação o voluntariado tem repercutido com inúmeros projetos nos diversos níveis de ensino tais como Universidade Solidária, Educação Solidária, Alfabetização Solidária, Projeto Amigos da Escola, entre outros. Estas iniciativas são financiadas por diversas fontes como UNESCO, empresas privadas, ONG's e entidades governamentais (SILVA, 2009).

De todas os projetos supracitados, o de maior destaque foi o Projeto Amigos da Escola – Todos pela Educação, lançado pela rede Globo durante o segundo mandato do governo FHC. Visa fortalecer o ensino público, sendo um dos empreendimentos sociais mais importantes do país (GLOBO, 2012)(SILVA, 2009).



Figura 8 – Logo Amigos da Escola – Todos pela Educação (DIÁRIO CATARINENSE, 2012).

2.2 APLICAÇÕES WEB

Mora (2002) define aplicações web como sendo uma categoria especial de cliente-servidor. Caivano e Villoria (2009) complementam ressaltando que aplicações web não são mais do que ferramentas acessíveis por meio de um navegador web e acesso à rede, internet ou intranet, na web 2.0.

Tanenbaum e Steen (2007) descreve a arquitetura cliente-servidor como um sistema distribuído dividido em dois processos, que se comunicam por meio de um protocolo definido, o HTTP no caso de aplicações web (MORA, 2002), o servidor que implementa um serviço específico e o cliente requisita o serviço, e espera o retorno, do servidor (TANENBAUM; STEEN, 2007) como exemplificado na figura 9.

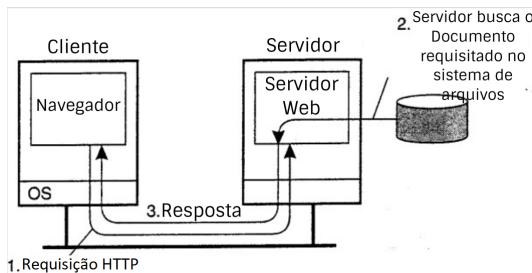


Figura 9 – Típica organização cliente-servidor de um web site/sistema (TANENBAUM; STEEN, 2007).

2.2.1 Tecnologias

Na seção abaixo são apresentadas as tecnologias utilizadas para o servidor e cliente no contexto deste trabalho, são elas ReST e Spring Project para o servidor e no lado do cliente ReactJS e Flux.

2.2.1.1 Servidor

2.2.1.1.1 Arquitetura Rest

Representational State Transfer, ReST, foi concebido por Roy Thomas Fielding na sua tese de doutorado em 2000, como uma arquitetura para sistemas de hipermídia distribuídos. Fielding (2000) elenca algumas características, restrições, que compõem a arquitetura ReST:

1. Arquitetura cliente-servidor - Para Fielding (2000) a arquitetura cliente-servidor é a resposta para uma requisição na qual servidor envia os dados em um formato fixo (representação). Separando a interface do armazenamento dos dados, assim, melhorando a portabilidade e escalabilidade.
2. Stateless - O servidor não detém conhecimento nenhum sobre os clientes a ele conectado, assim como conteúdo dos dados e como são executadas as funções do cliente (BERKENBROCK, 2005). Assim cada interação do cliente com o servidor deve conter todas as informações necessárias para a execução do pedido (FIELDING, 2000).

3. Cache - A utilização de cache, uma restrição facultativa, pode reduzir as interações entre cliente e servidor aumentando a eficiência e escalabilidade do sistema (FIELDING, 2000). Berkenbrock (2005) realça que a adoção de cache reduz a confiabilidade nos dados apresentados no cliente uma vez que podem não corresponder a versão mais recente no servidor.
4. Interface uniforme - Para Fielding (2000) a principal distinção do ReST para outras arquiteturas baseadas em rede. A interface uniforme degrada a eficiência por ser aplicado o princípio de generalidade da engenharia de software.
5. Sistemas em camadas - Fielding (2000) explicita que a divisão do sistema em camadas hierárquicas melhoram a escalabilidade, performance e segurança do sistema, por um melhor balanceamento da carga de processamento.
6. Código sob demanda - Outra restrição facultativa permite a funcionalidade opcional de estender código para execução no cliente por intermédio de download e instalação de applets e scripts (FIELDING, 2000).

2.2.1.1.2 Spring Project

O Spring Project ou Spring IO é uma plataforma modular do ecossistema Java mantida pela Pivotal (2018a) para prover a infraestrutura para as aplicações, desde segurança, aplicações web até big data. Os Módulos usados neste projeto são:

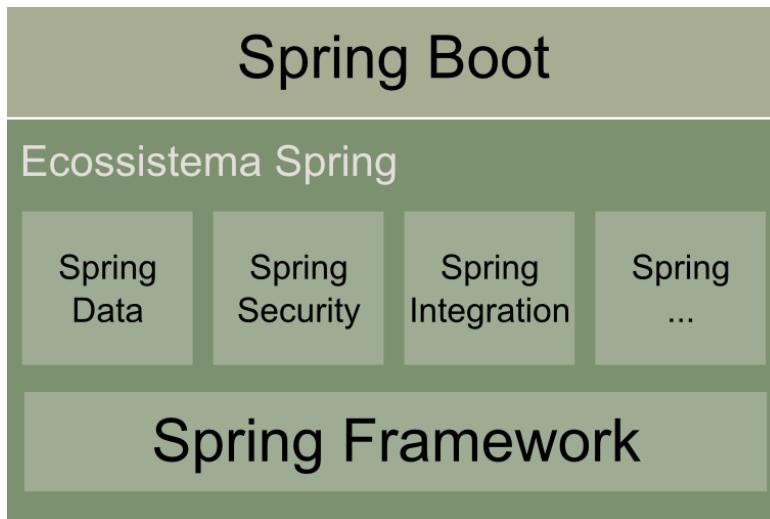


Figura 10 – Ecossistema da plataforma Spring IO (DEVMIDIA, 2015).

- Spring Boot - Facilita a criação de projetos Spring. Contém injecção de dependências, servidor embarcado e configurações mais amigáveis.
- Spring Data - Abstração de ORM construída em cima da JPA e da JTA
- Spring Security - Provê mecanismos de autenticação e autorização que podem ser facilmente estendidos para atender as necessidades da aplicação.

2.2.1.1.3 MySQL

Segundo a Oracle Corporation (2018), o MySQL é o banco de dados de *Open source* mais conhecido no mundo. Com desempenho, confiabilidade e facilidade de uso comprovados por gigantes da Web, incluindo Facebook, Twitter, YouTube, sendo a principal opção de banco de dados para aplicação Web. Além disso, é uma opção grande como banco de dados integrado (ORACLE CORPORATION, 2018).

2.2.1.2 Cliente

2.2.1.2.1 Virtual DOM

A virtual DOM, é uma cópia em memória da DOM renderizada, sendo um recurso implementado por diversos frameworks e bibliotecas JS para otimização de re-renderização da interface. A virtual DOM ao sofrer uma alteração de estado é comparada com a DOM real e esta por sua vez é alterada somente no ponto necessário da árvore (TECH, 2017) como apresentado na figura 11.

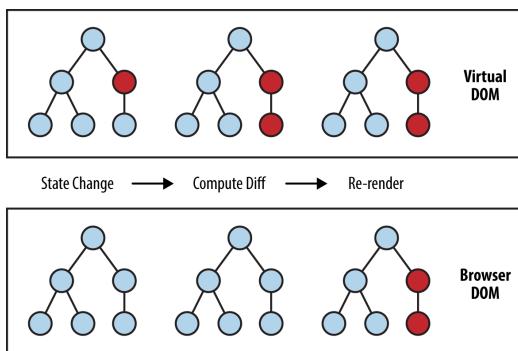


Figura 11 – Virtual Dom calculando a diferença apóis alteração de estado para otimização da nova renderização (EISENMAN, 2015).

2.2.1.2.2 ReactJS

ReactJS, ou simplesmente React, é uma biblioteca JS, criada pelo Facebook (2018), para a construção e gerenciamento de estado de interface. Utilizando-se de uma virtual DOM para otimização de renderização durante mudança de estado da aplicação (TECH, 2017).

O React se propõem a mudar a mentalidade ao construir a interface web, quebrando-a em componentes flexíveis e de fácil manutenção, e utiliza-se da sintaxe, opcional, JSX para uma maior legibilidade do código(HUNT, 2013). Outras vantagens do React é a melhor performance quando há alteração do estado visto que somente o nó da DOM que sofreu a alteração será novamente renderizado. A possibilidade de renderização em *server side* o que melhorar o SEO (TECH, 2017)(HUNT,

2013).

2.2.1.2.3 Arquitetura Flux

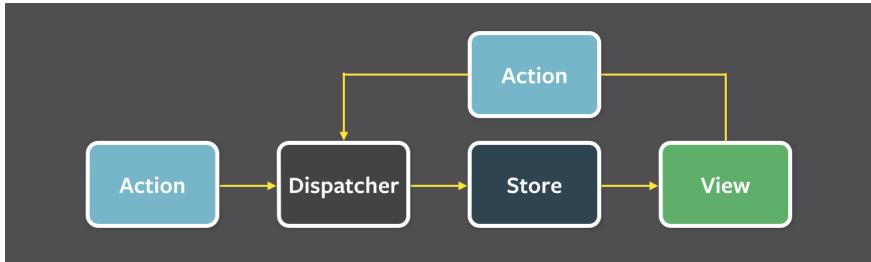


Figura 12 – Fluxo de dados proposto pela arquitetura Flux (FACEBOOK, 2015).

Flux é a proposta do Facebook (2015) para o fluxo dos dados dentro das aplicações construídas com ReactJS, trabalhando com um fluxo unidirecional como ilustrado na figura 12, dividido em quatro estruturas diferentes que são:

- *Action* - Capturam as maneiras pelas quais qualquer coisa pode interagir com a aplicação. São objetos simples que possuem um campo "type" e alguns dados.
- *Dispatcher* - Toda *action* é enviada para os *stores* registrados no *dispatcher*.
- *Store* - Após os *stores* serem atualizados em resposta a *action*, eles emitem um evento de alteração.
- *View/Controller-View* - Os dados dos *stores* são apresentados na *view* e esta quando esta recebe o sinal de um evento de alteração, realiza uma nova renderização utilizando os novos dados do *store*.

3 ESTADO DA ARTE

Este capítulo apresenta o estado da arte das ferramentas para gestão de processos de voluntariado. A análise do estado da arte é realizada seguindo as principais etapas do método de mapeamento sistemático de literatura definido por Petersen K. (2008), adaptadas para uma pesquisa de trabalho de conclusão de curso de graduação. Esse método é realizado definindo o mapeamento da literatura disponível, executando a busca e fazendo a extração dos resultados, bem como sua análise.

3.1 DEFINIÇÃO DO PROTOCOLO DE MAPEAMENTO

Este mapeamento tem como objetivo realizar uma pesquisa de ferramentas/sistemas web para o gerenciamento de processos de voluntariado. Com essa premissa foi definida a seguinte pergunta de pesquisa: “Existem sistemas informatizados para a gestão de processos para programas de voluntariado e quais seus benefícios?”.

A base de pesquisa utilizada foi o Google Scholar¹, pelo acesso a diversas bases de dados. As buscas foram realizadas com os termos em português e suas traduções em inglês. Os resultados analisados da busca foram apenas os que estavam nestes idiomas, e dentro do período de 2010 a 2017.

3.1.1 Critérios de inclusão/exclusão

O material escolhido deve obedecer ao seguintes critérios para serem considerados proveitosos no mapeamento:

- O material encontrado deve conter *features* que os sistemas existentes apresentam.
- O material encontrado deve ser focado em processos do terceiro setor, iniciativas de responsabilidade social e voluntariado.

¹<https://scholar.google.com.br>

3.1.2 Critério de qualidade

O critério de qualidade foi definido como:

- Deve preferencialmente conter necessidades que o sistema deveria atender.

3.1.3 Dados a serem extraídos

Para responder a pergunta de pesquisa com base nos trabalhos selecionados são extraídos os seguintes dados:

- Tecnologias utilizadas no desenvolvimento das ferramentas.
- Funcionalidades implementadas.
- Limitações encontradas.

3.1.4 Strings de busca

Os termos utilizados na busca são derivados da pergunta de pesquisa e são apresentados no Quadro 1:

Tabela 1 – Sinônimos e traduções dos termos de busca

Termo	Sinônimo	Tradução
"Sistema informatizado"	"Sistema web", "Aplicação"	"Computerized system", "Web system", "Application"
"Gestão de processos"	"Gerenciamento"	"Process Management", "Management"
"Empresa de Responsabilidade Social"	"Voluntariado", "Terceiro Setor"	"Corporate Social Responsibility/ CSR", "voluntary", "voluntary sector/community sector"
"Benefícios"		"Benefits"

As Strings de busca derivadas aparecem no Quadro 2, bem como o número total de resultados retornados:

Tabela 2 – Termos de busca

Termos de busca	Resultados retornados
("Sistema informatizado"OR "Sistema web"OR "Aplicação"OR "Computerized system"OR "Web system"OR "Application") AND ("Gestão de processos"OR "Gerenciamento"OR "Process Management"OR "Management") AND ("Empresa de Responsabilidade Social"OR "Voluntariado"OR "Terceiro Setor"OR "Corporate Social Responsibility/ CSR"OR "voluntary"OR "voluntary sector/community sector") AND ("Benefícios"OR "Benefits") PERÍODO (2010 a 2017)	13
("Web system"OR "Application") AND "Process Management" AND ("voluntary"OR "voluntary sector/community sector") AND "Benefits"PERÍODO (2010 a 2017)	23
"volunteer management web system"	995.000

3.2 EXECUÇÃO DE BUSCA

As buscas foram feitas em dezembro de 2017. No total foram necessárias 3 iterações de busca, apresentadas na sequência.

3.2.1 Primeira e Segunda Execução

A primeira execução retornou um total de 13 registros, nenhum, contudo atendeu aos critérios de inclusão, e um reforçou as dificuldades de gestão neste setor.

A segunda execução, usando somente alguns termos de busca na língua inglesa, elencados na segunda linha da tabela 2 também se mostrou escassa.

3.2.2 Terceira Execução

Em razão do baixo número de resultados encontrados, a busca por ferramentas correlatas também foi realizada utilizando o Google. Os critérios para busca foram considerados os mesmos.

Após uma pesquisa sobre ferramentas de gerência de voluntariado, foram elencadas para análise 6 ferramentas existentes no mercado (KNOW HOW NON PROFIT(ORG.), 2018), são elas:

1. Better Impact ².
2. Team Kinetic ³.
3. Volistics ⁴.
4. Do-it ⁵ ⁶ ⁷.
5. Samaritan ⁸.
6. Volunteer Hub ⁹.

3.2.3 Ferramentas selecionadas

Cada uma das ferramentas selecionadas é brevemente apresentada a seguir:

3.2.3.1 Better Impact

Better Impact é uma ferramenta web paga para gestão de voluntariado. A ferramenta tem a preocupação de ser genérica permitindo que o gestor da iniciativa voluntária escolher entre templates já existentes para os mais diversos e comuns tipos de iniciativas voluntárias, ou um template branco com menos customização previamente já realizada.

²<https://www.betterimpact.co.uk/>

³<https://teamkinetic.co.uk/>

⁴<https://www.volistics.com/>

⁵<https://do-it.org/>

⁶<https://doit.life/>

⁷<http://doittrust.org/>

⁸<https://samaritan.com/>

⁹<https://www.volunteerhub.com/>

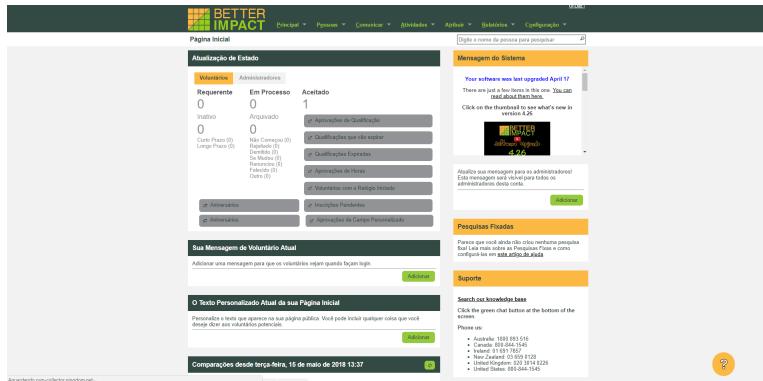


Figura 13 – Tela inicial do sistema Better Impact (2018).

3.2.3.2 Team Kinetic

Team Kinetic é uma ferramenta web paga com uma versão gratuita até 9 voluntários, a partir de 10 voluntários para gerir os preços variam entre £20pm à £150pm. A principal preocupação dessa ferramenta é com o recrutamento e a retenção de voluntários. Permitindo o envio de e-mails e mensagens de texto(SMS). A versão enterprise(£150pm) possibilita ao gestor elencar um perfil de usuário que pode visualizar as oportunidades.

Search Volunteers

Name or Email:

Quick Finds:

ACTIVE VOLUNTEERS RETURNED

VOLUNTER	DATE REG	VOL HRS	PRO HRS	CRIMINAL CHECK	STATUS
Total Hours					

Click to confirm block: Block Selected Volunteers Unblock Selected Volunteers

ADD Selected Volunteers To:

Figura 14 – Listagem de Voluntários (TEAM KINECTIC, 2018).

3.2.3.3 Volistics

Outra ferramenta web existente no mercado, Volistics, é customizável para a necessidade da iniciativa. Inicialmente o gestor utiliza uma versão de testes com alguns recursos limitados por 60 dias, podendo evoluir a qualquer momento para versão paga ou instantaneamente após os 60 dias. A Volistics conta com suporte de 13 horas/dia de segunda a sexta-feira.

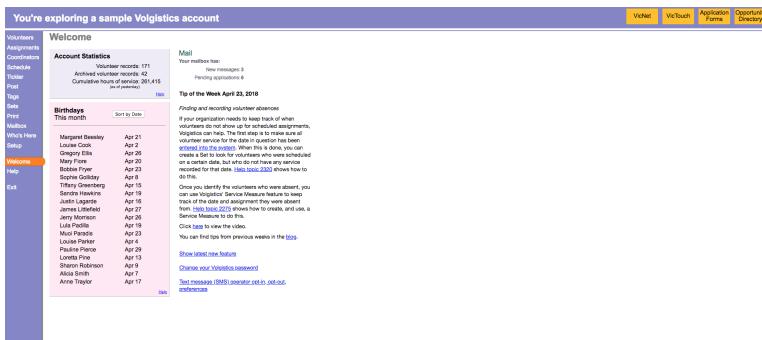


Figura 15 – Tela de boas vindas do Volistics (2018).

3.2.3.4 Do-it

A Do-it, é uma organização de caridade que disponibiliza uma plataforma web gratuita para gestão de iniciativas voluntárias. Porém diferentemente dos sistemas/plataformas anteriores há a necessidade de uma verificação dos dados da organização antes da divulgação das vagas de voluntariado. Ela não contém uma possibilidade alta de customização, somente elencar um perfil de voluntário que possa se candidatar a vaga.

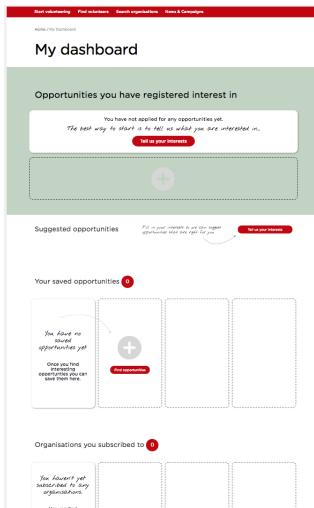


Figura 16 – Dashboard do voluntário (DO-IT, 2018).

3.2.3.5 Samaritan

A ferramenta Samaritan, é uma plataforma completa para gestão de voluntários. Contando com todas as funcionalidades dos outros sistemas supracitados e outras funções tais como verificação de antecedentes criminais e sistema para treinamento dos voluntários. O preço inicial da plataforma é de US\$2500 anuais, entretanto, para organizações com orçamento anual menor de US\$400 mil ou membro da Association of Leaders in Volunteer Engagement (AL!VE) tem direito a uma versão gratuita com os recursos mais básicos da plataforma.



Figura 17 – Site da Samaritan Technologies (2018).

3.2.3.6 Volunteer Hub

O Volunteer Hub, está no mercado desde 1996, sendo o sistema mais antigo dos estudados. Possui integração com diversos CRMs, ferramentas de angariação de fundos, deduplicação. Diferentemente das demais ferramentas não existe uma versão gratuita para teste, sendo o plano de menor valor de US\$150 mensais e US\$595 de instalação e configuração inicial.



Figura 18 – Site da Volunteer Hub (2018).

3.3 ANÁLISE DOS RESULTADOS

A Know How Non Profit(Org.) (2018) lista algumas funcionalidades que devem ser avaliadas pelo gestor na escolha de uma ferramenta para o fim de gerir uma organização sem fins lucrativos, dentre elas:

1. A inscrição de novos voluntários.
2. Coleta de dados dos voluntários.
3. Cadastro de oportunidades de voluntariado.
4. Cadastro e manutenção de Eventos voluntários.
5. Relatórios e feedbacks.
6. Gestão de comunicação com os envolvidos.
7. Incentivos ao voluntário.
8. Treinamento e cursos.

Para analisar as ferramentas encontradas, foram definidos alguns requisitos iniciais, tomando por base aqueles citados pela Know How Non Profit(Org.) (2018), requisitos funcionais, e por meio de entrevista com um dos coordenadores da CnE, requisitos não funcionais, são elas:

- RF1 - O sistema deve permitir o cadastro dos voluntários.
- RF2 - O sistema deve permitir o cadastro dos beneficiados.
- RF3 - O sistema deve permitir o cadastro de eventos de voluntariado.
- RF4 - O sistema deve permitir a alocação de voluntários.
- RF5 - O sistema deve permitir o gerenciamento de treinamentos.
- RF6 - O sistema deve permitir a avaliação dos voluntários.
- RF7 - O sistema deve permitir a avaliação dos treinamentos.
- RNF1 - O sistema deve ser Integrado ao site da CnE.
- RNF2 - Gratuito e Open-Source.
- RNF3 - Aplicação web.

A Tabela 3 apresenta um comparativo entre as ferramentas analisadas, de acordo com os requisitos funcionais de mais alto nível, identificados para a ferramenta desenvolvida neste trabalho. Para avaliar o grau de atendimento de cada requisito, foram usados os seguintes conceitos: T (Totalmente Atendido), P (Parcialmente Atendido), N (Não Atendido), NT (Não Testado).

Tabela 3 – Funcionalidades atendidas

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RNF1	RNF2	RNF3
Better Impact	T	N	N	T	P	T	N	N	N	T
Team Kinetic	N	P	T	T	N	T	N	N	P	T
Volistics	T	T	P	T	N	T	N	N	N	T
Do-it	T	N	T	P	N	N	N	N	P	T
Samaritan	NT	N	P	T						
Volunteer Hub	NT	N	N	T						

Como apresentado na Tabela 3 as ferramentas "Samaritan" e "Volunteer Hub" não poderam ser testadas devido ao custo de licença além do orçamento disponível para o projeto.

Por meio de uma breve pesquisa nos sites destas ferramentas, foi constatado que ambas ferramentas não atenderiam as funcionalidades RF5 e RF7, entretanto elas contemplam as demais funcionalidades não testadas.

Dentre as demais plataformas elencadas as mais próximas do ideal são "Team Kinetic" e "Volistics" atendendo a maior parte das funcionalidades em sua totalidade. Há possibilidade de que a ferramenta "Better Impact" tenha sido prejudicada na pesquisa devido a escolha de um *template* de testes, voltado para gerenciamento escolar.

Devido a nenhum dos sistemas serem de código aberto não foi possível encontrar informações sobre tecnologias e método de desenvolvimento utilizados, deixando a análise carente de informações para este tópico.

Existem diversas aplicações para o auxílio do controle de voluntários. Porém nenhuma das ferramentas supracitadas pode atender a todos os requisitos elencados para o projeto, resultando na necessidade da implementação de um sistema específico para a Computação na Escola.

4 PROPOSTA DE SOLUÇÃO

Este capítulo apresenta o processo de elaboração da solução para o presente trabalho. Primeiramente é abordada a elicitação dos requisitos funcionais e não-funcionais, para então ser apresentado o fluxo das atividades da solução. Também é apresentado o diagrama de casos de uso e seus respectivos detalhamentos, sendo que, para cada detalhamento de caso de uso, são apresentados alguns protótipos de telas. Após a definição dos requisitos, é apresentada uma pesquisa comparativa para escolha das tecnologias da solução implementada.

A solução desenvolvida tem como objetivo apoiar a iniciativa Computação na Escola do GQS/INE/UFSC durante o processo de seleção dos tutores voluntários e avaliações dos mesmos, sendo um sistema para controle de tutores voluntários (CuLTiVAR).

Para o desenvolvimento da solução deste trabalho foi adotado a linguagem Java para criação do *backend* devido a grande probabilidade de encontrar programadores dentro da universidade com conhecimento prévio na linguagem, visto ser a linguagem ensinada, caso haja necessidade de implementações de novas funcionalidade no futuro.

4.1 REQUISITOS

Visando o desenvolvimento de uma solução que atendesse às necessidades da iniciativa CnE e expectativas quanto às funcionalidades contempladas pela mesma, o levantamento dos requisitos foi realizado em 3 etapas.

Em um primeiro momento por meio de entrevista com o coordenador do projeto foi criado um esboço inicial dos requisitos tendo como base o fluxo, que automatizado pela solução proposta neste trabalho, apresentado na figura 19 fornecida por um dos coordenadores da CnE. Após isso foi realizado o mapeamento bibliográfico onde foram identificados novos requisitos e lapidados os requisitos já existentes usando como base os itens citados pela Know How Non Profit(Org.) (2018).

Em um terceiro momento, com telas já prototipadas utilizando os requisitos elencados anteriormente, foram entrevistados um ex-voluntário e uma pessoa responsável na CnE para confirmação de requisitos dos requisitos.

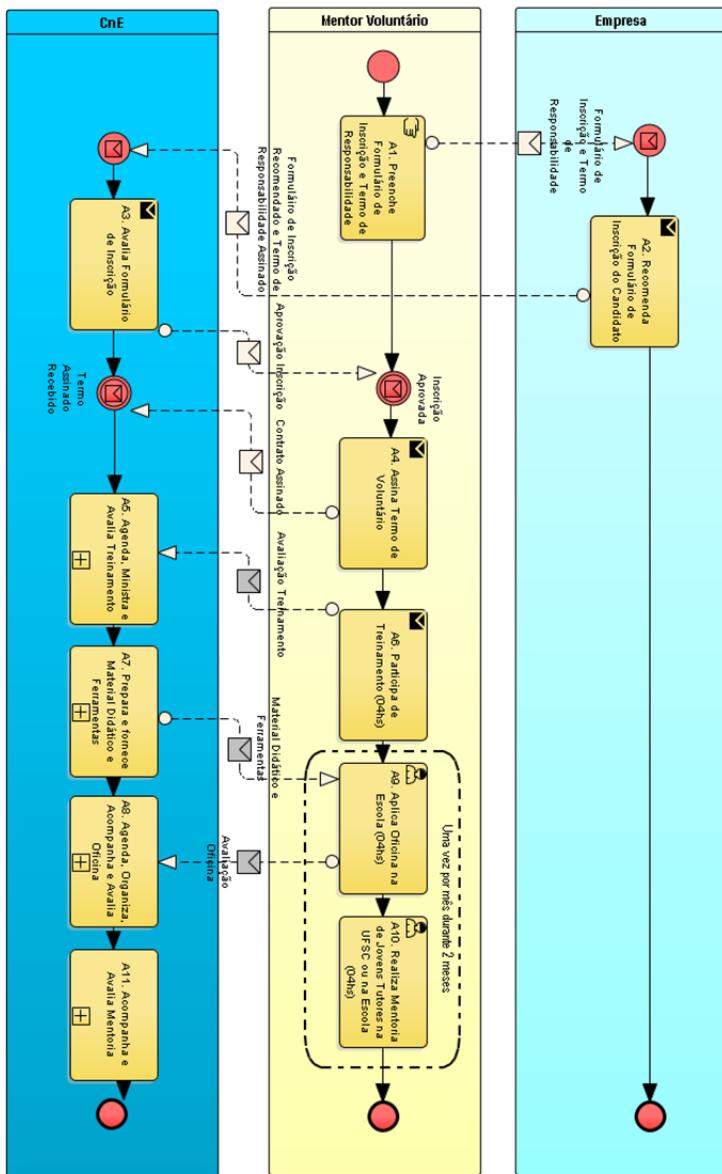


Figura 19 – Diagrama do processo de voluntariado (fornecido por um dos coordenadores da CnE).

Como resultado deste processo foram elicitados os seguintes requisitos vistos nas tabelas 4 e 5:

Tabela 4 – Requisitos Funcionais

Identificação	Descrição
RF 1	O sistema deve permitir ao voluntário se cadastrar da plataforma.
RF 2	O sistema deve permitir ao voluntário enviar a documentação necessária, termos de responsabilidade e de voluntariado.
RF 3	O sistema deve permitir ao responsável pela empresa recomendar a inscrição do voluntário.
RF 4	O sistema deve permitir ao responsável na CnE avaliar a inscrição recomendada.
RF 5	O sistema deve permitir ao responsável na CnE disponibilizar materiais para treinamento via documentos digitais.
RF 6	O sistema deve permitir ao responsável na CnE fornecer materiais didáticos para as oficinas.
RF 7	O sistema deve permitir ao responsável na CnE agendar eventos (Treinamentos, Oficinas, Reuniões, etc...).
RF 8	O sistema deve permitir ao Voluntário avaliar o treinamento recebido.
RF 9	O sistema deve permitir ao Voluntário e ao responsável da escola avaliar a mentoria.
RF 10	O sistema deve permitir ao responsável na CnE visualizar relatórios sobre os treinamentos.
RF 11	O sistema deve permitir ao responsável na CnE visualizar relatórios sobre os voluntários e mentorias.
RF 12	O sistema deve avisar os usuários envolvidos nos eventos.
RF 13	O sistema deve permitir ao responsável na CnE cadastrar novas empresas parceiras.
RF 14	O sistema deve permitir ao responsável na CnE cadastrar novas escolas beneficiadas.
RF 15	O sistema deve permitir ao responsável na CnE cadastrar novos locais para eventos.
RF 16	O sistema deve permitir ao Usuário alterar seu dados cadastrais.
RF 17	O Sistema deve permitir ao Responsável na CnE Cadastrar novos projetos

Continua na próxima página

Tabela 4 – *Continuação da página anterior*

Identificação	Descrição
RF 18	O Sistema deve permitir ao Responsável na CnE Cadastrar novos tipos de evento

Tabela 5 – Requisitos não-Funcionais

Identificação	Descrição
RNF 1	O sistema deve ser uma aplicação web
RNF 2	O sistema deve ser implementado na linguagem Java
RNF 3	O sistema deve ter integração com o Portal da CnE via banco de dado MySQL

Conforme abordado anteriormente, uma parte dos requisitos da solução deste trabalho foram criados com base na fundamentação teórica, algumas recomendações realizadas pela Know How Non Profit(Org.) (2018) sobre sistemas de gerenciamento de voluntariado:

- Rec 1 - Os voluntários se cadastram via website.
- Rec 2 - Definir questionário de cadastro.
- Rec 3 - Definir quais informações devem ser coletadas após o cadastro (CV e qualificações).
- Rec 4 - O administrador do sistema cadastrá todas as oportunidades de voluntariado.
- Rec 5 - Você precisará de ferramentas para comunicação com uma grande quantidade de voluntários?
- Rec 6 - Você gostaria que os voluntários gravassem suas próprias horas?
- Rec 7 - Você quer registrar o feedback do voluntário?
- Rec 8 - Deseja enviar e-mails automatizados ou mensagens de texto para voluntários (quando eles se inscrevem, participam de uma oportunidade, um dia antes de sua oportunidade acontecer, etc.)

- Rec 9 - Você quer anunciar cursos de treinamento para seus voluntários?

A tabela 6 apresenta uma comparação dos requisitos da fundamentação teórica com os da solução.

Tabela 6 – Comparação dos requisitos da fundamentação teórica com os da solução

Recomendação	Requisitos
Rec 1	RF 1
Rec 2	RF 1
Rec 3	RF 2
Rec 4	RF 7, RF 17, RF 18
Rec 5	RF 12
Rec 6	Não atendido
Rec 7	RF 8
Rec 8	RF 12
Rec 9	RF 5, RF 7, RF 8 e RF 12

O demais requisitos foram elencados por meio da análise do fluxo apresentado na imagem 19 e pelas entrevistas com participantes da iniciativa CnE.

4.2 CASOS DE USO

Por meio da análise do fluxo apresentado na imagem 19 foram identificadas quatro entidades externas que irão interagir com a solução a ser desenvolvida, ou seja, os atores do diagrama de casos de uso. São estes: Responsável na CnE, Responsável pela empresa, Responsável da escola e Voluntário.

A imagem 20 apresenta o diagrama de casos de uso referente ao sistema CuLTiVAR, na tabela 7 são apresentados de forma descritiva o que cada ator representa no mundo real e cada uma das funcionalidades acessivas a ele no sistema.

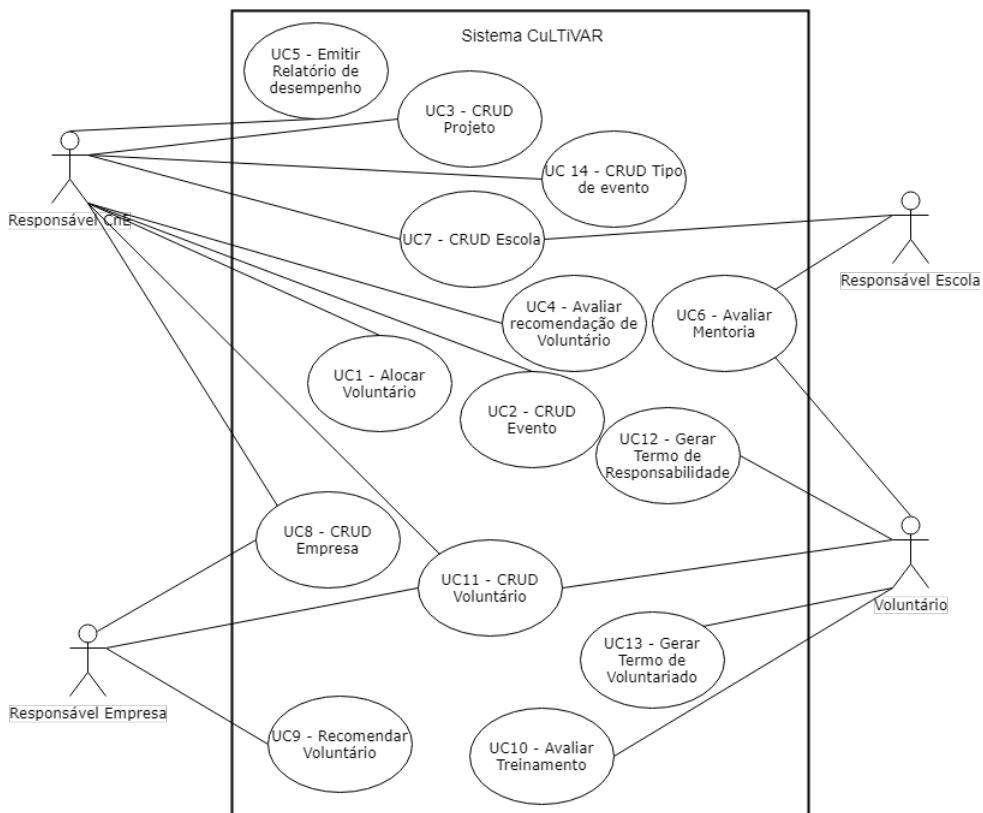


Figura 20 – Diagrama de casos de uso.

Tabela 7 – Atores e responsabilidades

Ator	Mundo Real	Casos de Uso Realizados
Responsável na CnE	Representa alguém da gestão da CnE	UC 1 - Alocar Voluntário. UC 2 - CRUD Evento. UC 3 - CRUD Projeto UC 4 - Avaliar recomendação de voluntário. UC 5 - Emitir relatório de desempenho UC 7 - CRUD escola UC 8 - CRUD empresa UC 11 - CRUD Voluntário UC 14 - CRUD Tipo de evento
Responsável pela Empresa	Representa a pessoa responsável na empresa parceira pela comunicação com a CnE	UC 8 - CRUD empresa UC 9 - Recomendar voluntário UC 11 - CRUD Voluntário
Responsável da Escola	Representa a pessoa responsável na escola parceira pela comunicação com a CnE	UC 6 - Avaliar mentoria UC 7 - CRUD escola
Voluntário	Representa a pessoa realizará a mentoria em nome da CnE	UC 6 - Avaliar mentoria UC 10 - Avaliar treinamento UC 11 - CRUD Voluntário UC 12 - Gerar termo de responsabilidade UC 13 - Gerar termo de Voluntariado

Cada casos de uso está relacionado com os requisitos funcionais por ele atendido na tabela 8 abaixo:

Tabela 8 – Rastreamento Casos de Uso - Requisitos Funcionais

Casos de Uso	Requisitos
UC 1	RF 7, RF 12
UC 2	RF 5, RF 7, RF 12, RF15
UC 3	RF 17
UC 4	RF 4
UC 5	RF 10, RF 11
UC 6	RF 9
UC 7	RF 14, RF 16
UC 8	RF 13, RF 16
UC 9	RF 3
UC 10	RF 8
UC 11	RF 1, RF 16
UC 12	RF 2
UC 13	RF 2
UC 14	RF 5, RF 6, RF 18

Como visto na tabela 8 é possível atender todos os requisitos elicitados com os caso de usos apresentados na imagem 20.

As tabelas 24, 10 e 11 abaixo apresentam, como exemplo, o detalhamento dos casos de uso “Alocar Voluntário”, “Recomendar Voluntário” e “Avaliar Treinamento” respectivamente.

Tabela 9 – Detalhamento de caso de uso: UC1 - Alocar Voluntário.

Nome do Caso de Uso	UC1 - Alocar Voluntário.
Pré-Condições	Existir pelo menos um evento e um voluntário.
Atores Envolvidos	Responsável na CnE.
Resumo	Aborda os passos necessários para o Responsável na CnE alojar um voluntário em algum evento (Treinamento, Reunião, mentoria, etc...).
Cenário principal:	
1. O Responsável CnE seleciona o evento em que quer adicionar o voluntário.	
2. O sistema apresenta os detalhes do evento.	
3. O responsável CnE clica em “Adicionar Voluntário”.	
4. O Sistema Mostra a lista de Voluntários aptos para o evento.	
5. O Responsável CnE seleciona o Voluntário a ser alocado.	
6. O sistema atualiza a alocação do Voluntário.	
7. O sistema envia uma notificação ao Voluntário por e-mail.	
Cenário alternativo: No passo 3 caso o evento já tenha alcançado o limite de de Voluntários alocado.	
3.1. O sistema deve apresentar a mensagem “O evento escolhido já tem o limite de Voluntários”.	
3.2. O usuário deverá desalocar algum Voluntário clicando no “-” ao lado do nome do Voluntário.	
3.3. Volta ao passo 3.	

Tabela 10 – Detalhamento de caso de uso: UC 9 - Recomendar Voluntário.

Nome do Caso de Uso	UC 9 - Recomendar Voluntário.
Pré-Condições	Existir um Voluntário que já tenha preenchido e enviado os termos de responsabilidade e voluntariado que ainda não tenha sido aprovado pelo Responsável na CnE.
Atores Envolvidos	Responsável pela Empresa.
Resumo	Aborda os passos necessários para o voluntário ser recomendado pelo Responsável pela Empresa.
Cenário principal:	
1. O Responsável pela Empresa seleciona o funcionário que deseja recomendar. [Imagen 21]	
2. O sistema mostra os detalhes do usuário. [Imagen 22]	
3. O Responsável Empresa clica em “Recomendar”.	
4. O sistema envia um e-mail ao Responsável CnE e ao Voluntário informando da recomendação.	
Cenário alternativo: No passo 2 caso o usuário não esteja no status “Aguardando recomendação da Empresa”	
2.1. O sistema não mostrará o botão “Recomendar”.	
2.2 Termina o caso de uso.	

Tabela 11 – Detalhamento de caso de uso: UC 10 - Avaliar Treinamento.

Nome do Caso de Uso	UC 10 - Avaliar Treinamento.
Pré-Condições	O voluntário ter participado de um treinamento e ainda não o ter avaliado.
Atores Envolvidos	Voluntário.
Resumo	Aborda os passos necessários para o voluntário avaliar o treinamento a ele oferecido.
Cenário principal:	
1.	O voluntário deve clicar no evento anterior de treinamento na tela inicial [Imagen 23]
2.	O sistema mostra os um pop-up com o formulário avaliação. [Imagen 24]
3.	O Voluntário deve preencher o formulário e clicar em "Avaliar".

4.3 PROTOTIPAÇÃO DE TELAS

Esta Seção apresenta alguns dos protótipos de tela para os casos de uso "Recomendar Voluntário" e "Avaliar Treinamento":

4.3.1 UC 9 - Recomendar Voluntário

Usuário teste

	Nome	CPF	Função	Status	Termo de Responsabilidade	Termo de Voluntariado
	Alan do Espírito Santo	508.272.561-08	Arquiteto de software	Aguardando envio Termo de Responsabilidade	Enviar Aviso	Enviar Aviso
	Fábio Kukoski	679.207.578-26	Programador II	Aguardando envio Termo de Voluntariado	Visualizar	Enviar Aviso
	Gustavo King	316.472.310-51	Estagiário	Aguardando encaminhamento de Empresa	Visualizar	Enviar Aviso
	Luiz Ricardo Flores Maestri	050.020.139-02	Programador II	Aprovado	Visualizar	Enviar Aviso
	Victor Pereira Goulart	826.284.470-02	Programador I	Aguardando Aprovação CNE	Visualizar	Enviar Aviso

Show 10 • entradas Search:

Showing 1 to 5 of 5 entries

Previous [1](#) Next

by Luís Masetti

Figura 21 – Tela inicial do Responsável pela empresa.

The screenshot shows a web application interface for managing employees. At the top, there is a search bar labeled "Search:" and a dropdown menu "Show 10 ▾ entries". Below this is a table with columns: "Name", "Phone", "Email", "Last login", "Actions", and "Aguardando encaminhamento da Empresa". The table contains five rows of employee data.

Name	Phone	Email	Last login	Actions
Alan do Espírito Santo	509.2772.561-08		2014-07-31 00:00:00	Visualizar Enviar Aviso
Fábiozzi Kruckoski	679.907.278-26		2014-07-31 00:00:00	Visualizar Enviar Aviso
Gustavo Kring	316.472.10.51	gkring@cdf.com.br	2014-07-31 00:00:00	Visualizar Enviar Aviso
Luis Ricardo Flores Mastri	(030) 020.139-02	luisrmaf@cdf.com.br	2014-07-31 00:00:00	Visualizar Enviar Aviso
Victor Pereira Goulart	826.284.470-02	victor.goulart@cdf.com.br	2014-07-31 00:00:00	Visualizar Enviar Aviso

Below the table, there is a message: "Aguardando encaminhamento da Empresa". To the right of this message are two buttons: "Recomendado" (highlighted in blue) and "Fechar". Further down, there is a section titled "Funcionário" with a circular profile picture of Gustavo Kring. The details shown are: Name "Gustavo Kring - Estagiário", Phone "316.472.10.51", Email "gkring@cdf.com.br", Last login "2014-07-31 00:00:00", and Actions "Visualizar" and "Enviar Aviso".

At the bottom of the page, there are navigation links: "Showing 1 to 5 of 5 entries", "Previous", "Next", and "by Luiz Mastri".

Figura 22 – Tela de detalhes do funcionário.

4.3.2 UC 10 - Avaliar Treinamento.

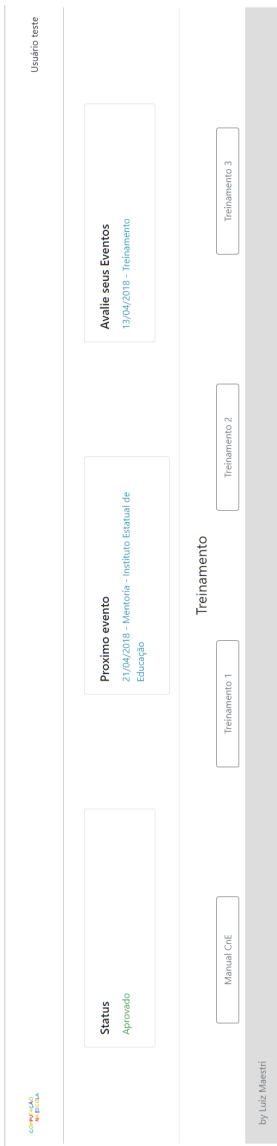


Figura 23 – Tela inicial do Voluntário.



Figura 24 – Tela de avaliação de treinamento.

4.4 MODELAGEM DO BANCO DE DADOS

Esta seção apresenta informações de suma importância para que seja atendido o requisito "RNF 3 - O sistema deve ter integração com o Portal da CnE via banco de dado MySQL", na figura 25, é apresentado o mapeamento de entidades relacionais do sistema CuLTiVAR, sem as tabelas relativas ao site da CnE.

No presente trabalho, o sistema de gerenciamento de banco de dados (SGBD) adotado para desenvolver a ferramenta para gerenciar as inscrições das oficinas e o ambiente colaborativo foi o MySQL, SGBD esse aceito pelo WordPress (WORDPRESS, 2018), tecnologia em que foi criada o atual site da Computação na Escola e demais sistemas preexistentes da iniciativa.

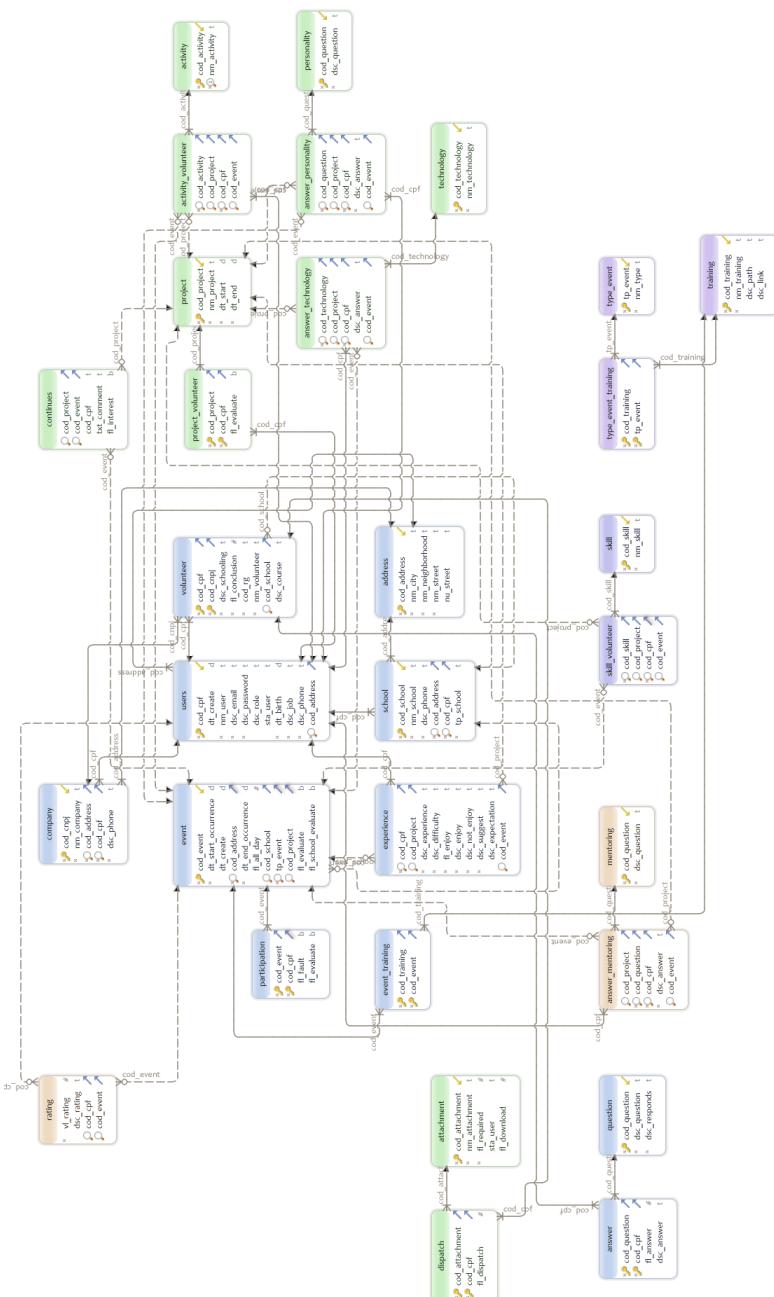


Figura 25 – Diagrama Entidade-Relacional

5 DESENVOLVIMENTO

Este Capítulo tem como objetivo apresentar passo a passo do desenvolvimento, explicando decisões de projeto como adoções de *Design Patterns*, tais como *Service*, *Builder*, *Immutable* entre outros. Na sequência são apresentados detalhes do desenvolvimentos das camadas de backend e de frontend.

5.1 BACKEND

Esta seção tem como objetivo apresentar a camada de backend do sistema desenvolvido, utilizando o framework spring e a biblioteca Java, Lombok.

5.1.1 Spring

Para a implementação do backend foi utilizado o framework Spring¹, mais especificamente o Spring Boot², um dos projetos da Pivotal³, que tem como objetivo simplificar *setup* inicial para a plataforma, trazendo algumas configurações já realizadas internamente, e permitindo que sejam reescritas programaticamente ou por meio de um arquivo nomeado *application.properties*(PIVOTAL, 2018a). Exemplos de ambas configurações apresentadas, em ordem, na Figuras 26 e 27. O trecho de código apresentado, habilita a capacidade de execução assíncrona para o Spring, e injeta no seu contexto um objeto do tipo Executor para utilização (explicado na Tabela 12).

¹<https://spring.io/>

²<https://spring.io/projects/spring-boot>

³<https://pivotal.io/>

```

1  @Configuration
2  @EnableAsync
3  public class AsyncConfig {
4      @Bean
5      public Executor asyncExecutor() {
6          ThreadPoolTaskExecutor executor = new
7              ThreadPoolTaskExecutor();
8          executor.setCorePoolSize(2);
9          executor.setMaxPoolSize(2);
10         executor.setQueueCapacity(50);
11         executor.setThreadNamePrefix("email-");
12         executor.initialize();
13     return executor;
14 }

```

Figura 26 – Código de configuração para funcionamento assíncrono

```

1  spring.datasource.url=jdbc:mysql://localhost:3306/cultivar?
2      useSSL=false&allowPublicKeyRetrieval=true
3  spring.datasource.username===== //Dados sensveis
4  spring.datasource.password===== //Dados sensveis
5
6  logging.level.org.springframework.data=INFO
7  logging.level.org.springframework.jdbc.core.JdbcTemplate=INFO
8
9  scheduler.email.send.cron=0 27 3 * * *
10
11  spring.mail.host=smtp.gmail.com
12  spring.mail.port=465
13  spring.mail.username=====@gmail.com //Dados sensveis
14  spring.mail.password===== //Dados sensveis
15  spring.mail.properties.mail.smtp.auth=true
16  spring.mail.properties.mail.smtp.starttls.enable=true
17  spring.mail.properties.mail.smtp.starttls.required=true
18  spring.mail.properties.mail.smtp.ssl.enable=true
19  spring.mail.test-connection=true

```

Figura 27 – Exemplo de application.properties para configuração do projeto

O arquivo apresentado na figura 27 exemplifica a configuração por meio de arquivos de chave-valor, esse exemplo traz algumas configurações como conexão ao banco de dados, nível de *log* e credenciais para envio de e-mail. Por padrão esse arquivo deve ser salvo no servidor no mesmo diretório onde se encontra o executável gerado na compilação.

O framework Spring fornece abstrações, JSR-330⁴, *Dependency Injection*, e utiliza da JSR-303⁵, *Bean Validation*, por meio das Java Annotations `@Autowired` e `@Valid` respectivamente, ambas utilizadas para o projeto apresentado, como demonstrado no trecho de código abaixo:

```

1  @RestController
2  @RequestMapping(path = "/activity")
3  @AllArgsConstructor(onConstructor = @_(@Autowired))
4  @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5  public class ActivityResource {

6
7      ActivityService service;
8
9      @PostMapping(consumes =
10         MediaType.APPLICATION_JSON_UTF8_VALUE)
11      @ResponseStatus(code = HttpStatus.CREATED)
12      public void create(@Valid @RequestBody final Activity
13          activity) {
14          service.create(activity);
15      }
16      ...
17      @DeleteMapping(path = "/{codQuestion}", produces =
18         MediaType.APPLICATION_JSON_UTF8_VALUE)
19      public Activity delete(@PathVariable final Long
20          codQuestion) throws ServiceException{
21          return service.delete(codQuestion);
22      }
23  }

```

Figura 28 – Classe ActivityResource

Outras Java Annotations do Spring apresentadas nos trechos de código acima são `@Configuration`, `@EnableAsync`, `@Bean`, `@RestController`, `@RequestMapping`, `@PostMapping`, `@DeleteMapping`, `@Respon-`

⁴<https://www.jcp.org/en/jsr/detail?id=330>

⁵<https://www.jcp.org/en/jsr/detail?id=303>

seBody, *@RequestBody* *@ResponseStatus* e *@PathVariable*, a Tabela 12 traz uma breve explicação de cada *Annotations* apresentada juntamente com *@Controller* e *@RequestBody*.

Tabela 12 – Spring Java Annotations (PIVOTAL, 2018b)

Identificação	Descrição
<i>@Configuration</i>	Indica que uma classe declara ao menos um método <i>@Bean</i> e deve ser processada pelo contêiner/contexto do Spring gerando definições de <i>bean</i> .
<i>@EnableAsync</i>	Habilita a capacidade de execução de métodos de modo assíncrono pelo Spring.
<i>@Bean</i>	Indica que o método gerará um <i>bean</i> gerenciado pelo contexto do Spring, geralmente utilizando o <i>pattern</i> de <i>Singleton</i> (explicado a anteriormente).
<i>@Controller</i>	A classe anotada com esta <i>Annotation</i> é considerada um <i>Controller</i> para o Spring e permite que a mesma seja detectada pelo scan de <i>classpath</i> .
<i>@ResponseBody</i>	Indica que o valor retornado pelo método deve ser encapsulado no corpo da resposta.
<i>@RestController</i>	Composição das <i>Annotations</i> <i>@Controller</i> e <i>@ResponseBody</i> .
<i>@RequestBody</i>	O parâmetro com está <i>Annotation</i> será extraído do corpo da requisição.
<i>@RequestMapping</i>	<i>Annotation</i> para mapeamento de requisições Web para métodos em classes de manipulação de Requisições.
<i>@PostMapping</i>	É um abstração da <i>Annotation</i> <i>@RequestMapping</i> para responder somente requisições utilizando o verbo http POST.
<i>@DeleteMapping</i>	É um abstração da <i>Annotation</i> <i>@RequestMapping</i> para responder somente requisições utilizando o verbo http DELETE.
<i>@ResponseStatus</i>	Mapeia o <i>status</i> http da resposta, caso não seja usado os <i>status</i> padrões são 200 para sucesso e 500 para qualquer exceção não tratada.
<i>@PathVariable</i>	Indica que o parâmetro é mapeado pela url da requisição, que é indicada na propriedade <i>path</i> da <i>Annotation</i> <i>@RequestMapping</i> e suas abstrações, a variável é indicada na propriedade por {propriedade}.

5.1.2 Project Lombok

O Lombok⁶ é uma biblioteca java que utiliza-se de *Annotations* para gerar código em tempo de compilação, deixando o código Java mais simples e amigável, este é um exemplo de como código fica com as *Annotations* do Lombok, Figura 29.

```
1  @Value
2  @Wither
3  @Builder(builderClassName = "Builder")
4  @...
5  @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
6  public class School {
7      Long codSchool;
8      @...
9      String name;
10     @...
11     String phone;
12     @...
13     Address address;
14     @...
15     User responsible;
16     @...
17     SchoolType type;
18
19     ...
20 }
```

Figura 29 – Exemplo de Classe Java utilizando *Annotations* do Lombok

⁶<https://projectlombok.org/>

Para cada *Annotation* apresentada na figura Figura 29 será compilada para os trechos representados nas figuras 30, 31, 32 e 33.

A figura 30 apresenta alguns trechos de código gerados pela *Annotation* `@Value` responsável por gerar um construtor com todos os atributos da classe, os métodos de acesso, *Getters*, e os métodos equals e hashCode.

```

1  public class School {
2      ...
3      @java.beans.ConstructorProperties({"codSchool", "name",
4          "phone", "address", "responsible", "type"})
5      School(Long codSchool, @NotNull @NotNull String name,
6          @NotNull @NotNull String phone, @Valid @NotNull
7          Address address, @Valid @NotNull User responsible,
8          @NotNull SchoolType type) {
9      ...
10 }
11
12     public Long getCodSchool() {
13         return this.codSchool;
14     }
15
16     public @NotNull @NotNull String getName() {
17         return this.name;
18     }
19
20     public boolean equals(Object o) {
21         ...
22     }
23 }
```

Figura 30 – Código gerado pela *Annotation* `@Value`

A *Annotation* `@Wither` gera métodos, figura 31, para "alteração" dos atributos do objeto, explicado na seção 5.1.3.

```

1  public class School {
2      ...
3      public School withPhone(@NotBlank @NotNull String phone)
4          {
5              return this.phone == phone ? this : new
6                  School(this.codSchool, this.name, phone,
7                      this.address, this.responsible, this.type);
8          }
9      ...
10     ...
11 }
```

Figura 31 – Método gerado pela *Annotation @Wither*

A *Annotation @Builder*, como apresentado na figura 32, é utilizada para se trabalhar como o *Pattern Builder*, mais explicações na seção 5.1.3.

```

1  public class School {
2      ...
3      public static Builder builder() {
4          return new Builder();
5      }
6      ...
7      public static class Builder{
8          ...
9      }
10     ...
11 }
```

Figura 32 – código gerado pela *Annotation @Builder*

A *Annotation @FieldDefaults* tem o objetivo de configurar a visibilidade dos atributos da classe por ela anotada, como mostra a Figura 33

```

1  public class School {
2      private final Long codSchool;
3      @NotBlank
4      @NotNull
5      private final String name;
6      @NotBlank
7      @NotNull
8      private final String phone;
9      @Valid
10     @NotNull
11     private final Address address;
12     @Valid
13     @NotNull
14     private final User responsible;
15     @NotNull
16     private final SchoolType type;
17     ...
18 }
```

Figura 33 – código gerado pela *Annotation @FieldDefaults*

5.1.3 Design Patterns

Esta subseção é destinada a explicações das adoções dos *patterns* adotados no backend.

O Spring trabalha com o conceito de *Proxy*, um *Design Pattern* estrutural definido pela GoF. Esse *Design Pattern* encapsula um objeto através de um outro objeto que possui a mesma interface de comunicação, métodos, de forma que o segundo objeto, *Proxy*, controla o acesso ao primeiro (DEVMIDIA, 2006).

Outro *Pattern* do Spring é o *Singleton*, permite a criação de somente um instância da classe implementada utilizando o *Pattern*, evitando assim a criação de instâncias desnecessárias e por sua vez o consumo de memória.(DEVMIDIA, 2010)

Uma aplicação construída com Spring, é dividida em camada chamadas no ecossistema da plataforma de *Stereotypes*, são elas, *Repository* para acesso ao banco de dados, comumente chamada de DAO , Service sendo a camada de negócio, sendo a camada mais robusta da aplicação e a camada de *Controller* que faz a comunicação entre a visão, *View*, e o *Service*.

Alguns dos *Design Patterns* usados no projeto foram *Builder*, *Service*, *DAO*, *DTO*, *PO*, *dependency injection* e *Immutable*, que serão exemplificados por meios dos trechos de códigos da figura 34 à figura 38:

```

1   @Value
2   @Wither
3   @Builder(builderClassName = "Builder")
4   @JsonDeserialize(builder = Address.Builder.class)
5   @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
6   public class Address {
7       Long codAddress;
8       @NotBlank
9       @NotNull
10      String city;
11      @NotBlank
12      @NotNull
13      String neighborhood;
14      @NotBlank
15      @NotNull
16      String street;
17      String number;
18
19      @JsonPOJOBuilder(withPrefix = "")
20      public static class Builder{
21          public Builder(){}
22      }
23 }
```

Figura 34 – Exemplo de PO com as *Annotations* do Lombok

Por meio das *Annotations* *@Value*, *@Wither*, *@Builder* e *@FieldDefaults* providas lombok, a classe *Address* é uma representação do modelo de dados, *PO*, com os atributos privados e *final*, impedindo alterações após a instanciação da classe, *immutable*, somente com os métodos de acesso e comparação gerados pela *Annotation* *@Value*. A *Annotation* *@Wither* gera métodos similares ao *Setters*, contanto, esses métodos não alteram o próprio objeto, mas instanciam e retornam um novo objeto com somente o atributo que o método propõe-se a alterar diferente do objeto original, como mostra a figura 35.

```

1   public Address withCodAddress(Long codAddress) {
2       return this.codAddress == codAddress ? this : new
3           Address(codAddress, this.city, this.neighborhood,
4               this.street, this.number);
5   }

```

Figura 35 – Exemplo método gerado pela *Annotation @Wither* do Lombok

```

1  @Repository
2  @AllArgsConstructor(onConstructor = @__(@Autowired))
3  @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4  public class PersonalityRepository {
5
6      NamedParameterJdbcTemplate jdbcTemplate;
7
8      ...
9
10     private Personality build(ResultSet rs , int i) throws
11         SQLException {
12         return Personality.builder()
13             .codQuestion(rs.getLong("cod_question"))
14             .question(rs.getString("dsc_question"))
15             .build();
16     }
17
18     ...
19 }

```

Figura 36 – Exemplo de uso de *Builder*

O código da figura 36 o uso de *Builder* para construção de objetos dentro do método **build(ResultSet rs , int i)**. A classe *PersonalityRepository* de acesso a base de dados comumente chamada de *DAO*. Dentro da arquitetura do Spring recebe o nome de *repository*. Sobre esta classe é também aplicado o *Pattern de dependency injection* por meio *Annotation @Autowired*, que será aplicada sobre o construtor gerado durante compilação pelo Lombok, figura 37.

```

1  @java.beans.ConstructorProperties({"jdbcTemplate"})
2  @Autowired
3  public PersonalityRepository(NamedParameterJdbcTemplate
4      jdbcTemplate) {
5      this.jdbcTemplate = jdbcTemplate;
6  }

```

Figura 37 – Exemplo de uso de *Builder*

A figura 38 apresenta um exemplo de classe Service, este tipo de classe segundo Fowler et al. (2002) encapsulam a lógica de negócios da aplicação, controlando transações e coordenando respostas na implementação de suas operações. O método da classe apresentado na figura 38 é a criação de um novo usuário onde pode encontrar uma validação do objeto *Address* antes da execução da inserção no banco de dados.

```

1  @Service
2  @...
3  public class UserService {
4      ...
5      public void create(final User user) throws
6          ServiceException {
7          val address = user.getAddress();
8          if (!ValidateUtils.isValid(address)) {
9              throw new InvalidException(null);
10         }
11         userRepository.create(
12             user.withAddress(
13                 address.withCodAddress(
14                     addressRepository.create(
15                         address
16                     )
17                 )
18             );
19     }
20     ...
21 }

```

Figura 38 – Classe *Service*

5.1.3.1 Immutable

Esta subseção visa explicar o motivo do uso de *immutable* e como a JVM trabalha com esse tipo de objetos. Infelizmente o servidor fornecida para implantação do sistema implementado tinha um hardware limitando quando a memória. Assim obrigando a utilização de patterns que utilizem uma menor quantidade deste recurso, tal qual, é o pattern Immutable, explicado abaixo. Os objetos imutáveis muitas vezes não são empregados por programadores, devido a uma superestimação do custo para criação de um novo objeto no lugar de somente alterar o existente. Porém, pode ser compensado por algumas das eficiências associadas a objetos imutáveis(ORACLE CORPORATION, 2014). Isso inclui uma otimização no coletor de lixo, visto que não se faz necessário a execução do código para proteger os objetos mutáveis contra corrupção(WIKIWIKIWEB, 2011)(ORACLE CORPORATION, 2014).

5.2 FRONT-END

Nesta seção são apresentados alguns detalhes da interface de interação com o usuário, como principais telas e funcionalidades e componentes.

5.2.1 React JS

Para implementação da camada de front-end foi utilizada a biblioteca de JavaScript React JS⁷. React foi pensado para que os elementos da tela podessem ter seus comportamentos encapsulados em forma de componentes flexíveis, independente e reutilizável(EIS, 2016)(TABLELESS, 2018), como mostram as figuras 39, 40.

⁷<https://reactjs.org>

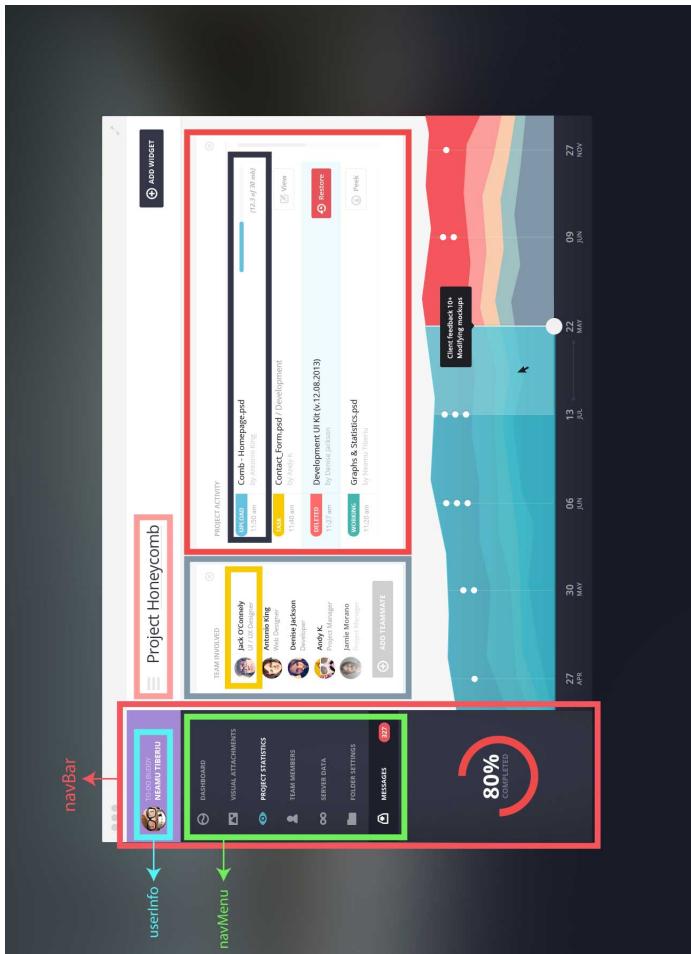


Figura 39 – Exemplos de Tela criada com React JS e componentes (EIS, 2016).

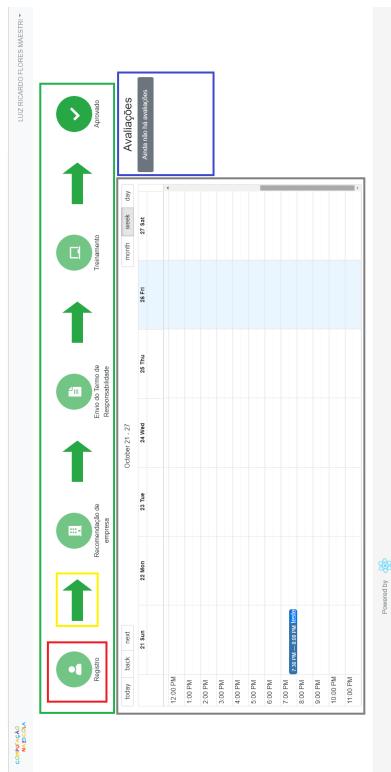


Figura 40 – Tela desenvolvida para o Sistema Cultivar com alguns componentes identificados.

Com a ideia de reuso de componentes em primeiro plano, foram utilizadas bibliotecas de componentes prontos, tais como: Reactstrap⁸, um biblioteca de componentes já estilizados com bootstrap⁹.

React-big-calendar¹⁰ para criação de uma agenda com layout similar ao Google Calendar¹¹

O código apresentado na figura 41 representa o componente da tela principal do voluntário responsável por iniciar a avaliação de um evento ou projeto que o voluntário tenha participado.

⁸<http://reactstrap.github.io/>

⁹<https://getbootstrap.com>

¹⁰<https://www.npmjs.com/package/react-big-calendar>

¹¹<https://www.google.com/calendar>

```
1 import React, { Component, Fragment } from 'react';
2 import { Row, Col, Button } from 'reactstrap';
3 import Form from './form';
4
5 export default class extends Component{
6     constructor(){
7         super();
8         this.state = {
9             isOpen : false
10        };
11        this.toggle = this.toggle.bind(this);
12    }
13
14    toggle(){
15        const { isOpen } = this.state;
16        this.setState({ isOpen: !isOpen });
17    }
18
19    render(){
20        const { evaluate, cpf, afterSubmit } = this.props;
21        const { isOpen } = this.state
22        (
23            <Fragment>
24                <Row>
25                    <Col>
26                        <Button color="primary" style={{ width: 'inherit', height: '40px' }} onClick={this.toggle}>
27                            {evaluate.title}
28                        </Button>
29                    </Col>
30                </Row>
31                {isOpen && <Form title={evaluate.title}
32                    code={evaluate.codEvent || evaluate.codProject} cpf={cpf}
33                    afterSubmit={afterSubmit}
34                    isOpen={isOpen} close={this.toggle}
35                    isProject={evaluate.isProject}/>}
36                    <br />
37                </Fragment>
38            )
39        }
40    }
41 }
```

Figura 41 – Exemplo de Componente React.

O componente apresentado na figura 41, foi escrito utilizando ECMAScript 6¹² e JSX¹³, que são, a linguagem script padronizada base para o JavaScript (MOZILLA, 2016) e uma extensão do ECMAScript (FACEBOOK, 2014), respectivamente.

Para Facilitar a explicação do código apresentado na figura 41, será explicado método a método nas figuras 42, 43 e 44.

A figura 42 demonstra o construtor do componente configurando o estado inicial do mesmo.

```

1   constructor(){
2     super();
3     this.state = {
4       isOpen : false
5     };
6     this.toggle = this.toggle.bind(this);
7 }
```

Figura 42 – Construtor do componente.

O método *toggle* demonstrado na figura 43 é responsável por abrir o formulário de avaliação alterando o estado do componente por meio do método *setState*, esta alteração faz com que o componente execute a função *render*, explicada na figura 44, novamente utilizando agora o novo estado.

```

1   toggle(){
2     const { isOpen } = this.state;
3     this.setState({ isOpen: !isOpen });
4 }
```

Figura 43 – Método *toggle*.

A figura 44 apresenta o método *render*, que é o responsável por transformar renderizar o JSX na tela.

O componente é composto por outros componentes são eles:

- *Row* que será renderizado como uma `<div class="row">`.
- *Col* que será renderizado como uma `<div class="col">`.

¹²<http://www.ecma-international.org/memento/tc39.htm>

¹³<https://facebook.github.io/jsx/>

- *Button* que será renderizado como uma `<button type="button" class="btn btn-primary">` e executará o método `toggle` quando clicado.
- *Form*, que é o formulário de avaliação. em forma de um *modal*.

```

1  render(){
2      const { evaluate, cpf, afterSubmit } = this.props;
3      const { isOpen } = this.state
4      return (
5          <Fragment>
6              <Row>
7                  <Col>
8                      <Button color="primary" style={{ width:
9                          'inherit', height: '40px' }}
10                         onClick={this.toggle}>
11                             {evaluate.title}
12                         </Button>
13                     </Col>
14                 </Row>
15                 {isOpen && <Form title={evaluate.title}
16                     code={evaluate.codEvent || evaluate.codProject} cpf={cpf}
17                     afterSubmit={afterSubmit} isOpen={isOpen}
18                     close={this.toggle}
19                     isProject={evaluate.isProject}/>}
20                     <br />
21                 </Fragment>
22             )
23     }

```

Figura 44 – Método render.

5.2.2 Interfaces

Esta seção tem como objetivo demonstrar a utilização das telas do Sistema Cultivar.

5.2.2.1 Login

Na figura 45 é demonstrada a tela de *login* com um campo Usuário(1) e Senha(2) que o usuário deve preencher com seu cpf/email e senha no sistema, e clicar no botão "Entrar"(3) para realizar sua autenticação no sistema. Logo abaixo do botão "Entrar" há o link "Quero ser voluntário"(4) que redireciona para o formulário inicial de cadastro do voluntário.

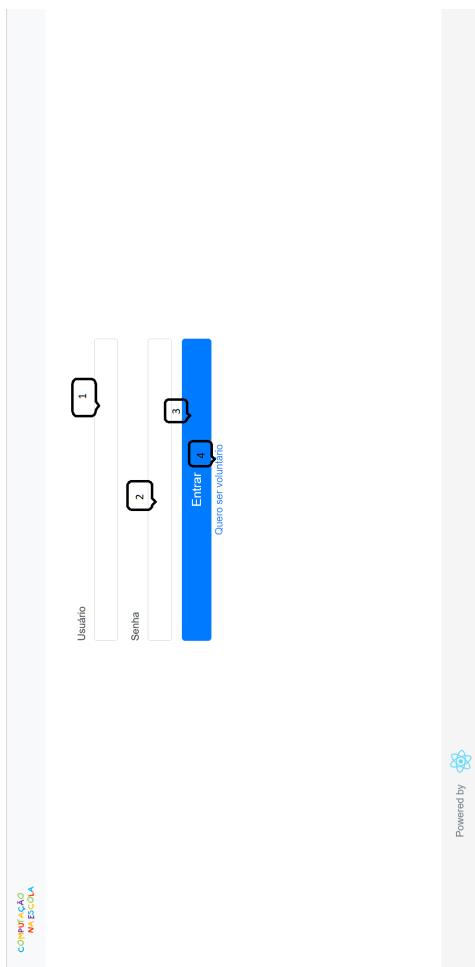


Figura 45 – Tela de Login.

5.2.2.2 Voluntário

As figuras 46, 47 e 48 demonstram o formulário inicial, termo de voluntariado, para o seu cadastro como de acordo com os casos de uso UC11 e UC13.

Dados do Voluntário

Nome Completo*

CPF*

RG*

Número para Contato*

Empresa*

 Seleccione ▾

Cargo na Empresa*

Escolaridade *

Completo Não

- 1º Grau (Fundamental)
- 2º Grau (Ensino Médio)
- Ensino Superior (Graduação)
- Pós-Graduação

Nascimento*

 dd/mm/aaaa

Email*

Senha*

Confirmar Senha*

[Cancelar](#)

[Avançar](#)

Figura 46 – Primeiro passo do *wizard* de registro do de voluntário.

Enderço do Voluntário

Cidade*

Bairro*

Lougradouro*

Número

[Voltar](#)

[Avançar](#)

Figura 47 – Segundo passo do *wizard* de registro do voluntário.

teste 1

Resposta

Não

[Voltar](#)

[cadastrar](#)

Figura 48 – Último passo do *wizard* de registro do voluntário.

A Figura 49 apresenta a tela do voluntário na primeira vez que entrar no sistema onde permite que o mesmo termine seu registro realizando *uploads* dos arquivos solicitados/necessários, como especificado pelo RF 2 e UC11.

Após executar o envio dos arquivos o usuário será transferido para tela presente na figura 50, onde terá de esperar o Responsável da empresa ligado a ele o recomendar para a CnE.

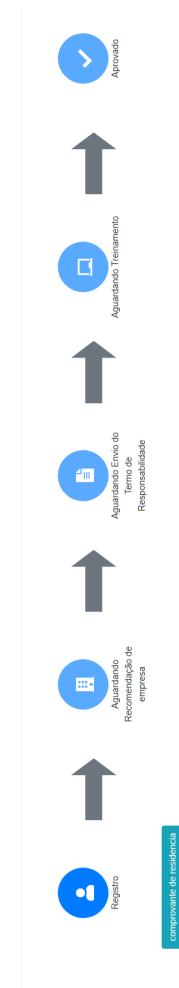


Figura 49 – Primeiro Login do voluntário.



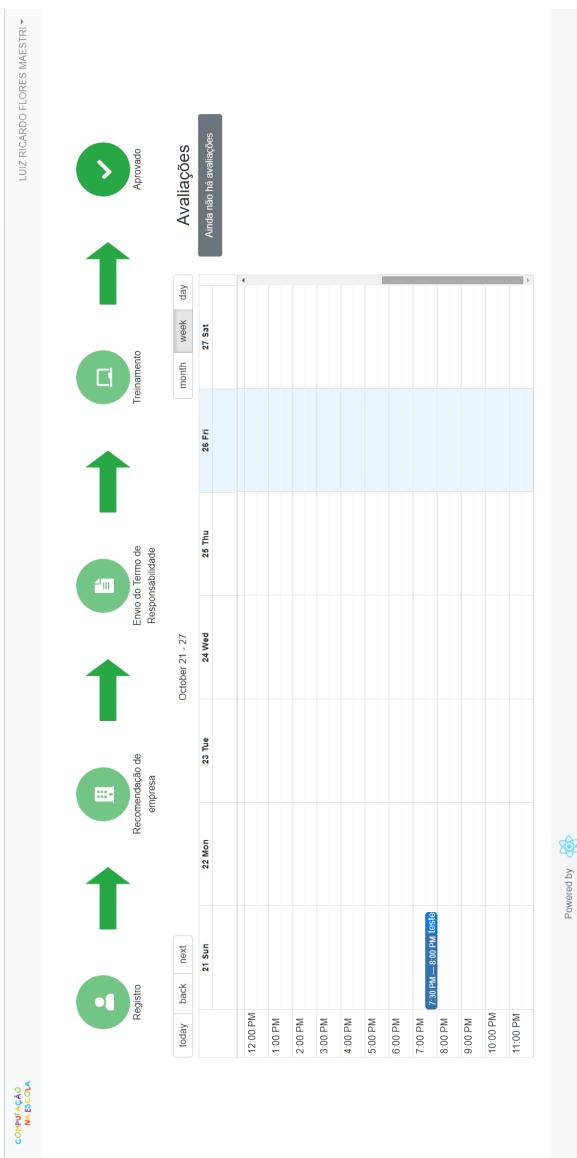
Figura 50 – Tela de Espera pela recomendação da empresa.

Na Figura 51 apresenta a tela referente ao UC12 - Gerar termo de responsabilidade, onde o usuário poderá realizar o *download* e *upload* do mesmo.



Figura 51 – Tela de envio do termo de responsabilidade.

A figuras 52 e 53 apresentam o *dashbord* principal do Voluntário. Onde ele pode visualizar os eventos do quais participará pela CnE, e poderá avaliar os eventos anteriores como explicitado nos UCs 6 e 10.

Figura 52 – *Dashboard* voluntário.

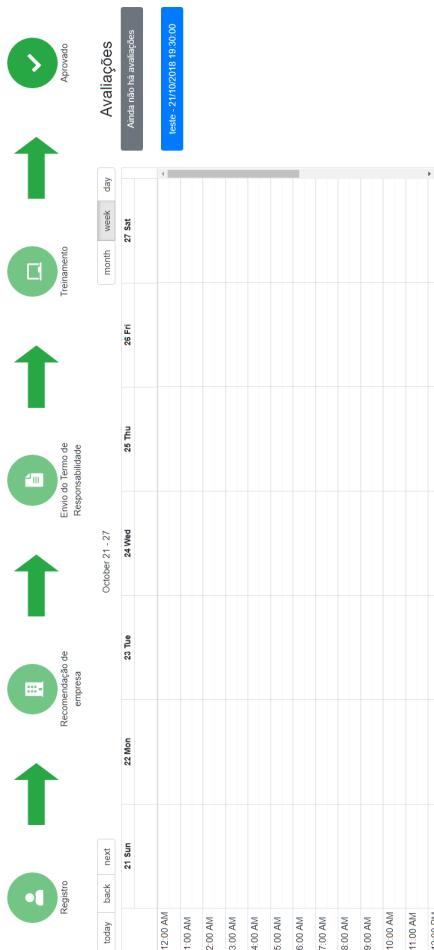


Figura 53 – Dashboard voluntário.

Ao clicar no evento marcado no calendário ou no botão em avaliações abrirá um *pop-up*, figura 54, com dados do evento ou o um *pop-up*, figura 55, com o formulário para avaliar o evento já ocorrido como previstos nos UC 6 e 10.

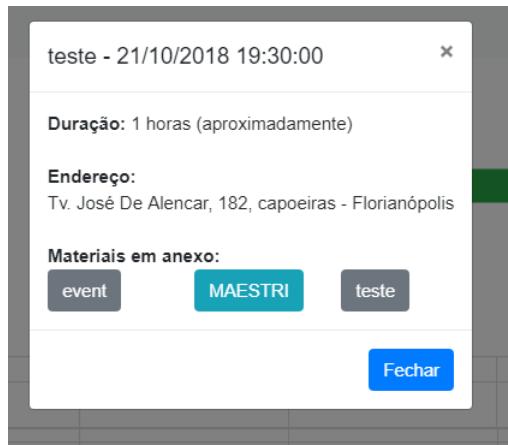


Figura 54 – Detalhes do evento.

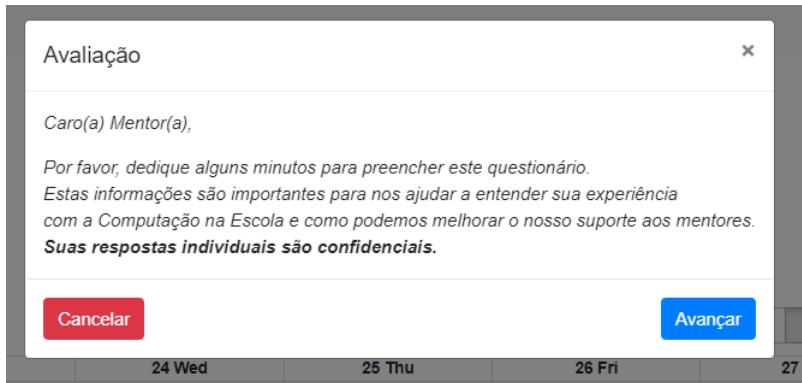


Figura 55 – Formulário de avaliação de eventos do Voluntário.

5.2.2.3 Responsável da Empresa

A figura 56 apresenta a tela principal do Responsável da empresa, onde lhe é apresentado uma listagem dos voluntários da empresa(1) e o Responsável a empresa pode realizar as seguintes ações:

- Filtrar a tabela pelo nome do voluntário(2), após a terceira letra começar a filtrar.
- Visualizar o status da candidatura do Voluntário(3).
- Visualizar mais detalhes do Voluntário(4), como previsto no UC11. Ao clicar no botão abrirá um *pop-up* com as informações do Voluntário como pode ser visto na figura 57.
- Visualizar locais onde o Voluntário já esteve representando a empresa na CnE(5).
- Recomendar a CnE a candidatura do Voluntário(6), como previsto no UC9. Ao clicar no botão abrirá um *pop-up* com perguntas de sim ou não sobre a recomendação do Voluntário como pode ser visto na figura 58.
- Deletar o Voluntário(7), como previsto no UC11, quando o mesmo não tem nenhum evento futuro associado a ele.

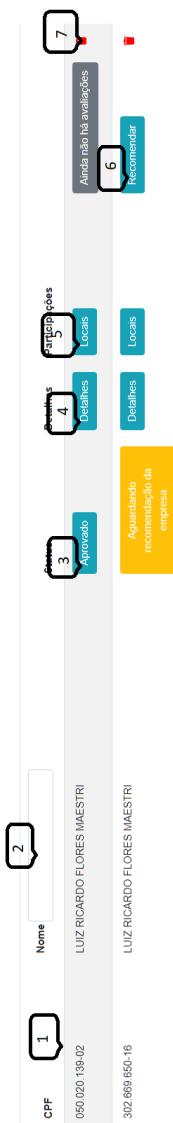


Figura 56 – Tela de Espera pela recomendação da empresa.



Figura 57 – Tela de detalhes do usuário.



Figura 58 – Tela de recomendação da empresa.

5.2.2.4 Responsável na CnE

A figura 59 apresenta a tela inicial do Responsável na CnE.
Onde o mesmo pode realiza:

- UC 8 - CRUD de Empresa, clicando nos itens 1 e 2 que abrirá um *pop-up* com o formulário para cadastro da empresa ou a excluirá a empresa, respectivamente.
- UC 2 - CRUD Evento, selecionando uma faixa de tempo dentro do calendário.
- UC 7 - CRUD de escola, clicando nos itens 4 e 5 que abrirá um *pop-up* com o formulário para cadastro da escola ou a excluirá a empresa, respectivamente.

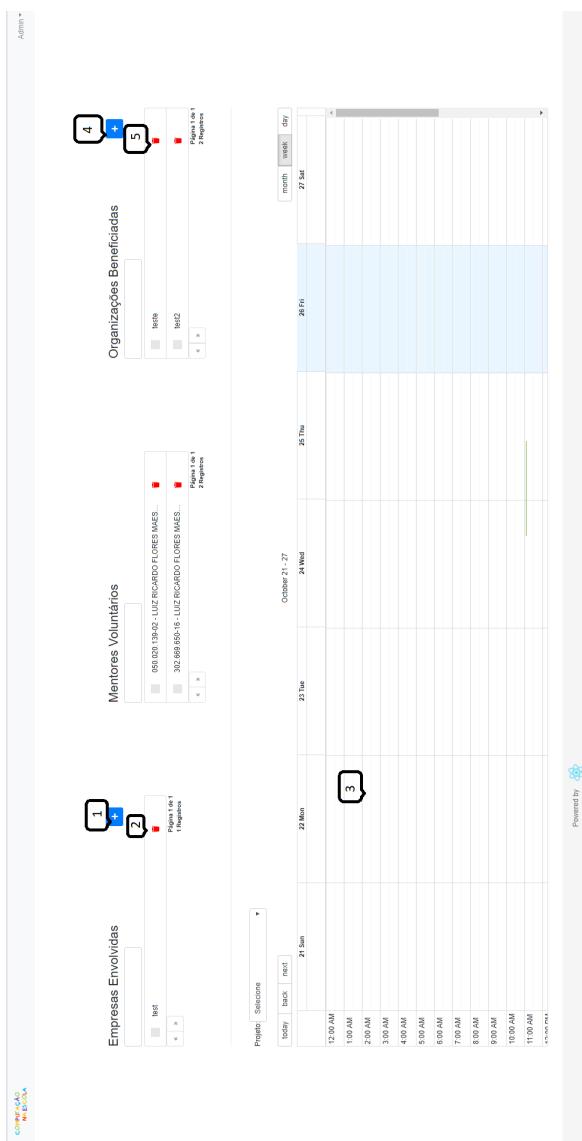


Figura 59 – Dashboard Responsável na CnE.

6 AVALIAÇÃO DO SISTEMA CULTIVAR

Com o objetivo de avaliar a qualidade do Sistema Cultivar foram realizadas avaliações com usuários entre os dias 20 e 26 de outubro de 2018, em relação a utilidade, funcionalidade, eficiência e usabilidade pela perspectiva de dos voluntários.

6.1 TESTE COM USUÁRIOS

Segundo Konaigari (2017) o teste com usuários é projetado para avaliar o produto segundo a visão de usuários reais, buscando descobrir problemas e pontos de melhorias(CAELUM, 2018).

Os testes foram feitos com ex-voluntários da CnE, para que houvesse uma comparação do processo atual, realizado manualmente, com o proporcionado pelo sistema desenvolvido neste projeto.

Todos os voluntários participaram da CnE entre os anos de 2014 e 2015, quando estavam entre 2^a e 4^a fase dos cursos de Ciências da Computação e Sistemas de Informação. Os mesmos foram selecionados por meio do grupo do curso de Sistemas.

O objetivo desta avaliação é analisar a qualidade do Sistema Cultivar em termos de utilidade, funcionalidade, e usabilidade pelo ponto de vista dos voluntários.

6.1.1 Utilidade

Sobre a utilidade do sistema foi aplicado um breve questionário para usuários, apresentado na tabela 13. De forma geral os voluntários consideraram o sistema útil para sua finalidade de facilitar o "dia-a-dia" dentro da Computação na Escola, apesar de reclamações sobre os email enviados diariamente ao invés de semanalmente alertando de compromissos da CnE e da falta de sincronização com o *google calendar*.

Sendo este o motivo pelo qual metade dos usuários consideraram muito mais incomodo do que útil para essa função.

Questão de Análise	SIM	NÃO
O Sistema Cultivar foi útil para sua candidatura a vaga de voluntário?	4	0
O Sistema Cultivar foi útil para organizar-se com os horários das oficinas/projetos?	2	2
O Sistema Cultivar foi útil para auxiliar a preparação das oficinas?	4	0

Tabela 13 – Respostas do questionário de utilidade

6.1.2 Funcionalidade

Para o quesito funcionalidade, o sistema foi avaliado de maneira levemente positiva pelos usuários como apresentado na tabela 14, não apresentando erros durante a execução do módulo do voluntário.

Entretanto foi quase unânime para os usuários que somente uma *label* colorida não é o suficiente para o usuário compreender o motivo do *feedback* da escola quando negativo. Mesmo que compreendam que os textos de *feedbacks* fornecidos pelas escolas possam acabar ferindo suas pessoas caso expostos sem um "filtro" de linguagem antes.

Questão de Análise	SIM	NÃO
Você acha que as informações disponibilizadas no evento são suficientes?(localização, data e horário)	4	0
Você observou algum erro (<i>bug</i>) em relação a funcionalidade da ferramenta?	0	4
Você acha que as informações de feedback são suficiente?	1	3

Tabela 14 – Respostas do questionário de Funcionalidade

6.1.3 Usabilidade

A avaliação de usabilidade, em termos de satisfação foi realizada por meio de um diagrama de SUS, que consiste em um questionário de 10 perguntas, com respostas em uma escala de 1 a 5, onde 1 significa Discordo Completamente e 5 significa Concordo Completely(TEIXEIRA, 2015). São as questões(TEIXEIRA, 2015):

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Observou-se avaliações acima de 85 pontos. Como apresentado na Tabela 15, as respostas de usuário para cada questão são apresentadas na tabela 16.

Tabela 15 – Resultado da pesquisa SUS

Pontuação SUS				Média
97,5	92,5	92,5	87,5	92,5

Tabela 16 – Respostas dos voluntários

Voluntário	1	2	3	4	5	6	7	8	9	10	Nota
Victor Goulart	5	1	5	1	5	1	5	1	4	1	97,5
Felipe Senna	5	1	5	1	4	1	4	1	4	1	92,5
Fabrizio Krukoski	5	2	4	1	5	1	5	1	4	1	92,5
Gabriela Mattos	4	2	4	1	5	2	5	1	4	1	87,5

6.1.4 Pontos Fortes

Baseado no feedback dos usuários o modulo de voluntário do Sistema Cultivar, é considerado simples de usar por adotar padrões de mercado como o uso do componente de calendário similar ao componente utilizado no Google Calendar.

Motivo pelo qual os voluntários sentiram-se bem familiarizados com o sistema, não precisando de muito tempo para se adaptarem ao sistema.

6.2 AMEAÇAS A VALIDADE DA AVALIAÇÃO

A maior ameaça a validade da avaliação feita é a quantidade de usuários disponíveis para avaliar a ferramenta e a falta de acesso a usuários representantes dos atores Responsável da Empresa e Responsável da Escola, devido a falta de tempo para o contato com os mesmos, entende-se que para uma maior precisão e confiabilidade do resultados é necessário mais usuários e representantes dos atores não testados.

7 CONCLUSÃO

O objetivo principal deste trabalho foi desenvolver e avaliar uma ferramenta web de gestão do processo de voluntariado para o ensino de computação nas escolas, no contexto da Iniciativa Computação na Escola do Departamento de Informática e Estatística da UFSC. Neste contexto, foi feita a análise da fundamentação teórica sobre gestão de voluntariado e desenvolvimento web (O1).

Também foi realizada a análise do estado da arte identificando as principais web ferramentas de auxílio ao voluntariado (O2).

Deste modo, baseando-se na fundamentação teórica e na análise do estado da arte, juntamente a entrevistas com responsáveis pela Computação na Escola, foram elicitados os requisitos, modelado a base de dados e arquitetura do sistema, para viabilizar a implementação do mesmo (O3).

Os resultados da avaliação do Sistema Cultivar fornecem uma indicação inicial da sua utilidade, funcionalidade, desempenho e usabilidade muito positiva (O4), porém ainda incompleta.

O sistema Cultivar permite cadastro e avaliação dos mentores voluntários, disponibilizando uma ferramenta de feedback mais segura para a CnE, e simplificando a iniciativa do voluntário de participar como mentor.

Como trabalhos futuros, recomenda-se algumas pequenas alterações da interface para deixa-lá mais amigável a dispositivos móveis permitindo a compatibilidade com PWAs, aplicações webs que, nos dispositivos moveis, progressivamente assumem funcionalidades que antes eram possíveis somente em aplicações nativas(LIMA, 2017). E com o possível aumento dos voluntários em projetos da Computação na Escola, a troca da máquina servidor para uma máquina mais robusta.

REFERÊNCIAS

- ABDUL-ALIM, J. Students not prepared for careers in computer science. *Diverse – Issues in Higher Education*, 2015. <<http://diverseeducation.com/article/77408/>>. Acessado em 10/09/2017.
- ALLENDE, F. O entra e sai nas universidades. *G1*, São Paulo, Brasil, p. 1–1, Out 2012. <<http://g1.globo.com/sp/santos-regiao/blog-do-allende/platb/2012/10/17/o-entra-esai-nas-universidades/>>. Acessado em 21/09/2017.
- BERKENBROCK, C. D. M. Mestre em Ciências da Computação, *Investigação e Implementação de Estratégias de Notificação de Invalidação para Coerência de Cache em Ambientes de Computação Móvel sem Fio*. Florianópolis, Brasil: [s.n.], 2005. <<https://repositorio.ufsc.br/bitstream/handle/123456789/102464/214169.pdf?sequence=1>>. Acessado em 24/05/2018.
- BETTER IMPACT. *Dashboard*. 2018. <<https://www.betterimpact.co.uk/>>. Acessado em 16/06/2018.
- BRASIL. Lei nº 9.608, de 18 de fevereiro de 1998. dispõe sobre o serviço voluntário e dá outras providências. *Diário Oficial [da] República Federativa do Brasil*, Brasília, DF, 18 fev. 1998. <http://www.planalto.gov.br/ccivil_03/LEIS/L9608.htm>. Acessado em 03/06/2018.
- CAELUM. *Apêndice - Testes de Usabilidade*. 2018. <<https://www.caelum.com.br/apostila-ux-usabilidade-mobile-web/usabilidade/>>. Acessado em 30/10/2018.
- CAIVANO, R. M.; VILLORIA, L. N. *Aplicaciones WEB 2.0 - Google Docs*. Villa María, Argentina: Editorial Universitaria de Villa María, 2009.
- CENTRO DE VOLUNTARIADO DE SÃO PAULO. *Palestra Voluntariado e a Transformação Social - 2016*. 2016. <<http://www.voluntariado.org.br/sms/files/2016%20%20Palestra%20Voluntariado%20e%20Transforma%C3%A7%C3%A3o%20Social.pdf>>. Acessado em 25/05/2018.

CNE, C. N. E. F. *Iniciando o projeto Jovens Tutores de Programação*. 2017. <<http://www.computacaonaescola.ufsc.br/?p=2217>>. Acessado em 14/09/2017.

DEVMIDIA. *Design Patterns – Singleton – Parte 3*. 2006. <<https://www.devmedia.com.br/conheca-o-pattern-proxy-gof-gang-of-four/4066>>. Acessado em 25/10/2018.

DEVMIDIA. *Conheça o Pattern Proxy - GoF (Gang of Four)*. 2010. <<https://www.devmedia.com.br/design-patterns-singleton-parte-3/16782>>. Acessado em 25/10/2018.

DEVMIDIA. *Spring Boot: simplificando o Spring*. 2015. <<https://www.devmedia.com.br/spring-boot-simplificando-o-spring/31979>>. Acessado em 26/10/2018.

DIÁRIO CATARINENSE. Terceira edição do de 2012 do projeto amigos da escola acontece hoje. *Diário Catarinense*, Florianópolis, Brasil, 2012. <<http://dc.clicrbs.com.br/sc/noticias/noticia/2012/08/terceira-edicao-do-de-2012-do-projeto-amigos-da-escola-acontece-hoje-3871028.html>>. Acessado em 16/06/2018.

DO-IT. *Dashboard*. 2018. <<https://do-it.org/>>. Acessado em 16/06/2018.

DUBOW, W. *By the numbers*. Boulder, 2013. <<http://www.newit.org/bythenumbers>>. Acessado em 02/09/2017.

EIS, D. *Hello World com React, do rascunho até o primeiro componente*. 2016. <<https://tableless.com.br/hello-world-com-react-do-rascunho-ate-o-primeiro-componente/>>. Acessado em 30/10/2018.

EISENMAN, B. *Learning React Native*. Sebastopol, Califórnia, EUA: O'Reilly Media, 2015.

FACEBOOK. *Draft: JSX Specification*. 2014. <<https://facebook.github.io/jsx/>>. Acessado em 30/10/2018.

FACEBOOK. *Flux*. 2015. <<https://facebook.github.io/flux/>>. Acessado em 23/05/2018.

FACEBOOK. *React*. 2018. <<https://reactjs.org/>>. Acessado em 23/05/2018.

- FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado em Ciências da Computação) — UNIVERSITY OF CALIFORNIA, Irvine, EUA, 2000. <<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>. Acessado em 23/05/2018.
- FOWLER, M. et al. *Patterns of Enterprise Application Architecture*. Boston, Massachusetts, EUA: Addison-Wesley Professional, 2002.
- GLOBO, R. *Manual AMIGOS DA ESCOLA - Guia de Ação*. 2012. <<http://download.globo.com/amigosdaescola/manual.pdf>>. Acessado em 30/11/2018.
- GOLDSTEIN, I. S. *Responsabilidade Social - Das Grandes Corporações ao Terceiro Setor*. São Paulo, Brasil: Ed. Atica, 2007.
- HUNT, P. *Why did we build React?* 2013. <<https://reactjs.org/blog/2013/06/05/why-react.html>>. Acessado em 16/06/2018.
- INEP. *Sinopse Estatística de Educação Superior 2015*. Brasília, 2016. <<http://portal.inep.gov.br/web/guest/sinopses-estatisticas-da-educacao-superior>>. Acessado em 14/09/2017.
- INEP. *Censo Escolar da Educação Básica 2016 : Notas Estatísticas*. Brasília, 2017. <http://download.inep.gov.br/educacao_basica/censo_escolar/notas_estatisticas/2017/notas_estatisticas_censo_escolar_da_educacao_basica_2016.pdf>. Acessado em 14/09/2017.
- JENKINS, T. *THE MOTIVATION OF STUDENTS OF PROGRAMMING*. Tese (Mestrado em Ciências da Computação) — University Of Kent, Cantuária, Reino Unido, 2001. <<https://www.cs.kent.ac.uk/pubs/2001/1401/content.pdf>>. Acessado em 21/09/2017.
- JENKINS, T. On the difficulty of learning to program. In: *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*. Leeds, Reino Unido: [s.n.], 2002. v. 1, p. 53–58. <<http://www.psy.gla.ac.uk/~steve/localed/jenkins.html>>. Acessado em 20/08/2017.
- KNOW HOW NON PROFIT(ORG.). *HOW TO CHOOSE A VOLUNTEER MANAGEMENT SYSTEM*. 2018.

<<https://knowhownonprofit.org/how-to/how-to-choose-a-volunteer-management-system>>. Acessado em 22/04/2018.

KONAIAGARI, A. *Teste de usuário: Introdução, benefícios, como criar e melhores práticas*. 2017. <<https://imasters.com.br/design-ux/teste-de-usuario-introducao-beneficios-como-criar-e-melhores-praticas>>. Acessado em 30/10/2018.

LACERDA, M. Informática como disciplina obrigatória na educação básica. In: *Anais do Encontro Virtual de Documentação em Software Livre e Congresso Internacional de Linguagem e Tecnologia Online*. Belo Horizonte, Brasil: [s.n.], 2012. v. 1, n. 1. <http://www.periodicos.letras.ufmg.br/index.php/anais_linguagem_tecnologia/article/view/1882>. Acessado em 03/11/2017.

LANGDON, D. et al. Stem: Good jobs now and for the future. *ESA Issue Brief #03-11*, US Department of Commerce, Jul 2011. <http://www.esa.doc.gov/sites/default/files/stemfinaljuly14_1.pdf>. Acessado em 21/09/2017.

LIMA, M. *Introdução aos Progressive Web Apps*. 2017. <<https://tableless.com.br/introducao-aos-progressive-web-apps/>>. Acessado em 02/12/2018.

MARCOS, V.; AMADOR, C. A gestão do voluntariado. *EMPREENDEDORISMO SOCIAL EM PORTUGAL*, UNIVERSIDADE DO PORTO, 2014. <<https://repositorio-aberto.up.pt/bitstream/10216/74145/2/88717.pdf>>. Acessado em 12/06/2018.

MARQUES, D. L. et al. Atraindo alunos do ensino médio para a computação: Uma experiência prática de introdução a programação utilizando jogos e python. In: *SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*. Aracajú, Brasil: [s.n.], 2011. p. 1138 – 1147. <<http://www.br-ie.org/pub/index.php/wie/article/view/1954/1713>>. Acessado em 20/11/2017.

MELLO, S. P. T. de et al. O fenômeno evasão nos cursos superiores de tecnologia: Um estudo de caso em uma universidade pública no sul do brasil. In: *Xiii Coloquio de Gestión Universitaria En Américas*. Arequipa, Peru: [s.n.], 2013. v. 0, n. 0, p. 1–10. <<https://repositorio.ufsc.br/bitstream/handle/123456789/113096/2013129>- O fenômeno evasão nos cursos

superiores.pdf?sequence=1&isAllowed=y>. Acessado em 01/09/2017.

MENDES, N. C. *Análise do Perfil Socioeconômico e Motivacional dos Voluntários da Copa do Mundo FIFA 2014*. Tese (BACHARELADO EM TURISMO) — UNIVERSIDADE DE BRASÍLIA, Brasília, Brasil, 2015. <http://www.bdm.unb.br/bitstream/10483/12986/1/2015_NathaliaCorreaMendes.pdf>. Acessado em 14/06/2018.

MORA, S. L. *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante, Espanha: Editorial Club Universitario, 2002.

MOZILLA. *Recursos de linguagem JavaScript*. 2016. <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/LanguageResources>>. Acessado em 30/10/2018.

ORACLE CORPORATION. *Immutable Objects*. 2014. <<https://docs.oracle.com/javase/tutorial/essential/concurrency/immutability.html>>. Acessado em 26/10/2018.

ORACLE CORPORATION. *MySQL / O Banco de Dados de Código Aberto mais Conhecido no Mundo*. 2018. <<https://www.oracle.com/br/mysql/index.html>>. Acessado em 16/06/2018.

PETERSEN K., F. R. M. S. . M. M. Systematic mapping studies in software engineering. In: *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*. Swindon, Reino Unido: [s.n.], 2008. p. 68–77. <http://www.robertfeldt.net/publications/petersen_ease08sysmap_studies_inse.pdf>

PIVOTAL. *Spring*. 2018. <<https://spring.io/>>. Acessado em 25/05/2018.

PIVOTAL. *Spring Docs*. 2018. <<https://docs.spring.io/>>. Acessado em 25/10/2018.

SAMARITAN TECHNOLOGIES. *Home Page*. 2018. <<https://samaritan.com/>>. Acessado em 16/06/2018.

SILVA, A. M. Dilemas do voluntariado na educação em Aracaju/SE. *IV Jornada Internacional de políticas públicas*, São Luís, Brasil, 2009. <http://www.joinpp.ufma.br/jornadas/joinppIV/OLD/eixos_OLD/11.%20Impasses%20e%20Desafios%20das%20Políticas%20de%20Educação%20em%20Aracaju.pdf>

%A7%C3%A3o/DILEMAS%20DO%20VOLUNTARIADO%20NA%20EDUCA%C3%87%C3%83O%20EM%20ARACAJU%20SE1.pdf>. Acessado em 16/06/2018.

TABLELESS. *Componentes alem do reuso*. 2018. <<https://tableless.com.br/componentes-alem-do-reuso/>>. Acessado em 30/10/2018.

TANENBAUM, A. S.; STEEN, M. V. *Distributed Systems: Principles and Paradigms*. Nova Jersey, EUA: Pearson, Prentice Hall, 2007.

TEAM KINECTIC. *Dashboard*. 2018. <<https://teamkinetic.co.uk/>>. Acessado em 16/06/2018.

TECH, H. ponto. *React: o framework onipresente*. out. 2017. 1 MP3 (48min). (Hipsters). Podcast. <<https://hipsters.tech/react-o-framework-onipresente-hipsters-66/>>. Acessado em 23/05/2018.

TEIXEIRA, F. *O que é o SUS (System Usability Scale) e como usá-lo em seu site*. 2015. <<https://brasil.uxdesign.cc/o-que-%C3%A9-o-sus-system-usability-scale-e-como-us%C3%A1-lo-em-seu-site-6d63224481c8>>. Acessado em 30/10/2018.

TENÓRIO, F. *Gestão de ongs: principais funções gerenciais*. Rio de Janeiro, Brasil: Ed.Fgv, 2001.

VALENTE, J. A. Diferentes usos do computador na educação. In: *Em Aberto*. Brasília, Brasil: [s.n.], 1993. v. 12, n. 53, p. 1–14. <<http://emaberto.inep.gov.br/index.php/emaberto/article/view/1876>>. Acessado em 21/07/2017.

VIDAL, F. A. B. et al. Gestão participativa e voluntariado: Sinais de uma racionalidade substantiva na administração de organizações do terceiro setor. In: *Anima - Revista da Faculdade Integrada do Ceará*. Fortaleza, Brasil: [s.n.], 2007. <<http://publica-estaciofic.com.br/edicoes/ANIMA/ANIMA-2007-N11.pdf#page=28>>. Acessado em 12/06/2018.

VOLGISTICS. *Dashboard*. 2018. <<https://www.volgistics.com/>>. Acessado em 16/06/2018.

VOLUNTEER HUB. *Home Page*. 2018. <<https://www.volunteerhub.com/>>. Acessado em 16/06/2018.

WIKIWEB. *Immutable Objects And Garbage Collection*. 2011. <<http://wiki.c2.com/?ImmutableObjectsAndGarbageCollection>>. Acessado em 26/10/2018.

WORDPRESS. *Home Page*. 2018. <<https://wordpress.org/>>. Acessado em 21/06/2018.

YAZBEK, M. C. Voluntariado e profissionalidade na intervenção social. *Revista Intervenção Social*, Brasil, 2002. <<http://www.periodicoseletronicos.ufma.br/index.php/rppublica/article/view/3718/1749>>. Acessado em 14/06/2018.

APÊNDICE A - Artigo

Processo de voluntariado dentro da iniciativa Computação na Escola

Luiz Ricardo Flores Maestrí¹, Jean Carlo Rossa Hauck²

¹Departamento de Informática e Estatística (DNE) – Universidade Federal de Santa Catarina (UFSC)

luiz.maestrif@grad.ufsc.br, jean.hauck@ufsc.br

Abstract. Currently, the market in technology is in great growth, however, in Brazil, there has not been perceived enough interest in the training of young people to work in the area. Initiatives such as Computação na Escola, which are being offered, to support basic education for children and young people, are born seeking to address this deficiency in basic education, yet managing the entire process of volunteering becomes complex and cumbersome, especially due to communication difficulties due to the physical distance between the volunteers and the institution. In this context, this work proposes the development and implementation of a web system in order to assist the management of the volunteer process.

Resumo. Atualmente, o mercado de tecnologia está em grande crescimento, no entanto, no Brasil não existe um grande interesse suficiente na capacitação de jovens para atuarem nessa área. Iniciativas como a Computação na Escola (CnE), que utilizam esforços voluntários para apoiar o ensino de computação para crianças e jovens, nasceram procurando atender essa deficiência na educação básica, todavia, gerenciar o processo de voluntariado se torna algo complexo e trabalhoso, principalmente pelas dificuldades de comunicação devido à troca de documentos por meio físico entre as entidades envolvidas.

1. Introdução

A demanda profissional nas áreas de Ciências, Tecnologia, Engenharia e Matemática está a crescer, nos Estados Unidos, de triplicar no período entre 2011 e 2013 [LANGDON 2011].

Mesmo com essa demanda crescente na área da TI, em 2015 um estudo divulgado [ABDIJL-ALIM 2015] aponta que alunos do ensino médio não seriam capacitados em sala de aula com a facilidade de ocupar essas vagas.

¹ <http://www.oefuturo.org.br/educação/nave/>
² <http://cscle.org>
³ <http://www.computacaonaescola.ufsc.br>

Análise e interpretação do estudo: Nesta etapa são analisados os dados obtidos perguntas/requisitos de pesquisa, usando métodos quantitativos e qualitativos. Ao final, os resultados são interpretados e discutidos.

4. Principais Conclusões

Existem diversas aplicações de para o auxílio do controle de voluntários, porém nem todas as ferramentas supracitadas podem atender a todos os requisitos elencados para o projeto, realçando na necessidade da implementação de um sistema específico para a Computação na Escola.

Referências

- ABDIJL-ALIM, J. "Students not prepared for careers in computer science." 2015. Disponível: <http://diversifiededucation.com/article/77408/>. Acesso Set. 2017.
Basili, V. R.; Caldera, G.; Ronħebek, H. D. "Goal Question Metric Paradigm. Encyclopedia of Software Engineering," John Wiley & Sons, 1994.
CNE, N. E. S. "Iniciando o seu jovem Turma de Programação." 2017. Disponível: <http://www.computacaonaescola.ufsc.br/p-2217>. Acesso Set. 2017.
KNOW HOW NON PROFIT (ORG). "HOW TO CHOOSE A VOLUNTEER MANAGEMENT SYSTEM." 2018. Disponível: <http://www.knowhownonprofit.org/how-to-choose-a-volunteer-management-system>. Acesso Set. 2017.
LANGDON, D. et al. "Smart: Good jobs now and for the future." 2011. Disponível: http://www.esadoc.gov/sites/default/files/semifinalJuly14_1.pdf. Acesso: Set. 2017.
TENÓRIO, F. Gestão de ongs: principais funções gerenciais. Rio de Janeiro, Brasil: EdFgv, 2001.

Iniciativas públicas-privadas como a escola NAVE¹ e voluntárias como code.org² e Computação na escola CnE³ surgiram com a intenção de levar o ensino da computação para crianças e jovens a um maior público.

A Iniciativa Computação na Escola tem utilizado o envolvimento de voluntários no curso de Computação para crianças entre 2011 e 2013 para auxiliar no encerramento do processo de voluntariado e complementar esse apoio de uma forma eficiente. Tipicamente o processo de voluntariado na Iniciativa Computação na Escola envolve:

- A CnE - o voluntário de alguma empresa parceira, a empresa e uma escola alvo.
- Envios de diversos documentos e Termos de Responsabilidade por e-mail/fax.
- Avaliações de perfil ao início do processo e avaliações de desempenho ao fim do mesmo.
- Agendamento das datas de treinamento e monitoria.

Para Tenório (2001) ONG (voluntárias) tem um "estilo próprio de gestão" construído ao longo de sua existência e demonstrando uma resistência aos modelos mais estruturados, apresentando dificuldades na execução de tarefas administrativas.

2. Contexto

A experiência relatada consiste na busca de ferramentas que possam auxiliar no gestão do processo de voluntariado. Com base em alguns requisitos elencados pela KnH, como o Profil (Org) [2018], que é o resultado de um levantamento feito por meio de entrevista com um dos coordenadores da CnE, requisitos não funcionais, são:

- RFI -1 - O sistema deve permitir o cadastro dos voluntários.
- RFI -2 - O sistema deve permitir a visualização de voluntários.
- RFI -3 - O sistema deve permitir o cadastro de eventos de voluntariado.
- RFI -4 - O sistema deve permitir a premiação de voluntários.
- RIS -5 - O sistema deve permitir o gerenciamento de treinamentos.
- RIS -6 - O sistema deve permitir a avaliação do voluntários.
- RIF -7 - O sistema deve permitir a visualização das avaliações dos voluntários.
- RNF1 - O sistema deve ser Integrado ao site da CnE.
- RNF2 - Gratuito e Open-Source.
- RNFM - Aplicação web.

3. Procedimentos metodológicos (coleção/análise dos dados)

A análise do estudo da arte 1 e realizada segundo os principais etapas do método de implementação descritivo de literatura definido por Pelsmér K. (2008). Esse método é realmente definindo o mapeamento da literatura disponível, executando a busca e fazendo a extração dos resultados, bem como sua análise.

Definição do estudo: O estudo é definido em termos do objetivo e perguntas de pesquisa e o design de pesquisa. A partir do objetivo e das perguntas de análise são sistematicamente derivados as metas para a coleta de dados utilizando o método QQM [Basil, Caldeira e Reutzbach 1994].

Figura 60 – Artigo

APÊNDICE B – Detalhamento dos casos de uso

Tabela 17 – Detalhamento de caso de uso: UC 2 - CRUD Evento.

Nome do Caso de Uso	UC 2 - CRUD Evento.
Pré-Condições	Existir escola já cadastrada e voluntários associados a escola
Atores Envolvidos	Responsável na CnE
Resumo	Aborda os passos necessários para o Responsável na CnE cadastrar um novo evento.
Cenário principal:	<ol style="list-style-type: none"> 1. O Responsável na CnE seleciona no calendário a data e e horários que o evento ocorrerá. 2. O sistema abre um modal com formulário, em forma de <i>wizard</i> com para o cadastro de evento, apresentando os campos relativos ao endereço como desabilitados. 3. O Responsável na CnE preenche os dados e seleciona a escola participante. 4. O Sistema preenche os dados de endereço com os dados da escola e habilita para edição. 5. O Responsável na CnE seleciona os voluntários participantes do evento dentre os associados a escola. 6. O Responsável na CnE seleciona documentos digitais para disponibilizar aos voluntários participantes.

Tabela 18 – Detalhamento de caso de uso: UC 3 - CRUD Projeto.

Nome do Caso de Uso	UC 3 - CRUD Projeto.
Pré-Condições	Sem precondições
Atores Envolvidos	Responsável na CnE
Resumo	Aborda os passos necessários para o Responsável na CnE Cadastrar um projeto
Cenário principal:	
1. O Responsável na CnE clica em "Administração" no canto superior direito.	
2. O sistema Abre a tela de administração.	
3. O Responsável na CnE clica em "Projetos".	
4. O Sistema apresenta a tela de projetos com uma listagem dos projetos existentes a esquerda e um formulário a direita.	
5. O Responsável preenche os dados e clica em "Salvar".	
6. O Sistema apresenta o novo projeto na listagem a direita e configura os campos do formulário para em branco.	

Tabela 19 – Detalhamento de caso de uso: UC 4 -Avaliar recomendação de voluntário.

Nome do Caso de Uso	UC 4 -Avaliar recomendação de voluntário.
Pré-Condições	Voluntário com o <i>status</i> "esperando recomendação da empresa"
Atores Envoltos	Responsável pela Empresa
Resumo	Aborda os procedimentos realizados pelo Responsável na Empresa para recomendar um voluntário.
Cenário principal:	
1.	O Responsável pela Empresa clica em "Recomendar"
2.	O sistema apresenta um modal com as respostas do voluntário ao questionário de cadastro.
3.	O Responsável pela Empresa clica em "Avançar".
4.	O sistema apresenta um questionário sobre a recomendação do voluntário.
5.	O Responsável pela Empresa preenche o questionário e clica em "Recomendar".
6.	O Sistema altera o <i>status</i> do voluntário para "aguardando envio do termo de responsabilidade".

Tabela 20 – Detalhamento de caso de uso: UC 5 - Emitir relatório de desempenho.

Nome do Caso de Uso	UC 5 - Emitir relatório de desempenho.
Pré-Condições	Existir evento avaliado.
Atores Envolvidos	Responsável na CnE.
Resumo	Aborda como o Responsável na CnE emite os relatórios sobre os voluntários.
Cenário principal:	
1. O Responsável na CnE clica em "Relatórios" no canto superior direito.	
2. O sistema apresenta a tela de relatórios com dois botões "Relatório por escola" e "Relatório por empresa".	
3. O Responsável na CnE clica em "Relatório por escola".	
4. O sistema apresenta o relatório de eventos separados por escola.	
Cenário alternativo: No passo 4 caso o o Responsável na CnE tenha clicado em "Relatório por empresa" no passo 3.	
4.1. O sistema apresenta o relatório de eventos separados por empresa.	

Tabela 21 – Detalhamento de caso de uso: UC 7 - CRUD escola.

Nome do Caso de Uso	UC 7 - CRUD escola.
Pré-Condições	Sem pré-condições
Atores Envolvidos	Responsável na CnE
Resumo	Aborda os passos realizados pelo Responsável na CnE para cadastrar uma escola beneficiada
Cenário principal:	
1. O Responsável na CnE clica no botão "+" acima da listagem de escolas beneficiadas.	
2. O sistema abre o formulário em forma de <i>wizard</i> de cadastro de escola beneficiada em um modal.	
3. O Responsável na CnE preenche os dados da escola e clica em "Salvar".	
4. O Sistema salva os dados e atualiza a listagem de escolas beneficiadas.	

Tabela 22 – Detalhamento de caso de uso: UC 8 - CRUD empresa.

Nome do Caso de Uso	UC 8 - CRUD empresa.
Pré-Condições	Sem pré-condições
Atores Envolvidos	Responsável na CnE
Resumo	Aborda os passos realizados pelo Responsável na CnE para cadastrar uma Empresa Parceira
Cenário principal:	
1. O Responsável na CnE clica no botão "+" acima da listagem de escolas beneficiadas.	
2. O sistema abre o formulário em forma de <i>wizard</i> de cadastro de escola beneficiada em um modal.	
3. O Responsável na CnE preenche os dados da escola e clica em "Salvar"	
4. O Sistema salva os dados e atualiza a listagem de escolas beneficiadas.	

Tabela 23 – Detalhamento de caso de uso: UC 11 - CRUD Voluntário.

Nome do Caso de Uso	UC 11 - CRUD Voluntário.
Pré-Condições	Ter Empresa parceira cadastrada.
Atores Envolvidos	Voluntário.
Resumo	Aborda os passos realizados pelo Voluntário
Cenário principal:	
1. O Voluntário clica em "Quero ser voluntário".	
2. O sistema apresenta o formulário de cadastro de voluntário.	
3. O Voluntário preenche seus dados e clica em cadastrar.	
4. O sistema salva os dados envia um e-mail e realiza o <i>login</i> do voluntário.	
5. O sistema apresenta a tela para envio de documentos.	
6. O Voluntário realiza o <i>upload</i> dos documentos.	
7. Após o último <i>upload</i> o sistema avisa o Voluntário por um <i>pop-up</i> e altera o <i>status</i> para "aguardando recomendação da empresa".	

Tabela 24 – Detalhamento de caso de uso: UC 14 - CRUD Tipo de evento.

Nome do Caso de Uso	UC 14 - CRUD Tipo de evento.
Pré-Condições	Sem precondições
Atores Envolvidos	Responsável na CnE
Resumo	Aborda os passos necessários para o Responsável na CnE Cadastrar um tipo de evento
Cenário principal:	<ol style="list-style-type: none"> 1. O Responsável na CnE clica em "Administração" no canto superior direito. 2. O sistema Abre a tela de administração. 3. O Responsável na CnE clica em "Tipo de Evento". 4. O Sistema apresenta a tela de projetos com uma listagem dos tipo de evento existentes a esquerda e um formulário a direita. 5. O Responsável empresa preenche os dados e clica em "Salvar". 6. O Sistema apresenta o novo projeto na listagem a direita e configura os campos do formulário para em branco.

APÊNDICE C – Diagramas de Classe

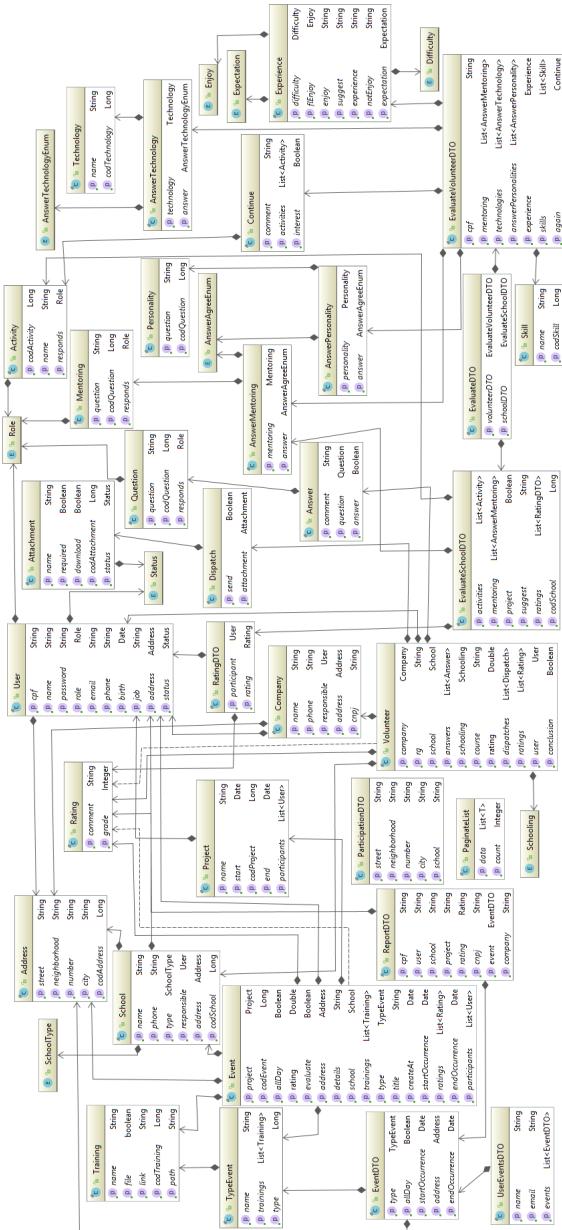


Figura 61 – Diagrama de classes - classes de modelo e DTOs.

APÊNDICE D - Código

D.1 BACKEND

```
1
2
3 src/main/java/br/ufsc/cultivar/dto/ParticipationDTO.java
4
5 package br.ufsc.cultivar.dto;
6
7 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
8 import lombok.AccessLevel;
9 import lombok.Builder;
10 import lombok.Value;
11 import lombok.experimental.FieldDefaults;
12
13 @Value
14 @Builder(builderClassName = "Builder")
15 @JsonDeserialize(builder = ParticipationDTO.Builder.class)
16 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
17 public class ParticipationDTO {
18     String city;
19     String neighborhood;
20     String street;
21     String number;
22     String school;
23 }
24
25
26 src/main/java/br/ufsc/cultivar/dto/EvaluateDTO.java
27
28 package br.ufsc.cultivar.dto;
29
30 import br.ufsc.cultivar.model.Answer;
31 import br.ufsc.cultivar.model.evaluate.AnswerTechnology;
32 import br.ufsc.cultivar.model.evaluate.AnswerPersonality;
33 import br.ufsc.cultivar.model.evaluate.Experience;
34 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
35 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
36 import lombok.AccessLevel;
37 import lombok.Builder;
38 import lombok.Value;
39 import lombok.experimental.FieldDefaults;
40
41 import java.util.List;
```

```
42
43 @Value
44 @Builder(builderClassName = "Builder")
45 @JsonDeserialize(builder = EvaluateDTO.Builder.class)
46 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
47 public class EvaluateDTO {
48
49     String cpf;
50     Long project;
51     List<AnswerTechnology> technologies;
52     List<AnswerPersonality> answerPersonalities;
53     Experience experience;
54
55     @JsonPOJOBuilder(withPrefix = "")
56     public static class Builder{
57         public Builder(){}
58     }
59 }
60
61
62 src/main/java/br/ufsc/cultivar/dto/EventDTO.java
63
64 package br.ufsc.cultivar.dto;
65
66 import br.ufsc.cultivar.model.Address;
67 import br.ufsc.cultivar.model.TypeEvent;
68 import lombok.AccessLevel;
69 import lombok.Builder;
70 import lombok.Value;
71 import lombok.experimental.FieldDefaults;
72
73 import java.util.Date;
74
75 @Value
76 @Builder
77 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
78 public class EventDTO {
79     TypeEvent type;
80     Address address;
81     Date startOccurrence;
82     Date endOccurrence;
83     Boolean allDay;
84 }
85
86
```

```
87 src/main/java/br/ufsc/cultivar/dto/PaginateList.java
88
89 package br.ufsc.cultivar.dto;
90
91 import br.ufsc.cultivar.model.Answer;
92 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
93 import lombok.AccessLevel;
94 import lombok.Builder;
95 import lombok.Value;
96 import lombok.experimental.FieldDefaults;
97
98 import java.util.List;
99
100 @Value
101 @Builder(builderClassName = "Builder")
102 @JsonDeserialize(builder = PaginateList.Builder.class)
103 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
104 public class PaginateList<T> {
105     Integer count;
106     List<T> data;
107 }
108
109
110 src/main/java/br/ufsc/cultivar/dto/UserEventsDTO.java
111
112 package br.ufsc.cultivar.dto;
113
114 import lombok.AccessLevel;
115 import lombok.Builder;
116 import lombok.Value;
117 import lombok.experimental.FieldDefaults;
118
119 import java.util.List;
120
121 @Value
122 @Builder
123 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
124 public class UserEventsDTO {
125     String email;
126     String name;
127     List<EventDTO> events;
128 }
129
130
131 src/main/java/br/ufsc/cultivar/repository/evaluate/PersonalityRepository.java
```

```
132  
133 package br.ufsc.cultivar.repository.evaluate;  
134  
135 import br.ufsc.cultivar.model.evaluate.Personality;  
136 import br.ufsc.cultivar.utils.DatabaseUtils;  
137 import lombok.AccessLevel;  
138 import lombok.AllArgsConstructorConstructor;  
139 import lombok.experimental.FieldDefaults;  
140 import org.springframework.beans.factory.annotation.Autowired;  
141 import  
    org.springframework.jdbc.core.namedparam.MapSqlParameterSource;  
142 import  
    org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;  
143 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;  
144 import org.springframework.stereotype.Repository;  
145  
146 import java.sql.ResultSet;  
147 import java.sql.SQLException;  
148 import java.util.List;  
149  
150 @Repository  
151 @AllArgsConstructor(onConstructor = @__(@Autowired))  
152 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
153 public class PersonalityRepository {  
154  
    NamedParameterJdbcTemplate jdbcTemplate;  
155  
    public void create(Personality question) {  
        new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())  
            .withTableName("personality")  
            .usingGeneratedKeyColumns("cod_question")  
            .execute(  
                new MapSqlParameterSource("dsc_question",  
                    question.getQuestion())  
            );  
    }  
165  
    public List<Personality> get() {  
        return jdbcTemplate.query(  
            "select * from personality",  
            (rs, i) -> this.build(rs)  
        );  
    }  
172  
    public Personality get(Long codQuestion) {
```

```
174     return jdbcTemplate.query(
175         "select * from personality where
176             cod_question=:cod_question",
177         new MapSqlParameterSource("cod_question",
178             codQuestion),
179         this::build
180     );
181 }
182
183 private Personality build(ResultSet rs) throws SQLException {
184     if(!DatabaseUtils.isNotEmpty(rs)){
185         return null;
186     }
187     return Personality.builder()
188         .codQuestion(rs.getLong("cod_question"))
189         .question(rs.getString("dsc_question"))
190         .build();
191 }
192
193 public void delete(Long codQuestion) {
194     jdbcTemplate.update(
195         "delete from personality where
196             cod_question=:cod_question",
197         new MapSqlParameterSource("cod_question",
198             codQuestion)
199     );
200 }
201
202 }
203
204 src/main/java/br/ufsc/cultivar/repository/evaluate/MentoringRepository.java
205
206 package br.ufsc.cultivar.repository.evaluate;
207
208 import br.ufsc.cultivar.model.evaluate.Mentoring;
209 import br.ufsc.cultivar.utils.DatabaseUtils;
210 import lombok.AccessLevel;
211 import lombok.NoArgsConstructor;
212 import lombok.experimental.FieldDefaults;
213 import org.springframework.beans.factory.annotation.Autowired;
214 import
215     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
216 import
217     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
218 import
219     org.springframework.jdbc.core.simple.SimpleJdbcInsert;
```

```
213 import org.springframework.stereotype.Repository;  
214  
215 import java.sql.ResultSet;  
216 import java.sql.SQLException;  
217 import java.util.List;  
218  
219 @Repository  
220 @AllArgsConstructor(onConstructor = @_(@Autowired))  
221 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
222 public class MentoringRepository {  
223     NamedParameterJdbcTemplate jdbcTemplate;  
224  
225     public void create(Mentoring mentoring){  
226         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())  
227             .withTableName("mentoring")  
228             .usingGeneratedKeyColumns("cod_question")  
229             .execute(  
230                 new MapSqlParameterSource("dsc_question",  
231                     mentoring.getQuestion())  
232             );  
233     }  
234  
235     public List<Mentoring> get(){  
236         return jdbcTemplate.query(  
237             "select * from mentoring",  
238             (rs, i) -> this.build(rs)  
239         );  
240     }  
241  
242     public Mentoring get(Long codQuestion){  
243         return jdbcTemplate.query(  
244             "select * from mentoring where  
245                 cod_question=:cod_question",  
246             new MapSqlParameterSource("cod_question",  
247                 codQuestion),  
248             this::build  
249         );  
250     }  
251  
252     public void delete(Long codQuestion){  
253         jdbcTemplate.update(  
254             "delete from mentoring where  
255                 cod_question=:cod_question",  
256             new MapSqlParameterSource("cod_question",  
257                 codQuestion)
```

```
253     );
254 }
255
256     private Mentoring build(ResultSet rs) throws SQLException {
257         if(!DatabaseUtils.isNotEmpty(rs)){
258             return null;
259         }
260         return Mentoring.builder()
261             .codQuestion(rs.getLong("cod_question"))
262             .question(rs.getString("dsc_question"))
263             .build();
264     }
265 }
266
267
268 src/main/java/br/ufsc/cultivar/repository/evaluate/SkillRepository.java
269
270 package br.ufsc.cultivar.repository.evaluate;
271
272 import br.ufsc.cultivar.model.evaluate.Skill;
273 import br.ufsc.cultivar.utils.DatabaseUtils;
274 import lombok.AccessLevel;
275 import lombok.AllArgsConstructor;
276 import lombok.experimental.FieldDefaults;
277 import org.springframework.beans.factory.annotation.Autowired;
278 import
279     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
280 import
281     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
282 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
283 import org.springframework.stereotype.Repository;
284
285 import java.sql.ResultSet;
286 import java.sql.SQLException;
287 import java.util.List;
288
289 @Repository
290 @AllArgsConstructor(onConstructor = @_(@Autowired))
291 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
292 public class SkillRepository {
293     NamedParameterJdbcTemplate jdbcTemplate;
294
295     public void create(Skill skill){
296         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
297             .withTableName("skill")
```

```
296         .usingGeneratedKeyColumns("cod_skill")
297         .execute(
298             new MapSqlParameterSource("nm_skill",
299                 skill.getName())
300         );
301     }
302
303     public Skill get(Long codSkill){
304         return jdbcTemplate.query(
305             "select * from skill where cod_skill=:cod_skill",
306             new MapSqlParameterSource("cod_skill", codSkill),
307             this::build
308         );
309     }
310
311     public List<Skill> get(){
312         return jdbcTemplate.query(
313             "select * from skill",
314             (rs, i) -> this.build(rs)
315         );
316     }
317
318     public void delete(Long codSkill){
319         jdbcTemplate.update(
320             "delete from skill where cod_skill=:cod_skill",
321             new MapSqlParameterSource("cod_skill", codSkill)
322         );
323     }
324
325     private Skill build(ResultSet rs) throws SQLException {
326         if (!DatabaseUtils.isNotEmpty(rs)){
327             return null;
328         }
329         return Skill.builder()
330             .codSkill(rs.getLong("cod_skill"))
331             .name(rs.getString("nm_skill"))
332             .build();
333     }
334
335
336     src/main/java/br/ufsc/cultivar/repository/evaluate/EvaluateRepository.java
337
338     package br.ufsc.cultivar.repository.evaluate;
339
```

```
340 import br.ufsc.cultivar.model.evaluate.*;
341 import lombok.AccessLevel;
342 import lombok.AllArgsConstructor;
343 import lombok.experimental.FieldDefaults;
344 import lombok.val;
345 import org.springframework.beans.factory.annotation.Autowired;
346 import
347     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
348 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
349 import org.springframework.stereotype.Repository;
350
351 import java.util.List;
352
353 @Repository
354 @AllArgsConstructor(onConstructor = @_(@Autowired))
355 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
356 public class EvaluateRepository {
357
358     NamedParameterJdbcTemplate jdbcTemplate;
359
360     public void saveTechnology(AnswerTechnology
361         answerTechnology, Long codProject, String cpf) {
362         val technology = answerTechnology.getTechnology();
363         val answer = answerTechnology.getAnswer();
364         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
365             .withTableName("answer_technology")
366             .execute(
367                 new MapSqlParameterSource()
368                     .addValue("cod_technology",
369                         technology.getCodTechnology())
370                     .addValue("cod_project", codProject)
371                     .addValue("cod_cpf", cpf)
372                     .addValue("dsc_answer", answer.name())
373             );
374     }
375
376     public void saveAnswerVolunteer(AnswerPersonality
377         answerPersonality, Long codProject, String cpf) {
378         val question = answerPersonality.getQuestion();
379         val answer = answerPersonality.getAnswer();
380         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
381             .withTableName("answer_technology")
382             .execute(
```

```
380     new MapSqlParameterSource()
381         .addValue("cod_question",
382             question.getCodQuestion())
383         .addValue("cod_project", codProject)
384         .addValue("cod_cpf", cpf)
385         .addValue("dsc_answer", answer.name())
386     );
387
388     public void saveExperience(Experience experience, Long
389         codProject, String cpf) {
390         val difficulty = experience.getDifficulty();
391         val enjoy = experience.getFlEnjoy();
392         val expectation = experience.getExpectation();
393         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
394             .withTableName("experience")
395             .execute(
396                 new MapSqlParameterSource()
397                     .addValue("cod_project", codProject)
398                     .addValue("cod_cpf", cpf)
399                     .addValue("dsc_experience",
400                         experience.getExperience())
401                     .addValue("dsc_difficulty", difficulty.name())
402                     .addValue("fl_enjoy", enjoy.name())
403                     .addValue("dsc_enjoy", experience.getEnjoy())
404                     .addValue("dsc_not_enjoy",
405                         experience.getNotEnjoy())
406                     .addValue("dsc_suggest",
407                         experience.getSuggest())
408                     .addValue("dsc_expectation",
409                         expectation.name())
410             );
411     }
412
413     public List<String> getCpfs(Long codProject) {
414         return jdbcTemplate.queryForList(
415             "select cod_cpf from experience where
416             cod_project=:cod_project",
417             new MapSqlParameterSource("cod_project",
418                 codProject),
419             String.class
420         );
421     }
422 }
```

```

416     public List<AnswerPersonality> getAnswersVolunteer(Long
417         codProject, String cpf) {
418         return jdbcTemplate.query(
419             "select av.dsc_answer, vq.* from
420             answer_personality av natural join
421             personality vq where cod_project=:cod_project
422             and cod_cpf=:cod_cpf",
423             new MapSqlParameterSource("cod_project",
424                 codProject).addValue("cod_cpf", cpf),
425             (rs, i) ->
426                 AnswerPersonality.builder()
427                     .question(
428                         Personality.builder()
429                             .codQuestion(rs.getLong("cod_question"))
430                             .question(rs.getString("dsc_question"))
431                             .build()
432                     ).answer(
433                         AnswerPersonalityEnum
434                             .valueOf(rs.getString("dsc_answer"))
435                     ).build()
436             );
437     }
438
439     public List<AnswerTechnology> getAnswersTechnologies(Long
440         codProject, String cpf) {
441         return jdbcTemplate.query(
442             "select at.dsc_answer ,t.* from answer_technology at
443             natural join technology t where
444             cod_project=:cod_project and cod_cpf=:cod_cpf",
445             new MapSqlParameterSource("cod_project",
446                 codProject).addValue("cod_cpf", cpf),
447             (rs, i) ->
448                 AnswerTechnology.builder()
449                     .technology(
450                         Technology.builder()
451                             .name("")
452                             .build()
453                     )
454                     .build()
455             );
456     }
457 }
458
459 src/main/java/br/ufsc/cultivar/repository/evaluate/TechnologyRepository.java

```

```
452  
453 package br.ufsc.cultivar.repository.evaluate;  
454  
455 import br.ufsc.cultivar.model.evaluate.Technology;  
456 import br.ufsc.cultivar.utils.DatabaseUtils;  
457 import lombok.AccessLevel;  
458 import lombok.AllArgsConstructorConstructor;  
459 import lombok.experimental.FieldDefaults;  
460 import org.springframework.beans.factory.annotation.Autowired;  
461 import  
        org.springframework.jdbc.core.namedparam.MapSqlParameterSource;  
462 import  
        org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;  
463 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;  
464 import org.springframework.stereotype.Repository;  
465  
466 import java.sql.ResultSet;  
467 import java.sql.SQLException;  
468 import java.util.List;  
469  
470 @Repository  
471 @AllArgsConstructor(onConstructor = @_(@Autowired))  
472 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
473 public class TechnologyRepository {  
    NamedParameterJdbcTemplate jdbcTemplate;  
475  
476     public void create(Technology technology) {  
        new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())  
            .withTableName("technology")  
            .usingGeneratedKeyColumns("cod_technology")  
            .execute(  
                new MapSqlParameterSource("nm_technology",  
                    technology.getName())  
            );  
    }  
484  
485     public List<Technology> get() {  
        return jdbcTemplate.query("select * from technology",  
            (rs, i) -> this.build(rs));  
    }  
488  
489     private Technology build(ResultSet rs) throws SQLException {  
        if(!DatabaseUtils.isNotEmpty(rs)){  
            return null;  
        }  
492 }
```

```
493     return Technology.builder()
494         .codTechnology(rs.getLong("cod_technology"))
495         .name(rs.getString("nm_technology"))
496         .build();
497     }
498
499     public void delete(Long codTechnology) {
500         jdbcTemplate.update(
501             "delete from technology where"
502                 "cod_technology=:cod_technology",
503                 new MapSqlParameterSource("cod_technology",
504                     codTechnology)
505             );
506     }
507
508     public Technology get(Long codTechnology) {
509         return jdbcTemplate.query(
510             "select * from technology where"
511                 "cod_technology=:cod_technology",
512                 new MapSqlParameterSource("cod_technology",
513                     codTechnology),
514                     this::build
515             );
516     }
517
518 package br.ufsc.cultivar.repository;
519
520 import br.ufsc.cultivar.model.Training;
521 import br.ufsc.cultivar.utils.DatabaseUtils;
522 import lombok.AccessLevel;
523 import lombok.NoArgsConstructor;
524 import lombok.experimental.FieldDefaults;
525 import org.springframework.beans.factory.annotation.Autowired;
526 import
527     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
528 import
529     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
530 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
531 import org.springframework.stereotype.Repository;
532
533 import java.sql.ResultSet;
```

```
532 import java.sql.SQLException;
533 import java.util.List;
534
535 @Repository
536 @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
537 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
538 public class TrainingRepository {
539
540     NamedParameterJdbcTemplate jdbcTemplate;
541
542
543     public Long create(final Training training, final Long
544                         codEvent) {
545         return new
546             SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
547                 .withTableName("training")
548                 .usingGeneratedKeyColumns("cod_training")
549                 .executeAndReturnKey(
550                     new MapSqlParameterSource()
551                         .addValue("nm_training",
552                             training.getName())
553                         .addValue("dsc_path",
554                             training.getPath())
555                         .addValue("dsc_link",
556                             training.getLink())
557                 ).longValue();
558     }
559
560
561     public List<Training> getByTypeEvent(final Long tpEvent) {
562         return jdbcTemplate.query(
563             "select t.* from training t natural join
564             type_event_training where tp_event=:tp_event",
565             new MapSqlParameterSource("tp_event", tpEvent),
566             (rs, i) -> this.build(rs)
567         );
568     }
569
570
571     public List<Training> getByEvent(final Long codEvent) {
572         return jdbcTemplate.query(
573             "select t.* from training t natural join
574             event_training where cod_event=:cod_event",
575             new MapSqlParameterSource("cod_event", codEvent),
576             (rs, i) -> this.build(rs)
577         );
578     }
579 }
```

```
570
571     public Training get(final Long codTraining) {
572         return jdbcTemplate.query(
573             "select * from training where",
574             "cod_training=:cod_training",
575             new MapSqlParameterSource("cod_training",
576             codTraining),
577             this::build
578         );
579     }
580
581     public void deleteByEvent(final Long tpEvent) {
582         jdbcTemplate.update(
583             "select * from training where tp_event=:tp_event",
584             new MapSqlParameterSource("tp_event", tpEvent)
585         );
586     }
587
588     private Training build(ResultSet rs) throws SQLException {
589         if(!DatabaseUtils.isNotEmpty(rs)){
590             return null;
591         }
592         return Training.builder()
593             .codTraining(rs.getLong("cod_training"))
594             .name(rs.getString("nm_training"))
595             .path(rs.getString("dsc_path"))
596             .link(rs.getString("dsc_link"))
597             .build();
598     }
599 }
600
601
602 src/main/java/br/ufsc/cultivar/repository/QuestionRepository.java
603
604 package br.ufsc.cultivar.repository;
605
606 import br.ufsc.cultivar.model.Question;
607 import br.ufsc.cultivar.model.Role;
608 import br.ufsc.cultivar.utils.DatabaseUtils;
609 import lombok.AccessLevel;
610 import lombok.AllArgsConstructor;
611 import lombok.experimental.FieldDefaults;
612 import org.springframework.beans.factory.annotation.Autowired;
```

```
613 import
614     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
615 import
616     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
617 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
618 import org.springframework.stereotype.Repository;
619 import java.sql.ResultSet;
620 import java.sql.SQLException;
621 import java.util.List;
622
623 @Repository
624 @AllArgsConstructor(onConstructor = @__(@Autowired))
625 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
626 public class QuestionRepository {
627
628     NamedParameterJdbcTemplate jdbcTemplate;
629
630     public void create(final Question question) {
631         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
632             .withTableName("question")
633             .usingGeneratedKeyColumns("cod_question")
634             .execute(
635                 new MapSqlParameterSource("dsc_question",
636                     question.getQuestion())
637                     .addValue("dsc_responds",
638                         question.getResponds().name())
639             );
640     }
641
642     public List<Question> get() {
643         return jdbcTemplate.query(
644             "select * from question",
645             (rs, i) -> this.build(rs)
646         );
647     }
648
649     public List<Question> get(final Role responds) {
650         return jdbcTemplate.query(
651             "select * from question where
652                 dsc_responds=:dsc_responds",
653             new MapSqlParameterSource("dsc_responds",
654                 responds.name()),
655             (rs, i) -> this.build(rs)
656         );
657     }
658 }
```

```
652     }
653
654     public Question get(final Long codQuestion) {
655         return jdbcTemplate.query(
656             "select * from question where
657                 cod_question=:cod_question",
658             new MapSqlParameterSource("cod_question",
659                 codQuestion),
660             this::build
661         );
662     }
663
664     public void delete(Long codQuestion) {
665         jdbcTemplate.update(
666             "delete from question where
667                 cod_question=:cod_question",
668             new MapSqlParameterSource("cod_question",
669                 codQuestion)
670         );
671     }
672
673     public void update(Question question) {
674         jdbcTemplate.update(
675             "update question set dsc_question=:dsc_question
676                 where cod_question=:cod_question",
677             new MapSqlParameterSource("dsc_question",
678                 question.getQuestion())
679                 .addValue("cod_question",
680                     question.getQuestion())
681         );
682     }
683
684     private Question build(ResultSet rs) throws SQLException {
685         if(!DatabaseUtils.isNotEmpty(rs)){
686             return null;
687         }
688         return Question.builder()
689             .codQuestion(rs.getLong("cod_question"))
690             .question(rs.getString("dsc_question"))
691             .responds(
692                 Role.valueOf(rs.getString("dsc_responds"))
693             )
694             .build();
695     }
696 }
```

```
690
691
692 src/main/java/br/ufsc/cultivar/repository/CompanyRepository.java
693
694 package br.ufsc.cultivar.repository;
695
696 import br.ufsc.cultivar.model.Address;
697 import br.ufsc.cultivar.model.Company;
698 import br.ufsc.cultivar.model.User;
699 import br.ufsc.cultivar.utils.DatabaseUtils;
700 import lombok.AccessLevel;
701 import lombok.AllArgsConstructor;
702 import lombok.experimental.FieldDefaults;
703 import lombok.val;
704 import org.springframework.beans.factory.annotation.Autowired;
705 import
    org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
706 import
    org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
707 import
    org.springframework.jdbc.core.namedparam.SqlParameterSource;
708 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
709 import org.springframework.stereotype.Repository;
710
711 import java.sql.ResultSet;
712 import java.sql.SQLException;
713 import java.util.List;
714 import java.util.Map;
715 import java.util.Objects;
716 import java.util.Optional;
717
718 @Repository
719 @AllArgsConstructor(onConstructor = @___(@Autowired))
720 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
721 public class CompanyRepository {
722
723     NamedParameterJdbcTemplate jdbcTemplate;
724
725     public void create(final Company company) {
726         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
727             .withTableName("company")
728             .execute(getParams(company));
729     }
730
```

```

731     public List<Company> get(final String filter, final Long
732         page) {
733         val sql = new StringBuilder("select * from company");
734         val params = new MapSqlParameterSource();
735         Optional.ofNullable(filter)
736             .ifPresent(
737                 s -> {
738                     sql.append(" where nm_company like
739                         :nm_company");
740                     params.addValue("nm_company", filter + "%");
741                 }
742             );
743         Optional.ofNullable(page)
744             .ifPresent(
745                 aLong -> {
746                     sql.append(" limit 5 offset :offset");
747                     params.addValue("offset", page*5);
748                 }
749             );
750         return jdbcTemplate.query(
751             sql.toString(),
752             params,
753             (rs, i) -> this.build(rs)
754         );
755     }
756
757     public Integer count(final String filter) {
758         val sql = new StringBuilder("select count(cod_cnpj) from
759             company");
760         val params = new MapSqlParameterSource();
761         Optional.ofNullable(filter)
762             .ifPresent(s -> {
763                 sql.append(" where nm_company like :nm_company");
764                 params.addValue("nm_company", filter + "%");
765             }
766             );
767         return jdbcTemplate.queryForObject(sql.toString(),
768             params, Integer.class);
769     }
770
771     public Company get(final String cnpj) {
772         return jdbcTemplate.query(
773             "select * from company where cod_cnpj=:cod_cnpj",
774             new MapSqlParameterSource("cod_cnpj", cnpj),
775             this::build
776     }

```

```
772         );
773     }
774
775     public void delete(final String cnpj) {
776         jdbcTemplate.update(
777             "delete from company where cod_cnpj=:cod_cnpj",
778             new MapSqlParameterSource("cod_cnpj", cnpj)
779         );
780     }
781
782     public void update(final Company company) {
783         jdbcTemplate.update(
784             "update company set nm_company=:nm_company,
785                 dsc_phone=:dsc_phone,
786                 cod_address=:cod_address," +
787                 "cod_cpf=:cod_cpf where
788                 cod_cnpj=:cod_cnpj",
789             getParams(company)
790         );
791     }
792
793     private SqlParameterSource getParams(final Company company) {
794         return new MapSqlParameterSource()
795             .addValue("cod_cnpj", company.getCnpj())
796             .addValue("nm_company", company.getName())
797             .addValue("dsc_phone", company.getPhone())
798             .addValue("cod_address",
799                 company.getAddress().getCodAddress())
800             .addValue("cod_cpf",
801                 company.getResponsible().getCpf());
802     }
803
804     private Company build(final ResultSet rs) throws
805         SQLException {
806         if(!DatabaseUtils.isNotEmpty(rs)){
807             return null;
808         }
809         return Company.builder()
810             .cnpj(rs.getString("cod_cnpj"))
811             .name(rs.getString("nm_company"))
812             .phone(rs.getString("dsc_phone"))
813             .address(
814                 Address.builder()
815                     .codAddress(rs.getLong("cod_address"))
816                     .build()
817             )
818     }
819 }
```

```
811         )
812         .responsible(
813             User.builder()
814                 .cpf(rs.getString("cod_cpf"))
815                 .build()
816         )
817         .build();
818     }
819 }
820
821
822 src/main/java/br/ufsc/cultivar/repository/DispatchRepository.java
823
824 package br.ufsc.cultivar.repository;
825
826 import br.ufsc.cultivar.model.Attachment;
827 import br.ufsc.cultivar.model.Dispatch;
828 import br.ufsc.cultivar.utils.DatabaseUtils;
829 import lombok.AccessLevel;
830 import lombok.AllArgsConstructor;
831 import lombok.experimental.FieldDefaults;
832 import org.springframework.beans.factory.annotation.Autowired;
833 import
834     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
835 import
836     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
837 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
838 import org.springframework.stereotype.Repository;
839
840 import java.sql.ResultSet;
841 import java.sql.SQLException;
842 import java.util.List;
843
844 @Repository
845 @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
846 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
847 public class DispatchRepository {
848
849     NamedParameterJdbcTemplate jdbcTemplate;
850
851     public void save(Dispatch dispatch, String cpf) {
852         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
853             .withTableName("dispatch")
854             .execute()
```

```

854             new MapSqlParameterSource()
855                 .addValue("cod_cpf", cpf)
856                 .addValue("cod_attachment",
857                     dispatch.getAttachment().getCodAttachment())
858                 .addValue("fl_dispatch",
859                     dispatch.getSend())
860         );
861     }
862
863     public Dispatch get(String cpf, Long codAttachment) {
864         return jdbcTemplate.query(
865             "select * from dispatch where cod_cpf=:cod_cpf and
866             cod_attachment=:cod_attachment",
867             new MapSqlParameterSource()
868                 .addValue("cod_cpf", cpf)
869                 .addValue("cod_attachment", codAttachment),
870             this::build
871         );
872     }
873
874     public List<Dispatch> get(String cpf) {
875         return jdbcTemplate.query(
876             "select * from dispatch where cod_cpf=:cod_cpf",
877             new MapSqlParameterSource()
878                 .addValue("cod_cpf", cpf),
879             (rs, i) -> this.build(rs)
880         );
881     }
882
883     private Dispatch build(ResultSet rs) throws SQLException {
884         if(!DatabaseUtils.isNotEmpty(rs)){
885             return null;
886         }
887         return Dispatch.builder()
888             .attachment(
889                 Attachment.builder()
890                     .codAttachment(rs.getLong("cod_attachment"))
891                     .build()
892             ).send(rs.getBoolean("fl_dispatch"))
893             .build();
894     }
895 }

src/main/java/br/ufsc/cultivar/repository/AttachmentRepository.java

```

```
896
897 package br.ufsc.cultivar.repository;
898
899 import br.ufsc.cultivar.model.Attachment;
900 import br.ufsc.cultivar.model.Status;
901 import lombok.AccessLevel;
902 import lombok.AllArgsConstructorConstructor;
903 import lombok.experimental.FieldDefaults;
904 import lombok.val;
905 import org.springframework.beans.factory.annotation.Autowired;
906 import
907     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
908 import
909     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
910 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
911 import org.springframework.stereotype.Repository;
912
913 import java.sql.ResultSet;
914 import java.sql.SQLException;
915 import java.util.List;
916 import java.util.Optional;
917
918 @Repository
919 @AllArgsConstructorConstructor(onConstructor = @___(@Autowired))
920 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
921 public class AttachmentRepository {
922
923     NamedParameterJdbcTemplate jdbcTemplate;
924
925     public long create(final Attachment attachment) {
926         return new
927             SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
928                 .withTableName("attachment")
929                 .usingGeneratedKeyColumns("cod_attachment")
930                 .executeAndReturnKey(getParams(attachment))
931                 .longValue();
932     }
933
934     public List<Attachment> get(final String filter, final Long
935         page) {
936         val sql = new StringBuilder("select * from attachment ");
937         val params = new MapSqlParameterSource();
938         Optional.ofNullable(filter)
939             .ifPresent(
940                 s -> {
```

```

937             sql.append("where nm_attachment like
938                 :nm_attachment ");
939             params.addValue("nm_attachment", filter + "%");
940         }
941     );
942     sql.append("limit 20 offset :offset");
943     params.addValue("offset", page*20);
944     return jdbcTemplate.query(sql.toString(), params, (rs,
945         i) -> this.build(rs));
946 }
947
948 public Integer count(final String filter){
949     val sql = new StringBuilder("select
950         count(cod_attachment) from attachment ");
951     val params = new MapSqlParameterSource();
952     Optional.ofNullable(filter)
953         .ifPresent(
954             s -> {
955                 sql.append("where nm_attachment like
956                     :nm_attachment ");
957                 params.addValue("nm_attachment", filter + "%");
958             }
959         );
960     return jdbcTemplate.queryForObject(sql.toString(),
961         params, Integer.class);
962 }
963
964 public Attachment get(final Long codAttachment) {
965     return jdbcTemplate.query(
966         "select * from attachment where
967             cod_attachment=:cod_attachment",
968         new MapSqlParameterSource("cod_attachment",
969             codAttachment),
970         this::build
971     );
972 }
973
974 public List<Attachment> get(final Status status) {
975     return jdbcTemplate.query(
976         "select * from attachment where
977             sta_user=:sta_user",
978         new MapSqlParameterSource("sta_user",
979             status.name()),
980         (rs, i) -> this.build(rs)
981     );
982 }
```

```
973
974     public void delete(Long codAttachment) {
975         jdbcTemplate.update(
976             "delete from attachment where
977                 cod_attachment=:cod_attachment",
978             new MapSqlParameterSource("cod_attachment",
979                 codAttachment)
980         );
981     }
982
983     public void update(final Attachment attachment) {
984         jdbcTemplate.update(
985             "update attachment set
986                 nm_attachment=:nm_attachment,
987                 fl_required=:fl_required "
988             + "where cod_attachment=:cod_attachment",
989             getParams(attachment).addValue("cod_attachment",
990                 attachment.getCodAttachment())
991         );
992     }
993
994     private Attachment build(ResultSet rs) throws SQLException {
995         if(rs.isBeforeFirst()){
996             rs.first();
997         }
998         return Attachment.builder()
999             .codAttachment(rs.getLong("cod_attachment"))
1000            .name(rs.getString("nm_attachment"))
1001            .status(
1002                Status.valueOf(rs.getString("sta_user"))
1003            )
1004            .required(rs.getBoolean("fl_required"))
1005            .download(rs.getBoolean("fl_download"))
1006            .build();
1007     }
1008
1009     private MapSqlParameterSource getParams(final Attachment
1010         attachment) {
1011         return new MapSqlParameterSource()
1012             .addValue("nm_attachment", attachment.getName())
1013             .addValue("sta_user",
1014                 attachment.getStatus().name())
1015             .addValue("fl_required", attachment.getRequired())
1016             .addValue("fl_download",
1017                 attachment.getDownload());
1018     }
1019 }
```

```
1010     }
1011 }
1012
1013
1014 src/main/java/br/ufsc/cultivar/repository/RatingRepository.java
1015
1016 package br.ufsc.cultivar.repository;
1017
1018 import br.ufsc.cultivar.model.Rating;
1019 import lombok.AccessLevel;
1020 import lombok.AllArgsConstructorConstructor;
1021 import lombok.experimental.FieldDefaults;
1022 import org.springframework.beans.factory.annotation.Autowired;
1023 import
1024     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1025 import
1026     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1027 import
1028     org.springframework.jdbc.core.namedparam.SqlParameterSource;
1029 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1030 import org.springframework.stereotype.Repository;
1031
1032 @Repository
1033 @AllArgsConstructor(onConstructor = @_(@Autowired))
1034 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1035 public class RatingRepository {
1036
1037     NamedParameterJdbcTemplate jdbcTemplate;
1038
1039     public void create(Rating rating, String cpf) {
1040         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1041             .withTableName("rating")
1042             .usingColumns("vl_rating", "dsc_rating",
1043                           "cod_cpf")
1044             .execute(getParams(rating, cpf));
1045     }
1046
1047     public void create(Rating rating, Long codEvent) {
1048         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1049             .withTableName("rating")
1050             .usingColumns("vl_rating", "dsc_rating",
1051                           "cod_event")
1052             .execute(getParams(rating, codEvent));
```

```
1050     }
1051
1052     private SqlParameterSource getParams(Rating rating, Object
1053         cod) {
1054         return new MapSqlParameterSource()
1055             .addValue("vl_rating", rating.getGrade())
1056             .addValue("dsc_rating", rating.getComment())
1057             .addValue("cod_cpf", cod)
1058             .addValue("cod_event", cod);
1059     }
1060
1061     public List<Rating> get(String cpf) {
1062         return jdbcTemplate.query(
1063             "select * from rating where cod_cpf=:cod_cpf",
1064             new MapSqlParameterSource("cod_cpf", cpf),
1065             (rs, i) -> Rating.builder()
1066                 .comment(rs.getString("dsc_rating"))
1067                 .grade(rs.getInt("vl_rating"))
1068                 .build()
1069         );
1070     }
1071
1072     public List<Rating> get(Long codEvent) {
1073         return jdbcTemplate.query(
1074             "select * from rating where cod_event=:cod_event",
1075             new MapSqlParameterSource("cod_event", codEvent),
1076             (rs, i) -> Rating.builder()
1077                 .comment(rs.getString("dsc_rating"))
1078                 .grade(rs.getInt("vl_rating"))
1079                 .build()
1080         );
1081     }
1082 }
1083
1084 src/main/java/br/ufsc/cultivar/repository/SchoolRepository.java
1085
1086 package br.ufsc.cultivar.repository;
1087
1088 import br.ufsc.cultivar.model.Address;
1089 import br.ufsc.cultivar.model.School;
1090 import br.ufsc.cultivar.model.SchoolType;
1091 import br.ufsc.cultivar.model.User;
1092 import br.ufsc.cultivar.utils.DatabaseUtils;
1093 import lombok.AccessLevel;
```

```
1094 import lombok.AllArgsConstructor;
1095 import lombok.experimental.FieldDefaults;
1096 import lombok.val;
1097 import org.springframework.beans.factory.annotation.Autowired;
1098 import
1099     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1100 import
1101     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1102 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1103 import org.springframework.stereotype.Repository;
1104
1105 import java.sql.ResultSet;
1106 import java.sql.SQLException;
1107 import java.util.List;
1108 import java.util.Map;
1109 import java.util.Objects;
1110 import java.util.Optional;
1111
1112 @Repository
1113 @AllArgsConstructor(onConstructor = @___(@Autowired))
1114 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1115 public class SchoolRepository {
1116
1117     NamedParameterJdbcTemplate jdbcTemplate;
1118
1119     public Long create(final School school) {
1120         return new
1121             SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1122                 .withTableName("school")
1123                 .usingGeneratedKeyColumns("cod_school")
1124                 .executeAndReturnKey(getParams(school))
1125                 .longValue();
1126     }
1127
1128     public List<School> get(final String filter, final Long
1129                             page) {
1130         val sql = new StringBuilder("select * from school");
1131         val params = new MapSqlParameterSource();
1132         Optional.ofNullable(filter)
1133             .ifPresent(
1134                 s -> {
1135                     sql.append(" where nm_school like :nm_school");
1136                     params.addValue("nm_school", filter + "%");
1137                 }
1138             );
1139     }
1140 }
```

```
1135     Optional.ofNullable(page)
1136         .ifPresent(
1137             aLong -> {
1138                 sql.append(" limit 5 offset :offset");
1139                 params.addValue("offset", page * 5);
1140             }
1141         );
1142     return jdbcTemplate.query(
1143         sql.toString(),
1144         params,
1145         (rs, i) -> this.build(rs)
1146     );
1147 }
1148
1149 public Integer count(final String filter) {
1150     val sql = new StringBuilder("select count(cod_school)"
1151         + " from school");
1152     val params = new MapSqlParameterSource();
1153     Optional.ofNullable(filter)
1154         .ifPresent(s -> {
1155             sql.append(" where nm_school like "
1156                 + " :nm_school");
1157             params.addValue("nm_school", filter +
1158                 "%");
1159         }
1160     );
1161     return jdbcTemplate.queryForObject(sql.toString(),
1162         params, Integer.class);
1163 }
1164
1165 public School get(final Long codSchool) {
1166     return jdbcTemplate.query(
1167         "select * from school where"
1168             + " cod_school=:cod_school",
1169             new MapSqlParameterSource("cod_school",
1170                 codSchool),
1171             this::build
1172     );
1173 }
1174
1175 public void delete(final Long codSchool) {
1176     jdbcTemplate.update(
1177         "delete from school where cod_school=:cod_school",
1178         new MapSqlParameterSource("cod_school", codSchool)
1179     );
1180 }
```

```
1174     }
1175
1176     public void update(final School school) {
1177         jdbcTemplate.update(
1178             "update school set nm_school=:nm_school,
1179                 dsc_phone=:dsc_phone,
1180                 cod_address=:cod_address," +
1181                 "cod_cpf=:cod_cpf where
1182                 cod_school=:cod_school",
1183             getParams(school).addValue("cod_school",
1184                 school.getCodSchool())
1185         );
1186     }
1187
1188     private MapSqlParameterSource getParams(final School school)
1189     {
1190         return new MapSqlParameterSource()
1191             .addValue("nm_school", school.getName())
1192             .addValue("dsc_phone", school.getPhone())
1193             .addValue("cod_address",
1194                 school.getAddress().getCodAddress())
1195             .addValue("cod_cpf",
1196                 school.getResponsible().getCpf())
1197             .addValue("tp_school", school.getType().name());
1198     }
1199
1200     private School build(final ResultSet rs) throws SQLException
1201     {
1202         if(!DatabaseUtils.isNotEmpty(rs)){
1203             return null;
1204         }
1205         return School.builder()
1206             .codSchool(rs.getLong("cod_school"))
1207             .name(rs.getString("nm_school"))
1208             .phone(rs.getString("dsc_phone"))
1209             .type(
1210                 SchoolType.valueOf(rs.getString("tp_school")))
1211             .address(
1212                 Address.builder()
1213                     .codAddress(rs.getLong("cod_address"))
1214                     .build()
1215             )
1216             .responsible(
1217                 User.builder()
1218                     .cpf(rs.getString("cod_cpf")))
```

```
1211             .build()
1212         )
1213     .build();
1214 }
1215 }
1216
1217
1218 src/main/java/br/ufsc/cultivar/repository/VolunteerRepository.java
1219
1220 package br.ufsc.cultivar.repository;
1221
1222 import br.ufsc.cultivar.model.*;
1223 import br.ufsc.cultivar.utils.DatabaseUtils;
1224 import javafx.scene.chart.PieChartBuilder;
1225 import lombok.AccessLevel;
1226 import lombok.AllArgsConstructorArgsConstructor;
1227 import lombok.experimental.FieldDefaults;
1228 import lombok.val;
1229 import org.springframework.beans.factory.annotation.Autowired;
1230 import
        org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1231 import
        org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1232 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1233 import org.springframework.stereotype.Repository;
1234
1235 import java.sql.ResultSet;
1236 import java.sql.SQLException;
1237 import java.util.*;
1238
1239 @Repository
1240 @AllArgsConstructor(onConstructor = @___(@Autowired))
1241 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1242 public class VolunteerRepository {
1243
1244     NamedParameterJdbcTemplate jdbcTemplate;
1245
1246     public void create(Volunteer volunteer) {
1247         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1248             .withTableName("volunteer")
1249             .execute(getParams(volunteer));
1250     }
1251
1252     public List<Volunteer> get(final List<String> filterCompany,
1253         final List<Long> filterSchool,
```

```

1253                     final String filter, final Integer
1254                     page) {
1255             val sql = new StringBuilder("select * from volunteer
1256                 where 1=1");
1257             val params = new MapSqlParameterSource();
1258
1259             if (!Optional.ofNullable(filterCompany)
1260                 .orElseGet(ArrayList::new)
1261                 .isEmpty())
1262             ){
1263                 sql.append(" and cod_cnpj in(:cod_cnpj)");
1264                 params.addValue("cod_cnpj", filterCompany);
1265             }
1266             if (!Optional.ofNullable(filterSchool)
1267                 .orElseGet(ArrayList::new)
1268                 .isEmpty())
1269             ){
1270                 sql.append(" and cod_school in(:cod_school)");
1271                 params.addValue("cod_school", filterSchool);
1272             }
1273             Optional.ofNullable(filter)
1274                 .ifPresent(
1275                     s -> {
1276                         sql.append(" and nm_volunteer like
1277                             :nm_volunteer");
1278                         params.addValue("nm_volunteer", filter + "%");
1279                     }
1280                 );
1281             Optional.ofNullable(page)
1282                 .ifPresent(
1283                     aLong -> {
1284                         sql.append(" limit 5 offset :offset");
1285                         params.addValue("offset", page * 5);
1286                     }
1287                 );
1288             return jdbcTemplate.query(
1289                 sql.toString(),
1290                 params,
1291                 (rs, i) -> this.build(rs)
1292             );
1293         }
1294
1295         public Volunteer get(String cpf) {
1296             return jdbcTemplate.query(
1297                 "select * from volunteer where cod_cpf=:cod_cpf",

```

```

1295                 new MapSqlParameterSource("cod_cpf", cpf),
1296                 this::build
1297             );
1298         }
1299
1300     public void delete(String cpf) {
1301         jdbcTemplate.update(
1302             "delete from volunteer where cod_cpf=:cod_cpf",
1303             new MapSqlParameterSource("cod_cpf", cpf)
1304         );
1305     }
1306
1307     public void update(Volunteer volunteer) {
1308         jdbcTemplate.update(
1309             "update volunteer set
1310                 dsc_schooling=:dsc_schooling,
1311                 fl_conclusion=:fl_conclusion, " +
1312                 "cod_rg=:cod_rg, cod_school=:cod_school
1313                 where cod_cpf=:cod_cpf",
1314             getParams(volunteer)
1315         );
1316     }
1317
1318     private MapSqlParameterSource getParams(Volunteer volunteer)
1319     {
1320         val user = volunteer.getUser();
1321         val school = volunteer.getSchool();
1322         return new MapSqlParameterSource()
1323             .addValue("cod_cpf", user.getCpf())
1324             .addValue("nm_volunteer", user.getName())
1325             .addValue("cod_cnpj",
1326                     volunteer.getCompany().getCnpj())
1327             .addValue("cod_school", Objects.nonNull(school) ?
1328                     school.getCodSchool() : null)
1329             .addValue("dsc_schooling",
1330                     volunteer.getSchooling().name())
1331             .addValue("fl_conclusion",
1332                     volunteer.getConclusion())
1333             .addValue("cod_rg", volunteer.getRg())
1334             .addValue("dsc_course", volunteer.getCourse());
1335     }
1336
1337     private Volunteer build(ResultSet rs) throws SQLException {
1338         if(!DatabaseUtils.isNotEmpty(rs)){
1339             return null;

```

```
1332     }
1333     return Volunteer.builder()
1334         .user(
1335             User.builder()
1336                 .cpf(rs.getString("cod_cpf"))
1337                 .build()
1338         )
1339         .company(
1340             Company.builder()
1341                 .cnpj(rs.getString("cod_cnpj"))
1342                 .build()
1343         ).school(
1344             School.builder()
1345                 .codSchool(rs.getLong("cod_school"))
1346                 .build()
1347         ).schooling(
1348             Schooling.valueOf(rs.getString("dsc_schooling"))
1349         ).course(rs.getString("dsc_course"))
1350         .conclusion(rs.getBoolean("fl_conclusion"))
1351         .rg(rs.getString("cod_rg"))
1352         .build();
1353     }
1354
1355     public Integer count(final List<String> filterCompany, final
1356         List<Long> filterSchool, final String filter) {
1357         val sql = new StringBuilder("select count(cod_cpf) from
1358             volunteer where 1=1");
1359         val params = new MapSqlParameterSource();
1360
1361         if (!Optional.ofNullable(filterCompany)
1362             .orElseGet(ArrayList::new)
1363             .isEmpty())
1364         ){
1365             sql.append(" and cod_cnpj in(:cod_cnpj)");
1366             params.addValue("cod_cnpj", filterCompany);
1367         }
1368         if (!Optional.ofNullable(filterSchool)
1369             .orElseGet(ArrayList::new)
1370             .isEmpty())
1371         ){
1372             sql.append(" and cod_school in(:cod_school)");
1373             params.addValue("cod_school", filterSchool);
1374         }
1375         Optional.ofNullable(filter)
1376             .ifPresent(
```

```
1375             s -> {
1376                 sql.append(" and nm_volunteer like
1377                             :nm_volunteer");
1378                 params.addValue("nm_volunteer", filter
1379                               + "%");
1380             }
1381         );
1382     }
1383
1384
1385     src/main/java/br/ufsc/cultivar/repository/EventRepository.java
1386
1387     package br.ufsc.cultivar.repository;
1388
1389     import br.ufsc.cultivar.dto.EventDTO;
1390     import br.ufsc.cultivar.model.*;
1391     import br.ufsc.cultivar.utils.DatabaseUtils;
1392     import br.ufsc.cultivar.utils.DateUtils;
1393     import lombok.AccessLevel;
1394     import lombok.AllArgsConstructor;
1395     import lombok.experimental.FieldDefaults;
1396     import lombok.val;
1397     import org.springframework.beans.factory.annotation.Autowired;
1398     import
1399         org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1400     import
1401         org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1402     import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1403     import org.springframework.stereotype.Repository;
1404
1405     import java.sql.ResultSet;
1406     import java.sql.SQLException;
1407     import java.util.*;
1408
1409     @Repository
1410     @AllArgsConstructor(onConstructor = @_(@Autowired))
1411     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1412     public class EventRepository {
1413
1414         NamedParameterJdbcTemplate jdbcTemplate;
1415
1416         public Long create(final Event event) {
```

```

1415     return new
1416         SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1417             .withTableName("event")
1418             .usingGeneratedKeyColumns("cod_event")
1419             .usingColumns(
1420                 "dt_start_occurrence",
1421                 "dt_end_occurrence",
1422                 "fl_all_day",
1423                 "tp_event",
1424                 "cod_address",
1425                 "fl_evaluate",
1426                 "cod_school"
1427             ).executeAndReturnKey(getParams(event))
1428             .longValue();
1429
1430     public List<Event> get(final List<String> filterVolunteer,
1431         final List<Long> filterSchool, final Long filterProject)
1432     {
1433         val sql = new StringBuilder("select e.* , te.nm_type from
1434             event e join participation natural join type_event
1435             te where 1=1");
1436         val params = new MapSqlParameterSource();
1437         if (!Optional.ofNullable(filterVolunteer)
1438             .orElseGet(ArrayList::new)
1439             .isEmpty())
1440         {
1441             sql.append(" and cod_cpf in(:cod_cpf)");
1442             params.addValue("cod_cpf", filterVolunteer);
1443         }
1444         if (!Optional.ofNullable(filterSchool)
1445             .orElseGet(ArrayList::new)
1446             .isEmpty())
1447         {
1448             sql.append(" and cod_school in(:cod_school)");
1449             params.addValue("cod_school", filterSchool);
1450         }
1451         if (Objects.nonNull(filterProject)){
1452             sql.append(" and cod_project in(:cod_project)");
1453             params.addValue("cod_project", filterProject);
1454         }
1455         return jdbcTemplate.query(
1456             sql.toString(),
1457             params,
1458             (rs, i) -> this.build(rs)

```

```
1455     );
1456 }
1457
1458 public Event get(final Long codEvent) {
1459     return jdbcTemplate.query(
1460         "select e.* , te.nm_type from event e natural join
1461             type_event te where cod_event=:cod_event",
1462         new MapSqlParameterSource("cod_event", codEvent),
1463         this::build
1464     );
1465 }
1466
1467 public void delete(final Long codEvent) {
1468     jdbcTemplate.update(
1469         "delete from event where cod_event=:cod_event",
1470         new MapSqlParameterSource("cod_event", codEvent)
1471     );
1472
1473 public void update(final Event event) {
1474     jdbcTemplate.update(
1475         "update event set
1476             dt_start_occurrence=:dt_start_occurrence,
1477             dt_end_occurrence=:dt_end_occurrence," +
1478             "tp_event=:tp_event,
1479             fl_all_day=:fl_all_day,
1480             cod_address=:cod_address,
1481             fl_evaluate=:fl_evaluate," +
1482             "cod_school=:cod_school where
1483             cod_event=:cod_event",
1484             getParams(event)
1485     );
1486 }
1487
1488 private Event build(final ResultSet rs) throws SQLException {
1489     if(!DatabaseUtils.isNotEmpty(rs)){
1490         return null;
1491     }
1492     val codProject = rs.getLong("cod_project");
1493     val isNull = rs.wasNull();
1494     return Event.builder()
1495         .codEvent(rs.getLong("cod_event"))
1496         .startOccurrence(DateUtils.toDate(rs.getTimestamp("dt_start_occurrence")))
1497         .endOccurrence(DateUtils.toDate(rs.getTimestamp("dt_end_occurrence")))
1498         .allDay(rs.getBoolean("fl_all_day"))
1499 }
```

```

1493     .createAt(DateUtils.toDate(rs.getTimestamp("dt_create")))
1494     .type(
1495         TypeEvent.builder()
1496             .type(rs.getLong("tp_event"))
1497             .name(rs.getString("nm_type"))
1498             .build()
1499     )
1500     .address(
1501         Address.builder()
1502             .codAddress(rs.getLong("cod_address"))
1503             .build()
1504     ).school(
1505         School.builder()
1506             .codSchool(rs.getLong("cod_school"))
1507             .build()
1508     ).project(
1509        isNull ?
1510             Project.builder()
1511                 .codProject(codProject)
1512                 .build() :
1513             null
1514     ).evaluate(rs.getBoolean("fl_evaluate"))
1515     .build();
1516 }
1517
1518 private MapSqlParameterSource getParams(final Event event) {
1519     return new MapSqlParameterSource()
1520         .addValue("dt_start_occurrence",
1521             event.getStartOccurrence())
1522         .addValue("dt_end_occurrence",
1523             event.getEndOccurrence())
1524         .addValue("fl_all_day", event.getAllDay())
1525         .addValue("tp_event", event.getType().getType())
1526         .addValue("cod_address",
1527             event.getAddress().getCodAddress())
1528         .addValue("cod_school",
1529             event.getSchool().getCodSchool())
1530         .addValue("cod_event", event.getCodEvent())
1531         .addValue("fl_evaluate", event.getEvaluate());
1532 }
1533
1534 public List<Event> eventsByVolunteer(final String cpf, final
1535     Long type) {
1536     val sb = new StringBuilder("select e.*,
1537         te.nm_type from
1538         event e natural join participation p natural join

```

```

        type_event te where cod_cpf=:cod_cpf");
1532     val params = new MapSqlParameterSource("cod_cpf", cpf);
1533     Optional.ofNullable(type)
1534         .ifPresent(
1535             aLong -> {
1536                 sb.append(" and te.tp_event=:tp_event");
1537                 params.addValue("tp_event", type);
1538             }
1539         );
1540     return jdbcTemplate.query(sb.toString(), params, (rs, i)
1541         -> this.build(rs));
1542 }
1543
1544 public List<Event> eventsBySchool(Long codSchool, Long type)
1545 {
1546     val sb = new StringBuilder("select * from event e
1547         natural join type_event te where
1548         cod_school=:cod_school");
1549     val params = new MapSqlParameterSource("cod_school",
1550         codSchool);
1551     Optional.ofNullable(type)
1552         .ifPresent(
1553             aLong -> {
1554                 sb.append(" and te.tp_event=:tp_event");
1555                 params.addValue("tp_event", type);
1556             }
1557         );
1558     return jdbcTemplate.query(sb.toString(), params, (rs, i)
1559         -> this.build(rs));
1560 }
1561
1562 public List<EventDTO> getEventsToAlert(String cpf) {
1563     val sql = "select dt_start_occurrence,
1564         dt_end_occurrence, "
1565         + "a.* , fl_all_day, te.nm_type from event "
1566         + "natural join type_event te natural join address a
1567             natural join participation "
1568         + "where cod_cpf=:cod_cpf and "
1569         + "dt_start_occurrence between CURDATE() and (CURDATE()
1570             + INTERVAL 7 DAY)";
1571     return jdbcTemplate.query(
1572         sql,
1573         new MapSqlParameterSource("cod_cpf", cpf),
1574         (rs, i) ->
1575             EventDTO.builder()
1576

```

```

1567     .type(
1568         TypeEvent.builder()
1569             .name(rs.getString("nm_type"))
1570             .build()
1571     ).startOccurrence(rs.getTimestamp("dt_start_occurrence"))
1572     .endOccurrence(rs.getTimestamp("dt_end_occurrence"))
1573     .allDay(rs.getBoolean("fl_all_day"))
1574     .address(
1575         Address.builder()
1576             .city(rs.getString("nm_city"))
1577             .neighborhood(rs.getString("nm_neighborhood"))
1578             .street(rs.getString("nm_street"))
1579             .number(rs.getString("nu_street"))
1580             .build()
1581     ).build()
1582 );
1583 }
1584
1585     public List<Event> getEventsToEvaluateByVolunteer(final
1586         String cpf) {
1587         return null;
1588     }
1589
1590
1591 src/main/java/br/ufsc/cultivar/repository/AddressRepository.java
1592
1593 package br.ufsc.cultivar.repository;
1594
1595 import br.ufsc.cultivar.model.Address;
1596 import lombok.AccessLevel;
1597 import lombok.NoArgsConstructor;
1598 import lombok.experimental.FieldDefaults;
1599 import lombok.val;
1600 import org.springframework.beans.factory.annotation.Autowired;
1601 import
1602     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1603 import
1604     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1605 import
1606     org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1607 import
1608     org.springframework.stereotype.Repository;
1609
1610     import java.util.Objects;
1611
1612     @Repository

```

```

1609     @AllArgsConstructor(onConstructor = @__(@Autowired))
1610     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1611     public class AddressRepository {
1612
1613         NamedParameterJdbcTemplate jdbcTemplate;
1614
1615         public Long create(final Address address) {
1616             val insert = new
1617                 SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1618                     .withTableName("address")
1619                     .usingGeneratedKeyColumns("cod_address");
1620             if (Objects.isNull(address.getNumber())){
1621                 insert.setColumns("nm_city", "nm_neighborhood",
1622                                 "nm_street");
1623             }
1624             return
1625                 insert.executeAndReturnKey(getParams(address)).longValue();
1626         }
1627
1628         public Address get(final Long codAddress) {
1629             return jdbcTemplate.query(
1630                 "select * from address where
1631                     cod_address=:cod_address",
1632                 new MapSqlParameterSource("cod_address", codAddress),
1633                 rs -> {
1634                     if(rs.isBeforeFirst()){
1635                         rs.first();
1636                     }
1637                     return Address.builder()
1638                         .codAddress(codAddress)
1639                         .city(rs.getString("nm_city"))
1640                         .neighborhood(rs.getString("nm_neighborhood"))
1641                         .street(rs.getString("nm_street"))
1642                         .number(rs.getString("nu_street"))
1643                         .build();
1644                 }
1645             );
1646         }
1647
1648         public void update(final Address address) {
1649             jdbcTemplate.update(
1650                 "update address set nm_city=:nm_city,
1651                     nm_neighborhood=:nm_neighborhood,
1652                     nm_street=:nm_street," +

```

```
1647             "nu_street=:nu_street where  
1648                 cod_address=:cod_address",  
1649             getParams(address)  
1650                 .addValue("cod_address",  
1651                     address.getCodAddress())  
1652         );  
1653     }  
1654  
1655     private MapSqlParameterSource getParams(final Address  
1656             address) {  
1657         return new MapSqlParameterSource()  
1658             .addValue("nm_city", address.getCity())  
1659             .addValue("nm_neighborhood",  
1660                 address.getNeighborhood())  
1661             .addValue("nm_street", address.getStreet())  
1662             .addValue("nu_street", address.getNumber());  
1663     }  
1664  
1665     src/main/java/br/ufsc/cultivar/repository/ParticipationRepository.java  
1666  
1667     package br.ufsc.cultivar.repository;  
1668  
1669     import br.ufsc.cultivar.dto.ParticipationDTO;  
1670     import lombok.AccessLevel;  
1671     import lombok.AllArgsConstructor;  
1672     import lombok.experimental.FieldDefaults;  
1673     import lombok.val;  
1674     import org.springframework.beans.factory.annotation.Autowired;  
1675     import  
1676         org.springframework.jdbc.core.namedparam.MapSqlParameterSource;  
1677     import  
1678         org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;  
1679     import org.springframework.jdbc.core.simple.SimpleJdbcInsert;  
1680     import org.springframework.stereotype.Repository;  
1681  
1682     import java.util.List;  
1683  
1684     @Repository  
1685     @AllArgsConstructor(onConstructor = @___(@Autowired))  
1686     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
1687     public class ParticipationRepository {  
1688  
1689         NamedParameterJdbcTemplate jdbcTemplate;
```

```
1686
1687     public void create(Long codEvent, String cpf) {
1688         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1689             .withTableName("participation")
1690             .execute(
1691                 new MapSqlParameterSource()
1692                     .addValue("cod_cpf", cpf)
1693                     .addValue("cod_event", codEvent)
1694             );
1695     }
1696
1697     public List<ParticipationDTO> getVolunteerLocals(final
1698         String codCpf){
1699         val sql = "select a.* , nm_school from volunteer v" +
1700             " natural join participation p" +
1701             " join event e on p.cod_event = e.cod_event " +
1702             " natural join address a" +
1703             " join school s on e.cod_school = s.cod_school "
1704             +
1705             "where p.cod_cpf=:cod_cpf and e.dt_end_occurrence
1706                 < current_date";
1707         return jdbcTemplate.query(
1708             sql,
1709             new MapSqlParameterSource("cod_cpf", codCpf),
1710             (rs, i) -> ParticipationDTO.builder()
1711                 .school(rs.getString("nm_school"))
1712                 .city(rs.getString("nm_city"))
1713                 .neighborhood(rs.getString("nm_neighborhood"))
1714                 .street(rs.getString("nm_street"))
1715                 .number(rs.getString("nu_street"))
1716                 .build()
1717             );
1718     }
1719
1720     src/main/java/br/ufsc/cultivar/repository/ProjectRepository.java
1721
1722     package br.ufsc.cultivar.repository;
1723
1724     import br.ufsc.cultivar.model.Project;
1725     import br.ufsc.cultivar.utils.DatabaseUtils;
1726     import lombok.AccessLevel;
1727     import lombok.AllArgsConstructor;
1728     import lombok.experimental.FieldDefaults;
```

```
1728 import lombok.val;
1729 import org.springframework.beans.factory.annotation.Autowired;
1730 import
1731     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1732 import
1733     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1734 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1735 import org.springframework.stereotype.Repository;
1736
1737 import java.sql.ResultSet;
1738 import java.sql.SQLException;
1739 import java.util.List;
1740 import java.util.Optional;
1741
1742 @Repository
1743 @AllArgsConstructor(onConstructor = @_(@Autowired))
1744 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1745 public class ProjectRepository {
1746
1747     NamedParameterJdbcTemplate jdbcTemplate;
1748
1749     public void create(Project project) {
1750         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1751             .withTableName("project")
1752             .usingGeneratedKeyColumns("cod_project")
1753             .execute(
1754                 new MapSqlParameterSource()
1755                     .addValue("nm_project", project.getName())
1756                     .addValue("dt_start", project.getStart())
1757                     .addValue("dt_end", project.getEnd())
1758             );
1759     }
1760
1761     public List<Project> get(final String filter, final Long
1762         page) {
1763         val sql = new StringBuilder("select * from project ");
1764         val params = new MapSqlParameterSource();
1765         Optional.ofNullable(filter)
1766             .ifPresent(
1767                 s -> {
1768                     sql.append("where nm_project like :nm_project
1769                         ");
1770                     params.addValue("nm_project", filter + "%");
1771                 }
1772             );
1773     }
1774 }
```

```
1769         sql.append("order by dt_start desc ");
1770         Optional.ofNullable(page)
1771             .ifPresent(
1772                 aLong -> {
1773                     sql.append("limit 20 offset :offset");
1774                     params.addValue("offset", page*20);
1775                 }
1776             );
1777         return jdbcTemplate.query(sql.toString(), params, (rs,
1778             i) -> this.build(rs));
1779     }
1780
1781     public Integer count(final String filter) {
1782         val sql = new StringBuilder("select count(cod_project)
1783             from project ");
1784         val params = new MapSqlParameterSource();
1785         Optional.ofNullable(filter)
1786             .ifPresent(
1787                 s -> {
1788                     sql.append("where nm_project like
1789                         :nm_project ");
1790                     params.addValue("nm_project", filter +
1791                         "%");
1792                 }
1793             );
1794         return jdbcTemplate.queryForObject(sql.toString(),
1795             params, Integer.class);
1796     }
1797
1798     public Project get(Long codProject) {
1799         return jdbcTemplate.query(
1800             "select * from project where
1801                 cod_project=:cod_project",
1802             new MapSqlParameterSource("cod_project", codProject),
1803             this::build
1804         );
1805     }
1806
1807     public void delete(Long codProject) {
1808         jdbcTemplate.update(
1809             "delete from project where cod_project=:cod_project",
1810             new MapSqlParameterSource("cod_project", codProject)
1811         );
1812     }
1813
1814 }
```

```
1808     private Project build(ResultSet rs) throws SQLException {
1809         if(!DatabaseUtils.isNotEmpty(rs)){
1810             return null;
1811         }
1812         return Project.builder()
1813             .codProject(rs.getLong("cod_project"))
1814             .name(rs.getString("nm_project"))
1815             .start(rs.getDate("dt_start"))
1816             .end(rs.getDate("dt_end"))
1817             .build();
1818     }
1819 }
1820
1821
1822 src/main/java/br/ufsc/cultivar/repository/AnswerRepository.java
1823
1824 package br.ufsc.cultivar.repository;
1825
1826 import br.ufsc.cultivar.model.Answer;
1827 import br.ufsc.cultivar.model.Question;
1828 import lombok.AccessLevel;
1829 import lombok.AllArgsConstructor;
1830 import lombok.experimental.FieldDefaults;
1831 import org.springframework.beans.factory.annotation.Autowired;
1832 import
1833     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1834 import
1835     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1836 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1837 import org.springframework.stereotype.Repository;
1838
1839 @Repository
1840 @AllArgsConstructor(onConstructor = @___(@Autowired))
1841 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1842 public class AnswerRepository {
1843
1844     NamedParameterJdbcTemplate jdbcTemplate;
1845
1846     public void create(final Answer answer, final String cpf) {
1847         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1848             .withTableName("answer")
1849             .execute(
1850                 new MapSqlParameterSource()
```

```
1851                     .addValue("cod_cpf", cpf)
1852                     .addValue("cod_question",
1853                         answer.getQuestion().getCodQuestion())
1854                     .addValue("fl_answer",
1855                         answer.getAnswer())
1856                     .addValue("dsc_answer",
1857                         answer.getComment())
1858                 );
1859             }
1860
1861             public List<Answer> get(final String cpf) {
1862                 return jdbcTemplate.query(
1863                     "select * from answer where cod_cpf=:cod_cpf",
1864                     new MapSqlParameterSource("cod_cpf", cpf),
1865                     (rs, i) -> Answer.builder()
1866                         .answer(rs.getBoolean("fl_answer"))
1867                         .comment(rs.getString("dsc_answer"))
1868                         .question(
1869                             Question.builder()
1870                               .codQuestion(rs.getLong("cod_question"))
1871                               .build()
1872                         )
1873                     );
1874             }
1875
1876             public void delete(String cpf) {
1877                 jdbcTemplate.update(
1878                     "delete from answer where cod_cpf=:cod_cpf",
1879                     new MapSqlParameterSource("cod_cpf", cpf)
1880                 );
1881             }
1882
1883             src/main/java/br/ufsc/cultivar/repository/UserRepository.java
1884
1885             package br.ufsc.cultivar.repository;
1886
1887             import br.ufsc.cultivar.model.Address;
1888             import br.ufsc.cultivar.model.Role;
1889             import br.ufsc.cultivar.model.Status;
1890             import br.ufsc.cultivar.model.User;
1891             import lombok.AccessLevel;
1892             import lombok.AllArgsConstructor;
```

```
1893 import lombok.experimental.FieldDefaults;
1894 import lombok.val;
1895 import org.springframework.beans.factory.annotation.Autowired;
1896 import
1897     org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
1898 import
1899     org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
1900 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
1901 import org.springframework.stereotype.Repository;
1902
1903 import java.sql.ResultSet;
1904 import java.sql.SQLException;
1905 import java.util.List;
1906 import java.util.Map;
1907 import java.util.Optional;
1908
1909 @Repository
1910 @AllArgsConstructor(onConstructor = @_(@Autowired))
1911 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
1912 public class UserRepository {
1913     NamedParameterJdbcTemplate jdbcTemplate;
1914
1915     public void create(final User user) {
1916         new SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
1917             .withTableName("users")
1918             .usingColumns("cod_cpf", "nm_user", "dsc_email",
1919                         "dsc_password", "dsc_role",
1920                         "sta_user", "dt_birth", "cod_address",
1921                         "dsc_job", "dsc_phone")
1922             .execute(getParams(user));
1923     }
1924
1925     public List<User> get(final Map<String, Object> filter) {
1926         val sql = new StringBuilder("select * from users where");
1927         sql.append(" 1=1");
1928         val params = new MapSqlParameterSource();
1929         Optional.ofNullable(filter)
1930             .ifPresent(
1931                 map -> map.forEach(
1932                     (key, value) -> {
1933                         sql.append(" and ")
1934                             .append(key)
1935                             .append("=:")
1936                             .append(key);
1937                         params.addValue(key, value);
1938                     })
1939             );
1940     }
1941 }
```

```
1933                     }
1934             )
1935         );
1936     return jdbcTemplate.query(
1937         sql.toString(),
1938         params,
1939         (rs, i) -> this.build(rs)
1940     );
1941 }
1942
1943 public User get(final String cpf) {
1944     return jdbcTemplate.query(
1945         "select * from users where cod_cpf=:cod_cpf",
1946         new MapSqlParameterSource("cod_cpf", cpf),
1947         this::build
1948     );
1949 }
1950
1951 public void delete(final String cpf) {
1952     jdbcTemplate.update(
1953         "delete from users where cod_cpf=:cod_cpf",
1954         new MapSqlParameterSource("cod_cpf", cpf)
1955     );
1956 }
1957
1958 public void update(final User user) {
1959     val sql = "update users " +
1960         "set nm_user=:nm_user, dsc_email=:dsc_email,
1961             dsc_password=:dsc_password, dsc_role=:dsc_role," +
1962         "sta_user=:sta_user, dt_birth=:dt_birth,
1963             cod_address=:cod_address, dsc_job=:dsc_job," +
1964         "dsc_phone=:dsc_phone where cod_cpf=:cod_cpf;";
1965     jdbcTemplate.update(sql, getParams(user));
1966 }
1967
1968 public List<User> getParticipants(final Long codEvent) {
1969     return jdbcTemplate.query(
1970         "select u.* from users u join participation p on
1971             u.cod_cpf=p.cod_cpf where
1972                 p.cod_event=:cod_event",
1973         new MapSqlParameterSource("cod_event", codEvent),
1974         (rs, i) -> this.build(rs)
1975     );
1976 }
1977 }
```

```
1974 private MapSqlParameterSource getParams(final User user){
1975     return new MapSqlParameterSource()
1976         .addValue("cod_cpf", user.getCpf())
1977         .addValue("nm_user", user.getName())
1978         .addValue("dsc_email", user.getEmail())
1979         .addValue("dsc_password", user.getPassword())
1980         .addValue("dsc_role", user.getRole().name())
1981         .addValue("sta_user", user.getStatus().name())
1982         .addValue("dt_birth", user.getBirth())
1983         .addValue("cod_address",
1984             user.getAddress().getCodAddress())
1985         .addValue("dsc_job", user.getJob())
1986         .addValue("dsc_phone", user.getPhone());
1987 }
1988
1989 private User build(final ResultSet rs) throws SQLException {
1990     if(rs.isBeforeFirst()){
1991         rs.first();
1992     }
1993     return User.builder()
1994         .cpf(rs.getString("cod_cpf"))
1995         .name(rs.getString("nm_user"))
1996         .email(rs.getString("dsc_email"))
1997         .password(rs.getString("dsc_password"))
1998         .role(Role.valueOf(rs.getString("dsc_role")))
1999         .status(Status.valueOf(rs.getString("sta_user")))
2000         .birth(rs.getDate("dt_birth"))
2001         .address(
2002             Address.builder()
2003                 .codAddress(rs.getLong("cod_address"))
2004                 .build()
2005         )
2006         .job(rs.getString("dsc_job"))
2007         .phone(rs.getString("dsc_phone"))
2008         .build();
2009 }
2010
2011
2012 src/main/java/br/ufsc/cultivar/repository/TypeEventRepository.java
2013
2014 package br.ufsc.cultivar.repository;
2015
2016 import br.ufsc.cultivar.model.Project;
2017 import br.ufsc.cultivar.model.TypeEvent;
```

```
2018 import br.ufsc.cultivar.utils.DatabaseUtils;
2019 import javafx.scene.chart.PieChartBuilder;
2020 import lombok.AccessLevel;
2021 import lombok.AllArgsConstructor;
2022 import lombok.experimental.FieldDefaults;
2023 import lombok.val;
2024 import org.springframework.beans.factory.annotation.Autowired;
2025 import
        org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
2026 import
        org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
2027 import org.springframework.jdbc.core.simple.SimpleJdbcInsert;
2028 import org.springframework.stereotype.Repository;
2029
2030 import java.sql.ResultSet;
2031 import java.sql.SQLException;
2032 import java.util.List;
2033 import java.util.Optional;
2034
2035 @Repository
2036 @AllArgsConstructor(onConstructor = @__(@Autowired))
2037 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2038 public class TypeEventRepository {
2039
2040     NamedParameterJdbcTemplate jdbcTemplate;
2041
2042     public Long create(String name) {
2043         return new
                SimpleJdbcInsert(jdbcTemplate.getJdbcTemplate())
                    .withTableName("type_event")
                    .usingGeneratedKeyColumns("tp_event")
                    .executeAndReturnKey(
                        new MapSqlParameterSource("nm_type", name)
                    ).longValue();
2044     }
2045
2046     public List<TypeEvent> get(final String filter, final Long
2047         page) {
2048         val sql = new StringBuilder("select * from type_event ");
2049         val params = new MapSqlParameterSource();
2050         Optional.ofNullable(filter)
2051             .ifPresent(
2052                 s -> {
2053                     sql.append("where nm_type like :nm_type ");
2054                     params.addValue("nm_type", filter + "%");
2055                 }
2056             );
2057     }
2058 }
```

```
2059         }
2060     );
2061     Optional.ofNullable(page)
2062         .ifPresent(
2063             aLong -> {
2064                 sql.append("limit 20 offset :offset");
2065                 params.addValue("offset", page*20);
2066             }
2067         );
2068     return jdbcTemplate.query(sql.toString(), params, (rs,
2069         i) -> this.build(rs));
2070 }
2071
2072 public TypeEvent get(Long tpEvent) {
2073     return jdbcTemplate.query(
2074         "select * from type_event where
2075             tp_event=:tp_event",
2076         new MapSqlParameterSource("tp_event", tpEvent),
2077         this::build
2078     );
2079 }
2080
2081 private TypeEvent build(ResultSet rs) throws SQLException {
2082     if(!DatabaseUtils.isNotEmpty(rs)){
2083         return null;
2084     }
2085     return TypeEvent.builder()
2086         .type(rs.getLong("tp_event"))
2087         .name(rs.getString("nm_type"))
2088         .build();
2089 }
2090
2091 public void delete(Long tpEvent) {
2092     jdbcTemplate.update(
2093         "delete from type_event where tp_event=:tp_event",
2094         new MapSqlParameterSource("tp_event", tpEvent)
2095     );
2096 }
2097
2098 public Integer count(String filter) {
2099     val sql = new StringBuilder("select count(tp_event) from
2100         type_event ");
2101     val params = new MapSqlParameterSource();
2102     Optional.ofNullable(filter)
2103         .ifPresent(
```

```

2101             s -> {
2102                 sql.append("where nm_type like :nm_type ");
2103                 params.addValue("nm_type", filter + "%");
2104             }
2105         );
2106         return jdbcTemplate.queryForObject(sql.toString(),
2107             params, Integer.class);
2108     }
2109 }
2110
2111 src/main/java/br/ufsc/cultivar/config/SecurityConfig.java
2112
2113 package br.ufsc.cultivar.config;
2114
2115 import org.springframework.context.annotation.Bean;
2116 import
2117 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
2118 import
2119 import org.springframework.security.config.annotation.web.configuration.EnableWebSe
2120 import
2121 import
2122 import org.springframework.security.config.annotation.web.configuration.WebSecurity
2123 import org.springframework.web.cors.CorsConfiguration;
2124 import org.springframework.web.cors.CorsConfigurationSource;
2125 import
2126 import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
2127
2128 @EnableWebSecurity
2129 public class SecurityConfig extends WebSecurityConfigurerAdapter
2130 {
2131
2132     @Override
2133     protected void configure(HttpSecurity http) throws Exception
2134     {
2135         http.cors()
2136             .and()
2137             .csrf()
2138             .disable();
2139     }
2140
2141     @Bean
2142     CorsConfigurationSource corsConfigurationSource(){
2143         CorsConfiguration corsConfiguration = new
2144             CorsConfiguration();
2145         corsConfiguration.addAllowedHeader("*");
2146     }
2147 }
```

```
2138     corsConfiguration.addAllowedMethod("*");
2139     corsConfiguration.addAllowedOrigin("*");
2140
2141     UrlBasedCorsConfigurationSource source = new
2142         UrlBasedCorsConfigurationSource();
2143     source.registerCorsConfiguration("/**",
2144         corsConfiguration);
2145     return source;
2146 }
2147
2148 src/main/java/br/ufsc/cultivar/config/EmailConfig.java
2149
2150 package br.ufsc.cultivar.config;
2151
2152 import lombok.Getter;
2153 import org.springframework.beans.factory.annotation.Value;
2154 import org.springframework.context.annotation.Configuration;
2155
2156 @Configuration
2157 @Getter
2158 public class EmailConfig {
2159     @Value("${spring.mail.username}")
2160     String from;
2161 }
2162
2163
2164 src/main/java/br/ufsc/cultivar/config/AsyncConfig.java
2165
2166 package br.ufsc.cultivar.config;
2167
2168 import org.springframework.context.annotation.Bean;
2169 import org.springframework.context.annotation.Configuration;
2170 import org.springframework.scheduling.annotation.EnableAsync;
2171 import
2172     org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor;
2173
2174 import java.util.concurrent.Executor;
2175
2176 @Configuration
2177 @EnableAsync
2178 public class AsyncConfig {
2179     @Bean
2180     public Executor asyncExecutor() {
```

```
2180     ThreadPoolTaskExecutor executor = new
2181         ThreadPoolTaskExecutor();
2182     executor.setCorePoolSize(2);
2183     executor.setMaxPoolSize(2);
2184     executor.setQueueCapacity(50);
2185     executor.setThreadNamePrefix("email-");
2186     executor.initialize();
2187     return executor;
2188 }
2189 }
2190
2191 src/main/java/br/ufsc/cultivar/config/JdbcConfig.java
2192
2193 package br.ufsc.cultivar.config;
2194
2195 import org.springframework.context.annotation.Configuration;
2196 import
2197     org.springframework.transaction.annotation.EnableTransactionManagement;
2198
2199 @Configuration
2200 @EnableTransactionManagement
2201 public class JdbcConfig {
2202 }
2203
2204
2205 src/main/java/br/ufsc/cultivar/config/SwaggerConfig.java
2206
2207 package br.ufsc.cultivar.config;
2208
2209 import org.springframework.context.annotation.Bean;
2210 import org.springframework.context.annotation.Configuration;
2211 import
2212     org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
2213 import
2214     org.springframework.web.servlet.config.annotation.WebMvcConfigurerSupport;
2215 import
2216     springfox.documentation.builders.ApiInfoBuilder;
2217 import
2218     springfox.documentation.builders.RequestHandlerSelectors;
2219 import
2220     springfox.documentation.service.ApiInfo;
2221 import
2222     springfox.documentation.service.Contact;
2223 import
2224     springfox.documentation.spi.DocumentationType;
2225 import
2226     springfox.documentation.spring.web.plugins.Docket;
2227 import
2228     springfox.documentation.swagger2.annotations.EnableSwagger2;
```

```
2220  
2221 import static springfox.documentation.builders.PathSelectors.ant;  
2222  
2223 @Configuration  
2224 @EnableSwagger2  
2225 public class SwaggerConfig extends WebMvcConfigurationSupport {  
2226  
2227     @Bean  
2228  
2229     public Docket productApi() {  
2230         return new Docket(DocumentationType.SWAGGER_2)  
2231             .select()  
2232             .apis(RequestHandlerSelectors.basePackage("br.ufsc.cultivar.  
2233             .paths(ant("/api/**"))  
2234             .build()  
2235             .apiInfo(metaData());  
2236  
2237     }  
2238  
2239     private ApiInfo metaData() {  
2240         return new ApiInfoBuilder()  
2241             .title("Sistema CultiVar")  
2242             .description("Sistema de Controle de Tutores  
2243                 Voluntários")  
2244             .version("0.2.0")  
2245             .license("Apache License Version 2.0")  
2246             .licenseUrl("https://www.apache.org/licenses/LICENSE-2.0")  
2247             .contact(new Contact("Luiz Maestri", null,  
2248                 "luizricardomaestri@gmail.com"))  
2249             .build();  
2250     }  
2251     @Override  
2252     protected void addResourceHandlers(ResourceHandlerRegistry  
2253         registry) {  
2254         registry.addResourceHandler("swagger-ui.html")  
2255             .addResourceLocations("classpath:/META-INF/resource/");  
2256  
2257         registry.addResourceHandler("/webjars/**")  
2258             .addResourceLocations("classpath:/META-INF/resource/webjars/  
2259     }  
2260 }  
2261 src/main/java;br\ufsc\cultivar\security\AuthResponseDTO.java
```

```
2262  
2263 package br.ufsc.cultivar.security;  
2264  
2265 import br.ufsc.cultivar.model.User;  
2266 import com.fasterxml.jackson.annotation.JsonInclude;  
2267 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;  
2268 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;  
2269 import lombok.AccessLevel;  
2270 import lombok.Builder;  
2271 import lombok.Value;  
2272 import lombok.experimental.FieldDefaults;  
2273 import lombok.experimental.Wither;  
2274  
2275 @Value  
2276 @Wither  
2277 @Builder(builderClassName = "Builder")  
2278 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
2279 @JsonDeserialize(builder = AuthResponseDTO.Builder.class)  
2280 @JsonInclude(JsonInclude.Include.NON_NULL)  
2281 class AuthResponseDTO {  
2282     User user;  
2283     String token;  
2284  
2285     @JsonPOJOBuilder(withPrefix = "")  
2286     static class Builder{}  
2287 }  
2288  
2289 src/main/java(br/ufsc/cultivar/security/AuthException.java  
2290  
2291 package br.ufsc.cultivar.security;  
2292  
2293 public class AuthException extends Exception {  
2294     AuthException(String message) {  
2295         super(message);  
2296     }  
2297 }  
2298 }  
2299  
2300  
2301 src/main/java(br/ufsc/cultivar/security/AuthService.java  
2302  
2303 package br.ufsc.cultivar.security;  
2304  
2305 import br.ufsc.cultivar.exception.ServiceException;  
2306 import br.ufsc.cultivar.model.User;
```

```
2307 import br.ufsc.cultivar.service.UserService;
2308 import lombok.AccessLevel;
2309 import lombok.AllArgsConstructorConstructor;
2310 import lombok.experimental.FieldDefaults;
2311 import org.springframework.beans.factory.annotation.Autowired;
2312 import org.springframework.stereotype.Service;
2313
2314 @Service
2315 @AllArgsConstructor(onConstructor = @_(@Autowired))
2316 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2317 class AuthService {
2318     UserService userService;
2319
2320     AuthResponseDTO authenticate(AuthRequestDTO dto) throws
2321         AuthException, ServiceException {
2322         User user = userService.get(dto.getCpf());
2323         if (user.getPassword().equals(dto.getPassword())) {
2324             return AuthResponseDTO.builder()
2325                 .user(user)
2326                 .token("")
2327                 .build();
2328         }
2329         throw new AuthException("Usuário e/ou senha incorretos.");
2330     }
2331
2332
2333 src/main/java(br/ufsc/cultivar/security/AuthRequestDTO.java
2334
2335 package br.ufsc.cultivar.security;
2336
2337 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2338 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2339 import lombok.AccessLevel;
2340 import lombok.Builder;
2341 import lombok.Value;
2342 import lombok.experimental.FieldDefaults;
2343 import org.hibernate.validator.constraints.br.CPF;
2344
2345 import javax.validation.constraints.NotBlank;
2346
2347 @Value
2348 @Builder
2349 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
```

```
2350     @JsonDeserialize(builder =
2351         AuthRequestDTO.AuthRequestDTOBuilder.class)
2352     class AuthRequestDTO {
2353         @CPF
2354         String cpf;
2355         @NotBlank
2356         String password;
2357
2358         @JsonPOJOBuilder(withPrefix = "")
2359         public static class AuthRequestDTOBuilder{}
2360     }
2361
2362     src/main/java/br/ufsc/cultivar/security/AuthResource.java
2363
2364     package br.ufsc.cultivar.security;
2365
2366     import br.ufsc.cultivar.exception.ServiceException;
2367     import lombok.AccessLevel;
2368     import lombok.AllArgsConstructor;
2369     import lombok.experimental.FieldDefaults;
2370     import org.springframework.beans.factory.annotation.Autowired;
2371     import org.springframework.http.MediaType;
2372     import org.springframework.web.bind.annotation.PostMapping;
2373     import org.springframework.web.bind.annotation.RequestBody;
2374     import org.springframework.web.bind.annotation.RequestMapping;
2375     import org.springframework.web.bind.annotation.RestController;
2376
2377     import javax.validation.Valid;
2378
2379     @RestController
2380     @RequestMapping(value = "/auth")
2381     @AllArgsConstructor(onConstructor = @__(@Autowired))
2382     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2383     public class AuthResource {
2384
2385         AuthService service;
2386
2387         @PostMapping(consumes =
2388             MediaType.APPLICATION_JSON_UTF8_VALUE,
2389             produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
2390         public AuthResponseDTO authenticate(@RequestBody @Valid
AuthRequestDTO dto) throws AuthException,
ServiceException {
return service.authenticate(dto);
```

```
2391     }
2392 }
2393
2394
2395
2396 src/main/java/br/ufsc/cultivar/scheduler/EventEmailScheduler.java
2397
2398 package br.ufsc.cultivar.scheduler;
2399
2400 import br.ufsc.cultivar.dto.UserEventsDTO;
2401 import br.ufsc.cultivar.email.EmailClient;
2402 import br.ufsc.cultivar.model.User;
2403 import br.ufsc.cultivar.service.EventService;
2404 import lombok.AccessLevel;
2405 import lombok.AllArgsConstructorConstructor;
2406 import lombok.experimental.FieldDefaults;
2407 import lombok.extern.slf4j.Slf4j;
2408 import lombok.val;
2409 import org.springframework.beans.factory.annotation.Autowired;
2410 import org.springframework.scheduling.annotation.Scheduled;
2411 import org.springframework.stereotype.Component;
2412 import org.thymeleaf.context.Context;
2413 import org.thymeleaf.spring5.SpringTemplateEngine;
2414
2415 import java.util.List;
2416
2417 @Slf4j
2418 @Component
2419 @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
2420 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2421 public class EventEmailScheduler {
2422
2423     SpringTemplateEngine templateEngine;
2424     EventService service;
2425     EmailClient client;
2426
2427     @Scheduled(cron = "${scheduler.email.send.cron}")
2428     public void sendWarning(){
2429         service.getEventsToAlert()
2430             .forEach(
2431                 userEventsDTO ->
2432                     client.sendEmailSync(
2433                         userEventsDTO.getEmail(),
2434                         "Eventos",
2435                         buildEmail(userEventsDTO)
```

```
2436             )
2437         );
2438     }
2439
2440     private String buildEmail(UserEventsDTO userEventsDTO){
2441         val context = new Context();
2442         context.setVariable("message", "teste");
2443         context.setVariable("user", userEventsDTO);
2444         return templateEngine.process("eventEmail", context);
2445     }
2446 }
2447
2448
2449 src/main/java/br/ufsc/cultivar/utils/FileUtils.java
2450
2451 package br.ufsc.cultivar.utils;
2452
2453 import br.ufsc.cultivar.exception.ServiceException;
2454 import br.ufsc.cultivar.exception.Type;
2455 import br.ufsc.cultivar.exception.UploadException;
2456 import br.ufsc.cultivar.model.Attachment;
2457 import br.ufsc.cultivar.model.Dispatch;
2458 import lombok.extern.slf4j.Slf4j;
2459 import lombok.val;
2460 import org.springframework.core.io.FileSystemResource;
2461 import org.springframework.core.io.Resource;
2462 import org.springframework.core.io.UrlResource;
2463 import org.springframework.stereotype.Component;
2464 import org.springframework.stereotype.Service;
2465 import org.springframework.web.multipart.MultipartFile;
2466
2467 import java.io.File;
2468 import java.io.IOException;
2469 import java.net.MalformedURLException;
2470 import java.nio.file.Files;
2471 import java.nio.file.Paths;
2472
2473 @Component
2474 @Slf4j
2475 public class FileUtils {
2476     public Dispatch save(final Attachment attachment, final
2477         MultipartFile multipartFile, final String cpf) throws
2478         UploadException {
2479         val path = String.format(
2480             "./files/users/%s/attachments/%s.pdf",
2481             cpf,
2482             attachment.getName()
2483         );
2484         try {
2485             Files.createDirectories(Paths.get(path).getParent());
2486             Files.write(Paths.get(path), multipartFile.getBytes());
2487         } catch (IOException e) {
2488             log.error("Error saving file: " + e.getMessage());
2489             throw new UploadException("Error saving file");
2490         }
2491     }
2492 }
```

```
2479         cpf,
2480         attachment.getName()
2481             .replaceAll(" ", "_")
2482     );
2483     val file = new File(path);
2484     save(file, multipartFile);
2485     return
2486         Dispatch.builder().attachment(attachment).send(true).build();
2487     }
2488
2489     public String save(final MultipartFile multipartFile, final
2490         Long codEvent) throws UploadException {
2491     val path = String.format(
2492         "./files/events/%d/attachments/%d.pdf",
2493         codEvent,
2494         System.currentTimeMillis()
2495     );
2496     return save(new File(path), multipartFile);
2497     }
2498
2499     public String saveAttachment(final MultipartFile
2500         multipartFile, final Long codAttachment) throws
2501         UploadException {
2502     val path = String.format(
2503         "./files/attachments/%d.pdf",
2504         codAttachment
2505     );
2506     return save(new File(path), multipartFile);
2507     }
2508
2509     private String save(final File file, final MultipartFile
2510         multipart) throws UploadException{
2511     try {
2512         if (file.getParentFile().mkdirs()){
2513             log.info("create dir: " + file.getParent());
2514         }
2515         Files.copy(multipartFile.getInputStream(),
2516             Paths.get(file.toURI()));
2517         return file.getName();
2518     } catch (IOException e) {
2519         throw new UploadException(String.format("No foi
2520             possivel salvar o arquivo %s.", file.getPath()),
2521         e);
2522     }
2523 }
```

```
2516     public FileSystemResource get(final Long codAttachment) {  
2517         val path = String.format(  
2518             "./files/attachments/%d.pdf",  
2519             codAttachment  
2520         );  
2521         return new FileSystemResource(path);  
2522     }  
2523 }  
2524 }  
2525  
2526 src/main/java/br/ufsc/cultivar/utils/ValidateUtils.java  
2528  
2529 package br.ufsc.cultivar.utils;  
2530  
2531 import lombok.val;  
2532  
2533 import javax.validation.Validation;  
2534  
2535 public class ValidateUtils {  
2536  
2537     public static <T> Boolean isValid(T entity){  
2538         if (entity == null){  
2539             return false;  
2540         }  
2541         val factory = Validation.buildDefaultValidatorFactory();  
2542         val validator = factory.getValidator();  
2543         val validate = validator.validate(entity);  
2544         return validate.isEmpty();  
2545     }  
2546 }  
2547  
2548  
2549 src/main/java/br/ufsc/cultivar/utils/DatabaseUtils.java  
2550  
2551 package br.ufsc.cultivar.utils;  
2552  
2553 import java.sql.ResultSet;  
2554 import java.sql.SQLException;  
2555  
2556 public class DatabaseUtils {  
2557  
2558     public static boolean isEmpty(ResultSet rs) throws  
2559         SQLException {  
2560         if (rs.isBeforeFirst()) {
```

```
2560         rs.first();
2561     }
2562     return rs.isLast() || rs.next() && rs.previous();
2563   }
2564 }
2565
2566
2567 src/main/java/br/ufsc/cultivar/utils/DateUtils.java
2568
2569 package br.ufsc.cultivar.utils;
2570
2571 import java.sql.Date;
2572 import java.sql.Timestamp;
2573
2574 public class DateUtils {
2575     public static java.util.Date toDate(Timestamp timestamp) {
2576         return Date.from(timestamp.toInstant());
2577     }
2578 }
2579
2580
2581 src/main/java/br/ufsc/cultivar/model/evaluate/AnswerTechnologyEnum.java
2582
2583 package br.ufsc.cultivar.model.evaluate;
2584
2585 public enum AnswerTechnologyEnum {
2586     NOTHING, BEGINNER, PROFICIENT, EXPERT
2587 }
2588
2589
2590 src/main/java/br/ufsc/cultivar/model/evaluate/Skill.java
2591
2592 package br.ufsc.cultivar.model.evaluate;
2593
2594 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2595 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2596 import lombok.AccessLevel;
2597 import lombok.Builder;
2598 import lombok.Value;
2599 import lombok.experimental.FieldDefaults;
2600
2601 import javax.validation.constraints.NotBlank;
2602 import javax.validation.constraints.NotNull;
2603
2604 @Value
```

```
2605 @Builder(builderClassName = "Builder")
2606 @JsonDeserialize(builder = Skill.Builder.class)
2607 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2608 public class Skill {
2609     Long codSkill;
2610     @NotNull
2611     @NotBlank
2612     String name;
2613
2614     @JsonPOJOBuilder(withPrefix = "")
2615     public static class Builder{
2616         public Builder(){}
2617     }
2618 }
2619
2620
2621 src/main/java/br/ufsc/cultivar/model/evaluate/Expectation.java
2622
2623 package br.ufsc.cultivar.model.evaluate;
2624
2625 public enum Expectation {
2626     ALL, SOME, NOTHING
2627 }
2628
2629
2630 src/main/java/br/ufsc/cultivar/model/evaluate/Difficulty.java
2631
2632 package br.ufsc.cultivar.model.evaluate;
2633
2634 public enum Difficulty {
2635     VERY_EASY, EASY, HARD, VERY_HARD
2636 }
2637
2638
2639 src/main/java/br/ufsc/cultivar/model/evaluate/Enjoy.java
2640
2641 package br.ufsc.cultivar.model.evaluate;
2642
2643 public enum Enjoy {
2644     VERY_FUNNY, FUNNY, BORED, VERY_BORED
2645 }
2646
2647
2648 src/main/java/br/ufsc/cultivar/model/evaluate/Technology.java
2649
```

```
2650 package br.ufsc.cultivar.model.evaluate;
2651
2652 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2653 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2654 import lombok.AccessLevel;
2655 import lombok.Builder;
2656 import lombok.Value;
2657 import lombok.experimental.FieldDefaults;
2658
2659 @Value
2660 @Builder(builderClassName = "Builder")
2661 @JsonDeserialize(builder = Technology.Builder.class)
2662 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2663 public class Technology {
2664
2665     Long codTechnology;
2666     String name;
2667
2668     @JsonPOJOBuilder(withPrefix = "")
2669     public static class Builder{
2670         public Builder(){}
2671     }
2672 }
2673
2674
2675 src/main/java(br/ufsc/cultivar/model/evaluate/AnswerPersonalityEnum.java
2676
2677 package br.ufsc.cultivar.model.evaluate;
2678
2679 public enum AnswerPersonalityEnum {
2680     TOTALLY_AGREE, AGREE, DISAGREE, TOTALLY_DISAGREE
2681 }
2682
2683
2684 src/main/java(br/ufsc/cultivar/model/evaluate/Experience.java
2685
2686 package br.ufsc.cultivar.model.evaluate;
2687
2688 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2689 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2690 import lombok.AccessLevel;
2691 import lombok.Builder;
2692 import lombok.Value;
2693 import lombok.experimental.FieldDefaults;
2694
```

```
2695     @Value
2696     @Builder(builderClassName = "Builder")
2697     @JsonDeserialize(builder = Experience.Builder.class)
2698     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2699     public class Experience {
2700
2701         String experience;
2702         Difficulty difficulty;
2703         Expectation expectation;
2704         Enjoy f1Enjoy;
2705         String enjoy;
2706         String notEnjoy;
2707         String suggest;
2708
2709         @JsonPOJOBuilder(withPrefix = "")
2710         public static class Builder{
2711             public Builder(){}
2712         }
2713     }
2714
2715
2716     src/main/java/br/ufsc/cultivar/model/evaluate/Personality.java
2717
2718     package br.ufsc.cultivar.model.evaluate;
2719
2720     import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2721     import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2722     import lombok.AccessLevel;
2723     import lombok.Builder;
2724     import lombok.Value;
2725     import lombok.experimental.FieldDefaults;
2726
2727     import javax.validation.constraints.NotBlank;
2728     import javax.validation.constraints.NotNull;
2729
2730     @Value
2731     @Builder(builderClassName = "Builder")
2732     @JsonDeserialize(builder = Personality.Builder.class)
2733     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2734     public class Personality {
2735
2736         Long codQuestion;
2737         @NotNull
2738         @NotBlank
2739         String question;
```

```
2740
2741     @JsonPOJOBuilder(withPrefix = "")
2742     public static class Builder{
2743         public Builder(){}
2744     }
2745 }
2746
2747
2748 src/main/java/br/ufsc/cultivar/model/evaluate/Mentoring.java
2749
2750 package br.ufsc.cultivar.model.evaluate;
2751
2752 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2753 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2754 import lombok.AccessLevel;
2755 import lombok.Builder;
2756 import lombok.Value;
2757 import lombok.experimental.FieldDefaults;
2758
2759 @Value
2760 @Builder(builderClassName = "Builder")
2761 @JsonDeserialize(builder = Mentoring.Builder.class)
2762 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2763 public class Mentoring {
2764     Long codQuestion;
2765     String question;
2766
2767     @JsonPOJOBuilder(withPrefix = "")
2768     public static class Builder{
2769         public Builder(){}
2770     }
2771 }
2772
2773
2774 src/main/java/br/ufsc/cultivar/model/evaluate/AnswerTechnology.java
2775
2776 package br.ufsc.cultivar.model.evaluate;
2777
2778 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2779 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2780 import lombok.AccessLevel;
2781 import lombok.Builder;
2782 import lombok.Value;
2783 import lombok.experimental.FieldDefaults;
2784
```

```
2785     @Value
2786     @Builder(builderClassName = "Builder")
2787     @JsonDeserialize(builder = Technology.Builder.class)
2788     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2789     public class AnswerTechnology {
2790         Technology technology;
2791         AnswerTechnologyEnum answer;
2792
2793         @JsonPOJOBuilder(withPrefix = "")
2794         public static class Builder{
2795             public Builder(){}
2796         }
2797     }
2798
2799
2800     src/main/java/br/ufsc/cultivar/model/evaluate/AnswerPersonality.java
2801
2802     package br.ufsc.cultivar.model.evaluate;
2803
2804     import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2805     import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2806     import lombok.AccessLevel;
2807     import lombok.Builder;
2808     import lombok.Value;
2809     import lombok.experimental.FieldDefaults;
2810
2811     @Value
2812     @Builder(builderClassName = "Builder")
2813     @JsonDeserialize(builder = AnswerPersonality.Builder.class)
2814     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2815     public class AnswerPersonality {
2816         Personality question;
2817         AnswerPersonalityEnum answer;
2818
2819         @JsonPOJOBuilder(withPrefix = "")
2820         public static class Builder{
2821             public Builder(){}
2822         }
2823     }
2824
2825
2826     src/main/java/br/ufsc/cultivar/model/Answer.java
2827
2828     package br.ufsc.cultivar.model;
2829
```

```
2830 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2831 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2832 import lombok.AccessLevel;
2833 import lombok.Builder;
2834 import lombok.Value;
2835 import lombok.experimental.FieldDefaults;
2836 import lombok.experimental.Wither;
2837
2838 import javax.validation.Valid;
2839 import javax.validation.constraints.NotBlank;
2840 import javax.validation.constraints.NotNull;
2841
2842 @Value
2843 @Wither
2844 @Builder(builderClassName = "Builder")
2845 @JsonDeserialize(builder = Answer.Builder.class)
2846 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2847 public class Answer {
2848     @Valid
2849     @NotNull
2850     Question question;
2851     @NotBlank
2852     @NotNull
2853     Boolean answer;
2854     String comment;
2855
2856     @JsonPOJOBuilder(withPrefix = "")
2857     public static class Builder{
2858         public Builder(){}
2859     }
2860 }
2861
2862
2863 src/main/java/br/ufsc/cultivar/model/Status.java
2864
2865 package br.ufsc.cultivar.model;
2866
2867 public enum Status {
2868     APPROVED,
2869     WAIT_STATEMENT,
2870     WAIT_COMPANY,
2871     WAIT_TRAINING,
2872     REGISTER,
2873
2874     public boolean isValid(Status next){
```

```
2875     switch (this){
2876         case APPROVED: return this.equals(next);
2877         case WAIT_TRAINING: return APPROVED.equals(next) ||
2878             this.equals(next);
2879         case WAIT_STATEMENT: return WAIT_TRAINING.equals(next)
2880             || this.equals(next);
2881         case WAIT_COMPANY: return WAIT_STATEMENT.equals(next)
2882             || this.equals(next);
2883         case REGISTER: return WAIT_COMPANY.equals(next) ||
2884             this.equals(next);
2885         default: return false;
2886     }
2887 }
2888 }
2889 src/main/java/br/ufsc/cultivar/model/School.java
2890
2891 package br.ufsc.cultivar.model;
2892
2893 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2894 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2895 import lombok.AccessLevel;
2896 import lombok.Builder;
2897 import lombok.Value;
2898 import lombok.experimental.FieldDefaults;
2899 import lombok.experimental.Wither;
2900
2901 import javax.validation.Valid;
2902 import javax.validation.constraints.NotBlank;
2903 import javax.validation.constraints.NotNull;
2904
2905 @Value
2906 @Wither
2907 @Builder(builderClassName = "Builder")
2908 @JsonDeserialize(builder = School.Builder.class)
2909 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2910 public class School {
2911     Long codSchool;
2912     @NotBlank
2913     @NotNull
2914     String name;
2915     @NotBlank
```

```
2916     @NotNull
2917     String phone;
2918     @Valid
2919     @NotNull
2920     Address address;
2921     @Valid
2922     @NotNull
2923     User responsible;
2924     @NotNull
2925     SchoolType type;
2926
2927     @JsonPOJOBuilder(withPrefix = "")
2928     public static class Builder{
2929         public Builder(){}
2930     }
2931 }
2932
2933
2934 src/main/java/br/ufsc/cultivar/model/Volunteer.java
2935
2936 package br.ufsc.cultivar.model;
2937
2938 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
2939 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
2940 import lombok.AccessLevel;
2941 import lombok.Builder;
2942 import lombok.Value;
2943 import lombok.experimental.FieldDefaults;
2944 import lombok.experimental.Wither;
2945
2946 import javax.validation.Valid;
2947 import javax.validation.constraints.NotBlank;
2948 import javax.validation.constraints.NotEmpty;
2949 import javax.validation.constraints.NotNull;
2950 import java.util.ArrayList;
2951 import java.util.List;
2952 import java.util.Optional;
2953
2954 @Value
2955 @Wither
2956 @Builder(builderClassName = "Builder")
2957 @JsonDeserialize(builder = Volunteer.Builder.class)
2958 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
2959 public class Volunteer {
2960     @Valid
```

```
2961     @NotNull
2962     User user;
2963     @Valid
2964     @NotNull
2965     Company company;
2966     @NotNull
2967     Schooling schooling;
2968     @NotNull
2969     Boolean conclusion;
2970     @NotBlank
2971     @NotNull
2972     String course;
2973     @NotBlank
2974     @NotNull
2975     String rg;
2976     @Valid
2977     @NotNull
2978     School school;
2979     List<@Valid Dispatch> dispatches;
2980     @NotNull
2981     @NotEmpty
2982     List<Answer> answers;
2983     List<@Valid Rating> ratings;
2984
2985     public Double getRating() {
2986         return Optional.ofNullable(ratings)
2987             .orElse(new ArrayList<>())
2988             .stream()
2989             .mapToInt(Rating::getGrade)
2990             .average()
2991             .orElse(0.);
2992     }
2993
2994     @JsonPOJOBuilder(withPrefix = "")
2995     public static class Builder{
2996         public Builder(){}
2997     }
2998 }
2999
3000
3001 src/main/java/br/ufsc/cultivar/model/User.java
3002
3003 package br.ufsc.cultivar.model;
3004
3005 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
```

```
3006 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3007 import lombok.AccessLevel;
3008 import lombok.Builder;
3009 import lombok.Value;
3010 import lombok.experimental.FieldDefaults;
3011 import lombok.experimental.Wither;
3012
3013 import javax.validation.Valid;
3014 import javax.validation.constraints.NotBlank;
3015 import javax.validation.constraints.NotNull;
3016 import java.util.Date;
3017
3018 @Value
3019 @Wither
3020 @Builder(builderClassName = "Builder")
3021 @JsonDeserialize(builder = User.Builder.class)
3022 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3023 public class User {
3024     String cpf;
3025     @NotBlank
3026     @NotNull
3027     String name;
3028     @NotBlank
3029     @NotNull
3030     String email;
3031     @NotBlank
3032     String password;
3033     @NotNull
3034     Role role;
3035     @NotNull
3036     Status status;
3037     @NotNull
3038     Date birth;
3039     @NotBlank
3040     @NotNull
3041     String job;
3042     @NotBlank
3043     @NotNull
3044     String phone;
3045     @Valid
3046     @NotNull
3047     Address address;
3048
3049     @JsonPOJOBuilder(withPrefix = "")
3050     public static class Builder{
```

```
3051     public Builder(){}
3052 }
3053 }
3054
3055
3056 src/main/java/br/ufsc/cultivar/model/SchoolType.java
3057
3058 package br.ufsc.cultivar.model;
3059
3060 public enum SchoolType {
3061     MUNICIPAL,
3062     STATE,
3063     FEDERAL
3064 }
3065
3066 src/main/java/br/ufsc/cultivar/model/Address.java
3067
3068 package br.ufsc.cultivar.model;
3069
3070 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3071 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3072 import lombok.AccessLevel;
3073 import lombok.Builder;
3074 import lombok.Value;
3075 import lombok.experimental.FieldDefaults;
3076 import lombok.experimental.Wither;
3077
3078 import javax.validation.constraints.NotBlank;
3079 import javax.validation.constraints.NotNull;
3080
3081 @Value
3082 @Wither
3083 @Builder(builderClassName = "Builder")
3084 @JsonDeserialize(builder = Address.Builder.class)
3085 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3086 public class Address {
3087     Long codAddress;
3088     @NotBlank
3089     @NotNull
3090     String city;
3091     @NotBlank
3092     @NotNull
3093     String neighborhood;
3094     @NotBlank
3095     @NotNull
```

```
3096     String street;
3097     String number;
3098
3099     @JsonPOJOBuilder(withPrefix = "")
3100     public static class Builder{
3101         public Builder(){}
3102     }
3103 }
3104
3105
3106 src/main/java/br/ufsc/cultivar/model/Event.java
3107
3108 package br.ufsc.cultivar.model;
3109
3110 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3111 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3112 import lombok.AccessLevel;
3113 import lombok.Builder;
3114 import lombok.Value;
3115 import lombok.experimental.FieldDefaults;
3116 import lombok.experimental.Wither;
3117
3118 import javax.validation.Valid;
3119 import javax.validation.constraints.NotBlank;
3120 import javax.validation.constraints.NotEmpty;
3121 import javax.validation.constraints.NotNull;
3122 import java.util.ArrayList;
3123 import java.util.Date;
3124 import java.util.List;
3125 import java.util.Optional;
3126
3127 @Value
3128 @Wither
3129 @Builder(builderClassName = "Builder")
3130 @JsonDeserialize(builder = Event.Builder.class)
3131 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3132 public class Event {
3133     Long codEvent;
3134     @NotBlank
3135     @NotNull
3136     Date startOccurrence;
3137     @NotBlank
3138     @NotNull
3139     Date endOccurrence;
3140     Date createdAt;
```

```
3141     @NotBlank
3142     @NotNull
3143     TypeEvent type;
3144     @NotNull
3145     Boolean allDay;
3146     @Valid
3147     @NotNull
3148     Address address;
3149     @NotNull
3150     Boolean evaluate;
3151     @Valid
3152     @NotNull
3153     School school;
3154     @Valid
3155     Project project;
3156     List<@Valid Training> trainings;
3157     @NotNull
3158     @NotEmpty
3159     List<@Valid User> participants;
3160     List<@Valid Rating> ratings;
3161
3162     public Double getRating() {
3163         return Optional.ofNullable(ratings)
3164             .orElseGet(ArrayList::new)
3165             .stream()
3166             .mapToInt(Rating::getGrade)
3167             .average()
3168             .orElse(0.);
3169     }
3170
3171     @JsonPOJOBuilder(withPrefix = "")
3172     public static class Builder{
3173         public Builder(){}
3174     }
3175 }
3176
3177
3178 src/main/java/br/ufsc/cultivar/model/Dispatch.java
3179
3180 package br.ufsc.cultivar.model;
3181
3182 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3183 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3184 import lombok.AccessLevel;
3185 import lombok.Builder;
```

```
3186 import lombok.Value;
3187 import lombok.experimental.FieldDefaults;
3188
3189 import javax.validation.Valid;
3190 import javax.validation.constraints.NotNull;
3191
3192 @Value
3193 @Builder(builderClassName = "Builder")
3194 @JsonDeserialize(builder = Dispatch.Builder.class)
3195 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3196 public class Dispatch {
3197     @NotNull
3198     @Valid
3199     Attachment attachment;
3200     Boolean send;
3201
3202     @JsonPOJOBuilder(withPrefix = "")
3203     public static class Builder{
3204         public Builder(){}
3205     }
3206 }
3207
3208
3209 src/main/java/br/ufsc/cultivar/model/Question.java
3210
3211 package br.ufsc.cultivar.model;
3212
3213 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3214 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3215 import lombok.AccessLevel;
3216 import lombok.Builder;
3217 import lombok.Value;
3218 import lombok.experimental.FieldDefaults;
3219
3220 import javax.validation.constraints.NotBlank;
3221 import javax.validation.constraints.NotNull;
3222
3223 @Value
3224 @Builder(builderClassName = "Builder")
3225 @JsonDeserialize(builder = Question.Builder.class)
3226 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3227 public class Question {
3228     Long codQuestion;
3229     @NotBlank
3230     @NotNull
```

```
3231     String question;
3232     @NotBlank
3233     @NotNull
3234     Role responds;
3235
3236     @JsonPOJOBuilder(withPrefix = "")
3237     public static class Builder{
3238         public Builder(){}
3239     }
3240 }
3241
3242
3243 src/main/java/br/ufsc/cultivar/model/Schooling.java
3244
3245 package br.ufsc.cultivar.model;
3246
3247 public enum Schooling {
3248     ELEMENTARY_SCHOOL,
3249     HIGH SCHOOL,
3250     UNIVERSITY_GRADUATE,
3251     POSTGRADUATE_STUDIES
3252 }
3253
3254
3255 src/main/java/br/ufsc/cultivar/model/Attachment.java
3256
3257 package br.ufsc.cultivar.model;
3258
3259 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3260 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3261 import lombok.AccessLevel;
3262 import lombok.Builder;
3263 import lombok.Value;
3264 import lombok.experimental.FieldDefaults;
3265
3266 import javax.validation.constraints.NotBlank;
3267 import javax.validation.constraints.NotNull;
3268
3269 @Value
3270 @Builder(builderClassName = "Builder")
3271 @JsonDeserialize(builder = Attachment.Builder.class)
3272 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3273 public class Attachment {
3274     Long codAttachment;
3275     @NotBlank
```

```
3276     @NotNull
3277     String name;
3278     @NotBlank
3279     @NotNull
3280     Boolean required;
3281     @NotBlank
3282     @NotNull
3283     Status status;
3284     Boolean download;
3285
3286     @JsonPOJOBuilder(withPrefix = "")
3287     public static class Builder{
3288         public Builder(){}
3289     }
3290 }
3291
3292
3293 src/main/java/br/ufsc/cultivar/model/Rating.java
3294
3295 package br.ufsc.cultivar.model;
3296
3297 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3298 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3299 import lombok.AccessLevel;
3300 import lombok.Builder;
3301 import lombok.Value;
3302 import lombok.experimental.FieldDefaults;
3303
3304 import javax.validation.constraints.NotBlank;
3305 import javax.validation.constraints.NotNull;
3306
3307 @Value
3308 @Builder(builderClassName = "Builder")
3309 @JsonDeserialize(builder = Rating.Builder.class)
3310 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3311 public class Rating {
3312     @NotBlank
3313     @NotNull
3314     Integer grade;
3315     String comment;
3316
3317     @JsonPOJOBuilder(withPrefix = "")
3318     public static class Builder{
3319         public Builder(){}
3320     }
```

```
3321     }
3322
3323
3324 src/main/java/br/ufsc/cultivar/model/Company.java
3325
3326 package br.ufsc.cultivar.model;
3327
3328 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3329 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3330 import lombok.AccessLevel;
3331 import lombok.Builder;
3332 import lombok.Value;
3333 import lombok.experimental.FieldDefaults;
3334 import lombok.experimental.Wither;
3335 import org.hibernate.validator.constraints.br.CNPJ;
3336
3337 import javax.validation.Valid;
3338 import javax.validation.constraints.NotBlank;
3339 import javax.validation.constraints.NotNull;
3340
3341 @Value
3342 @Wither
3343 @Builder(builderClassName = "Builder")
3344 @JsonDeserialize(builder = Company.Builder.class)
3345 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3346 public class Company {
3347     @NotBlank
3348     @NotNull
3349     @CNPJ
3350     String cnpj;
3351     @NotBlank
3352     @NotNull
3353     String name;
3354     @NotBlank
3355     @NotNull
3356     String phone;
3357     @Valid
3358     @NotNull
3359     Address address;
3360     @Valid
3361     @NotNull
3362     User responsible;
3363
3364     @JsonPOJOBuilder(withPrefix = "")
3365     public static class Builder{
```

```
3366     public Builder(){}
3367 }
3368 }
3369
3370
3371 src/main/java/br/ufsc/cultivar/model/Project.java
3372
3373 package br.ufsc.cultivar.model;
3374
3375 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;
3376 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;
3377 import lombok.AccessLevel;
3378 import lombok.Builder;
3379 import lombok.Value;
3380 import lombok.experimental.FieldDefaults;
3381
3382 import javax.validation.constraints.NotBlank;
3383 import javax.validation.constraints.NotNull;
3384 import java.util.Date;
3385
3386 @Value
3387 @Builder(builderClassName = "Builder")
3388 @JsonDeserialize(builder = Project.Builder.class)
3389 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3390 public class Project {
3391
3392     Long codProject;
3393     @NotNull
3394     @NotBlank
3395     String name;
3396     @NotNull
3397     Date start;
3398     @NotNull
3399     Date end;
3400
3401     @JsonPOJOBuilder(withPrefix = "")
3402     public static class Builder{
3403         public Builder(){}
3404     }
3405 }
3406
3407
3408 src/main/java/br/ufsc/cultivar/model/TypeEvent.java
3409
3410 package br.ufsc.cultivar.model;
```

```
3411  
3412 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;  
3413 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;  
3414 import lombok.AccessLevel;  
3415 import lombok.Builder;  
3416 import lombok.Value;  
3417 import lombok.experimental.FieldDefaults;  
3418 import lombok.experimental.Wither;  
3419  
3420 import javax.validation.Valid;  
3421 import javax.validation.constraints.NotBlank;  
3422 import javax.validation.constraints.NotEmpty;  
3423 import javax.validation.constraints.NotNull;  
3424 import java.util.List;  
3425  
3426 @Value  
3427 @Wither  
3428 @Builder(builderClassName = "Builder")  
3429 @JsonDeserialize(builder = TypeEvent.Builder.class)  
3430 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
3431 public class TypeEvent {  
3432  
3433     Long type;  
3434     @NotNull  
3435     @NotBlank  
3436     String name;  
3437     @NotNull  
3438     @NotEmpty  
3439     List<@Valid @NotNull Training> trainings;  
3440  
3441     @JsonPOJOBuilder(withPrefix = "")  
3442     public static class Builder{  
3443         public Builder(){  
3444     }  
3445 }  
3446  
3447  
3448 src/main/java/br/ufsc/cultivar/model/Role.java  
3449  
3450 package br.ufsc.cultivar.model;  
3451  
3452 public enum Role {  
3453     ADMIN,  
3454     VOLUNTEER,  
3455     COMPANY_ADMIN,
```

```
3456     SCHOOL_ADMIN  
3457 }  
3458  
3459  
3460 src/main/java/br/ufsc/cultivar/model/Training.java  
3461  
3462 package br.ufsc.cultivar.model;  
3463  
3464 import com.fasterxml.jackson.databind.annotation.JsonDeserialize;  
3465 import com.fasterxml.jackson.databind.annotation.JsonPOJOBuilder;  
3466 import lombok.AccessLevel;  
3467 import lombok.Builder;  
3468 import lombok.Value;  
3469 import lombok.experimental.FieldDefaults;  
3470 import lombok.experimental.Wither;  
3471  
3472 import javax.validation.Valid;  
3473 import javax.validation.constraints.NotNull;  
3474  
3475 @Value  
3476 @Wither  
3477 @Builder(builderClassName = "Builder")  
3478 @JsonDeserialize(builder = Training.Builder.class)  
3479 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
3480 public class Training {  
3481     Long codTraining;  
3482     @Valid  
3483     @NotNull  
3484     String name;  
3485     String path;  
3486     String link;  
3487  
3488     @JsonPOJOBuilder(withPrefix = "")  
3489     public static class Builder{  
3490         public Builder(){}
3491     }
3492 }
3493  
3494  
3495 src/main/java/br/ufsc/cultivar/service/evaluate/SkillService.java  
3496  
3497 package br.ufsc.cultivar.service.evaluate;  
3498  
3499 import br.ufsc.cultivar.exception.ServiceException;
3500 import br.ufsc.cultivar.exception.Type;
```

```
3501 import br.ufsc.cultivar.model.evaluate.Skill;
3502 import br.ufsc.cultivar.repository.evaluate.SkillRepository;
3503 import lombok.AccessLevel;
3504 import lombok.AllArgsConstructor;
3505 import lombok.experimental.FieldDefaults;
3506 import lombok.val;
3507 import org.springframework.beans.factory.annotation.Autowired;
3508 import org.springframework.stereotype.Service;
3509
3510 import java.util.List;
3511 import java.util.Optional;
3512
3513 @Service
3514 @AllArgsConstructor(onConstructor = @_(@Autowired))
3515 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3516 public class SkillService {
3517     SkillRepository repository;
3518
3519     public void create(Skill skill){
3520         repository.create(skill);
3521     }
3522
3523     public List<Skill> get(){
3524         return repository.get();
3525     }
3526
3527     public Skill delete(Long codSkill) throws ServiceException {
3528         val skill = Optional.ofNullable(
3529             repository.get(codSkill)
3530         ).orElseThrow(
3531             () -> new ServiceException(null, null, Type.NOT_FOUND)
3532         );
3533         repository.delete(codSkill);
3534         return skill;
3535     }
3536 }
3537
3538
3539 src/main/java;br/ufsc/cultivar/service/evaluate/TechnologyService.java
3540
3541 package br.ufsc.cultivar.service.evaluate;
3542
3543 import br.ufsc.cultivar.exception.ServiceException;
3544 import br.ufsc.cultivar.exception.Type;
3545 import br.ufsc.cultivar.model.evaluate.Technology;
```

```
3546 import br.ufsc.cultivar.repository.evaluate.TechologyRepository;
3547 import lombok.AccessLevel;
3548 import lombok.AllArgsConstructor;
3549 import lombok.experimental.FieldDefaults;
3550 import org.springframework.beans.factory.annotation.Autowired;
3551 import org.springframework.stereotype.Service;
3552
3553 import java.util.List;
3554 import java.util.Optional;
3555
3556 @Service
3557 @AllArgsConstructor(onConstructor = @_(@Autowired))
3558 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3559 public class TechologyService {
3560     TechologyRepository repository;
3561
3562     public void create(Techology technology) throws
3563         ServiceException {
3564         repository.create(technology);
3565     }
3566
3567     public List<Techology> get() throws ServiceException {
3568         return repository.get();
3569     }
3570
3571     public Techology delete(Long codTechnology) throws
3572         ServiceException {
3573         Techology technology = repository.get(codTechnology);
3574         Optional.ofNullable(technology).orElseThrow(() -> new
3575             ServiceException(null, null, Type.NOT_FOUND));
3576         repository.delete(codTechnology);
3577         return technology;
3578     }
3579
3580 src/main/java;br\ufsc\cultivar\service\evaluate\MentoringService.java
3581
3582 package br.ufsc.cultivar.service.evaluate;
3583
3584 import br.ufsc.cultivar.model.evaluate.Mentoring;
3585 import br.ufsc.cultivar.repository.evaluate.MentoringRepository;
3586 import lombok.AccessLevel;
3587 import lombok.AllArgsConstructor;
```

```
3588 import lombok.experimental.FieldDefaults;
3589 import lombok.val;
3590 import org.springframework.beans.factory.annotation.Autowired;
3591 import org.springframework.stereotype.Service;
3592
3593 import java.util.List;
3594
3595 @Service
3596 @AllArgsConstructor(onConstructor = @_(@Autowired))
3597 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3598 public class MentoringService {
3599
3600     MentoringRepository repository;
3601
3602     public void create(Mentoring mentoring){
3603         repository.create(mentoring);
3604     }
3605
3606     public List<Mentoring> get(){
3607         return repository.get();
3608     }
3609
3610     public Mentoring delete(Long codQuestion){
3611         val mentoring = repository.get(codQuestion);
3612         repository.delete(codQuestion);
3613         return mentoring;
3614     }
3615 }
3616
3617
3618 src/main/java/br/ufsc/cultivar/service/evaluate/EvaluateService.java
3619
3620 package br.ufsc.cultivar.service.evaluate;
3621
3622 import br.ufsc.cultivar.dto.EvaluateDTO;
3623 import br.ufsc.cultivar.exception.ServiceException;
3624 import br.ufsc.cultivar.repository.evaluate.EvaluateRepository;
3625 import lombok.AccessLevel;
3626 import lombok.AllArgsConstructor;
3627 import lombok.experimental.FieldDefaults;
3628 import lombok.val;
3629 import org.springframework.beans.factory.annotation.Autowired;
3630 import org.springframework.stereotype.Service;
3631
3632 import java.util.List;
```

```
3633 import java.util.stream.Collectors;
3634
3635 @Service
3636 @AllArgsConstructor(onConstructor = @_(@Autowired))
3637 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3638 public class EvaluateService {
3639     EvaluateRepository repository;
3640
3641     public void create(EvaluateDTO dto) throws ServiceException {
3642         val cpf = dto.getCpf();
3643         val codProject = dto.getProject();
3644         dto.getTechnologies()
3645             .forEach(
3646                 answerTechnology ->
3647                     repository.saveTechnology(answerTechnology,
3648                         codProject, cpf)
3649             );
3650         dto.getAnswerPersonalities()
3651             .forEach(
3652                 answerVolunteer ->
3653                     repository.saveAnswerVolunteer(answerVolunteer,
3654                         codProject, cpf)
3655             );
3656         repository.saveExperience(dto.getExperience(),
3657             codProject, cpf);
3658     }
3659
3660     public List<EvaluateDTO> get(Long codProject) {
3661         return repository.getCpfs(codProject)
3662             .stream()
3663             .map(
3664                 cpf -> EvaluateDTO.builder()
3665                     .cpf(cpf)
3666                     .project(codProject)
3667                     .answerPersonalities(
3668                         repository.getAnswersVolunteer(codProject,
3669                             cpf)
3670                         .technologies(
3671                             repository.getAnswersTechnologies(codProject,
3672                                 cpf)
3673                         .build()
3674                     ).collect(Collectors.toList());
3675     }
3676 }
```

```
3671  
3672 src/main/java/br/ufsc/cultivar/service/evaluate/PersonalityService.java  
3673  
3674 package br.ufsc.cultivar.service.evaluate;  
3675  
3676 import br.ufsc.cultivar.model.evaluate.Personality;  
3677 import  
     br.ufsc.cultivar.repository.evaluate.PersonalityRepository;  
3678 import lombok.AccessLevel;  
3679 import lombok.AllArgsConstructor;  
3680 import lombok.experimental.FieldDefaults;  
3681 import org.springframework.beans.factory.annotation.Autowired;  
3682 import org.springframework.stereotype.Service;  
3683  
3684 import java.util.List;  
3685  
3686 @Service  
@AllArgsConstructor(onConstructor = @_(_(@Autowired)))  
@FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
3689 public class PersonalityService {  
3690     PersonalityRepository repository;  
3691  
3692     public void create(Personality question) {  
3693         repository.create(question);  
3694     }  
3695  
3696     public List<Personality> get() {  
3697         return repository.get();  
3698     }  
3699  
3700     public Personality delete(Long codQuestion) {  
3701         Personality question = repository.get(codQuestion);  
3702         repository.delete(codQuestion);  
3703         return question;  
3704     }  
3705 }  
3706 }  
3707  
3708  
3709 src/main/java/br/ufsc/cultivar/service/UserService.java  
3710  
3711 package br.ufsc.cultivar.service;  
3712  
3713 import br.ufsc.cultivar.exception.ServiceException;  
3714 import br.ufsc.cultivar.model.User;
```

```
3715 import br.ufsc.cultivar.repository.AddressRepository;
3716 import br.ufsc.cultivar.repository.UserRepository;
3717 import br.ufsc.cultivar.utils.ValidateUtils;
3718 import lombok.AccessLevel;
3719 import lombok.AllArgsConstructor;
3720 import lombok.experimental.FieldDefaults;
3721 import lombok.val;
3722 import org.springframework.beans.factory.annotation.Autowired;
3723 import org.springframework.stereotype.Service;
3724
3725 import java.util.List;
3726 import java.util.Map;
3727
3728 @Service
3729 @AllArgsConstructor(onConstructor = @_(@Autowired))
3730 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3731 public class UserService {
3732
3733     UserRepository userRepository;
3734     AddressRepository addressRepository;
3735
3736     public void create(final User user) throws ServiceException {
3737         val address = user.getAddress();
3738         if (!ValidateUtils.isValid(address)) {
3739             throw new ServiceException(null, null, null);
3740         }
3741         userRepository.create(
3742             user.withAddress(
3743                 address.withCodAddress(
3744                     addressRepository.create(
3745                         address
3746                     )
3747                 )
3748             );
3749     };
3750
3751 }
3752
3753     public List<User> get(final Map<String, Object> filter)
3754         throws ServiceException {
3755         try {
3756             return userRepository.get(filter);
3757         } catch (RuntimeException e){
3758             val throwable = e.getCause();
3759             if (throwable instanceof ServiceException) {
```

```
3759             throw (ServiceException) throwable;
3760         }
3761         throw e;
3762     }
3763 }
3764
3765 public User get(final String cpf) throws ServiceException {
3766     val user = userRepository.get(cpf);
3767     return user.withAddress(
3768         addressRepository.get(
3769             user.getAddress().getCodAddress()
3770         )
3771     );
3772 }
3773
3774 public User delete(final String cpf) throws ServiceException
3775 {
3776     val user = get(cpf);
3777     userRepository.delete(cpf);
3778     return user;
3779 }
3780
3781 public void update(final User user, final String cpf) throws
3782     ServiceException {
3783     if (!user.getCpf().equals(cpf)) {
3784         throw new ServiceException(null, null, null);
3785     }
3786     if (!get(cpf).getStatus().isValid(user.getStatus())){
3787         throw new ServiceException(null, null, null);
3788     }
3789     userRepository.update(user);
3790 }
3791
3792 List<User> getParticipants(final Long codEvent) {
3793     return userRepository.getParticipants(codEvent);
3794 }
3795
3796 src/main/java/br/ufsc/cultivar/service/CompanyService.java
3797
3798 package br.ufsc.cultivar.service;
3799
3800 import br.ufsc.cultivar.dto.PaginateList;
3801 import br.ufsc.cultivar.exception.ServiceException;
3802 import br.ufsc.cultivar.model.Company;
```

```
3802 import br.ufsc.cultivar.repository.AddressRepository;
3803 import br.ufsc.cultivar.repository.CompanyRepository;
3804 import lombok.AccessLevel;
3805 import lombok.AllArgsConstructorConstructor;
3806 import lombok.experimental.FieldDefaults;
3807 import lombok.val;
3808 import org.springframework.beans.factory.annotation.Autowired;
3809 import org.springframework.stereotype.Service;
3810
3811 import java.util.ArrayList;
3812 import java.util.List;
3813 import java.util.Map;
3814 import java.util.Optional;
3815 import java.util.stream.Collectors;
3816
3817 @Service
3818 @AllArgsConstructor(onConstructor = @__(@Autowired))
3819 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
3820 public class CompanyService {
3821
3822     CompanyRepository companyRepository;
3823     UserService userService;
3824     AddressRepository addressRepository;
3825
3826     public void create(final Company company) throws
3827         ServiceException {
3828         userService.create(company.getResponsible());
3829         val address = company.getAddress();
3830         companyRepository.create(
3831             company.withAddress(
3832                 address.withCodAddress(
3833                     addressRepository.create(
3834                         address
3835                     )
3836                 )
3837             );
3838     }
3839
3840     public PaginateList get(final String filter, final Long
3841         page) throws ServiceException {
3842         return PaginateList.builder()
3843             .count(companyRepository.count(filter))
3844             .data(
```

```
3844         new ArrayList<>(companyRepository.get(filter,
3845             page))
3846     ).build();
3847 }
3848
3849     public Company get(final String cnpj) throws
3850         ServiceException {
3851         val company = companyRepository.get(cnpj);
3852         return Optional.ofNullable(company)
3853             .orElseThrow(
3854                 () -> new ServiceException(null, null, null))
3855             .withAddress(
3856                 addressRepository.get(
3857                     company.getAddress()
3858                         .getCodAddress()
3859                 )
3860             .withResponsible(
3861                 userService.get(
3862                     company.getResponsible()
3863                         .getCpf()
3864                 )
3865             );
3866
3867     public Company delete(String cnpj) throws ServiceException {
3868         val company = get(cnpj);
3869         companyRepository.delete(cnpj);
3870         return Optional.ofNullable(company).orElseThrow(() ->
3871             new ServiceException(null, null, null));
3872     }
3873
3874     public void update(Company company, String cnpj) throws
3875         ServiceException {
3876         if(!company.getCnpj().equals(cnpj)){
3877             throw new ServiceException(null, null, null);
3878         }
3879         addressRepository.update(company.getAddress());
3880         companyRepository.update(company);
3881     }
3882     src/main/java/br/ufsc/cultivar/service/VolunteerService.java
3883
3884     package br.ufsc.cultivar.service;
```

```
3885  
3886 import br.ufsc.cultivar.dto.PaginateList;  
3887 import br.ufsc.cultivar.email.EmailClient;  
3888 import br.ufsc.cultivar.exception.ServiceException;  
3889 import br.ufsc.cultivar.exception.Type;  
3890 import br.ufsc.cultivar.model.Volunteer;  
3891 import br.ufsc.cultivar.repository.*;  
3892 import br.ufsc.cultivar.utils.ValidateUtils;  
3893 import lombok.AccessLevel;  
3894 import lombok.AllArgsConstructorConstructor;  
3895 import lombok.experimental.FieldDefaults;  
3896 import lombok.extern.slf4j.Slf4j;  
3897 import lombok.val;  
3898 import org.springframework.beans.factory.annotation.Autowired;  
3899 import org.springframework.stereotype.Service;  
3900 import org.thymeleaf.context.Context;  
3901 import org.thymeleaf.spring5.SpringTemplateEngine;  
3902  
3903 import java.util.ArrayList;  
3904 import java.util.Date;  
3905 import java.util.List;  
3906 import java.util.Optional;  
3907 import java.util.stream.Collectors;  
3908  
3909 @Slf4j  
3910 @Service  
3911 @AllArgsConstructor(onConstructor = @__(@Autowired))  
3912 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
3913 public class VolunteerService {  
3914  
3915     VolunteerRepository volunteerRepository;  
3916     UserService userService;  
3917     CompanyService companyService;  
3918     RatingRepository ratingRepository;  
3919     AnswerRepository answerRepository;  
3920     QuestionRepository questionRepository;  
3921     EventRepository eventRepository;  
3922     DispatchService dispatchService;  
3923     EmailClient emailClient;  
3924     SpringTemplateEngine templateEngine;  
3925  
3926     public void create(final Volunteer volunteer) throws  
3927         ServiceException {  
3928         val user = volunteer.getUser();  
             if (!ValidateUtils.isValid(user)) {
```

```
3929         throw new ServiceException(null, null, null);
3930     }
3931     userService.create(user);
3932     volunteerRepository.create(volunteer);
3933     Optional.ofNullable(volunteer.getAnswers())
3934         .ifPresent(
3935             answers -> answers.forEach(
3936                 answer -> answerRepository.create(
3937                     answer,
3938                     user.getCpf()
3939                     )
3940                 )
3941             );
3942     emailClient.sendEmailAsync(user.getEmail(), "Bem vindo",
3943         buildEmail());
3944 }
3945
3946 private String buildEmail(){
3947     val context = new Context();
3948     context.setVariable("message", "teste volunteer");
3949     return templateEngine.process("newVolunteerEmail",
3950         context);
3951 }
3952
3953 public Volunteer get(final String cpf) throws
3954     ServiceException {
3955     val volunteer = volunteerRepository.get(cpf);
3956     return Optional.ofNullable(volunteer)
3957         .orElseThrow(() -> new ServiceException(null,
3958             null, Type.NOT_FOUND))
3959         .withUser(
3960             userService.get(cpf)
3961         ).withCompany(
3962             companyService.get(
3963                 volunteer.getCompany().getCnpj()
3964             )
3965         ).withAnswers(
3966             answerRepository.get(cpf)
3967                 .stream()
3968                 .map(answer -> answer.withQuestion(
3969                     questionRepository.get(
3970                         answer.getQuestion().getCodQuestion()
3971                     )
3972                 )
3973             ).collect(Collectors.toList())
3974 }
```

```

3970         ).withRatings(
3971             ratingRepository.get(cpf)
3972         ).withDispatches(
3973             dispatchService.get(cpf)
3974         );
3975     }
3976
3977     public Volunteer delete(final String cpf) throws
3978         ServiceException {
3979         val hasEvents = eventRepository.eventsByVolunteer(cpf,
3980             null)
3981             .stream()
3982             .filter(
3983                 event -> event.getEndOccurrence()
3984                     .after(new Date())
3985             ).collect(Collectors.toList())
3986             .isEmpty();
3987         if(hasEvents){
3988             throw new ServiceException(null, null, null);
3989         }
3990         val volunteer = get(cpf);
3991         volunteerRepository.delete(cpf);
3992         return volunteer;
3993     }
3994
3995     public void update(final Volunteer volunteer, final String
3996         cpf) throws ServiceException {
3997         val user = volunteer.getUser();
3998         if (!user.getCpf().equals(cpf)){
3999             throw new ServiceException(null, null, null);
4000         }
4001         userService.update(user, cpf);
4002         answerRepository.delete(cpf);
4003         volunteer.getAnswers()
4004             .forEach(answer ->
4005                 answerRepository.create(answer, cpf));
4006         volunteerRepository.update(volunteer);
4007     }
4008
4009     public PaginateList get(final List<String> filterCompany,
4010         final List<Long> filterSchool,
4011             final String filter, final Integer
4012                 page) throws ServiceException {
4013         try {
4014             return PaginateList.builder()

```

```
4009     .count(volunteerRepository.count(filterCompany,
4010             filterSchool, filter))
4011     .data(
4012         volunteerRepository.get(filterCompany,
4013             filterSchool, filter, page)
4014         .stream()
4015         .map(volunteer -> {
4016             try {
4017                 return volunteer.withUser(
4018                     userService.get(
4019                         volunteer.getUser()
4020                             .getCPF()
4021                         )
4022                     ).withRatings(
4023                         ratingRepository.get(
4024                             volunteer.getUser()
4025                             .getCPF()
4026                         )
4027                     );
4028             } catch (ServiceException e) {
4029                 throw new RuntimeException(e);
4030             }
4031         }).collect(Collectors.toList())
4032     ).build();
4033 } catch (RuntimeException e){
4034     throw new ServiceException(null, null, null);
4035 }
4036 }
4037
4038 src/main/java/br/ufsc/cultivar/service/TypeEventService.java
4039
4040 package br.ufsc.cultivar.service;
4041
4042 import br.ufsc.cultivar.dto.PaginateList;
4043 import br.ufsc.cultivar.exception.ServiceException;
4044 import br.ufsc.cultivar.exception.Type;
4045 import br.ufsc.cultivar.model.Training;
4046 import br.ufsc.cultivar.model.TypeEvent;
4047 import br.ufsc.cultivar.repository.TrainingRepository;
4048 import br.ufsc.cultivar.repository.TypeEventRepository;
4049 import lombok.AccessLevel;
4050 import lombok.AllArgsConstructor;
4051 import lombok.experimental.FieldDefaults;
```

```
4052 import lombok.val;
4053 import org.springframework.beans.factory.annotation.Autowired;
4054 import org.springframework.stereotype.Service;
4055
4056 import java.util.ArrayList;
4057 import java.util.List;
4058 import java.util.Optional;
4059
4060 @Service
4061 @AllArgsConstructor(onConstructor = @__(@Autowired))
4062 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4063 public class TypeEventService {
4064
4065     TypeEventRepository typeEventRepository;
4066     TrainingRepository trainingRepository;
4067
4068     public void create(TypeEvent typeEvent) {
4069         val tpEvent =
4070             typeEventRepository.create(typeEvent.getName());
4071         typeEvent.getTrainings()
4072             .forEach(
4073                 training ->
4074                     trainingRepository.create(training,
4075                         tpEvent)
4076             );
4077     }
4078
4079     public PaginateList get(final String filter, final Long
4080                             page) {
4081         return PaginateList.builder()
4082             .count(typeEventRepository.count(filter))
4083             .data(
4084                 new
4085                     ArrayList<>(typeEventRepository.get(filter,
4086                                     page))
4087             ).build();
4088     }
4089
4090     public TypeEvent get(Long tpEvent) {
4091         return typeEventRepository.get(tpEvent)
4092             .withTrainings(trainingRepository.getByTypeEvent(tpEvent));
4093     }
4094
4095     public TypeEvent delete(Long tpEvent) {
4096         val typeEvent = get(tpEvent);
4097
4098
4099 }
```

```
4091         typeEventRepository.delete(tpEvent);
4092         return typeEvent;
4093     }
4094
4095     public List<Training> getTrainings(Long tpEvent) {
4096         return trainingRepository.getByTypeEvent(tpEvent);
4097     }
4098 }
4099
4100 src/main/java/br/ufsc/cultivar/service/DispatchService.java
4102
4103 package br.ufsc.cultivar.service;
4104
4105 import br.ufsc.cultivar.exception.ServiceException;
4106 import br.ufsc.cultivar.exception.Type;
4107 import br.ufsc.cultivar.exception.UploadException;
4108 import br.ufsc.cultivar.model.Attachment;
4109 import br.ufsc.cultivar.model.Dispatch;
4110 import br.ufsc.cultivar.repository.DispatchRepository;
4111 import br.ufsc.cultivar.utils.FileUtils;
4112 import lombok.AccessLevel;
4113 import lombok.AllArgsConstructor;
4114 import lombok.experimental.FieldDefaults;
4115 import org.springframework.beans.factory.annotation.Autowired;
4116 import org.springframework.stereotype.Service;
4117 import org.springframework.web.multipart.MultipartFile;
4118
4119 import java.util.List;
4120 import java.util.Optional;
4121
4122 @Service
4123 @AllArgsConstructor(onConstructor = @_(@Autowired))
4124 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4125 public class DispatchService {
4126
4127     DispatchRepository repository;
4128     FileUtils fileUtils;
4129
4130     public void save(final Attachment attachment, final
4131                     MultipartFile file, final String cpf) throws
4132                     ServiceException {
4133         try {
4134             repository.save(fileUtils.save(attachment, file,
4135                                     cpf), cpf);
4136         } catch (Exception e) {
4137             throw new ServiceException("Error saving attachment");
4138         }
4139     }
4140 }
```

```
4133     } catch (UploadException e) {
4134         throw new ServiceException(null, null, Type.FILE);
4135     }
4136 }
4137
4138 public Dispatch get(final String cpf, final Long
4139 codAttachment) {
4140     return repository.get(cpf, codAttachment);
4141 }
4142
4143 public List<Dispatch> get(String cpf) throws
4144 ServiceException {
4145     return Optional.ofNullable(repository.get(cpf))
4146         .orElseThrow(() -> new ServiceException(null,
4147             null, Type.NOT_FOUND));
4148 }
4149
4150 src/main/java/br/ufsc/cultivar/service/AttachmentService.java
4151
4152 package br.ufsc.cultivar.service;
4153
4154 import br.ufsc.cultivar.dto.PaginateList;
4155 import br.ufsc.cultivar.exception.ServiceException;
4156 import br.ufsc.cultivar.exception.Type;
4157 import br.ufsc.cultivar.exception.UploadException;
4158 import br.ufsc.cultivar.model.Attachment;
4159 import br.ufsc.cultivar.model.Status;
4160 import br.ufsc.cultivar.repository.AttachmentRepository;
4161 import br.ufsc.cultivar.utils.FileUtils;
4162 import lombok.AccessLevel;
4163 import lombok.AllArgsConstructor;
4164 import lombok.experimental.FieldDefaults;
4165 import lombok.experimental.val;
4166 import org.springframework.beans.factory.annotation.Autowired;
4167 import org.springframework.core.io.Resource;
4168 import org.springframework.stereotype.Service;
4169 import org.springframework.web.multipart.MultipartFile;
4170 import java.util.ArrayList;
4171 import java.util.List;
4172 import java.util.Objects;
4173 import java.util.Optional;
4174
```

```
4175     @Service
4176     @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
4177     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4178     public class AttachmentService {
4179
4180         AttachmentRepository repository;
4181         FileUtils fileUtils;
4182
4183         public void create(final Attachment attachment, final
4184             MultipartFile file) throws ServiceException {
4185             if(attachment.getDownload() && Objects.isNull(file)){
4186                 throw new ServiceException(null, null, null);
4187             }
4188             val codAttachment = repository.create(attachment);
4189             if (attachment.getDownload()) {
4190                 try {
4191                     fileUtils.saveAttachment(file, codAttachment);
4192                 } catch (UploadException e) {
4193                     throw new ServiceException(e.getMessage(), e,
4194                         Type.FILE);
4195                 }
4196             }
4197         }
4198
4199         public PaginateList get(final String filter, final Long
4200             page) throws ServiceException {
4201             return PaginateList.builder()
4202                 .data(
4203                     new ArrayList<>(repository.get(filter,
4204                         page)))
4205                 .count(
4206                     repository.count(filter)
4207                 ).build();
4208         }
4209         public List<Attachment> get(final Status status) throws
4210             ServiceException {
4211             return repository.get(status);
4212         }
4213
4214         public Attachment get(final Long codAttachment) throws
4215             ServiceException {
4216             val attachment = repository.get(codAttachment);
4217             return Optional.ofNullable(attachment)
4218                 .orElseThrow(() -> new ServiceException(null, null,
4219                     null));
4220     }
```

```
4213     }
4214
4215     public Resource getAsFile(final Long codAttachment) throws
4216         ServiceException {
4216         return fileUtils.get(
4217             Optional.ofNullable(repository.get(codAttachment))
4218                 .orElseThrow(
4219                     () -> new ServiceException(null, null,
4220                         Type.NOT_FOUND)
4221                 ).getCodAttachment()
4222         );
4223
4224     public Attachment delete(final Long codAttachment) throws
4225         ServiceException {
4226         val attachment = get(codAttachment);
4227         repository.delete(codAttachment);
4228         return attachment;
4229     }
4230
4231     public void update(final Attachment attachment, final Long
4232         codAttachment) throws ServiceException {
4233         if (attachment.getCodAttachment().equals(codAttachment)){
4234             throw new ServiceException(null, null, null);
4235         }
4236         repository.update(attachment);
4237     }
4238
4239     src/main/java/br/ufsc/cultivar/service/ProjectService.java
4240
4241     package br.ufsc.cultivar.service;
4242
4243     import br.ufsc.cultivar.dto.PaginateList;
4244     import br.ufsc.cultivar.model.Project;
4245     import br.ufsc.cultivar.repository.ProjectRepository;
4246     import lombok.AccessLevel;
4247     import lombok.AllArgsConstructor;
4248     import lombok.experimental.FieldDefaults;
4249     import lombok.val;
4250     import org.springframework.beans.factory.annotation.Autowired;
4251     import org.springframework.stereotype.Service;
4252
4253     import java.util.ArrayList;
```

```
4254 import java.util.List;  
4255  
4256 @Service  
4257 @AllArgsConstructor(onConstructor = @_(@Autowired))  
4258 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
4259 public class ProjectService {  
4260  
4261     ProjectRepository projectRepository;  
4262  
4263     public void create(Project project) {  
4264         projectRepository.create(project);  
4265     }  
4266  
4267     public PaginateList get(final String filter, final Long  
4268         page) {  
4269         return PaginateList.builder()  
4270             .count(projectRepository.count(filter))  
4271             .data(  
4272                 new ArrayList<>(projectRepository.get(filter,  
4273                     page))  
4274             ).build();  
4275     }  
4276  
4277     public Project get(Long codProject) {  
4278         return projectRepository.get(codProject);  
4279     }  
4280  
4281     public Project delete(Long codProject) {  
4282         val project = get(codProject);  
4283         projectRepository.delete(codProject);  
4284         return project;  
4285     }  
4286  
4287     src/main/java/br/ufsc/cultivar/service/QuestionService.java  
4288  
4289     package br.ufsc.cultivar.service;  
4290  
4291     import br.ufsc.cultivar.exception.ServiceException;  
4292     import br.ufsc.cultivar.model.Question;  
4293     import br.ufsc.cultivar.model.Role;  
4294     import br.ufsc.cultivar.repository.QuestionRepository;  
4295     import lombok.AccessLevel;  
4296     import lombok.AllArgsConstructor;
```

```
4297 import lombok.experimental.FieldDefaults;
4298 import lombok.val;
4299 import org.springframework.beans.factory.annotation.Autowired;
4300 import org.springframework.stereotype.Service;
4301
4302 import java.util.List;
4303 import java.util.Map;
4304
4305 @Service
4306 @AllArgsConstructor(onConstructor = @__(@Autowired))
4307 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4308 public class QuestionService {
4309
4310     QuestionRepository repository;
4311
4312     public void create(final Question question) throws
4313         ServiceException {
4314         repository.create(question);
4315     }
4316
4317     public List<Question> get() throws ServiceException {
4318         return repository.get();
4319     }
4320
4321     public List<Question> get(Role responds) throws
4322         ServiceException {
4323         return repository.get(responds);
4324     }
4325
4326     public Question get(final Long codQuestion) throws
4327         ServiceException {
4328         return repository.get(codQuestion);
4329     }
4330
4331     public Question delete(final Long codQuestion) throws
4332         ServiceException {
4333         val question = get(codQuestion);
4334         repository.delete(codQuestion);
4335         return question;
4336     }
4337
4338     public void update(final Question question, final Long
4339         codQuestion) throws ServiceException {
4340         if (question.getCodQuestion().equals(codQuestion)){
4341             throw new ServiceException(null, null, null);
4342         }
4343     }
4344 }
```

```
4337         }
4338         repository.update(question);
4339     }
4340 }
4341
4342
4343 src/main/java/br/ufsc/cultivar/service/SchoolService.java
4344
4345 package br.ufsc.cultivar.service;
4346
4347 import br.ufsc.cultivar.dto.PaginateList;
4348 import br.ufsc.cultivar.exception.ServiceException;
4349 import br.ufsc.cultivar.model.School;
4350 import br.ufsc.cultivar.repository.AddressRepository;
4351 import br.ufsc.cultivar.repository.SchoolRepository;
4352 import br.ufsc.cultivar.utils.ValidateUtils;
4353 import lombok.AccessLevel;
4354 import lombok.AllArgsConstructorConstructor;
4355 import lombok.experimental.FieldDefaults;
4356 import lombok.val;
4357 import org.springframework.beans.factory.annotation.Autowired;
4358 import org.springframework.stereotype.Service;
4359
4360 import java.util.ArrayList;
4361 import java.util.List;
4362 import java.util.Map;
4363
4364 @Service
4365 @AllArgsConstructor(onConstructor = @_(@Autowired))
4366 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4367 public class SchoolService {
4368
4369     SchoolRepository schoolRepository;
4370     AddressRepository addressRepository;
4371     UserService userService;
4372
4373     public School create(final School school) throws
4374         ServiceException {
4375         val address = school.getAddress();
4376         if (!ValidateUtils.isValid(address)){
4377             throw new ServiceException(null, null, null);
4378         }
4379         userService.create(school.getResponsible());
4380         return school.withCodSchool(
4381             schoolRepository.create(
```

```
4381             school.withAddress(
4382                 address.withCodAddress(
4383                     addressRepository.create(address)
4384                 )
4385             )
4386         );
4387     }
4388
4389
4390     public PaginateList get(final String filter, final Long
4391         page) throws ServiceException {
4392         return PaginateList.builder()
4393             .count(schoolRepository.count(filter))
4394             .data(new
4395                 ArrayList<>(schoolRepository.get(filter,
4396                     page)))
4397             .build();
4398     }
4399
4400
4401     public School get(final Long codSchool) throws
4402         ServiceException {
4403         val school = schoolRepository.get(codSchool);
4404         return school.withAddress(
4405             addressRepository.get(
4406                 school.getAddress().getCodAddress()
4407             )
4408             ).withResponsible(
4409                 userService.get(
4410                     school.getResponsible().getCpf()
4411             )
4412         );
4413     }
4414
4415
4416     public School delete(final Long codSchool) throws
4417         ServiceException {
4418         val school = get(codSchool);
4419         schoolRepository.delete(codSchool);
4420         return school;
4421     }
4422
4423
4424     public void update(final School school, final Long
4425         codSchool) throws ServiceException {
4426         if (school.getCodSchool().equals(codSchool)){
4427             throw new ServiceException(null, null, null);
4428         }
4429     }
```

```
4420         schoolRepository.update(school);
4421     }
4422 }
4423
4424
4425 src/main/java/br/ufsc/cultivar/service/EventService.java
4426
4427 package br.ufsc.cultivar.service;
4428
4429 import br.ufsc.cultivar.dto.ParticipationDTO;
4430 import br.ufsc.cultivar.dto.UserEventsDTO;
4431 import br.ufsc.cultivar.exception.ServiceException;
4432 import br.ufsc.cultivar.exception.Type;
4433 import br.ufsc.cultivar.exception.UploadException;
4434 import br.ufsc.cultivar.model.Event;
4435 import br.ufsc.cultivar.repository.*;
4436 import br.ufsc.cultivar.utils.FileUtils;
4437 import lombok.AccessLevel;
4438 import lombok.AllArgsConstructor;
4439 import lombok.experimental.FieldDefaults;
4440 import lombok.val;
4441 import org.springframework.beans.factory.annotation.Autowired;
4442 import org.springframework.stereotype.Service;
4443 import org.springframework.web.multipart.MultipartFile;
4444
4445 import java.util.*;
4446 import java.util.stream.Collectors;
4447
4448 @Service
4449 @AllArgsConstructor(onConstructor = @_(@Autowired))
4450 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4451 public class EventService {
4452
4453     FileUtils fileUtils;
4454     EventRepository eventRepository;
4455     AddressRepository addressRepository;
4456     RatingRepository ratingRepository;
4457     UserRepository userRepository;
4458     ParticipationRepository participationRepository;
4459     TrainingRepository trainingRepository;
4460
4461     public String upload(MultipartFile file, Long codEvent)
4462         throws ServiceException{
4463         try {
4464             return fileUtils.save(file, codEvent);
4465         }
```

```

4464     } catch (UploadException e) {
4465         throw new ServiceException(e.getMessage(), e,
4466             Type.FILE);
4467     }
4468 }
4469
4470 public void create(final Event event) throws
4471     ServiceException {
4472     val address = event.getAddress();
4473     val codEvent = eventRepository.create(
4474         event.withAddress(
4475             address.withCodAddress(
4476                 addressRepository.create(
4477                     address
4478                 )
4479             )
4480         );
4481     Optional.ofNullable(event.getParticipants())
4482         .orElseGet(ArrayList::new)
4483         .forEach(
4484             user ->
4485                 participationRepository.create(codEvent,
4486                     user.getPf())
4487             );
4488 }
4489
4490 public Set<Event> get(final List<String> filterVolunteer,
4491     final List<Long> filterSchool,
4492         final Long filterProject) throws
4493             ServiceException {
4494     return new
4495         HashSet<>(eventRepository.get(filterVolunteer,
4496             filterSchool, filterProject));
4497 }
4498
4499 public Event get(final Long codEvent) throws
4500     ServiceException {
4501     val event =
4502         Optional.ofNullable(eventRepository.get(codEvent))
4503             .orElseThrow(() -> new ServiceException(null, null,
4504                 null));
4505     val type = event.getType();
4506     return event.withAddress(
4507         addressRepository.get(

```

```
4498             event.getAddress().getCodAddress()
4499         )
4500     ).withParticipants(
4501         userRepository.getParticipants(codEvent)
4502     ).withRatings(
4503         ratingRepository.get(codEvent)
4504     ).withTrainings(
4505         trainingRepository.getByEvent(codEvent)
4506     ).withType(
4507         type.withTrainings(
4508             trainingRepository.getByTypeEvent(
4509                 type.getType()
4510             )
4511         )
4512     );
4513 }
4514
4515 public Event delete(final Long codEvent) throws
4516     ServiceException {
4517     val event = get(codEvent);
4518     eventRepository.delete(codEvent);
4519     return event;
4520 }
4521
4522 public void update(Event event, final Long codEvent)
4523     throws ServiceException {
4524     if (event.getCodEvent().equals(codEvent)){
4525         throw new ServiceException(null, null, null);
4526     }
4527     eventRepository.update(event);
4528 }
4529
4530 public Set<ParticipationDTO> getVolunteerLocals(final String
4531     codCpf){
4532     return new
4533         HashSet<>(participationRepository.getVolunteerLocals(codCpf));
4534 }
4535
4536 public List<Event> eventsByVolunteer(final String cpf, final
4537     Long type) {
4538     return eventRepository.eventsByVolunteer(cpf, type);
4539 }
4540
4541 public List<Event> eventsBySchool(Long codSchool, Long type)
4542     {
```

```
4537         return eventRepository.eventsBySchool(codSchool, type);
4538     }
4539
4540     public List<UserEventsDTO> getEventsToAlert() {
4541         return userRepository.get((Map<String, Object>) null)
4542             .stream()
4543             .map(
4544                 user ->
4545                     UserEventsDTO.builder()
4546                         .email(user.getEmail())
4547                         .name(user.getName())
4548                         .events(
4549                             eventRepository.getEventsToAlert(user.getCpf())
4550                         ).build()
4551                 ).collect(Collectors.toList());
4552     }
4553
4554     public List<Event> getEventsToEvaluateByVolunteer(final
4555         String cpf) {
4556         return
4557             eventRepository.getEventsToEvaluateByVolunteer(cpf);
4558     }
4559
4560 src/main/java/br/ufsc/cultivar/service/RatingService.java
4561
4562 package br.ufsc.cultivar.service;
4563
4564 import br.ufsc.cultivar.model.Rating;
4565 import br.ufsc.cultivar.repository.RatingRepository;
4566 import lombok.AccessLevel;
4567 import lombok.AllArgsConstructor;
4568 import lombok.experimental.FieldDefaults;
4569 import org.springframework.beans.factory.annotation.Autowired;
4570 import org.springframework.stereotype.Service;
4571
4572 @Service
4573 @AllArgsConstructor(onConstructor = @___(@Autowired))
4574 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4575 public class RatingService {
4576
4577     RatingRepository repository;
4578
4579     public void create(Rating rating, String cpf) {
```

```
4580         repository.create(rating, cpf);
4581     }
4582
4583     public void create(Rating rating, Long codEvent) {
4584         repository.create(rating, codEvent);
4585     }
4586 }
4587
4588
4589 src/main/java/br/ufsc/cultivar/resource/evaluate/PersonalityResource.java
4590
4591 package br.ufsc.cultivar.resource.evaluate;
4592
4593 import br.ufsc.cultivar.exception.ServiceException;
4594 import br.ufsc.cultivar.model.evaluate.Personality;
4595 import br.ufsc.cultivar.service.evaluate.PersonalityService;
4596 import lombok.AccessLevel;
4597 import lombok.AllArgsConstructor;
4598 import lombok.experimental.FieldDefaults;
4599 import org.springframework.beans.factory.annotation.Autowired;
4600 import org.springframework.http.HttpStatus;
4601 import org.springframework.http.MediaType;
4602 import org.springframework.web.bind.annotation.*;
4603
4604 import java.util.List;
4605
4606 @RestController
4607 @RequestMapping(path = "/personality")
4608 @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
4609 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4610 public class PersonalityResource {
4611     PersonalityService service;
4612
4613     @PostMapping(consumes =
4614         MediaType.APPLICATION_JSON_UTF8_VALUE)
4615     @ResponseStatus(code = HttpStatus.CREATED)
4616     public void create(@RequestBody final Personality question)
4617         throws ServiceException {
4618         service.create(question);
4619     }
4620
4621     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
4622     public List<Personality> get() throws ServiceException{
4623         return service.get();
4624     }
4625 }
```

```
4623
4624     @DeleteMapping(path =("/{codQuestion}", produces =
4625         MediaType.APPLICATION_JSON_UTF8_VALUE)
4626     public Personality delete(@PathVariable final Long
4627         codQuestion) throws ServiceException{
4628         return service.delete(codQuestion);
4629     }
4630
4631     src/main/java/br/ufsc/cultivar/resource/evaluate/MentoringResource.java
4632
4633     package br.ufsc.cultivar.resource.evaluate;
4634
4635     import br.ufsc.cultivar.exception.ServiceException;
4636     import br.ufsc.cultivar.model.evaluate.Mentoring;
4637     import br.ufsc.cultivar.model.evaluate.Personality;
4638     import br.ufsc.cultivar.service.evaluate.MentoringService;
4639     import br.ufsc.cultivar.service.evaluate.PersonalityService;
4640     import lombok.AccessLevel;
4641     import lombok.AllArgsConstructor;
4642     import lombok.experimental.FieldDefaults;
4643     import org.springframework.beans.factory.annotation.Autowired;
4644     import org.springframework.http.HttpStatus;
4645     import org.springframework.http.MediaType;
4646     import org.springframework.web.bind.annotation.*;
4647
4648     import java.util.List;
4649
4650     @RestController
4651     @RequestMapping(path = "/mentoring")
4652     @AllArgsConstructor(onConstructor = @___(@Autowired))
4653     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4654     public class MentoringResource {
4655         MentoringService service;
4656
4657         @PostMapping(consumes =
4658             MediaType.APPLICATION_JSON_UTF8_VALUE)
4659         @ResponseStatus(code = HttpStatus.CREATED)
4660         public void create(@RequestBody final Mentoring question)
4661             throws ServiceException {
4662                 service.create(question);
4663             }
4664
4665         @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
```

```
4664     public List<Mentoring> get() throws ServiceException{
4665         return service.get();
4666     }
4667
4668     @DeleteMapping(path =("/{codQuestion}", produces =
4669         MediaType.APPLICATION_JSON_UTF8_VALUE)
4670     public Mentoring delete(@PathVariable final Long
4671         codQuestion) throws ServiceException{
4672         return service.delete(codQuestion);
4673     }
4674
4675     src/main/java/br/ufsc/cultivar/resource/evaluate/SkillResource.java
4676
4677     package br.ufsc.cultivar.resource.evaluate;
4678
4679     import br.ufsc.cultivar.exception.ServiceException;
4680     import br.ufsc.cultivar.model.evaluate.Skill;
4681     import br.ufsc.cultivar.service.evaluate.SkillService;
4682     import lombok.AccessLevel;
4683     import lombok.AllArgsConstructor;
4684     import lombok.experimental.FieldDefaults;
4685     import org.springframework.beans.factory.annotation.Autowired;
4686     import org.springframework.http.HttpStatus;
4687     import org.springframework.http.MediaType;
4688     import org.springframework.web.bind.annotation.*;
4689
4690     import java.util.List;
4691
4692     @RestController
4693     @RequestMapping(path = "/skill")
4694     @AllArgsConstructor(onConstructor = @___(@Autowired))
4695     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4696     public class SkillResource {
4697         SkillService service;
4698
4699         @PostMapping(consumes =
4700             MediaType.APPLICATION_JSON_UTF8_VALUE)
4701         @ResponseStatus(code = HttpStatus.CREATED)
4702         public void create(@RequestBody final Skill skill) throws
4703             ServiceException {
4704             service.create(skill);
4705         }
4706
4707         @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
```

```
4705 public List<Skill> get() throws ServiceException{
4706     return service.get();
4707 }
4708
4709 @DeleteMapping(path = "/{codSkill}", produces =
4710     MediaType.APPLICATION_JSON_UTF8_VALUE)
4711 public Skill delete(@PathVariable final Long codSkill)
4712     throws ServiceException{
4713     return service.delete(codSkill);
4714 }
4715
4716 src/main/java/br/ufsc/cultivar/resource/evaluate/EvaluateProjectResource.j
4717
4718 package br.ufsc.cultivar.resource.evaluate;
4719
4720 import br.ufsc.cultivar.dto.EvaluateDTO;
4721 import br.ufsc.cultivar.exception.ServiceException;
4722 import br.ufsc.cultivar.service.evaluate.EvaluateService;
4723 import lombok.AccessLevel;
4724 import lombok.AllArgsConstructor;
4725 import lombok.experimental.FieldDefaults;
4726 import org.springframework.beans.factory.annotation.Autowired;
4727 import org.springframework.http.HttpStatus;
4728 import org.springframework.http.MediaType;
4729 import org.springframework.web.bind.annotation.*;
4730
4731 import javax.validation.Valid;
4732 import java.util.List;
4733
4734 @RestController
4735 @RequestMapping(path = "/evaluate/project")
4736 @AllArgsConstructor(onConstructor = @___(@Autowired))
4737 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4738 public class EvaluateProjectResource {
4739
4740     EvaluateService evaluateService;
4741
4742     @PostMapping(consumes =
4743         MediaType.APPLICATION_JSON_UTF8_VALUE)
4744     @ResponseStatus(HttpStatus.CREATED)
4745     public void create(@RequestBody @Valid final EvaluateDTO
4746         dto) throws ServiceException {
4747         evaluateService.create(dto);
```

```
4746     }
4747
4748     @GetMapping(path = "/{codProject}", produces =
4749         MediaType.APPLICATION_JSON_UTF8_VALUE)
4750     public List<EvaluateDTO> get(@PathVariable final Long
4751         codProject){
4752         return evaluateService.get(codProject);
4753     }
4754
4755     src/main/java/br/ufsc/cultivar/resource/evaluate/TechnologyResource.java
4756
4757     package br.ufsc.cultivar.resource.evaluate;
4758
4759     import br.ufsc.cultivar.exception.ServiceException;
4760     import br.ufsc.cultivar.model.evaluate.Techology;
4761     import br.ufsc.cultivar.service.evaluate.TechologyService;
4762     import lombok.AccessLevel;
4763     import lombok.AllArgsConstructor;
4764     import lombok.experimental.FieldDefaults;
4765     import org.springframework.beans.factory.annotation.Autowired;
4766     import org.springframework.http.HttpStatus;
4767     import org.springframework.http.MediaType;
4768     import org.springframework.web.bind.annotation.*;
4769
4770     import java.util.List;
4771
4772     @RestController
4773     @RequestMapping(path = "/technology")
4774     @AllArgsConstructor(onConstructor = @___(@Autowired))
4775     @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4776     public class TechnologyResource {
4777         TechologyService service;
4778
4779         @PostMapping(consumes =
4780             MediaType.APPLICATION_JSON_UTF8_VALUE)
4781         @ResponseStatus(code = HttpStatus.CREATED)
4782         public void create(@RequestBody final Techology technology)
4783             throws ServiceException {
4784             service.create(technology);
4785         }
4786
4787         @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
4788         public List<Techology> get() throws ServiceException{
```

```
4787         return service.get();
4788     }
4789
4790     @DeleteMapping(path = "/{codTechnology}", produces =
4791         MediaType.APPLICATION_JSON_UTF8_VALUE)
4792     public Technology delete(@PathVariable final Long
4793         codTechnology) throws ServiceException{
4794         return service.delete(codTechnology);
4795     }
4796
4797 src/main/java/br/ufsc/cultivar/resource/AttachmentResource.java
4798
4799 package br.ufsc.cultivar.resource;
4800
4801 import br.ufsc.cultivar.dto.PaginateList;
4802 import br.ufsc.cultivar.exception.ServiceException;
4803 import br.ufsc.cultivar.model.Attachment;
4804 import br.ufsc.cultivar.model.Status;
4805 import br.ufsc.cultivar.service.AttachmentService;
4806 import lombok.AccessLevel;
4807 import lombok.AllArgsConstructorConstructor;
4808 import lombok.experimental.FieldDefaults;
4809 import org.springframework.beans.factory.annotation.Autowired;
4810 import org.springframework.core.io.Resource;
4811 import org.springframework.http.HttpStatus;
4812 import org.springframework.http.MediaType;
4813 import org.springframework.lang.Nullable;
4814 import org.springframework.web.bind.annotation.*;
4815 import org.springframework.web.multipart.MultipartFile;
4816
4817 import java.util.List;
4818
4819 @RestController
4820 @RequestMapping(path = "/attachment")
4821 @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
4822 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4823 public class AttachmentResource {
4824
4825     AttachmentService service;
4826
4827     @PostMapping(consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
4828     @ResponseStatus(code = HttpStatus.NO_CONTENT)
```

```
4829     public void create(@RequestPart final Attachment attachment,
        @RequestPart(required=false) final MultipartFile file)
        throws ServiceException {
        service.create(attachment, file);
    }
4832
4833     @GetMapping(path = "/{status}", produces =
        MediaType.APPLICATION_JSON_UTF8_VALUE)
4834     public List<Attachment> get(@PathVariable final Status
        status) throws ServiceException{
        return service.get(status);
    }
4837
4838     @GetMapping(path = "/{codAttachment}/download", produces =
        MediaType.APPLICATION_PDF_VALUE)
4839     public Resource get(@PathVariable final Long codAttachment)
        throws ServiceException{
        return service.getAsFile(codAttachment);
    }
4842
4843     @GetMapping(path = "/page/{page}", produces =
        MediaType.APPLICATION_JSON_UTF8_VALUE)
4844     public PaginateList get(@RequestParam(required = false)
        final String filter, @PathVariable final Long page)
        throws ServiceException{
        return service.get(filter, page);
    }
4847
4848     @DeleteMapping(path = "/{codAttachment}", produces =
        MediaType.APPLICATION_JSON_UTF8_VALUE)
4849     public Attachment delete(@PathVariable final Long
        codAttachment) throws ServiceException{
        return service.delete(codAttachment);
    }
4852
4853     @PutMapping(path = "/{codAttachment}", consumes =
        MediaType.APPLICATION_JSON_UTF8_VALUE,
        produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
4854     @ResponseStatus(code = HttpStatus.NO_CONTENT)
4855     public void update(@RequestBody final Attachment attachment,
        @PathVariable final Long codAttachment) throws
        ServiceException{
        service.update(attachment, codAttachment);
    }
4858 }
```

```
4860  
4861  
4862 src/main/java/br/ufsc/cultivar/resource/CompanyResource.java  
4863  
4864 package br.ufsc.cultivar.resource;  
4865  
4866 import br.ufsc.cultivar.dto.PaginateList;  
4867 import br.ufsc.cultivar.exception.ServiceException;  
4868 import br.ufsc.cultivar.model.Company;  
4869 import br.ufsc.cultivar.service.CompanyService;  
4870 import lombok.AccessLevel;  
4871 import lombok.AllArgsConstructorConstructor;  
4872 import lombok.experimental.FieldDefaults;  
4873 import org.springframework.beans.factory.annotation.Autowired;  
4874 import org.springframework.http.HttpStatus;  
4875 import org.springframework.http.MediaType;  
4876 import org.springframework.web.bind.annotation.*;  
4877  
4878 import java.util.List;  
4879 import java.util.Map;  
4880  
4881 @RestController  
4882 @RequestMapping(path = "/company")  
4883 @AllArgsConstructor(onConstructor = @__(@Autowired))  
4884 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)  
4885 public class CompanyResource {  
4886  
4887     CompanyService service;  
4888  
4889     @PostMapping(consumes =  
4890         MediaType.APPLICATION_JSON_UTF8_VALUE)  
4891     @ResponseStatus(code = HttpStatus.NO_CONTENT)  
4892     public void create(@RequestBody final Company company)  
4893         throws ServiceException {  
4894         service.create(company);  
4895     }  
4896  
4897     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)  
4898     public List<Company> get(@RequestParam(required=false) String filter)  
4899         throws ServiceException{  
4900         return service.get(filter, null).getData();  
4901     }  
4902  
4903     @GetMapping(path = "/page/{page}", produces =  
4904         MediaType.APPLICATION_JSON_UTF8_VALUE)
```

```
4901     public PaginateList get(@RequestParam(required=false) final
4902                               String filter,
4903                               @PathVariable final Long page) throws
4904                               ServiceException{
4905         return service.get(filter, page);
4906     }
4907
4908     @GetMapping(path = "/{cnpj}", produces =
4909                   MediaType.APPLICATION_JSON_UTF8_VALUE)
4910     public Company getOne(@PathVariable final String cnpj)
4911         throws ServiceException{
4912         return service.get(cnpj);
4913     }
4914
4915     @DeleteMapping(path = "/{cnpj}", produces =
4916                   MediaType.APPLICATION_JSON_UTF8_VALUE)
4917     public Company delete(@PathVariable final String cnpj)
4918         throws ServiceException{
4919         return service.delete(cnpj);
4920     }
4921
4922     }
4923
4924
4925     src/main/java/br/ufsc/cultivar/resource/QuestionResource.java
4926
4927     package br.ufsc.cultivar.resource;
4928
4929     import br.ufsc.cultivar.exception.ServiceException;
4930     import br.ufsc.cultivar.model.Question;
4931     import br.ufsc.cultivar.model.Role;
4932     import br.ufsc.cultivar.service.QuestionService;
4933     import lombok.AccessLevel;
4934     import lombok.NoArgsConstructor;
4935     import lombok.experimental.FieldDefaults;
4936     import org.springframework.beans.factory.annotation.Autowired;
4937     import org.springframework.http.HttpStatus;
```

```
4938 import org.springframework.http.MediaType;
4939 import org.springframework.web.bind.annotation.*;
4940
4941 import java.util.List;
4942 import java.util.Map;
4943
4944 @RestController
4945 @RequestMapping(path = "/question")
4946 @AllArgsConstructor(onConstructor = @___(@Autowired))
4947 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
4948 public class QuestionResource {
4949
4950     QuestionService service;
4951
4952     @PostMapping(consumes =
4953         MediaType.APPLICATION_JSON_UTF8_VALUE)
4954     @ResponseStatus(code = HttpStatus.NO_CONTENT)
4955     public void create(@RequestBody final Question question)
4956         throws ServiceException {
4957         service.create(question);
4958     }
4959
4960     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
4961     public List<Question> get() throws ServiceException{
4962         return service.get();
4963     }
4964
4965     @GetMapping(path = "/{responds}", produces =
4966         MediaType.APPLICATION_JSON_UTF8_VALUE)
4967     public List<Question> get(@PathVariable final Role responds)
4968         throws ServiceException{
4969         return service.get(responds);
4970     }
4971
4972     @DeleteMapping(path = "/{codQuestion}", produces =
4973         MediaType.APPLICATION_JSON_UTF8_VALUE)
4974     public Question delete(@PathVariable final Long codQuestion)
4975         throws ServiceException{
4976         return service.delete(codQuestion);
4977     }
4978
4979     @PutMapping(path = "/{codQuestion}", consumes =
4980         MediaType.APPLICATION_JSON_UTF8_VALUE,
4981         produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
4982     @ResponseStatus(code = HttpStatus.NO_CONTENT)
```

```
4976     public void update(@RequestBody final Question question,
4977         @PathVariable final Long codQuestion) throws
4978             ServiceException{
4979     service.update(question, codQuestion);
4980 }
4981 }
4982 src/main/java/br/ufsc/cultivar/resource/ProjectResource.java
4983
4984 package br.ufsc.cultivar.resource;
4985
4986 import br.ufsc.cultivar.dto.PaginateList;
4987 import br.ufsc.cultivar.exception.ServiceException;
4988 import br.ufsc.cultivar.model.Project;
4989 import br.ufsc.cultivar.service.ProjectService;
4990 import lombok.AccessLevel;
4991 import lombok.AllArgsConstructor;
4992 import lombok.experimental.FieldDefaults;
4993 import org.springframework.beans.factory.annotation.Autowired;
4994 import org.springframework.http.HttpStatus;
4995 import org.springframework.http.MediaType;
4996 import org.springframework.web.bind.annotation.*;
4997
4998 import javax.validation.Valid;
4999 import java.util.List;
5000
5001 @RestController
5002 @RequestMapping(path = "/project")
5003 @AllArgsConstructor(onConstructor = @___(@Autowired))
5004 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5005 public class ProjectResource {
5006
5007     ProjectService projectService;
5008
5009     @PostMapping(consumes =
5010         MediaType.APPLICATION_JSON_UTF8_VALUE)
5011     @ResponseStatus(HttpStatus.CREATED)
5012     public void create(@Valid @RequestBody final Project
5013         project){
5014         projectService.create(project);
5015     }
5016
5017     @GetMapping(path = "/page/{page}", produces =
5018         MediaType.APPLICATION_JSON_UTF8_VALUE)
```

```
5016     public PaginateList get(@RequestParam(required = false)
5017         final String filter, @PathVariable final Long page)
5018         throws ServiceException {
5019             return projectService.get(filter, page);
5020     }
5021
5022     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5023     public List<Project> get(){
5024         return projectService.get(null, null).getData();
5025     }
5026
5027     @GetMapping(path = "/{codProject}", produces =
5028         MediaType.APPLICATION_JSON_UTF8_VALUE)
5029     public Project get(@PathVariable final Long codProject){
5030         return projectService.get(codProject);
5031     }
5032
5033     @DeleteMapping(path = "/{codProject}", produces =
5034         MediaType.APPLICATION_JSON_UTF8_VALUE)
5035     public Project delete(@PathVariable final Long codProject){
5036         return projectService.delete(codProject);
5037     }
5038
5039     src/main/java/br/ufsc/cultivar/resource/EventResource.java
5040
5041     package br.ufsc.cultivar.resource;
5042
5043     import br.ufsc.cultivar.exception.ServiceException;
5044     import br.ufsc.cultivar.model.Event;
5045     import br.ufsc.cultivar.model.Rating;
5046     import br.ufsc.cultivar.service.EventService;
5047     import br.ufsc.cultivar.service.RatingService;
5048     import lombok.AccessLevel;
5049     import lombok.experimentalArgsConstructor;
5050     import org.springframework.beans.factory.annotation.Autowired;
5051     import org.springframework.http.HttpStatus;
5052     import org.springframework.http.MediaType;
5053     import org.springframework.web.bind.annotation.*;
5054     import org.springframework.web.multipart.MultipartFile;
5055
5056     import java.util.List;
5057     import java.util.Map;
```

```
5057 import java.util.Set;
5058
5059 @RestController
5060 @RequestMapping(path = "/event")
5061 @AllArgsConstructor(onConstructor = @_(_(@Autowired)))
5062 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5063 public class EventResource {
5064
5065     EventService eventService;
5066     RatingService ratingService;
5067
5068     @PostMapping(consumes =
5069         MediaType.APPLICATION_JSON_UTF8_VALUE)
5070     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5071     public void create(@RequestBody final Event event) throws
5072         ServiceException {
5073         eventService.create(event);
5074     }
5075
5076     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5077     public Set<Event> get(
5078         @RequestParam(value = "cod_cpf", required = false)
5079             final List<String> filterVolunteer,
5080         @RequestParam(value = "cod_school", required = false)
5081             final List<Long> filterSchool,
5082         @RequestParam(value = "cod_project", required =
5083             false) final Long filterProject) throws
5084         ServiceException{
5085         return eventService.get(filterVolunteer, filterSchool,
5086             filterProject);
5087     }
5088
5089     @GetMapping(path = "/{codEvent}", produces =
5090         MediaType.APPLICATION_JSON_UTF8_VALUE)
5091     public Event get(@PathVariable final Long codEvent) throws
5092         ServiceException{
5093         return eventService.get(codEvent);
5094     }
5095
5096     @DeleteMapping(path = "/{codEvent}", produces =
5097         MediaType.APPLICATION_JSON_UTF8_VALUE)
5098     public Event delete(@PathVariable final Long codEvent)
5099         throws ServiceException{
5100         return eventService.delete(codEvent);
5101     }
```

```
5091
5092     @PutMapping(path = "/{codEvent}", consumes =
5093         MediaType.APPLICATION_JSON_UTF8_VALUE,
5094         produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5095     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5096     public void update(@RequestBody final Event event,
5097         @PathVariable final Long codEvent) throws
5098             ServiceException{
5099                 eventService.update(event, codEvent);
5100             }
5101
5102     @PostMapping(path="/{codEvent}/upload", consumes =
5103         MediaType.APPLICATION_PDF_VALUE,
5104         produces = MediaType.TEXT_PLAIN_VALUE)
5105     public String upload(@RequestPart final MultipartFile file,
5106         @PathVariable final Long codEvent) throws
5107             ServiceException {
5108                 return eventService.upload(file, codEvent);
5109             }
5110
5111     @PutMapping(path = "/{codEvent}/evaluate", consumes =
5112         MediaType.APPLICATION_JSON_UTF8_VALUE)
5113     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5114     public void evaluate(@RequestBody final Rating rating,
5115         @PathVariable Long codEvent){
5116             ratingService.create(rating, codEvent);
5117         }
5118
5119     src/main/java/br/ufsc/cultivar/resource/VolunteerResource.java
5120
5121     package br.ufsc.cultivar.resource;
5122
5123     import br.ufsc.cultivar.dto.PaginateList;
5124     import br.ufsc.cultivar.dto.ParticipationDTO;
5125     import br.ufsc.cultivar.exception.ServiceException;
5126     import br.ufsc.cultivar.model.*;
5127     import br.ufsc.cultivar.service.DispatchService;
5128     import br.ufsc.cultivar.service.EventService;
5129     import br.ufsc.cultivar.service.RatingService;
5130     import br.ufsc.cultivar.service.VolunteerService;
5131     import lombok.AccessLevel;
5132     import lombok.AllArgsConstructor;
5133     import lombok.experimental.FieldDefaults;
```

```
5128 import org.springframework.beans.factory.annotation.Autowired;
5129 import org.springframework.http.HttpStatus;
5130 import org.springframework.http.MediaType;
5131 import org.springframework.web.bind.annotation.*;
5132 import org.springframework.web.multipart.MultipartFile;
5133
5134 import java.util.List;
5135 import java.util.Set;
5136
5137 @RestController
5138 @RequestMapping(path = "/volunteer")
5139 @AllArgsConstructor(onConstructor = @__(@Autowired))
5140 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5141 public class VolunteerResource {
5142
5143     VolunteerService volunteerService;
5144     RatingService ratingService;
5145     DispatchService dispatchService;
5146     EventService eventService;
5147
5148     @PostMapping(consumes =
5149                 MediaType.APPLICATION_JSON_UTF8_VALUE)
5150     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5151     public void create(@RequestBody final Volunteer volunteer)
5152         throws ServiceException {
5153         volunteerService.create(volunteer);
5154     }
5155
5156     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5157     public List get(
5158         @RequestParam(value = "cod_cnpj", required = false)
5159         final List<String> filterCompany,
5160         @RequestParam(value = "cod_school", required = false)
5161         final List<Long> filterSchool,
5162         @RequestParam(required = false) final String filter)
5163         throws ServiceException{
5164         return volunteerService.get(filterCompany, filterSchool,
5165             filter, null).getData();
5166     }
5167
5168     @GetMapping(path = "/page/{page}", produces =
5169                 MediaType.APPLICATION_JSON_UTF8_VALUE)
5170     public PaginateList get(
5171         @RequestParam(value = "cod_cnpj", required = false)
5172         final List<String> filterCompany,
```

```
5165     @RequestParam(value = "cod_school", required = false)
5166         final List<Long> filterSchool,
5167     @RequestParam(required = false) final String filter,
5168     @PathVariable final Integer page) throws
5169         ServiceException{
5170     return volunteerService.get(filterCompany, filterSchool,
5171         filter, page);
5172 }
5173
5174 @GetMapping(path = "/{cpf}", produces =
5175     MediaType.APPLICATION_JSON_UTF8_VALUE)
5176 public Volunteer get(@PathVariable final String cpf) throws
5177     ServiceException{
5178     return volunteerService.get(cpf);
5179 }
5180
5181 @DeleteMapping(path = "/{cpf}", produces =
5182     MediaType.APPLICATION_JSON_UTF8_VALUE)
5183 public Volunteer delete(@PathVariable final String cpf)
5184     throws ServiceException{
5185     return volunteerService.delete(cpf);
5186 }
5187
5188 @PutMapping(path = "/{cpf}", consumes =
5189     MediaType.APPLICATION_JSON_UTF8_VALUE,
5190     produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5191 @ResponseStatus(code = HttpStatus.NO_CONTENT)
5192 public void update(@RequestBody final Volunteer volunteer,
5193     @PathVariable final String cpf) throws ServiceException{
5194     volunteerService.update(volunteer, cpf);
5195 }
5196
5197 @GetMapping(path = "{/cpf}/event", "{/cpf}/event/{type}")
5198 public List<Event> getEvents(@PathVariable final String cpf,
5199     @PathVariable(required = false) final Long type){
5200     return eventService.eventsByVolunteer(cpf, type);
5201 }
5202
5203 @GetMapping(path = "/{cpf}/attachment/{codAttachment}")
5204 public Dispatch getDispatch(@PathVariable final String cpf,
5205     @PathVariable Long codAttachment){
5206     return dispatchService.get(cpf, codAttachment);
5207 }
5208
```

```

5199     @PostMapping(path = "/{cpf}/upload", consumes =
5200         MediaType.MULTIPART_FORM_DATA_VALUE,
5201         produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5202     public void upload(@RequestPart final Attachment attachment,
5203                         @RequestPart final MultipartFile file,
5204                         @PathVariable final String cpf) throws
5205                             ServiceException {
5206         dispatchService.save(attachment, file, cpf);
5207     }
5208
5209     @PutMapping(path = "/{cpf}/evaluate", consumes =
5210         MediaType.APPLICATION_JSON_UTF8_VALUE)
5211     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5212     public void evaluate(@RequestBody final Rating rating,
5213                           @PathVariable final String cpf) throws ServiceException {
5214         ratingService.create(rating, cpf);
5215     }
5216
5217     @GetMapping(path = "/{cpf}/evaluate", consumes =
5218         MediaType.APPLICATION_JSON_UTF8_VALUE)
5219     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5220     public List<Event> evaluate(@PathVariable final String cpf)
5221         throws ServiceException {
5222         return eventService.getEventsToEvaluateByVolunteer(cpf);
5223     }
5224
5225     @GetMapping(path = "/{cpf}/participations", produces =
5226         MediaType.APPLICATION_JSON_UTF8_VALUE)
5227     public Set<ParticipationDTO>
5228         getVolunteerLocals(@PathVariable final String cpf){
5229         return eventService.getVolunteerLocals(cpf);
5230     }
5231
5232     }
5233
5234     src/main/java/br/ufsc/cultivar/resource/UserResource.java
5235
5236     package br.ufsc.cultivar.resource;
5237
5238     import br.ufsc.cultivar.exception.ServiceException;
5239     import br.ufsc.cultivar.model.User;
5240     import br.ufsc.cultivar.service.CompanyService;
5241     import br.ufsc.cultivar.service.UserService;
5242     import lombok.AccessLevel;
5243     import lombok.AllArgsConstructor;
5244     import lombok.NoArgsConstructor;

```

```
5235 import lombok.experimental.FieldDefaults;
5236 import lombok.val;
5237 import org.springframework.beans.factory.annotation.Autowired;
5238 import org.springframework.http.HttpStatus;
5239 import org.springframework.http.MediaType;
5240 import org.springframework.web.bind.annotation.*;
5241
5242 import java.util.List;
5243 import java.util.Map;
5244
5245 @RestController
5246 @RequestMapping(path = "/user")
5247 @AllArgsConstructor(onConstructor = @__(@Autowired))
5248 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5249 public class UserResource {
5250
5251     UserService service;
5252
5253     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5254     public List<User> get(@RequestParam final Map<String,
5255         Object> filter) throws ServiceException{
5256         return service.get(filter);
5257     }
5258
5259     @GetMapping(path = "/{cpf}", produces =
5260         MediaType.APPLICATION_JSON_UTF8_VALUE)
5261     public User get(@PathVariable final String cpf) throws
5262         ServiceException{
5263         return service.get(cpf);
5264     }
5265
5266     @DeleteMapping(path = "/{cpf}", produces =
5267         MediaType.APPLICATION_JSON_UTF8_VALUE)
5268     public User delete(@PathVariable final String cpf) throws
5269         ServiceException{
5270         return service.delete(cpf);
5271     }
5272
5273     @PutMapping(path = "/{cpf}", consumes =
5274         MediaType.APPLICATION_JSON_UTF8_VALUE,
5275         produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5276     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5277     public void update(@RequestBody final User user,
5278         @PathVariable final String cpf) throws ServiceException{
5279         service.update(user, cpf);
```

```
5273     }
5274 }
5275
5276
5277 src/main/java/br/ufsc/cultivar/resource/TypeEventResource.java
5278
5279 package br.ufsc.cultivar.resource;
5280
5281 import br.ufsc.cultivar.dto.PaginateList;
5282 import br.ufsc.cultivar.exception.ServiceException;
5283 import br.ufsc.cultivar.model.Training;
5284 import br.ufsc.cultivar.model.TypeEvent;
5285 import br.ufsc.cultivar.service.TypeEventService;
5286 import lombok.AccessLevel;
5287 import lombok.AllArgsConstructor;
5288 import lombok.experimental.FieldDefaults;
5289 import org.springframework.beans.factory.annotation.Autowired;
5290 import org.springframework.http.HttpStatus;
5291 import org.springframework.http.MediaType;
5292 import org.springframework.web.bind.annotation.*;
5293
5294 import java.util.List;
5295
5296 @RestController
5297 @RequestMapping(path = "/typeEvent")
5298 @AllArgsConstructor(onConstructor = @_(@Autowired))
5299 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5300 public class TypeEventResource {
5301
5302     TypeEventService typeEventService;
5303
5304     @PostMapping(consumes =
5305         MediaType.APPLICATION_JSON_UTF8_VALUE)
5306     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5307     public void create(@RequestBody final TypeEvent typeEvent)
5308         throws ServiceException {
5309         typeEventService.create(typeEvent);
5310     }
5311
5312     @GetMapping(path = "/page/{page}", produces =
5313         MediaType.APPLICATION_JSON_UTF8_VALUE)
5314     public PaginateList get(@RequestParam(required = false)
5315         final String filter, @PathVariable final Long page)
5316         throws ServiceException {
5317         return typeEventService.get(filter, page);
5318 }
```

```
5313     }
5314
5315     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5316     public List<TypeEvent> get() throws ServiceException{
5317         return typeEventService.get(null, null).getData();
5318     }
5319
5320     @GetMapping(path = "/{tpEvent}", produces =
5321         MediaType.APPLICATION_JSON_UTF8_VALUE)
5322     public TypeEvent get(@PathVariable final Long tpEvent)
5323         throws ServiceException{
5324         return typeEventService.get(tpEvent);
5325     }
5326
5327     @DeleteMapping(path = "/{tpEvent}", produces =
5328         MediaType.APPLICATION_JSON_UTF8_VALUE)
5329     public TypeEvent delete(@PathVariable final Long tpEvent)
5330         throws ServiceException{
5331         return typeEventService.delete(tpEvent);
5332     }
5333
5334 }
5335
5336
5337 src/main/java/br/ufsc/cultivar/resource/SchoolResource.java
5338
5339 package br.ufsc.cultivar.resource;
5340
5341 import br.ufsc.cultivar.dto.PaginateList;
5342 import br.ufsc.cultivar.exception.ServiceException;
5343 import br.ufsc.cultivar.model.Event;
5344 import br.ufsc.cultivar.model.School;
5345 import br.ufsc.cultivar.model.TypeEvent;
5346 import br.ufsc.cultivar.service.EventService;
5347 import br.ufsc.cultivar.service.SchoolService;
5348 import lombok.AccessLevel;
5349 import lombok.AllArgsConstructor;
5350 import lombok.experimental.FieldDefaults;
5351 import org.springframework.beans.factory.annotation.Autowired;
```

```
5352 import org.springframework.http.HttpStatus;
5353 import org.springframework.http.MediaType;
5354 import org.springframework.web.bind.annotation.*;
5355
5356 import java.util.List;
5357 import java.util.Map;
5358
5359 @RestController
5360 @RequestMapping(path = "/school")
5361 @AllArgsConstructor(onConstructor = @__(@Autowired))
5362 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5363 public class SchoolResource {
5364     SchoolService service;
5365     EventService eventService;
5366
5367     @PostMapping(consumes =
5368         MediaType.APPLICATION_JSON_UTF8_VALUE)
5369     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5370     public void create(@RequestBody final School school) throws
5371         ServiceException {
5372         service.create(school);
5373     }
5374
5375     @GetMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5376     public List get(@RequestParam(required=false) String filter)
5377         throws ServiceException{
5378         return service.get(filter, null).getData();
5379     }
5380
5381     @GetMapping(path = "/page/{page}", produces =
5382         MediaType.APPLICATION_JSON_UTF8_VALUE)
5383     public PaginateList get(@RequestParam(required=false) String
5384         filter, @PathVariable final Long page) throws
5385         ServiceException{
5386         return service.get(filter, page);
5387     }
5388
5389     @GetMapping(path =("/{codSchool}", produces =
5390         MediaType.APPLICATION_JSON_UTF8_VALUE)
5391     public School get(@PathVariable final Long codSchool) throws
5392         ServiceException{
5393         return service.get(codSchool);
5394     }
5395
```

```
5388     @GetMapping(path = {"/{codSchool}/event",
5389                   "/{codSchool}/event/{type}"}, produces =
5390                   MediaType.APPLICATION_JSON_UTF8_VALUE)
5391     public List<Event> getEvents(@PathVariable final Long
5392                                   codSchool, @PathVariable(required = false) final Long
5393                                   type){
5394         return eventService.eventsBySchool(codSchool, type);
5395     }
5396
5397     @DeleteMapping(path = "/{codSchool}", produces =
5398                   MediaType.APPLICATION_JSON_UTF8_VALUE)
5399     public School delete(@PathVariable final Long codSchool)
5400         throws ServiceException{
5401         return service.delete(codSchool);
5402     }
5403
5404     @PutMapping(path = "/{codSchool}", consumes =
5405                   MediaType.APPLICATION_JSON_UTF8_VALUE,
5406                   produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
5407     @ResponseStatus(code = HttpStatus.NO_CONTENT)
5408     public void update(@RequestBody final School school,
5409                         @PathVariable final Long codSchool) throws
5410                         ServiceException{
5411         service.update(school, codSchool);
5412     }
5413
5414
5415     src/main/java/br/ufsc/cultivar/exception/ServiceException.java
5416
5417     package br.ufsc.cultivar.exception;
5418
5419     import lombok.Getter;
5420
5421     public class ServiceException extends Exception {
5422
5423         @Getter
5424         private final Type type;
5425
5426         public ServiceException(String message, Throwable cause,
5427                               Type type) {
5428             super(message, cause);
5429             this.type = type;
5430         }
5431
5432
5433     }
```

```
5423     public ServiceException(String message, Throwable cause) {
5424         super(message, cause);
5425         this.type = Type.DEFAULT;
5426     }
5427 }
5428
5429
5430 src/main/java/br/ufsc/cultivar/exception/Type.java
5431
5432 package br.ufsc.cultivar.exception;
5433
5434 public enum Type {
5435     NOT_FOUND,
5436     INVALID,
5437     FILE,
5438     DEFAULT
5439 }
5440
5441
5442 src/main/java/br/ufsc/cultivar/exception/UploadException.java
5443
5444 package br.ufsc.cultivar.exception;
5445
5446 public class UploadException extends Exception {
5447     public UploadException(String message, Throwable cause) {
5448         super(message, cause);
5449     }
5450 }
5451
5452
5453 src/main/java/br/ufsc/cultivar/email/EmailClient.java
5454
5455 package br.ufsc.cultivar.email;
5456
5457 import br.ufsc.cultivar.config.EmailConfig;
5458 import lombok.AccessLevel;
5459 import lombok.NoArgsConstructor;
5460 import lombok.experimental.FieldDefaults;
5461 import org.springframework.beans.factory.annotation.Autowired;
5462 import org.springframework.mail.javamail.JavaMailSender;
5463 import org.springframework.mail.javamail.MimeMessageHelper;
5464 import org.springframework.mail.javamail.MimeMessagePreparator;
5465 import org.springframework.scheduling.annotation.Async;
5466 import org.springframework.stereotype.Component;
5467
```

```
5468 import java.util.Optional;
5469 import java.util.concurrent.CompletableFuture;
5470
5471 @Component
5472 @AllArgsConstructor(onConstructor = @_(@Autowired))
5473 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
5474 public class EmailClient {
5475     EmailConfig emailConfig;
5476     JavaMailSender mailSender;
5477
5478     public void sendEmailSync(String to, String subject, String
5479         message){
5480         MimeMessagePreparator messagePreparator = mimeMessage ->
5481             {
5482                 MimeMessageHelper messageHelper = new
5483                     MimeMessageHelper(mimeMessage);
5484                 messageHelper.setFrom(emailConfig.getFrom());
5485                 messageHelper.setTo(to);
5486                 messageHelper.setSubject(subject);
5487                 messageHelper.setText(message, true);
5488             };
5489             mailSender.send(messagePreparator);
5490
5491     }
5492
5493     @Async
5494     public CompletableFuture<Void> sendEmailAsync(String to,
5495         String subject, String message){
5496         sendEmailSync(to, subject, message);
5497         return CompletableFuture.completedFuture(null);
5498     }
5499 }
5500
5501
5502 src/main/java/br/ufsc/cultivar/Application.java
5503
5504 package br.ufsc.cultivar;
5505
5506 import org.springframework.boot.SpringApplication;
5507 import
5508     org.springframework.boot.autoconfigure.SpringBootApplication;
5509 import
5510     org.springframework.scheduling.annotation.EnableScheduling;
5511 import
5512     org.springframework.web.bind.annotation.RequestMapping;
```

```
5507 @SpringBootApplication  
5508 @EnableScheduling  
5509 public class Application {  
5510     public static void main(String... args) {  
5511         SpringApplication.run(Application.class, args);  
5512     }  
5513 }
```

D.2 FRONTEND

```

1
2
3 webapp/src/utils/formatter.js
4
5 export default {
6     cnpj: s => s.replace(/^(\\d{2})(\\d{3})(\\d{3})(\\d{4})(\\d{2})/, 
7         "$1.$2.$3/$4-$5"),
8     cpf: s => s.replace(/^(\\d{3})(\\d{3})(\\d{3})(\\d{2})/, 
9         "$1.$2.$3-$4"),
10    phone: s => s.length === 10 ?
11        s.replace(/^(\\d{2})(\\d{4})(\\d{4})/, "($1)$2-$3") :
12        s.replace(/^(\\d{2})(\\d{5})(\\d{4})/, "($1)$2-$3"),
13    }
14
15 webapp/src/utils/http.js
16
17
18 import axios from 'axios';
19 import { get } from './storage';
20
21 axios.defaults.baseURL = process.env.REACT_APP_ENDPOINT;
22
23
24 export const setHeaders = (headers) =>
25     axios.defaults.headers.common = {
26         'X-Requested-With': 'XMLHttpRequest',
27         'Authorization': get('token'),
28         ...headers
29     }
30
31
32 const empty = () => {};
33
34 const setCallback = (request, success, failed) =>
35     request.then(success ? success : empty).catch(failed ?
36         failed : empty);
37
38
39 export const getRequest = (url, success, failed) =>
40     setCallback(axios.get(url), success, failed);
41
42 export const postRequest = (url, data, success, failed) =>
43     setCallback(axios.post(url, data), success, failed);
44
45 export const putRequest = (url, data, success, failed) =>
46     setCallback(axios.put(url, data), success, failed);

```

```
31  export const deleteRequest = (url, success, failed) =>
      setCallback(axios.delete(url), success, failed);
32
33  setHeaders();
34
35  webapp/src/utils/storage.js
36
37  export const saveObject = (key, obj) =>
      localStorage.setItem(key, JSON.stringify(obj));
38
39  export const getAsObject = key =>
      JSON.parse(localStorage.getItem(key));
40
41  export const save = (key, data) => localStorage.setItem(key,
      data);
42
43  export const get = key => localStorage.getItem(key);
44
45  webapp/src/components/pagination/index.js
46
47  import Pagination from './Pagination.jsx';
48  export default Pagination;
49
50  webapp/src/components/pagination/Pagination.jsx
51
52  import React from 'react';
53  import { Row, Col, Pagination, PaginationItem, PaginationLink }
      from 'reactstrap';
54
55  export default class Example extends React.Component {
56    constructor(){
57      super();
58      this.state = {
59        page: 0
60      };
61      this.increment = this.increment.bind(this);
62      this.decrement = this.decrement.bind(this);
63    }
64
65    increment(){
66      const page = this.state.page + 1;
67      this.props.onChangePage(page);
68      this.setState({page});
69    }
70
```

```
71     decrement() {
72         const page = this.state.page - 1;
73         this.props.onChangePage(page);
74         this.setState({ page });
75     }
76
77     render() {
78         const { pages, count } = this.props;
79         const { page } = this.state;
80         return (
81             <Row>
82                 <Col>
83                     <Pagination aria-label="Page navigation
84                         example">
85                         <PaginationItem disabled={page === 0}>
86                             <PaginationLink previous href="#" onClick={this.decrement}/>
87                         </PaginationItem>
88                         <PaginationItem disabled={page ===
89                             parseInt(pages)}>
90                             <PaginationLink next href="#" onClick={this.increment}/>
91                         </PaginationItem>
92                     </Pagination>
93                 </Col>
94                 <Col>
95                     <strong className="float-right"
96                         style={{fontSize: '.8rem'}}>
97                         Página {page + 1} de {parseInt(pages) + 1}
98                         <br/>
99                         {count} Registros
100                     </strong>
101                 </Col>
102             </Row>
103         );
104     }
105 }
106
107 import React from 'react';
108 import { Container, Col, Row } from 'reactstrap';
109 import logo from './logo.svg';
110 import './footer.css';
```

```

111  export default () => (
112      <footer className="footer">
113          <Container>
114              <Row>
115                  <Col>
116                      <span className="text-muted">Powered by <img
117                          className="react-logo" src={logo}
118                          alt="React"/></span>
119                  </Col>
120              </Row>
121          </Container>
122      </footer>
123  );
124
125  $footer-height: 60px
126
127  .footer
128      position: fixed
129      bottom: 0
130      width: 100%
131      height: $footer-height
132      line-height: 60px /* Vertically center the text there */
133      background-color: #f5f5f5
134  .react-logo
135      animation: react-logo-spin infinite 20s linear
136      height: $footer-height - 15px
137
138  @keyframes react-logo-spin
139      from
140          transform: rotate(0deg)
141      to
142          transform: rotate(360deg)
143
144
145  webapp/src/components/footer/logo.svg
146
147  <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 841.9
148      595.3">
149      <g fill="#61DAFB">
150          <path d="M666.3 296.5c0-32.5-40.7-63.3-103.1-82.4
151              14.4-63.6
152              8-114.2-20.2-130.4-6.5-3.8-14.1-5.6-22.4-5.6v22.3c4.6
153              0 8.3.9 11.4 2.6 13.6 7.8 19.5 37.5 14.9 75.7-1.1

```

9.4-2.9 19.3-5.1
 29.4-19.6-4.8-41-8.5-63.5-10.9-13.5-18.5-27.5-35.3-41.6-50
 32.6-30.3 63.2-46.9 84-46.9V78c-27.5 0-63.5
 19.6-99.9
 53.6-36.4-33.8-72.4-53.2-99.9-53.2v22.3c20.7 0 51.4
 16.5 84 46.6-14 14.7-28 31.4-41.3 49.9-22.6 2.4-44
 6.1-63.6 11-2.3-10-4-19.7-5.2-29-4.7-38.2 1.1-67.9
 14.6-75.8 3-1.8 6.9-2.6 11.5-2.6V78.5c-8.4 0-16
 1.8-22.6 5.6-28.1 16.2-34.4 66.7-19.9 130.1-62.2
 19.2-102.7 49.9-102.7 82.3 0 32.5 40.7 63.3 103.1
 82.4-14.4 63.6-8 114.2 20.2 130.4 6.5 3.8 14.1 5.6
 22.5 5.6 27.5 0 63.5-19.6 99.9-53.6 36.4 33.8 72.4
 53.2 99.9 53.2 8.4 0 16-1.8 22.6-5.6 28.1-16.2
 34.4-66.7 19.9-130.1 62-19.1 102.5-49.9
 102.5-82.3zm-130.2-66.7c-3.7 12.9-8.3 26.2-13.5
 39.5-4.1-8-8.4-16-13.1-24-4.6-8-9.5-15.8-14.4-23.4
 14.2 2.1 27.9 4.7 41 7.9zm-45.8 106.5c-7.8 13.5-15.8
 26.3-24.1 38.2-14.9 1.3-30 2-45.2 2-15.1
 0-30.2-7-45-1.9-8.3-11.9-16.4-24.6-24.2-38-7.6-13.1-14.5-26.4-
 6.2-13.4 13.2-26.8 20.7-39.9 7.8-13.5 15.8-26.3
 24.1-38.2 14.9-1.3 30-2 45.2-2 15.1 0 30.2.7 45 1.9
 8.3 11.9 16.4 24.6 24.2 38 7.6 13.1 14.5 26.4 20.8
 39.8-6.3 13.4-13.2 26.8-20.7 39.9zm32.3-13c5.4 13.4
 10 26.8 13.8 39.8-13.1 3.2-26.9 5.9-41.2 8 4.9-7.7
 9.8-15.6 14.4-23.7 4.6-8 8.9-16.1 13-24.1zM421.2
 430c-9.3-9.6-18.6-20.3-27.8-32 9 .4 18.2.7 27.5.7
 9.4 0 18.7-.2 27.8-.7-9 11.7-18.3 22.4-27.5
 32zm-74.4-58.9c-14.2-2.1-27.9-4.7-41-7.9 3.7-12.9
 8.3-26.2 13.5-39.5 4.1 8 8.4 16 13.1 24 4.7 8 9.5
 15.8 14.4 23.4zM420.7 163c9.3 9.6 18.6 20.3 27.8
 32-9-.4-18.2-.7-27.5-.7-9.4 0-18.7.2-27.8.7 9-11.7
 18.3-22.4 27.5-32zm-74 58.9c-4.9 7.7-9.8 15.6-14.4
 23.7-4.6 8-8.9 16-13 24-5.4-13.4-10-26.8-13.8-39.8
 13.1-3.1 26.9-5.8 41.2-7.9zm-90.5
 125.2c-35.4-15.1-58.3-34.9-58.3-50.6 0-15.7
 22.9-35.6 58.3-50.6 8.6-3.7 18-7 27.7-10.1 5.7 19.6
 13.2 40 22.5 60.9-9.2 20.8-16.6 41.1-22.2
 60.6-9.9-3.1-19.3-6.5-28-10.2zM310
 490c-13.6-7.8-19.5-37.5-14.9-75.7 1.1-9.4 2.9-19.3
 5.1-29.4 19.6 4.8 41 8.5 63.5 10.9 13.5 18.5 27.5
 35.3 41.6 50-32.6 30.3-63.2 46.9-84
 46.9-4.5-.1-8.3-1-11.3-2.7zm237.2-76.2c4.7 38.2-1.1
 67.9-14.6 75.8-3 1.8-6.9 2.6-11.5 2.6-20.7
 0-51.4-16.5-84-46.6 14-14.7 28-31.4 41.3-49.9
 22.6-2.4 44-6.1 63.6-11 2.3 10.1 4.1 19.8 5.2

```

29.1zm38.5-66.7c-8.6 3.7-18 7-27.7
10.1-5.7-19.6-13.2-40-22.5-60.9 9.2-20.8 16.6-41.1
22.2-60.6 9.9 3.1 19.3 6.5 28.1 10.2 35.4 15.1 58.3
34.9 58.3 50.6-.1 15.7-23 35.6-58.4 50.6zM320.8
78.4z"/>
150     <circle cx="420.9" cy="296.5" r="45.7"/>
151     <path d="M520.5 78.1z"/>
152   </g>
153 </svg>
154
155
156 webapp/src/components/footer/footer.css
157
158 .footer {
159   position: fixed;
160   bottom: 0;
161   width: 100%;
162   height: 60px;
163   line-height: 60px;
164   background-color: #f5f5f5; }
165 .footer .react-logo {
166   animation: react-logo-spin infinite 20s linear;
167   height: 45px; }

168
169 @keyframes react-logo-spin {
170   from {
171     transform: rotate(0deg); }
172   to {
173     transform: rotate(360deg); } }
174
175
176 webapp/src/components/form/maskInput/index.js
177
178 import MaskInput from './Mask'
179 export default MaskInput;
180
181 webapp/src/components/form/maskInput/Mask.jsx
182
183 import React, { Component } from 'react';
184 import Input from '../input';
185 import InputMask from 'react-input-mask';
186 import PropTypes from 'prop-types';
187 import requiredIf from 'react-required-if';
188
189 export default class extends Component {

```

```
190 static propTypes = {
191     id: PropTypes.string.isRequired,
192     mask: PropTypes.string.isRequired,
193     label: PropTypes.string.isRequired,
194     invalidMessage: requiredIf(PropTypes.string, props =>
195         props.required),
196     required: PropTypes.bool,
197     placeholder: PropTypes.string,
198     maskChar: PropTypes.string,
199     onChange: PropTypes.func
200 }
201
202 static defaultProps = {
203     maskChar: '_',
204 }
205
206 componentDidMount() {
207     const { id, value } = this.props;
208     document.getElementById(id).value = value;
209 }
210
211 render() {
212     const { id, label, placeholder, invalidMessage,
213             onChange, required, mask, maskChar, value, ...rest } =
214         this.props
215     return (
216         <InputMask onChange={onChange} mask={mask}>
217             <maskChar={maskChar}>
218                 {
219                     inputProps =>
220                     (
221                         <Input id={id} {...inputProps}>
222                             label={label}
223                             invalidMessage={invalidMessage}
224                             placeholder={placeholder}
225                             required={required} {...rest} />
226                     )
227                 }
228             </InputMask>
229         );
230     }
231 }
232
233 webapp/src/components/form/input/Input.jsx
234
```

```
227 import React, { Component } from 'react';
228 import PropTypes from 'prop-types';
229 import { FormGroup, Input, Label, FormFeedback } from
230   'reactstrap';
231 import Required from '../Required.jsx'
232 import requiredIf from 'react-required-if';
233
234 export default class extends Component{
235   static propTypes = {
236     id: PropTypes.string.isRequired,
237     label: PropTypes.string.isRequired,
238     invalidMessage: requiredIf(PropTypes.string, props =>
239       props.required),
240     type: PropTypes.string,
241     placeholder: PropTypes.string,
242     required: PropTypes.bool,
243     children: requiredIf(PropTypes.node, props => props.type
244       === 'select')
245   }
246
247   static defaultProps = {
248     type: 'text',
249   }
250
251   componentDidMount() {
252     const { id, value } = this.props;
253     document.getElementById(id).value = value;
254   }
255
256   render(){
257     const { id, label, type, placeholder, invalidMessage,
258       required, children, ...rest } = this.props
259     return (
260       <FormGroup>
261         <Label for={id}>{label}{required ? <Required /> :
262           null}</Label>
263         <Input id={id} type={type}
264           placeholder={placeholder} {...rest}>
265           {children}
266         </Input>
267         <FormFeedback>{invalidMessage}</FormFeedback>
268       </FormGroup>
269     );
270   }
271 }
```

```
266
267  webapp/src/components/form/input/index.js
268
269  import Input from './Input'
270  export default Input;
271
272  webapp/src/components/form/datePicker/datePicker.css
273
274  input[type="date"] {
275    background: #fff url(./calendar.png) 97% 50% no-repeat; }
276  input[type="date"]::-webkit-inner-spin-button {
277    display: none; }
278  input[type="date"]::-webkit-calendar-picker-indicator {
279    opacity: 0; }
280
281
282  webapp/src/components/form/datePicker/datePicker.sass
283
284  input[type="date"]
285    background: #fff url(./calendar.png) 97% 50% no-repeat
286    &::-webkit-inner-spin-button
287      display: none;
288    &::-webkit-calendar-picker-indicator
289      opacity: 0;
290
291
292  webapp/src/components/form/datePicker/index.js
293
294  import DatePicker from './DatePicker';
295  export default DatePicker;
296
297  webapp/src/components/form/datePicker/DatePicker.jsx
298
299  import React from 'react';
300  import PropTypes from 'prop-types';
301  import requiredIf from 'react-required-if';
302  import Input from '../input';
303  import './datePicker.css';
304
305  const DatePicker = ({ id, label, placeholder, invalidMessage,
306    required, ...rest }) =>
307    <Input id={id} type="date" label={label}
308      placeholder={placeholder}
309      invalidMessage={invalidMessage} required={required}
310      {...rest}/>
```

```

307 );
308
309 DatePicker.propTypes = {
310   id: PropTypes.string.isRequired,
311   label: PropTypes.string.isRequired,
312   invalidMessage: requiredIf(PropTypes.string, props =>
313     props.required),
314   placeholder: PropTypes.string,
315   required: PropTypes.bool
316 };
317
318 DatePicker.defaultProps = {
319   placeholder: 'DD/MM/AAAA',
320 };
321
322 export default DatePicker;
323
324 webapp/src/components/form/datePicker/calendar.png
325
326 PNG
327
328 IHDR7bKGD      pHYsHHFk > IDAT ( } M
329 P0          p5
330 j - vbKj , m6q@E ; SPf =
331 ; mbW2@TpRXkx ~ 1Xu |8 w6 - HEZW , ~ |: "0
332
333 fPD ~ {
334
335 j %tEXtdate:create2018-08-03T09:53:38-04:00 %tEXtdate:modify2017-10-0
336
337 webapp/src/components/form/wizard/Navigation.jsx
338
339 import React from 'react';
340 import PropTypes from 'prop-types';
341 import { WithWizard } from "react-albus/lib";
342 import { Button, Row, Col } from 'reactstrap';
343
344 const Navigation = ({ onCancel, validate, onSubmit, submitLabel,
345   nextLabel, previousLabel, cancelLabel }) => (
346   <WithWizard
347     render={({ next, previous, step, steps }) => (
348       <Row>
349         {
350           steps.indexOf(step) === 0 && (
351             <Col>
352               <Button type="button" onClick={onCancel}>Cancel</Button>
353             </Col>
354           )
355         }
356       )
357     )
358   )
359
360   <Col>
361     <Button type="button" onClick={cancelLabel}>{ cancelLabel }</Button>
362   </Col>
363   <Col>
364     <Button type="button" onClick={previousLabel}>{ previousLabel }</Button>
365   </Col>
366   <Col>
367     <Button type="button" onClick={nextLabel}>{ nextLabel }</Button>
368   </Col>
369   <Col>
370     <Button type="button" onClick={submitLabel}>{ submitLabel }</Button>
371   </Col>
372 </Row>
373 )
374
375 <Col>
376   <Button type="button" onClick={validate}>Validate</Button>
377 </Col>
378 <Col>
379   <Button type="button" onClick={onSubmit}>Submit</Button>
380 </Col>
381 </Row>
382
383 <Col>
384   <Text>{ step }</Text>
385 </Col>
386 <Col>
387   <Text>{ previous }</Text>
388 </Col>
389 <Col>
390   <Text>{ next }</Text>
391 </Col>
392 </Row>
393
394 <Col>
395   <Text>{ steps }</Text>
396 </Col>
397 </Row>
398
399 <Col>
400   <Text>{ cancelLabel }</Text>
401 </Col>
402 </Row>
403
404 <Col>
405   <Text>{ previousLabel }</Text>
406 </Col>
407 <Col>
408   <Text>{ nextLabel }</Text>
409 </Col>
410 </Row>
411
412 <Col>
413   <Text>{ submitLabel }</Text>
414 </Col>
415 </Row>
416
417 <Col>
418   <Text>{ validateLabel }</Text>
419 </Col>
420 </Row>
421
422 <Col>
423   <Text>{ cancelLabel }</Text>
424 </Col>
425 </Row>
426
427 <Col>
428   <Text>{ previousLabel }</Text>
429 </Col>
430 <Col>
431   <Text>{ nextLabel }</Text>
432 </Col>
433 </Row>
434
435 <Col>
436   <Text>{ submitLabel }</Text>
437 </Col>
438 </Row>
439
440 <Col>
441   <Text>{ validateLabel }</Text>
442 </Col>
443 </Row>
444
445 <Col>
446   <Text>{ cancelLabel }</Text>
447 </Col>
448 </Row>
449
450 <Col>
451   <Text>{ previousLabel }</Text>
452 </Col>
453 <Col>
454   <Text>{ nextLabel }</Text>
455 </Col>
456 </Row>
457
458 <Col>
459   <Text>{ submitLabel }</Text>
460 </Col>
461 </Row>
462
463 <Col>
464   <Text>{ validateLabel }</Text>
465 </Col>
466 </Row>
467
468 <Col>
469   <Text>{ cancelLabel }</Text>
470 </Col>
471 </Row>
472
473 <Col>
474   <Text>{ previousLabel }</Text>
475 </Col>
476 <Col>
477   <Text>{ nextLabel }</Text>
478 </Col>
479 </Row>
480
481 <Col>
482   <Text>{ submitLabel }</Text>
483 </Col>
484 </Row>
485
486 <Col>
487   <Text>{ validateLabel }</Text>
488 </Col>
489 </Row>
490
491 <Col>
492   <Text>{ cancelLabel }</Text>
493 </Col>
494 </Row>
495
496 <Col>
497   <Text>{ previousLabel }</Text>
498 </Col>
499 <Col>
500   <Text>{ nextLabel }</Text>
501 </Col>
502 </Row>
503
504 <Col>
505   <Text>{ submitLabel }</Text>
506 </Col>
507 </Row>
508
509 <Col>
510   <Text>{ validateLabel }</Text>
511 </Col>
512 </Row>
513
514 <Col>
515   <Text>{ cancelLabel }</Text>
516 </Col>
517 </Row>
518
519 <Col>
520   <Text>{ previousLabel }</Text>
521 </Col>
522 <Col>
523   <Text>{ nextLabel }</Text>
524 </Col>
525 </Row>
526
527 <Col>
528   <Text>{ submitLabel }</Text>
529 </Col>
530 </Row>
531
532 <Col>
533   <Text>{ validateLabel }</Text>
534 </Col>
535 </Row>
536
537 <Col>
538   <Text>{ cancelLabel }</Text>
539 </Col>
540 </Row>
541
542 <Col>
543   <Text>{ previousLabel }</Text>
544 </Col>
545 <Col>
546   <Text>{ nextLabel }</Text>
547 </Col>
548 </Row>
549
550 <Col>
551   <Text>{ submitLabel }</Text>
552 </Col>
553 </Row>
554
555 <Col>
556   <Text>{ validateLabel }</Text>
557 </Col>
558 </Row>
559
560 <Col>
561   <Text>{ cancelLabel }</Text>
562 </Col>
563 </Row>
564
565 <Col>
566   <Text>{ previousLabel }</Text>
567 </Col>
568 <Col>
569   <Text>{ nextLabel }</Text>
570 </Col>
571 </Row>
572
573 <Col>
574   <Text>{ submitLabel }</Text>
575 </Col>
576 </Row>
577
578 <Col>
579   <Text>{ validateLabel }</Text>
580 </Col>
581 </Row>
582
583 <Col>
584   <Text>{ cancelLabel }</Text>
585 </Col>
586 </Row>
587
588 <Col>
589   <Text>{ previousLabel }</Text>
590 </Col>
591 <Col>
592   <Text>{ nextLabel }</Text>
593 </Col>
594 </Row>
595
596 <Col>
597   <Text>{ submitLabel }</Text>
598 </Col>
599 </Row>
600
601 <Col>
602   <Text>{ validateLabel }</Text>
603 </Col>
604 </Row>
605
606 <Col>
607   <Text>{ cancelLabel }</Text>
608 </Col>
609 </Row>
610
611 <Col>
612   <Text>{ previousLabel }</Text>
613 </Col>
614 <Col>
615   <Text>{ nextLabel }</Text>
616 </Col>
617 </Row>
618
619 <Col>
620   <Text>{ submitLabel }</Text>
621 </Col>
622 </Row>
623
624 <Col>
625   <Text>{ validateLabel }</Text>
626 </Col>
627 </Row>
628
629 <Col>
630   <Text>{ cancelLabel }</Text>
631 </Col>
632 </Row>
633
634 <Col>
635   <Text>{ previousLabel }</Text>
636 </Col>
637 <Col>
638   <Text>{ nextLabel }</Text>
639 </Col>
640 </Row>
641
642 <Col>
643   <Text>{ submitLabel }</Text>
644 </Col>
645 </Row>
646
647 <Col>
648   <Text>{ validateLabel }</Text>
649 </Col>
650 </Row>
651
652 <Col>
653   <Text>{ cancelLabel }</Text>
654 </Col>
655 </Row>
656
657 <Col>
658   <Text>{ previousLabel }</Text>
659 </Col>
660 <Col>
661   <Text>{ nextLabel }</Text>
662 </Col>
663 </Row>
664
665 <Col>
666   <Text>{ submitLabel }</Text>
667 </Col>
668 </Row>
669
670 <Col>
671   <Text>{ validateLabel }</Text>
672 </Col>
673 </Row>
674
675 <Col>
676   <Text>{ cancelLabel }</Text>
677 </Col>
678 </Row>
679
680 <Col>
681   <Text>{ previousLabel }</Text>
682 </Col>
683 <Col>
684   <Text>{ nextLabel }</Text>
685 </Col>
686 </Row>
687
688 <Col>
689   <Text>{ submitLabel }</Text>
690 </Col>
691 </Row>
692
693 <Col>
694   <Text>{ validateLabel }</Text>
695 </Col>
696 </Row>
697
698 <Col>
699   <Text>{ cancelLabel }</Text>
700 </Col>
701 </Row>
702
703 <Col>
704   <Text>{ previousLabel }</Text>
705 </Col>
706 <Col>
707   <Text>{ nextLabel }</Text>
708 </Col>
709 </Row>
710
711 <Col>
712   <Text>{ submitLabel }</Text>
713 </Col>
714 </Row>
715
716 <Col>
717   <Text>{ validateLabel }</Text>
718 </Col>
719 </Row>
720
721 <Col>
722   <Text>{ cancelLabel }</Text>
723 </Col>
724 </Row>
725
726 <Col>
727   <Text>{ previousLabel }</Text>
728 </Col>
729 <Col>
730   <Text>{ nextLabel }</Text>
731 </Col>
732 </Row>
733
734 <Col>
735   <Text>{ submitLabel }</Text>
736 </Col>
737 </Row>
738
739 <Col>
740   <Text>{ validateLabel }</Text>
741 </Col>
742 </Row>
743
744 <Col>
745   <Text>{ cancelLabel }</Text>
746 </Col>
747 </Row>
748
749 <Col>
750   <Text>{ previousLabel }</Text>
751 </Col>
752 <Col>
753   <Text>{ nextLabel }</Text>
754 </Col>
755 </Row>
756
757 <Col>
758   <Text>{ submitLabel }</Text>
759 </Col>
760 </Row>
761
762 <Col>
763   <Text>{ validateLabel }</Text>
764 </Col>
765 </Row>
766
767 <Col>
768   <Text>{ cancelLabel }</Text>
769 </Col>
770 </Row>
771
772 <Col>
773   <Text>{ previousLabel }</Text>
774 </Col>
775 <Col>
776   <Text>{ nextLabel }</Text>
777 </Col>
778 </Row>
779
780 <Col>
781   <Text>{ submitLabel }</Text>
782 </Col>
783 </Row>
784
785 <Col>
786   <Text>{ validateLabel }</Text>
787 </Col>
788 </Row>
789
790 <Col>
791   <Text>{ cancelLabel }</Text>
792 </Col>
793 </Row>
794
795 <Col>
796   <Text>{ previousLabel }</Text>
797 </Col>
798 <Col>
799   <Text>{ nextLabel }</Text>
800 </Col>
801 </Row>
802
803 <Col>
804   <Text>{ submitLabel }</Text>
805 </Col>
806 </Row>
807
808 <Col>
809   <Text>{ validateLabel }</Text>
810 </Col>
811 </Row>
812
813 <Col>
814   <Text>{ cancelLabel }</Text>
815 </Col>
816 </Row>
817
818 <Col>
819   <Text>{ previousLabel }</Text>
820 </Col>
821 <Col>
822   <Text>{ nextLabel }</Text>
823 </Col>
824 </Row>
825
826 <Col>
827   <Text>{ submitLabel }</Text>
828 </Col>
829 </Row>
830
831 <Col>
832   <Text>{ validateLabel }</Text>
833 </Col>
834 </Row>
835
836 <Col>
837   <Text>{ cancelLabel }</Text>
838 </Col>
839 </Row>
840
841 <Col>
842   <Text>{ previousLabel }</Text>
843 </Col>
844 <Col>
845   <Text>{ nextLabel }</Text>
846 </Col>
847 </Row>
848
849 <Col>
850   <Text>{ submitLabel }</Text>
851 </Col>
852 </Row>
853
854 <Col>
855   <Text>{ validateLabel }</Text>
856 </Col>
857 </Row>
858
859 <Col>
860   <Text>{ cancelLabel }</Text>
861 </Col>
862 </Row>
863
864 <Col>
865   <Text>{ previousLabel }</Text>
866 </Col>
867 <Col>
868   <Text>{ nextLabel }</Text>
869 </Col>
870 </Row>
871
872 <Col>
873   <Text>{ submitLabel }</Text>
874 </Col>
875 </Row>
876
877 <Col>
878   <Text>{ validateLabel }</Text>
879 </Col>
880 </Row>
881
882 <Col>
883   <Text>{ cancelLabel }</Text>
884 </Col>
885 </Row>
886
887 <Col>
888   <Text>{ previousLabel }</Text>
889 </Col>
890 <Col>
891   <Text>{ nextLabel }</Text>
892 </Col>
893 </Row>
894
895 <Col>
896   <Text>{ submitLabel }</Text>
897 </Col>
898 </Row>
899
900 <Col>
901   <Text>{ validateLabel }</Text>
902 </Col>
903 </Row>
904
905 <Col>
906   <Text>{ cancelLabel }</Text>
907 </Col>
908 </Row>
909
910 <Col>
911   <Text>{ previousLabel }</Text>
912 </Col>
913 <Col>
914   <Text>{ nextLabel }</Text>
915 </Col>
916 </Row>
917
918 <Col>
919   <Text>{ submitLabel }</Text>
920 </Col>
921 </Row>
922
923 <Col>
924   <Text>{ validateLabel }</Text>
925 </Col>
926 </Row>
927
928 <Col>
929   <Text>{ cancelLabel }</Text>
930 </Col>
931 </Row>
932
933 <Col>
934   <Text>{ previousLabel }</Text>
935 </Col>
936 <Col>
937   <Text>{ nextLabel }</Text>
938 </Col>
939 </Row>
940
941 <Col>
942   <Text>{ submitLabel }</Text>
943 </Col>
944 </Row>
945
946 <Col>
947   <Text>{ validateLabel }</Text>
948 </Col>
949 </Row>
950
951 <Col>
952   <Text>{ cancelLabel }</Text>
953 </Col>
954 </Row>
955
956 <Col>
957   <Text>{ previousLabel }</Text>
958 </Col>
959 <Col>
960   <Text>{ nextLabel }</Text>
961 </Col>
962 </Row>
963
964 <Col>
965   <Text>{ submitLabel }</Text>
966 </Col>
967 </Row>
968
969 <Col>
970   <Text>{ validateLabel }</Text>
971 </Col>
972 </Row>
973
974 <Col>
975   <Text>{ cancelLabel }</Text>
976 </Col>
977 </Row>
978
979 <Col>
980   <Text>{ previousLabel }</Text>
981 </Col>
982 <Col>
983   <Text>{ nextLabel }</Text>
984 </Col>
985 </Row>
986
987 <Col>
988   <Text>{ submitLabel }</Text>
989 </Col>
990 </Row>
991
992 <Col>
993   <Text>{ validateLabel }</Text>
994 </Col>
995 </Row>
996
997 <Col>
998   <Text>{ cancelLabel }</Text>
999 </Col>
1000 </Row>
1001
1002 <Col>
1003   <Text>{ previousLabel }</Text>
1004 </Col>
1005 <Col>
1006   <Text>{ nextLabel }</Text>
1007 </Col>
1008 </Row>
1009
1010 <Col>
1011   <Text>{ submitLabel }</Text>
1012 </Col>
1013 </Row>
1014
1015 <Col>
1016   <Text>{ validateLabel }</Text>
1017 </Col>
1018 </Row>
1019
1020 <Col>
1021   <Text>{ cancelLabel }</Text>
1022 </Col>
1023 </Row>
1024
1025 <Col>
1026   <Text>{ previousLabel }</Text>
1027 </Col>
1028 <Col>
1029   <Text>{ nextLabel }</Text>
1030 </Col>
1031 </Row>
1032
1033 <Col>
1034   <Text>{ submitLabel }</Text>
1035 </Col>
1036 </Row>
1037
1038 <Col>
1039   <Text>{ validateLabel }</Text>
1040 </Col>
1041 </Row>
1042
1043 <Col>
1044   <Text>{ cancelLabel }</Text>
1045 </Col>
1046 </Row>
1047
1048 <Col>
1049   <Text>{ previousLabel }</Text>
1050 </Col>
1051 <Col>
1052   <Text>{ nextLabel }</Text>
1053 </Col>
1054 </Row>
1055
1056 <Col>
1057   <Text>{ submitLabel }</Text>
1058 </Col>
1059 </Row>
1060
1061 <Col>
1062   <Text>{ validateLabel }</Text>
1063 </Col>
1064 </Row>
1065
1066 <Col>
1067   <Text>{ cancelLabel }</Text>
1068 </Col>
1069 </Row>
1070
1071 <Col>
1072   <Text>{ previousLabel }</Text>
1073 </Col>
1074 <Col>
1075   <Text>{ nextLabel }</Text>
1076 </Col>
1077 </Row>
1078
1079 <Col>
1080   <Text>{ submitLabel }</Text>
1081 </Col>
1082 </Row>
1083
1084 <Col>
1085   <Text>{ validateLabel }</Text>
1086 </Col>
1087 </Row>
1088
1089 <Col>
1090   <Text>{ cancelLabel }</Text>
1091 </Col>
1092 </Row>
1093
1094 <Col>
1095   <Text>{ previousLabel }</Text>
1096 </Col>
1097 <Col>
1098   <Text>{ nextLabel }</Text>
1099 </Col>
1100 </Row>
1101
1102 <Col>
1103   <Text>{ submitLabel }</Text>
1104 </Col>
1105 </Row>
1106
1107 <Col>
1108   <Text>{ validateLabel }</Text>
1109 </Col>
1110 </Row>
1111
1112 <Col>
1113   <Text>{ cancelLabel }</Text>
1114 </Col>
1115 </Row>
1116
1117 <Col>
1118   <Text>{ previousLabel }</Text>
1119 </Col>
1120 <Col>
1121   <Text>{ nextLabel }</Text>
1122 </Col>
1123 </Row>
1124
1125 <Col>
1126   <Text>{ submitLabel }</Text>
1127 </Col>
1128 </Row>
1129
1130 <Col>
1131   <Text>{ validateLabel }</Text>
1132 </Col>
1133 </Row>
1134
1135 <Col>
1136   <Text>{ cancelLabel }</Text>
1137 </Col>
1138 </Row>
1139
1140 <Col>
1141   <Text>{ previousLabel }</Text>
1142 </Col>
1143 <Col>
1144   <Text>{ nextLabel }</Text>
1145 </Col>
1146 </Row>
1147
1148 <Col>
1149   <Text>{ submitLabel }</Text>
1150 </Col>
1151 </Row>
1152
1153 <Col>
1154   <Text>{ validateLabel }</Text>
1155 </Col>
1156 </Row>
1157
1158 <Col>
1159   <Text>{ cancelLabel }</Text>
1160 </Col>
1161 </Row>
1162
1163 <Col>
1164   <Text>{ previousLabel }</Text>
1165 </Col>
1166 <Col>
1167   <Text>{ nextLabel }</Text>
1168 </Col>
1169 </Row>
1170
1171 <Col>
1172   <Text>{ submitLabel }</Text>
1173 </Col>
1174 </Row>
1175
1176 <Col>
1177   <Text>{ validateLabel }</Text>
1178 </Col>
1179 </Row>
1180
1181 <Col>
1182   <Text>{ cancelLabel }</Text>
1183 </Col>
1184 </Row>
1185
1186 <Col>
1187   <Text>{ previousLabel }</Text>
1188 </Col>
1189 <Col>
1190   <Text>{ nextLabel }</Text>
1191 </Col>
1192 </Row>
1193
1194 <Col>
1195   <Text>{ submitLabel }</Text>
1196 </Col>
1197 </Row>
1198
1199 <Col>
1200   <Text>{ validateLabel }</Text>
1201 </Col>
1202 </Row>
1203
1204 <Col>
1205   <Text>{ cancelLabel }</Text>
1206 </Col>
1207 </Row>
1208
1209 <Col>
1210   <Text>{ previousLabel }</Text>
1211 </Col>
1212 <Col>
1213   <Text>{ nextLabel }</Text>
1214 </Col>
1215 </Row>
1216
1217 <Col>
1218   <Text>{ submitLabel }</Text>
1219 </Col>
1220 </Row>
1221
1222 <Col>
1223   <Text>{ validateLabel }</Text>
1224 </Col>
1225 </Row>
1226
1227 <Col>
1228   <Text>{ cancelLabel }</Text>
1229 </Col>
1230 </Row>
1231
1232 <Col>
1233   <Text>{ previousLabel }</Text>
1234 </Col>
1235 <Col>
1236   <Text>{ nextLabel }</Text>
1237 </Col>
1238 </Row>
1239
1240 <Col>
1241   <Text>{ submitLabel }</Text>
1242 </Col>
1243 </Row>
1244
1245 <Col>
1246   <Text>{ validateLabel }</Text>
1247 </Col>
1248 </Row>
1249
1250 <Col>
1251   <Text>{ cancelLabel }</Text>
1252 </Col>
1253 </Row>
1254
1255 <Col>
1256   <Text>{ previousLabel }</Text>
1257 </Col>
1258 <Col>
1259   <Text>{ nextLabel }</Text>
1260 </Col>
1261 </Row>
1262
1263 <Col>
1264   <Text>{ submitLabel }</Text>
1265 </Col>
1266 </Row>
1267
1268 <Col>
1269   <Text>{ validateLabel }</Text>
1270 </Col>
1271 </Row>
1272
1273 <Col>
1274   <Text>{ cancelLabel }</Text>
1275 </Col>
1276 </Row>
1277
1278 <Col>
1279   <Text>{ previousLabel }</Text>
1280 </Col>
1281 <Col>
1282   <Text>{ nextLabel }</Text>
1283 </Col>
1284 </Row>
1285
1286 <Col>
1287   <Text>{ submitLabel }</Text>
1288 </Col>
1289 </Row>
1290
1291 <Col>
1292   <Text>{ validateLabel }</Text>
1293 </Col>
1294 </Row>
1295
1296 <Col>
1297   <Text>{ cancelLabel }</Text>
1298 </Col>
1299 </Row>
1300
1301 <Col>
1302   <Text>{ previousLabel }</Text>
1303 </Col>
1304 <Col>
1305   <Text>{ nextLabel }</Text>
1306 </Col>
1307 </Row>
1308
1309 <Col>
1310   <Text>{ submitLabel }</Text>
1311 </Col>
1312 </Row>
1313
1314 <Col>
1315   <Text>{ validateLabel }</Text>
1316 </Col>
1317 </Row>
1318
1319 <Col>
1320   <Text>{ cancelLabel }</Text>
1321 </Col>
1322 </Row>
1323
1324 <Col>
1325   <Text>{ previousLabel }</Text>
1326 </Col>
1327 <Col>
1328   <Text>{ nextLabel }</Text>
1329 </Col>
1330 </Row>
1331
1332 <Col>
1333   <Text>{ submitLabel }</Text>
1334 </Col>
1335 </Row>
1336
1337 <Col>
1338   <Text>{ validateLabel }</Text>
1339 </Col>
1340 </Row>
1341
1342 <Col>
1343   <Text>{ cancelLabel }</Text>
1344 </Col>
1345 </Row>
1346
1347 <Col>
1348   <Text>{ previousLabel }</Text>
1349 </Col>
1350 <Col>
1351   <Text>{ nextLabel }</Text>
1352 </Col>
1353 </Row>
1354
1355 <Col>
1356   <Text>{ submitLabel }</Text>
1357 </Col>
1358 </Row>
1359
1360 <Col>
1361   <Text>{ validateLabel }</Text>
1362 </Col>
1363 </Row>
1364
1365 <Col>
1366   <Text>{ cancelLabel }</Text>
1367 </Col>
1368 </Row>
1369
1370 <Col>
1371   <Text>{ previousLabel }</Text>
1372 </Col>
1373 <Col>
1374   <Text>{ nextLabel }</Text>
1375 </Col>
1376 </Row>
1377
1378 <Col>
1379   <Text>{ submitLabel }</Text>
1380 </Col>
1381 </Row>
1382
1383 <Col>
1384   <Text>{ validateLabel }</Text>
1385 </Col>
1386 </Row>
1387
1388 <Col>
1389   <Text>{ cancelLabel }</Text>
1390 </Col>
1391 </Row>
1392
1393 <Col>
1394   <Text>{ previousLabel }</Text>
1395 </Col>
1396 <Col>
1397   <Text>{ nextLabel }</Text>
1398 </Col>
1399 </Row>
1400
1401 <Col>
1402   <Text>{ submitLabel }</Text>
1403 </Col>
1404 </Row>
1405
1406 <Col>
1407   <Text>{ validateLabel }</Text>
1408 </Col>
1409 </Row>
1410
1411 <Col>
1412   <Text>{ cancelLabel }</Text>
1413 </Col>
1414 </Row>
1415
1416 <Col>
1417   <Text>{ previousLabel }</Text>
1418 </Col>
1419 <Col>
1420   <Text>{ nextLabel }</Text>
1421 </Col>
1422 </Row>
1423
1424 <Col>
1425   <Text>{ submitLabel }</Text>
1426 </Col>
1427 </Row>
1428
1429 <Col>
1430   <Text>{ validateLabel }</Text>
1
```

```
348
349         <Col>
350             <Button color="danger"
351                 onClick={onCancel}>
352                 {cancelLabel}
353             </Button>
354         </Col>
355     )
356 }
357 {
358     steps.indexOf(step) > 0 && (
359         <Col>
360             <Button color="secondary"
361                 onClick={previous}>
362                 {previousLabel}
363             </Button>
364         </Col>
365     )
366 }
367 {
368     steps.indexOf(step) < steps.length - 1 && (
369         <Col>
370             <Button id="next" type="button"
371                 color="primary"
372                 className="float-right"
373                 onClick={() => {
374                     validate && validate(step);
375                     next();
376                 }}>
377                 {nextLabel}
378             </Button>
379         </Col>
380     )
381 }
382 {
383     steps.indexOf(step) === steps.length - 1 && (
384         <Col>
385             <Button id="next" type="button"
386                 color="primary"
387                 className="float-right"
388                 onClick={onSubmit}>
389                 {submitLabel}
390             </Button>
391         </Col>
392     )
393 }
```

```
385             </Row>
386         )}
387     />
388 );
389
390 Navigation.propTypes = {
391     onCancel: PropTypes.func.isRequired,
392     onSubmit: PropTypes.func.isRequired,
393     submitLabel : PropTypes.string.isRequired,
394     validate: PropTypes.func,
395     nextLabel: PropTypes.string,
396     previousLabel: PropTypes.string,
397     cancelLabel: PropTypes.string
398 };
399
400 Navigation.defaultProps = {
401     validate: () => null,
402     nextLabel: 'Avanar',
403     previousLabel: 'Voltar',
404     cancelLabel: 'Cancelar',
405 };
406
407 export default Navigation;
408
409 webapp/src/components/form/wizard/index.js
410
411 import Wizard from './Wizard';
412 export default Wizard;
413
414 webapp/src/components/form/wizard/Wizard.jsx
415
416 import React from 'react';
417 import PropTypes from 'prop-types';
418 import { Wizard, Steps, Step } from "react-albus/lib";
419 import { Form } from 'reactstrap';
420 import Navigation from './Navigation';
421
422 const WizardForm = ({ children, onCancel, validate, onSubmit,
423     submitLabel, nextLabel, previousLabel, cancelLabel }) => {
424     return (
425         <Form>
426             <Wizard>
427                 <Steps>
428                     {
```

```

428         React.Children.map(children, (child,
429             index) => (
430                 <Step id={'step-' + index}>
431                     {child}
432                 </Step>
433             ))
434         )
435     </Steps>
436     <Navigation {...{ onCancel, validate, onSubmit,
437         submitLabel, nextLabel, previousLabel,
438         cancelLabel}}/>
439     </Wizard>
440   );
441 }
442 Navigation.propTypes = {
443   children: PropTypes.element,
444   onCancel: PropTypes.func.isRequired,
445   onSubmit: PropTypes.func.isRequired,
446   submitLabel: PropTypes.string.isRequired,
447   validate: PropTypes.func,
448   nextLabel: PropTypes.string,
449   previousLabel: PropTypes.string,
450   cancelLabel: PropTypes.string
451 };
452 Navigation.defaultProps = {
453   validate: () => null,
454   nextLabel: 'Avanar',
455   previousLabel: 'Voltar',
456   cancelLabel: 'Cancelar'
457 };
458 }
459
460 export default WizardForm;
461
462 webapp/src/components/form/checkbox/Checkbox.jsx
463
464 import React from 'react';
465 import './checkbox.css';
466
467 export default ({ value, label, disabled, checked, onChange}) =>
468   (
469     <label className="control control--checkbox">{label}
```

```
469         <input type="checkbox" value={value} disabled={disabled}>
470             checked={checked} onChange={onChange}/>
471         <div className="control__indicator"></div>
472     </label>
473   )
474
475 webapp/src/components/form/checkbox/index.js
476
477 import Checkbox from './Checkbox.jsx';
478 export default Checkbox;
479
480 webapp/src/components/form/checkbox/checkbox.css
481
482 .control {
483   display: block;
484   position: relative;
485   padding-left: 30px;
486   margin-bottom: 15px;
487   cursor: pointer;
488   font-size: 18px; }
489
490 .control input {
491   position: absolute;
492   z-index: -1;
493   opacity: 0; }
494
495 .control input:focus ~ .control__indicator {
496   background: #ccc; }
497
498 .control input:checked ~ .control__indicator {
499   background: #2aa1c0; }
500
501 .control input:checked ~ .control__indicator:after {
502   display: block; }
503
504 .control input:checked:focus ~ .control__indicator {
505   background: #0e647d; }
506
507 .control input:disabled ~ .control__indicator {
508   background: #e6e6e6;
509   opacity: 0.6;
510   pointer-events: none; }
511
512 .control:hover input ~ .control__indicator {
513   background: #ccc; }
514
515 .control:hover input:not([disabled]):checked ~
516   .control__indicator {
517   background: #0e647d; }

518
519 .control__indicator {
520   position: absolute;
521   top: 2px;
```

```
512     left: 0;
513     height: 20px;
514     width: 20px;
515     background: #e6e6e6; }
516     .control__indicator:after {
517       content: '';
518       position: absolute;
519       display: none; }

520     .control--checkbox .control__indicator:after {
521       left: 8px;
522       top: 3px;
523       width: 5px;
524       height: 10px;
525       border: solid #fff;
526       border-width: 0 2px 2px 0;
527       transform: rotate(45deg); }

529     .control--checkboxinput:disabled ~ .control__indicator:after {
530       border-color: #7b7b7b; }

532

533     webapp/src/components/form/checkbox/checkbox.sass

535     .control
536       display: block
537       position: relative
538       padding-left: 30px
539       margin-bottom: 15px
540       cursor: pointer
541       font-size: 18px
542       input
543         position: absolute
544         z-index: -1
545         opacity: 0
546         &:focus ~ .control__indicator
547           background: #ccc
548         &:checked ~ .control__indicator
549           background: #2aa1c0
550           &:after
551             display: block
552             &:checked:focus ~ .control__indicator
553               background: #0e647d
554             &:disabled ~ .control__indicator
555               background: #e6e6e6
```

```

557         opacity: 0.6
558         pointer-events: none
559
560     &:hover
561         input ~ .control__indicator
562             background: #ccc
563         input:not([disabled]):checked ~ .control__indicator
564             background: #0e647d
565
566     .control__indicator
567         position: absolute
568         top: 2px
569         left: 0
570         height: 20px
571         width: 20px
572         background: #e6e6e6
573     &:after
574         content: ''
575         position: absolute
576         display: none
577
578     .control--checkbox
579         .control__indicator:after
580             left: 8px;
581             top: 3px;
582             width: 5px;
583             height: 10px;
584             border: solid #fff
585             border-width: 0 2px 2px 0
586             transform: rotate(45deg)
587     &input:disabled ~ .control__indicator:after
588         border-color: #7b7b7b
589
590
591 webapp/src/components/form/option/index.js
592
593 import Option from './Option.jsx'
594 export default Option;
595
596 webapp/src/components/form/option/Option.jsx
597
598 import React from 'react';
599 import { FormGroup, Input, Label, FormFeedback } from
600     'reactstrap';
601 import Required from '../Required.jsx'

```

```
601 import PropTypes from 'prop-types';
602 import requiredIf from 'react-required-if';
603
604 const Option = ({label, type, onChange, required,
605   invalidMessage, ...rest}) => (
606   <FormGroup check>
607     <Label check>
608       <Input type={type} onChange={onChange} {...rest}/>
609       {', ' + label}
610       {required? <Required/>: null}
611     </Label>
612     <FormFeedback>{invalidMessage}</FormFeedback>
613   </FormGroup>
614 );
615
616 Option.propTypes = {
617   label: PropTypes.string.isRequired,
618   invalidMessage: requiredIf(PropTypes.string, props =>
619     props.required),
620   type: PropTypes.oneOf(['checkbox', 'radio']),
621   required: PropTypes.bool,
622 };
623
624 Option.defaultProps = {
625   type: 'radio',
626 };
627
628 export default Option;
629
630 webapp/src/components/form/switch/Switch.jsx
631
632 import React from 'react';
633 import { Label } from 'reactstrap';
634 import Switch from 'rc-switch';
635 import 'rc-switch/assets/index.css';
636 import './switch.css';
637
638 export default ({ id, label, onChange, ...rest }) => (
639   <div>
640     <Label for={id}>{label}</Label>
641     <Switch id={id} onChange={onChange}
642       checkedChildren="Sim" unCheckedChildren="No"
643       style={{width: '50px'}} {...rest}/>
644   </div>
645 )
```

```
642
643 webapp/src/components/form/switch/switch.sass
644
645 .rc-switch-checked
646   &:after
647     left: 65%
648
649 webapp/src/components/form/switch/index.js
650
651 import Switch from './Switch.jsx'
652 export default Switch;
653
654 webapp/src/components/form/switch/switch.css
655
656 .rc-switch-checked:after {
657   left: 65%; }
658
659
660 webapp/src/components/form/fileInput/FileInput.jsx
661
662 import React from 'react';
663 import PropTypes from 'prop-types';
664 import { Input, Label } from 'reactstrap';
665 import './fileInput.css'
666
667 const FileInput = ({label, onChange, className, color, ...rest})
668   => (
669     <Label className={'btn btn-${color} ${className}'>
670       <Input type="file" name="file" onChange={onChange}>
671         {...rest}</Input>
672       {label}
673     </Label>
674   );
675
676 FileInput.propTypes = {
677   label: PropTypes.string.isRequired,
678   onChange: PropTypes.func.isRequired,
679   className: PropTypes.string,
680   color: PropTypes.oneOf([
681     'primary',
682     'success',
683     'info',
684     'warning',
685     'danger'
686   ])
687 }
```

```
685 };
686
687 FileInput.defaultProps = {
688   color: 'info',
689 };
690
691 export default FileInput;
692
693 webapp/src/components/form/fileInput/fileInput.sass
694
695 label
696   cursor: pointer
697   input[type=file]
698     display: none;
699
700 webapp/src/components/form/fileInput/index.js
701
702 import FileInput from './FileInput';
703 export default FileInput;
704
705 webapp/src/components/form/fileInput/fileInput.css
706
707 label {
708   cursor: pointer; }
709   label input[type=file] {
710     display: none; }
711
712
713 webapp/src/components/form/Required.jsx
714
715 import React from 'react';
716 export default () => (<span className="text-danger">*</span>);
717
718 webapp/src/components/form/index.js
719
720 export {default as Input} from './input';
721 export {default as FileInput} from './fileInput';
722 export {default as DatePicker} from './datePicker';
723 export {default as MaskInput} from './maskInput';
724 export {default as Wizard} from './wizard';
725 export {default as Option} from './option';
726 export {default as Switch} from './switch';
727 export {default as Checkbox} from './checkbox';
728 export {default as Required} from './Required.jsx';
729
```

```
730  webapp/src/components/calender/Calendar.jsx
731
732  import React, { Component } from 'react';
733  import BigCalendar from 'react-big-calendar',
734  import PropTypes from 'prop-types';
735  import requiredIf from 'react-required-if';
736  import dates from 'react-big-calendar/lib/utils/dates';
737  import moment from 'moment';
738  import 'react-big-calendar/lib/css/react-big-calendar.css';
739  import './calendar.css';
740
741  BigCalendar.setLocalizer(BigCalendar.momentLocalizer(moment))
742
743  const { MONTH, WEEK, DAY } = BigCalendar.Views
744  const allViews = [MONTH, WEEK, DAY]
745
746  export default class extends Component{
747      static propTypes = {
748          selectable: PropTypes.bool,
749          events: PropTypes.array,
750          onSelectEvent: requiredIf(PropTypes.func, props =>
751              props.selectable),
752          onSelectSlot: PropTypes.func
753      };
754      static defaultProps = {
755          events: [],
756          onSelectEvent: () => null,
757          onSelectSlot: () => null
758      };
759      render(){
760          const { events, selectable, onSelectEvent, onSelectSlot,
761                  defaultView } = this.props;
762          return (
763              <BigCalendar
764                  selectable={selectable}
765                  onSelectEvent={onSelectEvent}
766                  onSelectSlot={onSelectSlot}
767                  events={events}
768                  defaultView={defaultView}
769                  views={allViews}
770                  max={dates.endOf(new Date(), 'day')}
771                  defaultDate={new Date()}
772              />
773          );
774      }
775  }
```

```
773 }
774
775 webapp/src/components/calender/index.js
776
777 import Calendar from './Calendar.jsx';
778 export default Calendar;
779
780 webapp/src/components/calender/calendar.sass
781
782 .rbc-event-content
783   background-color: #007BFF
784   border-color: #007BFF
785
786 .rbc-toolbar
787   flex-wrap: wrap
788   justify-content: center
789
790
791 webapp/src/components/calender/calendar.css
792
793 .rbc-event-content {
794   background-color: #007BFF;
795   border-color: #007BFF; }
796
797 .rbc-toolbar {
798   flex-wrap: wrap;
799   justify-content: center; }
800
801
802 webapp/src/components/filter/Filter.jsx
803
804 import React from 'react';
805 import { Row, Col, Form } from 'reactstrap';
806 import PropTypes from 'prop-types';
807 import { Input } from '../form'
808
809 const Filter = ({ value, label, handlerFilter }) => (
810   <Row style={{ marginBottom: '5px' }}>
811     <Col>
812       <Form inline>
813         <Input id="filter" value={value} label={label}
814           onChange={handlerFilter} />
815       </Form>
816     </Col>
817   </Row>
```

```
817 );
818
819 Filter.propTypes = {
820     handlerFilter: PropTypes.func.isRequired,
821     value: PropTypes.string,
822     label: PropTypes.string
823 }
824
825 Filter.defaultProps ={
826     label: " "
827 }
828
829 export default Filter;
830
831 webapp/src/components/filter/index.js
832
833 import Filter from './Filter';
834 export default Filter;
835
836 webapp/src/components/header/Header.jsx
837
838 import React, { Fragment } from 'react';
839 import { Navbar, NavbarBrand, Nav, UncontrolledDropdown,
840         DropdownToggle, DropdownMenu, DropdownItem } from
841         'reactstrap';
842 import { NavLink } from 'react-router-dom';
843 import logo from './logo.png';
844 import { Roles } from '../../model';
845
846 export default ({role, name, logged, logout}) => (
847     <Navbar color="light" light>
848         <NavbarBrand tag="button" className="btn
849             btn-outline-secondary" style={{ borderWidth: '0' }}>
850             <NavLink to="/">
851                 <img src={logo} alt="Logo" height="35px"/>
852             </NavLink>
853         </NavbarBrand>
854         {
855             logged && (
856                 <Nav className="ml-auto" navbar>
857                     <UncontrolledDropdown nav inNavbar>
858                         <DropdownToggle nav caret>
859                             {name}
860                         </DropdownToggle>
861                         <DropdownMenu right>
```

```

859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
    {
      role === Roles.ADMIN && (
        <Fragment>
          <DropdownItem>
            <NavLink to="/admin">
              administrar
            </NavLink>
          </DropdownItem>
          <DropdownItem divider />
        </Fragment>
      )
    }
    <DropdownItem onClick={logout}>
      Sair
    </DropdownItem>
  </DropdownMenu>
</UncontrolledDropdown>
</Nav>
)
}
</Navbar>
);
webapp/src/components/index.js
export { default as Header } from './header/Header';
export { default as Footer } from './footer/Footer';
export {
  Input,
  FileInput,
  DatePicker,
  MaskInput,
  Wizard,
  Option,
  Switch,
  Required,
  Checkbox
} from './form';
export { default as Calendar } from './calender';
export { default as Pagination } from './pagination';
export { default as Filter } from './filter';
webapp/src/model/role.js
const Roles = {

```

```

904     ADMIN: 'ADMIN',
905     VOLUNTEER: 'VOLUNTEER',
906     COMPANY_ADMIN: 'COMPANY_ADMIN',
907     SCHOOL_ADMIN: 'SCHOOL_ADMIN',
908     values: () => [
909         Roles.ADMIN,
910         Roles.VOLUNTEER,
911         Roles.COMPANY_ADMIN,
912         Roles.SCHOOL_ADMIN
913     ],
914     has: role => Roles.values().includes(role),
915     translate: role =>{
916         return {
917             ADMIN: 'Administrador',
918             VOLUNTEER: 'Voluntrio',
919             COMPANY_ADMIN: 'Responsavel na empresa',
920             SCHOOL_ADMIN: 'Responsavel da escola',
921         }[role]
922     }
923 };
924
925 export default Roles;
926
927 webapp/src/model/question.js
928
929 export default class{
930     constructor(codQuestion, question = '', responds = '') {
931         this.codQuestion = codQuestion;
932         this.question = question;
933         this.responds = responds;
934     }
935 }
936
937 webapp/src/model/event.js
938
939 import Address from './address';
940 import School from './school';
941
942 export default class{
943     constructor(codEvent, startOccurrence, endOccurrence,
944                 createAt, project, type = '', allDay,
945                 address = new Address(), school = new School(),
946                 trainings = [],
947                 participants = [], ratings = [] ) {
948         this.codEvent = codEvent;

```

```
947         this.startOccurrence = startOccurrence;
948         this.endOccurrence = endOccurrence;
949         this.createAt = createAt;
950         this.project = project;
951         this.type = type;
952         this.allDay = allDay;
953         this.address = address;
954         this.trainings = trainings;
955         this.participants = participants;
956         this.ratings = ratings;
957         this.school = school;
958     }
959 }
960
961 webapp/src/model/user.js
962
963 import Address from './address';
964
965 export default class{
966     constructor(cpf='', name='', email='', password='', role='',
967                 status='', birth='', job='', phone='', address=new Address() ) {
968         this.cpf = cpf;
969         this.name = name;
970         this.email = email;
971         this.password = password;
972         this.role = role;
973         this.status = status;
974         this.birth = birth;
975         this.job = job;
976         this.phone = phone;
977         this.address = address;
978     }
979 }
980
981 webapp/src/model/school.js
982
983 import Address from './address';
984 import User from './user';
985 import Roles from './role';
986 import Status from './status';
987
988 export default class {
989     constructor(codSchool, name = '', phone = '', schoolType =
990                 '', address = new Address(), responsible = new User()) {
```

```
990     this.codSchool = codSchool;
991     this.name = name
992     this.phone = phone;
993     this.schoolType = schoolType;
994     this.address = address;
995     responsible.role = Roles.SCHOOL_ADMIN;
996     responsible.status = Status.APPROVED
997     this.responsible = responsible
998   }
999 }
1000
1001 webapp/src/model/address.js
1002
1003 export default class {
1004   constructor(codAddress, city = '', neighborhood = '', street
1005     = '', number = '') {
1006     this.codAddress = codAddress;
1007     this.city = city;
1008     this.neighborhood = neighborhood;
1009     this.street = street;
1010     this.number = number;
1011   }
1012   toString(){
1013     return `${this.street}, ${this.number},
1014       ${this.neighborhood} - ${this.city}`;
1015   }
1016 }
1017
1018 webapp/src/model/index.js
1019
1020 export { default as Address } from './address';
1021 export { default as Roles } from './role';
1022 export { default as Status } from './status';
1023 export { default as User } from './user';
1024 export { default as Volunteer } from './volunteer';
1025 export { default as Question } from './question';
1026 export { default as Answer } from './answer';
1027 export { default as Schooling } from './schooling';
1028 export { default as SchoolType } from './schoolType';
1029 export { default as School } from './school';
1030 export { default as Event } from './event';
1031 export { default as EventType } from './eventType';
1032 export { default as Attachment } from './attachment';
1033 export { default as Training } from './training';
1034 export { default as Project } from './project';
```

```
1033
1034 webapp/src/model/volunteer.js
1035
1036 import User from './user';
1037 import Roles from './role';
1038 import Status from './status';
1039 import Company from './company';
1040 import School from './school';
1041
1042 export default class{
1043     constructor(user = new User(), company = new Company(),
1044                 school = new School(),
1045                 schooling = '', conclusion = false, course = '', rg = '',
1046                 answers = [], ratings = []) {
1047         user.role = Roles.VOLUNTEER;
1048         user.status = Status.REGISTER;
1049         this.user = user;
1050         this.company = company;
1051         this.schooling = schooling;
1052         this.conclusion = conclusion;
1053         this.course = course;
1054         this.rg = rg;
1055         this.answers = answers;
1056         this.ratings = ratings;
1057         this.school = school;
1058     }
1059 }
1060
1061 webapp/src/model/answer.js
1062
1063 import Question from './question';
1064 export default class {
1065     constructor(question = new Question(), answer = false,
1066                 comment = '') {
1067         this.question = question;
1068         this.answer = answer;
1069         this.comment = comment;
1070     }
1071
1072 webapp/src/model/company.js
1073
1074 import Address from './address';
1075 import User from './user';
1076 import Roles from './role';
```

```
1075 import Status from './status';
1076
1077 export default class{
1078     constructor(cnpj='', name='', phone='', address = new
1079         Address(), responsible = new User()) {
1080         this.cnpj = cnpj;
1081         this.name = name;
1082         this.phone = phone;
1083         this.address = address;
1084         responsible.role = Roles.COMPANY_ADMIN;
1085         responsible.status = Status.APPROVED
1086         this.responsible = responsible
1087     }
1088 }
1089
1090 webapp/src/model/project.js
1091
1092 export default class{
1093     constructor(codProject, start, end, name='') {
1094         this.codProject = codProject;
1095         this.name = name;
1096         this.start = start;
1097         this.end = end;
1098     }
1099 }
1100
1101 webapp/src/model/status.js
1102
1103 const Status = {
1104     APPROVED: 'APPROVED',
1105     WAIT_STATEMENT: 'WAIT_STATEMENT',
1106     WAIT_COMPANY: 'WAIT_COMPANY',
1107     WAIT_TRAINING: 'WAIT_TRAINING',
1108     REGISTER: 'REGISTER',
1109     values: () => [
1110         Status.APPROVED,
1111         Status.WAIT_STATEMENT,
1112         Status.WAIT_COMPANY,
1113         Status.WAIT_TRAINING,
1114         Status.REGISTER
1115     ],
1116     has: status => Status.values().includes(status),
1117     translate: status => {
1118         return {
1119             APPROVED: 'Aprovado',
```

```

1119         WAIT_STATEMENT: 'Aguardando envio do termo de
1120             responsabilidade',
1121         WAIT_COMPANY: 'Aguardando recomendao da empresa',
1122         WAIT_TRAINING: 'Aguardando treinamento',
1123         REGISTER: 'Registrado',
1124     }[status];
1125 },
1126 next: status => {
1127     return {
1128         APPROVED: Status.APPROVED,
1129         WAIT_STATEMENT: Status.WAIT_TRAINING,
1130         WAIT_COMPANY: Status.WAIT_STATEMENT,
1131         WAIT_TRAINING: Status.APPROVED,
1132         REGISTER: Status.WAIT_COMPANY,
1133     }[status];
1134 }
1135 };
1136 export default Status;
1137
1138 webapp/src/model/training.js
1139
1140 export default class{
1141     constructor(codTraining, name = '', path = '', link = '') {
1142         this.codTraining = codTraining;
1143         this.name = name;
1144         this.path = path;
1145         this.link = link;
1146     }
1147 }
1148
1149 webapp/src/model/schooling.js
1150
1151 const Schooling = {
1152     ELEMENTARY_SCHOOL: 'ELEMENTARY_SCHOOL',
1153     HIGH SCHOOL: 'HIGH SCHOOL',
1154     UNIVERSITY_GRADUATE: 'UNIVERSITY_GRADUATE',
1155     POSTGRADUATE_STUDIES: 'POSTGRADUATE_STUDIES',
1156     values: () => [
1157         Schooling.ELEMENTARY_SCHOOL,
1158         Schooling.HIGH SCHOOL,
1159         Schooling.UNIVERSITY_GRADUATE,
1160         Schooling.POSTGRADUATE_STUDIES,
1161     ],
1162     has: schooling => Schooling.values().includes(schooling),

```

```
1163     translate: schooling => {
1164         return [
1165             ELEMENTARY_SCHOOL: '1 Grau (Fundamental)',
1166             HIGH SCHOOL: '2 Grau (Ensino Mdio)',
1167             UNIVERSITY_GRADUATE: 'Ensino Superior (Graduaao)',
1168             POSTGRADUATE_STUDIES: 'Ps- Graduaao',
1169         )[schooling];
1170     }
1171 }
1172 }
1173 export default Schooling;
1174
1175 webapp/src/model/attachment.js
1176
1177 export default class{
1178     constructor(codAttachment, name = '', status = '', required
1179         = false, download = false) {
1180         this.codAttachment = codAttachment;
1181         this.name = name;
1182         this.required = required;
1183         this.status = status;
1184         this.download = download;
1185     }
1186 }
1187 webapp/src/model/schoolType.js
1188
1189 const SchoolType = {
1190     MUNICIPAL: 'MUNICIPAL',
1191     STATE: 'STATE',
1192     FEDERAL: 'FEDERAL',
1193     values: () => [
1194         SchoolType.MUNICIPAL,
1195         SchoolType.STATE,
1196         SchoolType.FEDERAL
1197     ],
1198     has: type => SchoolType.values().includes(type),
1199     translate: type => {
1200         return [
1201             MUNICIPAL: 'Municipal',
1202             STATE: 'Estadual',
1203             FEDERAL: 'Federal',
1204         )[type];
1205     }
1206 }
```

```
1207
1208     export default SchoolType;
1209
1210 webapp/src/model/eventType.js
1211
1212 export default class{
1213     constructor(type, name, trainings=[]){
1214         this.type = type;
1215         this.name = name;
1216         this.trainings = trainings;
1217     }
1218 }
1219
1220 webapp/src/pages/public/register/Page.jsx
1221
1222 import React, { Component } from 'react';
1223 import { Row, Col, Form, Label } from 'reactstrap';
1224 import { Input, Wizard, DatePicker, MaskInput, Option, Switch,
1225     Required } from '../../../../../components';
1226 import { Volunteer, Schooling, Answer, Roles } from
1227     '../../../../../model';
1228 import { getRequest, postRequest } from '../../../../../utils/http';
1229 import axios from 'axios';
1230
1231
1232 export default class extends Component{
1233     constructor(){
1234         super();
1235         this.state = {
1236             volunteer: new Volunteer(),
1237             questions: [],
1238             companies: [],
1239             confirmPassword: ''
1240         }
1241         this.handlerName = this.handlerName.bind(this);
1242         this.handlerCpf = this.handlerCpf.bind(this);
1243         this.handlerRg = this.handlerRg.bind(this);
1244         this.handlerPhone = this.handlerPhone.bind(this);
1245         this.handlerComplete = this.handlerComplete.bind(this)
1246         this.handlerCourse = this.handlerCourse.bind(this)
1247         this.handlerSchooling = this.handlerSchooling.bind(this)
1248         this.handlerJob = this.handlerJob.bind(this);
1249         this.handlerCompany = this.handlerCompany.bind(this);
1250         this.handlerEmail = this.handlerEmail.bind(this);
1251         this.handlerBirth = this.handlerBirth.bind(this);
1252         this.handlerPassword = this.handlerPassword.bind(this);
```

```
1250     this.handlerConfirmPassword =
1251         this.handlerConfirmPassword.bind(this);
1252     this.handlerCity = this.handlerCity.bind(this);
1253     this.handlerNeighborhood =
1254         this.handlerNeighborhood.bind(this);
1255     this.handlerStreet = this.handlerStreet.bind(this);
1256     this.handlerNumber = this.handlerNumber.bind(this);
1257     this.handlerAnswer = this.handlerAnswer.bind(this);
1258     this.handlerAnswerComment =
1259         this.handlerAnswerComment.bind(this);
1260     this.handlerSubmit = this.handlerSubmit.bind(this);
1261 }
1262
1263 componentWillMount(){
1264     axios.all([
1265         getRequest('/question/${Roles.VOLUNTEER}', res =>
1266             res.data),
1267         getRequest('/company', res => res.data )
1268     ]).then(res => {
1269         const { volunteer } = this.state;
1270         volunteer.answers = res[0].map(question => new
1271             Answer(question))
1272         this.setState({ questions: res[0], companies: res[1]
1273             })
1274     });
1275 }
1276
1277 // user
1278 handlerName(event){
1279     const { volunteer } = this.state;
1280     const { user } = volunteer;
1281     user.name = event.target.value;
1282     volunteer.user = user;
1283     this.setState({ volunteer });
1284 }
1285
1286 handlerRg(event){
1287     const { volunteer } = this.state;
1288     volunteer.rg = event.target.value.replace('_', '');
1289     this.setState({ volunteer });
1290 }
1291
1292 handlerCpf(event){
1293     const { volunteer } = this.state;
1294     const { user } = volunteer;
```

```
1289     user.cpf = event.target.value.replace(/[-]/g, '');
1290     volunteer.user = user;
1291     this.setState({ volunteer });
1292 }
1293
1294 handlerPhone(event) {
1295     const { volunteer } = this.state;
1296     const { user } = volunteer;
1297     user.phone = event.target.value.replace(/[()]/g, '');
1298     volunteer.user = user;
1299     this.setState({ volunteer });
1300 }
1301
1302 handlerJob(event){
1303     const { volunteer } = this.state;
1304     const { user } = volunteer;
1305     user.job = event.target.value;
1306     volunteer.user = user;
1307     this.setState({ volunteer });
1308 }
1309
1310 handlerCompany(event){
1311     const { volunteer, companies } = this.state;
1312     for (const company of companies) {
1313         if(company.cnpj === event.target.value){
1314             volunteer.company = company;
1315             break;
1316         }
1317     }
1318     this.setState({ volunteer });
1319 }
1320
1321 handlerEmail(event){
1322     const { volunteer } = this.state;
1323     const { user } = volunteer;
1324     user.email = event.target.value;
1325     volunteer.user = user;
1326     this.setState({ volunteer });
1327 }
1328
1329 handlerBirth(event){
1330     const { volunteer } = this.state;
1331     const { user } = volunteer;
1332     const { value } = event.target;
1333     user.birth = value ? new Date(value) : null;
```

```
1334     volunteer.user = user;
1335     this.setState({ volunteer });
1336 }
1337
1338 handlerComplete(value){
1339     const { volunteer } = this.state;
1340     volunteer.conclusion = value;
1341     this.setState({ volunteer });
1342 }
1343
1344 handlerSchooling(event){
1345     const { volunteer } = this.state;
1346     volunteer.schooling = event.target.value;
1347     this.setState({ volunteer });
1348 }
1349
1350 handlerCourse(event){
1351     const { volunteer } = this.state;
1352     volunteer.course = event.target.value;
1353     this.setState({ volunteer });
1354 }
1355
1356 handlerPassword(event){
1357     const { volunteer } = this.state;
1358     const { user } = volunteer;
1359     user.password = event.target.value;
1360     volunteer.user = user;
1361     this.setState({ volunteer });
1362 }
1363
1364 handlerConfirmPassword(event){
1365     let { confirmPassword } = this.state;
1366     confirmPassword = event.target.value;
1367     this.setState({ confirmPassword });
1368 }
1369
1370 //address
1371 handlerCity(event) {
1372     const { volunteer } = this.state;
1373     const { address } = volunteer.user;
1374     address.city = event.target.value;
1375     volunteer.user.address = address;
1376     this.setState({ volunteer });
1377 }
1378
```

```
1379     handlerNeighborhood(event) {
1380         const { volunteer } = this.state;
1381         const { address } = volunteer.user;
1382         address.neighborhood = event.target.value;
1383         volunteer.user.address = address;
1384         this.setState({ volunteer });
1385     }
1386
1387     handlerStreet(event) {
1388         const { volunteer } = this.state;
1389         const { address } = volunteer.user;
1390         address.street = event.target.value;
1391         volunteer.user.address = address;
1392         this.setState({ volunteer });
1393     }
1394
1395     handlerNumber(event) {
1396         const { volunteer } = this.state;
1397         const { address } = volunteer.user;
1398         address.number = event.target.value;
1399         volunteer.user.address = address;
1400         this.setState({ volunteer });
1401     }
1402
1403 //Question
1404 handlerAnswer(value, codQuestion){
1405     const { volunteer } = this.state;
1406     for (const answer of volunteer.answers) {
1407         if (answer.question.codQuestion === codQuestion){
1408             answer.answer = value;
1409             break;
1410         }
1411     }
1412     this.setState({ volunteer });
1413 }
1414
1415 handlerAnswerComment(event, codQuestion){
1416     const { volunteer } = this.state;
1417     for (const answer of volunteer.answers) {
1418         if (answer.question.codQuestion === codQuestion){
1419             answer.answer = event.target.value;
1420             break;
1421         }
1422     }
1423     this.setState({ volunteer });
1424 }
```

```
1424     }
1425
1426     handlerSubmit(){
1427         const { afterSubmit } = this.props;
1428         postRequest(
1429             '/volunteer',
1430             this.state.volunteer,
1431             () => postRequest('/auth', {}, afterSubmit)
1432         );
1433     }
1434
1435     render(){
1436         const { questions, companies, volunteer, confirmPassword
1437             } = this.state;
1438         const { user, company } = volunteer;
1439         const maxDate = new Date().toJSON().split('T')[0];
1440         return (
1441             <Row>
1442                 <Col/>
1443                 <Col>
1444                     <Wizard onCancel={()=> null}
1445                         submitLabel="cadastrar"
1446                         onSubmit={this.handlerSubmit}>
1447                         <div>
1448                             <h3>Dados do Voluntrio</h3>
1449                             <Input id="nameReponsible" label="Nome
1450                                 Completo" invalidMessage="Nome
1451                                 Completo obrigatorio"
1452                                 value={user.name}
1453                                 onChange={this.handlerName}
1454                                 required />
1455                         <Row>
1456                             <Col>
1457                                 <MaskInput id="cpf" label="CPF"
1458                                     invalidMessage="CPF
1459                                     obrigatorio"
1460                                     mask="999.999.999-99"
1461                                     value={user.cpf}
1462                                     onChange={this.handlerCpf}
1463                                     required />
1464                             </Col>
1465                             <Col>
1466                                 <MaskInput id="rg" label="RG"
1467                                     invalidMessage="RG
1468                                     obrigatorio"
```

```

        mask="99999999999"
        value={volunteer.rg}
        onChange={this.handlerRg}
        required />
      </Col>
    </Row>
    <MaskInput id="phoneResponsible"
      label="Nmero para Contato"
      invalidMessage="Nmero para Contato
      obrigatrio" value={user.phone}
      mask="(99)9999-9999"
      onChange={this.handlerPhone}
      required />
    <Input id="company" type="select"
      label="Empresa"
      invalidMessage="Empresa
      obrigatrio" value={company.cnpj}
      onChange={this.handlerCompany}
      required >
      <option value="">Seleciona</option>
      {
        companies.length ?
          companies.map(
            company => <option
              key={company.cnpj}
              value={company.cnpj}>{company.name}</option>
          ) : null
      }
    </Input>
    <Input id="job" label="Cargo na
      Empresa" invalidMessage="Cargo na
      Empresa obrigatrio"
      value={user.job}
      onChange={this.handlerJob}
      required />
  <Row>
    <Col>
      <Label
        for="radio1">Escolaridade
        <Required/></Label>
      {
        Schooling.values().map(
          schooling =>
            <Option
              name="radio1"

```

```

key={schooling}
value={schooling}
label={Schooling.translate(schooling)
)
)
}
)
}
</Col>
<Col>
<Row>
<Col>
<Switch id="complete"
label="Completo"
onChange={this.handlerComplete}
value={volunteer.conclusion}
checkedChildren="Sim"
unCheckedChildren="No"/>
</Col>
</Row>
{
    volunteer.schooling ===
        Schooling.UNIVERSITY_GRADUATE
    ||
    volunteer.schooling ===
        Schooling.POSTGRADUATE_STUDIES?
        (
<div>
    <br/>
    <Row>
        <Col>
            <Form inline>
                <Input
                    id="course"
                    label="Curso"
                    invalidMessage="Curso
obrigatorio"
                    value={volunteer.course}
                    onChange={this.handlerCourse}
                    required/>
            </Form>
        </Col>
    </Row>
</div>
):

```

```
1497           null
1498       }
1499     </Col>
1500   </Row>
1501   <DatePicker id="birth"
1502     label="Nascimento"
1503     invalidMessage="Nascimento
1504     obrigatorio" max={maxDate}
1505     onChange={this.handlerBirth}
1506     required />
1507   <Input id="email" type="email"
1508     label="Email"
1509     invalidMessage="Email  obrigatorio"
1510     value={user.email}
1511     onChange={this.handlerEmail}
1512     required />
1513 <Row>
1514   <Col>
1515     <Input id="password"
1516       type="password"
1517       label="Senha"
1518       invalidMessage="Senha
1519       obrigatorio"
1520       value={user.password}
1521       onChange={this.handlerPassword}
1522       required />
1523   </Col>
1524   <Col>
1525     <Input id="passwordConfirm"
1526       type="password"
1527       label="Confirmar Senha"
1528       invalidMessage="As senhas
1529       devem ser idnticas"
1530       value={confirmPassword}
1531       onChange={this.handlerConfirmPassword}
1532       required />
1533   </Col>
1534 </Row>
1535 </div>
1536 <div>
1537   <h3> Endero do Voluntrio </h3>
1538   <Input id="city" label="Cidade"
1539     invalidMessage="Cidade
1540     obrigatorio"
1541     onChange={this.handlerCity}
```

```

        required />
1515     <Input id="neighborhood"
          label="Bairro"
          invalidMessage="Bairro
          obrigatorio"
          onChange={this.handlerNeighborhood}
          required />
1516     <Row>
1517       <Col>
1518         <Input id="street"
              label="Lougradouro"
              invalidMessage="Lougradouro
              obrigatorio"
              onChange={this.handlerStreet}
              required />
1519       </Col>
1520       <Col md="3">
1521         <Input id="number"
              label="Nmero"
              onChange={this.handlerNumber}
              />
1522       </Col>
1523     </Row>
1524   </div>
1525   <div>
1526     {
1527       questions.map((question, index) => (
1528         <div key={question.codQuestion}>
1529           <Row >
1530             <Col>
1531               <strong>{question.question}</strong>
1532             </Col>
1533             <Col md="2">
1534               <Switch
                  id={'question-${question.codQuest}'
                  label="Resposta"
                  onChange={(value)
                  =>
                    this.handlerAnswer(value,
                    question.codQuestion)}/>
1535             </Col>
1536             <Col>
1537               <textarea
                  id={'details-${question.codQuesti
                  cols="30"
```

```
rows="3"
onChange={(event)=>
this.handlerAnswerComment(e
question.codQuestion)})/>
1538 </Col>
1539 </Row>
1540 { index !==
1541     question.length-1
1542     ?<hr/> : null}
1543         </div>
1544     )
1545 </div>
1546 </Wizard>
1547 </Col>
1548 <Col/>
1549 </Row>
1550 );
1551 }
1552
1553 webapp/src/pages/public/register/index.js
1554
1555 import Page from './Page.jsx';
1556 export default Page;
1557
1558 webapp/src/pages/public/login/Page.jsx
1559
1560 import React, { Component } from 'react';
1561 import { Redirect, Link } from 'react-router-dom';
1562 import { Button, Row, Col, Form } from 'reactstrap';
1563 import { Input } from '../../components';
1564 import { postRequest } from '../../utils/http';
1565
1566 export default class extends Component{
1567     constructor(){
1568         super();
1569         this.state={
1570             cpf: '',
1571             password: ''
1572         };
1573         this.handlerUsername = this.handlerUsername.bind(this);
1574         this.handlerPassword = this.handlerPassword.bind(this);
1575         this.handlerSubmit = this.handlerSubmit.bind(this);
1576     }
}
```

```
1577
1578     handlerUsername(event){
1579         this.setState({ cpf: event.target.value});
1580     }
1581
1582     handlerPassword(event){
1583         this.setState({password: event.target.value});
1584     }
1585
1586     handlerSubmit(){
1587         postRequest('/auth', this.state, res =>
1588             this.props.afterLogin(res.data.user));
1589     }
1590
1591     render(){
1592         const { logged } = this.props;
1593         const { cpf, password } = this.state;
1594         return !logged ? (
1595             <Row>
1596                 <Col/>
1597                 <Col md="4" sm="10">
1598                     <Row>
1599                         <Col>
1600                             <Form onKeyPress={({ key }) => key ===
1601                                 'Enter' && this.handlerSubmit()}>
1602                             <Input id="login" label="Usurio"
1603                                 value={cpf}
1604                                 onChange={this.handlerUsername}/>
1605                             <Input id="password" label="Senha"
1606                                 type="password"
1607                                 value={password}
1608                                 onChange={this.handlerPassword}/>
1609                             <Button type="button"
1610                                 color="primary" size="lg"
1611                                 onClick={this.handlerSubmit}
1612                                 block>Entrar</Button>
1613                         </Form>
1614                     </Col>
1615                 </Row>
1616             <Row>
1617                 <Col className="text-center">
1618                     <Link to="cadastrar">Quero ser
1619                         voluntario</Link>
1620                 </Col>
1621             </Row>
```

```
1611             </Col>
1612             <Col/>
1613         </Row>
1614     ) : (
1615         <Redirect to="/dashboard"/>
1616     );
1617 }
1618 }
1619
1620 webapp/src/pages/public/login/index.js
1621
1622 import Page from './Page.jsx';
1623 export default Page;
1624
1625 webapp/src/pages/public/index.js
1626
1627 export {default as Login} from './login';
1628 export {default as Register} from './register';
1629
1630 webapp/src/pages/protected/admin/attachment/Form.jsx
1631
1632 import React, { Component } from 'react';
1633 import { Attachment, Status } from '../../../../../model';
1634 import PropTypes from 'prop-types';
1635 import { postRequest } from '../../../../../utils/http';
1636 import { Row, Col, Form, Button, Card,CardBody } from
1637     'reactstrap';
1638 import { Input, FileInput, Switch } from
1639     '../../../../../components';
1640
1641 export default class extends Component{
1642     constructor(){
1643         super();
1644         this.state= {
1645             attachment: new Attachment(),
1646             file: null
1647         };
1648         this.handlerAttachment =
1649             this.handlerAttachment.bind(this);
1650         this.handlerRequired = this.handlerRequired.bind(this);
1651         this.handlerDownload = this.handlerDownload.bind(this);
1652         this.handlerStatus = this.handlerStatus.bind(this);
1653         this.handlerFile = this.handlerFile.bind(this);
1654         this.handlerSubmit = this.handlerSubmit.bind(this);
1655     }
1656 }
```

```
1653
1654     static propTypes = {
1655         afterSubmit: PropTypes.func.isRequired
1656     }
1657
1658     handlerAttachment(event){
1659         const { attachment } = this.state;
1660         attachment.name = event.target.value;
1661         this.setState({ attachment });
1662     }
1663
1664     handlerRequired(value){
1665         const { attachment } = this.state;
1666         attachment.required = value;
1667         this.setState({ attachment });
1668     }
1669
1670     handlerDownload(value){
1671         const { attachment } = this.state;
1672         attachment.download = value;
1673         this.setState({ attachment });
1674     }
1675
1676     handlerStatus(event){
1677         const { attachment } = this.state;
1678         attachment.status = event.target.value;
1679         this.setState({ attachment });
1680     }
1681
1682     handlerFile(event){
1683         this.setState({ file: event.target.files[0] });
1684     }
1685
1686     handlerSubmit(){
1687         const { afterSubmit } = this.props;
1688         const { attachment, file } = this.state;
1689         const json = JSON.stringify(attachment);
1690         const blob = new Blob([json], {
1691             type: 'application/json'
1692         });
1693         let form = new FormData();
1694         form.append('attachment', blob);
1695         if(file){
1696             form.append('file', file);
1697         }
1698     }
1699
1700     render() {
1701         return (
1702             <Formik</pre>
```

```
1698     postRequest('/attachment', form, afterSubmit)
1699 }
1700
1701 render(){
1702     const { attachment } = this.state;
1703     return (
1704         <Card>
1705             <CardBody>
1706                 <Form>
1707                     <Input id="name" label="Anexo"
1708                         onChange={this.handlerAttachment}
1709                         invalidMessage="Anexo Obrigatorio"
1710                         required/>
1711                     <Row>
1712                         <Col md="5">
1713                             <Switch id="required" label="Envio
1714                                 Obrigatorio "
1715                                 onChange={this.handlerRequired}/>
1716                         </Col>
1717                         <Col>
1718                             <Switch id="download"
1719                                 label="Disponibilizado Pela CnE
1720                                 "
1721                                 onChange={this.handlerDownload}/>
1722                         </Col>
1723                     </Row>
1724             {
1725                 attachment.download && <FileInput
1726                     color="info" label="Upload"
1727                     onChange={this.handlerFile}
1728                     accept="application/pdf" required/>
1729             }
1730             <Input id="status" type="select"
1731                 label="Enviado Durante"
1732                 onChange={this.handlerStatus}
1733                 invalidMessage="Enviado Durante
1734                     Obrigatorio" required>
1735                 <option value="">Seleccione</option>
1736             {
1737                 Status.values().map(status =>
1738                     <option key={status}
1739                     value={status}>{Status.translate(status)}
```

```
1725             <Button type="button" color="primary"
1726                 className="float-right"
1727                 onClick={this.handlerSubmit}>Cadastrar</Button>
1728         </Form>
1729     </CardBody>
1730 </Card>
1731 );
1732 }
1733
1734 webapp/src/pages/protected/admin/attachment/index.js
1735
1736 import List from './List';
1737 export default List;
1738
1739 webapp/src/pages/protected/admin/attachment/ListItem.jsx
1740
1741 import React, { Component, Fragment } from 'react';
1742 import PropTypes from 'prop-types';
1743 import { deleteRequest } from '../../../../../utils/http';
1744 import { FaTrash } from 'react-icons/fa';
1745 import Details from './Details.jsx';
1746
1747 export default class extends Component{
1748     constructor(){
1749         super();
1750         this.state = {
1751             isOpen: false
1752         }
1753         this.handlerDelete = this.handlerDelete.bind(this);
1754         this.handlerDetails = this.handlerDetails.bind(this);
1755     }
1756
1757     static propTypes = {
1758         attachment: PropTypes.object.isRequired,
1759         afterDelete: PropTypes.func.isRequired
1760     }
1761
1762     handlerDelete(){
1763         const { attachment, afterDelete } = this.props;
1764         deleteRequest(
1765             '/attachment/${attachment.codAttachment}',
1766             afterDelete
1767         )
1768     }
1769 }
```

```

1768
1769     handlerDetails(){
1770         const { isOpen } = this.state;
1771         this.setState({ isOpen: !isOpen });
1772     }
1773
1774     render(){
1775         const { attachment } = this.props;
1776         const { isOpen } = this.state;
1777         return (
1778             <Fragment>
1779                 <tr>
1780                     <td onClick={this.handlerDetails} style={{ cursor: 'pointer' }}>
1781                         <strong>
1782                             {attachment.name}
1783                         </strong>
1784                     </td>
1785                     <td onClick={this.handlerDetails} style={{ cursor: 'pointer' }}>
1786                         <strong>
1787                             {attachment.required ? "Sim" : "No"}
1788                         </strong>
1789                     </td>
1790                     <td>
1791                         <FaTrash style={{ cursor: 'pointer' }}>
1792                             color="red"
1793                             onClick={this.handlerDelete} />
1794                     </td>
1795                 </tr>
1796             {
1797                 isOpen ? <Details attachment={attachment}
1798                         isOpen={isOpen}
1799                         close={this.handlerDetails}/> : null
1800             }
1801         );
1802     }
1803
1804     import React from 'react';
1805     import { Modal, ModalBody, ModalHeader, ModalFooter, Row, Col,
1806             Button } from 'reactstrap';

```

```
1806 import { Status } from '../../../../../model';
1807
1808 export default ({ close, attachment, isOpen }) => attachment && (
1809     <Modal toggle={close} isOpen={isOpen}>
1810         <ModalHeader>
1811             <strong>{attachment.name}</ModalHeader>
1812         <ModalBody>
1813             <Row>
1814                 <Col>
1815                     <strong>
1816                         Obrigatrio:
1817                     </strong>
1818                 </Col>
1819                 <Col>
1820                     {attachment.required ? 'Sim': 'No'}
1821                 </Col>
1822             </Row>
1823             <Row>
1824                 <Col>
1825                     <strong>
1826                         Disponibilizado pela CnE:
1827                     </strong>
1828                 </Col>
1829                 <Col>
1830                     {attachment.download ? 'Sim': 'No'}
1831                 </Col>
1832             </Row>
1833             <Row>
1834                 <Col>
1835                     <strong>
1836                         Necessrio envio durante:
1837                     </strong>
1838                 </Col>
1839                 <Col>
1840                     {Status.translate(attachment.status)}
1841                 </Col>
1842             </Row>
1843         <ModalFooter>
1844             <Button color="default" type="button"
1845                 onClick={close}>Fechar</Button>
1846         </ModalFooter>
1847     </Modal>
1848 );
```

```
1849 webapp/src/pages/protected/admin/attachment>List.jsx
1850
1851 import React, { Component } from 'react';
1852 import { getRequest } from '../../../../../utils/http';
1853 import { Row, Col, Table } from 'reactstrap';
1854 import ListItem from './ListItem.jsx';
1855 import Form from './Form.jsx';
1856 import { Filter, Pagination } from '../../../../../components';
1857
1858 export default class extends Component{
1859     constructor(){
1860         super();
1861         this.state = {
1862             attachments: [],
1863             filter: '',
1864             count: 0,
1865             pages: 0
1866         };
1867     }
1868
1869     componentWillMount(){
1870         getRequest(
1871             '/attachment/page/0',
1872             res => {
1873                 const { data: attachments, count } = res.data
1874                 this.setState({
1875                     pages: count / 20,
1876                     attachments,
1877                     count
1878                 })
1879             }
1880         );
1881     }
1882
1883     handlerFilter(event){
1884         const filter = event.target.value;
1885         if (filter.length >= 3 || filter.length === 0){
1886             getRequest(
1887                 '/attachment/pages/0?filter=${filter}',
1888                 res => {
1889                     const { data: attachments, count} = res.data
1890                     this.setState({
1891                         pages: count/20,
1892                         attachments,
1893                         count,
```

```
1894                     filter
1895                 })
1896             }
1897         );
1898     } else {
1899         this.setState({ filter })
1900     }
1901 }
1902
1903 onChangePage(pageNumber) {
1904     const { filter } = this.state;
1905     getRequest(
1906         `/attachment/page/${pageNumber}?filter=${filter.length
1907             > 3 ? filter : ''}`,
1908         res => {
1909             const { data: attachments, count } = res.data;
1910             this.setState({
1911                 pages: count / 20,
1912                 attachments,
1913                 count
1914             });
1915         }
1916     );
1917 }
1918
1919 render(){
1920     const { attachments, filter, pages, count } = this.state;
1921     return (
1922         <div>
1923             <Row>
1924                 <Col>
1925                     <h3>Anexos</h3>
1926                 </Col>
1927             </Row>
1928             <br/>
1929             <Row>
1930                 <Col>
1931                     <Table striped>
1932                         <thead>
1933                             <tr>
1934                                 <th>
1935                                     <Filter
1936                                         label="Nome &nbsp;&nbsp;">
1937                                         value={filter}>
1938                                         handlerFilter={this.handlerFilter.bind(t
```

```

                />
        </th>
        <th>Obrigatrio</th>
        <th></th>
    </tr>
</thead>
<tbody>
{
    attachments.length ?
        attachments.map(
            attachment => (
                <ListItem
                    key={attachment.codAttachment}
                    attachment={attachment}
                    afterDelete={this.component
)
) : (
<tr>
    <td colSpan="3">
        <strong>
            Nenhum
            Registro
            encontrado
        </strong>
    </td>
</tr>
)
}
</tbody>
</Table>
<Pagination pages={pages} count={count}
    onChangePage={this.onChangePage.bind(this)}
    />
</Col>
<Col md="4">
    <Form
        afterSubmit={this.componentWillMount.bind(this)}
    </Col>
</Row>
</div>
);
}
}

```

```
1971 webapp/src/pages/protected/admin/question/Form.jsx
1972
1973 import React, { Component } from 'react';
1974 import { Question, Roles } from '../../../../../model';
1975 import PropTypes from 'prop-types';
1976 import { postRequest } from '../../../../../utils/http';
1977 import { Form, Button, Card,CardBody } from 'reactstrap';
1978 import { Input } from '../../../../../components';
1979
1980 export default class extends Component{
1981     constructor() {
1982         super()
1983         this.state = {
1984             newQuestion: new Question(),
1985             invalid: false
1986         }
1987         this.handlerQuestion = this.handlerQuestion.bind(this);
1988         this.handlerResponds = this.handlerResponds.bind(this);
1989         this.handlerSubmit = this.handlerSubmit.bind(this);
1990     }
1991
1992     static propTypes = {
1993         afterSubmit: PropTypes.func.isRequired
1994     }
1995
1996     handlerQuestion(event) {
1997         const { newQuestion } = this.state;
1998         newQuestion.question = event.target.value;
1999         this.setState({ newQuestion, invalid: false });
2000     }
2001
2002     handlerResponds(event) {
2003         const { newQuestion } = this.state;
2004         newQuestion.responds = event.target.value;
2005         this.setState({ newQuestion, invalid: false });
2006     }
2007
2008     handlerSubmit() {
2009         const { newQuestion } = this.state;
2010         if (!newQuestion.question || !newQuestion.responds) {
2011             this.setState({ invalid: true })
2012             return;
2013         }
2014         const { afterSubmit } = this.props;
2015         postRequest(
```

```

2016         '/question',
2017         newQuestion,
2018         () =>{
2019             afterSubmit();
2020             this.setState({ newQuestion: new Question()});
2021         }
2022     );
2023 }
2024
2025 render(){
2026     const { newQuestion } = this.state;
2027     return (
2028         <Card>
2029             <CardBody>
2030                 <Form>
2031                     <Input id="newQuestion" label="Questo"
2032                         invalidMessage="Campo Obrigatrio"
2033                         value={newQuestion.question}
2034                         invalid={this.state.invalid}
2035                         onChange={this.handlerQuestion}
2036                         required />
2037                     <Input id="destiny" type="select"
2038                         label="Destinada"
2039                         invalidMessage="Campo Obrigatrio"
2040                         value={newQuestion.responds}
2041                         invalid={this.state.invalid}
2042                         onChange={this.handlerResponds}
2043                         required>
2044                         <option>Seleciona</option>
2045                         <option
2046                             value={Roles.VOLUNTEER}>{Roles.translate(Role
2047                             <option
2048                                 value={Roles.COMPANY_ADMIN}>{Roles.translate(
2049                                     </Input>
2050                                     <Button type="button" color="primary"
2051                                         className="float-right"
2052                                         onClick={this.handlerSubmit}>Cadastrar</Button>
2053                                     </Form>
2054                                 </CardBody>
2055                             </Card>
2056                         );
2057                     }
2058                 }
2059             )
2060         );
2061     }
2062 }
2063
2064 webapp/src/pages/protected/admin/question/index.js

```

```
2046
2047 import List from './List';
2048 export default List;
2049
2050 webapp/src/pages/protected/admin/question>List.jsx
2051
2052 import React, { Component, Fragment } from 'react';
2053 import { Question, Roles } from '../../../../../model';
2054 import { getRequest, deleteRequest } from
    '../../../../../utils/http';
2055 import { Row, Col, Table} from 'reactstrap';
2056 import { FaTrash } from 'react-icons/fa';
2057 import Form from './Form.jsx';
2058
2059 export default class extends Component{
2060     constructor(){
2061         super()
2062         this.state = {
2063             questions: []
2064         }
2065         this.handlerDelete = this.handlerDelete.bind(this);
2066     }
2067
2068     componentWillMount(){
2069         getRequest('/question', res => this.setState({
2070             questions: res.data }));
2071     }
2072
2073     handlerDelete(codQuestion){
2074         deleteRequest(`/question/${codQuestion}`,
2075             this.componentWillMount);
2076     }
2077
2078     render(){
2079         const { questions } = this.state;
2080         return (
2081             <Fragment>
2082                 <Row>
2083                     <Col>
2084                         <h3> Questes </h3>
2085                     </Col>
2086                 </Row>
2087                 <br/>
2088                 <Row>
2089                     <Col>
```

```

2088 <Table striped>
2089   <thead>
2090     <tr>
2091       <td>
2092         <strong>Questo</strong>
2093       </td>
2094       <td>
2095         <strong>Destinada</strong>
2096       </td>
2097       <td/>
2098     </tr>
2099   </thead>
2100   <tbody>
2101   {
2102     questions.length ?
2103     questions.map(
2104       question => (
2105         <tr>
2106           key={question.codQuestion}
2107           <td>
2108             <strong>{question.questi
2109           </td>
2110           <td>
2111             <strong>{Roles.translate(
2112               </td>
2113               <td>
2114                 <FaTrash
2115                   style={{cursor:
2116                     'pointer'}}
2117                   color="red"
2118                   onClick={() =>
2119                     this.handlerDelete(c
2120                   )
2121               )
2122             : (
2123               <tr>
2124                 <td colSpan="3">
2125                   Nenhuma Questo
2126                   cadastrada
2127                   Cadastrada
2128                 </td>
2129             </tr>
2130           )
2131         )
2132       )
2133     )
2134   )
2135 )

```

```

2124                     }
2125                 </tbody>
2126             </Table>
2127         </Col>
2128     <Col md="4">
2129         <Form
2130             afterSubmit={this.componentWillMount.bind(this)}
2131             />
2132         </Col>
2133     </Row>
2134   </Fragment>
2135 )
2136 }
2137
2138 webapp/src/pages/protected/admin/project/Form.jsx
2139
2140 import React, { Component } from 'react';
2141 import { Project } from '../../../../../model';
2142 import PropTypes from 'prop-types';
2143 import { postRequest } from '../../../../../utils/http';
2144 import { Form, Button, Card,CardBody } from 'reactstrap';
2145 import { Input, DatePicker } from '../../../../../components';
2146
2147 export default class extends Component{
2148     constructor(){
2149         super();
2150         this.state = {
2151             project: new Project(),
2152             invalid: false
2153         };
2154         this.handlerName = this.handlerName.bind(this);
2155         this.handlerStart = this.handlerStart.bind(this);
2156         this.handlerEnd = this.handlerEnd.bind(this);
2157         this.handlerSubmit = this.handlerSubmit.bind(this);
2158     }
2159
2160     static propTypes = {
2161         afterSubmit: PropTypes.func.isRequired
2162     }
2163
2164     handlerName(event) {
2165         const { project } = this.state;
2166         project.name = event.target.value;

```



```

        invalidMessage="Nome do Projeto
        Obrigatorio" value={project.name}
        invalid={invalid} required />
2210      <DatePicker id="start" label="Incio do
        Projeto" invalidMessage="Incio do
        Projeto obrigatorio"
        min={minDateStart}
        onChange={this.handlerStart}
        invalid={invalid} required />
2211      <DatePicker id="end" label="Final do
        Projeto" invalidMessage="Final do
        Projeto obrigatorio" min={minDateEnd}
        onChange={this.handlerEnd}
        invalid={invalid} required />
2212      <Button type="button" color="primary"
        className="float-right"
        onClick={this.handlerSubmit}>Cadastrar</Button>
2213    </Form>
2214  </CardBody>
2215 </Card>
2216  );
2217 }
2218 }
2219
2220 webapp/src/pages/protected/admin/project/index.js
2221
2222 import List from './List';
2223 export default List;
2224
2225 webapp/src/pages/protected/admin/project/ListItem.jsx
2226
2227 import React, { Component, Fragment } from 'react';
2228 import PropTypes from 'prop-types';
2229 import { deleteRequest } from '../../../../../utils/http';
2230 import { FaTrash } from 'react-icons/fa';
2231
2232 export default class extends Component {
2233
2234   static propTypes = {
2235     typeEvent: PropTypes.object.isRequired,
2236     afterDelete: PropTypes.func.isRequired
2237   }
2238
2239   handlerDelete() {
2240     const { project, afterDelete } = this.props;

```

```

2241     deleteRequest(
2242         '/project/${project.codProject}',
2243         afterDelete
2244     )
2245 }
2246
2247 render() {
2248     const { project } = this.props;
2249     return (
2250         <tr>
2251             <td>
2252                 <strong>
2253                     {project.name}
2254                 </strong>
2255             </td>
2256             <td>
2257                 <strong>
2258                     {new
2259                         Date(project.start).toLocaleDateString())
2260                     </strong>
2261             </td>
2262             <td>
2263                 <strong>
2264                     {new
2265                         Date(project.end).toLocaleDateString())
2266                     </strong>
2267             <td>
2268                 <FaTrash style={{ cursor: 'pointer' }}>
2269                     color="red"
2270                     onClick={this.handlerDelete.bind(this)} />
2271             </td>
2272         </tr>
2273     );
2274 }
2275
2276 webapp/src/pages/protected/admin/project/List.jsx
2277
2278 import React, { Component, Fragment } from 'react';
2279 import { getRequest } from '../../../../../utils/http';
2280 import { Row, Col, Table } from 'reactstrap';
2281 import ListItem from './ListItem.jsx';
2282 import Form from './Form.jsx';
2283 import { Filter, Pagination } from '../../../../../components';

```

```
2282
2283 export default class extends Component{
2284     constructor(){
2285         super();
2286         this.state ={
2287             projects: [],
2288             filter: '',
2289             count: 0,
2290             pages: 0
2291         };
2292     }
2293
2294     componentWillMount(){
2295         getRequest(
2296             '/project/page/0',
2297             res => {
2298                 const { data: projects, count } = res.data
2299                 this.setState({
2300                     pages: count / 20,
2301                     projects,
2302                     count
2303                 });
2304             }
2305         );
2306     }
2307
2308     handlerFilter(event) {
2309         const filter = event.target.value;
2310         if (filter.length >= 3 || filter.length === 0) {
2311             getRequest(
2312                 '/project/page/0?filter=${filter}',
2313                 res => {
2314                     const { data: projects, count } = res.data
2315                     this.setState({
2316                         pages: count / 20,
2317                         projects,
2318                         count,
2319                         filter
2320                     })
2321                 }
2322             );
2323         } else {
2324             this.setState({ filter })
2325         }
2326     }
}
```

```
2327
2328     onChangePage(pageNumber) {
2329         const { filter } = this.state;
2330         getRequest(
2331             `/project/page/${pageNumber}?filter=${filter.length} >
2332             3 ? filter : ''}`,
2333             res => {
2334                 const { data: projects, count } = res.data;
2335                 this.setState({
2336                     pages: count / 20,
2337                     projects,
2338                     count
2339                 });
2340             );
2341         }
2342
2343         render(){
2344             const { projects, filter, count, pages } = this.state;
2345             return (
2346                 <Fragment>
2347                     <Row>
2348                         <Col>
2349                             <h3>Projetos</h3>
2350                         </Col>
2351                     </Row>
2352                     <br/>
2353                     <Row>
2354                         <Col>
2355                             <Table striped>
2356                                 <thead>
2357                                     <tr>
2358                                         <th>
2359                                             <Filter
2360                                                 label="Nome &nbsp;&nbsp;">
2361                                                 value={filter}
2362                                                 handlerFilter={this.handlerFilter.
2363                                         />
2364                                         </th>
2365                                         <th> Incio </th>
2366                                         <th>Fim</th>
2367                                         <th></th>
2368                                         </th>
2369                                     </tr>
2370                                 </thead>
2371                                 <tbody>
```



```
2401         invalid: false
2402     };
2403     this.handlerSkill = this.handlerSkill.bind(this);
2404     this.handlerSubmit = this.handlerSubmit.bind(this);
2405 }
2406
2407 handlerSkill(event){
2408     this.setState({ value: event.target.value, invalid:
2409         false });
2410 }
2411
2412 handlerSubmit(){
2413     const { value } = this.state;
2414     if(!value){
2415         this.setState({invalid: true})
2416     }
2417     const { afterSubmit } = this.props;
2418     postRequest(
2419         '/skill',
2420         { name: value },
2421         () => {
2422             this.setState({value: ''});
2423             afterSubmit();
2424         }
2425     );
2426 }
2427
2428 render(){
2429     return (
2430         <Card>
2431             <CardBody>
2432                 <Form>
2433                     <Input id="skill" label="Competencia"
2434                         {...this.state} invalidMessage="Campo
2435                         Obrigatorio"
2436                         onChange={this.handlerSkill} required
2437                         />
2438                     <Button type="button" color="primary"
2439                         className="float-right"
2440                         onClick={this.handlerSubmit}>Cadastrar</Button>
2441                 </Form>
2442             </CardBody>
2443         </Card>
2444     )
2445 }
```

```
2439     }
2440
2441     webapp/src/pages/protected/admin/evaluate/skill/index.js
2442
2443     import List from './List';
2444     export default List;
2445
2446     webapp/src/pages/protected/admin/evaluate/skill/List.jsx
2447
2448     import React, { Component, Fragment } from 'react';
2449     import { getRequest, deleteRequest } from
2450         '../../../../../utils/http';
2451     import { Col, Row, ListGroup, ListGroupItem } from 'reactstrap';
2452     import { FaTrash } from 'react-icons/fa';
2453     import Form from './Form.jsx';
2454
2455     export default class extends Component{
2456         constructor(){
2457             super();
2458             this.state = {
2459                 skills: []
2460             };
2461             this.handlerDelete = this.handlerDelete.bind(this);
2462             this.componentWillMount =
2463                 this.componentWillMount.bind(this);
2464         }
2465
2466         componentWillMount(){
2467             getRequest('/skill', res => this.setState({ skills:
2468                 res.data }));
2469         }
2470
2471         handlerDelete(codSkill){
2472             deleteRequest(`/skill/${codSkill}`,
2473                 this.componentWillMount);
2474         }
2475
2476         render(){
2477             const { skills } = this.state;
2478             return (
2479                 <Fragment>
2480                     <Row>
2481                         <Col>
2482                             <h3>Competencias</h3>
2483                         </Col>
```

```
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
</Row>
<br/>
<Row>
    <Col>
        <ListGroup>
            {
                skills.length ?
                    skills.map(
                        skill =>
                            (
                                <ListGroupItem
                                    key={skill.codSkill}>
                                    <Row>
                                        <Col>
                                            <strong>{skill.name}</strong>
                                        </Col>
                                        <Col md="1">
                                            <FaTrash
                                                style={{{
                                                    cursor:
                                                    'pointer'
                                                }}}
                                                color="red"
                                                onClick={() =>
                                                    this.handlerDelete
                                                }
                                            </Col>
                                        </Row>
                                    </ListGroupItem>
                                )
                            )
            ) : (
                <ListGroupItem>
                    <strong>Nenhuma questo
                        sobre a Competncia
                        encontrada</strong>
                </ListGroupItem>
            )
        }
    <ListGroup>
    </Col>
    <Col md="4">
        <Form
            afterSubmit={this.componentWillMount}/>
    </Col>
```

```
2513             </Row>
2514         </Fragment>
2515     );
2516 }
2517 }
2518
2519 webapp/src/pages/protected/admin/evaluate/technology/Form.jsx
2520
2521 import React, { Component } from 'react';
2522 import { Card, CardBody, Form, Button } from 'reactstrap';
2523 import { Input } from '../../../../../components';
2524 import { postRequest } from '../../../../../utils/http';
2525
2526 export default class extends Component{
2527     constructor(){
2528         super();
2529         this.state = {
2530             value : '',
2531             invalid: false
2532         };
2533         this.handlerSkill = this.handlerSkill.bind(this);
2534         this.handlerSubmit = this.handlerSubmit.bind(this);
2535     }
2536
2537     handlerSkill(event){
2538         this.setState({ value: event.target.value, invalid:
2539             false });
2540     }
2541
2542     handlerSubmit(){
2543         const { value } = this.state;
2544         if(!value){
2545             this.setState({invalid: true})
2546         }
2547         const { afterSubmit } = this.props;
2548         postRequest(
2549             '/technology',
2550             { name: value },
2551             () => {
2552                 this.setState({value: ''});
2553                 afterSubmit();
2554             }
2555         );
2556     }
2557 }
```

```
2557     render(){
2558         return (
2559             <Card>
2560                 <CardBody>
2561                     <Form>
2562                         <Input id="skill" label="Conhecimento"
2563                             {...this.state} invalidMessage="Campo
2564                             Obrigatorio"
2565                             onChange={this.handlerSkill} required
2566                             />
2567                         <Button type="button" color="primary"
2568                             className="float-right"
2569                             onClick={this.handlerSubmit}>Cadastrar</Button>
2570                     </Form>
2571                 </CardBody>
2572             </Card>
2573         )
2574     }
2575 }
2576
2577 webapp/src/pages/protected/admin/evaluate/technology/index.js
2578
2579 import List from './List';
2580 export default List;
2581
2582 webapp/src/pages/protected/admin/evaluate/technology/List.jsx
2583
2584 import React, { Component, Fragment } from 'react';
2585 import { getRequest, deleteRequest } from
2586     '../../../../../utils/http';
2587 import { Col, Row, ListGroup, ListGroupItem } from 'reactstrap';
2588 import { FaTrash } from 'react-icons/fa';
2589 import Form from './Form.jsx';
2590
2591 export default class extends Component{
2592     constructor(){
2593         super();
2594         this.state = {
2595             technologies: []
2596         };
2597         this.handlerDelete = this.handlerDelete.bind(this);
2598         this.componentWillMount =
2599             this.componentWillMount.bind(this);
2600     }
2601 }
```



```
2628             </Row>
2629         </ListGroupItem>
2630     )
2631   ) : (
2632     <ListGroupItem>
2633       <strong>Nenhum tipo de
2634         Conhecimento
2635         encontrado</strong>
2636     </ListGroupItem>
2637   )
2638   </ListGroup>
2639 </Col>
2640 <Col md="4">
2641   <Form
2642     afterSubmit={this.componentWillMount}/>
2643   </Col>
2644 </Row>
2645 </Fragment>
2646 );
2647 }
2648
2649 webapp/src/pages/protected/admin/evaluate/personality/Form.jsx
2650
2651 import React, { Component } from 'react';
2652 import { Card, CardBody, Form, Button } from 'reactstrap';
2653 import { Input } from '../../../../../components';
2654 import { postRequest } from '../../../../../utils/http';
2655
2656 export default class extends Component{
2657   constructor(){
2658     super();
2659     this.state = {
2660       value : '',
2661       invalid: false
2662     };
2663     this.handlerQuestion = this.handlerQuestion.bind(this);
2664     this.handlerSubmit = this.handlerSubmit.bind(this);
2665   }
2666
2667   handlerQuestion(event){
2668     this.setState({ value: event.target.value, invalid:
2669       false });
2670   }
2671 }
```

```
2669     }
2670
2671     handlerSubmit(){
2672         const { value } = this.state;
2673         if(!value){
2674             this.setState({invalid: true})
2675         }
2676         const { afterSubmit } = this.props;
2677         postRequest(
2678             '/personality',
2679             { question: value },
2680             () => {
2681                 this.setState({value: ''});
2682                 afterSubmit();
2683             }
2684         );
2685     }
2686
2687     render(){
2688         return (
2689             <Card>
2690                 <CardBody>
2691                     <Form>
2692                         <Input id="personality" label="Questo"
2693                             {...this.state} invalidMessage="Campo
2694                             Obrigatório"
2695                             onChange={this.handlerQuestion}
2696                             required />
2697                         <Button type="button" color="primary"
2698                             className="float-right"
2699                             onClick={this.handlerSubmit}>Cadastrar</Button>
2700                     </Form>
2701                 </CardBody>
2702             </Card>
2703         )
2704     }
2705
2706     webapp/src/pages/protected/admin/evaluate/personality/index.js
2707
2708     import List from './List';
2709     export default List;
2710
2711     webapp/src/pages/protected/admin/evaluate/personality/List.jsx
2712
```

```
2708 import React, { Component, Fragment } from 'react';
2709 import { getRequest, deleteRequest } from
2710   '../../../../../utils/http';
2711 import { Col, Row, ListGroup, ListGroupItem } from 'reactstrap';
2712 import { FaTrash } from 'react-icons/fa';
2713 import Form from './Form.jsx';
2714
2715 export default class extends Component{
2716   constructor(){
2717     super();
2718     this.state = {
2719       personalities: []
2720     };
2721     this.handlerDelete = this.handlerDelete.bind(this);
2722     this.componentWillMount =
2723       this.componentWillMount.bind(this);
2724   }
2725
2726   componentWillMount(){
2727     getRequest('/personality', res => this.setState({
2728       personalities: res.data }));
2729   }
2730
2731   handlerDelete(codQuestion){
2732     deleteRequest(`/personality/${codQuestion}`,
2733       this.componentWillMount);
2734   }
2735
2736   render(){
2737     const { personalities } = this.state;
2738     return (
2739       <Fragment>
2740         <Row>
2741           <Col>
2742             <h3>Personalidade</h3>
2743           </Col>
2744         </Row>
2745         <br/>
2746         <Row>
2747           <Col>
2748             <ListGroup>
2749               {
2750                 personalities.length ?
2751                   personalities.map(
2752                     personality =>
```

```

2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
        (
            <ListGroupItem
                key={personality.codQuestion}>
                <Row>
                    <Col>
                        <strong>{personality.quest
                    </Col>
                    <Col md="1">
                        <FaTrash
                            style={{cursor:
                                'pointer'}}
                            color="red"
                            onClick={() =>
                                this.handlerDelete(per
                            )
                        }
                    </Col>
                </Row>
            </ListGroupItem>
        )
    ) : (
        <ListGroupItem>
            <strong>Nenhuma questo
                sobre a
                personalidade do
                voluntario
                encontrada</strong>
        </ListGroupItem>
    )
}

        }
    </ListGroup>
</Col>
<Col md="4">
    <Form
        afterSubmit={this.componentWillMount}/>
    </Col>
</Row>
</Fragment>
);
}
}
}

webapp/src/pages/protected/admin/evaluate/task/Form.jsx
import React, { Component } from 'react';

```



```

                    required />
2823             <Button type="button" color="primary"
2824                 className="float-right"
2825                 onClick={this.handlerSubmit}>Cadastrar</Button>
2826         </Form>
2827     </CardBody>
2828   </Card>
2829 }
2830 }
2831 webapp/src/pages/protected/admin/evaluate/task/index.js
2832
2833 import List from './List';
2834 export default List;
2835
2836 webapp/src/pages/protected/admin/evaluate/task/List.jsx
2837
2838 import React, { Component, Fragment } from 'react';
2839 import { getRequest, deleteRequest } from
2840     '../../../../../utils/http';
2841 import { Col, Row,ListGroup, ListGroupItem } from 'reactstrap';
2842 import { FaTrash } from 'react-icons/fa';
2843 import Form from './Form.jsx';
2844
2845 export default class extends Component{
2846     constructor(){
2847         super();
2848         this.state = {
2849             tasks: []
2850         };
2851         this.handlerDelete = this.handlerDelete.bind(this);
2852         this.componentWillMount =
2853             this.componentWillMount.bind(this);
2854     }
2855
2856     componentWillMount(){
2857         getRequest('/task', res => this.setState({
2858             personalities: res.data }));
2859     }
2860
2861     handlerDelete(codTask){
2862         deleteRequest(`/task/${codTask}`,
2863             this.componentWillMount);
2864     }

```

```
2861
2862     render(){
2863         const { tasks } = this.state;
2864         return (
2865             <Fragment>
2866                 <Row>
2867                     <Col>
2868                         <h3>Feedback do Voluntrio</h3>
2869                     </Col>
2870                 </Row>
2871                 <br/>
2872                 <Row>
2873                     <Col>
2874                         <ListGroup>
2875                             {
2876                                 tasks.length ?
2877                                     tasks.map(
2878                                         task =>
2879                                         (
2880                                             <ListGroupItem
2881                                                 key={task.codTask}>
2882                                                 <Row>
2883                                                     <Col>
2884                                                         <strong>{task.questi
2885                                                     </Col>
2886                                                     <Col md="1">
2887                                                         <FaTrash
2888                                                 style={{'
2889                                                 cursor:
2890                                                 'pointer'
2891                                                 }}}
2892                                                 color="red"
2893                                                 onClick={() =>
2894                                                     this.handlerDelete(
2895                                                         task)
2896                                                 }
2897                                         )
2898                                         ) : (
2899                                             <ListGroupItem>
2900                                                 <strong>Nenhuma questo
2901                                                 sobre a
2902                                                 personalidade do
2903                                                 voluntario
2904                                         )
2905                                     )
2906                                 )
2907                             )
2908                         )
2909                     )
2910                 )
2911             )
2912         )
2913     )
2914 }
```

```

                                encontrada</strong>
2894                         </ListGroupItem>
2895                     )
2896                 }
2897             }
2898         </ListGroup>
2899     <Col>
2900     <Col md="4">
2901         <Form
2902             afterSubmit={this.componentWillMount}/>
2903         </Col>
2904     </Row>
2905   </Fragment>
2906 );
2907 }
2908
2909 webapp/src/pages/protected/admin/evaluate/mentoring/Form.jsx
2910
2911 import React, { Component } from 'react';
2912 import { Card, CardBody, Form, Button } from 'reactstrap';
2913 import { Input } from '../../../../../components';
2914 import { postRequest } from '../../../../../utils/http';
2915
2916 export default class extends Component{
2917     constructor(){
2918         super();
2919         this.state = {
2920             value : '',
2921             invalid: false
2922         };
2923         this.handlerQuestion = this.handlerQuestion.bind(this);
2924         this.handlerSubmit = this.handlerSubmit.bind(this);
2925     }
2926
2927     handlerQuestion(event){
2928         this.setState({ value: event.target.value, invalid:
2929             false });
2930     }
2931
2932     handlerSubmit(){
2933         const { value } = this.state;
2934         if(!value){
2935             this.setState({invalid: true})
2936         }

```

```

2936     const { afterSubmit } = this.props;
2937     postRequest(
2938         '/mentoring',
2939         { question: value },
2940         () => {
2941             this.setState({value: ''});
2942             afterSubmit();
2943         }
2944     );
2945 }
2946
2947 render(){
2948     return (
2949         <Card>
2950             <CardBody>
2951                 <Form>
2952                     <Input id="mentoring" label="Questo"
2953                         {...this.state} invalidMessage="Campo
2954                         Obrigatorio"
2955                         onChange={this.handlerQuestion}
2956                         required />
2957                     <Button type="button" color="primary"
2958                         className="float-right"
2959                         onClick={this.handlerSubmit}>Cadastrar</Button>
2960                 </Form>
2961             </CardBody>
2962         </Card>
2963     )
2964 }
2965
2966 webapp/src/pages/protected/admin/evaluate/mentoring/index.js
2967
2968 import List from './List';
2969 export default List;
2970
2971 webapp/src/pages/protected/admin/evaluate/mentoring/List.jsx
2972
2973 import React, { Component, Fragment } from 'react';
2974 import { getRequest, deleteRequest } from
2975     './../../../../utils/http';
2976 import { Col, Row, ListGroup, ListGroupItem } from 'reactstrap';
2977 import { FaTrash } from 'react-icons/fa';
2978 import Form from './Form.jsx';
2979

```

```
2974 export default class extends Component{
2975   constructor(){
2976     super();
2977     this.state = {
2978       mentorings: []
2979     };
2980     this.handlerDelete = this.handlerDelete.bind(this);
2981     this.componentWillMount =
2982       this.componentWillMount.bind(this);
2983   }
2984
2985   componentWillMount(){
2986     getRequest('/mentoring', res => this.setState({
2987       personalites: res.data }));
2988   }
2989
2990   handlerDelete(codQuestion){
2991     deleteRequest('/mentoring/${codQuestion}',
2992       this.componentWillMount);
2993   }
2994
2995   render(){
2996     const { mentorings } = this.state;
2997     return (
2998       <Fragment>
2999         <Row>
3000           <Col>
3001             <h3>Feedback do Voluntrio</h3>
3002           </Col>
3003         </Row>
3004         <br/>
3005         <Row>
3006           <Col>
3007             <ListGroup>
3008               {
3009                 mentorings.length ?
3010                   mentorings.map(
3011                     mentoring =>
3012                       (
3013                         <ListGroupItem
3014                           key={mentoring.codQuestion}>
3015                           <Row>
3016                             <Col>
3017                               <strong>{mentoring.question}
3018                             </Col>
```

```

3015 <Col md="1">
3016   <FaTrash
3017     style={{
3018       cursor:
3019         'pointer',
3020     }}
3021     color="red"
3022     onClick={() =>
3023       this.handlerDelete()
3024     }
3025   )
3026 
3027   )
3028 </ListGroup>
3029 </Col>
3030 <Col md="4">
3031   <Form
3032     afterSubmit={this.componentWillMount}/>
3033   </Col>
3034 </Row>
3035 </Fragment>
3036 );
3037 }
3038
3039 webapp/src/pages/protected/admin/evaluate/Page.jsx
3040
3041 import React, { Component } from 'react';
3042 import { Row, Col, Button, Collapse } from 'reactstrap';
3043 import ListPersonality from './personality';
3044 import ListSkill from './skill';
3045 import ListTechnology from './technology';
3046 import ListMentoring from './mentoring';

```

```
3047
3048 const styleButton = {
3049   width: '100%'
3050 };
3051
3052 export default class extends Component {
3053   constructor() {
3054     super();
3055     this.state = {
3056       show: {
3057         knowledge: false,
3058         personality: false,
3059         skill: false,
3060         mentoring: false,
3061         others: false
3062       }
3063     };
3064   }
3065
3066   changeShow(collapse) {
3067     const { show } = this.state;
3068     for (const key in show) {
3069       if (show.hasOwnProperty(key)) {
3070         show[key] = key === collapse ? !show[key] : false;
3071       }
3072     }
3073     this.setState({ show })
3074   }
3075
3076   render() {
3077     const { role } = this.props;
3078     const { show } = this.state;
3079     return (
3080       <Row>
3081         <Col>
3082           <Row>
3083             <Col>
3084               <Button type="button" color="info"
3085                 style={styleButton} onClick={() =>
3086                   this.changeShow('knowledge')}>Conhecimento</Button>
3087             </Col>
3088             <Col md="1" />
3089             <Col>
3090               <Button type="button" color="info"
3091                 style={styleButton} onClick={() =>
```

```

            this.changeShow('personality')}>Personalidade
            do voluntario</Button>
        </Col>
        <Col md="1" />
        <Col>
            <Button type="button" color="info"
                style={styleButton} onClick={() =>
                    this.changeShow('skill')}>Competencias
                    Desenvolvidas</Button>
        </Col>
        <Col md="1" />
        <Col>
            <Button type="button" color="info"
                style={styleButton} onClick={() =>
                    this.changeShow('mentoring')}>Feedback
                    do Voluntario</Button>
        </Col>
        <Col md="1" />
        <Col>
            <Button type="button" color="info"
                style={styleButton} onClick={() =>
                    this.changeShow('others')}>Avaliao</Button>
        </Col>
    </Row>
    <br />
    <Row>
        <Col>
            <Collapse isOpen={show.knowledge}>
                <ListTechnology/>
            </Collapse>
            <Collapse isOpen={show.personality}>
                <ListPersonality/>
            </Collapse>
            <Collapse isOpen={show.skill}>
                <ListSkill/>
            </Collapse>
            <Collapse isOpen={show.mentoring}>
                <ListMentoring/>
            </Collapse>
            <Collapse isOpen={show.others}>
                teste
            </Collapse>
        </Col>
    </Row>
</Col>

```

3123

```
3124             </Row>
3125         );
3126     }
3127 }
3128
3129 webapp/src/pages/protected/admin/evaluate/index.js
3130
3131 import Page from './Page.jsx';
3132 export default Page;
3133
3134 webapp/src/pages/protected/admin/eventType/Form.jsx
3135
3136 import React, { Component, Fragment } from 'react';
3137 import { EventType, Training } from '../../../../../model';
3138 import PropTypes from 'prop-types';
3139 import { postRequest } from '../../../../../utils/http';
3140 import { Row, Col, Button, Form, Card, CardBody } from
3141   'reactstrap';
3142 import { Input, FileInput, Switch } from
3143   '../../../../../components';
3144
3145 export default class extends Component{
3146   constructor(){
3147     super();
3148     this.state = {
3149       typeEvent: new EventType(),
3150       files: []
3151     };
3152     this.handlerName = this.handlerName.bind(this);
3153     this.handlerAdd = this.handlerAdd.bind(this);
3154     this.handlerSubmit = this.handlerSubmit.bind(this);
3155   }
3156
3157   static propTypes = {
3158     afterSubmit: PropTypes.func.isRequired,
3159   }
3160
3161   handlerName(event){
3162     const { typeEvent } = this.state;
3163     typeEvent.name = event.target.value;
3164     this.setState({ typeEvent });
3165   }
3166
3167   handlerAdd() {
3168     const { typeEvent } = this.state;
```

```
3167     let training = new Training();
3168     training.isFile = false;
3169     typeEvent.trainings.push(training);
3170     this.setState({ typeEvent })
3171 }
3172
3173 handlerNameAttachemnt(event, index) {
3174     const { typeEvent } = this.state;
3175     typeEvent.trainings[index].name = event.target.value;
3176     /*const training = typeEvent.trainings[index];
3177     training.name = event.target.value;
3178     if(training.isFile && files[index]){
3179         this.upload()
3180     } else{
3181         this.setState({ typeEvent });
3182     }*/
3183     this.setState({ typeEvent });
3184 }
3185
3186 handlerLinkAttachemnt(event, index) {
3187     const { typeEvent } = this.state;
3188     typeEvent.trainings[index].link = event.target.value;
3189     this.setState({ typeEvent });
3190 }
3191
3192 handlerUploadAttachemnt(event, index) {
3193     const file = event.target.files[0];
3194     console.log(file);
3195 }
3196
3197 /*upload(index){
3198     a
3199 }*/
3200
3201 handlerRemove(index) {
3202     const { typeEvent } = this.state;
3203     typeEvent.trainings.splice(index, 1);
3204     this.setState({ typeEvent });
3205 }
3206
3207
3208 handlerIsFile(value, index) {
3209     const { typeEvent } = this.state;
3210     typeEvent.trainings[index].isFile = value
3211     this.setState({ typeEvent });
3212 }
```

```
3212     }
3213
3214     handlerSubmit(){
3215         const { afterSubmit } = this.props;
3216         const { typeEvent } = this.state;
3217         /*const json = JSON.stringify(typeEvent);
3218         const blob = new Blob([json], {
3219             type: 'application/json'
3220         });
3221         let form = new FormData();
3222         form.append('typeEvent', blob);
3223         if(file){
3224             form.append('file', file);
3225         }*/
3226         postRequest('/typeEvent', typeEvent, () => {
3227             afterSubmit();
3228             this.setState({ typeEvent: new EventType() });
3229         })
3230     }
3231
3232     render(){
3233         const { typeEvent } = this.state;
3234         const { trainings } = typeEvent;
3235         return (
3236             <Card>
3237                 <CardBody>
3238                     <Form>
3239                         <Row>
3240                             <Col>
3241                                 <Input id="name" label="Tipo de
3242                                     Evento"
3243                                     onChange={this.handlerName}
3244                                     invalidMessage="Tipo de Evento
3245                                     Obrigatorio"
3246                                     value={typeEvent.type}
3247                                     required/>
3248                             </Col>
3249                         </Row>
3250                         <Row>
3251                             <Col>
3252                                 <h4>Materiais</h4>
3253                             </Col>
3254                         </Row>
3255                         {
3256                             trainings.length ?
```



```

3268           this.handlerUpload
3269           index)}
3270           accept="application/
3271           required
3272           />
3273       ) : (
3274         <Input
3275           id={'link-${index}
3276           label="Link"
3277           invalidMessage="Lin
3278           Obrigatório"
3279           value={training.link}
3280           onChange={event
3281             =>
3282               this.handlerLinkAt
3283               index)}
3284             required
3285             />
3286           )
3287         }
3288       </Col>
3289       <Col md="3">
3290         {!training.isFile
3291           &&
3292             <label>&nbsp;</label>}
3293           <Button outline
3294             color="secondary"
3295             onClick={()}
3296             =>
3297               this.handlerRemove(index)}>Re
3298           </Col>
3299         </Row>
3300       </Fragment>
3301     )
3302   ): null
3303 }
3304 <Row>
3305   <Col>
3306     <Button outline color="secondary"
3307       onClick={this.handlerAdd}>Incluir</Button>
3308     </Col>
3309   </Row>
3310   <br/>
3311 <Row>

```

```
3289             <Col>
3290                 <Button type="button"
3291                     color="primary"
3292                     className="float-right"
3293                     onClick={this.handlerSubmit}>Cadastrar</B
3294             </Col>
3295         </Row>
3296     </Form>
3297 </CardBody>
3298 </Card>
3299 );
3300 }
3301
3302 webapp/src/pages/protected/admin/eventType/index.js
3303
3304 import List from './List';
3305 export default List;
3306
3307 webapp/src/pages/protected/admin/eventType/ListItem.jsx
3308
3309 import React, { Component, Fragment } from 'react';
3310 import PropTypes from 'prop-types';
3311 import { deleteRequest } from '../../../../../utils/http';
3312 import { FaTrash } from 'react-icons/fa';
3313 import Trainings from './Trainings.jsx';
3314
3315 export default class extends Component{
3316
3317     static propTypes = {
3318         typeEvent: PropTypes.object.isRequired,
3319         afterDelete: PropTypes.func.isRequired
3320     }
3321
3322     handlerDelete(){
3323         const { typeEvent, afterDelete } = this.props;
3324         deleteRequest(
3325             `/typeEvent/${typeEvent.type}`,
3326             afterDelete
3327         )
3328     }
3329
3330     render(){
3331         const { typeEvent } = this.props;
3332         return (
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4598
4599
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4678
4679
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4698
4699
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4778
4779
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4798
4799
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4878
4879
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4898
4899
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4978
4979
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4998
4999
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5098
5099
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5178
5179
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5278
5279
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5298
5299
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5378
5379
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5398
5399
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5478
5479
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5498
5499
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5578
5579
5579
5580
5581
5582
5583
5584
5585
5586
5587

```

```
3331     <Fragment>
3332         <tr>
3333             <td>
3334                 <strong>
3335                     {typeEvent.name}
3336                 </strong>
3337             </td>
3338             <td>
3339                 <Trainings type={typeEvent.type}/>
3340             </td>
3341             <td>
3342                 <FaTrash style={{ cursor: 'pointer' }}>
3343                     color="red"
3344                     onClick={this.handlerDelete.bind(this)}
3345                 />
3346             </td>
3347         </tr>
3348     </Fragment>
3349   );
3350 }
3351
3352 webapp/src/pages/protected/admin/eventType/Trainings.jsx
3353
3354 import React, { Component, Fragment } from 'react';
3355 import PropTypes from 'prop-types';
3356 import { Button, Tooltip } from 'reactstrap';
3357 import { getRequest } from '../../../../../utils/http';
3358
3359 export default class extends Component{
3360     constructor(){
3361         super();
3362         this.state = {
3363             isOpen: false,
3364             trainings: []
3365         };
3366         this.handlerTrainings = this.handlerTrainings.bind(this);
3367     }
3368
3369     static propTypes = {
3370         type: PropTypes.number.isRequired
3371     }
3372
3373     handlerTrainings(){
3374         const { isOpen } = this.state;
```



```
3413     };
3414     this.componentWillMount =
3415         this.componentWillMount.bind(this);
3416 }
3417
3418 componentWillMount() {
3419     getRequest(
3420         '/typeEvent/page/0',
3421         res => {
3422             const { data: typesEvent, count } = res.data
3423             this.setState({
3424                 pages: count / 20,
3425                 typesEvent,
3426                 count
3427             });
3428         }
3429     );
3430
3431     handlerFilter(event) {
3432         const filter = event.target.value;
3433         if (filter.length >= 3 || filter.length === 0) {
3434             getRequest(
3435                 '/typeEvent/page/0?filter=${filter}',
3436                 res => {
3437                     const { data: typesEvent, count } = res.data
3438                     this.setState({
3439                         pages: count / 20,
3440                         typesEvent,
3441                         count,
3442                         filter
3443                     })
3444                 }
3445             );
3446         } else {
3447             this.setState({ filter })
3448         }
3449     }
3450
3451     onChangePage(pageNumber) {
3452         const { filter } = this.state;
3453         getRequest(
3454             '/typeEvent/page/${pageNumber}?filter=${filter.length
3455                 > 3 ? filter : ''}',
3456             res => {
```



```

3496                                         typeEvent={typeEvent}
3497                                         afterDelete={this.componentWillMount}
3498                                         )
3499                                         ) : (
3500                                         <tr>
3501                                         <td colSpan="3">
3502                                         <strong>Nenhum
3503                                         Tipo de
3504                                         Evento
3505                                         cadastrado</strong>
3506                                         </td>
3507                                         </tr>
3508                                         )
3509                                         }
3510                                         </tbody>
3511                                         </Table>
3512                                         <Pagination pages={pages} count={count}
3513                                         onChangePage={this.onChangePage.bind(this)}
3514                                         />
3515                                         </Col>
3516                                         <Col md="4">
3517                                         <Form
3518                                         afterSubmit={this.componentWillMount}/>
3519                                         </Col>
3520                                         </Row>
3521                                         </Fragment>
3522                                         );
3523                                         }
3524                                         }
3525                                         webapp/src/pages/protected/admin/Page.jsx
3526
3527                                         import React, { Component } from 'react';
3528                                         import { Redirect } from 'react-router-dom';
3529                                         import { Row, Col, Button, Collapse } from 'reactstrap';
3530                                         import ListAttachment from './attachment';
3531                                         import ListQuestion from './question';
3532                                         import ListEventType from './eventType';
3533                                         import ListProject from './project';
3534                                         import EvaluatePage from './evaluate';
3535                                         import { Roles } from '../../../../../model';
3536
3537                                         const styleButton = {
3538                                         width: '100%'
3539                                         };

```

```
3533
3534 export default class extends Component{
3535     constructor(){
3536         super();
3537         this.state = {
3538             show: {
3539                 project: false,
3540                 eventType: false,
3541                 attachment: false,
3542                 question: false,
3543                 evaluate: false
3544             }
3545         };
3546     }
3547
3548     changeShow(collapse) {
3549         const { show } = this.state;
3550         for (const key in show) {
3551             if (show.hasOwnProperty(key)) {
3552                 show[key] = key === collapse ? !show[key] : false;
3553             }
3554         }
3555         this.setState({ show })
3556     }
3557
3558     render(){
3559         const { role, logged } = this.props;
3560         const { show } = this.state;
3561         if (role === Roles.ADMIN) {
3562             return (
3563                 <Row>
3564                     <Col md="1" />
3565                     <Col>
3566                         <Row>
3567                             <Col>
3568                                 <Button type="button" color="info"
3569                                     style={styleButton} onClick={() =>
3570                                         this.changeShow('project')}>Projetos</But
3571
3572                                     </Col>
3573                                     <Col md="1" />
3574                                     <Col>
3575                                         <Button type="button" color="info"
3576                                             style={styleButton} onClick={() =>
3577                                         this.changeShow('evaluate')}>Evaluaciones</But
```

```
            this.changeShow('eventType')}>Tipos  
            de Evento</Button>  
        </Col>  
        <Col md="1" />  
        <Col>  
            <Button type="button" color="info"  
                style={styleButton}  
                onClick={()=>this.changeShow('attachment')}>Ane  
        </Col>  
        <Col md="1" />  
        <Col>  
            <Button type="button" color="info"  
                style={styleButton}  
                onClick={()=>this.changeShow('question')}>Quest  
        </Col>  
        <Col md="1" />  
        <Col>  
            <Button type="button" color="info"  
                style={styleButton}  
                onClick={()=>this.changeShow('evaluate')}>Avali  
        </Col>  
    </Row>  
    <br/>  
    <Row>  
        <Col>  
            <Collapse isOpen={show.project}>  
                <ListProject />  
            </Collapse>  
            <Collapse isOpen={show.eventType}>  
                <ListEventType />  
            </Collapse>  
            <Collapse isOpen={show.attachment}>  
                <ListAttachment />  
            </Collapse>  
            <Collapse isOpen={show.question}>  
                <ListQuestion />  
            </Collapse>  
            <Collapse isOpen={show.evaluate}>  
                <EvaluatePage/>  
            </Collapse>  
        </Col>  
    </Row>  
    <Col md="1" />  
</Row>
```

```
3610         );
3611     }
3612     if(logged){
3613         return <Redirect to="/dashboard" />
3614     }
3615     return <Redirect to="/login" />
3616   }
3617 }
3618
3619 webapp/src/pages/protected/admin/index.js
3620
3621 import Page from './Page.jsx';
3622 export default Page;
3623
3624 webapp/src/pages/protected/dashboard/admin/volunteer/index.js
3625
3626 import List from './List.jsx';
3627 export default List;
3628
3629 webapp/src/pages/protected/dashboard/admin/volunteer/ListItem.jsx
3630
3631 import React, { Component } from 'react';
3632 import { Row, Col, ListGroupItem } from 'reactstrap';
3633 import { FaTrash } from 'react-icons/fa';
3634 import Details from './Details.jsx';
3635 import PropTypes from 'prop-types';
3636 import { Checkbox } from '../../../../../components';
3637 import { deleteRequest } from '../../../../../utils/http';
3638 import formatter from '../../../../../utils/formatter';
3639
3640 export default class extends Component{
3641   constructor(){
3642     super();
3643     this.state = {
3644       isOpen: false
3645     }
3646     this.toggle = this.toggle.bind(this);
3647     this.handlerDelete = this.handlerDelete.bind(this);
3648   }
3649
3650   static propTypes = {
3651     volunteer: PropTypes.object.isRequired,
3652     afterDelete: PropTypes.func.isRequired
3653   }
3654 }
```

```

3655     toggle(){
3656         const { isOpen } = this.state;
3657         this.setState({isOpen: !isOpen});
3658     }
3659
3660     handlerDelete(){
3661         const { volunteer, afterDelete } = this.props;
3662         const { user } = volunteer;
3663         deleteRequest('/user/${user.cpf}', afterDelete)
3664     }
3665
3666     render(){
3667         const { volunteer, onSelectVolunteer } = this.props;
3668         const { user } = volunteer;
3669         const { isOpen } = this.state;
3670         return (
3671             <ListGroupItem>
3672                 <Row>
3673                     <Col md="1">
3674                         <Checkbox value={user.cpf}
3675                             onChange={onSelectVolunteer} />
3676                     </Col>
3677                     <Col onClick={this.toggle}
3678                         className="text-ellipsis" style={{ cursor: 'pointer' }}>
3679                         {formatter.cpf(user.cpf)} - {user.name}
3680                     </Col>
3681                     <Col md="2">
3682                         <FaTrash style={{ cursor: 'pointer' }}
3683                             color="red"
3684                             onClick={this.handlerDelete} />
3685                     </Col>
3686                 </Row>
3687                 { isOpen && <Details cpf={user.cpf}
3688                     isOpen={isOpen} close={this.toggle}
3689                     deleteFunc={this.handlerDelete}/> }
3690             </ListGroupItem>
3691         );
3692     }
3693 }
3694
3695     webapp/src/pages/protected/dashboard/admin/volunteer/Details.jsx
3696
3697     import React, { Component } from 'react';

```

```
3692 import { getRequest, putRequest } from
  ' ../../../../../../utils/http';
3693 import formatter from ' ../../../../../../utils/formatter';
3694 import { Input } from ' ../../../../../../components';
3695 import { Status, Address, Schooling, School } from
  ' ../../../../../../model';
3696 import axios from ' axios';
3697 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
  Row, Col, Label, Form } from 'reactstrap';
3698 import pic from './pic.jpeg';
3699
3700 export default class extends Component{
3701   constructor(){
3702     super();
3703     this.state = {
3704       volunteer: null,
3705       trainings: [],
3706       schools: [],
3707       err: {
3708         invalid: false
3709       }
3710     };
3711     this.handlerAprroved = this.handlerAprroved.bind(this);
3712     this.HandlerAssociate = this.HandlerAssociate.bind(this);
3713     this.handlerDelete = this.handlerDelete.bind(this);
3714     this.HandlerSubmit = this.HandlerSubmit.bind(this);
3715   }
3716
3717   componentWillMount(){
3718     const { cpf } = this.props;
3719     axios.all([
3720       getRequest('/volunteer/${cpf}', res => res.data),
3721       getRequest('/school', res => res.data)
3722     ]).then(
3723       res => this.setState({
3724         volunteer: res[0],
3725         schools: res[1]
3726       })
3727     );
3728   }
3729
3730   handlerDelete(){
3731     const { deleteFunc, close } = this.props;
3732     deleteFunc();
3733     close();
```

```
3734     }
3735
3736     handlerApproved(){
3737         const { close } = this.props;
3738         const { volunteer } = this.state;
3739         const { user } = volunteer;
3740         user.status = Status.APPROVED;
3741         volunteer.user = user;
3742         putRequest('/volunteer/${user.cpf}', volunteer, close);
3743     }
3744
3745     HandlerAssociate(event){
3746         debugger;
3747         const { volunteer, schools } = this.state;
3748         const { value } = event.target;
3749         if(value){
3750             for (const school of schools) {
3751                 if(school.codSchool === parseInt(value, 10)){
3752                     volunteer.school = school;
3753                     break;
3754                 }
3755             }
3756         } else {
3757             volunteer.school = new School()
3758         }
3759         this.setState({
3760             volunteer,
3761             err: {
3762                 invalid: false
3763             }
3764         });
3765     }
3766
3767     HandlerSubmit(){
3768         const { volunteer } = this.state;
3769         const { user } = volunteer;
3770         if (volunteer.school.codSchool){
3771             putRequest('/volunteer/${user.cpf}', volunteer);
3772         } else {
3773             this.setState({
3774                 err: {
3775                     invalid: true
3776                 }
3777             })
3778         }
3779     }
```



```
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
</Row>
<Row>
    <Col>
        {
            Schooling.translate(volunteer.schooling)
        } -
        {volunteer.conclusion ?
        'Completo':
        'Incompleto'}
    {
        volunteer.course && <div>
            { volunteer.course }
        </div>
    }
    </Col>
</Row>
<br />
<Row>
    <Col>
        <strong>Contato:</strong>
    </Col>
</Row>
<Row>
    <Col>
        {user.email} /
        {formatter.phone(user.phone)}
    </Col>
</Row>
<br/>
<Row>
    <Col>
        <strong>Endereo :</strong>
    </Col>
</Row>
<Row>
    <Col>
        {address.toString()}
    </Col>
</Row>
<br />
<Row>
    <Col>
        <strong> Avaliaes :</strong>
    </Col>
</Row>
```

```

3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878

        <Row>
            <Col>
                {
                    volunteer.ratings.length?
                        ' Avaliao mdia de
                        ${user.name}
                        ${volunteer.rating}':
                        'Voluntrio ainda
                        no foi avaliado'
                }
            </Col>
        </Row>
        <br/>
        <Row>
            <Col>
                <Form inline>
                    <Input id="school"
                            type="select"
                            label="Escola"
                            invalidMessage="O
                            voluntrio deve ser
                            associado a uma
                            escola"
                            onChange={this.HandlerAssociate
                                value={school?
                                    school.codSchool
                                    : ''} {...err}
                                required>
                    <option
                            value="">Selecione</option>
                {
                    schools.map(school
                        => <option
                            key={school.codSchool}
                            value={school.codSchool
                                }
                        )
                }
            </Input>
            <Button color="primary"
                    type="button"
                    onClick={this.HandlerSubmit}>A
                </Form>
            </Col>
        </Row>
    </Col>
</Row>

```

```
3879         </ModalBody>
3880         <ModalFooter style={{ display: 'inline' }}>
3881             <Row>
3882                 <Col>
3883                     <Button color="danger"
3884                         onClick={this.handlerDelete}>Excluir</Button>
3885                 </Col>
3886                 <Col>
3887                     {
3888                         this.isApprovable() ? (
3889                             <Button color="info"
3890                                 className="float-right"
3891                                 onClick={this.handlerApproved}>Aprovar</Button>
3892                         ) : (
3893                             <Button color="default"
3894                                 className="float-right"
3895                                 onClick={close}>Fechar</Button>
3896                         )
3897                     )
3898                 }
3899             </Row>
3900             </ModalFooter>
3901         </Modal>
3902     )
3903 }
3904
3905 webapp/src/pages/protected/dashboard/admin/volunteer>List.jsx
3906
3907 import React, { Component, Fragment } from 'react';
3908 import { getRequest } from '../../../../../utils/http';
3909 import { Row, Col, ListGroup, ListGroupItem } from 'reactstrap';
3910 import VolunteerItem from './ListItem.jsx';
3911 import { Pagination, Filter } from '../../../../../components';
3912
3913 export default class extends Component{
3914     constructor() {
3915         super()
3916         this.state = {
3917             filter: '',
3918             volunteers: [] ,
```

```
3919         count: 0,
3920         pages: 0
3921     }
3922     this.onChangePage = this.onChangePage.bind(this);
3923     this.handlerFilter = this.handlerFilter.bind(this);
3924 }
3925
3926 componentWillMount() {
3927     const { filterByCompany, filterBySchool } = this.props;
3928     const { filter } = this.state;
3929     getRequest(
3930         '/volunteer/page/0?cod_cnpj=${filterByCompany.join(',')}&cod_sch'
3931     )
3932     res => {
3933         const page = res.data;
3934         this.setState({
3935             volunteers: page.data,
3936             count: page.count,
3937             pages: page.count / 5
3938         });
3939     );
3940 }
3941
3942 componentWillReceiveProps(){
3943     this.componentWillMount();
3944 }
3945
3946 handlerFilter(event) {
3947     const { filterByCompany, filterBySchool } = this.props;
3948     const filter = event.target.value;
3949     if (filter.length >= 3 || filter.length === 0){
3950         getRequest(
3951             '/volunteer/page/0?cod_cnpj=${filterByCompany.join(',')}&cod_sch'
3952         )
3953         res => {
3954             const page = res.data;
3955             this.setState({
3956                 volunteers: page.data,
3957                 count: page.count,
3958                 pages: page.count / 5,
3959                 filter
3960             });
3961         );
3962     } else {
3963         this.setState({filter});
3964     }
3965 }
```

```

3964     }
3965   }
3966
3967   onChangePage(pageNumber){
3968     const { filterByCompany, filterBySchool } = this.props;
3969     const { filter } = this.state;
3970     getRequest(
3971       `/volunteer/page/${pageNumber}?cod_cnpj=${filterByCompany.join(' , ')}`&
3972         > 3 ? filter : ''
3973     );
3974     res => {
3975       const page = res.data;
3976       this.setState({
3977         volunteers: page.data,
3978         count: page.count,
3979         pages: page.count / 5
3980       });
3981     }
3982   }
3983   render(){
3984     const { volunteers, pages, count } = this.state;
3985     const { onSelectVolunteer } = this.props;
3986     return (
3987       <Fragment>
3988         <Row>
3989           <Col>
3990             <h3>Mentores Voluntrios</h3>
3991           </Col>
3992         </Row>
3993         <Filter handlerFilter={this.handlerFilter} />
3994         <Row>
3995           <Col>
3996             <ListGroup>
3997               {
3998                 volunteers.length ?
3999                   volunteers.map(
4000                     volunteer => (
4001                       <VolunteerItem
4002                         key={volunteer.user.cpf}
4003                         volunteer={volunteer}
4004                         onSelectVolunteer={onSelectVolunteer}
4005                         afterDelete={this.componentWillMount
4006                           }
4007                     )
4008                   ) : (
4009                     <ListGroupItem>

```

```
4004             <strong>
4005                 Nenhum Voluntrio
4006                     Cadastrado
4007             </strong>
4008         </ListGroupItem>
4009     )
4010     </ListGroup>
4011   </Col>
4012 </Row>
4013   <Pagination pages={pages} count={count}
4014       onChangePage={this.onChangePage}/>
4015   </Fragment>
4016 );
4017 }
4018
4019 webapp/src/pages/protected/dashboard/admin/school/Form.jsx
4020
4021 import React, { Component } from 'react';
4022 import { School, SchoolType } from '../../../../../model';
4023 import PropTypes from 'prop-types';
4024 import { postRequest } from '../../../../../utils/http';
4025 import { Row, Col, Modal, ModalHeader, ModalBody } from
4026   'reactstrap';
4027 import { Wizard, Input, MaskInput, DatePicker } from
4028   '../../../../../components';
4029
4030 export default class extends Component{
4031   constructor(){
4032     super();
4033     this.state = {
4034       school: new School(),
4035       confirmPassword: null
4036     };
4037     this.handlerName = this.handlerName.bind(this);
4038     this.handlerPhone = this.handlerPhone.bind(this);
4039     this.handlerType = this.handlerType.bind(this);
4040     this.handlerNameResponsible =
4041       this.handlerNameResponsible.bind(this);
4042     this.handlerCpf = this.handlerCpf.bind(this);
4043     this.handlerPhoneResponsible =
4044       this.handlerPhoneResponsible.bind(this);
4045     this.handlerJob = this.handlerJob.bind(this);
4046     this.handlerEmail = this.handlerEmail.bind(this);
```

```
4043     this.handlerBirth = this.handlerBirth.bind(this);
4044     this.handlerPassword = this.handlerPassword.bind(this);
4045     this.handlerConfirmPassword =
4046         this.handlerConfirmPassword.bind(this);
4047     this.handlerCity = this.handlerCity.bind(this);
4048     this.handlerNeighborhood =
4049         this.handlerNeighborhood.bind(this);
4050     this.handlerStreet = this.handlerStreet.bind(this);
4051     this.handlerNumber = this.handlerNumber.bind(this);
4052     this.handlerSubmit = this.handlerSubmit.bind(this);
4053 }
4054
4055 static propTypes = {
4056     afterSubmit: PropTypes.func.isRequired,
4057     isOpen: PropTypes.bool.isRequired,
4058     close: PropTypes.func.isRequired
4059 }
4060
4061 //school
4062 handlerName(event){
4063     const { school } = this.state;
4064     school.name = event.target.value;
4065     this.setState({ school });
4066 }
4067
4068 handlerPhone(event){
4069     const { school } = this.state;
4070     school.phone = event.target.value.replace(/[\(\)-]/g, '');
4071     this.setState({ school });
4072 }
4073
4074 handlerType(event) {
4075     const { school } = this.state;
4076     school.type = event.target.value;
4077     this.setState({ school });
4078 }
4079
4080 // responsible
4081 handlerNameResponsible(event){
4082     const { school } = this.state;
4083     const { responsible } = school;
4084     responsible.name = event.target.value;
4085     school.responsible = responsible;
4086     this.setState({ school });
4087 }
```

```
4086     }
4087
4088     handlerCpf(event){
4089         const { school } = this.state;
4090         const { responsible } = school;
4091         responsible.cpf = event.target.value.replace(/[-]/g,
4092             '');
4093         school.responsible = responsible;
4094         this.setState({ school });
4095     }
4096
4097     handlerPhoneResponsible(event) {
4098         const { school } = this.state;
4099         const { responsible } = school;
4100         responsible.phone = event.target.value.replace(/[()]-/g,
4101             '');
4102         school.responsible = responsible;
4103         this.setState({ school });
4104     }
4105
4106     handlerJob(event){
4107         const { school } = this.state;
4108         const { responsible } = school;
4109         responsible.job = event.target.value;
4110         school.responsible = responsible;
4111         this.setState({ school });
4112     }
4113
4114     handlerEmail(event){
4115         const { school } = this.state;
4116         const { responsible } = school;
4117         responsible.email = event.target.value;
4118         school.responsible = responsible;
4119         this.setState({ school });
4120     }
4121
4122     handlerBirth(event){
4123         const { school } = this.state;
4124         const { responsible } = school;
4125         const { value } = event.target;
4126         responsible.birth = value ? new Date(value) : null;
4127         school.responsible = responsible;
4128         this.setState({ school });
4129     }
4130
```

```
4129     handlerPassword(event){
4130         const { school } = this.state;
4131         const { responsible } = school;
4132         responsible.password = event.target.value;
4133         school.responsible = responsible;
4134         this.setState({ school });
4135     }
4136
4137     handlerConfirmPassword(event){
4138         let { confirmPassword } = this.state;
4139         confirmPassword = event.target.value;
4140         this.setState({ confirmPassword });
4141     }
4142
4143 //address
4144     handlerCity(event){
4145         const { school } = this.state;
4146         const { responsible, address } = school;
4147         address.city = event.target.value;
4148         school.address = responsible.address = address;
4149         school.responsible = responsible;
4150         this.setState({ school });
4151     }
4152
4153     handlerNeighborhood(event){
4154         const { school } = this.state;
4155         const { responsible, address } = school;
4156         address.neighborhood = event.target.value;
4157         school.address = responsible.address = address;
4158         school.responsible = responsible;
4159         this.setState({ school });
4160     }
4161
4162     handlerStreet(event){
4163         const { school } = this.state;
4164         const { responsible, address } = school;
4165         address.street = event.target.value;
4166         school.address = responsible.address = address;
4167         school.responsible = responsible;
4168         this.setState({ school });
4169     }
4170
4171     handlerNumber(event){
4172         const { school } = this.state;
4173         const { responsible, address } = school;
```

```
4174     address.number = event.target.value;
4175     school.address = responsible.address = address;
4176     school.responsible = responsible;
4177     this.setState({ school });
4178 }
4179
4180 handlerSubmit(){
4181     const { afterSubmit, close } = this.props;
4182     const { school } = this.state;
4183     postRequest(
4184         '/school',
4185         school,
4186         () => {
4187             afterSubmit();
4188             close();
4189         }
4190     );
4191 }
4192
4193 render(){
4194     const { close, isOpen } = this.props;
4195     const { school, confirmPassword } = this.state;
4196     const { address, responsible } = school;
4197     const maxDate = new Date().toJSON().split('T')[0];
4198     return (
4199         <Modal isOpen={isOpen} toggle={close}>
4200             <ModalHeader toggle={close}>Nova
4201                 Empresa</ModalHeader>
4202             <ModalBody>
4203                 <Wizard onCancel={close}>
4204                     <SubmitLabel>cadastrar</SubmitLabel>
4205                     <OnSubmit>{this.handlerSubmit}</OnSubmit>
4206                     <div>
4207                         <h3>Dados da Escola</h3>
4208                         <Input id="nameCompany" label="Nome da
4209                             Escola" invalidMessage="Nome da
4210                             Escola obrigatorio"
4211                             value={school.name}
4212                             onChange={this.handlerName}
4213                             required/>
4214                         <MaskInput id="phone" label="Contato
4215                             da Empresa"
4216                             invalidMessage="Contato da Escola
4217                             obrigatorio" value={school.phone}
4218                             mask="(99)9999-9999"
```

```
4207          onChange={this.handlerPhone}
        required/>
<Input id="schoolType" type="select"
      label="Nvel Federativo"
      invalidMessage="Nvel Federativo
      obrigatorio"
      value={school.schoolType}
      onChange={this.handlerType}
      required >
<option value="">Selecione</option>
{
  SchoolType.values().map(
    type => <option key={type}
      value={type}>{SchoolType.translate(type)
    )
  }
</Input>
<hr className="row"/>
</div>
<div>
  <h3>Dados do responsvel</h3>
  <Input id="nameResponsible" label="Nome
    Completo" invalidMessage="Nome
    Completo obrigatorio"
    value={responsible.name}
    onChange={this.handlerNameResponsible}
    required />
  <MaskInput id="cpf" label="CPF"
    invalidMessage="CPF obrigatorio"
    value={responsible.cpf}
    mask="999.999.999-99"
    onChange={this.handlerCpf}
    required />
  <MaskInput id="phoneResponsible"
    label="Contato do Responsvel"
    invalidMessage="Contato do
    Responsvel obrigatorio"
    value={responsible.phone}
    mask="(99)9999-9999"
    onChange={this.handlerPhoneResponsible}
    required />
  <Input id="job" label="Cargo do
    Responsvel" invalidMessage="Cargo
    do Responsvel obrigatorio"
    value={responsible.job}
```

```
        onChange={this.handlerJob}
        required />
    <DatePicker id="birth"
        label="Nascimento"
        invalidMessage="Nascimento
        obrigatorio" max={maxDate}
        onChange={this.handlerBirth}
        required />
    <Input id="email" type="email"
        label="Email"
        invalidMessage="Email obrigatorio"
        value={responsible.email}
        onChange={this.handlerEmail}
        required />
<Row>
    <Col>
        <Input id="password"
            type="password"
            label="Senha"
            invalidMessage="Senha
            obrigatorio"
            value={responsible.password}
            onChange={this.handlerPassword}
            required />
    </Col>
    <Col>
        <Input id="passwordConfirm"
            type="password"
            label="Confirmar Senha"
            invalidMessage="As senhas
            devem ser idnticas"
            value={confirmPassword}
            onChange={this.handlerConfirmPassword}
            required />
    </Col>
</Row>
<hr className="row"/>
</div>
<div>
    <h3>Endereço da Escola</h3>
    <Input id="city" label="Cidade"
        invalidMessage="Cidade
        obrigatorio" value={address.city}
        onChange={this.handlerCity}
        required/>
```

```

4238             <Input id="neighborhood"
4239                 label="Bairro"
4240                 invalidMessage="Bairro
4241                     obrigatorio"
4242                     value={address.neighborhood}
4243                     onChange={this.handlerNeighborhood}
4244                     required/>
4245
4246             <Row>
4247                 <Col>
4248                     <Input id="street"
4249                         label="Lougradouro"
4250                         invalidMessage="Lougradouro
4251                             obrigatorio"
4252                             value={address.street}
4253                             onChange={this.handlerStreet}
4254                             required/>
4255
4256             </Col>
4257             <Col md="3">
4258                 <Input id="number"
4259                     label="Nmero"
4260                     value={address.number}
4261                     onChange={this.handlerNumber}/>
4262
4263         </Col>
4264     </Row>
4265     <hr className="row"/>
4266
4267         </div>
4268     </Wizard>
4269
4270     </ModalBody>
4271
4272     </Modal>
4273
4274     );
4275 }
4276 }
4277
4278 webapp/src/pages/protected/dashboard/admin/school/index.js
4279
4280 import List from './List.jsx';
4281 export default List;
4282
4283 webapp/src/pages/protected/dashboard/admin/school/ListItem.jsx
4284
4285 import React, { Component } from 'react';
4286 import PropTypes from 'prop-types';
4287 import { deleteRequest } from '../../../../../utils/http';
4288 import { Checkbox } from '../../../../../components';
4289 import { Row, Col, ListGroupItem } from 'reactstrap';

```

```
4268 import { FaTrash } from 'react-icons/fa';
4269 import Details from './Details.jsx';
4270
4271 export default class extends Component{
4272     constructor(){
4273         super();
4274         this.state = {
4275             isOpen: false
4276         }
4277         this.handlerDelete = this.handlerDelete.bind(this);
4278         this.handlerDetails = this.handlerDetails.bind(this);
4279     }
4280
4281     static propTypes = {
4282         school: PropTypes.object.isRequired,
4283         afterDelete: PropTypes.func.isRequired
4284     }
4285
4286     handlerDelete(){
4287         const { school, afterDelete } = this.props;
4288         deleteRequest(`/school/${school.codSchool}`,
4289                         afterDelete);
4290     }
4291
4292     handlerDetails(){
4293         const { isOpen } = this.state;
4294         this.setState({ isOpen: !isOpen });
4295     }
4296
4297     render(){
4298         const { school, onSelectSchool } = this.props;
4299         const { isOpen } = this.state;
4300         return (
4301             <ListGroupItem key={school.codSchool}>
4302                 <Row>
4303                     <Col md="1">
4304                         <Checkbox value={school.codSchool}
4305                             onChange={onSelectSchool}/>
4306                     </Col>
4307                     <Col md="9" onClick={this.handlerDetails}
4308                         className="text-ellipsis" style={{
4309                             cursor: 'pointer' }}>
4310                         {school.name}
4311                     </Col>
4312                     <Col>
```

```

4309             <FaTrash style={{ cursor: 'pointer' }}  
4310                 color="red"  
4311                 onClick={this.handlerDelete} />  
4312         </Col>  
4313     </Row>  
4314     {  
4315         isOpen ? <Details code={school.codSchool}  
4316             isOpen={isOpen}  
4317             close={this.handlerDetails}/> : null  
4318     }  
4319  
4320     </ListGroupItem>  
4321   );  
4322 }  
4323  
4324 webapp/src/pages/protected/dashboard/admin/school/Add.jsx  
4325  
4326 import React, { Component } from 'react';  
4327 import PropTypes from 'prop-types';  
4328 import { Button } from 'reactstrap';  
4329 import { FaPlus } from 'react-icons/fa';  
4330 import Form from './Form.jsx';  
4331  
4332 export default class extends Component {  
4333     constructor() {  
4334         super();  
4335         this.state = {  
4336             isOpen: false  
4337         };  
4338     }  
4339  
4340     static propTypes = {  
4341         afterSubmit: PropTypes.func.isRequired  
4342     }  
4343  
4344     handlerAdd() {  
4345         const { isOpen } = this.state;  
4346         this.setState({ isOpen: !isOpen });  
4347     }  
4348  
4349     render() {  
4350         const { isOpen } = this.state;  
4351         const { afterSubmit } = this.props;  
4352         return (  
4353             <div>

```

```
4350         <Button type="button" color="primary"
4351             onClick={this.handlerAdd.bind(this)}>
4352             <FaPlus />
4353         </Button>
4354         <Form isOpen={isOpen} afterSubmit={afterSubmit}
4355             close={this.handlerAdd.bind(this)} />
4356     </div>
4357   );
4358 }
4359
4360 webapp/src/pages/protected/dashboard/admin/school/Details.jsx
4361
4362 import React, { Component } from 'react';
4363 import PropTypes from 'prop-types';
4364 import { Address, SchoolType } from '../../../../../model';
4365 import { getRequest } from '../../../../../utils/http';
4366 import formatter from '../../../../../utils/formatter';
4367 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
4368         Row, Col } from 'reactstrap';
4369 import pic from './pic.jpeg';
4370
4371 export default class extends Component {
4372   constructor(){
4373     super();
4374     this.state = {
4375       school: null
4376     }
4377   }
4378   static propTypes = {
4379     code: PropTypes.string.isRequired,
4380     isOpen: PropTypes.bool.isRequired,
4381     close: PropTypes.func.isRequired
4382   };
4383   componentWillMount() {
4384     const { code } = this.props;
4385     getRequest(`/school/${code}`, res => this.setState({
4386       school: res.data }));
4387   }
4388   render() {
4389     const { isOpen, close } = this.props;
4390     const { school } = this.state;
4391     if (school) {
```

```
4391 const address = new Address();
4392 Object.assign(address, school.address);
4393 return (
4394     <Modal toggle={close} isOpen={isOpen} >
4395         <ModalHeader toggle={close}>
4396             <Row>
4397                 <Col>
4398                     {school.name}
4399                 </Col>
4400             </Row>
4401             <Row>
4402                 <Col>
4403                     <small>
4404                         <strong>Escola
4405                             {SchoolType.translate(school.type)}</strong>
4406                     </small>
4407                 </Col>
4408             </Row>
4409         </ModalHeader>
4410         <ModalBody>
4411             <Row>
4412                 <Col md={1}>
4413                     <img src={pic} alt=""
4414                         className="rounded-circle"
4415                         width="75px" />
4416                 </Col>
4417                 <Col md="2" />
4418                 <Col>
4419                     <Row>
4420                         <Col>
4421                             <strong>Endereço:</strong>
4422                         </Col>
4423                     </Row>
4424                     <Col>
4425                         {address.toString()}
4426                     </Col>
4427                 </Row>
4428                 <br />
4429                 <Row>
4430                     <Col>
4431                         <strong>Contato:</strong>
4432                     </Col>
4433                 </Row>
4434             <Row>
```

```

4433             <Col>
4434                 {school.responsible.email} /
4435                     {formatter.phone(school.phone)}
4436             </Col>
4437         </Row>
4438     </Col>
4439     <br />
4440     <Row>
4441         <Col md="3" />
4442         <Col>
4443             <strong>Responsavel:</strong>
4444         </Col>
4445     </Row>
4446     <Row>
4447         <Col md="3" />
4448         <Col>
4449             <div>{school.responsible.name}</div>
4450             <small>{formatter.cpf(school.responsible.cpf)}</small>
4451         </Col>
4452     </Row>
4453     </ModalBody>
4454     <ModalFooter>
4455         <Button color="primary"
4456             onClick={close}>Fechar</Button>
4457     </ModalFooter>
4458 </Modal>
4459 } else {
4460     return (
4461         <Modal toggle={close} isOpen={isOpen} ></Modal>
4462     );
4463 }
4464 }
4465 }
4466
4467 webapp/src/pages/protected/dashboard/admin/school>List.jsx
4468
4469 import React, { Component } from 'react';
4470 import { getRequest } from '../../../../../utils/http';
4471 import { Row, Col,ListGroup, ListGroupItem } from 'reactstrap';
4472 import { Pagination, Filter } from '../../../../../components';
4473 import SchoolItem from './ListItem.jsx';
4474 import Add from './Add.jsx';
4475

```

```
4476 export default class extends Component{
4477     constructor(){
4478         super();
4479         this.state = {
4480             filter: '',
4481             schools: [],
4482             count: 0,
4483             pages: 0
4484         };
4485         this.onChangePage = this.onChangePage.bind(this);
4486         this.handlerFilter = this.handlerFilter.bind(this);
4487         this.componentWillMount =
4488             this.componentWillMount.bind(this)
4489     }
4490
4491     componentWillMount(){
4492         getRequest(
4493             '/school/page/0',
4494             res => {
4495                 const page = res.data;
4496                 this.setState({
4497                     schools: page.data,
4498                     count: page.count,
4499                     pages: page.count / 5
4500                 });
4501             }
4502         );
4503     }
4504
4505     handlerFilter(event) {
4506         const filter = event.target.value;
4507         if (filter.length >= 3 || filter.length === 0){
4508             getRequest(
4509                 `/school/page/0?filter=${filter}`,
4510                 res => {
4511                     const page = res.data;
4512                     this.setState({
4513                         companies: page.data,
4514                         count: page.count,
4515                         pages: page.count / 5,
4516                         filter
4517                     });
4518                 }
4519             );
4520         } else {
4521     }
```

```
4520         this.setState({filter});
4521     }
4522 }
4523
4524 onChangePage(pageNumber) {
4525     const { filter } = this.state;
4526     getRequest(
4527         `/school/page/${pageNumber}?filter=${filter.length >
4528             3 ? filter : ''}`);
4529     res => {
4530         const page = res.data;
4531         this.setState({
4532             schools: page.data,
4533             count: page.count,
4534             pages: page.count / 5
4535         });
4536     );
4537 }
4538
4539 render(){
4540     const { schools, pages, count } = this.state;
4541     const { onSelectSchool } = this.props;
4542     return (
4543         <div>
4544             <Row>
4545                 <Col>
4546                     <h3>
4547                         Organizaes Beneficiadas
4548                     </h3>
4549                 </Col>
4550                 <Col md="2">
4551                     <Add afterSubmit={this.componentWillMount}>
4552                     </>
4553                 </Col>
4554             </Row>
4555             <Filter handlerFilter={this.handlerFilter} />
4556             <Row>
4557                 <Col>
4558                     <ListGroup>
4559                     {
4560                         schools.length ?
4561                             schools.map(
4562                                 school =>
```

```

4562
4563     ) : (
4564         <SchoolItem
4565             key={school.codSchool}
4566             onSelectSchool={onSelectSchool}
4567             school={school}
4568             afterDelete={this.componentWillMount}
4569             />
4570         )
4571         <ListGroupItem>
4572             <strong>
4573                 Nenhuma Organiza&oacute;
4574                 Cadastrada
4575             </strong>
4576         </ListGroupItem>
4577     )
4578
4579     </Col>
4580     </Row>
4581     <Pagination pages={pages} count={count}
4582         onChangePage={this.onChangePage} />
4583
4584     </div>
4585   );
4586 }
4587
4588 }
4589
4590 webapp/src/pages/protected/dashboard/admin/event/form/Form.jsx
4591
4592 import React, { Component, Fragment } from 'react';
4593 import PropTypes from 'prop-types';
4594 import { postRequest, getRequest } from
4595     '../../../../../utils/http';
4596 import formatter from '../../../../../utils/formatter';
4597 import { Event, School, Training } from
4598     '../../../../../model';
4599 import { Row, Col, Modal, ModalHeader, ModalBody, Button } from
4600     'reactstrap';
4601 import { Wizard, Input, Switch, FileInput } from
4602     '../../../../../components';
4603 import axios from 'axios';
4604 import './form.css';
4605
4606
4607 export default class extends Component {
4608     constructor(props){
4609         super(props);
4610         this.state = {

```

```
4596
4597     event: props.event,
4598     volunteers: [],
4599     schools: [],
4600     typesEvent: [],
4601     trainingsOfType: []
4602   };
4603   this.handlerType = this.handlerType.bind(this);
4604   this.handlerCity = this.handlerCity.bind(this);
4605   this.handlerNeighborhood =
4606     this.handlerNeighborhood.bind(this);
4607   this.handlerStreet = this.handlerStreet.bind(this);
4608   this.handlerNumber = this.handlerNumber.bind(this);
4609   this.handlerParticipants =
4610     this.handlerParticipants.bind(this);
4611   this.handlerSelectSchool =
4612     this.handlerSelectSchool.bind(this);
4613   this.handlerAdd = this.handlerAdd.bind(this);
4614   this.handlerRemove = this.handlerRemove.bind(this);
4615   this.handlerNameAttachemnt =
4616     this.handlerNameAttachemnt.bind(this);
4617   this.handlerLinkAttachemnt =
4618     this.handlerLinkAttachemnt.bind(this);
4619   this.handlerIsFile = this.handlerIsFile.bind(this);
4620   this.handlerEvaluate = this.handlerEvaluate.bind(this);
4621   this.handlerSubmit = this.handlerSubmit.bind(this);
4622 }
4623
4624 static propTypes = {
4625   event: PropTypes.objectOf(Event).isRequired,
4626   afterSubmit: PropTypes.func.isRequired,
4627   isOpen: PropTypes.bool.isRequired,
4628   close: PropTypes.func.isRequired
4629 }
4630
4631 componentWillMount(){
4632   const { event } = this.state;
4633   event.allDay = event.startOccurrence ===
4634     event.endOccurrence;
4635   axios.all([
4636     getRequest('/school', res => res.data),
4637     getRequest('/typeEvent', res => res.data)
4638   ]).then(
4639     res => this.setState({
4640       event,
4641       schools: res[0].
```

```

4635         typesEvent: res[1]
4636     })
4637   );
4638 }
4639
4640 handlerEvaluate(value){
4641   const { event } = this.state;
4642   event.evaluate = value;
4643   this.setState({event});
4644 }
4645
4646 handlerType(userEvent){
4647   const { event, typesEvent } = this.state;
4648   const { value } = userEvent.target;
4649   event.type = null;
4650   if(value){
4651     for (const key in typesEvent) {
4652       if (typesEvent.hasOwnProperty(key)) {
4653         const typeEvent = typesEvent[key];
4654         if(typeEvent.type === parseInt(value, 10)){
4655           event.type = typeEvent;
4656         }
4657       }
4658     }
4659     getRequest(
4660       `/typeEvent/${value}/trainings`,
4661       res => this.setState({ trainingsOfType: res.data,
4662         event }),
4663       () => this.setState({ trainingsOfType: [], event
4664         })
4665     );
4666   }
4667
4668 //address
4669 handlerCity(userEvent) {
4670   const { event } = this.state;
4671   const {address} = event;
4672   address.city = userEvent.target.value;
4673   event.address = address;
4674   this.setState({ event });
4675 }
4676
4677 handlerNeighborhood(userEvent) {
4678   const { event } = this.state;

```

```
4678     const {address } = event;
4679     address.neighborhood = userEvent.target.value;
4680     event.address = address;
4681     this.setState({ event });
4682 }
4683
4684 handlerStreet(userEvent) {
4685     const { event } = this.state;
4686     const {address } = event;
4687     address.street = userEvent.target.value;
4688     event.address = address;
4689     this.setState({ event });
4690 }
4691
4692 handlerNumber(userEvent) {
4693     const { event } = this.state;
4694     const {address } = event;
4695     address.number = userEvent.target.value;
4696     event.address = address;
4697     this.setState({ event });
4698 }
4699
4700 //participants
4701 handlerSelectSchool(userEvent){
4702     const { value } = userEvent.target;
4703     const { event, schools } = this.state
4704     event.school = new School()
4705     for(const school of schools){
4706         if(school.codSchool === parseInt(value, 10)){
4707             event.school = school;
4708         }
4709     }
4710     if(value){
4711         axios.all([
4712             getRequest(
4713                 '/volunteer?cod_school=${value}',
4714                 res => res.data
4715             ),
4716             getRequest(
4717                 '/school/${value}',
4718                 res => res.data
4719             )
4720         ]).then(
4721             res => {
4722                 const school = res[1];
```

```

4723             event.school = school;
4724             event.address = school.address;
4725             event.address.codAddress = null;
4726             this.setState({volunteers: res[0], event});
4727         }
4728     )
4729 } else{
4730     this.setState({volunteers: [], event});
4731 }
4732 }
4733 handlerParticipants(userEvent){
4734     const { event, volunteers } = this.state;
4735     let participants = [], opt, participant;
4736     for (let i = 0; !(opt =
4737         userEvent.target.options[i++]);) {
4738
4739         if (opt.selected) {
4740             for (let j = 0; !(participant =
4741                 volunteers[j++].user);){
4742                 if (opt.value === participant.cpf){
4743                     participants.push(participant);
4744                     break;
4745                 }
4746             }
4747         }
4748     }
4749     event.participants = participants;
4750     this.setState({ event });
4751 }
4752
4753 handlerAdd(){
4754     const { event } = this.state;
4755     let training = new Training();
4756     training.isFile = false;
4757     event.trainings.push(training);
4758     this.setState({event})
4759 }
4760
4761 handlerNameAttachemnt(userEvent, index){
4762     const { event } = this.state;
4763     event.trainings[index].name = userEvent.target.value;
4764     this.setState({event});
4765 }
4766
4767 handlerLinkAttachemnt(userEvent, index) {

```

```
4766     const { event } = this.state;
4767     event.trainings[index].link = userEvent.target.value;
4768     this.setState({ event });
4769 }
4770
4771     handlerUploadAttachemnt(userEvent, index){
4772         const file = userEvent.target.files[0];
4773         console.log(file);
4774     }
4775
4776     handlerRemove(index){
4777         const { event } = this.state;
4778         event.trainings.splice(index, 1);
4779         this.setState({event});
4780     }
4781
4782     handlerIsFile(value, index){
4783         const { event } = this.state;
4784         event.trainings[index].isFile = value
4785         this.setState({ event });
4786     }
4787
4788     handlerSubmit(){
4789         const { afterSubmit } = this.props
4790         let { event } = this.state;
4791         const [start, end] = [ event.startOccurrence,
4792             event.endOccurrence ]
4793         event.startOccurrence = event.startOccurrence.toJSON()
4794         event.endOccurrence = event.endOccurrence.toJSON()
4795         postRequest(
4796             '/event',
4797             event,
4798             afterSubmit,
4799             () => [event.startOccurrence, event.endOccurrence] =
4800                 [start, end]
4801         );
4802     }
4803
4804     render(){
4805         const { isOpen, close } = this.props;
4806         const { event, volunteers, schools, typesEvent,
4807             trainingsOfType } = this.state;
4808         const dateTitle = event.allDay ?
4809             event.startOccurrence.toLocaleString() :
```

```
4807     event.startOccurrence.toLocaleString() + ' - ' +
4808     event.endOccurrence.toLocaleString();
4809 const { codSchool } = event.school;
4810 const { trainings, type, address } = event;
4811 return (
4812     <Modal isOpen={isOpen} toggle={close} style={{width:
4813         'max-content'}}>
4814     <ModalHeader toggle={close}>Novo Evento -
4815         {dateTitle}</ModalHeader>
4816     <ModalBody>
4817         <Wizard onCancel={close}>
4818             <SubmitLabel>cadastrar</SubmitLabel>
4819             <OnSubmit>{this.handlerSubmit}</OnSubmit>
4820             <div>
4821                 <h3>Dados do evento</h3>
4822                 <Row>
4823                     <Col>
4824                         <Input id="type" type="select"
4825                             label="Tipo de Evento"
4826                             invalidMessage="Tipo de
4827                             Evento obrigatorio"
4828                             value={type.type}
4829                             onChange={this.handlerType}
4830                             required>
4831                         <option
4832                             value="">Selecione</option>
4833                         {
4834                             typesEvent.map(
4835                                 typeEvent => <option
4836                                     key={typeEvent.type}
4837                                     value={typeEvent.type}>{typeEvent
4838                                         .name}</option>
4839                             )
4840                         )
4841                     </Input>
4842                 </Col>
4843                 <Col md="3">
4844                     <Switch id="evaluate"
4845                         label="Avaliado"
4846                         value={event.evaluate}
4847                         onChange={this.handlerEvaluate}
4848                         />
4849                 </Col>
4850             </Row>
4851             <Input id="filter" type="select"
4852                 label="Escola" value={codSchool}>
```

```
        onChange={this.handlerSelectSchool}>
4833      <option>Selecione</option>
4834    {
4835      schools.map(school => <option
4836        key={school.codSchool}
4837        value={school.codSchool}>{school.name}
4838      )
4839    </Input>
4840    <Input id="city" label="Cidade"
4841      invalidMessage="Cidade
4842      obrigatorio" value={address.city}
4843      disabled={!codSchool}
4844      onChange={this.handlerCity}
4845      required />
4846    <Input id="neighborhood"
4847      label="Bairro"
4848      invalidMessage="Bairro
4849      obrigatorio"
4850      value={address.neighborhood}
4851      disabled={!codSchool}
4852      onChange={this.handlerNeighborhood}
4853      required />
4854  <Row>
4855    <Col>
4856      <Input id="street"
4857        label="Lougradouro"
4858        invalidMessage="Lougradouro
4859        obrigatorio"
4860        value={address.street}
4861        disabled={!codSchool}
4862        onChange={this.handlerStreet}
4863        required />
4864    </Col>
4865    <Col md="3">
4866      <Input id="number"
4867        label="Nmero"
4868        value={address.number}
4869        onChange={this.handlerNumber}
4870        disabled={!codSchool} />
4871    </Col>
4872  </Row>
4873  <hr className="row" />
4874</div>
4875<div>
4876  <h3>Participantes</h3>
4877</div>
```

```
4852 <Input id="type" type="select"
4853   label="Selecione"
4854   invalidMessage="Tipo de Evento
4855   obrigatorio"
4856   onChange={this.handlerParticipants}
4857   style={{ height: '500px' }}
4858   multiple required
4859   disabled={!codSchool} >
4860   {
4861     volunteers.map(
4862       volunteer => {
4863         const { user } =
4864           volunteer;
4865         return (
4866           <option
4867             key={user.cpf}
4868             value={user.cpf}>{formatter.cpf(u
4869             -
4870             {user.name})}</option>
4871           );
4872         }
4873       )
4874     }
4875   )
4876 </Input>
4877 <hr className="row" />
4878 </div>
4879 <div>
4880   <h3>Materias</h3>
4881   {
4882     trainingsOfType.length?
4883       trainingsOfType.map(
4884         training => (
4885           <Col
4886             key={training.codTraining}>
4887             {training.link &&
4888               <a
4889                 className="btn
4890                 btn-info"
4891                 href={training.link}
4892                 target="_blank">{training.nam
4893               </a>
4894             )
4895           )
4896         )
4897       : (
4898         <div>
```

```
4878                                     <strong>Para esse tipo  
4879                                         de evento no h  
4880                                         materiais prviamente  
4881                                         cadastrados</strong>  
4882                                 </div>  
4883                             )  
4884                         }  
4885                     <hr className="row" />  
4886                 </div>  
4887             <div>  
4888                 <h3>Materiais Extra</h3>  
4889             {  
4890                 trainings.map((training, index) => (  
4891                     <Fragment>  
4892                         <Row>  
4893                             <Col>  
4894                             <Input  
4895                               id={'training-${index}'}  
4896                               label="Nome"  
4897                               invalidMessage="Nome  
4898                                   Obrigatrio"  
4899                               value={training.name}  
4900                               onChange={event  
4901                                   =>  
4902                                   this.handlerNameAttachemnt  
4903                                       index)}  
4904                               required/>  
4905                         </Col>  
4906                         <Col md="5">  
4907                             <Switch  
4908                               id={'file-${index}'}  
4909                               label="Arquivo  
4910                                   pra Upload"  
4911                               value={training.isFile}  
4912                               onChange={value  
4913                                   =>  
4914                                   this.handlerIsFile(value,  
4915                                       index)} />  
4916                         </Col>  
4917                     </Row>  
4918                 <Row>  
4919                     <Col>  
4920                     {  
4921                         training.isFile ?  
4922                             (
```

4902

```

<FileInput
    id={'upload-${index}'}
    label="Anexar
    arquivo"
    invalidMessage="Upload

    Obrigatorio"
    onChange={event
    =>
    this.handlerUploadAtta
    index)}
    accept="application/pd
    required/>

) : (
<Input
    id={'link-${index}'}
    label="Link"
    invalidMessage="Link

    Obrigatorio"
    value={training.link}
    onChange={event
    =>
    this.handlerLinkAttach
    index)}
    required/>
)
}
</Col>
<Col md="3">
{!training.isFile &&
    <label>&nbsp;</label>
    <Button outline
        color="secondary"
        onClick={() =>
            this.handlerRemove(index)}>Remove
    </Col>
    </Row>
    </Fragment>
))
}
<Row>
    <Col>
        <Button outline
            color="secondary"
```

4903

4904

4905

4906

4907

4908

4909

4910

4911

4912

4913

4914

4915

4916

4917

4918

```
        onClick={this.handlerAdd}>Incluir</Button>
4919      </Col>
4920    </Row>
4921    <hr className="row" />
4922  </div>
4923  </Wizard>
4924  </ModalBody>
4925  </Modal>
4926  );
4927 }
4928 }
4929
4930 webapp/src/pages/protected/dashboard/admin/event/form/index.js
4931
4932 import Form from './Form.jsx';
4933 export default Form;
4934
4935 webapp/src/pages/protected/dashboard/admin/event/form/form.css
4936
4937 .modal-content {
4938   width: inherit; }
4939
4940
4941 webapp/src/pages/protected/dashboard/admin/event/form/form.sass
4942
4943 .modal-content
4944   width: inherit
4945
4946 webapp/src/pages/protected/dashboard/admin/event/list.css
4947
4948 .rbc-day-bg, .rbc-events-container {
4949   cursor: pointer; }
4950
4951
4952 webapp/src/pages/protected/dashboard/admin/event/index.js
4953
4954 import List from './List.jsx';
4955 export default List;
4956
4957 webapp/src/pages/protected/dashboard/admin/event/Details.jsx
4958
4959 import React, { Component } from 'react';
4960 import PropTypes from 'prop-types';
4961 import { getRequest, deleteRequest } from
4962   '../../../../../utils/http';
```

```
4962 import { Address } from '../../../../../model';
4963 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
4964     Row, Col } from 'reactstrap';
4965
4966 export default class extends Component{
4967     constructor(){
4968         super();
4969         this.state = {
4970             eventDetails : null
4971         }
4972         this.handlerDelete = this.handlerDelete.bind(this);
4973     }
4974
4975     static propTypes = {
4976         code: PropTypes.number.isRequired,
4977         isOpen: PropTypes.bool.isRequired,
4978         close: PropTypes.func.isRequired,
4979         afterDelete: PropTypes.func.isRequired,
4980     };
4981
4982     componentWillMount() {
4983         const { code } = this.props;
4984         getRequest(
4985             '/event/${code}',
4986             res => {
4987                 let eventDetails = res.data;
4988                 eventDetails.title = eventDetails.type.name + ' -
4989                     ' + new
4990                     Date(eventDetails.startOccurrence).toLocaleString();
4991                     this.setState({ eventDetails });
4992             }
4993         );
4994     }
4995
4996     handlerDelete() {
4997         const { code, afterDelete} = this.props;
4998         deleteRequest('/event/${code}', afterDelete);
4999     }
5000
5001     render(){
5002         const { isOpen, close } = this.props;
5003         const { eventDetails } = this.state;
5004         if (eventDetails){
```

```

5003     const { address: eventAddress, trainings:
5004         eventTrainings, participants, type } =
5005         eventDetails;
5006     const { trainings: typeTrainings } = type;
5007     const trainings =
5008         eventTrainings.concat(typeTrainings);
5009     const duration = Math.ceil(
5010         Math.abs(
5011             eventDetails.startOccurrence -
5012                 eventDetails.endOccurrence
5013             ) / (1000 * 3600)
5014     );
5015     const address = new Address();
5016     Object.assign(address, eventAddress);
5017     return (
5018         <Modal toggle={close} isOpen={isOpen} >
5019             <ModalHeader
5020                 toggle={close}>{eventDetails.title}</ModalHeader>
5021             <ModalBody>
5022                 <Row>
5023                     <Col>
5024                         <strong> Durao :</strong>
5025                             {eventDetails.allDay ? 'Dia
5026                             Inteiro' : duration + ' horas
5027                             (aproximadamente)'}
5028                     </Col>
5029                 </Row>
5030                 <br />
5031                 <Row>
5032                     <Col>
5033                         <strong>Endereço:</strong>
5034                     </Col>
5035                 </Row>
5036                 <br />
5037                 <Row>
5038                     <Col>
5039                         <strong>Participantes:</strong>

```

```

5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
{
    participants.map(
        participant => (
            <Col md="4"
                style={{marginTop:
                    '20px'}}>
                {participant.cpf} -
                {participant.name}
            </Col>
        )
    )
}
</Row>
<br />
<Row>
    <Col>
        <strong>Materiais em anexo:</strong>
    </Col>
</Row>
<Row>
    <Col>
        {
            trainings.length ?
            (
                <Row>
                {
                    trainings.map(
                        training => (
                            <Col
                                key={training.codTraining}
                                {training.link
                                    && <a
                                    className="btn
                                    btn-info"
                                    href={training.link}
                                    target="_blank">{train
}
</Col>
                )
            )
        }
    </Row>
    ): 'Nenhum anexo para esse
    evento.'
}
</Col>

```

```
5075     </Row>
5076   </ModalBody>
5077   <ModalFooter style={{display: 'inline'}}>
5078     <Row>
5079       <Col>
5080         <Button color="danger"
5081             onClick={this.handlerDelete}>Excluir</But
5082       </Col>
5083       <Col>
5084         <Button color="primary"
5085             className="float-right"
5086             onClick={close}>Fechar</Button>
5087       </Col>
5088     </Row>
5089   </ModalFooter>
5090 </Modal>
5091 )
5092 } else {
5093   return (
5094     <Modal toggle={close} isOpen={isOpen} ></Modal>
5095   );
5096 }
5097 }
5098
5099 import React, { Component } from 'react';
5100 import { Calendar, Input } from '../../../../../components';
5101 import { Row, Col, Form } from 'reactstrap';
5102 import BigCalendar from 'react-big-calendar';
5103 import FormEvent from './form';
5104 import Details from './Details.jsx';
5105 import { Event } from '../../../../../model';
5106 import { getRequest } from '../../../../../utils/http';
5107 import axios from 'axios';
5108 import './list.css';
5109
5110 const mapEvents =
5111   events => events.map(
5112     event =>{
5113       event.end = new Date(event.endOccurrence);
5114       event.start = new Date(event.startOccurrence);
5115       event.title = event.type.name + ' - ' +
5116         event.start.toLocaleString()
```

```

5116         return event
5117     }
5118 );
5119
5120 export default class extends Component{
5121     constructor(){
5122         super();
5123         this.state = {
5124             events: [],
5125             projects: [],
5126             isOpenForm: false,
5127             isOpenDetails: false,
5128             newEvent: new Event(),
5129             codEvent: null,
5130             codProject: ''
5131         };
5132         this.closeForm = this.closeForm.bind(this);
5133         this.closeDetails = this.closeDetails.bind(this);
5134         this.handlerSelectSlot =
5135             this.handlerSelectSlot.bind(this);
5136         this.handlerSelectEvent =
5137             this.handlerSelectEvent.bind(this);
5138         this.handlerSelectProject =
5139             this.handlerSelectProject.bind(this);
5140         this.componentWillMount =
5141             this.componentWillMount.bind(this);
5142     }
5143
5144     componentWillMount(){
5145         const { filterBySchool, filterByVolunteer } = this.props;
5146         axios.all([
5147             getRequest(
5148                 '/event?cod_cpf=${filterByVolunteer.join(',')}&cod_school=${filterBySchool}'
5149                 , res => res.data
5150             ),
5151             getRequest('/project', res => res.data)
5152         ]).then(
5153             res => this.setState({
5154                 events: mapEvents(res[0]),
5155                 projects: res[1],
5156                 isOpenForm: false,
5157                 newEvent: new Event(),
5158                 isOpenDetails: false,
5159                 codEvent: null
5160             })
5161     }

```

```
5157         )
5158     }
5159
5160     closeForm(){
5161         this.setState({isOpenForm: false, newEvent: new
5162             Event()});
5163     }
5164
5165     closeDetails(){
5166         this.setState({ isOpenDetails: false, codEvent: null });
5167     }
5168
5169     handlerSelectSlot({ start, end }){
5170         if(start < new Date()){
5171             return;
5172         }
5173         const { newEvent } = this.state;
5174         newEvent.startOccurrence = start;
5175         newEvent.endOccurrence = end;
5176         this.setState({isOpenForm: true, newEvent});
5177     }
5178
5179     handlerSelectEvent(event){
5180         const { codEvent } = event
5181         this.setState({ isOpenDetails: true, codEvent });
5182     }
5183
5184     handlerSelectProject(event){
5185         const { value } = event.target;
5186         const { filterBySchool, filterByVolunteer } = this.props;
5187         getRequest(
5188             `/event?cod_cpf=${filterByVolunteer.join(',')}&cod_school=${filterBySchool}
5189             res => this.setState({ events: mapEvents(res.data),
5190                 codProject: value})
5191         );
5192     }
5193
5194     render(){
5195         const { events, projects, isOpenForm, isOpenDetails,
5196             newEvent, codEvent, codProject } = this.state;
5197         return (
5198             <Row>
5199                 <Col>
5200                     <Row>
5201                         <Col>
```

```

5199 <Form inline>
5200   <Input id="filter" type="select"
5201     value={codProject}
5202     label="Projeto:"
5203     onChange={this.handlerSelectProject}>
5204   <option
5205     value="">Selecione</option>
5206   {
5207     projects.map(project =>
5208       <option
5209         key={project.codProject}
5210         value={project.codProject}>{project.name}
5211       }
5212     </Input>
5213   </Form>
5214 </Col>
5215 </Row>
5216 <br />
5217 <Row style={{ height: '600px', marginBottom:
5218   '1%' }}>
5219   <Col>
5220     <Calendar events={events}
5221       onSelectEvent={this.handlerSelectEvent}
5222       onSelectSlot={this.handlerSelectSlot}
5223       selectable
5224       defaultView={BigCalendar.Views.WEEK}
5225     />
5226   {
5227     isOpenForm ?
5228       <FormEvent isOpen={isOpenForm}
5229         close={this.closeForm}
5230         event={newEvent}
5231         afterSubmit={this.componentWillMount}
5232         /> :
5233       null
5234     }
5235   {
5236     isOpenDetails ?
5237       <Details code={codEvent}
5238         isOpen={isOpenDetails}
5239         close={this.closeDetails}
5240         afterDelete={this.componentWillMount}
5241         /> :
5242       null
5243   }

```

```
5223             </Col>
5224         </Row>
5225     </Col>
5226     </Row>
5227 );
5228 }
5229 }
5230
5231 webapp/src/pages/protected/dashboard/admin/event/list.sass
5232
5233 .rbc-day-bg, .rbc-events-container
5234     cursor: pointer
5235
5236 webapp/src/pages/protected/dashboard/admin/company/Form.jsx
5237
5238 import React, { Component } from 'react';
5239 import Company from '../../../../../model/company';
5240 import PropTypes from 'prop-types';
5241 import { postRequest } from '../../../../../utils/http';
5242 import { Row, Col, Modal, ModalHeader, ModalBody } from
5243     'reactstrap';
5244 import { Wizard, Input, MaskInput, DatePicker } from
5245     '../../../../../components';
5246
5247 export default class extends Component{
5248     constructor(){
5249         super();
5250         this.state = {
5251             company: new Company(),
5252             confirmPassword: ''
5253         };
5254         this.handlerName = this.handlerName.bind(this);
5255         this.handlerPhone = this.handlerPhone.bind(this);
5256         this.handlerCnpj = this.handlerCnpj.bind(this);
5257         this.handlerNameResponsible =
5258             this.handlerNameResponsible.bind(this);
5259         this.handlerCpf = this.handlerCpf.bind(this);
5260         this.handlerPhoneResponsible =
5261             this.handlerPhoneResponsible.bind(this);
5262         this.handlerJob = this.handlerJob.bind(this);
5263         this.handlerEmail = this.handlerEmail.bind(this);
5264         this.handlerBirth = this.handlerBirth.bind(this);
5265         this.handlerPassword = this.handlerPassword.bind(this);
5266         this.handlerConfirmPassword =
5267             this.handlerConfirmPassword.bind(this);
```

```
5263     this.handlerCity = this.handlerCity.bind(this);
5264     this.handlerNeighborhood =
5265         this.handlerNeighborhood.bind(this);
5266     this.handlerStreet = this.handlerStreet.bind(this);
5267     this.handlerNumber = this.handlerNumber.bind(this);
5268     this.handlerSubmit = this.handlerSubmit.bind(this);
5269 }
5270
5271     static propTypes = {
5272         afterSubmit: PropTypes.func.isRequired,
5273         isOpen: PropTypes.bool.isRequired,
5274         close: PropTypes.func.isRequired
5275     }
5276
5277 //company
5278 handlerName(event){
5279     const { company } = this.state;
5280     company.name = event.target.value;
5281     this.setState({ company });
5282 }
5283
5284 handlerCnpj(event){
5285     const { company } = this.state;
5286     company.cnpj = event.target.value.replace(/[-]/g, '');
5287     this.setState({ company });
5288 }
5289
5290 handlerPhone(event){
5291     const { company } = this.state;
5292     company.phone = event.target.value.replace(/[()]/g, '');
5293     this.setState({ company });
5294 }
5295
5296 // responsible
5297 handlerNameResponsible(event){
5298     const { company } = this.state;
5299     const { responsible } = company;
5300     responsible.name = event.target.value;
5301     company.responsible = responsible;
5302     this.setState({ company });
5303 }
5304
5305 handlerCpf(event){
5306     const { company } = this.state;
```

```
5307     const { responsible } = company;
5308     responsible.cpf = event.target.value.replace(/[-]/g,
5309         '');
5310     company.responsible = responsible;
5311     this.setState({ company });
5312 }
5313
5314 handlerPhoneResponsible(event) {
5315     const { company } = this.state;
5316     const { responsible } = company;
5317     responsible.phone = event.target.value.replace(/[()]/g,
5318         '');
5319     company.responsible = responsible;
5320     this.setState({ company });
5321 }
5322
5323 handlerJob(event){
5324     const { company } = this.state;
5325     const { responsible } = company;
5326     responsible.job = event.target.value;
5327     company.responsible = responsible;
5328     this.setState({ company });
5329 }
5330
5331 handlerEmail(event){
5332     const { company } = this.state;
5333     const { responsible } = company;
5334     responsible.email = event.target.value;
5335     company.responsible = responsible;
5336     this.setState({ company });
5337 }
5338
5339 handlerBirth(event){
5340     const { company } = this.state;
5341     const { responsible } = company;
5342     const { value } = event.target;
5343     responsible.birth = value ? new Date(value) : null;
5344     company.responsible = responsible;
5345     this.setState({ company });
5346 }
5347
5348 handlerPassword(event){
5349     const { company } = this.state;
5350     const { responsible } = company;
5351     responsible.password = event.target.value;
```

```
5350     company.responsible = responsible;
5351     this.setState({ company });
5352 }
5353
5354     handlerConfirmPassword(event){
5355         let { confirmPassword } = this.state;
5356         confirmPassword = event.target.value;
5357         this.setState({ confirmPassword });
5358 }
5359
5360 //address
5361     handlerCity(event){
5362         const { company } = this.state;
5363         const { responsible, address } = company;
5364         address.city = event.target.value;
5365         company.address = responsible.address = address;
5366         company.responsible = responsible;
5367         this.setState({ company });
5368 }
5369
5370     handlerNeighborhood(event){
5371         const { company } = this.state;
5372         const { responsible, address } = company;
5373         address.neighborhood = event.target.value;
5374         company.address = responsible.address = address;
5375         company.responsible = responsible;
5376         this.setState({ company });
5377 }
5378
5379     handlerStreet(event){
5380         const { company } = this.state;
5381         const { responsible, address } = company;
5382         address.street = event.target.value;
5383         company.address = responsible.address = address;
5384         company.responsible = responsible;
5385         this.setState({ company });
5386 }
5387
5388     handlerNumber(event){
5389         const { company } = this.state;
5390         const { responsible, address } = company;
5391         address.number = event.target.value;
5392         company.address = responsible.address = address;
5393         company.responsible = responsible;
5394         this.setState({ company });
5395 }
```



```
        obrigatorio" value={company.phone}
        mask="(99)9999-9999"
        onChange={this.handlerPhone}
        required/>
    <hr className="row"/>
</div>
<div>
    <h3>Dados do responsvel</h3>
    <Input id="nameReponsible" label="Nome
        Completo" invalidMessage="Nome
        Completo obrigatorio"
        value={responsible.name}
        onChange={this.handlerNameResponsible}
        required/>
    <MaskInput id="cpf" label="CPF"
        invalidMessage="CPF obrigatorio"
        value={responsible.cpf}
        mask="999.999.999-99"
        onChange={this.handlerCpf}
        required />
    <MaskInput id="phoneResponsible"
        label="Contato do Responsvel"
        invalidMessage="Contato do
        Responsvel obrigatorio"
        value={responsible.phone}
        mask="(99)9999-9999"
        onChange={this.handlerPhoneResponsible}
        required/>
    <Input id="job" label="Cargo do
        Responsvel" invalidMessage="Cargo
        do Responsvel obrigatorio"
        value={responsible.job}
        onChange={this.handlerJob}
        required/>
    <DatePicker id="birth"
        label="Nascimento"
        invalidMessage="Nascimento
        obrigatorio" max={maxDate}
        onChange={this.handlerBirth}
        required/>
    <Input id="email" type="email"
        label="Email"
        invalidMessage="Email obrigatorio"
        value={responsible.email}
        onChange={this.handlerEmail}>
```

```
        required/>
5435
5436
5437      <Row>
5438          <Col>
5439              <Input id="password"
5440                  type="password"
5441                  label="Senha"
5442                  invalidMessage="Senha
5443                      obrigatorio"
5444                  value={responsible.password}
5445                  onChange={this.handlerPassword}
5446                  required/>
5447          </Col>
5448      </Row>
5449      <hr className="row"/>
5450  </div>
5451  <div>
5452      <h3>Endereço da Empresa</h3>
5453      <Input id="city" label="Cidade"
5454          invalidMessage="Cidade
5455              obrigatorio" value={address.city}
5456          onChange={this.handlerCity}
5457          required />
5458      <Input id="neighborhood"
5459          label="Bairro"
5460          invalidMessage="Bairro
5461              obrigatorio"
5462          value={address.neighborhood}
5463          onChange={this.handlerNeighborhood}
5464          required />
5465  <Row>
5466      <Col>
5467          <Input id="street"
5468              label="Lougradouro"
5469              invalidMessage="Lougradouro
5470                  obrigatorio"
```

```
        value={address.street}
        onChange={this.handlerStreet}
        required />
5452    </Col>
5453    <Col md="3">
5454      <Input id="number"
5455        label="Número"
5456        value={address.number}
5457        onChange={this.handlerNumber}
5458      />
5459      </Col>
5460    </Row>
5461    <hr className="row"/>
5462  </div>
5463  </Wizard>
5464  </ModalBody>
5465</Modal>
5466);
5467}
5468}
5469
5470 webapp/src/pages/protected/dashboard/admin/company/index.js
5471
5472 import List from './List.jsx';
5473 export default List;
5474
5475 webapp/src/pages/protected/dashboard/admin/company/ListItem.jsx
5476
5477 import React, { Component } from 'react';
5478 import PropTypes from 'prop-types';
5479 import { deleteRequest } from '../../../../../utils/http';
5480 import { Checkbox } from '../../../../../components';
5481 import { Row, Col, ListGroupItem } from 'reactstrap';
5482 import { FaTrash } from 'react-icons/fa';
5483 import Details from './Details.jsx';
5484
5485 export default class extends Component{
5486   constructor(){
5487     super();
5488     this.state = {
5489       isOpen: false
5490     }
5491     this.handlerDelete = this.handlerDelete.bind(this);
5492     this.handlerDetails = this.handlerDetails.bind(this);
5493   }
5494 }
```

```
5490
5491     static propTypes = {
5492         company: PropTypes.object.isRequired,
5493         afterDelete: PropTypes.func.isRequired
5494     }
5495
5496     handlerDelete(){
5497         const { company, afterDelete } = this.props;
5498         deleteRequest(`/company/${company.cnpj}`, afterDelete);
5499     }
5500
5501     handlerDetails(){
5502         const { isOpen } = this.state;
5503         this.setState({ isOpen: !isOpen });
5504     }
5505
5506     render(){
5507         const { company, onSelectCompany } = this.props;
5508         const { isOpen } = this.state;
5509         return (
5510             <ListGroupItem key={company.cnpj}>
5511                 <Row>
5512                     <Col md="1">
5513                         <Checkbox value={company.cnpj}
5514                             onChange={onSelectCompany}/>
5515                     </Col>
5516                     <Col md="9" onClick={this.handlerDetails}
5517                         className="text-ellipsis" style={{
5518                             cursor: 'pointer' }}>
5519                         {company.name}
5520                     </Col>
5521                     <Col>
5522                         <FaTrash style={{ cursor: 'pointer' }}
5523                             color="red"
5524                             onClick={this.handlerDelete} />
5525                     </Col>
5526                 </Row>
5527             {
5528                 isOpen ? <Details cnpj={company.cnpj}
5529                     isOpen={isOpen}
5530                     close={this.handlerDetails}/> : null
5531             }
5532         );
5533     }
5534 }
```

```
5528     }
5529
5530     webapp/src/pages/protected/dashboard/admin/company/Add.jsx
5531
5532     import React, { Component } from 'react';
5533     import PropTypes from 'prop-types';
5534     import { Button } from 'reactstrap';
5535     import { FaPlus } from 'react-icons/fa';
5536     import Form from './Form.jsx';
5537
5538     export default class extends Component{
5539         constructor(){
5540             super();
5541             this.state = {
5542                 isOpen: false
5543             };
5544         }
5545
5546         static propTypes = {
5547             afterSubmit: PropTypes.func.isRequired
5548         }
5549
5550         handlerAdd(){
5551             const { isOpen } = this.state;
5552             this.setState({ isOpen: !isOpen });
5553         }
5554
5555         render(){
5556             const { isOpen } = this.state;
5557             const { afterSubmit } = this.props;
5558             return (
5559                 <div>
5560                     <Button type="button" color="primary"
5561                         onClick={this.handlerAdd.bind(this)}>
5562                         <FaPlus />
5563                     <Form isOpen={isOpen} afterSubmit={afterSubmit}
5564                         close={this.handlerAdd.bind(this)}/>
5565                 </div>
5566             );
5567         }
5568
5569     webapp/src/pages/protected/dashboard/admin/company/Details.jsx
5570
```

```
5571 import React, { Component } from 'react';
5572 import PropTypes from 'prop-types';
5573 import Address from '../../../../../model/address';
5574 import { getRequest } from '../../../../../utils/http';
5575 import formatter from '../../../../../utils/formatter';
5576 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
5577     Row, Col } from 'reactstrap';
5578 import pic from './pic.jpeg';
5579
5580 export default class extends Component {
5581     constructor(){
5582         super();
5583         this.state = {
5584             company: null
5585         }
5586     }
5587     static propTypes = {
5588         cnpj: PropTypes.string.isRequired,
5589         isOpen: PropTypes.bool.isRequired,
5590         close: PropTypes.func.isRequired
5591     };
5592     componentWillMount() {
5593         const { cnpj } = this.props;
5594         getRequest(`/company/${cnpj}`, res => this.setState({
5595             company: res.data }));
5596     }
5597     render() {
5598         const { isOpen, close } = this.props;
5599         const { company } = this.state;
5600         if (company) {
5601             const address = new Address();
5602             Object.assign(address, company.address);
5603             return (
5604                 <Modal toggle={close} isOpen={isOpen} >
5605                     <ModalHeader toggle={close}>{company.name} -
5606                         <small>{formatter.cnpj(company.cnpj)}</small></Modal
5607                         <ModalBody>
5608                             <Row>
5609                                 <Col md={1}>
5610                                     <img src={pic} alt="" className="rounded-circle"
5611                                         width="75px" />
5612                                 </Col>
```

```
5611 <Col md="2" />
5612 <Col>
5613 <Row>
5614
5615 <Col>
5616 <strong>Endereço:</strong>
5617 </Col>
5618 </Row>
5619 <Row>
5620 <Col>
5621 &{address.toString()}
5622 </Col>
5623 </Row>
5624 <br />
5625 <Row>
5626 <Col>
5627 <strong>Contato:</strong>
5628 </Col>
5629 </Row>
5630 <Row>
5631 <Col>
5632 &{company.responsible.email}
5633 /
5634 &{formatter.phone(company.phone)}
5635 </Col>
5636 </Row>
5637 <Col>
5638 <Row>
5639 <Col md="3" />
5640 <Col>
5641 <strong>Responsável:</strong>
5642 </Col>
5643 </Row>
5644 <Row>
5645 <Col md="3" />
5646 <Col>
5647 <div>&{company.responsible.name}</div>
5648 <small>&{formatter.cpf(company.responsible.cpf)}</small>
5649 </Col>
5650 </Row>
5651 </ModalBody>
5652 <ModalFooter>
```

```
5653             <Button color="primary"  
5654                 onClick={close}>Fechar</Button>  
5655         </ModalFooter>  
5656     );  
5657 } else {  
5658     return (  
5659         <Modal toggle={close} isOpen={isOpen} ></Modal>  
5660     );  
5661 }  
5662 }  
5663 }  
5664  
5665 webapp/src/pages/protected/dashboard/admin/company>List.jsx  
5666  
5667 import React, { Component } from 'react';  
5668 import { getRequest } from '../../../../../utils/http';  
5669 import { Row, Col, ListGroup, ListGroupItem, Form } from  
      'reactstrap';  
5670 import { Pagination, Filter } from '../../../../../components';  
5671 import CompanyItem from './ListItem.jsx';  
5672 import Add from './Add.jsx';  
5673  
5674 export default class extends Component {  
5675     constructor() {  
5676         super()  
5677         this.state = {  
5678             filter: '',  
5679             companies: [],  
5680             count: 0,  
5681             pages: 0  
5682         }  
5683         this.onChangePage = this.onChangePage.bind(this);  
5684         this.handlerFilter = this.handlerFilter.bind(this);  
5685         this.componentWillMount =  
              this.componentWillMount.bind(this)  
5686     }  
5687  
5688     componentWillMount() {  
5689         getRequest(  
5690             '/company/page/0',  
5691             res => {  
5692                 const page = res.data;  
5693                 this.setState({  
5694                     companies: page.data,
```

```
5695             count: page.count,
5696             pages: page.count / 5
5697         });
5698     );
5699 );
5700 }
5701
5702 handlerFilter(event) {
5703     const filter = event.target.value;
5704     if (filter.length >= 3 || filter.length === 0){
5705         getRequest(
5706             '/company/page/0?filter=${filter}',
5707             res => {
5708                 const page = res.data;
5709                 this.setState({
5710                     companies: page.data,
5711                     count: page.count,
5712                     pages: page.count / 5,
5713                     filter
5714                 });
5715             }
5716         );
5717     } else {
5718         this.setState({filter});
5719     }
5720 }
5721
5722 onChangePage(pageNumber){
5723     const { filter } = this.state;
5724     getRequest(
5725         '/company/page/${pageNumber}?filter=${filter.length > 3
5726             ? filter : ''}',
5727         res => {
5728             const page = res.data;
5729             this.setState({
5730                 companies: page.data,
5731                 count: page.count,
5732                 pages: page.count / 5
5733             });
5734         }
5735     );
5736 }
5737
5738 render() {
5739     const { companies, pages, count } = this.state;
```

```
5739     const { onSelectCompany } = this.props;
5740     return (
5741       <div>
5742         <Row>
5743           <Col>
5744             <h3>
5745               Empresas Envolvidas
5746             </h3>
5747           </Col>
5748           <Col md="2">
5749             <Add
5750               afterSubmit={this.componentWillMount}/>
5751           </Col>
5752         </Row>
5753         <Filter handlerFilter={this.handlerFilter}/>
5754       <Row>
5755         <Col>
5756           <ListGroup>
5757             {
5758               companies.length ?
5759                 companies.map(
5760                   company =>
5761                     <CompanyItem
5762                       onSelectCompany={onSelectCompany}
5763                         key={company.cnpj}
5764                           company={company}
5765                           afterDelete={this.componentWillMount}
5766                     ) : (
5767                       <ListGroupItem>
5768                         <strong>
5769                           Nenhuma Empresa
5770                             Cadastrada
5771                           </strong>
5772                             </ListGroupItem>
5773                       )
5774                     }
5775                   </ListGroup>
5776                 </Col>
5777               <Row>
5778                 <Pagination pages={pages} count={count}
5779                   onChangePage={this.onChangePage} />
5780               </div>
5781             );
5782           );
5783         );
5784       );
5785     );
5786   );
```

```
5777
5778  webapp/src/pages/protected/dashboard/admin/Page.jsx
5779
5780  import React, { Component, Fragment } from 'react';
5781  import ListCompany from './company';
5782  import ListVolunteer from './volunteer';
5783  import ListEvent from './event';
5784  import ListSchool from './school';
5785  import { Row, Col } from 'reactstrap';
5786
5787  const Hr = () => (
5788      <Fragment>
5789          <br />
5790          <hr />
5791          <br />
5792      </Fragment>
5793 );
5794
5795  export default class extends Component{
5796      constructor(){
5797          super();
5798          this.state = {
5799              filterByCompany: [],
5800              filterBySchool: [],
5801              filterByVolunteer: []
5802          }
5803          this.onSelectCompany = this.onSelectCompany.bind(this);
5804          this.onSelectSchool = this.onSelectSchool.bind(this);
5805          this.onSelectVolunteer =
5806              this.onSelectVolunteer.bind(this);
5807      }
5808
5809      onSelectCompany(event){
5810          const { filterByCompany } = this.state;
5811          const { value } = event.target;
5812          if(filterByCompany.includes(value)){
5813              filterByCompany.splice(filterByCompany.indexOf(value),
5814                  1);
5815          } else {
5816              filterByCompany.push(value);
5817          }
5818          this.setState({ filterByCompany });
5819      }
5820
5821      onSelectSchool(event) {
```

```
const { filterBySchool } = this.state;
const { value } = event.target;
if (filterBySchool.includes(value)) {
    filterBySchool.splice(filterBySchool.indexOf(value),
        1);
} else {
    filterBySchool.push(value);
}
this.setState({ filterBySchool });
}

onSelectVolunteer(event) {
    const { filterByVolunteer } = this.state;
    const { value } = event.target;
    if (filterByVolunteer.includes(value)) {
        filterByVolunteer.splice(filterByVolunteer.indexOf(value),
            1);
    } else {
        filterByVolunteer.push(value);
    }
    this.setState({ filterByVolunteer });
}

render(){
    const { filterByCompany, filterBySchool,
        filterByVolunteer } = this.state;
    return (
        <Row>
            <Col md="1" />
            <Col>
                <Row>
                    <Col />
                </Row>
                <Hr />
                <Row>
                    <Col>
                        <ListCompany
                            onSelectCompany={this.onSelectCompany}/>
                    </Col>
                    <Col md="1" />
                    <Col>
                        <ListVolunteer
                            onSelectVolunteer={this.onSelectVolunteer}
                            filterByCompany={filterByCompany}
                            filterBySchool={filterBySchool}/>
                    </Col>
                </Row>
            </Col>
        </Row>
    );
}
```

```
5858             </Col>
5859             <Col md="1" />
5860             <Col>
5861                 <ListSchool
5862                     onSelectSchool={this.onSelectSchool}
5863                     />
5864             </Col>
5865         </Row>
5866         <Hr />
5867         <Row>
5868             <Col>
5869                 <ListEvent
5870                     filterBySchool={filterBySchool}
5871                     filterByVolunteer={filterByVolunteer}/>
5872                     </Col>
5873             </Row>
5874         </Col>
5875     </Row>
5876 );
5877 }
5878 }
5879 import Page from './Page.jsx';
5880 export default Page;
5881
5882 webapp/src/pages/protected/dashboard/volunteer/training/index.js
5883
5884 import Step from './Step.jsx';
5885 export default Step;
5886
5887 webapp/src/pages/protected/dashboard/volunteer/training/Details.jsx
5888
5889 import React, { Component } from 'react';
5890 import PropTypes from 'prop-types';
5891 import { getRequest } from '../../../../../utils/http';
5892 import { Address } from '../../../../../model';
5893 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
5894             Row, Col } from 'reactstrap';
5895
5896 export default class extends Component{
5897     constructor(){
```

```

5898     super();
5899     this.state = {
5900       eventDetails : null
5901     }
5902   }
5903
5904   static propTypes = {
5905     code: PropTypes.number.isRequired,
5906     isOpen: PropTypes.bool.isRequired,
5907     close: PropTypes.func.isRequired
5908   };
5909
5910   componentWillMount() {
5911     const { code } = this.props;
5912     getRequest(
5913       `/event/${code}`,
5914       res => {
5915         let eventDetails = res.data;
5916         eventDetails.title = eventDetails.type.name + ' - '
5917         + new
5918           Date(eventDetails.startOccurrence).toLocaleString();
5919         this.setState({ eventDetails });
5920       }
5921     );
5922   }
5923
5924   render(){
5925     const { isOpen, close } = this.props;
5926     const { eventDetails } = this.state;
5927     if (eventDetails){
5928       const { address: eventAddress, trainings } =
5929         eventDetails;
5930       const duration = Math.ceil(
5931         Math.abs(
5932           eventDetails.startOccurrence -
5933             eventDetails.endOccurrence
5934           ) / (1000 * 3600)
5935         );
5936       const address = new Address();
5937       Object.assign(address, eventAddress);
5938       return (
5939         <Modal toggle={close} isOpen={isOpen} >
5940           <ModalHeader
5941             toggle={close}>{eventDetails.title}</ModalHeader>
5942           <ModalBody>

```

```
5938 <Row>
5939   <Col>
5940     <strong> Durao :</strong>
5941       {eventDetails.allDay ? 'Dia
5942         Inteiro' : duration + ' horas
5943         (aproximadamente)'}
5944     </Col>
5945   </Row>
5946   <br />
5947   <Row>
5948     <Col>
5949       <strong>Endereo:</strong>
5950     </Col>
5951   </Row>
5952   <Row>
5953     <Col>
5954       {address.toString())
5955     </Col>
5956   </Row>
5957   <br />
5958   <Row>
5959     <Col>
5960       <strong>Materiais em anexo:</strong>
5961     </Col>
5962   </Row>
5963   <Row>
5964     <Col>
5965     {
5966       trainings.length ?
5967         (
5968           <Row>
5969             {
5970               trainings.map(
5971                 training => (
5972                   <Col
5973                     key={training.codTrain
5974                     {training.link
5975                       &&
5976                         <a
5977                           className="btn
5978                             btn-info"
5979                           href={training.link
5980                           target="_blank">{t
5981                     )
5982                   )
5983                 )
5984               )
5985             )
5986           )
5987         )
5988       )
5989     )
5990   )
5991 )
5992 )
```



```
6014     }
6015 );
6016
6017 export default class extends Component{
6018     constructor(){
6019         super();
6020         this.state = {
6021             events: [],
6022             isOpenDetails: false,
6023             codEvent: null
6024         };
6025         this.closeDetails = this.closeDetails.bind(this);
6026         this.handlerSelectEvent =
6027             this.handlerSelectEvent.bind(this);
6028         this.componentWillMount =
6029             this.componentWillMount.bind(this);
6030     }
6031
6032     componentWillMount(){
6033         const { cpf } = this.props;
6034         getRequest(
6035             '/volunteer/${cpf}/event/',
6036             res => this.setState({
6037                 events: mapEvents(res.data),
6038                 isOpenDetails: false,
6039                 codEvent: null
6040             })
6041         );
6042     }
6043
6044     closeDetails(){
6045         this.setState({ isOpenDetails: false, codEvent: null });
6046     }
6047
6048     handlerSelectEvent(event){
6049         const { codEvent } = event
6050         this.setState({ isOpenDetails: true, codEvent });
6051     }
6052
6053     render(){
6054         const { events, isOpenDetails, codEvent } = this.state;
6055         return (
6056             <Row style={{ height: '600px', marginBottom: '1%' }}>
6057                 <Col>
```

```

6056             <Calendar events={events}>
6057                 onSelectEvent={this.handlerSelectEvent}
6058                 defaultView={BigCalendar.Views.WEEK}/>
6059             {
6060                 isOpenDetails ?
6061                     <Details code={codEvent}>
6062                         isOpen={isOpenDetails}
6063                         close={this.closeDetails}
6064                         afterDelete={this.componentWillMount}/>
6065                         :
6066                         null
6067                     }
6068             </Col>
6069         </Row>
6070     );
6071 }
6072 }
6073
6074 webapp/src/pages/protected/dashboard/volunteer/statement/Attachment.jsx
6075
6076 import React, { Component } from 'react';
6077 import { Row, Col, Button } from 'reactstrap';
6078 import { ClipLoader } from 'react-spinners';
6079 import { FileInput } from '../../../../../components';
6080 import { postRequest, getRequest } from
6081     '../../../../../utils/http';
6082 import { saveAs } from 'file-saver/FileSaver';
6083 import axios from 'axios';
6084
6085 export class AttachmentUploader extends Component{
6086     constructor(){
6087         super();
6088         this.state = {
6089             loading: false,
6090             success: false
6091         }
6092         this.upload = this.upload.bind(this);
6093     }
6094
6095     componentWillMount(){
6096         const { cpf, attachment } = this.props;
6097         const { codAttachment } = attachment;
6098         getRequest(`/volunteer/${cpf}/attachment/${codAttachment}`, {
6099             res => this.setState({success: res.data.send})
6100         })
6101     }

```

```
6093     upload(event){
6094         const file = event.target.files[0];
6095         const { attachment, cpf, afterUpload } = this.props;
6096         if (file){
6097             this.setState({loading: true});
6098             const json = JSON.stringify(attachment);
6099             const blob = new Blob([json], {
6100                 type: 'application/json'
6101             });
6102             let form = new FormData();
6103             form.append('attachment', blob);
6104             form.append('file', file);
6105             postRequest(
6106                 '/volunteer/${cpf}/upload',
6107                 form,
6108                 res => {
6109                     afterUpload(attachment.codAttachment);
6110                     this.setState({
6111                         loading: false,
6112                         success: true
6113                     });
6114                 },
6115                 () => this.setState({ loading: false, success:
6116                     false })
6117             );
6118         }
6119     }
6120
6121     render(){
6122         const { attachment } = this.props;
6123         const { loading, success } = this.state;
6124         return (
6125             <Row>
6126                 <Col md="9">
6127                     <FileInput color={success ? 'success' :
6128                         'info'} label={attachment.name}
6129                         onChange={this.upload}
6130                         accept="application/pdf"
6131                         disabled={loading || success}/>
6132                 </Col>
6133                 <Col md="2">
6134                     <ClipLoader sizeUnit="px" size={30}
6135                         color="#007bff" loading={loading} />
6136                 </Col>
6137             
```

```
6132             </Row>
6133         );
6134     }
6135 }
6136
6137 export class AttachmentDownloader extends Component{
6138
6139     handlerDownload(){
6140         const { codAttachment, name } = this.props.attachment;
6141         axios('/attachment/${codAttachment}/download', {
6142             method: 'GET',
6143             responseType: 'blob' //Force to receive data in a
6144             Blob Format
6145         }).then(res =>{
6146             const file = new Blob(
6147                 [res.data],
6148                 {type: 'application/pdf'}
6149             );
6150             saveAs(file, `${name}.pdf`);
6151         });
6152     }
6153
6154     render(){
6155         const { attachment } = this.props;
6156         return (
6157             <Row>
6158                 <Col md="9">
6159                     <Button
6160                         onClick={this.handlerDownload.bind(this)}>{attachmen
6161                     </Col>
6162                 </Row>
6163             );
6164         }
6165     webapp/src/pages/protected/dashboard/volunteer/statement/index.js
6166
6167     import Step from './Step.jsx';
6168     export default Step;
6169
6170     webapp/src/pages/protected/dashboard/volunteer/statement/Step.jsx
6171
6172     import React, { Component } from 'react';
6173     import { Row, Col } from 'reactstrap';
```

```
6174 import { AttachmentUploader, AttachmentDownloader } from
6175   './Attachment.jsx';
6176 import { getRequest } from '../../../../../utils/http';
6177 import { Status } from '../../../../../model';
6178
6179 export default class extends Component{
6180   constructor() {
6181     super();
6182     this.state = {
6183       attachments: []
6184     }
6185   }
6186   componentWillMount() {
6187     getRequest('/attachment/${Status.WAIT_STATEMENT}', res
6188       => this.setState({ attachments: res.data }));
6189   }
6190   render(){
6191     const { attachments } = this.state;
6192     const { cpf, afterUpload } = this.props;
6193     return (
6194       <Row style={{textAlign: 'center'}}>
6195         <Col>
6196           <Row>
6197             <Col>
6198               <a className="btn btn-info"
6199                 target="_blank" rel="noopener
6200                 noreferrer"
6201                 href="http://www.computacaonaescola.ufsc.br/wp-cont
6202                 Politica Protecao Criana
6203               </a>
6204             </Col>
6205           </Row>
6206           <br/>
6207           <Row>
6208             {
6209               attachments.map(
6210                 attachment => (
6211                   <Col
6212                     key={attachment.codAttachment}>
6213                     <Row>
6214                       <Col>
6215                         <AttachmentUploader
6216                           cpf={cpf}>
```

```
attachment={attachment}
afterUpload={afterUpload}
/>
6212         </Col>
6213     </Row>
6214     <Row>
6215         <Col>
6216             {attachment.download
6217                 &&
6218                 <AttachmentDownloader
6219                     attachment={attachment}/>
6220             </Col>
6221         </Row>
6222     </Col>
6223     )
6224   )
6225   </Row>
6226   )
6227 }
6228 }
6229
6230 webapp/src/pages/protected/dashboard/volunteer/register/Attachment.jsx
6231
6232 import React, { Component } from 'react';
6233 import { Row, Col } from 'reactstrap';
6234 import { ClipLoader } from 'react-spinners';
6235 import { FileInput } from '../../../../../components';
6236 import { postRequest, getRequest } from
6237   '../../../../../utils/http';
6238
6239 export default class extends Component{
6240   constructor(){
6241     super();
6242     this.state = {
6243       loading: false,
6244       success: false
6245     }
6246     this.upload = this.upload.bind(this);
6247   }
6248
6249   componentWillMount(){
6250     const { cpf, attachment } = this.props;
```

```
6250     const { codAttachment } = attachment;
6251     getRequest(`/volunteer/${cpf}/attachment/${codAttachment}`,
6252       res => this.setState({success: res.data.send}))
6253   }
6254
6255   upload(event){
6256     const file = event.target.files[0];
6257     const { attachment, cpf } = this.props;
6258     if (file){
6259       this.setState({loading: true});
6260       const json = JSON.stringify(attachment);
6261       const blob = new Blob([json], {
6262         type: 'application/json'
6263       });
6264       let form = new FormData();
6265       form.append('attachment', blob);
6266       form.append('file', file);
6267       postRequest(
6268         `/volunteer/${cpf}/upload`,
6269         form,
6270         res => {
6271           this.props.afterUpload(attachment.codAttachment);
6272           this.setState({
6273             loading: false,
6274             success: true
6275           });
6276           () => this.setState({ loading: false, success:
6277             false })
6278         );
6279       }
6280     }
6281     render(){
6282       const { attachment } = this.props;
6283       const { loading, success } = this.state;
6284       return (
6285         <Row>
6286           <Col md="9">
6287             <FileInput color={success ? 'success' :
6288               'info'} label={attachment.name}
6289                 onChange={this.upload}
6290                 accept="application/pdf"
6291                 disabled={loading || success}/>
6292           </Col>
6293         </Row>
6294       );
6295     }
6296   }
6297 }
```

```
6289         <Col md="2">
6290             <ClipLoader sizeUnit="px" size={30}
6291                 color="#007bff" loading={loading} />
6292         </Col>
6293     </Row>
6294   );
6295 }
6296
6297 webapp/src/pages/protected/dashboard/volunteer/register/index.js
6298
6299 import Step from './Step.jsx';
6300 export default Step;
6301
6302 webapp/src/pages/protected/dashboard/volunteer/register/Step.jsx
6303
6304 import React, { Component } from 'react';
6305 import { Row, Col } from 'reactstrap';
6306 import AttachmentUploader from './Attachment.jsx';
6307 import { getRequest } from '../../../../../utils/http';
6308 import { Status } from '../../../../../model';
6309
6310 export default class extends Component{
6311     constructor(){
6312         super();
6313         this.state = {
6314             attachments: []
6315         }
6316     }
6317
6318     componentWillMount(){
6319         getRequest('/attachment/${Status.REGISTER}', res =>
6320             this.setState({ attachments: res.data}));
6321     }
6322
6323     render(){
6324         const { attachments } = this.state;
6325         const { cpf, afterUpload } = this.props;
6326         if(cpf){
6327             return (
6328                 <Row>
6329                 {
6330                     attachments.map(
6331                         attachment => (
```

```
6331             <Col key= {  
6332                 attachment.codAttachment }>  
6333                 <AttachmentUploader cpf={cpf}  
6334                     attachment={attachment}  
6335                     afterUpload={afterUpload}/>  
6336             </Col>  
6337         )  
6338     )  
6339   )  
6340     </Row>  
6341   );  
6342 }  
6343  
6344 webapp/src/pages/protected/dashboard/volunteer/steps/index.js  
6345  
6346 import Steps from './RegisterSteps';  
6347 export default Steps;  
6348  
6349 webapp/src/pages/protected/dashboard/volunteer/steps/steps.css  
6350  
6351 .btn-circle {  
6352     width: 90px;  
6353     height: 90px;  
6354     padding: 6px 0px;  
6355     border-radius: 100% !important;  
6356     text-align: center;  
6357     font-size: 2rem !important;  
6358     line-height: 1.42857; }  
6359  
6360  
6361 webapp/src/pages/protected/dashboard/volunteer/steps/steps.sass  
6362  
6363 .btn-circle  
6364     width: 90px  
6365     height: 90px  
6366     padding: 6px 0px  
6367     border-radius: 100% !important  
6368     text-align: center  
6369     font-size: 2rem !important  
6370     line-height: 1.42857  
6371  
6372
```

```
6373 webapp/src/pages/protected/dashboard/volunteer/steps/RegisterSteps.jsx
6374
6375 import React from 'react';
6376 import { Row, Col, Button } from 'reactstrap';
6377 import { Status } from '../../../../../model';
6378 import { FaUser, FaCheck, FaFileAlt, FaBuilding, FaChalkboard }
6379     from 'react-icons/fa';
6380 import Arrow from 'react-arrow';
6381 import './steps.css';
6382
6383 const Step = ({userStatus, status, icon, successContition,
6384   label}) => (
6385   <Col>
6386     <Row>
6387       <Col>
6388         <Button color={ !successContition ? "primary" :
6389           'success' } size="lg" className="btn-circle"
6390             disabled={userStatus !== status}>
6391               {icon}
6392             </Button>
6393           </Col>
6394         </Row>
6395       <Row>
6396         <Col>
6397           {successContition ? label.replace('Aguardando ' ,
6398             '') : label}
6399         </Col>
6400       </Row>
6401     </Col>
6402   )
6403
6404
6405 const ArrowStep = ({successContition}) => {
6406   const color = successContition ? '#28a745' : '#6c757d';
6407   const style = {
6408     display: 'flex',
6409     alignItems: 'center',
6410     paddingBottom: '2%'
6411   };
6412   return (
6413     <Col md="1" style={style}>
6414       <Arrow direction="right" shaftWidth={30}
6415         shaftLength={80} headWidth={80} headLength={30}
6416         fill={color}/>
6417     </Col>
```

```

6411     )
6412 }
6413
6414 export default ({status}) => (
6415   <Row style={{ textAlign: 'center' }}>
6416     <Step userStatus={status} status={Status.REGISTER}
6417       icon={<FaUser />} successContition={Status.REGISTER
6418         !== status} label="Registro"/>
6419     <ArrowStep successContition={Status.REGISTER !==
6420       status}/>
6421     <Step userStatus={status} status={Status.WAIT_COMPANY}
6422       icon={<FaBuilding/>}
6423       successContition={! [Status.REGISTER,
6424         Status.WAIT_COMPANY].includes(status)}
6425       label="Aguardando Recomendao de empresa"/>
6426     <ArrowStep successContition={! [Status.REGISTER,
6427       Status.WAIT_COMPANY].includes(status)}/>
6428     <Step userStatus={status} status={Status.WAIT_STATEMENT}
6429       icon={<FaFileAlt/>}
6430       successContition={[Status.WAIT_TRAINING,
6431         Status.APPROVED].includes(status)} label="Aguardando
6432         Envio do Termo de Responsabilidade"/>
6433     <ArrowStep successContition={[Status.WAIT_TRAINING,
6434       Status.APPROVED].includes(status)}/>
6435     <Step userStatus={status} status={Status.WAIT_TRAINING}
6436       icon={<FaChalkboard/>}
6437       successContition={Status.APPROVED === status}
6438       label="Aguardando Treinamento"/>
6439     <ArrowStep successContition={Status.APPROVED ===
6440       status}/>
6441     <Step userStatus={status} status={Status.APPROVED}
6442       icon={<FaCheck/>} successContition={Status.APPROVED
6443         === status} label="Aprovado"/>
6444   </Row>
6445 );
6446
6447
6448 webapp/src/pages/protected/dashboard/volunteer/approved/Form.jsx
6449
6450 import React, { Component } from 'react';
6451 import Company from '../../../../../model/company';
6452 import PropTypes from 'prop-types';
6453 import { postRequest } from '../../../../../utils/http';
6454 import { Row, Col, Modal, ModalHeader, ModalBody } from
6455   'reactstrap';

```

```
6435 import { Wizard, Input, MaskInput, DatePicker } from
6436   ' ../../../../../../components';
6437
6438 export default class extends Component{
6439   constructor(){
6440     super();
6441     this.state = {
6442       evaluate: null
6443     };
6444     this.handlerSubmit = this.handlerSubmit.bind(this);
6445   }
6446
6447   static propTypes = {
6448     afterSubmit: PropTypes.func.isRequired,
6449     isOpen: PropTypes.bool.isRequired,
6450     close: PropTypes.func.isRequired
6451   }
6452
6453   handlerSubmit(){
6454     const { afterSubmit, close } = this.props;
6455     const { evaluate } = this.state;
6456     postRequest(
6457       '/company',
6458       evaluate,
6459       () => {
6460         afterSubmit();
6461         close();
6462       }
6463     );
6464   }
6465
6466   render(){
6467     const { close, isOpen } = this.props;
6468     const { evaluate } = this.state;
6469     return (
6470       <Modal isOpen={isOpen} toggle={close}>
6471         <ModalHeader toggle={close}>Avaliao</ModalHeader>
6472         <ModalBody>
6473           <Wizard onCancel={close}
6474             submitLabel="cadastrar"
6475             onSubmit={this.handlerSubmit}>
6476             <div>
6477               <h3></h3>
6478               <hr className="row"/>
6479             </div>
```

```
6477             </Wizard>
6478         </ModalBody>
6479     </Modal>
6480   );
6481 }
6482 }
6483
6484 webapp/src/pages/protected/dashboard/volunteer/approved/index.js
6485
6486 import Step from './Step.jsx';
6487 export default Step;
6488
6489 webapp/src/pages/protected/dashboard/volunteer/approved/Details.jsx
6490
6491 import React, { Component } from 'react';
6492 import PropTypes from 'prop-types';
6493 import { getRequest } from '../../../../../utils/http';
6494 import { Address } from '../../../../../model';
6495 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
6496   Row, Col } from 'reactstrap';
6497
6498 export default class extends Component{
6499   constructor(){
6500     super();
6501     this.state = {
6502       eventDetails : null
6503     }
6504   }
6505
6506   static propTypes = {
6507     code: PropTypes.number.isRequired,
6508     isOpen: PropTypes.bool.isRequired,
6509     close: PropTypes.func.isRequired
6510   };
6511
6512   componentWillMount() {
6513     const { code } = this.props;
6514     getRequest(
6515       '/event/${code}',
6516       res => {
6517         let eventDetails = res.data;
6518         eventDetails.title = eventDetails.type.name + ' - '
6519           + new
6520             Date(eventDetails.startOccurrence).toLocaleString();
```

```
6519         this.setState({ eventDetails });
6520     }
6521   );
6522 }
6523
6524 render(){
6525   const { isOpen, close } = this.props;
6526   const { eventDetails } = this.state;
6527   if (eventDetails){
6528     const { address: eventAddress, trainings } =
6529       eventDetails;
6530     const duration = Math.ceil(
6531       Math.abs(
6532         eventDetails.startOccurrence -
6533           eventDetails.endOccurrence
6534         ) / (1000 * 3600)
6535     );
6536     const address = new Address();
6537     Object.assign(address, eventAddress);
6538     return (
6539       <Modal toggle={close} isOpen={isOpen} >
6540         <ModalHeader
6541           toggle={close}>{eventDetails.title}</ModalHeader>
6542         <ModalBody>
6543           <Row>
6544             <Col>
6545               <strong> Durao :</strong>
6546               {eventDetails.allDay ? 'Dia
6547                 Inteiro' : duration + ' horas
6548                 (aproximadamente)'}
6549             </Col>
6550           </Row>
6551           <br />
6552           <Row>
6553             <Col>
6554               <strong>Endereo:</strong>
6555             </Col>
6556           </Row>
6557           <br />
6558           <Row>
```

```

6558             <Col>
6559                 <strong>Materiais em anexo:</strong>
6560             </Col>
6561         </Row>
6562     <Row>
6563         <Col>
6564             {
6565                 trainings.length ?
6566                     (
6567                         <Row>
6568                             {
6569                                 trainings.map(
6570                                     training => (
6571                                         <Col
6572                                             key={training.codTrain
6573                                             {training.link
6574                                                 &&
6575                                                 <a
6576                                                 className="btn
6577                                                 btn-info"
6578                                                 href={training.link
6579                                                 target="_blank">{t
6580                                         )
6581                                         )
6582                                         )
6583                                         }
6584                                         )
6585                                         )
6586                                         <ModalBody>
6587                                         <Row>
6588                                         <Col>
6589                                         <Button color="primary"
6590                                             className="float-right"
6591                                             onClick={close}>Fechar</Button>
6592                                         </Col>
6593                                         </Row>
6594                                     </ModalFooter>
6595                                 </Modal>
6596                             )
6597                         ) else {

```

```
6593         return (
6594             <Modal toggle={close} isOpen={isOpen} ></Modal>
6595         );
6596     }
6597 }
6598 }
6599
6600 webapp/src/pages/protected/dashboard/volunteer/approved/Evaluate.jsx
6601
6602 import React, { Component, Fragment } from 'react';
6603 import { Row, Col, Button } from 'reactstrap';
6604 import Form from './Form';
6605
6606 export default class extends Component{
6607     constructor(){
6608         super();
6609         this.state = {
6610             isOpen : false
6611         };
6612         this.toggle = this.toggle.bind(this);
6613     }
6614
6615     toggle(){
6616         const { isOpen } = this.state;
6617         this.setState({ isOpen: !isOpen });
6618     }
6619
6620     render(){
6621         const { evaluate, cpf, afterSubmit } = this.props;
6622         const { isOpen } = this.state
6623         return (
6624             <Fragment>
6625                 <Row>
6626                     <Col>
6627                         <Button color="primary" style={{ width:
6628                             'inherit', height: '40px' }}
6629                             onClick={this.toggle}>
6630                             {evaluate.title}
6631                         </Button>
6632                     </Col>
6633                 </Row>
6634                 {isOpen && <Form cpf={cpf}
6635                     afterSubmit={afterSubmit} isOpen={isOpen}
6636                     close={this.toggle}/>}
6637                 <br />
6638             
```

```

6634             </Fragment>
6635         )
6636     }
6637 }
6638
6639 webapp/src/pages/protected/dashboard/volunteer/approved/Step.jsx
6640
6641 import React, { Component,Fragment } from 'react';
6642 import { Calendar } from '../../../../../components';
6643 import { Row, Col, Label } from 'reactstrap';
6644 import BigCalendar from 'react-big-calendar';
6645 import Details from './Details';
6646 import Evaluate from './Evaluate';
6647 import { Volunteer } from '../../../../../model';
6648 import { getRequest } from '../../../../../utils/http';
6649 import axios from 'axios';
6650
6651 const mapEvents =
6652     events => events.map(
6653         event =>{
6654             event.end = new Date(event.endOccurrence);
6655             event.start = new Date(event.startOccurrence);
6656             event.title = event.type.name + ' - ' +
6657                 event.start.toLocaleString()
6658             return event
6659         }
6660     );
6661
6662 export default class extends Component{
6663     constructor(){
6664         super();
6665         this.state = {
6666             events: [],
6667             evaluates: [],
6668             isOpenDetails: false,
6669             codEvent: null,
6670             volunteer: new Volunteer()
6671         };
6672         this.closeDetails = this.closeDetails.bind(this);
6673         this.handlerSelectEvent =
6674             this.handlerSelectEvent.bind(this);
6675         this.componentWillMount =
6676             this.componentWillMount.bind(this);
6677     }
6678 }
```

```
componentWillMount(){
    const { cpf } = this.props;
    axios.all([
        getRequest('/volunteer/${cpf}', res => res.data),
        getRequest(
            '/volunteer/${cpf}/event',
            res => res.data
        )
    ]).then(
        res => {
            const events = mapEvents(res[1]);
            this.setState({
                volunteer: res[0],
                isOpenDetails: false,
                codEvent: null,
                evaluates: events.filter(event =>
                    event.evaluate),
                events
            });
        }
    );
}

getLabel(){
    const { ratings, rating } = this.state.volunteer;
    const lastRating = ratings[0];
    let className, text;
    if (lastRating){
        if (rating < 40 || lastRating.grade < 40){
            className = 'danger';
        } else if (rating < 70){
            className = 'warning';
        } else {
            className = 'success';
        }
    } else {
        className = 'secondary';
        text = 'Ainda no h avaliaes ';
    }

    return (
        <Row>
            <Col>
                <Label className={'btn btn-${{className}}'}
                    style={{ cursor: 'default', width:
```

```

6719           'inherit', height: '40px'}]}>
6720           {text}
6721       </Label>
6722     </Col>
6723   </Row>
6724 );
6725 }
6726
6727   closeDetails(){
6728     this.setState({ isOpenDetails: false, codEvent: null });
6729   }
6730
6731   handlerSelectEvent(event){
6732     const { codEvent } = event
6733     this.setState({ isOpenDetails: true, codEvent });
6734   }
6735
6736   render(){
6737     const { events, isOpenDetails, codEvent, evaluates,
6738           volunteer } = this.state;
6739     const { cpf } = volunteer.user;
6740     return (
6741       <Row style={{ height: '600px', marginBottom: '1%' }}>
6742         <Col>
6743           <Calendar events={events}
6744             onSelectEvent={this.handlerSelectEvent}
6745             defaultView={BigCalendar.Views.WEEK}>
6746             {
6747               isOpenDetails ?
6748                 <Details code={codEvent}
6749                   isOpen={isOpenDetails}
6750                   close={this.closeDetails}
6751                   afterDelete={this.componentWillMount}/>
6752                   :
6753                     null
6754               }
6755             </Col>
6756           <Col md="2">
6757             <h3 className="text-center">Avaliaes</h3>
6758             {this.getLabel()}
6759             <br/>
6760             {evaluates.length ? evaluates.map(evaluate =>
6761               <Evaluate evaluate={evaluate} cpf={cpf}
6762                 afterSubmit={()=> null}/>) : null}
6763           </Col>

```

```
6754         </Row>
6755     );
6756 }
6757 }
6758
6759 webapp/src/pages/protected/dashboard/volunteer/company/index.js
6760
6761 import Step from './Step.jsx';
6762 export default Step;
6763
6764 webapp/src/pages/protected/dashboard/volunteer/company/Step.jsx
6765
6766 import React, { Component } from 'react';
6767 import { Row, Col } from 'reactstrap';
6768 import { getRequest } from '../../../../../utils/http';
6769
6770 export default class extends Component{
6771     constructor(){
6772         super();
6773         this.state = {
6774             company: null
6775         }
6776     }
6777
6778     componentWillMount(){
6779         const { cnpj } = this.props;
6780         getRequest('/company/${cnpj}', res => this.setState({
6781             company: res.data }));
6782     }
6783
6784     render(){
6785         const { company } = this.state;
6786         if (company){
6787             const { responsible } = company;
6788             return (
6789                 <Row style={{textAlign: 'center'}}>
6790                     <Col>
6791                         <h2>
6792                             Aguarde resposta de sua empresa, para
6793                             mais informaes entre em contato com
6794                             <br/>
6795                             <strong>{responsible.name}</strong>
6796                         </h2>
6797                     </Col>
6798                 </Row>
6799 }
```

```
6797         );
6798     } else {
6799         return null;
6800     }
6801 }
6802 }
6803
6804 webapp/src/pages/protected/dashboard/volunteer/Page.jsx
6805
6806 import React, { Component } from 'react';
6807 import { Row, Col } from 'reactstrap';
6808 import RegisterSteps from './steps';
6809 import Register from './register';
6810 import Company from './company';
6811 import Statement from './statement';
6812 import Training from './training';
6813 import Approved from './approved';
6814 import { Volunteer, Status } from '../../model';
6815 import { getRequest, putRequest } from '../../utils/http';
6816
6817 export default class extends Component{
6818     constructor(){
6819         super();
6820         this.state = {
6821             volunteer: new Volunteer(),
6822             attachments: [],
6823             dispatches: []
6824         };
6825         this.completeUpload = this.completeUpload.bind(this);
6826     }
6827
6828     componentWillMount(){
6829         const { cpf } = this.props;
6830         getRequest('/volunteer/${cpf}', resVolunteer => {
6831             const volunteer = resVolunteer.data;
6832             const { user, dispatches } = volunteer;
6833             getRequest(
6834                 '/attachment/${user.status}',
6835                 res => {
6836                     const attachments = res.data;
6837                     dispatches.push(
6838                         ...
6839                         attachments.map(
6840                             attachment => {
6841                                 return {
```

```

6842                               codAttachment:
6843                               attachment.codAttachment,
6844                               required:
6845                               attachment.required,
6846                               send: false
6847                           }
6848                       )
6849                   )
6850               this.setState({
6851                   volunteer,
6852                   attachments,
6853                   dispatches
6854               });
6855           }
6856       );
6857   });
6858 }
6859
6860 completeUpload(codAttachment){
6861     let conclude = true
6862     const { dispatches, volunteer } = this.state;
6863     for (const dispatch of dispatches) {
6864         if (dispatch.codAttachment === codAttachment){
6865             dispatch.send = true
6866         }
6867     }
6868     for (const dispatch of dispatches) {
6869         conclude = dispatch.send || !dispatch.required
6870         if(!conclude){
6871             break;
6872         }
6873     };
6874     if (conclude){
6875         const confirm = window.confirm('Todos os anexo
6876         obrigatorios j foram submetidos, deseja finalizar
6877         esta etapa?');
6878         if (confirm){
6879             const { status } = volunteer.user;
6880             volunteer.user.status = Status.next(status);
6881             putRequest(
6882                 '/volunteer/${volunteer.user.cpf}',
6883                 volunteer,
6884                 res => this.setState({volunteer, dispatches})

```

```
        );
    }
} else {
    this.setState({ dispatches });
}
}

render(){
    const { volunteer } = this.state;
    const { user, company } = volunteer;
    return (
        <Row>
            <Col md="1"/>
            <Col>
                <Row>
                    <Col>
                        <RegisterSteps status={user.status}/>
                        <br/>
                        {user.status === Status.REGISTER &&
                            <Register cpf={user.cpf}
                            afterUpload={this.completeUpload}/>}
                        {user.status === Status.WAIT_COMPANY
                            && <Company cnpj={company.cnpj}/>}
                        {user.status === Status.WAIT_STATEMENT
                            && <Statement cpf={user.cpf}
                            afterUpload={this.completeUpload}/>}
                        {user.status === Status.WAIT_TRAINING
                            && <Training cpf={user.cpf} />}
                        {user.status === Status.APPROVED &&
                            <Approved cpf={user.cpf} />}
                    </Col>
                    </Row>
                </Col>
                <Col md="1"/>
            </Row>
        );
    }
}

webapp/src/pages/protected/dashboard/volunteer/index.js

import Page from './Page.jsx';
export default Page;

webapp/src/pages/protected/dashboard/school/volunteer/index.js
```

```
6921
6922 import List from './List.jsx';
6923 export default List;
6924
6925 webapp/src/pages/protected/dashboard/school/volunteer/ListItem.jsx
6926
6927 import React, { Component } from 'react';
6928 import { Row, Col, ListGroupItem } from 'reactstrap';
6929 import Details from './Details.jsx';
6930 import PropTypes from 'prop-types';
6931 import formatter from '../../../../../utils/formatter';
6932
6933 export default class extends Component{
6934     constructor(){
6935         super();
6936         this.state = {
6937             isOpen: false
6938         }
6939         this.toggle = this.toggle.bind(this);
6940     }
6941
6942     static propTypes = {
6943         user: PropTypes.object.isRequired
6944     }
6945
6946     toggle(){
6947         const { isOpen } = this.state;
6948         this.setState({isOpen: !isOpen});
6949     }
6950
6951     render(){
6952         const { user } = this.props;
6953         const { isOpen } = this.state;
6954         return (
6955             <ListGroupItem>
6956                 <Row>
6957                     <Col onClick={this.toggle}
6958                         className="text-ellipsis" style={{cursor: 'pointer'}}>
6959                         {formatter.cpf(user.cpf)} - {user.name}
6960                     </Col>
6961                 </Row>
6962                 { isOpen && <Details cpf={user.cpf}
6963                     isOpen={isOpen} close={this.toggle} /> }
6963             </ListGroupItem>
6964     }
6965 }
```

```

6963     );
6964   }
6965 }
6966
6967 webapp/src/pages/protected/dashboard/school/volunteer/Details.jsx
6968
6969 import React, { Component } from 'react';
6970 import { getRequest } from '../../../../../utils/http';
6971 import formatter from '../../../../../utils/formatter';
6972 import { Status, Address, Schooling } from
6973   '../../../../../model';
6974 import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
6975   Row, Col, Label } from 'reactstrap';
6976 import pic from './pic.jpeg';
6977
6978
6979 export default class extends Component{
6980   constructor(){
6981     super();
6982     this.state = {
6983       volunteer: null,
6984     };
6985   }
6986
6987   componentWillMount(){
6988     const { cpf } = this.props;
6989     getRequest('/volunteer/${cpf}', res =>
6990       this.setState({volunteer: res.data}));
6991   }
6992
6993   render(){
6994     const { volunteer } = this.state;
6995     const { close, isOpen } = this.props;
6996     if(volunteer){
6997       const { user } = volunteer;
6998       const address = new Address();
6999       Object.assign(address, user.address);
7000       return (
7001         <Modal toggle={close} isOpen={isOpen}>
7002           <ModalHeader toggle={close}>{user.name} -
7003             <small>{formatter.cpf(user.cpf)}</small></ModalHeader>
7004           <ModalBody>
7005             <Row>
7006               <Col md="1">
7007                 <img src={pic} alt="" className="rounded-circle"
7008               </Col>
7009             </Row>
7010           </ModalBody>
7011         </Modal>
7012       );
7013     }
7014   }
7015 }

```

```
        width="75px" />
7003    </Col>
7004    <Col md="2" />
7005    <Col>
7006        <Row>
7007            <Col>
7008                <Label className="btn
7009                    btn-info"
7010                    style={{cursor:
7011                        'default'}}>
7012                        {Status.translate(user.status)}
7013                    </Label>
7014                </Col>
7015            </Row>
7016        <Row>
7017            <Col>
7018                <strong>Esoclariade:</strong>
7019            </Col>
7020        <Row>
7021            <Col>
7022                {
7023                    Schooling.translate(volunteer.scho
7024                    } -
7025                    {volunteer.conclusion ?
7026                        'Completo':
7027                        'Incompleto'}
7028                {
7029                    volunteer.course && <div>
7030                        { volunteer.course }
7031                    </div>
7032                }
7033            </Col>
7034        <br />
7035        <Row>
7036            <Col>
7037                <strong>Contato:</strong>
7038            </Col>
7039        </Row>
7040        <Row>
7041            <Col>
7042                {user.email} /
7043                {formatter.phone(user.phone)}
7044            </Col>
```

```

7038         </Row>
7039         <br/>
7040         <Row>
7041             <Col>
7042                 <strong>Endereço:</strong>
7043             </Col>
7044         </Row>
7045         <Row>
7046             <Col>
7047                 {address.toString()}
7048             </Col>
7049         </Row>
7050         <br />
7051         <Row>
7052             <Col>
7053                 <strong> Avaliações:</strong>
7054             </Col>
7055         </Row>
7056         <Row>
7057             <Col>
7058             {
7059                 volunteer.ratings.length?
7060                     ' Avaliou   média de
7061                         ${user.name}
7062                         ${volunteer.rating}':
7063                         'Voluntário ainda
7064                         não foi avaliado'
7065                     }
7066             </Col>
7067         </Row>
7068         <br/>
7069     </Col>
7070 </Row>
7071 </ModalBody>
7072 <ModalFooter>
7073     <Button color="default"
7074         className="float-right"
7075         onClick={close}>Fechar</Button>
7076 </ModalFooter>
7077 </Modal>
7078 )
7079 }
7080 return (
7081     <Modal toggle={close} isOpen={isOpen}/>
7082 );
7083

```

```
7077     }
7078 }
7079
7080 webapp/src/pages/protected/dashboard/school/volunteer/List.jsx
7081
7082 import React, { Component } from 'react';
7083 import {ListGroup, ListGroupItem} from 'reactstrap';
7084 import VolunteerItem from './ListItem.jsx';
7085 import { getRequest } from '../../../../../utils/http';
7086
7087 export default class extends Component{
7088     constructor(){
7089         super();
7090         this.state = {
7091             users: []
7092         };
7093     }
7094
7095     componentWillMount(){
7096         const {codSchool} = this.props;
7097         getRequest(`/volunteer?cod_school=${codSchool}`, res =>
7098             this.setState({users: res.data.map(volunteer =>
7099                 volunteer.user)}))
7100     }
7101
7102     render(){
7103         const { users } = this.state;
7104         return (
7105             <ListGroup>
7106                 {
7107                     users.length ?
7108                         users.map(
7109                             user => <VolunteerItem key={user.cpf}
7110                                 user={user} />
7111                         ) :
7112                             <ListGroupItem>
7113                                 <strong>
7114                                     Nenhum Voluntrio Cadastrado
7115                                 </strong>
7116                             </ListGroupItem>
7117                         )
7118                 )
7119             }
7120         }
7121     }
7122 }
```

```
7119
7120  webapp/src/pages/protected/dashboard/school/event/index.js
7121
7122  import List from './List.jsx';
7123  export default List;
7124
7125  webapp/src/pages/protected/dashboard/school/event/Details.jsx
7126
7127  import React, { Component } from 'react';
7128  import PropTypes from 'prop-types';
7129  import { getRequest } from '../../../../../utils/http';
7130  import { Address } from '../../../../../model';
7131  import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
7132          Row, Col } from 'reactstrap';
7133
7134  export default class extends Component{
7135      constructor(){
7136          super();
7137          this.state = {
7138              eventDetails : null
7139          }
7140      }
7141
7142      static propTypes = {
7143          code: PropTypes.number.isRequired,
7144          isOpen: PropTypes.bool.isRequired,
7145          close: PropTypes.func.isRequired,
7146      };
7147
7148      componentWillMount() {
7149          const { code } = this.props;
7150          getRequest(
7151              `/event/${code}`,
7152              res => {
7153                  let eventDetails = res.data;
7154                  eventDetails.title = eventDetails.type.name + ' - '
7155                  + new
7156                  Date(eventDetails.startOccurrence).toLocaleString();
7157                  this.setState({ eventDetails });
7158              }
7159          );
7160      }
7161
7162      render(){
```

```
7161 const { isOpen, close } = this.props;
7162 const { eventDetails } = this.state;
7163 if (eventDetails){
7164     const { address: eventAddress, trainings } =
7165         eventDetails;
7166     const duration = Math.ceil(
7167         Math.abs(
7168             eventDetails.startOccurrence -
7169             eventDetails.endOccurrence
7170         ) / (1000 * 3600)
7171 );
7172     const address = new Address();
7173     Object.assign(address, eventAddress);
7174     return (
7175         <Modal toggle={close} isOpen={isOpen} >
7176             <ModalHeader
7177                 toggle={close}>{eventDetails.title}</ModalHeader>
7178             <ModalBody>
7179                 <Row>
7180                     <Col>
7181                         <strong> Durao :</strong>
7182                             {eventDetails.allDay ? 'Dia
7183                             Inteiro' : duration + ' horas
7184                             (aproximadamente)'}
7185                     </Col>
7186                 </Row>
7187                 <br />
7188                 <Row>
7189                     <Col>
7190                         <strong>Endereço:</strong>
7191                     </Col>
7192                 </Row>
7193                 <Row>
7194                     <Col>
7195                         {address.toString()}
7196                     </Col>
7197                 </Row>
7198                 <Row>
7199                     <Col>
```

```

7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
    {
        trainings.length ?
            (
                <Row>
                    {
                        trainings.map(
                            training => (
                                <Col
                                    key={training.codTrain}
                                    {training.link
                                        &&
                                        <a
                                            className="btn
                                            btn-info"
                                            href={training.link}
                                            target="_blank">{t
                                                </Col>
                                            )
                                        )
                                    }
                                </Row>
                            ) : 'Nenhum anexo para esse
                                evento..'
                        }
                    </Col>
                </Row>
            </ModalBody>
            <ModalFooter>
                <Button color="primary"
                    className="float-right"
                    onClick={close}>Fechar</Button>
            </ModalFooter>
        </Modal>
    )
} else {
    return (
        <Modal toggle={close} isOpen={isOpen} ></Modal>
    );
}
}
}

import React, { Component } from 'react';

```

```
7235 import { Calendar } from '../../../../../components';
7236 import BigCalendar from 'react-big-calendar';
7237 import Details from './Details.jsx';
7238 import { getRequest } from '../../../../../utils/http';
7239
7240 export default class extends Component{
7241     constructor(){
7242         super();
7243         this.state = {
7244             events: [],
7245             isOpenDetails: false,
7246             codEvent: null
7247         };
7248         this.closeDetails = this.closeDetails.bind(this);
7249         this.handlerSelectEvent =
7250             this.handlerSelectEvent.bind(this);
7251     }
7252
7253     componentWillMount(){
7254         const { codSchool } = this.props;
7255         getRequest(
7256             `/school/${codSchool}/event`,
7257             res => this.setState({
7258                 events: res.data.map(
7259                     event => {
7260                         event.end = new Date(event.endOccurrence);
7261                         event.start = new
7262                             Date(event.startOccurrence);
7263                         event.title = event.type.name + ' - ' +
7264                             event.start.toLocaleString()
7265                         return event
7266                     }
7267                 )
7268             );
7269     }
7270     closeDetails(){
7271         this.setState({ isOpenDetails: false, codEvent: null });
7272     }
7273     handlerSelectEvent(event){
7274         const { codEvent } = event
7275         this.setState({ isOpenDetails: true, codEvent });
7276     }
7277 }
```

```
7277     render(){
7278         const { events, isOpenDetails, codEvent } = this.state;
7280         return (
7281             <div style={{ height: '600px', marginBottom: '1%' }}>
7282                 <Calendar events={events}
7283                     onSelectEvent={this.handlerSelectEvent}
7284                     defaultView={BigCalendar.Views.WEEK}/>
7285                     { isOpenDetails && <Details code={codEvent}
7286                         isOpen={isOpenDetails}
7287                         close={this.closeDetails} />}
7288                 </div>
7289             );
7290         }
7291     }
7292 }
7293
7294 webapp/src/pages/protected/dashboard/school/Page.jsx
7295
7296 import React, { Component } from 'react';
7297 import { Row, Col } from 'reactstrap';
7298 import VolunteerList from './volunteer';
7299 import EventList from './event';
7300 import { getRequest } from '../../../../../utils/http';
7301
7302 export default class extends Component{
7303     constructor(){
7304         super();
7305         this.state = {
7306             school: null,
7307             volunteers: [],
7308             events: []
7309         };
7310     }
7311     componentWillMount(){
7312         const { cpf } = this.props;
7313         getRequest('/school?cod_cpf=${cpf}', res =>
7314             this.setState({ school: res.data[0]}));
7315     }
7316     render() {
7317         const { school } = this.state;
7318         if (school){
7319             return(
7320                 <Row>
7321                     <Col md="1"/>
```

```
7317             <Col>
7318                 <EventList codSchool={school.codSchool}/>
7319             </Col>
7320             <Col md="3">
7321                 <VolunteerList
7322                     codSchool={school.codSchool}/>
7323                 </Col>
7324             <Col md="1"/>
7325         </Row>
7326     );
7327 } else {
7328     return null;
7329 }
7330 }
7331
7332 webapp/src/pages/protected/dashboard/school/index.js
7333
7334 import Page from './Page.jsx';
7335 export default Page;
7336
7337 webapp/src/pages/protected/dashboard/company/Line.jsx
7338
7339 import React, { Component } from 'react';
7340 import { Button, Label } from 'reactstrap';
7341 import { FaTrash } from 'react-icons/fa';
7342 import Details from './Details';
7343 import Recommend from './Recommend';
7344 import Paticipation from './Paricipations';
7345 import formatter from '../../../../../utils/formatter';
7346 import { deleteRequest } from '../../../../../utils/http';
7347 import { Status } from '../../../../../model';
7348
7349 export default class extends Component{
7350     constructor(){
7351         super();
7352         this.state = {
7353             openDetails: false,
7354             openForm: false
7355         };
7356         this.handlerDelete = this.handlerDelete.bind(this);
7357         this.toggleRecommend = this.toggleRecommend.bind(this);
7358         this.toggleDetails = this.toggleDetails.bind(this);
7359     }
7360 }
```

```
7361     toggleRecommend(){
7362         const { openForm } = this.state;
7363         this.setState({ openForm: !openForm, openDetails: false
7364             });
7365     }
7366
7367     toggleDetails() {
7368         const { openDetails } = this.state;
7369         this.setState({ openDetails: !openDetails, openForm:
7370             false });
7371     }
7372
7373     handlerDelete(){
7374         const { volunteer, afterDelete } = this.props;
7375         const { user, company } = volunteer;
7376         deleteRequest('/user/${user.cpf}', () =>
7377             afterDelete(company));
7378     }
7379
7380     getLabel() {
7381         const { ratings, rating } = this.props.volunteer;
7382         const lastRating = ratings[0];
7383         let className, text;
7384         if (lastRating) {
7385             if (rating < 40 || lastRating.grade < 40) {
7386                 className = 'danger';
7387             } else if (rating < 70) {
7388                 className = 'warning';
7389             } else {
7390                 className = 'success';
7391             }
7392         } else {
7393             className = 'secondary';
7394             text = 'Ainda no h    avaliaes ';
7395         }
7396
7397         return (
7398             <Label className={'btn btn-${className}'} style={{{
7399                 cursor: 'default', width: 'inherit', height:
7400                     '40px' }}}>
7401                 {text}
7402             </Label>
7403         );
7404     }
7405 }
```

```
7401 render(){
7402     const { volunteer, afterDelete } = this.props;
7403     const { user } = volunteer;
7404     const { openDetails, openForm } = this.state;
7405     return (
7406         <tr>
7407             <td style={{ width: '15%' }}>{
7408                 formatter.cpf(user.cpf) }</td>
7409             <td style={{ maxWidth: '130px' }}>{ user.name
7410                 }</td>
7411             <td style={{ maxWidth: '115px' }}>
7412                 <Label className={'btn btn-${user.status ===
7413                     Status.WAIT_COMPANY ? 'warning':
7414                         'info'}'} style={{ color: 'white',
7415                         cursor: 'default', whiteSpace: 'unset'}}>
7416                     { Status.translate(user.status) }
7417                 </Label>
7418             </td>
7419             <td style={{ width: '40px' }}>
7420                 <Button type="button" color="info"
7421                     onClick={this.toggleDetails}>
7422                     Detalhes
7423                 </Button>
7424                 {
7425                     openDetails && (
7426                         <Details close={this.toggleDetails}
7427                             openForm={openForm} cpf={user.cpf}
7428                             isOpen={true}
7429                             closeRecommend={this.toggleRecommend}
7430                             deleteFunc={this.handlerDelete}/>
7431                     )
7432                 }
7433             </td>
7434             <td>
7435                 <Participation cpf={user.cpf}/>
7436             </td>
7437             <td style={{ width: user.status ===
7438                 Status.WAIT_COMPANY ? '60px': '200px' }}>
7439                 {
7440                     user.status === Status.WAIT_COMPANY ? (
7441                         <div>
7442                             <Button type="button" color="info"
7443                                 onClick={this.toggleRecommend}>
7444                                 Recomendar
7445                             </Button>
```

```

7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451 webapp/src/pages/protected/dashboard/company/Paricipations.jsx
7452
7453 import React, { Component, Fragment } from 'react';
7454 import { Button, Popover, PopoverHeader, PopoverBody } from
    'reactstrap';
7455 import { getRequest } from '../../../../../utils/http';
7456 import { Address } from '../../../../../model';
7457
7458 export default class extends Component{
7459     constructor(){
7460         super();
7461         this.state = {
7462             visible: false,
7463             participations: []
7464         };
7465         this.toggle = this.toggle.bind(this)
7466     }
7467
7468     toggle(){
7469         const { visible } = this.state;
7470         const { cpf } = this.props;
7471         if(!visible){

```

```

7472     getRequest(
7473       'volunteer/${cpf}/participations',
7474     res => {
7475       let participations = res.data.map(
7476         participation => {
7477           const { school } = participation;
7478           const address = new Address();
7479           Object.assign(address, participation);
7480           return {
7481             address, school
7482           }
7483         }
7484       );
7485       this.setState({ visible: true, participations
7486         });
7487     }
7488   } else{
7489     this.setState({ visible: false })
7490   }
7491 }
7492
7493 render(){
7494   const { cpf } = this.props;
7495   const { visible, participations } = this.state;
7496   const { length } = participations;
7497   return (
7498     <Fragment>
7499       <Button id={'volunteer-${cpf}'} color="info"
7500         onClick={this.toggle}>Locais</Button>
7501       <Popover placement="auto" isOpen={visible}
7502         target={'volunteer-${cpf}'}
7503         toggle={this.toggle}>
7504         <PopoverHeader> Paticipaes </PopoverHeader>
7505         <PopoverBody>
7506           {
7507             length ?
7508               participations.map(
7509                 (participation, index) => (
7510                   <Fragment>
7511                     {participation.school}
7512                     ->
7513                     {participation.address.toString()
7514                     index !== length-1 &&
7515                     <hr/>})

```

```
7510                                     </Fragment>
7511                                 )
7512     ) : <strong>Este voluntário ainda não
7513         participou de nenhum evento
7514         pela CnE.</strong>
7515     }
7516   </PopoverBody>
7517 </Popover>
7518 </Fragment>
7519 );
7520
7521 webapp/src/pages/protected/dashboard/company/Page.jsx
7522
7523 import React, { Component } from 'react';
7524 import { Row, Col, Table } from 'reactstrap';
7525 import { Filter } from '../../../../../components';
7526 import Line from './Line.jsx';
7527 import { getRequest } from '../../../../../utils/http';
7528
7529 export default class extends Component{
7530   constructor(){
7531     super();
7532     this.state = {
7533       filter: '',
7534       company: null,
7535       volunteers: []
7536     };
7537     this.listVolunteers = this.listVolunteers.bind(this);
7538     this.handlerFilter = this.handlerFilter.bind(this);
7539   }
7540
7541   componentWillMount(){
7542     const { cpf } = this.props;
7543     getRequest(
7544       `company?cod_cpf=${cpf}`,
7545       companyRes => {
7546         const company = companyRes.data[0];
7547         if (company){
7548           this.listVolunteers(company)
7549         }
7550       }
7551     )
7552   }
}
```

```
7553
7554     handlerFilter(event) {
7555         const { company } = this.state;
7556         const filter = event.target.value;
7557         if (filter.length >= 3 || filter.length === 0){
7558             this.listVolunteers(company, filter);
7559         } else {
7560             this.setState({filter});
7561         }
7562     }
7563
7564     listVolunteers(company, filter=''){
7565         getRequest(
7566             `/volunteer?cod_cnpj=${company.cnpj}&filter=${filter}`,
7567             res => this.setState({ company, filter, volunteers:
7568                 res.data })
7569         )
7570     }
7571
7572     render(){
7573         const { company, volunteers, filter } = this.state;
7574         if ( company ){
7575             return (
7576                 <Row>
7577                     <Col md="1"/>
7578                     <Col>
7579                         <Table striped>
7580                             <thead>
7581                                 <tr>
7582                                     <th>CPF</th>
7583                                     <th>
7584                                         <Filter label="Nome " value={filter} handlerFilter={this.handlerFilter}>
7585                                     </th>
7586                                     <th>Status</th>
7587                                     <th>Detalhes</th>
7588                                     <th> Participaes </th>
7589                                     <th colSpan="2"></th>
7590                             </tr>
7591                         </thead>
7592                         <tbody>
7593                             {
7594                                 volunteers.length ?
```

```
7594     volunteers.map(volunteer =>
7595         <Line
7596             key={volunteer.user.cpf}
7597             volunteer={volunteer}
7598             afterDelete={this.listVolunteers}/>
7599             :
7600             (
7601                 <tr>
7602                     <td colSpan="3">
7603                         <strong>
7604                             Nenhum Voluntrio
7605                             Cadastrado
7606                         </strong>
7607                     </td>
7608                 </tr>
7609             )
7610         }
7611     </tbody>
7612     </Table>
7613     </Col>
7614     <Col md="1"/>
7615     </Row>
7616 );
7617 }
7618 return (null);
7619 }
7620 }
7621 webapp/src/pages/protected/dashboard/company/index.js
7622
7623 import Page from './Page.jsx';
7624 export default Page;
7625
7626 webapp/src/pages/protected/dashboard/company/Recommend.jsx
7627
7628 import React, { Component } from 'react';
7629 import { Modal, ModalHeader, ModalBody, Row, Col } from
7630     'reactstrap';
7631 import { Wizard, Switch } from '../../../../../components';
7632 import { getRequest, putRequest } from '../../../../../utils/http';
7633 import formatter from '../../../../../utils/formatter';
7634 import { Roles, Answer, Status } from '../../../../../model';
7635 import axios from 'axios';
7636
7637 export default class extends Component{
```

```

7632     constructor(){
7633         super();
7634         this.state = {
7635             volunteer : null,
7636             questions: [],
7637             answers: [],
7638             volunteerAnswers: []
7639         };
7640         this.handlerAnswer = this.handlerAnswer.bind(this);
7641         this.handlerAnswerComment =
7642             this.handlerAnswerComment.bind(this);
7643         this.handlerSubmit = this.handlerSubmit.bind(this);
7644     }
7645
7646     componentWillMount(){
7647         const { cpf } = this.props;
7648         axios.all([
7649             getRequest('/volunteer/${cpf}', res => res.data),
7650             getRequest('/question/${Roles.COMPANY_ADMIN}', res =>
7651                 res.data)
7652         ]).then(res => {
7653             this.setState({
7654                 volunteer: res[0],
7655                 questions: res[1],
7656                 answers: res[1].map(question => new
7657                     Answer(question)),
7658                 volunteerAnswers: res[0].answers
7659             });
7660         });
7661     }
7662     handlerAnswer(value, codQuestion) {
7663         const { answers } = this.state;
7664         for (const answer of answers) {
7665             if (answer.question.codQuestion === codQuestion) {
7666                 answer.answer = value;
7667                 break;
7668             }
7669         }
7670         this.setState({ answers });
7671     }
7672     handlerAnswerComment(event, codQuestion) {
7673         const { answers } = this.state;

```

```
7674     for (const answer of answers) {
7675         if (answer.question.codQuestion === codQuestion) {
7676             answer.answer = event.target.value;
7677             break;
7678         }
7679     }
7680     this.setState({ answers });
7681 }
7682
7683 handlerSubmit(){
7684     const { volunteer, answers, volunteerAnswers } =
7685         this.state;
7686     const { user } = volunteer;
7687     const { afterSubmit } = this.props;
7688     user.status = Status.WAIT_STATEMENT;
7689     volunteer.user = user;
7690     volunteer.answers.push(...answers);
7691     putRequest(
7692         '/volunteer/${user.cpf}',
7693         volunteer,
7694         afterSubmit,
7695         () =>{
7696             user.status = Status.WAIT_COMPANY;
7697             volunteer.answers = volunteerAnswers;
7698             volunteer.user = user;
7699             this.setState({volunteer});
7700         }
7701     );
7702 }
7703
7704 render(){
7705     const { volunteer, questions } = this.state;
7706     const { close, isOpen } = this.props;
7707     if(volunteer){
7708         const { user } = volunteer;
7709         return (
7710             <Modal toggle={close} isOpen={isOpen} >
7711                 <ModalHeader toggle={close}>
7712                     {user.name} -
7713                     <small>
7714                         {formatter.cpf(user.cpf)}
7715                     </small>
7716                 </ModalHeader>
7717                 <ModalBody>
```

```
7717 <Wizard onCancel={close}>
7718     submitLabel="Recomendar"
7719     onSubmit={this.handlerSubmit}>
7720     <div>
7721         {
7722             volunteer.answers.map((answer,
7723                 index) => (
7724                 <div
7725                     key={answer.question.codQuestao}
7726                     <Row >
7727                         <Col>
7728                             <strong>{answer.question.titulo}</strong>
7729                         </Col>
7730                         <Col md="2">
7731                             <Switch
7732                                 id={'question-$index'}
7733                                 label="Resposta"
7734                                 value={answer.answer}
7735                                 disabled
7736                             />
7737                         </Col>
7738                         <Col>
7739                             <textarea
7740                                 id={'details-$index'}
7741                                 cols="30"
7742                                 rows="3"
7743                                 value={answer.comentario}
7744                                 disabled
7745                             />
7746                         </Col>
7747                     </Row>
7748                     {index !==
7749                         answer.length -
7750                         1 ? <hr /> :
7751                         null}
7752                     </div>
7753                 ))
7754             }
7755             <hr/>
7756         </div>
7757         <div>
7758             {
7759                 questions.map((question, index)
7760                     => (
```

```
7742     <div  
7743         key={question.codQuestion}>  
7744         <Row >  
7745             <Col>  
7746                 <strong>{question.question}</strong>  
7747             </Col>  
7748             <Col md="2">  
7749                 <Switch  
7750                     id={'question-${question.codQ  
7751                         label="Resposta"  
7752                         onChange={(value)  
7753                             =>  
7754                             this.handlerAnswer(value,  
7755                             question.codQuestion)}  
7756                         />  
7757                 </Col>  
7758             </Row>  
7759             {index !==  
7760                 question.length - 1  
7761                 ? <hr /> : null}  
7762         </div>  
7763     ))  
7764     }  
7765     <hr/>  
7766     </div>  
7767     </Wizard>  
7768     </ModalBody>  
7769     </Modal>  
7770 );  
7771 }  
7772 return (null);  
7773 }  
7774 }
```

```
7768
7769  webapp/src/pages/protected/dashboard/company/Details.jsx
7770
7771  import React, { Component } from 'react';
7772  import { Button, Modal, ModalHeader, ModalBody, ModalFooter,
7773      Row, Col } from 'reactstrap';
7774  import Recommend from './Recommend.jsx';
7775  import { getRequest } from '../../../../../utils/http';
7776  import formatter from '../../../../../utils/formatter';
7777  import { Address, Schooling, Status } from '../../../../../model';
7778  import pic from './pic.jpeg';
7779
7780  export default class extends Component{
7781      constructor(){
7782          super();
7783          this.state = {
7784              volunteer: null
7785          }
7786
7787          componentWillMount(){
7788              const { cpf } = this.props;
7789              getRequest(`/volunteer/${cpf}`, res =>
7790                  this.setState({volunteer: res.data}));
7791
7792          handlerDelete(){
7793              const { deleteFunc, close } = this.props;
7794              deleteFunc();
7795              close();
7796          }
7797
7798          render(){
7799              const { volunteer } = this.state;
7800              const { closeRecommend, openForm, close, isOpen } =
7801                  this.props;
7802              if(volunteer){
7803                  const { user } = volunteer;
7804                  const address = new Address();
7805                  Object.assign(address, user.address);
7806                  return (
7807                      <Modal toggle={close} isOpen={isOpen} >
7808                          <ModalHeader toggle={close}>
7809                              {user.name} -

```

```
7810          {formatter.cpf(user.cpf)}
7811      </small>
7812  </ModalHeader>
7813  <ModalBody>
7814      <Row>
7815          <Col md="1">
7816              <img src={pic} alt=""
7817                  className="rounded-circle"
7818                  width="75px" />
7819      </Col>
7820      <Col md="2" />
7821      <Col>
7822          <Row>
7823              <Col>
7824                  <strong>Escolaridade:</strong>
7825              </Col>
7826          </Row>
7827          <Col>
7828              {
7829                  Schooling.translate(volunteer.schooling)
7830              } -
7831              {volunteer.conclusion ?
7832                  'Completo',
7833                  'Incompleto'}
7834              {
7835                  volunteer.course && <div>
7836                      { volunteer.course }
7837                  </div>
7838              }
7839          </Col>
7840      </Row>
7841      <br />
7842      <Row>
7843          <Col>
7844              <strong>Contato:</strong>
7845          </Col>
7846      </Row>
7847      <br/>
```

```

7847 <Row>
7848   <Col>
7849     <strong>Endereço:</strong>
7850   </Col>
7851 </Row>
7852 <Row>
7853   <Col>
7854     {address.toString()}
7855   </Col>
7856 </Row>
7857 <br />
7858 <Row>
7859   <Col>
7860     <strong> Avaliações :</strong>
7861   </Col>
7862 </Row>
7863 <Row>
7864   <Col>
7865   {
7866     volunteer.ratings.length?
7867       ' Avaliou média de
7868       ${user.name}
7869       ${volunteer.rating}':
7870       'Voluntário ainda
7871       não foi avaliado'
7872     }
7873   </Col>
7874 </Row>
7875 </ModalBody>
7876 <ModalFooter style={{ display: 'inline' }}>
7877   <Row>
7878     <Col>
7879       <Button color="danger"
7880         onClick={this.handlerDelete}>Excluir</But
7881     </Col>
7882     <Col>
7883       {
7884         user.status ===
7885           Status.WAIT_COMPANY? (
7886             <Button color="info"
7887               className="float-right"
7888               onClick={closeRecommend}>Recomenda
7889             ) : (
7890

```

```
7884             <Button color="default"
7885                 className="float-right"
7886                 onClick={close}>Fechar</Button>
7887         )
7888     }
7889     {
7890         openForm && (
7891             <Recommend
7892                 close={closeRecommend}
7893                 cpf={user.cpf}
7894                 isOpen={true} />
7895         )
7896     }
7897     </Col>
7898     </Row>
7899     </ModalFooter>
7900     </Modal>
7901 );
7902 } else {
7903     return null
7904 }
7905 }
7906 }
7907 }
7908 webapp/src/pages/protected/dashboard/index.js
7909
7910 import Dashboard from './Dashboard.jsx';
7911 export default Dashboard;
7912
7913 webapp/src/pages/protected/dashboard/Dashboard.jsx
7914
7915 import React, { Fragment } from 'react';
7916 import { Redirect } from 'react-router-dom';
7917 import AdminDashboard from './admin';
7918 import VolunteerDashboard from './volunteer';
7919 import CompanyDashboard from './company';
7920 import SchoolDashboard from './school';
7921 import { Roles } from '../.../model';
7922
7923 export default ({cpf, role, logged}) => {
7924     return logged ?
7925         <Fragment>
7926         {
7927             role === Roles.ADMIN && (
7928                 <AdminDashboard/>
```

```
7924         )
7925     }
7926     {
7927         role === Roles.VOLUNTEER && (
7928             <VolunteerDashboard cpf={cpf}/>
7929         )
7930     }
7931     {
7932         role === Roles.COMPANY_ADMIN && (
7933             <CompanyDashboard cpf={cpf}/>
7934         )
7935     }
7936     {
7937         role === Roles.SCHOOL_ADMIN && (
7938             <SchoolDashboard cpf={cpf}/>
7939         )
7940     }
7941     </Fragment>
7942 ) : <Redirect to={`/login`} />
7943 }
7944
7945 webapp/src/pages/protected/index.js
7946
7947 export {default as Dashboard} from './dashboard';
7948 export {default as Admin} from './admin';
7949
7950 webapp/src/pages/index.js
7951
7952 export {Dashboard, Admin} from './protected';
7953 export {Login, Register} from './public';
7954
7955 webapp/src/App.css
7956
7957 a.no-underline {
7958     color: inherit;
7959     text-decoration: None; }
7960 a.no-underline:hover {
7961     color: inherit;
7962     text-decoration: None; }
7963
7964 textarea {
7965     resize: none !important; }
7966
7967 .text-ellipsis {
7968     white-space: nowrap;
```

```
7969     overflow: hidden;
7970     text-overflow: ellipsis; }
7971
7972
7973 webapp/src/index.js
7974
7975 import React from 'react';
7976 import ReactDOM from 'react-dom';
7977 import registerServiceWorker from './registerServiceWorker';
7978 import App from './App.jsx';
7979 import 'bootstrap/dist/css/bootstrap.min.css';
7980 import './index.css';
7981 import './utils/http';
7982
7983 ReactDOM.render(<App />, document.getElementById('root'));
7984 registerServiceWorker();
7985
7986
7987 webapp/src/index.css
7988
7989 body {
7990     margin: 0;
7991     padding: 0;
7992     font-family: sans-serif; }
7993
7994
7995 webapp/src/App.sass
7996
7997 a
7998     &.no-underline
7999         color: inherit
8000         text-decoration: None
8001         &:hover
8002             color: inherit
8003             text-decoration: None
8004
8005 textarea
8006     resize: none !important
8007
8008 .text-ellipsis
8009     white-space: nowrap
8010     overflow: hidden
8011     text-overflow: ellipsis
8012
8013 webapp/src/index.sass
```

```
8014
8015   body
8016     margin: 0
8017     padding: 0
8018     font-family: sans-serif
8019
8020
8021
8022 webapp/src/App.test.js
8023
8024 import React from 'react';
8025 import ReactDOM from 'react-dom';
8026 import App from './App';
8027
8028 it('renders without crashing', () => {
8029   const div = document.createElement('div');
8030   ReactDOM.render(<App />, div);
8031   ReactDOM.unmountComponentAtNode(div);
8032 });
8033
8034
8035 webapp/src/App.jsx
8036
8037 // @flow
8038 import React, { Component } from 'react';
8039 import { Header, Footer } from './components';
8040 import { BrowserRouter, Route, Redirect } from
  'react-router-dom';
8041 import { Container, Row, Col } from 'reactstrap';
8042 import { Dashboard, Login, Register, Admin } from './pages';
8043 import { User } from './model';
8044 import { getAsObject, saveObject } from './utils/storage';
8045 import './App.css';
8046
8047 export default class App extends Component {
8048   constructor(){
8049     super();
8050     const user = getAsObject('user');
8051     this.state = {
8052       logged: getAsObject('logged'),
8053       user: user ? user : new User()
8054     };
8055     this.handlerLogin = this.handlerLogin.bind(this);
8056     this.handlerLogout = this.handlerLogout.bind(this);
8057   }
```

```
8058     handlerLogin(user){
8059         saveObject('user', user);
8060         saveObject('logged', true);
8061         this.setState({ logged: true, user });
8062     }
8063
8064
8065     handlerLogout(){
8066         let user = new User();
8067         saveObject('user', user);
8068         saveObject('logged', false);
8069         this.setState({ logged: false, user });
8070     }
8071
8072     render() {
8073         const { logged, user } = this.state;
8074         return (
8075             <BrowserRouter>
8076                 <div>
8077                     <Header role={user.role} logged={logged}
8078                         logout={this.handlerLogout} name={user.name}/>
8079                     <Container style={{ margin: '3% 0' }} fluid>
8080                         <Row>
8081                             <Col>
8082                                 <Route path="/dashboard" render={() =>
8083                                     (<Dashboard logged={logged}
8084                                         cpf={user.cpf} role={user.role}/>)}
8085                                 <Route path="/login" render={() => <Login
8086                                     logged={logged}
8087                                     afterLogin={this.handlerLogin}/>}/>
8088                                 <Route path="/cadastrar" render={() =>
8089                                     <Register
8090                                         afterSubmit={this.handlerLogin}/>}/>
8091                                 <Route path="/admin" render={() => <Admin
8092                                     logged={logged} role={user.role}/>}/>
8093                                 <Route exact path="/" render={() =>
8094                                     <Redirect to="/login"/>}/>
8095                             </Col>
8096                         </Row>
8097                     </Container>
8098                     <Footer/>
8099                 </div>
8100             </BrowserRouter>
8101         );
8102     }
8103 }
```

```
8094 }
8095
8096
8097 webapp/src/registerServiceWorker.js
8098
8099 // In production, we register a service worker to serve assets
     from local cache.
8100
8101 // This lets the app load faster on subsequent visits in
     production, and gives
8102 // it offline capabilities. However, it also means that
     developers (and users)
8103 // will only see deployed updates on the "N+1" visit to a page,
     since previously
8104 // cached resources are updated in the background.
8105
8106 // To learn more about the benefits of this model, read
     https://goo.gl/KvvDMy.
8107 // This link also includes instructions on opting out of this
     behavior.
8108
8109 const isLocalhost = Boolean(
8110   window.location.hostname === 'localhost' ||
8111     // [::1] is the IPv6 localhost address.
8112   window.location.hostname === '[::1]' ||
8113     // 127.0.0.1/8 is considered localhost for IPv4.
8114   window.location.hostname.match(
8115     /^127(?:\.(?:25[0-5]|2[0-4][0-9]|0[1-9][0-9]|0)|[0-9]{3})$/
8116   )
8117 );
8118
8119 export default function register() {
8120   if (process.env.NODE_ENV === 'production' && 'serviceWorker' in navigator) {
8121     // The URL constructor is available in all browsers that
     support SW.
8122   const publicUrl = new URL(process.env.PUBLIC_URL,
     window.location);
8123   if (publicUrl.origin !== window.location.origin) {
8124     // Our service worker won't work if PUBLIC_URL is on a
     different origin
8125     // from what our page is served on. This might happen if a
     CDN is used to
8126     // serve assets; see
     https://github.com/facebookincubator/create-react-app/issues/2374
```

```

8127     return;
8128 }
8129
8130 window.addEventListener('load', () => {
8131   const swUrl =
8132     `${process.env.PUBLIC_URL}/service-worker.js`;
8133
8134   if (isLocalhost) {
8135     // This is running on localhost. Lets check if a service
8136     // worker still exists or not.
8137     checkValidServiceWorker(swUrl);
8138
8139     // Add some additional logging to localhost, pointing
8140     // developers to the
8141     // service worker/PWA documentation.
8142     navigator.serviceWorker.ready.then(() => {
8143       console.log(
8144         'This web app is being served cache-first by a
8145         service ' +
8146         'worker. To learn more, visit https://goo.gl/SC7cgQ'
8147       );
8148     });
8149   });
8150 }
8151 }

8152 function registerValidSW(swUrl) {
8153   navigator.serviceWorker
8154     .register(swUrl)
8155     .then(registration => {
8156       registration.onupdatefound = () => {
8157         const installingWorker = registration.installing;
8158         installingWorker.onstatechange = () => {
8159           if (installingWorker.state === 'installed') {
8160             if (navigator.serviceWorker.controller) {
8161               // At this point, the old content will have been
8162               // purged and
8163               // the fresh content will have been added to the
8164               // cache.
8165               // It's the perfect time to display a "New content
8166               // is

```

```
8165 // available; please refresh." message in your web
     app.
8166 console.log('New content is available; please
               refresh.');
8167 } else {
8168     // At this point, everything has been precached.
8169     // It's the perfect time to display a
8170     // "Content is cached for offline use." message.
8171     console.log('Content is cached for offline use.');
8172 }
8173 }
8174 };
8175 }
8176 }
8177 .catch(error => {
8178     console.error('Error during service worker registration:',
8179     error);
8200 });
8181
8182 function checkValidServiceWorker(swUrl) {
8183     // Check if the service worker can be found. If it can't
     reload the page.
8184 fetch(swUrl)
8185     .then(response => {
8186         // Ensure service worker exists, and that we really are
             getting a JS file.
8187         if (
8188             response.status === 404 ||
8189             response.headers.get('content-type').indexOf('javascript')
                 === -1
8200     ) {
8211         // No service worker found. Probably a different app.
             Reload the page.
8212         navigator.serviceWorker.ready.then(registration => {
8213             registration.unregister().then(() => {
8214                 window.location.reload();
8215             });
8216         });
8217     } else {
8218         // Service worker found. Proceed as normal.
8219         registerValidSW(swUrl);
8220     }
8202 }.catch(() => {
```

```
8203     console.log(  
8204       'No internet connection found. App is running in offline  
8205         mode.'  
8206     );  
8207   }  
8208  
8209   export function unregister() {  
8210     if ('serviceWorker' in navigator) {  
8211       navigator.serviceWorker.ready.then(registration => {  
8212         registration.unregister();  
8213       });  
8214     }  
8215   }
```
