

Bruno Ribeiro da Silva

***FOREST: Uma ferramenta para gerência de
Sprints/Maratonas de Software***

Florianópolis, Santa Catarina

Bruno Ribeiro da Silva

FOREST: Uma ferramenta para gerência de Sprints/Maratonas de Software

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação

Orientador:

José Eduardo De Lucca

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Florianópolis, Santa Catarina

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação pela Universidade Federal de Santa Catarina.

Título: FOREST: Uma ferramenta para gerência de Sprints/Maratonas de Software

Autor: Bruno Ribeiro da Silva

Orientador:

Prof. José Eduardo De Lucca

Banca Avaliadora:

Prof. José Mazzucco Jr.

Prof. Luís Fernando Friedrich

AGRADECIMENTOS

Gostaria de agradecer toda minha família, por todo o suporte que me foi dado em todos esses anos da minha vida. Agradeço aos meus colegas de trabalho pela compreensão do tempo que tive de empenhar para a conclusão deste trabalho e do curso.

Agradeço à todos os professores que encontrei por toda a graduação e por tudo aquilo que me foi passado de conhecimento.

RESUMO

No campo de desenvolvimento de aplicativos para computadores tem surgido cada vez mais a necessidade de utilização de metodologias ágeis para a execução de projetos. Comumente utilizado em projetos de código fonte aberto e muitas vezes até em aplicativos de código fechado o Sprint é uma espécie de maratona visando a correção mais rápida dos problemas que um determinado aplicativo apresenta. Por virtude da falta de uma ferramenta no auxílio e implementação do mesmo foi desenvolvida a ferramenta FOREST. O desenvolvimento da ferramenta procurou obedecer a padrões de projeto assim como se utilizar de técnicas ágeis na construção de software.

ABSTRACT

In the field of developing computer applications have increasingly emerged the need to use agile methods for project execution. Commonly used in open source projects and often even in the closed source applications Sprint is a kind of marathon in order to fix problems faster than a determined application presents. By virtue of the lack of a tool to aid in implementation of the same, a tool named FOREST was developed. The tool development sought to comply with design standards as well as use of agile techniques in software development.

SUMÁRIO

Lista de Figuras

1	Introdução	p. 10
1.1	Software livre	p. 10
1.1.1	História	p. 10
1.1.2	Motivação	p. 11
1.2	Objetivos	p. 11
1.2.1	Objetivo Geral	p. 11
1.2.2	Objetivos Específicos	p. 11
1.3	Justificativa	p. 12
1.4	Contribuição	p. 12
2	Revisão bibliográfica: Desenvolvimento de Software e plataformas tecnológicas	p. 13
2.1	Metodologias de desenvolvimento	p. 13
2.1.1	Metodologia clássica (Industrial)	p. 13
2.1.2	Metodologia ágil (Iterativa)	p. 14
2.2	Sprint	p. 15
2.2.1	Introdução	p. 15
2.2.2	História	p. 16

2.2.3	Importância	p. 16
2.2.4	Exemplos	p. 17
2.2.5	Ferramentas Existentes	p. 17
2.3	Ferramentas Google	p. 18
2.3.1	Introdução	p. 18
2.3.2	AppEngine	p. 18
2.3.3	Maps	p. 19
2.3.4	Chart	p. 20
2.3.5	Accounts	p. 21
2.3.6	Google Web Toolkit (GWT)	p. 21
2.4	Django	p. 21
2.4.1	Arquitetura	p. 22
2.4.2	Mapeador Objeto-Relacional	p. 23
2.4.3	Internacionalização e localização	p. 23
2.4.4	Testes	p. 23
3	Projeto	p. 25
3.1	Metodologia de desenvolvimento	p. 25
3.2	Padrões de projeto	p. 25
3.2.1	Inicialização tardia	p. 26
3.2.2	Decorador	p. 26
3.3	Frameworks e Bibliotecas	p. 27
3.3.1	Google App Engine	p. 27
3.3.2	Django	p. 28

3.4	Entidades e Funcionalidades	p. 29
3.4.1	Projeto	p. 29
3.4.2	Sprint	p. 29
3.4.3	Usuário	p. 30
3.5	Dados	p. 30
3.5.1	Modelo: Projeto	p. 31
3.5.2	Modelo: Sprint	p. 31
3.5.3	Modelo: Usuário	p. 33
3.5.4	Exemplo do modelo Sprint implementado no ORM	p. 33
3.6	Interface Gráfica	p. 34
3.7	Instalação	p. 36
3.7.1	Requisitos	p. 36
3.7.2	Instalação (local)	p. 36
3.7.3	Execução	p. 36
3.7.4	Instalação (Google App Engine)	p. 37
4	Conclusão	p. 39
4.1	Futuro	p. 40
4.1.1	Futuras implementações	p. 40
	Referências Bibliográficas	p. 41
	Apêndice A – Código Fonte	p. 43

LISTA DE FIGURAS

2.1	Representação visual do modelo cascata	p. 14
2.2	Exemplo de mapa estático gerado pela api do Google Maps	p. 20
2.3	Exemplo de gráfico gerado pela api do Google Charts	p. 20
2.4	Representação visual da arquitetura MVC	p. 22
3.1	Tela de cadastro de um novo projeto	p. 32
3.2	Validação no lado do cliente	p. 35
3.3	Campo validado	p. 35

1 INTRODUÇÃO

O sprint de software é uma ação cada vez mais comum entre projetos de software livre e de código aberto, uma peça fundamental para a continuação dos mesmos e muitas vezes também utilizado para promover a interação entre os desenvolvedores de um mesmo projeto. No entanto hoje esses eventos são realizados de maneira descentralizada e com gerência dificultada.

1.1 Software livre

1.1.1 História

No início de 1984, Richard Stallman, que na época era empregado pelo laboratório de Inteligência Artificial do MIT, deixou seu trabalho e começou a trabalhar no projeto GNU. Stallman considerava-se um hacker do tipo que gosta de compartilhar seus interesses tecnológicos e seu código. Ele não gostou da maneira que sua recusa em assinar acordos de exclusividade ou não-compartilhamento fez dele um exilado em seu próprio mundo, e como o uso de software proprietário em seu ambiente o tornou impotente diante de situações que poderiam facilmente ser resolvidas antes. (BARAHONA; PASCUAL; ROBLES, 2009)

Stallman então motivou-se a criar ferramentas que pudessem substituir as ferramentas proprietárias, ao conjunto dessas ferramentas ele criou o projeto GNU. A primeira ferramenta desenvolvida foi o GNU Emacs, um editor de texto. Como o interesse em usar o Emacs foi crescendo, outras pessoas se envolveram no projeto GNU, e Stallman decidiu que era hora de buscar financiamento. Assim, em 1985 criou a Free Software Foundation, uma instituição filantrópica para o desenvolvimento de software livre. (STALLMAN, 2002)

1.1.2 Motivação

Tem havido muita especulação sobre as motivações que os desenvolvedores tem para desenvolver software livre, especialmente quando é feito no tempo livre.

Segundo Barahona, Pascual e Robles (2009), a maioria [dos desenvolvedores] querem aprender e desenvolver novas habilidades (cerca de 80%) e que muitos fazem isso para compartilhar conhecimentos e habilidades (50%) ou para participar de uma nova forma de cooperação (um terço).

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é a busca de uma solução que ajude e centralize a organização, coordenação e gerência do sprint de software aplicado à diversos projetos, de modo que a realização de um sprint se torne trivial e viável à qualquer projeto que tenha interesse em realizá-lo.

1.2.2 Objetivos Específicos

Como objetivo específico e parte do processo de busca do objetivo geral, este trabalho tem como meta, levantar os requisitos e projetar um sistema que auxilie na centralização das atividades relacionadas à gerência de um sprint de software.

Esta aplicação deve utilizar-se da maior parte possível de bibliotecas, ferramentas e frameworks já desenvolvidos visando agilidade no processo de implementação da própria e manter os princípios básicos da reutilização de código, tanto pregado durante todo o tempo como uma das principais justificativas ao paradigma de programação orientada à objetos.

Como parte dessas bibliotecas e frameworks, surgiu durante o projeto a idéia de utilizar-se de todo um ecossistema de software oferecido pela empresa Google gratuitamente aos desenvolvedores, uma vez que foram identificados bons candidatos à poupar esforço na etapa de desenvolvimento, tais como Google Maps, Google Calendar, Google Accounts, Google App Engine e

etc. Estes todos serão melhores explicados e justificados no corpo deste trabalho.

1.3 Justificativa

A idéia de desenvolver um sistema de gerência de sprints de software partiu do problema que comumente é observado durante a realização desses eventos e pela não existência de uma ferramenta dedicada à esta atividade com fins de agilizar todo o processo.

1.4 Contribuição

Uma vez projetada e implementada uma ferramenta que melhore a tarefa de criação e gerência de sprints de software, automaticamente todos os projetos que se beneficiam desta prática estarão recebendo o acesso à esta ferramenta e como contribuição a possibilidade de melhor organização e maior foco nos objetivos dos sprints. Tornando assim a gerência do sprint um processo periférico e de fácil acesso aos interessados.

Todos os projetos ganham em tempo e foco se tiverem acesso à uma ferramenta que lhes facilitem a gerência de sprints, quando dos mesmos se beneficiam.

2 REVISÃO BIBLIOGRÁFICA: DESENVOLVIMENTO DE SOFTWARE E PLATAFORMAS TECNOLÓGICAS

2.1 Metodologias de desenvolvimento

Com a total consolidação das empresas de software no mercado de tecnologia faz-se necessário que os sistemas sejam desenvolvidos com o menor tempo possível e dentro de elevados padrões de qualidade, tornando esta tarefa desafiadora e muito difícil para gerentes de projetos espalhados por todo mundo. Neste capítulo aborda-se duas das principais vertentes quando se fala de metodologias no processo de construção de aplicativos, sejam eles para computadores, celulares, dispositivos móveis, sistemas embarcados e etc.

2.1.1 Metodologia clássica (Industrial)

À metodologia clássica remete-se aos projetos de software que tomam como base o modelo de desenvolvimento em cascata. Este modelo segue do início ao final por uma série de etapas às quais cada uma depende da anterior para que seja iniciada. As etapas que o modelo em cascata segue são: a análise de requisitos, o projeto, a implementação, testes (validação), integração, e, por fim, a manutenção do software.

A metodologia clássica é naturalmente seqüencial, portanto torna-se um processo que não admite erros durante as etapas, uma vez que a mudança do projeto durante a etapa de testes

por exemplo acarreta em possíveis correções em todas as etapas subsequentes e traz um custo de desenvolvimento muito alto para a empresa que emprega este modelo na construção de seus sistemas.

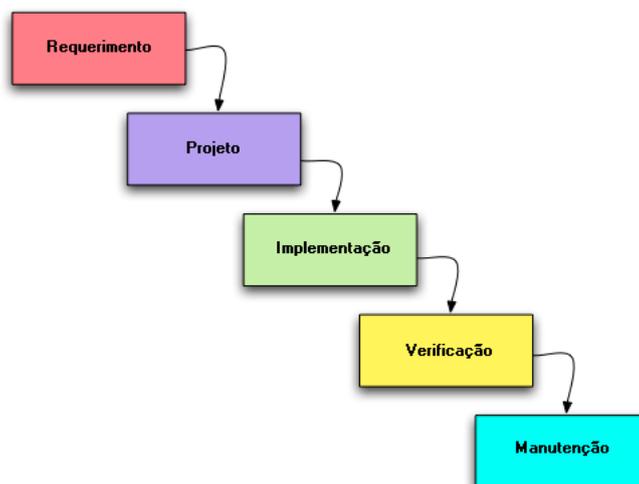


Figura 2.1: Representação visual do modelo cascata

Como podemos observar na tradução feita por Teles (2004):

"A falácia básica do modelo em cascata é que ele assume que o projeto irá passar pelo processo uma vez, que a arquitetura é excelente e fácil de utilizar, o design é correto e a implementação pode ser corrigida enquanto os testes estiverem sendo executados. Outra forma de dizer isso é que o modelo em cascata assume que todos os erros irão aparecer na implementação e, portanto, os reparos podem ser feitos sem maiores problemas testando os componentes e o sistema.

A segunda falácia do modelo em cascata é que ele assume que o sistema inteiro é construído de uma vez. Para que se possa fazer o teste de todo sistema, as diferentes partes precisam ser combinadas depois que todo o design e toda a codificação já foram feitos." (BROOKS, 1995)

2.1.2 Metodologia ágil (Iterativa)

À metodologia ágil refere-se ao modelo iterativo aplicado no desenvolvimento de softwares. A abordagem da metodologia ágil veio de maneira a eliminar com os problemas apresentados

pela metodologia clássica, herdada dos modelos industriais de produção. O modelo iterativo se baseia no princípio da realimentação e permite que o cliente aprenda ao longo do projeto gerando feedback para a equipe de desenvolvimento.

No desenvolvimento iterativo o sistema é feito de forma incremental seguindo um ciclo de vida espiral, fazendo com que cada volta da espiral seja composta por um módulo do sistema que é desenvolvido em cima das etapas descritas na metodologia clássica (em cascata). Desta maneira aplica-se as etapas de análise, design, implementação, teste, implantação e manutenção por módulos e não ao sistema inteiro (metodologia clássica).

2.2 Sprint

2.2.1 Introdução

O Sprint, também conhecido como maratona de software, é uma espécie de evento criado com os objetivos de sanar o maior número de problemas de um programa em menor espaço de tempo possível e/ou para adicionar novas funcionalidades ao programa.

Um Sprint de software pode ser realizado virtualmente, através de reuniões em canais de IRC (Internet Relay Chat), geralmente comandadas por mentores que auxiliam a gerenciar grupos de desenvolvedores interessados em áreas comuns do software em questão.

Eventualmente, quando possível, os Sprints também são realizados em um local físico, estimulando a interação entre os desenvolvedores e melhorando a comunicação entre os grupos.

Atualmente não existe nenhum tipo de ferramenta voltada à gerência desses Sprints de software, de maneira que os mesmos costumam ser realizados e gerenciados com diversas ferramentas separadas, tais como bugtracker, e-mail, forum de discussão, páginas wiki, IRC, sistema de controle de versão, e quando dispõe de locação física à mesma não possui lugar centralizado para divulgação e coordenação do evento.

2.2.2 História

Os primeiros registros oficiais que se tem de um evento de tal tipo, datam de 9 de Junho de 1999 (HACKATHON...), dia este que foi realizado o primeiro Hackathon (OPENBSD...), termo derivado de uma composição de hacker e marathon, ou seja, uma maratona de hackers que se encontraram na cidade de Calgary em Alberta, Estados Unidos, para resolver problemas legais relacionados à regulamentação de exportação de software criptográfico naquele país, o software em questão era o IPSec, sistema este utilizado para o fechamento de VPNs (Virtual Private Networks). Este evento foi organizado por um grupo de desenvolvedores do OpenBSD (distribuição do BSD focada em segurança).

Logo em seguida, em 15 de Junho de 1999 a Sun Microsystems viria a utilizar o termo Hackathon e o mesmo conceito na JavaOne Conference, onde o palestrante John Gage desafiou os espectadores à escrever um programa em Java para o novo Palm V utilizando a porta de infravermelho para intercomunicação entre os outros usuários de Palm. (HACKATHON...)

O Sprint é uma técnica poderosa de temporizar um projeto e canalizar os recursos para trabalhar nos requisitos de maior prioridade. (WOODWARD; SURDEK; GANIS, 2010)

2.2.3 Importância

Como citado anteriormente, o Sprint tornou-se muito forte e difundido em projetos de software livre e de código aberto pelo fato da natureza desses projetos serem muito favoráveis a este tipo de adoção, uma vez que grande parte dos desenvolvedores destes projetos estão distribuídos através do globo. Fato este que exige de tempos em tempos um evento coordenado onde são resolvidas interdependências entre os desenvolvedores do projeto.

Esses eventos também são muito importantes por trazerem novos adeptos e até novos desenvolvedores aos projetos, é comum pessoas dispostas à ajudar na resolução de bugs em um Sprint se envolverem-se tanto e tornarem-se desenvolvedores ativos do sistema, alguns até de muito destaque.

Para alguns projetos os sprints são até mesmo de vital importância, como no caso do PyPy (PYPY...,) que é praticamente desenvolvido em sprints regulares, em que desenvolvedores

internacionais se unem para implementar novas funcionalidades e realizar correções das já existentes.

2.2.4 Exemplos

Como exemplos reais da aplicação de sprints de software, temos o framework Django (DJANGO SOFTWARE FOUNDATION, a) em que frequentemente são realizados sprints em locais físicos através do globo em conjunto com sprints virtuais, onde são resolvidos o maior número de bugs possíveis e tendo como meta o lançamento da próxima versão estável do framework. Cada sprint em local físico é realizado em uma cidade diferente, em países diferentes. (DJANGO SOFTWARE FOUNDATION, b)

Um outro projeto que realiza eventos de sprint e que é muito conhecido por tais eventos é o projeto Zope (ZOPE CORPORATION, b), um servidor de aplicação para construção de sistemas de gerenciamento de conteúdo, intranets, portais e aplicações customizadas. (ZOPE CORPORATION, a)

Tem-se também muitos outros projetos como o já citado OpenBSD, a Sun em suas conferências sobre Java, PyPy, PHP, Apache, etc. O sprint também é utilizado em empresas que adotam a metodologia de desenvolvimento ágil Scrum, que faz uso direto de sprints em ciclos iterativos.

2.2.5 Ferramentas Existentes

Durante o percurso do trabalho foram pesquisadas algumas ferramentas já existentes que buscam auxiliar a gerência de Sprints de software. Algumas das ferramentas encontradas trazem o foco no Sprint utilizado no processo de desenvolvimento ágil chamado Scrum e este tipo de Sprint apesar de ter ligação direta com o Sprint de software possui algumas diferenças. No entanto nenhuma ferramenta focada em Sprint de software foi encontrada, somente as ferramentas com foco no Scrum, por virtude da similaridade entre ambas optou-se por citar algumas das melhores ferramentas que foram encontradas, para fins de avaliação e comparação.

- TeamPulse: Ferramenta proprietária e paga. (TELERIK,)
- pmScrum: Ferramenta proprietária e paga. (TENFORCE,)

- ScrumDo: Ferramenta de código fonte aberto e gratuita. (SCRUMDO,)
- kunagi: Ferramenta de código fonte aberto e gratuita. (KUNAGI,)
- Bright Green: Ferramenta proprietária e paga. (BRIGHT GREEN PROJECTS,)

2.3 Ferramentas Google

2.3.1 Introdução

O Google é uma das maiores empresas de tecnologia do mundo e também conhecida por contribuir bastante com projetos de código aberto e de utilização gratuita. Seus mais famosos sistemas são o buscador e o gmail. No entanto além destes dois sistemas citados o Google ainda nos oferece uma gama de bibliotecas e serviços que podem ser utilizados gratuitamente.

Neste capítulo serão explicados e demonstrados os principais serviços e bibliotecas disponibilizados pelo Google para desenvolvedores interessados em fazer uso de ferramentas livres e gratuitamente disponibilizadas.

2.3.2 AppEngine

O AppEngine é a solução que o Google encontrou para resolver os que problemas que muitas empresas enfrentam quando precisam desenvolver uma aplicação web escalável, rápida, robusta e segura. Imagine ter de gerenciar uma enorme quantidade de servidores, bancos de dados, aplicações, etc, tudo isto distribuído em datacenters ao redor do mundo.

O GAE (Google App Engine) traz para o desenvolvedor a abstração de tudo isso, fazendo com que o mesmo necessite se concentrar apenas na aplicação. Sem se preocupar em quais servidores seus dados estarão localizados, como será feita a redundância para casos de contingência, a consistência dos dados, etc.

Em resumo, você estará rodando sua aplicação na infraestrutura do maior buscador e de uma das maiores empresas do mundo voltada para a Internet, em um ambiente seguro, estável, redundante e veloz.

Atualmente é possível escrever suas aplicações em duas linguagens que são suportadas diretamente no GAE: Java e Python.

Os dados são elementos fundamentais de qualquer aplicação web e necessitam de especial atenção quando o banco que os vai armazenar é projetado. Como forma de acesso ao banco de dados que roda na infraestrutura do Google, o App Engine nos fornece uma API de acesso ao Google Datastore, que abstrai do programador o banco de dados utilizado e traz o mesmo em forma de objeto para manipulação/criação/edição. O Google Datastore é assunto tratado na sessão seguinte.

Datastore

Quando utiliza-se o Google App Engine se tem acesso à uma base de dados relacional como Oracle, MySQL, PostgreSQL, etc. Ao invés disso tem-se acesso ao Google Datastore que é baseada em uma tecnologia do Google chamada BigTable, desenvolvida para explorar o máximo possível de paralelismo e escalabilidade quando se trata de manipulação e armazenamento de dados. O Datastore utiliza uma abordagem orientada à objetos para o armazenamento de dados, possibilitando que manipulemos nossos dados diretamente como objetos em nossa aplicação.

O Datastore armazena e executa consultas sobre objetos-dados, conhecidos como entidades. Uma entidade tem uma ou mais propriedades, conhecidas como valores de um ou mais diferentes tipos suportados de dados e essas propriedades podem fazer referência à outra entidade.

O Datastore tem a característica de poder executar operações múltiplas em uma única transação e voltar a mesma em caso de falha em alguma das operações, o que é especialmente útil em aplicações web que constantemente manipulam os mesmos dados de forma concorrente.

2.3.3 Maps

O Maps é a aplicação de mapas geográficos e rodoviários que a Google disponibiliza para uso gratuito através da internet. Esta aplicação possui mapas atuais e precisos, até mesmo com fotos por satélite, para geolocalização. Além da aplicação também é disponibilizado uma biblioteca que permite embutir mapas específicos e marcados dentro da aplicação, através de javascript.

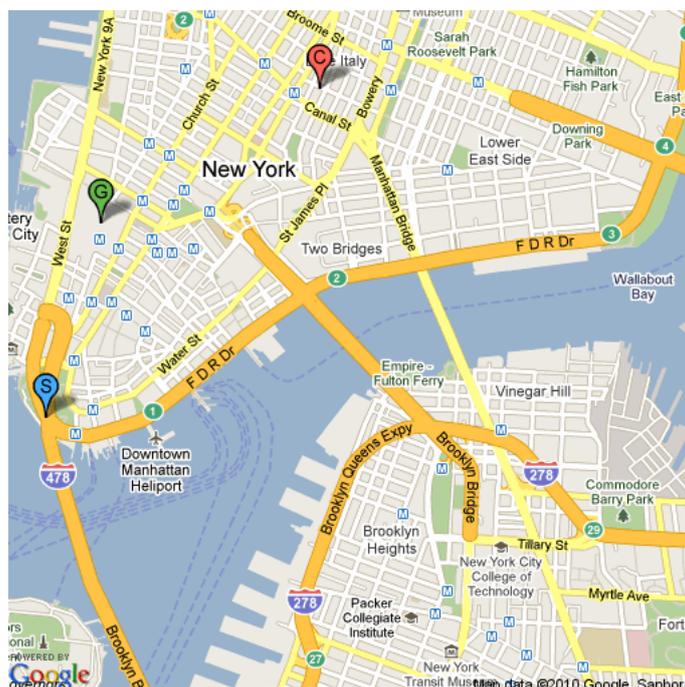


Figura 2.2: Exemplo de mapa estático gerado pela api do Google Maps

2.3.4 Chart

A biblioteca de gráficos da Google permite gerar gráficos personalizados com uma sintaxe bastante simplificada e transferindo a carga onerada para a nuvem. Através dessa api podem-se gerar gráficos do tipo pizza, barra, caixa, 3D, etc.



Figura 2.3: Exemplo de gráfico gerado pela api do Google Charts

2.3.5 Accounts

O Google disponibiliza para utilização através do GAE o login de seus usuários como forma de centralização de contas de usuário. A API de utilização é chamada de Google Accounts e dispõe de mecanismos de acesso baseado em permissões. Sendo assim qualquer aplicação desenvolvida no GAE que utilize da Accounts API já está habilitada a oferecer o serviço à toda a base de usuários do Google.

2.3.6 Google Web Toolkit (GWT)

Toda aplicação interativa precisa de uma apresentação através de telas para que o usuário realize as operações, normalmente em aplicações que rodam fora do navegador (aquelas que rodam diretamente no sistema operacional) utilizam-se de uma toolkit gráfica.

Desenvolvida pelos engenheiros da Google como uma toolkit genérica com o objetivo de ser utilizada em todas as aplicações dessa empresa, a GWT é a solução encontrada para manter a consistência das aplicações entre diversos navegadores diferentes.

Com a GWT programa-se as telas através da linguagem Java, podendo se beneficiar de IDEs avançadas, que ajudam na redução de erros de programação e agilizam no processo do desenvolvimento. No final do processo há uma compilação do código para o correspondente em Javascript e compatível com os principais navegadores do mercado.

A GWT também oferece um bom suporte para internacionalização e localização da aplicação.

2.4 Django

Atualmente o maior movimento no desenvolvimento de software é o foco na web, uma vez que as empresas estão cada vez mais migrando suas aplicações para a chamada nuvem, onde as mesmas podem ser acessadas de qualquer lugar no mundo sem nenhuma dificuldade.

Com o crescimento desta área, faz-se cada vez mais necessário a utilização de frameworks que trazem o pragmatismo do desenvolvimento guiado, a agilidade no processo, a padronização

do processo, uma melhor documentação e toda uma comunidade de desenvolvedores (no caso de frameworks de código livre).

O Django é um framework web desenvolvido inteiramente em python e que traz bibliotecas de utilidades em todas as camadas do desenvolvimento do software, dentre as facilidades estão: um poderoso Mapeador Objeto Relacional (ORM), um sistema de templates para construção das telas em html, um motor para a lógica da arquitetura (modelo-template-visão), sistema de controle de acesso baseado em usuário e senhas, biblioteca para geração de formulários, suporte à internacionalização e localização, um módulo para a construção de testes de unidade e muitas outras funcionalidades as quais serão omitidas por fugir do escopo deste trabalho.

2.4.1 Arquitetura

O framework Django utiliza uma arquitetura bastante utilizada por outros frameworks web, que é a chamada Modelo-Visão-Controlador ou simplesmente MVC. A implementação do MVC no Django também é chamada Modelo-Template-Visão uma vez que nomenclaturas diferentes foram adotadas para cada componente no caso específico deste framework.

A essência desta arquitetura é a separação em três camadas especializadas e comunicantes entre si, sendo assim a camada Modelo trata de tudo que for referente ao modelo de dados e a lógica de conversação com o sistema gerenciador de banco de dados que vai armazenar os mesmos. Já a camada Visão é aquela que trata de apresentar os dados obtidos na camada Modelo ao usuário da aplicação. A terceira e última camada, a Controladora é a responsável de interpretar as requisições feitas na camada Visão fazer o processamento devido, requisitar dados se necessário à camada Modelo e processá-los e entregá-los de volta à camada de Visão.

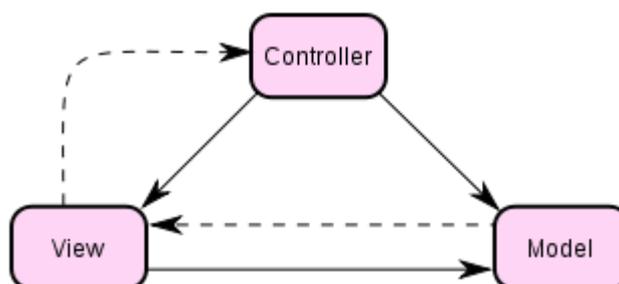


Figura 2.4: Representação visual da arquitetura MVC

2.4.2 Mapeador Objeto-Relacional

Apesar do uso generalizado de SQL em software corporativo, ainda existem armadilhas em usá-lo. Muitos desenvolvedores de aplicativos não entendem bem SQL e, como resultado, têm problemas de definição de comandos e procedimentos eficazes. Embora existam várias técnicas para a incorporação de SQL em uma linguagem de programação, todas elas são um pouco desajeitadas. Seria melhor usar mecanismos para acessar os dados que se encaixem no desenvolvimento das aplicações. (FOWLER et al., 2002)

O Django possui integrado um poderoso Mapeador Objeto-Relacional (ORM) que permite ao programador manipular os dados contidos nas tabelas de um banco relacional através de objetos que representam as entidades lá contidas. Assim é possível realizar qualquer operação de manipulação dos dados na linguagem em que a aplicação está sendo desenvolvida, sem a necessidade do conhecimento da sintaxe SQL.

O ORM é uma ferramenta que por abstrair a sintaxe SQL para a manipulação dos dados também elimina a dependência de um sistema gerenciador de banco de dados específico, sendo assim a escolha do SGBD fica à escolha da empresa onde o software será instalado, sejam eles Oracle, MySQL, PostgreSQL ou qualquer outro SGBD que possua um backend suportado.

2.4.3 Internacionalização e localização

O Django possui incluso em suas bibliotecas uma especialmente desenhada para a internacionalização e localização dos projetos, o processo de marcação das strings traduzíveis e geração do arquivo de tradução é muito simplificado, tornando triviais as tarefas de internacionalização e localização.

2.4.4 Testes

Uma parte muito importante no desenvolvimento de qualquer software é a criação e execução de testes rotineiros para assegurar que nenhuma modificação do código gerou problemas para em outros módulos do sistema, ou até mesmo no próprio módulo que está sendo alterado.

O Django traz consigo um mini-framework voltado para testes de unidade de software tor-

nando este tipo de tarefa ágil, trivial e completamente automatizável.

3 PROJETO

Este capítulo compreende da aplicação das ferramentas estudadas ao escopo do sistema proposto, desde a estrutura dos dados armazenados, passando pelo desenho das telas (interface gráfica do usuário) e processo aplicado durante o desenvolvimento.

Ressalta-se que a aplicação foi projetada desde o início para estar adaptada às atuais tecnologias da web, sendo assim a técnica de Ajax (Javascript assíncrono) foi utilizada intensivamente, em parte graças à flexibilidade dos frameworks adotados para a construção das telas trazem ao programadores no quesito de chamadas assíncronas ao servidor.

3.1 Metodologia de desenvolvimento

Optou-se por adotar uma metodologia de desenvolvimento ágil no desenvolvimento da aplicação FOREST descrita neste trabalho por apresentar maior dinâmica e velocidade no desenvolvimento, também permitindo experimentação sem que todo o projeto necessitasse ser repensado.

3.2 Padrões de projeto

Segundo Alexander (1977), um padrão descreve um problema que sempre ocorre em nosso ambiente, e então descreve a raiz da solução para aquele problema, de tal maneira que você pode utilizar essa solução um milhão de vezes, sem nunca fazer do mesmo jeito.

Durante a etapa de avaliação das funcionalidades do projeto e adoção do framework dois padrões de projetos foram identificados como candidatos para a solução de problemas específicos porém bem conhecidos. A inicialização tardia e o decorador, explicados nas seguintes

subsessões.

3.2.1 Inicialização tardia

Durante a inicialização de uma instância ou de uma classe podem-se ter atributos que consomem demasiados recursos para serem inicializados, neste tipo de caso faz-se aplicação do padrão de inicialização tardia, onde a variável só é inicializada quando for requisitada, poupando-se assim recursos dos servidores com tarefas custosas e que improvavelmente serão utilizadas.

No escopo da aplicação FOREST o conceito de inicialização tardia foi utilizado principalmente no módulo de internacionalização, onde as strings que serão mostradas para o utilizador só são inicializadas no momento em que são utilizadas (mostradas), já que a qualquer momento o usuário pode alterar a língua de visualização, sendo assim o recurso de inicialização tardia economiza recursos computacionais.

3.2.2 Decorador

O padrão decorador é utilizado para estender dinamicamente (em tempo de execução) as funcionalidades de um objeto. Normalmente utiliza-se este tipo de padrão evitando a criação de uma subclasse para cada novo comportamento que queremos adicionar ao objeto. Ele traz muita flexibilidade ao projeto uma vez que pode-se adicionar comportamento à um objeto sem precisar alterar seu código.

O único cuidado que deve ser tomado ao se utilizar este tipo de padrão é que se houver muitos decoradores no projeto pode-se aumentar demais a complexidade de sua utilização.

No projeto em questão este tipo de padrão foi bastante utilizado para a verificação de autorização de acesso às camadas de visualização, assim uma vez criado o decorador que verificasse os atributos do usuário logado no sistema bastaram-se marcar as views que precisavam de controle de acesso com o novo decorador, sem precisar do código das mesmas.

3.3 Frameworks e Bibliotecas

Feito estudos sobre frameworks e bibliotecas que poderiam ser utilizadas no trabalho, algumas delas foram escolhidas por se encaixar de melhor maneira na proposta do projeto.

3.3.1 Google App Engine

Datastore

Neste trabalho o Datastore possui um papel fundamental, uma vez que todos os dados sobre os projetos, wikis, agendamentos, dados pessoais, informações dos sprints, etc, estarão armazenados no Google Datastore, com exceção dos templates de páginas e arquivos estáticos.

Maps

Na maioria das vezes os Sprints realizados por projetos de Software Livre são feitos em lugares físicos com o intuito de gerar contato entre a comunidade desenvolvedora e dar a oportunidade para que problemas mais difíceis sejam resolvidos de maneira mais fácil, já que a comunicação é um ponto importante.

Com esta necessidade identificada no projeto foi resolvido por optar pela api de mapas da Google como maneira de mostrar para o participante o local físico do evento e que o mesmo consiga ver como chegar até lá.

Chart

Como em qualquer aplicação gerencial, relatórios gráficos são de base fundamental para análise dos dados produzidos, com base nesta afirmação e de conhecimento de que muitos dados são gerados pelos Sprints o Google Chart foi adotado como biblioteca para geração dos gráficos.

Accounts

A aplicação de gerência de Sprints FOREST desde o início de seu projeto visava a necessidade de um controle de acesso baseado em usuários e senhas, durante a pesquisa sobre as

bibliotecas disponibilizadas pela Google, a biblioteca Accounts trouxe a solução ideal já totalmente integrada com a base de usuários da Google fazendo por dispensar o cadastro de usuário e senha na aplicação quando o mesmo já tivesse uma conta em alguma aplicação da Google.

A biblioteca de utilização da api em python para ser utilizada no GAE foi um outro ponto positivo na escolha do Accounts para a aplicação em objeto do trabalho.

Google Web Toolkit (GWT)

A aplicação em questão foi pensada desde o início para ser um sistema web e ai que se apresenta o papel da Google Web Toolkit. Um ponto muito forte que a GWT traz para a aplicação deste trabalho é um desenvolvimento focado em uma api consistente e ágil, sem interferência no tratamento dos erros entre diversos navegadores.

3.3.2 Django

Internacionalização

Como uma característica dos projetos de software livre é de ter desenvolvedores espalhados pelo mundo todo, falando diferentes línguas, tornou-se claro que o projeto precisaria de uma ferramenta simples e que atendesse à necessidade de tornar a aplicação internacionalizada. O framework Django por possuir toda a estrutura necessária para a internacionalização foi utilizado em todo o escopo do projeto.

Testes

Mais uma vez a escolha do framework mostrou-se bastante apta para o projeto já que o mesmo possui uma espécie de mini-framework para criação, execução e depuração dos testes, o mesmo se encarrega de gerar banco de dados temporários e de não interferir com os dados de produção na hora da execução dos testes.

A criação de testes trouxe agilidade para o projeto e maior segurança do correto funcionamento da aplicação.

3.4 Entidades e Funcionalidades

Como funcionalidades da aplicação em questão compreende-se:

- Criação/alteração/remoção de um projeto
- Criação/alteração/remoção de um sprint
- Criação/alteração/remoção de um objetivo
- Criação/alteração/remoção de um usuário

3.4.1 Projeto

Compreende uma entidade que representa um projeto e/ou software, o qual irá se beneficiar da aplicação na organização de seus Sprints. Entre os principais atributos estão: nome, descrição e apontador via url para o site oficial.

Cada projeto possui uma lista de funcionalidades a ser implementada na aplicação:

- Associação de um ou mais moderadores
- Edição de página tipo wiki para informações relevantes ao projeto
- Marcação de Sprints
- Envio de e-mails para todos os usuários cadastrados no projeto
- Envio de convites para participar do projeto

3.4.2 Sprint

Cada objeto representa um Sprint único, pertencendo à somente um projeto e tendo data de início e fim, sendo que é automaticamente fechado para novas inscrições depois de iniciado e/ou terminado. Um Sprint pode ser de 2 tipos ou ambos, sendo eles: virtual, físico e ambos.

Quando um Sprint for do tipo virtual é possível o cadastro de locais virtuais de encontros para a comunicação, tais como canais de IRC (Internet Relay Chat), ou outro mensageiro em tempo real. Quando for do tipo físico tem-se a possibilidade de uma indicação visual através de mapa rodoviário.

Cada Sprint possui a seguinte lista de funcionalidades:

- Marcação do local físico via API do google maps
- Visualização de gráficos estatísticos no decorrer do Sprint
- Disparo automático de e-mail de aviso de início do Sprint

Objetivo

Compreende-se um marco dentro de um Sprint, seja ele a resolução de uma falha, um conjunto de falhas associadas à uma certa funcionalidade ou até mesmo uma nova funcionalidade. Ele faz referência ao Sprint através do qual o mesmo foi criado e possui uma descrição sobre o tal marco definido pela equipe de desenvolvedores do projeto.

As entidades que representam um objetivo são passíveis das seguintes alterações:

- Alteração da descrição do objetivo
- Sinalização de objetivo alcançado

3.4.3 Usuário

A entidade de um usuário representa a pessoa que fará a utilização do sistema, seja ela um administrador (dono do projeto), um moderador, um colaborador ou um visitante. Um usuário pode estar associado (participar) à um ou mais projetos.

3.5 Dados

Uma vez definido o problema o primeiro passo no processo de desenvolvimento foi o de levantar quais os dados pertencentes ao domínio da aplicação deveriam estar armazenados para

serem utilizados pela aplicação.

A implementação da representação dos dados foi feita diretamente através do ORM fornecido pelo Google Apps Engine, bastando para cada modelo criar uma classe e descrever seus atributos, tudo isso na sintaxe da linguagem Python.

3.5.1 Modelo: Projeto

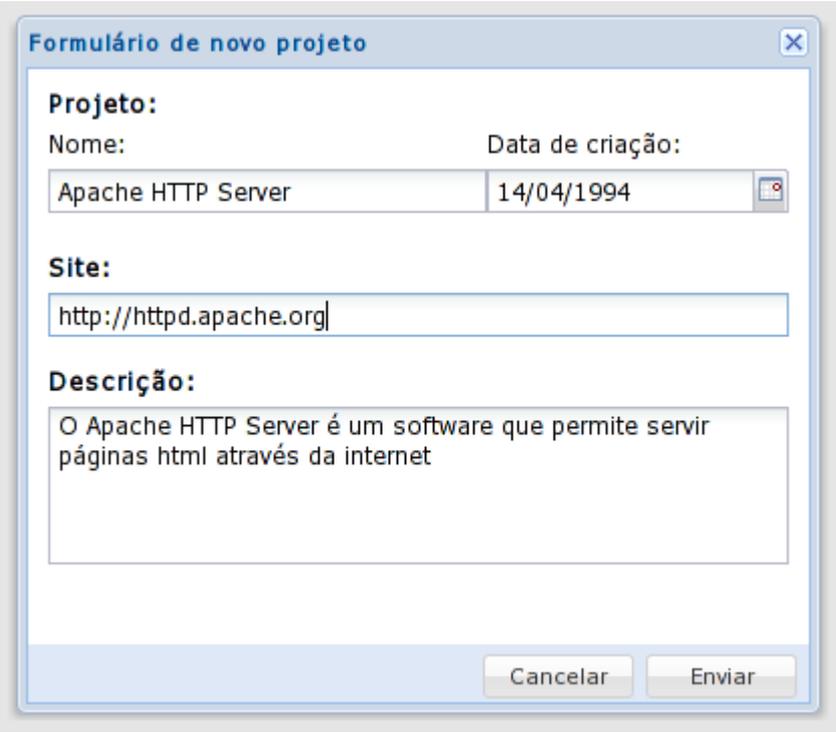
Este modelo representa um projeto cadastrado no sistema onde são armazenados os seguintes dados:

- Nome do projeto: atributo "name", obrigatório;
- Descrição do projeto: atributo "description", obrigatório;
- URL do projeto: atributo "url", opcional;
- Criador: atributo "owner", obrigatório, selecionado a partir do usuário logado no momento da criação do projeto;

3.5.2 Modelo: Sprint

Este modelo representa um Sprint cadastrado no sistema onde são armazenados os seguintes dados:

- Nome do Sprint: atributo "name", obrigatório;
- Tipo do Sprint: atributo "s_type", obrigatório, define se é virtual, "físico" ou "ambos";
- Data de início: atributo "begin", obrigatório;
- Data de término: atributo "end", obrigatório;
- Local físico: atributo "p_location", obrigatório caso o tipo for "físico" ou "ambos"
- Local virtual: atributo "v_location", obrigatório caso o tipo for "virtual" ou "ambos"



Formulário de novo projeto

Projeto:

Nome: Apache HTTP Server Data de criação: 14/04/1994

Site:

http://httpd.apache.org

Descrição:

O Apache HTTP Server é um software que permite servir páginas html através da internet

Cancelar Enviar

Figura 3.1: Tela de cadastro de um novo projeto

Modelo: Objetivo

Este modelo representa um objetivo específico de um Sprint cadastrado no sistema onde são armazenados os seguintes dados:

- Referência ao Sprint: atributo "sprint", obrigatório, este é uma chave estrangeira para um Sprint;
- Descrição do objetivo: atributo "description", obrigatório;

3.5.3 Modelo: Usuário

O modelo Usuário representa a pessoa que está acessando o sistema, independente dos níveis de permissão de acesso que a mesma venha a possuir, todas elas são representadas por este mesmo modelo.

- Nome do Usuário: atributo "name", obrigatório;
- Login do Usuário: atributo "username", obrigatório, é utilizado no campo de login;
- Senha: atributo "password", obrigatório, é utilizado no campo de login;
- Data de criação: atributo "creation_date", obrigatório (porém não será visível ao usuário);
- E-mail: atributo "email", obrigatório;
- Nível de acesso: atributo "access_level", obrigatório, é um elemento do conjunto (admin, moderador, colaborador, visitante);

3.5.4 Exemplo do modelo Sprint implementado no ORM

```
class Sprint(db.Model):
    name = db.StringProperty(
        verbose_name=_('nome'),
        required=True,
```

```

)
s_type = db.StringProperty(
    verbose_name=_(u'tipo'),
    required=True,
    choices=set([_(u'físico'), _(u'virtual'), _(u'ambos')])
)
begin = db.DateTimeProperty(
    verbose_name=_(u'data de abertura'),
    required=True
)
end = db.DateTimeProperty(
    verbose_name=_(u'data de encerramento'),
    required=True
)

```

3.6 Interface Gráfica

O desenvolvimento da parte gráfica com que o usuário interage com o sistema é uma tarefa que demanda muito tempo dos programadores, uma vez que ela precisa refletir visualmente os dados armazenados em uma aplicação e por vezes até mesmo sintetizar esses dados em informações para o usuário.

O framework Google Web Toolkit foi um fator culminante no desenvolvimento das telas dessa aplicação, uma vez que traz consigo meios para utilização de Ajax e ferramentas de ajuda à internacionalização e localização do software. Combinado com o framework ExtJS trouxe uma interface gráfica rica em widgets, visualmente moderna e portátil entre computadores pessoais e dispositivos móveis. Este framework também traz validação dos dados no lado do cliente reduzindo assim parte do processamento no lado do servidor. (Mostrado nas figuras 3.2 e 3.3)

Outro fator de vantagem na utilização dos dois frameworks escolhidos foi a presença de mecanismos de portabilidade entre diferentes versões e modelos de navegadores trazendo consistência na experiência vivenciada por diferentes usuários da ferramenta.

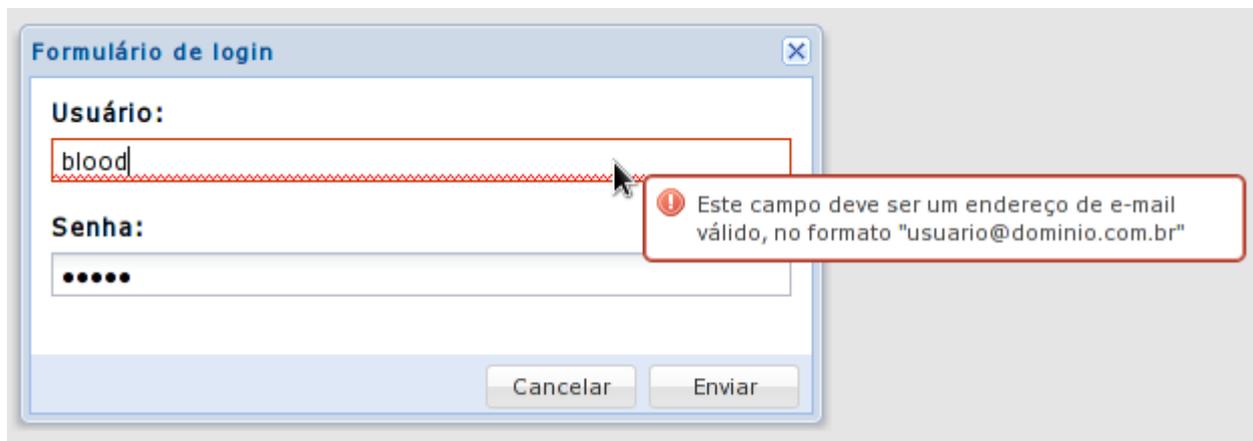


Figura 3.2: Validação no lado do cliente

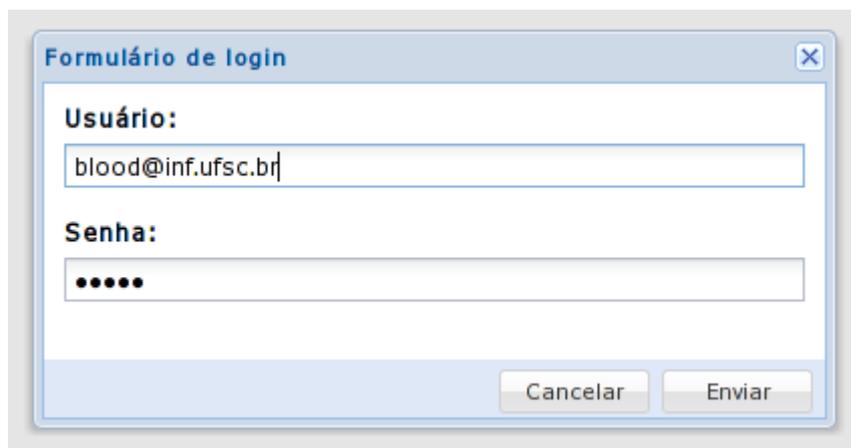


Figura 3.3: Campo validado

3.7 Instalação

3.7.1 Requisitos

São requisitos para a instalação do sistema em servidor local alguns componentes, são eles (lista de pacotes para distribuições Red Hat e derivadas):

- git
- python >= 2.5
- python-sqlite2
- sqlite >= 3

Também é necessário que se tenha conectividade com a internet, pois alguns componentes de terceiros são utilizados diretamente via web.

Os pacotes são mínimos uma vez que grande parte das bibliotecas necessárias estão incluídas próprio fonte do projeto. O código fonte e as bibliotecas se encontram no site:

`https://github.com/loop0/FOREST`

3.7.2 Instalação (local)

Para baixar o código fonte do projeto deve-se utilizar o git (uma ferramenta de controle de versão). Deve-se escolher um diretório onde serão baixados os fontes e executar a seguinte linha de comando:

```
git clone https://github.com/loop0/FOREST.git
```

3.7.3 Execução

Para colocar o servidor da aplicação em execução (funcionamento), deve-se seguir o passo anterior de instalação e após ter baixado o código fonte executar os seguintes comandos em linha de comando:

```
cd FOREST
source ./vars
cd forest
./manage.py runserver
```

Após a execução dos comandos acima será mostrado na tela alguns avisos e mensagens de inicialização.

Basta-se então abrir um navegador e apontar a url para o endereço <http://localhost:8080> e a primeira tela da aplicação deve aparecer para ser iniciada a utilização do sistema.

3.7.4 Instalação (Google App Engine)

Para instalação do sistema em uma conta do Google App Engine deve-se baixar o código fonte como descrito na seção Instalação (local) e editar o seguinte arquivo:

```
1 application: sprintogo
2 version: 1
3 runtime: python
4 api_version: 1
5
6 builtins:
7 - remote_api: on
8
9 inbound_services:
10 - warmup
11
12 handlers:
13 - url: /_ah/queue/deferred
14   script: djangoappengine/deferred/handler.py
15   login: admin
16
17 - url: /_ah/stats/.*
18   script: djangoappengine/appstats/ui.py
19
20 - url: /media/admin
21   static_dir: django/contrib/admin/media
```

```
22 expiration: '0'  
23  
24 - url: /*  
25 script: djangoappengine/main/main.py
```

FOREST/forest/app.yaml

Deve-se então mudar a primeira linha deste arquivo alterando o nome da aplicação de sprinto para o nome da sua aplicação no Google App Engine, feito isto basta executar os seguintes comandos para fazer o deploy nos servidores do Google:

```
cd FOREST  
source ./vars  
cd forest  
./manage.py deploy
```

Ao último comando será perguntado pelo seu login e senha no Google App Engine, deve-se entrar com os dados corretos e aguardar o final da instalação.

4 CONCLUSÃO

O objetivo deste trabalho foi buscar as deficiências nas atuais ferramentas utilizadas em um Sprint e reunir estas informações para o projeto de uma ferramenta de gerência completa e centralizada, que pudesse eliminar os pontos de ruptura da comunicação entre os grupos de desenvolvedores envolvidos em projetos de Software Livre que adotam a prática do Sprint como forma de evoluir e corrigir problemas em sua aplicação.

Para tanto foram-se pesquisados entre projetos de Software Livre o tipo de ferramentas e o fluxo adotado durante os eventos do Sprint. Feito o levantamento dos dados necessários partiu-se para a parte de estudo em diversas áreas disciplinares tais como engenharia de software, banco de dados, sistemas operacionais, internacionalização e localização de software, a fim de se formar uma base para o projeto da aplicação proposta como gerência de Sprints.

Na solução do problema de gerência de Sprints foi proposta uma aplicação que tivesse os elementos de comunicação necessários à uma organização centralizada, tal como informações dos Sprints de locais físicos, dos Sprints de locais virtuais, link para página do projeto, controle dos objetivos, quadro de avisos e disparo de e-mail para os envolvidos.

O desenvolvimento de um software desde seu projeto até sua implementação e testes demonstrou-se que esta é uma atividade que pode trazer à tona conhecimentos interdisciplinares aprendidos e amadurecidos ao longo de todo o curso da faculdade e o quanto se faz importante uma formação própria nesta área da tecnologia.

O tipo de iniciativa que uma empresa como a Google tem ao disponibilizar diversas ferramentas para uso gratuito mostra a importância de impulsionar a indústria de software a gastar mais tempo nos problemas do que na implementação das aplicações.

As atividades de pesquisa foram muito importantes na descoberta de ferramentas que podem

trazer agilidade no processo de desenvolvimento, no processo de internacionalização e localização, no processo de documentação e na esfera dos testes de aplicação.

Principalmente a criação de testes se mostrou indispensável em qualquer ambiente de produção de aplicações, sem os mesmos se torna muito difícil a tarefa de depuração e a garantia da consistência das aplicações.

4.1 Futuro

4.1.1 Futuras implementações

Para futuras implementações foi levantado durante o desenvolvimento do projeto que poderia ser interessante adicionar a funcionalidade da importação direta dos objetivos do sprint a partir de sistemas utilizados juntamente com os repositórios de código e sistemas de controle de versão.

Para trabalhos futuros fica a implementação de um plugin para importação dos objetivos diretamente da ferramenta Trac, que é utilizada para gerenciar projetos que fazem uso do Subversion (sistema de controle de versões de código).

REFERÊNCIAS BIBLIOGRÁFICAS

- ALEXANDER, C. *A Pattern Language*. [S.l.]: Oxford, 1977.
- BARAHONA, J. M. G.; PASCUAL, J. S.; ROBLES, G. *Introduction to Free Software*. [S.l.]: Free Technology Academy, 2009.
- BRIGHT GREEN PROJECTS. *Bright Green Projects*. Disponível em: <<http://brightgreenprojects.com>>.
- BROOKS, F. *The Mythical Man-Month: essays on software engineering*. [S.l.]: Addison Wesley Longman, 1995.
- DJANGO SOFTWARE FOUNDATION. *Django: The web framework for perfectionists with deadlines*. Disponível em: <<http://www.djangoproject.com>>.
- DJANGO SOFTWARE FOUNDATION. *Sprints - Django*. Disponível em: <<https://code.djangoproject.com/wiki/Sprints>>.
- FOWLER, M. et al. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison Wesley, 2002.
- HACKATHON - Wikipedia the free encyclopedia. Disponível em: <<http://en.wikipedia.org/wiki/Hackathon>>.
- KUNAGI. *kunagi*. Disponível em: <<http://kunagi.org>>.
- OPENBSD hackatons. Disponível em: <<http://www.openbsd.org/hackathons.html>>.
- PYPY: site oficial. Disponível em: <<http://pypy.org>>.
- SCRUMDO. *ScrumDo*. Disponível em: <<http://www.scrumdo.com>>.
- STALLMAN, R. M. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. [S.l.]: Free Software Foundation, 2002.
- TELERIK. *Team Pulse*. Disponível em: <<http://www.telerik.com/agile-project-management-tools.aspx>>.
- TELES, V. M. *Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade*. [S.l.]: Novatec, 2004.

TENFORCE. *pmScrum*. Disponível em: <<http://www.tenforce.com/pmscrum>>.

WOODWARD, E.; SURDEK, S.; GANIS, M. *A practical guide to distributed Scrum*. [S.l.]: IBM Press, 2010.

ZOPE CORPORATION. *Zope*. Disponível em: <<http://www.zope.org>>.

ZOPE CORPORATION. *Zope 3 wiki Sprint Schedule*. Disponível em: <<http://wiki.zope.org/zope3/SprintSchedule>>.

APÊNDICE A - CÓDIGO FONTE

```
1 # -*- coding: utf-8 -*-
2
3 # django imports
4 from django.http import HttpResponse
5 from django.template import RequestContext
6 from django.shortcuts import render_to_response as render
7
8 def home(request):
9
10     return render('index.html',
11                  {},
12                  context_instance=RequestContext(request)
13                 )
14
15 def profile(request):
16
17     return render('index.html',
18                  {},
19                  context_instance=RequestContext(request)
20                 )
21
22 def login(request):
23
24     return render('common_login_success.html',
25                  {},
26                  context_instance=RequestContext(request)
27                 )
28
```

```

29 def verification(request):
30     return HttpResponseRedirect("google-site-verification: google6260374b9af7fe79.
        html")

```

FOREST/forest/common/views.py

```

1 from django.conf.urls.defaults import patterns, include, url
2 from django.conf import settings
3
4 urlpatterns = patterns('common.views',
5     url(r'^$', 'home', name='common_index'),
6     url(r'^google6260374b9af7fe79.html$', 'verification', name='
        common_verification'),
7     url(r'^login/success/$', 'login', name='common_login_success'),
8     url(r'^profile/$', 'profile', name='common_profile'),
9 )
10
11 urlpatterns += patterns('django.contrib.auth.views',
12     url(r'^login/$', 'login', {'template_name': 'common_login.html'}, name='
        common_login'),
13     url(r'^logout/$', 'logout', {'next_page': '/'}, name='common_logout'),
14 )

```

FOREST/forest/common/urls.py

```

1 Ext.require([
2     'Ext.form.*'
3 ]);
4
5 Ext.onReady(function() {
6
7     var win;
8
9     showSprintForm = function showSprintForm() {
10         if (!win) {
11             var form = Ext.widget('form', {
12                 layout: {
13                     type: 'vbox',
14                     align: 'stretch'

```

```
15     },
16     border: false ,
17     bodyPadding: 10,
18
19     fieldDefaults: {
20         labelAlign: 'top',
21         labelWidth: 100,
22         labelStyle: 'font-weight:bold'
23     },
24     defaults: {
25         margins: '0 0 10 0'
26     },
27
28     items: [{
29         name: 'name',
30         xtype: 'textfield',
31         fieldLabel: 'Nome do Sprint',
32         allowBlank: false
33     },new Ext.form.ComboBox({
34         name: 's_type',
35         editable: false ,
36         disableKeyFilter: true ,
37         forceSelection: true ,
38         emptyText: 'selecione um...',
39         triggerAction: 'all',
40         mode: 'local',
41         store: new Ext.data.SimpleStore({
42             id: 0,
43             fields: ['value', 'text'],
44             data : [
45                 ['virtual', 'virtual'],
46                 ['physical', 'f sico'],
47                 ['both', 'virtual + f sico']
48             ]
49         }),
50         valueField: 'value',
51         displayField: 'text',
52         hiddenName: 's_type',
```

```
53         fieldLabel: 'Tipo do Sprint'
54     }),{
55         xtype: 'fieldcontainer',
56         fieldLabel: 'Período (início)',
57         labelStyle: 'font-weight:bold;padding:0',
58         layout: 'hbox',
59         defaultType: 'textfield',
60
61         fieldDefaults: {
62             labelAlign: 'top'
63         },
64         items: [{
65             name: 'begin_date',
66             fieldLabel: 'Data',
67             xtype: 'datefield',
68         },{
69             name: 'begin_time',
70             fieldLabel: 'Hora',
71             xtype: 'timefield',
72             format: 'H:i'
73         }]
74     },{
75         xtype: 'fieldcontainer',
76         fieldLabel: 'Período (término)',
77         labelStyle: 'font-weight:bold;padding:0',
78         layout: 'hbox',
79         defaultType: 'textfield',
80
81         fieldDefaults: {
82             labelAlign: 'top'
83         },
84         items: [{
85             name: 'end_date',
86             fieldLabel: 'Data',
87             xtype: 'datefield',
88         },{
89             name: 'begin_date',
90             fieldLabel: 'Hora',
```

```
91         xtype: 'timefield',
92         format: 'H:i'
93     }],
94 },
95
96     buttons: [{
97         text: 'Cancelar',
98         handler: function () {
99             this.up('form').getForm().reset();
100             this.up('window').hide();
101         }
102     }, {
103         text: 'Enviar',
104         handler: function () {
105             if (this.up('form').getForm().isValid()) {
106                 // In a real application, this would submit the
107                 form to the configured url
108                 this.up('form').getForm().submit({
109                     url: '/p/project/save/',
110                     waitMsg: 'Salvando dados do Sprint',
111                     waitTitle: 'Por favor aguarde...',
112                     success: function () {
113                         win.hide();
114                         Ext.MessageBox.alert('Sucesso!', 'Seu
115                             Sprint foi cadastrado com sucesso!');
116                         window.location.reload();
117                     },
118                 });
119             }
120         }
121     }
122 });
123
124 win = Ext.widget('window', {
125     title: 'Formulário de novo sprint',
126     closeAction: 'hide',
127     width: 400,
128     height: 400,
```

```

127         minHeight: 400,
128         layout: 'fit',
129         resizable: true,
130         modal: true,
131         items: form
132     });
133
134     // load fields from server
135     //form.getForm().load({ url: '/p/project/load/', method: 'GET' });
136 }
137 win.show();
138 }
139
140 });

```

FOREST/forest/media/js/sprint-form.js

```

1 Ext.require([
2     'Ext.form.*'
3 ]);
4
5 Ext.onReady(function() {
6     var win;
7
8     showLoginForm = function showLoginForm() {
9         if (!win) {
10            var form = Ext.widget('form', {
11                layout: {
12                    type: 'vbox',
13                    align: 'stretch'
14                },
15                border: false,
16                bodyPadding: 10,
17
18                fieldDefaults: {
19                    labelAlign: 'top',
20                    labelWidth: 100,
21                    labelStyle: 'font-weight:bold'
22                },

```

```
23     defaults: {
24         margins: '0 0 10 0'
25     },
26
27     items: [{
28         name: 'username',
29         xtype: 'textfield',
30         fieldLabel: 'Usu&aacute;rio',
31         vtype: 'email',
32         allowBlank: false
33     }, {
34         name: 'password',
35         xtype: 'textfield',
36         fieldLabel: 'Senha',
37         allowBlank: false,
38         inputType: 'password'
39     }],
40
41     buttons: [{
42         text: 'Cancelar',
43         handler: function () {
44             this.up('form').getForm().reset();
45             this.up('window').hide();
46         }
47     }, {
48         text: 'Enviar',
49         handler: function () {
50             if (this.up('form').getForm().isValid()) {
51                 this.up('form').getForm().submit({
52                     url: '/login/',
53                     waitMsg: 'Realizando login no sistema',
54                     waitTitle: 'Por favor aguarde...',
55                     success: function () {
56                         win.hide();
57                         Ext.MessageBox.alert('Sucesso!', 'Seu
58                             login foi efetuado com sucesso!');
59                         window.location.reload();
60                     }
61                 });
62             }
63         }
64     }
65     ],
66     win.hide();
67     Ext.MessageBox.alert('Sucesso!', 'Seu login foi efetuado com sucesso!');
68     window.location.reload();
69 }
```

```

60         });
61     }
62 }
63 ]]
64 });
65
66     win = Ext.widget('window', {
67         title: 'Formul&aacute;rio de login',
68         closeAction: 'hide',
69         width: 400,
70         height: 200,
71         minHeight: 200,
72         layout: 'fit',
73         resizable: true,
74         modal: true,
75         items: form
76     });
77 }
78 win.show();
79 }
80 });

```

FOREST/forest/media/js/login-form.js

```

1 Ext.require([
2     'Ext.form.*'
3 ]);
4
5 Ext.onReady(function() {
6
7     var win;
8
9     function showProjectForm() {
10         if (!win) {
11             var form = Ext.widget('form', {
12                 layout: {
13                     type: 'vbox',
14                     align: 'stretch'
15                 },

```

```
16         border: false ,
17         bodyPadding: 10,
18
19         fieldDefaults: {
20             labelAlign: 'top',
21             labelWidth: 100,
22             labelStyle: 'font-weight:bold'
23         },
24         defaults: {
25             margins: '0 0 10 0'
26         },
27
28         items: [{
29             xtype: 'fieldcontainer',
30             fieldLabel: 'Projeto',
31             labelStyle: 'font-weight:bold;padding:0',
32             layout: 'hbox',
33             defaultType: 'textfield',
34
35             fieldDefaults: {
36                 labelAlign: 'top'
37             },
38             items: [
39                 {
40                     flex: 1,
41                     name: 'name',
42                     fieldLabel: 'Nome',
43                     allowBlank: false
44                 }, {
45                     name: 'creation_date',
46                     fieldLabel: 'Data de cria&ccedil;&atilde;o',
47                     xtype: 'datefield',
48                     maxValue: new Date()
49                 }
50             ],{
51                 name: 'url',
52                 xtype: 'textfield',
53                 fieldLabel: 'Site',
```

```

54         vtype: 'url',
55         allowBlank: true
56     },{
57         name: 'description',
58         xtype: 'textareafield',
59         fieldLabel: 'Descri&ccedil;& atilde;o',
60         allowBlank: false,
61     },{
62         name: 'id',
63         xtype: 'hiddenfield',
64         allowBlank: false,
65     }],
66
67     buttons: [{
68         text: 'Cancelar',
69         handler: function() {
70             this.up('form').getForm().reset();
71             this.up('window').hide();
72         }
73     }, {
74         text: 'Enviar',
75         handler: function() {
76             if (this.up('form').getForm().isValid()) {
77                 // In a real application, this would submit the
78                 form to the configured url
79                 this.up('form').getForm().submit({
80                     url: '/s/project/save/',
81                     waitMsg: 'Salvando dados do projeto',
82                     waitTitle: 'Por favor aguarde...'
83                 });
84                 // this.up('window').hide();
85                 //Ext.MessageBox.alert('Thank you!', 'Your
86                 inquiry has been sent. We will respond as
87                 soon as possible.');
```

```

89
90     win = Ext.widget('window', {
91         title: 'Formul&aacute;rio de novo projeto',
92         closeAction: 'hide',
93         width: 400,
94         height: 350,
95         minHeight: 350,
96         layout: 'fit',
97         resizable: true,
98         modal: true,
99         items: form
100     });
101
102     // load fields from server
103     form.getForm().load({ url: '/s/project/load/', method: 'GET' });
104 }
105 win.show();
106 }
107
108 showProjectForm();
109 });

```

FOREST/forest/media/js/objective-form.js

```

1 Ext.require([
2     'Ext.form.*'
3 ]);
4
5 Ext.onReady(function() {
6
7     var win;
8
9     showProjectForm = function showProjectForm() {
10         if (!win) {
11             var form = Ext.widget('form', {
12                 layout: {
13                     type: 'vbox',
14                     align: 'stretch'
15                 },

```

```
16         border: false ,
17         bodyPadding: 10,
18
19         fieldDefaults: {
20             labelAlign: 'top',
21             labelWidth: 100,
22             labelStyle: 'font-weight:bold'
23         },
24         defaults: {
25             margins: '0 0 10 0'
26         },
27
28         items: [{
29             xtype: 'fieldcontainer',
30             fieldLabel: 'Projeto',
31             labelStyle: 'font-weight:bold;padding:0',
32             layout: 'hbox',
33             defaultType: 'textfield',
34
35             fieldDefaults: {
36                 labelAlign: 'top'
37             },
38             items: [
39                 {
40                     flex: 1,
41                     name: 'name',
42                     fieldLabel: 'Nome',
43                     allowBlank: false
44                 }, {
45                     name: 'creation_date',
46                     fieldLabel: 'Data de cria&ccedil;&atilde;o',
47                     xtype: 'datefield',
48                     maxValue: new Date()
49                 }
50             ],{
51                 name: 'url',
52                 xtype: 'textfield',
53                 fieldLabel: 'Site',
```

```
54         vtype: 'url',
55         allowBlank: true
56     },{
57         name: 'description',
58         xtype: 'textareafield',
59         fieldLabel: 'Descri&ccedil;&atilde;o',
60         allowBlank: false,
61     },{
62         name: 'id',
63         xtype: 'hiddenfield',
64         allowBlank: false,
65     }],
66
67     buttons: [{
68         text: 'Cancelar',
69         handler: function() {
70             this.up('form').getForm().reset();
71             this.up('window').hide();
72         }
73     }, {
74         text: 'Enviar',
75         handler: function() {
76             if (this.up('form').getForm().isValid()) {
77                 // In a real application, this would submit the
78                 form to the configured url
79                 this.up('form').getForm().submit({
80                     url: '/p/project/save/',
81                     waitMsg: 'Salvando dados do projeto',
82                     waitTitle: 'Por favor aguarde...',
83                     success: function() {
84                         win.hide();
85                         Ext.MessageBox.alert('Sucesso!', 'Seu
86                             projeto foi cadastrado com sucesso!')
87                             ;
88                         window.location.reload();
89                     }
90                 });
91             }
92         }
93     }
94     ]
95 }
```

```

89         }
90     }}
91 });
92
93     win = Ext.widget('window', {
94         title: 'Formul&aacute;rio de novo projeto',
95         closeAction: 'hide',
96         width: 400,
97         height: 350,
98         minHeight: 350,
99         layout: 'fit',
100        resizable: true,
101        modal: true,
102        items: form
103    });
104
105        // load fields from server
106        //form.getForm().load({ url: '/p/project/load/', method: 'GET' });
107    }
108    win.show();
109 }
110
111 });

```

FOREST/forest/media/js/project-form.js

```

1 # -*- coding: utf-8 -*-
2
3 # python imports
4 import datetime
5 import logging
6
7 # django imports
8 from django.http import HttpResponse
9 from django.shortcuts import render_to_response as render
10 from django.template import RequestContext
11 from django.contrib.auth.decorators import login_required
12
13 # app imports

```

```
14 from sprint.models import Project
15 from sprint.forms import ProjectForm
16 from utils.http import JsonResponse
17 from utils.models import model_to_json
18
19
20 def project(request, project_id=None):
21     project = Project.objects.get(id=project_id)
22     return render('project.html', {
23         'project': project,
24     },
25         context_instance=RequestContext(request)
26     )
27
28 def project_load(request, project_id=None):
29     project = Project.objects.get()
30     content = {
31         'success': True,
32         'data': model_to_json(project)
33     }
34
35     return JsonResponse(content)
36
37 @login_required
38 def project_save(request):
39     if request.method == 'POST':
40         id = request.POST.get('id')
41         if id:
42             instance = Project.objects.get(id=id)
43             form = ProjectForm(request.POST, instance=instance)
44         else:
45             form = ProjectForm(request.POST)
46
47
48     if form.is_valid():
49         project = form.save(commit=False)
50         project.owner = request.user
51         project.save()
```

```

52         return JsonResponse({'success': True})
53     else:
54         return JsonResponse({'success': False, 'errors': form.errors
55                               })
56 @login_required
57 def list_projects(request):
58     user_projects = request.user.project_set.all()
59     projects = Project.objects.all()
60
61     return render('projects.html', {
62         'user_projects': user_projects,
63         'projects': projects,
64     },
65         context_instance=RequestContext(request)
66     )
67
68 @login_required
69 def list_sprints(request, project_id=None):
70     project = Project.objects.get(id=project_id)
71     sprints = project.sprint_set.all()
72
73     return render('sprints.html', {
74         'project': project,
75         'sprints': sprints
76     },
77         context_instance=RequestContext(request)
78     )

```

FOREST/forest/sprint/views.py

```

1 # -*- coding: utf-8 -*-
2
3 # django imports
4 from django import forms
5
6 # app imports
7 from sprint.models import Project
8

```

```

9 class ProjectForm(forms.ModelForm):
10     creation_date = forms.DateField(input_formats=( '%d/%m/%Y' ,))
11     class Meta:
12         model = Project
13         exclude = ('owner',)

```

FOREST/forest/sprint/forms.py

```

1 # -*- coding: utf-8 -*-
2 from django.db import models
3 from django.utils.translation import ugettext_lazy as _
4 from django.contrib.auth.models import User
5
6 class Project(models.Model):
7     name = models.CharField(
8         verbose_name=_(u'nome do projeto'),
9         max_length=255,
10    )
11     creation_date = models.DateField(
12         verbose_name=_(u'data de criação'),
13    )
14     url = models.URLField(
15         verbose_name=_(u'site'),
16         blank=True,
17    )
18     description = models.TextField(
19         verbose_name=_(u'descrição'),
20    )
21     owner = models.ForeignKey(
22         User,
23         verbose_name=_(u'dono')
24    )
25
26     def __unicode__(self):
27         return self.name
28
29     class Meta:
30         verbose_name = _(u'projeto')
31         verbose_name_plural = _(u'projetos')

```

```
32     ordering = ('name',)
33
34 class Sprint(models.Model):
35     TYPES = (
36         ('virtual', _(u'virtual')),
37         ('physical', _(u'f sico')),
38         ('both', _(u'virtual e f sico')),
39     )
40     project = models.ForeignKey(
41         Project,
42         verbose_name=_(u'projeto'),
43     )
44     name = models.CharField(
45         verbose_name=_(u'nome do sprint'),
46         max_length=100,
47     )
48     s_type = models.CharField(
49         verbose_name=_(u'tipo do sprint'),
50         max_length=8,
51         choices=TYPES,
52     )
53     p_location = models.TextField(
54         verbose_name=_(u'informa es do local f sico'),
55     )
56     v_location = models.TextField(
57         verbose_name=_(u'informa es do local virtual'),
58     )
59     begin_date = models.DateTimeField(
60         verbose_name=_(u'data de in cio'),
61     )
62     end_date = models.DateTimeField(
63         verbose_name=_(u'data de t rmino'),
64     )
65
66     def __unicode__(self):
67         return self.name
68
69     class Meta:
```

```

70     verbose_name = _(u' sprint ')
71     verbose_name_plural = _(u' sprints ')
72
73 class Objective(models.Model):
74     sprint = models.ForeignKey(
75         Sprint,
76         verbose_name=_(u' sprint '),
77     )
78     description = models.TextField(
79         verbose_name=_(u' descri    o '),
80     )
81     done = models.BooleanField(
82         verbose_name=_(u' finalizado '),
83         default=False,
84         blank=True,
85     )
86
87     def __unicode__(self):
88         return self.description
89
90     class Meta:
91         verbose_name = _(u' objetivo ')
92         verbose_name_plural = _('objetivos')

```

FOREST/forest/sprint/models.py

```

1 from django.conf.urls.defaults import patterns, include, url
2 from django.conf import settings
3
4 urlpatterns = patterns('sprint.views',
5     url(r'^$', 'list_projects', name='sprint_projects'),
6     url(r'^project/(?P<project_id>\d+)/$', 'project', name='sprint_project')
7     ,
8     url(r'^project/(?P<project_id>\d+)/s/$', 'list_sprints', name='
9     sprint_sprints'),
10    url(r'^project/load/$', 'project_load', name='sprint_project_load'),
11    url(r'^project/save/$', 'project_save', name='sprint_project_save'),
12 )

```

FOREST/forest/sprint/urls.py

```
1 {% extends "base.html" %}
2
3 {% block menu %}
4     <ul>
5         <li><a href="{% url common_index %}">In&iacute;cio </a></li>
6         <li><a href="{% url sprint_projects %}">Projetos </a></li>
7         {% if user.is_authenticated %}
8             <li class="current_page_item"><a href="{% url sprint_project
9                 project.id %}">{{ project.name }}</a></li>
10            <li><a href="{% url common_profile %}">Perfil </a></li>
11        {% endif %}
12        <li><a href="{% url common_logout %}">Sair </a></li>
13    </ul>
14
15
16 {% block submenu %}
17
18     <ul>
19         {% if user.is_authenticated %}
20             <li><a href="{% url sprint_project project.id %}">Projeto </a></li>
21             <li class="current_page_item"><a href="{% url sprint_sprints project
22                 .id %}">Sprints </a></li>
23             <li><a href="#" id="new_sprint">Novo Sprint </a></li>
24         {% endif %}
25     </ul>
26
27
28
29 {% block content %}
30 {% for sprint in sprints %}
31     <div class="post">
32
33         <div class="post_title"><h2><a href="#">{{ sprint.name }}</a></h2></
```

```

        div>
34     <div class="post_body">
35
36     </div>
37
38     </div>
39 {% endfor %}
40 {% endblock %}
41
42 {% block sidebar %}
43     <div class="box">
44
45         <div class="box_title">Sobre </div>
46
47         <div class="box_content">
48             <p>FOREST &eacute; uma aplica&ccedil;&atilde;o para ger&ecirc;
                ncia desenvolvida como Trabalho de Conclu&atilde;o de Curso.
49             Ela serve de aux&iacute; na organiza&ccedil;&atilde;o de Sprints
                de software.</p>
50         </div>
51
52     </div>
53 {% endblock %}

```

FOREST/forest/sprint/templates/sprints.html

```

1 {% extends "base.html" %}
2
3 {% block menu %}
4     <ul>
5         <li><a href="{% url common_index %}">In&iacute;cio </a></li>
6         <li class="current_page_item"><a href="{% url sprint_projects %}">
            Projetos </a></li>
7         {% if user.is_authenticated %}
8             <li><a href="{% url common_profile %}">Perfil </a></li>
9         {% endif %}
10        <li><a href="{% url common_logout %}">Sair </a></li>
11    </ul>
12 {% endblock %}

```

```
13
14
15 {% block submenu %}
16
17     <ul>
18         <li class="current_page_item"><a href="{% url sprint_projects %}">
19             Listar projetos </a></li>
20         {% if user.is_authenticated %}
21         <li><a href="#" id="new_project" onClick="showProjectForm();">Novo
22             projeto </a></li>
23         {% for project in user_projects %}
24         <li><a href="{% url sprint_project project.id %}">{{ project.name
25             }}</a></li>
26         {% endfor %}
27         {% endif %}
28     </ul>
29
30 {% endblock %}
31
32 {% block content %}
33 {% for project in projects %}
34     <div class="post">
35         <div class="post_title"><h2><a href="#">{{ project.name }}</a></h2
36             ></div>
37         <div class="post_body">
38             <p>{{ project.description }}</p>
39         </div>
40     </div>
41 {% endfor %}
42 {% endblock %}
43
44 {% block sidebar %}
45 <div class="box">
```

```

47
48     <div class="box_title">Sobre </div>
49
50     <div class="box_content">
51         <p>FOREST &eacute; uma aplica&ccedil;&atilde;o para ger&ecirc;
          ncia desenvolvida como Trabalho de Conclu&atilde;o de Curso.
52         Ela serve de aux&iacute; na organiza&ccedil;&atilde;o de Sprints
          de software.</p>
53     </div>
54
55 </div>
56 {% endblock %}

```

FOREST/forest/sprint/templates/projects.html

```

1 {% extends "base.html" %}
2
3 {% block menu %}
4     <ul>
5         <li><a href="{% url common_index %}">In&iacute;cio </a></li>
6         <li><a href="{% url sprint_projects %}">Projetos </a></li>
7         {% if user.is_authenticated %}
8             <li class="current_page_item"><a href="{% url sprint_project
          project.id %}">{{ project.name }}</a></li>
9             <li><a href="{% url common_profile %}">Perfil </a></li>
10            {% endif %}
11            <li><a href="{% url common_logout %}">Sair </a></li>
12        </ul>
13 {% endblock %}
14
15
16 {% block submenu %}
17
18     <ul>
19         {% if user.is_authenticated %}
20             <li class="current_page_item"><a href="{% url sprint_project project
          .id %}">Projeto </a></li>
21             <li><a href="{% url sprint_sprints project.id %}">Sprints </a></li>
22             <li><a href="#" id="new_sprint" onClick="showSprintForm();">Novo

```

```

                Sprint </a></li >
23         {% endif %}
24     </ul >
25
26 {% endblock %}
27
28
29 {% block content %}
30
31     <div class="post">
32
33         <b>{{ project.name }}</b>
34         <br >
35         <br >
36         <div class="post_body">
37             <p><b>Data de cria&ccedil;&atilde;o:</b> {{ project.
                 creation_date | date: "d/m/Y" }}</p>
38             <p><b>URL:</b> <a href="{{ project.url }}" target="_blank">{{
                 project.url }}</a></p>
39             <p><b>Descri&ccedil;&atilde;o:</b> {{ project.description }}</p>
40
41         </div >
42
43     </div >
44 {% endblock %}
45
46 {% block sidebar %}
47     <div class="box">
48
49         <div class="box_title">Sobre </div >
50
51         <div class="box_content">
52             <p>FOREST &eacute; uma aplica&ccedil;&atilde;o para ger&ecirc;
                 ncia desenvolvida como Trabalho de Conclu&atilde;o de Curso.
53             Ela serve de aux&iacute; na organiza&ccedil;&atilde;o de Sprints
                 de software.</p>
54         </div >
55

```

```

56     </div>
57 {% endblock %}

```

FOREST/forest/sprint/templates/project.html

```

1  {% extends "base.html" %}
2
3  {% block menu %}
4      <ul>
5          <li class="current_page_item"><a href="{% url common_index %}">In&
            iacute;cio </a></li>
6          <li><a href="{% url sprint_projects %}">Projetos </a></li>
7          {% if user.is_authenticated %}
8              <li><a href="{% url common_profile %}">Perfil </a></li>
9              <li><a href="{% url common_logout %}">Sair </a></li>
10         {% endif %}
11     </ul>
12 {% endblock %}
13
14
15 {% block submenu %}
16     {% if not user.is_authenticated %}
17     <ul>
18         <li class="current_page_item"><a href="#" id="do_login" onClick="
            showLoginForm();">Fazer login </a></li>
19     </ul>
20     {% else %}
21     {% endif %}
22 {% endblock %}
23
24
25 {% block content %}
26     <div class="post">
27
28         <div class="post_title"><h2><a href="#">Bem vindo(a) </a></h2></div>
29         <div class="post_body">
30
31             <p>Seja bem vindo(a) ao FOREST, uma aplica&ccedil;&atilde;o para
                ger&ecirc;ncia de Sprints de software </p>

```

```

32
33     </div>
34
35 </div>
36 {% endblock %}
37
38 {% block sidebar %}
39     <div class="box">
40
41         <div class="box_title">Sobre</div>
42
43         <div class="box_content">
44             <p>FOREST &eacute; uma aplica&ccedil;&atilde;o para ger&ecirc;
                ncia desenvolvida como Trabalho de Conclu&atilde;o de Curso.
45             Ela serve de aux&iacute; na organiza&ccedil;&atilde;o de Sprints
                de software.</p>
46         </div>
47
48     </div>
49 {% endblock %}

```

FOREST/forest/templates/index.html

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr">
3
4 <head>
5     <title>FOREST</title>
6     <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
7     <meta name="description" content="" />
8     <meta name="keywords" content="" />
9     <meta name="author" content="" />
10
11     <!-- CSS -->
12     <link rel="stylesheet" type="text/css" href="{{ MEDIA_URL }}/css/style.
        css" media="screen" />
13     <link rel="stylesheet" type="text/css" href="http://cdn.sencha.io/ext
        -4.0.2a/resources/css/ext-all.css" />

```

```
14
15 <!-- ExtJS -->
16 <script type="text/javascript" charset="utf-8" src="http://cdn.sencha.io
    /ext-4.0.2a/ext-all.js"></script>
17 <script type="text/javascript" charset="utf-8" src="http://cdn.sencha.io
    /ext-4.0.2a/locale/ext-lang-pt_BR.js"></script>
18 <script type="text/javascript" charset="utf-8" src="http://cdn.sencha.io
    /ext-4.0.2a/ext-all.js"></script>
19
20 <!-- FOREST JS -->
21 <script type="text/javascript" src="{ { MEDIA_URL } }/js/login-form.js"></
    script>
22 {% if user.is_authenticated %}
23 <script type="text/javascript" src="{ { MEDIA_URL } }/js/project-form.js"
    ></script>
24 <script type="text/javascript" src="{ { MEDIA_URL } }/js/sprint-form.js"
    ></script>
25 {% endif %}
26 </head>
27
28 <body id="top">
29
30 <div id="layout_wrapper_outer">
31 <div id="layout_wrapper">
32
33 <div id="layout_top">
34
35 <div id="site_title">
36 <h1><a href="#">FOREST</span></a></h1>
37 <h2>Uma aplica&ccedil;&atilde;o para ger&ecirc;ncia de Sprints </
    h2>
38 </div>
39
40 </div>
41
42 <div id="layout_body_outer">
43 <div id="layout_body">
44
```

```
45 <div id="navigation">
46
47     <div id="nav1">
48         {% block menu %}
49             <ul>
50                 <li class="current_page_item"><a href="{% url
                    common_index %}">In&iacute;cio </a></li>
51                 <li><a href="{% url sprint_projects %}">Projetos </a></li
                    >
52                 {% if request.user %}
53                     <li><a href="{% url common_profile %}">Perfil </a></
                    li >
54                 {% endif %}
55             </ul>
56         {% endblock %}
57
58         <div class="clearer">&nbsp;</div>
59
60     </div>
61
62     <div id="nav2">
63         {% block submenu %}
64             <ul>
65                 <li class="current_page_item"><a href="{% url
                    common_login %}">Fazer login </a></li>
66             </ul>
67         {% endblock %}
68
69         <div class="clearer">&nbsp;</div>
70
71     </div>
72 </div>
73
74 <div id="main">
75
76     <div class="left" id="content_outer">
77         <div id="content">
78             {% block content %}
```

```
79         <div class="post">
80
81             <div class="post_title"><h2><a href="#">Bem vindo(a)
82                 </a></h2></div>
83             <div class="post_body">
84
85                 <p>Seja bem vindo(a) ao FOREST, uma aplica&
86                     ccedil;&atilde;o para ger&ecirc;ncia de
87                     Sprints de software </p>
88
89             </div>
90         </div>
91     </div>
92
93     <div class="right" id="sidebar_outer">
94         <div id="sidebar">
95             {% block sidebar %}
96                 <div class="box">
97
98                     <div class="box_title">Sobre </div>
99
100                    <div class="box_content">
101                        <p>FOREST &eacute; uma aplica&ccedil;&atilde;o
102                            para ger&ecirc;ncia desenvolvida como
103                            Trabalho de Conclu&atilde;o de Curso.
104                            Ela serve de aux&iacute; na organiza&ccedil;&
105                            atilde;o de Sprints de software.</p>
106                    </div>
107                </div>
108            {% endblock %}
109        </div>
110
111     <div class="clearer">&nbsp;</div>
```

```
111
112     </div>
113
114     <div id="dashboard">
115         <div id="dashboard_inner">
116
117             <div class="clearer">&nbsp;</div>
118
119         </div>
120     </div>
121
122 </div>
123 </div>
124
125 <div id="footer">
126
127     <div class="left">
128         &copy; 2011 sprintogo.appspot.com – FOREST
129     </div>
130     <div class="right">
131         <a href="http://templates.arcsin.se/">Website template </a> by <a
132             href="http://arcsin.se/">Arcsin </a>
133     </div>
134     <div class="clearer">&nbsp;</div>
135
136 </div>
137
138 </div>
139 </div>
140
141 </body>
142 </html>
```

FOREST/forest/templates/base.html

```
1 # -*- coding: utf-8 -*-
2
3 # python imports
```

```
4 from libs import simplejson as json
5
6 # django imports
7 from django.http import HttpResponse
8
9 class JsonResponse(HttpResponse):
10
11     def __init__(self, content='', mimetype='application/javascript', status
12                 =None, content_type=None):
13         content = json.dumps(content)
14         super(JsonResponse, self).__init__(content, mimetype, status,
15                                           content_type)
```

FOREST/forest/utils/http.py