

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**EVOLUÇÃO DE UMA FERRAMENTA PARA COLETA DE  
DADOS EM PESQUISAS QUANTITATIVAS**

**Lucas Barzan Demétrio**

**Florianópolis - SC**

**2022 - 2**

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Evolução de uma Ferramenta para Coleta de Dados  
em Pesquisas Quantitativas

Lucas Barzan Demétrio

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

Florianópolis - SC

2022 - 2

Lucas Barzan Demétrio

Evolução de uma Ferramenta para Coleta de Dados  
em Pesquisas Quantitativas

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

Orientador(a): Prof. Raul Sidnei Wazlawick

Banca examinadora

Patricia Vilain

Jean Carlo Rossa Hauck

## SUMÁRIO

<b>SUMÁRIO</b>	<b>3</b>
<b>LISTA DE FIGURAS</b>	<b>5</b>
<b>LISTA DE QUADROS</b>	<b>8</b>
<b>RESUMO</b>	<b>9</b>
<b>ABSTRACT</b>	<b>10</b>
<b>1. INTRODUÇÃO</b>	<b>11</b>
1.1. Objetivos	13
1.2. Metodologia de pesquisa	14
<b>2. REFERENCIAL TEÓRICO</b>	<b>15</b>
2.1. Questionários	15
2.2. Framework	23
2.3. Flutter	24
2.4. API	26
2.5. Flask	27
2.6. Firebase	28
2.8. Sage	29
<b>3. TRABALHOS CORRELATOS</b>	<b>33</b>
3.1. Análise dos requisitos	33
3.2. Desk Research	34
<b>4. DESENVOLVIMENTO</b>	<b>39</b>
4.1. Atualização de pacotes e ferramentas	39
4.2. Correção de erros	42
4.3. Aplicação web	43
4.4. Novos tipos de perguntas	45
4.4.1. Escala Likert	45
4.4.2. Data/horário	55
4.4.3. Data	60
4.4.4. Horário	63
4.5. Convites entre entrevistadores	65
4.5.1. Entidade de Convite	65
4.5.2. Lógica de negócio (business logic)	68
4.5.3. Rotas	69
4.5.4. Testes unitários	71
4.5.5. Aplicação	72
4.6. Resultados	78

<b>5. CONCLUSÃO</b>	<b>80</b>
5.1. Trabalhos futuros	80
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>82</b>
<b>APÊNDICE A - Artigo</b>	<b>84</b>
<b>APÊNDICE B - Código-fonte</b>	<b>104</b>

## LISTA DE FIGURAS

Figura 1: Exemplo de interface de máscara de questionário eletrônico	18
Figura 2: Exemplo de interface de aplicativo de questionário eletrônico	19
Figura 3: Exemplo de interface de plataforma de questionário on-line	19
Figura 4: Etapas da construção de uma pesquisa com questionário on-line	21
Figura 5: Flutter Framework	25
Figura 6: Estrutura de base de dados Firebase	28
Figura 7: Tela de Login do Sage	30
Figura 8: Tela de Cadastro de questionários do Sage	31
Figura 9: Tela de preenchimento de questionários do Sage	32
Figura 10: Telas de listagem de questionários e de sincronização do Sage	32
Figura 11: Git diff referente a atualização dos pacotes do novo Sage	40
Figura 12: Recorte da lista de erros de tipagem no projeto	42
Figura 13: Sage em execução no emulador	43
Figura 14: Tela inicial do novo Sage (versão web)	44
Figura 15: Tela de detalhe de questionário do novo Sage (versão web)	44
Figura 16: Tela de resposta de questionário do novo Sage (versão web)	45
Figura 17: Tela de criação de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão mobile)	51
Figura 18: Tela de criação de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão web)	51
Figura 19: Tela de resposta de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão mobile)	54
Figura 20: Tela de resposta de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão web)	54

Figura 21: Tela de criação de questionário do novo Sage, pergunta “Data/horário” (versão mobile)	56
Figura 22: Tela de criação de questionário do novo Sage, pergunta “Data/horário” (versão web)	56
Figura 23: Tela de resposta de questionário do novo Sage, pergunta “Data/horário” (versão mobile) – parte 1	58
Figura 24: Tela de resposta de questionário do novo Sage, pergunta “Data/horário” (versão mobile) – parte 2	58
Figura 25: Tela de resposta de questionário do novo Sage, pergunta “Data/horário” (versão web) – parte 1	59
Figura 26: Tela de resposta de questionário do novo Sage, pergunta “Data/horário” (versão web) – parte 2	59
Figura 27: Tela de criação de questionário do novo Sage, pergunta “Data” (versão mobile)	60
Figura 28: Tela de criação de questionário do novo Sage, pergunta “Data” (versão web)	61
Figura 29: Tela de resposta de questionário do novo Sage, pergunta “Data” (versão mobile)	62
Figura 30: Tela de resposta de questionário do novo Sage, pergunta “Data” (versão web)	62
Figura 31: Tela de criação de questionário do novo Sage, pergunta “Horário” (versão mobile)	63
Figura 32: Tela de criação de questionário do novo Sage, pergunta “Horário” (versão web)	64

Figura 33: Tela de resposta de questionário do novo Sage, pergunta “Horário” (versão mobile)	64
Figura 34: Tela de resposta de questionário do novo Sage, pergunta “Horário” (versão web)	65
Figura 35: Diagrama Entidade-Relacionamento do banco de dados do servidor do novo Sage	66
Figura 36: Diagrama de estados de um Convite no novo Sage	68
Figura 37: Tela de detalhe de questionário do novo Sage, detalhe menu (versão mobile)	73
Figura 38: Tela de compartilhamento de questionário do novo Sage (versão mobile)	74
Figura 39: Tela de compartilhamento de questionário do novo Sage, detalhe menus (versão mobile)	75
Figura 40: Tela de compartilhamento de questionário do novo Sage, detalhe diálogo novo convite (versão mobile)	76
Figura 41: Tela inicial do novo Sage, detalhe menu; e Tela de convites recebidos no novo Sage (versão mobile)	77
Figura 42: Tela de convites recebidos no novo Sage, detalhe menu (versão mobile)	78

## LISTA DE QUADROS

Quadro 1: Comparação entre os métodos de aplicação de questionário	15
Quadro 2: Ferramentas para construção de questionários on-line	19
Quadro 3: Arquitetura do Flutter	24
Quadro 4: Desk research	36
Quadro 5: Rotas de convite no servidor do novo Sage	72
Quadro 6: Comparação do novo Sage	78

## RESUMO

Os questionários são frequentemente utilizados em pesquisas científicas mundialmente. Dados os problemas dos tradicionais questionários em papel, estes estão sujeitos à digitalização na forma de plataformas e aplicativos, por exemplo. Esta informatização traz benefícios em relação à sua alternativa analógica, mas também apresenta limitações, como indisponibilidade do sinal de internet, problemas de usabilidade e preocupações sobre a segurança dos dados dos participantes. Tais questões foram o foco do TCC denominado "Desenvolvimento de ferramenta para coleta de dados em pesquisas quantitativas utilizando dispositivos móveis", de Caio Pereira Oliveira, que resultou na criação do app Sage. Mesmo relevante, a aplicação desenvolvida por Caio tem espaço para melhorias e inclusão de novas funcionalidades. Este projeto visa, então, implementar estas alterações de forma a facilitar a adoção e utilização do app no contexto de coleta de dados em pesquisas científicas. Dentre elas, estão: novos tipos de perguntas, convite entre entrevistadores, e o desenvolvimento de uma aplicação web que seja intercambiável com a aplicação mobile. Como resultado do trabalho, tem-se o novo Sage com a implementação dos requisitos supracitados, verificados através de testes unitários automatizados.

**Palavras-chave:** Questionário, Aplicativo, Multiplataforma.

## ABSTRACT

Questionnaires are frequently used in scientific research worldwide. Given the problems of traditional paper questionnaires, they are subject to digitization in the form of platforms and applications, for example. This computerization brings benefits over its analog alternative, but also has limitations, such as unavailability of the internet signal, usability issues and concerns about the security of participants' data. Such questions were the focus of the TCC called "Desenvolvimento de ferramenta para coleta de dados em pesquisas quantitativas utilizando dispositivos móveis", by Caio Pereira Oliveira, which resulted in the creation of the Sage app. Even relevant, the application developed by Caio has room for improvement and the inclusion of new features. This project therefore aims to implement these changes in order to facilitate the adoption and use of the app in the context of data collection in scientific research. Among them are: new types of questions, invitation between interviewers, and the development of a web application that is interchangeable with the mobile application. As a result of the work, the new Sage received the implementation of the aforementioned requirements, verified through automated unit tests.

**Keywords:** Questionnaire, Application, Multiplatform.

## 1. INTRODUÇÃO

O método científico pode ser definido como o caminho a ser percorrido numa pesquisa para alcançar os resultados da melhor forma possível. Suas técnicas e instrumentos de coleta de dados devem ser adequados ao objeto em estudo e devem permitir alcançar os objetivos, responder às indagações da investigação científica e produzir conhecimento válido (LAKATOS, 2003).

A partir da definição dos objetivos de pesquisa a serem alcançados, deve-se propor o método que mais se adequa à pergunta de pesquisa, ou seja, que seja capaz de respondê-la. Portanto, podem-se ter pesquisas com abordagem essencialmente qualitativa, pesquisa de natureza quantitativa ou, ainda, pode ser feita a utilização de estratégias múltiplas, que abordem o qualitativo e o quantitativo simultaneamente, para aumentar o espectro de visão de interpretação da realidade, explorando diversas técnicas de coleta de dados (LAKATOS, 2003).

Na abordagem quantitativa, o estudo é de caráter experimental e matemático, tem predileção pelo fenômeno extenso e se caracteriza pela descrição empírica e pela medida e análise de variáveis. Tem como pressuposto o cultivo do rigor científico, da objetividade e da neutralidade no ato investigativo. Atua em níveis de realidade quantificáveis e deve ser utilizado para abarcar grandes aglomerados de dados ou de conjuntos demográficos, classificando-os e tornando-os inteligíveis por meio de variáveis (DEMOS, 1985).

Na pesquisa quantitativa, os instrumentos de coleta de dados são: a observação direta, os questionários, os bancos de dados (como o Sinan<sup>1</sup> – Sistema Nacional de Agravos de Notificação, o Sinasc<sup>2</sup> – Sistemas de Informação dos Nascidos Vivos), entre outros.

---

1 <https://portalsinan.saude.gov.br/>

2 <https://svs.aids.gov.br/daent/cgiae/sinasc/apresentacao/>

Neste contexto, os dados se apresentam como forma de medidas, mensurações ou estatísticas. Sua interação pode ser definida como objetiva e descritiva do fenômeno. Em um estudo quantitativo, a amostra deve ser ampla, estratificada, com apoio de grupos de controle, intencional ou aleatória, dependendo das variáveis necessárias para atingir os objetivos da pesquisa (ABDAL, 2016).

As técnicas e os procedimentos utilizados para a coleta de dados dependerão do tipo de método escolhido para a realização da pesquisa. No caso do presente trabalho desenvolvido, será analisado o tipo de coleta de dados que envolve uma entrevista fechada ou estruturada, que também é denominada de questionário. Na entrevista fechada, todas as questões têm respostas previstas para serem escolhidas (ABDAL, 2016).

Tais questionários são frequentemente usados em pesquisas científicas, porém há diversos problemas associados ao uso dos mesmos quando preenchidos em papel. Dentre eles, a aplicação do questionário em papel requer uma etapa complementar de digitação das respostas em uma planilha eletrônica, o que, além de exigir tempo, está sujeito a erros de digitação, o que pode impactar nos resultados do estudo. Por outro lado, a aplicação do questionário on-line faz o envio dos dados de forma imediata e automática, poupando, assim, boa parte do trabalho e do tempo destinados à digitação por parte do pesquisador. De forma semelhante, o surgimento dos questionários on-line permite que a mesma pesquisa possa ser feita de forma mais ágil e com redução de custos para o pesquisador, uma vez que o valor unitário gasto para a replicação do questionário é próximo de zero (ABDAL, 2016).

O funcionamento de um questionário on-line possui em termos gerais as mesmas características de um questionário de papel, e deverá ser enviado, recebido e preenchido, com o auxílio da internet (ABDAL, 2016).

Dessa forma, a informatização dos questionários se mostra uma alternativa, porém também possui limitações, como indisponibilidade do sinal de internet, problemas de usabilidade e preocupações sobre a segurança dos dados dos participantes (GUILHAM, 2008).

Tais questões foram o foco do Trabalho de Conclusão de Curso denominado "Desenvolvimento de ferramenta para coleta de dados em pesquisas quantitativas utilizando dispositivos móveis", de Caio Pereira Oliveira, que resultou na criação do aplicativo Sage. Mesmo relevante, o app desenvolvido por Caio tem espaço para melhorias e inclusão de novas funcionalidades. Este projeto visa, então, implementar estas alterações de forma a facilitar a adoção e utilização do app no contexto de coleta de dados em pesquisas científicas.

## **1.1. Objetivos**

O objetivo geral deste trabalho é evoluir o app Sage com novas funcionalidades e desenvolver uma aplicação web para o sistema, a fim de torná-lo multiplataforma.

Como objetivos específicos, tem-se:

- Extensão do backend e app mobile existentes com as novas funcionalidades:
  - implementação de novos tipos de perguntas;
  - implementação da funcionalidade de convites entre entrevistadores;
- Desenvolvimento de uma aplicação web com as mesmas funcionalidades do app mobile.

## 1.2. Metodologia de pesquisa

Este trabalho almeja atingir os objetivos propostos na seção 1.1 através das seguintes etapas:

- Pesquisa bibliográfica das linguagens/frameworks necessários através da própria documentação e artigos na internet;
- Desenvolvimento das funcionalidades no servidor, na aplicação, e na sua versão web;
- Desenvolvimento de testes unitários automatizados para o servidor.

Quanto às ferramentas necessárias para realizar o desenvolvimento, o aplicativo mobile Sage, ponto de partida deste trabalho, foi desenvolvido utilizando Flutter, enquanto o servidor utiliza Flask (framework Python) e integração com Firebase. Essas ferramentas foram mantidas, uma vez que suprem as necessidades deste trabalho. Para a aplicação web, também foi utilizado Flutter, já que ele permite a escrita de interfaces cross-platform e desta forma evita-se demasiado retrabalho.

## **2. REFERENCIAL TEÓRICO**

Nesta seção, são apresentados conceitos essenciais para o entendimento das seções seguintes. Primeiramente, a definição dos tipos de questionários existentes, a escolha do tipo a ser considerado no escopo deste trabalho, bem como um resumo das etapas para a construção de um questionário. Em seguida, são contextualizados termos técnicos que também se fazem necessários: Framework, Flutter, API, Flask, e Firebase. Por fim, encerra-se esta seção com uma apresentação do app Sage existente, anterior à sua evolução tecnológica neste trabalho.

### **2.1. Questionários**

A partir do surgimento da Web 2.0, modificou-se a forma como os internautas interagem na produção de conteúdos em sites e redes sociais, o que provocou um aumento de forma significativa na troca de informações entre os usuários. É neste contexto que os questionários de papel ganham um novo formato, onde a coleta de dados agora pode ser feita utilizando-se a internet por meio de formulários e questionários on-line.

Por mais que estudos apontem que a ideia de se conseguir informações dos usuários seja antiga, sua expansão e popularização no meio digital são recentes (VASCONCELLOS-GUEDES, 2007). Com o auxílio da computação em nuvem, que oportuniza o armazenamento virtual de dados e documentos em grande escala, a expansão dos formulários e questionários no formato on-line aumentou consideravelmente.

No quadro 1, pode-se observar um comparativo entre as formas de aplicação de um questionário, ressaltando seus pontos positivos e negativos.

Quadro 1: Comparação entre os métodos de aplicação de questionário

<b>Característica</b>	<b>Descrição e Observações</b>	<b>Entrevista pessoal</b>	<b>Entrevista por telefone</b>	<b>Questionário autopreenchido (correio)</b>
Versatilidade	Possibilidade de esclarecimento de dúvidas e utilização de material de apoio.	Alta	Média	Baixa
Custo	Horas de trabalho, deslocamento e nível da equipe.	Alto	Médio	Baixo
Tempo para aplicação	No caso da entrevista o tempo pode ser reduzido aumentando o número de entrevistadores. A devolução do questionário ocorre, em maior volume, de 2 a 3 semanas após a postagem. Para reduzir o prazo pode ser feito o envio de postagem estimuladora.	Alto	Baixo	Médio
Controle amostral	Controle em relação a quem respondeu e se este constitui uma amostra representativa da população.	Alto	Médio	Baixo
Quantidade de dados	Quantidade de dados possível de se obter.	Alta	Média	Média
Qualidade dos dados	Grau em que os dados estão livres de vieses em função do método de coleta.	A qualidade dos dados depende da fonte de viés considerada.		
Garantia de anonimato	O anonimato ajuda a coleta de dados sinceros, mas dificulta o controle amostral.	Baixa	Baixa	Média
Habilidade exigida para aplicação	Perfil de qualificações dos entrevistadores	Alta	Alta	Baixa
Uniformidade da mensuração	Padronização de frases, ordem das perguntas, opções de respostas.	Baixa	Média	Alta

Índice de resposta	Pressão para obtenção de respostas.	Alto	Alto	Médio
Nível educacional exigido dos respondentes	O questionário auto preenchido é mais adequado para públicos com educação formal elevada, restringindo sua possibilidade de aplicação.	Baixo	Baixo	Alto
Possibilidade de verificação da sinceridade das respostas	Possibilidade de o entrevistador observar o comportamento do entrevistado.	Alta	Baixa	Baixa
Tamanho da amostra	_____	Pequena	Grande	Grande

Fonte: Vasconcellos-Guedes e Guedes 2007, p.4-5

Em relação aos questionários on-line, pode-se ressaltar três tipos principais, que utilizam a internet em pelo menos uma de suas etapas de aplicação. Abaixo, são exemplificados os principais questionários on-line (ABDAL, 2016).

- **Máscaras e arquivos de bases de dados:** são instrumentos eletrônicos construídos, aplicados e respondidos de forma off-line, onde o respondente recebe o questionário em um editor de texto/planilha, como Word, Excel, Power Point e, após o seu preenchimento, envia os dados por e-mail ou outro serviço de transferência eletrônica de arquivos. Usando a internet, para receber o questionário e para o envio dos dados. Um exemplo seria os questionários do IBGE.

Figura 1: Exemplo de interface de máscara de questionário eletrônico  
 Fonte: Abdal 2016, p.56

- **Aplicativos para dispositivos móveis:** são programas para celulares, smartphones e tablets cujos questionários também podem ser respondidos de forma off-line usando a rede somente para encaminhar os dados. Um diferencial é que esse tipo de questionário eletrônico apresenta uma interface que facilita o envio automático desses dados (geralmente por meio de um botão), sem a necessidade de transferir os arquivos por e-mail ou por qualquer outra ferramenta externa de transferência de dados. Utilizado, por exemplo, pela Receita Federal brasileira para o preenchimento e o envio das informações relativas à Declaração Anual de Imposto de Renda.

The image shows two side-by-side screenshots of the 'RASCUNHO IRPF' mobile application. The left screenshot displays a main menu with five large, colorful buttons: 'Identificação' (blue), 'Info. Terceiros' (yellow), 'Rendimentos' (green), 'Pagamentos' (red), and a home icon (blue). The right screenshot shows a form titled 'Adicionar Rendimento' with a 'Pessoa Juridica' header. It contains several input fields: 'CNPJ/CPF da fonte pagadora', 'Nome da fonte pagadora', 'Rendimento recebido', and 'Previdência Oficial'. At the top of both screens, it says 'Exercício: 2016 CPF: Rascunho: Não Salvo'.

Figura 2: Exemplo de interface de aplicativo de questionário eletrônico  
Fonte: Abdal 2016, p.56

The image shows two columns of a web-based questionnaire. The left column contains several dropdown menus for the following questions: 'Ocupação \* Qual é sua ocupação atual?', 'Trabalho/Emprego \* Se você trabalha, em que setor está empregado?', 'Trabalho/Emprego \* Se você trabalha, em que setor está empregado?', 'Município de Moradia \*', 'Estado de Moradia \*', 'Instituição de Ensino \* Qual o nome da instituição em que você estudou ou estuda?', 'Ano de formação \* Em que ano você terminou o seu curso?', 'Ano de ingresso \* Em que ano você ingressou no seu curso?', 'Modalidade do Curso \* Seu curso é / foi:', and 'Estudos anteriores \* Onde você fez seus estudos de ensino fundamental e do ensino médio?'. The right column contains two sections of radio button options. The first section is 'Manutenção dos estudos \* Como você se mantém (ou se mantém) no decorrer do seu curso?' with options: 'Somente com recursos da família', 'Trabalhando, mas contando, para o essencial, com os recursos da família', 'Trabalhando para participar do rateio das despesas da família', 'Por conta própria, com recursos oriundos de trabalho remunerado', 'Com bolsa de estudos ou crédito educativo', 'Com bolsa ou crédito educativo, trabalhando e contando ainda com o apoio da família', and 'Outros'. The second section is 'Trabalho/Emprego \* Caso tenha exercido alguma atividade remunerada durante o seu curso, de que tipo ela foi?' with options: 'Em meio-período (até 20 horas semanais)', 'Em tempo semi-integral (de 21 a 32 horas semanais)', 'Em tempo integral (mais de 32 horas semanais)', 'Eventualmente', and 'Não exerci atividade remunerada'. Below these is another section 'Atividades extra-curriculares (mais de uma opção) \* Você realizou algum tipo de atividade extra-curricular? Se sim, qual foi? (Você pode selecionar mais de uma opção)' with options: 'Iniciação Científica (com bolsa)', 'Iniciação Científica (sem bolsa)', 'Estágio remunerado', 'Estágio não-remunerado', 'Bolsa-trabalho ou afins', 'Outros tipos de atividades existentes em meu curso/instituição', and 'Não participei atividades extra-curriculares'.

Figura 3: Exemplo de interface de plataforma de questionário on-line  
Fonte: Abdal 2016, p.57

- **Web surveys:** atualmente, há na internet mais de 80 plataformas que oferecem o serviço de construção de questionários on-line (ABDAL, 2016). Sendo que em sua maioria, a disponibilidade de tais serviços é gratuita, mas, difere de algumas opções premium, oferecendo uma quantidade limitada de ferramentas para elaboração do questionário e sua coleta dos dados. No quadro 2, apresenta-se alguns dos principais serviços disponíveis no país, que oferecem as ferramentas mais completas para a construção e aplicação de questionários do tipo on-line.

Quadro 2: Ferramentas para construção de questionários on-line

Google Formulários	Trata-se da plataforma mais popular para a construção de formulários e questionários on-line. Embora tenha um conjunto limitado de ferramentas para a elaboração dos formulários e poucas opções de formatação, o serviço é totalmente gratuito e não tem limite de questionários aplicados.
Lime Survey	Com ferramentas muito mais sofisticadas, esta é a plataforma de código aberto mais completa disponível hoje aos pesquisadores. No entanto, ela exige conhecimentos básicos em programação e design de páginas da internet.
Survey Monkey	Bastante amigável e intuitiva, possui um leque completo de ferramentas, das mais simples às mais complexas. O pesquisador tem à disposição dezenas de modelos de questões, e basta selecionar as perguntas desejadas para montar rapidamente o questionário.
Qualtrics	Juntamente com a anterior, é a plataforma mais completa e intuitiva à disposição hoje para aqueles que não dominam uma linguagem de programação e desejam montar um questionário mais complexo. Além de dezenas de ferramentas de coleta, também oferece técnicas de metrificação sofisticadas e automáticas que são bastante úteis.
Sphinx on-line	Versão on-line de um dos mais populares programas de digitação e montagem de bancos de dados.
Survio	Versão muito semelhante à Survey Monkey e à Qualtrics, com a vantagem de oferecer gratuitamente a elaboração de um número ilimitado de questões.

Fonte: Abdal 2016, p.57-58 adaptado

Para os fins de definição de “questionário on-line” no escopo deste trabalho, considerou-se questionários digitais na forma de aplicativo para dispositivos móveis

e na forma de web surveys, com foco em aplicação presencial pelo entrevistador (como ocorre, por exemplo, com o Censo demográfico do Brasil).

Em se tratando das etapas que envolvem a aplicação e uso dos questionários, observa-se que é necessário o desenvolvimento do planejamento, elaboração e aplicação como em qualquer instrumento quantitativo de coleta de dados. Na etapa do planejamento de uma pesquisa quantitativa, se verificam importantes pontos a considerar e decisões a tomar que são anteriores à própria construção de um questionário, que independe do seu tipo (on-line ou papel). Ao menos cinco passos teórico-metodológicos são essenciais até que o pesquisador possa decidir pela viabilidade do uso de um questionário on-line, e os mesmos estão destacados na Figura 4.

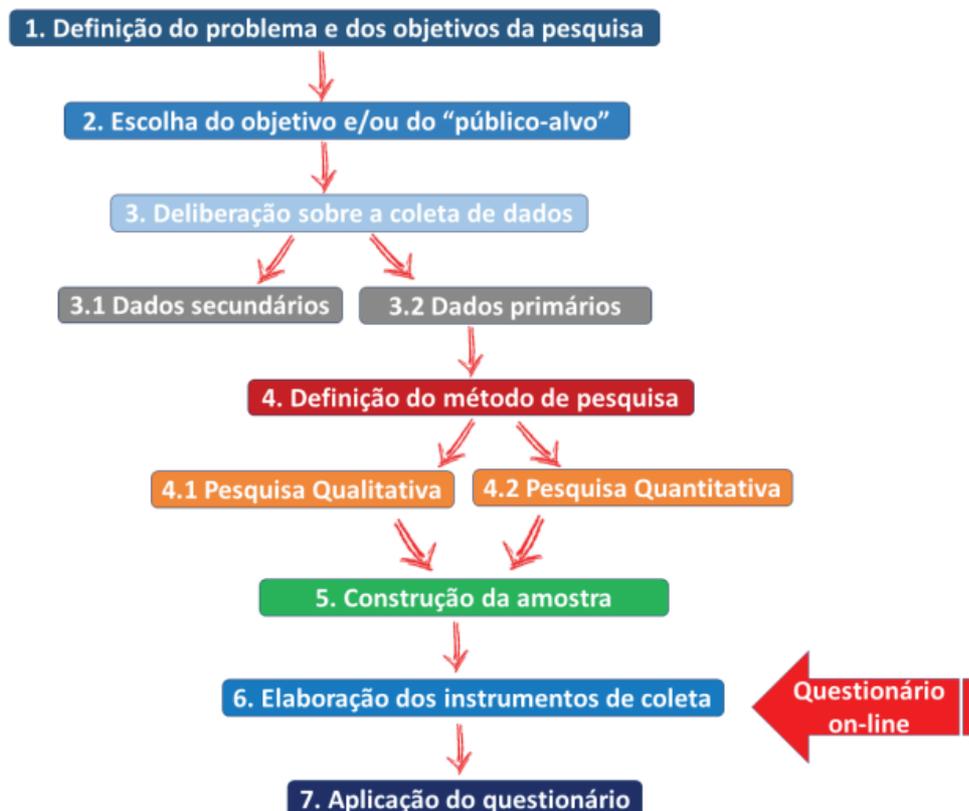


Figura 4: Etapas da construção de uma pesquisa com questionário on-line

Fonte: Abdal 2016, p.60

Será por meio da fase de aplicação do questionário que a opção pelo instrumento on-line passa a ter implicações mais diretas sobre as etapas que se sucederão. Sendo necessária a aplicação de um “piloto” do questionário com um número reduzido de respondentes a fim de verificar possíveis erros de construção, fluxo e dificuldades de compreensão do questionário.

Após a finalização do período de coleta dos dados, se observa a construção de um banco de dados completo formado por todas as respostas registradas. É comum que cada opção de resposta do questionário tenha um código numérico no banco de dados, facilitando o processamento estatístico. Atualmente, se observa a opção de baixar os dados em um arquivo compatível com programas de análise estatística, como o Excel<sup>3</sup>, SPSS<sup>4</sup>, Matlab<sup>5</sup>, SAS<sup>6</sup>, R<sup>7</sup>, Stata<sup>8</sup>, entre outros.

Com o banco de dados, inicia-se o processo de checagem, crítica e consistência onde as informações são revisadas em busca de possíveis incoerências ou erros de preenchimento. Caso se observe que alguma questão permite múltiplas respostas ou que tenha a possibilidade de citações não definidas previamente no questionário (as chamadas questões abertas), será atribuído um código para cada resposta diferente que aparecer. Essa é a etapa chamada de codificação (ABDAL, 2016).

Por fim, para efeitos de comparação, no caso do questionário de papel, além dessas etapas seria necessária uma etapa de processamento, para inserir as respostas no banco de dados. Portanto, o questionário on-line apresenta um diferencial neste quesito, uma vez que tudo já é registrado automaticamente.

---

3 <https://www.microsoft.com/pt-br/microsoft-365/excel>

4 <https://www.ibm.com/br-pt/spss>

5 <https://www.mathworks.com/products/matlab.html>

6 [https://www.sas.com/pt\\_br/home.html](https://www.sas.com/pt_br/home.html)

7 <https://www.r-project.org/>

8 <https://www.stata.com/>

## 2.2. Framework

Um framework pode ser compreendido como um conjunto de códigos que são usados como pacotes para que uma pessoa desenvolvedora possa criar uma aplicação. Logo, os frameworks podem ser vistos como uma forma de auxiliar na construção de um sistema, uma vez que permitem o reuso de código pronto ao invés de ser necessário desenvolvê-lo do zero (BARRO, 2022).

Logo, a principal função de um framework, que leva a sua utilidade, é a facilidade no que diz respeito ao processo de desenvolvimento de um software ou aplicação. Além disso, muitos dos projetos de frameworks são abertos e tendem a passar por testes e processos de otimização.

Isto é, caso alguma falha ou bug seja encontrado, eles podem ser rapidamente reportados e resolvidos pela comunidade. Desta forma, se verifica mais segurança e consistência nos seus padrões.

Assim, o funcionamento de um framework se dá por meio da reutilização dos códigos genéricos oferecidos pelo mesmo, que podem ser aplicados como base para diferentes projetos (BARRO, 2022). Como vantagens proporcionadas pelo amplo emprego dos frameworks nos softwares destacam-se: a otimização do tempo de desenvolvimento, definição e padronização das melhores práticas de programação, maior segurança, prevenção de códigos duplicados, diminuição da ocorrência de bugs, maior consistência no processo de desenvolvimento e nas aplicações criadas, redução das chances de erro no código. Por fim, frameworks possibilitam que os desenvolvedores se dediquem aos elementos específicos do projeto e simplificam/encurtam a curva de aprendizado do time de desenvolvimento (BARRO, 2022).

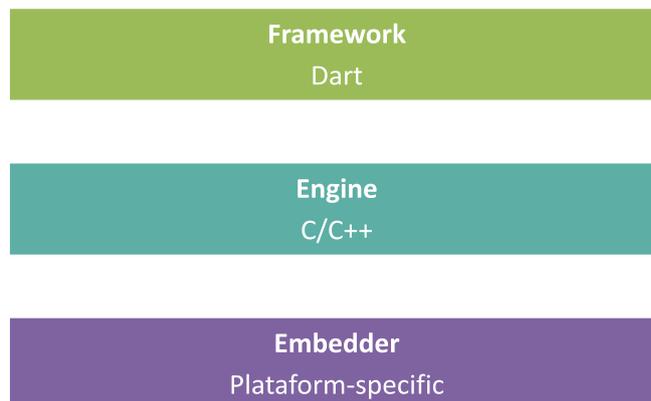
## 2.3. Flutter

Flutter<sup>9</sup> é um framework voltado para o desenvolvimento de interfaces de usuário para aplicativos multiplataforma, como: Android, iOS, Windows, Mac, Linux, Fuchsia e Web. Nele, é utilizada a linguagem de programação Dart que, assim como o framework, foi criada pela Google. O paradigma utilizado em Flutter é o de programação reativa para interfaces, difundido pela biblioteca React que, por sua vez, foi criada pelo Facebook.

De acordo com a documentação oficial, as aplicações desenvolvidas em Flutter, durante a fase de desenvolvimento, são executadas em uma Virtual Machine (VM) que possibilita uma recompilação rápida apenas dos componentes que sofreram mudanças, não necessitando reconstruir toda a aplicação. Entretanto, na etapa de *releasing* (lançamento da aplicação) todo o código é compilado diretamente para código de máquina (Intel x64, ARM ou JavaScript).

O Flutter é construído com C, C++, Dart e Skia (uma motor gráfico de renderização 2D) e, como mostrado no quadro 3, possui uma arquitetura de camadas.

Quadro 3: Arquitetura do Flutter



Fonte: Próprio autor, adaptado de Bueno 2021

---

<sup>9</sup> <https://flutter.dev/>

Na última camada do diagrama está o Embedder. Cada plataforma possui um Embedder específico que pode ser escrito em Java e C++ para Android, Objective-C/Objective-C++ para iOS/macOS, e C++ para Windows/Linux. O Embedder acaba por ser um provedor de ponto de entrada para a aplicação sendo também responsável por coordenar acesso a serviços do sistema operacional, como: renderização, acessibilidade, entrada de dados e também gerencia o loop de eventos.

Ainda tomando como referência o quadro 3, tem-se a Engine ou também chamada de Flutter Engine, é escrita em C++ e tem as primitivas necessárias para suportar todas as aplicações Flutter. Tal mecanismo é responsável por rasterizar os componentes da tela sempre que for preciso atualizar a interface.



Figura 5: Flutter Framework  
Fonte: Bueno 2021, p.10

No topo do diagrama está o Framework Flutter. Este é o componente utilizado pelos desenvolvedores de software aplicativo, que não atuam no desenvolvimento do framework em si. A Figura 7 ilustra os principais componentes do framework, que apresentam as seguintes funcionalidades:

- **Foundation:** provê abstrações para serem utilizadas por serviços como Animation (animação), Painting (pintura) e Gestures (gestos).

- **Rendering:** prove abstração para trabalhar com layouts. Nesta camada é possível criar uma árvore de componentes renderizados.
- **Widgets:** esta camada possibilita criar abstração de composição. Basicamente as classes utilizadas nesta camada serão tratadas como *render objects* na camada de renderização.
- **Material e Cupertino:** esta camada provê um conjunto de widgets pré-definidos de acordo com ambos Material e iOS design systems, e que podem ser utilizados na criação de interfaces.

## 2.4. API

API é um acrônimo de *Application Programming Interface* e corresponde a um conjunto de softwares que tem como função intermediar a comunicação entre as aplicações. Desta forma, uma API acaba por oportunizar, também, que uma ou mais aplicações possam se comunicar de forma transparente com o usuário, como se fosse somente uma. Assim, sua agregação acaba por integrar em nível de dados onde estes são enviados e/ou recebidos entre as aplicações envolvidas no processo (JASSE, 2017).

Logo, uma *Application Programming Interface* (API) é um conjunto de rotinas utilizado para descrever a comunicação entre diferentes componentes de software. São responsáveis pelo controle de acesso a funções de software ou dispositivos de hardware. Existem APIs de sistemas operacionais, de bibliotecas ou pacotes de software, para a Web, dentre outras.

Para uma empresa, uma API é um ponto de conexão entre ela e seus parceiros ou clientes, compartilhando processos de negócio, serviços ou ainda

conteúdos, onde as equipes de trabalhos internos podem estar conectadas de maneira fácil e segura (JASSE, 2017).

## 2.5. Flask

O Python<sup>10</sup>, por ser uma linguagem de programação interpretada, pode rodar em qualquer plataforma, seja ela Linux, Windows, ou Mac OS, por exemplo. Isso torna o desenvolvimento de aplicações multiplataformas mais acessível. Sua sintaxe em si é uma de suas melhores características e deixa o código facilmente legível a nível humano. Permite desenvolver de modo imperativo, orientado a objetos ou funcional e possui tipagem dinâmica e forte. Além do mais, possui uma vasta quantidade de módulos/bibliotecas que já são *built-in*, e se bibliotecas de terceiros forem necessárias basta utilizar o *pip*, um módulo *built-in* do Python, para instalá-las. Outro exemplo de uma biblioteca interna é a *multiprocessing*, que é utilizada para se atingir paralelismo nos programas.

Flask<sup>11</sup> é um framework Python para a criação de servidores web. É um dos pacotes mais utilizados do ecossistema Python. Destaca-se por implementar a especificação WSGI (Web Server Gateway Interface), padrão utilizado por frameworks web a fim de facilitar a portabilidade das aplicações em diferentes servidores web.

Segundo Ronacher (2018) Flask é um micro framework para Python baseado no Werkzeug, Jinja2 e boas práticas. É usado para construir e executar um web server, que pode ser gerenciado por URLs customizadas, onde cada uma aponta para uma anotação no script em Python que executará sua devida função, sendo

---

10 <https://www.python.org/>

11 <https://flask.palletsprojects.com/>

que cada função possui as definições de uma página ou ações padronizadas, assim como o tratamento de requisições GET e POST.

## 2.6. Firebase

A Firebase<sup>12</sup> é uma plataforma pertencente ao Google para desenvolvimento de aplicações web e móveis, que implementa diversas funcionalidades comuns a estas aplicações. Dentre elas, podemos citar os serviços a seguir oferecidos pelo Firebase: *Auth*, para criação de usuários e autenticação, com suporte a *social login*; *Crashlytics*, para receber informações sobre encerramentos inesperados no app por conta de erro; *Analytics*, para coletar dados de uso do aplicativo; e *Cloud Messaging*, para enviar push notifications para os dispositivos.

O Firebase, desenvolvido pela Google, é um conjunto de produtos distribuído gratuitamente, com um limite de utilização (FIREBASE, 2017). Entre esses produtos, existem serviços de hospedagem, armazenamento em nuvem e banco de dados. No caso do banco, a utilização gratuita permite até 100 acessos simultâneos. Essa ferramenta permite, com poucas linhas de código, adicionar o banco de dados em aplicações web, Android e iOS para que se conectem ao mesmo banco, sem, contudo, requerer conhecimentos sobre a infraestrutura do sistema. (UEDA 2020, p.16).

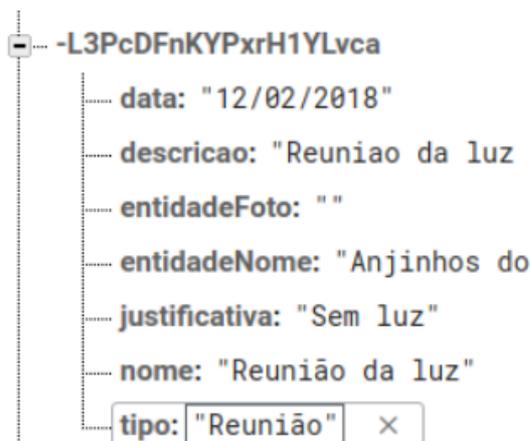


Figura 6: Estrutura de base de dados Firebase  
Fonte: Silva 2018, p.35

12 <https://firebase.google.com/?hl=pt>

Por meio dos estudos desenvolvidos por Silva (2018), pode-se mencionar que o Firebase tem uma estrutura de base de dados que é diferente de muitos bancos de dados comuns. Cada base de dados Firebase é guardada como uma árvore de objeto JSON. Esta estrutura de árvore é muito flexível para todos os tipos de dados, como pode ser visto na Figura 6.

Assim a plataforma acaba por oferecer serviços e produtos que têm como foco desenvolver e testar aplicativos, objetivando o desenvolvimento de aplicativos com alta qualidade e ótimo tempo de resposta. (SILVA, 2018).

## **2.8. Sage**

O aplicativo utilizado como base para desenvolvimento deste trabalho, o *Sage*, se propõe a servir como ferramenta para coleta de dados em pesquisas quantitativas. Ele foi projetado para que o aplicativo em execução num dispositivo móvel (tablet ou smartphone) seja entregue ao participante para que preencha o questionário de forma autônoma, mas o pesquisador também pode optar por ler as perguntas e campos ao participante, registrando suas escolhas por ele.

Ao iniciar o app, é apresentada a tela de login para autenticação com o Google Sign in, provido pelo Firebase, conforme a Figura 7.

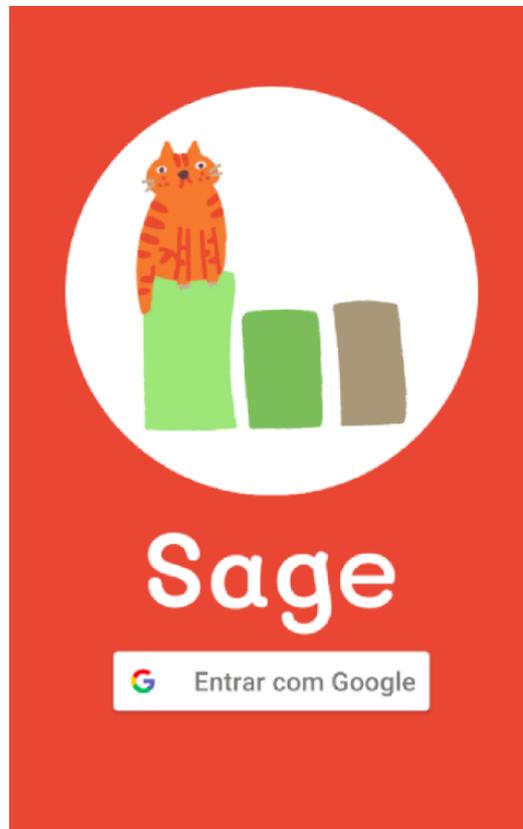


Figura 7: Tela de Login do Sage

Após autenticado, o usuário pode utilizar a funcionalidade de cadastro de questionários, conforme a Figura 8, para criar novos questionários e adicionar perguntas aos mesmos, de acordo com os tipos de perguntas desejados. Os tipos de perguntas disponíveis são: escolha múltipla, escolha única, texto livre e numérica.

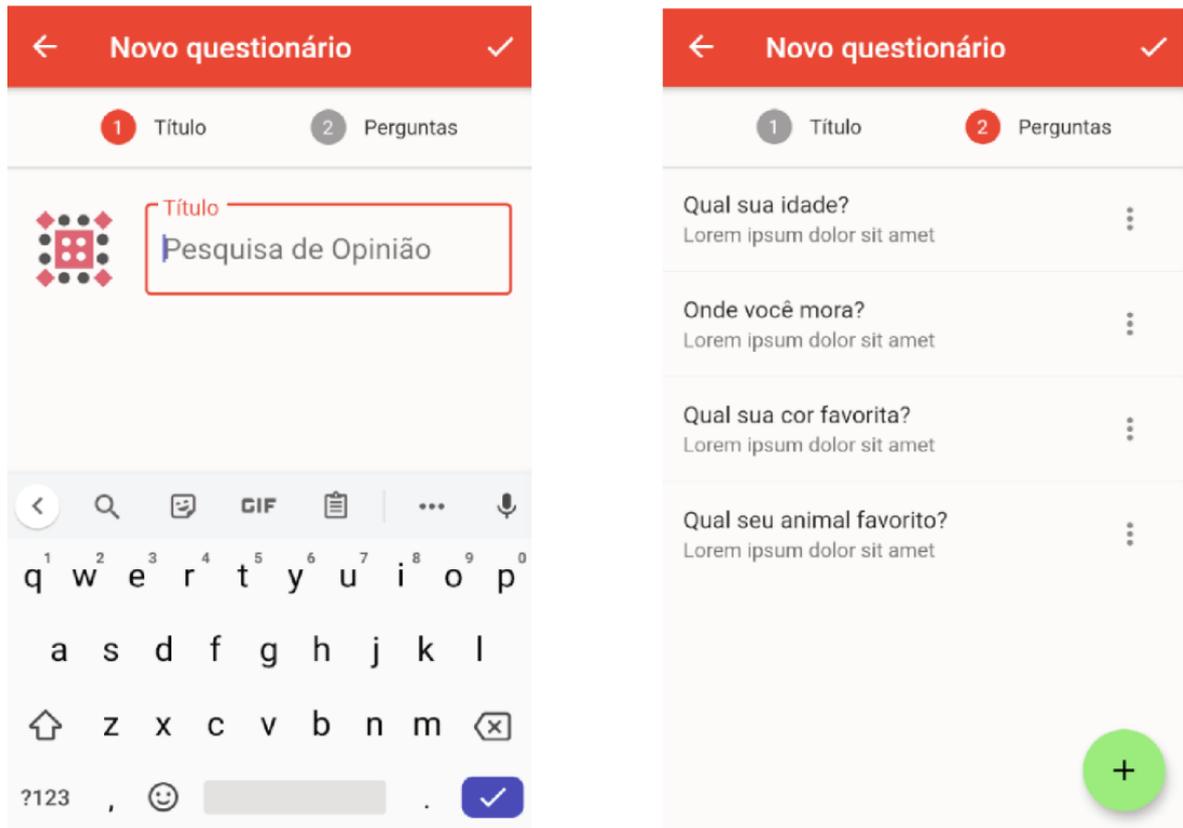


Figura 8: Tela de Cadastro de questionários do Sage

Tendo criado um ou mais questionários, é possível respondê-los, conforme a Figura 9, avançando pelas perguntas e respeitando as devidas regras de validação de cada pergunta. Ao salvar, as respostas são criptografadas e armazenadas no dispositivo.

← Pesquisa de Opinião

Qual sua cor preferida?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce venenatis ut mauris a dignissim.

Amarelo

Azul

Preto

Verde

Vermelho

AVANÇAR

Figura 9: Tela de preenchimento de questionários do Sage

Na tela inicial, de listagem de questionários, é possível visualizar os questionários criados e sincronizar com o servidor as respostas armazenadas, conforme a Figura 10.

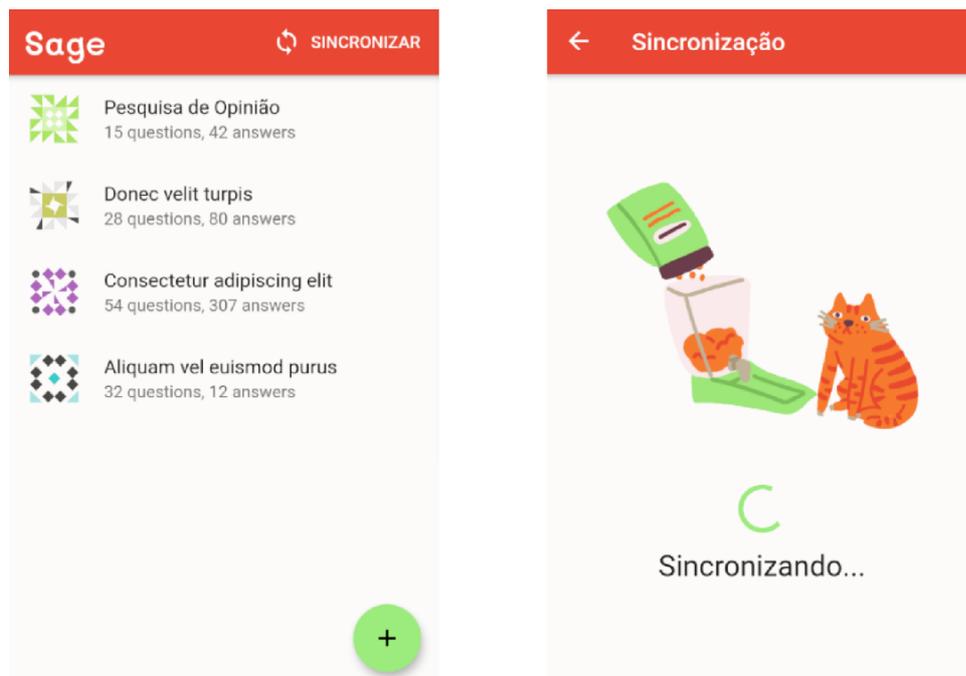


Figura 10: Telas de listagem de questionários e de sincronização do Sage

### 3. TRABALHOS CORRELATOS

Nesta seção, serão apresentados os requisitos que foram definidos para a evolução tecnológica do Sage, seguidos por uma “desk research” em busca de ferramentas correlatas.

#### 3.1. Análise dos requisitos

Baseado nas recomendações de trabalhos futuros apresentados ao final do TCC de (OLIVEIRA, 2019), e complementando-se com sugestões próprias, obteve-se os requisitos a seguir para evolução técnica da ferramenta.

##### 1. Versão web da ferramenta

**Requisito 1.1:** O *web app* deverá permitir acesso às mesmas funcionalidades já presentes no aplicativo móvel.

**Requisito 1.2:** O *web app* deverá permitir o uso das novas funcionalidades (requisitos a seguir) que serão implementadas no aplicativo móvel.

##### 2. Adição de novos tipos de perguntas

**Requisito 2.1:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “escala de Likert”.

**Requisito 2.2:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “data + horário”.

**Requisito 2.3:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “data”.

**Requisito 2.4:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “horário”.

### 3. Adição de convites entre entrevistadores

**Requisito 3.1:** O aplicativo e o web app deverão permitir que um usuário convide outros usuários para os seus questionários, usando o e-mail cadastrado na conta do entrevistador a ser convidado.

**Requisito 3.2:** O aplicativo e o web app deverão permitir que o usuário liste os seus convites recebidos.

**Requisito 3.3:** O aplicativo e o web app deverão permitir que o usuário liste os convites que foram enviados em um determinado questionário de sua autoria.

### 3.2. Desk Research

Com o objetivo de identificar a existência de uma ferramenta de questionários semelhante à nova versão do Sage que é sugerida neste trabalho, foi realizado um levantamento das principais ferramentas de questionário existentes atualmente. Neste trabalho, este processo será referido como *Desk research*, isto é, uma pesquisa secundária baseada em informações, documentos e fontes já existentes (AELA CONTENTS, 2020). Para isto, não seria viável utilizar um motor de busca determinístico como o Scopus<sup>13</sup>, já que este busca apenas por artigos e trabalhos científicos, portanto, as ferramentas foram encontradas através do motor de busca Google, utilizando-se as strings de busca "survey software", "questionnaire software" e "form software".

Foram consideradas as ferramentas presentes nas 3 primeiras páginas de cada consulta, com exceção de resultados provenientes de anúncios. Apesar do Google personalizar os resultados de acordo com o usuário, este problema foi

---

<sup>13</sup> <https://www.scopus.com/>

mitigado realizando-se todas as buscas a partir de uma guia anônima e utilizando um IP fornecido pela universidade. Após realizar a busca, 34 ferramentas foram listadas: *Alchemer, BangTheTable, Checkbox, CognitoForms, Creative Research Systems, Forms.io, FormStack, FreeOnlineSurveys, G3plus, Google Forms, IdSurvey, Jotform, KeySurvey, LimeSurvey, MachForm, Opinio, Qualaroo, Qualtrics, QuestionPro, SmartSurvey, SnapSurveys, SoftExpert, Submittable, SurveyLab, SurveyMethods, SurveyMonkey, SurveyPlanet, Survio, Typeform, VisualParadigm, Voxco, WebSurveyCreator, Wufoo, e Zoho.*

Após listar as ferramentas, todas que não possuíam planos gratuitos foram excluídas da comparação, visto que seria inviável pagar por todas para testar as funcionalidades. As ferramentas removidas por este motivo foram: *Alchemer, BangTheTable, Checkbox, Creative Research Systems, FormStack, G3plus, IdSurvey, KeySurvey, MachForm, Opinio, Qualaroo, Qualtrics, SnapSurveys, SoftExpert, Submittable, SurveyLab, VisualParadigm, e Voxco.*

Por fim, também foram removidas ferramentas que não permitiam a exportação de dados (em um formato como .CSV ou .XLSX, por exemplo) no plano gratuito. Foram elas: *SmartSurvey, SurveyPlanet, Survio, e Zoho.*

A análise das ferramentas restantes foi feita com base tanto nos requisitos provenientes do TCC de (OLIVEIRA, 2019) (precificação, uso offline, e ciframento no armazenamento das respostas) quanto nos requisitos determinados na proposta deste TCC (aplicação web, tipos de perguntas, e convites entre entrevistadores).

Quadro 4: Desk research

Nome	Site	Precificação	Layout Adaptado	Offline	Ciframento no armazenamento das respostas	Aplicação web	Tipos de perguntas	Convites entre entrevistadores
CognitoForms	<a href="https://www.cognitofrms.com/">https://www.cognitofrms.com/</a>	Gratuito (500 respostas/mês; 100 MB de armazenamento)	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala/matriz Likert, data, intervalo de data, envio de arquivo	Apenas no plano pago
Forms.io	<a href="https://forms.io/">https://forms.io/</a>	Gratuito (500 respostas)	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (somente estrelas), data, horário, envio de arquivo	Sim, até 10 usuários no plano gratuito
FreeOnlineSurveys	<a href="https://freeonlinesurveys.com/">https://freeonlinesurveys.com/</a>	Gratuito (limite de 1000 respostas, duração máxima de 2 semanas para cada survey)	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala/matriz Likert, data, intervalo de tempo, ranking, escolha de imagem	Apenas nos planos Business
Google Forms	<a href="https://www.google.com/forms/about/">https://www.google.com/forms/about/</a>	Gratuito	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), data, horário, upload de arquivo	Sim
Jotform	<a href="https://www.jotform.com/">https://www.jotform.com/</a>	Gratuito (5 pesquisas; 100 respostas/mês; 100 MB de armazenamento; 1000 views/mês)	Sim	Sim	Sim	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), horário, upload de arquivo	Apenas nos planos para Empresas
LimeSurvey	<a href="https://www.limesurvey.org/pt/">https://www.limesurvey.org/pt/</a>	Gratuito (limite de 25 respostas/mês, 10 MB de armazenamento)	Sim	Apenas através de app de terceiros, cujo plano gratuito inclui limitações	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), data (selects)	Sim

QuestionPro	<a href="https://www.questionpro.com/pt-br/">https://www.questionpro.com/pt-br/</a>	Gratuito (1000 respostas/pesquisa; 100 perguntas/pesquisa; 5000 convites de e-mail/mês)	Sim	Apenas nos planos pagos	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala/matriz Likert (diversos), max diff, escolha de imagem, data/horário	Apenas em planos pagos
SurveyMethods	<a href="https://surveymethods.com/">https://surveymethods.com/</a>	Gratuito (30 pesquisas/conta; 20 perguntas/pesquisa; 500 respostas/pesquisa/mês)	Não	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala/matriz Likert (diversos), upload de arquivo	Sim
SurveyMonkey	<a href="https://pt.surveymonkey.com/">https://pt.surveymonkey.com/</a>	Gratuito (10 perguntas/pesquisa; 10 respostas/pesquisa)	Sim	Apenas em planos pagos	Apenas em planos pagos	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), data, horário, upload de arquivo	Apenas em planos pagos
Typeform	<a href="https://www.typeform.com/">https://www.typeform.com/</a>	Gratuito (10 perguntas/pesquisa; 10 respostas/mês)	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala/matriz Likert (diversos), escolha de imagem, data/horário	Apenas em planos pagos
WebSurveyCreator	<a href="https://www.websurveycreator.com/c/best-online-survey-software.aspx">https://www.websurveycreator.com/c/best-online-survey-software.aspx</a>	Gratuito (100 respondentes/pesquisas; 100 respostas/pesquisa)	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala/matriz Likert (diversos), data/horário, mais outros nos planos pagos	Apenas em planos pagos
Wufoo	<a href="https://www.wufoo.com/">https://www.wufoo.com/</a>	Gratuito (5 pesquisas; 100 respostas/mês; 10 campos)	Sim	Requer ferramenta externa paga	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), data/horário, mais outros nos planos pagos	Apenas em alguns planos pagos
Sage		Gratuito	Sim	Sim	Sim	Não	Texto, numérico, múltipla escolha (diversos)	Não

A partir do desk research, é possível observar que as ferramentas existentes que mais se aproximam da ferramenta proposta, isto é, que satisfazem mais requisitos, são: Google Forms e Jotform.

No entanto, o Google Forms não dispõe de uso offline e, por consequência, também não realiza ciframento no armazenamento das respostas, já que elas não são armazenadas no dispositivo.

Por outro lado, o Jotform restringe o uso da funcionalidade de convite entre entrevistadores a usuários dos planos empresariais, ao mesmo tempo em que possui limite do plano gratuito (5 pesquisas; 100 respostas/mês; 100 MB de armazenamento; 1000 views/mês).

Dessa forma, partindo do Sage existente, é possível evoluí-lo tecnicamente para que se satisfaça todos os requisitos listados.

## 4. DESENVOLVIMENTO

Nesta seção, é apresentada a sequência de passos executados no desenvolvimento deste trabalho a fim de evoluir o Sage de seu estado inicial até o estado proposto na análise de requisitos já exposta. Cada subseção apresenta uma funcionalidade desenvolvida ou agrupa um conjunto de tarefas correlacionadas. Primeiramente, é descrita a atualização dos pacotes e ferramentas da aplicação, seguida pela correção de erros diversos. Depois, o desenvolvimento da aplicação web, novos tipos de perguntas, e funcionalidade de convites entre entrevistadores, respectivamente.

### 4.1. Atualização de pacotes e ferramentas

Para executar o servidor existente do Sage localmente, bastou instalar as dependências necessárias e alterar alguns detalhes relacionados à importação dos módulos. Já no caso do app mobile, foi necessário realizar uma série de tarefas antes de ser possível iniciar o desenvolvimento.

Primeiramente, os pacotes utilizados se encontravam em versões muito antigas, o que foi proibitivo por conta da adoção de “Android v2 embedding” no Flutter. Mais informações disponíveis no artigo “Android v1 embedding app and plugin creation deprecation”<sup>14</sup> da documentação oficial do Flutter.

“Existing v1 applications consuming a plugin that targets only the v2 embedding won’t build and must migrate. This has been the case since v1.12.”

Portanto, foi necessário atualizar as versões dos pacotes, como mostrado na Figura 11, tomando cuidado para utilizar versões compatíveis entre si (já que muitas

---

<sup>14</sup> <https://docs.flutter.dev/release/breaking-changes/android-v1-embedding-create-deprecation>

vezes um pacote depende de uma versão específica de outro pacote e desrespeitar isso pode causar problemas).

```

... @@ -1,7 +1,7 @@
1 name: sage
2 description: A new Flutter application.
3 version: 0.1.0+00100
4 - publish_to: !none
5
6 environment:
7   sdk: ">=2.8.0 <3.0.0"
... @@ -11,48 +11,49 @@ dependencies:
11 flutter:
12   sdk: flutter
13
14 - animations: ^1.0.0+5
15 - auto_route: ^0.5.0
16 - auto_size_text: ^2.1.0
17 - cached_network_image: ^2.2.0+1
18 - encrypt: ^4.0.1
19 - dio: ^3.0.9
20 - firebase_auth: ^0.16.0
21 - firebase_core: ^0.4.4+3
22 - flutter_auth_buttons: ^0.8.0
23 - flutter_reorderable_list: ^0.1.3
24 - flutter_svg: ^0.17.4
25 - get_it: ^4.0.2
26 - google_fonts: ^1.1.0
27 - google_sign_in: ^4.5.1
28 - injectable: ^0.4.0
29 - intl: ^0.16.1
30 - jdenticon_dart: ^1.2.1+1
31 - json_annotation: ^3.0.1
32 - list_diff: ^1.3.0
33 - moor: ^3.1.0
34 - moor_ffi: ^0.6.0
35 path: ^1.6.4
36 - path_provider: ^1.6.9
37 - package_info: ^0.4.0+18
38 pretty_dio_logger: ^1.1.1
39 - provider: ^4.1.2
40 - shared_preferences: ^0.5.7+3
41 - uuid: ^2.0.4
42 webcolors: ^0.1.0
43
44 dev_dependencies:
45   flutter_test:
46     sdk: flutter
47
48   async: ^2.4.1
49 - auto_route_generator: ^0.5.0
50 - build_runner: ^1.10.0
51 - flutter_launcher_icons: ^0.7.5
52 - grinder: ^0.8.4
53 - injectable_generator: ^0.4.0
54 - json_serializable: ^3.3.0
55 - moor_generator: ^3.1.0
56
57 flutter:
58   uses-material-design: true
11 name: sage
12 description: A new Flutter application.
13 version: 0.1.0+00100
14 + publish_to: !none"
15
16 environment:
17   sdk: ">=2.8.0 <3.0.0"
... @@ -11,48 +11,49 @@ dependencies:
11 flutter:
12   sdk: flutter
13
14 + animations: ^2.0.3
15 + auto_route: ^4.2.1
16 + auto_size_text: ^3.0.0
17 + cached_network_image: ^3.2.1
18 + encrypt: ^5.0.1
19 + dio: ^4.0.6
20 + drift: ^1.7.1
21 + firebase_auth: ^3.5.0
22 + firebase_core: ^1.20.0
23 + flutter_auth_buttons_list: ^0.10.0
24 + flutter_reorderable_list: ^1.2.0
25 + flutter_svg: ^1.1.1+1
26 + get_it: ^7.2.0
27 + google_fonts: ^3.0.1
28 + google_sign_in: ^5.4.0
29 + injectable: ^1.5.3
30 + intl: ^0.17.0
31 + jdenticon_dart: ^2.0.0
32 + json_annotation: ^4.6.0
33 + list_diff: ^2.0.1
34 + # moor: ^3.1.0
35 path: ^1.6.4
36 + path_provider: ^2.0.11
37 + package_info: ^2.0.2
38 pretty_dio_logger: ^1.1.1
39 + provider: ^6.0.3
40 + shared_preferences: ^2.0.15
41 + uuid: ^3.0.6
42 webcolors: ^0.1.0
43
44 dev_dependencies:
45   flutter_test:
46     sdk: flutter
47
48   async: ^2.4.1
49 + auto_route_generator: ^4.2.0
50 + build_runner: ^2.2.0
51 + drift_dev: ^1.7.0
52 + flutter_launcher_icons: ^0.9.3
53 + grinder: ^0.9.2
54 + injectable_generator: ^1.1.2
55 + json_serializable: ^6.3.1
56 + # moor_generator: ^3.1.0
57 flutter:
58   uses-material-design: true

```

Figura 11: Git diff referente a atualização dos pacotes do novo Sage

No caso de um pacote específico, o *moor*, o pacote foi renomeado pelos criadores para *drift*. Mais informações sobre esta mudança podem ser encontradas na documentação oficial do pacote: “*Moor and Drift*”<sup>15</sup>.

<sup>15</sup> <https://drift.simonbinder.eu/name/>

Uma vez que a migração automática sugerida não funcionou, foi necessário realizar a migração manual de um pacote para outro, e isso incluiu:

- Alterar o arquivo *pubspec.yaml*;
- Alterar as importações;
- Alterar o código, para refletir os novos nomes das entidades disponibilizadas pela API do pacote;
- Alterar o arquivo de configuração *build.yaml*;
- Renomear os arquivos *.moor*;

Ao tentar executar o *build\_runner*, ferramenta que gera alguns arquivos necessários para a aplicação (no caso do Sage, principalmente relacionados ao banco e às rotas), a execução falhou com o erro “*Generator cannot target libraries that have not been migrated to null-safety*”.

A “*sound null safety*”, suportada pelo Dart (linguagem de programação utilizada para desenvolver com Flutter), torna os tipos das variáveis *non-nullable* por padrão, isto é, as variáveis não podem conter nulo a não ser que seja informado o contrário. Por conta da incompatibilidade apontada no erro, foi preciso alterar a versão mínima do SDK do Dart utilizado pela aplicação para 2.12.0. A partir dessa versão do Dart, *sound null safety* é suportado.

Também fez-se necessário a atualização da versão do *Gradle* (ferramenta utilizada no processo de *build* do app Android), e do *Kotlin Gradle plugin*, assim como a versão esperada do SDK do Android nos dispositivos em que o app for instalado (por conta de restrições dos pacotes utilizados).

## 4.2. Correção de erros

Ao tentar compilar e executar o aplicativo, foram identificados 289 problemas no código (vide Figura 12), dos quais 211 eram erros impeditivos, e 78 eram *warnings* com sugestões de melhoria.

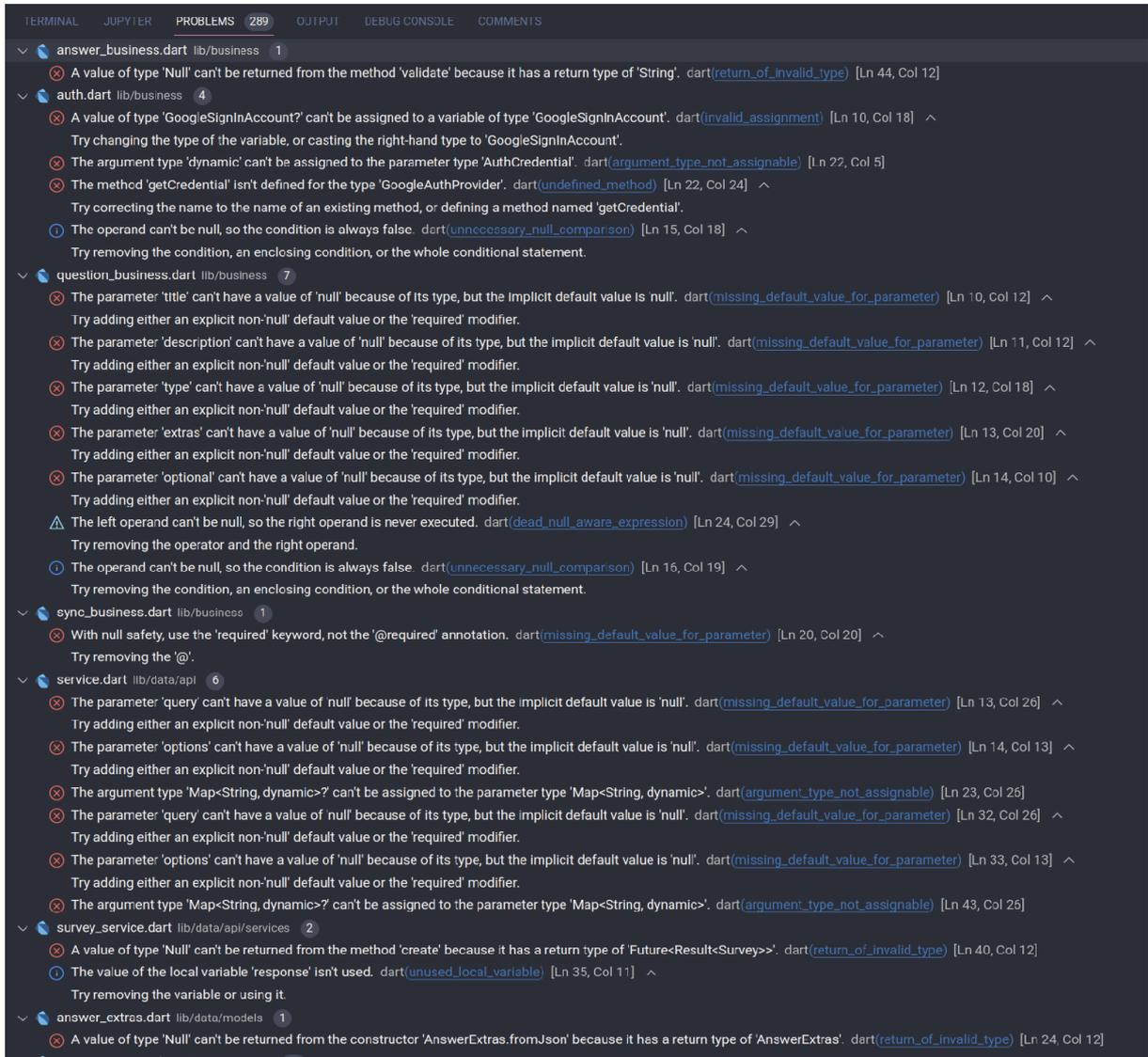


Figura 12: Recorte da lista de erros de tipagem no projeto

Como não existe forma automática de corrigir estes erros, foi preciso passar pelos arquivos e avaliar cada caso. Havia alguns erros de tipagem, outros por conta de *breaking changes* nas bibliotecas utilizadas, dentre outros. Uma vez corrigidos, foi possível rodar a aplicação.

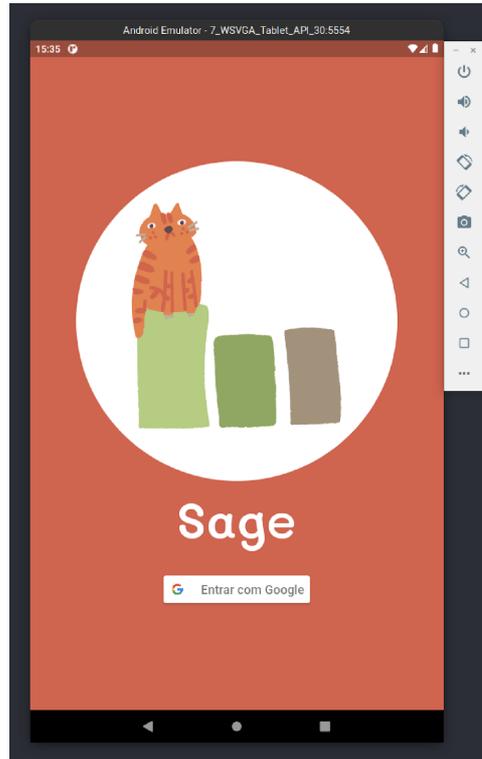


Figura 13: Sage em execução no emulador

### 4.3. Aplicação web

A aplicação existente foi portada para sua versão web, de forma que as funcionalidades seguintes pudessem ser desenvolvidas e validadas em ambas as plataformas. Para adicionar suporte web a um app existente, conforme a documentação<sup>16</sup> do Flutter, é preciso rodar o comando “flutter create .”. Isso adiciona alguns arquivos necessários para a execução da página web.

No contexto do Sage, também foi preciso realizar algumas adaptações em relação à biblioteca utilizada para o banco de dados, mais especificamente a inicialização do módulo correto de acordo com o ambiente em que a aplicação está sendo executada (mobile ou web).

A seguir, podem ser visualizadas algumas telas na versão web (Figura 14, Figura 15 e Figura 16).

---

<sup>16</sup> <https://docs.flutter.dev/get-started/web#add-web-support-to-an-existing-app>

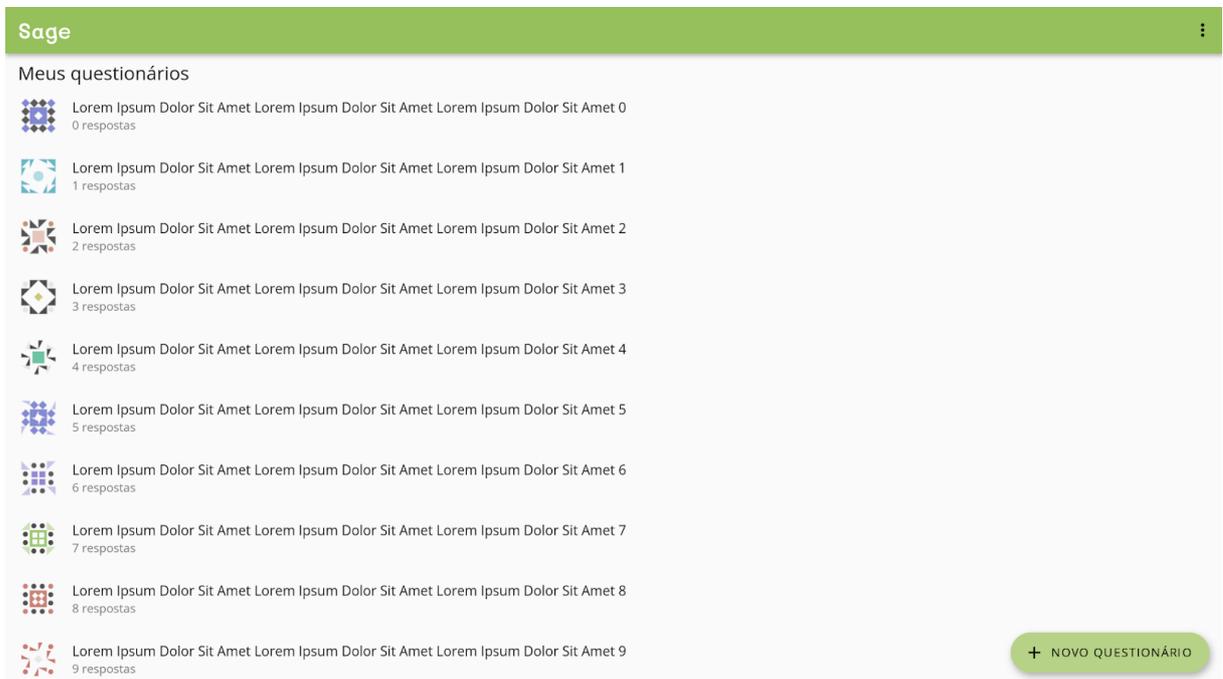


Figura 14: Tela inicial do novo Sage (versão web)

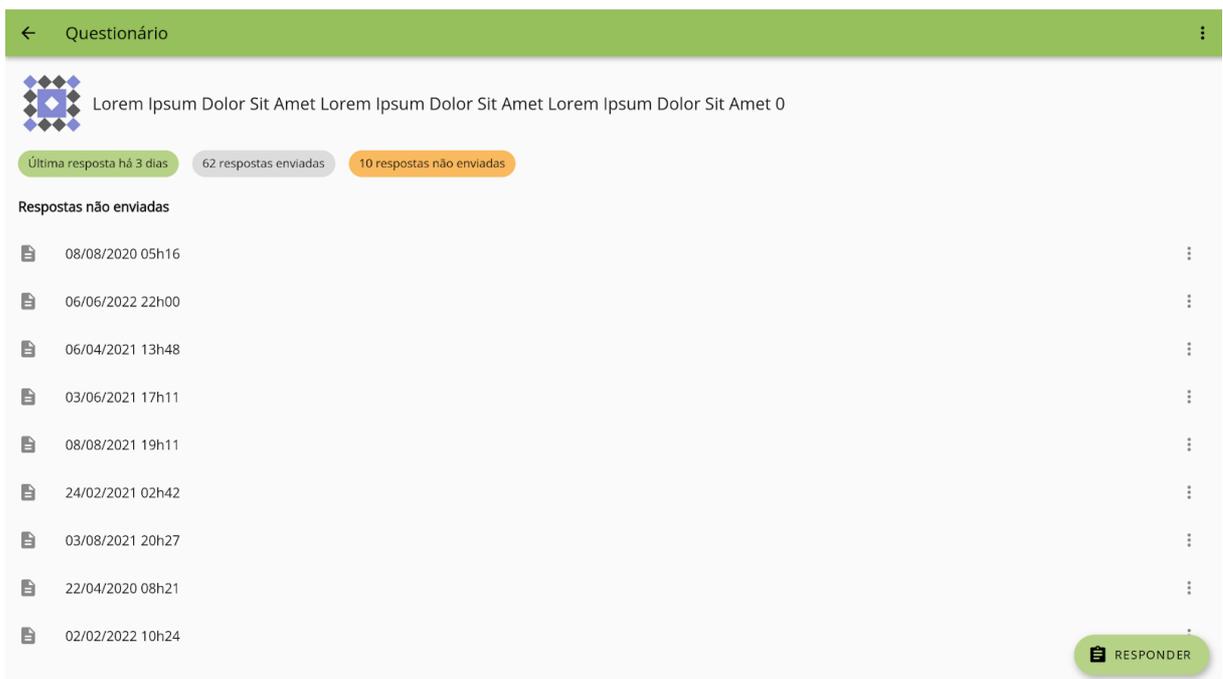


Figura 15: Tela de detalhe de questionário do novo Sage (versão web)

Questionário

Qual sua cor preferida? Opcional

Azul

Amarelo

Vermelho

Verde

Preto

PRÓXIMA →

Figura 16: Tela de resposta de questionário do novo Sage (versão web)

Com isto, atingiu-se o requisito 1.1.

## 4.4. Novos tipos de perguntas

### 4.4.1. Escala Likert

Introduzida por Rensis Likert em 1932, a escala de Likert é uma escala de mensuração de cinco pontos amplamente utilizada como instrumento de medida, em especial na área das ciências sociais. É possível fazer uso de escalas com um número de itens diferente de cinco, porém esta não se configura uma escala Likert, mas sim do “tipo Likert”. De acordo com Dalmoro (2014), a escala de três pontos é menos confiável e tem menor capacidade de demonstrar com precisão a opinião do entrevistado. Além disso, a escala de cinco pontos teve, em média, a mesma precisão e mostrou-se mais fácil e mais rápida que a escala de sete pontos.

Portanto, neste trabalho, optou-se por desenvolver este tipo de pergunta seguindo o formato padrão de cinco pontos.

Para suportar a escala Likert do lado do servidor, foi preciso declarar o novo tipo de pergunta juntamente com os demais:

#### **server/sage/models/schemas.py**

```
class QuestionType(DbEnum):
    SINGLE = 'single'
    MULTI = 'multi'
    NUMERIC = 'numeric'
    TEXT = 'text'
    LIKERT5 = 'likert5'

    ALL = [
        SINGLE,
        MULTI,
        NUMERIC,
        TEXT,
        LIKERT5,
    ]
```

Os objetos de perguntas recebidos pelo servidor do Sage são validados de acordo com uma determinada estrutura fixa (ou *schema*), que é definida no servidor e encontra-se replicada a seguir. Esse schema é definido com auxílio da biblioteca “schema”, comumente utilizada para validar estruturas de dados em Python.

A documentação<sup>17</sup> da biblioteca apresenta como diversas estruturas complexas podem ser modeladas. Apenas a título de exemplificação, é possível observar no schema a seguir que uma pergunta deve possuir um atributo “title” do tipo *str* (cadeia de caracteres), e um atributo “required” do tipo *bool* (booleano). Além disso, uma pergunta pode possuir, opcionalmente, um atributo “description” do tipo *str* que, caso não informado, recebe uma string vazia como valor padrão.

---

<sup>17</sup> <https://pypi.org/project/schema/>

**server/sage/models/utils.py**

```
question_schema = Schema({
    Optional('uuid', default=None): Or(str, None),
    'type': Or(*QuestionType.ALL),
    'title': str,
    'required': bool,
    Optional('description', default=''): str,
    Optional('extras', default=dict): {str: object},
})
```

Essa estrutura se mostrou suficiente para diversos tipos de perguntas graças ao seu atributo “extras”, que recebe as restrições específicas de cada tipo de pergunta (caso haja). Por exemplo, perguntas do tipo “numérica”, já suportadas pelo Sage, tem seus “extras” definidos pelo schema:

**server/sage/models/utils.py**

```
QuestionType.NUMERIC: Schema({
    'min': int,
    'max': int,
}),
```

Em alto nível, perguntas do tipo “numérica” podem receber dois inteiros: um que servirá como mínimo e outro que servirá como máximo, de forma que só serão aceitas respostas dentro desse intervalo. Analogamente, neste trabalho, estruturou-se o schema para os extras das perguntas do tipo escala Likert:

**server/sage/models/utils.py**

```
QuestionType.LIKERT5: Schema({
    'labels_from_1_to_5': [str],
}),
```

Conforme o schema, para este tipo de pergunta espera-se que seja informada uma lista de strings que correspondem aos rótulos das 5 opções da escala.

Uma vez implementados os aspectos supracitados no lado do servidor, uma das alterações desenvolvidas do lado do aplicativo foi a implementação do componente usado na criação de uma pergunta deste novo tipo. É através deste componente que o usuário define, pela interface, os rótulos para os pontos da escala.

O componente consiste em 5 caixas de texto (*TextFormField*) organizadas verticalmente numa coluna (*Column*). A cada caixa de texto foi atribuído um rótulo para identificar sua finalidade (por exemplo, “Rótulo para pontuação = 1”). Dentre as demais propriedades, é possível citar também a passagem de uma função que valida se o valor digitado é válido; caso este não seja, a caixa de texto é exibida num estado de erro. Por padrão, as caixas de texto são preenchidas com estes valores, respectivamente: “Muito insatisfeito”, “Insatisfeito”, “Indiferente”, “Satisfeito”, “Muito satisfeito”.

Uma versão resumida do código que define este componente pode ser visualizada a seguir.

#### **app/lib/pages/question\_create/question\_type\_layouts.dart**

```
class CreateLikert5Question extends StatefulWidget {
  final ValueChanged<QuestionExtras> onChanged;
  final QuestionExtras extras;

  const CreateLikert5Question(
    {Key? key, required this.onChanged, required this.extras})
    : super(key: key);

  @override
  _CreateLikert5QuestionState createState() =>
  _CreateLikert5QuestionState();
}

class _CreateLikert5QuestionState extends
State<CreateLikert5Question> {
  List<String> labelsFrom1To5 = [
    'Muito insatisfeito',
```

```

    'Insatisfeito',
    'Indiferente',
    'Satisfeito',
    'Muito satisfeito'
  ];

  @override
  void initState() {
    super.initState();
    final extras = widget.extras;
    print(extras);
    if (extras != null && extras is QuestionExtrasLikert5) {
      labelsFrom1To5 = extras.labelsFrom1To5;
    }
  }

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.only(
        left: 16,
        right: 16,
        top: 16,
      ),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          TextFormField(
            decoration: InputDecoration(
              border: OutlineInputBorder(),
              labelText: 'Rótulo para pontuação = 1',
              errorMaxLines: 2,
            ),
            initialValue: labelsFrom1To5[0],
            keyboardType: TextInputType.number,
            onChanged: (value) {
              setState(() {
                labelsFrom1To5[0] = value;
              });
            },
            validator: _validator,
            onSave: (_) {
              widget.onChanged(
                QuestionExtrasLikert5(labelsFrom1To5:
labelsFrom1To5));
            },
          ),
          SizedBox(height: 24),

          [...],

          SizedBox(height: 24),
          TextFormField(
            decoration: InputDecoration(
              border: OutlineInputBorder(),
              labelText: 'Rótulo para pontuação = 5',

```

```

        errorMaxLines: 2,
      ),
      initialValue: labelsFrom1To5[4],
      keyboardType: TextInputType.number,
      onChanged: (value) {
        setState(() {
          labelsFrom1To5[4] = value;
        });
      },
      validator: _validator,
      onSave: (_) {
        widget.onChanged(
          QuestionExtrasLikert5(labelsFrom1To5:
labelsFrom1To5));
      },
    ),
  ],
));
}

String? _validator(String? str) {
  if (str == null || str.isEmpty) {
    return 'Insira um rótulo';
  }

  if (str.length > 45) {
    return 'O rótulo não pode ter mais de 45 caracteres';
  }

  return null;
}
}
}

```

14:05

← Nova pergunta

Pergunta

Qual o seu nível de satisfação com o nosso produto?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta

Likert de 5 pontos

Obrigatório

Obrigatória

Rótulo para pontuação = 1

Muito insatisfeito

Rótulo para pontuação = 2

Insatisfeito

Rótulo para pontuação = 3

Indiferente

Rótulo para pontuação = 4

Satisfeito

Rótulo para pontuação = 5

Muito satisfeito

Figura 17: Tela de criação de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão mobile)

← Nova pergunta

Pergunta

Qual o seu nível de satisfação com o nosso produto?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta

Likert de 5 pontos

Obrigatório

Obrigatória

Rótulo para pontuação = 1

Muito insatisfeito

Rótulo para pontuação = 2

Insatisfeito

Rótulo para pontuação = 3

Indiferente

Rótulo para pontuação = 4

Satisfeito

Rótulo para pontuação = 5

Figura 18: Tela de criação de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão web)

Outro ponto implementado foi o componente usado pelo usuário para responder uma pergunta deste tipo. Para isso, 5 botões de opção (*Radio*) foram dispostos horizontalmente numa linha (*Row*). Para cada botão de opção foi atribuído um valor interno entre “1” e “5”, bem como o seu respectivo rótulo (exibido abaixo do botão), obtido a partir dos extras da pergunta. A seguir, pode ser visualizada uma versão resumida do código que estrutura este componente.

### **app/lib/pages/answer/answer\_type\_layouts.dart**

```
class AnswerLikert5 extends StatefulWidget {
  final Question question;
  final int value;
  final ValueChanged<dynamic> onChanged;
  final String validation;

  const AnswerLikert5({
    Key? key,
    required this.question,
    required this.value,
    required this.onChanged,
    required this.validation,
  }) : super(key: key);

  @override
  _AnswerLikert5State createState() => _AnswerLikert5State();
}

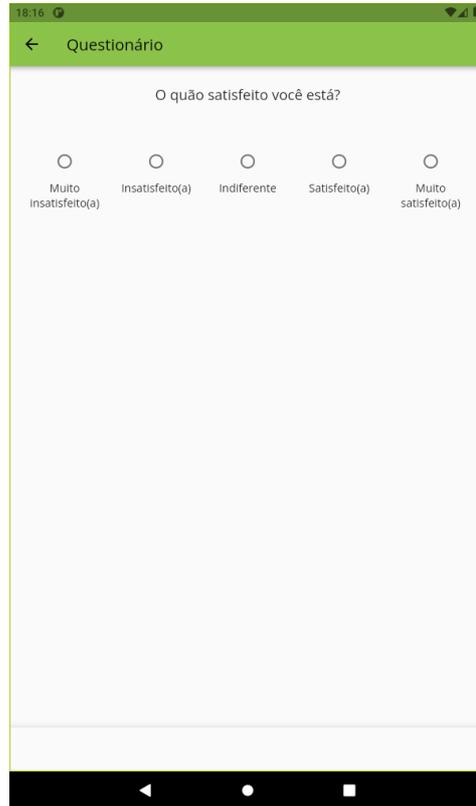
class _AnswerLikert5State extends State<AnswerLikert5> {
  @override
  Widget build(BuildContext context) {
    final question = widget.question;

    assert(question.type == QuestionType.likert5);
    assert(question.extras is QuestionExtrasLikert5);

    final extras = question.extras as QuestionExtrasLikert5;

    return Row(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        SizedBox(width: 16),
        Expanded(
          child: Column(
            children: [
              Radio(
                value: "1",
```





18:16

← Questionário

O quão satisfeito você está?

Muito insatisfeito(a)

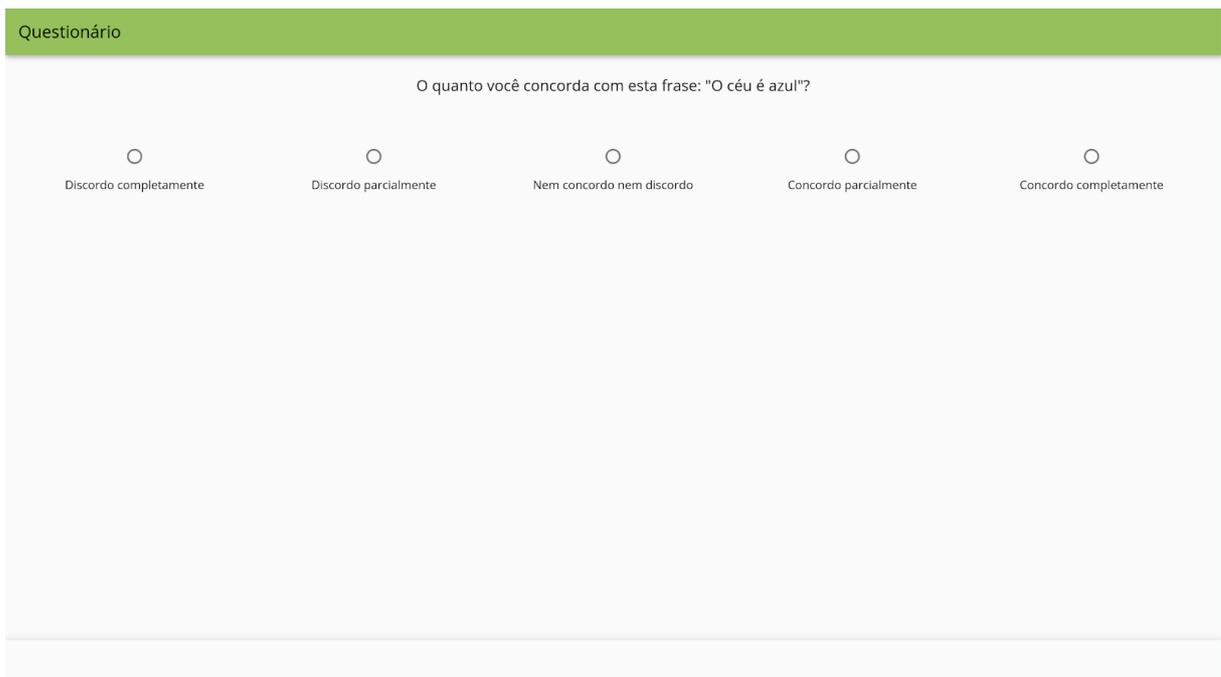
Insatisfeito(a)

Indiferente

Satisfeito(a)

Muito satisfeito(a)

Figura 19: Tela de resposta de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão mobile)



Questionário

O quanto você concorda com esta frase: "O céu é azul"?

Discordo completamente

Discordo parcialmente

Nem concordo nem discordo

Concordo parcialmente

Concordo completamente

Figura 20: Tela de resposta de questionário do novo Sage, pergunta “Likert de 5 pontos” (versão web)

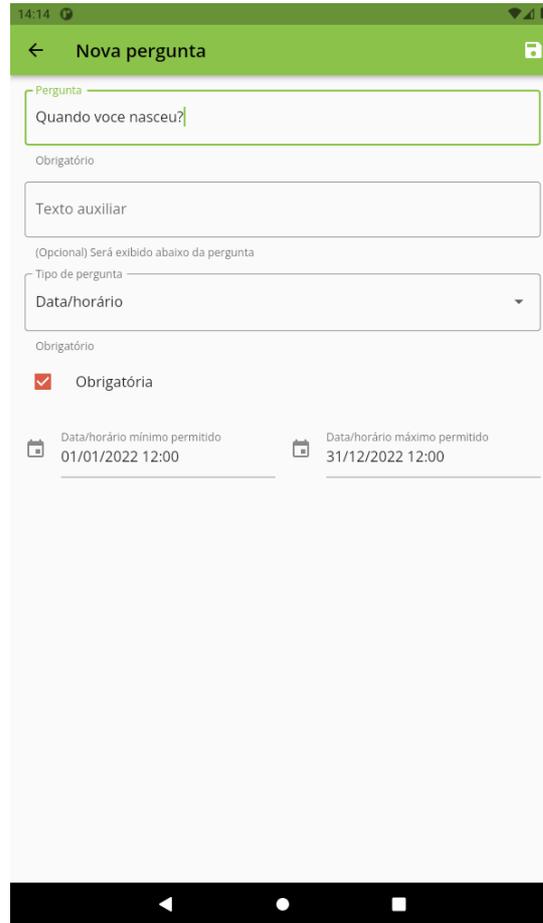
Com isto, atingiu-se o requisito 2.1.

#### **4.4.2. Data/horário**

O novo tipo Data/horário consiste em uma forma de coletar a data e horário de ocorrência de um evento; em outras palavras, uma espécie de *timestamp*. Em alguns domínios, é importante registrar tanto data quanto horário para maior precisão.

O processo de desenvolvimento deste tipo de pergunta é análogo ao da escala Likert, uma vez que também foi necessário registrá-lo juntamente com os demais tipos no servidor e modelar os seus “extras” adequadamente.

Para este tipo de pergunta, os extras consistiram em uma data/horário mínimo e uma data/horário máximo. Isto porque, por exemplo, para a pergunta “Quando você prefere que seja realizado o evento de confraternização da empresa?”, pode ser interessante restringir o conjunto de respostas a datas no futuro dentro de um determinado intervalo que faça sentido no contexto.



14:14

← Nova pergunta

Pergunta

Quando voce nasceu?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta

Data/horário

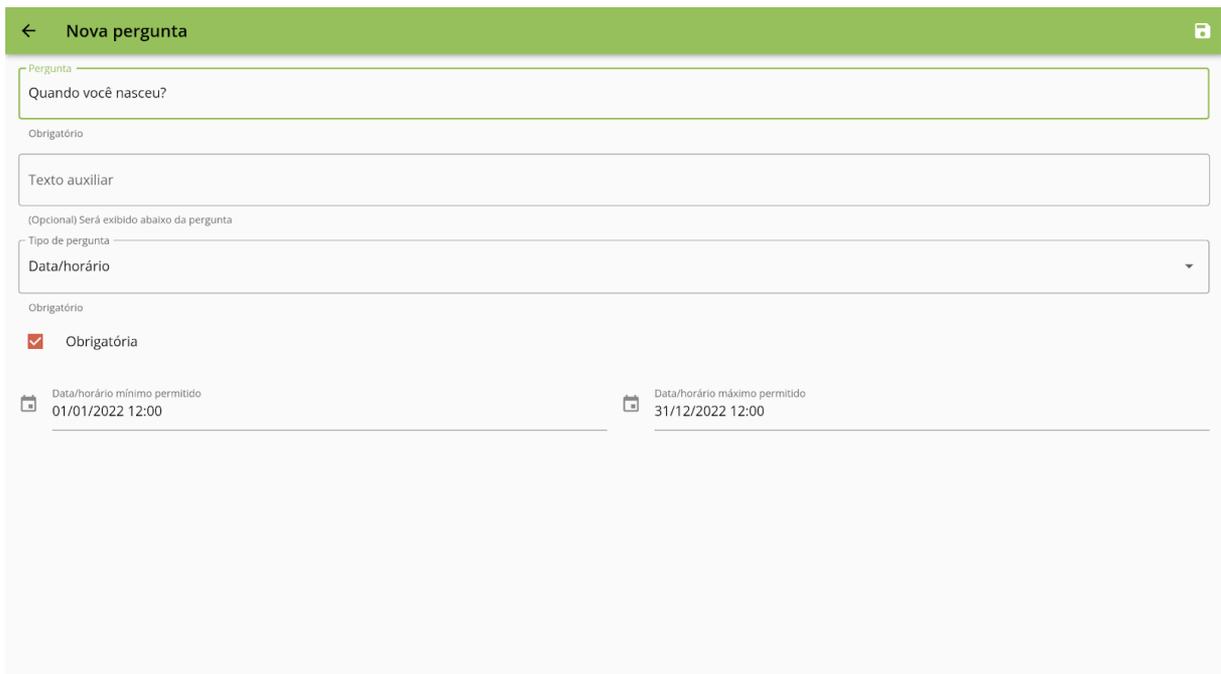
Obrigatório

Obrigatória

Data/horário mínimo permitido 01/01/2022 12:00

Data/horário máximo permitido 31/12/2022 12:00

Figura 21: Tela de criação de questionário do novo Sage, pergunta “Data/horário” (versão mobile)



← Nova pergunta

Pergunta

Quando você nasceu?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta

Data/horário

Obrigatório

Obrigatória

Data/horário mínimo permitido 01/01/2022 12:00

Data/horário máximo permitido 31/12/2022 12:00

Figura 22: Tela de criação de questionário do novo Sage, pergunta “Data/horário” (versão web)

Ao clicar nas caixas de texto, tanto na tela de criação da pergunta quanto na tela de resposta, é aberto um diálogo que permite a seleção da data e, em sequência, a seleção do horário. Ao finalizar, o resultado é salvo e exibido na caixa de texto com a formatação adequada. Para exibição do diálogo de seleção da data e horário, utilizou-se a biblioteca *date\_time\_picker* disponível para Flutter, que provê um widget para esse fim.

É possível passar alguns parâmetros para esse widget de forma a customizá-lo, dentre eles:

- **type:** o tipo de diálogo a ser apresentado (apenas horário, apenas data, ou ambos data e horário em sequência);
- **dateMask:** máscara usada para formatar o valor selecionado (exemplo: 'dd/MM/yyyy HH:mm');
- **initialValue:** valor inicial que será pré-selecionado;
- **firstDate:** valor mínimo permitido;
- **lastDate:** valor máximo permitido.

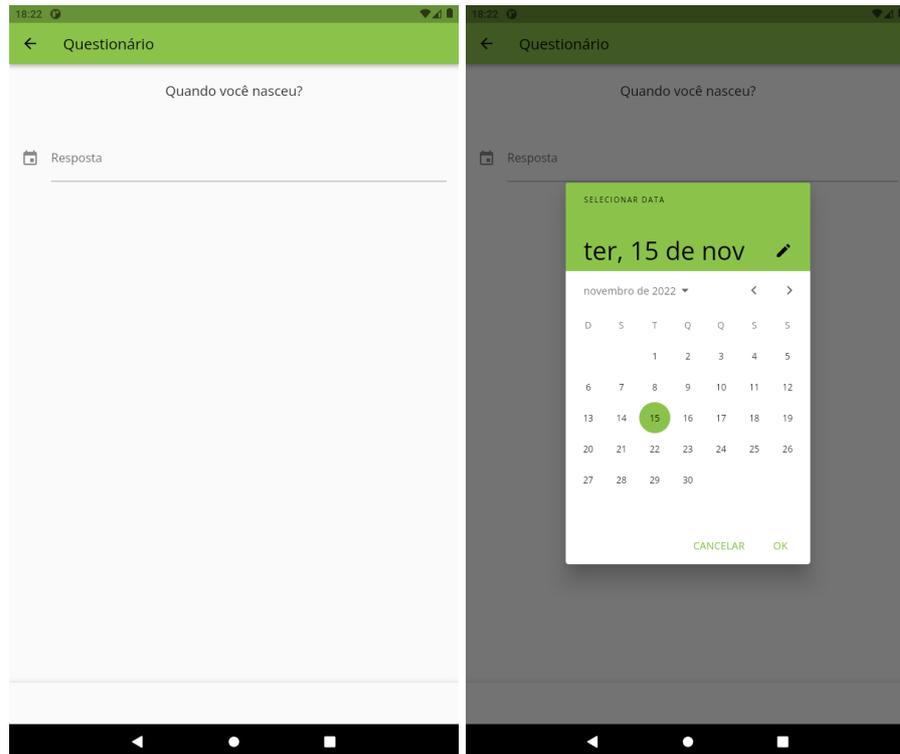


Figura 23: Tela de resposta de questionário do novo Sage, pergunta “Data/horário” (versão mobile) – parte 1

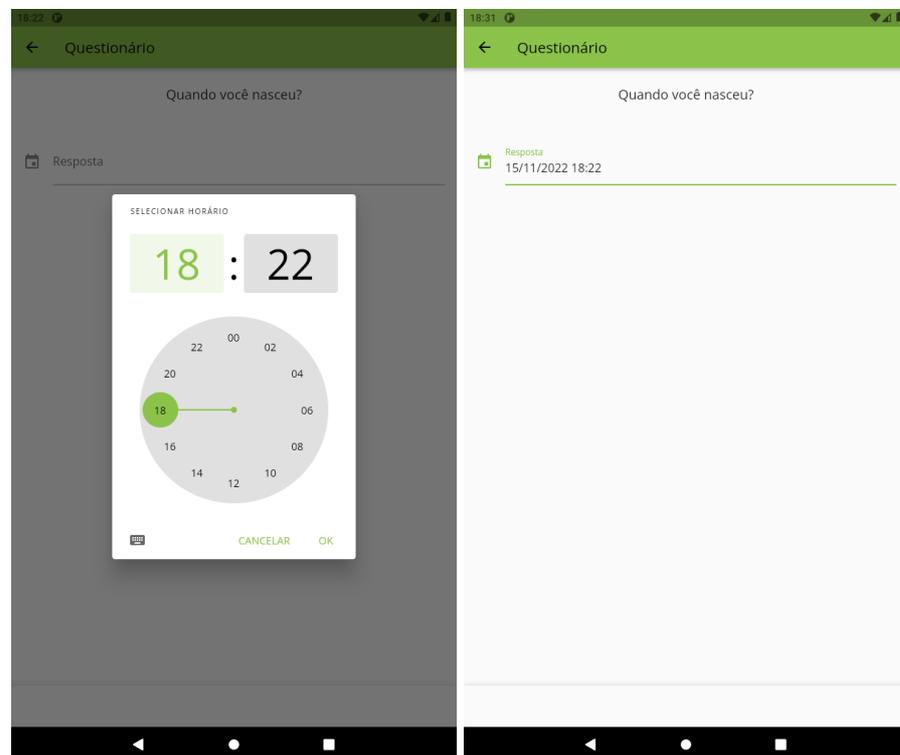


Figura 24: Tela de resposta de questionário do novo Sage, pergunta “Data/horário” (versão mobile) – parte 2

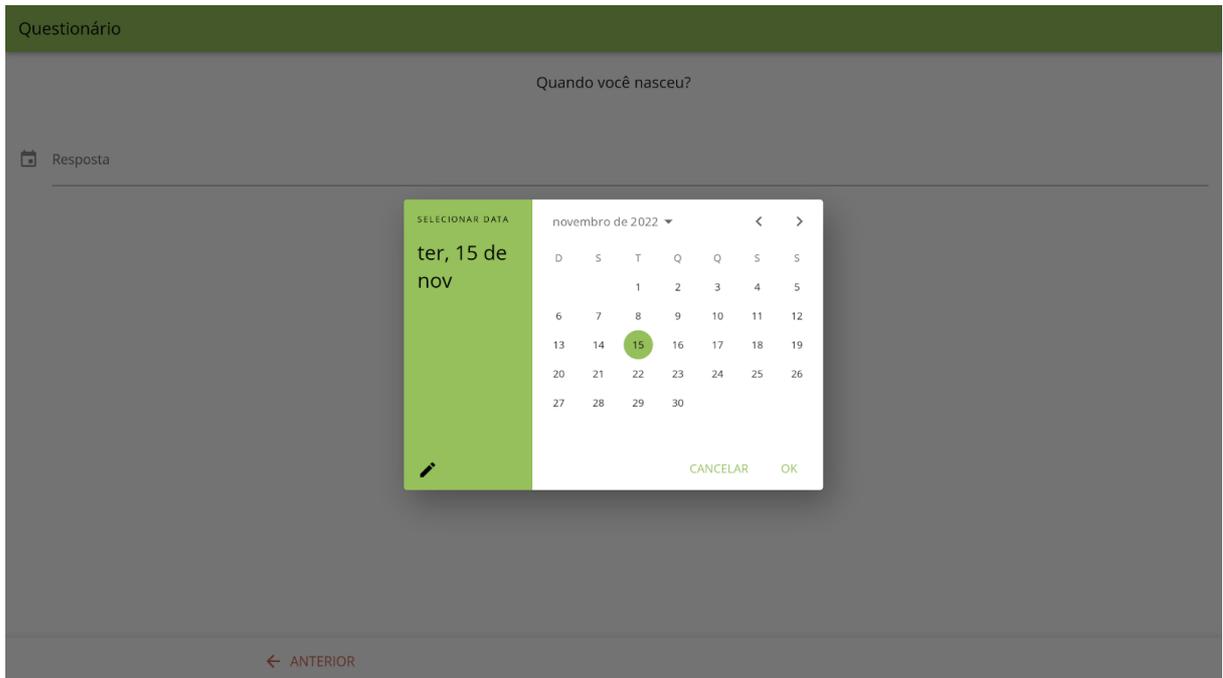


Figura 25: Tela de resposta de questionário do novo Sage, pergunta "Data/horário" (versão web) – parte 1

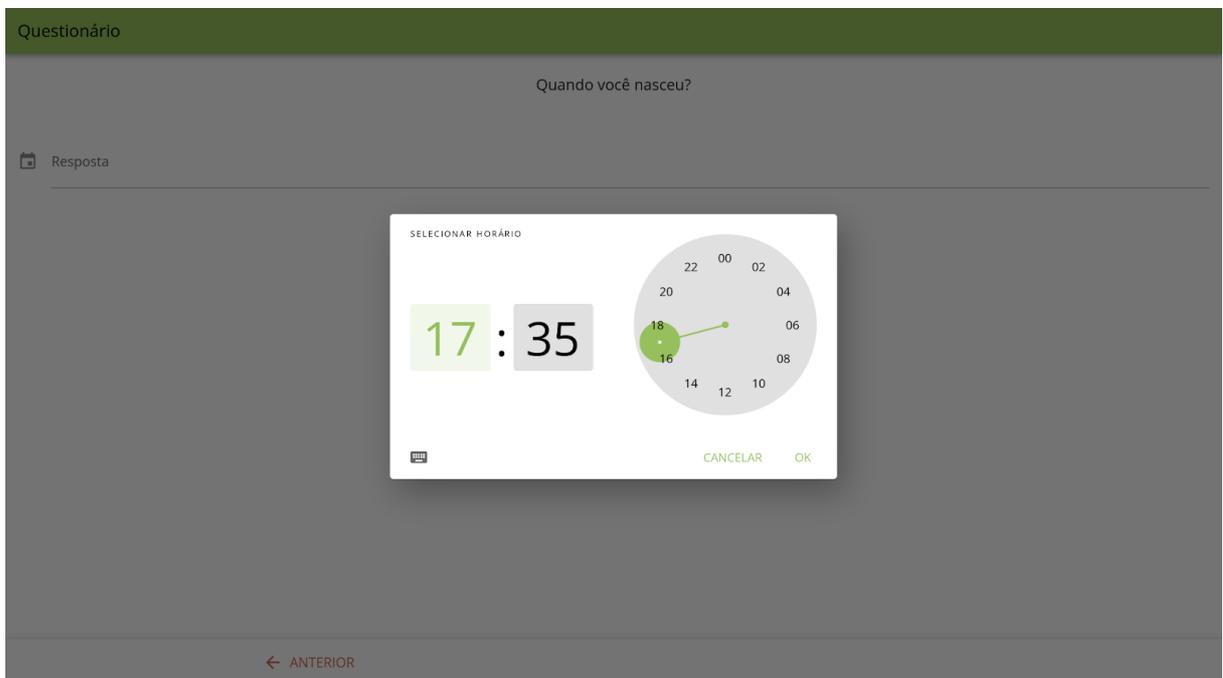


Figura 26: Tela de resposta de questionário do novo Sage, pergunta "Data/horário" (versão web) – parte 2

Com isto, atingiu-se o requisito 2.2.

#### 4.4.3. Data

O novo tipo Data é uma espécie de subtipo do tipo Data/horário. Ele se mostra mais adequado quando o horário de ocorrência do evento não é relevante. Por exemplo, para a pergunta “Quando foi a última Páscoa?” é suficiente saber apenas a data (dia/mês/ano), sem que seja necessário o horário.

Para desenvolver este tipo de pergunta, o processo é análogo aos demais. Os extras foram modelados com uma data mínima permitida e uma data máxima permitida, seguindo o mesmo princípio do tipo de pergunta anterior.

14:15

← Nova pergunta

Pergunta

Quando voce nasceu?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta

Data

Obrigatório

Obrigatória

Data mínima permitida 01/01/2022

Data máxima permitida 31/12/2022

Figura 27: Tela de criação de questionário do novo Sage, pergunta “Data” (versão mobile)

← Nova pergunta

Pergunta

Quando você nasceu?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta

Data

Obrigatório

Obrigatória

Data mínima permitida  
01/01/2022

Data máxima permitida  
31/12/2022

Figura 28: Tela de criação de questionário do novo Sage, pergunta “Data” (versão web)

Para exibição do diálogo de seleção de data, utilizou-se a mesma biblioteca já citada anteriormente, com a diferença do parâmetro de tipo passado (*DateTimePickerType.date* ao invés de *DateTimePickerType.dateTime*).

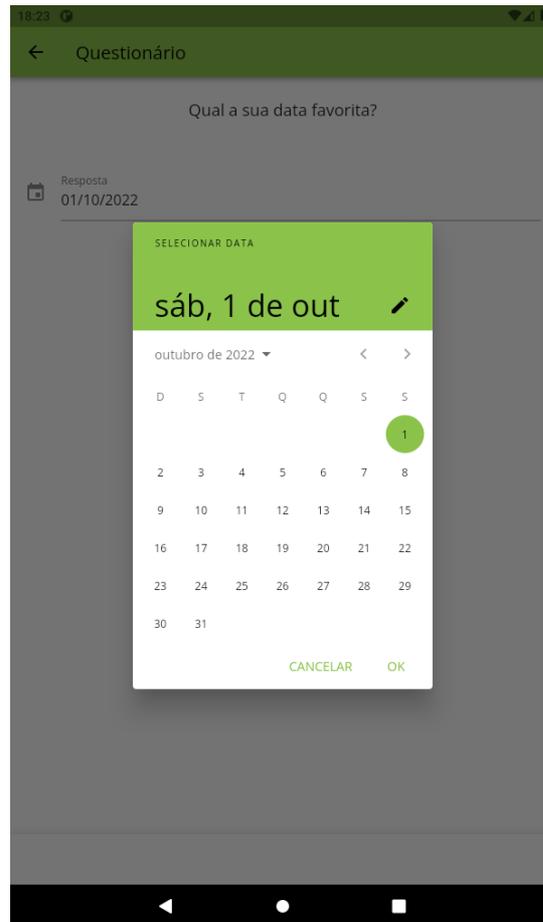


Figura 29: Tela de resposta de questionário do novo Sage, pergunta “Data” (versão mobile)

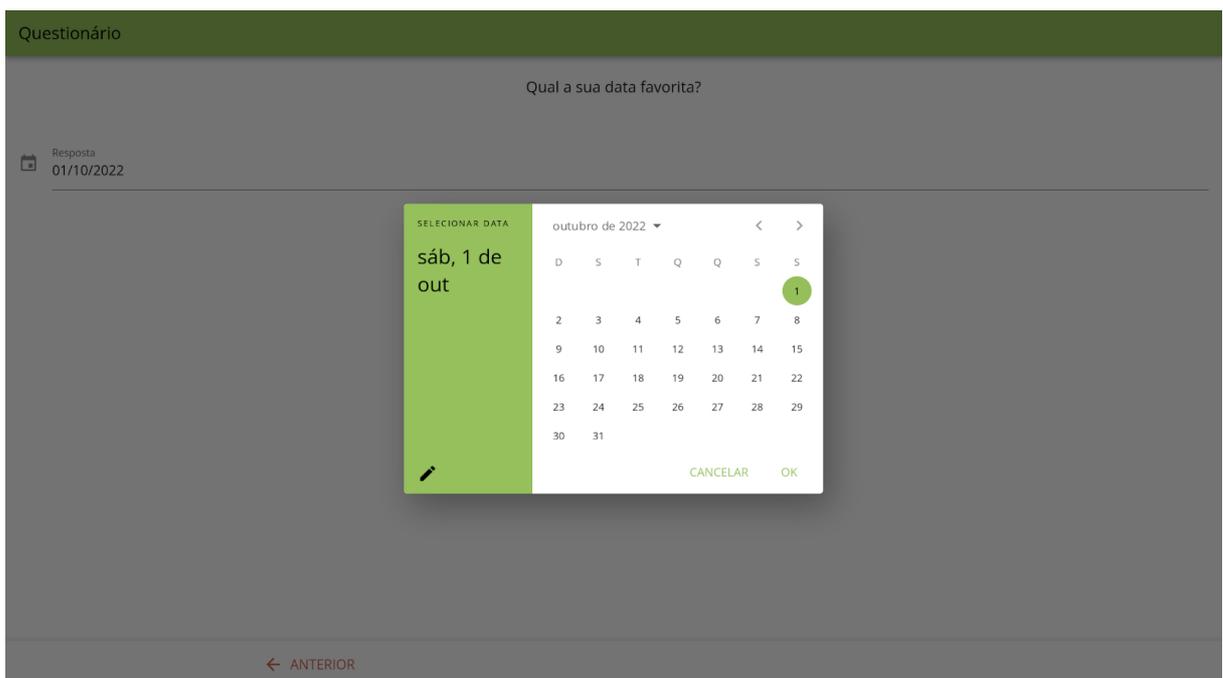
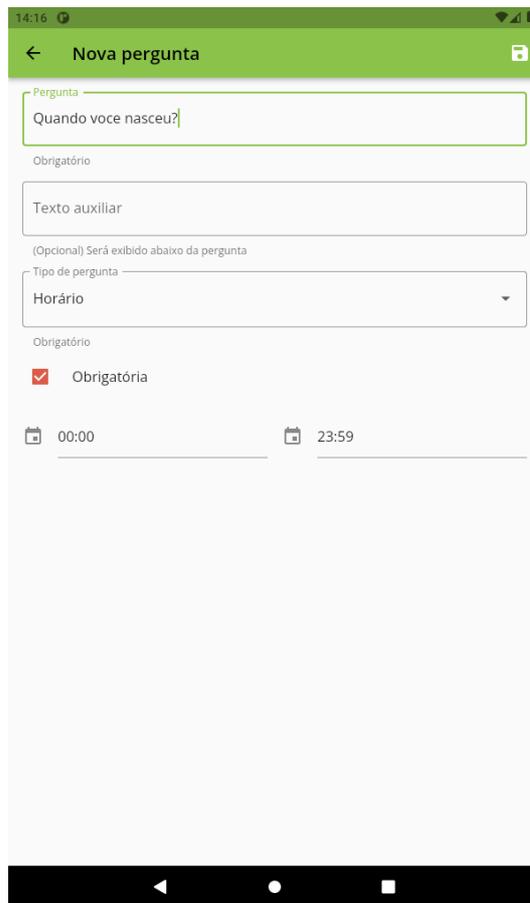


Figura 30: Tela de resposta de questionário do novo Sage, pergunta “Data” (versão web)

Com isto, atingiu-se o requisito 2.3.

#### 4.4.4. Horário

Por fim, o novo tipo Horário se mostra mais adequado quando a data não é importante. Por exemplo, para as perguntas “Em qual horário você costuma acordar diariamente?” ou “Em qual horário você prefere sair do trabalho?”, apenas o horário pode ser suficiente como resposta.



The screenshot shows a mobile application interface for creating a new question. The title bar is green and contains a back arrow, the text "Nova pergunta", and a share icon. Below the title bar, there is a text input field labeled "Pergunta" containing the text "Quando voce nasceu?". Below this field is a label "Obrigatório". There is a text input field labeled "Texto auxiliar". Below this field is a label "(Opcional) Será exibido abaixo da pergunta". There is a dropdown menu labeled "Tipo de pergunta" with the selected option "Horário". Below this dropdown is a label "Obrigatório". There is a checked checkbox labeled "Obrigatória". At the bottom, there is a time range selection with a calendar icon, "00:00", a horizontal line, another calendar icon, and "23:59". The bottom of the screen shows the standard Android navigation bar.

Figura 31: Tela de criação de questionário do novo Sage, pergunta “Horário” (versão mobile)

← Nova pergunta

Pergunta  
Quando você nasceu?

Obrigatório

Texto auxiliar

(Opcional) Será exibido abaixo da pergunta

Tipo de pergunta  
Horário

Obrigatório

Obrigatória

00:00 23:59

Figura 32: Tela de criação de questionário do novo Sage, pergunta “Horário” (versão web)

Análogo aos dois tipos anteriores, ao clicar na caixa de texto, é aberto um diálogo de seleção de horário que foi feito com a mesma biblioteca, alterando-se o parâmetro de tipo utilizado (*DateTimePickerType.time*).

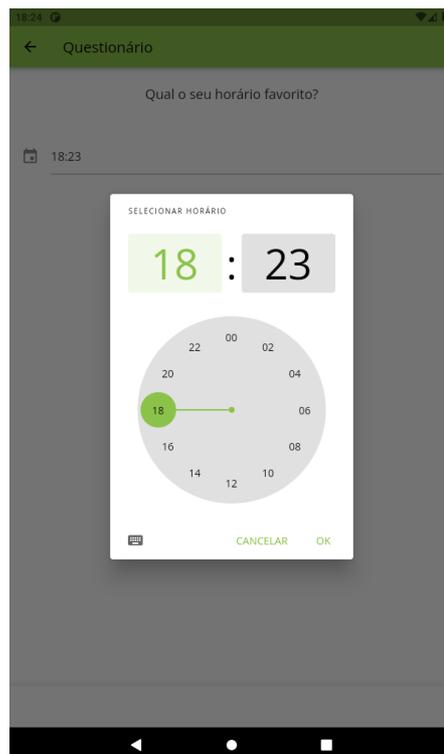


Figura 33: Tela de resposta de questionário do novo Sage, pergunta “Horário” (versão mobile)

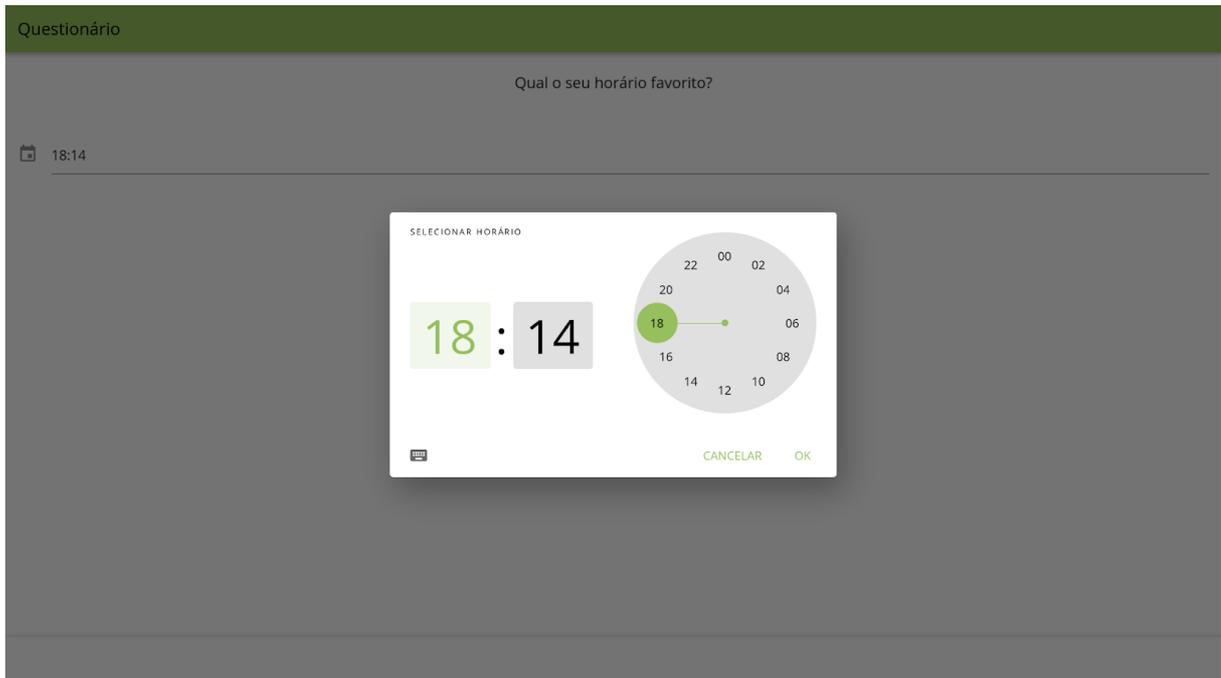


Figura 34: Tela de resposta de questionário do novo Sage, pergunta “Horário” (versão web)

Com isto, atingiu-se o requisito 2.4.

## 4.5. Convites entre entrevistadores

A funcionalidade de convite entre entrevistadores visa permitir que um entrevistador crie um questionário e compartilhe-o com outros entrevistadores, a fim de que eles apliquem o mesmo questionário, reaproveitando o conjunto de perguntas.

### 4.5.1. Entidade de Convite

Para esta funcionalidade, foi necessário modelar uma entidade de Convite (*Invite*), de forma que o diagrama Entidade-Relacionamento original do banco de dados do servidor foi atualizado, conforme a Figura 35.

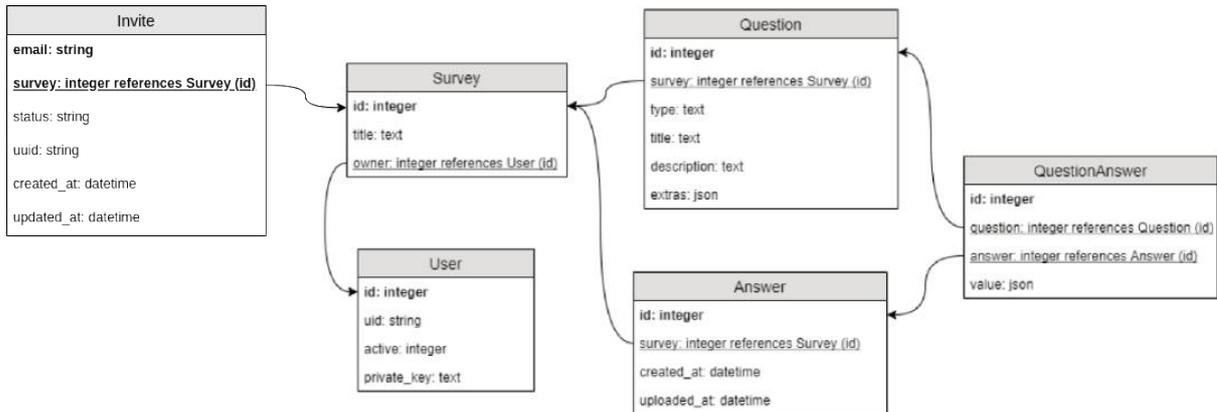


Figura 35: Diagrama Entidade-Relacionamento do banco de dados do servidor do novo Sage

À entidade de Convite, atribuiu-se as seguintes propriedades:

- **email:** e-mail do entrevistador que receberá o convite;
- **survey:** o questionário para o qual o entrevistador será convidado;
- **status:** o status do convite, dentre as opções definidas no enumerador *InviteStatus* (pendente, aceito, recusado ou cancelado);
- **uuid:** um identificador único para o convite;
- **created\_at** e **updated\_at:** timestamps para controle do momento em que os registros foram criados/atualizados.

Destas, utilizou-se *email* e *survey* como chave primária composta para a tabela de convites, uma vez que um entrevistador não pode receber mais de um convite por questionário, e os dois atributos em conjunto são uma forma de identificar um convite.

Prosseguindo, ORM (*Object Relational Mapper*) ou, em português, mapeamento objeto-relacional, é uma técnica que aproxima o paradigma de desenvolvimento de aplicações orientadas a objetos ao paradigma do banco de dados relacional. O uso desta técnica é realizado através de um mapeador

objeto-relacional que geralmente é uma biblioteca ou framework que ajuda no mapeamento e uso do banco de dados (FOWLER, 2002).

No servidor do Sage, a biblioteca Python que já estava sendo utilizada como ORM era a PonyORM<sup>18</sup>. Com auxílio dela, iniciou-se o desenvolvimento da funcionalidade de convite entre entrevistadores por meio da definição da nova entidade apresentada. A seguir, é possível observar o código resumido das classes envolvidas.

### **server/sage/models/db.py**

```
class Invite(db.Entity):
    email = Required(str)
    survey = Required(Survey)
    PrimaryKey(email, survey)
    status = Required(
        str,
        py_check=InviteStatus.validate,
        default=InviteStatus.PENDING
    )

    uuid = Required(
        uuid.UUID,
        index=True,
        unique=True,
        default=uuid.uuid4
    )

    created_at = Required(datetime, default=datetime.now)
    updated_at = Required(datetime, default=datetime.now)
```

### **server/sage/models/utils.py**

```
class InviteStatus(DbEnum):
    PENDING = 'pending'
    ACCEPTED = 'accepted'
    REFUSED = 'refused'
    CANCELED = 'canceled'
    ALL = [
        PENDING,
        ACCEPTED,
        REFUSED,
        CANCELED,
```

---

<sup>18</sup> <https://ponyorm.org/>

]

As transições válidas entre os estados de um convite são ilustradas no diagrama da Figura 36.

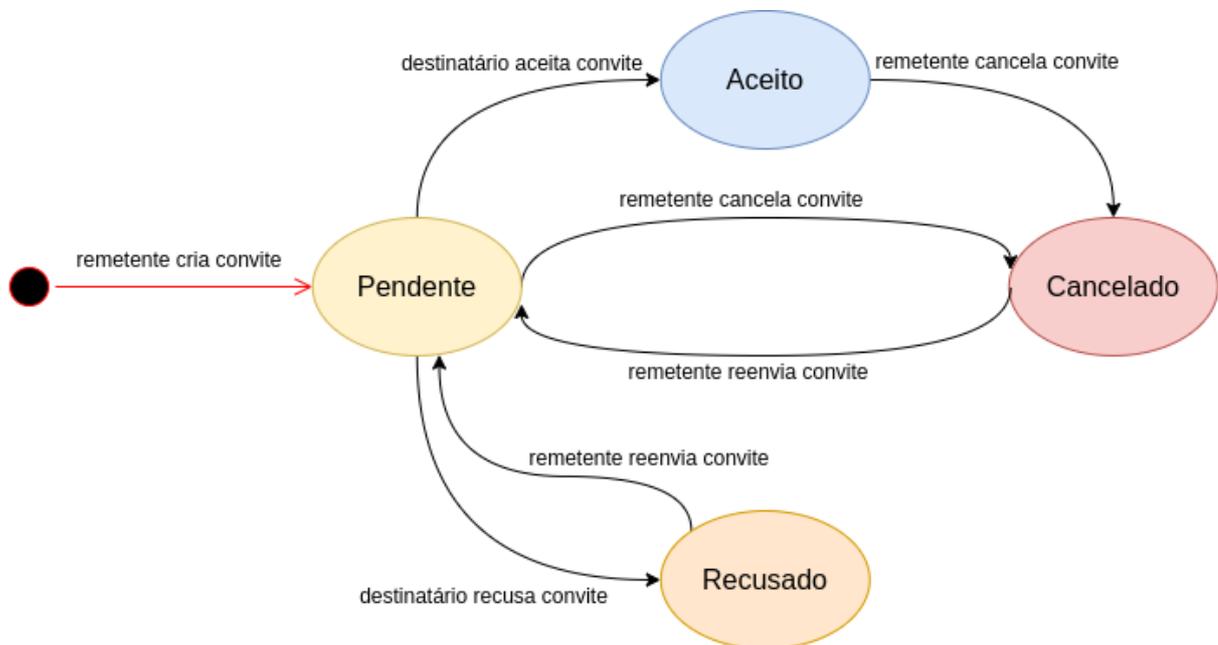


Figura 36: Diagrama de estados de um Convite no novo Sage

#### 4.5.2. Lógica de negócio (*business logic*)

Em seguida, foram desenvolvidas as funções responsáveis pela lógica de negócio relacionada aos convites. Por exemplo, a função *create\_invite* (reproduzida a seguir) recebe os dados necessários para criar um convite e faz algumas verificações antes de criá-lo de fato: é preciso checar se o questionário informado existe e se o usuário fazendo a requisição é dono deste questionário (para que possa convidar outros entrevistadores).

**server/sage/business/invites.py**

```
def create_invite(invite_data, user: User):
    survey = select(
```

```

    s for s in Survey if s.uuid == UUID(invite_data['survey_uuid'])
).first()

if survey is None:
    raise Forbidden('survey not found')

if survey.owner != user:
    raise Forbidden('user is not the owner of this survey')

invite = Invite(
    email=invite_data['email'],
    survey=survey,
)

return invite

```

De maneira análoga, outras funções desenvolvidas foram:

- **cancel\_invite:** usada para cancelar um convite que foi enviado pelo usuário em questão; checa se o usuário é dono do questionário;
- **accept\_invite:** usada para aceitar um convite recebido pelo usuário em questão; checa se o usuário recebeu o convite em questão e se o convite ainda está pendente;
- **refuse\_invite:** usada para recusar um convite recebido pelo usuário em questão; checa se o usuário recebeu o convite em questão e se o convite ainda está pendente;
- **list\_sent\_invites:** usada para listar convites enviados pelo usuário em questão;
- **list\_received\_invites:** usada para listar convites recebidos pelo usuário em questão;

### 4.5.3. Rotas

Para que as aplicações (web e mobile) pudessem acessar no servidor os recursos relacionados aos convites, foi preciso definir as rotas a serem

disponibilizadas. Elas estão listadas no Quadro 5, acompanhadas de sua finalidade, método HTTP esperado, endpoint, e corpo da requisição esperado.

Quadro 5: Rotas de convite no servidor do novo Sage

Finalidade	Método HTTP	Endpoint	Body
Criar convite	POST	/invites/	{ "email": str, "survey_uuid": str }
Cancelar convite enviado	POST	/invites/:inviteUuid/cancel	-
Aceitar convite recebido	POST	/invites/:inviteUuid/accept	-
Recusar convite recebido	POST	/invites/:inviteUuid/refuse	-
Listar convites enviados	GET	/invites/sent	-
Listar convites recebidos	GET	/invites/received	-

A título de exemplo, é possível visualizar a seguir como a rota para cancelar convite é declarada no código.

#### **server/sage/routes/invites.py**

```
@bp.route('/<string:inviteUuid>/cancel', methods=['POST'])
@auth_required
@db_session
def invite_cancel(user, inviteUuid):
    invite = Invite.get(uuid=inviteUuid)

    invite = cancel_invite(invite, user)

    data = invite.as_dict()
    response = jsonify(data)

    return response
```

Observa-se que esta rota:

- Recebe na própria rota o parâmetro *inviteUuid* (uma string), que é o *uuid* (identificador único) do convite em questão;
- Responde apenas ao método HTTP “POST”;
- Através da annotation “*@auth\_required*” já definida no servidor, exige autenticação através de token (para identificar o usuário que fez a chamada);
- Obtém o convite em questão no banco de dados, cancela-o usando a função de negócio *cancel\_invite*, converte a resposta para o formato JSON e a retorna.

As demais rotas seguem uma estrutura semelhante.

#### 4.5.4. Testes unitários

A fim de garantir um correto funcionamento das novas rotas relacionadas a convites, foram desenvolvidos testes unitários que simulam requisições para as rotas e verificam se os resultados recebidos estão conforme esperado.

Por exemplo, no teste da rota usada para aceitar um convite, primeiramente são criados os registros necessários para o teste (o usuário dono do questionário, o questionário, o usuário a ser convidado e o convite), e então é chamada a rota. Neste caso, considerando que o retorno da rota é o convite atualizado, espera-se I) que ele seja o convite em questão (isto é, que o *uuid* corresponda), e II) que ele tenha recebido o status de “Aceito”.

#### **server/sage/tests/test\_invites.py**

```
@db_session
def test_invites_invite_accept(client):
    owner_email = 'owner@email.com'
```

```

invited_email = 'invited@email.com'
owner = User(
    uid = str(uuid4()),
    email = owner_email,
    private_key = RSAKey.generate().to_private_pem()
)
survey = Survey(
    owner = owner,
    title = 'Dummy survey',
)
invited = User(
    uid = str(uuid4()),
    email = invited_email,
    private_key = RSAKey.generate().to_private_pem()
)
invite = Invite(
    email = invited.email,
    survey = survey,
)

response = client.post(f'/invites/{invite.uuid}/accept', headers={
'mockUserId': invited.uid })

assert response.json['uuid'] == str(invite.uuid)
assert response.json['status'] == InviteStatus.ACCEPTED

```

Os demais testes seguem uma estrutura semelhante. As novas rotas possuem cobertura de 100%. No entanto, as rotas existentes não possuíam testes automatizados.

#### 4.5.5. Aplicação

A seguir, são descritas algumas alterações que foram desenvolvidas para suportar esta nova funcionalidade no lado da aplicação. Na tela de detalhe de questionário, foi adicionada ao menu a opção de “Compartilhar” questionário, conforme Figura 37. A exemplo, as próximas capturas de tela são do aplicativo móvel, porém a funcionalidade também está disponível na aplicação web e o layout é análogo.

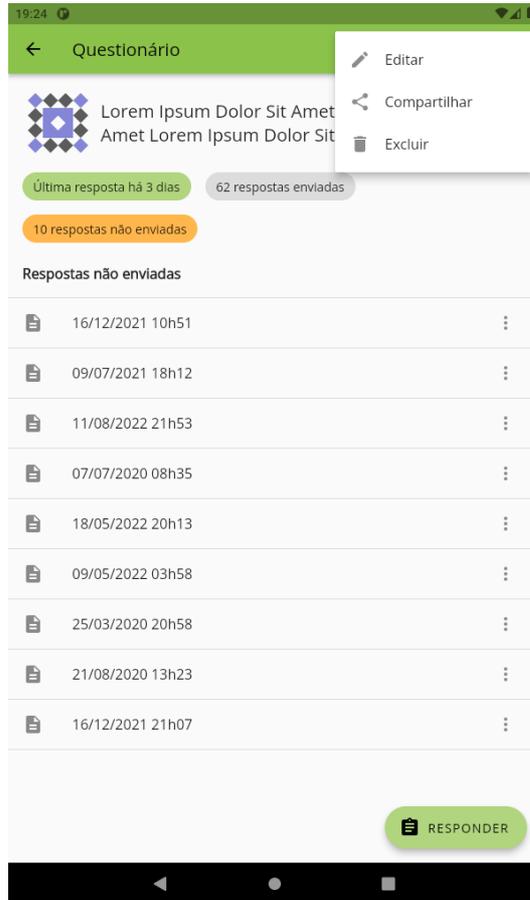


Figura 37: Tela de detalhe de questionário do novo Sage, detalhe menu (versão mobile)

Ao clicar nela, o usuário é redirecionado para uma nova tela (Figura 38), onde são listados os convites para este questionário que já foram enviados e seus respectivos status.

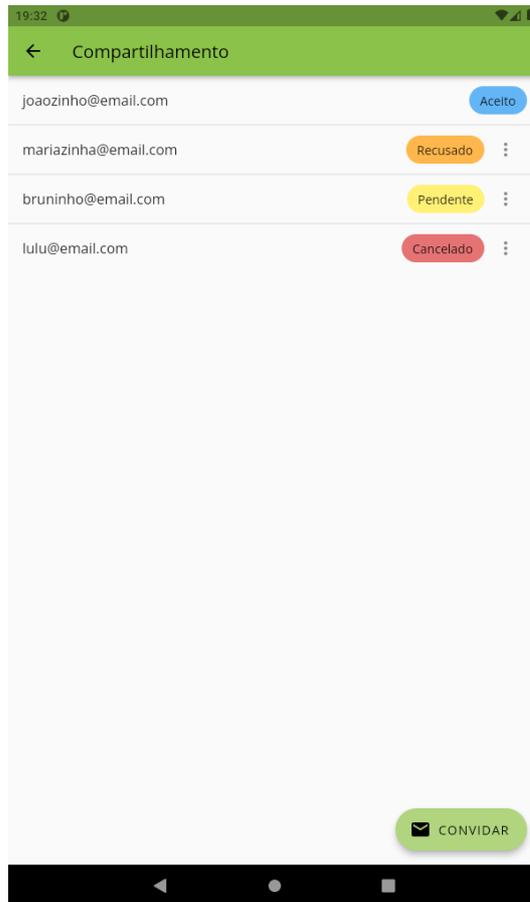


Figura 38: Tela de compartilhamento de questionário do novo Sage (versão mobile)

Nesta lista, convites pendentes podem ser cancelados, e convites recusados ou cancelados podem ser reenviados, ações que podem ser visualizadas na Figura 39.

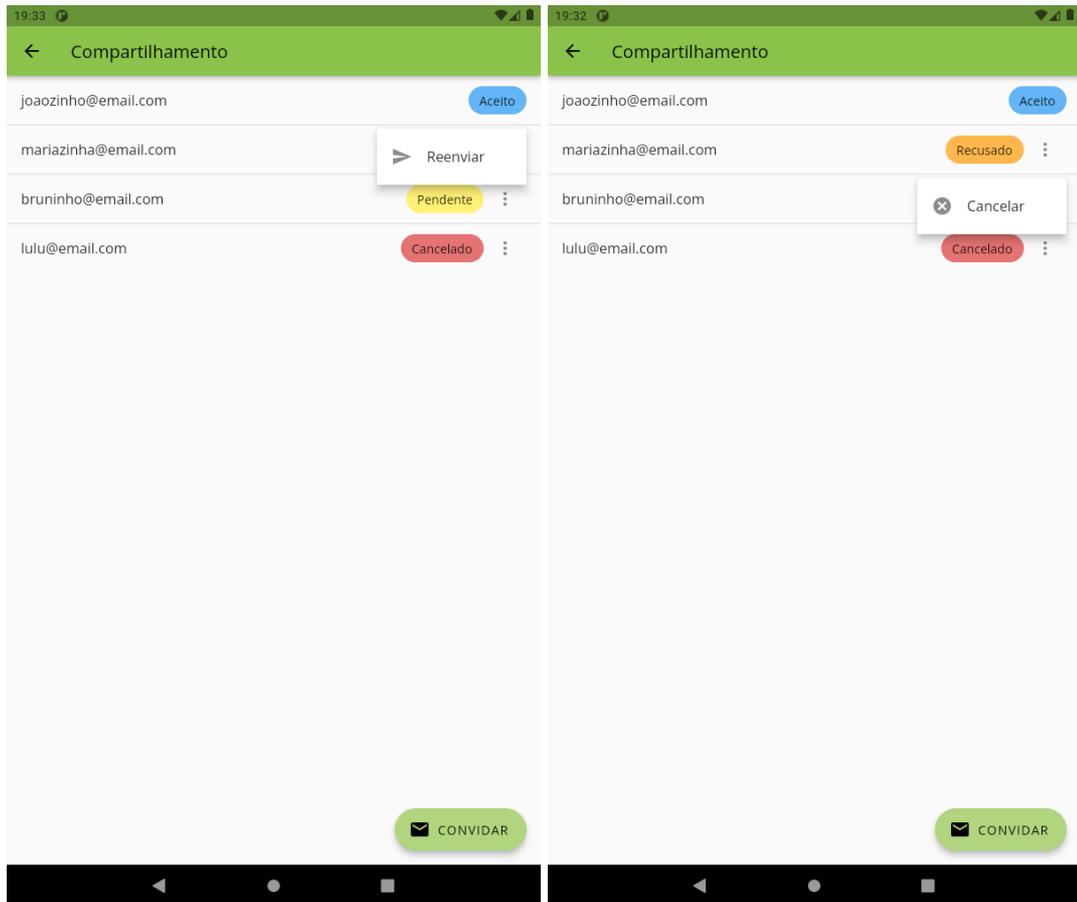


Figura 39: Tela de compartilhamento de questionário do novo Sage, detalhe menus (versão mobile)

Por fim, clicando no botão “Convidar” no canto inferior direito, abre-se um diálogo no qual é possível informar o e-mail do entrevistador para criar um novo convite.

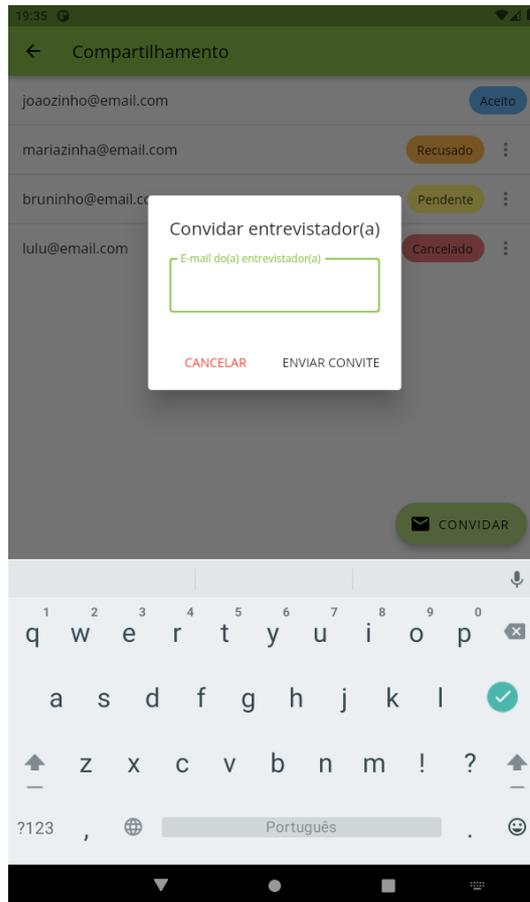


Figura 40: Tela de compartilhamento de questionário do novo Sage, detalhe diálogo novo convite (versão mobile)

Por outro lado, o entrevistador que foi convidado pode visualizar o(s) seu(s) convite(s) a partir da tela inicial, clicando na opção de “Ver convites recebidos”, mostrada na Figura 41.

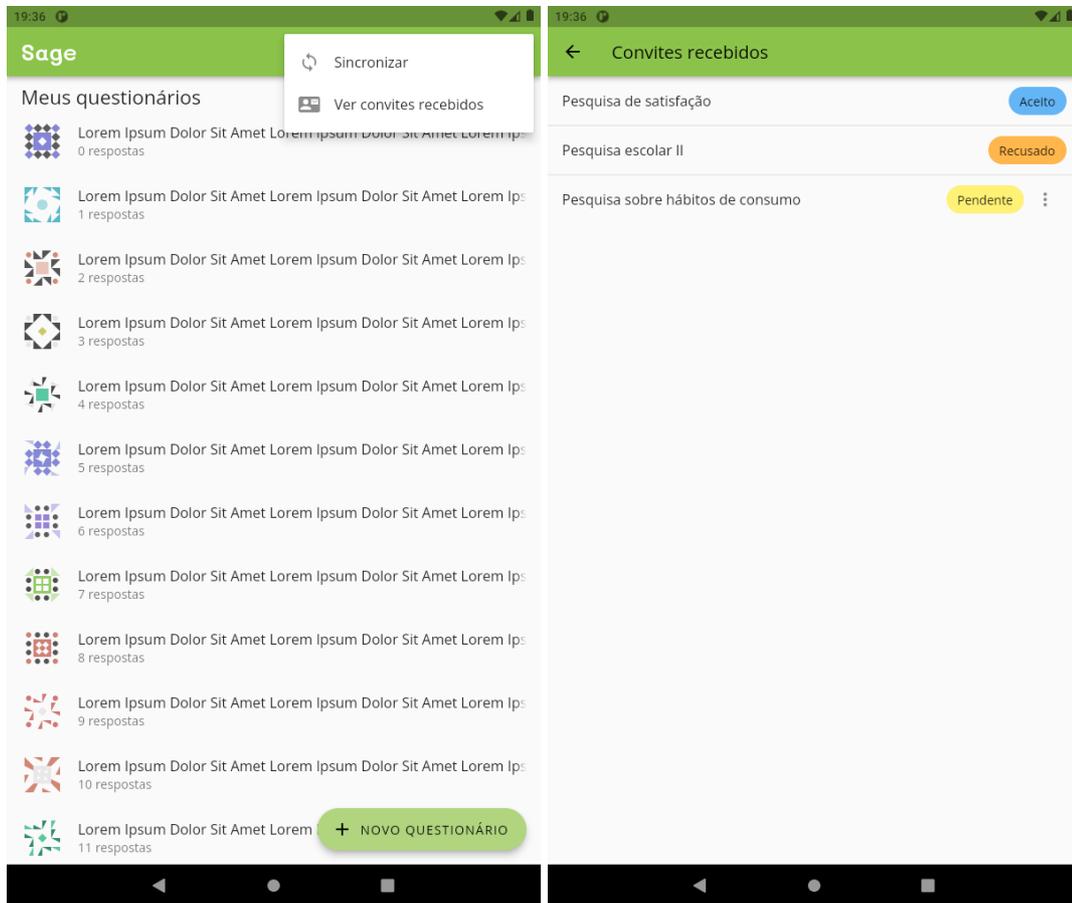


Figura 41: Tela inicial do novo Sage, detalhe menu; e  
Tela de convites recebidos no novo Sage (versão mobile)

Nesta tela, o usuário pode aceitar ou recusar os convites que recebeu, conforme Figura 42.

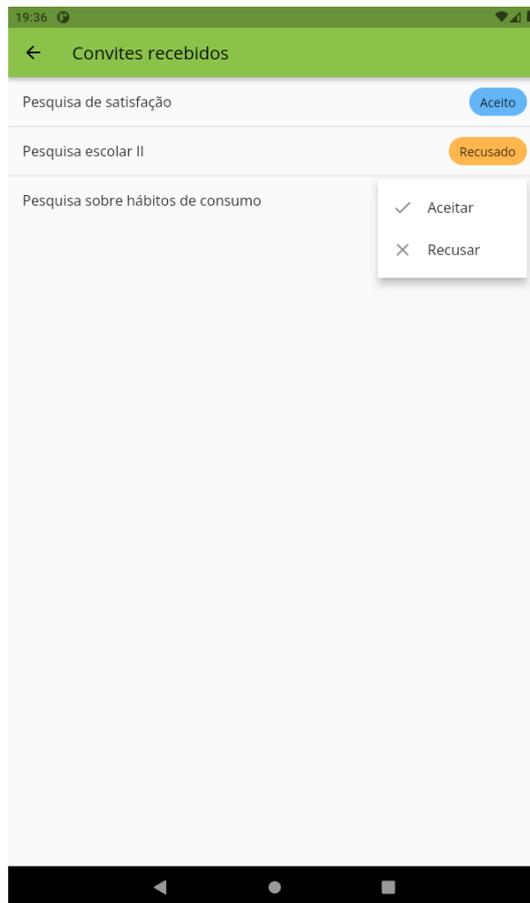


Figura 42: Tela de convites recebidos no novo Sage, detalhe menu (versão mobile)

Através do que foi desenvolvido e apresentado nas seções anteriores, atingiu-se os requisitos 3.1, 3.2 e 3.3.

#### 4.6. Resultados

No quadro 6, é possível observar a comparação do novo Sage com as ferramentas existentes que mais se aproximam da ferramenta proposta, isto é, que satisfazem mais requisitos, são: Google Forms e Jotform, além do Sage original.

Sendo assim, verifica-se que o Novo Sage preenche todos os requisitos listados.

Quadro 6: Comparação do Novo Sage

Nome	Site	Precificação	Layout Adaptado	Offline	Ciframento no armazenamento das respostas	Aplicação web	Tipos de perguntas	Convites entre entrevistadores
Google Forms	<a href="https://www.google.com/forms/about/">https://www.google.com/forms/about/</a>	Gratuito	Sim	Não	N/A	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), data, horário, upload de arquivo	Sim
Jotform	<a href="https://www.jotform.com/">https://www.jotform.com/</a>	Gratuito (5 pesquisas; 100 respostas/mês; 100 MB de armazenamento; 1000 views/mês)	Sim	Sim	Sim	Sim	Texto (diversos), múltipla escolha (diversos), escala Likert (diversos), horário, upload de arquivo	Apenas nos planos para Empresas
Sage		Gratuito	Sim	Sim	Sim	Não	Texto, numérico, múltipla escolha (diversos)	Não
Novo Sage		Gratuito	Sim	Sim	Sim	Sim	Texto, numérico, múltipla escolha (diversos), escala Likert, data/horário, data, horário	Sim

## 5. CONCLUSÃO

A aplicação *Sage* original mostrou-se relevante dentro do que havia sido proposto no TCC de (OLIVEIRA, 2019), porém ainda possuía potencial para novas funcionalidades. Com base em suas recomendações de trabalhos futuros e complementando-se com sugestões próprias, foram elaborados novos requisitos.

Observou-se, através da desk research, que não foi encontrada ferramenta no mercado que preenchesse todos os requisitos listados e, portanto, este trabalho visou desenvolver as funcionalidades em questão.

O desenvolvimento teve início com algumas tarefas necessárias para que se pudesse executar a aplicação, tais como atualização dos pacotes e ferramentas, bem como a correção de erros de compilação.

Em seguida, foi possível portar a aplicação para sua versão web e, então, desenvolver as funcionalidades previstas com disponibilidade em ambas as plataformas (mobile e web). Além do web app, o novo *Sage* conta com novos tipos de perguntas e a possibilidade de enviar convites entre entrevistadores, de forma que pode-se esperar uma maior facilidade na adoção da ferramenta no futuro ao colocá-la em produção, considerando o contexto de pesquisas quantitativas informatizadas no Brasil.

### 5.1. Trabalhos futuros

Como sugestões de pontos para trabalhos futuros que atuem como extensão deste, tem-se:

- Avaliação de usabilidade da aplicação;
- Funcionalidade de ramificação de perguntas;
- Aumento da cobertura de testes automatizados;

- Lançamento da aplicação em ambiente de produção e acompanhamento do seu uso com usuários reais.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABDAL, Alexandre. **Métodos e técnicas de pesquisa em Ciências Sociais: bloco quantitativo**. Sesc São Paulo/CEBRAP São Paulo, 2016. Disponível em: [https://bibliotecavirtual.cebrap.org.br/arquivos/2017\\_E-BOOK%20Sesc-Cebrap\\_%20Metodos%20e%20tecnicas%20em%20CS%20-%20Bloco%20Quantitativo.pdf](https://bibliotecavirtual.cebrap.org.br/arquivos/2017_E-BOOK%20Sesc-Cebrap_%20Metodos%20e%20tecnicas%20em%20CS%20-%20Bloco%20Quantitativo.pdf). Acesso em: 11 set. 2022.

AELA CONTENTS. **Desk Research: O que é e como Efetuar uma Pesquisa Secundária**. [S. l.], 16 nov. 2020. Disponível em: <https://medium.com/aela/desk-research-o-que-%C3%A9-e-como-efetuar-uma-pesquisa-secund%C3%A1ria-784ee064c10b>. Acesso em: 1 dez. 2022.

BARRO, Bruna. **O Que São Frameworks e Quais os Mais Utilizados**. [S. l.], 11 mar. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/frameworks>. Acesso em: 1 dez. 2022.

BUENO, Carlos Eduardo de Oliveira. **Desenvolvimento de um Aplicativo Utilizando o Framework Flutter e Clean Architecture**. Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos Requisitos para obtenção do título de Bacharel em Ciências da Computação. 2021. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/1861/1/TCC%20%20-%20CARLOS.pdf>. Acesso em: 01 set. 2022.

CARDANO, Mario. **Manual de Pesquisa Qualitativa: a contribuição da teoria da argumentação**. Petrópolis, RJ: Vozes, 2017.

COSTA, Adriana de Bastos da. **Fundamentos de Gestão de Projeto: da teoria à prática, como gerenciar projetos de sucesso**. Curitiba: InterSaberes, 2019.

CRUZ, Fábio. **Scrum e Agile em Projetos Guia Completo**. 2.ed. Rio de Janeiro: Brasport, 2018.

DALMORO, Marlon; VIEIRA, Kelmara Mendes. Dilemas na construção de escalas Tipo Likert: o número de itens e a disposição influenciam nos resultados?. *Revista Gestão Organizacional*, [s. l.], 28 jul. 2014. Disponível em: <https://bell.unochapeco.edu.br/revistas/index.php/rgo/article/view/1386>. Acesso em: 22 nov. 2022.

FOGGETTI, Cristiano. **Gestão Ágil de Projetos**. São Paulo: Education do Brasil, 2014.

FOWLER, Martin. **Patterns of Enterprise Application Architecture**. [S. l.]: Addison-Wesley, 2002.

GIRAFFA, Lucia Maria Martins. **(Re)invenção pedagógica? Reflexões acerca do uso de tecnologias digitais na educação**. Porto Alegre: EdiPUCRS, 2012.

JASSE, Ermínio Pita. **Application programming interface - API para integração de dados em agricultura de precisão**. 2017. 45 f. Dissertação (Mestrado em Tecnologias Computacionais para o Agronegócio) - Universidade Tecnológica Federal do Paraná, Medianeira, 2017. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/3420>. Acesso em: 03 set. 2022.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Técnicas de pesquisa**. 5. ed. São Paulo: Atlas, 2003. 310 p.

MUNCK, Luciano; BANSI, Ana Claudia; GALLELI, Bárbara; CELLA-DE-OLIVEIRA, Flavio Augusto. EM BUSCA DA SUSTENTABILIDADE ORGANIZACIONAL: a proposição de um framework. **Revista Alcance**, [S.L.], v. 20, n. 4, p. 460-477, 28 maio 2014. Editora UNIVALI. Disponível em: <https://periodicos.univali.br/index.php/ra/article/view/3430>. Acesso em: 18 set. 2022.

RONACHER, ARMIN. **Flask web development, one drop at a time**. 2018.

SILVA, Werliton Carlos Sousa da. **Aplicações Móveis Nativas com React Native e Firebase: Um Estudo de Caso**. 2018. 72 p. Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luis, 2018. Disponível em: <https://monografias.ufma.br/jspui/bitstream/123456789/3498/1/WERLITON-Silva.pdf>. Acesso em: 03 set. 2022.

UEDA, Daniel Yudi. **Aplicação Android com Firebase para administrar o uso de medicações**. 2020. Trabalho de Conclusão de Curso (Especialização em Tecnologia Java) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2020. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/25926>. Acesso em: 04 set. 2022.

VASCONCELLOS-GUEDES, L.; GUEDES, L. F. A. E-surveys: Vantagens e Limitações dos Questionários Eletrônicos via Internet no Contexto da Pesquisa Científica. In: X SemeAd - Seminário em Administração FEA/USP (São Paulo, Brasil), 2007. Disponível em: [https://www.researchgate.net/profile/Luis-Guedes-5/publication/233852786\\_E-survey\\_s\\_Vantagens\\_e\\_limitacoes\\_dos\\_questionarios\\_eletronicos\\_via\\_internet\\_no\\_contexto\\_da\\_pesquisa\\_cientifica/links/55b635b608ae9289a08aad62/E-surveys-Vantagens-e-limitacoes-dos-questionarios-eletronicos-via-internet-no-contexto-da-pesquisa-cientifica.pdf](https://www.researchgate.net/profile/Luis-Guedes-5/publication/233852786_E-survey_s_Vantagens_e_limitacoes_dos_questionarios_eletronicos_via_internet_no_contexto_da_pesquisa_cientifica/links/55b635b608ae9289a08aad62/E-surveys-Vantagens-e-limitacoes-dos-questionarios-eletronicos-via-internet-no-contexto-da-pesquisa-cientifica.pdf). Acesso em: 11 set. 2022.

## **APÊNDICE A - Artigo**

# Evolução de uma Ferramenta para Coleta de Dados em Pesquisas Quantitativas

Lucas Barzan Demétrio<sup>1</sup>, Raul Sidnei Wazlawick<sup>1</sup>

<sup>1</sup>Universidade Federal de Santa Catarina

**Abstract.** *Questionnaires are frequently used in scientific research worldwide. Its computerization brings benefits over its analog alternative, but it also has limitations, which were the focus of (OLIVEIRA, 2019), which resulted in the creation of the Sage app. Even relevant, the developed application has room for improvement and inclusion of new features. This project therefore aims to implement these changes in order to facilitate the adoption and use of the app in the context of data collection in scientific research. As a result of the work, we have the new Sage with the implementation of the proposed requirements, verified through automated unit tests.*

**Resumo.** *Questionários são frequentemente utilizados em pesquisas científicas mundialmente. Sua informatização traz benefícios em relação à sua alternativa analógica, mas também apresenta limitações, que foram o foco de (OLIVEIRA, 2019), que resultou na criação do app Sage. Mesmo relevante, a aplicação desenvolvida tem espaço para melhorias e inclusão de novas funcionalidades. Este projeto visa, então, implementar estas alterações de forma a facilitar a adoção e utilização do app no contexto de coleta de dados em pesquisas científicas. Como resultado do trabalho, tem-se o novo Sage com a implementação dos requisitos propostos, verificados através de testes unitários automatizados.*

## 1. Introdução

O método científico pode ser definido como o caminho a ser percorrido numa pesquisa para alcançar os resultados da melhor forma possível. Suas técnicas e instrumentos de coleta de dados devem ser adequados ao objeto em estudo e devem permitir alcançar os objetivos, responder às indagações da investigação científica e produzir conhecimento válido (LAKATOS, 2003).

A partir da definição dos objetivos de pesquisa a serem alcançados, deve-se propor o método que mais se adequa à pergunta de pesquisa, ou seja, que seja capaz de respondê-la. Portanto, podem-se ter pesquisas com abordagem essencialmente qualitativa, pesquisa de natureza quantitativa ou, ainda, pode ser feita a utilização de estratégias múltiplas, que abordem o qualitativo e o quantitativo simultaneamente, para aumentar o espectro de visão de interpretação da realidade, explorando diversas técnicas de coleta de dados (LAKATOS, 2003).

Na abordagem quantitativa, o estudo é de caráter experimental e matemático, tem predileção pelo fenômeno extenso e se caracteriza pela descrição empírica e pela medida e análise de variáveis. Tem como pressuposto o cultivo do rigor científico, da objetividade e da neutralidade no ato investigativo. Atua em níveis de realidade

quantificáveis e deve ser utilizado para abarcar grandes aglomerados de dados ou de conjuntos demográficos, classificando-os e tornando-os inteligíveis por meio de variáveis (DEMOS, 1985).

Na pesquisa quantitativa, os instrumentos de coleta de dados são: a observação direta, os questionários, os bancos de dados (como o Sinan<sup>1</sup> – Sistema Nacional de Agravos de Notificação, o Sinasc<sup>2</sup> – Sistemas de Informação dos Nascidos Vivos), entre outros.

Neste contexto, os dados se apresentam como forma de medidas, mensurações ou estatísticas. Sua interação pode ser definida como objetiva e descritiva do fenômeno. Em um estudo quantitativo, a amostra deve ser ampla, estratificada, com apoio de grupos de controle, intencional ou aleatória, dependendo das variáveis necessárias para atingir os objetivos da pesquisa (ABDAL, 2016).

As técnicas e os procedimentos utilizados para a coleta de dados dependerão do tipo de método escolhido para a realização da pesquisa. No caso do presente trabalho desenvolvido, será analisado o tipo de coleta de dados que envolve uma entrevista fechada ou estruturada, que também é denominada de questionário. Na entrevista fechada, todas as questões têm respostas previstas para serem escolhidas (ABDAL, 2016).

Tais questionários são frequentemente usados em pesquisas científicas, porém há diversos problemas associados ao uso dos mesmos quando preenchidos em papel. Dentre eles, a aplicação do questionário em papel requer uma etapa complementar de digitação das respostas em uma planilha eletrônica, o que, além de exigir tempo, está sujeito a erros de digitação, o que pode impactar nos resultados do estudo. Por outro lado, a aplicação do questionário on-line faz o envio dos dados de forma imediata e automática, poupando, assim, boa parte do trabalho e do tempo destinados à digitação por parte do pesquisador. De forma semelhante, o surgimento dos questionários on-line permite que a mesma pesquisa possa ser feita de forma mais ágil e com redução de custos para o pesquisador, uma vez que o valor unitário gasto para a replicação do questionário é próximo de zero (ABDAL, 2016).

O funcionamento de um questionário on-line possui em termos gerais as mesmas características de um questionário de papel, e deverá ser enviado, recebido e preenchido, com o auxílio da internet (ABDAL, 2016).

Dessa forma, a informatização dos questionários se mostra uma alternativa, porém também possui limitações, como indisponibilidade do sinal de internet, problemas de usabilidade e preocupações sobre a segurança dos dados dos participantes (GUILHAM, 2008).

Tais questões foram o foco do Trabalho de Conclusão de Curso denominado "Desenvolvimento de ferramenta para coleta de dados em pesquisas quantitativas utilizando dispositivos móveis", de Caio Pereira Oliveira, que resultou na criação do aplicativo *Sage*. Mesmo relevante, o app desenvolvido por Caio tem espaço para melhorias e inclusão de novas funcionalidades. Este projeto visa, então, implementar

---

1 <https://portalsinan.saude.gov.br/>

2 <https://svs.aids.gov.br/daent/cgiae/sinasc/apresentacao/>

estas alterações de forma a facilitar a adoção e utilização do app no contexto de coleta de dados em pesquisas científicas.

### 1.1. Objetivos

O objetivo geral deste trabalho é evoluir o app Sage com novas funcionalidades e desenvolver uma aplicação web para o sistema, a fim de torná-lo multiplataforma.

Como objetivos específicos, tem-se:

- Extensão do backend e app mobile existentes com as novas funcionalidades:
  - implementação de novos tipos de perguntas;
  - implementação da funcionalidade de convites entre entrevistadores;
- Desenvolvimento de uma aplicação web com as mesmas funcionalidades do app mobile.

### 1.2. Metodologia de pesquisa

Este trabalho almeja atingir os objetivos propostos na seção 1.1 através das seguintes etapas:

- Pesquisa bibliográfica das linguagens/frameworks necessários através da própria documentação e artigos na internet;
- Desenvolvimento das funcionalidades no servidor, na aplicação, e na sua versão web;
- Desenvolvimento de testes unitários automatizados para o servidor.

Quanto às ferramentas necessárias para realizar o desenvolvimento, o aplicativo mobile *Sage*, ponto de partida deste trabalho, foi desenvolvido utilizando Flutter, enquanto o servidor utiliza Flask (framework Python) e integração com Firebase. Essas ferramentas foram mantidas, uma vez que suprem as necessidades deste trabalho. Para a aplicação web, também foi utilizado Flutter, já que ele permite a escrita de interfaces cross-platform e desta forma evita-se demasiado retrabalho.

## 2. Referencial teórico

Nesta seção, são apresentados conceitos essenciais para o entendimento das seções seguintes. Primeiramente, a definição dos tipos de questionários existentes, a escolha do tipo a ser considerado no escopo deste trabalho, bem como um resumo das etapas para a construção de um questionário. Em seguida, são contextualizados termos técnicos que também se fazem necessários: Framework, Flutter, API, Flask, e Firebase. Por fim, encerra-se esta seção com uma apresentação do app Sage existente, anterior à sua evolução tecnológica neste trabalho.

### 2.1. Questionários

A partir do surgimento da Web 2.0, modificou-se a forma como os internautas interagem na produção de conteúdos em sites e redes sociais, o que provocou um aumento de forma significativa na troca de informações entre os usuários. É neste contexto que os questionários de papel ganham um novo formato, onde a coleta de dados agora pode ser feita utilizando-se a internet por meio de formulários e questionários on-line.

Por mais que estudos apontem que a ideia de se conseguir informações dos usuários seja antiga, sua expansão e popularização no meio digital são recentes (VASCONCELLOS-GUEDES, 2007). Com o auxílio da computação em nuvem, que oportuniza o armazenamento virtual de dados e documentos em grande escala, a expansão dos formulários e questionários no formato on-line aumentou consideravelmente.

Em relação aos questionários on-line, pode-se ressaltar três tipos principais, que utilizam a internet em pelo menos uma de suas etapas de aplicação. Abaixo, são exemplificados os principais questionários on-line (ABDAL, 2016).

- **Máscaras e arquivos de bases de dados:** são instrumentos eletrônicos construídos, aplicados e respondidos de forma off-line, onde o respondente recebe o questionário em um editor de texto/planilha, como Word, Excel, Power Point e, após o seu preenchimento, envia os dados por e-mail ou outro serviço de transferência eletrônica de arquivos. Usando a internet, para receber o questionário e para o envio dos dados. Um exemplo seria os questionários do IBGE.
- **Aplicativos para dispositivos móveis:** são programas para celulares, smartphones e tablets cujos questionários também podem ser respondidos de forma off-line usando a rede somente para encaminhar os dados. Um diferencial é que esse tipo de questionário eletrônico apresenta uma interface que facilita o envio automático desses dados (geralmente por meio de um botão), sem a necessidade de transferir os arquivos por e-mail ou por qualquer outra ferramenta externa de transferência de dados. Utilizado, por exemplo, pela Receita Federal brasileira para o preenchimento e o envio das informações relativas à Declaração Anual de Imposto de Renda.
- **Web surveys:** atualmente, há na internet mais de 80 plataformas que oferecem o serviço de construção de questionários on-line (ABDAL, 2016). Sendo que em sua maioria, a disponibilidade de tais serviços é gratuita, mas, difere de algumas opções premium, oferecendo uma quantidade limitada de ferramentas para elaboração do questionário e sua coleta dos dados.

Para os fins de definição de “questionário on-line” no escopo deste trabalho, considerou-se questionários digitais na forma de aplicativo para dispositivos móveis e na forma de web surveys, com foco em aplicação presencial pelo entrevistador (como ocorre, por exemplo, com o Censo demográfico do Brasil).

Em se tratando das etapas que envolvem a aplicação e uso dos questionários, observa-se que é necessário o desenvolvimento do planejamento, elaboração e aplicação como em qualquer instrumento quantitativo de coleta de dados. Na etapa do planejamento de uma pesquisa quantitativa, se verificam importantes pontos a considerar e decisões a tomar que são anteriores à própria construção de um questionário, que independe do seu tipo (on-line ou papel). Ao menos cinco passos teórico-metodológicos são essenciais até que o pesquisador possa decidir pela viabilidade do uso de um questionário on-line.

Será por meio da fase de aplicação do questionário que a opção pelo instrumento on-line passa a ter implicações mais diretas sobre as etapas que se sucederão. Sendo necessária a aplicação de um “piloto” do questionário com um número reduzido de

respondentes a fim de verificar possíveis erros de construção, fluxo e dificuldades de compreensão do questionário.

Após a finalização do período de coleta dos dados, se observa a construção de um banco de dados completo formado por todas as respostas registradas. É comum que cada opção de resposta do questionário tenha um código numérico no banco de dados, facilitando o processamento estatístico. Atualmente, se observa a opção de baixar os dados em um arquivo compatível com programas de análise estatística, como o Excel<sup>3</sup>, SPSS<sup>4</sup>, Matlab<sup>5</sup>, SAS<sup>6</sup>, R<sup>7</sup>, Stata<sup>8</sup>, entre outros.

Com o banco de dados, inicia-se o processo de checagem, crítica e consistência onde as informações são revisadas em busca de possíveis incoerências ou erros de preenchimento. Caso se observe que alguma questão permite múltiplas respostas ou que tenha a possibilidade de citações não definidas previamente no questionário (as chamadas questões abertas), será atribuído um código para cada resposta diferente que aparecer. Essa é a etapa chamada de codificação (ABDAL, 2016).

Por fim, para efeitos de comparação, no caso do questionário de papel, além dessas etapas seria necessária uma etapa de processamento, para inserir as respostas no banco de dados. Portanto, o questionário on-line apresenta um diferencial neste quesito, uma vez que tudo já é registrado automaticamente.

## 2.2. Framework

Um framework pode ser compreendido como um conjunto de códigos que são usados como pacotes para que uma pessoa desenvolvedora possa criar uma aplicação. Logo, os frameworks podem ser vistos como uma forma de auxiliar na construção de um sistema, uma vez que permitem o reuso de código pronto ao invés de ser necessário desenvolvê-lo do zero (BARRO, 2022).

Logo, a principal função de um framework, que leva a sua utilidade, é a facilidade no que diz respeito ao processo de desenvolvimento de um software ou aplicação. Além disso, muitos dos projetos de frameworks são abertos e tendem a passar por testes e processos de otimização.

Isto é, caso alguma falha ou bug seja encontrado, eles podem ser rapidamente reportados e resolvidos pela comunidade. Desta forma, se verifica mais segurança e consistência nos seus padrões.

Assim, o funcionamento de um framework se dá por meio da reutilização dos códigos genéricos oferecidos pelo mesmo, que podem ser aplicados como base para diferentes projetos (BARRO, 2022). Como vantagens proporcionadas pelo amplo emprego dos frameworks nos softwares destacam-se: a otimização do tempo de desenvolvimento, definição e padronização das melhores práticas de programação,

---

3 <https://www.microsoft.com/pt-br/microsoft-365/excel>

4 <https://www.ibm.com/br-pt/spss>

5 <https://www.mathworks.com/products/matlab.html>

6 [https://www.sas.com/pt\\_br/home.html](https://www.sas.com/pt_br/home.html)

7 <https://www.r-project.org/>

8 <https://www.stata.com/>

maior segurança, prevenção de códigos duplicados, diminuição da ocorrência de bugs, maior consistência no processo de desenvolvimento e nas aplicações criadas, redução das chances de erro no código. Por fim, frameworks possibilitam que os desenvolvedores se dediquem aos elementos específicos do projeto e simplificam/encurtam a curva de aprendizado do time de desenvolvimento (BARRO, 2022).

### 2.3. Flutter

Flutter<sup>9</sup> é um framework voltado para o desenvolvimento de interfaces de usuário para aplicativos multiplataforma, como: Android, iOS, Windows, Mac, Linux, Fuchsia e Web. Nele, é utilizada a linguagem de programação Dart que, assim como o framework, foi criada pela Google. O paradigma utilizado em Flutter é o de programação reativa para interfaces, difundido pela biblioteca React que, por sua vez, foi criada pelo Facebook.

De acordo com a documentação oficial, as aplicações desenvolvidas em Flutter, durante a fase de desenvolvimento, são executadas em uma Virtual Machine (VM) que possibilita uma recompilação rápida apenas dos componentes que sofreram mudanças, não necessitando reconstruir toda a aplicação. Entretanto, na etapa de *releasing* (lançamento da aplicação) todo o código é compilado diretamente para código de máquina (Intel x64, ARM ou JavaScript).

O Flutter é construído com C, C++, Dart e Skia (uma motor gráfico de renderização 2D) e possui uma arquitetura de camadas.

Na última camada do diagrama está o Embedder. Cada plataforma possui um Embedder específico que pode ser escrito em Java e C++ para Android, Objective-C/Objective-C++ para iOS/macOS, e C++ para Windows/Linux. O Embedder acaba por ser um provedor de ponto de entrada para a aplicação sendo também responsável por coordenar acesso a serviços do sistema operacional, como: renderização, acessibilidade, entrada de dados e também gerencia o loop de eventos.

Tem-se a Engine ou também chamada de Flutter Engine, é escrita em C++ e tem as primitivas necessárias para suportar todas as aplicações Flutter. Tal mecanismo é responsável por rasterizar os componentes da tela sempre que for preciso atualizar a interface.

No topo do diagrama está o Framework Flutter. Este é o componente utilizado pelos desenvolvedores de software aplicativo, que não atuam no desenvolvimento do framework em si. Os principais componentes do framework apresentam as seguintes funcionalidades:

- **Foundation:** provê abstrações para serem utilizadas por serviços como Animation (animação), Painting (pintura) e Gestures (gestos).
- **Rendering:** prove abstração para trabalhar com layouts. Nesta camada é possível criar uma árvore de componentes renderizados.

---

<sup>9</sup> <https://flutter.dev/>

- **Widgets:** esta camada possibilita criar abstração de composição. Basicamente as classes utilizadas nesta camada serão tratadas como *render objects* na camada de renderização.
- **Material e Cupertino:** esta camada provê um conjunto de widgets pré-definidos de acordo com ambos Material e iOS design systems, e que podem ser utilizados na criação de interfaces.

## 2.4. API

API é um acrônimo de *Application Programming Interface* e corresponde a um conjunto de softwares que tem como função intermediar a comunicação entre as aplicações. Desta forma, uma API acaba por oportunizar, também, que uma ou mais aplicações possam se comunicar de forma transparente com o usuário, como se fosse somente uma. Assim, sua agregação acaba por integrar em nível de dados onde estes são enviados e/ou recebidos entre as aplicações envolvidas no processo (JASSE, 2017).

Logo, uma *Application Programming Interface* (API) é um conjunto de rotinas utilizado para descrever a comunicação entre diferentes componentes de software. São responsáveis pelo controle de acesso a funções de software ou dispositivos de hardware. Existem APIs de sistemas operacionais, de bibliotecas ou pacotes de software, para a Web, dentre outras.

Para uma empresa, uma API é um ponto de conexão entre ela e seus parceiros ou clientes, compartilhando processos de negócio, serviços ou ainda conteúdos, onde as equipes de trabalhos internos podem estar conectadas de maneira fácil e segura (JASSE, 2017).

## 2.5. Flask

O Python<sup>10</sup>, por ser uma linguagem de programação interpretada, pode rodar em qualquer plataforma, seja ela Linux, Windows, ou Mac OS, por exemplo. Isso torna o desenvolvimento de aplicações multiplataformas mais acessível. Sua sintaxe em si é uma de suas melhores características e deixa o código facilmente legível a nível humano. Permite desenvolver de modo imperativo, orientado a objetos ou funcional e possui tipagem dinâmica e forte. Além do mais, possui uma vasta quantidade de módulos/bibliotecas que já são *built-in*, e se bibliotecas de terceiros forem necessárias basta utilizar o *pip*, um módulo *built-in* do Python, para instalá-las. Outro exemplo de uma biblioteca interna é a *multiprocessing*, que é utilizada para se atingir paralelismo nos programas.

Flask<sup>11</sup> é um framework Python para a criação de servidores web. É um dos pacotes mais utilizados do ecossistema Python. Destaca-se por implementar a especificação WSGI (Web Server Gateway Interface), padrão utilizado por frameworks web a fim de facilitar a portabilidade das aplicações em diferentes servidores web.

Segundo Ronacher (2018) Flask é um micro framework para Python baseado no Werkzeug, Jinja2 e boas práticas. É usado para construir e executar um web server, que pode ser gerenciado por URLs customizadas, onde cada uma aponta para uma anotação

---

<sup>10</sup> <https://www.python.org/>

<sup>11</sup> <https://flask.palletsprojects.com/>

no script em Python que executará sua devida função, sendo que cada função possui as definições de uma página ou ações padronizadas, assim como o tratamento de requisições GET e POST.

## 2.6. Firebase

A Firebase<sup>12</sup> é uma plataforma pertencente ao Google para desenvolvimento de aplicações web e móveis, que implementa diversas funcionalidades comuns a estas aplicações. Dentre elas, podemos citar os serviços a seguir oferecidos pelo Firebase: *Auth*, para criação de usuários e autenticação, com suporte a *social login*; *Crashlytics*, para receber informações sobre encerramentos inesperados no app por conta de erro; *Analytics*, para coletar dados de uso do aplicativo; e *Cloud Messaging*, para enviar push notifications para os dispositivos.

Por meio dos estudos desenvolvidos por Silva (2018), pode-se mencionar que o Firebase tem uma estrutura de base de dados que é diferente de muitos bancos de dados comuns. Cada base de dados Firebase é guardada como uma árvore de objeto JSON. Esta estrutura de árvore é muito flexível para todos os tipos de dados.

Assim a plataforma acaba por oferecer serviços e produtos que têm como foco desenvolver e testar aplicativos, objetivando o desenvolvimento de aplicativos com alta qualidade e ótimo tempo de resposta. (SILVA, 2018).

## 2.8. Sage

O aplicativo utilizado como base para desenvolvimento deste trabalho, o *Sage*, se propõe a servir como ferramenta para coleta de dados em pesquisas quantitativas. Ele foi projetado para que o aplicativo em execução num dispositivo móvel (tablet ou smartphone) seja entregue ao participante para que preencha o questionário de forma autônoma, mas o pesquisador também pode optar por ler as perguntas e campos ao participante, registrando suas escolhas por ele.

Ao iniciar o app, é apresentada a tela de login para autenticação com o Google Sign in, provido pelo Firebase.

Após autenticado, o usuário pode utilizar a funcionalidade de cadastro de questionários para criar novos questionários e adicionar perguntas aos mesmos, de acordo com os tipos de perguntas desejados. Os tipos de perguntas disponíveis são: escolha múltipla, escolha única, texto livre e numérica.

Tendo criado um ou mais questionários, é possível respondê-los, avançando pelas perguntas e respeitando as devidas regras de validação de cada pergunta. Ao salvar, as respostas são criptografadas e armazenadas no dispositivo.

Na tela inicial, de listagem de questionários, é possível visualizar os questionários criados e sincronizar com o servidor as respostas armazenadas.

## 3. Trabalhos correlatos

---

<sup>12</sup> <https://firebase.google.com/?hl=pt>

Nesta seção, serão apresentados os requisitos que foram definidos para a evolução tecnológica do Sage, seguidos por uma “desk research” em busca de ferramentas correlatas.

### 3.1. Análise dos requisitos

Baseado nas recomendações de trabalhos futuros apresentados ao final do TCC de (OLIVEIRA, 2019), e complementando-se com sugestões próprias, obteve-se os requisitos a seguir para evolução técnica da ferramenta.

- **1. Versão web da ferramenta**

**Requisito 1.1:** O *web app* deverá permitir acesso às mesmas funcionalidades já presentes no aplicativo móvel.

**Requisito 1.2:** O *web app* deverá permitir o uso das novas funcionalidades (requisitos a seguir) que serão implementadas no aplicativo móvel.

- **2. Adição de novos tipos de perguntas**

**Requisito 2.1:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “escala de Likert”.

**Requisito 2.2:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “data + horário”.

**Requisito 2.3:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “data”.

**Requisito 2.4:** O aplicativo e o *web app* deverão permitir que o usuário adicione perguntas do tipo “horário”.

- **3. Adição de convites entre entrevistadores**

**Requisito 3.1:** O aplicativo e o *web app* deverão permitir que um usuário convide outros usuários para os seus questionários, usando o e-mail cadastrado na conta do entrevistador a ser convidado.

**Requisito 3.2:** O aplicativo e o *web app* deverão permitir que o usuário liste os seus convites recebidos.

**Requisito 3.3:** O aplicativo e o *web app* deverão permitir que o usuário liste os convites que foram enviados em um determinado questionário de sua autoria.

### 3.2. Desk Research

Com o objetivo de identificar a existência de uma ferramenta de questionários semelhante à nova versão do Sage que é sugerida neste trabalho, foi realizado um levantamento das principais ferramentas de questionário existentes atualmente. Neste trabalho, este processo será referido como Desk research, isto é, uma pesquisa secundária baseada em informações, documentos e fontes já existentes (AELA CONTENTS, 2020). Para isto, não seria viável utilizar um motor de busca determinístico como o Scopus<sup>13</sup>, já que este busca apenas por artigos e trabalhos científicos, portanto, as ferramentas foram encontradas através do motor de busca

---

<sup>13</sup> <https://www.scopus.com/>

Google, utilizando-se as strings de busca "survey software", "questionnaire software" e "form software".

Foram consideradas as ferramentas presentes nas 3 primeiras páginas de cada consulta, com exceção de resultados provenientes de anúncios. Apesar do Google personalizar os resultados de acordo com o usuário, este problema foi mitigado realizando-se todas as buscas a partir de uma guia anônima e utilizando um IP fornecido pela universidade. Após realizar a busca, 34 ferramentas foram listadas: Alchemer, BangTheTable, Checkbox, CognitoForms, Creative Research Systems, Forms.io, FormStack, FreeOnlineSurveys, G3plus, Google Forms, IdSurvey, Jotform, KeySurvey, LimeSurvey, MachForm, Opinio, Qualaroo, Qualtrics, QuestionPro, SmartSurvey, SnapSurveys, SoftExpert, Submittable, SurveyLab, SurveyMethods, SurveyMonkey, SurveyPlanet, Survio, Typeform, VisualParadigm, Voxco, WebSurveyCreator, Wufoo, e Zoho.

Após listar as ferramentas, todas que não possuíam planos gratuitos foram excluídas da comparação, visto que seria inviável pagar por todas para testar as funcionalidades. As ferramentas removidas por este motivo foram: Alchemer, BangTheTable, Checkbox, Creative Research Systems, FormStack, G3plus, IdSurvey, KeySurvey, MachForm, Opinio, Qualaroo, Qualtrics, SnapSurveys, SoftExpert, Submittable, SurveyLab, VisualParadigm, e Voxco.

Por fim, também foram removidas ferramentas que não permitiam a exportação de dados (em um formato como .CSV ou .XLSX, por exemplo) no plano gratuito. Foram elas: SmartSurvey, SurveyPlanet, Survio, e Zoho.

A análise das ferramentas restantes foi feita com base tanto nos requisitos provenientes do TCC de (OLIVEIRA, 2019) (precificação, uso offline, e ciframento no armazenamento das respostas) quanto nos requisitos determinados na proposta deste TCC (aplicação web, tipos de perguntas, e convites entre entrevistadores).

A partir do desk research, foi possível observar que as ferramentas existentes que mais se aproximam da ferramenta proposta, isto é, que satisfazem mais requisitos, são: Google Forms e Jotform.

No entanto, o Google Forms não dispõe de uso offline e, por consequência, também não realiza ciframento no armazenamento das respostas, já que elas não são armazenadas no dispositivo.

Por outro lado, o Jotform restringe o uso da funcionalidade de convite entre entrevistadores a usuários dos planos empresariais, ao mesmo tempo em que possui limite do plano gratuito (5 pesquisas; 100 respostas/mês; 100 MB de armazenamento; 1000 views/mês).

Dessa forma, partindo do Sage existente, é possível evoluí-lo tecnicamente para que se satisfaça todos os requisitos listados.

#### **4. Desenvolvimento**

Nesta seção, é apresentada a sequência de passos executados no desenvolvimento deste trabalho a fim de evoluir o Sage de seu estado inicial até o estado proposto na análise de requisitos já exposta. Cada subseção apresenta uma funcionalidade desenvolvida ou agrupa um conjunto de tarefas correlacionadas. Primeiramente, é descrita a atualização

dos pacotes e ferramentas da aplicação, seguida pela correção de erros diversos. Depois, o desenvolvimento da aplicação web, novos tipos de perguntas, e funcionalidade de convites entre entrevistadores, respectivamente.

#### 4.1. Atualização de pacotes e ferramentas

Para executar o servidor existente do Sage localmente, bastou instalar as dependências necessárias e alterar alguns detalhes relacionados à importação dos módulos. Já no caso do app mobile, foi necessário realizar uma série de tarefas antes de ser possível iniciar o desenvolvimento.

Primeiramente, os pacotes utilizados se encontravam em versões muito antigas, o que foi proibitivo por conta da adoção de “Android v2 embedding” no Flutter. Mais informações disponíveis no artigo “Android v1 embedding app and plugin creation deprecation”<sup>14</sup> da documentação oficial do Flutter.

Portanto, foi necessário atualizar as versões dos pacotes, tomando cuidado para utilizar versões compatíveis entre si (já que muitas vezes um pacote depende de uma versão específica de outro pacote e desrespeitar isso pode causar problemas).

No caso de um pacote específico, o moor, o pacote foi renomeado pelos criadores para drift. Mais informações sobre esta mudança podem ser encontradas na documentação oficial do pacote: “Moor and Drift”<sup>15</sup>.

Uma vez que a migração automática sugerida não funcionou, foi necessário realizar a migração manual de um pacote para outro, e isso incluiu:

- Alterar o arquivo pubspec.yaml;
- Alterar as importações;
- Alterar o código, para refletir os novos nomes das entidades disponibilizadas pela API do pacote;
- Alterar o arquivo de configuração build.yaml;
- Renomear os arquivos .moor;

Ao tentar executar o build\_runner, ferramenta que gera alguns arquivos necessários para a aplicação (no caso do Sage, principalmente relacionados ao banco e às rotas), a execução falhou com o erro “Generator cannot target libraries that have not been migrated to null-safety”.

A “sound null safety”, suportada pelo Dart (linguagem de programação utilizada para desenvolver com Flutter), torna os tipos das variáveis non-nullable por padrão, isto é, as variáveis não podem conter nulo a não ser que seja informado o contrário. Por conta da incompatibilidade apontada no erro, foi preciso alterar a versão mínima do SDK do Dart utilizado pela aplicação para 2.12.0. A partir dessa versão do Dart, sound null safety é suportado.

Também fez-se necessário a atualização da versão do Gradle (ferramenta utilizada no processo de build do app Android), e do Kotlin Gradle plugin, assim como

---

<sup>14</sup> <https://docs.flutter.dev/release/breaking-changes/android-v1-embedding-create-deprecation>

<sup>15</sup> <https://drift.simonbinder.eu/name/>

a versão esperada do SDK do Android nos dispositivos em que o app for instalado (por conta de restrições dos pacotes utilizados).

## **4.2. Correção de erros**

Ao tentar compilar e executar o aplicativo, foram identificados 289 problemas no código, dos quais 211 eram erros impeditivos, e 78 eram warnings com sugestões de melhoria.

Como não existe forma automática de corrigir estes erros, foi preciso passar pelos arquivos e avaliar cada caso. Havia alguns erros de tipagem, outros por conta de breaking changes nas bibliotecas utilizadas, dentre outros. Uma vez corrigidos, foi possível rodar a aplicação.

## **4.3. Aplicação web**

A aplicação existente foi portada para sua versão web, de forma que as funcionalidades seguintes pudessem ser desenvolvidas e validadas em ambas as plataformas. Para adicionar suporte web a um app existente, conforme a documentação<sup>16</sup> do Flutter, é preciso rodar o comando “flutter create .”. Isso adiciona alguns arquivos necessários para a execução da página web.

No contexto do Sage, também foi preciso realizar algumas adaptações em relação à biblioteca utilizada para o banco de dados, mais especificamente a inicialização do módulo correto de acordo com o ambiente em que a aplicação está sendo executada (mobile ou web).

Com isto, atingiu-se o requisito 1.1.

## **4.4. Novos tipos de perguntas**

### **4.4.1. Escala Likert**

Introduzida por Rensis Likert em 1932, a escala de Likert é uma escala de mensuração de cinco pontos amplamente utilizada como instrumento de medida, em especial na área das ciências sociais. É possível fazer uso de escalas com um número de itens diferente de cinco, porém esta não se configura uma escala Likert, mas sim do “tipo Likert”. De acordo com Dalmoro (2014), a escala de três pontos é menos confiável e tem menor capacidade de demonstrar com precisão a opinião do entrevistado. Além disso, a escala de cinco pontos teve, em média, a mesma precisão e mostrou-se mais fácil e mais rápida que a escala de sete pontos. Portanto, neste trabalho, optou-se por desenvolver este tipo de pergunta seguindo o formato padrão de cinco pontos.

Para suportar a escala Likert do lado do servidor, foi preciso declarar o novo tipo de pergunta juntamente com os demais.

Os objetos de perguntas recebidos pelo servidor do Sage são validados de acordo com uma determinada estrutura fixa (ou schema), que é definida no servidor.. Esse schema é definido com auxílio da biblioteca “schema”, comumente utilizada para validar estruturas de dados em Python.

---

<sup>16</sup> <https://docs.flutter.dev/get-started/web#add-web-support-to-an-existing-app>

A documentação<sup>17</sup> da biblioteca apresenta como diversas estruturas complexas podem ser modeladas. Apenas a título de exemplificação, uma pergunta deve possuir um atributo “title” do tipo str (cadeia de caracteres), e um atributo “required” do tipo bool (booleano). Além disso, uma pergunta pode possuir, opcionalmente, um atributo “description” do tipo str que, caso não informado, recebe uma string vazia como valor padrão.

Essa estrutura se mostrou suficiente para diversos tipos de perguntas graças ao seu atributo “extras”, que recebe as restrições específicas de cada tipo de pergunta (caso haja). Por exemplo, perguntas do tipo “numérica”, já suportadas pelo Sage, tem seus “extras” definidos por um determinado schema. Em alto nível, perguntas do tipo “numérica” podem receber dois inteiros: um que servirá como mínimo e outro que servirá como máximo, de forma que só serão aceitas respostas dentro desse intervalo. Analogamente, neste trabalho, estruturou-se o schema para os extras das perguntas do tipo escala Likert.

Conforme o schema, para este tipo de pergunta espera-se que seja informada uma lista de strings que correspondem aos rótulos das 5 opções da escala.

Uma vez implementados os aspectos supracitados no lado do servidor, uma das alterações desenvolvidas do lado do aplicativo foi a implementação do componente usado na criação de uma pergunta deste novo tipo. É através deste componente que o usuário define, pela interface, os rótulos para os pontos da escala.

O componente consiste em 5 caixas de texto (TextFormField) organizadas verticalmente numa coluna (Column). A cada caixa de texto foi atribuído um rótulo para identificar sua finalidade (por exemplo, “Rótulo para pontuação = 1”). Dentre as demais propriedades, é possível citar também a passagem de uma função que valida se o valor digitado é válido; caso este não seja, a caixa de texto é exibida num estado de erro. Por padrão, as caixas de texto são preenchidas com estes valores, respectivamente: “Muito insatisfeito”, “Insatisfeito”, “Indiferente”, “Satisfeito”, “Muito satisfeito”.

Outro ponto implementado foi o componente usado pelo usuário para responder uma pergunta deste tipo. Para isso, 5 botões de opção (Radio) foram dispostos horizontalmente numa linha (Row). Para cada botão de opção foi atribuído um valor interno entre “1” e “5”, bem como o seu respectivo rótulo (exibido abaixo do botão), obtido a partir dos extras da pergunta.

Com isto, atingiu-se o requisito 2.1.

#### **4.4.2. Data/horário**

O novo tipo Data/horário consiste em uma forma de coletar a data e horário de ocorrência de um evento; em outras palavras, uma espécie de timestamp. Em alguns domínios, é importante registrar tanto data quanto horário para maior precisão.

O processo de desenvolvimento deste tipo de pergunta é análogo ao da escala Likert, uma vez que também foi necessário registrá-lo juntamente com os demais tipos no servidor e modelar os seus “extras” adequadamente.

---

<sup>17</sup> <https://pypi.org/project/schema/>

Para este tipo de pergunta, os extras consistiram em uma data/horário mínimo e uma data/horário máximo. Isto porque, por exemplo, para a pergunta “Quando você prefere que seja realizado o evento de confraternização da empresa?”, pode ser interessante restringir o conjunto de respostas a datas no futuro dentro de um determinado intervalo que faça sentido no contexto.

Ao clicar nas caixas de texto, tanto na tela de criação da pergunta quanto na tela de resposta, é aberto um diálogo que permite a seleção da data e, em sequência, a seleção do horário. Ao finalizar, o resultado é salvo e exibido na caixa de texto com a formatação adequada. Para exibição do diálogo de seleção da data e horário, utilizou-se a biblioteca *date\_time\_picker* disponível para Flutter, que provê um widget para esse fim.

É possível passar alguns parâmetros para esse widget de forma a customizá-lo, dentre eles:

- **type:** o tipo de diálogo a ser apresentado (apenas horário, apenas data, ou ambos data e horário em sequência);
- **dateMask:** máscara usada para formatar o valor selecionado (exemplo: ‘dd/MM/yyyy HH:mm’);
- **initialValue:** valor inicial que será pré-selecionado;
- **firstDate:** valor mínimo permitido;
- **lastDate:** valor máximo permitido.

Com isto, atingiu-se o requisito 2.2.

#### 4.4.3. Data

O novo tipo Data é uma espécie de subtipo do tipo Data/horário. Ele se mostra mais adequado quando o horário de ocorrência do evento não é relevante. Por exemplo, para a pergunta “Quando foi a última Páscoa?” é suficiente saber apenas a data (dia/mês/ano), sem que seja necessário o horário.

Para desenvolver este tipo de pergunta, o processo é análogo aos demais. Os extras foram modelados com uma data mínima permitida e uma data máxima permitida, seguindo o mesmo princípio do tipo de pergunta anterior.

Para exibição do diálogo de seleção de data, utilizou-se a mesma biblioteca já citada anteriormente, com a diferença do parâmetro de tipo passado (*DateTimePickerType.date* ao invés de *DateTimePickerType.dateTime*).

Com isto, atingiu-se o requisito 2.3.

#### 4.4.4. Horário

Por fim, o novo tipo Horário se mostra mais adequado quando a data não é importante. Por exemplo, para as perguntas “Em qual horário você costuma acordar diariamente?” ou “Em qual horário você prefere sair do trabalho?”, apenas o horário pode ser suficiente como resposta.

Análogo aos dois tipos anteriores, ao clicar na caixa de texto, é aberto um diálogo de seleção de horário que foi feito com a mesma biblioteca, alterando-se o parâmetro de tipo utilizado (*DateTimePickerType.time*).

Com isto, atingiu-se o requisito 2.4.

## 4.5. Convites entre entrevistadores

A funcionalidade de convite entre entrevistadores visa permitir que um entrevistador crie um questionário e compartilhe-o com outros entrevistadores, a fim de que eles apliquem o mesmo questionário, reaproveitando o conjunto de perguntas.

### 4.5.1. Entidade de Convite

Para esta funcionalidade, foi necessário modelar uma entidade de Convite (Invite), de forma que o diagrama Entidade-Relacionamento original do banco de dados do servidor foi atualizado.

À entidade de Convite, atribuiu-se as seguintes propriedades:

- **email:** e-mail do entrevistador que receberá o convite;
- **survey:** o questionário para o qual o entrevistador será convidado;
- **status:** o status do convite, dentre as opções definidas no enumerador InviteStatus (pendente, aceito, recusado ou cancelado);
- **uuid:** um identificador único para o convite;
- **created\_at e updated\_at:** timestamps para controle do momento em que os registros foram criados/atualizados.

Destas, utilizou-se email e survey como chave primária composta para a tabela de convites, uma vez que um entrevistador não pode receber mais de um convite por questionário, e os dois atributos em conjunto são uma forma de identificar um convite.

Prosseguindo, ORM (Object Relational Mapper) ou, em português, mapeamento objeto-relacional, é uma técnica que aproxima o paradigma de desenvolvimento de aplicações orientadas a objetos ao paradigma do banco de dados relacional. O uso desta técnica é realizado através de um mapeador objeto-relacional que geralmente é uma biblioteca ou framework que ajuda no mapeamento e uso do banco de dados (FOWLER, 2002).

No servidor do Sage, a biblioteca Python que já estava sendo utilizada como ORM era a PonyORM<sup>18</sup>. Com auxílio dela, iniciou-se o desenvolvimento da funcionalidade de convite entre entrevistadores por meio da definição da nova entidade apresentada.

As transições válidas entre os estados de um convite foram identificadas e dispostas num diagrama.

### 4.5.2. Lógica de negócio (business logic)

Em seguida, foram desenvolvidas as funções responsáveis pela lógica de negócio relacionada aos convites. Por exemplo, a função create\_invite recebe os dados necessários para criar um convite e faz algumas verificações antes de criá-lo de fato: é preciso checar se o questionário informado existe e se o usuário fazendo a requisição é dono deste questionário (para que possa convidar outros entrevistadores).

De maneira análoga, outras funções desenvolvidas foram:

---

<sup>18</sup> <https://ponyorm.org/>

- **cancel\_invite:** usada para cancelar um convite que foi enviado pelo usuário em questão; checa se o usuário é dono do questionário;
- **accept\_invite:** usada para aceitar um convite recebido pelo usuário em questão; checa se o usuário recebeu o convite em questão e se o convite ainda está pendente;
- **refuse\_invite:** usada para recusar um convite recebido pelo usuário em questão; checa se o usuário recebeu o convite em questão e se o convite ainda está pendente;
- **list\_sent\_invites:** usada para listar convites enviados pelo usuário em questão;
- **list\_received\_invites:** usada para listar convites recebidos pelo usuário em questão;

### 4.5.3. Rotas

Para que as aplicações (web e mobile) pudessem acessar no servidor os recursos relacionados aos convites, foi preciso definir as rotas a serem disponibilizadas. Foram identificadas as rotas acompanhadas de sua finalidade, método HTTP esperado, endpoint, e corpo da requisição esperado.

A título de exemplo, a rota para cancelar convite:

- Recebe na própria rota o parâmetro `inviteUuid` (uma string), que é o `uuid` (identificador único) do convite em questão;
- Responde apenas ao método HTTP “POST”;
- Através da annotation “`@auth_required`” já definida no servidor, exige autenticação através de token (para identificar o usuário que fez a chamada);
- Obtém o convite em questão no banco de dados, cancela-o usando a função de negócio `cancel_invite`, converte a resposta para o formato JSON e a retorna.

As demais rotas seguem uma estrutura semelhante.

### 4.5.4. Testes unitários

A fim de garantir um correto funcionamento das novas rotas relacionadas a convites, foram desenvolvidos testes unitários que simulam requisições para as rotas e verificam se os resultados recebidos estão conforme esperado.

Por exemplo, no teste da rota usada para aceitar um convite, primeiramente são criados os registros necessários para o teste (o usuário dono do questionário, o questionário, o usuário a ser convidado e o convite), e então é chamada a rota. Neste caso, considerando que o retorno da rota é o convite atualizado, espera-se I) que ele seja o convite em questão (isto é, que o `uuid` corresponda), e II) que ele tenha recebido o status de “Aceito”.

Os demais testes seguem uma estrutura semelhante. As novas rotas possuem cobertura de 100%. No entanto, as rotas existentes não possuíam testes automatizados.

### 4.5.5. Aplicação

A seguir, são descritas algumas alterações que foram desenvolvidas para suportar esta nova funcionalidade no lado da aplicação. Na tela de detalhe de questionário, foi adicionada ao menu a opção de “Compartilhar” questionário. A exemplo, as próximas

capturas de tela são do aplicativo móvel, porém a funcionalidade também está disponível na aplicação web e o layout é análogo.

Ao clicar nela, o usuário é redirecionado para uma nova tela, onde são listados os convites para este questionário que já foram enviados e seus respectivos status.

Nesta lista, convites pendentes podem ser cancelados, e convites recusados ou cancelados podem ser reenviados.

Por fim, clicando no botão “Convidar” no canto inferior direito, abre-se um diálogo no qual é possível informar o e-mail do entrevistador para criar um novo convite.

Por outro lado, o entrevistador que foi convidado pode visualizar o(s) seu(s) convite(s) a partir da tela inicial, clicando na opção de “Ver convites recebidos”.

Nesta tela, o usuário pode aceitar ou recusar os convites que recebeu.

Através do que foi desenvolvido e apresentado nas seções anteriores, atingiu-se os requisitos 3.1, 3.2 e 3.3.

#### **4.6. Resultados**

Foi possível observar a comparação do novo Sage com as ferramentas existentes que mais se aproximam da ferramenta proposta, isto é, que satisfazem mais requisitos, são: Google Forms e Jotform, além do Sage original.

Sendo assim, verifica-se que o Novo Sage preenche todos os requisitos listados.

#### **5. Conclusão**

A aplicação Sage original mostrou-se relevante dentro do que havia sido proposto no TCC de (OLIVEIRA, 2019), porém ainda possuía potencial para novas funcionalidades. Com base em suas recomendações de trabalhos futuros e complementando-se com sugestões próprias, foram elaborados novos requisitos.

Observou-se, através da desk research, que não foi encontrada ferramenta no mercado que preenchesse todos os requisitos listados e, portanto, este trabalho visou desenvolver as funcionalidades em questão.

O desenvolvimento teve início com algumas tarefas necessárias para que se pudesse executar a aplicação, tais como atualização dos pacotes e ferramentas, bem como a correção de erros de compilação.

Em seguida, foi possível portar a aplicação para sua versão web e, então, desenvolver as funcionalidades previstas com disponibilidade em ambas as plataformas (mobile e web). Além do web app, o novo Sage conta com novos tipos de perguntas e a possibilidade de enviar convites entre entrevistadores, de forma que pode-se esperar uma maior facilidade na adoção da ferramenta no futuro ao colocá-la em produção, considerando o contexto de pesquisas quantitativas informatizadas no Brasil.

##### **5.1. Trabalhos futuros**

Como sugestões de pontos para trabalhos futuros que atuem como extensão deste, tem-se:

- Avaliação de usabilidade da aplicação;
- Funcionalidade de ramificação de perguntas;
- Aumento da cobertura de testes automatizados;
- Lançamento da aplicação em ambiente de produção e acompanhamento do seu uso com usuários reais.

## Referências

- ABDAL, Alexandre. Métodos e técnicas de pesquisa em Ciências Sociais: bloco quantitativo. Sesc São Paulo/CEBRAP São Paulo, 2016. Disponível em: [https://bibliotecavirtual.cebrap.org.br/arquivos/2017\\_E-BOOK%20Sesc-Cebrap\\_%20Metodos%20e%20tecnicas%20em%20CS%20-%20Bloco%20Quantitativo.pdf](https://bibliotecavirtual.cebrap.org.br/arquivos/2017_E-BOOK%20Sesc-Cebrap_%20Metodos%20e%20tecnicas%20em%20CS%20-%20Bloco%20Quantitativo.pdf). Acesso em: 11 set. 2022.
- AELA CONTENTS. Desk Research: O que é e como Efetuar uma Pesquisa Secundária. [S. l.], 16 nov. 2020. Disponível em: <https://medium.com/aela/desk-research-o-que-%C3%A9-e-como-efetuar-uma-pesqui-sa-secund%C3%A1ria-784ee064c10b>. Acesso em: 1 dez. 2022.
- BARRO, Bruna. O Que São Frameworks e Quais os Mais Utilizados. [S. l.], 11 mar. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/frameworks>. Acesso em: 1 dez. 2022.
- BUENO, Carlos Eduardo de Oliveira. Desenvolvimento de um Aplicativo Utilizando o Framework Flutter e Clean Architecture. Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos Requisitos para obtenção do título de Bacharel em Ciências da Computação. 2021. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/1861/1/TCC%20%20-%20CARLOS.pdf>. Acesso em: 01 set. 2022.
- CARDANO, Mario. Manual de Pesquisa Qualitativa: a contribuição da teoria da argumentação. Petrópolis, RJ: Vozes, 2017.
- COSTA, Adriana de Bastos da. Fundamentos de Gestão de Projeto: da teoria à prática, como gerenciar projetos de sucesso. Curitiba: InterSaberes, 2019.
- CRUZ, Fábio. Scrum e Agile em Projetos Guia Completo. 2.ed. Rio de Janeiro: Brasport, 2018.
- DALMORO, Marlon; VIEIRA, Kelmara Mendes. Dilemas na construção de escalas Tipo Likert: o número de itens e a disposição influenciam nos resultados?. Revista Gestão Organizacional, [s. l.], 28 jul. 2014. Disponível em: <https://bell.unochapeco.edu.br/revistas/index.php/rgo/article/view/1386>. Acesso em: 22 nov. 2022.
- FOGGETTI, Cristiano. Gestão Ágil de Projetos. São Paulo: Education do Brasil, 2014.
- FOWLER, Martin. Patterns of Enterprise Application Architecture. [S. l.]: Addison-Wesley, 2002.
- GIRAFFA, Lucia Maria Martins. (Re)invenção pedagógica? Reflexões acerca do uso de tecnologias digitais na educação. Porto Alegre: EdiPUCRS, 2012.

- JASSE, Ermínio Pita. Application programming interface - API para integração de dados em agricultura de precisão. 2017. 45 f. Dissertação (Mestrado em Tecnologias Computacionais para o Agronegócio) - Universidade Tecnológica Federal do Paraná, Medianeira, 2017. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/3420>. Acesso em: 03 set. 2022.
- LAKATOS, Eva Maria; MARCONI, Marina de Andrade. Técnicas de pesquisa. 5. ed. São Paulo: Atlas, 2003. 310 p.
- MUNCK, Luciano; BANSI, Ana Claudia; GALLELI, Bárbara; CELLA-DE-OLIVEIRA, Flavio Augusto. EM BUSCA DA SUSTENTABILIDADE ORGANIZACIONAL: a proposição de um framework. Revista Alcance, [S.L.], v. 20, n. 4, p. 460-477, 28 maio 2014. Editora UNIVALI. Disponível em: <https://periodicos.univali.br/index.php/ra/article/view/3430>. Acesso em: 18 set. 2022.
- RONACHER, ARMIN. Flask web development, one drop at a time. 2018.
- SILVA, Werliton Carlos Sousa da. Aplicações Móveis Nativas com React Native e Firebase: Um Estudo de Caso. 2018. 72 p. Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luis, 2018. Disponível em: <https://monografias.ufma.br/jspui/bitstream/123456789/3498/1/WERLITON-Silva.pdf>. Acesso em: 03 set. 2022.
- UEDA, Daniel Yudi. Aplicação Android com Firebase para administrar o uso de medicações. 2020. Trabalho de Conclusão de Curso (Especialização em Tecnologia Java) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2020. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/25926>. Acesso em: 04 set. 2022.
- VASCONCELLOS-GUEDES, L.; GUEDES, L. F. A. E-surveys: Vantagens e Limitações dos Questionários Eletrônicos via Internet no Contexto da Pesquisa Científica. In: X SemeAd - Seminário em Administração FEA/USP (São Paulo, Brasil), 2007. Disponível em: [https://www.researchgate.net/profile/Luis-Guedes-5/publication/233852786\\_E-surveys\\_Vantagens\\_e\\_limitacoes\\_dos\\_questionarios\\_eletronicos\\_via\\_internet\\_no\\_contexto\\_da\\_pesquisa\\_cientifica/links/55b635b608ae9289a08aad62/E-surveys-Vantagens-e-limitacoes-dos-questionarios-eletronicos-via-internet-no-contexto-da-pesquisa-cientifica.pdf](https://www.researchgate.net/profile/Luis-Guedes-5/publication/233852786_E-surveys_Vantagens_e_limitacoes_dos_questionarios_eletronicos_via_internet_no_contexto_da_pesquisa_cientifica/links/55b635b608ae9289a08aad62/E-surveys-Vantagens-e-limitacoes-dos-questionarios-eletronicos-via-internet-no-contexto-da-pesquisa-cientifica.pdf). Acesso em: 11 set. 2022.

## **APÊNDICE B - Código-fonte**

O código-fonte pode ser encontrado no repositório “novo-sage-2022”, hospedado no GitLab da UFSC em:

<https://codigos.ufsc.br/lucas.barzan/novo-sage-2022>