



UNIVERSIDADE D
COIMBRA

Flávia Daniela Passos Cardoso Costa

**DESENVOLVIMENTO DE PLATAFORMA DE GESTÃO DE
QUALIDADE**

Relatório de estágio no âmbito do Mestrado em Engenharia Informática, especialização em Sistemas de Informação orientado pelo Professor Doutor Paulo Rupino da Cunha e apresentado à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Junho de 2020

Está página é intencionalmente deixada em branco.

Agradecimentos

Agradeço a Deus por me permitir viver este momento.

Quero agradecer à minha mãe e aos meus irmãos, Carolina, José e Joaquim por me terem acompanhado ao longo de todo o meu percurso académico.

O meu obrigada aos meus amigos, por me terem apoiado e dado uma palavra de incentivo sempre que precisei.

Agradeço ao meu gatinho, pela companhia nos dias de estudo intensivo.

Agradeço ao meu orientador, professor Dr. Paulo Rupino, pelo apoio e dedicação e também aos meus orientadores da Talkdesk, em particular ao Daniel Afonso por toda a ajuda que me deu e pelos ensinamentos transmitidos.

Agradeço também a quem me deu a oportunidade de estagiar na Talkdesk.

A todos, o meu muito obrigada!

Está página é intencionalmente deixada em branco.

Abstract

Over the years there has been a growing use of *call centers*, so it is important to control service quality to determine the most positive points and which ones need to be improved.

Talkdesk develops systems to improve the customer experience and increase the organizational efficiency. These solutions integrate a set of services, such as: Communication recordings, for quality assurance or speech analysis, quality management to determine how well agents adhere to internal procedures, observation of the agent desktop, to control everything that happens in the work space, text analysis to extract information from unstructured data and performance management to align the call center activities with the company objectives.

This internship aims to develop a tool for evaluating interactions between customers and *call center* agents and assists in quality management. The platform allows to create different models of personalized forms with a variety of questions, according to the interaction and the evaluation that is intended to be carried out. These questionnaires are created by the *call center* administrator and then filled out by a supervisor, who listens to the phone calls recordings and assigns an assessment. The supervisor is able to identify the agent, evaluate how the call started, verify if the company policies were respected, determine if the agent was able to understand customer needs, identify methods to solve the problem and check if the customer was satisfied with the service.

Keywords

Call Centers, Quality Management, Customer service, Optimization, Evaluation, Form Builder, Forms.

Está página é intencionalmente deixada em branco.

Resumo

Ao longo dos anos, tem havido um acentuado crescimento da utilização de *call centers*, pelo que é importante controlar a qualidade do serviço de atendimento para determinar os pontos mais positivos e quais os que precisam de ser melhorados.

A Talkdesk desenvolve sistemas para melhorar a experiência dos clientes e aumentar a eficiência organizacional. Estas soluções integram um conjunto de serviços, tais como: Gravações das comunicações, para garantia de qualidade ou análise da fala, gestão da qualidade para determinar quão bem os agentes aderem às políticas e procedimentos internos, observação do *desktop* do agente, para monitorizar tudo o que acontece na área de trabalho, análise de texto para extrair informações de dados não estruturados e gestão de desempenho para alinhar as atividades do *call center* com os objetivos da empresa.

Este estágio teve como objetivo desenvolver uma ferramenta para avaliar as interações entre clientes e agentes de *call centers* e auxiliar na gestão da qualidade. A plataforma permite criar diferentes modelos de formulários personalizados com uma variedade de questões, de acordo com a interação e a avaliação que se pretende realizar. Esses questionários são criados pelo administrador da central de atendimento e preenchidos por um supervisor, que ouve as gravações das chamadas telefónicas e atribui uma avaliação. O supervisor pode identificar o agente, avaliar como a chamada foi iniciada, verificar se as políticas da empresa foram respeitadas, determinar se o agente foi capaz de entender as necessidades do cliente, identificar os métodos para resolver o problema e verificar se o cliente ficou satisfeito com o serviço.

Palavras-Chave

Call Centers, Gestão da qualidade, Atendimento ao cliente, Otimização, Avaliação, Construtor de formulários, Formulários.

Está página é intencionalmente deixada em branco.

Conteúdo

1	Introdução	1
1.1	Local de estágio	1
1.2	Enquadramento e Objetivos	2
1.3	Motivação	3
1.4	Estrutura do documento	4
2	Estado de arte	6
2.1	Construtores de formulários	6
2.2	Soluções	7
2.2.1	Typeform	7
2.2.2	Google Forms	12
2.2.3	SurveyMonkey	14
2.2.4	ZohoSurvey	18
2.2.5	SoGoSurvey	22
2.3	Vetores de comparação	24
2.4	Análise comparativa das soluções	25
3	Abordagem de Desenvolvimento	27
3.1	Equipa do projeto	27
3.2	Ciclo de vida	27
3.3	Controlo de versões de <i>software</i>	29
3.4	Plano de estágio	30
3.4.1	Primeiro semestre	30
3.4.2	Segundo semestre	32
3.5	Análise de Riscos	34
3.5.1	Lista de riscos	35
4	Requisitos	37
4.1	Levantamento de requisitos	37
4.2	Requisitos funcionais	37
4.3	Requisitos Não funcionais e restrições técnicas	39
4.3.1	Requisitos do produto	39
4.3.2	Requisitos externos	40
4.3.3	Restrições técnicas	40
4.4	Diagramas de casos-de-uso	40
4.5	Especificação de requisitos	42
5	Arquitetura e Modelo de dados	44
5.1	Arquitetura global da aplicação	44
5.2	Arquitetura da aplicação na cloud	46
5.3	Modelo de dados	47

6	Tecnologias e ferramentas	55
7	Qualidade do código	59
7.1	Github	59
7.1.1	Criação de <i>branches</i>	59
7.1.2	Criação de <i>pull requests</i>	60
7.2	Git Hooks	62
7.3	Análise estática do código	64
7.4	Cobertura do código	64
7.5	Github Actions: Integração contínua	66
8	Testes	68
8.1	Testes Unitários	68
8.2	Testes de Sistema	70
9	Conclusões	72
A	Requisitos	77
A.1	Diagramas de casos de uso	77
A.2	Especificação de requisitos	79
B	Manual de Utilizador	103
B.1	Introdução	103
B.2	Efetuar login	103
B.2.1	Página de Login	103
B.2.2	Login - Dados incorretos	104
B.3	Página inicial do Administrador	104
B.4	Página inicial do Supervisor	105
B.5	Efetuar logout	105
B.6	Criar novo formulário - Administrador	106
B.6.1	Criar formulário - Validações	108
B.7	Página para criar formulário	109
B.8	Adicionar elementos ao formulário	110
B.8.1	New section	110
B.8.2	Short Text	110
B.8.3	Long Text	111
B.8.4	Full Name	112
B.8.5	Date Picker	112
B.8.6	Time	113
B.8.7	Number	113
B.8.8	Opinion Scale	114
B.8.9	Favorite Rating	114
B.8.10	Star Rating	115
B.8.11	Slider Scale	115
B.8.12	NPS	116
B.8.13	Single Choice	116
B.8.14	Multiple Choice	118
B.9	Eliminar elementos do formulário	121
B.10	Duplicar elementos	122
B.11	Adicionar opções de resposta	123
B.12	Eliminar opções de resposta	124
B.13	Pré-visualizar formulário	125

B.14 Guardar formulário	125
B.15 Filtrar formulários	127
B.16 Eliminar formulários	127
B.17 Editar informação do formulário	128
B.18 Consultar formulários	130
B.19 Publicar formulário	131
B.20 Consultar formulários publicados	132
B.21 Preencher formulário	133
C Plano de testes	134
C.1 Testes unitários	134
C.2 Testes de Sistema	144

Acrónimos

API Application programming interface. 45, 47, 56

CSS Cascading Style Sheets. 55

FCTUC Faculdade de Ciências e Tecnologias da Universidade de Coimbra. 1

HTML Hypertext Markup Language. 13, 45, 55

HTTP HyperText Transfer Protocol. 45

JSON JavaScript Object Notation. 45, 47, 56

MEI Mestrado em Engenharia Informática. 1

QM Quality Management. 2

REST Representational State Transfer. 44, 45, 47, 56

SAAS Software as a service. 46

URL Uniform Resource Locator. 7

WFO Workforce Otimization. 2

Está página é intencionalmente deixada em branco.

Lista de Figuras

2.1	Planos do Typeform	7
2.2	Ecrã inicial do Typeform	8
2.3	Criar formulário no Typeform	8
2.4	Criar formulário em branco	9
2.5	Tipos de questões do Typeform	9
2.6	Exemplo de um formulário no Typeform	10
2.7	Exemplo de um formulário no Typeform	10
2.8	Publicar formulário	11
2.9	Meios de partilha de formulários no Typeform	11
2.10	Ecrã inicial do Google Forms	12
2.11	Tipos de questões no Google Forms	13
2.12	Partilha de formulários no Google Forms	13
2.13	Planos da plataforma Survey Monkey	14
2.14	Ecrã inicial da SurveyMonkey	15
2.15	Modelos de formulários no SurveyMonkey	15
2.16	Criar formulário no SurveyMonkey	16
2.17	Tipos de questões no SurveyMonkey	16
2.18	Modelo de formulário no SurveyMonkey	17
2.19	Personalizar modelo de formulário no SurveyMonkey	17
2.20	Partilhar formulários no SurveyMonkey	18
2.21	Planos do ZohoSurvey	18
2.22	Ecrã inicial do ZohoSurvey	19
2.23	Criar formulário no ZohoSurvey	19
2.24	Tipo de questões do ZohoSurvey	20
2.25	Modelos do ZohoSurvey	20
2.26	Personalização de layout dos formulários	21
2.27	Partilha de formulários no ZohoSurvey	21
2.28	Planos da SoGoSurvey	22
2.29	Ecrã inicial da SoGoSurvey	22
2.30	Modelos de formulários da SoGoSurvey	23
2.31	Tipos de questões	23
2.32	Meios de partilha de formulários no SoGoSurvey	24
3.1	Funcionamento do Scrum	28
3.2	Funcionamento do Git	29
3.3	Planeamento inicial - 1º semestre	30
3.4	Execução real - 1º semestre	31
3.5	Planeamento inicial - 2º semestre	32
3.6	Execução real - 2º semestre	33
3.7	Matriz de probabilidade/impacto	35
3.8	Matriz de probabilidade/impacto dos riscos identificados	36

4.1	Interação do sistema com o administrador	41
4.2	Interação do sistema com o supervisor	41
4.3	Caso de uso "Adicionar elementos ao formulário"	42
4.4	Elementos adicionados ao formulário: <i>New section, Yes/No, ShortText e OpinionScale</i>	43
4.5	Elementos adicionados ao formulário: <i>SingleChoice e MultipleChoice</i>	43
5.1	Representação da arquitetura do sistema	44
5.2	Legenda dos componentes da arquitetura	45
5.3	Representação da arquitetura da aplicação na cloud	46
5.4	Ficheiro db.json	47
5.5	Exemplo da representação de um formulário	47
5.6	Representação de elemento do formulário - Section	48
5.7	Representação de elemento do formulário - Short Text	49
5.8	Representação de elemento do formulário - Long Text	49
5.9	Representação de elemento do formulário - Full name	50
5.10	Representação de elemento do formulário - Yes/No	50
5.11	Representação de elemento do formulário - Date Picker	50
5.12	Representação de elemento do formulário - Time	51
5.13	Representação de elemento do formulário - Number	51
5.14	Representação de elemento do formulário - Opinion Scale	52
5.15	Representação de elemento do formulário - Star Rating	52
5.16	Representação de elemento do formulário - Favorite Rating	52
5.17	Representação de elemento do formulário - Slider Scale	53
5.18	Representação de elemento do formulário - Single Choice	53
5.19	Representação de elemento do formulário - Multiple Choice	54
7.1	Representação de <i>branches</i> no Github	60
7.2	Representação de <i>pull requests</i> no GitHub	60
7.3	Descrição de <i>Pull Request</i> no GitHub	61
7.4	Alterações - Pull Request	61
7.5	Execução do hook "Pre-commit"	62
7.6	Hook "Pre-commit"	62
7.7	Hook "commit-msg"	63
7.8	Hook "pre-push"	63
7.9	Hook pre-push - Testes	63
7.10	Verificações em ficheiros	64
7.11	Verificações em ficheiros	64
7.12	Cobertura do código em cada ficheiro	65
7.13	Cobertura do código em cada <i>commit</i>	66
8.1	Cobertura total do código-fonte	69
8.2	Cobertura do código por ficheiro	69
8.3	Exemplo de teste de sistema	71
A.1	Casos de uso e interação com o administrador	77
A.2	Casos de uso e interação com o supervisor	78
A.3	Criar novo formulário	82
A.4	Preencher dados do formulário	82
A.5	Elementos adicionados: <i>New section, Yes/No, ShortText, OpinionScale</i>	83
A.6	Elementos adicionados: <i>Single Choice e Multiple Choice</i>	84
A.7	Apagar elementos do formulário	85

A.8	Duplicar elemento	86
A.9	Adicionar opção de resposta	87
A.10	Opção adicionada	87
A.11	Eliminar opções	88
A.12	Opção eliminada	89
A.13	Pré-visualização de formulário	90
A.14	Guardar formulário	91
A.15	Lista de formulários criados	92
A.16	Ver formulário	93
A.17	Formulário	93
A.18	Eliminar formulário	94
A.19	Formulário eliminado	95
A.20	Editar informação do formulário	96
A.21	Pesquisar pelo nome do formulário	97
A.22	Publicar formulário	98
A.23	Lista de formulários publicados	99
A.24	Preencher formulário	100
A.25	Escolher o tipo de formulário	101
A.26	Selecionar modelo de formulário	101
A.27	Formulário submetido	102
B.1	Página de login	103
B.2	Página de login	104
B.3	Página inicial do administrador	104
B.4	Página inicial do Supervisor	105
B.5	Logout - Administrador	106
B.6	Logout - Supervisor	106
B.7	Logout - Supervisor	107
B.8	Criar novo formulário	107
B.9	Criar novo formulário - Validação do nome e descrição	108
B.10	Criar novo formulário - Erro	108
B.11	Criar novo formulário - Erro	109
B.12	Página para criar formulário	109
B.13	Adicionar elementos - "New section"	110
B.14	Adicionar elementos - "Short Text"	111
B.15	Adicionar elementos - "Long Text"	111
B.16	Adicionar elementos - "Full Name"	112
B.17	Adicionar elementos - "Date Picker"	112
B.18	Adicionar elementos - "Time"	113
B.19	Adicionar elementos - "Number"	113
B.20	Adicionar elementos - "Opinion Scale"	114
B.21	Adicionar elementos - "Favorite Rating"	114
B.22	Adicionar elementos - "Star Rating"	115
B.23	Adicionar elementos - "Slider Scale"	115
B.24	Adicionar elementos - "NPS"	116
B.25	Adicionar elementos - "Single Choice"	116
B.26	Adicionar opções - "Single Choice"	117
B.27	Eliminar opções - "Single Choice"	117
B.28	Adicionar elementos - "Multiple Choice"	118
B.29	Adicionar opções - "Multiple Choice"	118
B.30	Eliminar opções - "Multiple Choice"	119
B.31	Elementos <i>New section, Yes/No, Star Rating, Long Text, New section e Slider Scale</i>	119

B.32 Elementos <i>Date Picker, Time, Yes/No, Number e Favorite Rating</i>	120
B.33 Elementos <i>Date Picker, Time, Yes/No, Number e Favorite Rating</i>	120
B.34 Eliminar elemento do formulário	121
B.35 Eliminar elemento do formulário	121
B.36 Opção para duplicar elemento	122
B.37 Elemento duplicado	122
B.38 Opção para adicionar opções de resposta	123
B.39 Opções adicionadas	123
B.40 Eliminar opção de resposta	124
B.41 Opção eliminada	124
B.42 Pré-visualização de formulário	125
B.43 Opção para guardar formulário	125
B.44 Formulário a ser guardado	126
B.45 Lista de formulários guardados no sistema	126
B.46 Filtrar formulários	127
B.47 Eliminar formulário	127
B.48 Eliminar formulário	128
B.49 Editar informação do formulário	128
B.50 Editar informação do formulário	129
B.51 Editar informação do formulário	129
B.52 Editar informação do formulário	130
B.53 Consultar formulário	130
B.54 Formulário	131
B.55 Publicar formulário	131
B.56 Formulário formulário	132
B.57 Lista de formulários publicados	132
B.58 Formulário preenchido	133

Está página é intencionalmente deixada em branco.

Lista de Tabelas

2.1	Avaliação das diferentes soluções	25
4.1	Lista de requisitos funcionais	38
8.1	Exemplo de teste realizado	69

Está página é intencionalmente deixada em branco.

Capítulo 1

Introdução

O presente relatório descreve o trabalho realizado pela aluna Flávia Daniela Passos Cardoso Costa, no âmbito da disciplina "Dissertação/Estágio" do Mestrado em Engenharia Informática (MEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC). A aluna tem como orientadores o professor Doutor Paulo Rupino, do Departamento de Engenharia Informática da Universidade de Coimbra e os engenheiros André Simões e Daniel Afonso da Talkdesk.

Na secção 1.1 é fornecido um enquadramento do local de estágio. Na secção 1.2 são apresentados conceitos relacionados com o tema em estudo, bem como os objetivos que se procuram atingir. Seguidamente, na secção 1.3, é especificada a motivação para a realização deste projeto e finalmente, na secção 1.4, é descrita a estrutura do documento.

1.1 Local de estágio

O estágio foi realizado de Setembro de 2019 a Junho de 2020, na empresa Talkdesk, localizada no Instituto Pedro Nunes, em Coimbra.

A Talkdesk foi fundada em 2011, em São Francisco, nos Estados Unidos da América. Atualmente, tem escritórios em Londres, Lisboa, Madrid, Salt Lake *city*, Coimbra, Aveiro e Porto. O seu propósito é desenvolver soluções de *call centers* baseadas na nuvem para ajudar as empresas a melhorar a satisfação dos seus clientes e, ao mesmo tempo, reduzir os custos de suporte [3]. A tecnologia existente centra-se numa plataforma que permite criar um *call center* facilmente e integrá-lo com uma variedade de sistemas, tais como: *Salesforce*, *Zendesk*, *Slack* e *ServiceNow* [13].

As soluções desenvolvidas permitem que um cliente que liga para um *call center* seja automaticamente reencaminhado para o agente mais qualificado para atender às suas necessidades. Este tem acesso, em tempo real, aos dados do cliente, como o nome, as informações de contacto e o seu histórico de chamadas [2]. Além disso, quando o cliente autoriza, as chamadas são gravadas para posterior análise da qualidade das interações. Caso exista a necessidade de divulgar informação privada, como por exemplo, dados bancários, a gravação é interrompida.

Na Talkdesk, "todo o processo é realizado apenas com um computador e uma ligação à internet" [2].

O *call center* criado adapta-se assim às necessidades dos clientes e equipas, para melhorar a sua experiência, aumentar a produtividade da empresa e reduzir os custos de suporte [3].

1.2 Enquadramento e Objetivos

Os *call centers* são centrais de atendimento telefónico utilizados como elo de ligação entre as empresas e os seus clientes. A sua construção requer estratégias bem definidas, sendo assim importante determinar previamente quais as funções que devem ser suportadas e quais os objetivos concretos de cada organização, sempre com o intuito de melhorar a experiência do cliente [36]. De entre os principais serviços oferecidos num *call center* [33] estão:

- **Apoio ao cliente** - Responsável por fortalecer o relacionamento com os clientes, esclarecer as suas dúvidas, atender reclamações e dar orientações gerais.
- **Vendas** - Serviço utilizado para comunicar com os clientes a fim de vender determinados produtos ou serviços oferecidos pela empresa.
- **Telemarketing** - Serviço responsável por divulgar mensagens publicitárias das empresas, realizar pesquisas de mercado e opinião com os consumidores.

A má gestão de um *call center* pode originar uma má experiência para o cliente e para os funcionários envolvidos no atendimento [36]. Assim sendo, é importante definir metas e aplicar estratégias que garantam um bom desempenho.

Os líderes destes setores estão a apostar na otimização da força de trabalho (WFO) como uma **"alternativa para melhorar a experiência dos clientes, diminuir o tempo de resposta e, ao mesmo tempo melhorar a produtividade dos agentes"** [29].

Estas soluções integram um conjunto de serviços projetados para planejar, desenvolver e implementar estratégias que aumentem a eficiência operacional [1], [8], tais como:

Gravações das comunicações, para garantia de qualidade ou análise da fala, gestão da qualidade (QM) para determinar quão bem os agentes aderem às políticas e procedimentos internos, observação do *desktop* do agente, para monitorizar tudo o que acontece na área de trabalho, análise de texto para extrair informações de dados não estruturados, gestão de desempenho para alinhar as atividades do *call center* com os objetivos da empresa, entre outros.

Através do módulo **"Gestão da Qualidade"**, em particular, é possível avaliar o nível de interação dos agentes com os clientes e obter diagnósticos e *insights* sobre os processos internos e externos [22]. A partir destes, podem surgir novas oportunidades de negócio e informação que permita efetuar melhorias e definir novas estratégias.

O objetivo deste projeto é o desenvolvimento de uma aplicação web que permita construir formulários para monitorizar a qualidade do atendimento prestado em *call centers*. Esta plataforma está inserida no módulo de gestão de qualidade (QM), da Talkdesk. Os questionários criados através deste sistema têm o intuito de avaliar o serviço prestado por um agente de atendimento após uma interação com um cliente. A partir destas avaliações, pretende-se identificar erros e possibilidades de melhoria, para tornar o serviço mais eficiente e produtivo. Todos os modelos de questões são criados pelo administrador do *call center*. Esta é a pessoa responsável por definir as questões que o formulário deve conter e o conjunto de pontos que devem ser avaliados, tendo em conta as necessidades da empresa e o motivo do contacto. Assim, é possível adaptar os questionários a cada tipo de atendimento, onde a avaliação da cordialidade, agilidade e outras competências dos agentes podem ser completamente diferentes.

Após serem criados, os formulários podem ser publicados. Desta forma, o supervisor do *call center*, pode analisar as gravações das chamadas e preencher as questões de acordo com a qualidade do atendimento. Para tal, acede à plataforma, vê os formulários publicados previamente pelo administrador e efetua o seu preenchimento. Deste modo, pode identificar o agente, avaliar a forma como a chamada começou, verificar se as políticas da empresa foram respeitadas, determinar se o responsável pelo atendimento foi capaz de atender às necessidades do cliente, identificar os métodos propostos para solucionar o problema, verificar se o cliente ficou satisfeito com o atendimento, selecionar a data e hora da chamada, entre outros.

Todos os questionários existentes na aplicação são personalizáveis para que consigam atender às necessidades de controlo da qualidade em diversas situações. As diferenças entre os vários tipos de questões centram-se essencialmente nos itens que serão avaliados, de acordo com o tipo de atendimento.

Através do preenchimento dos formulários, os agentes podem, posteriormente, ser informados pelo supervisor relativamente ao seu desempenho e alertados para os pontos que merecem atenção, sendo assim possível aprimorar a transparência e a consistência das práticas de garantia de qualidade, além de compreender as necessidades dos clientes e melhorar os índices de satisfação.

Este projeto centra-se essencialmente no desenvolvimento de um sistema que permita:

- **Criação e gestão de formulários**, de acordo com os itens que precisam de ser avaliados em cada atendimento.
- **Partilha de formulários**, entre os dois utilizadores do sistema, o administrador e o supervisor. Enquanto o administrador é a pessoa responsável por criar os vários modelos de formulários e publicá-los, o supervisor é quem pode vê-los e preenchê-los.

1.3 Motivação

Tendo por base os conceitos referidos no enquadramento, o desenvolvimento de um sistema eficaz e confiável para controlo da qualidade em *call centers*, é muito vantajoso, uma vez que além de poder melhorar os resultados das empresas a longo prazo, pode aumentar o nível de satisfação dos clientes e dos funcionários responsáveis pelo atendimento.

Atualmente, a Talkdesk está a implementar soluções para amplificar a qualidade das interações e melhorar a experiência dos clientes. Contudo, ainda não possui sistemas que ofereçam suporte a todas as suas necessidades, nomeadamente um sistema que permita a criação de formulários personalizados para avaliar as interações dos agentes com os clientes.

A recolha dessas métricas é bastante importante, uma vez que a partir destas é possível tomar decisões para melhorar o serviço prestado e identificar as áreas que precisam de melhorias.

Apesar da plataforma desenvolvida estar inserida na aplicação de controlo de qualidade da Talkdesk, esta não será integrada com os restantes módulos de Gestão de Qualidade, durante o decorrer deste estágio, nomeadamente, o módulo que permite a gravação e análise das chamadas. Isto deve-se ao facto deste último não se encontrar totalmente desenvolvido pela Talkdesk e não fazer parte do âmbito deste estágio.

1.4 Estrutura do documento

Este relatório encontra-se estruturado em diversos capítulos, enumerados de seguida:

- **Capítulo 1: Introdução** - Identificação e descrição do local de estágio, enquadramento, objetivos, motivação e estrutura do documento.
- **Capítulo 2: Estado de arte** - Análise e comparação de soluções existentes no mercado, as suas vantagens e desvantagens e identificação dos motivos que levam à necessidade de uma nova solução.
- **Capítulo 3: Abordagem** - Constituição da equipa, metodologias escolhidas para a gestão do projeto e para controlo de versões do *software* desenvolvido, planeamento do estágio e por fim, análise de riscos.
- **Capítulo 4: Levantamento de requisitos** - Identificação dos requisitos da aplicação e especificação através de casos de uso e *mockups*.
- **Capítulo 5: Arquitetura e Modelo de dados** - Apresentação da arquitetura da aplicação e modelo de dados.
- **Capítulo 6: Tecnologias e ferramentas** - Apresentação das tecnologias e ferramentas utilizadas, bem como as suas vantagens e desvantagens.
- **Capítulo 7: Qualidade do código** - Descrição do processo realizado durante a implementação do projeto para garantir a qualidade do código-fonte.
- **Capítulo 8: Testes** - Descrição dos testes realizados para validar as funcionalidades implementadas.
- **Capítulo 9: Conclusões** - Apresentação das conclusões obtidas com a realização deste projeto.

Está página é intencionalmente deixada em branco.

Capítulo 2

Estado de arte

Neste capítulo é descrito o estudo do estado da arte realizado no âmbito deste projeto de estágio. A secção 2.1 apresenta uma introdução aos construtores de formulários, quais as suas principais funções e utilidade. Em seguida, na secção 2.2 são identificadas e descritas algumas das soluções mais conhecidas e referenciadas na web. Na secção 2.3 são apresentados os fatores utilizados para as comparar e por fim, na secção 2.4 é feita uma análise das mesmas, tendo em conta os vetores de comparação definidos na secção 2.3 e também são identificadas as razões que levam à necessidade de uma nova solução.

2.1 Construtores de formulários

Num *call center*, o preenchimento de questionários é importante para determinar a qualidade do serviço prestado pelos agentes de atendimento. Para facilitar este processo, é necessário obter informação a partir de questões concretas, organizadas em diferentes modelos prontos para preenchimento. Neste contexto, surge a necessidade de ter um construtor de formulários - sistema que permite criar questões personalizadas e específicas de acordo com as necessidades de cada utilizador.

Através dos recursos presentes neste tipo de aplicações, é possível obter informação mais facilmente e utilizá-la para os mais diversos fins, como por exemplo:

- Obter *feedback* do produto - Analisar quais os pontos positivos e os que precisam de melhorias;
- Realizar pesquisas de mercado - Obter conhecimento sobre um determinado mercado e o seu público para identificar oportunidades e reduzir o risco na tomada de decisões estratégicas;
- Obter *feedback* do serviço ao cliente - Determinar a satisfação dos clientes;
- Avaliar os funcionários - Analisar o desempenho dos funcionários de acordo com o serviço prestado aos clientes.

2.2 Soluções

Atualmente existem muitas soluções que permitem criar formulários para os mais diversos fins, como registo, contacto, *feedback*, pagamento, pesquisa, entre outros. Porém, uma vez que o âmbito deste projeto é a construção de formulários de avaliação, optou-se por analisar apenas as soluções mais utilizadas para a criação deste tipo de questionários.

De seguida são descritas algumas das mais populares e referenciadas na web [37],[28] bem como as suas características mais relevantes, utilizadas como referência ao longo do desenvolvimento do projeto.

2.2.1 Typeform

O Typeform é uma plataforma web que permite construir formulários dinâmicos e criativos [28] e que pode ser acessada a partir do seguinte URL: <https://www.typeform.com/>. Contém diversos planos: o gratuito, disponibilizado após a criação de uma conta através de um endereço de *email* e três com um custo associado.

A figura 2.1 representa algumas das diferenças existentes nos vários planos.

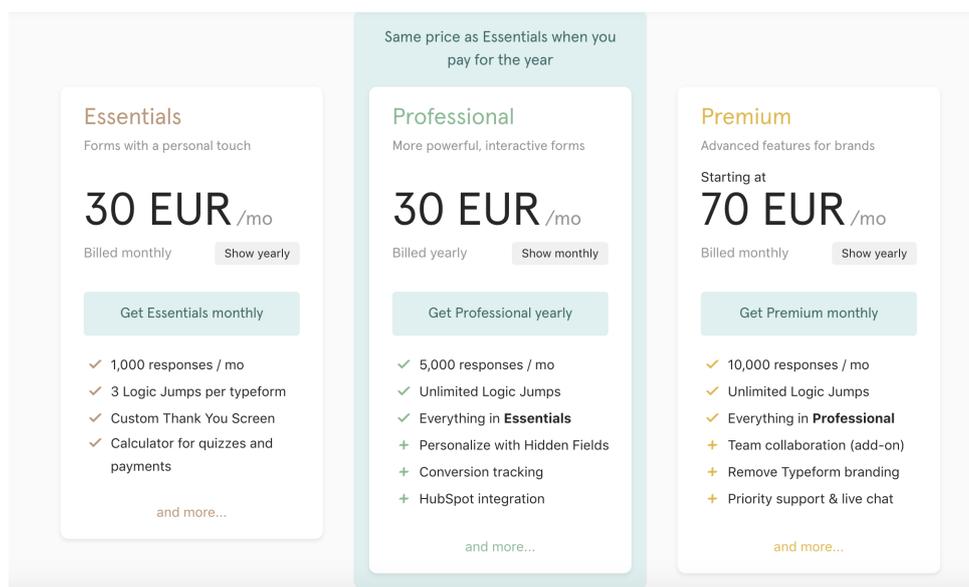


Figura 2.1: Planos do Typeform

Fonte: *Printscreen* do Website do Typeform

Estes planos distinguem-se essencialmente pelo número de respostas que é possível obter através da utilização dos questionários e também pelas funcionalidades que se podem utilizar. Por exemplo, os planos que têm um custo associado permitem integrar os questionários com outros *softwares*.

Apresentam-se, de seguida, figuras que ilustram o funcionamento da aplicação.

Através da figura 2.2 é possível ver que quando o utilizador entra na aplicação, é-lhe apresentada a opção para criar um novo formulário (*New Typeform*).

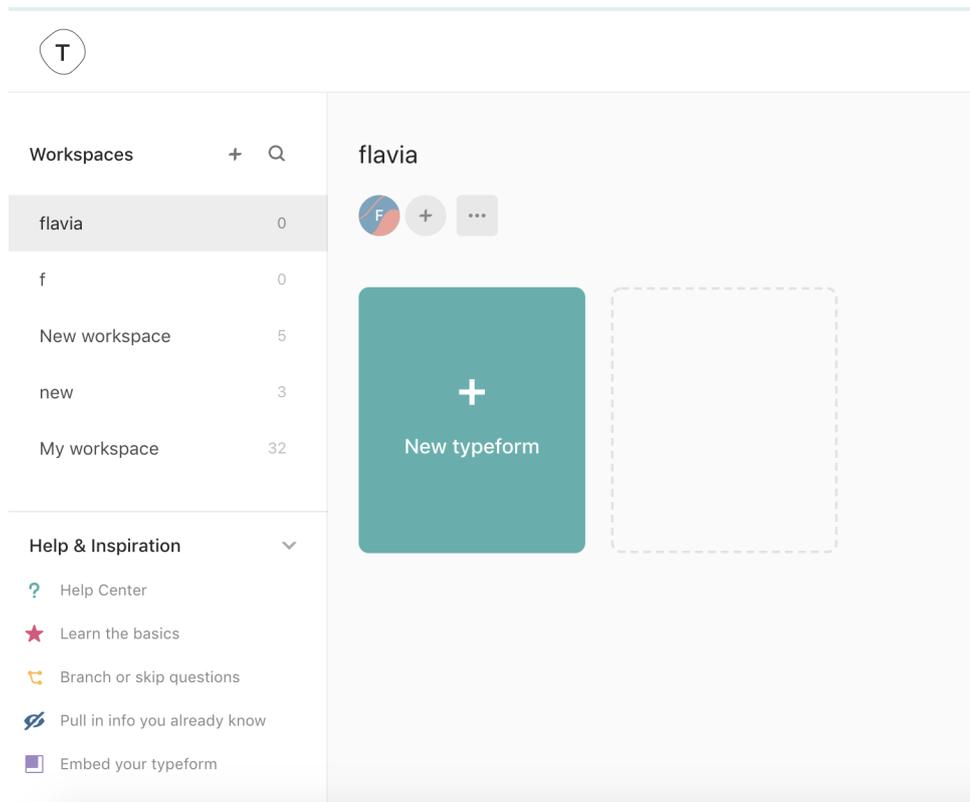


Figura 2.2: Ecrã inicial do Typeform

Fonte: *Printscreen* do Website do Typeform

Quando seleciona essa opção, é direcionado para uma nova página (Figura 2.3).

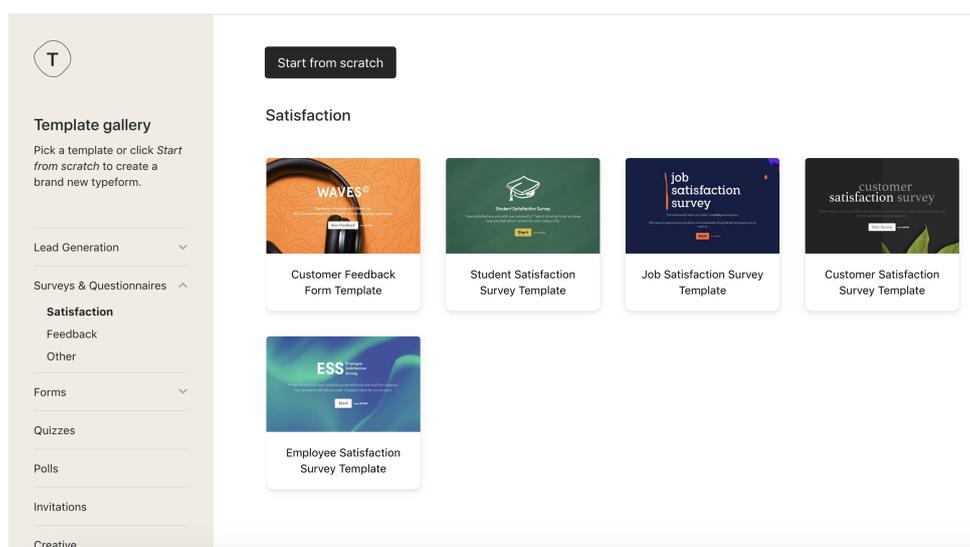


Figura 2.3: Criar formulário no Typeform

Fonte: *Printscreen* do Website do Typeform

Através desta interface tem a possibilidade de escolher entre um questionário em branco, isto é, sem qualquer elemento adicionado, ou seleccionar um dos modelos existentes na galeria (Avaliação da satisfação, *feedback*, entre outros).

Quando o utilizador selecciona a opção para criar um novo formulário em branco, deve preencher os seus dados e, em seguida, um novo projeto é criado, tal como está representado na figura 2.4.

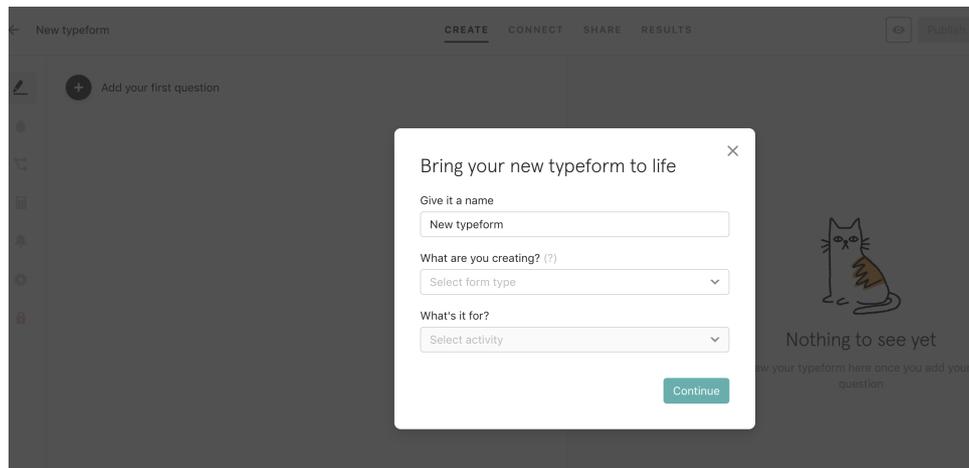


Figura 2.4: Criar formulário em branco

Fonte: *Printscreen* do Website do Typeform

Posteriormente é-lhe disponibilizada uma opção para adicionar questões ao questionário (Figura 2.5).

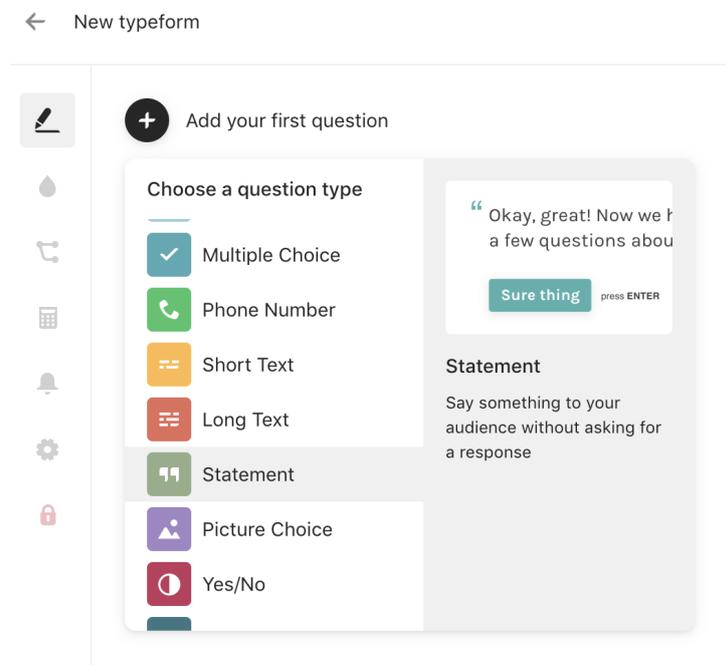


Figura 2.5: Tipos de questões do Typeform

Fonte: *Printscreen* do Website do Typeform

As questões adicionadas podem ser de vários tipos, desde escolha múltipla, avaliação, seleção, classificação, escalas de opinião, resposta curta, resposta longa, entre outras. Após adicionar as questões, o utilizador pode pré-visualizar o modelo criado.

Como se vê observa nas figuras 2.6 e 2.7, cada questão é mostrada individualmente no formulário, isto é, o utilizador apenas consegue ver a questão seguinte quando preenche a anterior.

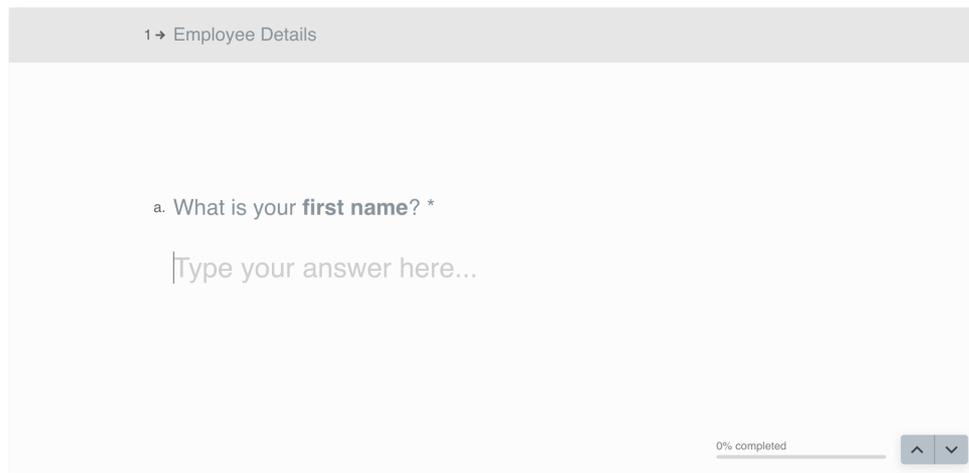
A screenshot of a Typeform interface. At the top, a grey header bar contains the text "1 → Employee Details". Below this, the question "a. What is your first name? *" is displayed. Underneath the question is a text input field with the placeholder text "Type your answer here...". In the bottom right corner of the question area, there is a progress indicator showing "0% completed" and two small square buttons, one with an upward arrow and one with a downward arrow.

Figura 2.6: Exemplo de um formulário no Typeform

Fonte: *Printscreen* do Website do Typeform

A screenshot of a Typeform interface. At the top, a grey header bar contains the text "2 → Self-evaluation". Below this, the question "a. Quality of Work *" is displayed. Underneath the question is the text "I complete my work thoroughly and with care, correctly following established _____ processes and procedures." Below this text is a Likert scale consisting of five rectangular buttons labeled "1", "2", "3", "4", and "5". Underneath the scale, the text "Strongly disagree" is aligned with the "1" button and "Strongly agree" is aligned with the "5" button. In the bottom right corner of the question area, there is a progress indicator showing "0% completed" and two small square buttons, one with an upward arrow and one with a downward arrow.

Figura 2.7: Exemplo de um formulário no Typeform

Fonte: *Printscreen* do Website do Typeform

Quando completos, os questionários podem ser publicados, isto é, tornados públicos para poderem ser preenchidos. Para tal, o utilizador apenas tem de selecionar a opção "Publish", como se verifica através da figura 2.8.

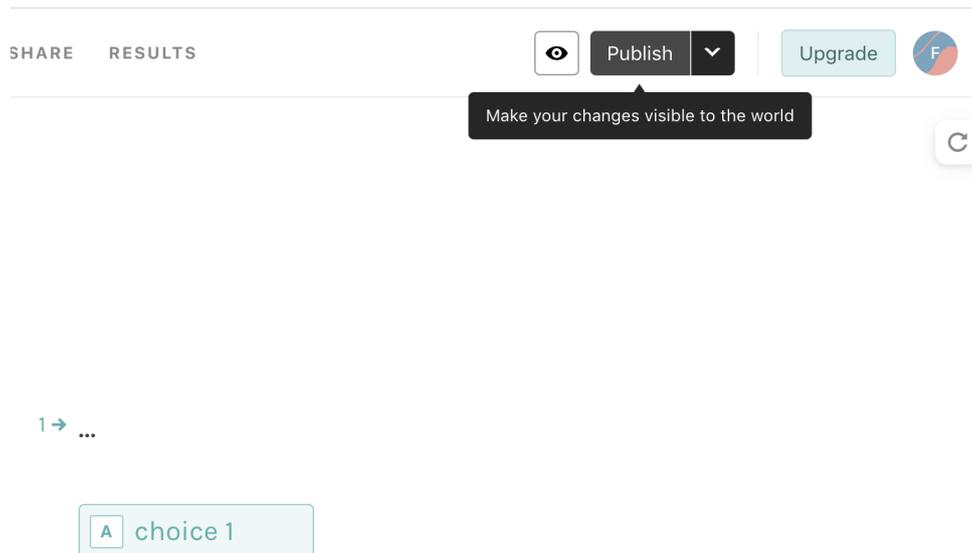


Figura 2.8: Publicar formulário

Fonte: *Printscreen* do Website do Typeform

Em seguida, os modelos de formulários criados podem ainda ser partilhados através de um *link* direto, nas redes sociais, numa página web, num *chat* ou ainda através de um *email*, ficando assim disponíveis para utilização (Figura 2.9).

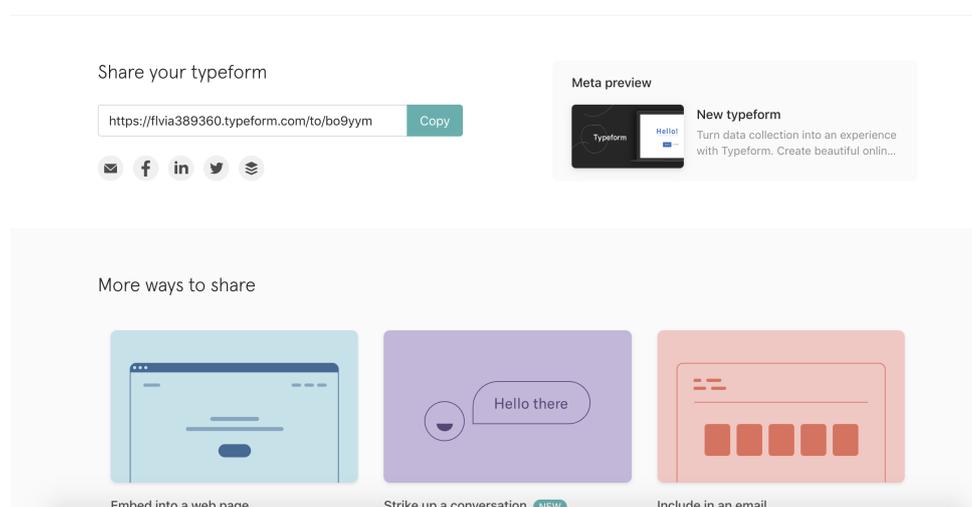


Figura 2.9: Meios de partilha de formulários no Typeform

Fonte: *Printscreen* do Website do Typeform

O Typeform é uma plataforma que se destaca positivamente pelo design dos questionários, uma vez que apresenta diversas opções de personalização. No entanto, também apresenta algumas limitações, tais como:

1. Não é possível alterar a apresentação das questões, isto é, apenas é possível ver uma questão de cada vez. Isto poderá limitar a utilização da plataforma, uma vez que poderão existir utilizadores que preferem criar formulários com as questões todas visíveis na mesma página;
2. Permite criar questionários sem qualquer nome ou descrição que os identifiquem e diferenciem dos restantes;

2.2.2 Google Forms

A Google Forms é uma plataforma para gerir e criar diversos tipos de questionários, incluindo pesquisas ou avaliações. É totalmente gratuita e disponibilizada pelo Google, em: <https://www.google.com/forms/about/>.

O facto de não ter qualquer custo associado, estar sempre disponível e permitir criar formulários ilimitados tornou-o um sistema muito popular [37].

Na figura 2.10 está representado o ecrã inicial da aplicação.

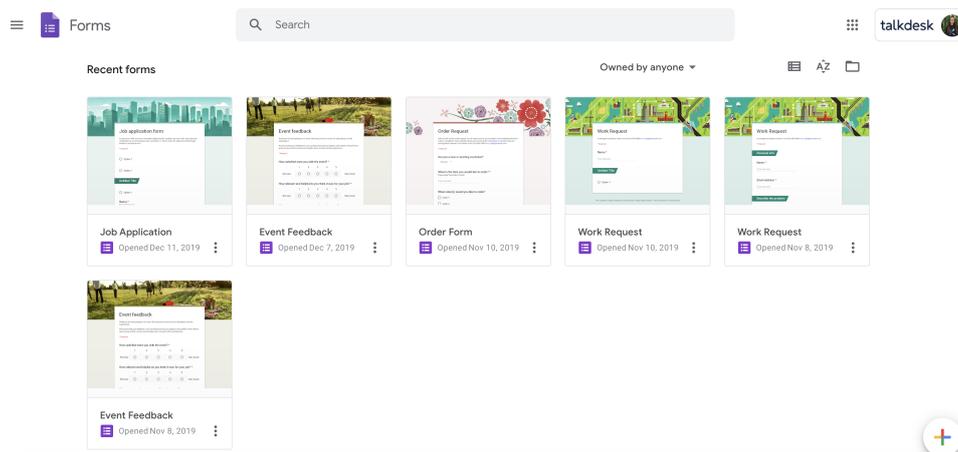


Figura 2.10: Ecrã inicial do Google Forms

Fonte: *Printscreen* do Website do Google Forms

A partir desta página, o utilizador pode seleccionar um dos formulários existentes no sistema ou criar um novo sem qualquer elemento adicionado. Se optar por criar um formulário em branco, pode adicionar vários tipos de questões (Figura 2.11).

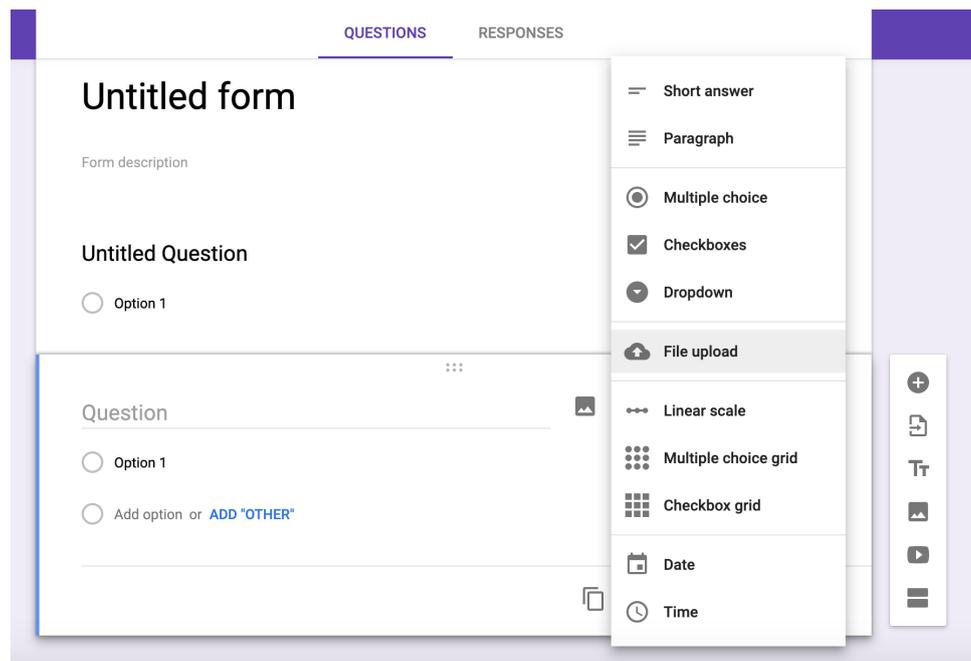


Figura 2.11: Tipos de questões no Google Forms

Fonte: *Printscreen* do Website do Google Forms

Embora existam várias elementos que podem ser adicionados, estes não são suficientes para a criação de questionários de avaliação. Por exemplo, a plataforma não contém elementos a partir dos quais o utilizador pode expressar a sua opinião, através de símbolos, tais como *emojis* ou estrelas. Também não contém elementos que permitam adicionar dados numéricos, ou respostas mais longas.

Depois de criados, os formulários podem ser pré-visualizados e publicados através de um endereço eletrónico, *link*, incorporados em código HTML ou ainda em redes sociais (Figura 2.12).

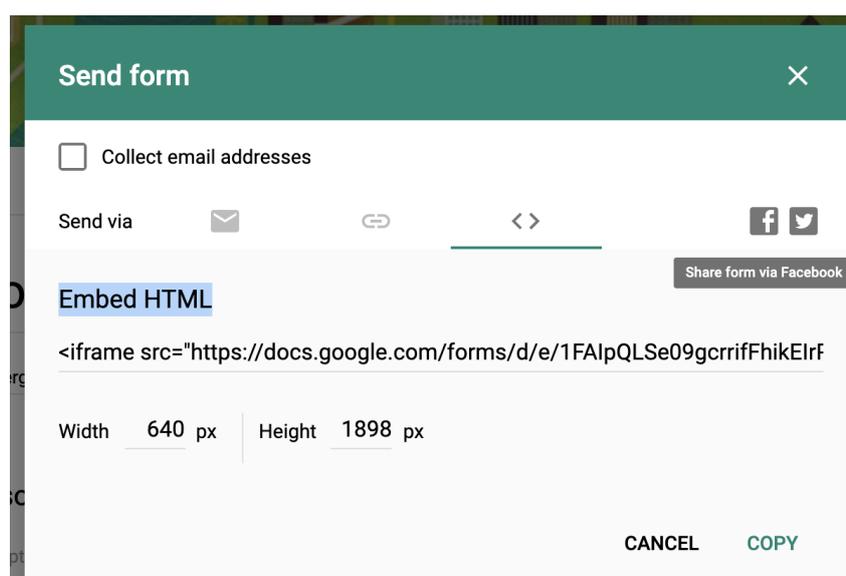


Figura 2.12: Partilha de formulários no Google Forms

Fonte: *Printscreen* do Website do Google Forms

A principal vantagem desta aplicação é a facilidade e simplicidade de utilização, uma vez que apresenta uma interface intuitiva [37]. Contudo, também contém algumas limitações, apresentadas de seguida:

1. Não contém alguns elementos que podem ser utilizados em formulários de avaliação, tal como referido anteriormente.
2. Permite a criação de inúmeros formulários com o mesmo nome, não existindo assim qualquer forma de os distinguir.

2.2.3 SurveyMonkey

A SurveyMonkey é uma ferramenta direcionada à criação de pesquisas [28] e está disponível no seguinte url: <https://www.surveymonkey.com/>.

Pode ser utilizada através da criação de uma conta para teste, embora tenha muitos outros planos com um custo associado, tal como está representado na figura 2.13.

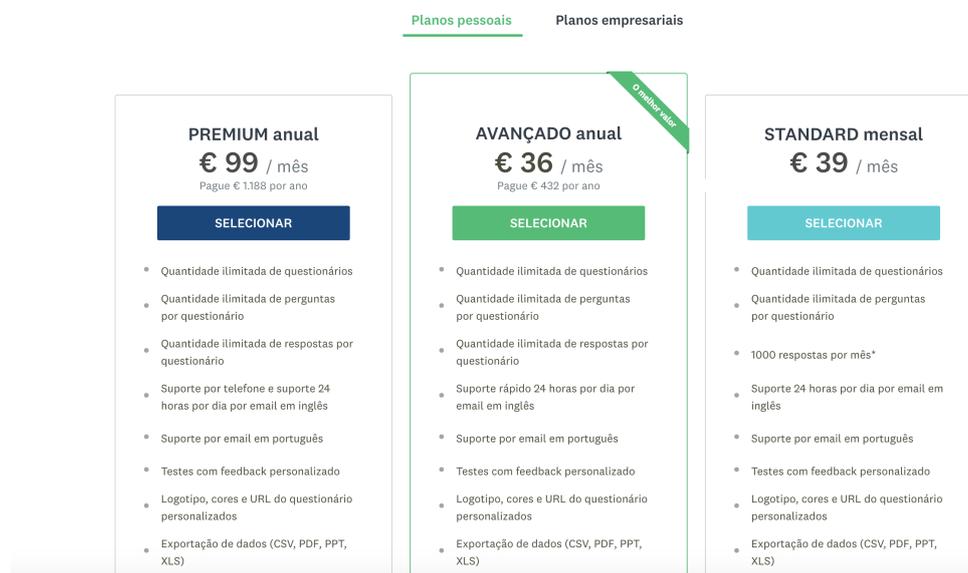


Figura 2.13: Planos da plataforma Survey Monkey

Fonte: *Printscreen* do Website do SurveyMonkey

As suas versões pagas permitem o acesso a recursos adicionais, como por exemplo: Adicionar ficheiros aos formulários, filtrar pesquisas de modo a destacar determinados resultados e construir gráficos.

Na figura 2.14 é apresentado o ecrã inicial da aplicação.

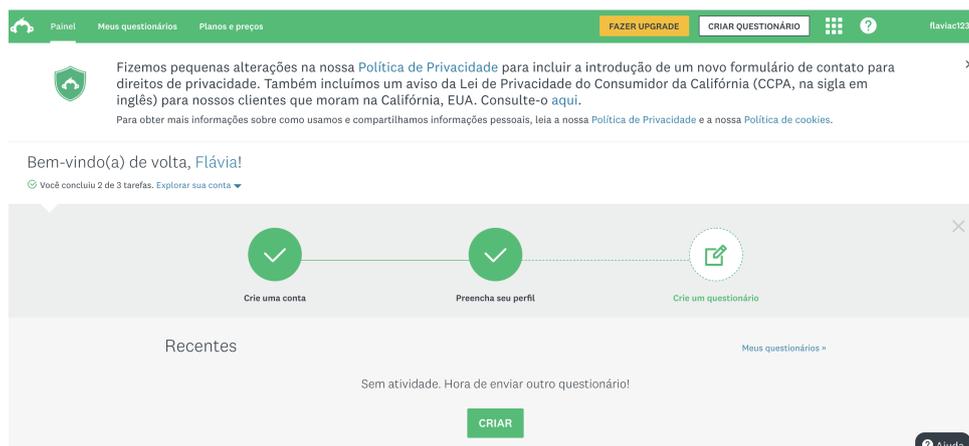


Figura 2.14: Ecrã inicial da SurveyMonkey

Fonte: *Printscreen* do Website da SurveyMonkey

Através deste ecrã, o utilizador tem a possibilidade de criar um novo questionário. Este pode ser iniciado do zero ou a partir de um dos modelos existentes (Figura 2.15).

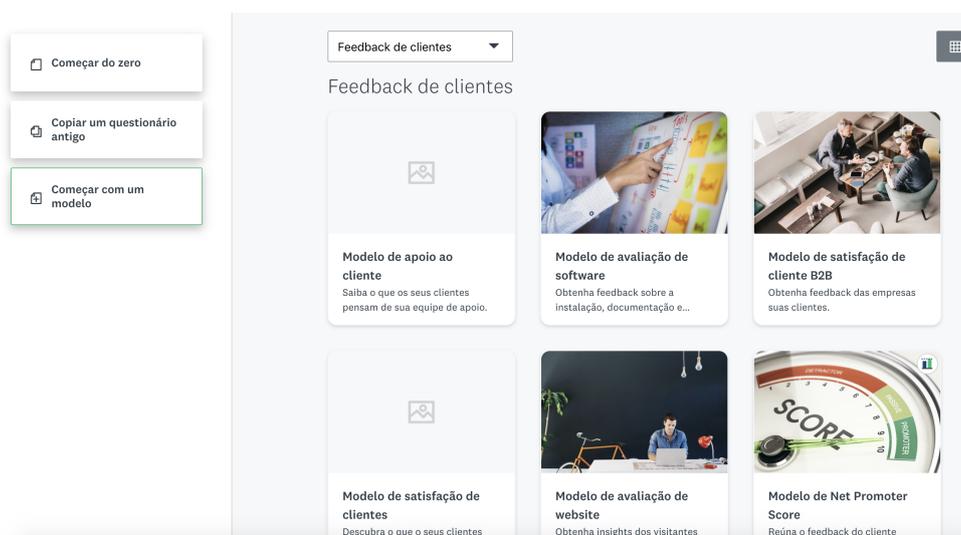


Figura 2.15: Modelos de formulários no SurveyMonkey

Fonte: *Printscreen* do Website do SurveyMonkey

Para ser iniciado a partir do zero, o utilizador tem de indicar algumas informações, como o título, a categoria e o formato do questionário. A partir desta página pode também escolher o *layout* que pretende (Figura 2.16).

Dê um título para o questionário

Título do questionário

Categoria do questionário

Minhas perguntas já foram criadas.

Escolha um formato de questionário (opcional) Qual é o melhor formato para mim?

- Uma pergunta de cada vez**
Role automaticamente para a próxima pergunta
- Clássico**
Mostrar todas as perguntas de uma página de uma vez
- Conversa BETA**
Transforme seu questionário em uma conversa de bate-papo

CRIAR QUESTIONÁRIO

Figura 2.16: Criar formulário no SurveyMonkey

Fonte: *Printscreen* do Website do SurveyMonkey

Em seguida, pode escolher o tipo de questão que quer adicionar (Figura 2.17).

Multiple Choice

- Multiple Choice ✓
- Dropdown
- Matrix / Rating Scale
- Ranking
- Slider
- Multiple Textboxes
- Contact Information
- Date / Time
- Single Textbox
- Comment Box
- File Upload
- Star Rating
- Checkboxes

SurveyMonkey

Figura 2.17: Tipos de questões no SurveyMonkey

Fonte: *Printscreen* do Website do SurveyMonkey

O utilizador tem diversas opções, incluindo questões utilizadas em formulários de avaliação e *feedback*, tais como: *Slider*, *star rating*, *ranking*, *date/time*, *multiple choice*, entre outras.

Se optar por utilizar um dos modelos pré-criados pode aceder a várias informações, como o nome e descrição, o número de vezes que foi utilizado e a quantidade de perguntas que contém (Figura 2.18).

Figura 2.18: Modelo de formulário no SurveyMonkey

Fonte: *Printscreen* do Website do SurveyMonkey

Os modelos podem ainda ser editados e personalizados quanto às cores, imagens, tema, tipo de questões e opções de resposta (Figura 2.19).

Figura 2.19: Personalizar modelo de formulário no SurveyMonkey

Fonte: *Printscreen* do Website do SurveyMonkey

Depois de completos, os questionários podem ser partilhados com qualquer pessoa que tenha um endereço de email e podem ser definidas permissões para escolher o que elas podem ver

e editar. Contudo, esta funcionalidade apenas está disponível para as versões pagas, como se verifica através da figura 2.20.

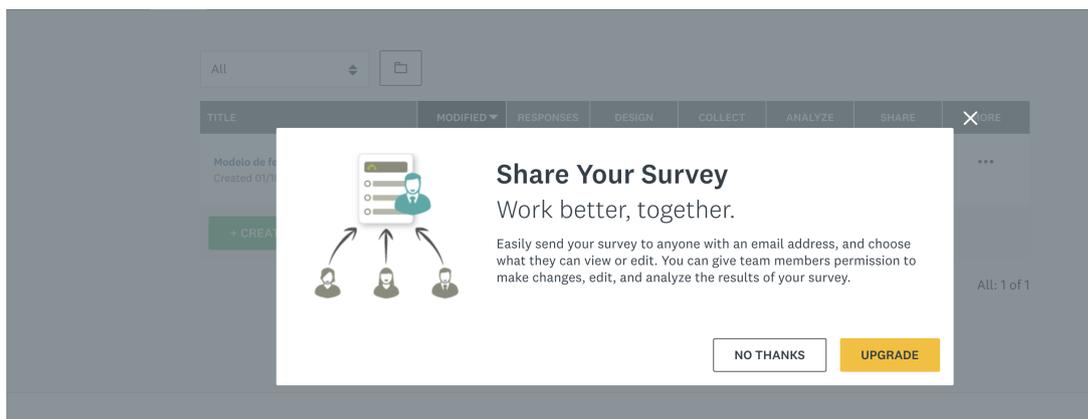


Figura 2.20: Partilhar formulários no SurveyMonkey

Fonte: *Printscreen* do Website do SurveyMonkey

As principais vantagens desta plataforma são a possibilidade de escolher o tipo de formulário pretendido, com todas as questões numa só página ou apenas uma de cada vez e também a variedade de questões que podem ser adicionadas para criar avaliações. A principal desvantagem é o facto de não ser totalmente intuitiva, uma vez que ao apresentar tantas recursos numa só página pode tornar-se confusa sobretudo para utilizadores iniciantes.

2.2.4 ZohoSurvey

O ZohoSurvey é uma ferramenta direccionada à criação de pesquisas, que podem ser das mais simples às mais complexas e encontra-se disponível em: <https://www.zoho.com/survey/>.

Tem um plano gratuito, disponível durante 7 dias, e três planos pagos que se distinguem essencialmente pelo tipo de funcionalidades permitidas (Figura 2.21).

FREE	PLUS	* MOST POPULAR *	ENTERPRISE
€0	€20	€25	€60
	/month billed annually	/month billed annually	/month billed annually
SIGN UP NOW	TRY FOR FREE	TRY FOR FREE	TRY FOR FREE
<ul style="list-style-type: none"> ✓ Unlimited surveys ✓ 10 Questions / Survey ✓ 100 Responses / Survey ✓ In-mail surveys ✓ Over 250 survey templates ✓ Embed survey in website ✓ Scoring ✓ HTTPS encryption ✓ Share on social ✓ Password protection ✓ 1 response collector/survey ✓ Real-time responses ✓ Export as PDF 	<ul style="list-style-type: none"> ✓ Unlimited surveys ✓ Unlimited questions ✓ Unlimited responses ✓ Over 250 survey templates ✓ Embed survey in website ✓ HTTPS encryption ✓ Share on social ✓ Password protection ✓ Unlimited response collectors ✓ Real-time responses ✓ Survey builder in 26 languages ✓ Skip, end page, and auto-fill logic 	<ul style="list-style-type: none"> Includes everything in Plus ✓ Piping logic ✓ Custom variable ✓ White label survey ✓ Upload favicon ✓ Multilingual surveys ✓ Multilingual reports ✓ Send email notifications ✓ URL callback notification (webhook) ✓ Export reports in SPSS ✓ Zoho CRM integration ✓ Trigger function 	<ul style="list-style-type: none"> Includes everything in Pro ✓ Admin + 2 users ✓ User management ✓ Department creation

Figura 2.21: Planos do ZohoSurvey

Fonte: *Printscreen* do Website do ZohoSurvey

Seguidamente são apresentadas algumas imagens que ilustram o seu funcionamento.

Na figura 2.22 é ilustrado o ecrã inicial da aplicação.

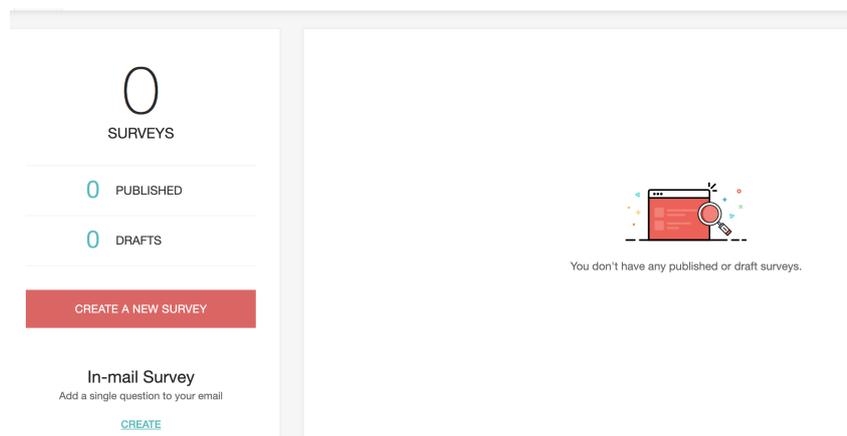


Figura 2.22: Ecrã inicial do ZohoSurvey

Fonte: *Printscreen* do Website do ZohoSurvey

À semelhança das anteriores, a partir da sua interface inicial é possível criar novos questionários. Quando essa opção é selecionada, o utilizador tem de inserir o nome e a categoria que pretende (Figura 2.23).

A screenshot of the ZohoSurvey form for creating a new survey. The form has a light gray background. At the top, it says 'SURVEY NAME *' in bold. Below this is a text input field containing 'Avaliar atendimento no call center'. Underneath that is 'CATEGORY NAME' in bold, followed by a dropdown menu showing 'Customer Satisfaction Survey' with a downward arrow. At the bottom of the form is a large red button with the text 'CREATE SURVEY' in white.

Figura 2.23: Criar formulário no ZohoSurvey

Fonte: *Printscreen* do Website do ZohoSurvey

Em seguida, pode escolher o tipo de questões que pretende adicionar, de entre as opções existentes (Figura 2.24).

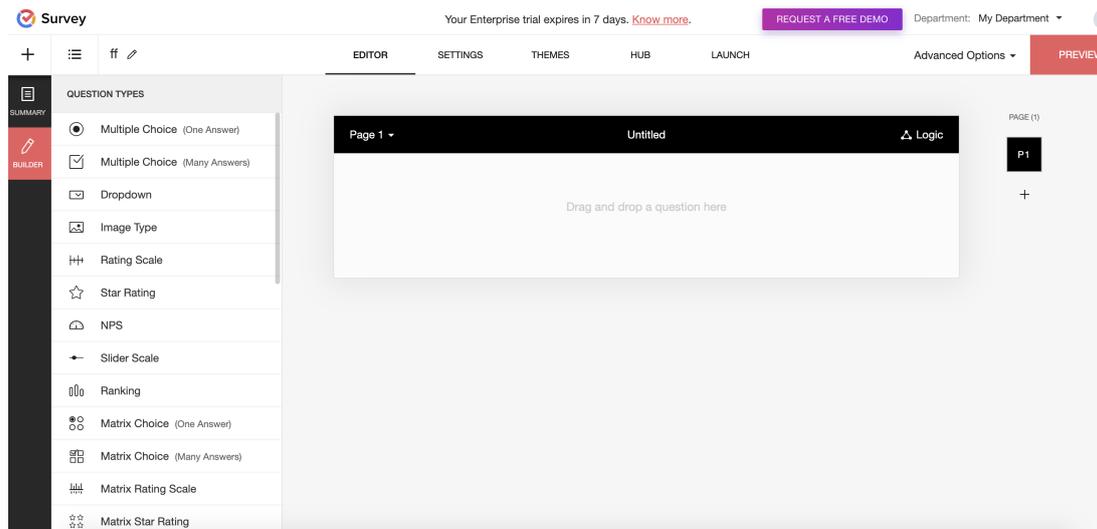


Figura 2.24: Tipo de questões do ZohoSurvey

Fonte: *Printscreen* do Website do ZohoSurvey

Também pode escolher utilizar os modelos pré-criados e personalizar o seu *layout* (Figuras 2.25 e 2.26).

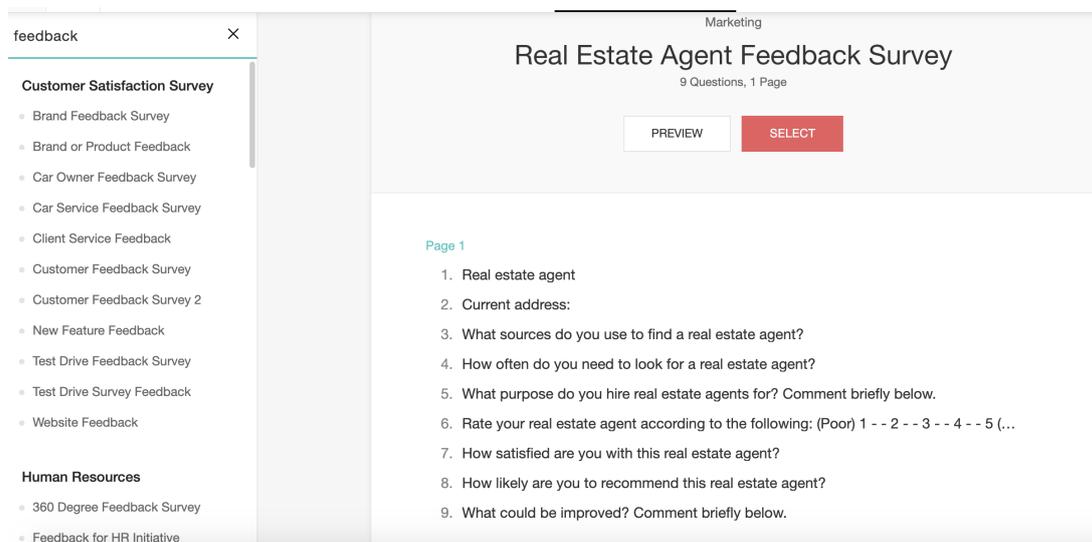


Figura 2.25: Modelos do ZohoSurvey

Fonte: *Printscreen* do Website do ZohoSurvey

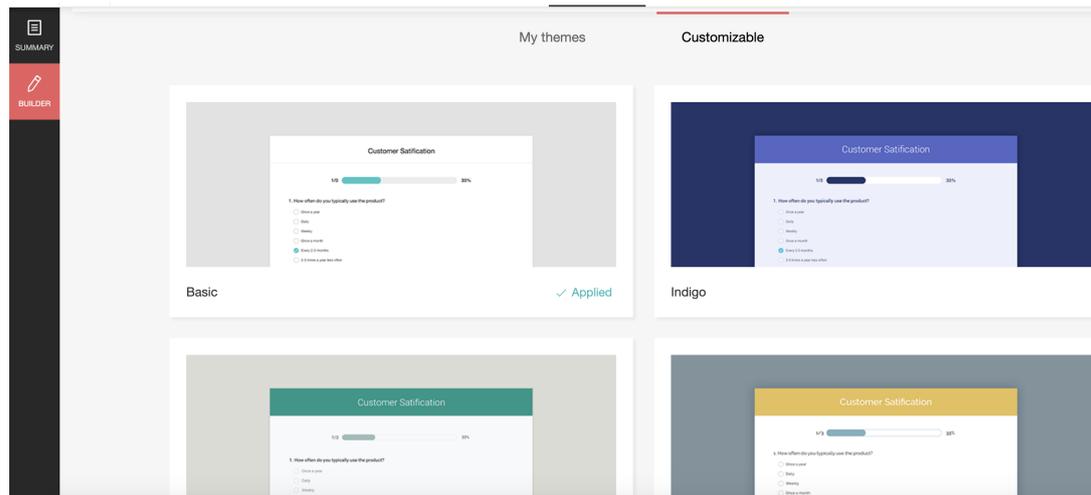


Figura 2.26: Personalização de layout dos formulários

Fonte: *Printscreen* do Website do ZohoSurvey

Para partilhar os formulários, pode utilizar um link direto ou as redes sociais(Figura 2.27).

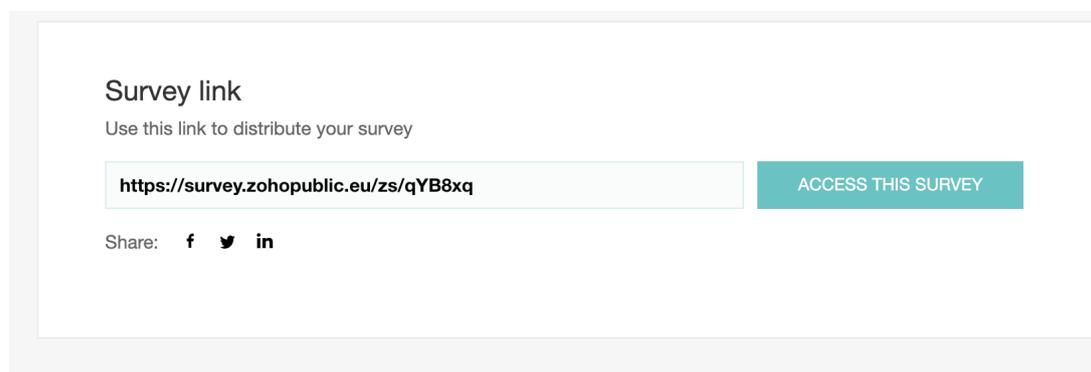


Figura 2.27: Partilha de formulários no ZohoSurvey

Fonte: *Printscreen* do Website do ZohoSurvey

O ponto mais positivo da aplicação é a diversidade de questões que podem ser utilizadas para fins de avaliação.

Como ponto negativo, destaca-se a dificuldade para editar o nome do formulário, uma vez que esta funcionalidade apenas está disponível se o utilizador clicar para ver o questionário. Além disto, a edição das questões também não é totalmente intuitiva, principalmente para utilizadores iniciantes.

2.2.5 SoGoSurvey

A SoGoSurvey é uma plataforma para a criação de formulários de pesquisa e pode ser acessada em: <https://www.sogosurvey.com/>.

Construída para uma ampla gama de indústrias - desde empresas de serviços financeiros até organizações governamentais - ajuda a armazenar o *feedback* dos clientes e a realizar pesquisas do envolvimento dos funcionários [12]. À semelhança das anteriores, tem também uma variedade de planos, sendo o gratuito disponibilizado durante um período de tempo limitado e os restantes com um custo associado (Figura 2.28).

Plus	Pro	Premium	Looking for an enterprise solution?
\$ 25 /month Billed \$299 Annually	\$ 40 /month Billed \$480 Annually	\$ 99 /month Billed \$1188 Annually	MORE USERS. MORE FEATURES. MORE ANALYTICS.
STARTER PACKAGE Good for individuals running straightforward, simple surveys.	INTERMEDIATE RESEARCH PLAN Gain more from your research with expanded options.	SMALL BUSINESS PLAN Expand your potential with advanced features and support.	REQUEST A DEMO
FREE 10-DAY TRIAL (no credit card required)	FREE 10-DAY TRIAL (no credit card required)	FREE 10-DAY TRIAL (no credit card required)	
ALL BASIC FEATURES AND:	ALL PLUS FEATURES AND:	ALL PRO FEATURES AND:	ALL PREMIUM FEATURES AND:
<ul style="list-style-type: none"> ✓ Add your own survey logo 	<ul style="list-style-type: none"> ✓ More question types, including Image Choice and Grids 	<ul style="list-style-type: none"> ✓ Advanced reporting and analysis 	<ul style="list-style-type: none"> ✓ Branded URL
<ul style="list-style-type: none"> ✓ Customized thank you page 	<ul style="list-style-type: none"> ✓ Hide irrelevant questions and question skip logic 	<ul style="list-style-type: none"> ✓ Guaranteed anonymous surveys 	<ul style="list-style-type: none"> ✓ Multiple users for collaboration and enterprise deployment
<ul style="list-style-type: none"> ✓ Export data to Excel 	<ul style="list-style-type: none"> ✓ Full control of survey themes, colors, fonts and more 	<ul style="list-style-type: none"> ✓ Advanced skip logic 	<ul style="list-style-type: none"> ✓ Admin dashboard and user controls
<ul style="list-style-type: none"> ✓ Simple skip logic 	<ul style="list-style-type: none"> ✓ Graphical analysis of results ready to export to PowerPoint and Word 	<ul style="list-style-type: none"> ✓ Pre-fill known data into visible and hidden fields 	<ul style="list-style-type: none"> ✓ Powerful data segmentation and data visualization

Figura 2.28: Planos da SoGoSurvey

Fonte: *Printscreen* do Website do SoGoSurvey

Na figura 2.29 é apresentado o seu ecrã inicial.

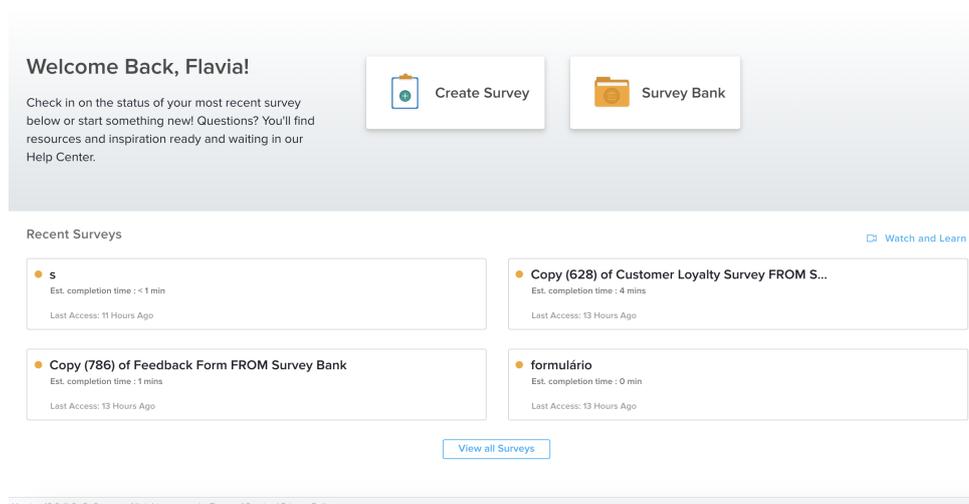


Figura 2.29: Ecrã inicial da SoGoSurvey

Fonte: *Printscreen* do Website do SoGoSurvey

Nesta página é apresentada a opção para criar novo formulário, escolher de entre os existentes e é também mostrada uma lista dos questionários já criados. Quando se opta por visualizar os modelos pré-criados verifica-se que estes estão distribuídos por 12 categorias, entre as quais avaliação, *feedback* e satisfação do cliente (Figura 2.30).

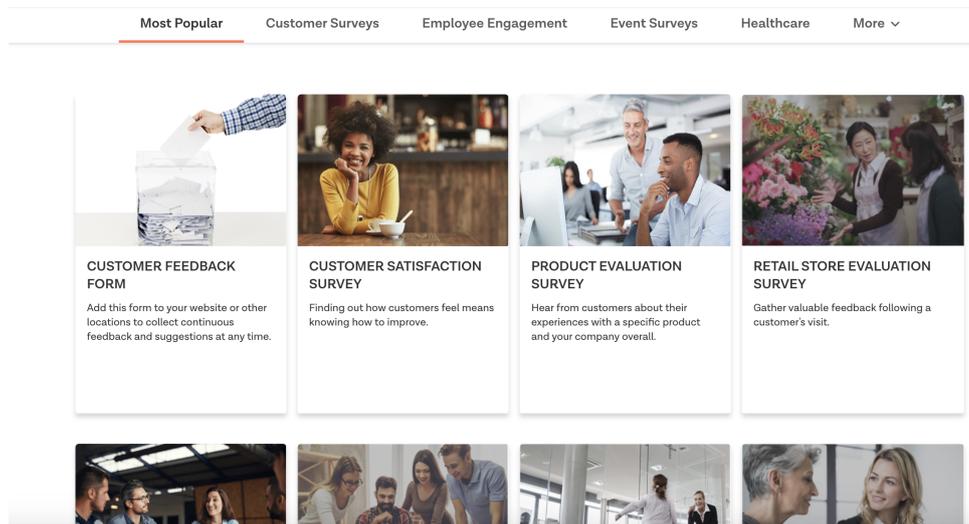


Figura 2.30: Modelos de formulários da SoGoSurvey

Fonte: *Printscreen* do Website do SoGoSurvey

Se o formulário for criado em branco, podem ser adicionados vários tipos de questões. Contudo, algumas das mais utilizadas em formulários de avaliação, como questões de classificação e escalas de opinião, não estão disponíveis no plano gratuito, como se verifica através da figura 2.31.

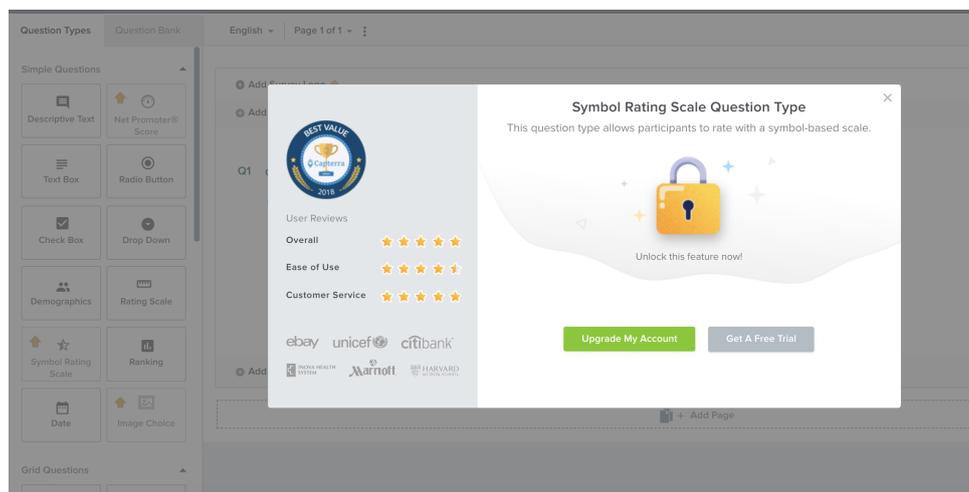


Figura 2.31: Tipos de questões

Fonte: *Printscreen* do Website do SoGoSurvey

Quando os formulários estão prontos para serem partilhados, é apresentada uma variedade de opções, incluindo email, redes sociais e *websites* (Figura 2.32).

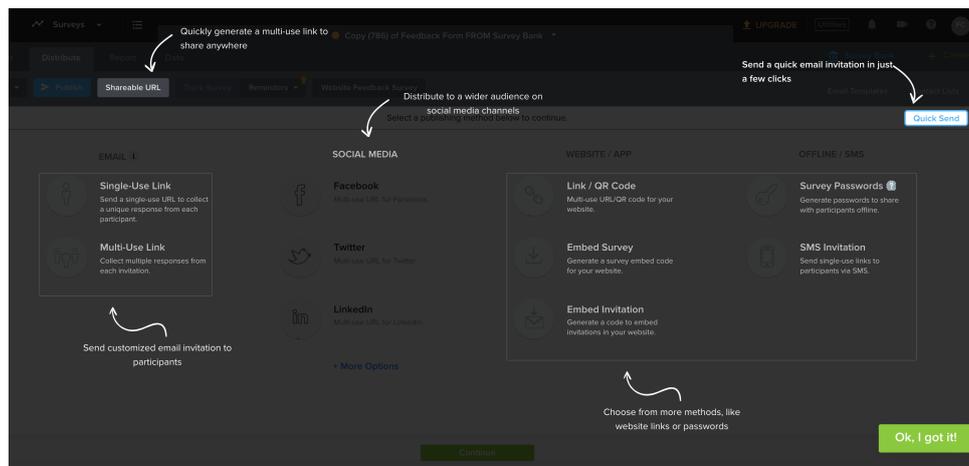


Figura 2.32: Meios de partilha de formulários no SoGoSurvey

Fonte: *Printscreen* do Website do SoGoSurvey

O ponto mais positivo desta aplicação é o facto das questões estarem organizadas em diferentes grupos, o que torna mais fácil a sua identificação e seleção de acordo com a utilização pretendida.

Como ponto negativo, considera-se que a plataforma não é totalmente intuitiva, uma vez nem todas as suas funcionalidades são facilmente identificáveis, como por exemplo, a opção para pesquisar os formulários ou a edição da informação.

2.3 Vetores de comparação

Tendo em conta a análise realizada anteriormente às diferentes soluções e o âmbito deste projeto, a aluna considerou os seguintes fatores, para comparar as diferentes plataformas:

- Usabilidade - Todas as funcionalidades da aplicação são fáceis de utilizar e identificáveis facilmente;
- Facilidade de criação de formulários - O processo de criação de formulários é simples e adequado às necessidades dos diferentes tipos de utilizadores;
- Design dos formulários - O design dos formulários contém funcionalidade combinada com uma apresentação elegante;
- Diversidade de questões - A plataforma permite adicionar questões de vários tipos, que são úteis na construção de formulários de avaliação, como por exemplo: Escalas de avaliação, escolha múltipla, classificação por estrelas, entre outras;
- Acesso aos formulários - Os questionários apenas são partilhados com as pessoas devidamente autorizadas para o efeito.

2.4 Análise comparativa das soluções

Nesta secção é apresentada uma análise detalhada das soluções descritas anteriormente, tendo em conta os vetores de comparação definidos na secção 2.3 e o âmbito deste projeto.

Na tabela 2.1 é atribuída uma pontuação para cada uma das características. Para tal, a aluna utilizou uma escala de *Likert* com valores de 1 a 5, em que "1" representa a pontuação mais baixa e "5" a pontuação mais alta.

Tabela 2.1: Avaliação das diferentes soluções

Fatores a avaliar/ Plataforma	Typeform	Google Forms	Survey Monkey	ZohoSurvey	SoGoSurvey
Usabilidade	3	4	2	2	3
Facilidade de criação de formulários	3	3	4	4	3
Design de formulários	4	3	3	3	3
Diversidade de questões	3	2	4	5	3
Acesso aos formulários	2	2	4	2	2

Feita a avaliação dos principais pontos presentes em cada plataforma analisada, pode verificar-se que cada uma possui características próprias e únicas que a distinguem das restantes.

Relativamente à usabilidade, considerou-se que a Google Forms é a solução que mais se destaca, uma vez que é uma plataforma que não exige muitos conhecimentos e procedimentos para ser utilizada, sendo assim adequada para vários tipos de utilizadores, incluindo os iniciantes que procuram criar formulários simples sem investir muito tempo.

Em seguida analisou-se a facilidade de criação de formulários e concluiu-se que a SurveyMonkey e a ZohoSurvey são as plataformas que mais investiram neste aspeto, uma vez que permitem criar formulários facilmente, de uma forma simples e intuitiva.

Analisando o design dos formulários criados, conclui-se que a Typeform é a plataforma que apresenta questionários com um design mais elegante. No entanto, a plataforma não apresenta a opção para criar formulários com todas as questões na mesma página, o que pode ser considerada uma desvantagem em determinadas situações.

Relativamente à diversidade de questões que podem ser adicionadas, é a ZohoSurvey a mais valorizada, pois contém mais tipos de questões do que as restantes.

Por fim, relativamente ao acesso aos formulários, é a SurveyMonkey que mais se destaca, uma vez que apenas permite a partilha de formulários através de um endereço de email, evitando assim a partilha com pessoas não autorizadas.

Após a análise das características destes sistemas, concluiu-se que todos apresentam semelhanças entre si. No entanto, também têm características únicas que lhes conferem vantagens ou desvantagens, de acordo com as preferências de cada utilizador.

Uma vez que a Talkdesk desenvolve as suas próprias soluções de *software* para serem utilizadas unicamente em *call centers* e que estas contém requisitos específicos, as plataformas analisadas anteriormente não são adequadas para a sua utilização. Assim, surgiu a necessidade de desenvolver uma nova solução direcionada para a criação de formulários de avaliação e que respeite os requisitos impostos pelo cliente. Posto isto, e utilizando os aspetos analisados como referência para o desenvolvimento da plataforma, procurou-se desenvolver uma aplicação que suportasse as seguintes características:

- Fácil de utilizar e intuitiva para qualquer utilizador;
- Permite criar formulários de avaliação, com questões específicas para esse efeito;
- Contém formulários com um design simples mas apelativo visualmente;
- Apenas permite a visualização e preenchimento de formulários a pessoas devidamente autorizadas.

A plataforma desenvolvida procurou melhorar algumas das falhas encontradas nas soluções analisadas anteriormente, principalmente na usabilidade e também na diversidade de questões que podem ser adicionadas aos formulários. Esta apresenta um conjunto diverso de elementos que podem ser utilizados, entre as quais: secções, para dividir as questões em diferentes grupos, texto, escolha múltipla, numéricas, data, tempo, opção única, múltiplas opções, escalas de classificação através da utilização de ícones, entre outras. Além disso, contém também requisitos específicos que a caracterizam e distinguem das restantes, como por exemplo: Apenas é permitida a partilha de formulários com utilizadores devidamente autorizados para o efeito; Não é possível criar um formulário sem nome ou descrição ou com um nome já existente; Os formulários são eliminados automaticamente depois de publicados, para que não possam ser modificados.

Capítulo 3

Abordagem de Desenvolvimento

Neste capítulo é descrita a abordagem seguida durante o desenvolvimento deste projeto. Na secção 3.1 é apresentada a constituição da equipa. Em seguida, na secção 3.2, é ilustrado o processo de desenvolvimento, isto é, o ciclo de vida utilizado bem como a fundamentação dessa escolha. Na secção 3.3, é apresentada a metodologia seguida para o controlo de versões de *software*. Por último, na secção 3.4, é apresentado o planeamento do estágio e a execução real, isto é, como foi feita a divisão das tarefas ao longo dos dois semestres e a duração das mesmas.

3.1 Equipa do projeto

A equipa envolvida neste projeto é constituída pela aluna estagiária, Flávia Costa, e pelos orientadores da Talkdesk, os engenheiros André Simões e Daniel Afonso.

A aluna ficou responsável por executar todos os processos de desenvolvimento de *software* necessários para a criação da aplicação, bem como a realização de testes aos diversos módulos implementados.

3.2 Ciclo de vida

Para a gestão e planeamento do projeto utilizou-se uma metodologia ágil baseada no Scrum [34]. As razões para a escolha desta metodologia encontram-se descritas seguidamente:

- O processo de desenvolvimento do *software* foi dividido em várias iterações;
- O projeto contém requisitos que poderiam ser alterados ao longo do tempo;
- O Scrum é a metodologia ágil utilizada na Talkdesk.

A figura 3.1 ilustra o funcionamento da metodologia Scrum.

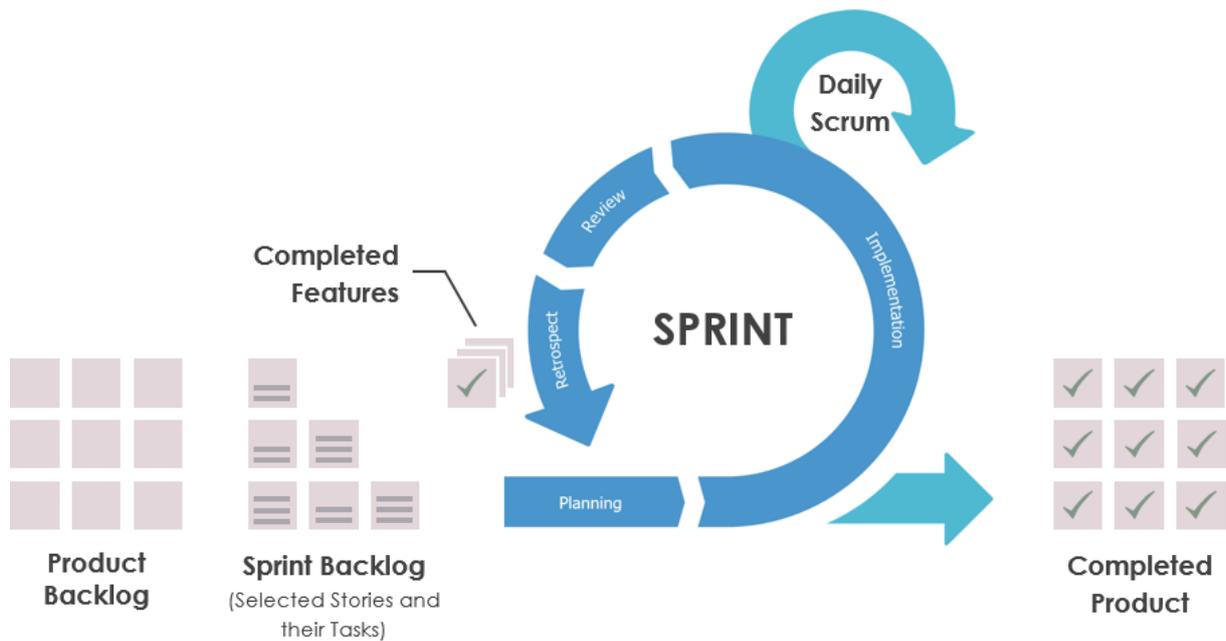


Figura 3.1: Funcionamento do Scrum

Fonte: Visual Paradigm

De acordo com os princípios estabelecidos por esta metodologia, todas as funcionalidades necessárias para o desenvolvimento da plataforma foram identificadas numa lista, conhecida por **Product Backlog**. As tarefas definidas no *Product Backlog* foram distribuídas em vários ciclos, também conhecidos por **Sprints**. Inicialmente, estava previsto que *sprints* deste projeto tivessem a duração de 5 dias úteis. Contudo, uma vez que existiram algumas dificuldades na aprendizagem das tecnologias, as *sprints* tiveram a duração de duas semanas.

De acordo com a metodologia Scrum, é o **Product Owner** que é normalmente o responsável por priorizar as tarefas definidas no backlog. No entanto, neste projeto a priorização foi realizada pela estagiária.

O desenvolvimento do *software* foi assim dividido em diferentes fases, de acordo com os vários momentos da metodologia Scrum:

- **Planeamento da Sprint**

A fase de planeamento de cada *sprint* foi muito importante no desenvolvimento do projeto, pois através desta definiram-se as tarefas a ser realizadas, bem como a sua prioridade. Nesta fase existiram reuniões entre a estagiária e os orientadores da Talkdesk. Estas foram divididas em duas partes: Na primeira parte, a aluna descreveu o objetivo da *sprint* e quais os itens do *product backlog* que contribuíam para esse objetivo. Em seguida, decidiu o que podia ser feito para que o objetivo fosse atingido.

- **Durante a Sprint**

Ao longo de cada *Sprint* foram realizadas reuniões, também conhecidas como **Daily Scrum**, com o orientador Daniel Afonso. Estas tiveram o objetivo de avaliar o progresso do desenvolvimento, identificar eventuais impedimentos e definir o trabalho para o dia seguinte [34].

- **Revisão da Sprint**

No fim de cada *sprint*, a estagiária apresentou as funcionalidades desenvolvidas.

3.3 Controlo de versões de *software*

Para estruturação e controlo de versões do código desenvolvido foi utilizado o Git [26]. O Git é um sistema flexível e poderoso, que permite gerir e acompanhar o desenvolvimento de um projeto [26]. Nos seus diretórios são registadas todas as alterações e versões de um determinado *software*.

Uma das suas funcionalidades é a possibilidade de se poder fazer o controlo de versões de modo colaborativo, ou seja, é possível que o mesmo ficheiro seja editado simultaneamente por dois desenvolvedores diferentes e que ambas as alterações sejam guardadas.

Na figura 3.2, encontra-se representado o funcionamento genérico do Git.

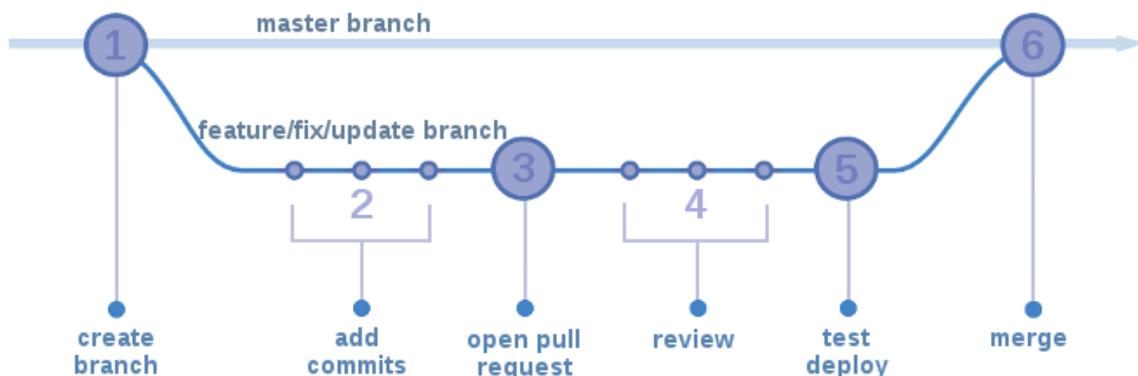


Figura 3.2: Funcionamento do Git

Fonte: *Git Workflow* - arccwiki

Como se vê na figura anterior, o Git permite a criação de **branches** (em português, ramificações). Para este projeto, foram criadas diferentes *branches*, ao longo do desenvolvimento de cada funcionalidade. Isto permitiu que o conteúdo de um repositório fosse divergido em várias direções [26], para que fosse possível o desenvolvimento de novas funcionalidades, mantendo-as separadas umas das outras.

Através dos *commits* foram adicionadas as alterações nos respetivos *branches*. Em seguida, foi criado um novo *pull request*, para descrever a funcionalidade implementada e as alterações efetuadas desde o último *commit*. A partir desta funcionalidade, o código foi revisto pelo engenheiro Daniel Afonso, para assim determinar se estava de acordo com os padrões definidos para o projeto. Se estivesse, podia então ser feito o *test deploy* da aplicação e por fim podia ser adicionado ao *master branch*, o *branch* padrão do projeto.

No capítulo 7, Qualidade do Código, é descrito em detalhe todo o processo realizado através do Git durante a implementação do *software*.

3.4 Plano de estágio

Nesta secção é apresentado o planeamento do estágio, através dos diagramas de Gantt. A partir destes é possível identificar as várias tarefas realizadas, bem como a duração das mesmas. Todos os diagramas foram elaborados a partir da ferramenta *Office Timeline Online*, disponível em: <https://online.officetimeline.com>.

A planificação dos dois semestres foi elaborada maioritariamente pela aluna estagiária, tendo sido feitas pequenas alterações por parte dos orientadores da Talkdesk.

3.4.1 Primeiro semestre

Esta secção apresenta os diagramas de Gantt relativos ao primeiro semestre. A figura 3.3 ilustra o planeamento inicial e a figura 3.4 contém a execução real.



Figura 3.3: Planeamento inicial - 1º semestre

- **Início do estágio** - 16 de Setembro de 2019
- **Onboarding** - Introdução aos processos existentes na empresa através de sessões de esclarecimento acerca dos produtos que esta desenvolve, além de formações relativas às tecnologias que são usadas.
- **Especificação do projeto e planeamento do 1º semestre** - Reuniões com os orientadores para especificar os detalhes do projeto e definir as várias etapas a serem realizadas ao longo do semestre.
- **Análise e aprendizagem das tecnologias** - Estudo e aprendizagem das várias tecnologias que serão utilizadas no desenvolvimento do projeto.
- **Estado de arte** - Estudo e comparação das soluções existentes no mercado.
- **Levantamento de requisitos** - Levantamentos dos requisitos funcionais e não funcionais do sistema.
- **Elaboração de protótipos** - Elaboração de protótipos que representam as interfaces a serem desenvolvidas no 2º semestre.
- **Definição da arquitetura** - Identificação dos componentes da plataforma e as suas relações com outros *softwares*.
- **Escrita do relatório intermédio** - Escrita e revisão dos vários capítulos do relatório intermédio.



Figura 3.4: Execução real - 1º semestre

Através da figura 3.4, é possível ver que ocorreram alguns desvios face ao plano previsto inicialmente, nomeadamente no tempo necessário para a concretização de algumas tarefas.

Seguidamente, apresentam-se as justificações para a ocorrência desses desvios:

- **Especificação do projeto e planeamento do 1º semestre** - Os detalhes do projeto só foram definidos com clareza no dia 1 de Novembro, pois ocorreram várias alterações na proposta de estágio por parte da Talkdesk.
- **Análise e aprendizagem das tecnologias** - A fase de aprendizagem das tecnologias demorou mais do que o previsto e decorreu durante quase todo o semestre, uma vez que a estagiária sentiu algumas dificuldades à medida que ia realizando os exercícios propostos pelo orientador, André Simões.
- **Estado de arte** - O estado de arte demorou mais do que o previsto, uma vez que foi necessário analisar diferentes soluções e seleccionar apenas as mais relevantes para o âmbito deste projeto.
- **Levantamento de requisitos e elaboração de protótipos** - As etapas de levantamento de requisitos e elaboração de protótipos demoraram menos do que o previsto, uma vez que após o estado de arte estar concluído e os detalhes do projeto estarem bem definidos foi mais fácil determinar quais eram os requisitos necessários e assim esquematizar as interfaces da aplicação.
- **Definição da arquitetura preliminar do sistema** - Esta etapa demorou menos tempo do que o previsto, uma vez que ainda só foi definida uma versão preliminar da arquitetura.

3.4.2 Segundo semestre

Esta secção apresenta os diagramas de Gantt relativos ao segundo semestre. A figura 3.5 ilustra o planeamento inicial e a figura 3.6 contém a execução real.

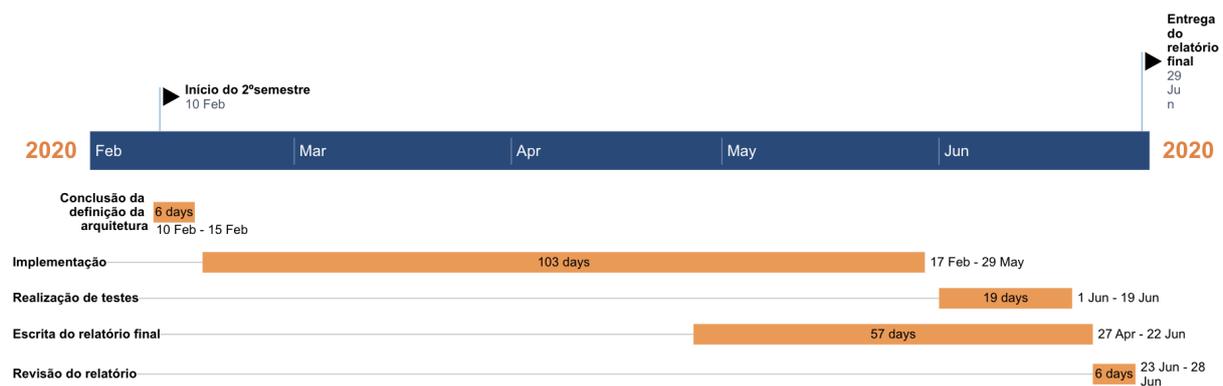


Figura 3.5: Planeamento inicial - 2º semestre

- **Conclusão da definição da arquitetura** - Especificação mais detalhada da arquitetura e criação do modelo de dados do sistema.
- **Implementação** - Desenvolvimento dos vários módulos da aplicação. Esta etapa corresponde a 15 sprints.

- **Realização de testes** - Fase para implementação de testes com o objetivo de validar as funcionalidades implementadas.
- **Escrita do relatório final** - Escrita dos restantes capítulos do relatório e revisão dos capítulos escritos anteriormente.
- **Revisão do relatório** - Fase para deteção e revisão de eventuais erros e inconsistências no relatório.



Figura 3.6: Execução real - 2º semestre

Também no 2º semestre, ocorreram alguns desvios face ao planeado inicialmente, essencialmente no tempo necessário para a conclusão de tarefas. As justificações para a ocorrência dessas oscilações encontram-se descritas em seguida:

- **Implementação** - As *sprints* tiveram a duração de duas semanas, ao invés de apenas uma como foi definido no planeamento do 2º semestre, uma vez que ocorreram algumas dificuldades na aprendizagem das tecnologias e nem sempre existiu disponibilidade, por parte dos restantes membros da equipa para verificar o trabalho desenvolvido. No entanto, para a conclusão desta fase foi necessária apenas mais uma semana do que o previsto inicialmente.
Esta etapa correspondeu a 8 sprints.
- **Realização de testes** - Considerou-se importante realizar testes unitários ao longo do semestre e simultaneamente com a fase de desenvolvimento e não apenas após a solução estar completamente desenvolvida, tal como previsto inicialmente. Assim, estes foram desenvolvidos a partir da conclusão da implementação da primeira funcionalidade. Posteriormente, quando se concluiu o desenvolvimento do projeto, realizaram-se também testes de sistema, para garantir que a plataforma já completamente integrada cumpre com os requisitos especificados.
- **Elaboração do manual de utilizador** - Esta tarefa foi acrescentada ao cronograma, uma vez que se considerou essencial, o desenvolvimento de um manual de utilizador que ilustre o funcionamento da aplicação.
- **Escrita do relatório final** - Esta etapa demorou mais tempo do que o planeado, uma vez que o relatório final foi escrito ao longo do semestre e simultaneamente com a fase de desenvolvimento.

- **Revisão do relatório** - Para esta tarefa foi necessário mais tempo, uma vez que se considerou importante enviar, ao longo do semestre, os diversos capítulos ao professor Dr. Paulo Rupino para assim obter um maior *feedback*.

3.5 Análise de Riscos

Qualquer processo de desenvolvimento de *software* pode envolver alguns fatores de risco. Assim, é importante identificá-los previamente e tomar medidas que permitam reduzir o seu impacto. A presente secção descreve a análise de riscos realizada para este projeto.

Todos os riscos identificados contém uma probabilidade de ocorrência, o impacto que podem ter no projeto e o momento em que foram identificados. Esta classificação foi efetuada de acordo com a percepção da estagiária. Para cada um dos riscos existe também um plano de mitigação, constituído por medidas preventivas e/ou corretivas para reduzir o seu impacto.

A probabilidade de ocorrência de um risco pode ser classificada de acordo com os seguintes níveis [27]:

- Probabilidade quase certa;
- Probabilidade elevada;
- Probabilidade média;
- Probabilidade baixa.

O impacto de um risco pode também ser classificado de acordo com os seguintes níveis [27]:

- Impacto catastrófico;
- Impacto crítico;
- Impacto moderado;
- Impacto insignificante.

Uma forma possível de categorizar os riscos, tendo em conta a sua probabilidade e impacto é através de uma matriz de exposição. Numa matriz de exposição, quanto maior for a probabilidade e impacto de um risco, maior serão as consequências negativas no projeto.

A figura 3.7 representa a matriz utilizada durante a análise dos riscos, sendo que os que se encontram numa posição de cor vermelha e laranja podem causar um impacto mais negativo no sucesso do projeto e a verde e amarelo o inverso.

Impacto \ Probabilidade	Insignificante	Moderado	Crítico	Catastrófico
Quase certa				
Elevada				
Média				
Baixa				

Figura 3.7: Matriz de probabilidade/impacto

3.5.1 Lista de riscos

Os riscos foram identificados ao longo do ciclo de vida do projeto e encontram-se identificados em seguida.

Risco 01: Pouca experiência na utilização das tecnologias

- **Descrição** - A estagiária não tem experiência com algumas das tecnologias utilizadas, como o React.js e Jest, pelo que poderão existir atrasos na concretização das tarefas e consequente atraso do projeto.
- **Probabilidade** - Elevada
- **Impacto** - Crítico
- **Início do risco** - Primeiro semestre.
- **Mitigação** - Realização de tutoriais, visualização de exemplos e esclarecimento regular das dúvidas com os colegas da Talkdesk experientes na utilização das tecnologias em causa.
- **Estado do Risco** - Este risco ocorreu durante a maior parte do segundo semestre, uma vez que a falta de experiência na utilização das tecnologias provocou alguns atrasos na conclusão das *sprints* e exigiu que fosse feito um esforço adicional por parte da estagiária.

Risco 02: Pouca experiência em design de *User Interface*.

- **Descrição** - A estagiária possui pouca experiência em design de *User Interface*, o que pode levar à existência de falhas durante a experiência do utilizador com a aplicação.
- **Probabilidade** - Média
- **Impacto** - Crítico
- **Identificação do risco** - Primeiro semestre.
- **Mitigação** - Utilizar, como referência, bibliotecas com padrões de design e layouts bem definidos, tais como: Material-UI, Materialize ou Bootstrap.
- **Estado do Risco** - Este risco ocorreu até ao início do 2º semestre e foi ultrapassado a partir da ação de mitigação definida anteriormente.

Risco 03: Alteração de funcionalidades da aplicação

- **Descrição** - Necessidade de alteração de funcionalidades e características relevantes do produto.
- **Probabilidade** - Baixa
- **Impacto** - Moderado
- **Identificação do risco** - Início do 2º semestre
- **Mitigação** - Comunicar com o cliente para determinar quais são os requisitos prioritários que devem ser implementados para garantir o sucesso do projeto.
- **Estado do Risco** - Este risco foi ultrapassado após a primeira reunião de planeamento entre a estagiária e os orientadores. Aqui, todas as funcionalidades a implementar foram definidas, com clareza, no *product backlog*.

Risco 04: Tempo de desenvolvimento dos requisitos, testes e finalização do relatório.

- **Descrição** - É necessário conciliar o desenvolvimento dos requisitos com a realização de testes e finalização do relatório final, pelo que poderão existir incoerências ou erros no desenvolvimento.
- **Probabilidade** - Média
- **Impacto** - Crítico
- **Identificação do risco** - Início do 2º semestre.
- **Mitigação** - Definir um planeamento inicial e desenvolver os requisitos de acordo com a sua prioridade, garantindo espaço temporal para a finalização deste documento.
- **Estado do Risco** - Este risco ocorreu ao longo do semestre mas foi ultrapassado a partir da ação de mitigação definida.

Os riscos identificados, foram colocados na matriz de probabilidade/impacto, resultando na distribuição ilustrada na figura 3.8.

Impacto \ Probabilidade	Insignificante	Moderado	Crítico	Catastrófico
Quase certa				
Elevada			Risco 1	
Média			Risco 2 Risco 4	
Baixa		Risco 3		

Figura 3.8: Matriz de probabilidade/impacto dos riscos identificados

A partir desta matriz foram definidas prioridades na mitigação dos riscos. Verifica-se que o risco nº1 foi o que teve uma estratégia de mitigação mais prioritária, pois é o risco com a probabilidade mais elevada e poderia causar um impacto crítico no sucesso do projeto. O risco nº3 foi o menos prioritário, uma vez que a probabilidade da sua ocorrência é baixa e o impacto no projeto é moderado.

Capítulo 4

Requisitos

Este capítulo apresenta o processo realizado durante o levantamento dos requisitos da aplicação. Na secção 4.1 são identificadas as motivações que permitiram a sua identificação. Na secção 4.2 são listados todos os requisitos funcionais, as suas prioridades e as fundamentações dessa escolha. Em seguida, na secção 4.3, são apresentados os requisitos não funcionais e as restrições técnicas. Na secção 4.4 são apresentados os diagramas de casos-de-uso e as suas interações com os atores do sistema. Por fim, na secção 4.5, é descrita a metodologia seguida para a especificação dos requisitos.

4.1 Levantamento de requisitos

O processo de levantamento de requisitos foi efetuado na Talkdesk, a partir das necessidades do cliente (*product owner*) e das lacunas existentes nas soluções analisadas no estudo do estado de arte. Após várias reuniões e análise da documentação por este disponibilizada foi iniciada a elaboração do documento de requisitos.

À medida que os requisitos iam sendo adicionados eram também validados, tendo-se efetuado alguns ajustes e correções. Terminada a sua identificação percebeu-se que alguns dos que tinham sido inicialmente identificados deviam ser removidos uma vez que eram desnecessários e outros deviam ser adicionados para acrescentar mais valor à solução.

4.2 Requisitos funcionais

Os requisitos funcionais indicam o que o "sistema deve fazer" [25], isto é, as funcionalidades que deve conter.

A tabela 4.1 apresenta a lista de todas as funcionalidades identificadas para este projeto, bem como a sua prioridade.

Tabela 4.1: Lista de requisitos funcionais

Requisito	Prioridade
RF01 - Efetuar login	MUST HAVE
RF02 - Efetuar logout	MUST HAVE
RF03 - Criar novo formulário	MUST HAVE
RF04 - Adicionar elementos ao formulário	MUST HAVE
RF05 - Eliminar elementos do formulário	MUST HAVE
RF06 - Duplicar elementos do formulário	SHOULD HAVE
RF07 - Adicionar opções de resposta	MUST HAVE
RF08 - Eliminar opções de resposta	MUST HAVE
RF09 - Pré-visualizar formulário	MUST HAVE
RF010 - Guardar formulários	MUST HAVE
RF011 - Consultar formulários	MUST HAVE
RF012 - Abrir determinado formulário	SHOULD HAVE
RF013 - Apagar formulários	MUST HAVE
RF014 - Editar nome e/ou descrição dos formulários	MUST HAVE
RF015 - Filtrar formulários	SHOULD HAVE
RF016 - Publicar formulários	MUST HAVE
RF017 - Consultar formulários publicados	MUST HAVE
RF018 - Preencher formulários	MUST HAVE
RF019 - Visualizar modelos de formulários	WON'T HAVE
RF020 - Submeter formulários	WON'T HAVE

Por forma a estabelecer a prioridade das funcionalidades, foi utilizada a técnica de análise de MoScow [30]. Esta consiste na atribuição de uma prioridade a cada requisito, de acordo com as seguintes categorias:

- **Must Have (Deve ter):** Representa tudo o que é imprescindível para o desenvolvimento do projeto, isto é, as funcionalidades mais importantes, para que a solução seja considerada um sucesso.
- **Should Have (Deveria ter):** Representa as funcionalidades que deveriam ser incluídas no projeto, mas que não são imprescindíveis e que se por ventura não forem implementadas, não farão com que o produto final perca o seu valor.
- **Could Have (Poderia ter):** Requisitos que são considerados desejáveis, mas não são necessários, e que serão incluídos caso o tempo e os recursos permitam.

- **Won't Have (Não terá por enquanto)**: Funcionalidades que não serão desenvolvidas por enquanto, mas que podem ser consideradas no futuro.

De acordo com a tabela 4.1, os requisitos deste projeto estão distribuídos em três categorias: **Essenciais** (Têm de ser implementados para o sucesso do projeto), **Importantes** (Podem ser implementados mas não são essenciais) e **Dispensáveis** (Não serão implementados agora, mas podem ser considerados no futuro).

Os critérios utilizados para a sua priorização foram os seguintes:

- Cliente - Através de várias reuniões com o *product owner* foi possível identificar quais eram as funcionalidades mais importantes para o negócio e as que deveriam ser implementadas primeiro;
- Riscos - A identificação dos riscos para este projeto também foi um critério decisivo, uma vez que permitiu compreender que é mais importante dar prioridade ao que é absolutamente necessário para o sistema funcionar, de forma a maximizar a probabilidade de sucesso do projeto.

4.3 Requisitos Não funcionais e restrições técnicas

Requisitos que "expressam condições ou qualidades específicas que o sistema deve atender" [18]. Seguidamente apresentam-se os requisitos não funcionais deste projeto, distribuídos pelas seguintes categorias: Usabilidade, Segurança, Confiabilidade e Interoperabilidade, bem como as restrições técnicas.

4.3.1 Requisitos do produto

Requisitos que descrevem "propriedades de um sistema ou produto" [11].

- **Usabilidade:**
 - RNF01: O sistema deve ser fácil de usar, isto é, os utilizadores devem conseguir utilizar as várias funcionalidades da aplicação sem muitas dificuldades após um período de treino.
- **Segurança:**
 - RNF02: O sistema não deve mostrar dados pessoais do utilizador, tais como a palavra-passe.
- **Confiabilidade:**
 - RNF03: O sistema deve conter todas as funcionalidades *must have* especificadas na lista de requisitos.

4.3.2 Requisitos externos

Decorrem de fatores externos ao sistema e ao processo de desenvolvimento [11].

- **Interoperabilidade**

- RNF04: O sistema deverá comunicar com a base de dados.

4.3.3 Restrições técnicas

Restrições estabelecidas ainda no domínio do problema e que não devem ser modificadas ao longo do desenvolvimento da solução [38].

- O sistema deverá ser desenvolvido utilizando a linguagem JavaScript.

-O sistema deverá ser desenvolvido utilizando a biblioteca React.Js.

4.4 Diagramas de casos-de-uso

Os casos de uso descrevem o comportamento que o sistema deverá apresentar na interação com os seus utilizadores (atores e entidades externas) [23].

Neste sistema existem dois tipos de atores: O Administrador e o Supervisor. Ambos podem utilizar a aplicação, contudo têm permissões distintas:

O Administrador pode aceder às funcionalidades que permitem criar e gerir formulários, enquanto o Supervisor apenas pode ver os formulários publicados e preenchê-los.

Nas figuras 4.1 e 4.2 são apresentados os casos de uso da aplicação e a interação destes com os atores.



Figura 4.1: Interação do sistema com o administrador

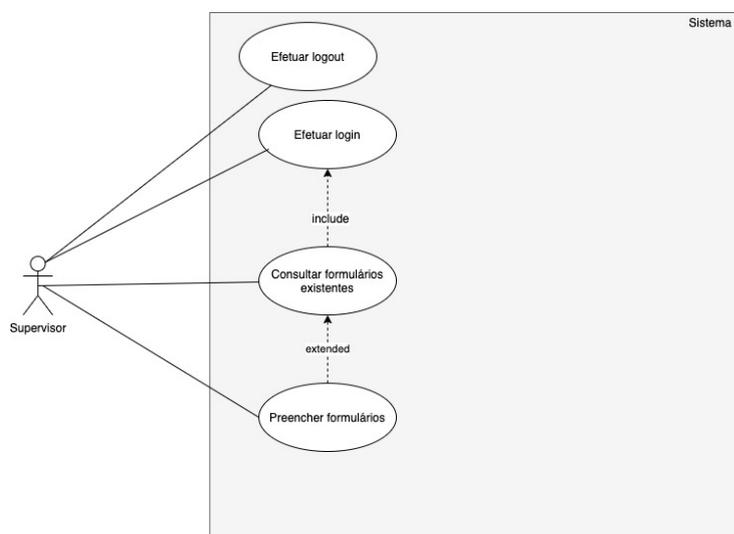


Figura 4.2: Interação do sistema com o supervisor

4.5 Especificação de requisitos

Esta secção descreve as técnicas utilizadas para especificação de requisitos.

Inicialmente, fez-se uma descrição detalhada dos casos de uso de modo a perceber como é que os atores interagem com o sistema e em seguida elaborou-se *mockups* que representam as várias interfaces da aplicação. Este processo foi acompanhado e validado pelos orientadores do estágio.

Cada caso de uso é descrito recorrendo aos seguintes campos, que ajudam a preencher a sua especificação:

- **Descrição:** Apresentação dos objetivos do caso de uso.
- **Atores:** Utilizadores que interagem diretamente com a aplicação.
- **Pré-Condições:** Condições que devem ser atendidas para que o caso de uso possa ser executado.
- **Fluxo esperado:** Sequência de eventos que mostram as interações entre o utilizador e o sistema. O cenário esperado descreve a situação em caso de sucesso.
- **Fluxo alternativo:** Pode ser utilizado para descrever cenários opcionais do caso de uso.
- **Pós-Condições:** Estado em que o sistema estará após a execução do caso de uso.

Em seguida é apresentado um exemplo de um caso de uso e de um *mockup*. Os restantes podem ser consultados, com mais detalhe, no Anexo A (Requisitos).

UC-04:	Adicionar elementos ao formulário
<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende adicionar elementos ao formulário.
<i>Atores:</i>	Administrador
<i>Pré-condições:</i>	Criar um formulário
<i>Fluxo esperado:</i>	<ol style="list-style-type: none"> 1. O sistema apresenta um menu com vários tipos de opções que podem ser adicionadas; 2. O utilizador escolhe o elemento que pretende adicionar; 3. O utilizador clica para adicionar o elemento ao formulário.
<i>Fluxo alternativo:</i>	
<i>Pós-condições:</i>	Em caso de sucesso, um novo elemento é adicionado ao formulário.

Figura 4.3: Caso de uso "Adicionar elementos ao formulário"

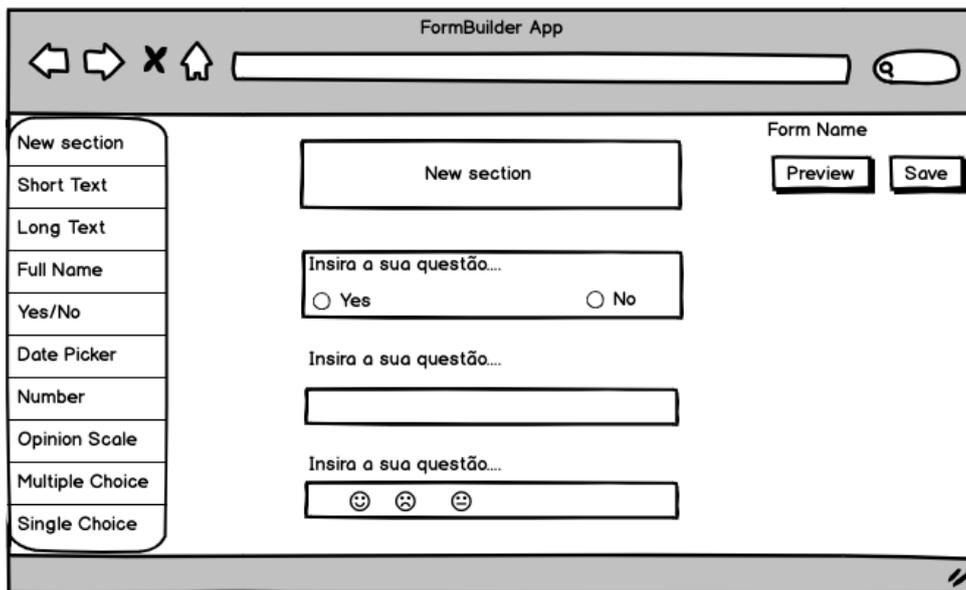


Figura 4.4: Elementos adicionados ao formulário: *New section*, *Yes/No*, *ShortText* e *OpinionScale*

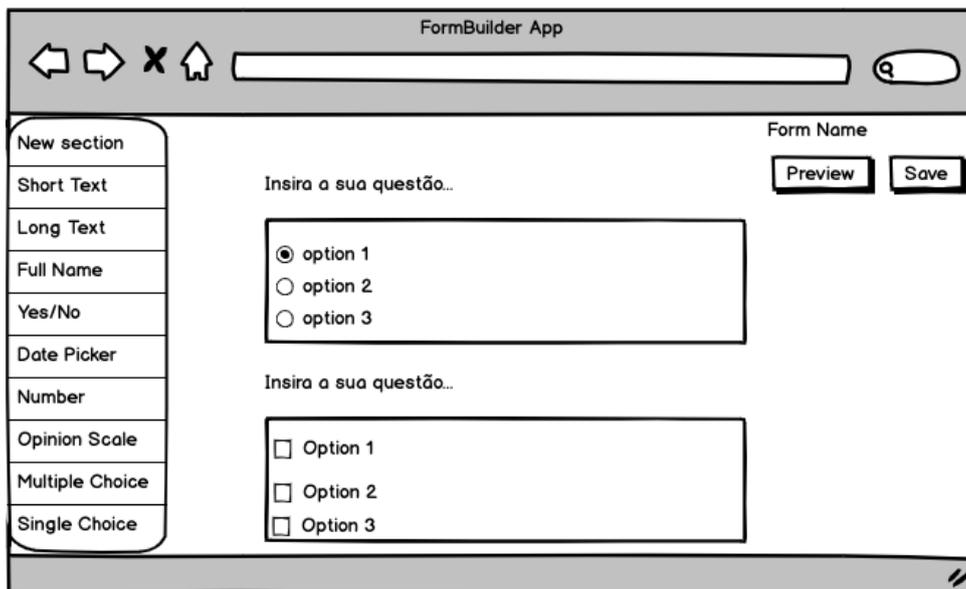


Figura 4.5: Elementos adicionados ao formulário: *SingleChoice* e *MultipleChoice*

Capítulo 5

Arquitetura e Modelo de dados

Este capítulo apresenta a arquitetura e o modelo de dados da aplicação desenvolvida. Na secção 5.1 são identificados os componentes da aplicação e é descrito o processo que permite que estes comuniquem entre si. Na secção 5.2 é apresentada a arquitetura da plataforma na *cloud*. Por fim, na secção 5.3, é apresentado o modelo de dados.

5.1 Arquitetura global da aplicação

Na figura 5.1 está representada a arquitetura global da aplicação, desenvolvida a partir do modelo arquitetural REST.

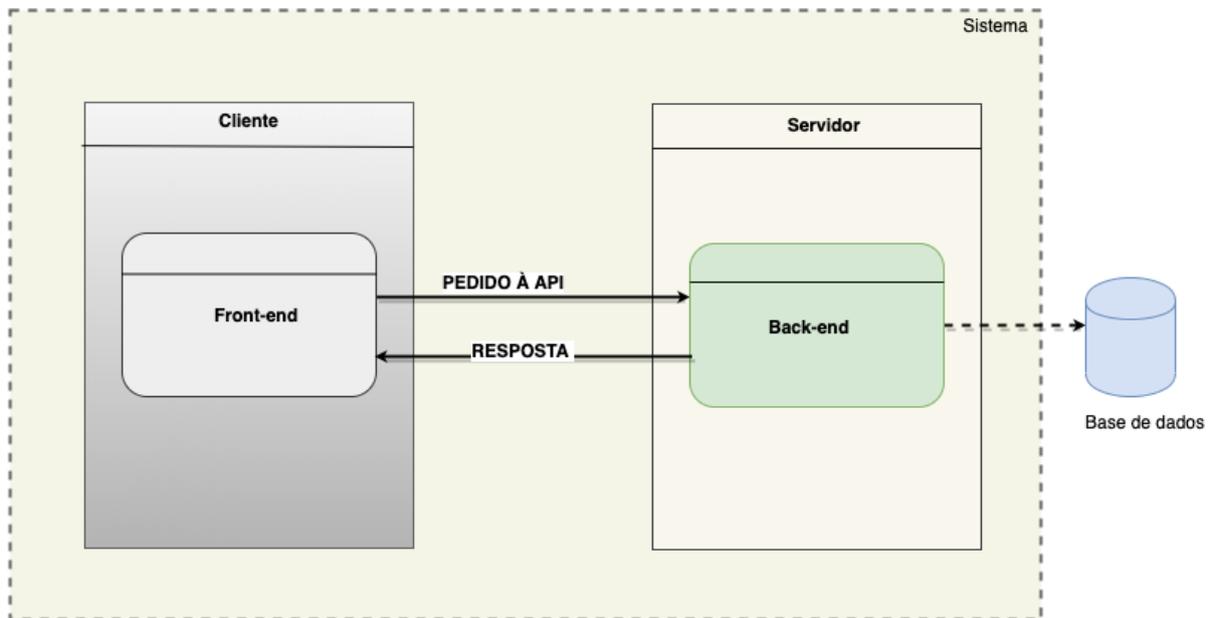


Figura 5.1: Representação da arquitetura do sistema

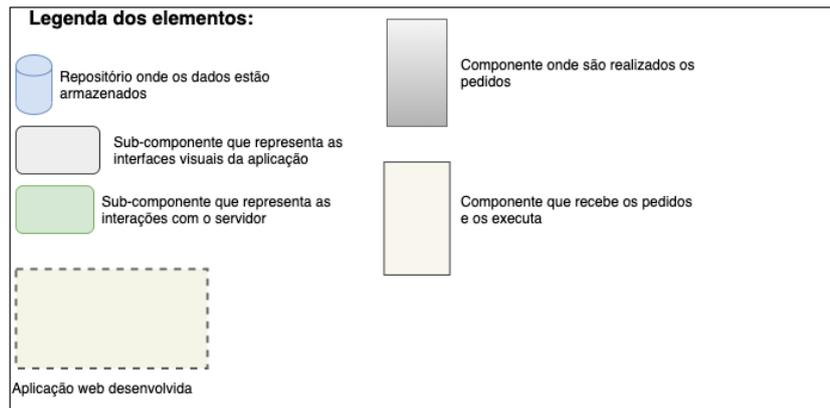


Figura 5.2: Legenda dos componentes da arquitetura

Uma das vantagens de se utilizar este tipo de arquitetura é a divisão da aplicação em duas camadas: o cliente e o servidor.

Como se observa na figura 5.1, o sistema desenvolvido possui duas camadas: o *front-end* e o *back-end*. A primeira representa o cliente e é composta pelas interfaces visuais da aplicação, com as quais o utilizador interage diretamente. A segunda camada (*back-end*) representa o servidor, componente que permite a execução da aplicação e a comunicação com a base de dados.

Neste tipo de aplicações, o servidor aguarda pelas requisições HTTP provenientes do cliente, executa-as e devolve uma resposta. Um mesmo cliente pode enviar vários pedidos para o servidor, contudo, cada um deles é independente, ou seja, cada requisição deve conter todas as informações necessárias para que o servidor consiga processá-la adequadamente.

Do lado do cliente, utilizou-se a biblioteca *React.js* para desenvolver as interfaces visuais da aplicação, uma vez que é a tecnologia utilizada na Talkdesk. Contudo, poderiam ser consideradas outras alternativas, tais como Angular ou Vue. Do lado do servidor, utilizou-se o JSON-Server, uma REST API para aceder aos dados, isto é, para ler, adicionar, atualizar e eliminar. A razão para a escolha desta tecnologia deve-se ao facto do âmbito do estágio ser desenvolvimento front-end e não existir um back-end previamente desenvolvido pela Talkdesk. Como o nome indica, uma API (*Application programming interface*) consiste numa interface que permite a utilização das funcionalidades desenvolvidas, sem que se conheça o código-fonte. Os dados estão armazenados num ficheiro JSON, que representa a base de dados do sistema.

A plataforma desenvolvida é uma *Single Page Application*, isto é, uma aplicação onde todas as funcionalidades estão concentradas numa única página [32]. Assim, a maior parte da lógica da interface do utilizador está no navegador, que comunica com o servidor por meio de requisições à API. O *front-end* é carregado de uma só vez no lado do cliente. Isso permite que este envie e receba apenas os dados do seu interesse. Ao invés da página toda ser recarregada sempre que há um novo pedido ou redirecionar o utilizador para uma página nova, "apenas o conteúdo principal é atualizado de forma assíncrona, mantendo toda a estrutura da página estática" [32]. O cliente obtém uma representação JSON dos dados solicitados, em vez de receber uma página HTML completa como resposta.

5.2 Arquitetura da aplicação na cloud

A figura 5.3 representa a arquitetura na *cloud*, isto é, o processo realizado para hospedar a aplicação online e disponibilizá-la como um serviço SAAS.

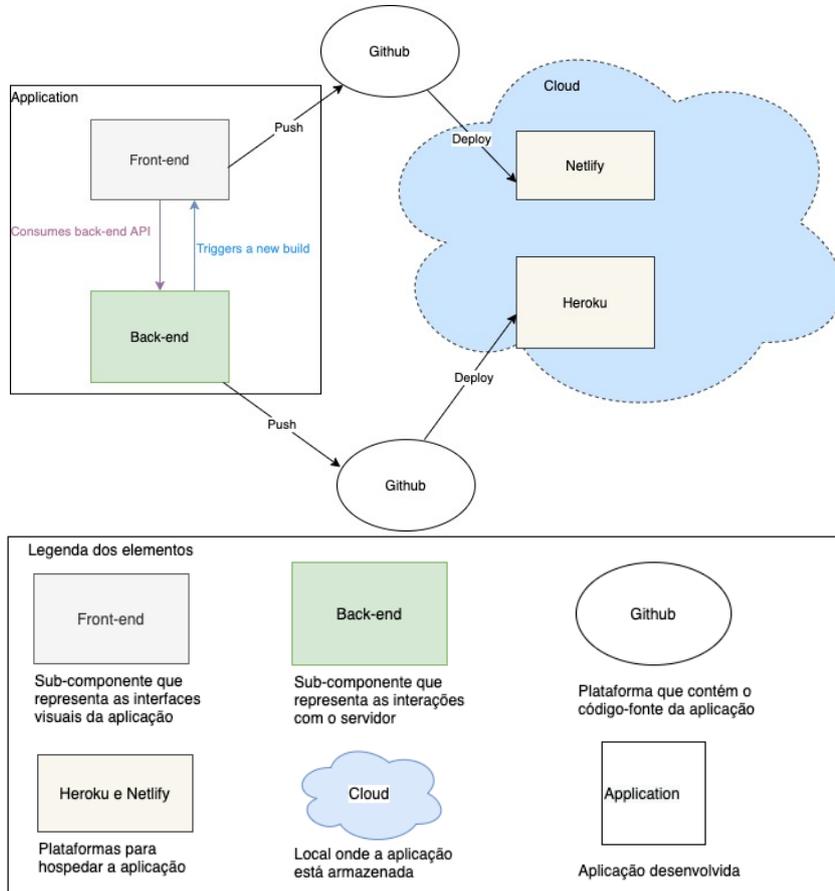


Figura 5.3: Representação da arquitetura da aplicação na cloud

Após o desenvolvimento da aplicação, foi necessário fazer o seu *deploy*, isto é, um conjunto de atividades que permitem que esta fique disponível para utilização. Como se vê na figura 5.3 e como referido anteriormente, o front-end e o back-end são módulos independentes, pelo que, optou-se por fazer o *deploy* das duas camadas em separado. Após alguma pesquisa e por sugestão do orientador André Simões, optou-se por utilizar a plataforma Netlify para hospedar o front-end da aplicação e o Heroku para hospedar o back-end. Estas são soluções que podem ser configuradas facilmente e integradas com o github, o que permite que as alterações, tanto das interfaces visuais como da base de dados sejam automaticamente enviadas e atualizadas.

Apesar de se ter optado pela escolha destas tecnologias devido às razões mencionadas anteriormente, poderiam ser utilizadas outras alternativas, tais como, *Firebase* ou *Amazon Web Services*.

5.3 Modelo de dados

Nesta secção é apresentado o modelo de dados da aplicação. A base de dados utilizada é do tipo não relacional, uma vez que toda a informação se encontra agrupada num único documento JSON e não existem relações entre tabelas.

Nas imagens seguintes, estão representados os elementos contidos no documento e é descrito o seu significado. Inicialmente, é apresentado o ficheiro db.json que contém os dados utilizados pela API REST (Figura 5.4).

```
{
  "forms": [],
  "publishedForms": [],
  "users": [
    {
      "id": 1,
      "usernameAdmin": "admin",
      "passwordAdmin": "admin",
      "usernameSupervisor": "supervisor",
      "passwordSupervisor": "supervisor"
    }
  ]
}
```

Figura 5.4: Ficheiro db.json

Quando são efetuadas solicitações POST, PUT, PATCH ou DELETE, as alterações são guardadas automaticamente neste ficheiro.

Em seguida, apresenta-se a descrição dos elementos mostrados na figura 5.4:

- forms: Representa os modelos de formulários criados pelo administrador.
- publishedForms: Representa os formulários publicados pelo administrador e que podem ser consultados pelo supervisor.
- users: Tipos de utilizadores que podem aceder à aplicação. Cada um tem permissões diferentes e é identificado pelo seu *username* e *password*.

A figura 5.5 representa a informação guardada nos formulários, isto é, os elementos contidos no campo "forms", no ficheiro db.json.

```
[
  {
    "name": "formulário de avaliação",
    "description": "Avaliar atendimento do agente.",
    "componentsState": [],
    "id": 1
  }
]
```

Figura 5.5: Exemplo da representação de um formulário

Cada formulário é constituído pelos seguintes elementos:

- name: Elemento que representa o nome do formulário. Este campo é de preenchimento obrigatório e apenas é permitida a criação de formulários com nomes que ainda não existam na base de dados, para que seja mais fácil identificá-los e distingui-los dos restantes.
- description: Campo, de preenchimento obrigatório, onde é armazenada a descrição do formulário.
- componentsState: Conjunto de elementos que podem ser adicionados ao formulário, isto é, o tipo de questões.

As figuras seguintes ilustram todos os elementos que podem ser adicionados aos formulários, bem como os atributos que os caracterizam.

Na figura 5.6, é apresentado um exemplo da criação de um elemento "Section". Este é utilizado para agrupar as questões em diferentes grupos, de acordo com as suas características, isto é, podem ser criadas secções para dividir as perguntas por temas, ou por uma ordem definida pelo administrador para avaliar cada agente. Cada formulário pode ter uma ou mais secções, de acordo com as preferências do utilizador.

```

"componentsState": [
  {
    "values": {
      "type": "newsection",
      "inputQuestion": "Nova secção ",
      "id": "New Section_0"
    },
    "id": "New Section_0"
  }
],

```

Figura 5.6: Representação de elemento do formulário - Section

Em seguida, apresenta-se a legenda dos atributos identificados na figura 5.6:

- values - Representa os valores que aquele elemento possui, isto é, o que é definido pelo utilizador nos campos de entrada.
- type - Tipo de elemento adicionado. Neste caso é do tipo "newsection", pois este elemento é uma secção e aqui apenas pode ser definido um título ou uma breve descrição, isto é, não é permitido adicionar respostas.
- id - Identificação do elemento

Na figura 5.7, é apresentado um exemplo da criação do elemento "Short Text". Este é utilizado para adicionar respostas curtas. Cada formulário pode ter um ou mais elementos "Short Text".

```
"componentsState": [  
  {  
    "values": {  
      "value": "Resposta ",  
      "type": "shorttext",  
      "inputQuestion": "Questão",  
      "id": "Short text_0"  
    },  
    "id": "Short text_0"  
  }  
],
```

Figura 5.7: Representação de elemento do formulário - Short Text

Na figura 5.8, é apresentado um exemplo da criação do elemento "Long Text". Este é utilizado para adicionar respostas longas. Cada formulário pode ter um ou mais elementos do tipo "Long Text".

```
"componentsState": [  
  {  
    "values": {  
      "value": "Resposta ",  
      "type": "longtext",  
      "inputQuestion": "Questão",  
      "id": "Long text_0"  
    },  
    "id": "Long text_0"  
  }  
],
```

Figura 5.8: Representação de elemento do formulário - Long Text

Na figura 5.9, é apresentado um exemplo da criação do elemento "Full Name". Este é utilizado para adicionar respostas que contém o primeiro e o último nome. Cada formulário pode ter um ou mais elementos do tipo "Full Name".

```

"componentsState": [
  {
    "values": {
      "value": [
        "Primeiro nome",
        "Último nome "
      ],
      "type": "fullname",
      "inputQuestion": "Questão",
      "id": "Full name_0"
    },
    "id": "Full name_0"
  }
],

```

Figura 5.9: Representação de elemento do formulário - Full name

Na figura 5.10, é apresentado um exemplo da criação do elemento "Yes/No". Este é utilizado para adicionar questões que podem ter duas opções de resposta: Sim ou Não. Cada formulário pode ter um ou mais elementos do tipo "Yes/No".

```

"componentsState": [
  {
    "values": {
      "value": 1,
      "type": "yesno",
      "inputQuestion": "Questão",
      "id": "Yes No_0"
    },
    "id": "Yes No_0"
  }
],

```

Figura 5.10: Representação de elemento do formulário - Yes/No

Na figura 5.11, é apresentado um exemplo da criação do elemento "Date Picker". Este é utilizado para adicionar questões que podem ter uma data como resposta. Cada formulário pode ter um ou mais elementos do tipo "Date Picker".

```

"componentsState": [
  {
    "values": {
      "value": "2020-01-01T10:30",
      "type": "datepicker",
      "inputQuestion": "Questão",
      "id": "Date Picker_0"
    },
    "id": "Date Picker_0"
  }
],

```

Figura 5.11: Representação de elemento do formulário - Date Picker

Na figura 5.12, é apresentado um exemplo da criação do elemento "Time". Este é utilizado para adicionar questões que podem conter uma escala temporal (horas, minutos e segundos). Cada formulário pode ter um ou mais elementos do tipo "Time".

```
"componentsState": [  
  {  
    "values": {  
      "value": "10:20",  
      "type": "time",  
      "inputQuestion": "Questão",  
      "id": "Time_0"  
    },  
    "id": "Time_0"  
  }  
],
```

Figura 5.12: Representação de elemento do formulário - Time

Na figura 5.13, é apresentado um exemplo da criação do elemento "Number". Este é utilizado para adicionar questões que apenas aceitam números como resposta. Cada formulário pode ter um ou mais elementos do tipo "Number".

```
"componentsState": [  
  {  
    "values": {  
      "value": "1",  
      "type": "number",  
      "inputQuestion": "Questão",  
      "id": "Number_0"  
    },  
    "id": "Number_0"  
  }  
],
```

Figura 5.13: Representação de elemento do formulário - Number

Na figura 5.14, é apresentado um exemplo da criação do elemento "Opinion Scale". Este é utilizado para adicionar questões que aceitam respostas com *emojis* para definir o índice de satisfação do utilizador. Cada formulário pode ter um ou mais elementos do tipo "Opinion Scale".

```
"componentsState": [  
  {  
    "values": {  
      "value": "4",  
      "type": "opinionscale",  
      "inputQuestion": "Questão",  
      "id": "Opinion Scale_0"  
    },  
    "id": "Opinion Scale_0"  
  }  
],
```

Figura 5.14: Representação de elemento do formulário - Opinion Scale

Na figura 5.15, é apresentado um exemplo da criação do elemento "Star Rating". Este é utilizado para adicionar questões que aceitam respostas com ícones de estrelas para definir o índice de satisfação do utilizador. Cada formulário pode ter um ou mais elementos do tipo "Star Rating".

```
"componentsState": [  
  {  
    "values": {  
      "value": "3",  
      "type": "starrating",  
      "id": "Star Rating_0"  
    },  
    "id": "Star Rating_0"  
  }  
],
```

Figura 5.15: Representação de elemento do formulário - Star Rating

Na figura 5.16, é apresentado um exemplo da criação do elemento "Favorite Rating". Este é utilizado para adicionar questões que aceitam respostas com ícones para representar a classificação favorita. Cada formulário pode ter um ou mais elementos do tipo "Favorite Rating".

```
"componentsState": [  
  {  
    "values": {  
      "value": "3",  
      "type": "favoriterating",  
      "inputQuestion": "Questão",  
      "id": "Favorite Rating_0"  
    },  
    "id": "Favorite Rating_0"  
  }  
],
```

Figura 5.16: Representação de elemento do formulário - Favorite Rating

Na figura 5.17, é apresentado um exemplo da criação do elemento "Slider Scale". Este é utilizado para adicionar questões que contêm uma escala como resposta e que pode ter valores definidos entre 0 a 100. Cada formulário pode ter um ou mais elementos do tipo "Slider Scale".

```
"componentsState": [
  {
    "values": {
      "type": "sliderscale",
      "inputQuestion": "Questão",
      "value": 70,
      "id": "Slider Scale_0"
    },
    "id": "Slider Scale_0"
  }
],
```

Figura 5.17: Representação de elemento do formulário - Slider Scale

Na figura 5.18, é apresentado um exemplo da criação do elemento "Single Choice". Este é utilizado para adicionar questões que podem ter uma única opção de resposta, de entre uma lista existente. Aqui, o utilizador além de poder definir a questão, pode também definir as opções de resposta, contudo, apenas poderá uma poderá ser selecionada. Cada formulário pode ter um ou mais elementos do tipo "Single Choice".

```
"componentsState": [
  {
    "values": {
      "name": [
        {
          "optionsList": [
            {
              "id": 1,
              "name": "option 1 "
            },
            {
              "id": 2,
              "name": "option 2"
            },
            {
              "id": 3,
              "name": "option 3"
            },
            {
              "id": 4,
              "name": "outra opção "
            },
            {
              "id": 5,
              "name": "outra opção 2"
            }
          ]
        }
      ]
    },
    "type": "singlechoice",
    "inputQuestion": "Questão",
    "id": "Single Choice_0"
  },
  "id": "Single Choice_0"
],
```

Figura 5.18: Representação de elemento do formulário - Single Choice

Na figura 5.19, é apresentado um exemplo da criação do elemento "Multiple Choice". Este é utilizado para adicionar questões que podem ter várias opções de resposta.

```

"componentsState": [
  {
    "values": {
      "value": [
        1
      ],
      "name": [
        {
          "optionsList": [
            {
              "id": 1,
              "name": "option 1"
            },
            {
              "id": 2,
              "name": "option 2"
            },
            {
              "id": 3,
              "name": "option 3"
            },
            {
              "id": 4,
              "name": "outra opção"
            }
          ]
        }
      ]
    },
    "type": "multiplechoice",
    "inputQuestion": "Questão",
    "id": "Multiple Choice_0"
  },
  "id": "Multiple Choice_0"
],

```

Figura 5.19: Representação de elemento do formulário - Multiple Choice

Capítulo 6

Tecnologias e ferramentas

Este capítulo apresenta uma descrição das tecnologias e ferramentas utilizadas ao longo do desenvolvimento do projeto, bem como as suas principais vantagens e desvantagens. A razão para a utilização destas tecnologias deve-se ao facto de serem as utilizadas atualmente no desenvolvimento dos projetos da Talkdesk, contudo poderiam ser consideradas outras alternativas, tal como descrito em seguida.

HTML

Linguagem de marcação utilizada para estruturação e apresentação do conteúdo presente nas páginas web. O código num ficheiro HTML consiste em texto contido dentro de *tags*. Essas *tags* indicam onde a formatação deve ser aplicada, como é que o *layout* deve aparecer em determinados locais e muito mais [40].

De acordo com a perceção da estagiária, esta é uma linguagem fácil de aprender e de utilizar. O facto de ser compatível com diversos *browsers* e ter um conjunto alargado de opções, permite melhorar a experiência do utilizador e assim trazer melhores resultados.

CSS

Linguagem utilizada para definir a apresentação das páginas web, como por exemplo, fontes, cores, margens e efeitos de transição, desenvolvidos a partir do HTML.

Um dos pontos positivos desta linguagem é permitir a separação entre a estruturação e a formatação. Além disso, o CSS permite criar diferentes estilos na mesma página, tornando as possibilidades de personalização quase infinitas. Contudo, embora não seja uma linguagem difícil aprender, é necessário conhecer as suas particularidades para poder utilizá-lo corretamente e melhorar a experiência do utilizador. Ao longo da implementação deste projeto, foi possível perceber que a organização do código CSS nem sempre é fácil, uma vez que, em algumas situações, as mudanças efetuadas em determinados elementos resultavam em alterações indesejadas noutras páginas, ou seja, o processo de estilização das páginas web requer uma constante atenção e manutenção.

JavaScript

Linguagem de programação dinâmica, utilizada para adicionar interatividade às páginas web [35]. "Graças à sua compatibilidade com a maioria dos navegadores modernos, o *JavaScript* é a linguagem de programação do lado do cliente mais utilizada" [35].

React.js

Biblioteca *JavaScript* utilizada para criar interfaces do utilizador [19].

Esta biblioteca distingue-se das restantes, pois introduz uma nova maneira de construir interfaces, uma vez que é "baseada em componentes". Os componentes permitem que a interface seja dividida em várias partes independentes e reutilizáveis e que sejam tratados de forma isolada [19]. Embora seja necessário utilizar React.js para o desenvolvimento deste projeto, existem muitas outras alternativas que também poderiam ser utilizadas, tais como: *Vue* ou *Angular.js*.

React Router

O React Router é uma biblioteca para lidar com rotas no React. Permite a navegação entre *views* de vários componentes numa aplicação de página única [9].

JSON-Server

É uma biblioteca *JavaScript* utilizada para simular APIs REST [35]. Permite criar a base de dados da aplicação no formato JSON e simular requisições (GET, POST, PUT, PATCH e DELETE). O seu principal ponto positivo é a facilidade de configuração e utilização.

Axios

O axios é um cliente HTTP, utilizado para consumir e mostrar os dados da API REST.

Netlify

Plataforma utilizada para hospedar o front-end da aplicação. É simples de configurar e permite a integração com o repositório no github.

Heroku

Plataforma utilizada para hospedar o back-end da aplicação. É simples de configurar e permite a integração com o repositório no github.

Jest

Framework *JavaScript* para criar, executar e estruturar testes [7].

Por ser um framework muito completo, não é necessário instalar outras bibliotecas para o utilizar. Com o Jest é possível escrever testes e *assertions*, executá-los, e por fim, visualizar o resultado da execução do teste.

React Testing Library

Biblioteca utilizada para testar os componentes no React [20]. Permite escrever testes de maneira a simular como um utilizador interage com a aplicação, ao contrário de outras bibliotecas como o Enzyme, que se concentram mais nos detalhes de implementação. Assim, é possível garantir que o que se espera que o utilizador veja, dada alguma interação, é realmente o ele vê.

Visual Studio Code

Utilizado para editar o código-fonte, é uma plataforma disponível para vários sistemas operativos. Suporta as principais linguagens de programação, e inclui ferramentas que permitem fazer *debug* e até gestão de versões, com uma integração com o GitHub [15].

Node.js

Ambiente de execução Javascript *server-side*.

ESLint

É um *code linter* para identificar problemas no código, tais como, variáveis declaradas que não são utilizadas, código duplicado, ausência de pontos e vírgulas, entre outros.

Através da utilização desta ferramenta, é possível escrever código mais organizado e utilizar boas práticas de programação, aumentando a optimização do código e prevenindo erros. Como alternativas ao ESLint, também poderiam ser utilizadas outras ferramentas, como por exemplo, o SonarQube ou JSHint.

Prettier

Ferramenta para formatar o código. Permite obter um estilo consistente ao analisá-lo e adaptá-lo conforme as regras definidas. Sempre que existem alterações, o Prettier é executado e o código é automaticamente formatado.

Git

Sistema utilizado para controlar as diferentes versões do *software* durante o desenvolvimento.

Uma das vantagens de utilização do Git é a rapidez. Uma vez que as operações são processadas localmente, não é necessário contactar o servidor para fazer um *commit* ou *log*, por exemplo. Além disto, também garante confiabilidade, uma vez que se existirem falhas no servidor, a equipa pode continuar o desenvolvimento localmente e os repositórios funcionam como cópias de *backup* de todo o projeto. **Material UI**

Framework utilizada para o design das interfaces da aplicação. De entre as vantagens da Material-UI, destacam-se a consistência no design, a reutilização de código e personalização e flexibilidade de componentes.

Esta tecnologia foi escolhida pela estagiária, contudo, também foram consideradas outras alternativas, tais como a Ant Design ou Bootstrap.

Slack

Ferramenta utilizada para a colaboração e comunicação entre os elementos da equipa [17]. É muito útil, uma vez que permite a criação de canais de comunicação e a partilha de ficheiros.

Trello

Plataforma utilizada para organizar as tarefas de cada sprint. É simples e intuitiva e permite atualizações em tempo real. Como alternativas ao Trello, também poderiam ser utilizadas outras plataformas, como a MeisterTask ou Jira.

Github

Plataforma utilizada para armazenar o código-fonte da aplicação. Através do Github, é possível armazenar o código e o histórico de versões num repositório online, independentemente do que acontece com a máquina local. [31].

Draw.IO

Software utilizado para desenhar a arquitetura da aplicação.

Capítulo 7

Qualidade do código

De entre os aspectos utilizados para avaliar a qualidade do código-fonte de um *software*, incluem-se [21]: facilidade de entendimento, formatação consistente, ausência de erros de compilação, capacidade adequada e consistente de tratamento de erros de execução e boas práticas de programação. Quando estes parâmetros não se verificam, o produto final pode ter falhas, o que resulta numa má experiência para os utilizadores. Além disso, a manutenção será mais difícil ao longo do tempo e a adição de novos recursos leva mais tempo e esforço do que seria necessário. Assim, a utilização de boas práticas, durante o processo de desenvolvimento, é importante, pois além de melhorar a qualidade do *software*, também facilita a organização da equipa.

Este capítulo descreve as técnicas utilizadas para garantir a qualidade do código-fonte da aplicação desenvolvida.

7.1 Github

O Github é uma plataforma que foi utilizada para armazenar o código-fonte da aplicação e para colaboração no desenvolvimento entre a equipa. Encontra-se disponível no seguinte URL: <https://github.com/>

7.1.1 Criação de *branches*

Através dos *branches*, ramificações criadas no GitHub para separar as diferentes funcionalidades da aplicação e trabalhar em diferentes versões do repositório ao mesmo tempo, foi possível registar modificações e ter um maior controlo ao longo do desenvolvimento.

Uma das práticas definidas para a criação de *branches* foi a utilização de uma nomenclatura padrão que permitisse facilmente identificar a funcionalidade desenvolvida. Apesar deste factor não causar um impacto negativo no projeto, considerou-se que podia facilitar a identificação do código e a organização do repositório. Assim, todos os *branches* foram nomeados, de acordo com a mesma nomenclatura. Inicialmente identificou-se o tipo de *branch*: **feat**, quando correspondia ao desenvolvimento de uma nova funcionalidade; **fix**, para a correção de um *bug* e **test** para o desenvolvimento de testes. Em seguida, efetuou-se uma breve descrição do *branch*.

Na figura 7.1, é apresentado um exemplo dos *branches* criados.

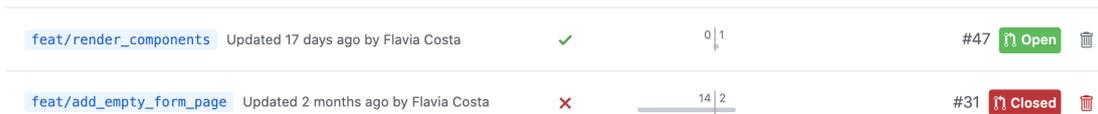


Figura 7.1: Representação de *branches* no Github

7.1.2 Criação de *pull requests*

Após a conclusão da implementação de cada funcionalidade e a atualização do código no repositório, foi criado um *pull request* (PR). Através desta funcionalidade, é possível comunicar aos restantes elementos da equipa que foram efetuadas novas alterações no código e assim solicitar que estas sejam adicionadas ao repositório principal (*master branch*). Para tal, os responsáveis pelo repositório, analisam se o código enviado está de acordo com os padrões definidos para o projeto. Se estiver, pode ser adicionado ao *branch* principal. Se não estiver, são solicitadas novas alterações no código.

Em cada *pull request* criado ao longo do desenvolvimento deste projeto foi adicionada uma descrição das alterações efetuadas e das suas características mais relevantes.

Na figura 7.2 é apresentado um exemplo de diferentes *pull requests* criados durante o desenvolvimento e na figura 7.3 é exemplificada a descrição de um dos *pull requests*. Por fim, na figura 7.5 é apresentado um exemplo de um pedido de alterações.

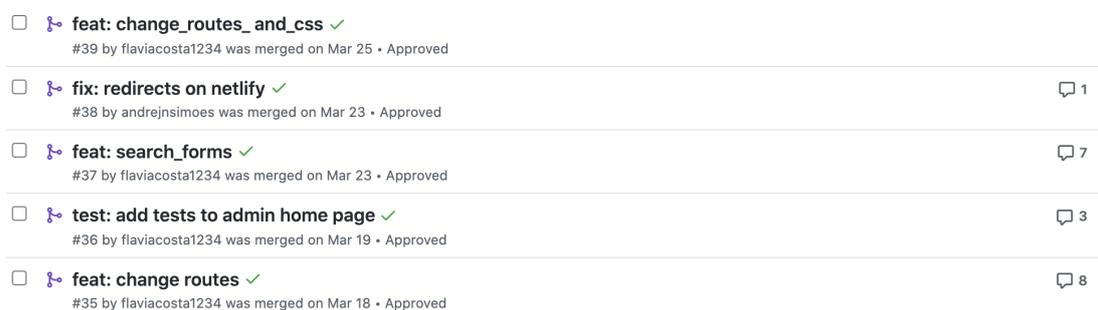


Figura 7.2: Representação de *pull requests* no GitHub

Em cada *pull request* é possível identificar qual é a *branch* onde se encontram as novas alterações e verificar se estas foram aceites pela pessoa responsável por rever o código (quando existe a mensagem "*Approved*").

Para cada um, é possível efetuar uma descrição das alterações efetuadas (Figura 7.3).

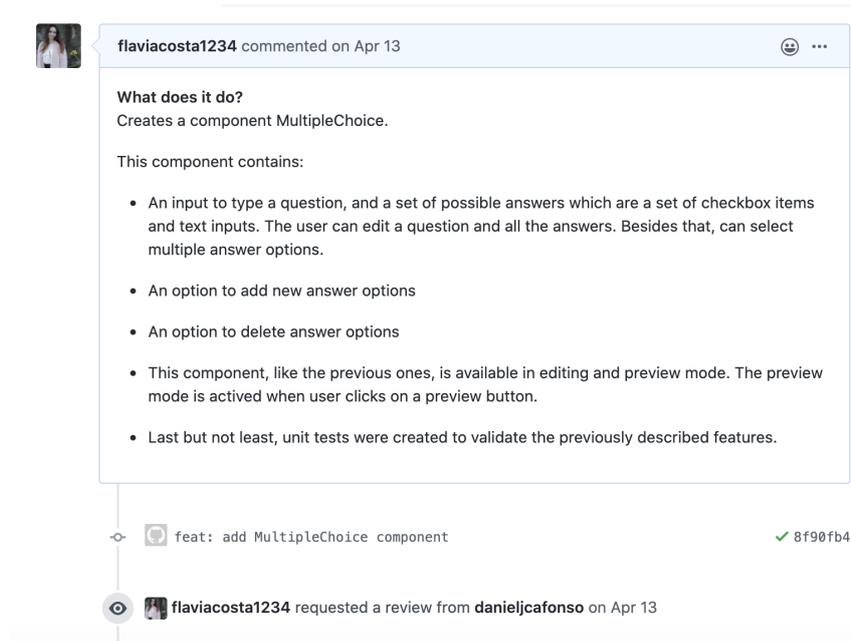


Figura 7.3: Descrição de *Pull Request* no GitHub

Por fim, também é possível visualizar comentários e pedidos de alterações (Figura 7.4).



Figura 7.4: Alterações - *Pull Request*

A aplicação de boas práticas nesta fase resultou na diminuição do tempo gasto pelos membros do equipa para rever o código, integrá-lo e atualizá-lo com o armazenado no repositório. Para garantir a qualidade do código desenvolvido utilizaram-se ferramentas complementares, descritas em seguida.

7.2 Git Hooks

Os Git Hooks são *scripts* executados pelo Git, quando ocorrem determinadas ações no repositório do GitHub, como por exemplo, um *commit* para adicionar as novas alterações ou o *push* para fazer o *upload* do conteúdo do repositório local para o remoto [6]. São úteis para identificar problemas no código antes deste ser revisto e adicionado ao *branch* principal.

Algumas das principais vantagens dos GitHooks são:

- Evitar *commits*, caso exista algum problema no código;
- Impedir *pushes* que não estejam de acordo com os padrões definidos;
- Garantir a execução casos de teste nos ficheiros que estão a ser adicionados com o *commit*.

Seguidamente, são descritos os hooks utilizados neste projeto para garantir a qualidade do código, assim como exemplos do seu funcionamento.

- **pre-commit:** Hook utilizado para garantir a execução dos testes e para identificar as alterações que devem ser feitas no código antes dos ficheiros serem enviados para o repositório.

Na figura 7.5 é apresentado o funcionamento do hook "Pre-commit".

```
husky > pre-commit (node v10.16.3)
△ Some of your tasks use `git add` command. Please remove it
by tasks will be automatically added to the git commit index

✓ Preparing...
✓ Running tasks...
✓ Applying modifications...
✓ Cleaning up...
```

Figura 7.5: Execução do hook "Pre-commit"

Como se vê na figura anterior, sempre que é efetuado um novo *commit*, é iniciado o hook que, por sua vez, efetua diversas verificações ao código. Se existir algum problema, como por exemplo, variáveis que são definidas no código mas não são utilizadas, o *commit* não é confirmado e os erros são mostrados no ecrã, como se verifica na figura 7.6.

```
16:8 error 'IconButton' is defined but never used no-unused-
vars
18:8 error 'Tooltip' is defined but never used no-unused-
vars
19:8 error 'DeleteIcon' is defined but never used no-unused-
vars
20:8 error 'VisibilityIcon' is defined but never used no-unused-
vars
39:3 error Unexpected console statement no-console
46:9 error 'deleteForms' is assigned a value but never used no-unused-
vars
124:25 warning The use of method `push` is not allowed as it might be a mutating method fp/no-muta-
ting-methods

× 38 problems (26 errors, 12 warnings)
husky > pre-commit hook failed (add --no-verify to bypass)
```

Figura 7.6: Hook "Pre-commit"

- **commit-msg:** Este hook foi utilizado para garantir que todos os *commits* continham uma mensagem descritiva. Através da figura 7.7, é possível ver que quando é criado um *commit* sem qualquer mensagem que o descreva, o sistema apresenta uma mensagem de erro e o *commit* não é confirmado.

```

CBR-MBP-0058:formbuilder-app flavia.costa$ git commit -m ""
husky > pre-commit (node v10.16.3)
▲ Some of your tasks use 'git add' command. Please remove it from the config since all modifications made by tasks will be automatically added to the git commit index.

✔ Preparing...
✔ Running tasks...
✔ Applying modifications...
✔ Cleaning up...
husky > commit-msg (node v10.16.3)
Aborting commit due to empty commit message.
CBR-MBP-0058:formbuilder-app flavia.costa$

```

Figura 7.7: Hook "commit-msg"

- **pre-push:** Hook utilizado para garantir que os testes passam antes de se poder fazer o *push* das alterações para o repositório remoto.

Quando é iniciado um *push*, o hook *pre-push* é executado (Figura 7.8).

```

CBR-MBP-0058:formbuilder-app flavia.costa$ git push
husky > pre-push (node v10.16.3)
$ react-scripts test --watchAll=false

```

Figura 7.8: Hook "pre-push"

Caso algum teste falhe, as alterações não são confirmadas (Figura 7.9).

```

    in WithStyles(ForwardRef(Paper)) (at Wrapper.js:16)
    in Wrapper (at SingleChoice.js:119)
    in SingleChoice (at EmptyFormPage.js:402)
    in div (at EmptyFormPage.js:401)
    in div (at EmptyFormPage.js:400)
    in EmptyFormPage (at EmptyFormPage.test.js:104)
    in FormsProvider (at EmptyFormPage.test.js:103)
    in AuthProvider (at EmptyFormPage.test.js:102)
    in Router (created by MemoryRouter)
    in MemoryRouter (at EmptyFormPage.test.js:101)

Test Suites: 28 passed, 28 total
Tests:       1 skipped, 83 passed, 84 total
Snapshots:  0 total
Time:        11.568s
Ran all test suites.
To https://github.com/flaviacosta1234/formbuilder_app.git
! [rejected]        feat/render_components -> feat/render_components (non-fast-forward)
error: failed to push some refs to 'https://github.com/flaviacosta1234/formbuilder_app.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

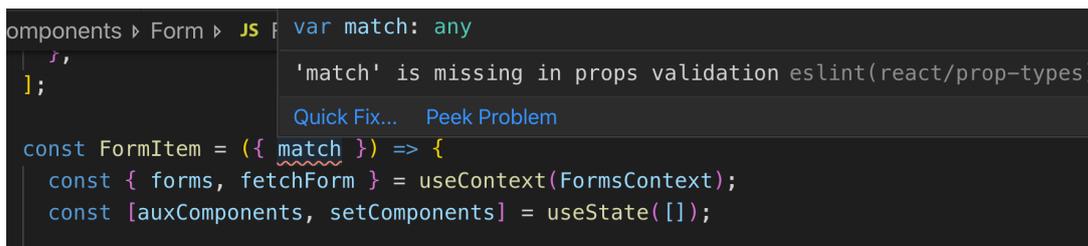
```

Figura 7.9: Hook pre-push - Testes

7.3 Análise estática do código

O ESLint foi a ferramenta utilizada para analisar o código e encontrar erros de sintaxe ou *bugs*. Porém, para evitar analisar constantemente todo o código desenvolvido, adicionou-se o Lint-staged, um script que permite executar verificações apenas nos arquivos enviados em cada commit. Ao analisar estes arquivos é possível garantir que todos têm o mesmo estilo de código e que não existem quaisquer erros ao enviar *commits* para o repositório.

As figuras 7.10 e 7.11 ilustram exemplos das verificações efetuadas pelo ESLint.



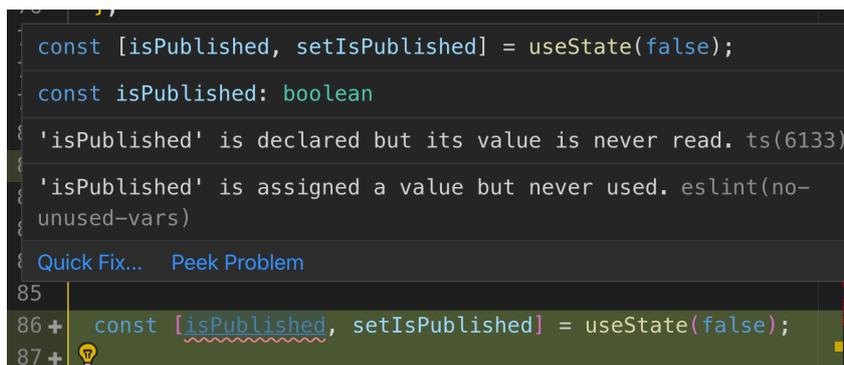
```

components > Form > JS {
  },
];
const FormItem = ({ match }) => {
  const { forms, fetchForm } = useContext(FormsContext);
  const [auxComponents, setComponents] = useState([]);
}
  
```

'match' is missing in props validation eslint(react/prop-types)

Quick Fix... Peek Problem

Figura 7.10: Verificações em arquivos



```

const [isPublished, setIsPublished] = useState(false);
const isPublished: boolean
}
}
  
```

'isPublished' is declared but its value is never read. ts(6133)

'isPublished' is assigned a value but never used. eslint(no-unused-vars)

Quick Fix... Peek Problem

85

86 + const [isPublished, setIsPublished] = useState(false);

87 +

Figura 7.11: Verificações em arquivos

Através destas figuras, é possível ver que o ESLint identifica diversos problemas no código, tais como a ausência de validações (Figura 7.10), ou variáveis declaradas mas que não são utilizadas (Figura 7.11). Desta forma, é mais fácil eliminar código redundante e que apenas dificulta o processo de leitura e compreensão.

7.4 Cobertura do código

Uma das métricas utilizadas para avaliar a qualidade do código no desenvolvimento de software é a realização de testes. No entanto, apenas a realização de testes não garante qualidade. É importante implementar casos de teste relevantes e eficientes e monitorizar e analisar adequadamente os resultados. Isso pode ser feito pelos sistemas de análise de cobertura de código.

Através destes é possível determinar quanto da base de código é que é realmente testada. Além disso, são fornecidas informações adicionais, como por exemplo, os arquivos que ainda não foram testados, as linhas de código mais relevantes para serem testadas, entre outros.

Neste projeto, utilizou-se a ferramenta *Coveralls.io* para monitorizar regularmente a cobertura do código desenvolvido e verificar quais os ficheiros que ainda não tinham sido testados. Após integrar este sistema no projeto, é possível ver um relatório que indica quais são as partes do código verificadas pelos testes e quais não são. Isso fornece as seguintes informações importantes:

- Percentagem de cobertura total de todo o projeto;
- Percentagem total de cobertura por ficheiro, um método (uma métrica integral que permite monitorizar a dinâmica da cobertura em diferentes partes do projeto);
- Linhas de código cobertas (métrica não integral para determinar se uma parte específica do código é coberta por testes).

Seguidamente, são apresentados exemplos da utilização da ferramenta Coveralls.io no projeto desenvolvido.

Na figura 7.12 é possível ver a percentagem de código testado em cada ficheiro, o número de linhas de código existentes, quais dessas linhas são mais relevantes, quais as que já foram verificadas, quais as que ainda não apresentam qualquer caso de teste, entre outros.

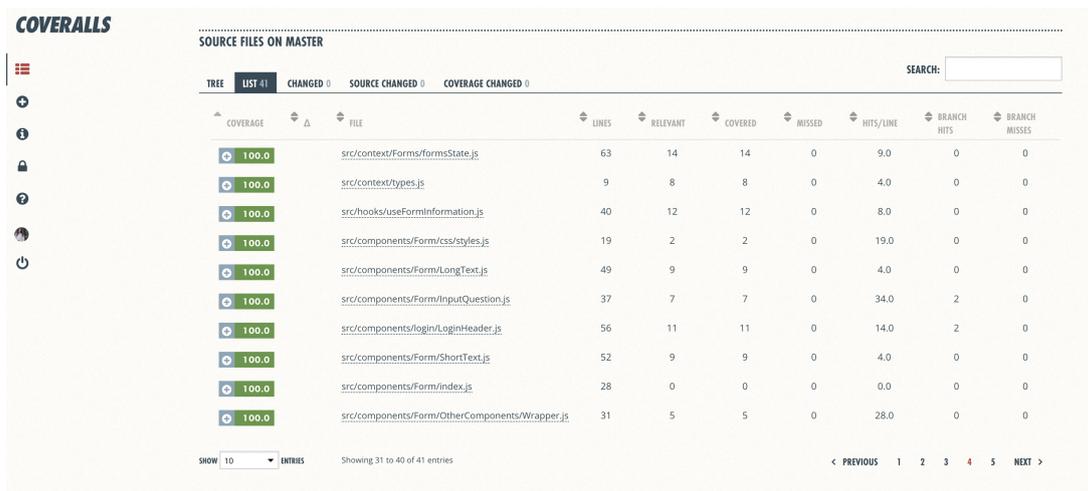


Figura 7.12: Cobertura do código em cada ficheiro

Também é possível ver a evolução da cobertura do código ao longo dos vários *commits* que são adicionados, isto é, verificar se aumenta ou diminui de acordo com as novas alterações (Figura 7.13).

BUILDS	BRANCH	COVERAGE	COMMIT	COMMITTER	TYPE	TIME	VIA
100119354	feat/render_components	72.56	Merge 35383 1c9b1 nto 99 a01fde b	GitHub	PULL #47	09 May 2020 06:34PM UTC	github
99343867	feat/render_components	72.56	Merge 5bc24 2bce i nto 99 a01fde b	GitHub	PULL #47	08 May 2020 05:24PM UTC	github
99289273	feat/render_components	72.33	Merge 968c7 dd1f i nto 99 a01fde b	GitHub	PULL #47	08 May 2020 04:07PM UTC	github
99131693	feat/render_components	72.33	Merge 968c7 dd1f i nto 99 a01fde b	GitHub	PULL #47	08 May 2020 12:39PM UTC	github

Figura 7.13: Cobertura do código em cada *commit*

7.5 Github Actions: Integração contínua

A integração contínua é a prática de automatizar a integração de alterações do código num único projeto de *software*. O processo é composto por diferentes etapas, nas quais o código é verificado e corrigido antes de ser integrado [16].

Através do *Github Actions* é possível criar diferentes tarefas, chamadas ações, que podem ser executadas automaticamente em determinados eventos, como um *commit* ou um *pull request* e combinadas para criar um *workflow* personalizado [4].

A cada *commit* realizado, o código passou por diferentes etapas, que se encontram descritas em seguida:

- *Checkout* do repositório - Através desta ação, é feito o *checkout* do repositório para que este fique disponível para utilização;
- *Setup do node* - Criar um ambiente node;
- Instalar dependências - Instalar bibliotecas externas das quais o código depende;
- Executar os testes - Garantir que os testes são executados;
- *Code linting* - Ação utilizada para detetar e corrigir erros no código;
- Validar mensagem de *commit* - Verificar se as mensagens de confirmação respeitam um determinado padrão;
- Verificar *code coverage* - Esta ação disponibiliza os dados de cobertura do conjunto de testes para análise e controlo de alterações.

Através deste conjunto de ações, é possível garantir a qualidade do código, antes deste ser integrado no repositório principal.

Com a utilização de todas estas ferramentas verificou-se que, garantir a qualidade do código desenvolvido, é muito importante, pois esta influencia diretamente a confiabilidade e a segurança da aplicação. Se os padrões não forem garantidos, o código poderá estar suscetível a ataques de fontes externas ou poderão ocorrer erros devido à falta de qualidade de codificação, além de que a existência de falhas também poderá provocar uma experiência negativa aos utilizadores e aumenta as dificuldades de organização da equipa de desenvolvimento.

Capítulo 8

Testes

Através da realização de testes é possível verificar se o sistema desenvolvido respeita os requisitos definidos [24] e se apresenta o comportamento esperado.

As atividades de teste incluem diversas ações, tais como: A definição das condições, a execução de casos de teste, a análise dos resultados e a avaliação dos critérios de saída.

Neste capítulo é apresentada uma breve descrição dos testes realizados para validar o *software* desenvolvido. Estes estão divididos em dois grupos: Testes Unitários e Testes de Sistema. Todos os casos de teste podem ser consultados no anexo C (Plano de testes).

8.1 Testes Unitários

São utilizados para testar as funcionalidades do sistema individualmente. O objetivo é assegurar que cada unidade do código, como um método ou uma classe, está a funcionar como esperado.

Ao longo do desenvolvimento da aplicação foram elaborados diferentes casos de teste, obtendo-se no total um conjunto de 83 testes unitários. Estes foram criados usando a biblioteca de testes do React (*React Testing Library*) [10] e o Jest [7].

Os testes foram escritos de acordo com o padrão AAA (*Arrange, Act, Assert*) [39]:

- Arrange (Organizar) - Inicializar os objetos a ser testados.
- Act (Agir) - Invocar o método a ser testado com os parâmetros organizados.
- Assert (Declarar) - Verificar se a ação do método em teste retorna a saída esperada.

A validação de cada unidade de código foi realizada ao longo do desenvolvimento, para ser mais fácil perceber onde se encontravam os *bugs* e proceder à sua correção antes de se iniciar o desenvolvimento da funcionalidade seguinte, evitando assim falhas com maior impacto no produto final.

Através da ferramenta *Coveralls.IO* [5], gerou-se o relatório geral de cobertura dos testes, que é apresentado de forma genérica na figura 8.1.

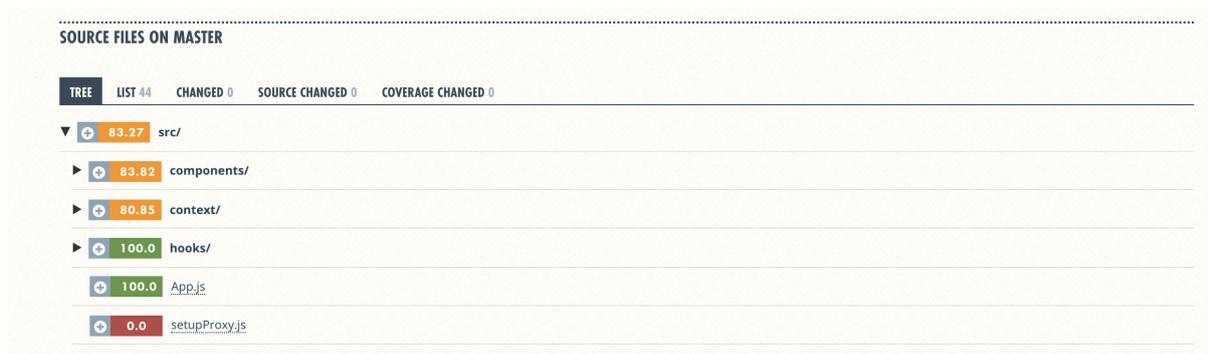


Figura 8.1: Cobertura total do código-fonte

Como se verifica na figura anterior, obteve-se uma cobertura de código de 83.27 %, o que significa que a maior parte do código-fonte desenvolvido foi testado, existindo apenas alguns ficheiros nos quais nem todas as linhas de código foram verificadas, tal como é possível ver através da figura 8.2.



Figura 8.2: Cobertura do código por ficheiro

A cobertura total não foi atingida devido essencialmente à dimensão da aplicação e também devido às restrições de tempo que se verificaram ao longo do desenvolvimento do projeto.

Na tabela 8.1, é representado um exemplo de um dos testes unitários realizados. Os restantes podem ser consultados no anexo C (Plano de Testes).

Tabela 8.1: Exemplo de teste realizado

Teste 01	Elementos na página de login
Descrição	Verificar se a página de login contém um elemento para adicionar o <i>username</i> , um elemento para adicionar a <i>password</i> e um botão para fazer login.
Resultado	PASSOU

8.2 Testes de Sistema

O teste de sistema é definido como o processo realizado para avaliar se um *software* cumpre com os requisitos especificados [14]. Para a sua realização, foi necessário criar diferentes casos de teste, de acordo com os requisitos do projeto.

Os testes foram realizados com a participação de dois utilizadores, que não estiveram envolvidos na fase de desenvolvimento do projeto. Estes foram convidados pela aluna via *skype*, onde lhes foi explicada a atividade e marcado o dia do teste. Um dos utilizadores ficou com a função de ser o administrador, responsável pelo processo da criação e gestão dos formulários, enquanto o outro representou o supervisor, ficando responsável por visualizar os formulários publicados e efetuar o seu preenchimento.

No dia anterior à realização do teste, ambos os utilizadores receberam o manual que contém a explicação do funcionamento da aplicação e que pode ser consultado no anexo B (Manual de Utilizador). Além disso, receberam também um guião com a lista de tarefas que cada um teria de realizar. Este encontra-se representado seguidamente.

Tarefas do Administrador

- Inserir dados de acesso incorretos, isto é, um *username* ou *password* que não existem no sistema;
- Inserir os dados de acesso corretos disponibilizados pela aluna, isto é, o *username* e *password* do administrador;
- Criar um formulário sem qualquer nome ou descrição;
- Criar um formulário com um nome que já exista num outro formulário guardado no sistema;
- Criar um formulário com um nome e descrição disponibilizados pela aluna;
- Adicionar todos os elementos do menu ao formulário, de acordo com as indicações da aluna;
- Adicionar várias opções de resposta, nos elementos que suportam escolha múltipla;
- Preencher cada elemento com as questões indicadas pela aluna;
- Clicar na opção para eliminar um elemento do formulário;
- Clicar na opção para duplicar um elemento do formulário;
- Clicar na opção para pré-visualizar formulário;
- Clicar na opção para guardar formulário;
- Inserir o nome de formulário guardado no sistema, na barra de pesquisa da página inicial;
- Editar o nome e descrição do formulário;
- Abrir formulário;
- Clicar na opção para publicar formulário;
- Clicar na opção para fazer logout.

Tarefas do Supervisor

- Aceder à aplicação com dados de acesso incorretos, isto é, com um *username* e *password* que não existem no sistema;
- Aceder à aplicação com dados de acesso corretos disponibilizados pela aluna, isto é, com o *username* e *password* do supervisor;
- Clicar na opção para preencher formulário;
- Adicionar texto em elementos que apenas aceitam números como resposta;
- Selecionar várias opções em elementos que aceitam apenas uma única opção;
- Inserir texto nos elementos que suportam apenas datas;
- Clicar na opção para fazer logout.

Para cada uma das tarefas descritas, foram criados diferentes casos de teste. Na figura 8.3, encontra-se representado um exemplo de um dos testes de sistema realizados. Os restantes podem ser consultados no anexo C (Plano de testes).

Caso de teste 001	Dados de acesso incorretos
Pré-condições	Aceder à página de login
Procedimento	1 - O utilizador insere dados incorretos 2 - O utilizador seleciona a opção "Login" 3 - O sistema verifica se os dados do utilizador existem no sistema.
Dados de entrada	Dados de acesso incorretos
Resultado esperado	Mensagem de erro no sistema

Figura 8.3: Exemplo de teste de sistema

Durante a realização dos testes unitários, foram encontrados alguns *bugs* na aplicação. Assim, estas falhas foram corrigidas ao longo do desenvolvimento, para evitar problemas maiores no produto final.

Posteriormente, através dos testes de sistema com utilizadores garantiu-se que a plataforma tinha o comportamento esperado e estava a funcionar na totalidade.

Capítulo 9

Conclusões

Neste capítulo são apresentadas as conclusões obtidas após o desenvolvimento deste projeto bem como alguns pontos de evolução a serem realizados em futuras versões da plataforma.

A aplicação foi concebida na sua totalidade pela autora do projeto, isto é desde o estudo de soluções existentes, análise e levantamento de requisitos, definição da arquitetura e modelo de dados, até à implementação e testes.

No 1º semestre, realizou-se um extenso trabalho de investigação, para identificar as soluções existentes no mercado, analisar os seus pontos positivos e negativos e determinar como deveria ser a solução proposta pela Talkdesk e quais os requisitos que esta deveria ter. Durante este semestre, ocorreram atrasos na definição do projeto e das várias etapas, uma vez que quando a estagiária iniciou o estágio curricular, o plano não estava completo e alinhado com os objetivos necessários à conclusão da unidade curricular de Estágio/Dissertação. Estes imprevistos originaram dificuldades na organização e gestão do tempo e consequentemente indecisões quanto ao trabalho que iria ser desenvolvido no 2º semestre.

Após reuniões com os orientadores da Talkdesk e com o *product owner* de um dos módulos da empresa, foi sugerido que a estagiária desenvolvesse uma ferramenta para dar suporte a uma das necessidades da Talkdesk, o controlo e gestão da qualidade de interações telefónicas entre clientes e agentes de *call centers*. O principal objetivo deste módulo é a criação de avaliações a partir da construção de formulários personalizados e específicos para monitorizar o serviço prestado por um agente ao cliente. A ideia desta aplicação seria estar integrada com outro módulo da empresa que permite ouvir as gravações das chamadas realizadas por cada agente e, em seguida, efetuar avaliações, tendo em conta o desempenho obtido. No entanto, uma vez que este último não se encontra totalmente desenvolvido pela Talkdesk, não foi possível realizar essa integração ainda durante o decorrer deste estágio.

Ao longo deste estágio foram sentidas diversas dificuldades, principalmente na fase de implementação onde foi necessária a aprendizagem de várias tecnologias para desenvolvimento web. Também o desenho do modelo de dados da aplicação foi um grande desafio, uma vez que não existia um *back-end* previamente desenvolvido pela Talkdesk e assim foi necessário pensar numa solução para obter uma plataforma funcional, tendo em conta o objetivo do estágio e as restrições de tempo a ele associadas.

Apesar das dificuldades sentidas, todos os requisitos propostos foram implementados e o código-fonte foi desenvolvido de acordo com boas práticas de programação.

A experiência de trabalhar num ambiente empresarial foi enriquecedora e motivadora. A autora percebeu que existem diferenças entre o desenvolvimento de projetos acadêmicos e de projetos reais, onde todos os pormenores devem ser analisados para que o produto final seja um sucesso. Além disso, foi possível adquirir novos conhecimentos técnicos, uma vez que foram utilizadas tecnologias com as quais a autora nunca tinha trabalhado e também foram seguidas metodologias que implementam boas práticas em projetos de desenvolvimento de *software*.

Na opinião da aluna, o estágio foi um dos maiores desafios que teve ao longo do seu percurso académico, uma vez que apesar de todos os obstáculos vividos, estes foram ultrapassados com sucesso, sempre com o objetivo de enriquecer os seus conhecimentos, quer a nível profissional como pessoal.

Futuramente, a aplicação poderá ser integrada com o módulo de gestão de qualidade da Talkdesk, nomeadamente o módulo que permite o acesso às gravações das chamadas. Prevê-se também que possa vir a ter requisitos adicionais, como por exemplo, o acesso a modelos de formulários já existentes no sistema, para que estes não tenham de ser criados pelo administrador.

Referências

- [1] 3 Keys to Workforce Optimization in the Call Center. <https://callminer.com/blog/3-keys-workforce-optimization-call-center/>. Consultado: 2019-12.
- [2] A história da Talkdesk, da bênção do pouco dinheiro ao estatuto de unicórnio. <https://visao.sapo.pt/atualidade/economia/2018-11-03-a-historia-da-talkdesk-da-bencao-do-pouco-dinheiro-ao-estatuto-de-unicornio/>. Consultado: 2019-09.
- [3] About Talkdesk. <https://www.talkdesk.com/about/>. Consultado: 2019-09.
- [4] Automate your workflow from idea to production. <https://github.com/features/actions>. Consultado: 2020-06.
- [5] Coveralls. <https://docs.coveralls.io/>. Consultado: 2020-06.
- [6] Customizing git - git hooks. <https://git-scm.com/docs/githooks>. Consultado: 2020-05.
- [7] Jest - delightful javascript testing. <https://jestjs.io/>. Consultado: 2020-05.
- [8] Learning Tree Launches IT Workforce Optimization Solutions. <https://www.businesswire.com/news/home/20151118006035/en/Learning-Tree-Launches-Workforce-Optimization-Solution/>. Consultado: 2019-12.
- [9] React router. <https://www.educative.io/edpresso/what-is-a-react-router>. Consultado: 2020-05.
- [10] React testing library. <https://testing-library.com/docs/react-testing-library/intro>. Consultado: 2020-06.
- [11] Requisitos. <https://pt.wikipedia.org/wiki/Requisito>. Consultado: 2020-05.
- [12] SoGoSurvey Review. <https://reviews.financesonline.com/p/sogosurvey/>. Consultado: 2019-12.
- [13] Talkdesk Integrations. <https://www.talkdesk.com/call-center-software/integrations/>. Consultado: 2019-09.
- [14] Teste de sistema. https://pt.wikipedia.org/wiki/Teste_de_software. Consultado: 2020-06.
- [15] Visual studio code. <https://code.visualstudio.com/>. Consultado: 2020-05.
- [16] What is continuous integration. <https://www.atlassian.com/continuous-delivery/continuous-integration>. Consultado: 2020-06.
- [17] What is slack? <https://slack.com/intl/en-pt/help/articles/115004071768-What-is-Slack->. Consultado: 2020-01.

-
- [18] V. Bajpai and R. P. Gorthi. On non-functional requirements: A survey. Conference on Electrical, Electronics and Computer Science.
- [19] Porcello Eve Banks Alex. *Learning React*. O'Reilly Media, 2017. Consultado: 2019-10.
- [20] Kent C. Dodds. Introducing the react-testing-library. <https://kentcdodds.com/blog/introducing-the-react-testing-library>. Consultado: 2020-05.
- [21] Darwin Ian F. *Checking Java Programs*. O'Reilly Media, Inc., 2007. Consultado: 2020-05.
- [22] Audrey Gilmore;. Call centre management: Is service quality a priority? Consultado: 2020-06.
- [23] Kurt Bittner Ivar Jacobson, Ian Spence. *Use-Case 2.0*. 2011. Consultado: 2020-05.
- [24] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad. Software testing techniques: A literature review. In *2016 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, 2016. Consultado: 2020-06.
- [25] Soren Lauesen. Task descriptions as functional requirements. 2003. Consultado: 2019-12.
- [26] Jon Loeliger. *Version Control with Git*. O'Reilly, May 2009. Consultado: 2019-11.
- [27] Vieira Marco. Project Management, Risk Management. Universidade de Coimbra, 2017/2018. Consultado: 2020-02.
- [28] Megan Marrs. 8 Best Survey Tools: Create Awesome Surveys For Free! <https://www.wordstream.com/blog/ws/2014/11/10/best-online-survey-tools>. Consultado: 2020-01.
- [29] Omer Minkara. CONTACT CENTER WORKFORCE OPTIMIZATION: PRODUCTIVE AGENTS, HAPPY CUSTOMERS AND REDUCED COSTS. Report, May-2017. Consultado: 2020-01.
- [30] Ronielton Oliveira. Prince2: A técnica de priorização moscow. 2014. Consultado: 2019-12.
- [31] Achilleas Pipinellis. *GitHub Essentials*. Packt Publishing, 2018. Consultado: 2020-05.
- [32] Josh Powell and Michael Mikowski. *Single Page Web Applications: JavaScript End-to-end*. Manning Publications, 2013. Consultado: 2020-04.
- [33] Felipe Rampinelli. O que é call center e contact center? <https://www.dds.com.br/blog/index.php/o-que-e-call-center-contact-center/>. Consultado: 2019-11.
- [34] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002. Consultado: 2019-11.
- [35] Steve Suehring. *JavaScript Step by Step*. Microsoft Press, 2013. Consultado: 2019-12.
- [36] Marco Sá, Dulce Magalhães Painho. CRM e call centers: customer relationship management e sistemas de informação em call centers. Working paper, Jun-2001. Consultado: 2020-01.
- [37] Narayanaswamy Vasantha Raju and Harinarayana N.S. Online survey tools: A case study of google forms. 01 2016. Consultado: 2020-01.
- [38] Carlos Eduardo Vazquez and Guilherme Siqueira Simões. *Engenharia de Requisitos*. Brasport, 2016. Consultado: 2020-01.
- [39] Khorikov Vladimir. *Unit Testing Principles, Practices, and Patterns*. Manning Publications, 2020. Consultado: 2020-05.
- [40] Faithe Wempen. *HTML5 Step by Step*. O'Reilly Media, 2011. Consultado: 2019-12.

Está página é intencionalmente deixada em branco.

Anexo A

Requisitos

O presente documento especifica as características e funcionalidades da aplicação, através de casos de uso e *mockups*.

A.1 Diagramas de casos de uso

Nesta secção são apresentados os diagramas de casos de uso que descrevem as funcionalidades do sistema e as interações com os seus elementos (atores).

Na figura A.1 estão representadas as funcionalidades executadas pelo administrador e na figura A.2 estão representadas as funcionalidades do supervisor.



Figura A.1: Casos de uso e interação com o administrador

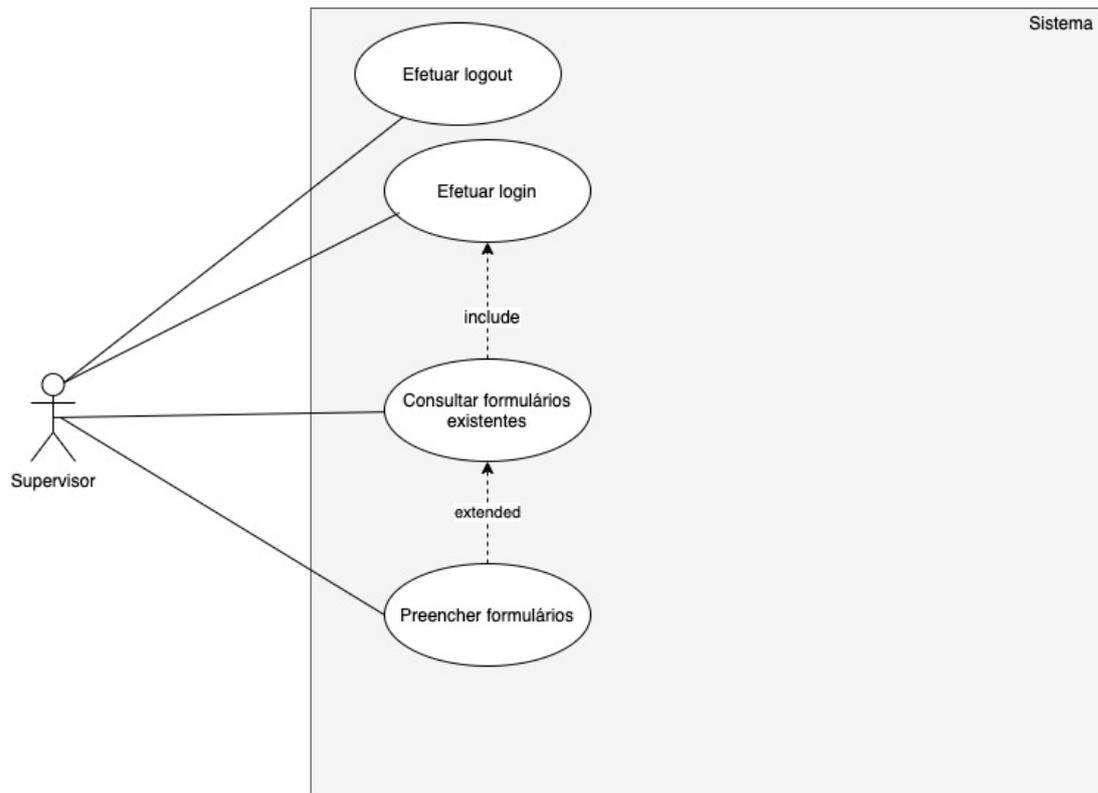


Figura A.2: Casos de uso e interação com o supervisor

A.2 Especificação de requisitos

UC-01:	Efetuar Login
---------------	----------------------

<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende aceder à aplicação.
-------------------	--

<i>Atores:</i>	Administrador, Supervisor
----------------	---------------------------

<i>Pré-condições:</i>	O utilizador possui uma conta registada no sistema.
-----------------------	---

<i>Fluxo esperado:</i>	<ol style="list-style-type: none">1. O sistema mostra o painel de autenticação;2. O utilizador insere as credencias de autenticação e tenta autenticar-se.
------------------------	---

<i>Fluxo alternativo:</i>	No passo 2 do fluxo esperado, caso as credenciais de acesso estejam incorretas: <ol style="list-style-type: none">1. O sistema mostra uma mensagem de erro;2. O caso de uso continua no ponto 1.
---------------------------	---

<i>Pós-condições:</i>	O utilizador acede à aplicação (tendo em conta as restrições de cada ator)
-----------------------	--

UC-02:	Efetuar Logout
---------------	-----------------------

<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende terminar sessão na aplicação.
-------------------	--

<i>Atores:</i>	Administrador, Supervisor
----------------	---------------------------

<i>Pré-condições:</i>	Estar autenticado no sistema.
-----------------------	-------------------------------

Fluxo esperado:

1. O sistema mostra uma opção para terminar sessão;
2. O utilizador clica na opção para terminar sessão;

Fluxo alternativo:

No passo 2 do fluxo esperado, caso o utilizador não consiga terminar sessão:

1. O utilizador permanece no mesmo ecrã.

<i>Pós-condições:</i>	O utilizador termina a sessão e é redirecionado para a interface de login.
-----------------------	--

UC-03:	Criar novo formulário
---------------	------------------------------

<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende criar um novo formulário de avaliação que não contem qualquer elemento.
-------------------	--

<i>Atores:</i>	Administrador
----------------	---------------

<i>Pré-condições:</i>	<ol style="list-style-type: none">1. O utilizador está autenticado na aplicação.2. O utilizador seleciona a opção para criar formulário.
-----------------------	---

<i>Fluxo esperado:</i>	<ol style="list-style-type: none">1. O sistema mostra uma opção para criar uma novo formulário.2. O utilizador clica na opção para criar novo formulário.3. O sistema mostra um painel para preencher o nome e descrição do formulário que pretende criar.4. O utilizador insere o nome e descrição do formulário.5. O utilizador clica em "Continuar".6. Um novo formulário é criado.
------------------------	---

<i>Fluxo alternativo:</i>	No passo 5 do fluxo esperado, caso o utilizador não preencha os dados corretamente: <ol style="list-style-type: none">1.1 O sistema mostra uma mensagem de erro se o utilizador não preencher o nome e/ou descrição do formulário, uma vez que estes são campos obrigatórios;1.2 O sistema mostra uma mensagem de erro se o utilizador inserir um nome muito pequeno no formulário.1.3 O sistema mostra uma mensagem de erro se o nome do formulário que o utilizador deseja atribuir já existir no sistema. <ol style="list-style-type: none">2. O utilizador permanece no mesmo ecrã para inserir os dados.
---------------------------	---

<i>Pós-condições:</i>	Em caso de sucesso, o utilizador é redirecionado para uma nova interface onde pode adicionar elementos ao formulário.
-----------------------	---

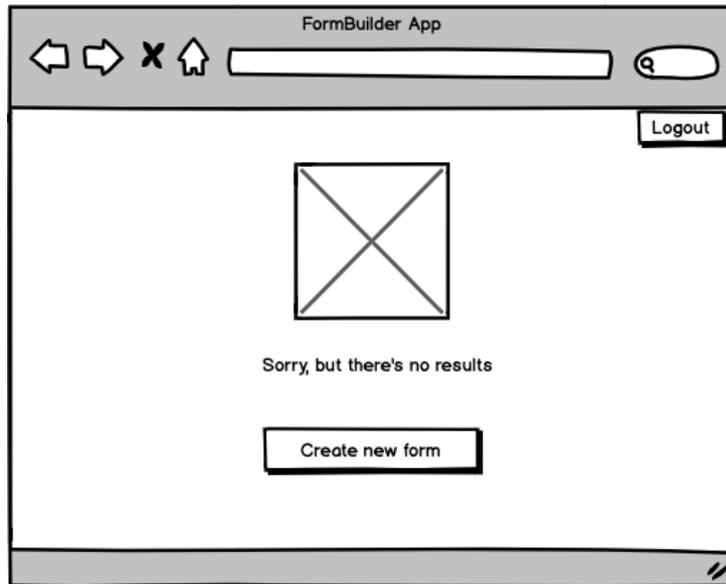


Figura A.3: Criar novo formulário

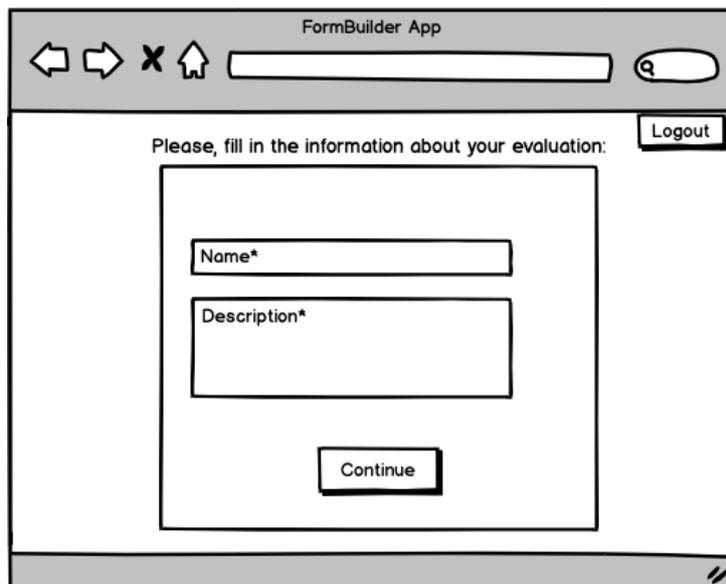


Figura A.4: Preencher dados do formulário

UC-04:	Adicionar elementos ao formulário
<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende adicionar elementos ao formulário.
<i>Atores:</i>	Administrador
<i>Pré-condições:</i>	Criar um formulário
<i>Fluxo esperado:</i>	
<ol style="list-style-type: none"> 1. O sistema apresenta um menu com vários tipos de opções que podem ser adicionadas; 2. O utilizador escolhe o elemento que pretende adicionar; 3. O utilizador clica numa das opções para adicionar o elemento ao formulário. 	
<i>Fluxo alternativo:</i>	
<i>Pós-condições:</i>	Em caso de sucesso, um novo elemento é adicionado ao formulário.

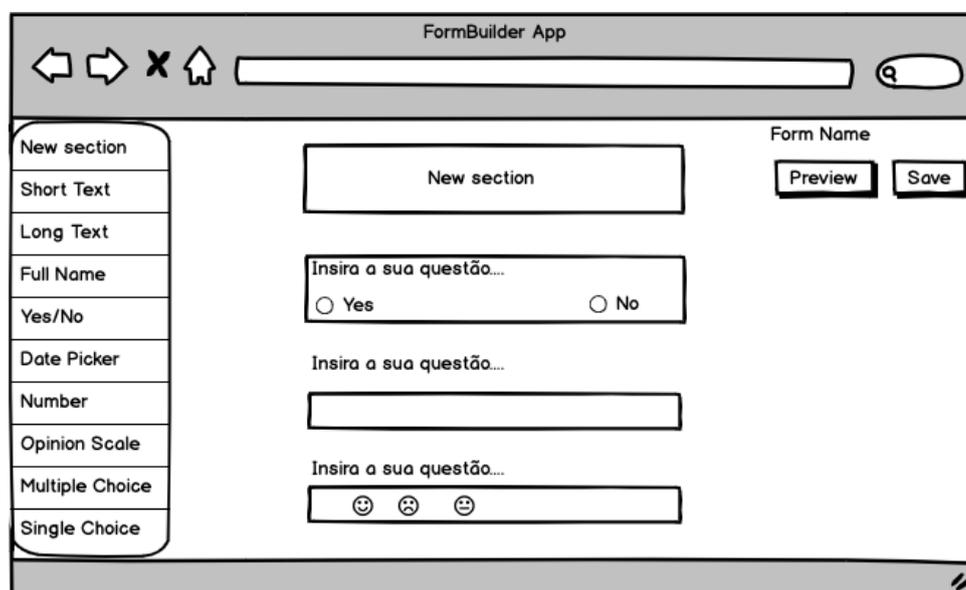


Figura A.5: Elementos adicionados: *New section*, *Yes/No*, *ShortText*, *OpinionScale*

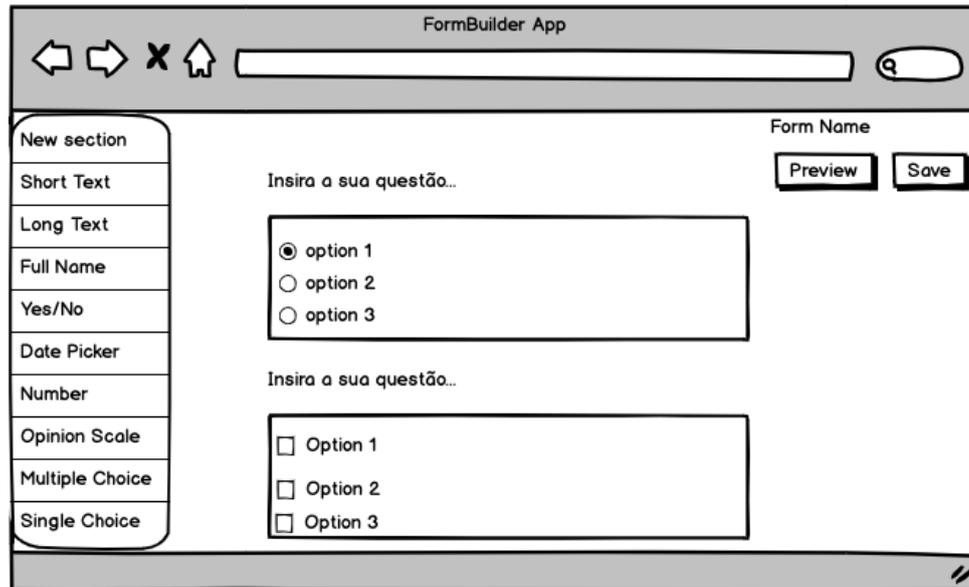


Figura A.6: Elementos adicionados: *Single Choice* e *Multiple Choice*

UC-05:	Eliminar elementos
<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende apagar elementos do formulário.
<i>Atores:</i>	Administrador
<i>Pré-condições:</i>	Criar um formulário e adicionar um ou mais elemento(s).
<i>Fluxo esperado:</i>	<ol style="list-style-type: none"> 1. O utilizador escolhe o elemento que pretende apagar; 2. O utilizador seleciona a opção para apagar o elemento.
<i>Fluxo alternativo:</i>	
<i>Pós-condições:</i>	Em caso de sucesso, o elemento é apagado do formulário.

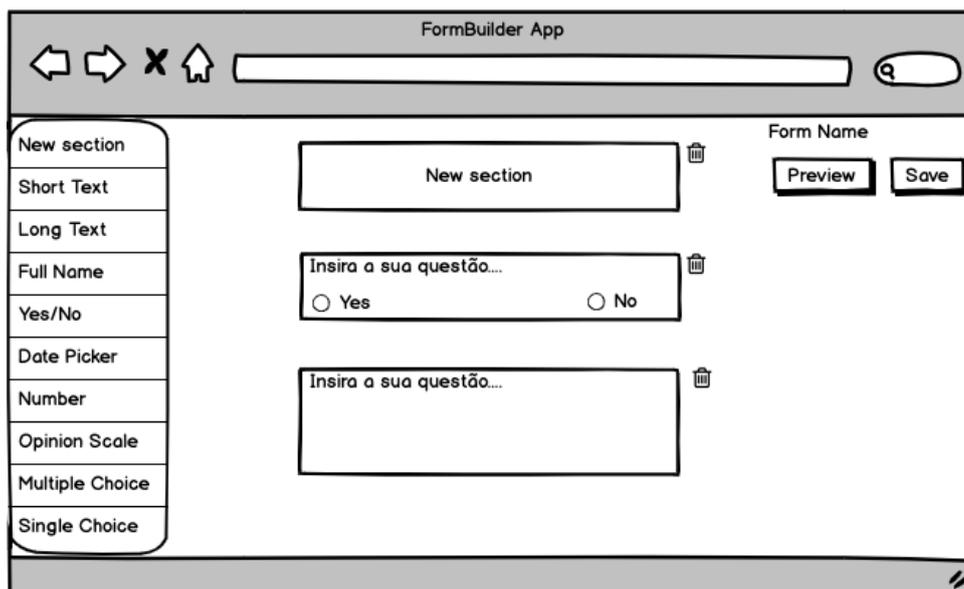


Figura A.7: Apagar elementos do formulário

UC-06: Duplicar elementos

Descrição: O caso de uso inicia-se quando o utilizador pretende adicionar um elemento igual a um pré existente no formulário.

Atores: Administrador

Pré-condições:

1. Criar um formulário no sistema;
 2. Adicionar um elemento ao formulário.
-

Fluxo esperado:

1. O utilizador seleciona o elemento que pretende duplicar;
 2. O utilizador clica na opção que permite duplicar o elemento.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o sistema adiciona um elemento igual a um pré-existente.

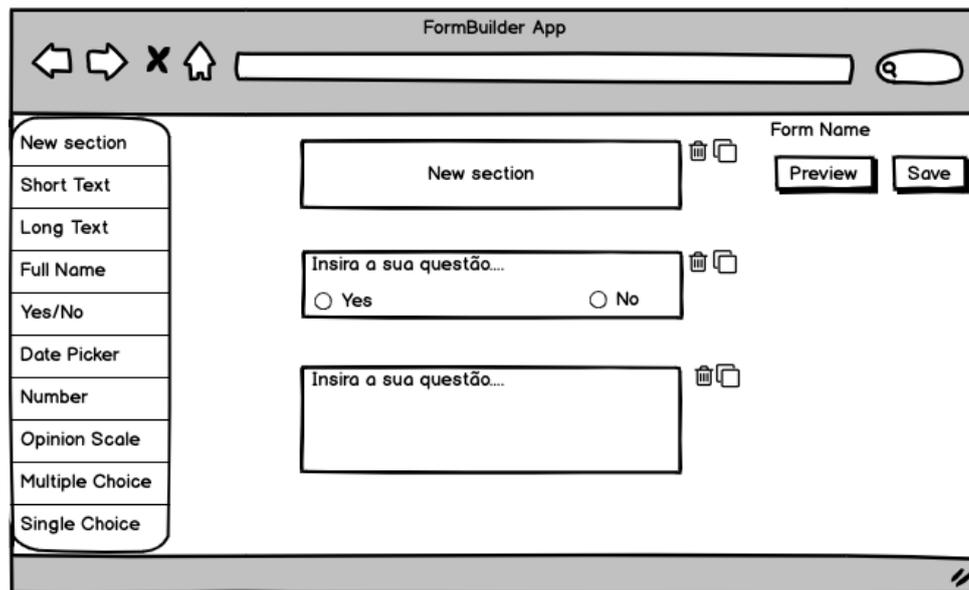


Figura A.8: Duplicar elemento

UC-07: Adicionar opção de resposta

Descrição: O caso de uso inicia-se quando o utilizador pretende adicionar mais opções de resposta além das existentes. (Aplicável em questões que podem ter várias opções, como de escolha múltipla.

Atores: Administrador

Pré-condições:

1. Criar formulário no sistema;
 2. Adicionar um ou mais elementos que contêm várias opções de resposta.
-

Fluxo esperado:

1. O sistema mostra uma opção para adicionar novas opções de resposta;
 2. O utilizador clica para adicionar nova opção.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, as opções de resposta são adicionadas.

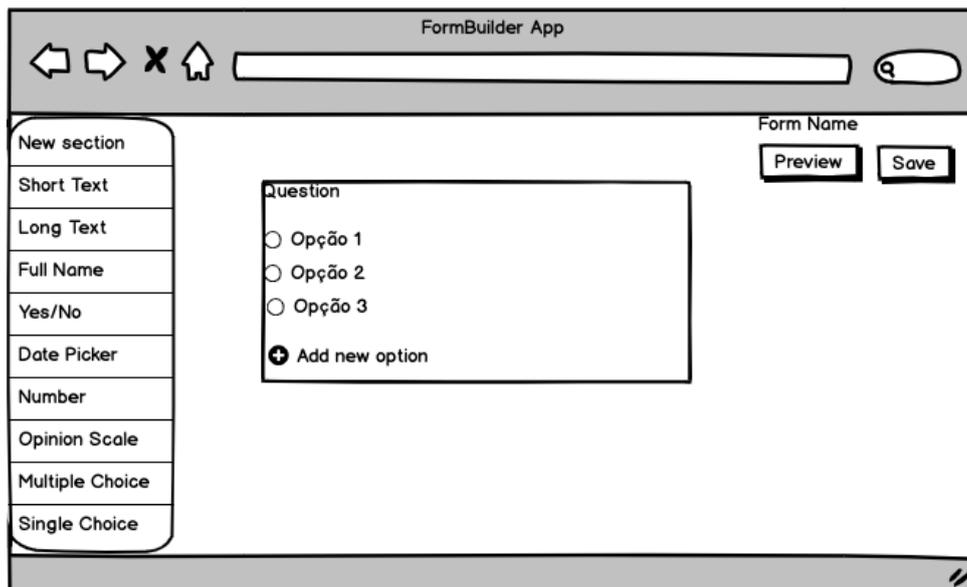


Figura A.9: Adicionar opção de resposta

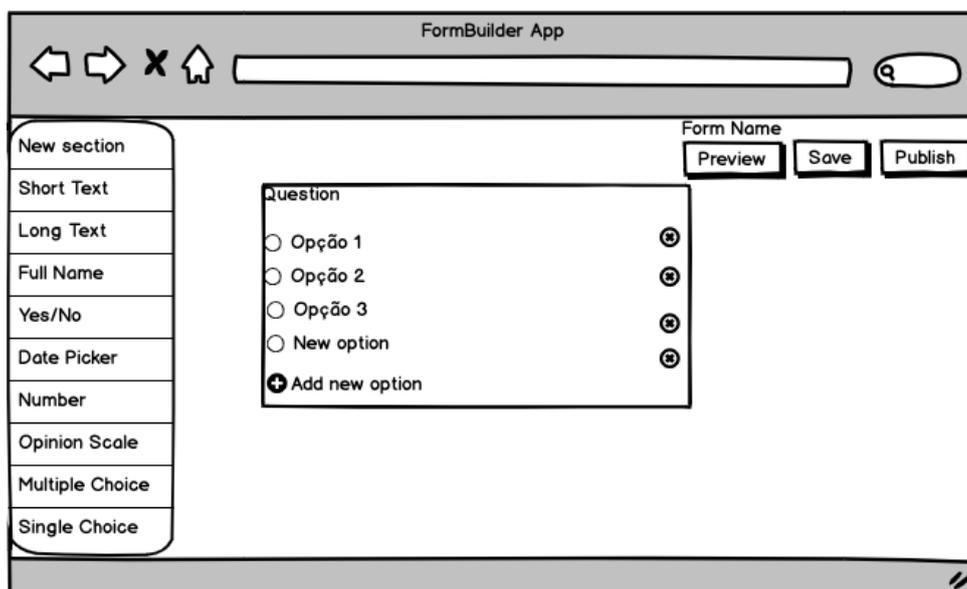


Figura A.10: Opção adicionada

UC-08: Eliminar opção de resposta

Descrição: O caso de uso inicia-se quando o utilizador pretende eliminar uma opção de resposta.

Atores: Administrador

Pré-condições:

1. Criar um formulário no sistema;
 2. Adicionar questões com várias opções.
-

Fluxo esperado:

1. O sistema mostra uma opção para eliminar opção de resposta;
 2. O utilizador clica na opção para eliminar a opção.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, a opção é eliminada.

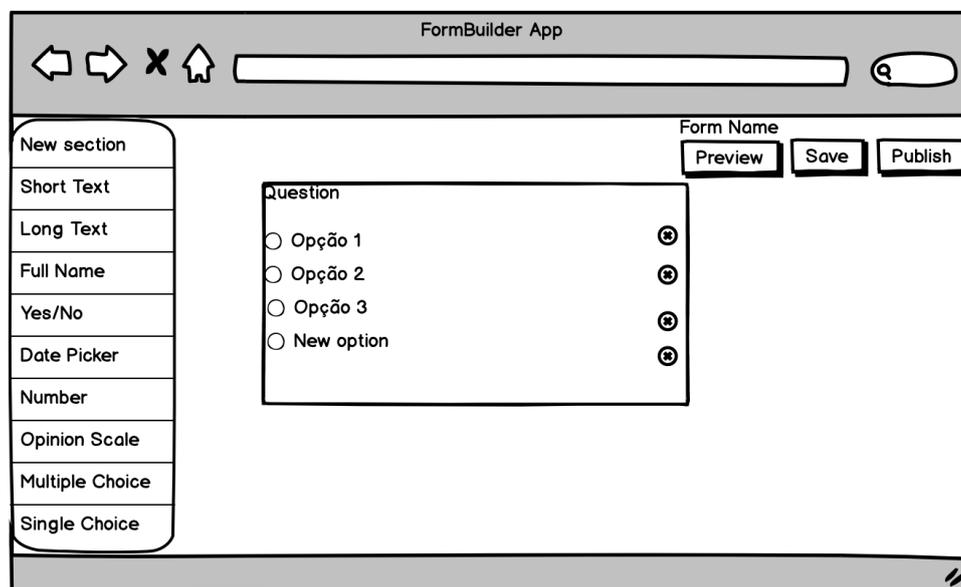


Figura A.11: Eliminar opções

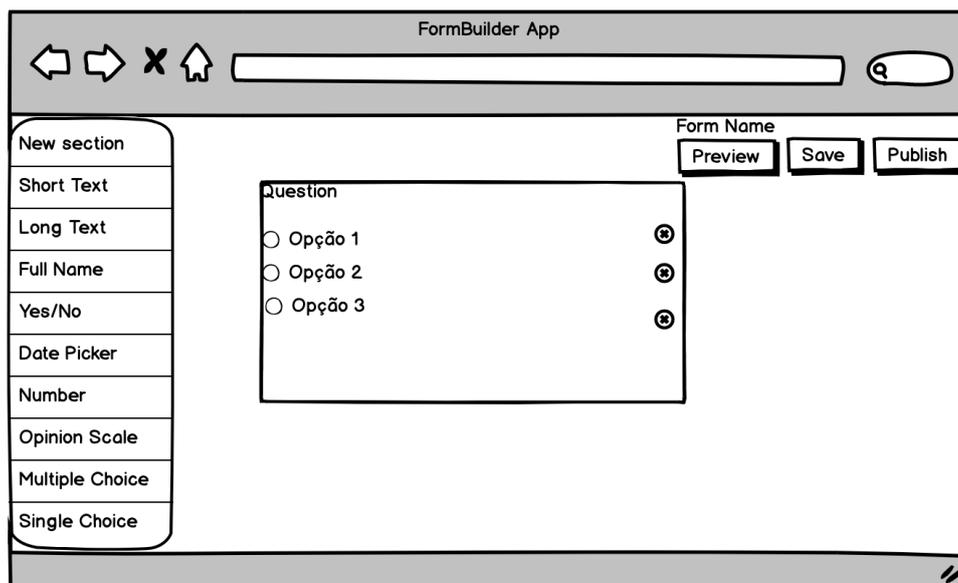


Figura A.12: Opção eliminada

UC-09: Pré-visualizar formulário

Descrição: O caso de uso inicia-se quando o utilizador pretende pré-visualizar um formulário, antes deste estar finalizado.

Atores: Administrador

Pré-condições:

1. Criar um formulário no sistema;
 2. Adicionar elementos ao formulário.
-

Fluxo esperado:

1. O sistema mostra uma opção para pré-visualizar o formulário;
 2. O utilizador clica na opção para pré-visualizar o formulário;
 3. O utilizador clica na opção para pré-visualizar.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o sistema apresenta uma pré-visualização do formulário.

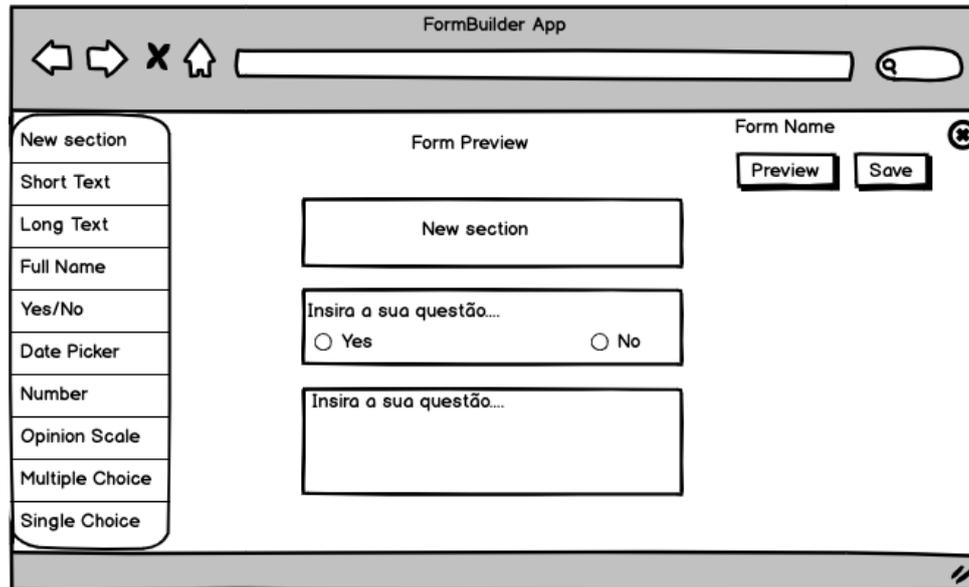


Figura A.13: Pré-visualização de formulário

UC-010: Guardar formulários

Descrição: O caso de uso inicia-se quando o administrador pretende guardar o formulário no sistema.

Atores: Administrador

Pré-condições:

1. Criar um formulário no sistema.
-

Fluxo esperado:

1. O sistema mostra uma opção para guardar o formulário;
 2. O utilizador clica na opção para guardar formulário.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o formulário é guardado no sistema.

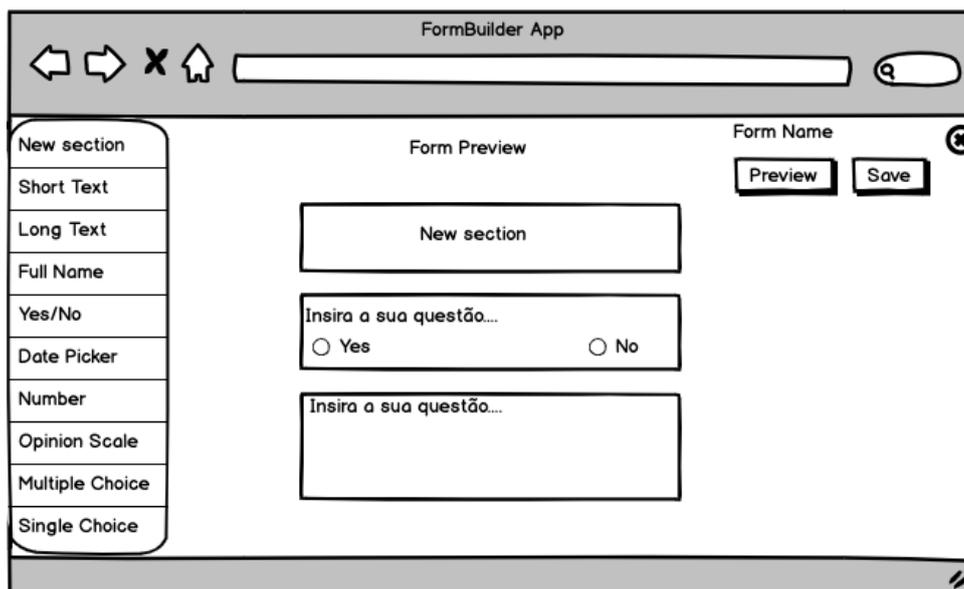


Figura A.14: Guardar formulário

UC-011: Consultar formulários

Descrição: O caso de uso inicia-se quando o utilizador pretende ver a lista de formulários criados.

Atores: Administrador

Pré-condições:

1. Criar um formulário no sistema.
-

Fluxo esperado:

1. O utilizador clica na opção para guardar formulário;
 2. O sistema redireciona o utilizador para a sua página inicial;
 3. O sistema mostra a lista de formulários existentes.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o utilizador consegue visualizar todos os formulários criados.

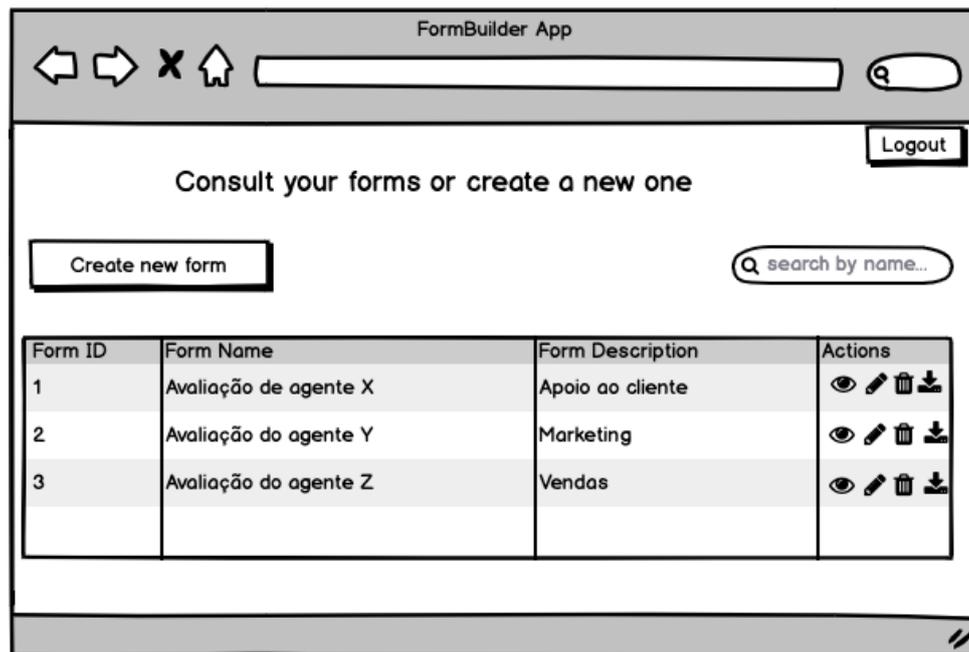


Figura A.15: Lista de formulários criados

UC-012: Abrir formulário

Descrição: O caso de uso inicia-se quando o utilizador pretende ver um dos formulários existentes no sistema.

Atores: Administrador

Pré-condições:

1. Criar um formulário no sistema.
-

Fluxo esperado:

1. O utilizador seleciona o formulário que pretende ver;
 2. O utilizador clica na opção para abrir formulário;
 3. O sistema apresenta o formulário selecionado.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o utilizador consegue visualizar o formulário.

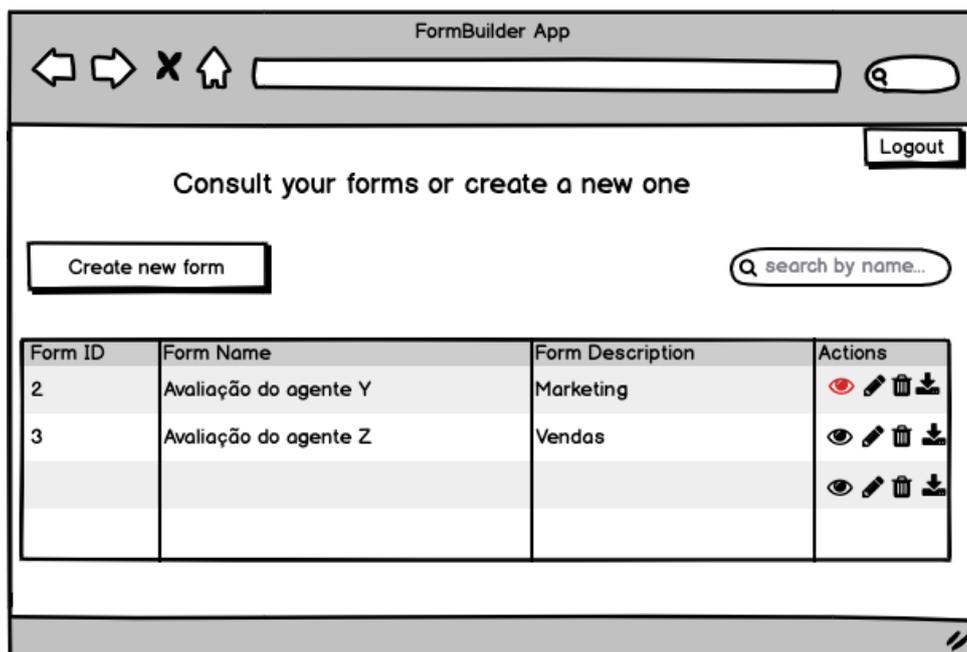


Figura A.16: Ver formulário

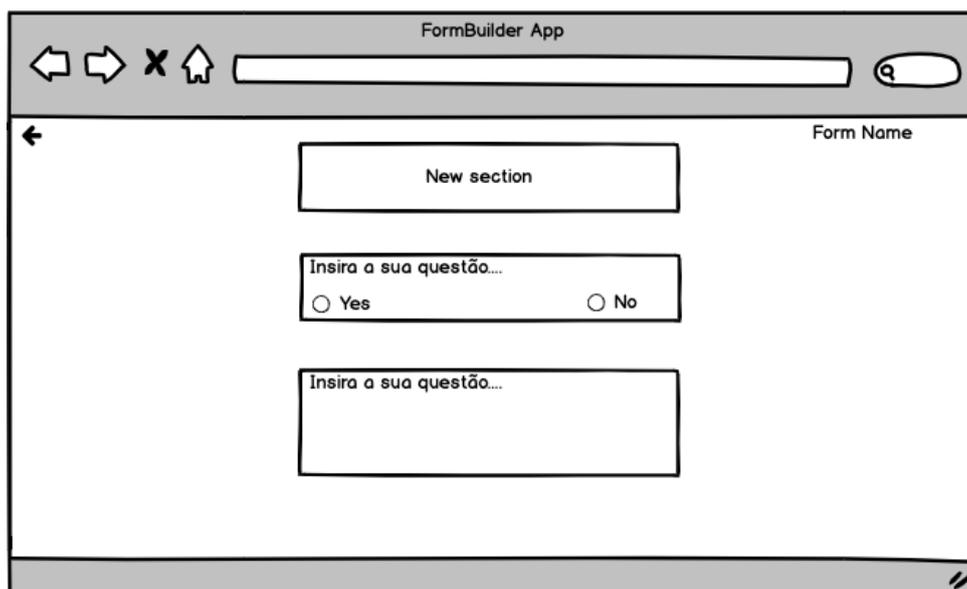


Figura A.17: Formulário

UC-013: Apagar formulários

Descrição: O caso de uso inicia-se quando o utilizador pretende remover permanentemente um determinado formulário do sistema.

Atores: Administrador

Pré-condições:

1. O administrador criou um formulário no sistema;

Fluxo esperado:

1. O sistema mostra uma opção para eliminar formulário;
2. O utilizador clica na opção para eliminar formulário;
3. O sistema mostra um alerta para o utilizador confirmar que pretende mesmo apagar o formulário;
4. O utilizador confirma que pretende apagar o formulário.

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o utilizador consegue eliminar um formulário.

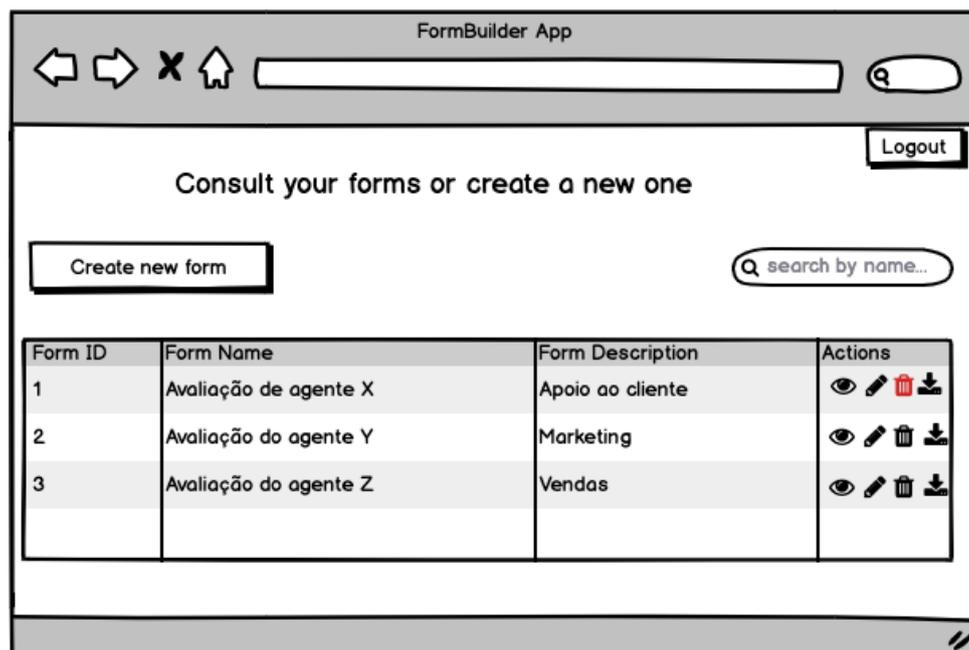


Figura A.18: Eliminar formulário

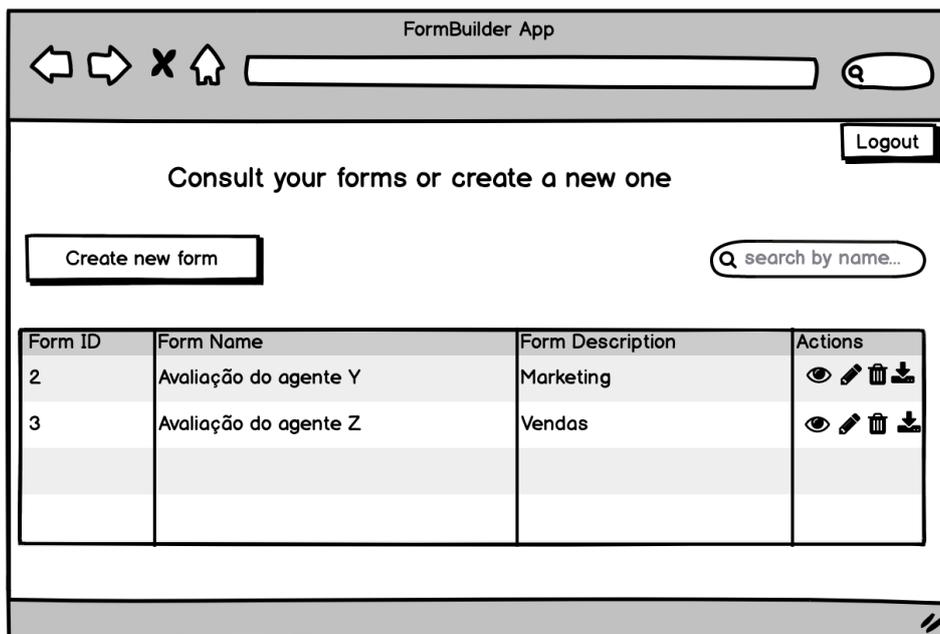


Figura A.19: Formulário eliminado

UC-014: Editar nome ou descrição do formulário

Descrição: O caso de uso inicia-se quando o utilizador pretende alterar o nome e/ou descrição de determinado formulário.

Atores: Administrador

Pré-condições: O utilizador criou um formulário no sistema.

Fluxo esperado:

1. O sistema mostra uma opção para alterar informação formulário;
 2. O utilizador clica na opção para alterar informação;
 3. O utilizador insere um novo nome e/ou nova descrição para o formulário;
 4. O utilizador clica na opção para confirmar.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o sistema altera o nome e/ou descrição do formulário.

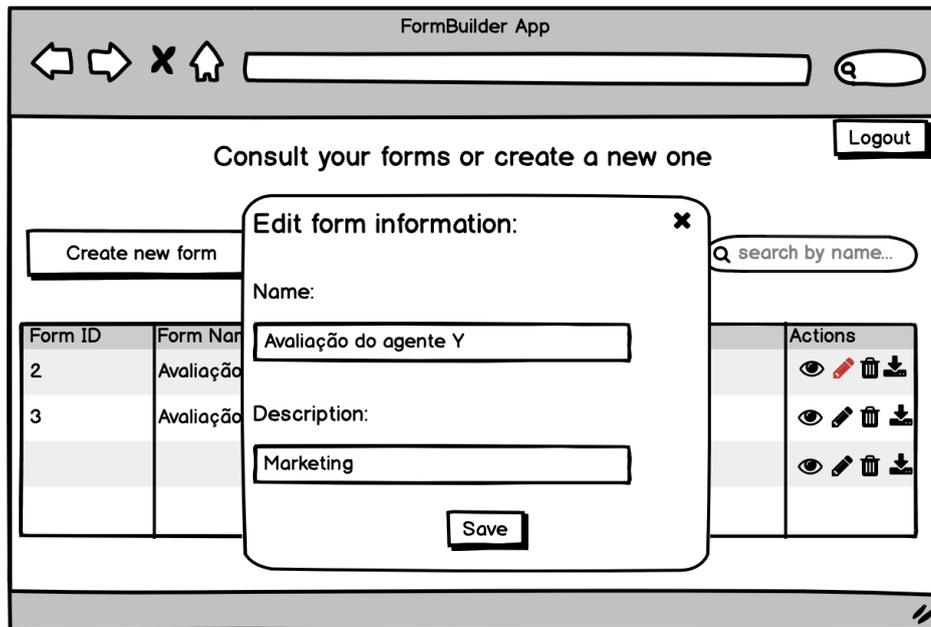


Figura A.20: Editar informação do formulário

UC-015: Filtrar formulários

Descrição: O caso de uso inicia-se quando o utilizador pretende procurar por um determinado formulário.

Atores: Administrador

Pré-condições:

1. O utilizador criou um formulário previamente no sistema.
-

Fluxo esperado:

1. O sistema mostra uma opção para pesquisar pelo nome do formulário;
 2. O utilizador insere o nome do formulário que pretende encontrar.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o sistema consegue filtrar os formulários correspondentes ao nome pesquisado pelo utilizador.

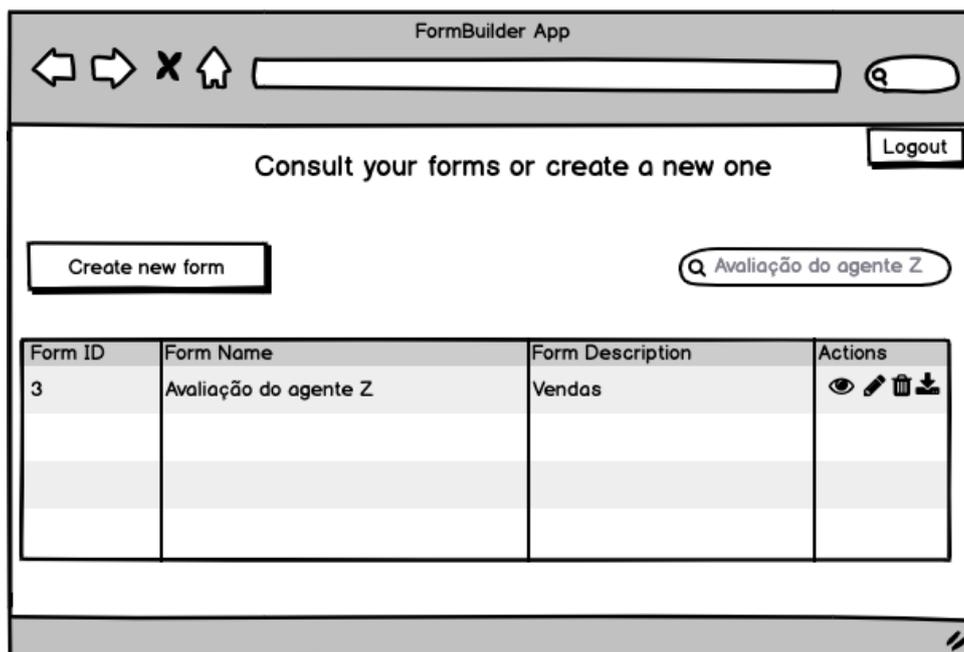


Figura A.21: Pesquisar pelo nome do formulário

UC-016: Publicar formulários

Descrição: O caso de uso inicia-se quando o utilizador pretende publicar um formulário, para que este fique disponível para preenchimento.

Atores: Administrador

Pré-condições:

1. Criar um ou mais formulários no sistema.
-

Fluxo esperado:

1. O sistema mostra uma opção para publicar formulário;
 2. O administrador clica na opção para publicar formulário.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o formulário é publicado e visível na página inicial do supervisor.

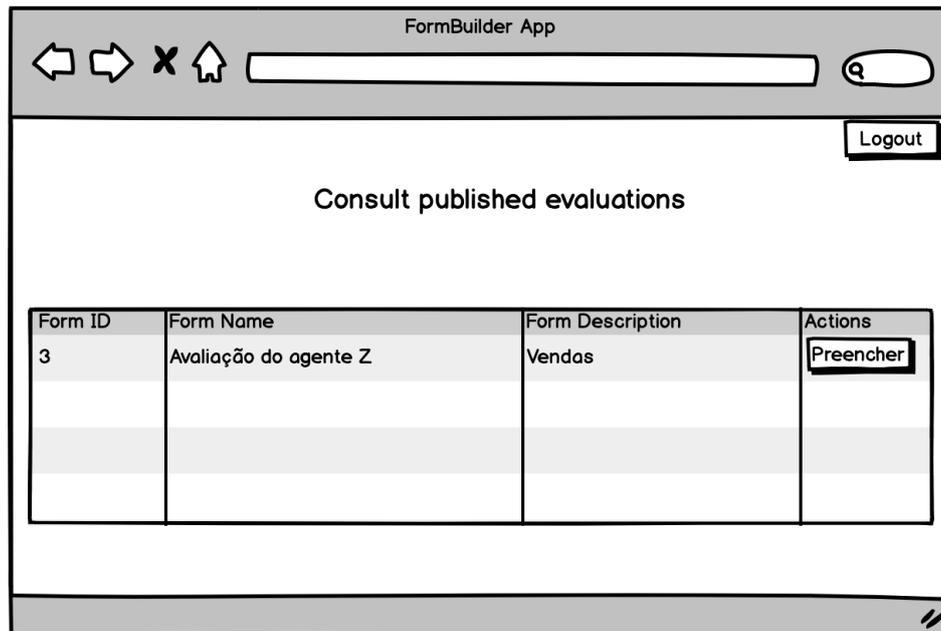


Figura A.22: Publicar formulário

UC-017: Consultar formulários publicados

Descrição: O caso de uso inicia-se quando o utilizador pretende visualizar os formulários publicados no sistema.

Atores: Supervisor

Pré-condições:

1. O administrador criou um ou mais formulários no sistema;
 2. O administrador publicou um ou mais formulários no sistema;
 3. O supervisor está autenticado na aplicação.
-

Fluxo esperado:

1. O sistema apresenta a lista de formulários disponíveis quando o supervisor acede à aplicação.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o supervisor consegue ver todos os formulários publicados pelo administrador.

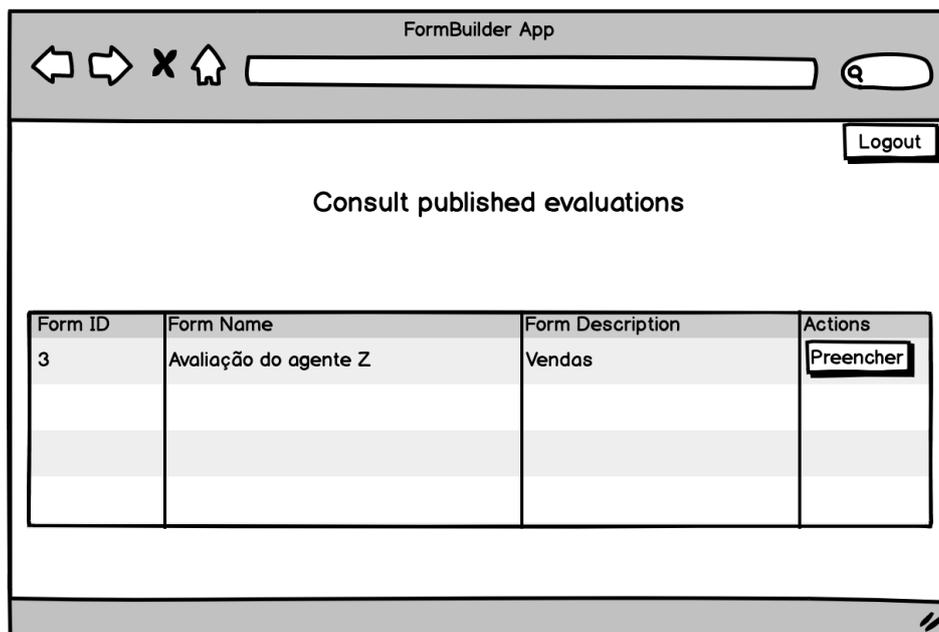


Figura A.23: Lista de formulários publicados

UC-018: Preencher formulários

Descrição: O caso de uso inicia-se quando o utilizador pretende preencher os formulários publicados.

Atores: Supervisor

Pré-condições:

1. O administrador criou um ou mais formulários;
 2. O supervisor autenticou-se na aplicação.
-

Fluxo esperado:

1. O sistema mostra a lista de formulários existentes;
 2. O utilizador escolhe o formulário que pretende preencher;
 3. O utilizador clica no formulário que pretende;
 4. O sistema apresenta o formulário.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o formulário é preenchido com sucesso.

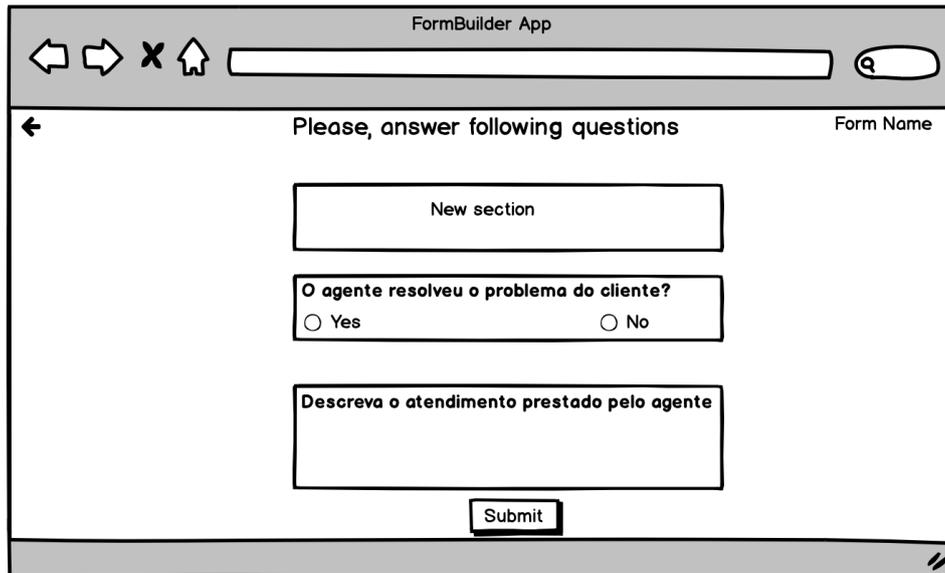


Figura A.24: Preencher formulário

UC-019: Visualizar modelos de formulários

Descrição: O caso de uso inicia-se quando o utilizador pretende visualizar modelos pré-criados de formulários.

Atores: Administrador

Pré-condições:

1. O sistema contém modelos pré-criados de formulários;
 2. O utilizador está autenticado no sistema.
-

Fluxo esperado:

1. O sistema mostra a lista de modelos existentes.
 2. O utilizador clica no formulário que pretende visualizar.
 3. O sistema apresenta o modelo de formulário ao utilizador.
-

Fluxo alternativo:

Pós-condições: Em caso de sucesso, o utilizador consegue ver o modelo de formulário.

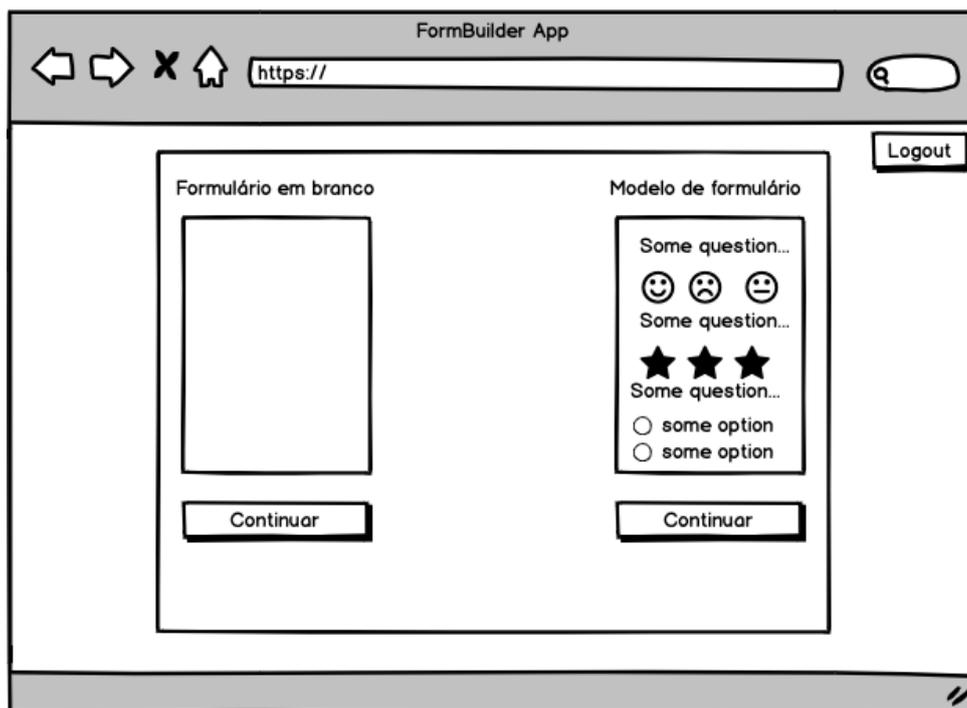


Figura A.25: Escolher o tipo de formulário

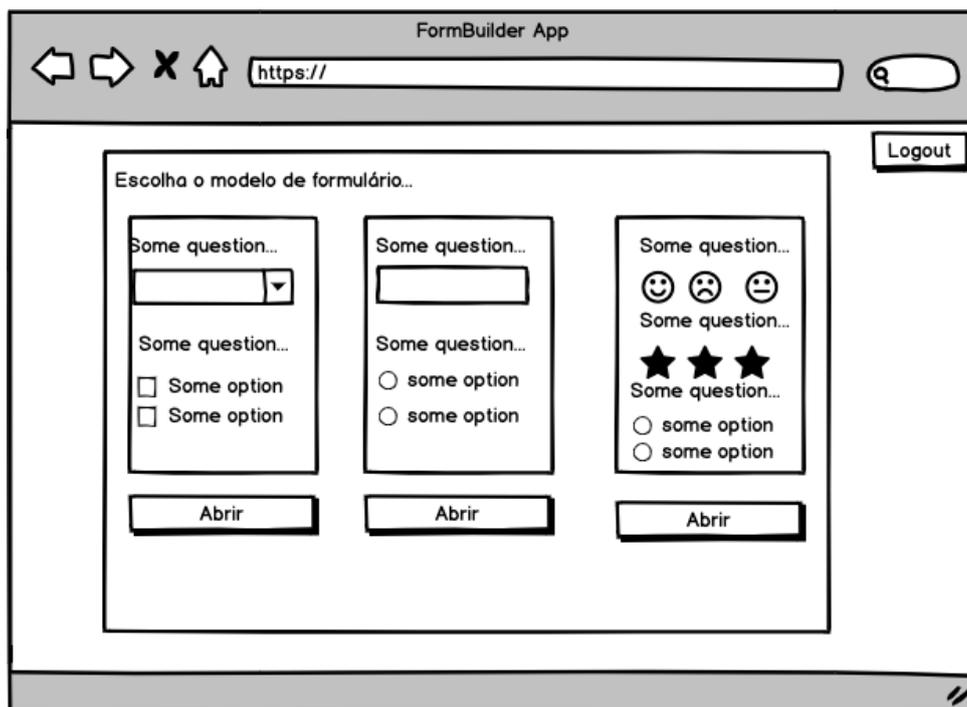


Figura A.26: Selecionar modelo de formulário

UC-020:	Submeter formulários
<i>Descrição:</i>	O caso de uso inicia-se quando o utilizador pretende submeter os formulários depois de preenchidos.
<i>Atores:</i>	Supervisor
<i>Pré-condições:</i>	<ol style="list-style-type: none">1. O sistema contém formulários publicados;2. O utilizador preenche o formulário.
<i>Fluxo esperado:</i>	<ol style="list-style-type: none">1. O utilizador clica para submeter formulário;2. O formulário é submetido e a avaliação é comunicada ao agente.
<i>Fluxo alternativo:</i>	
<i>Pós-condições:</i>	Em caso de sucesso, o formulário é submetido e o agente recebe a avaliação.



Figura A.27: Formulário submetido

Anexo B

Manual de Utilizador

B.1 Introdução

O presente documento apresenta todas as funcionalidades da aplicação, assim como as instruções para a sua utilização.

Os dados presentes nas imagens seguintes são fictícios, sendo apenas utilizados como exemplos para demonstrar as funcionalidades implementadas.

B.2 Efetuar login

B.2.1 Página de Login

Quando o utilizador acede à plataforma, é redirecionado para a página de login. A partir desta, pode inserir os seus dados de acesso para aceder à aplicação. Dependendo das suas permissões, este pode ter acesso como administrador, para criar os modelos de formulários, ou como supervisor, para efetuar as avaliações.

Para tal, tem de inserir o *username* e a *password*, associadas à sua conta (Figura B.1).

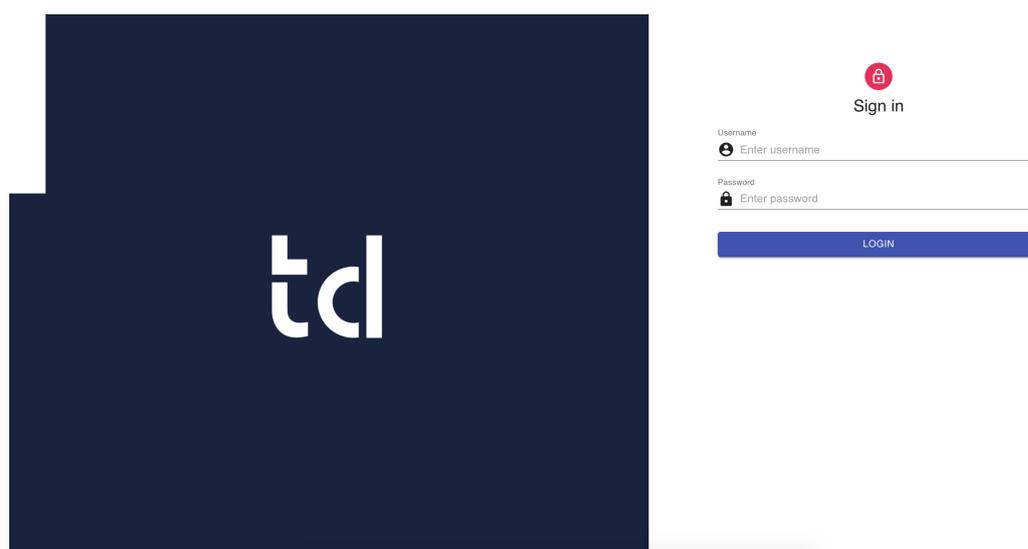


Figura B.1: Página de login

B.2.2 Login - Dados incorretos

Quando o utilizador tenta autenticar-se com dados incorretos, isto é, que não existem no sistema, é apresentada uma mensagem de erro para que este verifique os seus dados (Figura B.2). Assim, não é possível aceder à aplicação, enquanto não introduzir os dados corretos.

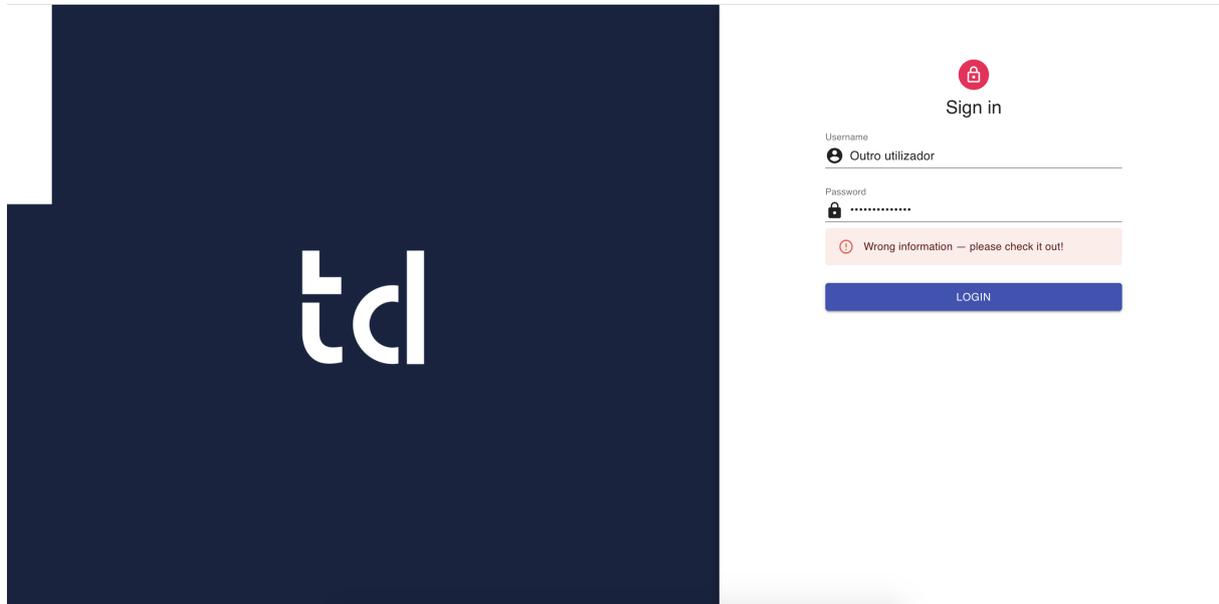


Figura B.2: Página de login

B.3 Página inicial do Administrador

Quando insere os seus dados de acesso de administrador, é direcionado para a sua página inicial (Figura B.3).

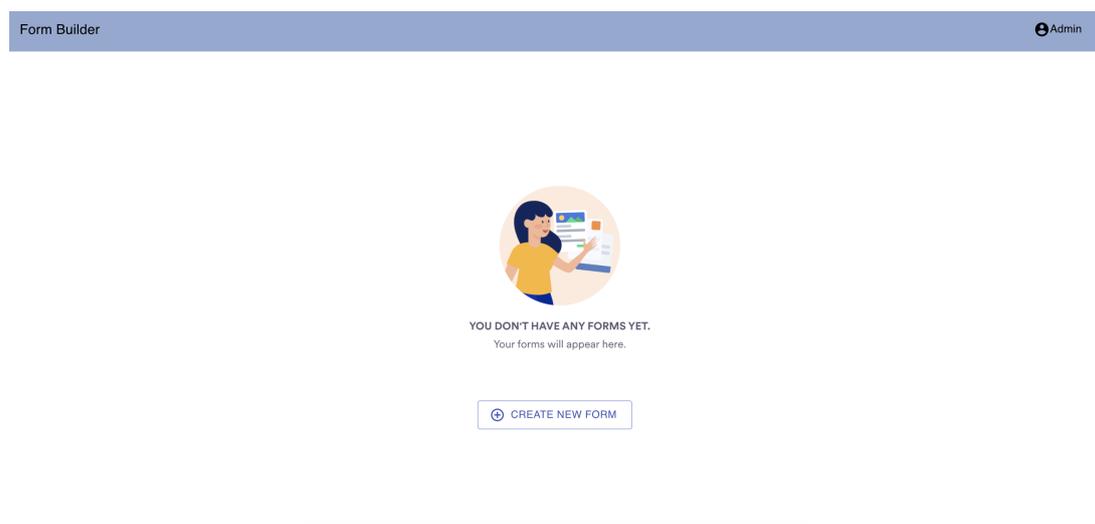


Figura B.3: Página inicial do administrador

Se ainda não existirem quaisquer formulários criados, o sistema apresenta uma mensagem informativa e também uma opção para criar um novo formulário.

B.4 Página inicial do Supervisor

Quando o utilizador insere os seus dados de acesso como supervisor, é redirecionado para a sua página inicial. Se ainda não existirem quaisquer formulários publicados, o sistema apresenta uma mensagem informativa (Figura B.4).

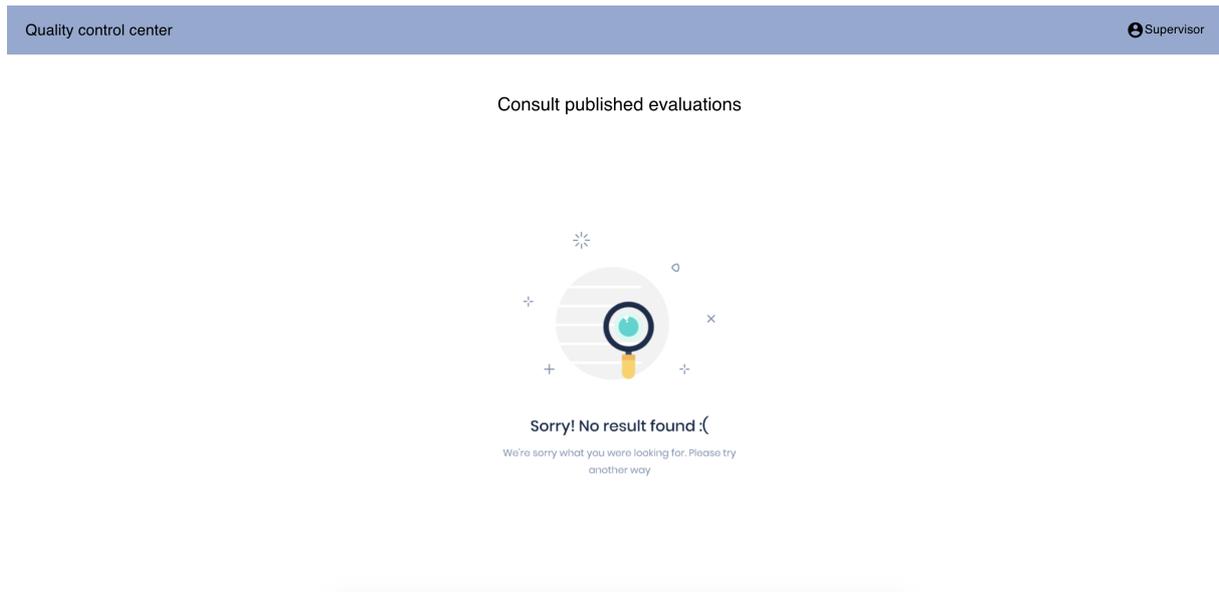


Figura B.4: Página inicial do Supervisor

B.5 Efetuar logout

Quando pretendem sair da aplicação, os utilizadores devem clicar na opção "Logout" presente na barra superior da aplicação, como se pode ver nas figuras B.5 e B.6.

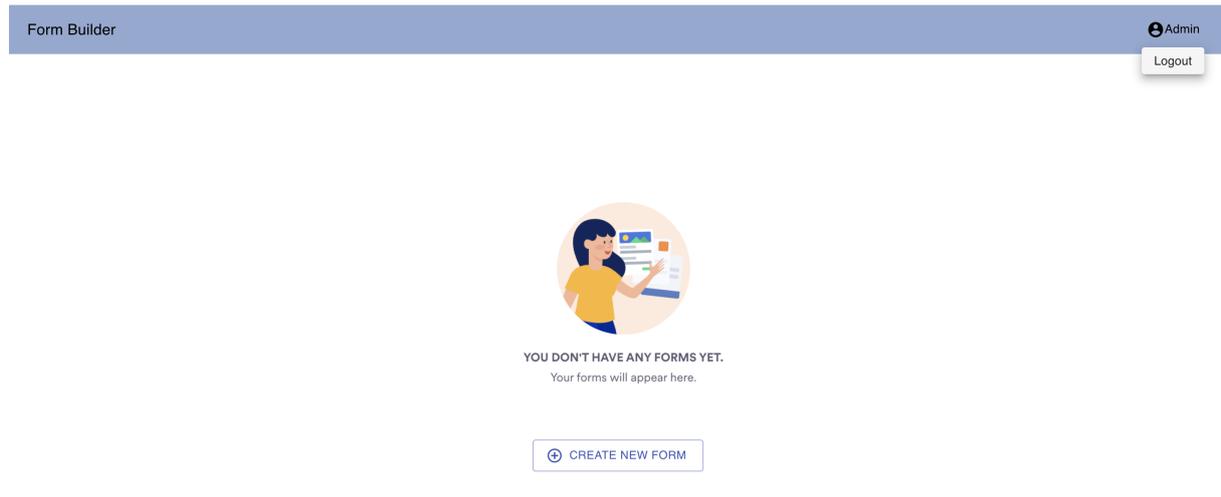


Figura B.5: Logout - Administrador

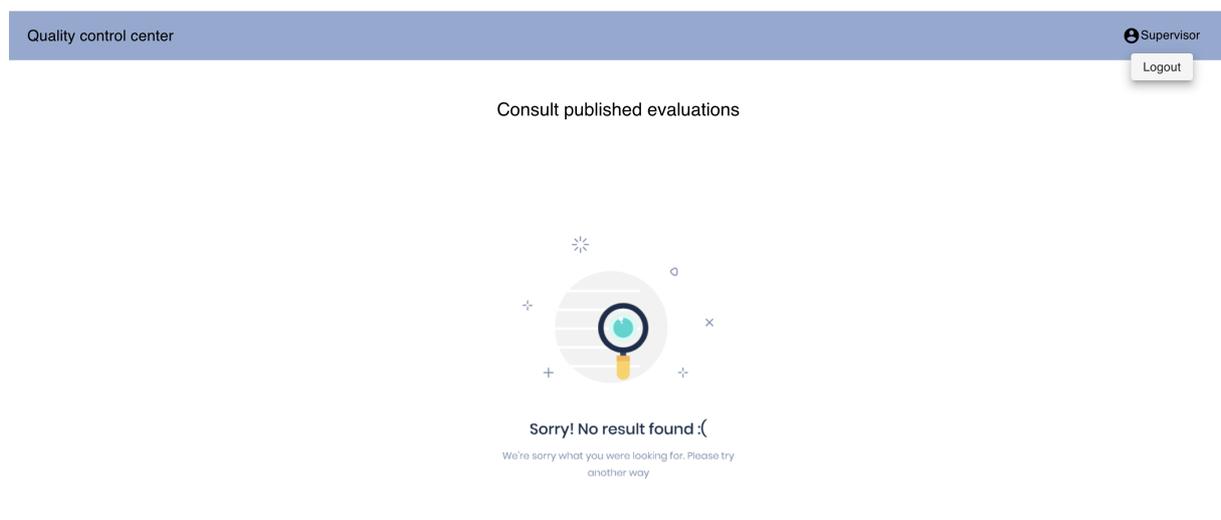


Figura B.6: Logout - Supervisor

B.6 Criar novo formulário - Administrador

Para criar um novo formulário, o utilizador deve seleccionar a opção "Create new form" (Figura B.7).

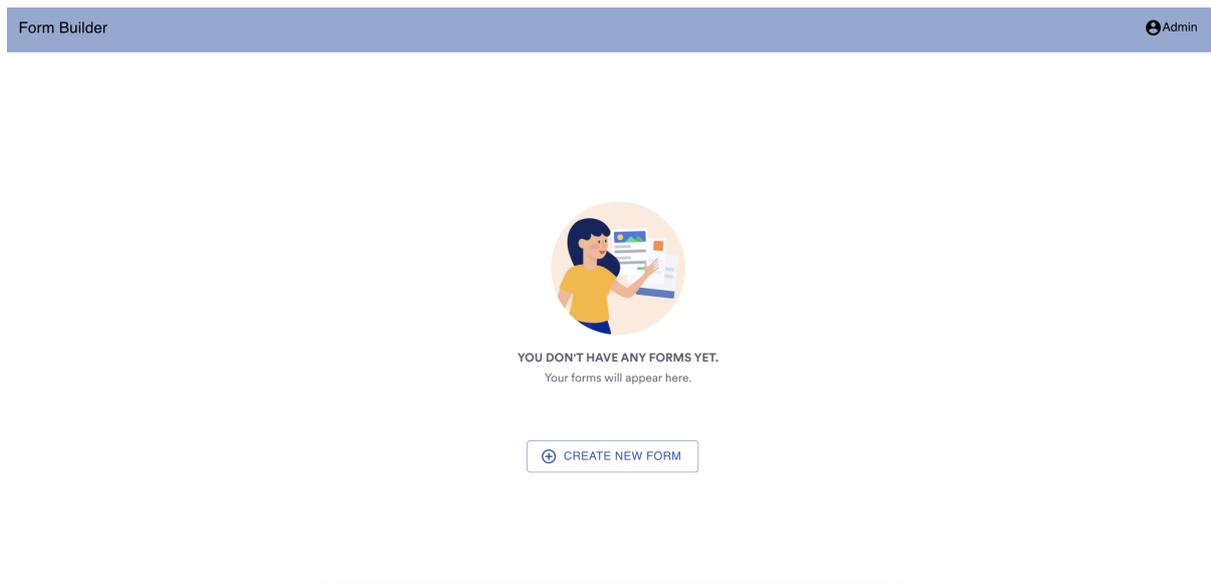


Figura B.7: Logout - Supervisor

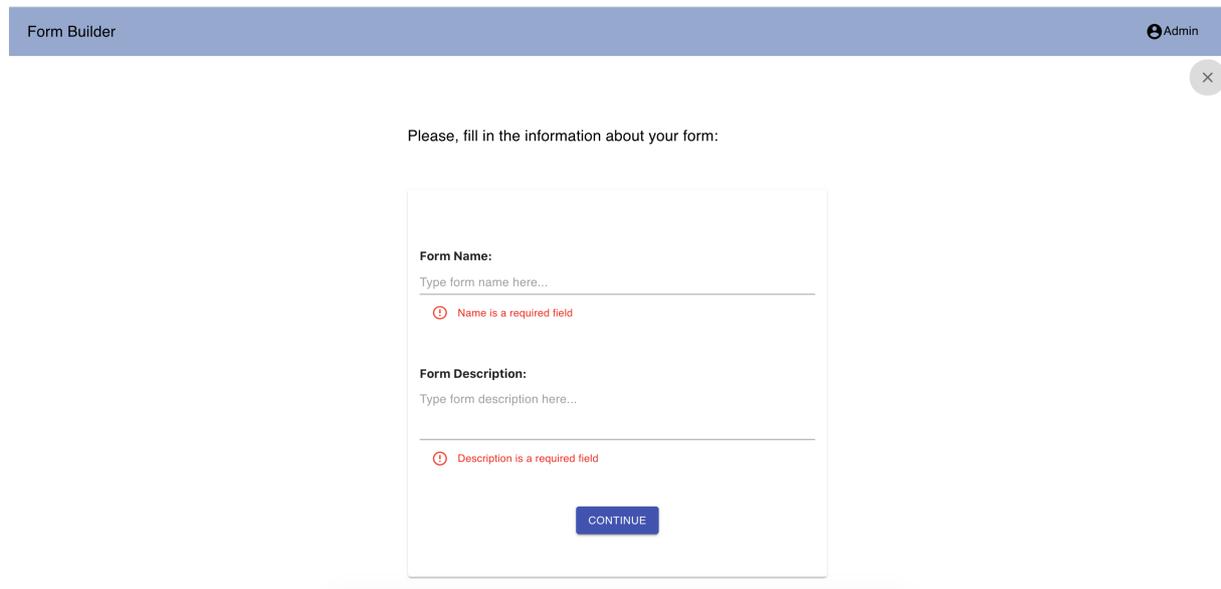
Quando clica nesta opção, é redirecionado para uma nova página (Figura B.8). Aqui, deve inserir um nome e uma descrição para o formulário.

The screenshot shows the top navigation bar with 'Form Builder' on the left and 'Admin' on the right. Below the navigation bar is a close button (X). The main content area contains the text 'Please, fill in the information about your form:'. Below this text is a white form box with a light gray border. Inside the form box, there are two sections: 'Form Name:' with a text input field containing the placeholder 'Type form name here...' and 'Form Description:' with a text input field containing the placeholder 'Type form description here...'. At the bottom of the form box is a blue button with the text 'CONTINUE'.

Figura B.8: Criar novo formulário

B.6.1 Criar formulário - Validações

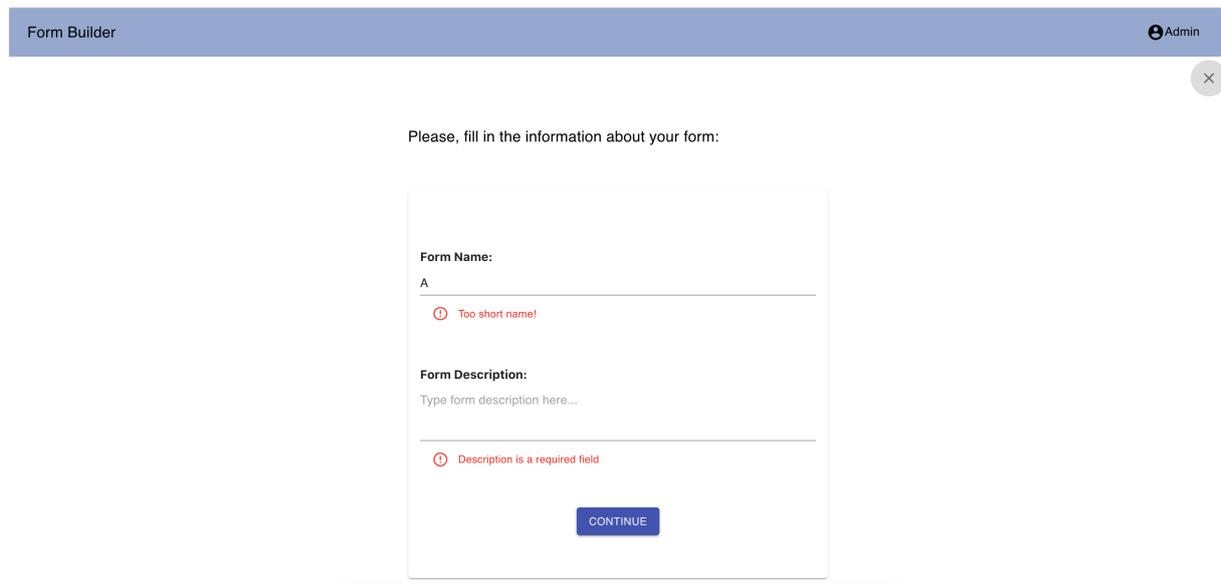
Se o utilizador tentar criar um formulário sem qualquer nome ou descrição, o sistema apresenta uma mensagem de erro pois estes campos são de preenchimento obrigatório (Figura B.9).



The screenshot shows the 'Form Builder' interface. At the top, there is a blue header bar with 'Form Builder' on the left and 'Admin' with a user icon on the right. Below the header, a grey button with an 'X' icon is visible. The main content area contains the text 'Please, fill in the information about your form:'. Below this text is a white form box with a light blue border. Inside the form box, there are two sections: 'Form Name:' and 'Form Description:'. The 'Form Name:' section has a text input field with the placeholder 'Type form name here...' and a red error message below it: 'Name is a required field'. The 'Form Description:' section has a text input field with the placeholder 'Type form description here...' and a red error message below it: 'Description is a required field'. At the bottom of the form box is a blue button labeled 'CONTINUE'.

Figura B.9: Criar novo formulário - Validação do nome e descrição

Caso o utilizador tente criar um formulário com um nome muito curto, o sistema também apresenta uma mensagem a alertar para o erro (Figura B.10).



The screenshot shows the 'Form Builder' interface. At the top, there is a blue header bar with 'Form Builder' on the left and 'Admin' with a user icon on the right. Below the header, a grey button with an 'X' icon is visible. The main content area contains the text 'Please, fill in the information about your form:'. Below this text is a white form box with a light blue border. Inside the form box, there are two sections: 'Form Name:' and 'Form Description:'. The 'Form Name:' section has a text input field containing the letter 'A' and a red error message below it: 'Too short name!'. The 'Form Description:' section has a text input field with the placeholder 'Type form description here...' and a red error message below it: 'Description is a required field'. At the bottom of the form box is a blue button labeled 'CONTINUE'.

Figura B.10: Criar novo formulário - Erro

Caso o utilizador pretenda criar um formulário cujo nome já exista num outro formulário guardado no sistema, a plataforma apresenta uma mensagem de alerta e este não é criado, até que o utilizador insira um novo nome (Figura B.11).

The screenshot shows the 'Form Builder' interface. At the top, there is a blue header with 'Form Builder' on the left and 'Admin' on the right. Below the header, a central message reads 'Please, fill in the information about your form:'. A white form box contains the following fields:

- Form Name:** 'Avaliação de Desempenho'. Below this field is a yellow warning banner with a triangle icon and the text: 'There is another form with the same name! Please change it.'
- Form Description:** 'Type form description here...'. Below this is a horizontal line.

At the bottom of the form box is a blue button labeled 'CONTINUE'.

Figura B.11: Criar novo formulário - Erro

B.7 Página para criar formulário

Após inserir o nome e descrição do formulário corretamente, o utilizador é redirecionado para uma nova página onde pode adicionar elementos de diferentes tipos, a partir do menu lateral (Figura B.12).

The screenshot shows the 'Form Builder' interface with the 'Form Elements' menu open on the left. The menu items are:

- New Section
- Short text
- Long text
- Full name
- Yes No
- Date Picker
- Time
- Number
- Opinion Scale
- Favorite Rating
- Star Rating
- Slider Scale
- Nps
- Single Choice
- Multiple Choice

The main area shows the form preview for 'Avaliação de desempenho - Apoio ao cliente'. At the top right of the preview area, there is a blue header with 'Admin'. Below the header, the form title 'Avaliação de desempenho - Apoio ao cliente' is displayed. At the bottom right of the preview area, there are two buttons: 'PREVIEW' and 'SAVE'.

Figura B.12: Página para criar formulário

B.8 Adicionar elementos ao formulário

A partir do menu existente no lado esquerdo da página, é possível adicionar elementos ao formulário. Estes podem ser utilizados para organizar o formulário, como por exemplo, através de secções, ou para criar diferentes tipos de questões e respostas, como elementos de escolha múltipla, texto, classificação, entre outras.

Nas figuras seguintes estão representados exemplos dos elementos que podem ser adicionados nos questionários, assim como a sua possível utilização, através da definição de diferentes questões.

B.8.1 New section

Se seleccionar a opção "New section", uma nova secção é adicionada ao formulário. Este elemento pode ser utilizado para dividir as questões em diferentes grupos, para facilitar a organização do formulário, tal como está ilustrado na figura B.13.

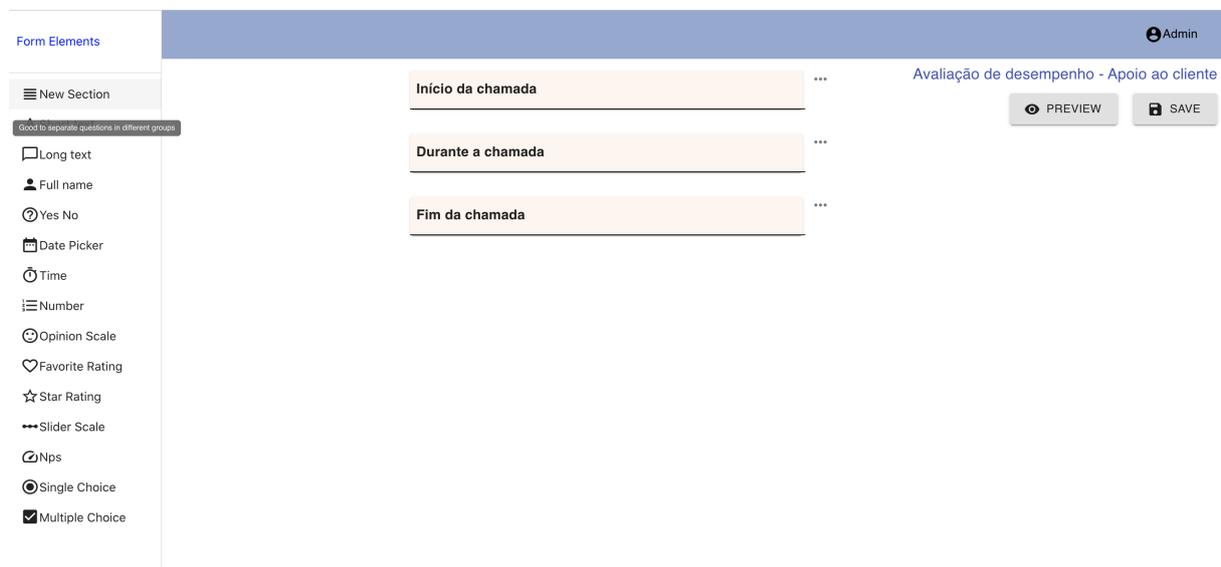


Figura B.13: Adicionar elementos - "New section"

B.8.2 Short Text

Se seleccionar a opção "Short Text", um novo elemento é adicionado ao formulário. Este pode ser utilizado para obter respostas curtas (Figura B.14).

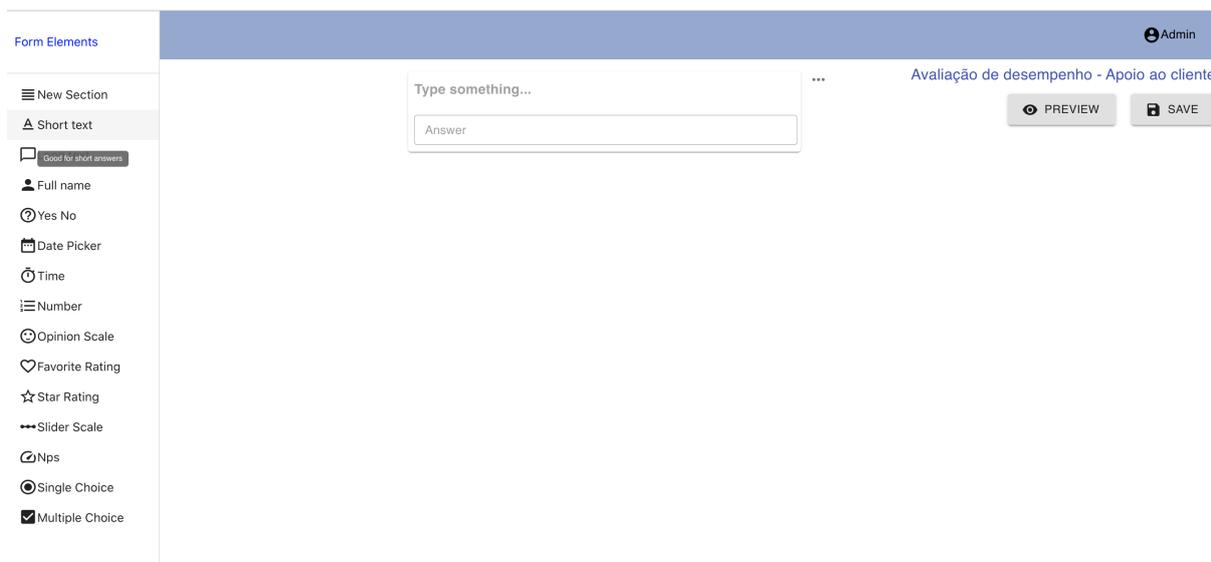


Figura B.14: Adicionar elementos - "Short Text"

B.8.3 Long Text

Se seleccionar a opção "Long Text", é adicionado um elemento que permite obter respostas longas (Figura B.15).

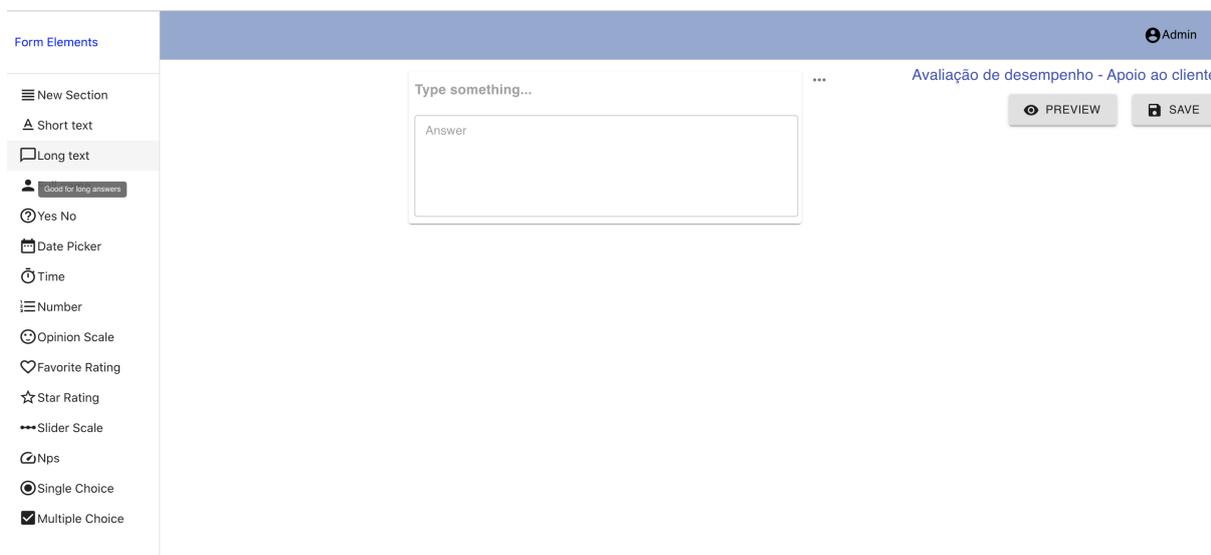


Figura B.15: Adicionar elementos - "Long Text"

B.8.4 Full Name

Se seleccionar a opção "Full Name", é adicionado um elemento que permite obter informações de contacto, tal como o primeiro e último nome (Figura B.16).

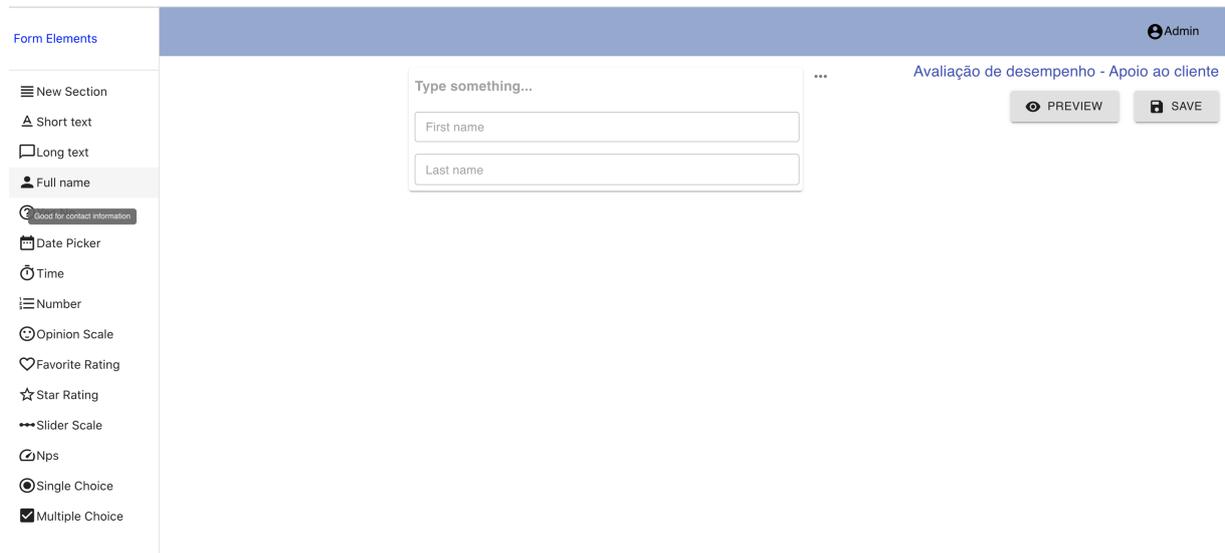


Figura B.16: Adicionar elementos - "Full Name"

B.8.5 Date Picker

Se seleccionar a opção "Date Picker", é adicionado um elemento que permite obter respostas no formato de datas (Figura B.17).

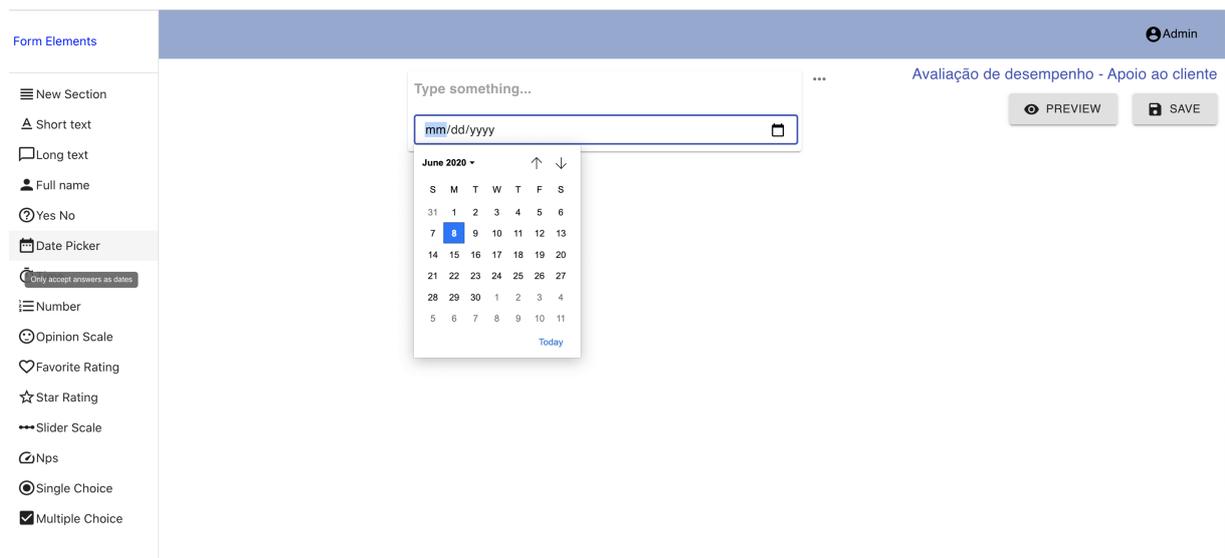


Figura B.17: Adicionar elementos - "Date Picker"

B.8.6 Time

Se seleccionar a opção "Time", é adicionado um elemento que permite obter respostas de acordo com uma escala temporal (Figura B.18).

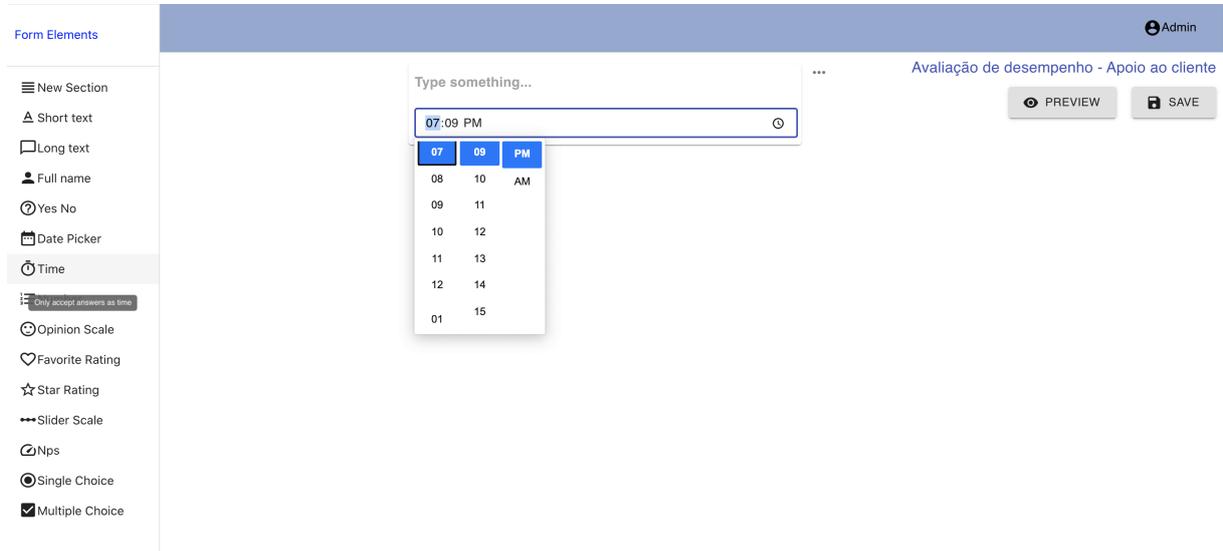


Figura B.18: Adicionar elementos - "Time"

B.8.7 Number

Se seleccionar a opção "Number", é adicionado um elemento que permite obter respostas com números (Figura B.19).

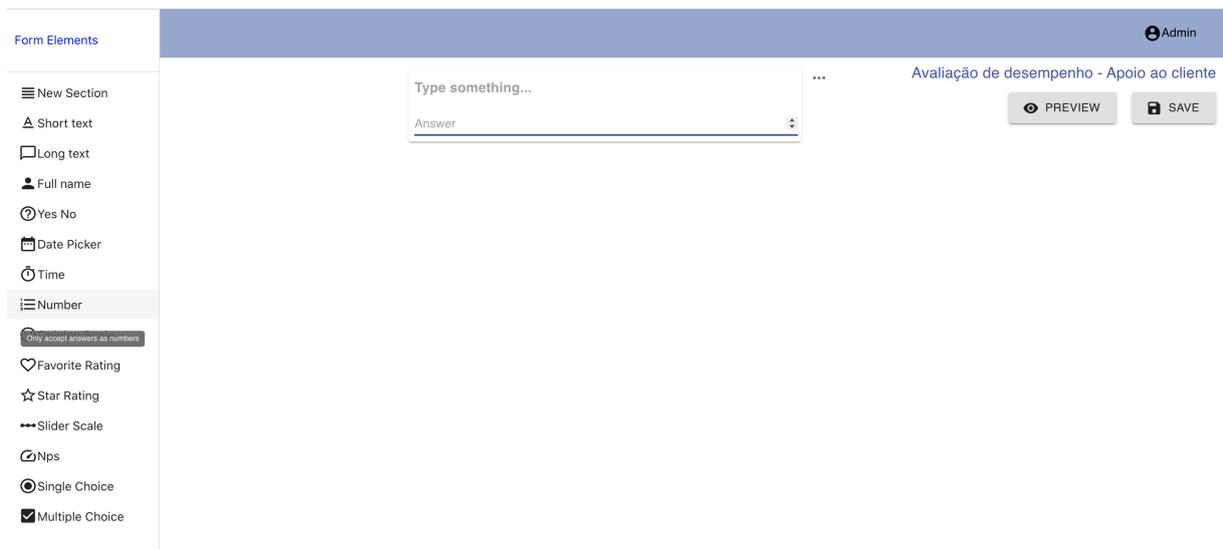


Figura B.19: Adicionar elementos - "Number"

B.8.8 Opinion Scale

Se selecionar a opção "Opinion Scale", é adicionado um elemento que permite ao utilizador expressar a sua opinião através de formas, como *emojis* (Figura B.20).

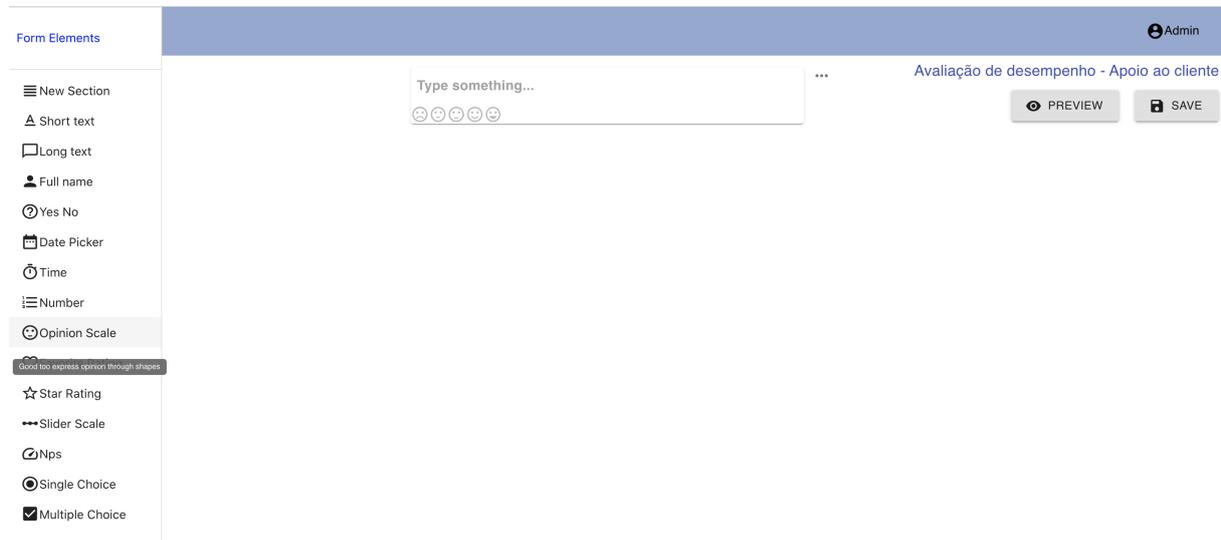


Figura B.20: Adicionar elementos - "Opinion Scale"

B.8.9 Favorite Rating

Se selecionar a opção "Favorite Rating", é adicionado um elemento que permite ao utilizador expressar a sua opinião através de formas, como corações (Figura B.21).



Figura B.21: Adicionar elementos - "Favorite Rating"

B.8.10 Star Rating

Se seleccionar a opção "Star Rating", é adicionado um elemento que permite ao utilizador expressar a sua opinião através de formas, como estrelas (Figura B.22).

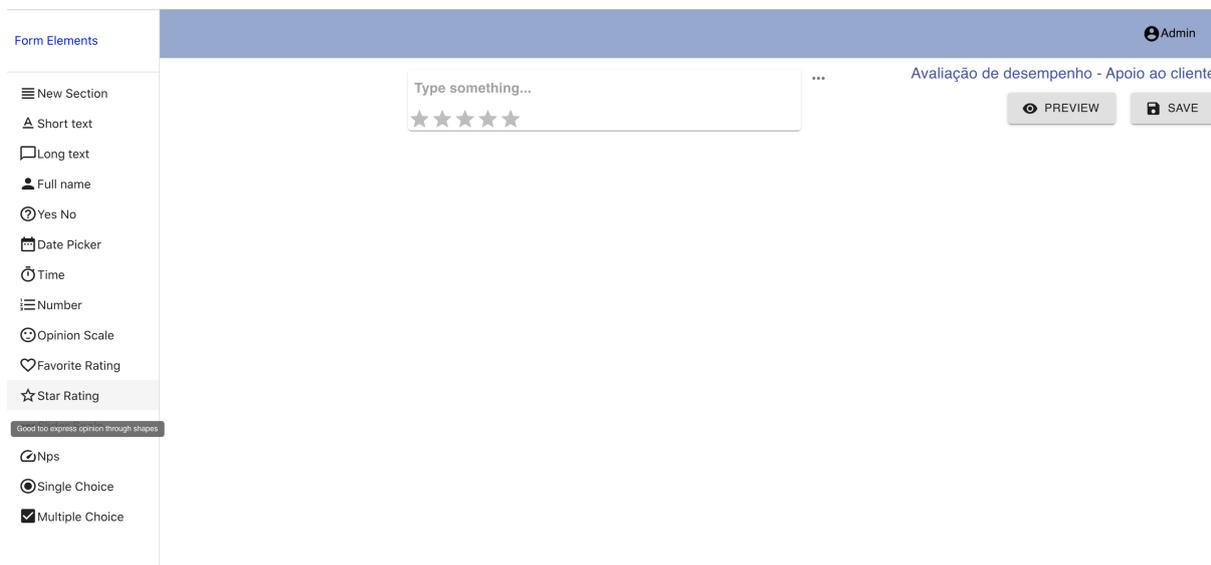


Figura B.22: Adicionar elementos - "Star Rating"

B.8.11 Slider Scale

Se seleccionar a opção "Slider Scale", é adicionado um elemento que permite ao utilizador expressar a sua opinião através de uma escala de classificação (Figura B.23).

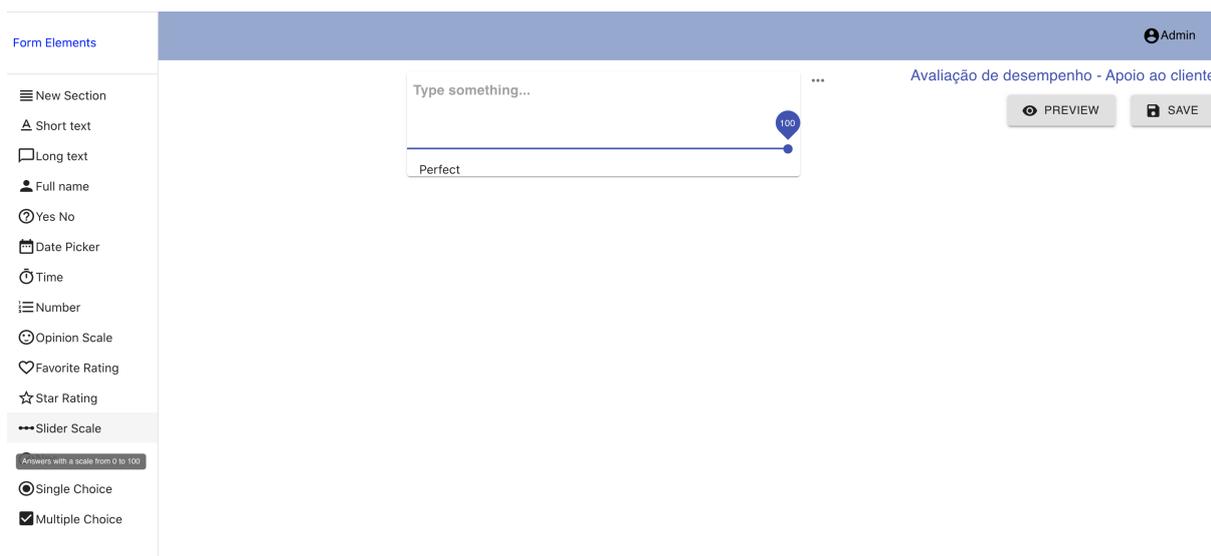


Figura B.23: Adicionar elementos - "Slider Scale"

B.8.12 NPS

Se selecionar a opção "NPS", é adicionado um elemento que permite ao utilizador expressar a sua opinião, numa escala de 0 a 10 (Figura B.24).



Figura B.24: Adicionar elementos - "NPS"

B.8.13 Single Choice

Se selecionar a opção "Single Choice", é adicionado um elemento que permite ao utilizador selecionar uma única opção de entre uma lista existente (Figura B.25).

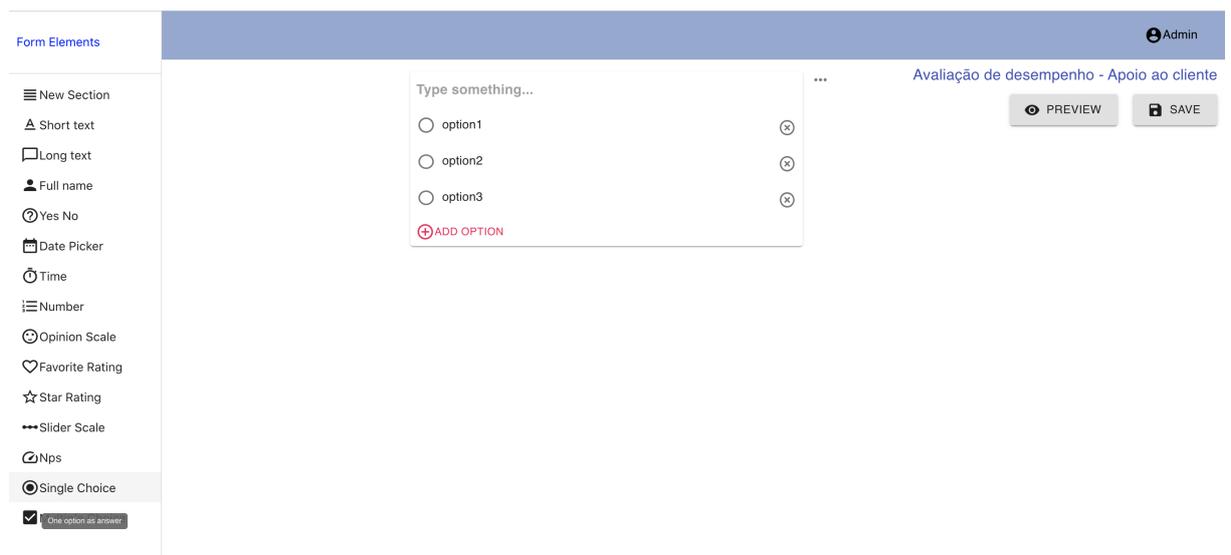


Figura B.25: Adicionar elementos - "Single Choice"

Além de poder editar as opções adicionadas, o utilizador pode também adicionar novas opções, clicando na opção "Add Option" (Figura B.26), ou eliminar as opções, clicando para isso na opção "Delete option" (Figura B.27).

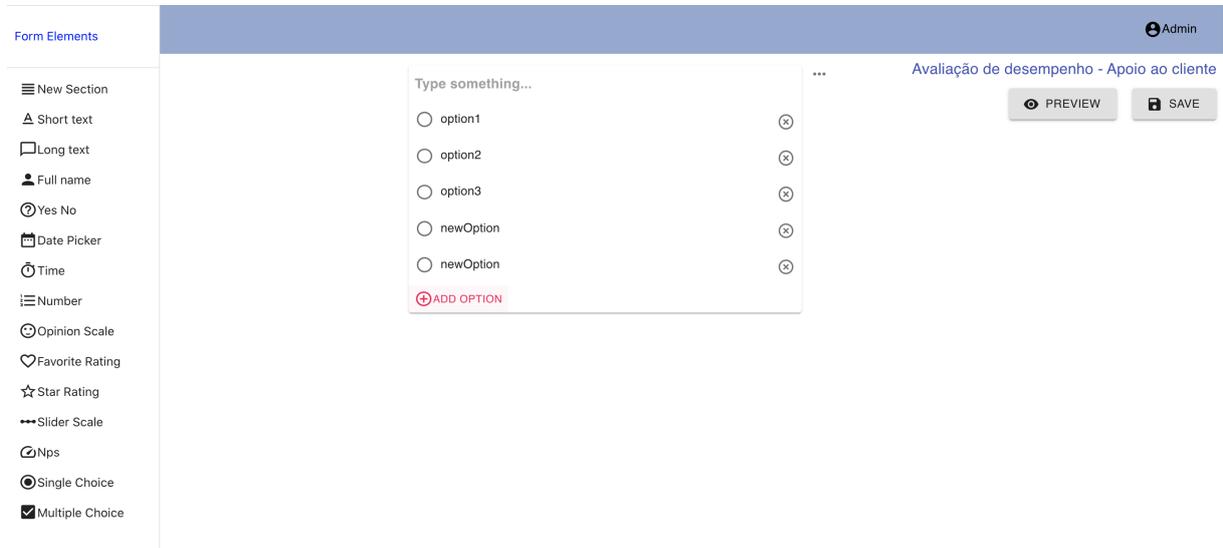


Figura B.26: Adicionar opções - "Single Choice"

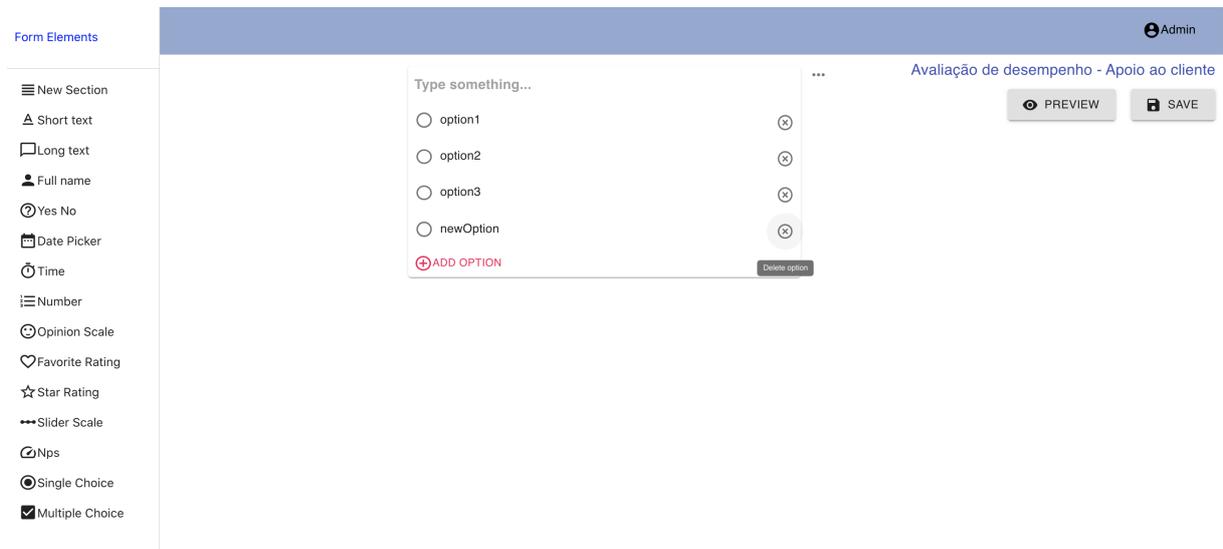


Figura B.27: Eliminar opções - "Single Choice"

B.8.14 Multiple Choice

Se selecionar a opção "Multiple Choice", é adicionado um elemento que permite selecionar várias opções de entre uma lista existente (Figura B.28).

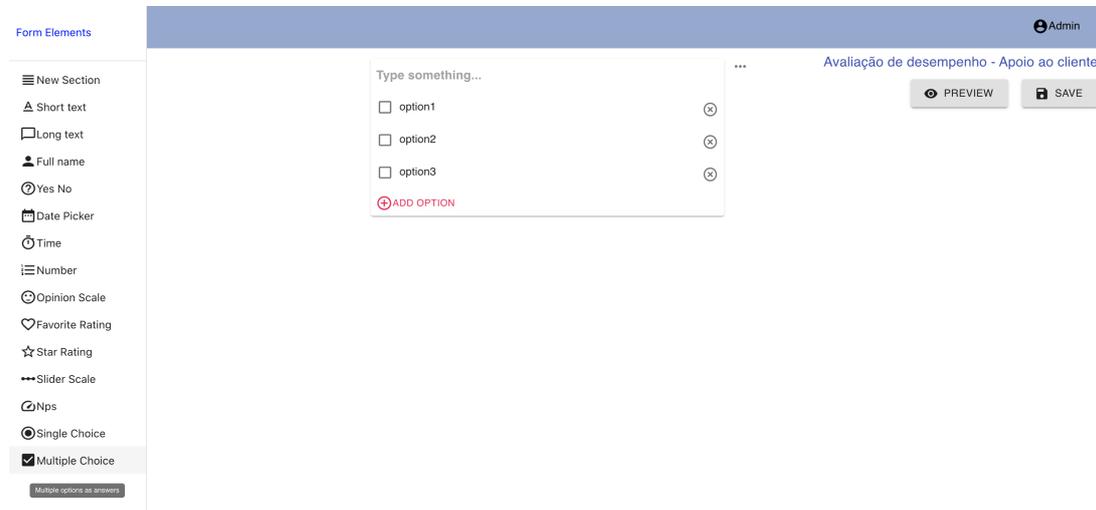


Figura B.28: Adicionar elementos - "Multiple Choice"

Além de poder editar as opções adicionadas por defeito, o administrador pode também adicionar novas opções, clicando na opção "Add Option" (Figura B.29), ou eliminar as opções que pretender, clicando para isso na opção "Delete Option" (Figura B.30).

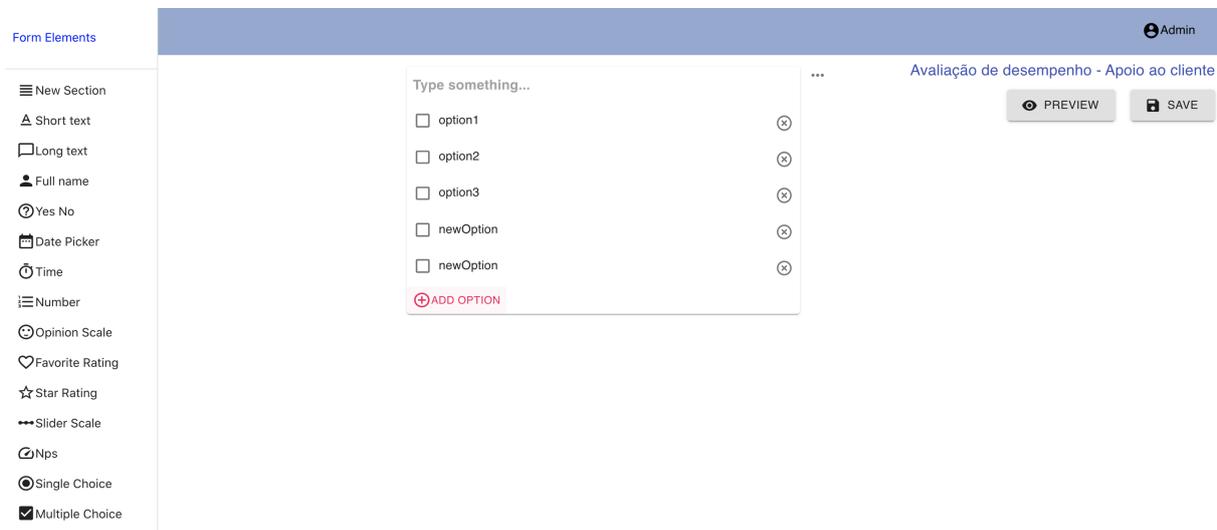


Figura B.29: Adicionar opções - "Multiple Choice"

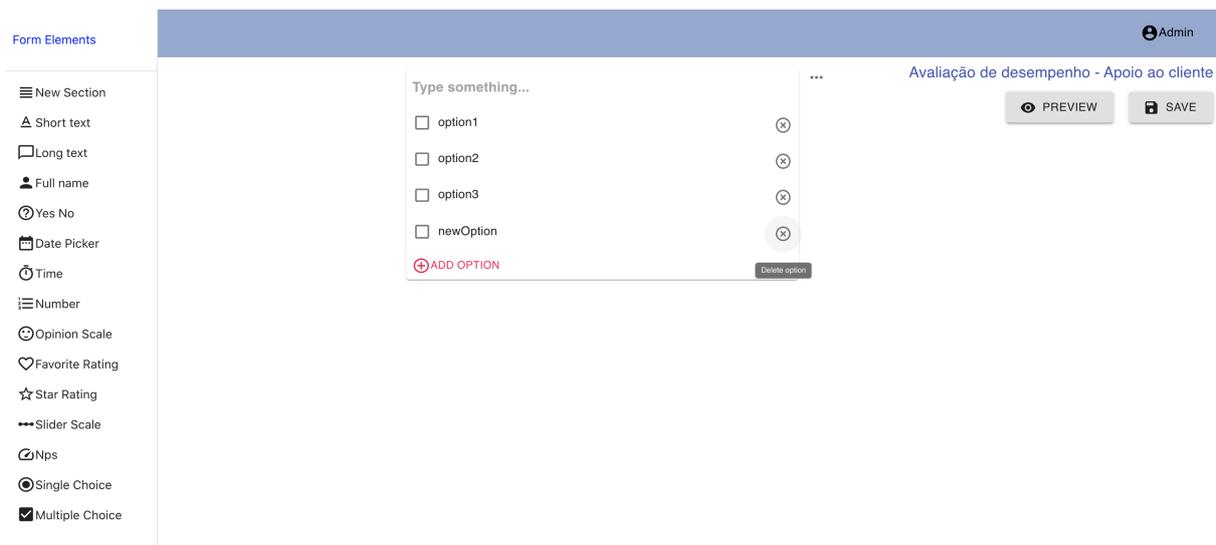


Figura B.30: Eliminar opções - "Multiple Choice"

Após serem adicionados ao formulário, os elementos podem ser personalizados com diferentes questões, tal como está representado nas figuras seguintes.

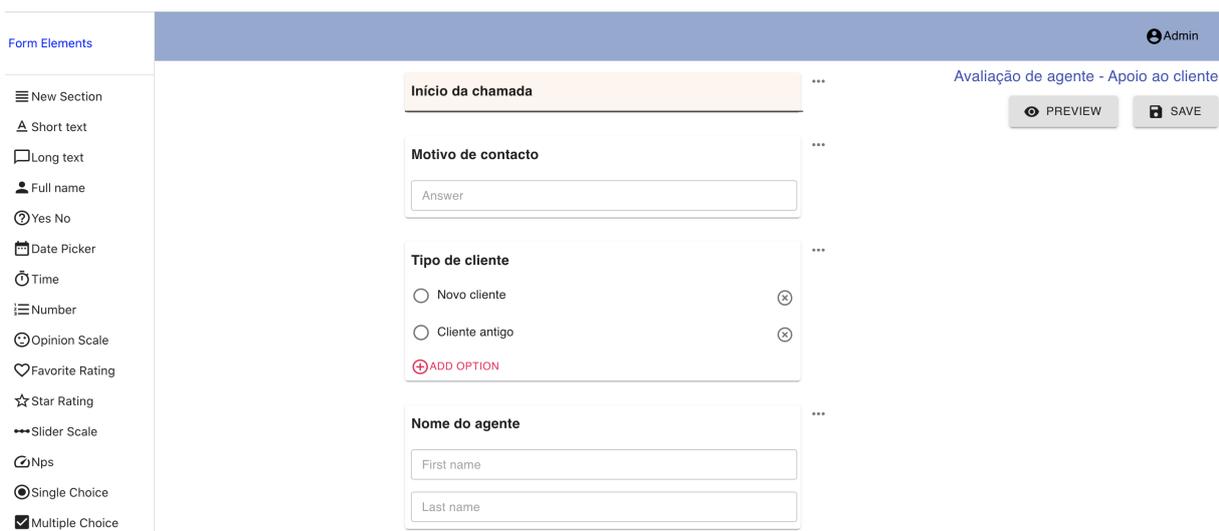


Figura B.31: Elementos *New section*, *Yes/No*, *Star Rating*, *Long Text*, *New section* e *Slider Scale*.

The screenshot shows a form builder interface with a sidebar on the left containing various form elements: New Section, Short text, Long text, Full name, Yes No, Date Picker, Time, Number, Opinion Scale, Favorite Rating, Star Rating, Slider Scale, Nps, Single Choice, and Multiple Choice. The main area displays a form titled "Avaliação de agente - Serviço de vendas" with the following elements:

- Data de contacto:** A date picker field with the placeholder "mm/dd/yyyy".
- Duração da chamada:** A time picker field with the placeholder "--:-- --".
- O agente já tinha contactado o cliente previamente?:** A radio button group with options "Yes" and "No".
- Se respondeu "Sim" anteriormente, indique o nº de vezes.:** A text input field with the placeholder "Answer".
- Classifique as capacidades de comunicação do agente.:** A favorite rating element consisting of five heart icons.

At the top right of the form area, there are "PREVIEW" and "SAVE" buttons, and an "Admin" user indicator.

Figura B.32: Elementos *Date Picker*, *Time*, *Yes/No*, *Number* e *Favorite Rating*

The screenshot shows a form builder interface with a sidebar on the left containing various form elements: Full name, Yes No, Date Picker, Time, Number, Opinion Scale, Favorite Rating, Star Rating, Slider Scale, Nps, Single Choice, and Multiple Choice. The main area displays a form with the following elements:

- Como classifica o agente?:** A list of checkboxes with options: "Concentrado", "Motivado", "Organizado", "Comunicativo", and "Nenhuma opção". There is an "ADD OPTION" button at the bottom.
- Indique o nível de satisfação com o atendimento prestado:** A smiley face rating element with five icons ranging from sad to happy.
- Atribua uma pontuação ao agente.:** A number scale element with buttons for numbers 1 through 10, and the text "Extremely likely" at the end.

Figura B.33: Elementos *Date Picker*, *Time*, *Yes/No*, *Number* e *Favorite Rating*

B.9 Eliminar elementos do formulário

Depois de adicionados ao formulário, os elementos podem ser eliminados. Para tal, o utilizador deve seleccionar a opção "More" (Figura B.34) e, em seguida, clicar no botão "Delete" (Figura B.35).

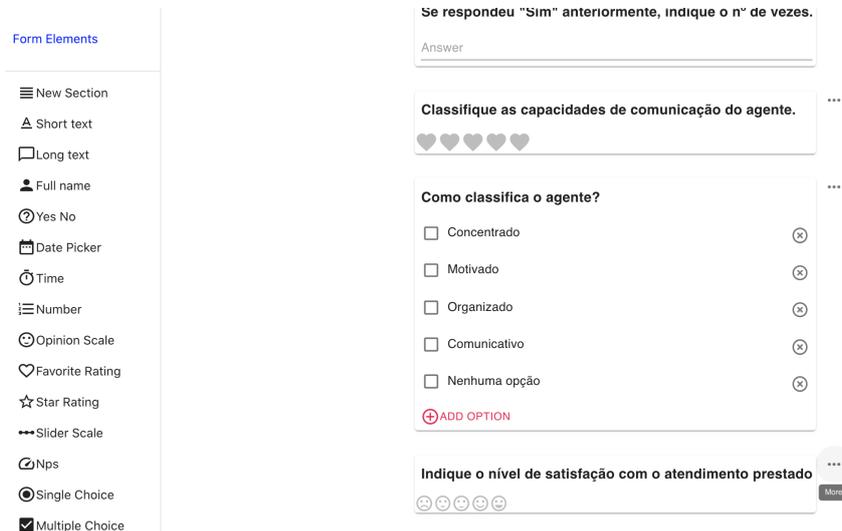


Figura B.34: Eliminar elemento do formulário

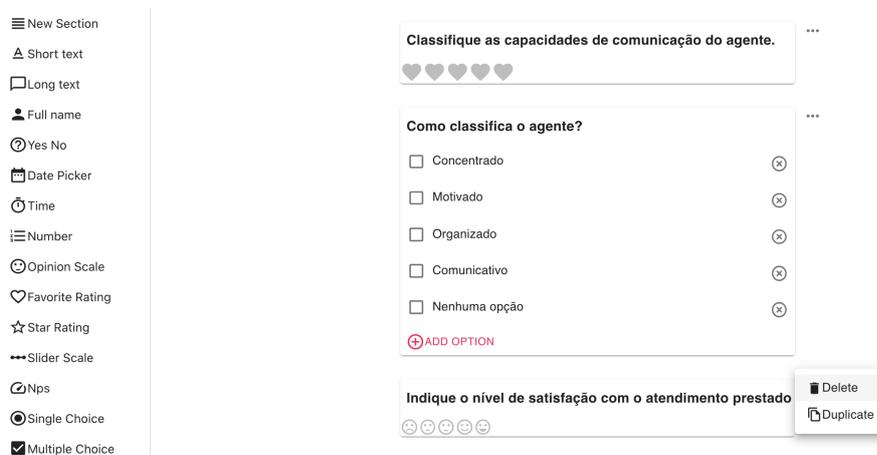


Figura B.35: Eliminar elemento do formulário

B.10 Duplicar elementos

Depois de adicionados ao formulário, os elementos podem ser duplicados. Esta funcionalidade permite que seja adicionado um novo elemento, igual ao anterior, para que o utilizador não tenha de procurá-lo novamente no menu e clicar para adicionar. Para duplicar o elemento, o utilizador deve clicar na opção "More" e, em seguida, seleccionar o botão "Duplicate", tal como se pode ver nas figuras B.36 e B.37.

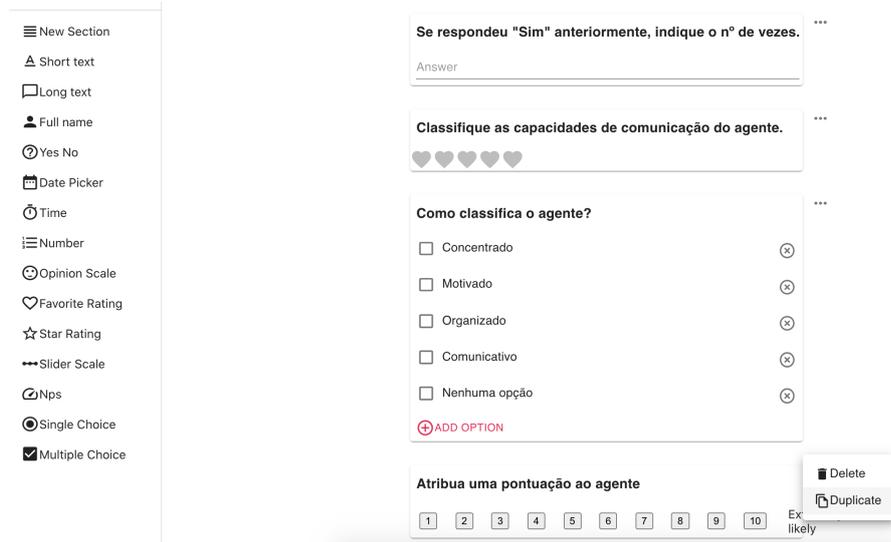


Figura B.36: Opção para duplicar elemento



Figura B.37: Elemento duplicado

B.11 Adicionar opções de resposta

Para elementos que suportem diversas opções de resposta, como por exemplo, de escolha múltipla, o utilizador pode clicar no botão "Add option" e novas opções serão adicionadas (Figuras B.38 e B.39).

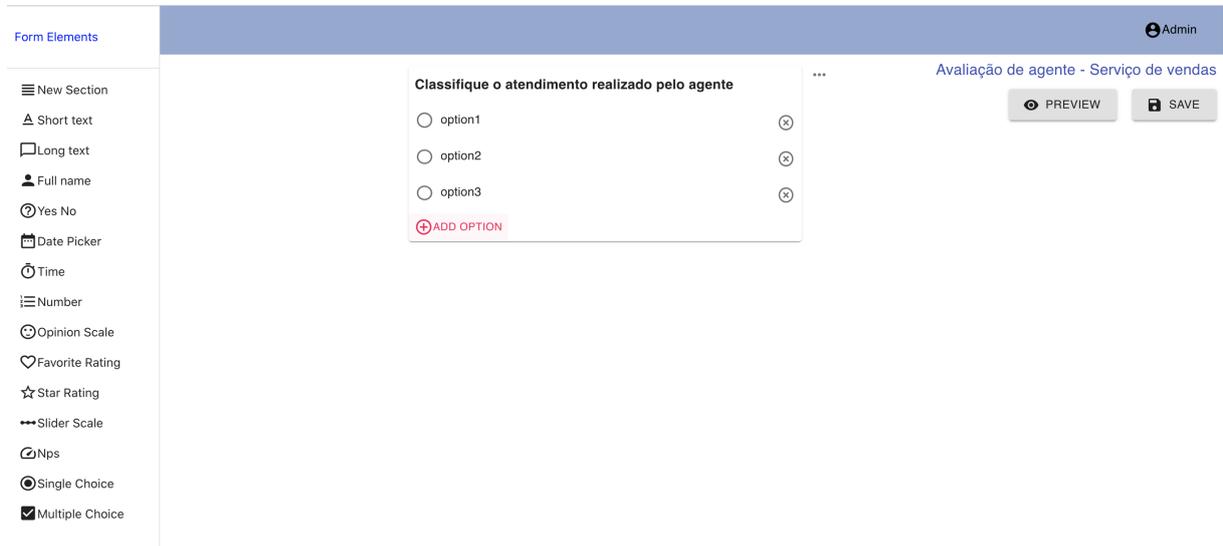


Figura B.38: Opção para adicionar opções de resposta

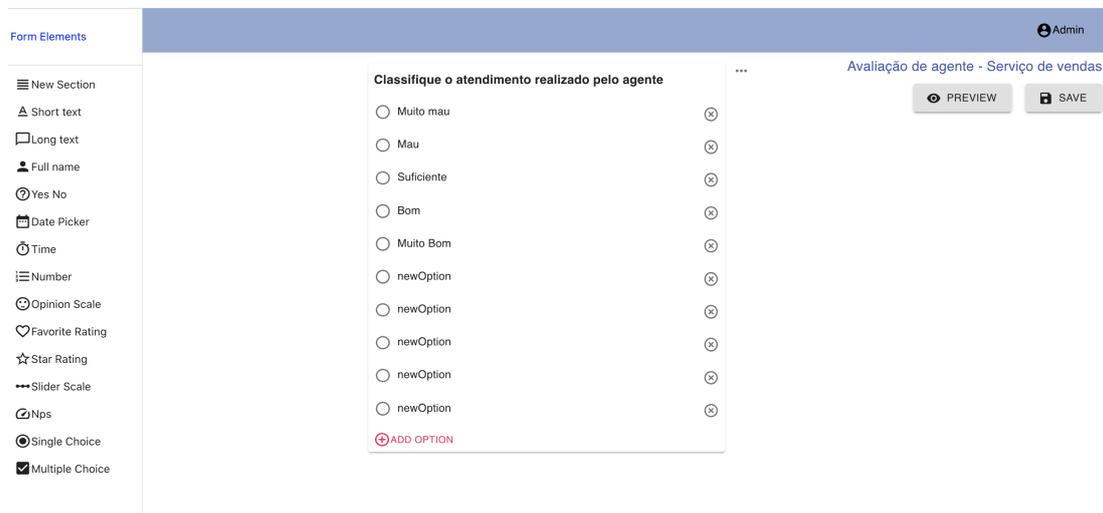


Figura B.39: Opções adicionadas

B.12 Eliminar opções de resposta

Para eliminar as opções de resposta, o utilizador deve escolher a opção que quer eliminar, clicar no botão "Delete option" e a opção será eliminada (Figuras B.40 e B.41).

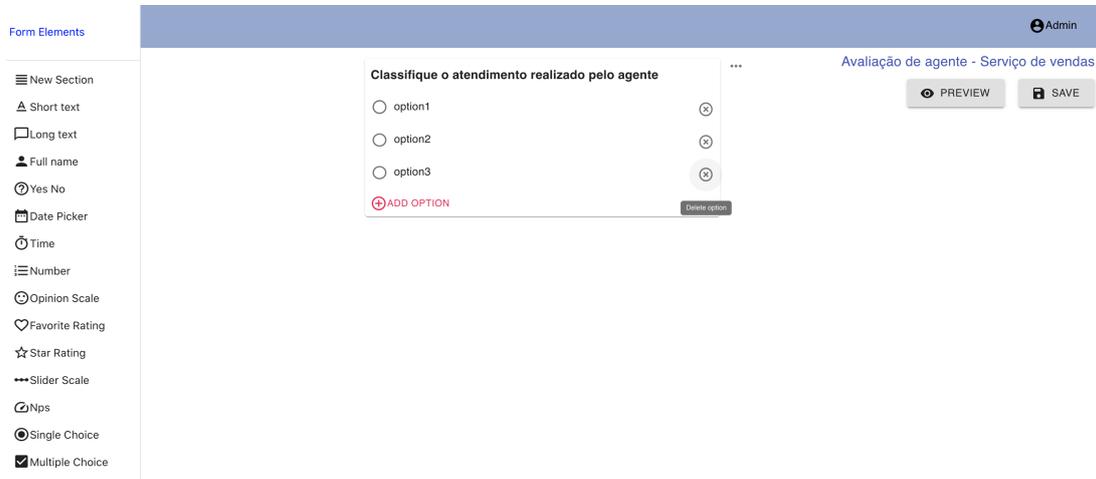


Figura B.40: Eliminar opção de resposta

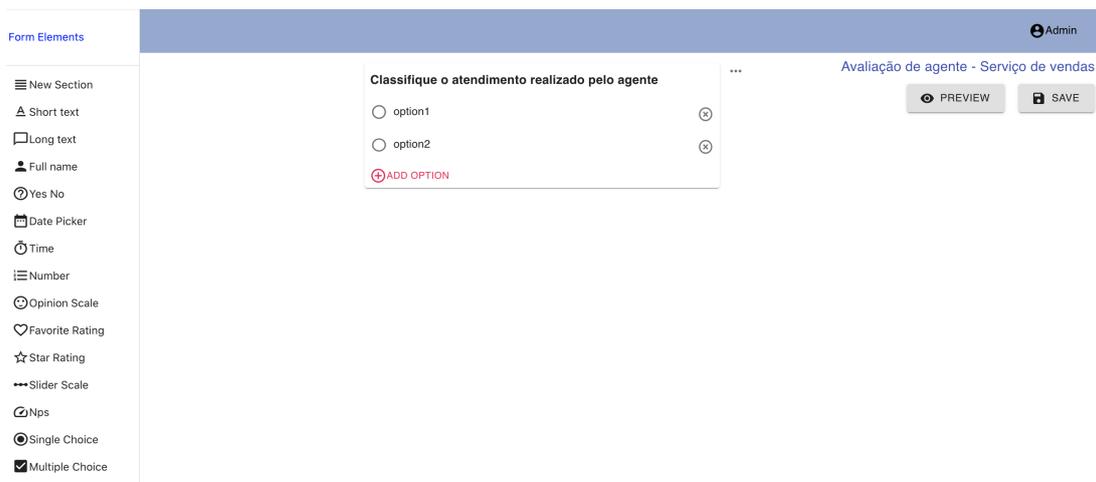


Figura B.41: Opção eliminada

B.13 Pré-visualizar formulário

O utilizador pode pré-visualizar o formulário, isto é, ver como será antes de guardá-lo. Para tal, deve clicar na opção "Preview" e todos os elementos ficarão assim em modo de leitura, isto é, não é possível editá-los, alterar os seus valores ou removê-los (Figura B.42).

Figura B.42: Pré-visualização de formulário

Para voltar ao modo de edição, o utilizador deve clicar na opção "Close", no canto superior direito.

B.14 Guardar formulário

Para guardar o formulário, o utilizador deve clicar na opção "Save" (Figura B.43).

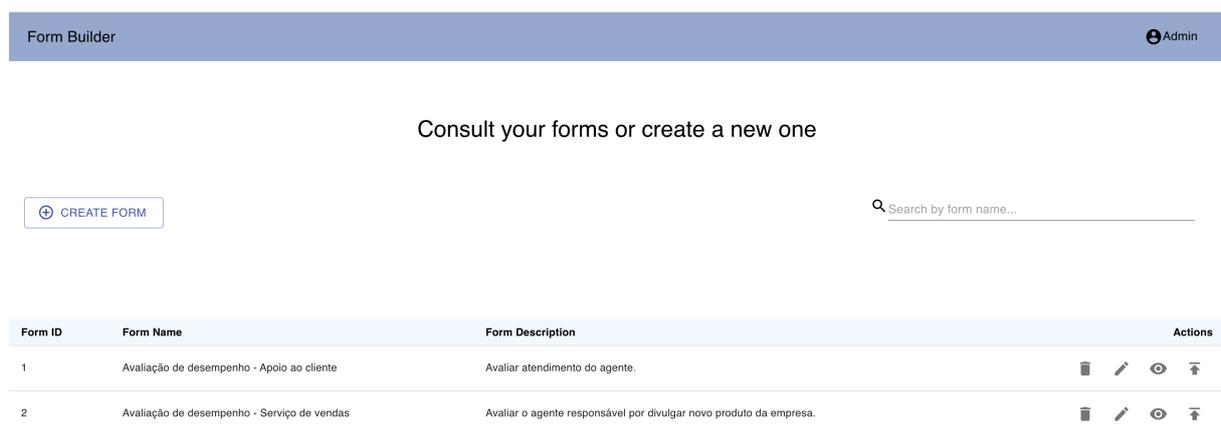
Figura B.43: Opção para guardar formulário

Quando esta opção é selecionada, o sistema inicia o processo para guardar o formulário (Figura B.44).



Figura B.44: Formulário a ser guardado

Após ser guardado, o formulário é apresentado na página inicial do administrador (Figura B.45).



The screenshot shows the 'Form Builder' interface. At the top, there is a header with 'Form Builder' on the left and 'Admin' on the right. Below the header, the text 'Consult your forms or create a new one' is centered. On the left, there is a 'CREATE FORM' button with a plus icon. On the right, there is a search bar with the placeholder text 'Search by form name...'. Below these elements is a table with the following data:

Form ID	Form Name	Form Description	Actions
1	Avaliação de desempenho - Apoio ao cliente	Avaliar atendimento do agente.	[Trash] [Edit] [Eye] [Filter]
2	Avaliação de desempenho - Serviço de vendas	Avaliar o agente responsável por divulgar novo produto da empresa.	[Trash] [Edit] [Eye] [Filter]

Figura B.45: Lista de formulários guardados no sistema

A partir desta página, o utilizador pode consultar os formulários criados ou criar um novo. Se optar por criar um novo, deve seleccionar a opção "Create Form" e repete todo o processo descrito anteriormente.

B.15 Filtrar formulários

Para pesquisar por determinado formulário, o utilizador deve escrever o nome do formulário na barra de pesquisa presente na página. O sistema irá mostrar todos os formulários que correspondem ao termo que o utilizador pesquisou (Figura B.46).

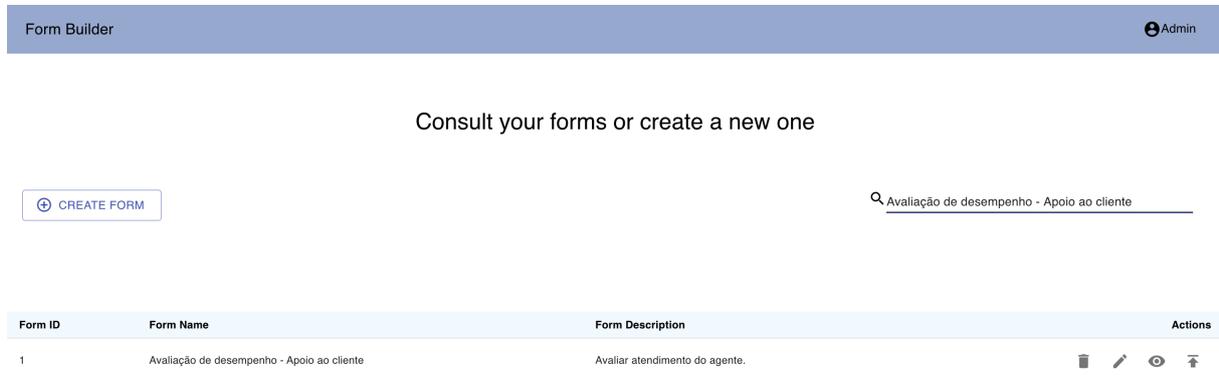


Figura B.46: Filtrar formulários

B.16 Eliminar formulários

Para eliminar determinado formulário, o utilizador deve seleccionar o que pretende e clicar na opção "Delete" (Figura B.47).

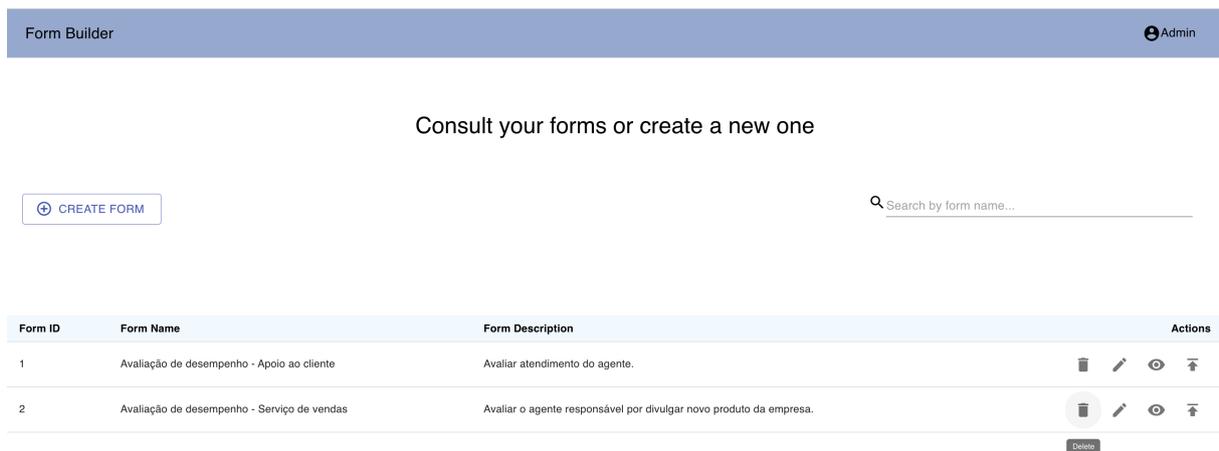


Figura B.47: Eliminar formulário

Quando clica nessa opção, o sistema mostra um alerta no ecrã, para que o utilizador confirme que deseja mesmo eliminar o formulário (Figura B.48).

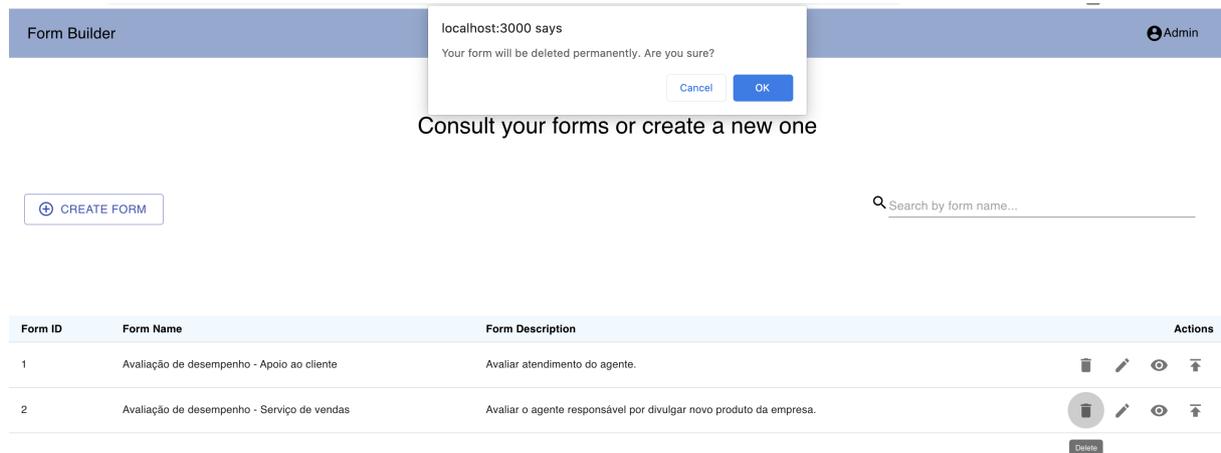


Figura B.48: Eliminar formulário

Assim, o utilizador deve seleccionar a opção "Ok" se quiser confirmar ou então clicar em "Cancel" para não eliminar o formulário.

B.17 Editar informação do formulário

Para alterar o nome e/ou descrição de um formulário, o utilizador deve seleccionar a opção "Rename" (Figura B.49).

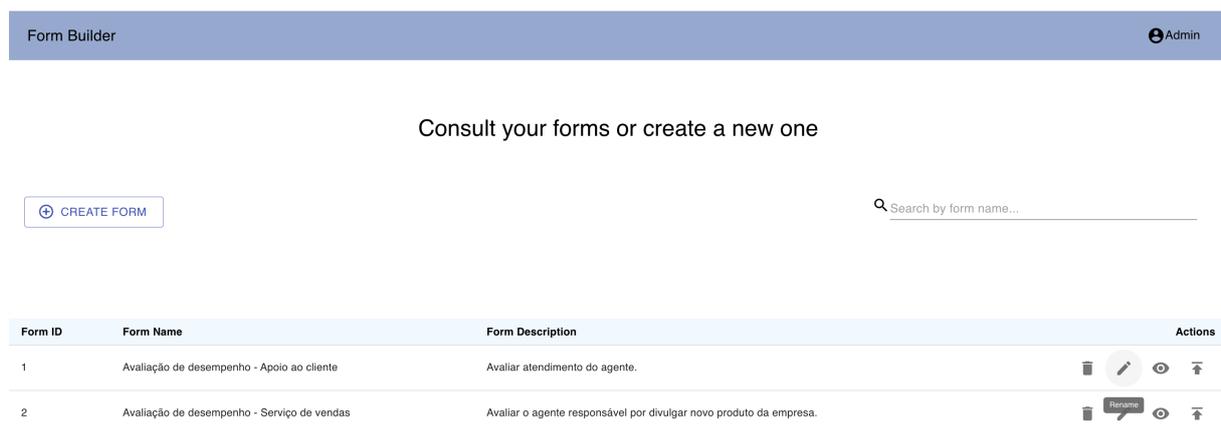
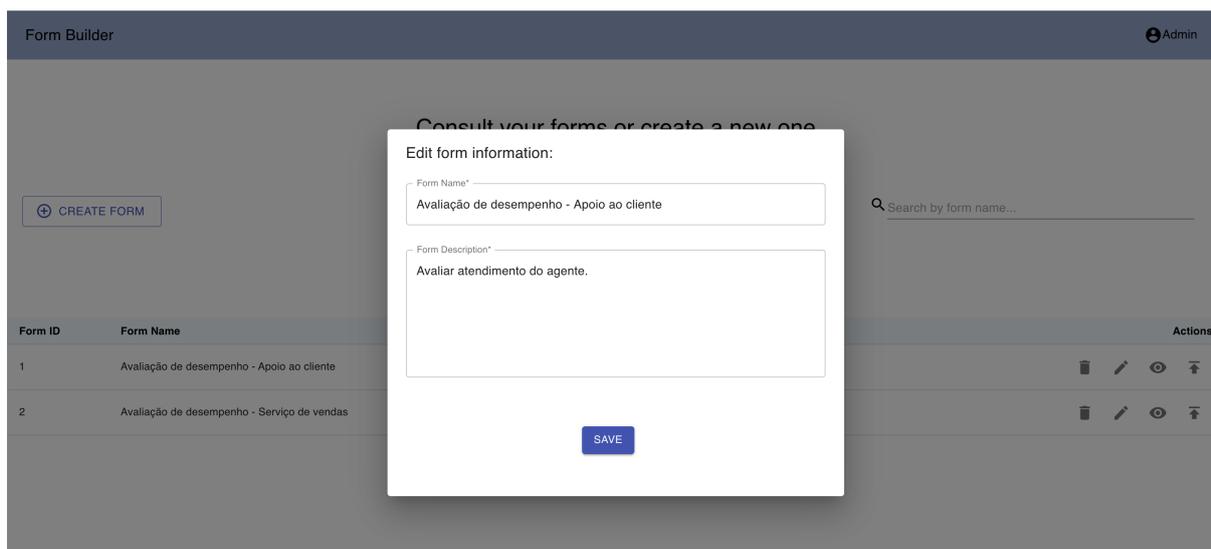


Figura B.49: Editar informação do formulário

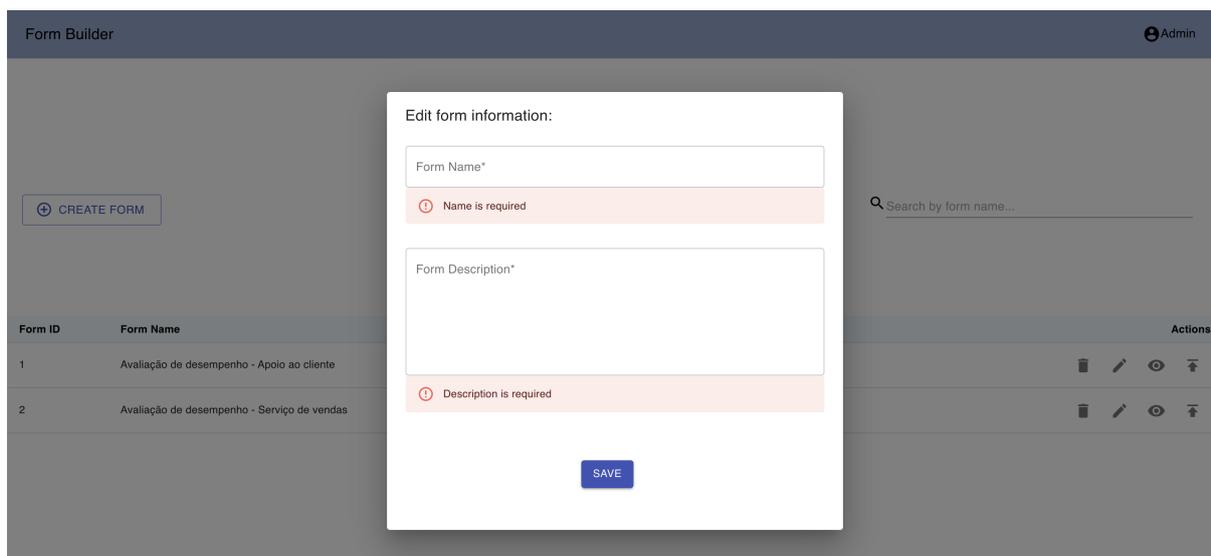
Quando clica nessa opção, pode ver a informação do respetivo formulário (Figura B.50).



The screenshot shows the 'Form Builder' interface. A modal dialog titled 'Edit form information:' is open. It contains two text input fields: 'Form Name*' with the value 'Avaliação de desempenho - Apoio ao cliente' and 'Form Description*' with the value 'Avaliar atendimento do agente.'. A blue 'SAVE' button is at the bottom. In the background, a table lists forms with columns 'Form ID' and 'Form Name'. The first row has ID '1' and name 'Avaliação de desempenho - Apoio ao cliente'. The second row has ID '2' and name 'Avaliação de desempenho - Serviço de vendas'. There is also a 'CREATE FORM' button and an 'Actions' column with icons for delete, edit, view, and refresh.

Figura B.50: Editar informação do formulário

Também aqui, o utilizador deve preencher obrigatoriamente os campos do nome e descrição (Figura B.51).



The screenshot shows the 'Form Builder' interface with the 'Edit form information:' dialog box. The 'Form Name*' field is empty and has a red error message 'Name is required'. The 'Form Description*' field is also empty and has a red error message 'Description is required'. A blue 'SAVE' button is at the bottom. The background interface is the same as in Figure B.50, showing the table of forms and the 'CREATE FORM' button.

Figura B.51: Editar informação do formulário

Se o utilizador pretender alterar, deve adicionar o novo nome e a nova descrição e clicar no botão "Save" (Figura B.52).

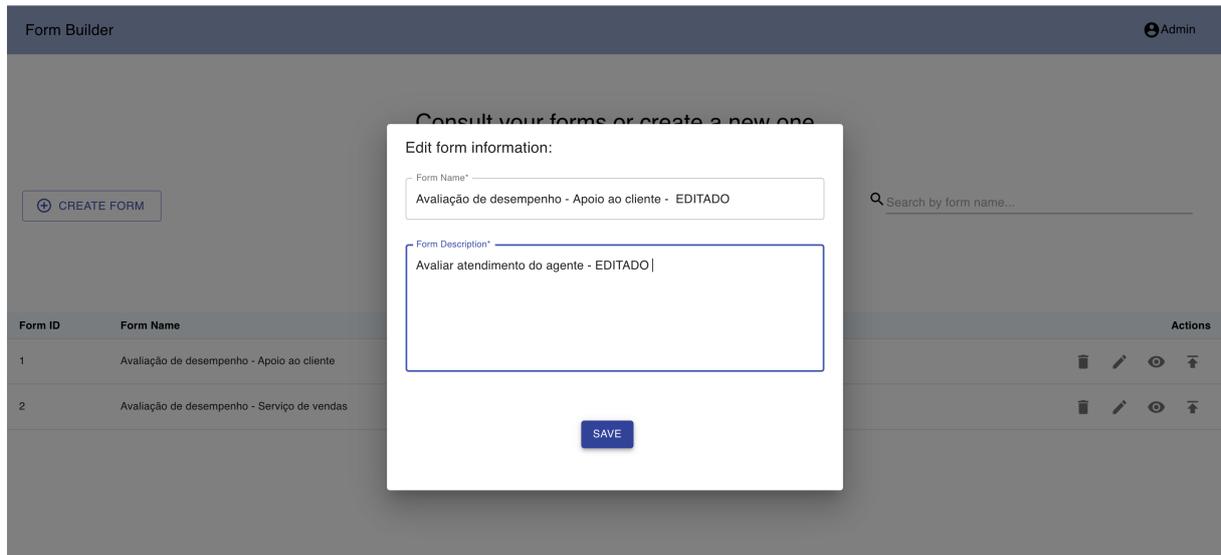


Figura B.52: Editar informação do formulário

B.18 Consultar formulários

Para ver um determinado formulário, o utilizador deve seleccionar a opção "Open" (Figura B.53).

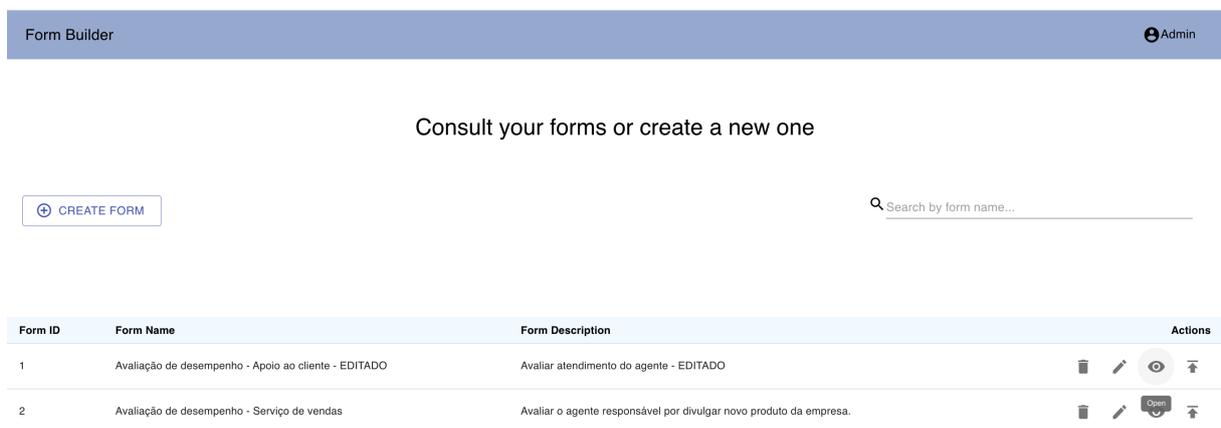


Figura B.53: Consultar formulário

Em seguida, poderá ver o formulário exatamente como o criou (Figura B.54).

Figura B.54: Formulário

B.19 Publicar formulário

O utilizador pode publicar o formulário, isto é, torná-lo visível para o supervisor. Para tal, deve escolher o formulário que pretende publicar e selecionar a opção "Publish", na sua página inicial (Figura B.55).

Form ID	Form Name	Form Description	Actions
1	Avaliação de agente - Serviço de Vendas EDITADO	Avaliar agente responsável por divulgar novo produto da empresa. EDITADO	[Icons: Delete, Edit, View, Publish]
2	Avaliação de agente	Apoio ao cliente	[Icons: Delete, Edit, View, Publish]
3	Avaliação do atendimento	Durante a divulgação de um novo produto.	[Icons: Delete, Edit, View, Publish]

Figura B.55: Publicar formulário

Depois de publicado, o formulário é automaticamente eliminado do sistema, para que o utilizador não o consiga alterar (Figura B.56).

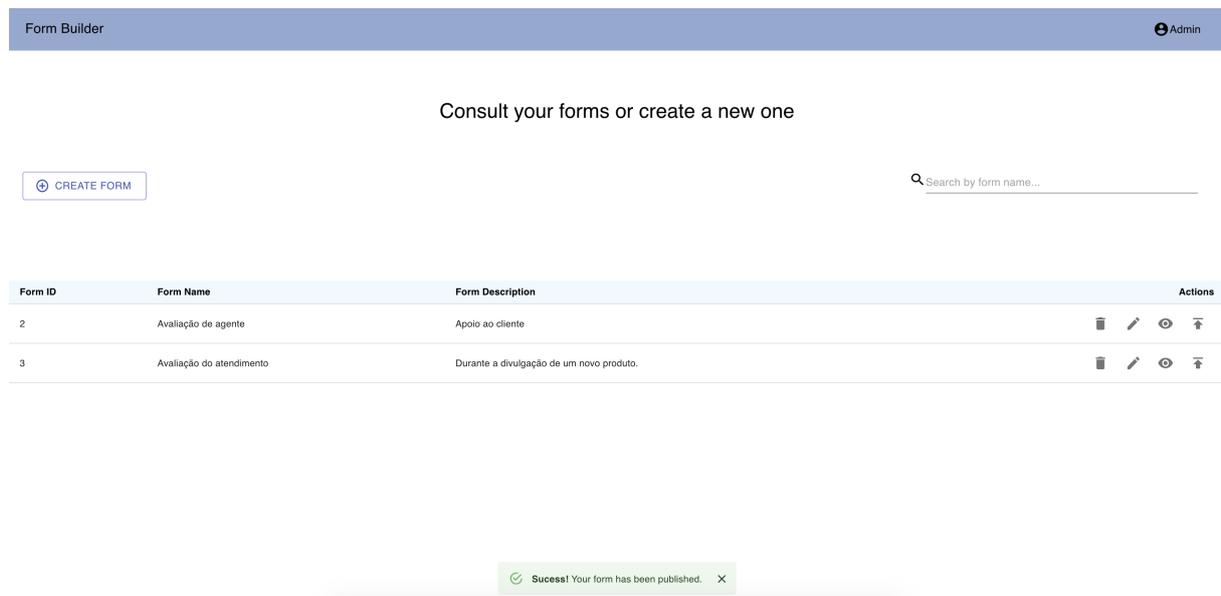


Figura B.56: Formulário formulário

B.20 Consultar formulários publicados

Quando o supervisor acede à aplicação, com os seus dados de acesso, pode ver a lista de formulários publicados (Figura 57).



Figura B.57: Lista de formulários publicados

B.21 Preencher formulário

Para preencher determinado formulário, o utilizador tem de seleccionar a opção "Preencher" na página inicial. Em seguida, o formulário é apresentado tal como foi criado e pode ser preenchido. (Figura B.58).

The image shows a survey form with the following sections:

- Como classifica o agente?**
 - Concentrado
 - Motivado
 - Organizado
 - Nenhuma opção
 - [ADD OPTION](#)
- Indique o nível de satisfação com o atendimento prestado**
 - ☹️☹️☹️☹️😊 Satisfied
- Atribua uma pontuação ao agente.**
 - 1 2 3 4 5 6 7 8 9 10 Extremely likely

Figura B.58: Formulário preenchido

Anexo C

Plano de testes

Neste documento são apresentados os testes realizados: Testes unitários e Testes de sistema.

C.1 Testes unitários

Caso de teste 01	Iniciar a aplicação
Descrição	Verificar se ao iniciar a aplicação o utilizador é redirecionado para a página de login.
Resultado	PASSOU

Caso de teste 02	Elementos na página de login
Descrição	Verificar se a página de login contém elementos que permitam inserir o <i>username</i> e a <i>password</i> e um botão para fazer login.
Resultado	PASSOU

Caso de teste 03	Autenticação na aplicação
Descrição	Verificar se é possível iniciar um pedido de autenticação, após inserir as credenciais de acesso e clicar na opção de login.
Resultado	PASSOU

Caso de teste 04	Alerta - Página de login
Descrição	Verificar se existe uma mensagem de alerta na página de login, quando o utilizador insere dados incorretos.
Resultado	PASSOU

Caso de teste 05	Página inicial - Administrador
Descrição	Verificar se a <i>search bar</i> que está na página inicial do administrador contém os elementos esperados.
Resultado	PASSOU

Caso de teste 06	Página inicial do administrador
Descrição	Verificar se existe uma mensagem de alerta na página inicial quando não existem quaisquer formulários no sistema.
Resultado	PASSOU

Caso de teste 07	Página inicial do administrador
Descrição	Verificar se existe uma opção para criar novo formulário quando ainda não existe qualquer formulário no sistema.
Resultado	PASSOU

Caso de teste 08	Página inicial do administrador
Descrição	Verificar se, quando o administrador se autentica na aplicação, é iniciado um pedido ao servidor para apresentar os formulários existentes.
Resultado	PASSOU

Caso de teste 09	Página inicial do administrador
Descrição	Verificar se existe um elemento (<i>loading</i>) que mostra ao utilizador que o pedido está a ser efetuado e os dados ainda não estão disponíveis.
Resultado	PASSOU

Caso de teste 010	Página inicial do administrador
Descrição	Verificar se, quando existem formulários criados previamente e guardados na base de dados, estes são apresentados ao utilizador na sua página inicial.
Resultado	PASSOU

Caso de teste 011	Página inicial do administrador
Descrição	Verificar se quando existem formulários, a página inicial contém um título, uma opção para adicionar novo formulário e uma <i>search bar</i> .
Resultado	PASSOU

Caso de teste 012	Página inicial do administrador
Descrição	Verificar se existe uma opção para abrir o formulário depois de guardado.
Resultado	PASSOU

Caso de teste 013	Página inicial do administrador
Descrição	Verificar se a página contém uma opção para editar o nome e/ou descrição do formulário.
Resultado	PASSOU

Caso de teste 014	Página inicial do administrador
Descrição	Verificar se ao clicar no botão para editar formulário, é apresentado um <i>pop-up</i> no ecrã.
Resultado	PASSOU

Caso de teste 015	Página inicial do administrador
Descrição	Verificar se são apresentados os elementos esperados para poder editar a informação do formulário.
Resultado	PASSOU

Caso de teste 016	Página inicial do administrador
Descrição	Verificar que não é possível alterar o nome e/ou descrição para campos em branco.
Resultado	PASSOU

Caso de teste 017	Página inicial do administrador
Descrição	Verificar se a página inicial contém uma opção para apagar formulário.
Resultado	PASSOU

Caso de teste 018	Página inicial do administrador
Descrição	Verificar se, quando o utilizador clica na opção para apagar formulário, é apresentado um alerta no ecrã.
Resultado	PASSOU

Caso de teste 019	Página para criar formulário
Descrição	Verificar se existe uma página para criar novo formulário e os elementos contidos na página são os esperados.
Resultado	PASSOU

Caso de teste 020	Página para criar formulário
Descrição	Verificar se existe uma opção para sair da página de criação de formulários e voltar à inicial.
Resultado	PASSOU

Caso de teste 021	Página para criar formulário
Descrição	Verificar se existe uma mensagem de erro na página quando o utilizador tenta criar um formulário com um nome inferior a 2 caracteres.
Resultado	PASSOU

Caso de teste 022	Página para criar formulário
Descrição	Verificar se o utilizador consegue inserir um nome e uma descrição e clicar em "Guardar".
Resultado	PASSOU

Caso de teste 023	Página para criar formulário
Descrição	Verificar se existe uma mensagem de erro na página quando o utilizador tenta criar um formulário sem qualquer descrição.
Resultado	PASSOU

Caso de teste 024	Página do formulário
Descrição	Verificar se existe um menu com diferentes elementos.
Resultado	PASSOU

Caso de teste 025	Página do formulário
Descrição	Verificar se ao clicar em determinado elemento do menu, este é adicionado à página.
Resultado	PASSOU

Caso de teste 026	Página do formulário
Descrição	Verificar se existe um botão para ver outras opções em cada elemento adicionado à página, nomeadamente as opções que permitem apagar e duplicar o elemento.
Resultado	PASSOU

Caso de teste 027	Página do formulário
Descrição	Verificar se existe uma opção que permite eliminar o elemento do formulário.
Resultado	PASSOU

Caso de teste 028	Página do formulário
Descrição	Verificar se o elemento é apagado permanentemente quando o utilizador clica na opção "Delete".
Resultado	PASSOU

Caso de teste 029	Página do formulário
Descrição	Verificar se existe uma opção para duplicar o elemento.
Resultado	PASSOU

Caso de teste 030	Página do formulário
Descrição	Verificar se, ao clicar para duplicar, o elemento anterior mantém-se na página.
Resultado	PASSOU

Caso de teste 031	Página do formulário
Descrição	Verificar se, quando o utilizador clica para pré-visualizar o formulário, os restantes elementos ficam em modo de leitura, isto é, não podem ser alterados nem removidos.
Resultado	PASSOU

Caso de teste 032	Página do formulário
Descrição	Verificar se existe uma opção para sair do modo de pré-visualização.
Resultado	PASSOU

Caso de teste 033	Página do formulário
Descrição	Verificar se ao clicar na opção para sair do modo de pré-visualização, é possível voltar ao modo de edição.
Resultado	PASSOU

Caso de teste 034	Página do formulário
Descrição	Verificar se a página de pré-visualização contém um título.
Resultado	PASSOU

Caso de teste 035	Elemento "New Section" do formulário
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 036	Elemento "New Section" do formulário
Descrição	Verificar se é possível adicionar texto nos campos da questão e da resposta.
Resultado	PASSOU

Caso de teste 037	Elemento "Short text"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 038	Elemento "Short Text"
Descrição	Verificar se é possível adicionar texto nos campos da questão e da resposta.
Resultado	PASSOU

Caso de teste 039	Elemento "Long text"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 040	Elemento "Long text"
Descrição	Verificar se é possível adicionar texto nos campos da questão e da resposta.
Resultado	PASSOU

Caso de teste 041	Elemento "Full Name"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 042	Elemento "Full Name"
Descrição	Verificar se é possível adicionar texto nos campos da questão e da resposta.
Resultado	PASSOU

Caso de teste 043	Elemento "Yes/No"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 044	Elemento "Yes/No"
Descrição	Verificar se as opções de resposta apresentadas no elemento são as esperadas.
Resultado	PASSOU

Caso de teste 045	Elemento "Yes/No"
Descrição	Verificar se é possível selecionar os <i>radio buttons</i> que estão no elemento.
Resultado	PASSOU

Caso de teste 046	Elemento "Yes/No"
Descrição	Verificar se é possível alterar o texto da questão.
Resultado	PASSOU

Caso de teste 047	Elemento "Date Picker"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 048	Elemento "Date Picker"
Descrição	Verificar se o utilizador pode alterar o texto da questão e inserir datas como resposta.
Resultado	PASSOU

Caso de teste 049	Elemento "Time"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 050	Elemento "Time"
Descrição	Verificar se é possível alterar a questão e a resposta.
Resultado	PASSOU

Caso de teste 051	Elemento "Number"
Descrição	Verificar se elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 052	Elemento "Number"
Descrição	Verificar se o utilizador pode alterar os valores apresentados no elemento.
Resultado	PASSOU

Caso de teste 053	Elemento "Number"
Descrição	Verificar se o utilizador não pode inserir outros caracteres, além de números, no campo da resposta.
Resultado	PASSOU

Caso de teste 054	Elemento "Opinion Scale"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 055	Elemento "Opinion Scale"
Descrição	Verificar se o utilizador pode alterar os valores na resposta.
Resultado	PASSOU

Caso de teste 056	Elemento "Opinion Scale"
Descrição	Verificar se o utilizador pode alterar a questão.
Resultado	PASSOU

Caso de teste 057	Elemento "Favorite Rating"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 058	Elemento "Favorite Rating"
Descrição	Verificar se o utilizador pode alterar os valores apresentados na resposta.
Resultado	PASSOU

Caso de teste 059	Elemento "Star Rating"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 060	Elemento "Star Rating"
Descrição	Verificar se o utilizador pode alterar os valores apresentados na resposta.
Resultado	PASSOU

Caso de teste 061	Elemento "Slider Scale"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 062	Elemento "Slider Scale"
Descrição	Verificar se o utilizador pode alterar os valores das respostas.
Resultado	PASSOU

Caso de teste 063	Elemento "Slider Scale"
Descrição	Verificar se o utilizador pode alterar os valores da questão.
Resultado	PASSOU

Caso de teste 064	Elemento "NPS"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 065	Elemento "NPS"
Descrição	Verificar se o utilizador pode alterar os valores apresentados nas respostas.
Resultado	PASSOU

Caso de teste 066	Elemento "NPS"
Descrição	Verificar se o utilizador pode alterar a questão.
Resultado	PASSOU

Caso de teste 067	Elemento "Single Choice"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 068	Elemento "Single Choice"
Descrição	Verificar se o elemento pode ser apagado quando o utilizador clica para eliminar.
Resultado	PASSOU

Caso de teste 069	Elemento "Single Choice"
Descrição	Verificar se uma nova opção é adicionada quando o utilizador clica para adicionar.
Resultado	PASSOU

Caso de teste 070	Elemento "Single Choice"
Descrição	Verificar se é possível alterar os valores das opções.
Resultado	PASSOU

Caso de teste 071	Elemento "Single Choice"
Descrição	Verificar se é possível alterar o texto apresentado na questão.
Resultado	PASSOU

Caso de teste 072	Elemento "Multiple Choice"
Descrição	Verificar se o elemento contém os campos esperados.
Resultado	PASSOU

Caso de teste 073	Elemento "Multiple Choice"
Descrição	Verificar se as opções podem ser apagadas quando o utilizador clica para apagar.
Resultado	PASSOU

Caso de teste 074	Elemento "Multiple Choice"
Descrição	Verificar se uma nova opção é adicionada quando o utilizador clica em adicionar.
Resultado	PASSOU

Caso de teste 075	Elemento "Multiple Choice"
Descrição	Verificar se é possível alterar o texto da questão.
Resultado	PASSOU

Caso de teste 076	Página do formulário
Descrição	Verificar se, quando o utilizador clica na opção para guardar o formulário, é iniciado um pedido ao servidor.
Resultado	PASSOU

Caso de teste 077	Página inicial do administrador
Descrição	Verificar se os dados do formulário são apresentados corretamente na página, quando o utilizador (administrador) clica na opção para abrir formulário.
Resultado	PASSOU

Caso de teste 078	Página inicial do Supervisor
Descrição	Verificar se os dados do formulário publicado são mostrados corretamente quando o utilizador (supervisor) clica na opção para abrir formulário.
Resultado	PASSOU

Caso de teste 079	Página inicial do Supervisor
Descrição	Verificar se existe um botão para submeter na página do supervisor.
Resultado	PASSOU

Caso de teste 080	Página inicial do Supervisor
Descrição	Verificar se a <i>search bar</i> presente na página contém os elementos esperados.
Resultado	PASSOU

Caso de teste 081	Página inicial do Supervisor
Descrição	Verificar se é apresentada uma mensagem na página quando não há quaisquer formulários publicados.
Resultado	PASSOU

Caso de teste 082	Página inicial do Supervisor
Descrição	Verificar se os formulários publicados pelo administrador são apresentados corretamente na página do supervisor.
Resultado	PASSOU

Caso de teste 083	Página inicial do Supervisor
Descrição	Verificar se a página do supervisor contém os elementos esperados.
Resultado	PASSOU

C.2 Testes de Sistema

Caso de teste 001	Dados de acesso incorretos
Pré-condições	Aceder à página de login
Procedimento	1 - O utilizador insere dados incorretos; 2 - O utilizador seleciona a opção "Login"; 3 - O sistema verifica se os dados do utilizador existem no sistema.
Dados de entrada	Dados de acesso incorretos.
Resultado esperado	Mensagem de erro no sistema.

Caso de teste 002	Dados de acesso corretos
Pré-condições	Aceder à página de login.
Procedimento	1 - O utilizador insere os dados de acesso como administrador corretamente; 2 - O utilizador seleciona a opção "Login"; 3 - O sistema verifica se os dados do utilizador existem no sistema.
Dados de entrada	Dados de acesso corretos.
Resultado esperado	O utilizador consegue aceder à aplicação e é redirecionado para a página inicial.

Caso de teste 003	Formulário com dados inválidos
Pré-condições	1- Aceder à página de login; 2 - Clicar na opção para criar formulário;
Procedimento	1 - O utilizador não insere qualquer nome ou descrição nos campos do formulário. 2 - O utilizador seleciona a opção "Continue".
Dados de entrada	Nome e descrição em branco.
Resultado esperado	O sistema mostra uma mensagem de erro e o formulário não é criado.

Caso de teste 004	Formulário com dados inválidos
Pré-condições	1- Aceder à página de login; 2 - Clicar na opção para criar formulário.
Procedimento	1 - O utilizador atribui um nome ao formulário; 2 - O utilizador seleciona a opção "Continue"; 3 - O sistema verifica se já existe um outro formulário com o mesmo nome.
Dados de entrada	Nome de um formulário que já exista no sistema.
Resultado esperado	O sistema mostra uma mensagem de alerta e o formulário não é criado.

Caso de teste 005	Formulário com dados corretos
Pré-condições	1- Aceder à página de login; 2 - Clicar na opção para criar formulário.
Procedimento	1 - O utilizador atribui um nome e descrição ao formulário, disponibilizados pela aluna; 2 - O utilizador seleciona a opção "Continue".
Dados de entrada	Nome e descrição.
Resultado esperado	O formulário é criado e o utilizador é redirecionado para uma página que contém o nome introduzido anteriormente e também um menu com opções para adicionar elementos ao formulário.

Caso de teste 006	Adicionar elementos ao formulário
Pré-condições	1- Criar um formulário
Procedimento	1 - O utilizador clica nos itens existentes no menu.
Dados de entrada	-
Resultado esperado	Os itens selecionados pelo utilizador são adicionados à página.

Caso de teste 007	Adicionar opções de resposta nos elementos de escolha múltipla.
Pré-condições	1- Criar um formulário; 2 - Adicionar um elemento de escolha múltipla.
Procedimento	1 - O utilizador clica para adicionar uma ou mais opções de resposta.
Dados de entrada	-
Resultado esperado	O formulário contém as opções de resposta adicionadas pelo utilizador.

Caso de teste 008	Preencher cada elemento adicionado com questões disponibilizadas pela aluna.
Pré-condições	1- Criar um formulário; 2 - Adicionar elementos ao formulário.
Procedimento	1 - Inserir questões em cada elemento adicionado.
Dados de entrada	Questões disponibilizadas pela aluna.
Resultado esperado	O formulário contém as questões adicionadas pelo utilizador.

Caso de teste 009	Remover elemento do formulário
Pré-condições	1- Criar um formulário; 2 - Adicionar elementos ao formulário.
Procedimento	1 - Clicar na opção para remover elemento.
Dados de entrada	-
Resultado esperado	O elemento selecionado pelo utilizador é removido do formulário.

Caso de teste 010	Duplicar elemento do formulário
Pré-condições	1- Criar um formulário; 2 - Adicionar elementos ao formulário.
Procedimento	1 - Clicar na opção para duplicar elemento.
Dados de entrada	-
Resultado esperado	O formulário contém um novo elemento, igual ao anterior.

Caso de teste 011	Pré-visualizar formulário
Pré-condições	1- Criar um formulário; 2 - Adicionar elementos ao formulário.
Procedimento	1 - Clicar na opção para pré-visualizar formulário.
Dados de entrada	-
Resultado esperado	O formulário é apresentado tal como criado pelo utilizador, mas os elementos não podem ser editados nem eliminados, ficando apenas disponíveis para visualização.

Caso de teste 012	Guardar formulário
Pré-condições	1- Criar um formulário; 2 - Adicionar elementos ao formulário.
Procedimento	1 - Clicar na opção para guardar formulário.
Dados de entrada	-
Resultado esperado	O formulário é guardado no sistema e o utilizador é redirecionado para a sua página inicial, onde pode ver a lista de todos os formulários existentes.

Caso de teste 013	Filtrar formulário
Pré-condições	1- Criar um formulário; 2 - Guardar formulário.
Procedimento	1 - O utilizador insere o nome do formulário na barra inicial de pesquisa.
Dados de entrada	Nome do formulário
Resultado esperado	O sistema mostra os formulários que correspondem ao nome inserido pelo utilizador.

Caso de teste 014	Alterar nome e descrição do formulário
Pré-condições	1- Criar um formulário; 2 - Guardar formulário.
Procedimento	1 - O utilizador clica na opção para alterar o nome e/ou descrição do formulário; 2 - O utilizador apaga os dados do formulário; 3 - O utilizador clica na opção "Save".
Dados de entrada	-
Resultado esperado	O sistema mostra uma mensagem de erro e a informação não é alterada.

Caso de teste 015	Publicar formulário
Pré-condições	1- Criar um formulário; 2 - Guardar formulário.
Procedimento	1 - O utilizador clica na opção para publicar formulário.
Dados de entrada	-
Resultado esperado	O formulário é publicado e o sistema apresenta uma mensagem de sucesso.

Caso de teste 016	Efetuar logout
Pré-condições	1- Estar autenticado na aplicação;
Procedimento	1 - O utilizador clica na opção para efetuar logout.
Dados de entrada	-
Resultado esperado	A sessão é terminada e o utilizador é redirecionado para a página de login.

Caso de teste 017	Dados de acesso corretos - Supervisor
Pré-condições	1- Aceder à página de login.
Procedimento	1 - O utilizador insere corretamente os dados de acesso como supervisor.
Dados de entrada	Nome do utilizador e palavra-passe
Resultado esperado	O utilizador autentica-se na aplicação e é redirecionado para a página inicial.

Caso de teste 018	Preencher formulário
Pré-condições	1- Estar autenticado na aplicação.
Procedimento	1 - O utilizador clica na opção para preencher o formulário que pretende.
Dados de entrada	-
Resultado esperado	O utilizador acede ao formulário e inicia o seu preenchimento.

Caso de teste 019	Adicionar texto no elemento <i>Number</i>
Pré-condições	1- Criar um formulário no sistema com um elemento do tipo <i>Number</i> .
Procedimento	1 - O utilizador clica na opção para preencher o formulário; 2 - O utilizador insere texto no campo de resposta.
Dados de entrada	-
Resultado esperado	O sistema não permite que o utilizador insira texto e apenas aceita números.

Caso de teste 020	Selecionar várias opções - Elemento <i>Single Choice</i>
Pré-condições	1- Criar um formulário no sistema com um elemento do tipo <i>Single Choice</i> .
Procedimento	1 - O utilizador clica na opção para preencher o formulário; 2 - O utilizador tenta selecionar várias opções de entre a lista existente.
Dados de entrada	-
Resultado esperado	O sistema não permite que o utilizador selecione várias opções.

Caso de teste 021	Inserir texto no campo de resposta - Elemento <i>Date Picker</i>
Pré-condições	1- Existir um formulário no sistema com um elemento do tipo <i>Date Picker</i> .
Procedimento	1 - O utilizador clica na opção para preencher o formulário; 2 - O utilizador tenta inserir texto no campo de resposta do elemento <i>Date Picker</i> .
Dados de entrada	-
Resultado esperado	O sistema não permite que o utilizador insira texto.

Caso de teste 022	Efetuar logout
Pré-condições	1- Estar autenticado na aplicação.
Procedimento	1 - O utilizador clica na opção para efetuar logout.
Dados de entrada	-
Resultado esperado	A sessão é terminada e o utilizador é redirecionado para a página de login.