



UNIVERSIDADE D
COIMBRA

José Diogo Tinoco Domingos

QUICKSERVE
PRESTAÇÃO DE SERVIÇOS COM RECURSO A RECONHECIMENTO
FACIAL

Relatório de estágio no âmbito do Mestrado em Engenharia Informática, na área de especialização em Sistemas Inteligentes, orientada pelo Professor Carlos Manuel Mira da Fonseca e coorientada inicialmente pela Professora Maria Paula de Brito Graça, e posteriormente pelo Engenheiro Diogo Pacheco, apresentada ao Departamento de Engenharia Informática da Universidade de Coimbra.

Julho 2022

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

QuickServe

Prestação de Serviços com Recurso a Reconhecimento Facial

José Diogo Tinoco Domingos

Relatório de estágio no âmbito do Mestrado em Engenharia Informática, na área de especialização em Sistemas Inteligentes, orientada pelo Professor Carlos Manuel Mira da Fonseca e coorientada inicialmente pela Professora Maria Paula de Brito Graça, e posteriormente pelo Engenheiro Diogo Pacheco, apresentada ao Departamento de Engenharia Informática da Universidade de Coimbra.

Julho de 2022



UNIVERSIDADE D
COIMBRA

Esta página foi intencionalmente deixada em branco

Abstract

With the technological advances in the area of intelligent systems, the execution of tasks of greater computational effort becomes increasingly accessible. An example of this phenomenon is the increasing prevalence of facial recognition tools capable of providing greater security to various tasks and services without requiring additional human resources. As such, the increase in the efficiency and overall performance of these technologies allows for a greater presence in everyday life.

This work aimed to develop a product that integrates facial recognition technology in the context of providing security services or technical assistance, with the goal of, on the one hand, facilitating the supervision of workers by the company, and on the other hand, providing greater security to its customers.

The OutSystems low-code development platform was used to build a mobile application capable of responding to the needs of companies and their customers, providing an opportunity to explore the potential of a low-code environment for implementing intelligent systems technologies.

To this end, it was necessary to gather the requirements to be implemented, the study of both the techniques used in the area of facial recognition and previously existing products, as well as the current state of the *low-code* market.

Furthermore, the process of developing the product is described in detail. Firstly, the architecture and data model are described, as well as the changes they suffered throughout the project, with a comparison being drawn between the final version of the project and the initial planning. Next, the process of integrating the services used to carry out the project is detailed, with emphasis on the Azure, notification and Google map services. Finally, the various development phases are described, as well as the implementation process of the identified requirements, with emphasis on those that involved the use of the technologies previously mentioned.

Finally, the testing strategy used to validate the product is also described, as well as an analysis of the results obtained and the conclusions derived from them.

Keywords

Facial Recognition; Intelligent Systems; Low-code; OutSystems; Azure Cognitive Services;

Esta página foi intencionalmente deixada em branco

Resumo

Com os avanços tecnológicos na área de sistemas inteligentes, torna-se cada vez mais acessível a execução de tarefas de maior esforço computacional. Um exemplo deste fenômeno é o aumento da prevalência de ferramentas de reconhecimento facial capazes de conferir maior segurança a serviços sem requerer recursos humanos adicionais. Deste modo, o aumento da eficiência e eficácia destas tecnologias permite a integração das mesmas em soluções cada vez mais presentes no cotidiano das populações.

Este trabalho consistiu no desenvolvimento de uma solução que integra a tecnologia de reconhecimento facial num contexto de prestação de serviços de segurança ou assistência técnica, com o objetivo de, por um lado, facilitar a supervisão dos trabalhadores por parte da empresa, e por outro lado conferir maior segurança aos clientes da mesma.

Assim, recorreu-se à plataforma de desenvolvimento *low-code* da OutSystems para construir uma aplicação móvel capaz de dar resposta às necessidades das empresas e dos seus clientes, proporcionando, desta forma, uma oportunidade para explorar o potencial de um ambiente *low-code* para a implementação de sistemas inteligentes.

Para tal, foi necessário o levantamento de requisitos a implementar, o estudo quer de técnicas utilizadas na área de reconhecimento facial, quer dos produtos já existentes, e ainda uma visão atual do mercado de soluções *low-code*.

De seguida, é descrito o processo de desenvolvimento da solução. O primeiro ponto abordado é a descrição da arquitetura e modelo de dados previstos para o projeto, assim como a sua progressão ao longo do projeto, sendo traçada uma comparação entre o final do projeto e a planificação inicial. De seguida, é detalhado o processo de integração dos serviços utilizados para a realização do projeto, dos quais se destacam os serviços da Azure, os serviços de notificações, e os serviços de mapas da Google. Finalmente, são descritas as várias fases de desenvolvimento, assim como o processo de implementação dos requisitos identificados, com destaque nos que envolveram o uso das tecnologias referidas anteriormente.

Finalmente, é ainda abordada a estratégia de testes adotada para a validação da solução, assim como uma discussão dos seus resultados e as conclusões retiradas dos mesmos.

Palavras-Chave

Reconhecimento Facial; Sistemas Inteligentes; *Low-code*; OutSystems; Azure Cognitive Services;

Esta página foi intencionalmente deixada em branco

Agradecimentos

Nada na vida se alcança sozinho, pelo que, na conclusão desta etapa da minha vida, não poderia deixar de agradecer a todos os que me apoiaram até aqui.

Em primeiro lugar, gostaria de agradecer à Do iT Lean por me terem recebido na realização deste projeto, principalmente aos Engenheiros Diogo Pacheco, Fabrice Rosa e Tiago Coito por me guiarem da melhor maneira possível. Gostaria também de agradecer aos Professores Carlos Fonseca e Paula Graça pelo acompanhamento, disponibilidade e todo o apoio que prestaram ao longo deste projeto.

Agradeço também aos meus pais por proporcionarem todas as condições para o meu sucesso em todo o meu percurso académico. Esta vitória é tanto minha como vossa.

Aos meus colegas e amigos de Engenharia Informática, Diogo, Eva, Gustavo, Inês, João, Madalena, Maria, Rodrigo, Rosarinho, Rui e Xavier agradeço por me acompanharem nesta jornada, terem sempre uma palavra de apoio e ouvirem todos os desabafos.

Finalmente, agradeço à Catarina, Daniela, Inês, Joana e à Laura pela sua longa amizade e pela sua influência na pessoa que sou hoje.

Muito obrigado a todos!

Esta página foi intencionalmente deixada em branco

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	1
1.3	Contributos	2
1.4	Metodologias	2
1.4.1	Metodologia de Investigação	2
1.4.2	Metodologia de Desenvolvimento	4
1.5	Planificação do Estágio	5
1.5.1	Primeiro Semestre - Relatório Intermédio	5
1.5.2	Segundo Semestre - Relatório Final	5
1.6	Riscos	6
2	Estado da Arte	9
2.1	Visão Por Computador	9
2.2	Deteção Facial	9
2.2.1	Haar Cascades	10
2.2.2	R-CNN	11
2.2.3	SPPnets	12
2.2.4	Fast R-CNN	12
2.2.5	Faster R-CNN	13
2.3	Extração de Características	13
2.3.1	Feature-Based	14
2.4	Reconhecimento Facial	15
2.4.1	Triplet Loss	15
2.4.2	Hard Triplet Loss	15
2.4.3	Quadruplet Loss	16
2.5	Tecnologias Disponíveis	17
2.5.1	Azure Cognitive Services	17
2.5.2	Amazon Rekognition	18
2.5.3	Seleção da API	19
2.6	Produtos Existentes no Mercado	21
2.6.1	Veriff	21
2.6.2	Jumio	21
2.7	Soluções de Desenvolvimento <i>Low-code</i>	22
3	Análise dos Requisitos	27
3.1	Descrição de Atores Principais	27
3.2	Requisitos Funcionais	28
3.2.1	Comuns a todos os utilizadores	28
3.2.2	Supervisor	29
3.2.3	Funcionário	32

3.2.4	Cliente	33
3.2.5	Comuns a todos os utilizadores	34
3.2.6	Supervisor	35
3.2.7	Funcionário	35
3.2.8	Cliente	35
3.3	Diagramas de Casos de Uso	36
3.4	<i>Mockups</i>	41
4	Desenvolvimento da Solução	43
4.1	Arquitetura da Solução	43
4.1.1	Componentes de Aplicações OutSystems	43
4.1.2	Planeamento da Arquitetura	44
4.1.3	Diagrama de Entidades	47
4.1.4	Arquitetura Final	49
4.2	Implementação	52
4.2.1	Plataforma Low-code	52
4.2.2	Plugins	55
4.2.3	Fases de Desenvolvimento	63
4.2.4	Notificações	70
4.2.5	Finalização de Serviços	71
5	Testes	77
5.1	Testes Unitários	77
5.2	Testes de Fluxo	77
5.3	Testes de Reconhecimento Facial	79
5.4	Testes de Usabilidade	82
5.4.1	Discussão de Resultados	82
6	Conclusão	83
6.1	Reflexão	83
6.2	Trabalho Futuro	84
6.3	Apreciações Finais	85
A	Protótipos	91
B	Produto Final	97
C	Fluxos de Teste	103
D	Questionários de Teste	117
E	Resultados dos Inquéritos de Teste	127

Lista de Figuras

1.1	Ciclos de processamento da <i>Design Science Research</i>	3
1.2	Plano Detalhado do Estágio.	6
2.1	Abordagem de Imagem Integral	10
2.2	Camada Convolutacional	11
2.3	Camada de <i>Max Pooling</i>	11
2.4	R-CNN	12
2.5	Fast R-CNN	13
2.6	Pontos Faciais	14
2.7	Traços Faciais Relevantes para a Azure	17
2.8	Veriff	21
2.9	Jumio	22
2.10	Visão do mercado de plataformas <i>low-code</i> , segundo a Gartner [28].	23
2.11	Visão do mercado de plataformas <i>low-code</i> , segundo a Forrester [29].	24
3.1	Diagrama de Casos de Uso do Supervisor	37
3.2	Diagrama de Casos de Uso do Funcionário	38
3.3	Diagrama de Casos de Uso do Cliente	39
3.4	Diagrama Completo dos Casos de Uso	40
3.5	Protótipos: Página Inicial	41
3.6	Protótipos: Detalhes do Serviço - Funcionário	41
3.7	Protótipos: FaceId	42
3.8	Protótipos: Autenticação por Código Pin	42
4.1	Componentes de uma Aplicação	44
4.2	Architecture Canvas	44
4.3	Regras da definição de arquitetura na <i>Architecture Canvas</i>	45
4.4	Disclose e Organize	46
4.5	Assemble 1	46
4.6	Assemble 2	47
4.7	Diagrama de Entidades	47
4.8	Diagrama de Entidades OutSystems	48
4.9	Diagrama Final de Entidades OutSystems	50
4.10	Arquitetura Atualizada	51
4.11	Arquitetura Final	51
4.12	Módulos da aplicação criados no Service Studio	52
4.13	Visão dos Componentes do Service Studio da OutSystems [33].	53
4.14	Vertentes de Desenvolvimento	54
4.15	Código C# para contabilizar correspondências numa lista.	54
4.16	Transposição de código C# para <i>low-code</i>	55
4.17	Serviços selecionados da <i>FaceAPI</i>	56
4.18	Lógica do bloco <i>Take Picture</i>	57

4.19	Localização do funcionário	57
4.20	Google Cloud Platform	58
4.21	<i>Pop-up</i> no mapa	59
4.22	AutocompleteWidget	60
4.23	Configuração do projeto Firebase.	61
4.24	Configuração do projeto na API OneSignal.	61
4.25	RegisterOneSignal	62
4.26	CreateAndSendOneSignalNotification	62
4.27	Ecrã de Recuperação de Password.	64
4.28	Apresentação de Serviços	65
4.29	Filtro de serviços por data	65
4.30	Consulta das informações do serviço.	66
4.31	Relação entre a entidade FaceAPIFaceRecord e os registos de caras na API da Azure.	67
4.32	Registo de funcionários na Face API	68
4.33	Conexão aos Serviços da Azure	69
4.34	Ecrã de Autenticação Manual de Funcionários.	70
4.35	Notificações	71
4.36	Ecrã de Inquérito de Qualidade	72
4.37	Versão final do ecrã de relatório.	73
4.38	<i>Pop-up</i> de navegação entre <i>dashboard</i>	73
4.39	KPI	74
4.40	Alerta de Serviço Atrasado	75
5.1	Estratégia de teste da solução.	77
5.2	Exemplo de um teste de fluxo.	78
5.3	Testes iniciais do reconhecimento facial	79
5.4	Fracassos do reconhecimento facial.	79
5.5	Testes da qualidade de imagem - Sucessos	80
5.6	Testes da qualidade de imagem - Fracassos.	80
5.7	Projeção de pontos infravermelhos [34].	81
5.8	Distribuição das medições do tempo de reconhecimento facial.	81
E.1	Respostas à pergunta “A interface do calendário é perceptível e de fácil utilização.”	127
E.2	Respostas à pergunta “A navegação pelos ecrãs é simples.”	128
E.3	Respostas à pergunta “A utilização do calendário representa uma funcionalidade útil.”	128
E.4	Respostas à pergunta “As interações com as notificações são claras.”	129
E.5	Respostas à pergunta “Compreendo o resultado de aceitar/rejeitar a recalendarização de um serviço.”	129
E.6	Respostas à pergunta “É fácil alterar as informações pessoais da minha conta.”	129
E.7	Respostas à pergunta “É fácil criar um serviço.”	130
E.8	Respostas à pergunta “É simples introduzir informações sobre o serviço.”	130
E.9	Respostas à pergunta “Foi simples utilizar o <i>pop-up</i> para visualizar as informações do serviço.”	130
E.10	Respostas à pergunta “O mecanismo de recuperação da password é claro.”	131
E.11	Respostas à pergunta “Seguir a localização do funcionário é simples.”	131
E.12	Respostas à pergunta “Utilizaria esta aplicação para requisitar serviços? Porquê?”	132
E.13	Respostas à pergunta “Que funcionalidades sentiu que mais contribuem para a sua segurança? Acrescentaria alguma?”	132

E.14	Respostas à pergunta “A autenticação por pin é uma boa alternativa.” . . .	133
E.15	Respostas à pergunta “A interface do calendário é perceptível e de fácil utilização.”	133
E.16	Respostas à pergunta “A navegação pelos ecrãs é simples.”	134
E.17	Respostas à pergunta “A progressão do estado dos serviços é clara.”	134
E.18	Respostas à pergunta “A utilização do calendário representa uma funcionalidade útil.”	134
E.19	Respostas à pergunta “É fácil alterar as informações pessoais da minha conta.”	135
E.20	Respostas à pergunta “É fácil concluir o registo na aplicação.”	135
E.21	Respostas à pergunta “O mapa é simples de utilizar para visualizar o local do serviço.”	135
E.22	Respostas à pergunta “O mecanismo de autenticação por reconhecimento facial é confiável.”	136
E.23	Respostas à pergunta “O preenchimento do relatório é simples.”	136
E.24	Respostas à pergunta “Como considera que as ferramentas de autenticação afetam o seu trabalho?”	137
E.25	Respostas à pergunta “Que funcionalidades consideraria úteis para apoiar a sua prestação de serviços”	137
E.26	Respostas à pergunta “A gestão de recalendarização de serviços é simples.” .	138
E.27	Respostas à pergunta “A interface do calendário é perceptível e de fácil utilização.”	138
E.28	Respostas à pergunta “A navegação da lista de serviços por estado é clara.”	139
E.29	Respostas à pergunta “A navegação entre o perfil de supervisor e funcionário é clara.”	139
E.30	Respostas à pergunta “A navegação pelos ecrãs é simples.”	139
E.31	Respostas à pergunta “A utilização do calendário representa uma funcionalidade útil.”	140
E.32	Respostas à pergunta “Alterar o modo de visualização da <i>dashboard</i> é simples.”	140
E.33	Respostas à pergunta “As interações com as notificações são claras.”	140
E.34	Respostas à pergunta “É fácil verificar a localização dos funcionários.” . . .	141
E.35	Respostas à pergunta “É simples adicionar/eliminar funcionários da plataforma.”	141
E.36	Respostas à pergunta “É simples consultar o relatório e classificação do serviço.”	142
E.37	Respostas à pergunta “Foi simples definir o intervalo relevante no ecrã de KPI.”	142
E.38	Respostas à pergunta “O mapa geral foi de fácil navegação.”	143

Esta página foi intencionalmente deixada em branco

Lista de Tabelas

1.1	Tabela de Riscos	7
2.1	Comparação de Funcionalidades de Detecção Facial	19
2.2	Preços da <i>FaceAPI</i> da Azure Cognitive Services	20
2.3	Preços da API Amazon Rekognition	20
2.4	Comparação entre os líderes do mercado <i>low-code</i> , adaptada de Forrester [29].	24
5.1	Medições do tempo de autenticação do funcionário em milissegundos.	81

Esta página foi intencionalmente deixada em branco

Capítulo 1

Introdução

1.1 Motivação

Com o aumento da presença de tecnologia na vida quotidiana, as indústrias têm cada vez mais recursos disponíveis para integrar novas ferramentas, com o objetivo de agilizar processos que, caso contrário, continuariam a representar custos (temporais ou humanos) desnecessariamente elevados. Em linha com este pensamento, a Comissão Europeia (CE) definiu a transição digital como um dos principais pontos da nova estratégia industrial europeia [1]. Assim, aliando a emergente necessidade de incorporar novas tecnologias nas ferramentas das empresas com os desenvolvimentos recentes nas tecnologias de sistemas inteligentes, a Do iT Lean propôs o desenvolvimento de uma aplicação móvel que possibilite a autenticação de funcionários prestadores de serviços de segurança ou de assistência técnica com recurso a reconhecimento facial.

Parceira da OutSystems desde a sua criação, a Do iT Lean lançou esta proposta de estágio e proporcionou a orientação necessária para que fosse realizado o potencial da plataforma *low-code* da OutSystems, e desse modo, criar um produto de reconhecimento facial capaz de dar resposta à procura cada vez maior de aplicações deste género. Em particular, pretendeu-se explorar a utilização de dispositivos móveis para este fim, não só por incorporarem câmaras de qualidade suficiente para desempenhar as tarefas propostas, como também pelo fácil acesso a este recurso por parte das entidades envolvidas no processo.

No contexto de utilização para o qual se pretende desenvolver o projeto, o reconhecimento facial representa uma tecnologia que pode ser adotada em grande escala, que se espera capaz de proporcionar segurança acrescida aos clientes das empresas, aumentando a confiança na identidade do funcionário que se apresenta para prestar o serviço.

1.2 Objetivos

Tendo em conta os requisitos iniciais do projeto, assim como a solução que se pretendia desenvolver, foram delineadas as seguintes questões de investigação:

- Quais as funcionalidades que conferem maior segurança na prestação dos serviços solicitados, assim como uma facilitada gestão dos mesmos?
- Quais são as ferramentas ou serviços cognitivos de reconhecimento facial mais adequados para o desenvolvimento da aplicação?

- Qual o valor que a nova solução poderá trazer para os seus utilizadores?

O objetivo do projeto de estágio passou por desenvolver uma aplicação móvel que, com recurso a tecnologia de reconhecimento facial, apoie as empresas prestadoras de serviços de segurança ou de assistência técnica, fornecendo serviços de localização e identificação dos técnicos no local. Assim, foi desenvolvida uma aplicação móvel com recurso à plataforma de *low-code* OutSystems, com integração de serviços de inteligência artificial fornecidos pela API (*Application Programming Interface*) de Serviços Cognitivos da Azure.

Deste modo, os objetivos do estágio são:

- Definição dos requisitos e funcionalidades relevantes para satisfazer as necessidades dos utilizadores;
- Estudo das tecnologias de reconhecimento facial e dos produtos existentes atualmente na área;
- Desenvolvimento da aplicação e subsequente teste com utilizadores reais, para validação da sua utilidade;
- Identificação do valor da aplicação sob a perspetiva dos clientes através de questionários.

1.3 Contributos

A principal contribuição deste projeto é o desenvolvimento de uma aplicação com uso de tecnologia de reconhecimento facial para satisfazer as necessidades de utilizadores reais (empresas e os seus clientes), assim como aumentar a segurança e confiança que estes sentem ao utilizar a aplicação. Deste modo, a aplicação desenvolvida representa uma aplicação útil para o agendamento de serviços, assim como uma opção segura para a autenticação dos funcionários que os prestam.

1.4 Metodologias

1.4.1 Metodologia de Investigação

Primeiramente, é abordada a metodologia de investigação, que serve como base à estrutura do relatório por representar as etapas do desenvolvimento do produto. A organização do estágio baseia-se no modelo de DSR (*Design Science Research*) proposto por Hevner et al. [2]. Apesar do estágio não apresentar uma forte vertente de investigação, esta metodologia foi adotada no sentido de servir de apoio no processo e organização da síntese do projeto, ou seja, serviu de ferramenta para delinear as etapas a executar para o desenvolvimento do produto final. Deste modo, representada na Figura 1.1 está uma adaptação dos 3 ciclos de DSR estabelecidos por Hevner. O primeiro destes ciclos é o ciclo de relevância, que relaciona as atividades do *design* com o ambiente de aplicação, através da contextualização das mesmas com os requisitos e pessoas, organizações e tecnologias envolvidas. De seguida, o ciclo de rigor proporciona conhecimento das soluções já existentes para garantir a utilidade do projeto. Como tal, é necessária pesquisa de forma a delinear a base de conhecimento e as contribuições do projeto para a mesma. Por fim, o ciclo de *design* itera entre as atividades centrais de construção e avaliação dos artefactos concebidos do projeto.

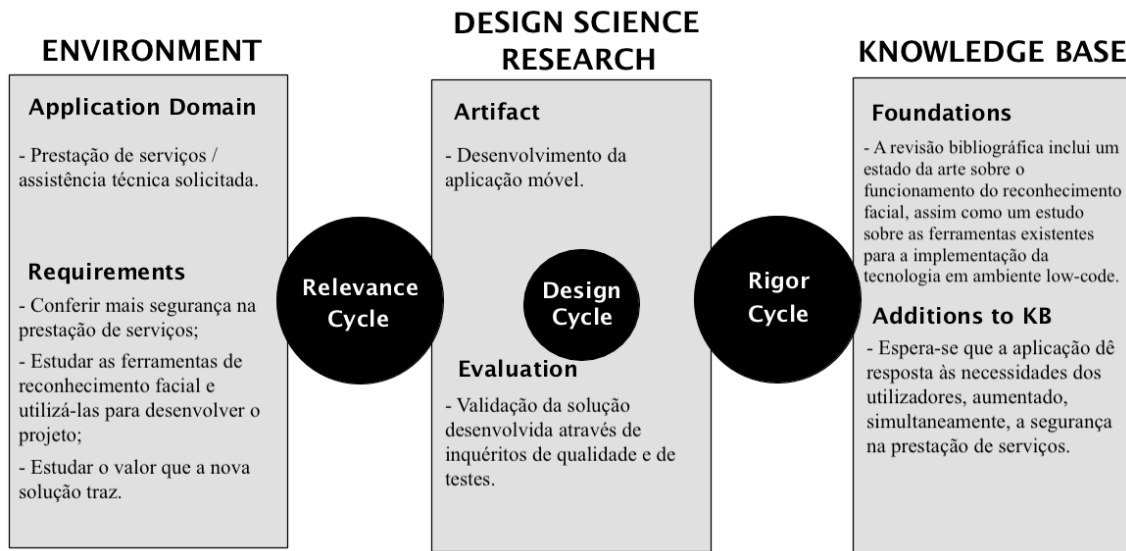


Figura 1.1: Ciclos de processamento da DSR (adaptado de Hevner et al. [2]).

Deste modo, são abordadas ao longo deste relatório 6 fases, decorrentes destes 3 ciclos, correspondentes aos seguintes capítulos:

- **Relevance Cycle - Capítulo 1**

Esta fase do modelo consiste na definição das pessoas, organizações e sistemas que interagem para atingir um objetivo. Assim, é importante identificar os problemas e oportunidades no ambiente a explorar, para que sirvam de base para a motivação do projeto.

Para a fase correspondente aos requisitos, foram consideradas as questões de investigação, ou seja, os objetivos primários do projeto para dar resposta às oportunidades identificadas na secção anterior. Assim, estes representam os requisitos gerais da aplicação, delineados na fase inicial do projeto.

- **Rigor Cycle - Capítulo 2**

Esta fase do modelo incide sobre os aspetos mais técnicos do projeto, nomeadamente a forma como as tecnologias que já existem no campo de reconhecimento facial e visão por computador, assim como os requisitos identificados, influenciaram a escolha das tecnologias adotadas para a realização do projeto. Assim, neste capítulo consta um breve estado da arte sobre o funcionamento do reconhecimento facial, assim como um estudo das soluções que atualmente existem no mercado. Desta fase resultam as contribuições do projeto para satisfazer as necessidades dos utilizadores e, simultaneamente, a utilidade da aplicação comparativamente com as soluções já existentes no mercado.

- **Design Cycle - Capítulos 3, 4 e 5**

No terceiro capítulo constam os processos de Engenharia de Software empregados para definição das funcionalidades a implementar na solução desenvolvida, concebidos a partir da visão da área de reconhecimento facial resultante do Capítulo 2. Foram ainda elaborados protótipos da aplicação, para organização das interações entre as várias vertentes do projeto.

A fase do artefacto diz respeito ao Capítulo 4, referente à implementação da solução em si, pelo que constam as definições da arquitetura e os modelos de dados utilizados na mesma, bem como os vários aspetos do *design* e o seu desenvolvimento. Para este projeto, o artefacto consiste numa aplicação móvel, desenvolvida na plataforma *low-code* da OutSystems, para uso de funcionários e clientes de uma empresa de prestação de serviços, com uma vertente de reconhecimento facial para fortalecer a segurança e confiança nas partes envolvidas no processo.

A avaliação da solução desenvolvida é fundamental para compreender se o artefacto consegue dar resposta aos problemas identificados no domínio da aplicação. Deste modo, estão descritos no Capítulo 5 os testes realizados para avaliar o desempenho da solução, assim como inquéritos de satisfação direcionados a voluntários que testaram as funcionalidades da aplicação e, posteriormente, a análise dos resultados observados.

1.4.2 Metodologia de Desenvolvimento

A OutSystems define metodologias ágeis como abordagens de desenvolvimento de *software* que permitem às equipas reagir a alterações no projeto, minimizando o tempo despendido em processos de aprendizagem e adaptação [3]. Assim, as metodologias ágeis diferenciam-se por envolverem o cliente no processo de desenvolvimento, uma vez que o projeto é dividido em etapas mais curtas, que resultam num artefacto apresentado ao cliente no final de cada uma. Deste modo, o cliente dá *feedback* de forma contínua ao longo do projeto, o que aumenta a probabilidade deste satisfazer as necessidades do cliente.

Assim, uma equipa ágil trabalha com um número reduzido de elementos, cujas prioridades são desenvolver o produto enquanto se minimiza o tempo gasto em documentação, incluir o cliente no processo de desenvolvimento e reagir a alterações em vez de seguir um plano rígido. O conceito de metodologia ágil permite, assim, definir a filosofia de trabalho adotada pela equipa. No entanto, não é suficiente para definir como as equipas devem organizar-se para concretizar os ideais ágeis. Com este objetivo, surgem as *frameworks* ágeis, como por exemplo, o *Scrum*.

Na sua generalidade, os projetos de *software*, pela sua própria natureza, beneficiam fortemente da adoção de metodologias ágeis, sendo, esta a metodologia de trabalho usada na Do iT Lean através da *framework Scrum*, que é recomendada para projetos cujos requisitos apresentam maior ambiguidade.

Ao seguir uma metodologia ágil, caracterizada por ser iterativa, devem ser delineados objetivos a cumprir no espaço de 1-4 semanas, no final do qual a equipa deve ser capaz de implementar uma versão utilizável das funcionalidades desenvolvidas. Este período de tempo designa-se por *sprint*. No início de cada *sprint* são selecionadas tarefas do *product backlog*, onde é feito o registo das tarefas a realizar e a sua descrição. Esta fase de planeamento, também conhecida como *sprint planning*, tem como objetivo principal responder a duas questões: o que deve ser entregue até ao final do *sprint*, e como a equipa deve proceder para atingir o objetivo.

Ao longo do *sprint*, são realizadas reuniões diárias para comunicar o trabalho desenvolvido ao final de cada dia, o trabalho que se pretende desenvolver presentemente, assim como as dificuldades encontradas. Estas reuniões designam-se por *Daily Scrum*, e não devem durar mais de 15 minutos. Como referido anteriormente, no final de cada *sprint* há uma reunião com o cliente para que este comente sobre o produto desenvolvido ao longo do *sprint*.

Finalmente, as equipas na *framework Scrum* contemplam 3 papéis diferentes a desempenhar pelos seus elementos:

- **Equipa de Desenvolvimento:** Composta por todos os membros da equipa responsáveis por desenvolver o produto, correspondente ao estagiário José Domingos;
- **Product Owner:** Responsável por priorizar tarefas e certificar-se que a equipa de desenvolvimento sabe o trabalho a realizar, desempenhado inicialmente pela orientadora Paula Graça e posteriormente pelo orientador Diogo Pacheco;
- **Scrum Master:** Este elemento tem como papel garantir que são seguidos os princípios da *framework Scrum*, assim como resolver eventuais dificuldades que surjam ao longo dos *sprints*. Este papel foi desempenhado pelo orientador técnico Tiago Coito.

1.5 Planificação do Estágio

1.5.1 Primeiro Semestre - Relatório Intermédio

Incluído na duração do estágio esteve um processo de formação proporcionado pela empresa Do iT Lean, com principal incidência sobre a plataforma de *low-code* utilizada no decorrer do estágio, assim como conceitos básicos de tecnologias web, SQL e metodologias ágeis. Durante esta etapa do estágio não foram registadas alterações ao plano, pelo que tudo correu conforme o previsto.

De seguida, deu-se início ao levantamento de requisitos funcionais, bem como a elaboração de protótipos rápidos da aplicação móvel. Simultaneamente, foram estudadas tecnologias do estado da arte do reconhecimento facial, assim como produtos já existentes no mercado para comparação dos mesmos, a fim de tomar uma decisão sobre os que seriam incluídos no produto final. Por fim, foi também delineada a arquitetura da aplicação a desenvolver.

Concluindo o primeiro semestre, foi desenvolvida uma pequena demonstração da aplicação, com objetivo de testar as tecnologias de reconhecimento facial selecionadas. Simultaneamente, foi finalizado o relatório intermédio que, face à situação pandémica vivida na altura, viu a data limite para a sua submissão estendida uma semana.

1.5.2 Segundo Semestre - Relatório Final

A planificação do segundo semestre consistiu, conforme estabelecido pela *framework Scrum*, em *sprints* consecutivos de aproximadamente três semanas para desenvolvimento da aplicação, finalizados com demonstrações das funcionalidades desenvolvidas na duração dos mesmos. Para finalizar o estágio foi incluída uma fase de teste da solução desenvolvida, seguida de um período dedicado ao relatório final.

Na Figura 1.2 é apresentada uma representação da planificação de todo o estágio, fornecida pela empresa Do iT Lean.

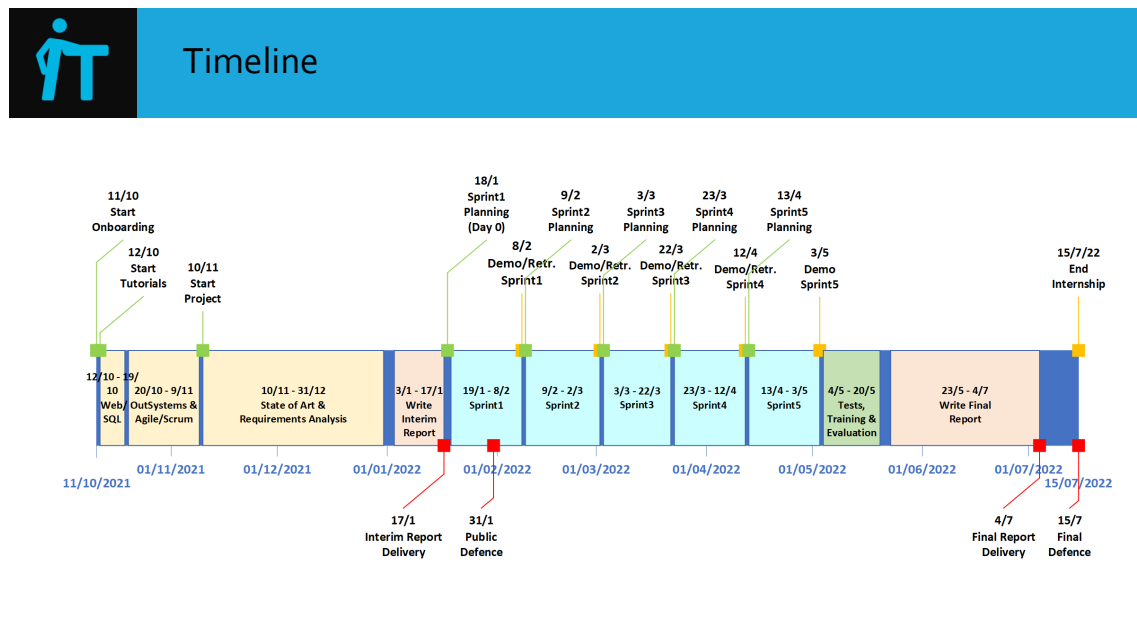


Figura 1.2: Plano Detalhado do Estágio.

1.6 Riscos

A ocorrência de imprevistos é um facto inevitável no desenvolvimento de qualquer projeto, pelo que é imperativa a identificação atempada dos mesmos e a elaboração de planos de mitigação correspondentes, para que as eventuais dificuldades possam ser minimizadas e, assim, o seu impacto sobre o projeto possa ser reduzido. Deste modo, no início do projeto foram identificados riscos, classificados pela sua probabilidade e o seu impacto segundo a seguinte classificação:

- **Probabilidade**

- Pouco Provável;
- Provável;
- Muito Provável.

- **Impacto**

- Baixo;
- Moderado;
- Elevado.

Assim, seguem-se os riscos identificados, bem como as suas consequências e respetivos planos de mitigação:

- **R1:** Questões de natureza ética.
Uma vez que o projeto implica a utilização de informações sensíveis, mantidas sob políticas de proteção de dados, deve ser revisto por uma comissão de ética.
- **R2:** As tecnologias adotadas não se revelarem adequadas ao projeto.
Caso as tecnologias adotadas se revelem inadequadas ao projeto, a alternativa seria

uma substituição de ferramentas numa fase avançada do projeto, o que apresenta um impacto enorme e implica a insatisfação de todos os requisitos propostos. Assim, deve ser feito um estudo atempado das funcionalidades disponíveis nas tecnologias adotadas e verificar se estão alinhadas com os requisitos funcionais.

- **R3:** Alteração dos requisitos numa fase avançada do projeto.
O plano de mitigação para este risco passa pela definição concreta dos requisitos mais importantes do projeto, assim como a implementação prioritária dos requisitos que apresentam menor probabilidade de sofrer alterações. Deste modo, existe uma menor probabilidade de alteração de requisitos depois da sua implementação.

Tendo identificado estes riscos, a tabela 5.1 apresenta as informações referentes aos mesmos de um forma mais estruturada.

ID	Data de levantamento	Probabilidade	Impacto	Plano de Mitigação
R1	14/12/2021	Provável	Moderado	Revisão do projeto por uma comissão de ética.
R2	14/12/2021	Pouco Provável	Elevado	Realização um estudo atempado das funcionalidades disponíveis nas tecnologias adotadas e verificar se estão alinhadas com os requisitos funcionais.
R3	16/12/2021	Provável	Elevado	Implementação prioritária dos requisitos que apresentam menor probabilidade de sofrer alterações.

Tabela 1.1: Tabela de Riscos
Identificação de riscos e respetivos planos de mitigação.

Depois da conclusão do projeto, foi possível constatar que o primeiro risco identificado não se verificou, uma vez que a API utilizada para o reconhecimento facial não guarda fotografias dos utilizadores. Quanto ao segundo risco, foi seguido o plano de mitigação, pelo que as tecnologias utilizadas revelaram-se adequadas para o desenvolvimento do projeto, pelo que o impacto deste risco no projeto foi inexistente. Finalmente, quanto ao terceiro risco, foi também seguido o plano de mitigação, pelo que os requisitos funcionais prioritários foram todos implementados atempadamente. O único requisito não implementado apresentava a prioridade mais baixa, não tendo sido levado a cabo não por uma questão de tempo, mas sim por se ter considerado que algumas melhorias na usabilidade trariam mais valor à solução.

Esta página foi intencionalmente deixada em branco

Capítulo 2

Estado da Arte

Neste capítulo é apresentada uma contextualização das tecnologias do Estado da Arte do ramo de reconhecimento facial. Primeiramente, são apresentadas soluções para o problema de detecção facial, seguidas de algumas abordagens para a extração de características das faces detetadas. De seguida, são exploradas técnicas para treinar a identificação facial. Finalmente, é feito um estudo das ferramentas de reconhecimento facial mais relevantes e a sua comparação, assim como um estudo do mercado de soluções de desenvolvimento *low-code*.

2.1 Visão Por Computador

Antes de abordar a temática do reconhecimento facial, é fundamental compreender o ramo mais abrangente da visão por computador. Este ramo de inteligência artificial permite que sistemas extraiam informações de imagens, vídeos ou outros tipos de informação visual. Deste modo, sistemas de visão por computador utilizam câmaras, sensores e algoritmos de processamento de dados para replicar a visão humana.

A visão por computador representa uma área extremamente importante dos sistemas inteligentes, uma vez que permite uma maior interação dos sistemas com o mundo que os rodeia. Se a inteligência artificial permite aos computadores pensar, a visão por computador permite-lhes ver, observar e compreender [4].

2.2 Detecção Facial

Qualquer solução de reconhecimento facial com aplicação no mundo real tem, necessariamente, de incorporar um mecanismo de detecção de caras. O principal objetivo da detecção facial é reconhecer o tamanho e localização de caras numa dada imagem, caso existam, conseguindo delinear uma *bounding box*, ou seja, um retângulo em torno das caras detetadas. Esta área de visão por computador já foi alvo de extenso estudo ao longo dos anos, pelo que existem diversas soluções para o problema.

2.2.1 Haar Cascades

Uma das primeiras abordagens que revolucionaram esta área de estudo passa pela decomposição da imagem em retângulos, que desempenham o papel de classificadores, sobre os quais serão aplicadas operações de soma e diferenças sobre os valores dos píxeis, com o objetivo de encontrar zonas de diferentes luminosidades. Esta técnica baseia-se no facto de diferentes zonas da cara apresentarem diferentes níveis de luminosidade numa imagem. A decomposição em si terá que ser aplicada de várias formas, com o objetivo de comparar diferentes zonas da imagem, uma vez que diferentes zonas da cara têm relações diferentes entre si.

Contudo, esta abordagem apresenta uma limitação no tempo computacional que estas operações podem requerer em imagens de grandes dimensões. Para contornar este problema, foi sugerida a utilização de uma Imagem Integral [5]. Esta técnica, representada na Figura 2.1, requer o tratamento da imagem para uma versão onde o novo valor em cada pixel equivale à soma dos valores de todos os píxeis que se encontram por cima e à esquerda do mesmo. Assim, as operações sobre esta nova imagem tornam-se muito mais rápidas quando aplicadas a grandes imagens.

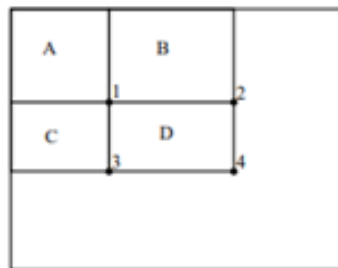


Figura 2.1: Abordagem de Imagem Integral proposta por [5]. O valor no ponto 1 é a soma dos valores na região A, no ponto 2 é a soma dos valores nas regiões A+B, no ponto 3 é a soma de A+C e no ponto 4 é a soma de todas as regiões. Assim, seja v o valor em cada um dos pontos, a soma dos valores na região D é dada por $v_4 + v_1 - (v_2 + v_3)$.

No entanto, o processo continua a ser demasiado demoroso, uma vez que existem vários classificadores para conjugar diferentes secções da imagem, que terão que ser aplicados em várias zonas da imagem. No entanto, nem todos os classificadores são igualmente complexos, pelo que surgiu a abordagem da Attentional Cascade [5]. Segundo este modelo, os classificadores serão usados em sucessão, começando pelos mais simples, e rejeitam apenas as zonas que nitidamente não podem conter uma cara. Desta forma, os classificadores mais complexos só serão utilizados em zonas candidatas que não tenham sido excluídas nos passos anteriores.

Apesar da relativa simplicidade destes métodos, os resultados obtidos são, no geral, satisfatórios, pelo que a utilização desta abordagem é frequente em várias aplicações. Contudo, esta solução apresenta alguns problemas no que diz respeito à deteção de caras em posições não frontais, pelo que surgem dificuldades em situações menos estruturadas, por exemplo, numa rua movimentada. Para tentar resolver esta dificuldade, têm sido estudadas soluções com recurso a algoritmos de *deep learning*, como por exemplo R-CNN (*Region-based Convolutional Network*) que foi, entretanto, adaptada para outras abordagens como, por exemplo, Fast R-CNN e Faster R-CNN.

2.2.2 R-CNN

Uma das primeiras abordagens a utilizar CNNs (*Convolutional Neural Networks*) foi a da janela deslizante, onde se treina uma rede convolucional para reconhecer a presença de uma cara humana dentro de uma área limitada, que será colocada em todas as possíveis posições na imagem [6]. Contudo, esta solução apresentava um elevado custo computacional devido a recorrentes computações da CNN. Assim, surgiu o objetivo de reduzir o número de posições candidatas a serem percorridas pela janela deslizante, que levou ao desenvolvimento de métodos de propostas de regiões, cujo objetivo é encontrar zonas na imagem com maior probabilidade de conter uma cara.

Desta forma surge a R-CNN, constituída por duas componentes. Primeiramente, é gerado um conjunto de aproximadamente 2000 propostas de regiões de interesse, recorrendo a várias técnicas como *Objectness*, Procura Seletiva ou Propostas de Objetos Independentes de Categorias, entre outros [7]. De seguida, a região circundante de cada proposta é convertida para uma imagem de tamanho fixo, e submetida à CNN com várias camadas de processamento das quais se destacam as camadas convolucionais e de *max pooling*. As camadas convolucionais têm como objetivo extrair um mapa de características a partir de uma imagem através da utilização de um filtro, ou seja, os valores nos vários píxeis da imagem são convolidos com os valores no filtro para obter a relação entre os mesmos, como se ilustra na Figura 2.2. O resultado deste processo é designado mapa de características. Por outro lado, uma camada de *max pooling* comprime este mapa ao agregar os valores dentro do mesmo e selecionando os maiores valores, como é ilustrado na Figura 2.3.

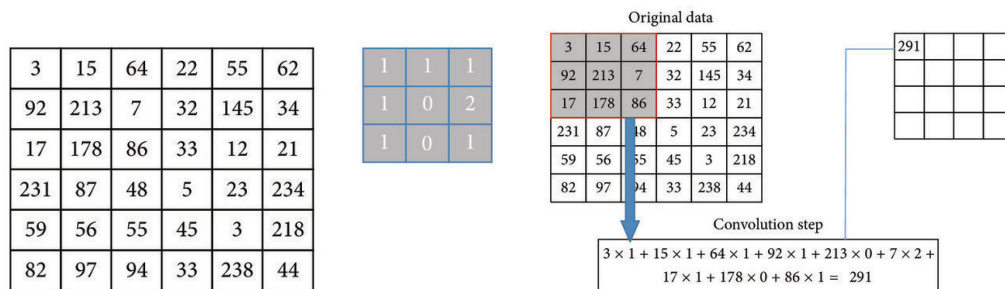


Figura 2.2: Na parte superior está representada a imagem original assim como o filtro da camada convolucional. Na parte inferior está um exemplo do cálculo convolucional [8].

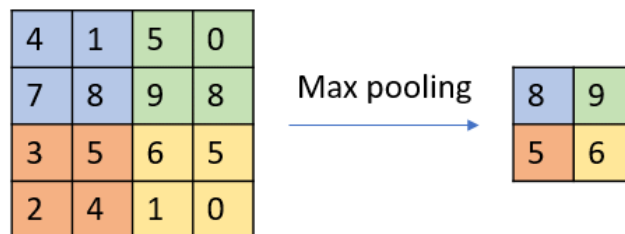


Figura 2.3: Exemplo da aplicação de uma camada de *max pooling* [9].

Por fim, o resultado do processamento da imagem pela CNN é um vetor de características de 4096 dimensões [7][10], que é submetido a um processo de classificação com recurso a uma SVM (*Support Vector Machine*) [7]. Através deste método é possível utilizar a elevada precisão das CNN em problemas de classificação na área da deteção de objetos. Na Figura 2.4 está apresentada a arquitetura da R-CNN descrita anteriormente.

Além destas abordagens, existe ainda a possibilidade de variar a configuração da CNN, como por exemplo, no caso da *Two Stream CNN* e da *Cascade CNN*, cuja utilização é mais vantajosa para problemas de detecção de objeto único e de multi-objects, respetivamente [6]. Contudo, para cada proposta de objeto gerada, a R-CNN necessita de recorrer à rede convolucional para poder extrair o vetor de características, resultando num elevado custo computacional. Uma das abordagens desenvolvida para contornar este problema foi a Fast R-CNN [6], que é discutida na secção correspondente.

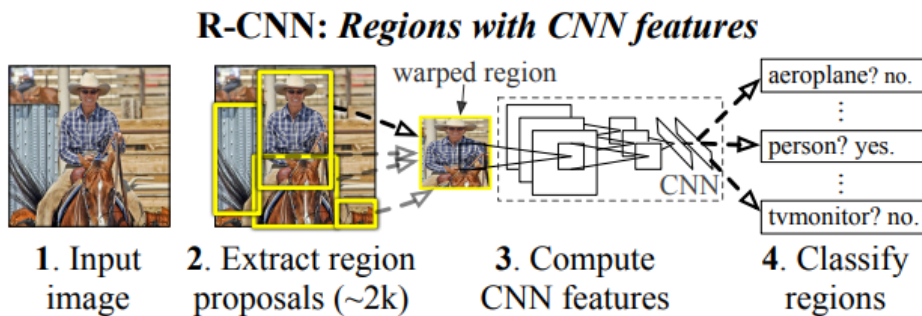


Figura 2.4: Arquitetura da R-CNN como sugerido em [7].

2.2.3 SPPnets

Tal como referido anteriormente, uma grande limitação do R-CNN é o uso recorrente da CNN para todas as propostas de objetos. Uma sugestão para mitigar este problema é o uso de SPPnets (*Spatial pyramid pooling networks*). Em vez de gerar um mapa de características para cada proposta de objeto, a SPPnet cria apenas um mapa para a imagem total (também com recurso a camadas convolucionais), extraindo posteriormente os vetores de características através de *max pooling* da região do mapa correspondente à proposta de objeto. Os resultados desta abordagem são melhores, embora continue a revelar-se muito limitada, em parte por partilhar grande parte da sua arquitetura com a R-CNN [11].

2.2.4 Fast R-CNN

Apesar dos resultados promissores da R-CNN, essa abordagem apresenta algumas limitações, nomeadamente o elevado custo do processo de treino em termos de tempo e espaço, assim como uma lenta deteção de objetos. Mesmo as SPPnets, com os resultados mais rápidos quando comparadas com a R-CNN, continuam a apresentar limitações com redes muito profundas. Assim, a tentativa de melhorar os resultados da R-CNN levou ao desenvolvimento da Fast R-CNN, cujo objetivo é fornecer toda a imagem à CNN uma única vez, evitando a utilização recorrente que ocorre no caso da R-CNN.

Assim, a Fast R-CNN recebe como parâmetros a imagem inteira e um conjunto de propostas de objetos. A imagem começa por ser processada por várias camadas convolucionais e *max pooling* para produzir um mapa de características, à semelhança da R-CNN. De seguida, para cada proposta, uma camada de *pooling* extrai do mapa de características um vetor de características de tamanho fixo. Cada um destes vetores é fornecido a um conjunto de camadas *fully connected*, que, por sua vez, fornecem o resultado a duas camadas de *output*. A primeira é uma camada *softmax*, que normaliza os valores numa escala entre 0 e 1, para que possam ser interpretados como probabilidades. Assim, são devolvidas as probabilidades do objeto pertencer a $K+1$ classes, uma para cada classe de objeto, mais

uma para objetos que não se enquadram em nenhuma. Finalmente, a outra camada de *output* devolve 4 parâmetros para cada classe, que servirão de base para a *bounding box* [11]. As etapas desta abordagem são representadas na imagem 2.5.

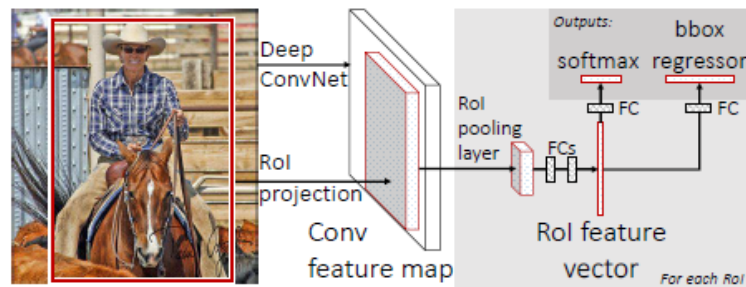


Figura 2.5: Arquitetura da Fast R-CNN como sugerido em [11].

Estas tecnologias mostram ser promissoras na medida em que os tempos de treino das redes são reduzidos o suficiente para utilizações em ambientes menos estruturados, como por exemplo uma rua movimentada.

2.2.5 Faster R-CNN

Como já foi visto, a R-CNN e a Fast R-CNN operam sobre as propostas de objetos genéricos que são normalmente criadas a partir de modelos como a Procura Seletiva. Como constatado em [10], estas abordagens para a geração de propostas de objetos representam um custo computacional elevado, chegando mesmo a representar a maior parte da duração da execução da Fast R-CNN. O facto de as abordagens com representações *deeply learned*, ou seja, treinadas para o desempenho de tarefas específicas, apresentarem normalmente melhor generalização, conjugado com um menor custo computacional da geração de propostas, levou ao surgimento da Faster R-CNN.

De um modo geral, esta abordagem é composta por dois módulos: a RPN (Rede de Propostas Regionais), que é uma rede convolucional responsável por gerar propostas de objetos, e uma Fast R-CNN, cuja função é aprimorar as propostas geradas pelo primeiro módulo. Assim, a ideia principal é partilhar as camadas convolucionais utilizadas por estas duas componentes, de modo a que a imagem só tenha de ser processada uma vez pela CNN a fim de gerar as propostas de objetos e proceder à sua classificação [10].

Deste modo, através da partilha das camadas convolucionais dos dois módulos, é possível a utilização de várias camadas convolucionais para a RPN sem um custo computacional acrescido.

2.3 Extração de Características

A extração de características é o processo de redução da dimensionalidade de um problema, de forma a representar a informação mais relevante. Para o reconhecimento facial, uma vez que este processo implica geralmente operações de mapeamento e localização, é frequente que esta etapa seja efetuada simultaneamente com a deteção facial.

2.3.1 Feature-Based

A abordagem *Feature-Based* é uma das principais abordagens utilizadas em aplicações de reconhecimento facial e consiste na localização dos traços do rosto presente numa imagem, para que possam, posteriormente, ser fornecidos a classificadores. Geralmente, é dada prioridade à localização dos olhos, uma vez que estes fornecem informações sobre as dimensões da cara e facilitam a localização de outros traços faciais. Além disso, são dos traços faciais que apresentam menor variabilidade entre caras, sendo, simultaneamente, pouco afetados por alterações nos pelos faciais.

A abordagem sugerida em [12] recebe como parâmetro de entrada um *face map*, ou seja, uma imagem binária, ou seja, constituída por duas cores (normalmente, preto e branco), com as regiões em que foi detetada uma cara. De seguida, são executados dois módulos para localização dos olhos, sendo o primeiro utilizado para uma deteção aproximada da região onde estes se encontram, e o segundo para um ajuste que garante maior precisão dos resultados. Para ambos os módulos são utilizadas SVM, sendo a principal diferença a área em que são utilizadas.

Para o primeiro módulo, a área é estimada com base no tamanho da cara detetada na fase de deteção facial. No segundo módulo, é utilizado o resultado do primeiro e repetida a utilização de uma SVM, desta vez considerando uma área mais reduzida na vizinhança da localização resultante do primeiro módulo. Depois de ter localizado o centro dos olhos, os limites dos mesmos são estimados com base no tamanho da cara, a sua posição, e a distância entre os dois olhos.

Quanto aos restantes traços faciais, a ponta do nariz é simples de estimar com base em contrastes de luminosidade com as áreas circundantes, enquanto para a boca é utilizado um modelo de contorno ativo para identificar os cantos e os pontos intermédios dos lábios. Finalmente, com estes pontos localizados, são inferidos os restantes. Na Figura 2.6 estão representados a verde os pontos referidos anteriormente, e a vermelho os pontos inferidos com base nos mesmos.

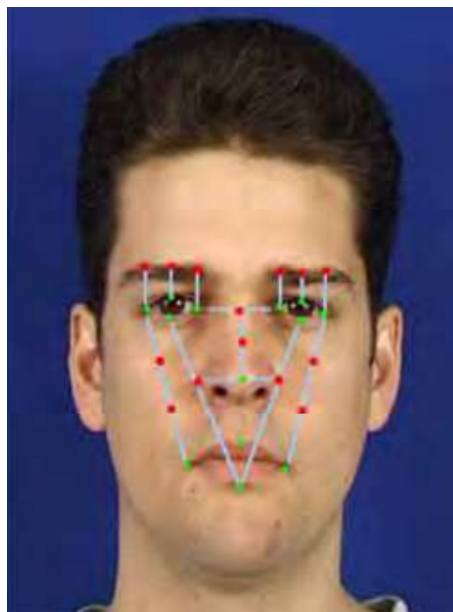


Figura 2.6: Os 27 pontos faciais obtidos pelo método descrito em [12].

2.4 Reconhecimento Facial

Após o processo de detecção facial, um dos passos mais importantes na área de reconhecimento facial é a identificação facial, ou seja, estabelecer correspondências entre caras identificadas. Para tal, é fundamental analisar os métodos de treino utilizados para que os classificadores consigam distinguir as correspondências positivas das correspondências negativas. Assim, seguem-se alguns dos métodos utilizados.

2.4.1 Triplet Loss

Uma das abordagens para o treino de soluções de reconhecimento facial é a utilização de funções *triplet loss*. A ideia que serve de fundamento a estas funções é a seleção de uma foto de uma cara que servirá de âncora, uma outra da mesma cara numa posição ligeiramente diferente que será uma correspondência positiva, e uma última cara de uma pessoa diferente, que será uma ocorrência negativa. Deste modo, pretende-se que a distância entre a âncora e a correspondência positiva seja menor do que a distância entre a âncora e a correspondência negativa por uma margem m [13].

Assim, o ponto de partida é definir o cálculo da distância entre as correspondências mencionadas anteriormente. Como é frequente em problemas desta natureza, existem vários tipos de métricas disponíveis para calcular distâncias no espaço de características, pelo que existe liberdade consoante a métrica a utilizar. Contudo, para a generalidade das situações, utiliza-se o quadrado da distância euclidiana.

Tendo isto em mente, e considerando cada indivíduo como uma classe distinta y , a função *triplet loss* é descrita pela expressão 2.1, onde θ representa a projeção no espaço de características, m a margem entre classificações positivas e negativas e D a distância entre dois pontos no espaço de características.

$$L_{tri}(\theta) = \sum_{\substack{a,p,n \\ y_a=y_p \neq y_n}} [m + D_{a,p} - D_{a,n}]_+ \quad (2.1)$$

O princípio que serve de base a esta abordagem é treinar o classificador para que, ao retirar as características mais relevantes das 3 imagens e projetar as mesmas no espaço de características (com a projeção θ), consiga compreender que o exemplo positivo (p) está muito mais próximo da âncora (a) do que o exemplo negativo (n). Deste modo, o processo de treino passará por conseguir que a solução consiga definir o quão próximas duas caras devem estar no espaço de características para que sejam consideradas como pertencentes a mesma pessoa.

2.4.2 Hard Triplet Loss

Um dos problemas identificados nas funções *triplet loss* é o facto de, para cada cara adicionada ao dataset, a quantidade de possíveis combinações aumentar a um ritmo muito elevado, uma vez que para cada cara são necessárias 3 fotos (uma âncora, e uma correspondência positiva e negativa). Assim, um dataset de grandes dimensões tende a apresentar tempos de treino elevados. Há ainda a referir que nem todas os exemplos serão igualmente relevantes. Por exemplo, devem ser prioritários os casos em que a classificação é mais difícil, por exemplo, duas pessoas diferentes com caras semelhantes (*Hard Negatives*), ou uma pessoa com duas expressões faciais muito diferentes (*Hard Positives*). No entanto, o

processo de treino não deve ser limitado a estes casos, uma vez que a escolha exclusiva de casos extremos resulta numa má generalização de associações mais frequentes.

Assim, para contornar estas limitações, foram sugeridas várias alternativas [13], sendo a abordagem dos Hard Triplets a mais relevante. Nesta abordagem, são selecionadas aleatoriamente P classes, correspondentes a P pessoas, e para cada uma delas K imagens também aleatórias. Deste modo, para cada âncora existente nas imagens da amostra, selecionam-se as correspondências mais difíceis, seguindo a expressão 2.2.

$$L_{BH}(\theta) = \sum_{i=1}^P \sum_{a=1}^K [m + \max_{p=1, \dots, K} D(x_a^i, x_p^i) - \min_{\substack{n=1, \dots, K \\ j=1, \dots, P \\ j \neq i}} D(x_a^i, x_n^j)]_+ \quad (2.2)$$

Note-se que, sendo n e j pessoas diferentes, o termo $\max D(x_a^i, x_p^i)$ representa as correspondências positivas mais difíceis de classificar, uma vez que a sua distância é máxima, e $\min D(x_a^i, x_n^j)$ as correspondências negativas mais difíceis de classificar, uma vez que a sua distância é mínima.

Assim, esta abordagem soluciona algumas das limitações referidas anteriormente, uma vez que as correspondências utilizadas para treino são referentes a casos mais desafiantes, pelo que a informação extraída dos mesmos é maior. Este modelo apresenta, também, maior escalabilidade, uma vez que não é necessário utilizar todo o *dataset*. Isto porque, ao parametrizar P e K , é possível que sejam selecionadas apenas as correspondências mais difíceis de uma parte mais reduzida do *dataset*. Esta ideia vai ao encontro com o que foi expresso anteriormente, pelo que, não devendo limitar o processo de treino a estes casos, são escolhidos apenas alguns dos casos mais difíceis.

2.4.3 Quadruplet Loss

Uma outra abordagem baseada nas funções *triplet loss* são as funções *quadruplet loss* que, como o nome indica, são funções que acrescentam um parâmetro de decisão à função *triplet loss*, sob a forma de uma outra correspondência negativa. Neste caso, além da comparação da distância entre correspondências positivas ou negativas, será também medida a distância entre duas correspondências negativas. Como observado anteriormente, podemos calcular a distância entre duas projeções de imagens para medir a proximidade entre as mesmas. Desta forma, a função *quadruplet loss* segue a expressão 2.3. Note-se que i e j pertencem à mesma classe, enquanto i , l e k pertencem a 3 classes distintas.

$$L_{quad} = \sum_{i,j,k}^N [D(x_i, x_j)^2 - D(x_i, x_k)^2 + \alpha_1]_+ + \sum_{i,j,k,l}^N [D(x_i, x_j)^2 - D(x_l, x_k)^2 + \alpha_2]_+ \quad (2.3)$$

O primeiro termo, tal como na *triplet loss*, mede a distância entre pares positivos e negativos. O segundo termo permite que a distância intra-classes seja menor do que a distância inter-classes, uma vez que, além de medir a distância entre pares positivos, irá também medi-la entre pares negativos. Ou seja, não só se minimiza a distância entre correspondências positivas, como também se maximiza a distância entre correspondências negativas. No entanto, é importante que o primeiro termo seja prioritário, uma vez que a distinção entre classificações positivas e negativas é de carácter mais relevante, enquanto o segundo termo serve para alargar a variação entre classes e melhorar os resultados obtidos, pelo que

assume um papel auxiliar. Desta forma, o valor de α_1 deve ser maior, reservando assim uma margem mais reduzida para o segundo termo [14].

2.5 Tecnologias Disponíveis

Nesta secção serão discutidas as tecnologias incorporadas na solução desenvolvida, bem como a fundamentação das escolhas efetuadas relativamente às mesmas. Em particular, serão discutidas as opções para a componente de reconhecimento facial.

Um dos critérios mais importantes a considerar para a seleção da API para reconhecimento facial foi a compatibilidade com o Service Studio da OutSystems, ou seja, a API selecionada tem, necessariamente, que ser compatível com a plataforma onde o projeto foi desenvolvido. Tendo em conta esta condição, as principais ferramentas consideradas foram os Azure Cognitive Services e a Amazon Rekognition. Segue-se a análise de cada uma delas e, posteriormente, a comparação entre as mesmas.

2.5.1 Azure Cognitive Services

Entre várias modalidades disponíveis nos serviços da Azure, é de particular relevância o módulo *Azure Face Services*, que foi desenvolvido especificamente para deteção e análise de caras e integra várias funcionalidades úteis para o desenvolvimento de projetos com recurso a reconhecimento facial. Primeiramente, a API fornece serviços de deteção facial, com os quais é possível fornecer uma imagem e obter as coordenadas da localização de várias caras humanas presentes nessa. Para efeitos de registo, é atribuído um identificador único aos dados referentes a cada cara detetada para utilização em operações posteriores. Há ainda a salientar a possibilidade de extrair atributos das caras detetadas. Um exemplo seria o dos traços faciais, que são, ao todo, os 27 pontos na cara mais simples de detetar, como por exemplo os olhos, as sobrancelhas, o nariz, a boca, etc. Todos os traços faciais encontram-se representados na Figura 2.7.

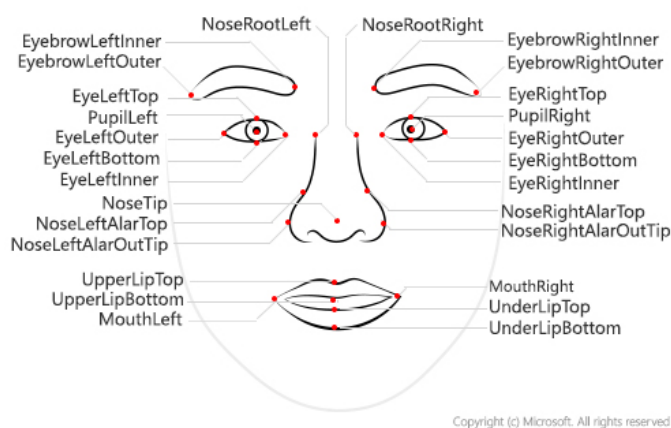


Figura 2.7: Os 27 traços faciais detetados pela API [15].

Além destes traços faciais, existem ainda vários atributos que podem ser extraídos da imagem, como por exemplo a idade, a existência de acessórios, as emoções demonstradas, orientação da cabeça, etc. [15]. Do ponto de vista da aplicação a desenvolver, estes atributos podem ser bastante úteis, por exemplo, para dar *feedback* ao funcionário, no caso da presença de acessórios impossibilitar a autenticação por reconhecimento facial.

Este último ponto é particularmente relevante no cenário pandêmico que tem vindo a ser sentido em todo o mundo.

Em segundo lugar, são fornecidos pela API serviços de identificação, ou seja, é possível definir um grupo de pessoas como um *dataset* (capaz de conter dados de até 1 milhão de pessoas, com um limite de 248 exemplos de caras por pessoa) e treinar o reconhecimento das pessoas no grupo com as caras existentes no mesmo. Assim, é possível verificar se a cara detetada pela aplicação pertence ao grupo definido pelo *dataset* utilizado para treino. Existe também a opção de validação, cujo objetivo é determinar se a cara recebida pela aplicação corresponde a uma cara específica do grupo. Difere da opção de identificação na medida em que é necessário haver correspondência direta com um único elemento, e não meramente pertencer ao grupo de treino. Trata-se, portanto, de uma diferença entre *One-to-Many* e *One-to-One*. Finalmente, existem ainda opções de seleção de caras semelhantes e de agrupamento de caras [16], não sendo estas particularmente relevantes para o projeto a desenvolver.

As imagens devem apresentar as seguintes características:

- Ser fornecidas no formato JPEG, PNG, GIF (utilizada a primeira frame) ou BMP.
- O tamanho máximo dos ficheiros é 6 MB;
- Para imagens de dimensões até 1920 x 1080 píxeis, o tamanho mínimo de uma cara detetável é 36 x 36 píxeis. Imagens com dimensões superiores ao indicado têm um tamanho mínimo para caras detetáveis proporcionalmente maior.
- O tamanho máximo de uma cara detetável é de 4096 x 4096 píxeis.
- Caras com dimensões fora dos limites especificados não são detetáveis.
- Algumas caras poderão não ser detetadas face a dificuldades técnicas, tais como:
 - Níveis inadequados de luminosidade;
 - Obstruções que bloqueiem um ou ambos os olhos;
 - Alterações no cabelo ou pêlos faciais;
 - Alterações nos traços faciais resultantes do envelhecimento;
 - Expressões faciais extremas.

2.5.2 Amazon Rekognition

De forma semelhante a outras API de serviços de reconhecimento facial, a Amazon Rekognition providencia serviços de deteção facial e comparação de caras [17].

Como é usual em muitas outras API de serviços da Amazon, as imagens utilizadas para os serviços de deteção facial podem ser fornecidas pelo seu serviço de armazenamento de dados, Amazon S3 (*Amazon Simple Storage Service*) [18]. O serviço de deteção de caras deteta as 100 maiores caras na imagem e devolve as características de cada uma delas, tais como as coordenadas da *bounding box*, os traços faciais (localização dos olhos, boca, nariz, etc.), atributos faciais (idade, género, presença de pelos faciais, etc.), emoções e ainda a qualidade da imagem, incluindo métricas como a luminosidade e o contraste. É relevante referir que muitos dos atributos listados são acompanhados por um nível de confiança quanto à deteção dos mesmos. [19].

A comparação entre caras, esta será feita com recurso a uma imagem de exemplo (por exemplo, uma foto de perfil) e uma imagem para autenticação, fornecida no momento de autenticação. Ao fornecer as imagens de referência, caso estas contenham mais do que uma cara, a Amazon Rekognition escolhe, unicamente, a maior detetada, que será comparada com todas as caras na imagem fornecida para autenticação.

Para as caras em que há correspondências, é devolvida a informação da localização na imagem de cada uma delas, assim como um nível de confiança para cada uma (pode ser definido um nível de confiança mínimo para ser considerada uma correspondência) e a informação dos traços faciais. Para todas as caras na imagem para autenticação onde não há qualquer correspondência, são devolvidas as mesmas informações. Por fim, é incorporada informação sobre a imagem de referência para a identificação [20] [21].

2.5.3 Seleção da API

Para a seleção da API, foram consideradas as respetivas funcionalidades, apresentadas nas tabelas seguintes.

Funcionalidades de Detecção Facial	Azure Cognitive Services	Amazon Rekognition
Coordenadas para cada ponto facial	✓	✓
Coordenadas e dimensões da <i>bounding box</i> à volta da cara	✓	✓
Nível de confiança que a <i>bounding box</i> contem uma cara	×	✓
Idade da cara	em anos	alta ou baixa
Possível género da cara detetada	✓	✓
Presença de óculos na cara	✓	✓
Detetar se os olhos estão abertos	×	✓
Detetar se a boca está aberta	×	✓
Emoção demonstrada	✓	✓
Intensidade do sorriso	✓	✓
Presença de pelos faciais (barba, bigode)	✓	✓
Posição da cabeça	✓	✓
Visibilidade do cabelo	✓	×
Nível de confiança da calvície da cara detetada	✓	×
Cor do cabelo, caso detetável	✓	×
Presença de maquilhagem (olhos ou lábios)	✓	×
Presença de obstruções (testa, olhos ou boca)	✓	×
Presença de acessórios, o seu tipo, e nível de confiança	✓	×
Nível de desfocagem da foto	✓	×
Nível de exposição da foto	✓	✓
Nível de ruído da foto	✓	✓
Nível de confiança nos atributos detetados	×	✓

Tabela 2.1: Comparação de Funcionalidades de Detecção Facial

Apesar de o nível de confiança apresentar uma vantagem, por exemplo, para efeitos de testes, no contexto da comparação entre caras a API da Amazon Rekognition é um pouco mais limitada, na medida que as definições de grupos não existem. Assim, esta API apenas permite a comparação direta entre caras presentes em duas fotografias, enquanto os serviços da Azure permitem a criação de listas para identificar se uma dada cara pertence a um grupo [22] [23]. Esta limitação, no âmbito da aplicação a desenvolver, poderá ser uma desvantagem significativa, uma vez que os funcionários da empresa representam um grupo restrito, pelo que poderá ser relevante manter este conceito de grupos de utilizadores.

Um dos aspetos relevantes ainda a considerar para a seleção da API é o custo da sua utilização. Assim, seguem-se as informações de preços das API mencionadas na secção anterior. Primeiramente, é importante referir que a modalidade *standard* da *FaceAPI* dos serviços da Azure suporta até **10 transações por segundo** [24]. Nesta modalidade, os preços, que variam consoante a utilização dos recursos, são apresentados na tabela 2.2:

Volume de Imagens(imagens/mês)	Azure Cognitive Services - <i>FaceAPI</i>
0 - 1 Milhão de imagens	0,001 USD por imagem
1 - 5 Milhões de imagens	0,0008 USD por imagem
5 - 100 Milhões de imagens	0,0006 USD por imagem
Mais de 100 Milhões de imagens	0,0004 USD por imagem

Tabela 2.2: Preços da *FaceAPI* da Azure Cognitive Services

É importante considerar, ainda, que o custo do armazenamento dos dados de **cada cara corresponde a 0.00001 USD**, pagos mensalmente.

Os preços da API Amazon Rekognition, são apresentados na Tabela 2.3. É de realçar que as diferentes funcionalidades estão divididas em dois grupos, ou seja, certas funcionalidades constam no grupo de serviços mais caro. Para suporte das funcionalidades exigidas pelo projeto é necessária a utilização do grupo 1, cujos preços se encontram na seguinte tabela [25].

Volume de Imagens(imagens/mês)	Amazon Rekognition
0 - 1 Milhão de imagens	0,001 USD por imagem
1 - 5 Milhões de imagens	0,0008 USD por imagem
5 - 35 Milhões de imagens	0,0006 USD por imagem
Mais de 35 Milhões de imagens	0,0004 USD por imagem

Tabela 2.3: Preços da API Amazon Rekognition

Tal como na *FaceAPI* da Azure, o armazenamento de metadados das faces corresponde a **0.00001 USD por cara**, pagos mensalmente.

Há ainda a comparar os planos gratuitos de cada API. No caso da *FaceAPI* o plano gratuito limita-se a **20 transações por minuto** e oferece **30 mil transações gratuitas por mês**. No caso da Amazon Rekognition, estão disponíveis 5 mil transações gratuitas por mês, assim como o armazenamento de **metadados de 1000 faces por mês**.

Com tudo isto em mente, conclui-se que o custo da *FaceAPI* dos serviços Azure é favorável quando comparado ao custo da API da Amazon Rekognition. Uma vez que a utilização da *FaceAPI* para autenticação de elementos pertencentes a um grupo restrito é mais clara, esta revela-se, mais uma vez, a melhor escolha.

2.6 Produtos Existentes no Mercado

Para melhor compreender as contribuições deste projeto para o domínio da aplicação, foi realizada uma pesquisa por soluções semelhantes no mercado. Desta pesquisa resultou a análise de dois produtos de autenticação por reconhecimento facial, que consta na secção seguinte.

2.6.1 Veriff

Um dos serviços disponibilizados pela Veriff é a autenticação de utilizadores através de uma fotografia tirada pelos próprios no momento de autenticação. Esta é depois comparada com um registo pré-existente de uma cópia de um documento de identificação. Assim, quando o utilizador efetua uma operação que necessite de autenticação, o sistema pede que seja tirada uma foto no momento, empregando simultaneamente uma avaliação de *liveness* para determinar se a pessoa detetada está, realmente, presente e que não se trata de uma foto [26]. Na Figura 2.8 é ilustrado o resultado esperado da utilização do produto.

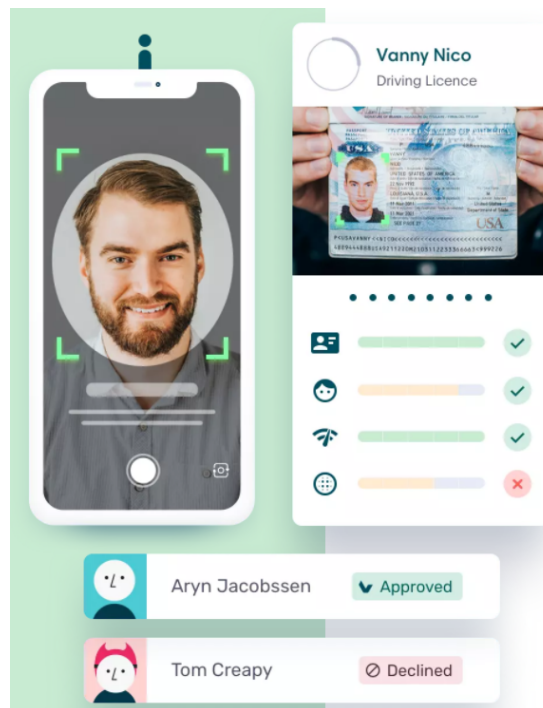


Figura 2.8: Simulação do uso da ferramenta Veriff [26].

2.6.2 Jumio

Uma outra alternativa são os serviços de reconhecimento Jumio [27]. Tal como a Veriff, é inserida uma cópia de um documento de identificação, que serve depois como referência para autenticar utilizadores através de uma foto. Mais uma vez, são utilizados processos de *liveness* para determinar que o rosto detetado não é de uma fotografia impressa, como é ilustrado na Figura 2.9.

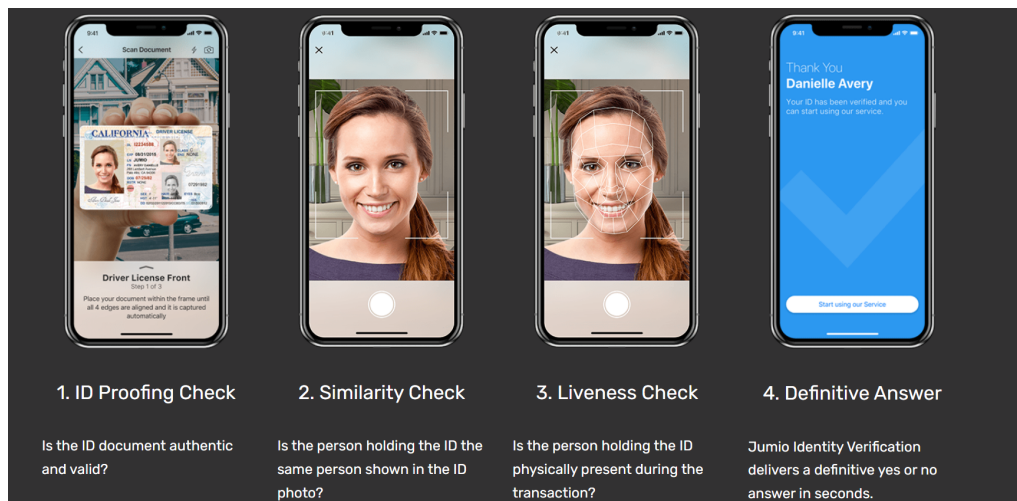


Figura 2.9: Etapas da utilização da ferramenta Jumio [27].

Dada a existência destes dois produtos, é possível concluir que as tecnologias utilizadas no desenvolvimento deste projeto não apresentam inovação no campo de reconhecimento facial. Contudo, em comparação com produtos como os apresentados, focados apenas na autenticação dos utilizadores, existe um fator de diferenciação, na medida em que são combinadas funcionalidades de reconhecimento facial com um *software* de gestão de serviços e funcionários de uma empresa. A inovação neste projeto está associada à integração da autenticação por reconhecimento facial neste tipo de *software*.

2.7 Soluções de Desenvolvimento *Low-code*

Segundo a Gartner, 70% das aplicações desenvolvidas por empresas em 2025 serão desenvolvidas com recurso a tecnologias *low-code* ou *no-code* [28]. Assim, é importante o reconhecimento do mercado destas tecnologias, de modo a compreender as alternativas existentes atualmente.

Na Figura 2.10 está representado o posicionamento das principais soluções *low-code* no mercado atual, segundo a Gartner.



Figura 2.10: Visão do mercado de plataformas *low-code*, segundo a Gartner [28].

A OutSystems, a Microsoft e a Mendix são consideradas as líderes do mercado. Esta classificação é devida à natureza inovadora das suas soluções, assim como a uma forte presença no mercado.

No caso da OutSystems, os aspetos salientados pela Gartner são a forte aposta na acessibilidade da sua interface e o foco na experiência do utilizador, assim como a inovação na área de automatização de testes com recurso a inteligência artificial e de gestão de APIs. Os pontos negativos estão relacionados com os modelos de faturação e baixa flexibilidade nos contratos.

Quanto à Microsoft, as principais vantagens consistem na quantidade de ferramentas complementares, que correspondem a serviços já disponibilizados pela Microsoft, como por exemplo a Azure, o Visual Studio e GitHub.

Finalmente, quanto à Mendix, as vantagens apontadas prendem-se com a inovação da solução, sobretudo na área da Internet of Things, assim como, de forma semelhante à OutSystems, a uma aposta na usabilidade da sua plataforma. Os pontos negativos consistem na sua estratégia de marketing e dificuldades na sua expansão em territórios fora da Europa e América do Norte.

Também a análise da Forrester [29] destaca estas 3 plataformas como líderes do mercado, como evidenciado na Figura 2.11.



Figura 2.11: Visão do mercado de plataformas *low-code*, segundo a Forrester [29].

Esta classificação tem por base as funcionalidades atualmente disponíveis, a sua estratégia de negócio e a sua presença no mercado, comparadas entre as 3 na Tabela 2.4.

	Peso atribuído	Mandix	Microsoft	OutSystems
Oferta atual	50%	4.40	3.99	4.18
Ferramentas de para dados e integrações	30%	5.00	5.00	5.00
Ferramentas para automatização de processos	20%	3.30	3.60	3.30
Ferramentas para experiência do utilizador	20%	5.00	5.00	5.00
Ferramentas para testes, <i>deployment</i> e gestão de aplicações	15%	4.50	1.70	3.70
Ferramentas para governo da plataforma e aplicações	10%	4.00	3.00	3.00
Ferramentas para relatórios, extensões e funcionalidades	5%	3.20	2.80	3.20
Estratégia	50%	4.50	4.50	4.50
Visão do produto	20%	5.00	5.00	5.00
Plano de inovação	20%	5.00	3.00	5.00
Ecosistema de parcerias	20%	3.00	5.00	3.00
Estratégia de marketing	25%	5.00	5.00	5.00
Melhoramentos planeados	10%	5.00	5.00	5.00
Modelo comercial	5%	3.00	3.00	3.00
Presença no mercado	0%	2.50	4.50	3.00
Receitas da plataforma de <i>low-code</i>	50%	3.00	4.00	3.00
Número de clientes	50%	2.00	5.00	3.00

Tabela 2.4: Comparação entre os líderes do mercado *low-code*, adaptada de Forrester [29].

A realização deste estado da arte permitiu a tomada de uma decisão quanto às tecnologias de reconhecimento facial a utilizar no decorrer do projeto, assim como uma visão sobre as oportunidades que existem no domínio de aplicação, o que permitiu a definição de requisitos mais específicos da solução, descritos no capítulo seguinte.

Esta página foi intencionalmente deixada em branco

Capítulo 3

Análise dos Requisitos

Uma vez que a proposta do estágio não considerava um cliente específico, foram definidos inicialmente requisitos gerais. Assim, o produto a desenvolver deveria ser uma aplicação móvel com os seguintes atributos gerais:

- Autenticação na aplicação para acesso ao calendário dos serviços agendados;
- Notificação do cliente do início de deslocação ao local de prestação do serviço por parte do funcionário;
- *Check-in* no local (*outdoor*) por identificação facial do técnico que presta o serviço;
- Para que a identificação seja mais fiável e segura, a localização também deve ser confirmada por GPS;
- *Check-out* no local após a conclusão do serviço;
- Envio de questionário de satisfação para o cliente.

Da revisão bibliográfica e pesquisa de mercado apresentadas no capítulo anterior, foi possível definir requisitos mais específicos para a aplicação, de modo a satisfazer as oportunidades identificadas no domínio da aplicação.

À medida que foram elaborados os casos de uso e protótipos para a aplicação foram surgindo outros requisitos, quer para conciliar outras funcionalidades, quer por representarem opções que poderiam dar apoio ao funcionamento da aplicação, sendo estes, no geral, facultativos e, por isso, menos prioritários.

Neste capítulo são apresentados os requisitos da aplicação, incluindo esclarecimentos pontuais de algumas decisões tomadas na sua implementação, assim como protótipos da aplicação elaborados para clarificar o fluxo de ações da mesma.

3.1 Descrição de Atores Principais

Primeiramente, foi necessário definir quais os atores intervenientes na aplicação. O ator principal mais evidente é o Funcionário, dado que a maioria dos requisitos se prende com este. De seguida, ao analisar os requisitos referentes à administração da plataforma (como por exemplo a criação de serviços), tornou-se evidente que seria necessário um perfil de Supervisor para gestão dos funcionários e dos serviços a desempenhar. Posteriormente,

para simplificar a integração do cliente neste processo, tomou-se a decisão de também o incluir como ator primário, de modo a simplificar processos como o preenchimento do questionário de satisfação. Assim, os atores primários definidos foram os seguintes:

- Supervisor - Responsável pela gestão de funcionários e serviços;
- Funcionário - Prestador de serviços;
- Cliente - Solicitador de serviços.

3.2 Requisitos Funcionais

Para a classificação da prioridades dos requisitos funcionais foram considerados 3 níveis de prioridades:

- **Elevado** - Representam as funcionalidades de caráter prioritário, quer por serem funcionalidades indispensáveis ao funcionamento da aplicação, quer por servirem de base a outras funcionalidades que o sejam. São requisitos obrigatórios e essenciais para o sucesso do projeto;
- **Intermédio** - Funcionalidades que devem ser implementadas, ainda que numa fase mais avançada do projeto. Assim, representam tarefas que podem ser interrompidas e retomadas noutra altura caso surja um imprevisto na execução do projeto;
- **Reduzido** - Prioridade mais reduzida, atribuída a funcionalidades que só serão implementadas se existirem recursos suficientes. A não implementação destes requisitos não afeta o sucesso do projeto.

3.2.1 Comuns a todos os utilizadores

- **Login**
 - **Descrição:** Para aceder à plataforma, todos os utilizadores (funcionários, supervisor e clientes) têm de se autenticar para poder usufruir das suas funcionalidades;
 - **Prioridade:** Elevada;
 - **Cenário de Sucesso:**
 1. O utilizador acede à página de login. O sistema pede para ser realizado o login com as suas credenciais;
 2. O utilizador introduz as credenciais de acesso, email e password. O utilizador é redirecionado para a sua página pessoal caso as credenciais estejam corretas, caso contrário é levantado um alerta.
- **Ver notificações.**
 - **Descrição:** Os utilizadores pretendem ver notificações referentes a vários aspetos do funcionamento da aplicação;
 - **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O utilizador seleciona o ícone das notificações e estas são apresentadas.

• **Sincronização com o *Google Calendar*.**

– **Descrição:** A plataforma poderá permitir ao utilizador integrar o calendário pessoal de cada funcionário com o Google Calendar, de forma que tenha acesso à calendarização dos seus serviços mais facilmente;

– **Prioridade:** Reduzida;

– **Cenário de Sucesso:**

1. O utilizador abre a sua página inicial, onde consta o seu calendário;
2. O utilizador escolhe a opção de integrar o calendário com o Google Calendar. O sistema gera um link para integrar com o Google Calendar;
3. O utilizador insere o link no Google Calendar. O calendário visível na app está de acordo com o seu Google Calendar.

• **Editar perfil.**

– **Descrição:** Caso o utilizador queira editar as informações pessoais ou de acesso à plataforma, deve escolher a respetiva opção. Ao preencher o formulário com os dados pedidos, recebe via email a confirmação da alteração dos dados na plataforma, que são atualizados na base de dados;

– **Prioridade:** Reduzida;

– **Cenário de Sucesso:**

1. O utilizador seleciona a opção para editar as suas informações e são apresentadas as informações editáveis;
2. O utilizador altera a informação que pretende e seleciona a opção “Guardar”. Uma mensagem de confirmação aparece no ecrã e é enviado um e-mail.

3.2.2 Supervisor

• **Visualizar informações referentes aos serviços criados.**

– **Descrição:** O supervisor pode ver as várias informações sobre um serviço, incluindo o estado e o funcionário a quem está atribuído, e no caso de estar concluído, o relatório e avaliação de qualidade feita pelo cliente;

– **Prioridade:** Reduzida;

– **Cenário de Sucesso:**

1. O supervisor seleciona o serviço que pretende e são apresentadas as informações sobre o serviço (local, horas, cliente), e caso o serviço já se encontre atribuído a um funcionário, esta informação também é visível. No caso de estar concluído, o relatório e a avaliação de qualidade feita pelo cliente também podem ser consultadas.

• **Visualizar funcionários na empresa.**

– **Descrição:** O supervisor tem acesso à listagem dos funcionários na empresa, assim como às suas informações;

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O supervisor seleciona a opção “Gerir Funcionários” e são apresentados os funcionários associados à empresa;
2. O supervisor introduz a informação para pesquisa do funcionário (nome, email). Caso haja correspondências para a informação introduzida, o perfil dos funcionários aparece;
3. O supervisor seleciona o funcionário cujas informações pretende consultar. São apresentadas as informações referentes ao funcionário selecionado;

• **Adicionar funcionários à empresa.**

– **Descrição:** O supervisor pode associar um funcionário à empresa, para que este possa visualizar os serviços que devem ser prestados;

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O supervisor seleciona a opção “Adicionar Funcionário”. O sistema pede para ser realizado o registo, preenchendo os campos requeridos;
2. Após preencher as informações, o supervisor seleciona a opção “Concluir Registo” e é devolvida uma mensagem de sucesso. O supervisor é levado para a página inicial.

• **Remover funcionários da empresa.**

– **Descrição:** Caso o funcionário deixe de trabalhar para a empresa, o supervisor deve salvar a segurança da aplicação, negando acesso à plataforma a funcionários que já não se encontrem ao serviço da empresa;

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O supervisor seleciona a opção “Remover Funcionário” e é apresentada uma barra de pesquisa;
2. O supervisor seleciona a opção “Gerir Funcionários” e são apresentados os funcionários associados à empresa;
3. O supervisor seleciona o funcionário que pretende e surge o perfil do trabalhador com as informações referentes ao mesmo;
4. O supervisor seleciona a opção “Remover”. O supervisor recebe uma mensagem de sucesso e o funcionário é removido da lista de funcionários da empresa.

• **Criar serviço.**

– **Descrição:** O supervisor cria um serviço consoante as necessidades de um cliente ao preencher as informações do serviço para que este seja adicionado ao sistema. Estas informações são adquiridas através de um contacto externo à aplicação (por exemplo, e-mail ou telefone). Alternativamente, o cliente pode fazer um pedido diretamente na plataforma.

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O supervisor seleciona a opção “Criar Serviço” na opção principal. É apresentado um formulário para introdução das informações referentes ao serviço;

2. O supervisor preenche o formulário e seleciona a opção “Guardar”. Uma mensagem de confirmação aparece no ecrã e o serviço é adicionado ao sistema.

- **Validar pedido de serviço.**

- **Descrição:** O supervisor deve validar o pedido de serviço efetuado pelo cliente, atribuindo assim, um funcionário para o desempenhar. Neste caso, as informações referentes ao serviço são definidas automaticamente, consoante o especificado pelo cliente;
- **Prioridade:** Elevada;
- **Cenário de Sucesso:**
 1. O supervisor seleciona um pedido da secção “Pedidos por Verificar” e surgem as informações do pedido introduzidas pelo cliente, assim como um campo para preencher a data e o funcionário prestador de serviço;
 2. O supervisor preenche a data da prestação do serviço e seleciona um funcionário da lista dos que estão disponíveis na data, e seleciona a opção “Validar”. É apresentada uma mensagem de sucesso e é enviada uma notificação ao cliente. O serviço torna-se visível para o funcionário.

- **Rejeitar pedido de serviço.**

- **Descrição:** Caso seja impossível prestar o serviço pedido pelo cliente, o supervisor deve rejeitá-lo, especificando o motivo (por exemplo, indisponibilidade dos funcionários ou falta de competências técnicas) e sugerir uma nova data;
- **Prioridade:** Intermédia;
- **Cenário de Sucesso:**
 1. O supervisor seleciona um pedido da secção “Pedidos por Verificar”, sendo apresentada as informações do pedido introduzidas pelo cliente, assim como um campo para preencher a data e o empregado prestador de serviço;
 2. O supervisor seleciona a opção “Rejeitar”. Num novo ecrã surge uma *check-list* de motivos para justificar a rejeição da proposta, assim como um campo de proposta de nova data para o serviço;
 3. O supervisor preenche o questionário e seleciona a opção “Rejeitar” e o pedido é retirado da secção “Pedidos por Verificar”, acompanhado de uma mensagem de sucesso. O cliente recebe uma notificação sobre a rejeição do seu pedido.

- **Autenticação manual dos funcionários.**

- **Descrição:** Se um funcionário não conseguir autenticar-se, será enviada uma notificação ao supervisor para que entre em contacto com o mesmo. De seguida, poderá, assim, validar a autenticação do funcionário;
- **Prioridade:** Intermédia;
- **Cenário de Sucesso:**
 1. O supervisor seleciona a opção “Autenticação Manual” e os pedidos são apresentados;
 2. O supervisor seleciona a opção “Autenticar”. O estado do serviço é alterado, acompanhado com uma mensagem de sucesso.

3.2.3 Funcionário

- **Início de deslocação.**

- **Descrição:** Ao selecionar um serviço no calendário, o funcionário seleciona a opção para iniciar a deslocação para o referido serviço. Esta ação gera uma notificação, que é enviada ao cliente;
- **Prioridade:** Intermédia;
- **Cenário de Sucesso:**
 1. O funcionário seleciona o serviço que vai prestar e são apresentadas as informações sobre o serviço (local, horas, cliente), bem como uma opção para “Iniciar Deslocação”;
 2. O funcionário seleciona a opção “Iniciar Deslocação”. Uma mensagem de confirmação aparece no ecrã e a notificação é enviada ao cliente;
 3. Às informações do serviço é adicionado um mapa com um trajeto desde a localização do funcionário até ao destino.

- **Check-In no local de serviço.**

- **Descrição:** Ao chegar ao local do serviço, o funcionário identifica-se com recurso a identificação facial e a localização GPS;
- **Prioridade:** Elevada;
- **Cenário de Sucesso:**
 1. O funcionário seleciona o serviço que já iniciou e são apresentadas as informações sobre o serviço (local, horas, cliente), bem como uma opção para “Check-In”;
 2. O funcionário escolhe a opção “Check-In”. É apresentado o ecrã que utiliza a câmara para a realização da identificação facial;
 3. O funcionário segue as instruções de utilização da câmara. O sistema fornece uma mensagem para indicar o sucesso do reconhecimento facial;
 4. O funcionário é redirecionado para o ecrã de preenchimento do relatório.

- **Autenticação por código pin.**

- **Descrição:** O funcionário não conseguiu autenticar-se com o reconhecimento facial ao fim de, pelo menos, 5 tentativas, pelo que pode alternativamente introduzir um código pin, definido no momento de registo;
- **Prioridade:** Intermédia;
- **Cenário de Sucesso:**
 1. O funcionário seleciona a opção “Autenticação por Pin Code” e é apresentado um teclado numérico para introdução do pin, assim como um contador para as tentativas restantes que o funcionário tem;
 2. O funcionário digita o seu código. Uma mensagem de confirmação aparece no ecrã e o funcionário é autenticado.

- **Preencher Relatório.**

- **Descrição:** Ao longo do serviço, o funcionário pode preencher o relatório com informações sobre o material usado e as tarefas desempenhadas;
- **Prioridade:** Reduzida;

– **Cenário de Sucesso:**

1. O funcionário navega para o ecrã de preenchimento de relatório, onde pode preencher as informações relativas à prestação do serviço.

• **Terminar Serviço.**

– **Descrição:** Ao terminar o preenchimento do relatório, o funcionário seleciona a opção de “Submeter” e é lhe apresentado o relatório final, onde pode conferir as informações e recolher a assinatura do cliente;

– **Prioridade:** Intermédia;

– **Cenário de Sucesso:**

1. O funcionário seleciona a opção de “Submeter” depois de preenchido o relatório. São apresentadas as informações sobre o serviço (local, horas, cliente), bem como uma opção para “Terminar Serviço”;
2. O funcionário seleciona o campo de assinatura, permitindo ao cliente que o preencha;
3. O funcionário seleciona a opção “Terminar Serviço”. Surge uma mensagem de sucesso e são enviadas notificações para o cliente e o supervisor. As informações que constam no relatório passam a ser visíveis nos detalhes dos serviços concluídos e é disponibilizado o inquérito de qualidade para o cliente.

• **Concluir Cadastro**

– **Descrição:** Após o registo inicial do funcionário por parte do supervisor, no primeiro acesso do funcionário à plataforma este deve proceder ao preenchimento das informações necessárias para a utilização da solução (foto de perfil e definição de pin);

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O funcionário recebe um e-mail com um link para a aplicação;
2. Ao abrir o link, é lhe pedido que defina a sua *password*. De seguida, seleciona a opção “Seguinte”.
3. O sistema requer ao funcionário que preencha informações referentes à autenticação (foto de perfil e definição de pin);
4. O funcionário segue as instruções de preenchimento. O sistema fornece uma mensagem consoante as etapas sejam bem sucedidas ou não;
5. O funcionário é redirecionado para a sua página inicial, pronto a utilizar a aplicação.

3.2.4 Cliente

• **Registo**

– **Descrição:** Para aceder à plataforma, os clientes têm que se registar para poder fazer login, através do preenchimento de um formulário de adesão;

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O cliente acede à página de registo. O sistema pede para ser realizado o registo, preenchendo os campos requeridos: nome, *password*, e-mail e morada;
2. O cliente preenche o formulário de registo na plataforma e confirma. O sistema verifica se todos os campos estão corretamente preenchidos em caso afirmativo surge uma mensagem de sucesso. A página inicial é apresentada ao cliente, que recebe um e-mail a confirmar as suas informações.

• **Preenchimento do inquérito de qualidade.**

– **Descrição:** De modo a facilitar o processo de recolha de informações por parte da empresa, quando um serviço é concluído é enviada ao cliente uma notificação que, ao expandir, dá início ao preenchimento do inquérito de qualidade;

– **Prioridade:** Reduzida;

– **Cenário de Sucesso:**

1. O cliente seleciona a opção “Avaliações” e são apresentados todos os inquéritos de qualidade dos seus serviços;
2. O cliente seleciona um inquérito por preencher e surge o formulário no ecrã;
3. O cliente atribui as classificações pedidas e seleciona a opção “Enviar”. Surge uma mensagem de sucesso e é enviada uma notificação ao supervisor. O inquérito torna-se visível nos detalhes do serviço.

• **Pedido de criação de serviço.**

– **Descrição:** O cliente pode comunicar à empresa que necessita da prestação de um serviço diretamente pela plataforma;

– **Prioridade:** Elevada;

– **Cenário de Sucesso:**

1. O cliente seleciona a opção “Pedir Serviço” e é-lhe apresentado o ecrã de pedidos de serviço, que contém um formulário a preencher;
2. O cliente preenche as informações necessárias para registo do pedido e seleciona a opção “Criar Pedido”. Surge uma mensagem de sucesso e o pedido fica visível na plataforma, sendo ainda enviada uma notificação ao supervisor. O serviço é acrescentado ao separador de pedidos por validar do supervisor.

Ao longo do projeto, estes casos de uso foram complementados por outros que surgiram do desenvolvimento de funcionalidades, dos quais se destacam os seguintes.

3.2.5 Comuns a todos os utilizadores

• **Editar e-mail do utilizador.**

– **Descrição:** Por ser uma informação particularmente sensível, a edição do e-mail é feita num ecrã para este fim.;

– **Prioridade:** Reduzida;

– **Cenário de Sucesso:**

1. O utilizador navega para o ecrã de definições do seu perfil e seleciona a opção “Editar E-mail”. É apresentado o e-mail atual e um campo para introdução do novo e-mail;
2. Depois de preencher o campo para o novo e-mail, o utilizador seleciona a opção “Alterar E-mail”. Surge um *pop-up* a pedir a confirmação da *password*;
3. O utilizador introduz a sua *password* e seleciona a opção “Confirmar”. Surge uma mensagem de sucesso e o utilizador recebe um e-mail na nova caixa de correio a confirmar a alteração das suas informações pessoais.

3.2.6 Supervisor

- **Restaurar acesso de um funcionário bloqueado.**

- **Descrição:** O supervisor deve restaurar acesso a um funcionário que tenha sido bloqueado por introduzir um código incorreto 3 vezes;
- **Prioridade:** Reduzida;
- **Cenário de Sucesso:**
 1. O supervisor navega para o ecrã “Gerir Funcionários”;
 2. O supervisor seleciona o separador “Bloqueados”. Aqui, são apresentados os funcionários que se encontram bloqueados, podendo filtrar por nome caso não se encontre imediatamente apresentado;
 3. O supervisor seleciona a opção “Restaurar Acesso”. É apresentada uma mensagem de sucesso e o funcionário é removido da lista de funcionários bloqueados.

3.2.7 Funcionário

- **Suspender atividade do funcionário.**

- **Descrição:** O funcionário pretende ausentar-se do serviço, por exemplo, gozar férias. Assim, não deve ser sugerido o seu nome para serviços a desempenhar enquanto estiver ausente;
- **Prioridade:** Reduzida;
- **Cenário de Sucesso:**
 1. O funcionário navega para o ecrã de definições do seu perfil;
 2. É selecionada a opção “Suspender Atividade”. Surge um aviso a comunicar ao funcionário que o seu acesso será restaurado da próxima vez que iniciar sessão na aplicação;
 3. O funcionário confirma a sua intenção de suspender a sua atividade e a sua sessão na aplicação é terminada.

3.2.8 Cliente

- **Consultar índices de desempenho.**

- **Descrição:** O supervisor pretende ter uma visão mais ampla do desempenho dos funcionários;

- **Prioridade:** Reduzida;
- **Cenário de Sucesso:**
 1. O supervisor navega para o ecrã “KPI”. São apresentadas estatísticas dos serviços dos últimos 7 dias;
 2. O supervisor seleciona o intervalo que pretende consultar, alterando a data de início e de fim. As informações apresentadas são atualizadas (média de classificações dos serviços, distância percorrida pelos funcionários, serviços concluídos, serviços iniciados a horas e com atrasos e percentagem de tempo gasto em deslocações);
 3. O supervisor seleciona um dos funcionários apresentado no mapa geral. Surge um *pop-up* com as suas informações mais básicas (nome e foto).

3.3 Diagramas de Casos de Uso

Conforme os requisitos identificados anteriormente, foram desenvolvidos diagramas de casos de uso de forma a evidenciar as interdependências entre os atores identificados. Deste modo, nas Figuras 3.2 a 3.3 constam os diagramas de casos de uso do supervisor, funcionário e cliente respetivamente. Finalmente, a Figura 3.4 ilustra a visão conjunta dos diagramas de casos de uso para todos os atores.

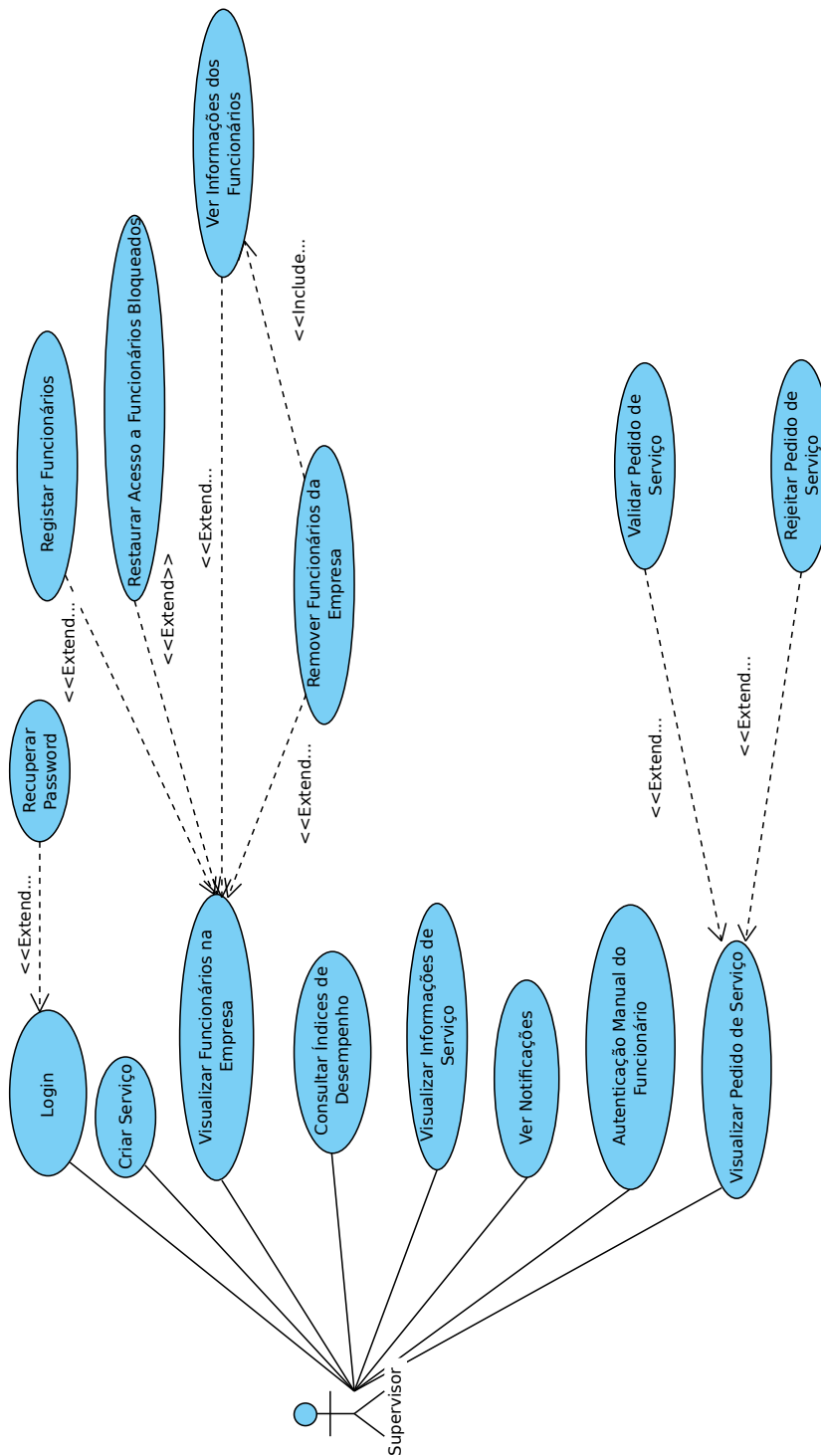


Figura 3.1: Diagrama de Casos de Uso do Supervisor

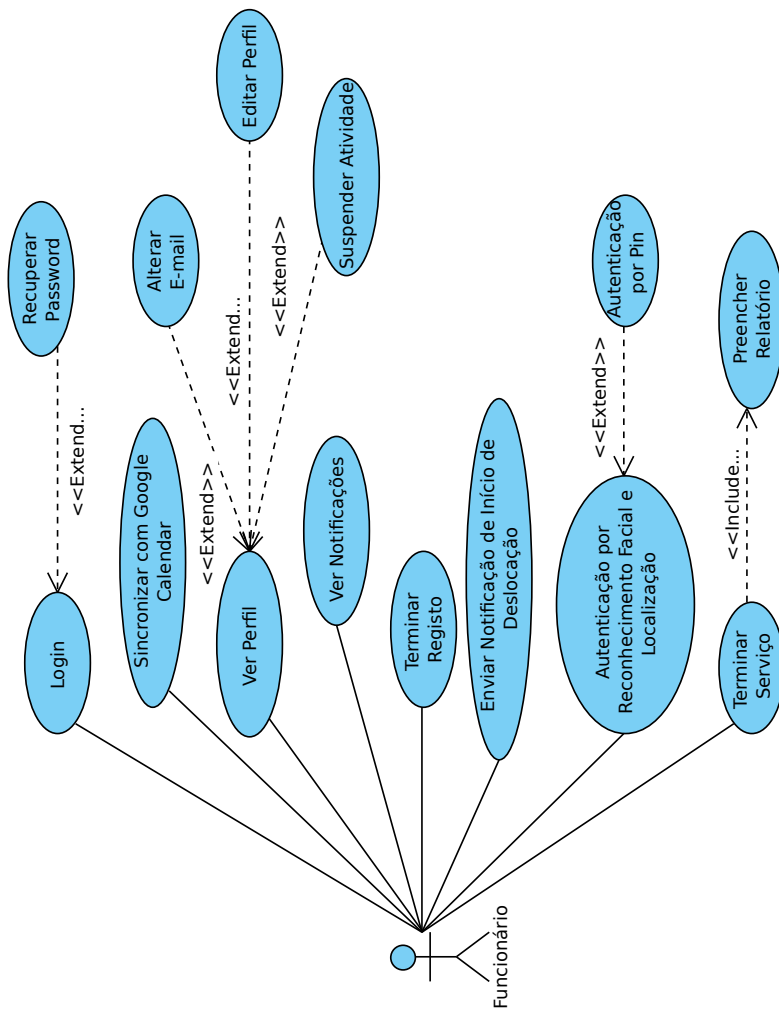


Figura 3.2: Diagrama de Casos de Uso do Funcionário

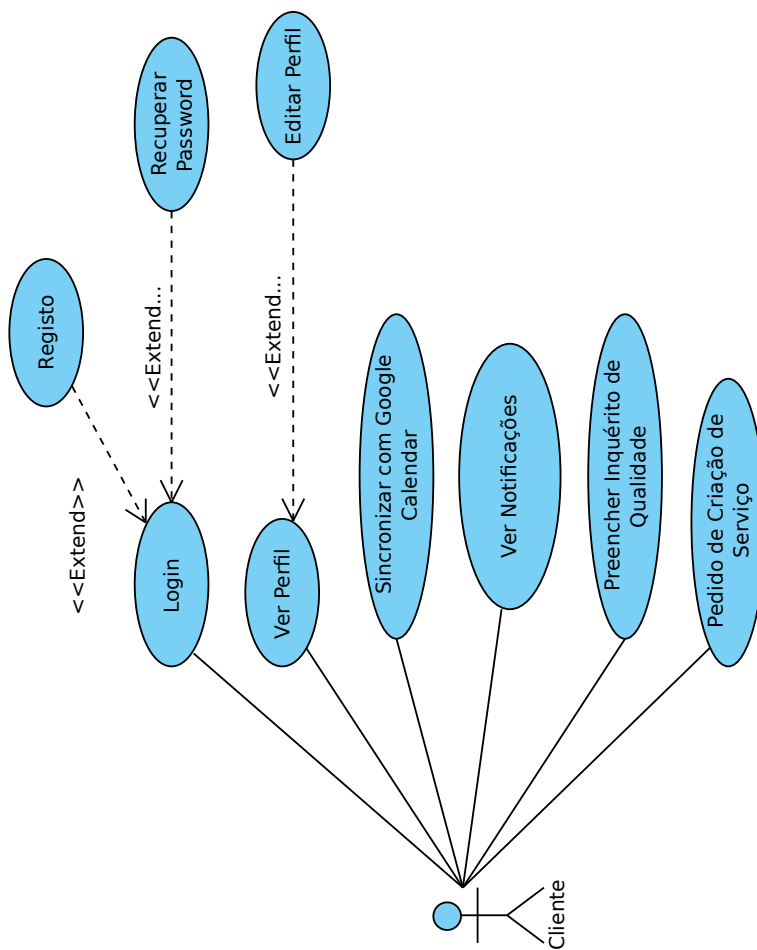


Figura 3.3: Diagrama de Casos de Uso do Cliente

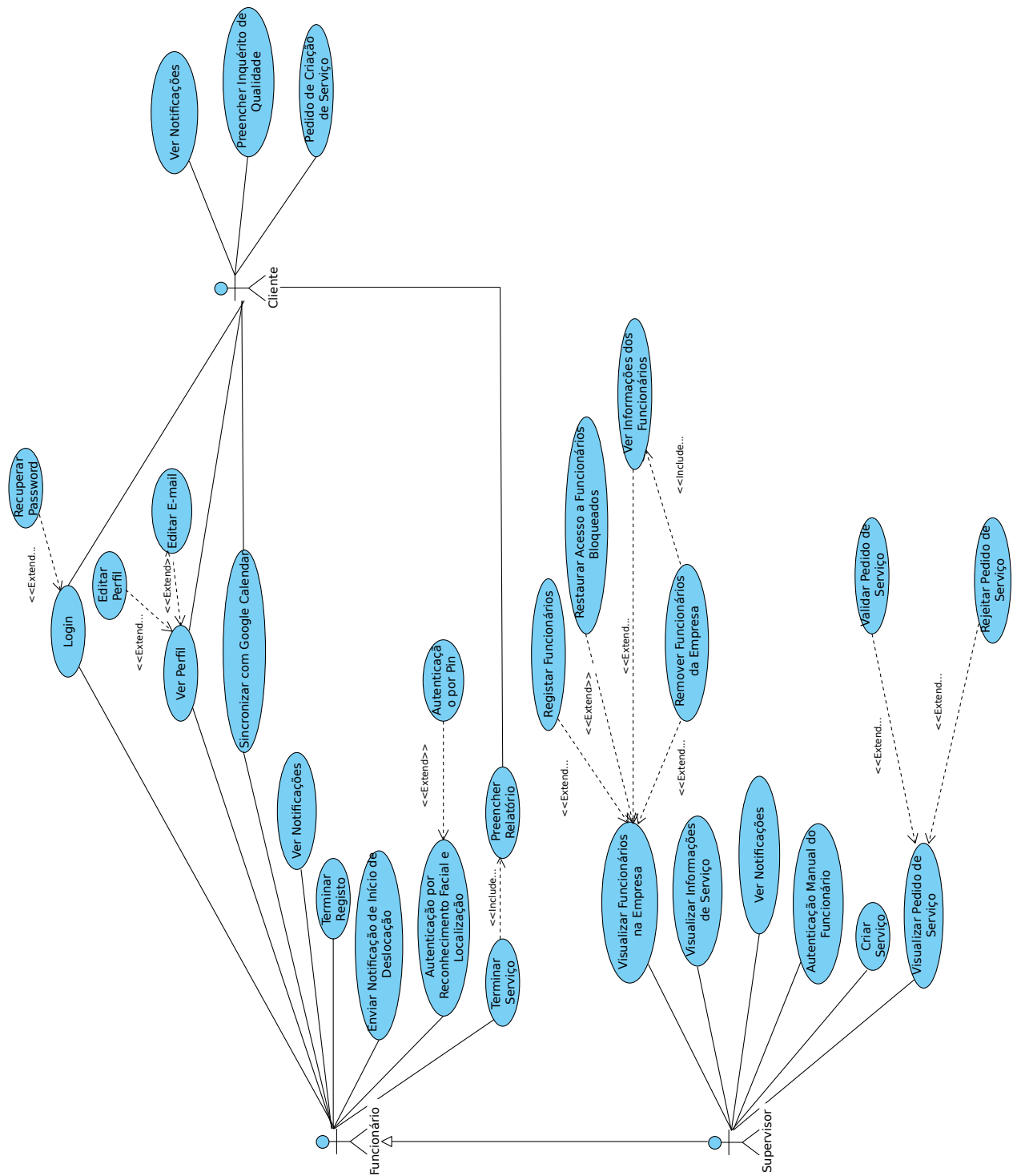


Figura 3.4: Diagrama Completo dos Casos de Uso

3.4 Mockups

Por fim, com base nos requisitos apresentados nas secções anteriores, foram desenvolvidos protótipos rápidos dos ecrãs da solução. De todos os protótipos, apresentados na secção Apêndices A, salientam-se os mais relevantes, representados nas Figuras 3.5 a 3.8.



Figura 3.5: Da esquerda para a direita: Página Inicial do Supervisor, do Funcionário e do Cliente.

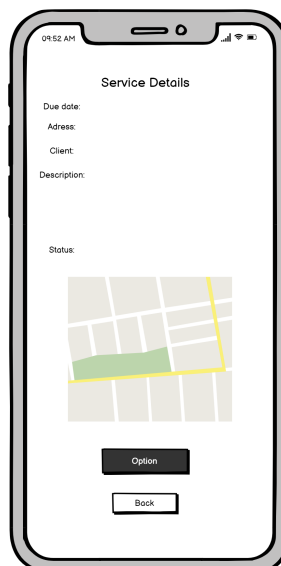


Figura 3.6: Apresentação dos detalhes de um serviço a um funcionário que já se encontra no local. Selecionar a opção “Check-In” inicia o processo de autenticação por reconhecimento facial.

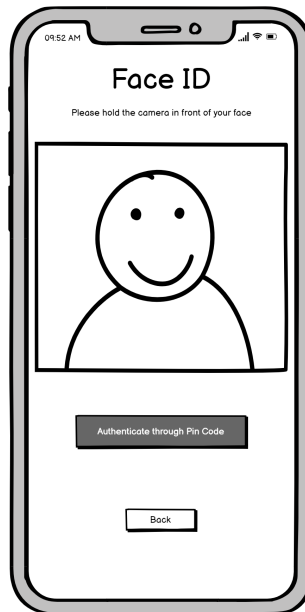


Figura 3.7: Ecrã com acesso a câmara para realização do reconhecimento facial. Caso a autenticação falhe pelo menos 5 vezes, surge a opção de autenticação com recurso a um código pin.

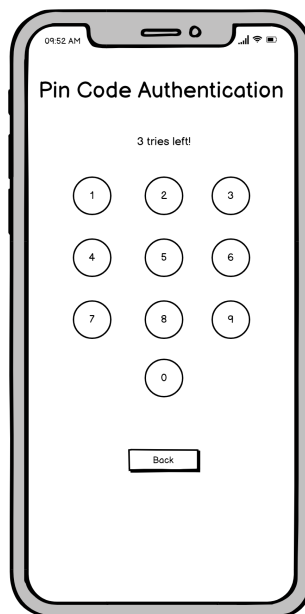


Figura 3.8: O utilizador introduz o pin definido no momento de registo na plataforma. Caso não introduza o pin correto ao fim de 3 tentativas, o supervisor será notificado para que entre em contacto com o funcionário.

Tendo definido os requisitos funcionais a implementar, deu-se início à implementação da solução, descrita no capítulo seguinte.

Capítulo 4

Desenvolvimento da Solução

Neste capítulo está exposto o processo de desenvolvimento da aplicação, dividido em duas secções principais. A primeira secção corresponde ao planeamento da arquitetura, que foi realizado com base no modelo definido para projetos desenvolvidos na plataforma *low-code* da OutSystems. Ainda nesta secção é exposto o modelo de dados utilizado para relacionar as entidades identificadas. A segunda secção corresponde à implementação das funcionalidades na plataforma da OutSystems, assim como uma visão sobre o funcionamento da mesma. É ainda abordada a integração dos vários *plugins* utilizados na solução.

4.1 Arquitetura da Solução

4.1.1 Componentes de Aplicações OutSystems

As aplicações desenvolvidas na plataforma da OutSystems são constituídas por alguns componentes seguindo a mesma estrutura. Em primeiro lugar, o *Platform Server* integra vários serviços, incluindo o ambiente de desenvolvimento da lógica referente à aplicação, ou seja, são programadas as ações a desempenhar nos servidores, assim como toda a lógica de *front-end* (*Widgets*, ecrãs, etc.). Associada a este componente está uma base de dados onde são guardadas as várias versões publicadas ao longo do desenvolvimento da aplicação.

Quando a aplicação é publicada, é gerado o código otimizado que reflete as funcionalidades implementadas no *Platform Server*. Este é incorporado no *Application Server*, que tem recurso a dados da aplicação guardados em base de dados, como por exemplo, informações sobre utilizadores. Existe também um componente de *External Systems* para tratar das integrações de tecnologias utilizadas na aplicação desenvolvida.

Assim, a Figura 4.1 representa de forma esquemática a interação entre os componentes descritos anteriormente.

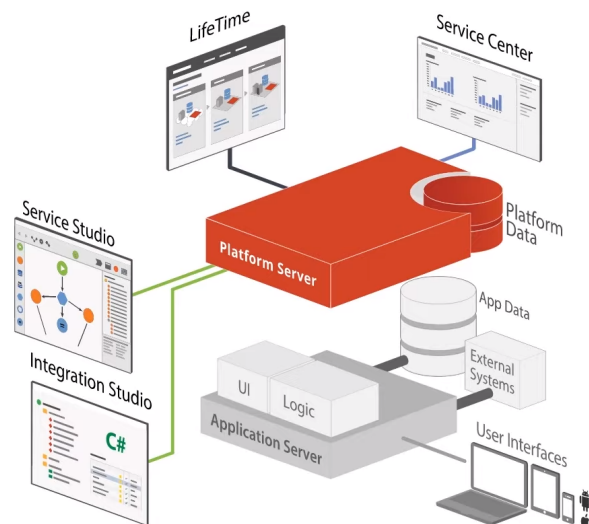


Figura 4.1: Componentes de um aplicação desenvolvida com a OutSystems.

4.1.2 Planeamento da Arquitetura

Para desenvolver aplicações com recurso à plataforma *low-code*, a OutSystems propõem que a sua arquitetura siga o modelo da *Architecture Canvas* [30], representada na Figura 4.2. Deste modo, na fase inicial do projeto, foi definida uma arquitetura para o mesmo, segundo este modelo.

A arquitetura é constituída por 3 camadas: *End-User*, *Core* e *Foundation*. A primeira camada contém as interfaces e processos que providenciam funcionalidades aos utilizadores. Por outro lado, a camada *Core* contém as entidades e conceitos de negócio usados na camada anterior, ou seja, a representação dos intervenientes nas interfaces é feita nesta camada. Por fim, a camada *Foundation* estabelece ligações a serviços externos à aplicação. Cada uma destas camadas deve ser claramente definida, pelo que os vários componentes da aplicação a desenvolver devem ser organizados por cada uma delas.

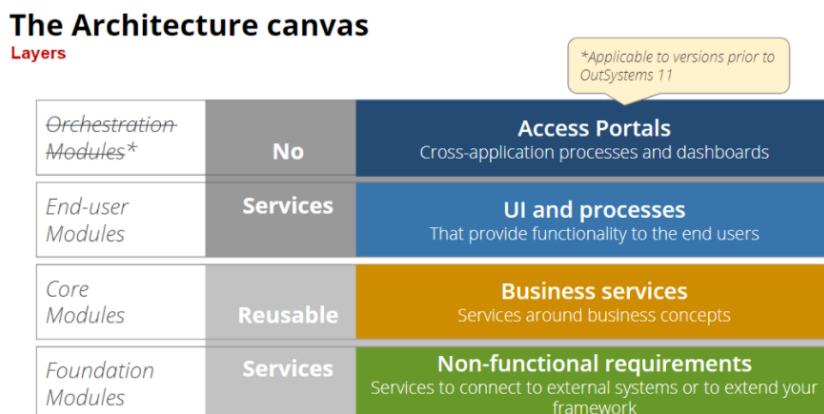


Figura 4.2: Visão da *Architecture Canvas* e das suas camadas. Note-se que, como referido na imagem, a camada superior não é utilizada em projetos desenvolvidos na versão mais recente da plataforma.

Além destas considerações, existem regras que devem ser cumpridas quanto à disposição

de elementos na *Architecture Canvas*, a fim de simplificar a manutenção e *deployment* das aplicações, das quais se destacam as seguintes [31]:

- Módulos das camadas inferiores não podem apresentar dependências de módulos de camadas superiores;
- Módulos na camada *End-User* não podem apresentar dependências de módulos na mesma camada;
- Dois módulos não podem ser dependentes um do outro (dependências circulares);
- Os módulos na camada *Core* não podem ter ecrãs de *front-end*, e as entidades definidas nos mesmos devem ser exclusivamente de leitura (*Read-Only*);
- Os módulos na camada *Foundation* não devem apresentar lógica referente as entidades definidas na camada *Core*.

Assim, estas regras podem ser resumidas na Figura 4.3.

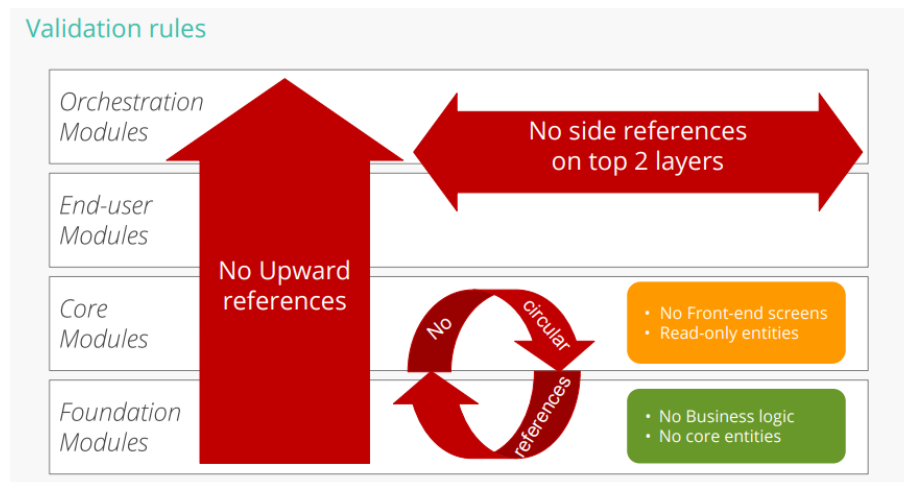


Figura 4.3: Regras da definição de arquitetura na *Architecture Canvas*.

O processo de definição da arquitetura é constituído por 3 fases, sendo elas a fase de *Disclose*, em que é feito um levantamento dos conceitos e necessidades da aplicação (que tipos de utilizadores e funções deve suportar), a fase de *Organize*, onde os conceitos identificados são organizados nas camadas correspondentes, e a fase de *Assemble*, onde os conceitos identificados nas fases anteriores são agrupados em módulos.

Na fase de *Disclose e Organize*, cujo resultado final se encontra representado na Figura 4.4, foram identificadas as interfaces a implementar, nomeadamente as várias interações que envolvem serviços, assim como o processo de reconhecimento facial. Para cada um destes processos, foram identificadas as entidades envolvidas (Clientes, Funcionários e Supervisores), assim como dois módulos extra, um para coordenação dos três tipos de utilizadores e um designado para o reconhecimento facial. Finalmente, foram identificados os serviços externos à aplicação que deveriam ser integrados para o desenvolvimento da mesma, nomeadamente serviços de notificações, localização, calendarização, temas da aplicação (integram vários aspetos visuais da aplicação) e integração da API de reconhecimento facial. Todos os componentes identificados foram colocados na camada da correspondente da *Architecture Canvas*.

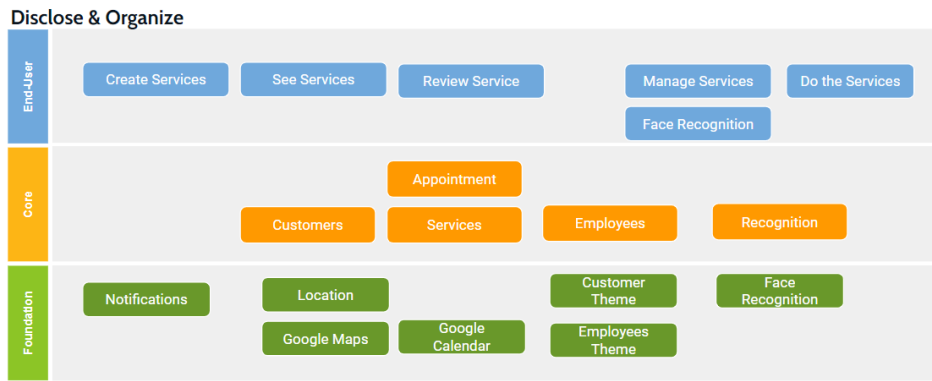


Figura 4.4: Resultado final das fases de *Disclose* e *Organize*.

A fase seguinte do processo de definição da arquitetura passa por agrupar os conceitos identificados anteriormente em módulos que contêm a lógica referente a cada conceito. Assim, as interfaces na camada superior são agrupadas em duas vertentes, uma para fins administrativos e uma outra para fins operacionais. Quanto à camada de *Core*, cada tipo de utilizador será refletido num módulo, acompanhados de módulos de coordenação e do reconhecimento facial. Deste modo, módulos com entidades, *actions* e blocos públicos a serem consumidas por outros módulos são sucedidos pela sigla *_CS* (*Core Services*), enquanto módulos intermédios utilizados para orquestrar os restantes módulos são sucedidos pela sigla *_MBL* (*Mobile Business Logic*). Finalmente, os serviços identificados na camada inferior são, também, atribuídos a módulos correspondentes. Caso se tratem de integrações com *actions* próprias para a aplicação, são sucedidos pela sigla *_IS* (*Integration Services*). Caso se tratem somente da inclusão de *plugins*, sem uma lógica específica para utilização na aplicação, são sucedidos pela sigla *_Plug*. Finalmente, os módulos que dizem respeito a aspetos visuais são sucedidos pela sigla *_MTh* (*Mobile Theme*). Assim, o resultado obtido é o da Figura 4.5.

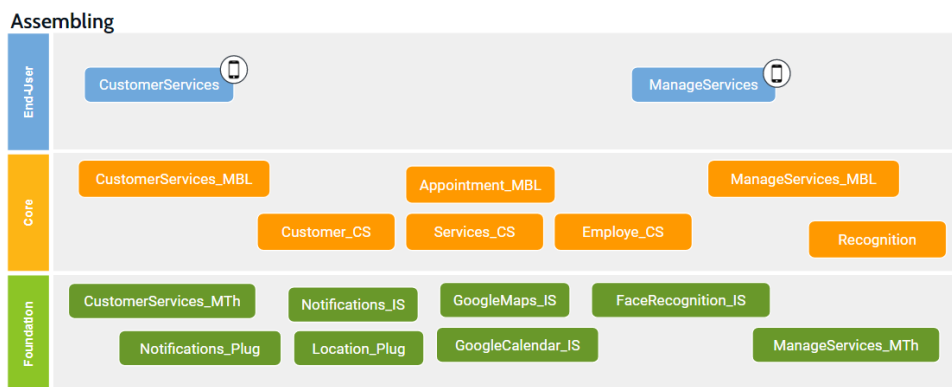


Figura 4.5: Passo intermédia da fase *Assemble*.

Contudo, uma vez que a escala do projeto não requer uma divisão tão exaustiva dos vários componentes, é possível agrupar ambas as interfaces, e por conseguinte os módulos da camada *Core*. Deste modo, as interfaces dos vários tipos de utilizador são implementadas num único módulo *End-User*, e também a lógica referente a todas as entidades é agregada num único módulo *Core*. Quanto à camada de *Foundation*, a única agregação possível é a dos temas da aplicação, que são unidos num só módulo, tal como as interfaces na camada superior. Finalmente, a visão final da arquitetura está completa, como ilustrado na Figura 4.6.

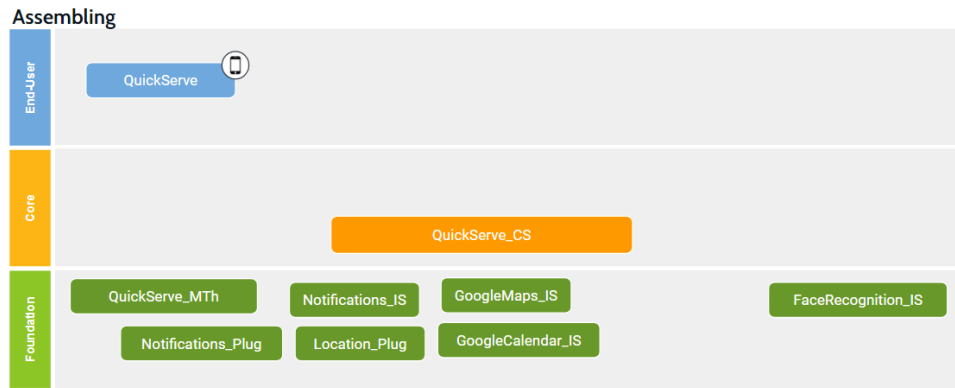


Figura 4.6: Resultado final da fase *Assemble*.

4.1.3 Diagrama de Entidades

O diagrama de entidades, representado na Figura 4.7, representa as relações entre diferentes conceitos armazenados em bases de dados. Assim, o passo inicial foi a definição das principais entidades a representar.

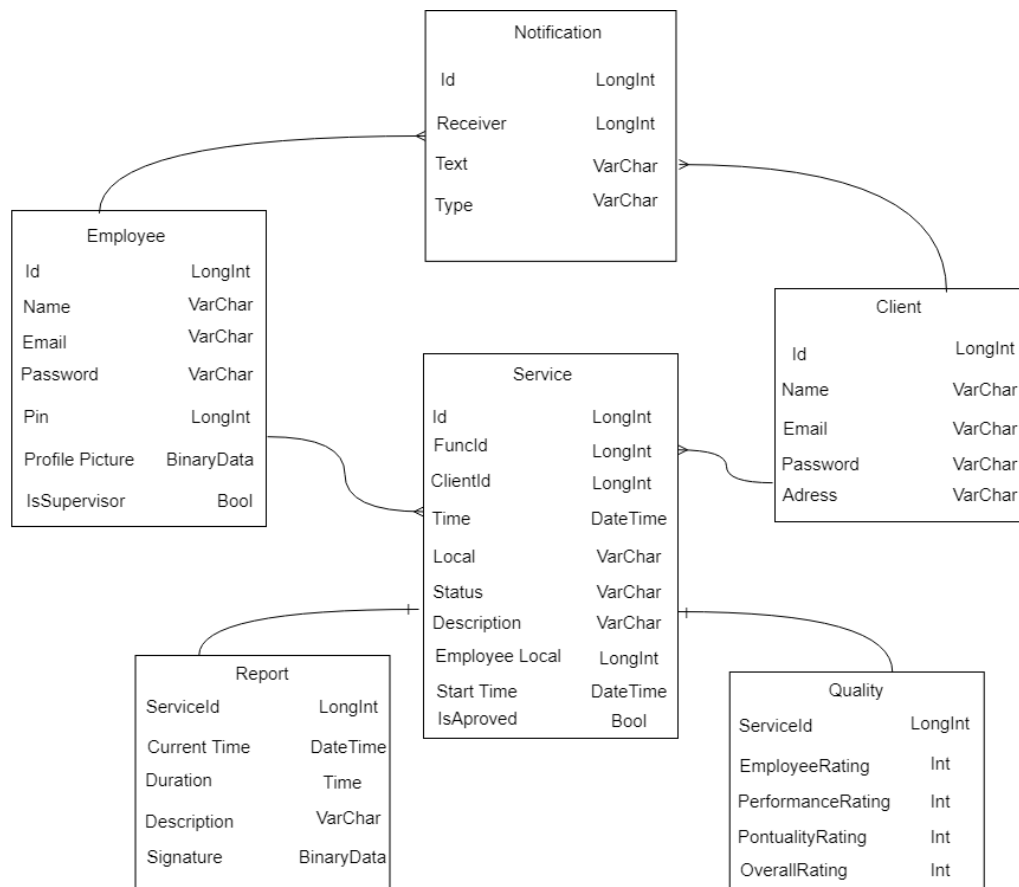


Figura 4.7: Representação das relações entre entidades na base de dados.

Para representar os vários tipos de utilizador foram necessárias entidades para Cliente e Funcionário, conferindo a cada uma os atributos necessários para a implementação dos requisitos funcionais identificados. Os supervisores são representados através de um atributo booleano na entidade Funcionários, não sendo justificada a sua representação numa entidade particular. Ainda referente aos utilizadores, existe uma entidade para representação de notificações, com uma relação *1-to-many*, quer com a entidade Funcionários, quer com a entidade Cliente, uma vez que qualquer tipo de utilizador pode ser destinatário de várias notificações.

Por fim, foi definida uma entidade para representação dos Serviços, com relações *1-to-many* com Clientes e Funcionários, uma vez que cada cliente pode solicitar mais do que um serviço e cada funcionário pode estar associado a mais do que um serviço de cada vez. Deste modo, os serviços apresentam uma relação de 1 para 1 para com as entidades de Inquérito e Relatório, uma vez que só pode existir um registo único associado a um serviço.

Tendo representado estas entidades no diagrama anterior, foi necessária a sua implementação na plataforma *low-code* da OutSystems, onde foi sujeita a algumas alterações. Primeiramente, é importante referir que, para além da noção mais usual de entidade, a OutSystems considera a existência de um tipo particular, designadas entidades estáticas. Estas representam dados imutáveis, ou seja, um conjunto de valores constante e pré-definidos.

Uma vez que o *Service Studio* da OutSystems tem, por omissão, uma entidade *User*, as entidades Cliente e Funcionário são representadas com uma relação de 1 para 1 com a entidade *User*. Deste modo, é possível limitar alguns atributos a utilizadores específicos, sendo possível, ao mesmo tempo, a referência a utilizadores gerais.

As restantes entidades permanecem inalteradas, com exceção das entidades Notifications e Services, sendo acrescentadas entidades estáticas em ambas. Nas notificações, o tipo de notificação é determinado pela entidade estática Notification Type, enquanto na entidade Service existe uma entidade estática para representar o estado do serviço, com registos “Por Validar”, “Por Iniciar”, “Por Concluir” e “Concluída”.

Assim, o modelo relacional da base de dados pode ser observado na Figura 4.8.

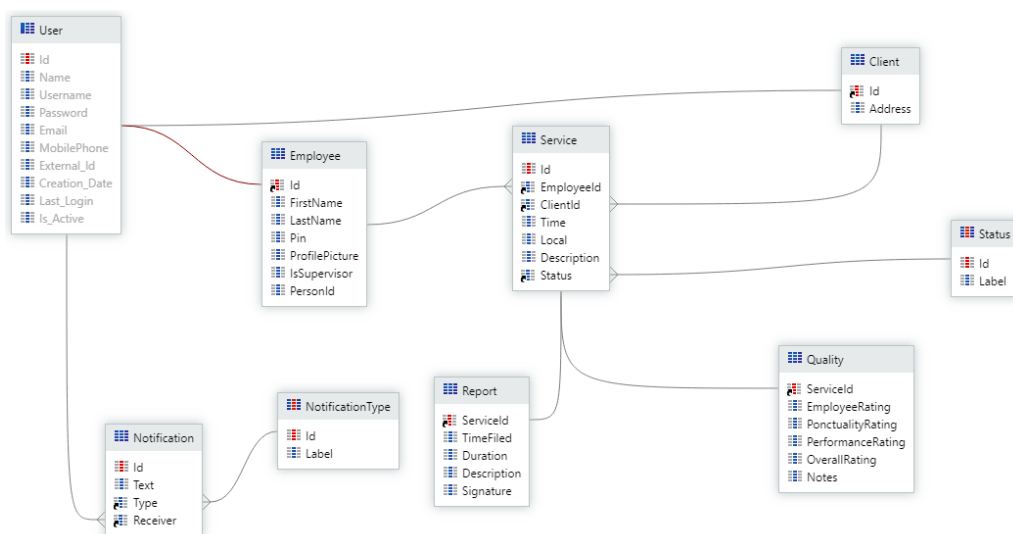


Figura 4.8: Representação das relações entre entidades na plataforma *low-code* da OutSystems.

4.1.4 Arquitetura Final

Ao longo do desenvolvimento do projeto, como é esperado da metodologia ágil, surgiram alguns aspetos que, para implementação dos requisitos identificados, tiveram que ser revistos.

Primeiramente, em relação ao modelo de dados, foram acrescentados alguns parâmetros em várias entidades para a implementação mais robusta de vários requisitos, como por exemplo, a adição de tempos de início e fim dos serviços, assim como a distância e tempo previsto para o deslocamento do funcionário para realização do mesmo. Outros exemplos são a inclusão de *tokens* para validação dos pedidos de recuperação de *password* e a sua data de validade, campos de custo nos relatórios, identificadores de serviço nas notificações, etc. Estes foram atributos acrescentados por uma questão de simplificar a visualização de informações por parte dos utilizadores, assim como simplificar as suas interações com a aplicação para utilizar as várias funcionalidades. Além disso, uma boa prática do desenvolvimento de soluções em OutSystems é a inclusão de atributos que façam referência ao utilizador que criou um registo de uma entidade, ou que o atualizou pela última vez. Assim, a cada entidade, foram adicionados dois atributos com identificadores de utilizadores, um para a criação do registo e outro para a sua última atualização.

Outra boa prática do desenvolvimento em OutSystems é a separação das fotos de perfil dos utilizadores (neste caso presentes apenas no registos de funcionários) numa entidade própria. Isto para evitar que, para cada vez que se pretendam visualizar informações sobre os funcionários, seja opcional o carregamento das fotos de perfil, uma vez que são informações mais pesadas e que podem não ser necessárias. Também por esta razão, a foto utilizada para autenticação é guardada numa entidade à parte, associada ao serviço correspondente. Pelo mesmo motivo, para implementar os casos de uso que envolvem a localização do funcionário, a mesma foi armazenada numa entidade própria para minimizar o custo das atualizações e consultas, uma vez que são muito frequentes.

Finalmente, foi criada a entidade FaceAPIFaceRecords para criar um registo das faces guardadas na API da Azure e assim facilitar as operações de atualização ou eliminação dos registos. Esta entidade é descrita em maior detalhe na secção da implementação do reconhecimento facial.

Assim, na Figura 4.9 é apresentado a versão final do modelo relacional da base de dados.

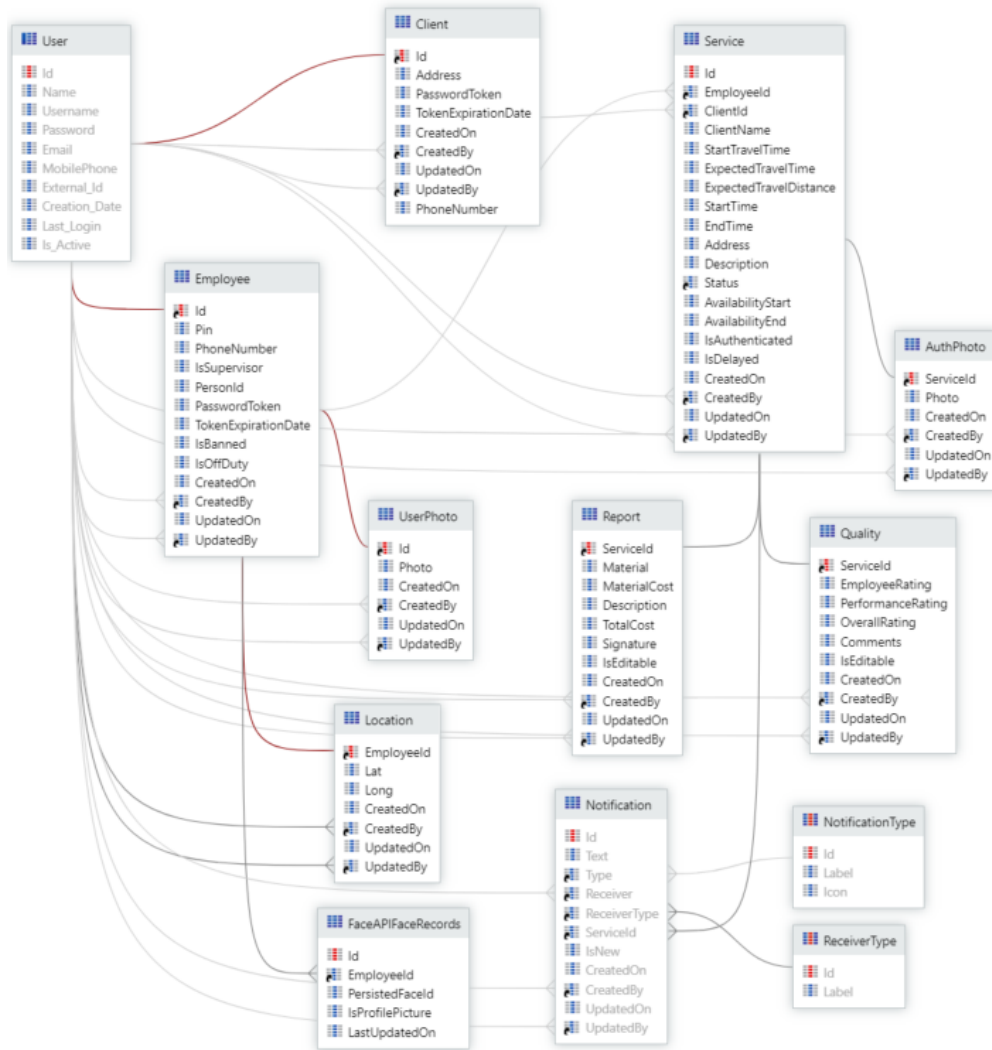


Figura 4.9: Representação final das relações entre entidades na plataforma *low-code* da OutSystems.

Além destas alterações no modelo de dados, foram também efetuadas algumas alterações na arquitetura definida inicialmente.

Primeiramente, a lógica das notificações foi separada do módulo *QuickServe_CS* por representarem contextos de uso diferentes. Deste modo, o módulo *QuickServe_CS* integra as entidades intervenientes na prestação de serviço (Client, Employee, Service, Status, Report, Quality, Location, FaceAPIFaceRecord, UserPhoto e AuthPhoto), assim como as *actions* referentes às mesmas, enquanto o módulo *Notifications_CS* disponibiliza um serviço auxiliar, que é a gestão de notificações, composto pelas entidades Notification, NotificationType e ReceiverType, bem como as *actions* correspondentes às mesmas. Assim, a arquitetura, foi atualizada para o modelo apresentado na Figura 4.10. Note-se que as integrações que se encontram fora da aplicação são usadas num contexto mais geral, uma vez que, não tendo sofrido alterações na sua lógica para serem usadas na aplicação, o seu contexto de utilização não se prende com a mesma.

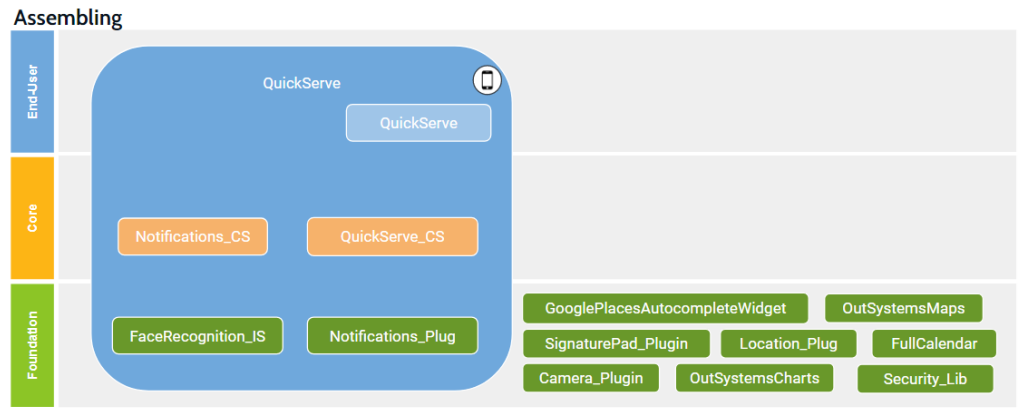
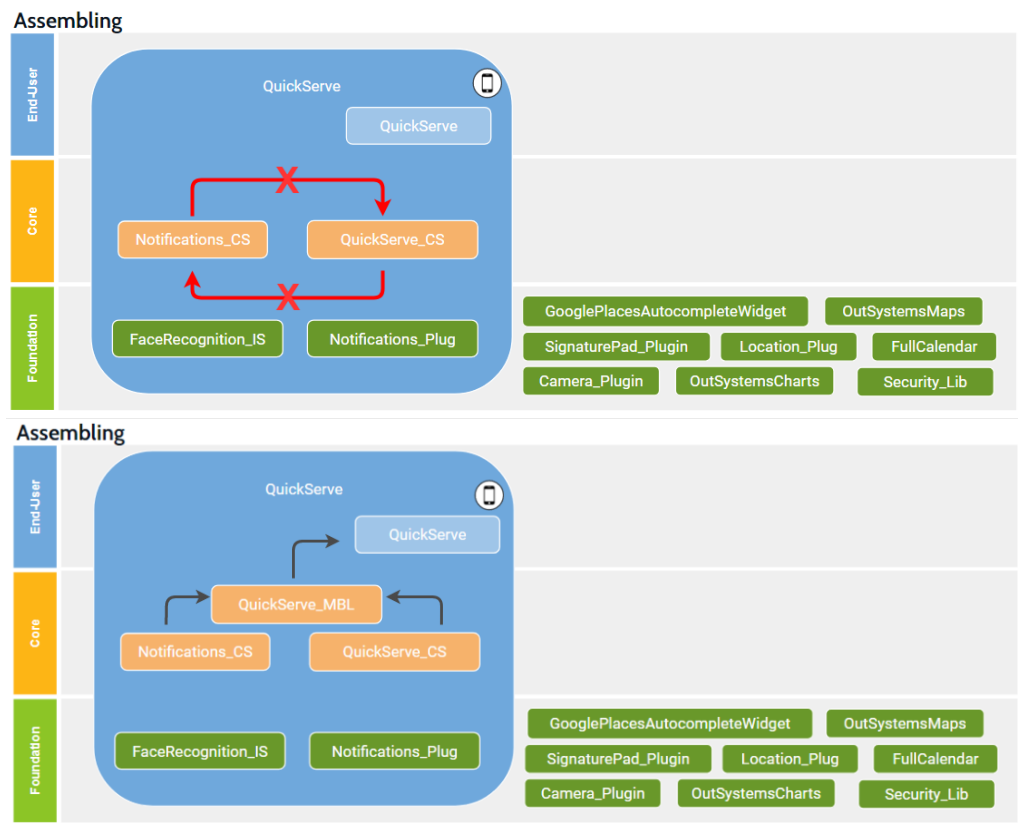


Figura 4.10: Arquitetura Atualizada

Contudo, tendo feita esta separação surge um problema de dependências circulares entre módulos da mesma camada, o que não é permitido na OutSystems, como referido anteriormente e ilustrado na Figura 4.3. Isto porque o módulo *QuickServe_CS* utiliza métodos definidos no módulo das notificações, e este, por sua vez, tem uma dependência de tipos de dados declarados no módulo *QuickServe_CS*, uma vez que cada notificação tem referências a identificadores de serviço e de utilizador.

Assim, é necessária a criação de um módulo intermédio que sirva para orquestrar estes dois módulos, pelo que foi criado o módulo *QuickServe_MBL*, como mostra a Figura 4.11. Note-se que o módulo *QuickServe_MBL* não contém entidades próprias, ou seja, consome apenas as entidades definidas nos módulos inferiores.

Figura 4.11: Arquitetura antes e depois da adição do módulo intermédio *QuickServe_MBL*.

4.2 Implementação

4.2.1 Plataforma Low-code

A OutSystems define *low-code* como uma abordagem que permite um desenvolvimento de aplicações mais rápido e com o mínimo de código escrito manualmente [32]. Assim, o Service Studio da OutSystems é uma plataforma de *low-code*, em que o desenvolvimento de aplicações é feito com recursos a interfaces, integrações, dados e lógica de uma forma acelerada, graças à representação visual fornecida pela plataforma e à ferramenta de *drag-and-drop* para produção dos elementos constituintes da aplicação.

O primeiro passo foi a criação de dois módulos para a aplicação *QuickServe*, como representado na Figura 4.12, um para desenvolvimento da lógica das interfaces e um *core module* para a lógica do servidor e das entidades, tal como descrito na secção da arquitetura. Assim, enquanto o módulo *QuickServe* se adequa para a implementação das interfaces dos utilizadores, no módulo *QuickServe_CS* criam-se todas as *server actions* necessárias para a manutenção de utilizadores na aplicação, como por exemplo o registo, a atualização e eliminação dos mesmos, assim como o modelo de dados e o respetivo diagrama de entidades.

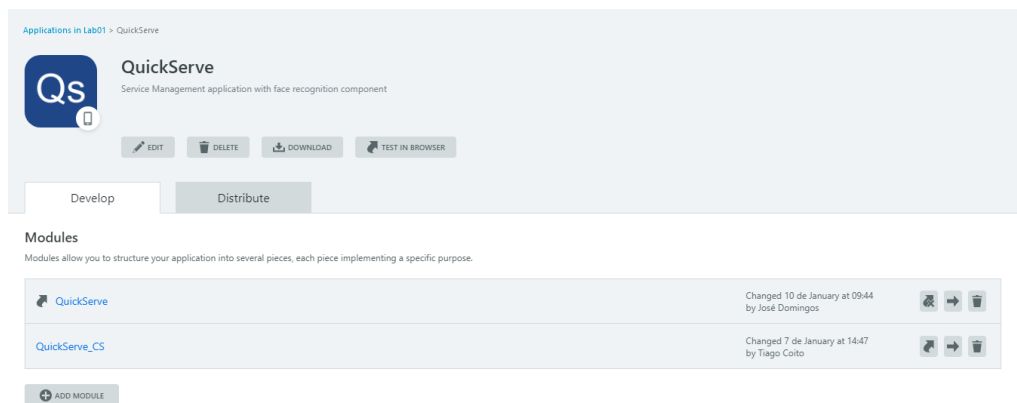


Figura 4.12: Módulos da aplicação criados no Service Studio

Tendo estes módulos criados, deu-se início ao desenvolvimento da aplicação na área de trabalho do Service Studio. Assim, segue-se na Figura 4.13 uma breve exposição dos vários componentes que integram este ambiente de desenvolvimento.

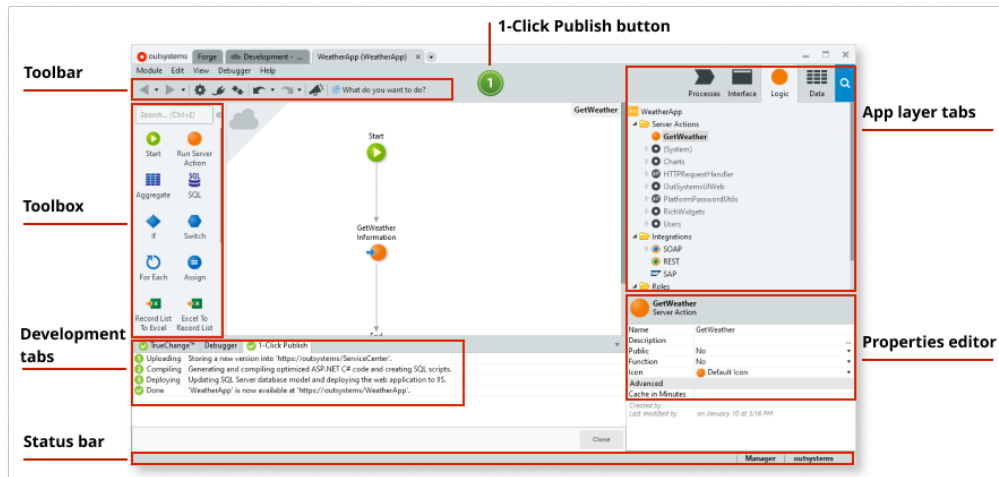


Figura 4.13: Visão dos Componentes do Service Studio da OutSystems [33].

- *Toolbar*: contém atalhos para as operações mais comum e recorrentes;
- *Toolbox*: contém as ferramentas necessárias para a criação da lógica da aplicação, bem como os *widgets* mais apropriados para a construção dos ecrãs;
- *Development tabs*:
 - *TrueChange tab*: exibe os erros e determinados avisos existentes no módulo;
 - *Debugger tab*: permite iniciar o debugger e consultar o conteúdo das variáveis passo a passo ao longo da lógica implementada;
 - *1-Click Publish tab*: mostra o progresso e resultado do processo de deployment;
- *1-Click Publish button*: inicia o deployment da aplicação. Se o modulo apresentar erros, esta operação fica indisponível;
- *Application layer tabs*: apresenta o conteúdo referente a cada camada da aplicação (*processes, interface, logic e data*);
- *Properties editor*: permite consultar e definir as propriedades de um dado elemento selecionado;
- *Status bar*: exibe informações sobre a última vez que o módulo foi atualizado, o utilizador daquele ambiente de desenvolvimento em específico e o servidor onde a app está a ser guardada.

A área central não assinalada corresponde à área de desenvolvimento, onde são colocados os componentes disponíveis na *toolbox*. Esta secção apresenta duas vertentes possíveis, correspondentes ao desenvolvimento de interfaces do utilizador e ao desenvolvimento da lógica das ações de cliente e servidor. Alternar entre estas duas vertentes influencia, também, os componentes disponíveis na *toolbox*, como evidenciado na Figura 4.14.

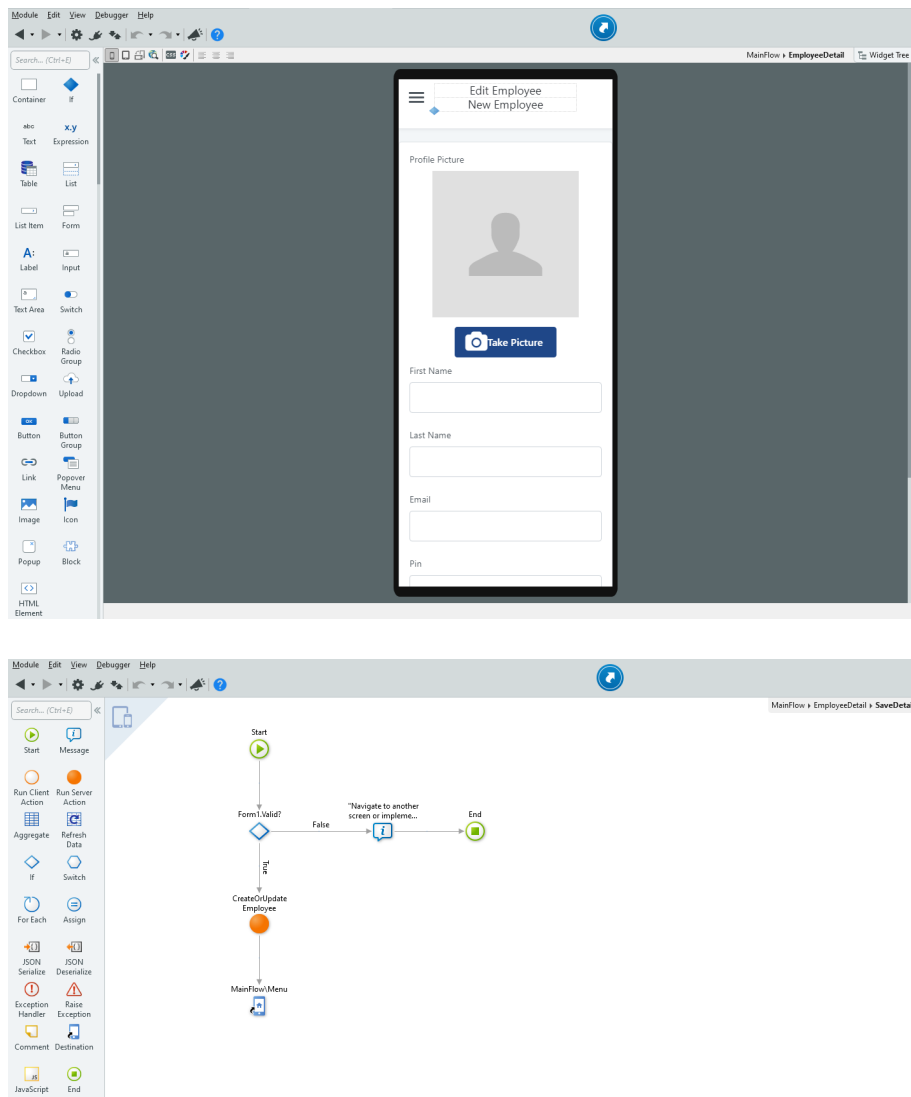


Figura 4.14: Desenvolvimento de Interfaces (superior) e Desenvolvimento de Ações (inferior).

Assim, é possível transpor para a OutSystems código escrito numa linguagem tradicional, como por exemplo, código em C#. Por exemplo, dado o código da Figura 4.15, cujo o objetivo é a iteração sobre uma lista de utilizadores e a contabilização do número de utilizadores cujo nome é “Ana”, é fácil a sua implementação em OutSystems, com recurso aos elementos *For Each* e *If*, como demonstrado na Figura 4.16.

```

static void Main() {
    int matches= GetMatches(userList);
    Console.WriteLine("There are "+matches+" matches");
}

static int GetMatches(List<User> userList){
    int matches=0;

    foreach(User u in userList)
        if(u.name == "Ana")
            matches+=1;
    return matches;
}
    
```

Figura 4.15: Código C# para contabilizar correspondências numa lista.

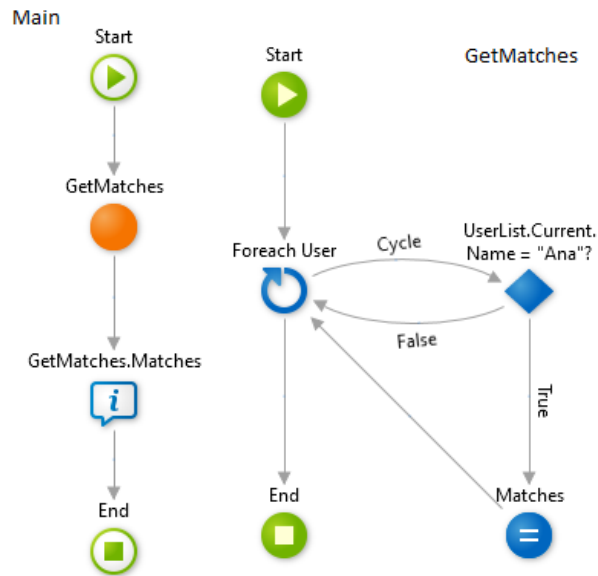


Figura 4.16: Este código executa a lógica na Figura 4.15. Tal como o código em C#, a *action* GetMatches recebe uma lista de utilizadores e tem como parâmetro de saída o número de correspondências, impresso na *action* Main através do elemento *Message*.

4.2.2 Plugins

Para poder integrar os *plugins* na solução é necessário proceder à sua instalação a partir da *Forge* da OutSystems. A *Forge* trata-se de um repositório de módulos, conectores e componentes de *interface* de utilizadores de uso livre para acelerar o processo de desenvolvimento de aplicações. Assim, os seguintes *plugins* foram utilizados no desenvolvimento da solução.

Azure Cognitive Services

Tal como foi referido no segundo capítulo, a API escolhida para a componente de reconhecimento facial foi a da Azure Cognitive Services. Uma das razões que levou a esta escolha foi a existência de um componente na *Forge* que facilita a integração destes serviços na solução. Deste modo, após a instalação deste mesmo componente, foram especificadas quais as funcionalidades a utilizar para a solução, pelo que foram selecionadas apenas algumas da *FaceAPI*, como mostra a Figura 4.17.

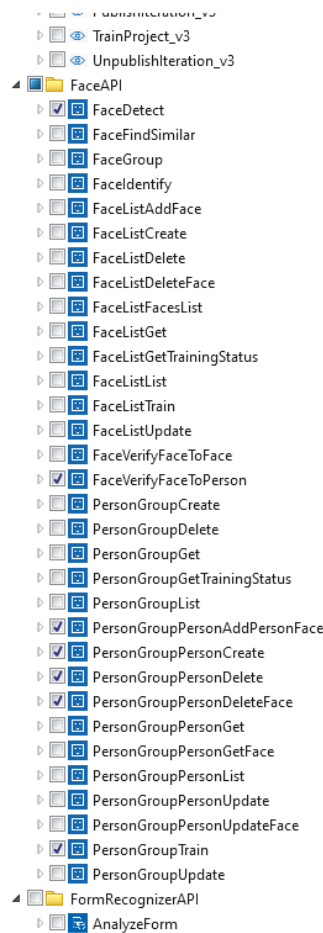


Figura 4.17: Serviços selecionados da *FaceAPI*.

Camera Plugin

Para integração dos serviços da câmara foi utilizado o *Camera Plugin* publicado pela OutSystems. Incluídas neste *plugin* estão três *client actions*: uma para verificar se a instalação do *plugin* foi feita corretamente, uma para selecionar imagens da galeria e uma última para tirar uma fotografia com a câmara do dispositivo.

Além destas funcionalidades, estão incluídos, também dois blocos OutSystems, que representam código reutilizável e com lógica própria para permitir a rápida integração dos mesmos numa variedade de aplicações.

Assim, foi utilizado o bloco *Take Picture*, que inclui um botão com as ações referidas anteriormente para tirar uma foto, devolvendo como parâmetro de saída uma variável do tipo *Binary Data*, que representa a foto tirada com a câmara. Esta lógica encontra-se representada na Figura 4.18. Finalmente, esta variável pode ser armazenada na base de dados, associada ao utilizador que a tirou.

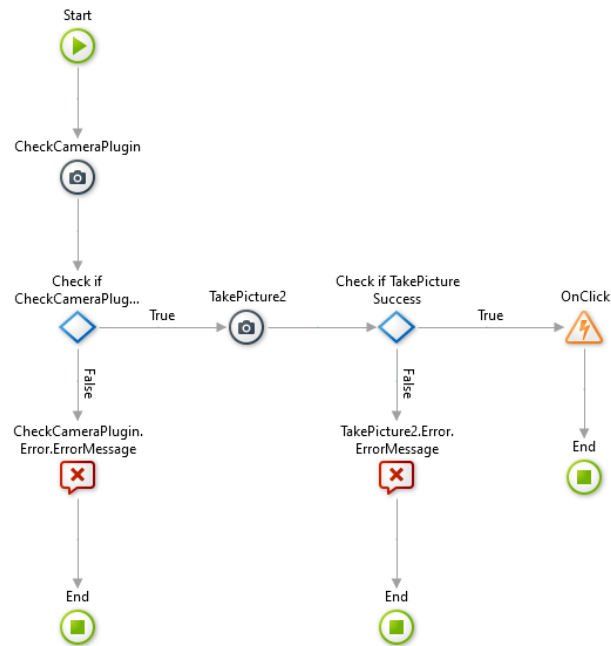


Figura 4.18: Lógica do bloco *Take Picture*.

Location Plugin

Os serviços de localização foram implementadas com recurso ao *Location Plugin*, também publicado pela OutSystems. Para a integração deste *plugin* foram utilizadas apenas duas das *client actions* disponíveis, com o objetivo de verificar a instalação do componente e iniciar o rastreo da posição do funcionário no momento de *login*, assim como atualizar a localização do funcionário em base de dados sempre que se regista uma alteração na mesma. Para este efeito, foi desenvolvida lógica apresentada na Figura 4.19.

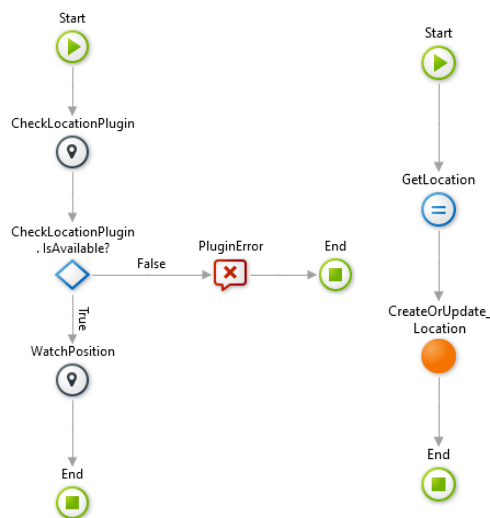


Figura 4.19: À esquerda é apresentada a lógica que dá início ao rastreo da posição do funcionário, e à direita é apresentada a lógica para atualização das coordenadas sempre há alteração das mesmas.

OutSystems Maps

A utilização dos serviços do Google Maps foi integrada com recurso ao componente de mapas da OutSystems, que permitiu a utilização de um bloco, ou seja, um componente de código reutilizável, para visualização dos mapas. Deste modo, a sua utilização nos ecrãs onde é necessária a visualização de mapas requereu simplesmente a definição de alguns parâmetros, como por exemplo, as coordenadas do ponto que se encontra no centro do mapa. A conjugação com o *Location Plugin* revelou-se muito útil, uma vez que tornou possível centrar o mapa na posição atual do funcionário, acompanhado com um marcador visual, também suportado pelo bloco.

Contudo, as funcionalidades do bloco revelaram-se pouco satisfatórias, uma vez que, apesar de ser possível a criação de vários marcadores, a visualização simultânea de marcadores a uma grande distância tornava complexa a navegação do mapa, pelo que se concluiu que seria vantajosa a inclusão de um sistema de direções, onde o trajeto a seguir seria apresentado ao funcionário. Um outro aspeto muito negativo era a necessidade de introduzir as coordenadas do local de prestação do serviço, não sendo suficiente a introdução de uma morada. No entanto, um dos parâmetros suportados pelo bloco corresponde a uma chave de API da Google, pelo que foi decidido explorar esta opção como forma de complementar os serviços já implementados. Assim, foi criado um projeto na Google Cloud Platform e definida uma chave API com acesso a 3 APIs: Directions API, Geocoding API e Maps Javascript API, como representado na Figura 4.20. A primeira das API permitiu a utilização de serviços de navegação entre 2 pontos, pelo que, quando um funcionário inicia um serviço, é possível visualizar uma rota desde a sua posição ao local onde será prestado o serviço. A Geocoding API permite traduzir uma morada nas coordenadas GPS associadas, o que representa uma ótima funcionalidade para o cliente no momento de criação do serviço. Finalmente, a inclusão da Maps Javascript API permite retirar a marca de água de desenvolvimento do bloco do mapa.

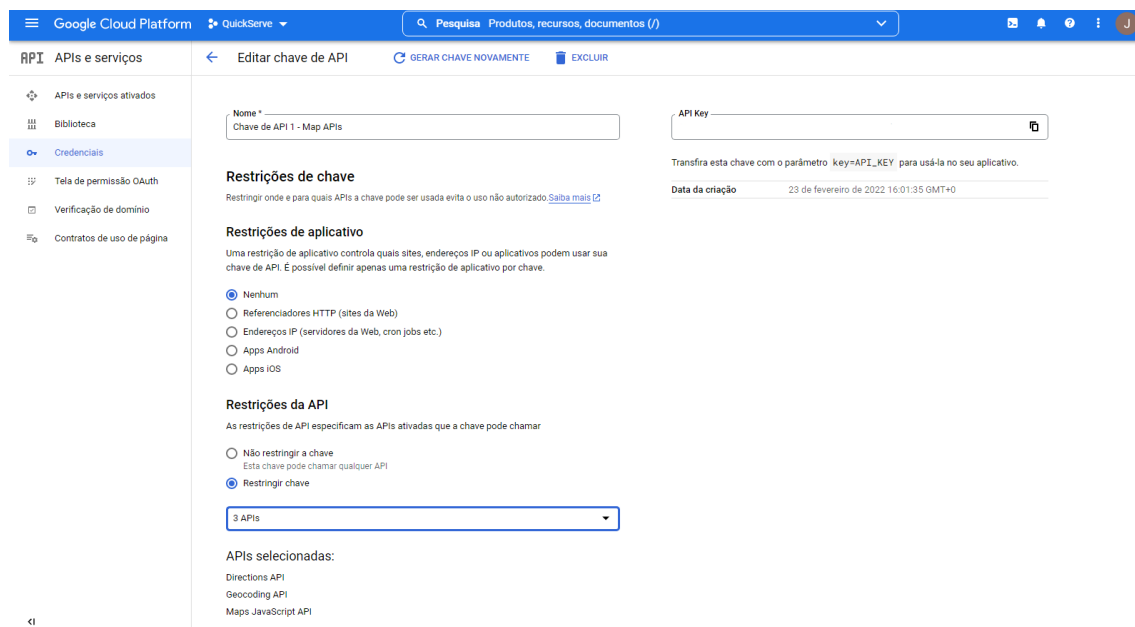


Figura 4.20: Criação da chave API para utilização de serviços do Google Maps.

Em termos de usabilidade, foram tomadas algumas decisões para facilitar a interação dos utilizadores com a solução. Nomeadamente, foram atribuídos ícones diferentes para marcadores de posição de funcionários e locais de prestação de serviço, de modo a possibilitar

uma diferenciação imediata. Além disso, para o supervisor, foi implementada a opção de expandir um *pop-up* com informações complementares, quer do funcionário, quer do serviço, como representado na Figura 4.21.



Figura 4.21: Visualização de informações complementares no mapa.

Finalmente, por motivos de segurança, a chave API criada na Google Cloud Platform foi definida como uma variável a ser chamada em todos os ecrãs que utilizam o bloco do *Outsystems Maps Plugin*.

Google Places Autocomplete Widget

Para simplificar o processo de inserir uma morada para a prestação de serviços, este componente foi utilizado para apresentar sugestões ao utilizador à medida que este introduz o local de prestação de serviço. Assim, através da utilização deste componente da Forge, foi possível simultaneamente simplificar o processo de introdução da morada, assim como proteger a mesma de eventuais erros do utilizador.

Para utilização deste componente foi necessária a chave API definida anteriormente, assim como especificar o país a que se deve limitar a procura (neste caso, Portugal). Finalmente, foi especificado o tipo de sugestões que deviam ser apresentadas, ou seja, nomes de ruas. Outras opções como o nome de regiões ou cidades não foram consideradas relevantes.

Deste modo, a lógica que atualiza as sugestões apresentadas pelo componente, representada na Figura 4.22, é executada sempre que houver uma alteração no campo da morada.

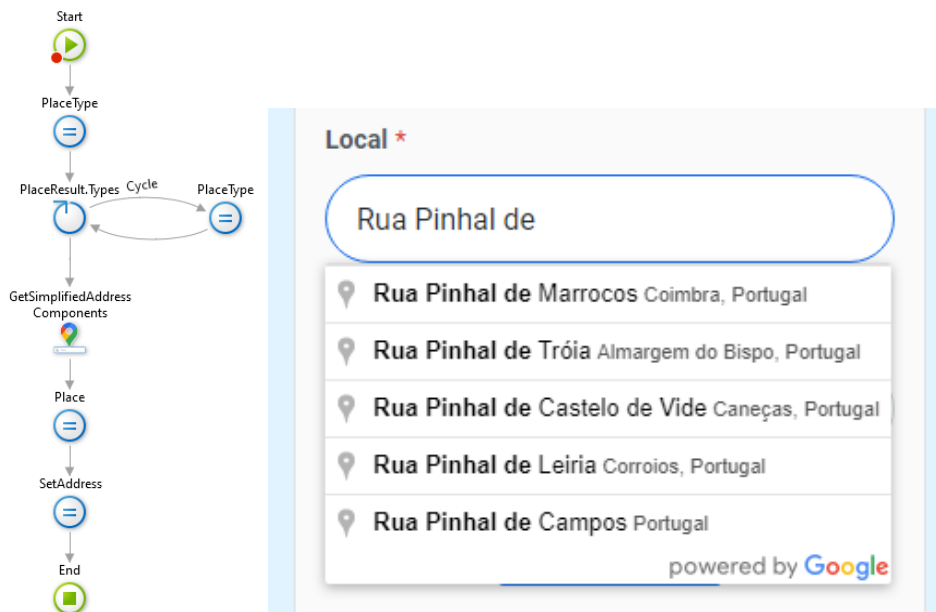


Figura 4.22: Cada um dos tipos de sugestões é inserido numa estrutura auxiliar (neste caso, os nomes das ruas), sendo devolvida a opção seleccionada pelo utilizador.

FullCalendar Reactive

Este *plugin* foi utilizado para integrar um calendário nas páginas iniciais de cada utilizador. Para tal, foi utilizado o componente da Forge, cuja parametrização consistiu na definição de uma data inicial para o calendário, a língua do mesmo e uma variável a partir da qual são carregados os eventos a apresentar, que será explicada na secção seguinte. A data inicial foi definida através da função da OutSystems *CurrDate*, que devolve a data no dispositivo atual.

OneSignal Notifications

Para implementação dos requisitos que incluem a troca de notificações foi utilizada uma combinação dos *plugins* OneSignalPlugin e OneSignalAPI. O primeiro diz respeito à criação de registos que relacionam utilizadores e dispositivos, ou seja, de cada vez que o utilizador inicia sessão é criado um registo para o dispositivo que foi utilizado, com um identificador do utilizador que iniciou sessão. O segundo disponibiliza várias *server actions* para utilizar a API OneSignal, cuja principal utilidade é a lógica de envio de notificações *push*.

Assim, antes de começar a integrar estes dois *plugins* na solução, é necessária uma configuração prévia do projeto na API OneSignal. Como se trata de um projeto para Android, o primeiro passo é a criação de um projeto Firebase. De seguida, nas definições do projeto criado, ao ativar a opção *Cloud Messaging*, são gerados um *token* e um identificador de remetente, conforme representado na Figura 4.23.

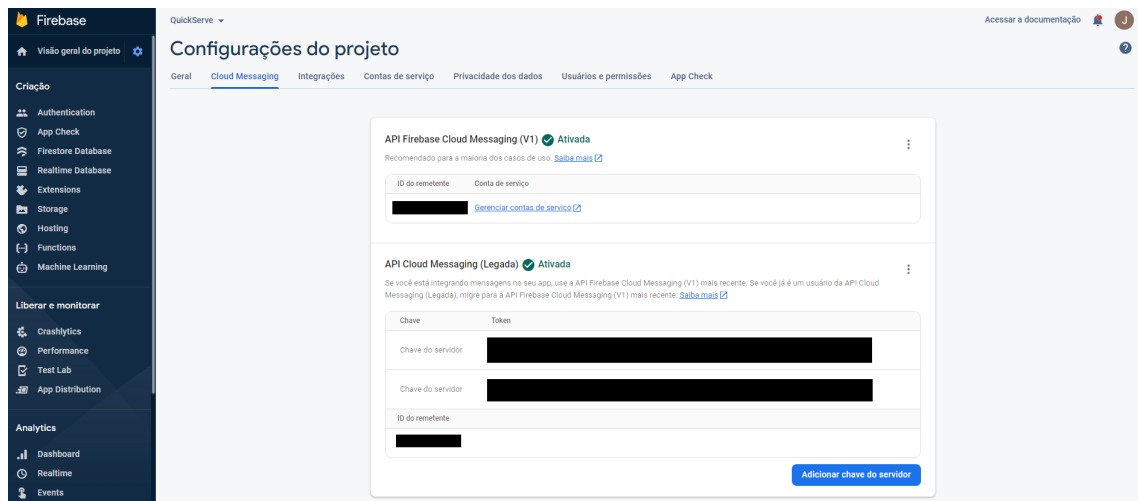


Figura 4.23: Configuração do projeto Firebase.

Com esta configuração finalizada, foi possível criar um projeto na API OneSignal com estes dois parâmetros, cuja plataforma selecionada foi o sistema Android, como representado na figura 4.24.

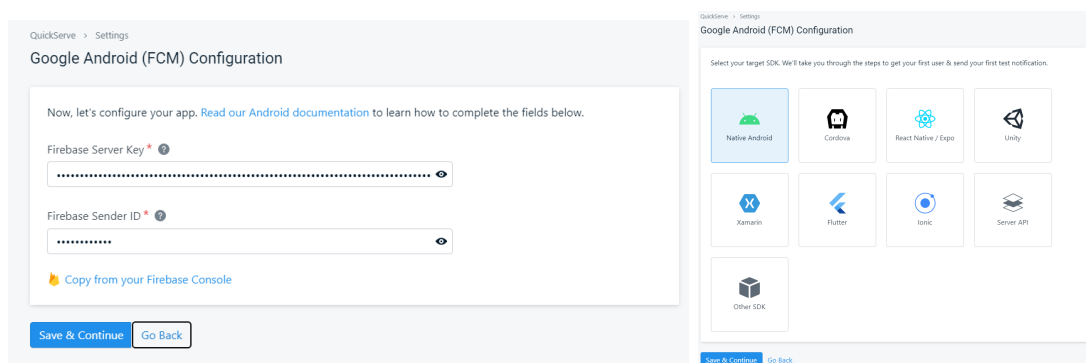


Figura 4.24: Configuração do projeto na API OneSignal.

Com a configuração concluída, foi devolvido um identificador de aplicação, que permite ao *QuickServe* fazer chamadas à API no projeto que foi criado. Assim, para a integração deste componente na solução, o primeiro passo foi a utilização da *action RegisterWithUser* do *OneSignalPlugin* durante o *login* na aplicação, como representado na Figura 4.25

Finalmente, para o envio das notificações foi definida a *action CreateAndSendOneSignalNotification* para criar um registo da notificação para visualização na aplicação, assim como utilizar a *action SendPushNotificationToUserId* do *plugin OneSignalAPI*, que envia a notificação *push* correspondente. A Figura 4.26 representa a *action* descrita anteriormente.

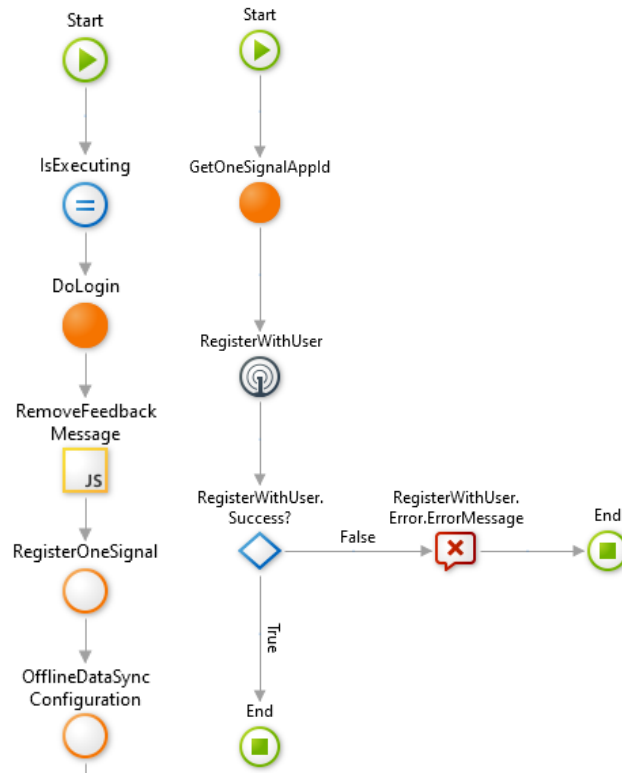


Figura 4.25: À esquerda é apresentada a lógica de *login*, e à direita é apresentada em detalhe a *action* RegisterOneSignal.

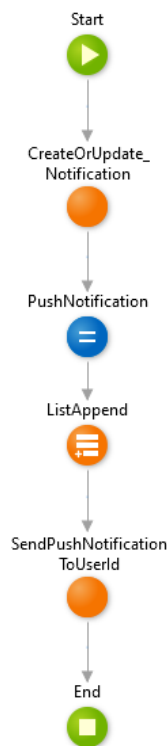


Figura 4.26: Lógica para criação e envio de notificações *push*.

SignaturePad

Este *plugin* foi utilizado para extrair um objeto binário, ou seja, uma imagem, a partir das deslocações de um dedo no ecrã. Assim, o bloco incluído neste *plugin* inclui um evento que atualiza o atributo *Signature* do relatório sempre que acaba a deslocação de um dedo no ecrã. Além deste evento, existem dois botões para apagar a assinatura (no momento de preenchimento do relatório) e outro para concluir o processo e fechar o *pop-up* onde este decorre.

OutSystems Charts

Tal como o *plugin* dos serviços de mapas, a OutSystems disponibiliza um componente para utilização de gráficos, que foi utilizado no ecrã dos índices de desempenho, visível apenas para supervisores. Deste modo, as dependências utilizadas deste componente limitaram-se à utilização de um gráfico circular.

4.2.3 Fases de Desenvolvimento

Preparação

Esta fase inicial consistiu no desenvolvimento dos ecrãs de *login* e registo dos diferentes tipos de utilizadores. Para cada um dos formulários de ingresso foram aplicadas proteções, de modo a que as informações inseridas sejam válidas e em conformidade com a base de dados. Por exemplo, não são aceites e-mails iguais para utilizadores diferentes. Estas informações podem ser alteradas posteriormente.

Simultaneamente, foram configurados os serviços de e-mail, de modo a que os utilizadores sejam notificados no caso de um registo bem sucedido ou de realização de pedidos de recuperação de *password*.

O mecanismo de recuperação de *password* foi implementado com recurso a um *token* gerado no momento em que é efetuado o pedido e associado ao utilizador através do campo *PasswordToken* representado na Figura 4.8. Assim, o *token* gerado é incluído no *link* para a página de recuperação de *password*, incluído no e-mail enviado ao utilizador que realizou o pedido. Finalmente, neste ecrã, representado na figura 4.27, o *token* é comparado com os registos na base de dados e a sua data de validade é confirmada com os mesmos. Caso as informações estejam válidas, a *password* é alterada conforme a indicação do utilizador.

No caso de registo de funcionários, foi criado um ecrã de gestão de funcionários, onde o supervisor pode consultar uma lista de funcionários, assim como informações sobre os mesmos. Caso pretenda pesquisar por um funcionário em específico, pode utilizar o filtro no topo do ecrã. Finalmente, neste ecrã o supervisor pode eliminar funcionários, assim como registar novos funcionários.

No registo de funcionários, o supervisor preenche o seu nome e e-mail, para o qual será enviado um e-mail com um *link* para terminar o processo de inscrição. Parte da lógica é igual ao processo de recuperação de *password*, uma vez que é gerada uma *password* aleatória no momento de registo, que deve ser posteriormente alterada pelo funcionário. Isto para que a presença do funcionário não seja necessária no momento do registo. Além disso, é definido um código de 4 dígitos para que o funcionário se possa autenticar caso o reconhecimento facial falhe. Finalmente, é pedido ao funcionário para definir uma foto

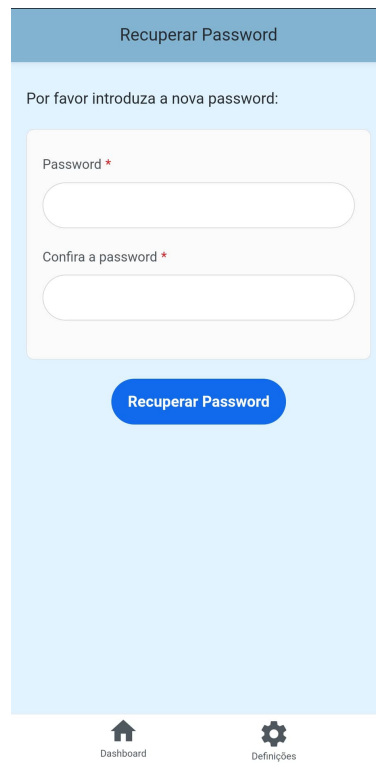


Figura 4.27: Ecrã de Recuperação de Password.

de perfil, não só para fins de identificação dentro da aplicação, mas também para criar o registo do funcionário na *FaceAPI* da Azure.

Agendamento de Serviços

O passo seguinte diz respeito à lógica de criação de serviços, uma vez que estes servem de base para as outras funcionalidades a implementar na solução. O processo de criação de serviços, acessível a supervisores e clientes, consiste na indicação de uma data e hora, assim como o local onde o serviço deve ser prestado. Este deverá ser introduzido pelo utilizador com recurso ao *plugin* de sugestão de moradas mencionado anteriormente, e confirmada através do mapa que surge com a localização depois de introduzida. O supervisor pode ainda preencher informações como a descrição do serviço e o funcionário atribuído para prestação do mesmo.

Para a representação dos serviços agendados foi utilizado o componente FullCalendar Reactive, da Forge da OutSystems. Este componente, muito utilizado para calendarização em projetos em React, permite não só a visualização do calendário e uma elevada customização da mesma, como também suporta lógica para agendamento de eventos, que podem ser visualizados no próprio calendário. Assim, ao inicializar um ecrã onde é visível o calendário, são selecionados da base de dados os serviços do utilizador com a sessão iniciada. Estes serviços são, posteriormente, convertidos para uma estrutura de dados suportada pelo componente, para que possam ser apresentados no calendário. Na Figura 4.28 está representada a *client action* com a lógica que correspondente a este processo.

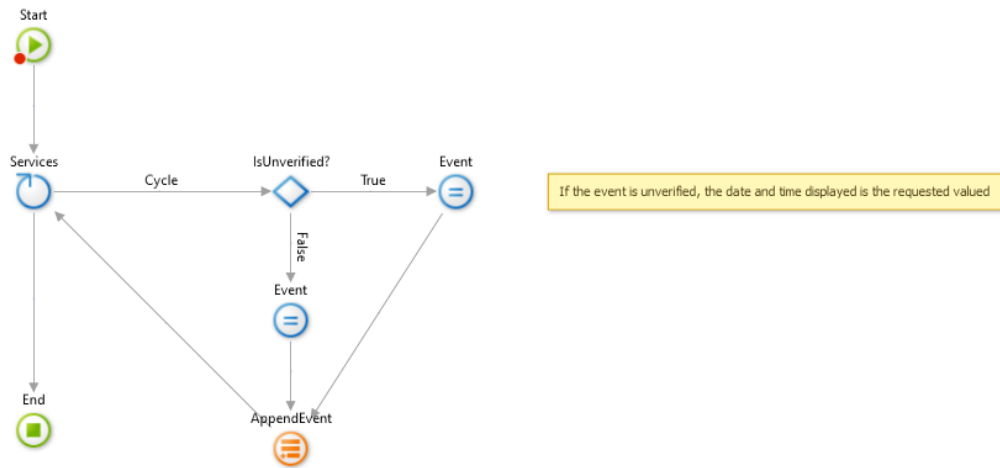


Figura 4.28: Lógica que converte os serviços guardados em base de dados para um formato apresentável pelo calendário.

Além desta lógica, foram utilizadas duas *actions* para gerir as interações com o calendário. Primeiramente, no caso do funcionário e do supervisor, é apresentada uma lista dos serviços agendados para o dia selecionado no calendário. Para tal, foi necessária uma *action* que filtre os serviços do calendário para apresentar apenas os agendados para a data atualmente selecionada, representada na Figura 4.29.



Figura 4.29: Lógica que filtra os serviços consoante a data selecionada no calendário.

De seguida, foi implementado um *pop-up*, representado na Figura 4.30, para todos os utilizadores, que permite visualizar os detalhes de um serviço quando este é selecionado no calendário.

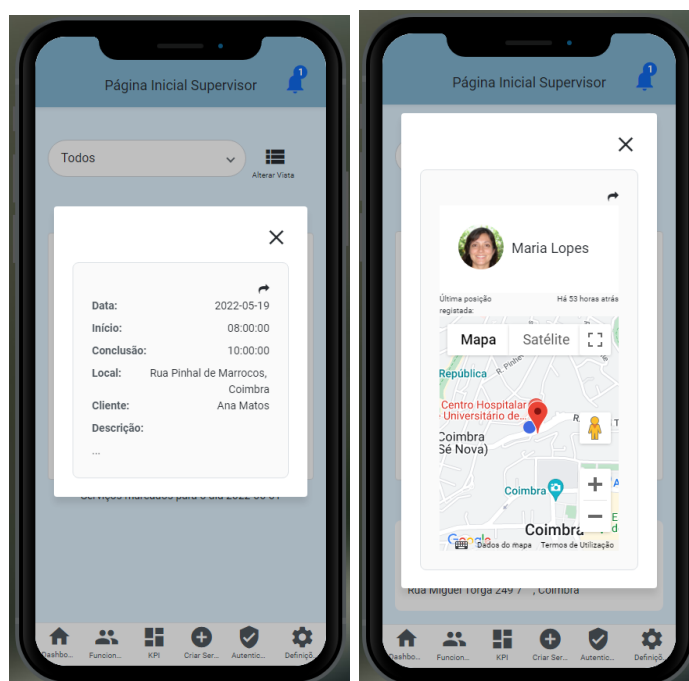


Figura 4.30: Consulta das informações do serviço.

Na página inicial do supervisor estão disponíveis duas formas de visualizar os serviços, quer pelo calendário, quer por uma lista de serviços. Para facilitar a consulta de um elevado número de serviços foi implementada uma *dropdown* cujas opções são os vários estados possíveis de um serviço. Deste modo, quer a lista quer o calendário apresentam os serviços com o estado selecionado. Contudo, para o calendário é acrescentada a possibilidade de visualizar todos os serviços, independentemente do seu estado.

Finalmente, foram tomadas certas considerações sobre a edição de informações de serviços. Caso o serviço ainda não tenha sido confirmado pelo supervisor, todas as informações são editáveis, exceto o local de prestação de serviço. Quando o serviço passa ao estado “Por Iniciar”, ou seja, quando o supervisor valida o serviço ou o cliente aceita um pedido de recalendarização, as únicas informações editáveis são a descrição e o funcionário associado ao serviço. Quando o serviço passa ao estado “A decorrer”, ou seja, quando o funcionário inicia a deslocação para o serviço, nenhuma informação é editável.

Autenticação de Funcionários

Depois de implementada a lógica de criação de serviços, foi finalmente possível implementar a autenticação de funcionários por reconhecimento facial.

Primeiramente, foi desenvolvida a lógica para registar funcionários, que inclui criar um registo da sua cara num grupo existente na API. Para tal, usou-se a *server action* `PersonGroupPersonCreate`, que associa um identificador ao novo funcionário adicionado. Este identificador será associado ao perfil deste funcionário através do atributo `PersonId`, que se encontra representado no diagrama de entidades da Figura 4.8. De seguida, o registo do funcionário e da sua foto de perfil são guardados na base de dados do *QuickServe*.

Tendo estes registos criados, é necessário fornecer aos serviços da Azure a cara associada a este funcionário, pelo que a `PersonId` e a foto do funcionário são fornecidas à *server action* `PersonGroupPersonAddPersonFace` para que o seu rosto seja associado a este registo na

API.

Finalmente, é recomendado o uso de pelo menos 5 imagens para o reconhecimento facial, pelo que se decidiu que, enquanto este limite não fosse atingido, todas as ações de autenticação ou atualização de foto de perfil seriam acompanhadas de uma repetição do passo anterior. Caso o limite já se encontre atingido, as informações da face mais antiga guardada na API são apagadas através da *server action* `PersonGroupPersonDeleteFace` antes de se dar o novo registo. Este passo é acompanhado de uma atualização dos registos da entidade `FaceRecord` na base de dados do *QuickServe*, uma vez que esta é a entidade que guarda os identificadores de cada cara para que possam ser referenciados numa chamada à API da Azure, como exemplificado na Figura 4.31.

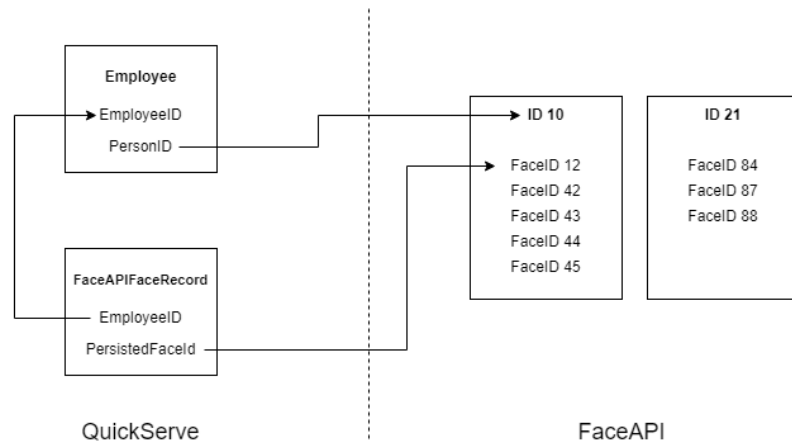


Figura 4.31: Relação entre a entidade `FaceAPIFaceRecord` e os registos de caras na API da Azure.

Assim, a *action* com a lógica de registo do funcionário encontra-se representada na Figura 4.32.

Quanto ao reconhecimento dos funcionários, utiliza-se o acesso à câmara para tirar uma fotografia do funcionário, e esta será fornecida à *server action* `FaceDetect`. Caso esta devolva uma cara detetada válida, servirá de parâmetro de entrada à *server action* `FaceVerifyFaceToPerson`, assim como o identificador `PersonId` do funcionário com sessão iniciada. Assim, a cara que existe no registo do funcionário será comparada com a cara detetada na etapa `FaceDetect` e, caso a correspondência seja positiva e com um grau de confiança superior a um valor pré-definido, o utilizador é autenticado. Para esta aplicação, o grau de confiança definido inicialmente foi de 60%, tendo sido testado posteriormente na fase de testes do reconhecimento facial. Como referido anteriormente, caso a autenticação seja bem sucedida e o número de caras associadas a este funcionário seja 5, o registo mais antigo é apagado através da *server action* `PersonGroupPersonDeleteFace` e substituído por um novo registo.

Assim, a Figura 4.33 representa as duas *actions* com a lógica descrita anteriormente, uma para a gestão dos registos da entidade `FaceRecord` e outra para a autenticação do funcionário através da API.

Como alternativa ao reconhecimento facial, caso este sofra dificuldades técnicas, é disponibilizada a opção de inserir o código definido no momento de registo caso a autenticação por reconhecimento facial falhe mais do que 5 vezes. Caso o funcionário insira o código errado 3 vezes será bloqueado da aplicação, tendo que entrar em contacto com o supervisor para restaurar o seu acesso e autenticá-lo manualmente. Para este efeito, foi definido um ecrã próprio para o supervisor, representado na Figura 4.34.

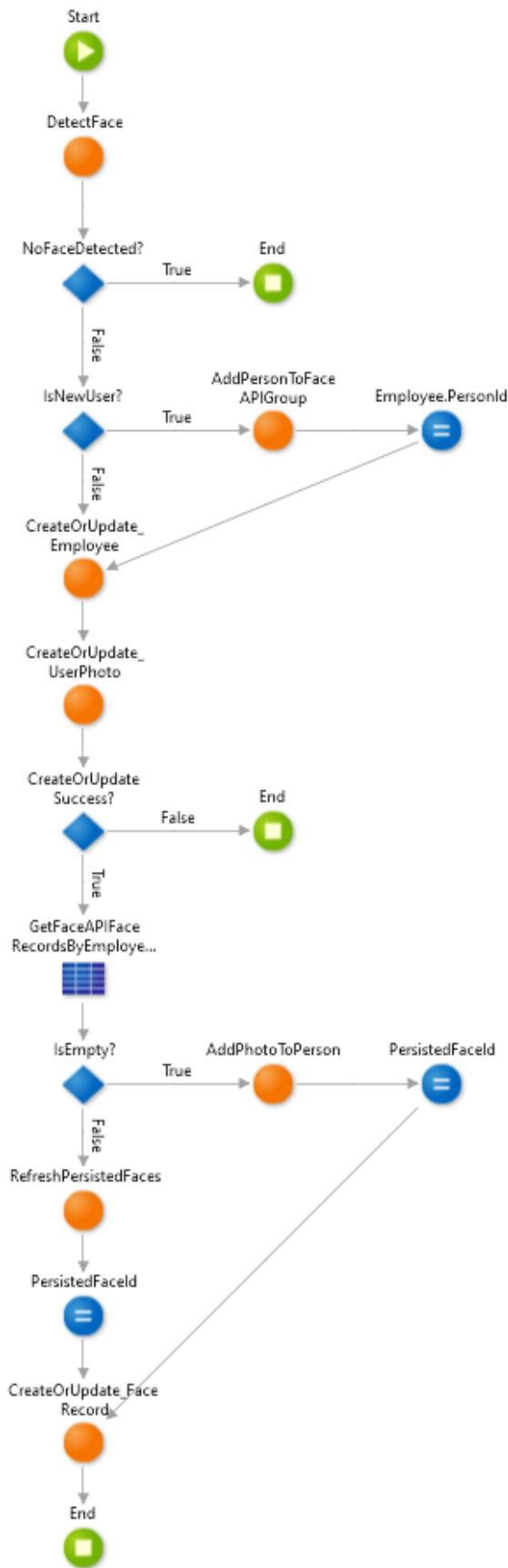


Figura 4.32: A *action* AddPersonToFaceAPIGroup contém a *action* PersonGroupPerson-Create enquanto as *actions* AddPhotoToPerson e RefreshPersistedFaces contém a *action* PersonGroupPersonAddPersonFace.

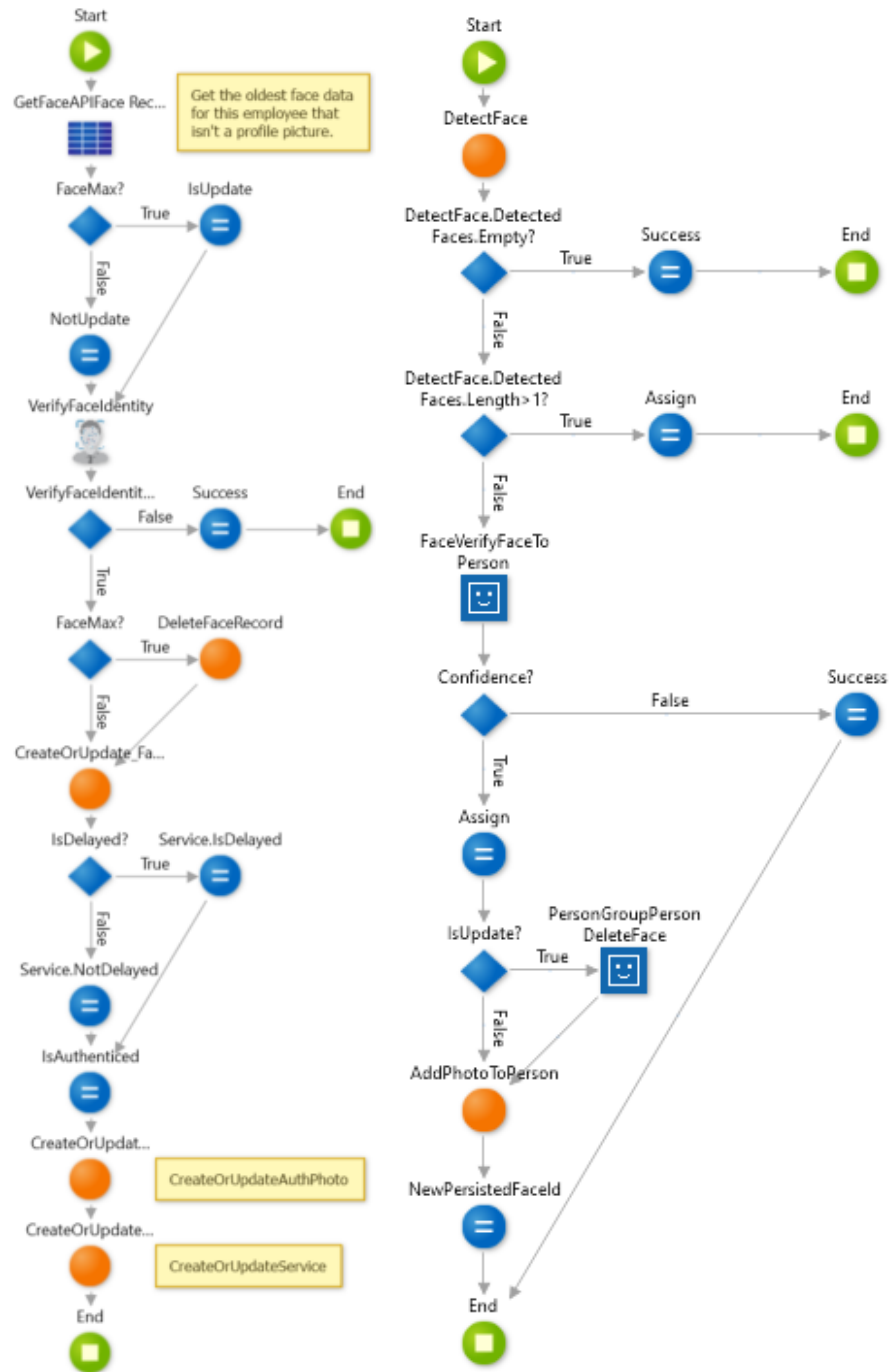


Figura 4.33: À esquerda está a gestão de FaceRecords do funcionário, e à direita o processo de autenticação.

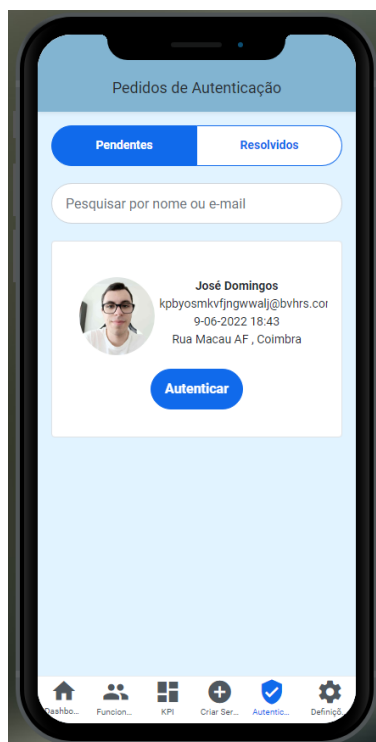


Figura 4.34: Ecrã de Autenticação Manual de Funcionários.

4.2.4 Notificações

A maior parte do tempo despendido na implementação das notificações foi empregue na configuração e integração dos *plugins* do OneSignal, processos já descritos na secção anterior. Tendo criadas as *actions* para envio de notificações, restou apenas incluí-las nas *actions* onde estas são relevantes, como a criação/edição de serviços, autenticação de funcionários, etc.

Assim, foi criado um ecrã para consulta das notificações de cada utilizador. Ao abrir a página inicial de cada utilizador, são carregadas as notificações do mesmo, e contabilizadas as notificações que ainda não foram lidas pelo utilizador. Este total é apresentado a flutuar sobre o ícone das notificações.

De seguida, ao abrir o ecrã das notificações são apresentadas as mais recentes e marcadas como lidas as que ainda se encontravam guardadas como notificações novas.

Para cada tipo de notificação e consoante as permissões do utilizador são definidas interações, como uma forma de atuação rápida. Por exemplo, para gerir recalendalizações, o cliente pode abrir o ecrã destinado a este fim, ou interagir diretamente com a notificação que recebe a esse respeito, como mostra a Figura 4.35.

Finalmente, foi definida uma ação para apagar uma notificação. Ao deslizar o dedo sobre uma notificação surge um ícone que ao ser clicado chama a *action* que apaga a notificação e atualiza a lista para que esta seja removida.

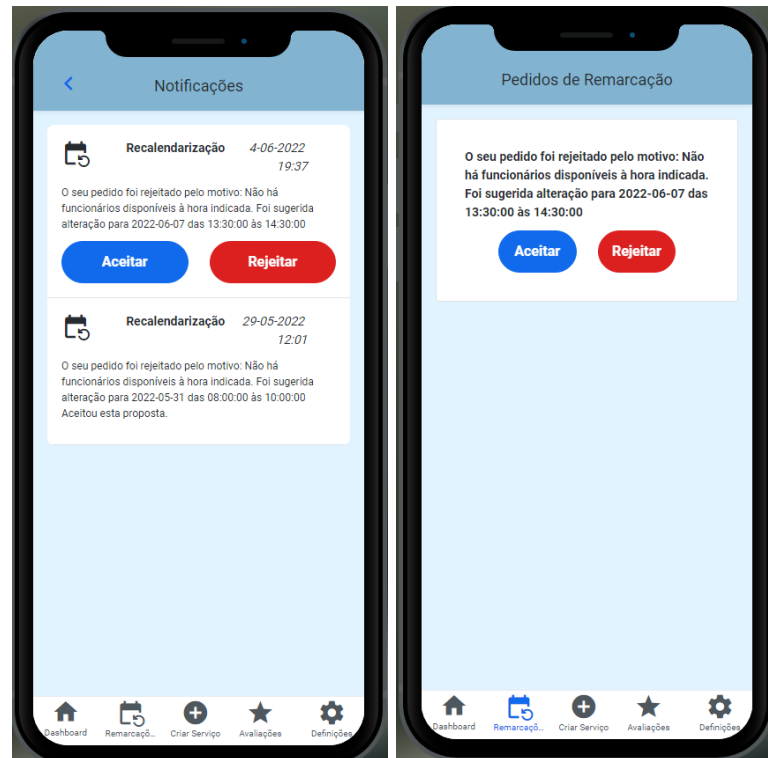


Figura 4.35: São apresentadas duas notificações de pedidos de recalendarização. O mais antigo já se encontra respondido, enquanto o mais recente ainda tem possibilidades de respostas. Note-se que o utilizador pode, alternativamente, usar o ecrã de recalendarizações para o mesmo efeito.

4.2.5 Finalização de Serviços

Depois de implementar a lógica de autenticação e das notificações foram implementadas algumas das funcionalidades da prestação de serviços com prioridade mais reduzida, nomeadamente o preenchimento do relatório e o preenchimento de um inquérito de qualidade por parte do cliente.

Inicialmente, o preenchimento do relatório era efetuado exclusivamente no final da prestação do serviço, sendo apresentado ao funcionário sob a forma de duas páginas. Este processo foi alterado na fase de desenvolvimento seguinte. Primeiramente, eram apresentadas algumas informações do serviço para confirmação, pelo que as informações a preencher pelo funcionário eram colocadas na segunda página, incluindo a recolha da assinatura do cliente.

Tendo preenchido o relatório, o cliente recebe uma notificação para preencher o inquérito de qualidade. Para a introdução das classificações, por se tratar de um componente que seria repetido várias vezes no mesmo ecrã, foi criando um bloco para este efeito, ou seja, uma componente de código facilmente replicável.

Deste modo, foi criado um bloco com um ícone de uma estrela, que se encontra preenchido consoante o valor atualmente selecionado pelo utilizador. Cada uma das 5 estrelas tem um valor associado, tendo a primeira um valor igual a 1 e a última um valor igual a 5. Ao ser clicada, cada estrela atualiza o valor global da pontuação do bloco, que por sua vez se reflete numa atualização visual de todas as estrelas, preenchendo os ícones com valor menor ou igual ao da estrela clicada pela última vez. O ecrã do inquérito de qualidade

encontra-se na Figura 4.36, onde cada uma das escalas corresponde a uma instância do bloco.

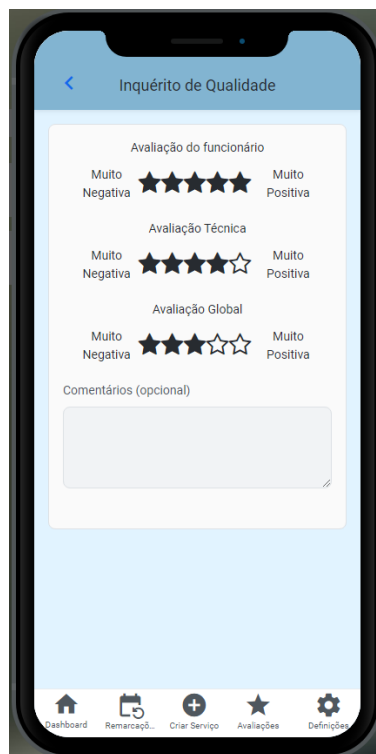


Figura 4.36: Ecrã de Inquérito de Qualidade

Melhorias de Usabilidade

Tendo implementado a maioria das funcionalidades, a única funcionalidade restante era a sincronização com os serviços do Google Calendar. Contudo, por se tratar de um requisito de baixa prioridade, considerou-se que a melhoria da usabilidade da aplicação traria mais valor à solução do que a implementação deste requisito. Assim, foi destinada uma fase para tornar mais claro o funcionamento de alguns ecrãs da aplicação, assim como dar algum destaque a algumas funcionalidades e torná-las mais visíveis.

Primeiramente, nos vários ecrãs de consulta de informações foram retirados os campos de edição caso as informações não fossem editáveis, como por exemplo, no preenchimento do relatório. Assim, o ecrã com campos a preencher pelo funcionário, referentes ao custo do serviço prestado, assim como uma listagem do material e das tarefas realizadas, é apresentado ao funcionário após a sua autenticação, de modo a poder preencher o relatório ao longo do decorrer do serviço. Tendo preenchido a informação necessária, o funcionário é levado a um ecrã onde poderá confirmar as informações do serviço, incluindo as que inseriu. No final deste ecrã é disponibilizado um campo para recolher a assinatura do cliente. Por fim, o funcionário pode terminar o serviço. O novo ecrã de relatório pode ser consultado na Figura 4.37.

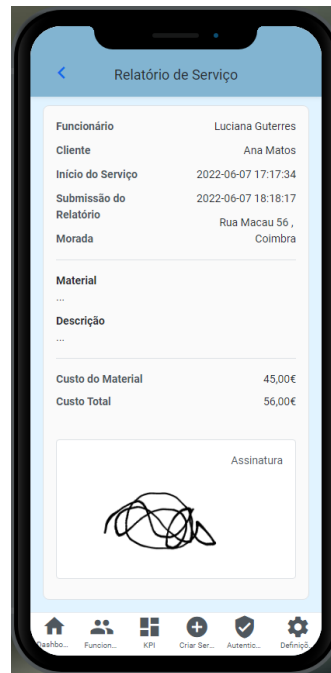


Figura 4.37: Versão final do ecrã de relatório.

De seguida, a navegação entre ecrãs passou a ser feita através de uma barra de ícones no fundo de todos os ecrãs. Além disso, a navegação entre perfis para utilizadores com várias funções passou a ser feita através do ícone da *dashboard*. Um simples toque no ícone remete o utilizador para a página inicial do perfil que está a visualizar, enquanto que um toque prolongado abre um *pop-up* para poder seleccionar o perfil para que pretende alterar, como mostra a Figura 4.38.

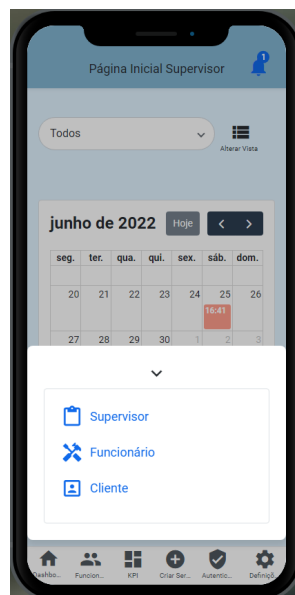


Figura 4.38: *Pop-up* de navegação entre *dashboard*.

Além disso, foi implementado um ecrã que permite ao supervisor visualizar índices de desempenho da empresa, como os serviços concluídos num dado intervalo de tempo, a distância total percorrida pelos funcionários para a prestação de serviços, a percentagem de serviços que são iniciados a horas, a percentagem de horas gastas em deslocações e na

prestação dos serviços, assim como a classificação média dos serviços.

Assim, de cada vez que o intervalo selecionado é alterado há uma atualização das informações carregadas da base de dados, para extração dos serviços e classificações inseridos no novo intervalo selecionado.

A Figura 4.39 representa a lógica executada sempre que há uma atualização destes valores.

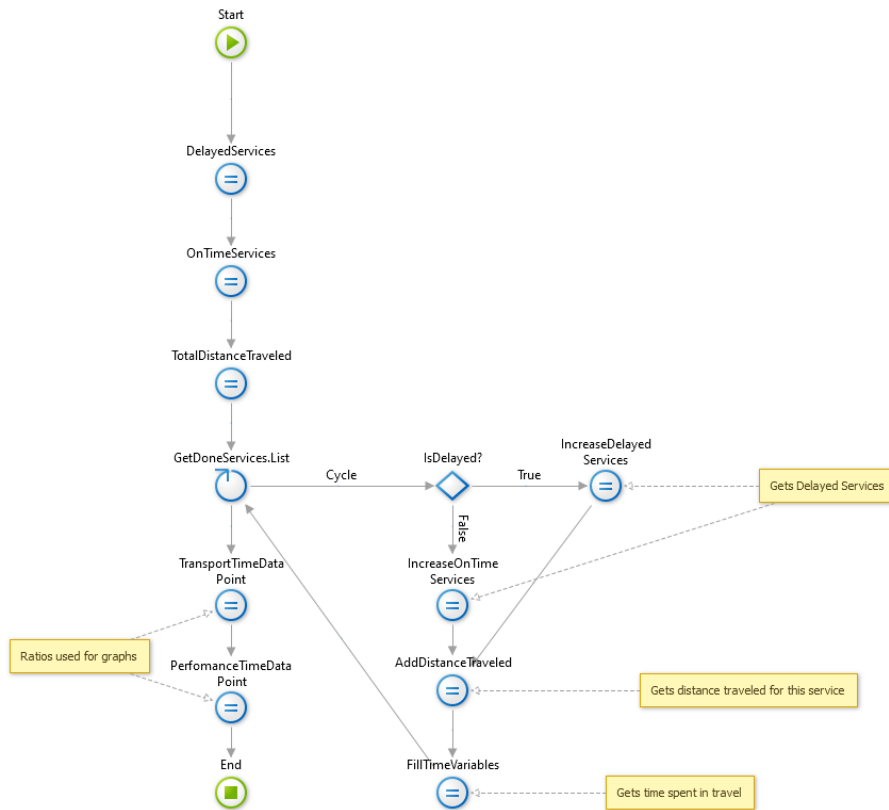


Figura 4.39: As informações dos serviços são reiniciadas a 0 e recontabilizadas para apresentar ao supervisor. No final, são calculadas as percentagens para os gráficos são atualizadas.

Para facilitar a implementação desta lógica, foram acrescentados parâmetros de distância percorrida e de pontualidade no serviço, que são preenchidos no momento de início de deslocamento e de autenticação do funcionário respetivamente. Para efeitos de pontualidade, considera-se que o serviço começou atrasado se a autenticação for feita passados 5 minutos da hora agendada.

Para finalizar este ecrã, foi incluído um mapa com as localizações de todos os funcionários e serviços agendados no intervalo definido, assim como um filtro para pesquisa pelo nome do funcionário ou morada do serviço.

Um outro melhoramento foi a inclusão de uma secção com as informações do próximo serviço agendado no *dashboard* do cliente. Assim, torna-se mais simples para o cliente visualizar informações sobre o estado do serviço, uma vez que não é necessária nenhuma interação com o calendário para consultar estas informações. Além disso, caso o funcionário esteja atrasado, essa informação é explicitamente comunicada ao cliente, uma vez que surge um alerta no topo da secção, como representado na Figura 4.40

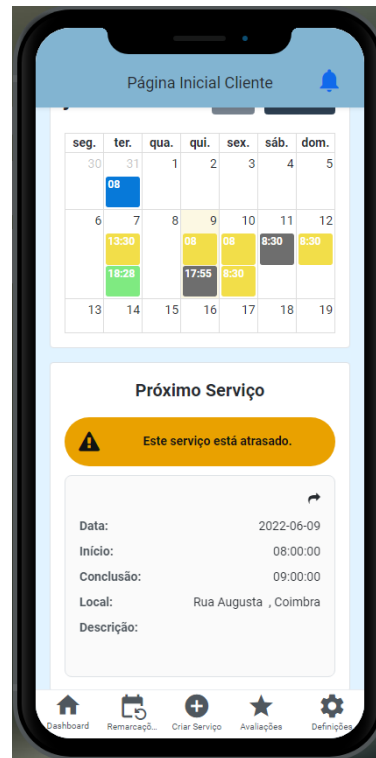


Figura 4.40: Alerta de Serviço Atrasado

Esta previsão é atualizada com frequência, uma vez que, quando a localização é atualizada do lado do funcionário, é também calculado o restante tempo previsto da deslocação. Caso este tempo exceda a hora agendada para o serviço, essa informação será apresentada ao cliente.

Outra alteração efetuada foi a adoção do mecanismo de bloqueio de funcionários, utilizado no momento de autenticação, para o caso do utilizador introduzir o seu código errado 3 vezes ao tentar editá-lo. Além disso, foi dada ao funcionário a opção de suspender a sua atividade, para o caso de se encontrar fora de serviço, como por exemplo, caso vá de férias. Deste modo, funcionários suspensos não serão sugeridos para atribuição de serviços por parte do supervisor.

Para facilitar a gestão destes dois estados de funcionários, foi adicionada uma secção para funcionários bloqueados e suspensos no ecrã de gestão de funcionários, acessível ao supervisor. Finalmente, foram efetuadas outras pequenas melhorias na disposição de alguns elementos nos vários ecrãs para tornar a aplicação mais simples de usar.

No final da fase de testes, depois de refletir sobre os resultados, foi implementada a opção de visualizar a foto utilizada para a autenticação do funcionário nas notificações do cliente, para que este possa também servir como agente de validação da autenticação.

Tendo concluído a implementação das funcionalidades definidas, deu-se início ao processo de teste das mesmas, cuja estratégia de avaliação, resultados e conclusões retiradas constam no capítulo seguinte.

Esta página foi intencionalmente deixada em branco

Capítulo 5

Testes

Este capítulo apresenta os testes que foram efetuados ao longo do projeto, inicialmente para fins de controlo de qualidade e dando lugar posteriormente a testes para a validação da solução, onde foram recolhidas informações para uma subsequente análise da utilidade da mesma. As etapas dos testes realizados encontram-se representadas na Figura 5.1.

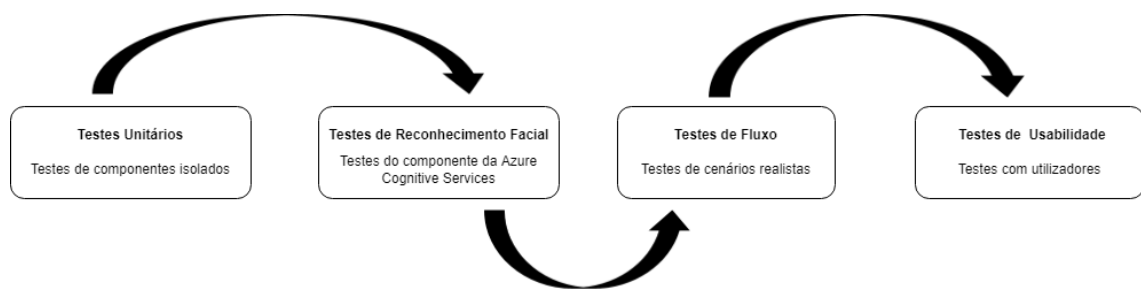


Figura 5.1: Estratégia de teste da solução.

5.1 Testes Unitários

No decorrer do projeto, foram realizados testes unitários de forma paralela com o desenvolvimento da aplicação e à medida que as funcionalidades foram implementadas, foram também testadas.

5.2 Testes de Fluxo

Ao terminar o desenvolvimento da solução foram efetuados testes com o objetivo de testar as funcionalidades implementadas em cenários mais complexos, envolvendo várias ações segundo fluxos de execução previamente definidos. Estes testes permitiram a simulação de casos de uso mais reais e conferir se a aplicação se encontrava funcional como um todo.

Na Figura 5.2 encontra-se um dos fluxos definidos para este tipo de testes, constando os restantes na secção de Apêndices C.

Registo e remoção de funcionários

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Navegar para o ecrã de gestão de funcionários	1. Selecionar o ícone "Gerir Funcionários" na <i>bottom bar</i> .	1. É apresentado o ecrã de gestão de funcionários.	
Gestão de Funcionários			
Navegar para o ecrã de registo de funcionários	1. Selecionar o ícone "+" no canto superior direito.	1. É apresentado o ecrã de registo de funcionários	
Registo de Funcionários			
Registo correto de funcionários	1. Preenchimento correto dos campos requeridos (e-mail, telemóvel e nome válidos); 2. Seleção da opção "Registar".	1. O utilizador regressa ao ecrã de gestão de funcionários; 2. É apresentada uma mensagem de sucesso referente ao registo.	
Gestão de Funcionários			
Utilização do filtro para pesquisa de funcionários	1. Preenchimento da barra de pesquisa com o nome do funcionário a remover; 2. Preenchimento da barra de pesquisa com o e-mail do funcionário a remover.	1. O perfil do funcionário surge como resultado das duas pesquisas, assim como a sua foto de perfil e a sua classificação média.	
Navegação para o ecrã de informações do funcionário	1. Seleção, através do toque, do funcionário em questão.	1. O ecrã que contém as informações do funcionário é apresentado.	
Informação de Funcionários			
Acionar o <i>popup</i> de remoção do funcionário	1. Seleção da opção "Remover Funcionário".	1. Surge um <i>popup</i> de confirmação.	
Remoção do funcionário	1. Seleção da opção "Remover Funcionário".	1. O utilizador é devolvido à página de gestão de funcionários; 2. O funcionário já não se encontra visível na lista.	
Navegar para o ecrã de funcionários inativos	1. Selecionar o separador "Funcionários Bloqueados".	1. É apresentada a lista de funcionários bloqueados.	
Restaurar acesso a funcionários banidos	1. Percorrer a lista de funcionários bloqueados até localizar o que se pretende; 2. Seleção da opção "Restaurar".	1. É apresentada uma mensagem de sucesso; 2. O funcionário é removido da lista.	
Visualização de pedidos resolvidos	1. Selecionar a opção de funcionários inativos através do <i>switch</i> no topo do ecrã; 2. Utilização do filtro para encontrar o funcionário em questão.	1. É apresentada a lista de funcionários com conta suspensa; 2. É apresentado o perfil do funcionário conforme especificado na barra de pesquisa.	

Figura 5.2: Exemplo de um teste de fluxo.

5.3 Testes de Reconhecimento Facial

Estes testes tiveram como objetivo averiguar a confiabilidade e robusteza do mecanismo de reconhecimento facial, ou seja, se era flexível nos critérios de luminosidade, qualidade de imagem, expressões faciais, etc, para ser utilizável em contextos reais, e se era simultaneamente seguro o suficiente para não ser manipulado.

Para tal, foi temporariamente eliminada a condição definida na secção de desenvolvimento, que implica que correspondências positivas tivessem necessariamente um nível de confiança superior a 60%. Deste modo foi possível averiguar a necessidade desta condição, e simultaneamente considerar se um outro limiar seria mais vantajoso.

Assim, os primeiros testes foram efetuados com ligeiras variações na posição da cara e acessórios, dos quais se apresentam alguns exemplos na Figura 5.3.

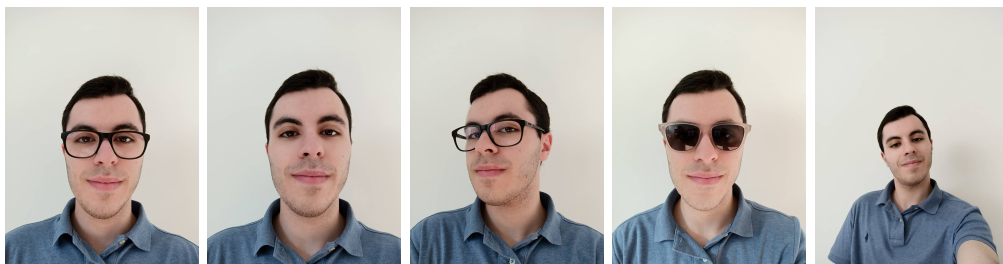


Figura 5.3: Alguns dos testes mais simples do reconhecimento facial. O nível de confiança mais alto atingido foi de 0.97535, enquanto o mais baixo foi de 0.72881.

De seguida foram testados alguns casos mais extremos, como por exemplo, acessórios com maior área de obstrução, rotações mais acentuadas da cabeça, ou expressões faciais mais pronunciadas. Apesar da existência de alguns sucessos, a maioria destes testes resultaram em fracassos na autenticação, alguns dos quais se encontram na Figura 5.4.

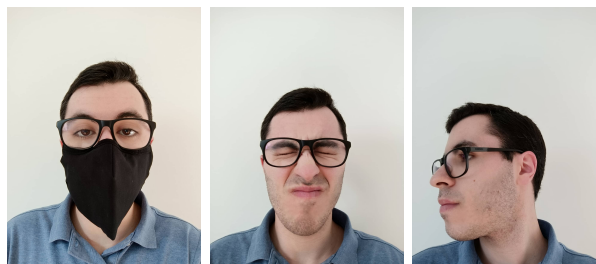


Figura 5.4: Fracassos do reconhecimento facial.

No geral, os fracassos nestes testes corresponderam às expectativas, uma vez que são situações limite, não esperadas de utilizadores reais.

Num contexto de utilização real, as condições de luminosidade e qualidade da imagem são, muitas vezes, variáveis, pelo que se considerou relevante testar estas situações. A maioria dos cenários verificou resultados positivos, apresentados na Figura 5.5, enquanto casos mais extremos apresentaram fracassos, visíveis na Figura 5.6.



Figura 5.5: O nível de confiança mais alto atingido foi de 0.76601, enquanto o mais baixo foi de 0.66390.

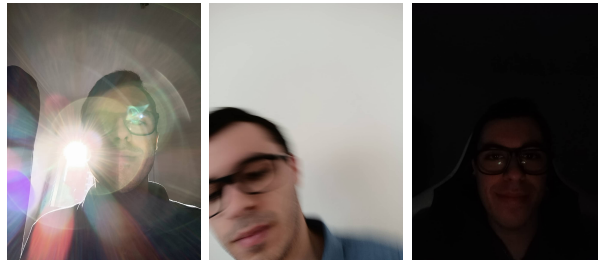


Figura 5.6: Testes da qualidade de imagem - Fracassos.

Face a estes resultados, o reconhecimento facial apresentou um nível de flexibilidade satisfatório, uma vez que a esmagadora maioria dos casos em que a autenticação fracassou envolveu o uso de fotos que não seguem as recomendações da Azure, enumeradas em [15], das quais se destacam:

1. Imagens com iluminação extrema, por exemplo, luz de fundo severa;
2. Obstruções que bloqueiam um ou ambos os olhos;
3. Expressões faciais extremas.

Finalmente, foi testado o reconhecimento facial com recurso a fotos (atuais e antigas). Para as fotos atuais, a autenticação foi bem sucedida, enquanto nas fotos mais antigas, o efeito do envelhecimento foi suficiente para a autenticação falhar. Foi observado apenas um caso com sucesso utilizando uma foto antiga, com um nível de confiança de 0.53428. Tentativas subsequentes com a mesma foto foram bem sucedidas, com um nível de confiança maior ao longo das tentativas. Isto porque para cada tentativa bem sucedida, é utilizada a *action* *PersonGroupTrain*, ou seja, o sistema é treinado com a nova foto registada. Deste modo, para proteger contra esta vulnerabilidade, o limiar para o nível de significância de 60% foi reimplementado. A partir desse momento, não foram detetadas novas autenticações com fotos antigas.

Contudo, apesar do reconhecimento facial apresentar alguma vulnerabilidade no que toca ao uso de fotografias, existem algumas melhorias possíveis para contornar este problema. Por um lado, poderia ser explorada a utilização de processamento de vídeo através da *Azure Cognitive Services*, que apesar de implicar um ritmo de utilização da API não suportado pela licença utilizada para o desenvolvimento do projeto, poderia conferir uma maior robustez ao sistema, uma vez que seriam possíveis várias medições dos traços faciais detetados. Além disso, poderia ser explorada a utilização de sistemas complementares ao reconhecimento facial, como por exemplo, a projeção de pontos utilizada atualmente por dispositivos móveis da Apple. Através da projeção de pontos infravermelhos, apresentada

na Figura 5.7 e de sensores de distancia, é possível fazer um mapeamento da cara do utilizador, que permite ao dispositivo averiguar se se trata de um objeto tridimensional.

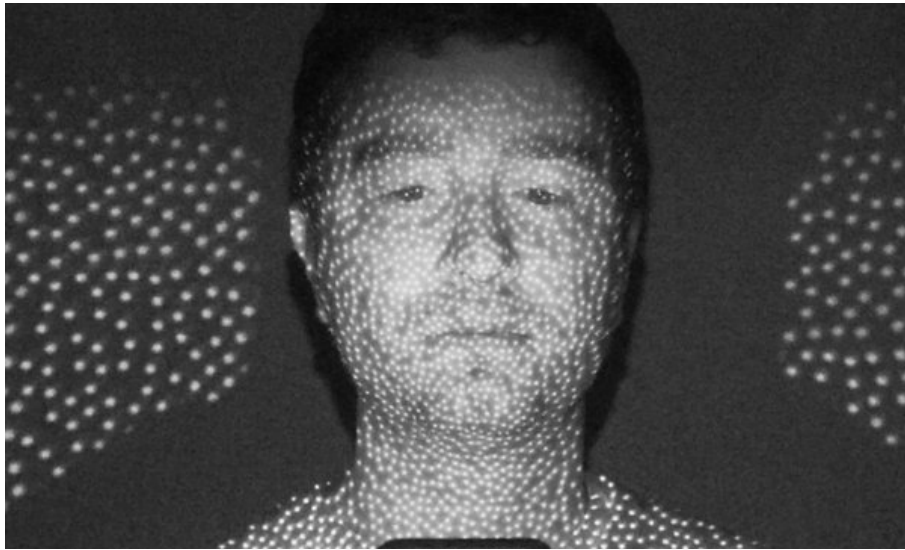


Figura 5.7: Projeção de pontos infravermelhos [34].

Finalmente, foi medido, em milissegundos, o tempo de autenticação para os casos em que esta foi bem sucedida, apresentados na Tabela 5.1.

797	719	703	641	656	800	636	718	734	672	764
781	766	672	860	672	657	625	718	8218	672	712
735	718	673	703	703	687	782	703	744	735	657

Tabela 5.1: Medições do tempo de autenticação do funcionário em milissegundos.

Como é evidente pela tabela, existe um valor que pode ser descartado por ser considerado um *outlier*, que é a medição correspondente a 8218ms. Deste modo, sem incluir este valor, obtém-se o valor médio de aproximadamente 713ms, assim como a distribuição das medições, representada na Figura 5.8.

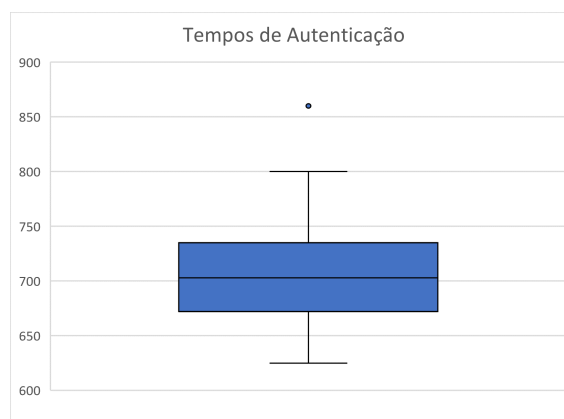


Figura 5.8: Distribuição das medições do tempo de reconhecimento facial.

5.4 Testes de Usabilidade

Na última fase dos testes, foram realizados testes de aceitação, em que as funcionalidades da aplicação foram testadas por utilizadores fora do âmbito do projeto. A realização dos testes consistiu na execução de tarefas correspondentes aos casos de uso delineados, assim como o subsequente preenchimento de um formulário de satisfação. Cada um dos formulários é constituído por dois tipos de questões, de escolha múltipla e de resposta aberta. Para as questões de escolha múltipla foram propostas afirmações sobre a usabilidade da aplicação, às quais os utilizadores responderam numa escala de concordância compreendida de 1 a 5. Deste modo, a escala consistiu nos níveis Discordo Totalmente, Discordo, Não Concordo Nem Discordo (NCND), Concordo e Concordo Totalmente.

Deste modo, foram realizados testes com 21 indivíduos externos ao projeto, com a maior faixa etária representada entre os 20 e os 25 anos, estudantes de Engenharia Informática e de Design. Não obstante, foram também realizados testes com elementos de faixas etárias mais elevadas, incluindo dos 50 aos 60 anos de idade. Todos os inquéritos realizados posteriormente aos testes foram considerados válidos, e podem ser consultados na secção Apêndices D, assim como resultados observados na secção Apêndices E .

5.4.1 Discussão de Resultados

As avaliações dos utilizadores são, no geral, bastante positivas, uma vez que a maioria das classificações são do nível 5, e todas são dos níveis 4 ou 5, que correspondem às classificações positivas.

Posto isto, no caso do cliente, a questão com pior classificação é a da localização do funcionário, uma vez que o mapa do cliente só mostra a sua localização. Uma possível melhoria seria a opção de ver o trajeto que resta fazer ao funcionário, tal como apresentado a este tipo de utilizadores nos detalhes do serviço que estão a prestar. Finalmente, as questões de resposta aberta foram também positivas, apresentando uma sugestão interessante para melhoria da segurança da aplicação, sob a forma de apresentação da foto utilizada para autenticação do funcionário. Deste modo, no caso de haver uma tentativa de fraude, por exemplo, através de uma foto impressa, seria fácil para o cliente detetar esta situação. Por representar uma funcionalidade simples e contribuir na vertente da segurança da aplicação, esta funcionalidade foi implementada posteriormente à realização dos testes.

No caso dos funcionários, a questão com a classificação mais baixa passa pela progressão do estado dos serviços, pelo que poderia ser incluído um identificador visual nos detalhes do serviço, que seria atualizado sempre que este fosse alterado. Quanto às questões de resposta aberta, as respostas foram, também, positivas, incluindo funcionalidades que podem ser úteis numa versão futura, como por exemplo, uma forma de contactar o supervisor diretamente a partir da aplicação.

Finalmente, para o supervisor, as classificações mais baixas foram registadas na questão de alternância do modo de visualização da *dashboard*, entre a lista de serviços e o calendário. Isto porque não foi imediatamente claro que o calendário apresenta os serviços no estado selecionado na *dropdown* no topo do ecrã. Deste modo, poderá ser interessante remover esta funcionalidade na visualização do calendário, reservando este modo de visualização para uma visão de todos os serviços agendados.

Com estes resultados em mente, pode dar-se como concluído o desenvolvimento da solução, restando por fim retirar as conclusões do projeto, que constam no capítulo que se segue.

Capítulo 6

Conclusão

Este capítulo tem como objetivo uma reflexão sobre o produto final desenvolvido, melhorias e outras funcionalidades que poderiam ser implementadas numa fase futura, assim como a satisfação dos objetivos definidos no início do estágio.

6.1 Reflexão

No início do projeto de estágio, foram definidas questões de investigação às quais se pretendia responder, assim como requisitos gerais da aplicação a desenvolver, dos quais se salienta a utilização de tecnologias de reconhecimento facial numa aplicação de apoio à prestação de serviços ou assistência técnica. De seguida, foi realizada uma investigação do funcionamento de algumas técnicas de reconhecimento facial, assim como as tecnologias e ferramentas mais utilizadas na área. Desta pesquisa, resultou um maior entendimento das oportunidades no domínio da aplicação, pelo que foi possível definir requisitos funcionais mais específicos para a solução desenvolvida, culminado na sua implementação e subsequente avaliação.

Tendo chegado ao final do estágio, é necessário refletir sobre os objetivos estabelecidos para o mesmo e avaliar se estes foram cumpridos com sucesso. Para tal, devem ser retomadas as questões de investigação definidas inicialmente.

Primeiramente, foi levantada a questão “Quais as funcionalidades que conferem maior segurança na prestação dos serviços solicitados, assim como uma facilitada gestão dos mesmos?”. Esta questão foi respondida ao longo do decorrer do estágio, principalmente na etapa de definição de requisitos funcionais. De facto, muitas das conclusões retiradas desta fase foram refletidas na solução desenvolvida, como é evidente pelos mecanismos de autenticação, quer por reconhecimento facial e localização, quer pelas suas alternativas. É também importante lembrar que as funcionalidades implementadas foram selecionadas com base na sua utilidade para a prestação de serviços, sendo esta vertente igualmente importante quando comparada com a autenticação de funcionários. Foi nesta medida que se selecionaram componentes e ferramentas como por exemplo, o suporte de calendário e notificações.

De seguida, a segunda pergunta - “Quais são as ferramentas ou serviços cognitivos de reconhecimento facial mais adequados para o desenvolvimento da aplicação?” - foi respondida principalmente durante a etapa de pesquisa para o estado da arte das soluções de reconhecimento facial que existem, pelo que a própria comparação destas tecnologias ocupou uma

secção com particular destaque no segundo capítulo, correspondente ao estado da arte. Deste modo, as conclusões retiradas desta questão foram particularmente relevantes para as escolhas feitas nas tecnologias de reconhecimento facial utilizadas, e corroboradas pelos resultados observados na fase de testes. Não obstante, tal como observado nessa mesma fase, e em concordância com a secção imediatamente anterior, podiam ter sido ainda estudadas algumas funcionalidades da tecnologia da Azure para melhorar ainda mais o seu desempenho. No entanto, considera-se que os resultados obtidos atualmente são satisfatórios.

Quanto à terceira questão - “Qual o valor que a nova solução poderá trazer para os seus utilizadores?” - a resposta pode ser extraída diretamente dos resultados da fase de testes. As classificações foram positivas em todos os aspetos, com os aspetos de conveniência e segurança destacados nas questões de resposta aberta. Deste modo, pode concluir-se que o valor da solução se prende com a simplificação do processo de marcação de serviços, reduzindo a interação entre as várias partes envolvidas e, deste modo, o tempo despendido para a requisição de serviços. Por outro lado, o aumento da segurança foi também destacado com atributo de valor da aplicação, particularmente positivo por não ter apresentado um transtorno significativo para a prestação de serviços por parte da empresa.

6.2 Trabalho Futuro

Apesar dos requisitos identificados no início do projeto terem sido cumpridos praticamente na sua totalidade, um destes ficou por implementar, correspondendo à implementação da sincronização da aplicação com os serviços do Google Calendar. Este requisito foi abandonado em prol de várias melhorias de usabilidade da aplicação, por um lado, por se considerar que estas trariam maior valor para a aplicação, e por outro por este requisito apresentar a prioridade mais baixa de todos os requisitos definidos. Contudo, numa fase mais avançada, seria interessantes explorar a possibilidade de incluir esta funcionalidade da solução desenvolvida.

Outro aspeto que poderia ser explorado prende-se com os serviços da Azure, nomeadamente o processamento de vídeo. Tal como mencionado no capítulo anterior, é possível que o sistema funcione com fotos impressas dos funcionários, pelo que o processamento de vídeo poderia ser explorado como forma de aumentar ainda mais a segurança do sistema. Contudo, a utilização desta e outras funcionalidades está dependente da adoção de licenças de utilização mais caras, razão pela qual não foram exploradas durante este projeto. Ainda assim, poderia ser relevante explorar em que medida estas funcionalidades melhorariam a qualidade da aplicação.

Finalmente, na fase de testes surgiram algumas sugestões, da parte dos utilizadores, para melhoria da aplicação, das quais se destacam as seguintes:

- Definição de vários tipos de serviço para seleção no momento de criação do serviço. Esta funcionalidade seria bastante útil no caso da aplicação ser adotada por uma empresa em concreto, de modo a poder definir o tipo de serviços a serem prestados, pelo que sua implementação estaria dependente da especificação do contexto de utilização;
- Maior customização do calendário, permitindo a visualização de eventos que não correspondem a serviços. Esta sugestão vai ao encontro da implementação da sincronização com os serviços da Google Calendar discutida anteriormente;

- Possibilidade de entrar em contacto com o supervisor diretamente a partir da aplicação (quer por chamada, quer por via de um sistema de mensagens, como por exemplo, um **chat**);

6.3 Apreciações Finais

A realização deste estágio e a resposta às perguntas de investigação representaram uma grande experiência de aprendizagem, não só sobre os fundamentos do reconhecimento facial, como também no processo de desenvolvimento de projetos no contexto profissional de uma empresa.

Finalmente, pode considerar-se que foi desenvolvido um produto final que incorporou as funcionalidades identificadas, satisfazendo os casos de uso definidos na fase inicial, e com valor confirmado por utilizadores externos. Além disso, esta solução, desenvolvida em ambiente de desenvolvimento *low-code*, com integração de várias tecnologias, incluindo de reconhecimento facial, exigiu uma compreensão do funcionamento destas tecnologias, assim como o atual estado de mercado de soluções semelhantes. Deste modo, podem ser dados como atingidos os objetivos definidos no início do estágio.

Esta página foi intencionalmente deixada em branco

Referências

- [1] Comissão Europeia. Comunicação da comissão - uma nova estratégia industrial para a europa. <https://eur-lex.europa.eu/legal-content/PT/TXT/?uri=CELEX:52020DC0102>, November 2021.
- [2] Alan Hevner, Salvatore March, Jinsoo Park, and Sudha Ram. Design science research in information systems. 2004.
- [3] OutSystems. Agile and Scrum: Understanding the differences. <https://www.outsystems.com/blog/posts/agile-and-scrum/>, January 2022.
- [4] IBM. What is computer vision? <https://www.ibm.com/topics/computer-vision>, June 2022.
- [5] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. Accepted Conference on Computer Vision and Pattern Recognition, 2001.
- [6] Yicheng An, Jiafu Wu, and Chang Yue. CNNs for face detection and recognition. IEEE 12th International Conference on Automatic Face & Gesture Recognition, 2017.
- [7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.
- [8] Danfeng Xie, Lei Zhang, and Li Bai. Deep learning in visual computing and signal processing. 2017.
- [9] Damian Podareanu, Valeriu Bogdan Codreanu, Sandra Aigner, Caspar van Leeuwen, and Volker Weinberg. Best practice guide - Deep Learning. 2019.
- [10] Huaizu Jiang and Erik Learned-Miller. Face detection with the Faster R-CNN. IEEE 12th International Conference on Automatic Face & Gesture Recognition, 2017.
- [11] Ross Girshick. Fast R-CNN. Accepted Conference on Computer Vision and Pattern Recognition, 2015.
- [12] Paola Campadelli, Raffaella Lanzarotti, and Giuseppe Lipori. Automatic facial feature extraction for face recognition. Università degli Studi di Milano, Italy, 2007.
- [13] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. IEEE 12th International Conference on Automatic Face & Gesture Recognition, 2017.
- [14] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. University of Chinese Academy of Sciences, 2017.

- [15] Microsoft Azure Cognitive Services. Face detection and attributes. <https://docs.microsoft.com/en-us/azure/cognitive-services/face/concepts/face-detection>, November 2021.
- [16] Microsoft Azure Cognitive Services. What is the azure face service? <https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview>, November 2021.
- [17] Amazon. Overview of face detection and face comparison. <https://docs.aws.amazon.com/rekognition/latest/dg/face-feature-differences.html>, November 2021.
- [18] Amazon. Amazon simple storage service. <https://aws.amazon.com/pt/s3/?nc=sn&loc=0>, June 2022.
- [19] Amazon. Detecting faces in an image. <https://docs.aws.amazon.com/rekognition/latest/dg/faces-detect-images.html>, November 2021.
- [20] Amazon. Comparing faces in images. <https://docs.aws.amazon.com/rekognition/latest/dg/faces-comparefaces.html>, November 2021.
- [21] Amazon. Compare faces in images (console). <https://docs.aws.amazon.com/rekognition/latest/dg/compare-faces-console.html>, November 2021.
- [22] Microsoft Azure. Face similarity API. <https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395237>, February 2022.
- [23] Microsoft Azure. Face identification API. <https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395239>, February 2022.
- [24] Microsoft Azure. Face API pricing. <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/>, January 2022.
- [25] Amazon. Amazon rekognition API pricing. <https://aws.amazon.com/pt/rekognition/pricing/>, January 2022.
- [26] Veriff. Veriff face matching. <https://www.veriff.com/product/face-match>, February 2022.
- [27] Jumio. Jumio identity verification. <https://www.jumio.com/products/identity-verification/>, February 2022.
- [28] Jason Wong, Kimihiko Iijima, Adrian Leow, Akash Jain, and Paul Vincent. Magic quadrant for enterprise low-code application platforms. Gartner, 2021.
- [29] John Bratincevic, Rob Koplowitz, Stephen Powers, Sara Sjoblom, and Kara Hartig. The forrester wave™: Low-code development platforms for professional developers, Q2 2021. Forrester, 2021.
- [30] OutSystems. The architecture canvas. https://success.outsystems.com/Support/Enterprise_Customers/Maintenance_and_Operations/Designing_the_Architecture_of_Your_OutSystems_Applications/The_Architecture_Canvas, November 2021.
- [31] OutSystems. Best practices on architecture design. <https://www.outsystems.com/training/lesson/2072/best-practices-on-architecture-design>, June 2022.

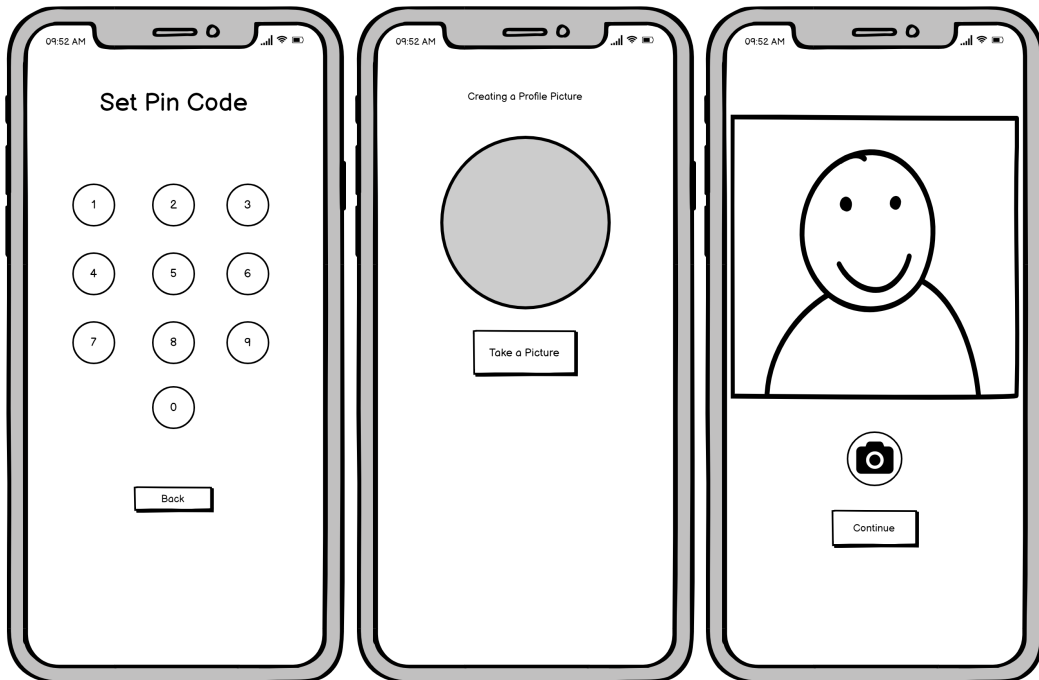
- [32] OutSystems. The low-code development guide. <https://www.outsystems.com/guide/low-code/>, November 2021.
- [33] OutSystems. Service studio overview. https://success.outsystems.com/Documentation/11/Getting_started/Service_Studio_Overview, January 2022.
- [34] Projeção de pontos infravermelhos. <https://www.quora.com/Can-Apple-Face-ID-be-fooled-by-a-photo>, 2019.

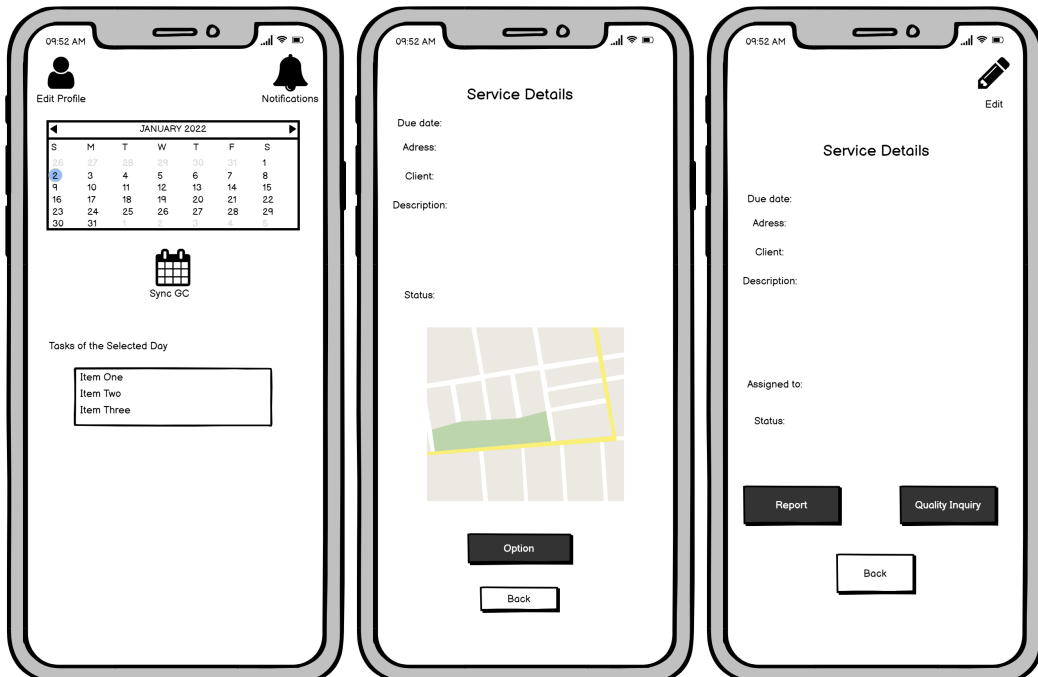
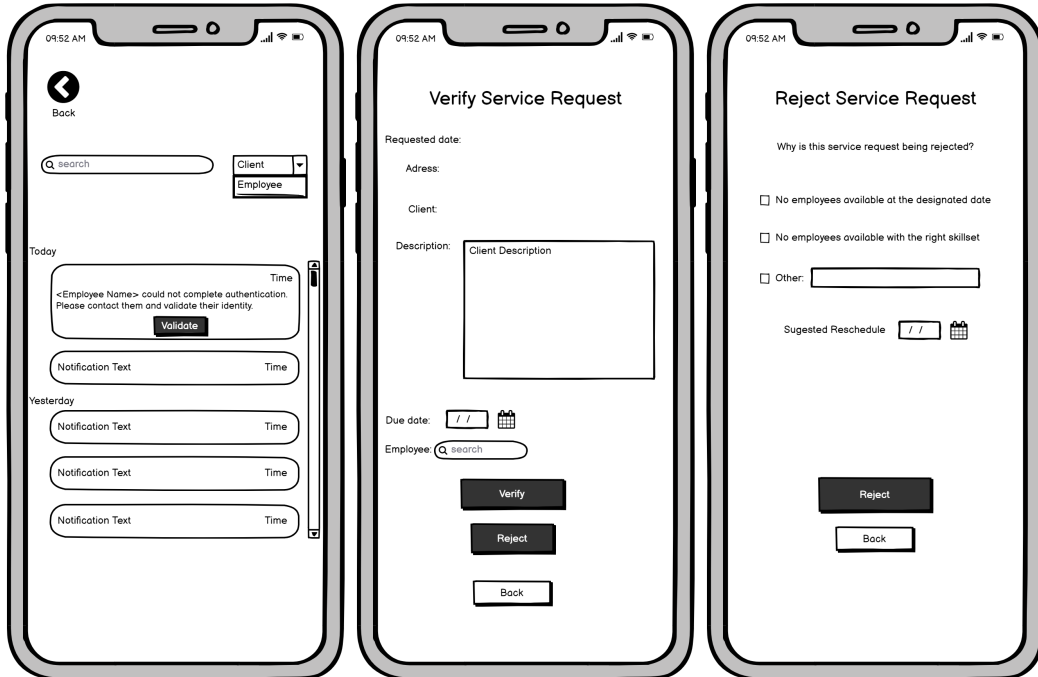
Esta página foi intencionalmente deixada em branco

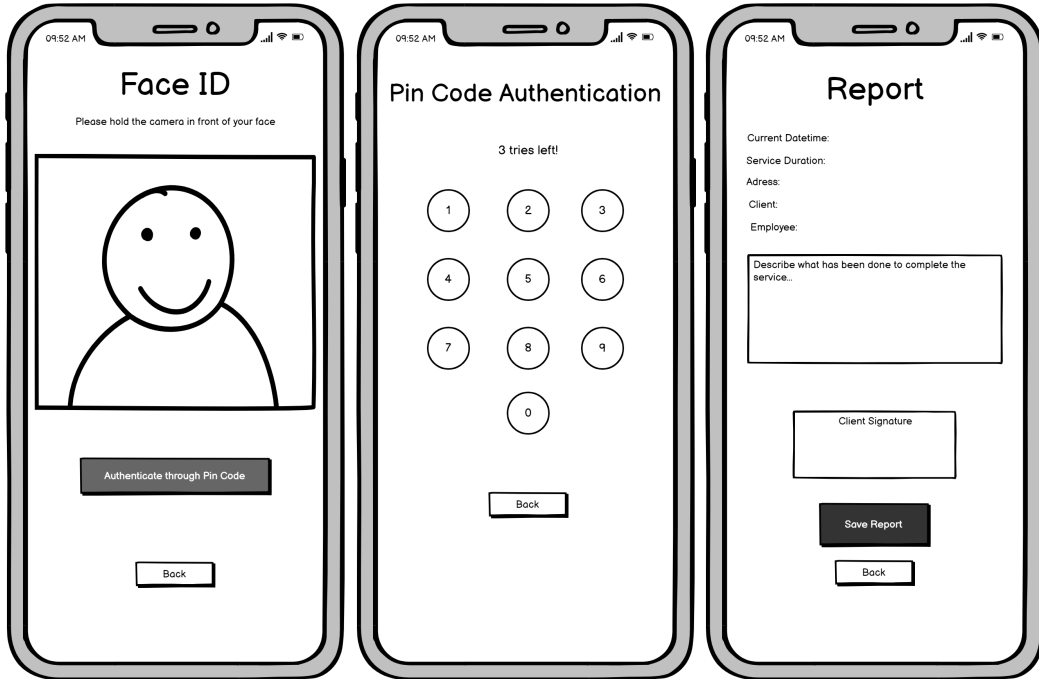
Apêndice A

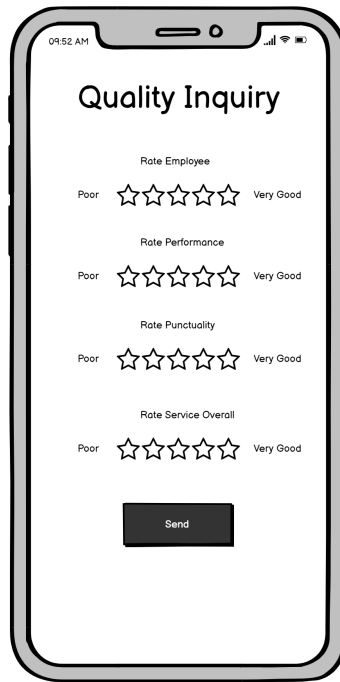
Protótipos







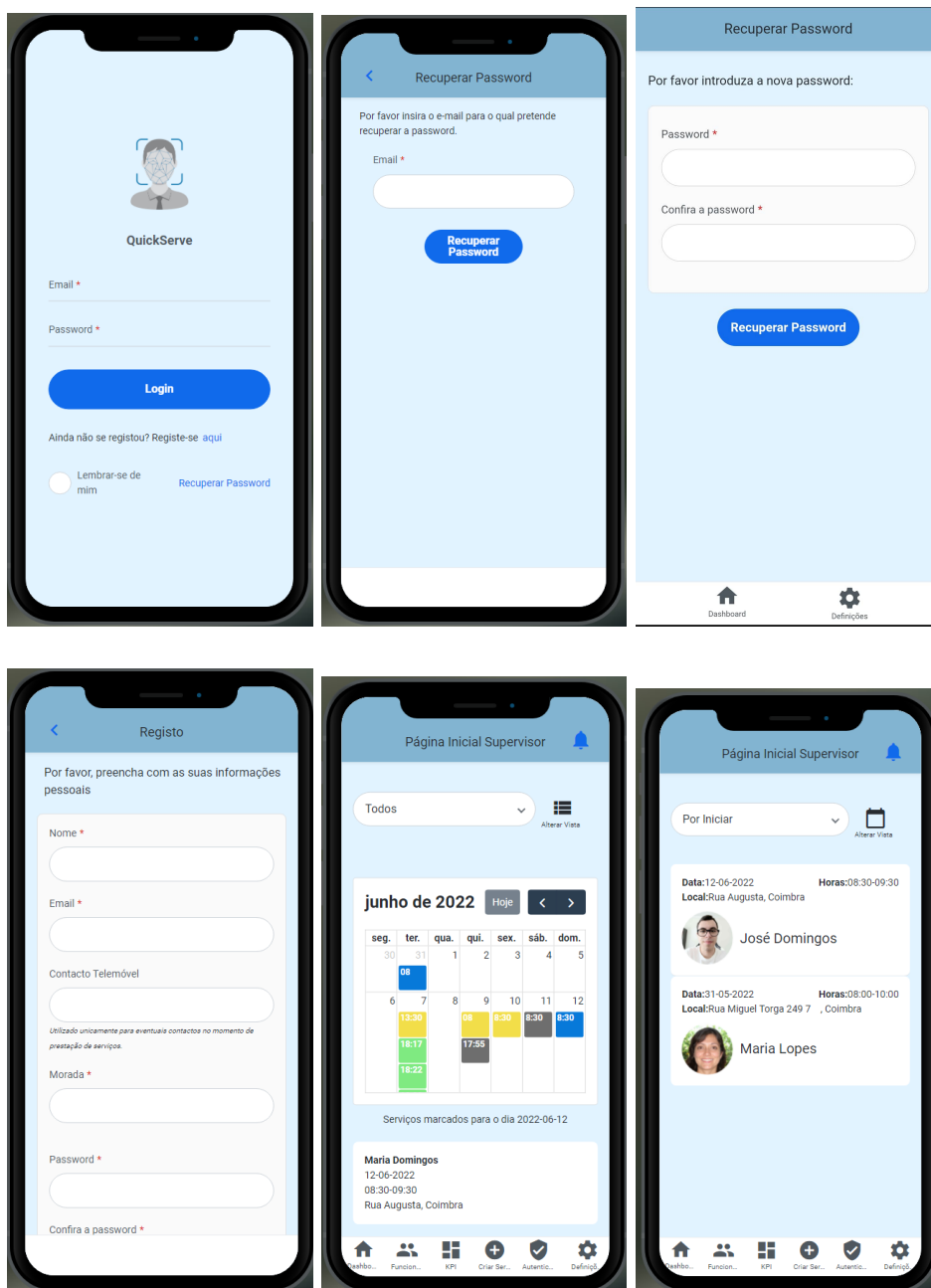


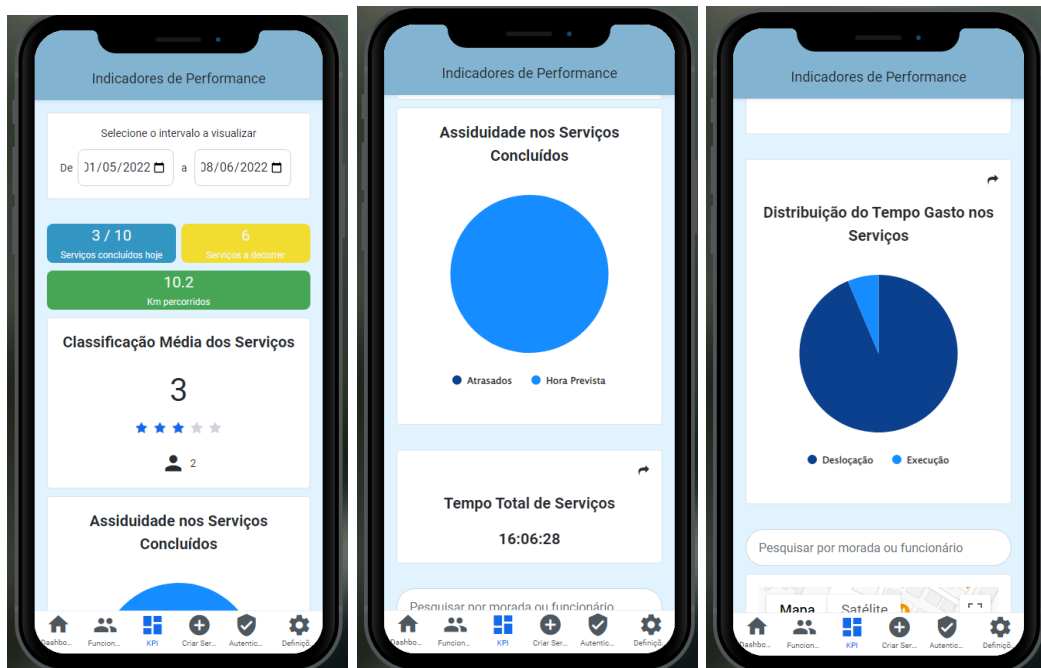
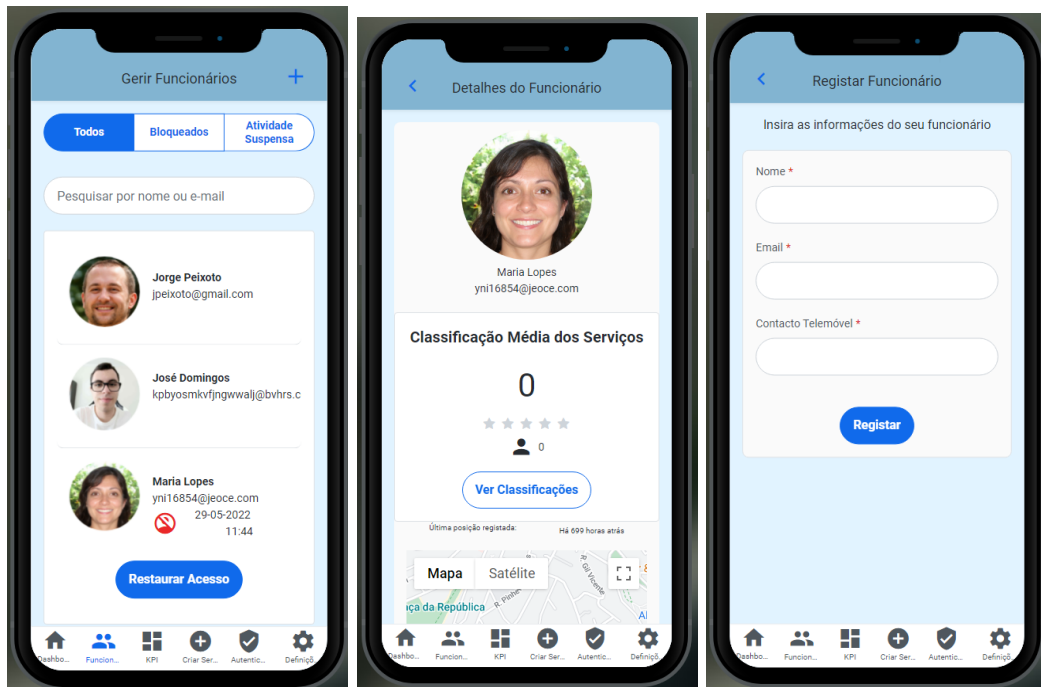


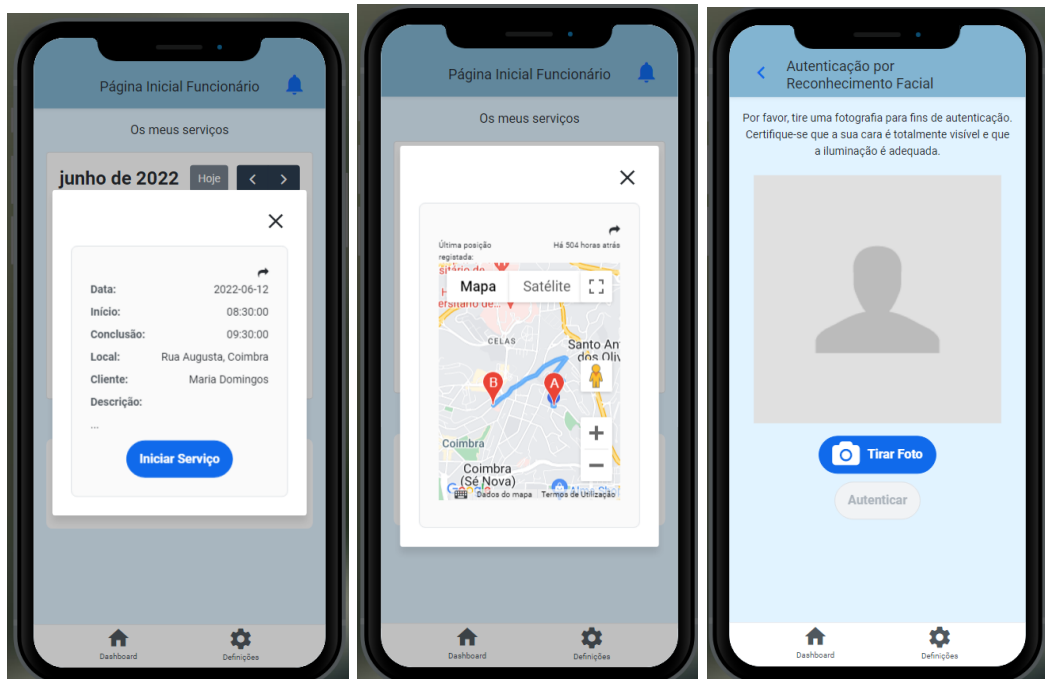
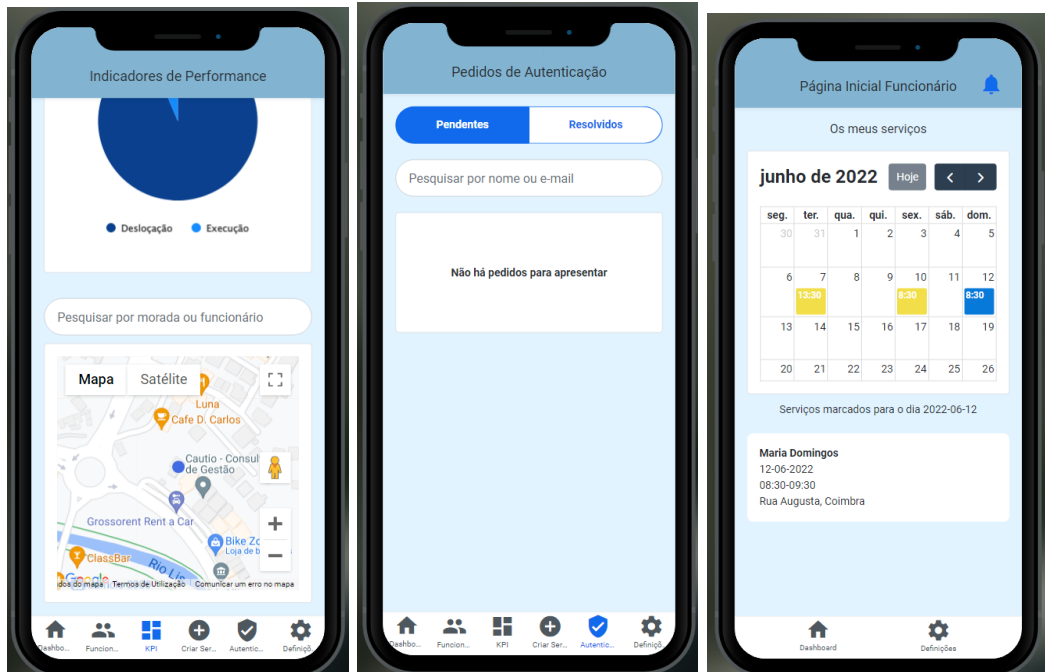
Esta página foi intencionalmente deixada em branco

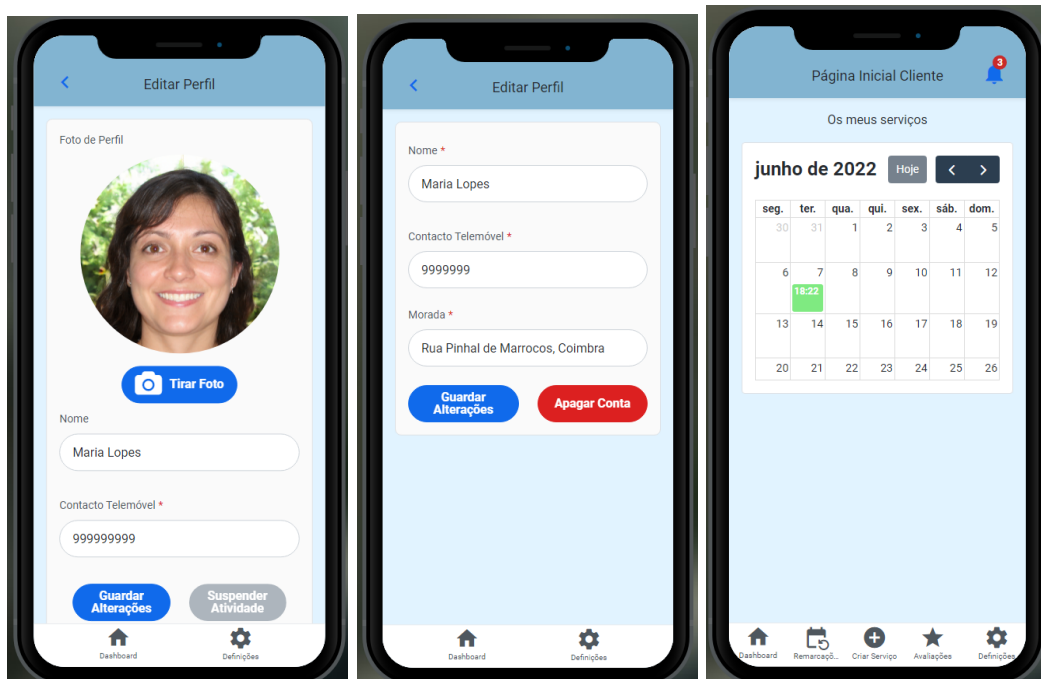
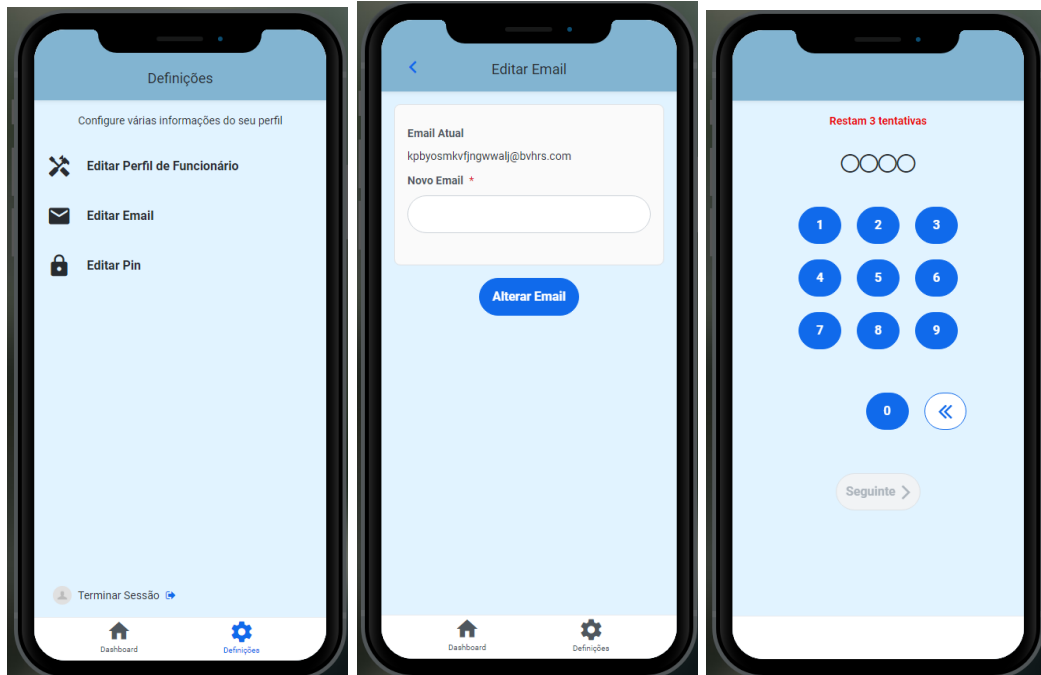
Apêndice B

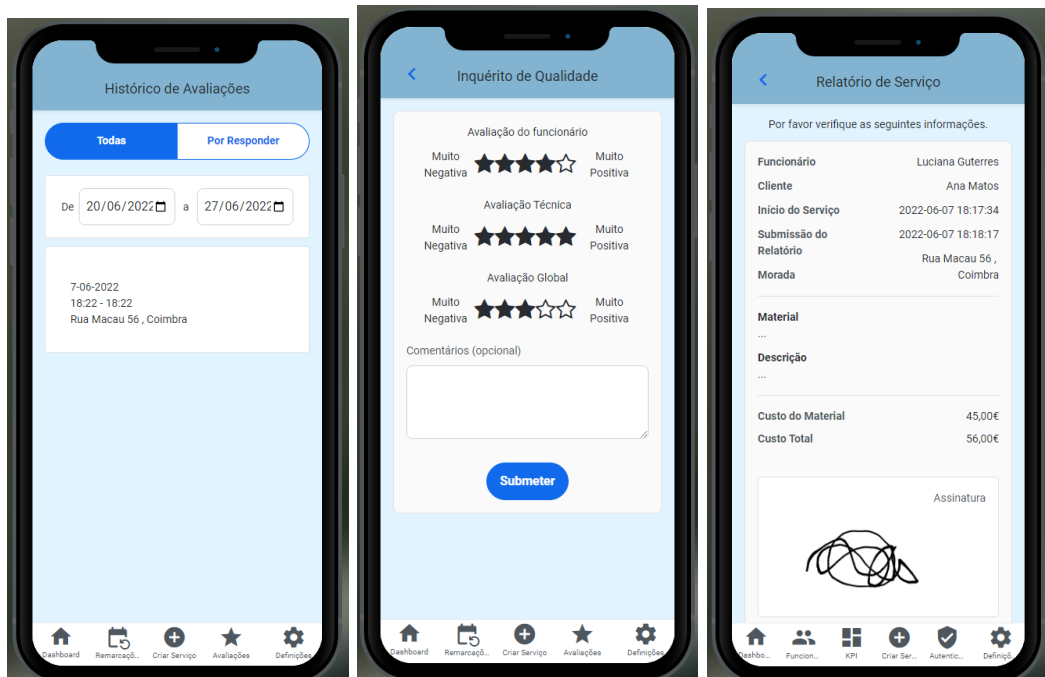
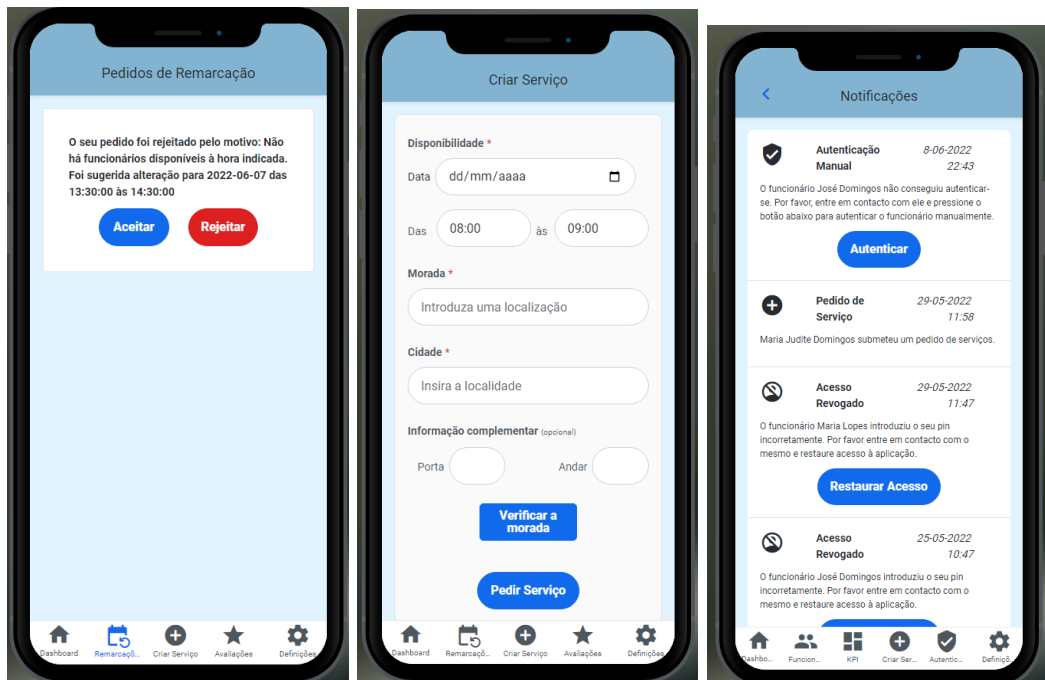
Produto Final











Esta página foi intencionalmente deixada em branco

Apêndice C

Fluxos de Teste

Registo e remoção de funcionários

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	<ol style="list-style-type: none"> 1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login". 	<ol style="list-style-type: none"> 1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor. 	
Dashboard Supervisor			
Navegar para o ecrã de gestão de funcionários	<ol style="list-style-type: none"> 1. Selecionar o ícone "Gerir Funcionários" na <i>bottom bar</i>. 	<ol style="list-style-type: none"> 1. É apresentado o ecrã de gestão de funcionários. 	
Gestão de Funcionários			
Navegar para o ecrã de registo de funcionários	<ol style="list-style-type: none"> 1. Selecionar o ícone "+" no canto superior direito. 	<ol style="list-style-type: none"> 1. É apresentado o ecrã de registo de funcionários 	
Registo de Funcionários			
Registo correto de funcionários	<ol style="list-style-type: none"> 1. Preenchimento correto dos campos requeridos (e-mail, telemóvel e nome válidos); 2. Seleção da opção "Registar". 	<ol style="list-style-type: none"> 1. O utilizador regressa ao ecrã de gestão de funcionários; 2. É apresentada uma mensagem de sucesso referente ao registo. 	
Gestão de Funcionários			
Utilização do filtro para pesquisa de funcionários	<ol style="list-style-type: none"> 1. Preenchimento da barra de pesquisa com o nome do funcionário a remover; 2. Preenchimento da barra de pesquisa com o e-mail do funcionário a remover. 	<ol style="list-style-type: none"> 1. O perfil do funcionário surge como resultado das duas pesquisas, assim como a sua foto de perfil e a sua classificação média. 	
Navegação para o ecrã de informações do funcionário	<ol style="list-style-type: none"> 1. Seleção, através do toque, do funcionário em questão. 	<ol style="list-style-type: none"> 1. O ecrã que contém as informações do funcionário é apresentado. 	
Informação de Funcionários			
Acionar o <i>popup</i> de remoção do funcionário	<ol style="list-style-type: none"> 1. Seleção da opção "Remover Funcionário". 	<ol style="list-style-type: none"> 1. Surge um <i>popup</i> de confirmação. 	
Remoção do funcionário	<ol style="list-style-type: none"> 1. Seleção da opção "Remover Funcionário". 	<ol style="list-style-type: none"> 1. O utilizador é devolvido à página de gestão de funcionários; 2. O funcionário já não se encontra visível na lista. 	
Navegar para o ecrã de funcionários inativos	<ol style="list-style-type: none"> 1. Selecionar o separador "Funcionários Bloqueados". 	<ol style="list-style-type: none"> 1. É apresentada a lista de funcionários bloqueados. 	
Restaurar acesso a funcionários banidos	<ol style="list-style-type: none"> 1. Percorrer a lista de funcionários bloqueados até localizar o que se pretende; 2. Seleção da opção "Restaurar". 	<ol style="list-style-type: none"> 1. É apresentada uma mensagem de sucesso; 2. O funcionário é removido da lista. 	
Visualização de pedidos resolvidos	<ol style="list-style-type: none"> 1. Selecionar a opção de funcionários inativos através do <i>switch</i> no topo do ecrã; 2. Utilização do filtro para encontrar o funcionário em questão. 	<ol style="list-style-type: none"> 1. É apresentada a lista de funcionários com conta suspensa; 2. É apresentado o perfil do funcionário conforme especificado na barra de pesquisa. 	

Criar Serviço como Supervisor

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	<ol style="list-style-type: none"> 1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login". 	<ol style="list-style-type: none"> 1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor. 	
Dashboard Supervisor			
Navegar para o ecrã de criação de serviços	<ol style="list-style-type: none"> 1. Selecionar o ícone "Criar Serviço". 	<ol style="list-style-type: none"> 1. É apresentado o ecrã de criação de serviços. 	
Criação de Serviços			
Preenchimento das informações do serviço	<ol style="list-style-type: none"> 1. Preencher a data e a duração do serviço; 2. Preencher a morada e selecionar uma sugestão do <i>autocomplete</i>; 3. Selecionar a opção "ok". 4. Confirmar a morada introduzida no mapa que surge. 5. Preencher a descrição; 6. Selecionar um funcionário disponível; 7. Procurar o cliente no campo destinado; 8. Preencher o seu nome quando a pesquisa não devolve resultados; 9. Selecionar a opção "Criar Serviço". 	<ol style="list-style-type: none"> 1. O utilizador é devolvido à página <i>dashboard</i> do supervisor; 2. É apresentada uma mensagem de sucesso. 	
Dashboard Supervisor			
Visualização do serviço criado	<ol style="list-style-type: none"> 1. Seleção da opção "Por Iniciar" na <i>dropdown</i> no topo do ecrã; 2. Seleção do serviço criado. 	<ol style="list-style-type: none"> 1. As informações do serviço criado são apresentadas. 	
Detalhes do Serviço			
Alteração das informações do serviço	<ol style="list-style-type: none"> 1. Alteração da data do serviço para outro dia; 2. Seleção de outro funcionário, uma vez que o funcionário escolhido já não se encontra disponível para a nova data; 3. Alteração do cliente para um cliente que não se encontra inscrito na plataforma; 4. Preenchimento manual do nome do cliente; 5. Selecionar a opção "Guardar alterações". 	<ol style="list-style-type: none"> 1. O utilizador é devolvido à página <i>dashboard</i> do supervisor; 2. É apresentada uma mensagem de sucesso. 	

Validação de um pedido

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Visualizar os pedidos por validar	1. Selecionar a opção "Por Validar" no <i>dropdown da dashboard</i> ; 2. Selecionar um dos serviços;	1. É apresentado o ecrã de detalhes do serviço.	
Detalhes do Serviço			
Aceitar o pedido a ser visualizado	1. Preencher as informações do formulário (definir uma data dentro da disponibilidade do cliente, descrição, etc); 2. Atribuir um funcionário da lista de funcionários disponíveis para a hora indicada; 3. Selecionar a opção "Validar Serviço".	1. Surge uma mensagem de sucesso no ecrã; 2. O utilizador é enviado para a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Visualizar os pedidos por iniciar	1. Selecionar a opção "Por Iniciar" no <i>dropdown da dashboard</i> ;	1. É apresentada uma lista de serviços, onde se pode ver o pedido validado.	

Recalendarização de um pedido

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Visualizar os pedidos por validar	1. Selecionar a opção "Por Validar" no <i>dropdown da dashboard</i> ; 2. Selecionar um dos serviços;	1. É apresentado o ecrã de detalhes do serviço.	
Detalhes do Serviço			
Rejeitar o pedido a ser visualizado	1. Conferir que não existem funcionários disponíveis para a hora pedida; 2. Selecionar a opção "Rejeitar Serviço".	1. É apresentado o ecrã de Rejeição do pedido.	
Rejeição de Serviço			
Preencher o pedido de recalendarização	1. Selecionar a justificação para a rejeição do pedido; 2. Selecionar uma nova data para a prestação do serviço. 3. Seleção da opção "Rejeitar Serviço"	1. Surge uma mensagem de sucesso no ecrã; 2. O utilizador é enviado para a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Visualizar os pedidos de recalendarização	1. Selecionar a opção "Remarcações" no <i>dropdown da dashboard</i> ;	1. É apresentada lista de serviços a aguardar resposta de pedidos de remarcação, onde se pode ver o pedido rejeitado anteriormente.	

Navegação entre *dashboards*

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Alterar a <i>dashboard</i> a visualizar	1. Selecionar a opção "dashboard" na <i>bottom bar</i> com um toque prolongado; 2. Selecionar a opção "Funcionário".	1. É apresentada a <i>dashboard</i> do funcionário.	
Dashboard Funcionário			
Alterar a <i>dashboard</i> a visualizar	1. Selecionar a opção "dashboard" na <i>bottom bar</i> com um toque prolongado; 2. Selecionar a opção "Supervisor".	1. É apresentada a <i>dashboard</i> do supervisor.	
Alterar a vista dos serviços			
Visualizar os serviços num calendário	1. Selecionar a opção "Alterar Vista";	1. A lista no ecrã desaparece e é substituída por um calendário com vários serviços, cujos ícones coloridos refletem o estado do serviço.	

Autenticação Manual

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Navegar para o ecrã de autenticação manual de funcionários	1. Selecionar o ícone "Autenticação" na <i>bottom bar</i> .	1. É apresentado o ecrã de autenticação de funcionários.	
Autenticação de funcionários			
Autenticação de um pedido pendente	1. Percorrer a lista de pedidos pendentes até localizar o que se pretende; 2. Seleção da opção "Autenticar".	1. É apresentada uma mensagem de sucesso; 2. O pedido selecionado é removido da lista.	
Visualização de pedidos resolvidos	1. Selecionar a opção de pedidos resolvidos através do <i>switch</i> no topo do ecrã; 2. Utilização do filtro para encontrar o pedido do funcionário em questão.	1. É apresentada a lista de pedidos resolvidos no passado; 2. É apresentado o pedido do funcionário conforme especificado na barra de pesquisa.	

Consultar Relatório e Inquérito de Qualidade

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do supervisor.	
Dashboard Supervisor			
Selecionar um serviço concluído	1. Selecionar a opção "Concluído" na <i>dropdown</i> no topo da página; 2. Selecionar um serviço com um clique.	1. É apresentado o ecrã dos detalhes do serviço.	
Detalhes do Serviço			
Visualizar o relatório do serviço.	1. Selecionar a opção "Relatório".	1. É apresentado o ecrã do relatório.	
Visualizar o inquérito de qualidade do serviço.	1. Selecionar a opção "Inquérito de Qualidade".	1. É apresentado o ecrã da avaliação com os campos não editáveis.	

Iniciar Serviço

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do funcionário.	
Dashboard Funcionário			
Abrir o pop-up com as informações do serviço	1. Selecionar no calendário um serviço com a cor azul.	1. É apresentada o pop-up com algumas informações sobre o serviço.	
Iniciar a deslocação para o local de serviço	1. Selecionar a opção "Iniciar Deslocação".	1. Surge no mapa um trajeto entre a posição atual do supervisor e o local de prestação de serviço; 2. A opção "Iniciar Deslocação" é substituída por "Autenticar", que deverá aparecer desativada até o funcionário se encontrar na proximidade do serviço.	
Autenticação por Reconhecimento Facial	1. Selecionar a opção "Autenticar"	1. O funcionário é redirecionado para o ecrã do reconhecimento facial.	

Autenticação por Reconhecimento Facial

Objetivo	Passos	Resultado Esperado	Resultado
Reconhecimento Facial			
Autenticação por reconhecimento facial	1. Selecionar a opção "Tirar Foto";	1. A câmara do dispositivo é inicializada.	
Câmara do dispositivo			
Tirar uma foto para o reconhecimento facial	1. Tirar uma foto com a sua cara visível; 2. Confirmação da foto tirada.	1. O ecrã do reconhecimento facial é apresentado novamente, com a foto tirada apresentada; 2. A opção "Autenticar" deverá tornar-se disponível.	
Reconhecimento Facial			
Autenticação por reconhecimento facial	1. Selecionar a opção "Autenticar";	1. Surge uma mensagem de sucesso; 2. O funcionário é redirecionado para o ecrã dos detalhes do preenchimento do relatório;	

Autenticação por Pin

Objetivo	Passos	Resultado Esperado	Resultado
Reconhecimento Facial			
Autenticação por reconhecimento facial	1. Selecionar a opção "Tirar Foto";	1. A câmara do dispositivo é inicializada.	
Câmara do dispositivo			
Tirar uma foto para o reconhecimento facial	1. Tirar uma foto com a sua cara não visível; 2. Confirmação da foto tirada.	1. O ecrã do reconhecimento facial é apresentado novamente, com a foto tirada apresentada; 2. A opção "Autenticar" deverá tornar-se disponível.	
Reconhecimento Facial			
Falhar a autenticação por reconhecimento facial	1. Selecionar a opção "Autenticar";	1. Surge uma mensagem de erro; 2. Ao fim de 5 tentativas falhadas, a opção "Autenticação por Pin" torna-se disponível.	
Autenticação por Pin			
Autenticação por pin	1. Introduzir o pin do funcionário; 2. Selecionar a opção "Autenticar"	1. Surge uma mensagem de sucesso; 2. O funcionário é redirecionado para o ecrã dos detalhes do serviço.	
Detalhes do Serviço			
Terminar serviço	1. Selecionar a opção "Terminar Serviço"	1. O funcionário é redirecionado para o ecrã de preenchimento do relatório.	

Preencher Relatório

Objetivo	Passos	Resultado Esperado	Resultado
Relatório Funcionário			
Submeter o relatório	1. Preencher as informações sobre o material utilizado e os serviços prestados, assim como o preço cobrado; 2. Pedir ao cliente para assinar o relatório; 3. Selecionar a opção "Submeter"	1. É apresentada uma mensagem de sucesso; 2. O funcionário é reencaminhado para a sua <i>dashboard</i> .	

Concluir Registo

Objetivo	Passos	Resultado Esperado	Resultado
E-mail			
Abrir a aplicação no ecrã de conclusão de registo	<ol style="list-style-type: none"> 1. Clicar no link fornecido no e-mail; 2. Abrir na aplicação. 	1. É apresentado o ecrã de recuperação de <i>password</i> , no modo de conclusão de registo.	
Recuperação de <i>Password</i>			
Definir uma <i>password</i> para substituir o <i>token</i> gerado no registo.	<ol style="list-style-type: none"> 1. Introduzir uma <i>password</i> com um mínimo de 8 caracteres; 2. Confirmar a <i>password</i>; 3. Selecionar a opção "Seguinte" 	<ol style="list-style-type: none"> 1. É apresentada uma mensagem de sucesso; 2. O funcionário é reencaminhado para o ecrã de definição de pin. 	
Definição de Pin			
Definir um pin pessoal para o funcionário.	<ol style="list-style-type: none"> 1. Introduzir um pin de 4 dígitos; 2. Utilizar o botão de apagar para corrigir o último dígito introduzido; 3. Seleção da opção "Seguinte". 	1. É apresentado o ecrã de confirmação de pin.	
Confirmação de pin			
Falhar a confirmação do pin introduzido	<ol style="list-style-type: none"> 1. Introduzir um pin diferente do anterior; 2. Selecionar a opção "Seguinte" 	1. É apresentada uma mensagem de erro a informar o utilizador que o pin não corresponde ao definido anteriormente.	
Confirmação do pin introduzido	<ol style="list-style-type: none"> 1. Introduzir o pin anterior; 2. Selecionar a opção "Seguinte" 	<ol style="list-style-type: none"> 1. É apresentada uma mensagem de sucesso; 2. O funcionário é reencaminhado para o ecrã de definição de foto de perfil. 	
Foto de Perfil			
Definir uma foto de perfil	<ol style="list-style-type: none"> 1. Selecionar a opção "Tirar Foto"; 	1. A câmara do dispositivo é inicializada.	
Câmara do dispositivo			
Tirar uma foto para o perfil / reconhecimento facial	<ol style="list-style-type: none"> 1. Tirar uma foto com a sua cara visível; 2. Confirmação da foto tirada. 	1. O ecrã do reconhecimento facial é apresentado novamente, com a foto tirada apresentada;	
Reconhecimento Facial			
Autenticação por reconhecimento facial	<ol style="list-style-type: none"> 1. Selecionar a opção "Concluir"; 	<ol style="list-style-type: none"> 1. Surge uma mensagem de sucesso; 2. O funcionário é redirecionado para a sua <i>dashboard</i>. 	

Edição do perfil

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do funcionário.	
Dashboard Funcionário			
Abrir as definições	1. Selecionar a opção "Definições" na <i>bottom bar</i> .	1. É apresentado o ecrã de definições.	
Definições			
Selecionar as informações a alterar	1. Selecionar a opção "Editar Perfil".	1. É apresentado o ecrã de perfil do funcionário.	
Editar Perfil			
Editar informações	1. Alterar o nome do funcionário; 2. Selecionar a opção "Guardar Alterações";	1. Surge uma mensagem de sucesso; 2. O funcionário é reenviado para a sua <i>dashboard</i> .	
Definições			
Selecionar as informações a alterar	1. Selecionar a opção "Editar E-mail".	1. É apresentado o ecrã de alteração de e-mail.	
Editar E-mail			
Editar informações	1. Alterar o e-mail do funcionário; 2. Selecionar a opção "Guardar Alterações"; 3. Introduzir a <i>password</i> no pop-up que surgiu; 4. Selecionar a opção "Guardar";	1. Surge uma mensagem de sucesso; 2. O funcionário é reenviado para a sua <i>dashboard</i> .	

Editar Pin

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do funcionário.	
Dashboard Funcionário			
Abrir as definições	1. Selecionar a opção "Definições" na <i>bottom bar</i> .	1. É apresentado o ecrã de definições.	
Selecionar as informações a alterar	1. Selecionar a opção "Editar Pin".	1. É apresentado o ecrã de perfil do funcionário.	
Editar Perfil			
Iniciar alteração do pin	1. Selecionar a opção "Alterar Pin";	1. É apresentado o ecrã de confirmação de pin.	
Confirmação de pin			
Confirmação do pin atual	1. Introduzir o pin atual; 2. Selecionar a opção "Seguinte"	1. É apresentado o ecrã de redefinição do pin.	
Definição de Pin			
Definir um pin pessoal para o funcionário.	1. Introduzir um pin de 4 dígitos; 2. Seleção da opção "Seguinte".	1. É apresentado o ecrã de confirmação de pin.	
Confirmação de pin			
Confirmação do pin introduzido	1. Introduzir o pin anterior; 2. Selecionar a opção "Seguinte"	1. É apresentada uma mensagem de sucesso; 2. O funcionário é reencaminhado para o seu perfil.	

Bloquear funcionário

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do funcionário.	
Dashboard Funcionário			
Abrir as definições	1. Selecionar a opção "Definições" na <i>bottom bar</i> .	1. É apresentado o ecrã de definições.	
Definições			
Iniciar alteração do pin	1. Selecionar a opção "Alterar Pin";	1. É apresentado o ecrã de confirmação de pin.	
Confirmação de pin			
Confirmação do pin atual	1. Introduzir o pin errado; 2. Selecionar a opção "Seguinte"	1. O contador de tentativas baixa com cada tentativa errada; 2. Surge uma mensagem de erro; 3. A sessão do funcionário é terminada; 4. O funcionário é redirecionado para o ecrã de Login.	

Suspender Conta

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do funcionário.	
Dashboard Funcionário			
Abrir edição de perfil	1. Abrir o menu lateral; 2. Abrir o submenu de <i>Editar Perfil</i> ao selecionar a opção com o mesmo nome; 3. Selecionar a opção "Funcionário"	1. É apresentado o perfil do funcionário.	
Editar Perfil			
Editar informações	1. Selecionar a opção "Suspender Atividade"; 2. Confirmar no pop-up.	1. Surge uma mensagem de sucesso; 2. O funcionário é reenviado para o ecrã de login.	

Reativar Conta

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do funcionário; 3. Surge um pop-up com um botão para reativar a conta.	
Dashboard Funcionário			
Reativar conta	1. Seleção da opção "Ativar Conta".	1. O pop-up é fechado.	

Registo de Cliente

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar o registo de cliente	1. Selecionar a opção "Registar".	1. É apresentado o ecrã de registo.	
Registo de Clientes			
Navegar para o ecrã de criação de serviços	1. Preencher as informações requeridas (nome, e-mail, morada e <i>password</i>). 2. Selecionar a opção "Registar".	1. É apresentada uma mensagem de sucesso; 2. O cliente é redirecionado para a página de Login.	
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e <i>password</i> corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do cliente.	

Criar Serviço como Cliente

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e <i>password</i> corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do cliente.	
Dashboard Supervisor			
Navegar para o ecrã de criação de serviços	1. Selecionar o ícone "Criar Serviço".	1. É apresentado o ecrã de criação de serviços.	
Criação de Serviços			
Preenchimento das informações do serviço	1. Preencher a data e a duração do serviço; 2. Preencher a morada e a descrição; 3. Confirmar a localização da morada no mapa; 4. Selecionar a opção "Criar Serviço".	1. O utilizador é devolvido à página <i>dashboard</i> do cliente; 2. É apresentada uma mensagem de sucesso.	

Pedidos de recalendarização

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do cliente.	
Dashboard Supervisor			
Navegar para o ecrã de pedidos de recalendarização	1. Selecionar o ícone "Recalendarizações" na <i>bottom bar</i> .	1. É apresentado o ecrã de recalendarização de pedidos.	
Recalendarizações			
Aceitar um pedido de recalendarização.	1. Percorrer a lista de pedidos até localizar o que se pretende; 2. Seleção da opção "Aceitar".	1. É apresentada uma mensagem de sucesso; 2. As opções "Aceitar" e "Rejeitar" são removidas.	
Rejeitar um pedido de recalendarização.	1. Percorrer a lista de pedidos até localizar o que se pretende; 2. Seleção da opção "Rejeitar"; 3. Clicar na opção "Rejeitar" no pop-up que resulta do passo 2.	1. É apresentada uma mensagem de sucesso; 2. O pop-up desaparece; 3. As opções "Aceitar" e "Rejeitar" são removidas.	

Inquéritos de qualidade

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Selecionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do cliente.	
Dashboard Supervisor			
Navegar para o ecrã de inquéritos de qualidade	1. Selecionar o ícone "Avaliações".	1. É apresentado o ecrã de inquéritos de qualidade.	
Avaliações			
Selecionar um serviço por avaliar	1. Percorrer a lista de pedidos até localizar o que se pretende; 2. Selecionar o serviço através de um clique.	1. É apresentado o ecrã do inquérito de qualidade.	
Inquérito de Qualidade			
Submeter avaliação do serviço	1. Preencher as várias avaliações com um toque na classificação pretendida. 2. Seleção da opção "Submeter"	1. Surge uma mensagem de sucesso; 2. O cliente é redirecionado para o ecrã dos inquéritos de qualidade.	
Avaliações			
Selecionar um serviço avaliado	1. Percorrer a lista de pedidos até localizar o que se pretende; 2. Selecionar o serviço através de um clique.	1. É apresentado o ecrã do inquérito de qualidade.	
Visualização de pedidos resolvidos	1. Desativar o filtro de inquéritos sem resposta através do switch no topo do ecrã;	1. É apresentada a lista de todos inquéritos, com resposta ou não;	

Edição do perfil

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Seleccionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do Cliente.	
Dashboard Cliente			
Abrir as definições	1. Seleccionar a opção "Definições" na <i>bottom bar</i> .	1. É apresentado o ecrã de definições.	
Definições			
Seleccionar as informações a alterar	1. Seleccionar a opção "Editar Perfil".	1. É apresentado o ecrã de perfil do Cliente.	
Editar Perfil			
Editar informações	1. Alterar o nome do cliente; 2. Seleccionar a opção "Guardar Alterações";	1. Surge uma mensagem de sucesso; 2. O cliente é reenviado para a sua <i>dashboard</i> .	
Definições			
Seleccionar as informações a alterar	1. Seleccionar a opção "Editar E-mail".	1. É apresentado o ecrã de alteração de e-mail.	
Editar E-mail			
Editar informações	1. Alterar o e-mail do cliente; 2. Seleccionar a opção "Guardar Alterações"; 3. Introduzir a <i>password</i> no pop-up que surgiu; 4. Seleccionar a opção "Guardar";	1. Surge uma mensagem de sucesso; 2. O cliente é reenviado para a sua <i>dashboard</i> .	

Ver Notificações

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar sessão corretamente	1. Introduzir o nome de utilizador e password corretamente; 2. Seleccionar a opção "Login".	1. Utilizar credenciais corretas permite o início de sessão; 2. É apresentada a <i>dashboard</i> do utilizador.	
Dashboard			
Visualizar notificações	1. Seleccionar o ícone "Notificações".	1. É apresentado o ecrã das notificações.	
Notificações			
Eliminar notificação	1. Deslizar uma notificação para a esquerda; 2. Clicar no ícone do caixote do lixo.	1. A notificação desaparece da lista.	
Navegar para outra página de notificações	1. Clicar na seta de navegação ao fundo da página.	1. São apresentadas 5 novas notificações	

Pedido de Recuperação de *Password*

Objetivo	Passos	Resultado Esperado	Resultado
Login			
Iniciar processo de recuperação de <i>password</i>	1. Selecionar a opção "Esqueceu-se da <i>password</i> ".	1. É apresentado ao utilizador o ecrã de pedido de recuperação de <i>password</i> .	
Pedido de Recuperação de <i>Password</i>			
Recuperar a <i>password</i>	1. Preencher o campo do e-mail para o qual será enviado o link de recuperação da <i>password</i> .	1. Surge uma mensagem de sucesso; 2. O utilizador é reencaminhado para o ecrã de Login.	

Recuperar *Password*

Objetivo	Passos	Resultado Esperado	Resultado
E-mail			
Abrir a aplicação no ecrã de recuperação de <i>password</i>	1. Clicar no link fornecido no e-mail; 2. Abrir na aplicação.	1. É apresentado o ecrã de recuperação de <i>password</i> .	
Recuperação de <i>Password</i>			
Definir uma nova <i>password</i> .	1. Introduzir uma <i>password</i> com um mínimo de 8 caracteres; 2. Confirmar a <i>password</i> ; 3. Selecionar a opção "Seguinte"	1. É apresentada uma mensagem de sucesso; 2. O funcionário é reencaminhado para o ecrã de Login.	

Esta página foi intencionalmente deixada em branco

Apêndice D

Questionários de Teste

QuickServe Cliente

Questionário de usabilidade da aplicação QuickServe para clientes.

[Inicie sessão no Google](#) para guardar o seu progresso. [Saiba mais](#)

***Obrigatório**

É fácil criar um serviço. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Compreendo o resultado de aceitar/rejeitar a recalendarização de um serviço. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A interface do calendário é perceptível e de fácil utilização. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Seguir a localização do funcionário é simples. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

É fácil alterar as informações pessoais da minha conta. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A navegação pelos ecrãs é simples. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A utilização do calendário representa uma funcionalidade útil. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

É simples introduzir informações sobre o serviço. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

As interações com as notificações são claras. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

O mecanismo de recuperação da password é claro. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Foi simples utilizar o pop-up para visualizar as informações do serviço. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Utilizaria esta aplicação para requisitar serviços? Porquê?

A sua resposta _____

Que funcionalidades sentiu que mais contribuem para a sua segurança?
Acrescentaria alguma?

A sua resposta _____

QuickServe Funcionário

Questionário de usabilidade da aplicação QuickServe para funcionários.

*Obrigatório

É fácil concluir o registo na aplicação. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

A interface do calendário é perceptível e de fácil utilização. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

O mapa é simples de utilizar para visualizar o local do serviço. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

É fácil alterar as informações pessoais da minha conta. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

A navegação pelos ecrãs é simples. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A utilização do calendário representa uma funcionalidade útil. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

O mecanismo de autenticação por reconhecimento facial é confiável. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A autenticação por pin é uma boa alternativa. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A progressão do estado dos serviços é clara. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

O preenchimento do relatório é simples. *

1 2 3 4 5

Discordo totalmente

Concordo totalmente

Como considera que as ferramentas de autenticação afetam o seu trabalho?

A sua resposta

Que funcionalidades consideraria úteis para apoiar a sua prestação de serviços?

A sua resposta

QuickServe Supervisor

Questionário de usabilidade da aplicação QuickServe para supervisores.

*Obrigatório

Alterar o modo de visualização da dashboard é simples. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A interface do calendário é perceptível e de fácil utilização. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A navegação da lista de serviços por estado é clara. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

É fácil verificar a localização dos funcionários. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

É simples adicionar/eliminar funcionários da plataforma. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

A navegação pelos ecrãs é simples. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

A utilização do calendário representa uma funcionalidade útil. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

A gestão de recalendarização de serviços é simples. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

As interações com as notificações são claras. *

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

É simples consultar o relatório e classificação do serviço. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A navegação entre o perfil de supervisor e funcionário é clara. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Foi simples definir o intervalo relevante no ecrã de KPI. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

O mapa geral foi de fácil navegação. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Esta página foi intencionalmente deixada em branco

Apêndice E

Resultados dos Inquéritos de Teste

Seguem-se os resultados dos testes realizados, separados pelo tipo de perfil.

Cliente

As respostas às perguntas de escolha múltipla colocadas depois dos testes dos casos de uso do cliente são apresentadas da Figura E.1 à Figura E.11.

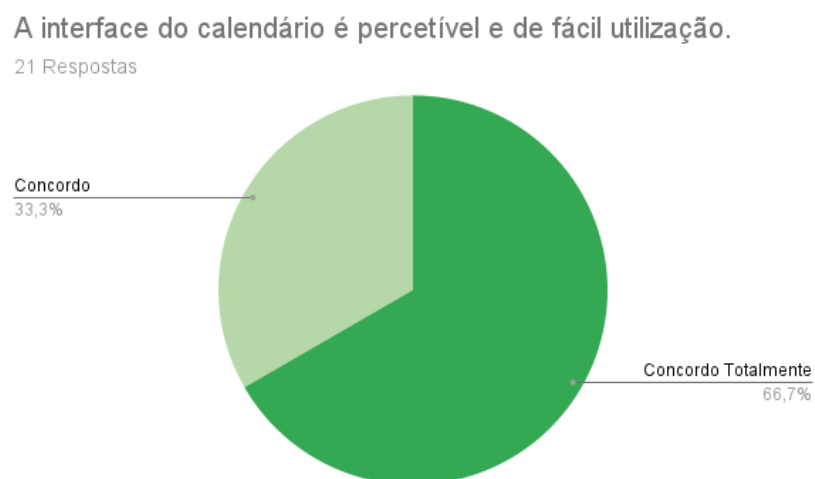


Figura E.1: Respostas à pergunta “A interface do calendário é perceptível e de fácil utilização.”

A navegação pelos ecrãs é simples.

21 Respostas

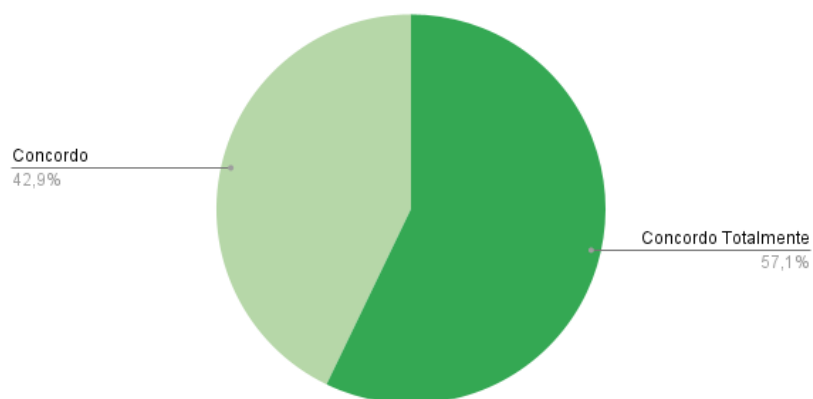


Figura E.2: Respostas à pergunta “A navegação pelos ecrãs é simples.”

A utilização do calendário representa uma funcionalidade útil.

21 Respostas

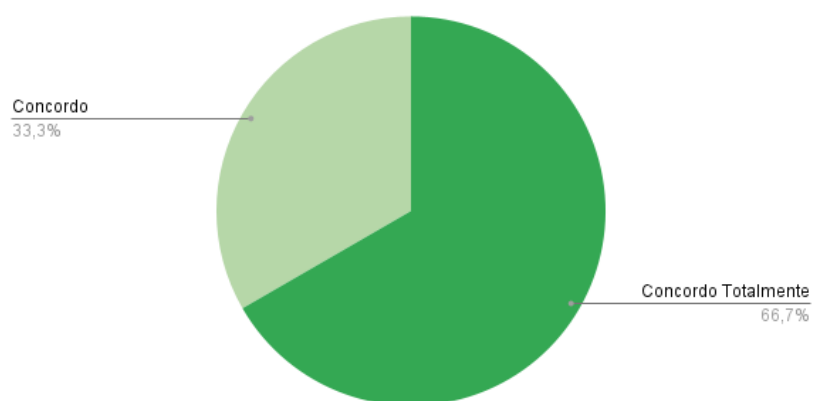


Figura E.3: Respostas à pergunta “A utilização do calendário representa uma funcionalidade útil.”

As interações com as notificações são claras.

21 Respostas

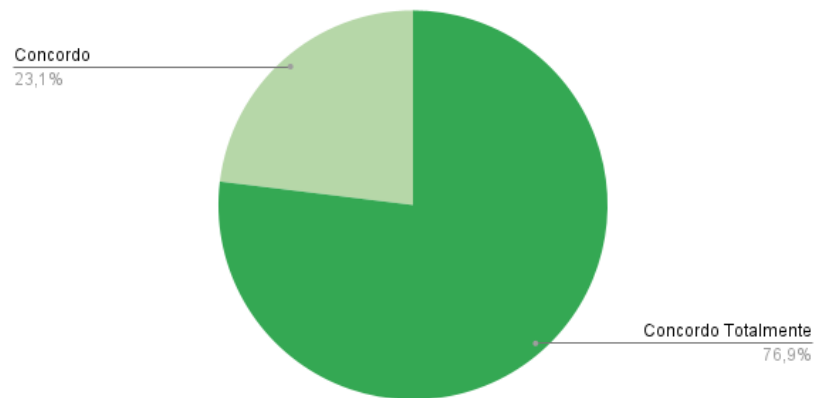


Figura E.4: Respostas à pergunta “As interações com as notificações são claras.”

Compreendo o resultado de aceitar/rejeitar a recalendarização de um serviço.

23 Respostas

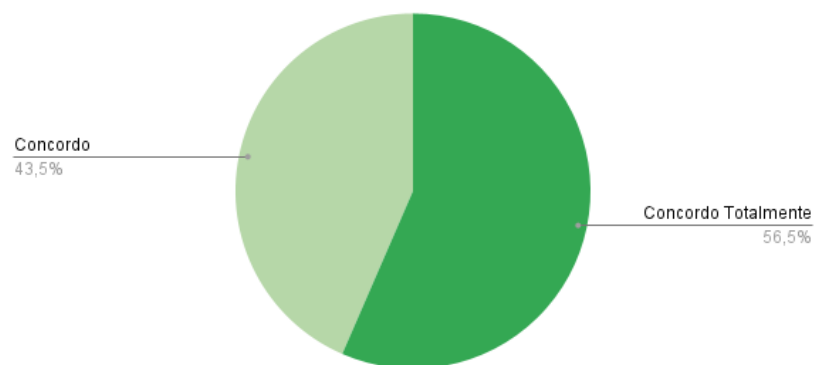


Figura E.5: Respostas à pergunta “Compreendo o resultado de aceitar/rejeitar a recalendarização de um serviço.”

É fácil alterar as informações pessoais da minha conta.

21 Respostas

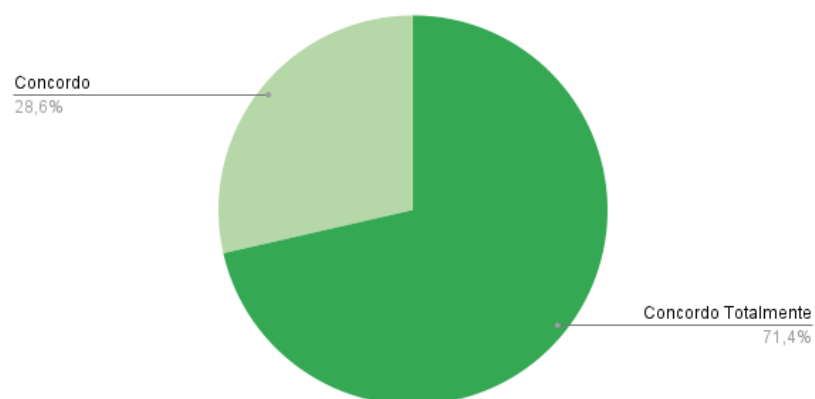


Figura E.6: Respostas à pergunta “É fácil alterar as informações pessoais da minha conta.”

É fácil criar um serviço.

21 Respostas

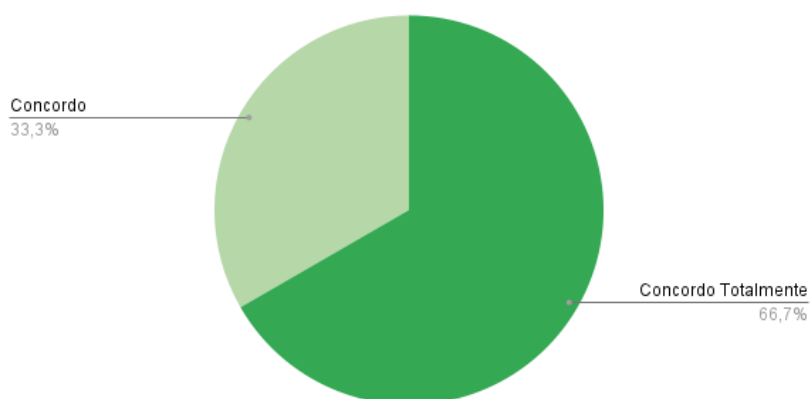


Figura E.7: Respostas à pergunta “É fácil criar um serviço.”

É simples introduzir informações sobre o serviço.

21 Respostas

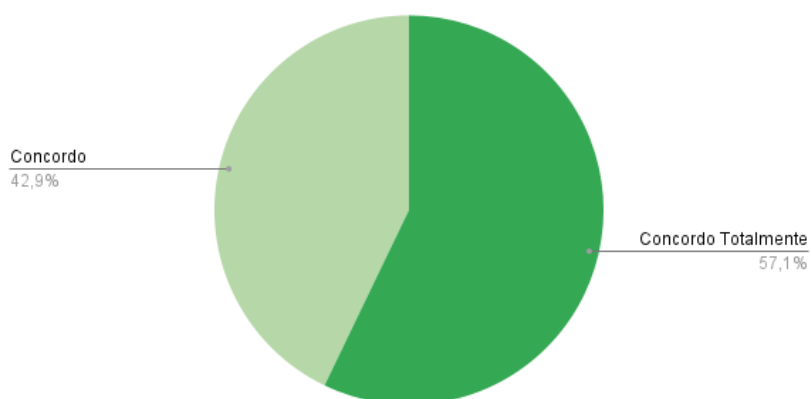


Figura E.8: Respostas à pergunta “É simples introduzir informações sobre o serviço.”

Foi simples utilizar o pop-up para visualizar as informações do serviço.

21 Respostas

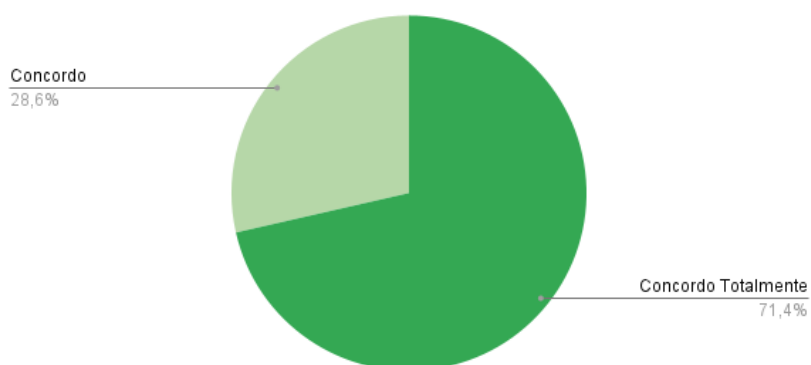


Figura E.9: Respostas à pergunta “Foi simples utilizar o *pop-up* para visualizar as informações do serviço.”

O mecanismo de recuperação da password é claro.

21 Respostas

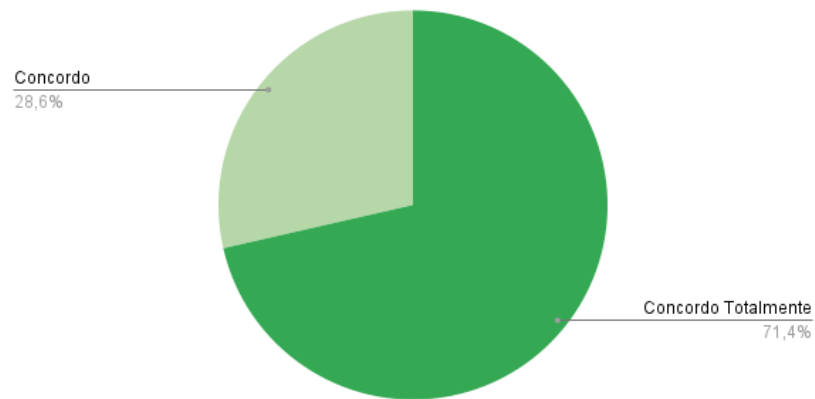


Figura E.10: Respostas à pergunta “O mecanismo de recuperação da password é claro.”

Seguir a localização do funcionário é simples.

21 Respostas

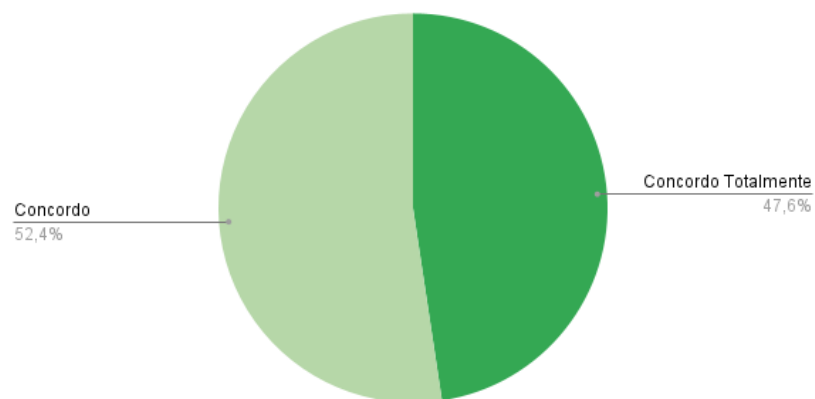


Figura E.11: Respostas à pergunta “Seguir a localização do funcionário é simples.”

Finalmente, as Figuras E.12 e E.13 apresentam as respostas obtidas para as perguntas de resposta aberta.

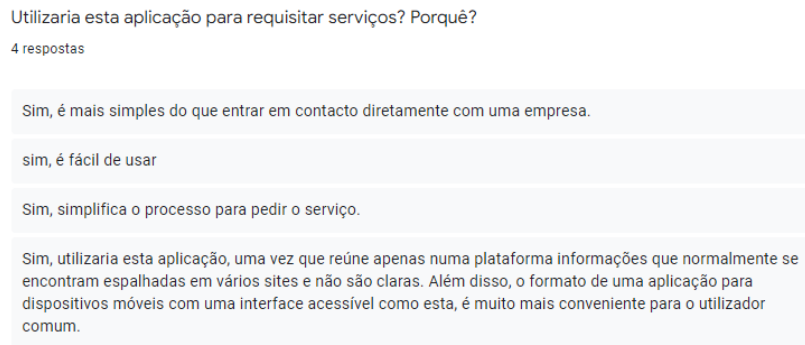


Figura E.12: Respostas à pergunta “Utilizaria esta aplicação para requisitar serviços? Porquê?”

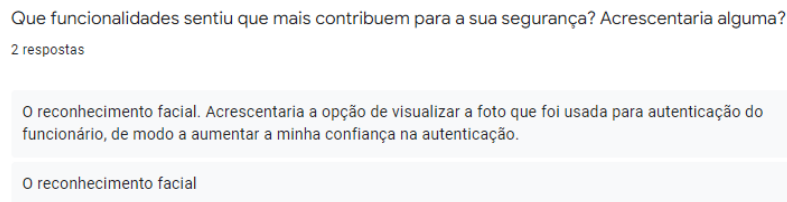


Figura E.13: Respostas à pergunta “Que funcionalidades sentiu que mais contribuem para a sua segurança? Acrescentaria alguma?”

Funcionário

As respostas às perguntas de escolha múltipla colocadas depois dos testes dos casos de uso do funcionário são apresentadas da Figura E.14 à Figura E.23.



Figura E.14: Respostas à pergunta “A autenticação por pin é uma boa alternativa.”

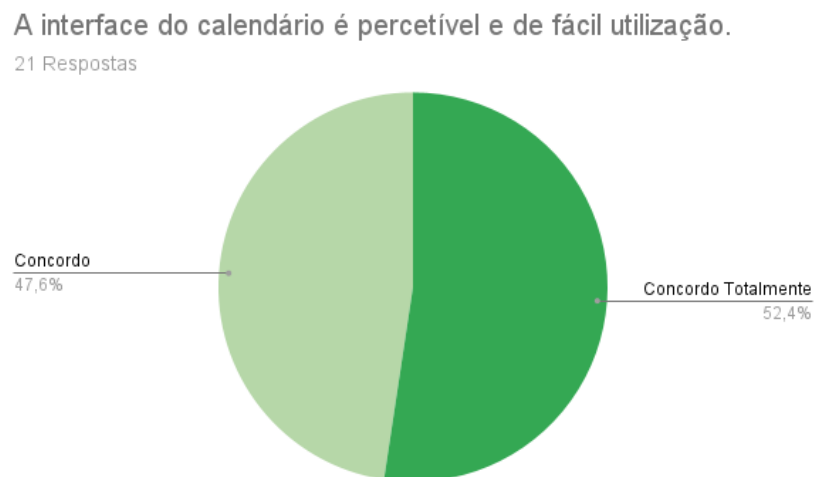


Figura E.15: Respostas à pergunta “A interface do calendário é perceptível e de fácil utilização.”

A navegação pelos ecrãs é simples.

21 Respostas

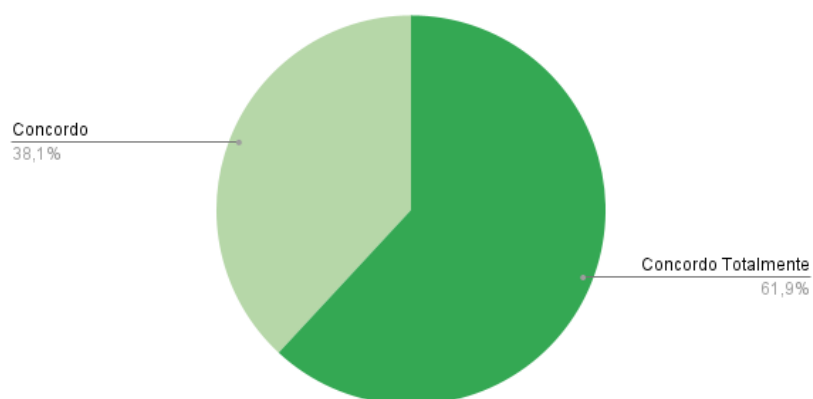


Figura E.16: Respostas à pergunta “A navegação pelos ecrãs é simples.”

A progressão do estado dos serviços é clara.

21 Respostas

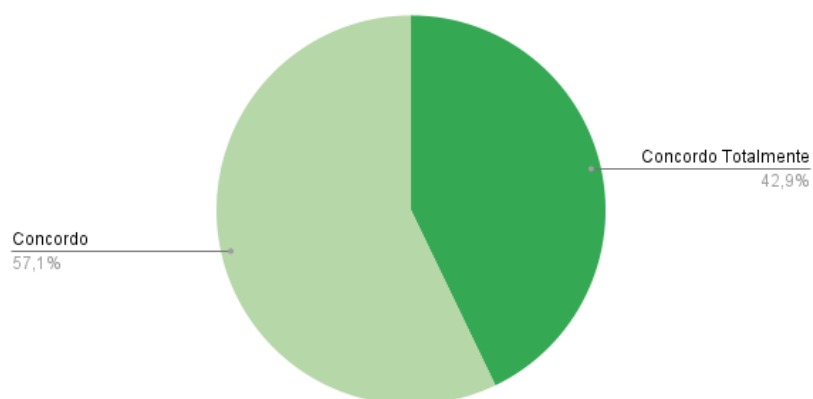


Figura E.17: Respostas à pergunta “A progressão do estado dos serviços é clara.”

A utilização do calendário representa uma funcionalidade útil.

21 Respostas

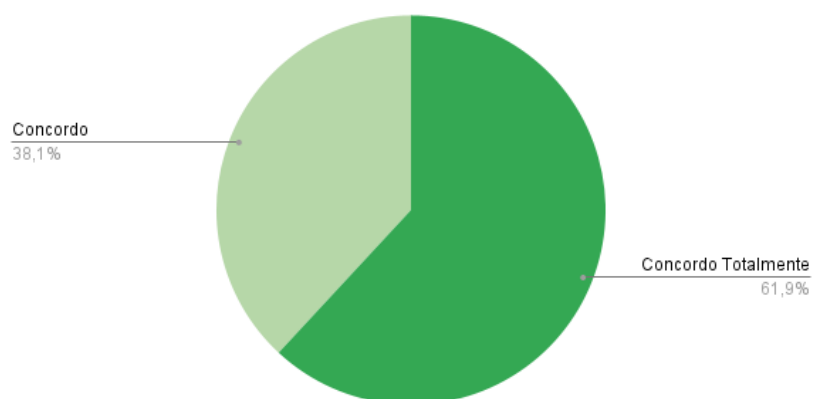


Figura E.18: Respostas à pergunta “A utilização do calendário representa uma funcionalidade útil.”

É fácil alterar as informações pessoais da minha conta.

21 Respostas

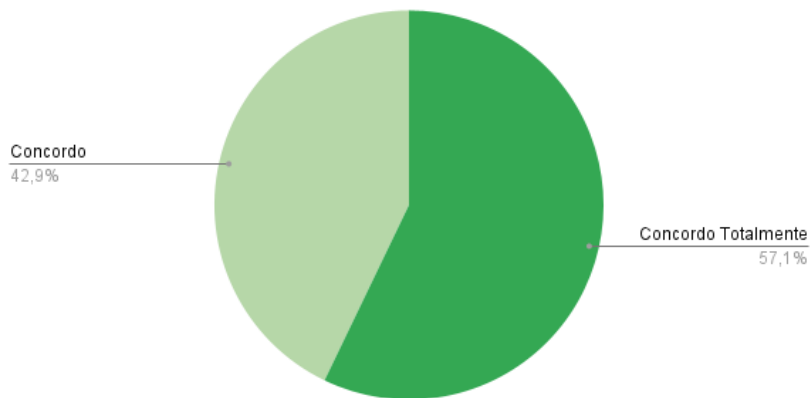


Figura E.19: Respostas à pergunta “É fácil alterar as informações pessoais da minha conta.”

É fácil concluir o registo na aplicação.

21 Respostas

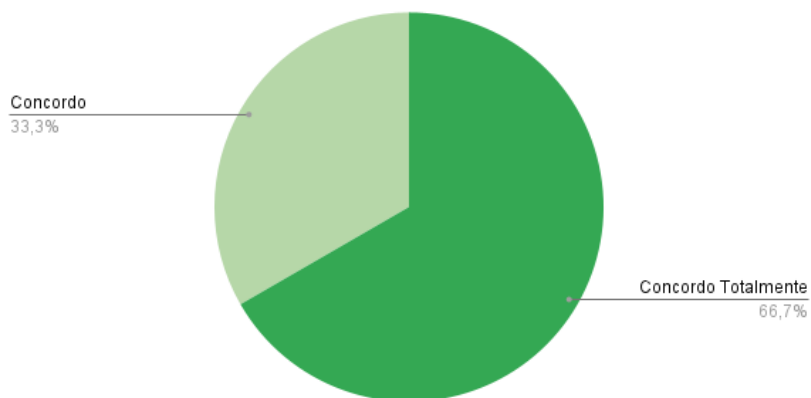


Figura E.20: Respostas à pergunta “É fácil concluir o registo na aplicação.”

O mapa é simples de utilizar para visualizar o local do serviço.

21 Respostas

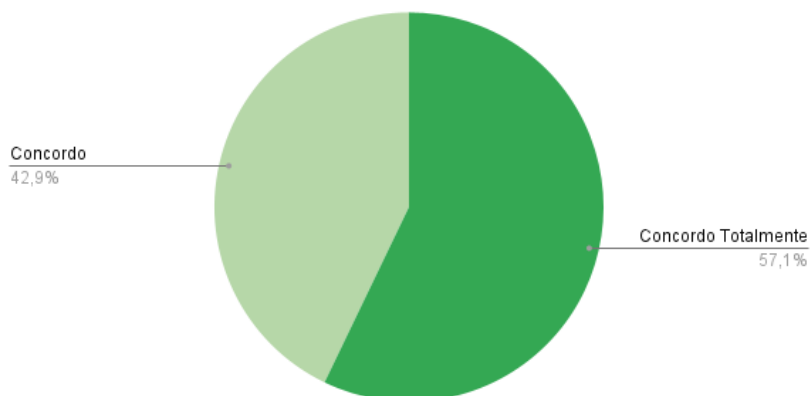


Figura E.21: Respostas à pergunta “O mapa é simples de utilizar para visualizar o local do serviço.”

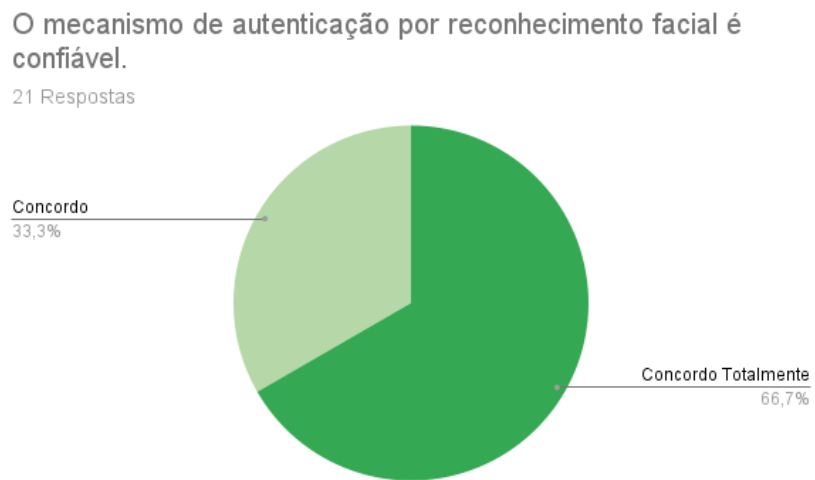


Figura E.22: Respostas à pergunta “O mecanismo de autenticação por reconhecimento facial é confiável.”

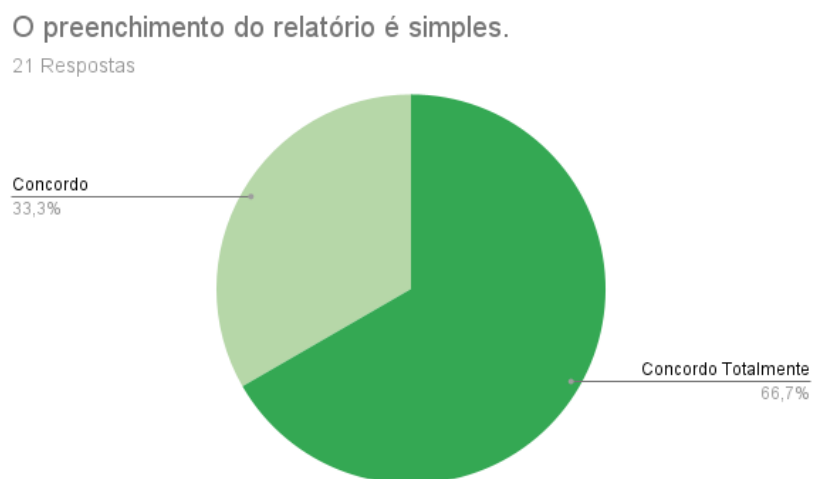


Figura E.23: Respostas à pergunta “O preenchimento do relatório é simples.”

Finalmente, as Figuras E.24 e E.25 apresentam as respostas obtidas para as perguntas de resposta aberta.

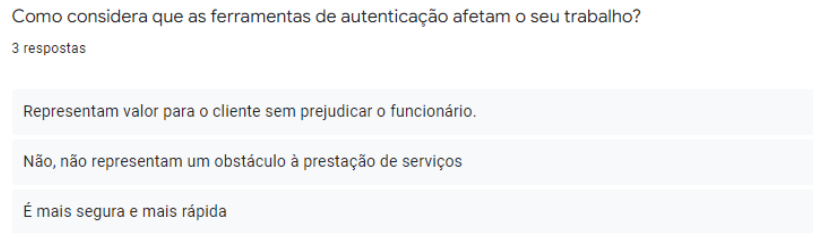


Figura E.24: Respostas à pergunta “Como considera que as ferramentas de autenticação afetam o seu trabalho?”

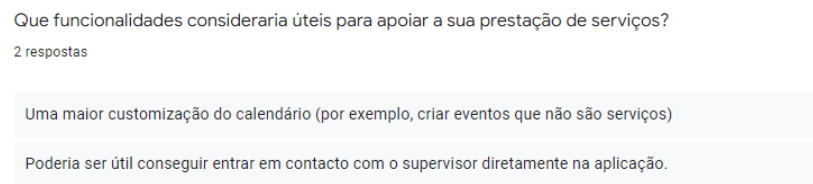


Figura E.25: Respostas à pergunta “Que funcionalidades consideraria úteis para apoiar a sua prestação de serviços?”

Supervisor

As respostas às perguntas de escolha múltipla colocadas depois dos testes dos casos de uso do supervisor são apresentadas da Figura E.26 à Figura E.38.

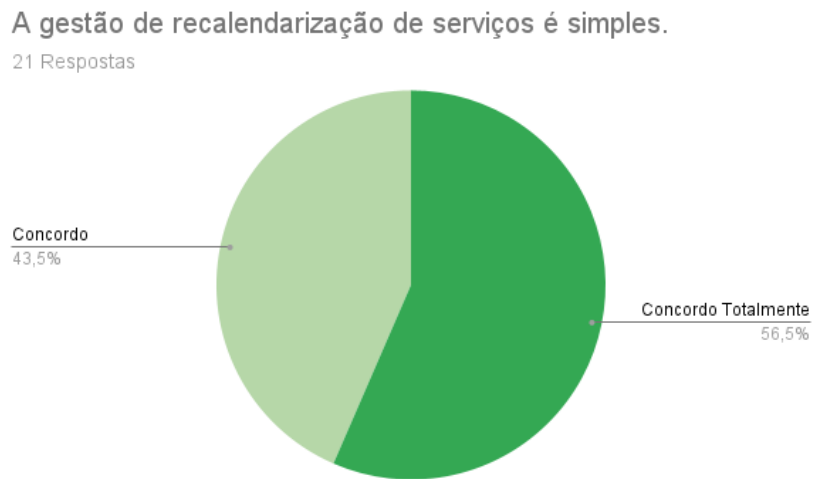


Figura E.26: Respostas à pergunta “A gestão de recalendarização de serviços é simples.”

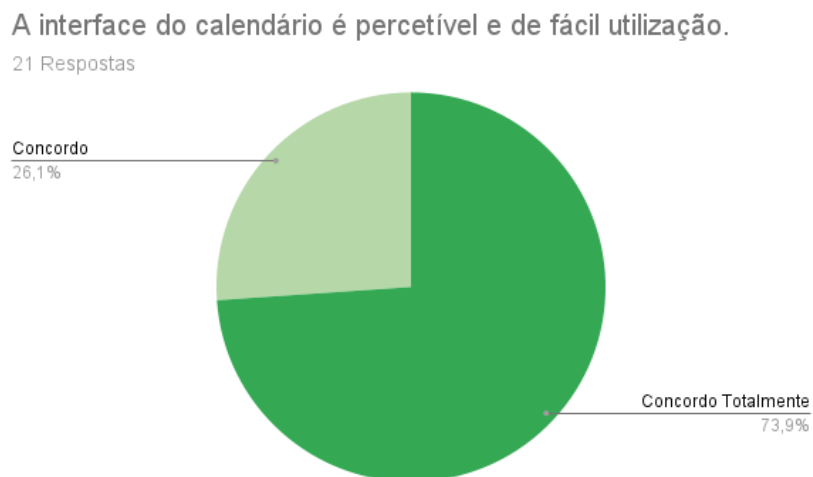


Figura E.27: Respostas à pergunta “A interface do calendário é perceptível e de fácil utilização.”

A navegação da lista de serviços por estado é clara.

21 Respostas

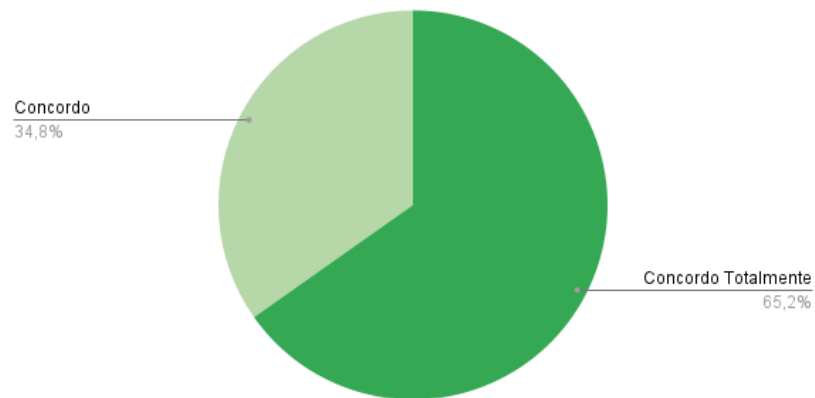


Figura E.28: Respostas à pergunta “A navegação da lista de serviços por estado é clara.”

A navegação entre o perfil de supervisor e funcionário é clara.

21 Respostas

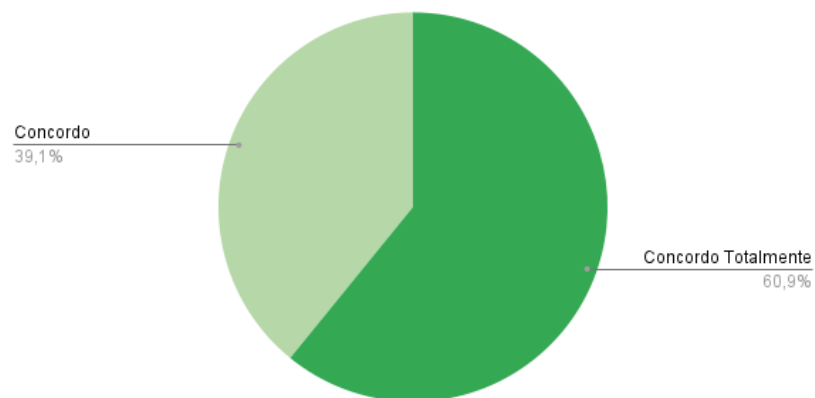


Figura E.29: Respostas à pergunta “A navegação entre o perfil de supervisor e funcionário é clara.”

A navegação pelos ecrãs é simples.

21 Respostas

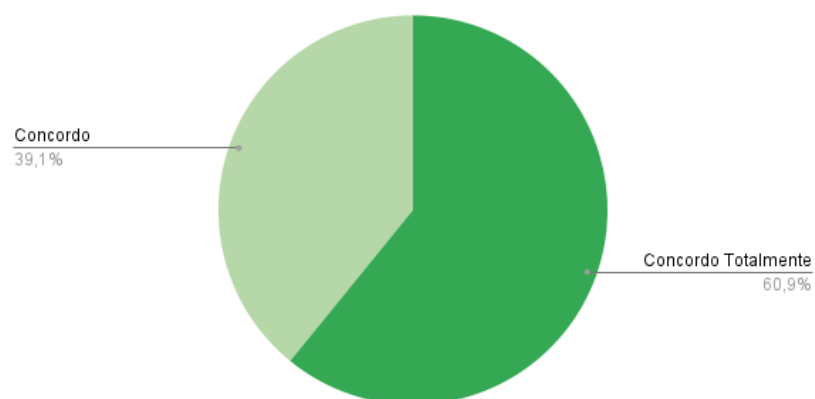


Figura E.30: Respostas à pergunta “A navegação pelos ecrãs é simples.”

A utilização do calendário representa uma funcionalidade útil.

21 Respostas

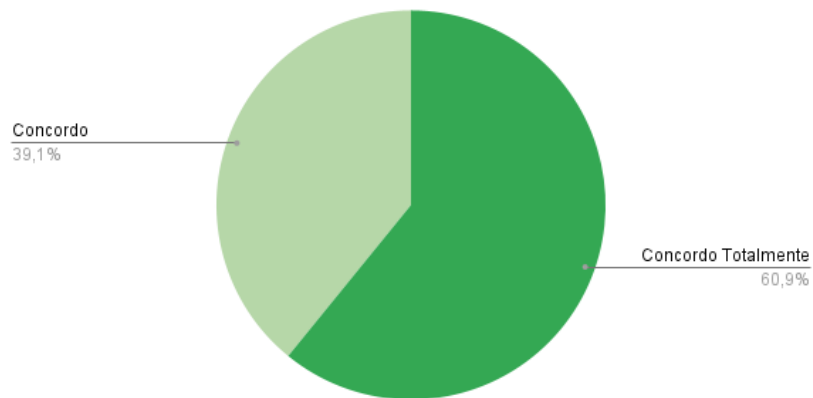


Figura E.31: Respostas à pergunta “A utilização do calendário representa uma funcionalidade útil.”

Alterar o modo de visualização da dashboard é simples.

21 Respostas

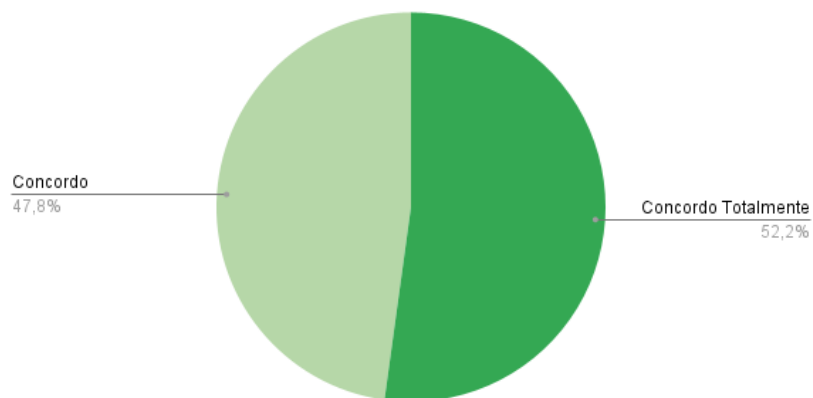


Figura E.32: Respostas à pergunta “Alterar o modo de visualização da *dashboard* é simples.”

As interações com as notificações são claras.

21 Respostas

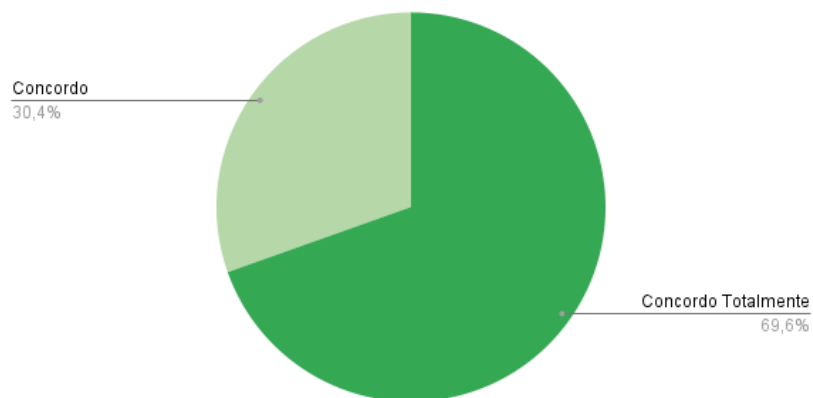


Figura E.33: Respostas à pergunta “As interações com as notificações são claras.”

É fácil verificar a localização dos funcionários.

21 Respostas

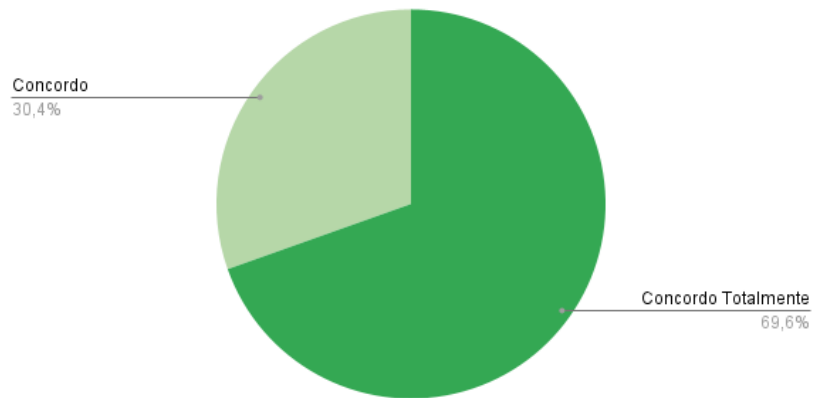


Figura E.34: Respostas à pergunta “É fácil verificar a localização dos funcionários.”

É simples adicionar/eliminar funcionários da plataforma.

23 Respostas

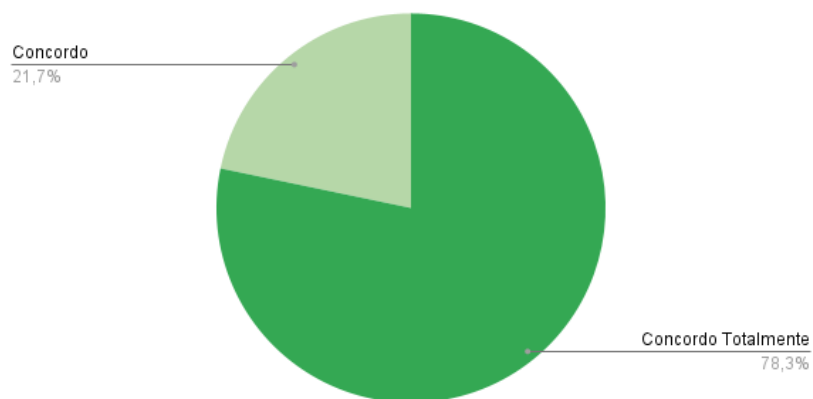


Figura E.35: Respostas à pergunta “É simples adicionar/eliminar funcionários da plataforma.”

É simples consultar o relatório e classificação do serviço.

21 Respostas

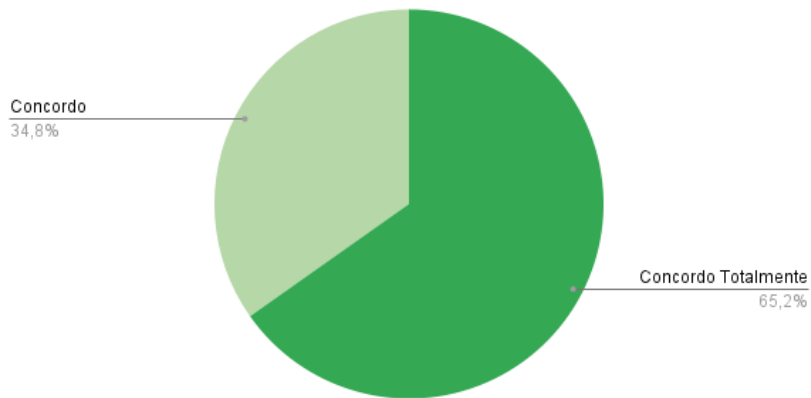


Figura E.36: Respostas à pergunta “É simples consultar o relatório e classificação do serviço.”

Foi simples definir o intervalo relevante no ecrã de KPI.

21 Respostas

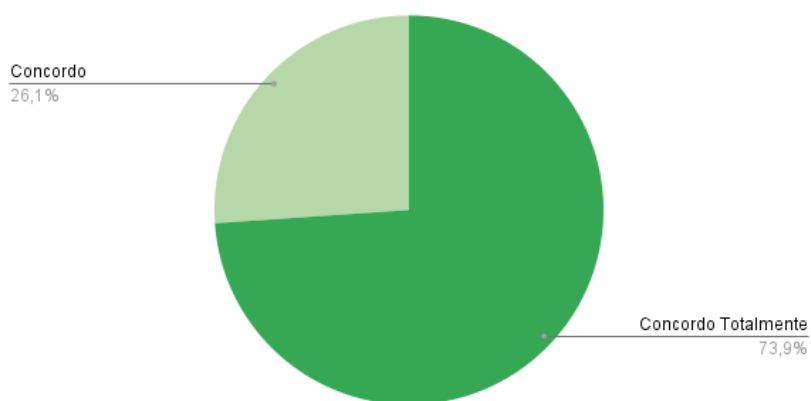


Figura E.37: Respostas à pergunta “Foi simples definir o intervalo relevante no ecrã de KPI.”

O mapa geral foi de fácil navegação.

21 Respostas

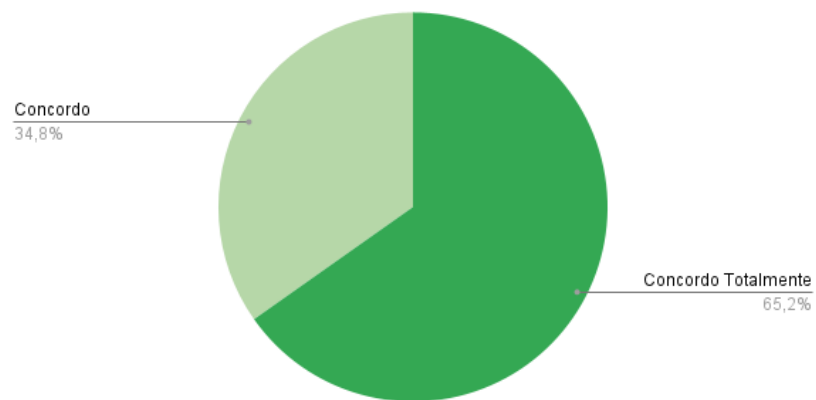


Figura E.38: Respostas à pergunta “O mapa geral foi de fácil navegação.”