



Licenciatura em Engenharia Informática e Multimédia (LEIM)

OnBoarding Document Capture



Realizado por:
Ana Padeiro, 45148
Artur Paredes, 46347

Orientadores:
Professora Paula Graça
Engenheiro Rúben Oliveira

Ano Letivo 2020/21
11 julho de 2021

Resumo

O presente relatório descreve o desenvolvimento de uma aplicação para dispositivos móveis, OnBoarding Document Capture, para a unidade curricular Projeto integrada na Licenciatura de Engenharia Informática e Multimédia no Instituto Superior de Engenharia de Lisboa, com a orientação da Professora Paula Graça e do Engenheiro Rúben Oliveira da empresa Do It Lean.

A motivação para este projeto primeiramente foi poder aprender uma nova tecnologia chamada OutSystems, que nunca foi utilizada ao longo da licenciatura, e utilizá-la para desenvolver uma aplicação que permita ao utilizador guardar tanto documentos pessoais como faturas.

Nesta aplicação o utilizador pode inserir os dados de faturas (Eletricidade, Água, etc.) ou documentos pessoais (Cartão de cidadão, Cartão bancário, Carta de condução e Passaporte) manualmente e/ou fazendo *upload* de uma imagem em que serão extraídos os dados e de seguida inseridos na aplicação a partir de um processo automático. Para esta extração foi utilizada a plataforma *Form Recognizer* da Microsoft Azure Cognitive Services que utiliza mecanismos de *Machine Learning (ML)* para extrair com precisão texto de tabelas e de documentos. O utilizador pode ainda ser notificado quando falta um número de dias definido por este, para terminar o prazo de um documento ou fatura. Permite ainda o acesso a um calendário com as datas de fins de prazo dos pagamentos e datas de vencimento dos documentos, organizar os documentos e faturas por categorias e ter a possibilidade de pesquisá-los pelo nome e ainda ver os seus gastos através de gráficos e estatísticas.

Concluindo, este projeto permitiu adquirir diferentes experiências tais como o desenvolvimento de uma aplicação para dispositivos móveis em OutSystems, uma metodologia de desenvolvimento ágil, o uso de ferramentas de *ML* da Microsoft Azure Cognitive Services e ainda melhorar o trabalho em equipa. O resultado deste projeto foi bastante gratificante para o grupo tanto a nível de aspetto gráfico da aplicação e todas as suas funcionalidades, como também a nível de conhecimentos adquiridos.

Abstract

This report describes the development of an application for mobile devices, OnBoarding Document Capture, for the subject Project integrated in the Degree in Engenharia Informática e Multimédia in Instituto Superior de Engenharia de Lisboa, under the guidance of Teacher Paula Graça and Engineer Rúben Oliveira from the company Do It Lean.

The motivation for this project was primarily to learn a new technology called OutSystems, which was never used throughout the degree, and use it to develop an application that allows the user to save both personal documents and invoices.

In this application the user can enter the data of invoices (Electricity, Water, etc) or personal documents (Citizen Card, Bank Card, Driver's License and Passport) manually and/or by upload of an image in which data will be extracted and then inserted into the application from an automatic process. For this extraction, Microsoft Azure Cognitive Services' Form Recognizer platform will be used, which uses *ML* mechanisms to accurately extract text from tables and documents. The user can also be notified when there is a number of days defined by him, to finish the deadline of a document or invoice. You will also have access to a calendar with the deadlines of payments and due dates of the documents, you will be able to organize the documents and invoices by categories and be able to search them by name and even see your expenses through graphics and statistics.

In conclusion, this project allowed us to acquire different experiences such as the development of an application for mobile devices in OutSystems, the agile development methodology, the use of *ML* tools from Microsoft Azure Cognitive Services and also improve teamwork. The result of this project was very gratifying for the group both in terms of the user interface design of the application and all its functionalities, as well as in terms of acquired knowledge.

Agradecimentos

A elaboração do presente relatório e o desenvolvimento da aplicação não seriam possíveis sem o apoio dos nossos orientadores. Assim sendo queremos agradecer por todo a orientação e apoio que nos deram no desenvolver deste projeto.

Queremos ainda agradecer:

Às nossas famílias, por nos terem sempre apoiado e nos terem dado tudo o que é necessário para lutarmos pelos nossos sonhos.

A todos os docentes que contribuíram para a nossa formação ao longo da licenciatura, por todos os conhecimentos, dedicação e contributo para o nosso crescimento pessoal e educacional.

Aos nossos amigos, que estiverem sempre connosco para nos apoiarem com todo o carinho e compreensão.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Lista de Figuras	ix
Lista de Tabelas	xiii
Lista de Acrónimos	xv
1 Introdução	1
2 Estado da Arte	3
2.1 Serviço Payper	3
2.2 Aplicação Billdu	5
2.3 Aplicação Expensify	7
2.4 Aplicação BlinkID	8
2.5 <i>Software IRISmart™</i>	9
2.6 Aplicação Veryfi Receipts OCR & Expenses	10
2.7 Aplicação CamScanner	11
2.8 Comparação entre Aplicações Analisadas	12
3 Análise de Requisitos	13
3.1 Síntese de Objetivos	13
3.2 Público alvo	13
3.3 Metas a Alcançar	14
3.4 Funções do Sistema	14
3.5 Casos de Utilização	17
3.6 <i>Mockups</i>	22

4 Plataformas	25
4.1 Trello	25
4.2 OutSystems	27
4.3 Microsoft Azure Cognitive Services	27
5 Implementação	29
5.1 Modelo da Base de Dados	29
5.1.1 Utilizador	31
5.1.2 Documento	31
5.1.3 Fatura	34
5.1.4 Categoria	35
5.1.5 Problema	36
5.1.6 Configuração de Notificação	36
5.1.7 Notificação	37
5.2 Arquitetura	38
5.3 Implementação OutSystems	39
5.3.1 Configuração e Criação de modelos	49
5.3.2 Extração de dados	54
6 Resultados Obtidos	57
6.1 Testes Funcionais	57
6.2 Testes de Usabilidade	62
7 Conclusões	67
Bibliografia	69
Anexos	71

Lista de Figuras

2.1 Aplicação Payper	4
2.2 Aplicação Billdu	6
2.3 Aplicação Expensify	7
2.4 Aplicação BlickID	8
2.5 Software IRISmart™	9
2.6 Software Veryfi Receipts OCR & Expenses	10
2.7 Aplicação CamScanner	11
3.1 Casos de Utilização - Simplificado	17
3.2 Casos de Utilização - Autenticar-se na aplicação	18
3.3 Casos de Utilização - Mudar as definições da aplicação	18
3.4 Casos de Utilização - Ver as notificações	19
3.5 Casos de Utilização - Adicionar Documento	19
3.6 Casos de Utilização - Adicionar Fatura	20
3.7 Casos de Utilização - Ver Lista	20
3.8 Casos de Utilização - Ver Gráficos e Estatísticas	21
3.9 Casos de Utilização - Ver Calendário	21
3.10 Mockups do Use Case - Adicionar Documento	22
3.11 Mockups do Use Case - Mudar as definições da aplicação	23
5.1 Modelo de Entidade-Associação Simplificado	30
5.2 Tabela de Utilizadores	31
5.3 Tabela do Documento	31
5.4 Tabela do Cartão de Cidadão	32
5.5 Tabela da Carta de Condução	32
5.6 Tabela do Cartão Bancário	33
5.7 Tabela do Passaporte	33
5.8 Tabela da FotoDocumento	34
5.9 Tabela da Fatura	34
5.10 Tabela da FotoFatura	35
5.11 Tabela da Categoria	35

5.12	Tabela do Problema	36
5.13	Tabela da ConfiguracaoNotificacao	36
5.14	Tabela da Notificacao	37
5.15	Arquitetura da Aplicação OnBoarding Document Capture . .	38
5.16	Variável do tipo Fatura	39
5.17	Agregado com duas tabelas	39
5.18	Exemplo de uma <i>client action</i>	40
5.19	Ecrã de Adicionar Fatura	41
5.20	Componentes dentro de um ecrã	41
5.21	Bloco de um ecrã	42
5.22	Componentes de um bloco	43
5.23	Widget Tree do ecrã "Adicionar Fatura"	43
5.24	Atributos de um input widget	44
5.25	Exemplos de Widgets disponíveis	44
5.26	Widgets do ecrã Adicionar Fatura	45
5.27	Validação de um campo	46
5.28	<i>Dropdown</i> da lista de categorias	46
5.29	Popup de adicionar uma categoria	47
5.30	Adicionar uma fatura (client action)	47
5.31	Adicionar uma fatura (server action)	48
5.32	Selecionar Connections	49
5.33	Lista de Contentores	50
5.34	Criação do Contentor	50
5.35	Dados para a criação da Conexão	51
5.36	Criar Projeto	52
5.37	Resultados do treino de um modelo	53
5.38	Fazer <i>upload</i> de uma imagem	54
5.39	Ligaçao da aplicação com os serviços cognitivos da Azure . .	55
5.40	Fatura exemplo	56
5.41	Dados extraídos de uma fatura	56
6.1	Validação dos campos obrigatórios no Cartão de Cidadão . .	58
6.2	Testes aos campos Nome e N°ID Civil no Cartão de Cidadão .	59
6.3	Testes da tarefa 'Alterar palavra-passe' - Erros	60
6.4	Testes da tarefa 'Alterar palavra-passe' - Validação	60
6.5	Teste à barra de pesquisa da lista de documentos	61
6.6	Exemplo de uma pergunta do questionário aos utilizadores .	62
6.7	Respostas à pergunta da Figura 6.6	62
6.8	Resultados da tarefa "Configurar Notificações" por parte dos utilizadores	63

6.9 Resultados da tarefa "Aceder ao calendário" por parte dos utilizadores	63
6.10 Resultados da tarefa "Ver página detalhada de uma fatura" por parte dos utilizadores	64
6.11 Resultados da tarefa "Apagar um documento" por parte dos utilizadores	64
6.12 Resultados da tarefa "Adicionar uma categoria" por parte dos utilizadores	65
6.13 Sugestões dos Utilizadores	66

Lista de Tabelas

2.1	Comparação das Aplicações	12
3.1	Funções do Sistema	16

Lista de Acrónimos

API Application Programming Interface. 27, 51

ML Machine Learning. i, iii, 1, 13

Capítulo 1

Introdução

O presente relatório descreve o desenvolvimento de uma aplicação para dispositivos móveis, OnBoarding Document Capture, para a unidade curricular Projeto integrada na Licenciatura de Engenharia Informática e Multimédia no Instituto Superior de Engenharia de Lisboa, com a orientação da Professora Paula Graça e do Engenheiro Rúben Oliveira da empresa Do It Lean.

A motivação para este projeto primeiramente foi poder aprender uma nova tecnologia chamada OutSystems, que nunca foi utilizada ao longo da licenciatura, e utilizá-la para desenvolver uma aplicação que permita ao utilizador guardar tanto documentos pessoais como faturas. Nesta aplicação o utilizador pode inserir os dados de faturas (Eletricidade, Água, etc) ou documentos pessoais (Cartão de cidadão, Cartão bancário, Carta de condução e Passaporte) manualmente e/ou fazendo *upload* de uma imagem em que serão extraídos os dados e de seguida inseridos na aplicação a partir de um processo automático. Para esta extração foi usada a plataforma *Form Recognizer* da Microsoft Azure Cognitive Services que utiliza mecanismos de *ML* para extrair com precisão texto de tabelas e de documentos. O utilizador pode ainda ser notificado quando falta um número de dias definido por este, para terminar o prazo de um documento ou fatura. Permite o acesso a um calendário com as datas de fins de prazo dos pagamentos e datas de vencimento dos documentos, organizar os documentos e faturas por categorias e ter a possibilidade de pesquisá-los pelo nome e ainda ver os seus gastos através de gráficos e estatísticas.

A OutSystems é uma plataforma *low-code* que permite o desenvolvimento de aplicações *web* e *mobile*. O *low-code* é uma metodologia virada para a criação de *software* e aplicações que utiliza pouco código sendo assim bastante útil pois garante um aumento de produtividade. Para ter um melhor conhecimento da plataforma antes de dar início ao desenvolvimento da

aplicação, foram feitos alguns tutoriais com o intuito de aprender como funciona e como se poderá utilizá-la para cumprir com o objetivo definido. Foram feitos os módulos *"Becoming a Reactive Web Developer"* [9] e *"Becoming a Mobile Developer"* [10] dos tutoriais da OutSystems.

Foi utilizada a ferramenta Trello, sugerida pelos orientadores, que permite o planeamento de projetos seguindo a metodologia ágil com a *framework* SCRUM. A *framework* SCRUM é uma *framework* de desenvolvimento ágil utilizada no desenvolvimento de *software* baseada em processos iterativos e incrementais. O principal objetivo do SCRUM é satisfazer a necessidade do cliente, sendo para isso dividido em *Sprints*. Um *Sprint* tem uma duração predefinida, em que durante esse tempo o programador terá uma lista de tarefas a desempenhar que serão definidas no início de cada um, e quando chegar ao fim do mesmo será apresentado ao cliente tudo o que foi implementado. Este método garante que exista uma melhor organização de tempo e que sejam alcançados todos os requisitos do cliente, permitindo que esteja sempre a par do que está a ser desenvolvido.

Antes de dar início ao desenvolvimento, foi feito um estudo do mercado para saber que aplicações já existiam em semelhança ao que se pretendia desenvolver para se poder melhorar o conceito e acrescentar funcionalidades que não tivessem sido ainda planeadas.

A partir do estudo das aplicações já existentes foi feita uma lista das funcionalidades e das tarefas desejáveis na aplicação. A partir desta lista foram desenvolvidos os casos de utilização, que demonstram as tarefas que o utilizador poderá desempenhar e o comportamento esperado pelo sistema a criar. De seguida foram desenhados os *mockups* da aplicação, sendo estes desenhos dos ecrãs e ideias base de design da interface gráfica.

Foi ainda desenvolvido o modelo de entidade-associação, que tem como objetivo descrever e definir a base de dados a implementar, e o diagrama da arquitetura, onde serão especificados os diferentes módulos da aplicação.

Finalizado o processo de pré-aplicação, foram aplicadas as ideias em OutSystems, desde a criação da base de dados como também os ecrãs. Durante o processo de desenvolvimento foram feitos alguns questionários a vários utilizadores com intuito de melhorar não só o *design* gráfico como também as diferentes funcionalidades.

Capítulo 2

Estado da Arte

O estudo do estado da arte permite compreender e conhecer ferramentas disponíveis com objetivos ou características semelhantes à aplicação a desenvolver, permitindo também obter novas ideias para *design* e funcionalidades interessantes para acrescentar.

Para isso serão apresentadas algumas aplicações/*software* disponíveis semelhantes ao produto que se pretende desenvolver, tendo ainda uma tabela comparativa entre as diferentes aplicações/*software*.

2.1 Serviço Payper

O Payper [1] é um serviço gratuito online, que permite gerir todas as faturas de casa. Podendo ser utilizado tanto no próprio *site* como na aplicação que está representada na Figura 2.1. Ao serem inseridas as faturas este será capaz de fazer a leitura dos dados para poder fazer a comparação a outras companhias.

A aplicação fornece as seguintes funcionalidades:

- Gestor online de todas as faturas;
- Avisa caso os preços aumentem;
- Avisa se existe um melhor tarifário;
- Avisa quando a fidelização termina.

Com isto será possível que o utilizador consiga aceder a melhores preços de outras companhias do mesmo serviço, podendo considerar alteração de contrato para poder poupar dinheiro.

Algumas lacunas da aplicação, tendo em conta os comentários dos utilizadores:

- Erro ao inserir uma nova fatura;
- Problemas na interface do utilizador.

Avaliação na Play Store: 3,1/5,0

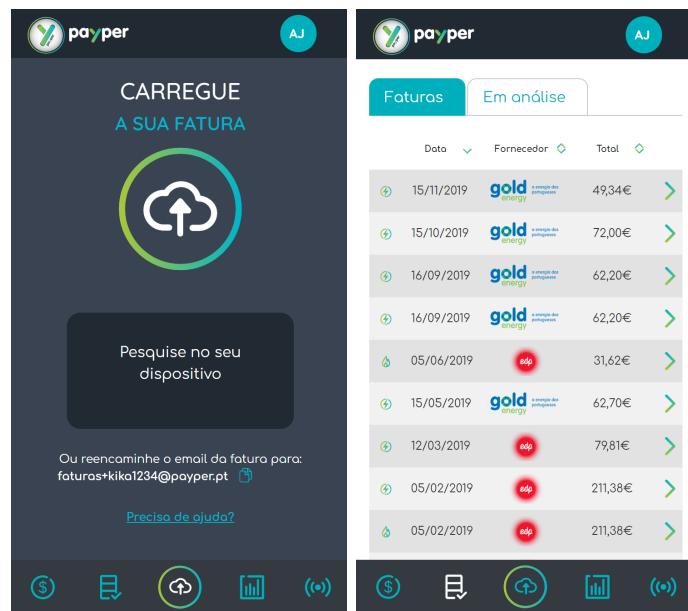


Figura 2.1: Aplicação Payper

2.2 Aplicação Billdu

A billdu [2] é uma aplicação que gera faturas, orçamentos e ordens de compra. É bastante útil para empresas porque tem imensas ofertas relacionadas a gerenciamento, nomeadamente criar faturas, permite acompanhar as despesas através da digitalização de faturas, os dados são sincronizados em todos os dispositivos por isso é possível partilhar com o contabilista e muito mais. Na Figura 2.2 é possível visualizar o aspetto gráfico da aplicação.

A aplicação fornece as seguintes funcionalidades:

- Digitalização de faturas para acompanhar as despesas;
- Receber notificações sobre que fatura foi aberta e por quem;
- Ligação com a contabilidade;
- Lembretes de pagamentos;
- Faturação online;
- Monitorização de inventário;
- Digitalização de códigos de barra para facilitar a administração de itens e a faturação.

Opiniões dos utilizadores que utilizaram a aplicação:

- Aplicação muito boa;
- Apenas é gratuita nas duas primeiras semanas de utilização.

A aplicação fornece ainda planos de assinatura que variam entre 2,99€/mês e 17,99€/mês, ainda existe a possibilidade de pagar pela anualidade que varia entre 23,99€/ano e 179,88€/ano.

Avaliação na Play Store: 4,7/5.0

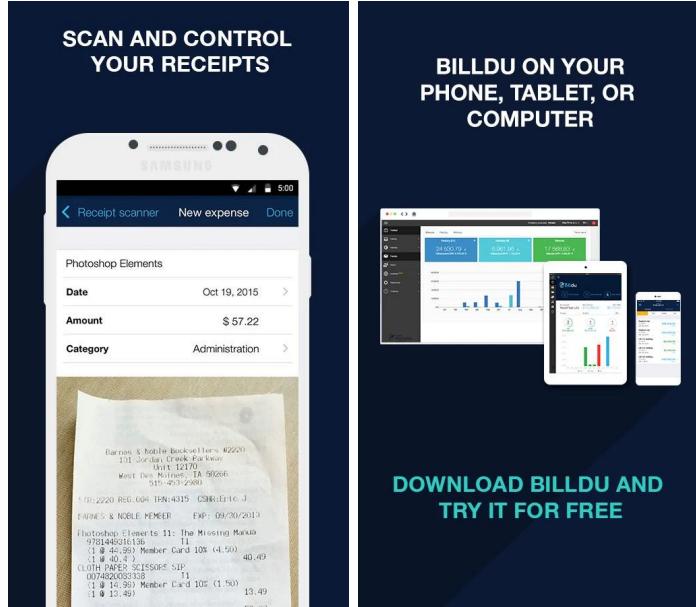


Figura 2.2: Aplicação Billdu

2.3 Aplicação Expensify

Esta aplicação [3], representada na Figura 2.3, permite digitalizar recibos podendo ver as despesas que estão a ser feitas e dividindo ainda os recibos por pastas de modo a haver uma maior organização. Ainda possui um assistente de marcação de viagem 14 horas por dia, onde pode marcar uma viagem completa, desde voo, hotel e carros.

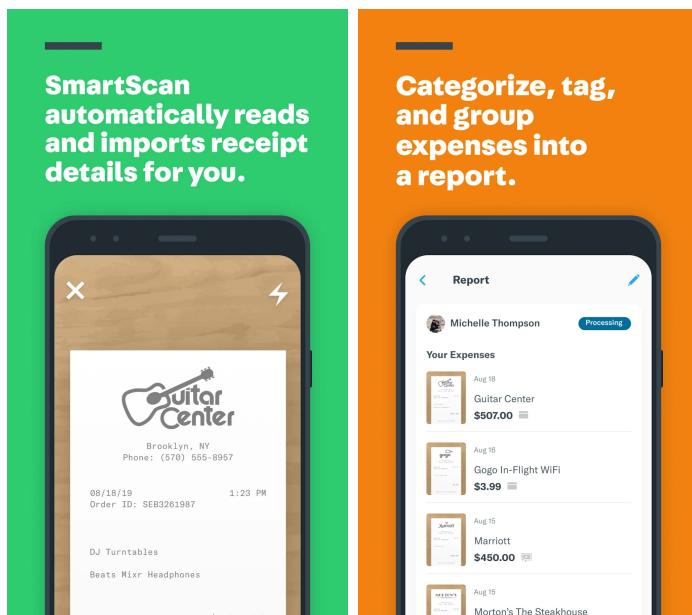


Figura 2.3: Aplicação Expensify

Utilidades da aplicação:

- Integrações de recibo;
- Exportações de relatórios personalizados;
- Conversor de moeda automático;
- Importação automática de cartão de crédito.

Algumas lacunas da aplicação, referidas por utilizadores:

- Processo muito lento;
- Utilidade demasiado complexa;
- Subscrição obrigatória e sem informação sobre isso.

Avaliação na Play Store: 3,5/5.0

2.4 Aplicação BlinkID

A BlinkID [4], representada na Figura 2.4, foi criada pela empresa Microblink Ltd. que tem como visão o desenvolvimento de tecnologia que resolva problemas da vida real.

A aplicação permite guardar todos os documentos na mesma aplicação, digitalizar um documento e de imediato extrai todas as informações pessoais de cada um. Ainda notifica os utilizadores quando os documentos estão a chegar ao fim da sua validade. A aplicação é livre de anúncios e é totalmente gratuita.

É possível extrair informação de documentos de identidade, passaportes, cartas de condução, vistos e permissões de trabalho.

Opiniões dos utilizadores:

- Demora a processar o documento;
- Aplicação muito boa no geral.

Avaliação na Play Store: 4,7/5.0

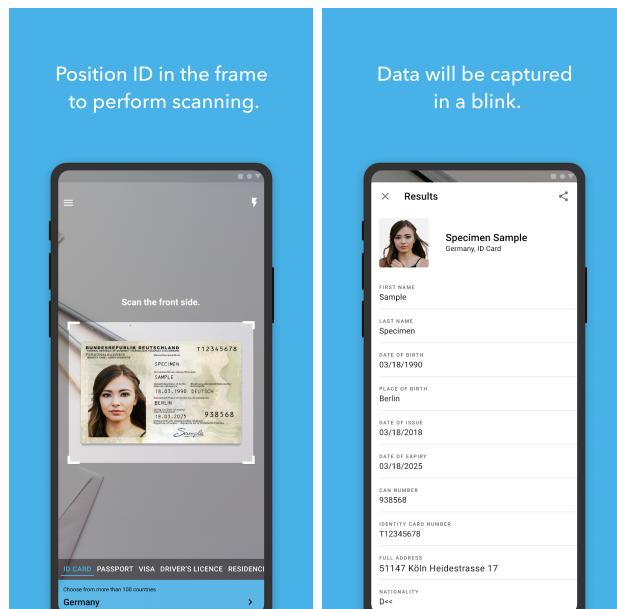


Figura 2.4: Aplicação BlinkID

2.5 Software IRISmart™

IRISmart™ [5] é um *software* criado para Windows que ajuda a extrair os dados pessoais através dos documentos de identificação, como é possível ver na Figura 2.5, pode ser bastante útil para um dono de um hotel ou de uma empresa pois terá acesso aos documentos pessoais do seu cliente ou do seu trabalhador.

Este *software* é gratuito nos primeiros 10 dias, depois o utilizador terá de o comprar por 199\$.



Figura 2.5: Software IRISmart™

2.6 Aplicação Veryfi Receipts OCR & Expenses

Esta aplicação [6] tem como principal funcionalidade a deteção de informação de faturas e recibos, tais como o vendedor, totais, itens comprados, data e hora. Tem também a capacidade de detetar informação escrita à mão. A aplicação é gratuita apenas nos primeiros 14 dias.

Na Figura 2.6 é representado uma utilização da aplicação onde é possível verificar que a partir de uma foto de uma fatura são extraídos os dados.

Opiniões dos utilizadores:

- Processamento rápido das faturas;
- Aplicação muito boa no geral;
- Alguns problemas com as subscrições.

Avaliação na Play Store: 4,4/5.0

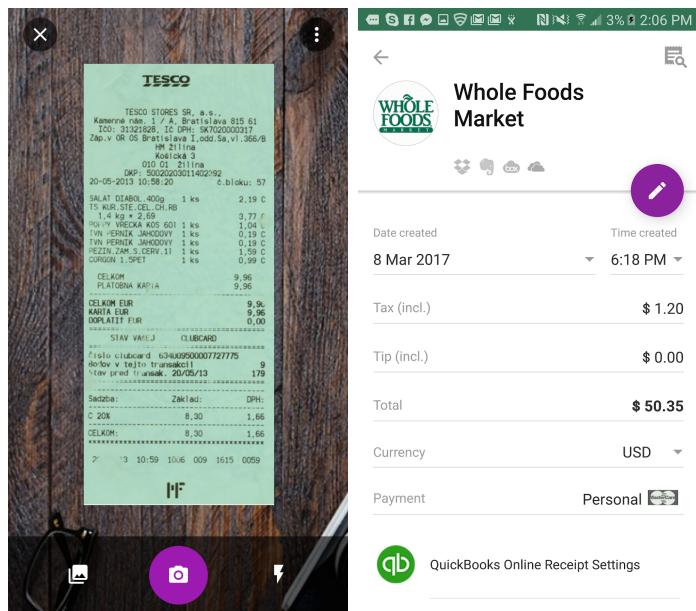


Figura 2.6: Software Veryfi Receipts OCR & Expenses

2.7 Aplicação CamScanner

O CamScanner [7] ajuda a digitalizar, guardar e sincronizar vários conteúdos através de *smartphones*, *iPads*, *tablets* e computadores. É possível visualizar o aspeto gráfico da aplicação na figura 2.7. Apenas é necessária a câmara do dispositivo, aceita documentos em formato PDF ou JPEG. Recursos da aplicação *Premium* com um custo de \$ 4.99/mês ou de \$49.99/ano.

Permite digitalizar e gerir:

- Recibos, faturas, contratos, cartões-de-visita;
- Quadros brancos, cartas;
- Notas, PPT, livros, artigos;
- Certificados, documentos oficiais.

A opinião dos utilizadores foi que quando o período de experimentação acaba têm alguns problemas e que a aplicação tem muitos anúncios. Mas num modo geral é muito boa e faz exatamente o que é prometido.

Avaliação na Play Store: 4,7/5.0



Figura 2.7: Aplicação CamScanner

2.8 Comparação entre Aplicações Analisadas

A Tabela 2.1 apresenta uma comparação entre as funcionalidades das aplicações referidas anteriormente, nomeadamente se suporta documentos pessoais, faturas, se tem custo associado à utilização da aplicação e ainda a avaliação dada pelos utilizadores na Play Store e o número de instalações e de avaliações (visto em 31/03/2021).

Nome	Suporta leitura de documentos pessoais	Suporta leitura de faturas	Tem custo	Número de instalações	Número de avaliações	Avaliação
Payper	Não	Sim	Não	50.000+	302	3,1
Billdu	Não	Sim	Opcional	100.000+	3.822	4,7
Expensify	Não	Sim	Sim	1.000.000+	15.186	3,5
BlinkID	Sim	Não	Não	100.000+	1.294	4,5
IRISmart™	Sim	Não	Sim	-	-	-
Veryfi Receipts	Não	Sim	Não	50.000+	748	4,4
CamScanner	Sim	Sim	Opcional	100.000.000+	3.280.678	4,7

Tabela 2.1: Comparação das Aplicações

Capítulo 3

Análise de Requisitos

A análise de requisitos é uma parte importante no processo de desenvolvimento de *software*, onde são identificadas as necessidades do utilizador ao utilizar a aplicação. Na análise do Onboarding Document Capture foi feita uma caracterização geral destas tarefas e outras ao pormenor.

A caracterização geral contém numa forma geral a aplicação, descrevendo os clientes que quererão utilizar a aplicação e definindo as metas a alcançar com este projeto.

Na caracterização ao pormenor irão ser definidas as funções do sistema de modo a se obter uma ideia clara das tarefas que o utilizador pode realizar.

3.1 Síntese de Objetivos

Neste projeto será criada uma aplicação utilizando a ferramenta OutSystems e mecanismos de *ML*.

Tem como principal objetivo adicionar e guardar documentos e faturas no mesmo local, podendo os dados destes serem escritos à mão ou extraídos a partir de imagens, sendo possível depois listá-los e agrupá-los como se pretender.

3.2 Públco alvo

O público alvo para a utilização desta aplicação são todos os utilizadores que tenham um telemóvel com acesso à Internet, e que desejem guardar as suas faturas e documentos todos no mesmo local para uma melhor organização e facilidade de acesso.

3.3 Metas a Alcançar

Pretende-se que este sistema contribua para:

- Guardar dados dos documentos pessoais e faturas do utilizador;
- Extração de dados de documentos pessoais ou faturas através de uma fotografia;
- Extração de dados de uma forma rápida e correta;
- Visualização agrupada de documentos ou faturas;
- Visualização de gráficos, estatísticas e calendário de pagamentos efetuados, pagamentos a efetuar e datas de documentos vencidos;
- Ser notificado da expiração de documentos e do prazo final de pagamentos de faturas.

3.4 Funções do Sistema

As funções do sistema representam aquilo que o sistema é suposto fazer, para isso foi utilizada a Tabela 3.1 para definir cada tarefa, em que a coluna Requisito trata-se do nome da tarefa, Função é a descrição da tarefa e Agrupamento é uma maneira de dividir as tarefas em diferentes conceitos, como por exemplo, agrupamento ”apagar” terá todas as tarefas que forem com o intuito de apagar algo.

Requisito	Função	Agrupamento
Adicionar documento	Adicionar novo documento	Adicionar
Adicionar fatura	Adicionar nova fatura	Adicionar
Adicionar dados da fatura	Adicionar dados da fatura manualmente	Adicionar
Adicionar fatura a categoria	Adicionar a fatura a uma categoria	Adicionar
Adicionar dados do documento	Adicionar dados de um documento manualmente	Adicionar

Requisito	Função	Agrupamento
Apagar fatura	Apagar fatura	Apagar
Apagar documento	Apagar documento	Apagar
Atualizar lista de documentos	Atualizar lista de documentos quando é adicionado/apagado um documento ou categoria	Atualização
Atualizar lista de faturas	Atualizar lista de faturas quando é adicionada/apagada uma nova fatura	Atualização
Atualizar dados do documento	Atualizar dados associados a um documento	Atualização
Atualizar dados da fatura	Atualizar dados associados a uma fatura	Atualização
Registo	Criar uma conta na aplicação	Autenticação
Login	Entrar na aplicação com email e palavra-passe	Autenticação
Logout	Sair da aplicação	Autenticação
Autenticar	Autenticar utilizador (email e palavra-passe)	Autenticação
Visualizar calendário	Visualizar calendário com data de pagamentos de faturas e com data de vencimento de documentos	Calendário
Copiar dados da fatura	Copiar os dados lidos da fatura para a <i>clipboard</i>	Copiar dados
Copiar dados do documento	Copiar os dados lidos do documento para a <i>clipboard</i>	Copiar dados
Definições de notificações	Mudar opções de notificações	Definições
Definições gerais	Aceder as definições gerais da aplicação	Definições
Editar fatura	Editar os dados extraídos da fatura	Editar
Editar fatura	Editar dados da fatura	Editar
Mudar o estado do pagamento	Poder dar marcar uma fatura como paga/não paga	Editar
Editar documento	Editar dados dos documentos	Editar
Visualizar um gráfico	Visualizar um gráfico fléxivel (dia, semana, mês, ano) dos pagamentos	Gráfico

Requisito	Função	Agrupamento
Visualizar lista de documentos	Ver lista de documentos	Lista de documentos
Visualizar lista de documentos	Ordenar os documentos pela data de vencimento	Lista de documentos
Visualizar lista de faturas	Ver a lista de faturas filtrada (categorias, pago, não pago)	Lista de faturas
Visualizar lista de faturas	Ver a lista de faturas ordenada (data)	Lista de faturas
Filtrar lista de faturas	Escolher filtro da visualização das faturas	Lista de faturas
Extração de dados	Extrair dados através de uma fotografia de uma fatura	Machine Learning
Visualizar notificações	Visualizar notificações	Notificações
Visualizar notificações	Apagar notificação	Notificações
Ocultar dados da fatura	Ocultar dados da fatura	Ocultar dados
Ocultar dados do documento	Ocultar dados do documento	Ocultar dados
Visualizar lista de faturas	Pesquisa de faturas pelo nome	Pesquisa
Visualizar lista de documentos	Pesquisa de documentos pelo nome	Pesquisa
Adicionar foto da fatura	Fazer <i>upload</i> ou tirar foto a uma fatura	<i>Upload/ tirar foto</i>
Adicionar foto do documento	Fazer <i>upload</i> ou tirar foto de um documento	<i>Upload/ tirar foto</i>

Tabela 3.1: Funções do Sistema

3.5 Casos de Utilização

Os casos de utilização descrevem as funcionalidades propostas do sistema a desenvolver, é de algum modo bastante semelhante aos elementos da tabela 3.1. Primeiramente foi feito um diagrama com as funcionalidades de forma mais generalizada, representado na figura 3.1, e de seguida, para cada uma destas, são representados os seus diagramas de forma detalhada.

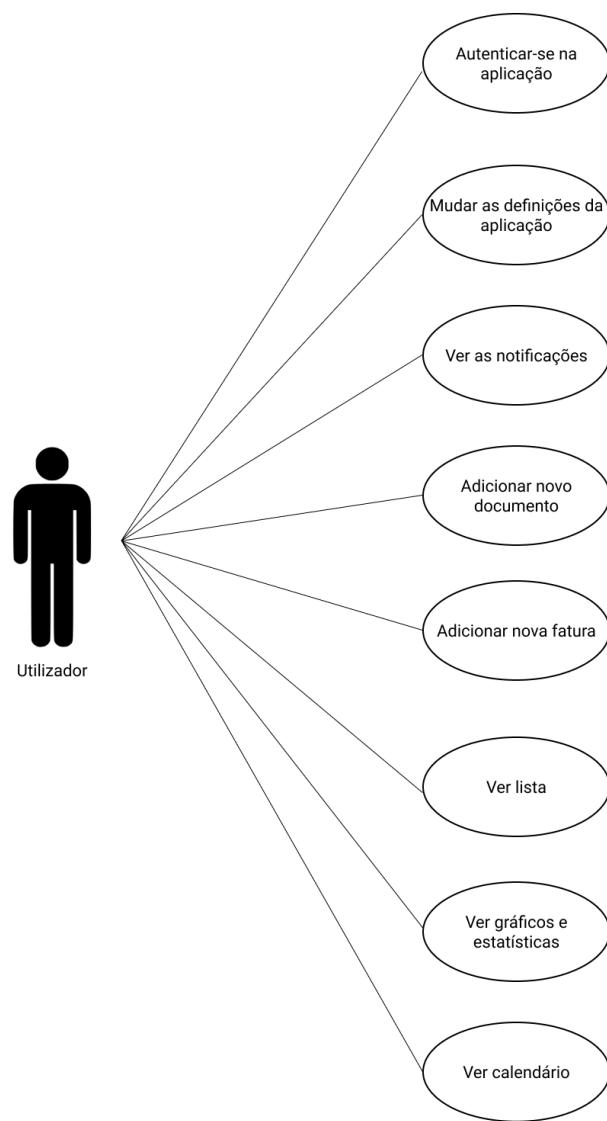


Figura 3.1: Casos de Utilização - Simplificado

O módulo ”Autenticar-se na aplicação” descreve o processo que o utilizador terá de completar para se autenticar na aplicação, como se pode ver na figura 3.2. Para a autenticação o utilizador terá de fazer o registo para poder efetuar o *login*.

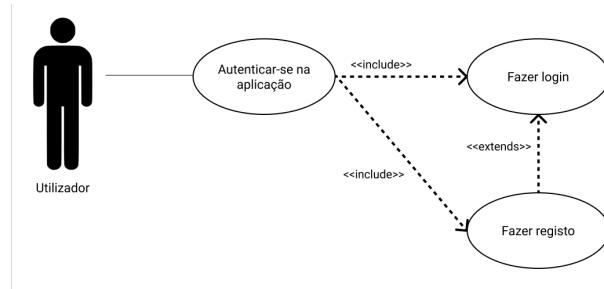


Figura 3.2: Casos de Utilização - Autenticar-se na aplicação

Na figura 3.3 é possível ver o módulo ”Mudar as definições da aplicação” que permite ao utilizador mudar a *password*, configurar as notificações, relatar um problema e fazer o *logout*.

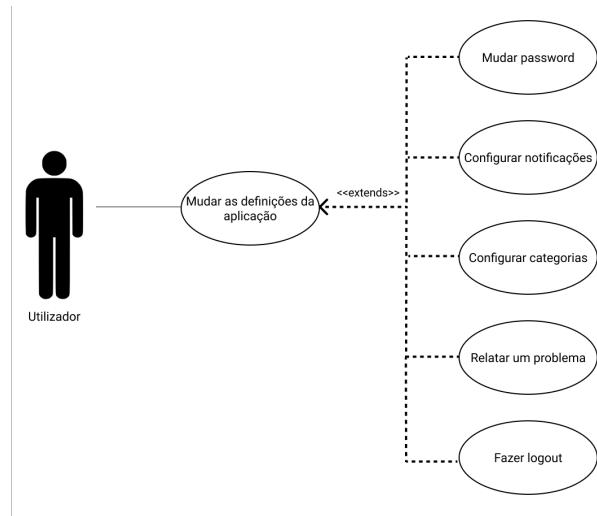


Figura 3.3: Casos de Utilização - Mudar as definições da aplicação

No módulo "Ver as notificações" o utilizador poderá apagar a notificação caso já não a pretenda ver novamente ou pode clicar na notificação e seguir para a fatura ou documento correspondente, como se pode observar na figura 3.4.

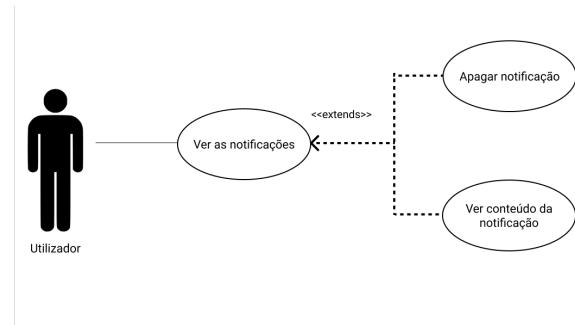


Figura 3.4: Casos de Utilização - Ver as notificações

No módulo "Adicionar Documento", ilustrado na figura 3.5, o utilizador terá a possibilidade de inserir dados manualmente ou fazer *upload* de uma foto do qual serão extraídos os dados, que podem ser alterados pelo utilizador.

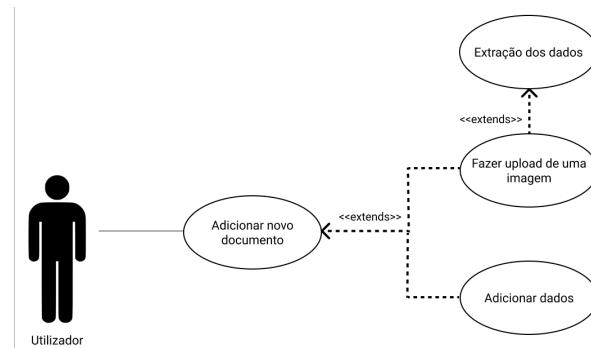


Figura 3.5: Casos de Utilização - Adicionar Documento

No módulo ”Adicionar Fatura” o utilizador tem a opção de adicionar os dados manualmente ou fazer *upload* de uma foto do qual serão extraídos os dados, onde poderão ainda ser alterados pelos utilizador. Este módulo está representado de seguida na figura 3.6.

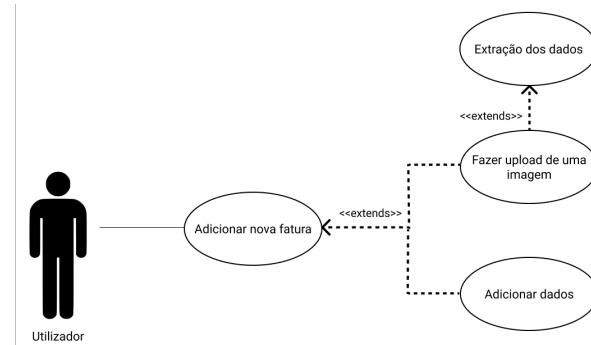


Figura 3.6: Casos de Utilização - Adicionar Fatura

No módulo ”Ver lista”, representado na figura 3.7, o utilizador tem a possibilidade de ver as listas de documentos e faturas, onde este poderá selecionar um elemento da lista e ver o seu conteúdo detalhadamente. Ao ver o conteúdo pode ainda editar o mesmo, apagá-lo, copiar os dados ou ocultá-los.

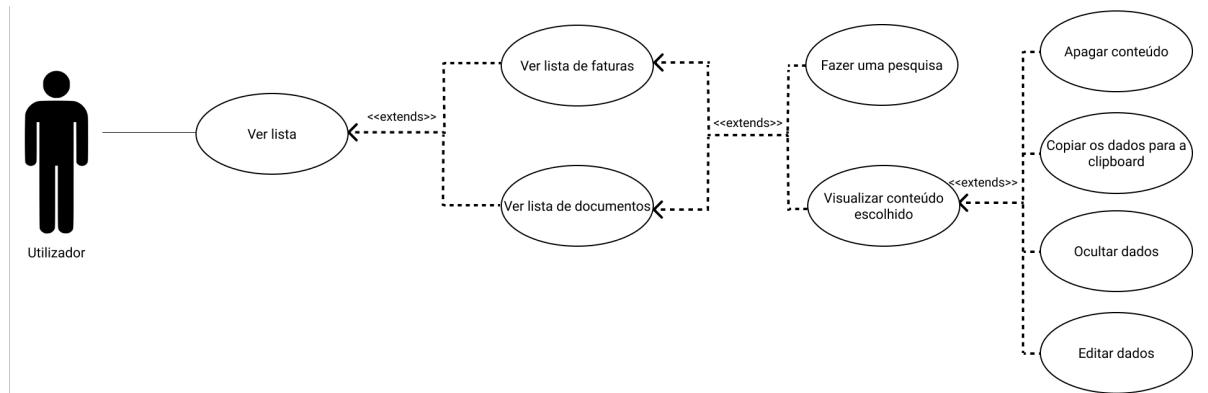


Figura 3.7: Casos de Utilização - Ver Lista

O utilizador poderá ver uma página onde tem os valores dos seus gastos, através de uma lista de valor total gasto por tipo de fatura, um gráfico circular e um histograma, podendo ainda escolher se quer ver os gastos diários, semanais, mensais ou anuais. O módulo está representado na figura 3.8.



Figura 3.8: Casos de Utilização - Ver Gráficos e Estatísticas

No módulo ”Ver Calendário”, ilustrado na figura 3.9, o utilizador poderá ter acesso a um calendário que indica os dias em que a data de validade de um documento/fatura expira.

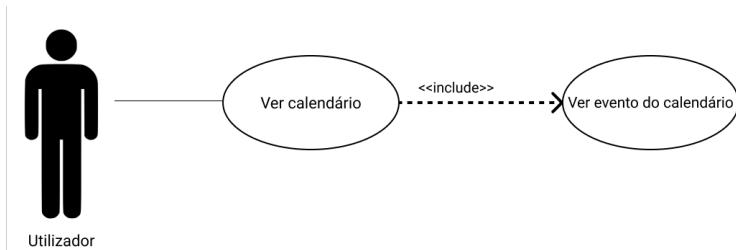


Figura 3.9: Casos de Utilização - Ver Calendário

3.6 *Mockups*

Para se obter uma ideia do que se pretendia desenvolver na aplicação, como a interface gráfica, tarefas, funcionalidades, limitações e experiência do utilizador em OutSystems, foram feitos alguns esboços dos ecrãs (denominados de *mockups*) que incluem as funcionalidades desejáveis. Na Figuras seguintes serão ilustrados alguns dos desenhos iniciais que foram desenvolvidos na ferramenta Figma, poderá ver os restantes no Capítulo 7 que é os Anexos.

Na figura 3.10 estão representados os ecrãs que representam o Use Case da figura 3.5 que é da tarefa ”Adicionar Documento” em que para isso o utilizador terá de ir ao ecrã ”Adicionar” onde poderá escolher o conteúdo a adicionar (documentos ou faturas), sendo redirecionado para a página onde poderá adicionar os dados do documento ou inserir uma imagem de onde serão extraídos os dados automaticamente.



Figura 3.10: *Mockups* do *Use Case* - Adicionar Documento

Na figura 3.11 será representado o *use case* da figura 3.3 que é da tarefa ”Mudar as definições da aplicação” onde o utilizador poderá mudar a palavra-passe, gerir as notificações que quer receber, gerir as categorias referentes às faturas, relatar possíveis problemas com a aplicação e por fim fazer o *logout* da aplicação.

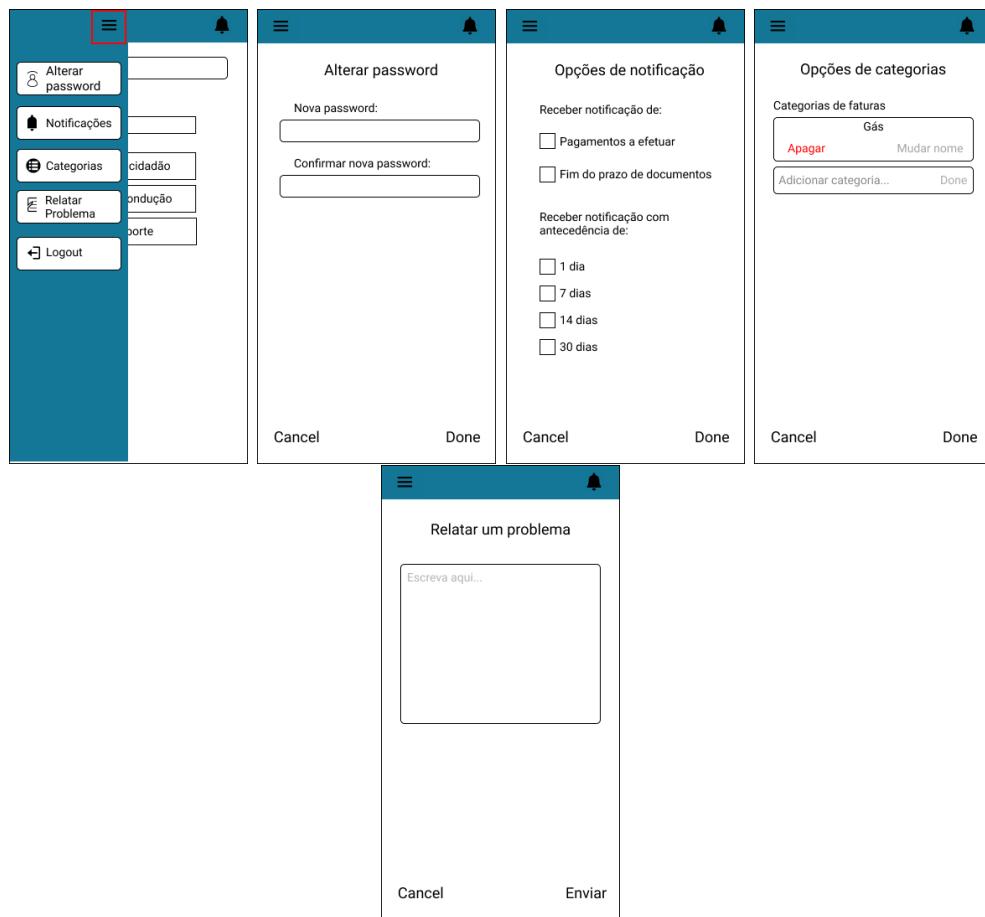


Figura 3.11: *Mockups* do *Use Case* - Mudar as definições da aplicação

Capítulo 4

Plataformas

Ao longo do desenvolvimento do projeto foram utilizadas três plataformas, o Trello que foi utilizada para a gestão e planeamento, a OutSystems para o desenvolvimento da aplicação e a Microsoft Azure Cognitive Services para a extração de dados.

4.1 Trello

O Trello é uma plataforma que permite a organização e gestão de tarefas a desenvolver num projeto, tendo uma interface intuitiva e recursos como *checklists*, *upload* de arquivos e etiquetas coloridas. Tem como principal característica as listas de cartões, sendo que cada cartão poderá conter uma imagem e texto, em que neste caso foi utilizada como fotografia um *mockup* que correspondia a uma funcionalidade da aplicação, e um texto que a descrevesse.

Foi apresentada e sugerida pelos orientadores como sendo uma plataforma que permite adotar uma metodologia ágil através da *framework* SCRUM. A *framework* SCRUM é uma *framework* de desenvolvimento ágil utilizada no desenvolvimento de software baseada em processos iterativos e incrementais. O principal objetivo do SCRUM é satisfazer a necessidade do cliente, sendo para isso dividido em *Sprints* que têm uma duração predefinida, em que durante esse tempo o programador terá uma lista de tarefas a desempenhar que serão definidas no início de cada um, e quando chegar ao fim do mesmo, será apresentado ao cliente tudo o que foi implementado. Este método garante que exista uma melhor organização de tempo por parte da equipa técnica, como também possibilita o acompanhamento regular do projeto por parte do cliente.

Nesta plataforma foram criadas 7 listas com o intuito de organizar todo o desenvolvimento:

- *Design*: Contém documentos pré-aplicação como o Estado da Arte, Análise de Requisitos, *Wireframes*, Casos de Utilização, Arquitetura e Modelo Entidade-Associação;
- *Backlog*: Contém todas as tarefas a desenvolver na aplicação, sendo que cada tarefa é representada por um cartão com uma imagem de um *mockup* e texto a explicar a sua funcionalidade;
- *To-Do*: No início de cada *Sprint* são selecionados alguns cartões do Backlog e inseridos nesta lista como sendo os objetivos para o *Sprint* atual, definindo em cada tarefa o tempo máximo para que esteja concluída;
- *Doing*: A equipa de programadores irá depois passar cada cartão do To-Do para esta lista, para que toda a equipa do projeto tenha conhecimento do que está a ser desenvolvido no momento;
- *Testing*: Assim que uma tarefa da lista *Doing* esteja concluída, esta será passada para a lista *Testing* onde será testada a sua funcionalidade;
- *Done*: Depois de passar a fase de testes, a tarefa será colocada nesta lista como estando concluída na sua totalidade;
- *Accepted*: No final do *Sprint* é realizada uma reunião em que a equipa de programadores irá mostrar aos orientadores as tarefas que foram desenvolvidas, e no caso de serem aceites irão ser colocadas nesta lista, caso contrário poderão ser pedidas melhorias ou alterações fazendo com que as tarefas voltem novamente a ser implementadas ou testadas.

4.2 OutSystems

A OutSystems é uma plataforma *low-code* que permite o desenvolvimento, publicação e manutenção de aplicações móveis e *web*, sendo principalmente conhecida pela sua facilidade de utilização, organização, implementação e gestão.

Uma aplicação poderá ter diferentes módulos, em que cada módulo contém 4 principais ferramentas:

- Processos: Onde serão criados processos, por exemplo *timers*, que irão chamar a uma hora definida uma atividade da aplicação. Neste caso foi utilizado como processo um *timer* que irá enviar notificações aos utilizadores a uma certa hora do dia;
- Interface: Onde é desenvolvida toda a interface gráfica da aplicação. Permite a criação de ecrãs, blocos, variáveis, agregados de dados da base de dados e *client actions* (lógica executada do lado do cliente);
- Lógica: Contém toda a lógica da aplicação como *client actions* e *server actions* (lógica executada do lado do servidor);
- Dados: Permite a criação, visualização e gestão de uma base de dados relacional.

É comum e uma boa prática ao desenvolver uma aplicação móvel em OutSystems, criar dois módulos, o módulo do *End User* onde será desenhada a interface gráfica, e o módulo do *Core* onde serão criados os processos, lógica e base de dados. Depois, no módulo do *End User* serão criadas dependências com o módulo do *Core*, para que sejam utilizadas as funcionalidades implementadas e permitindo assim uma melhor organização.

4.3 Microsoft Azure Cognitive Services

A Microsoft Azure é uma plataforma utilizada para a execução de aplicativos e serviços, sendo baseada no conceito de computação na nuvem. Os Serviços Cognitivos da Microsoft Azure permitem que desenvolvedores sem experiência em inteligência artificial, como aprendizagem automática, possam utilizá-la nas suas aplicações apenas com uma chamada de uma Application Programming Interface (API) . Destes serviços, foi utilizado o *Form Recognizer* que permite automaticamente extrair dados de um documento com texto.

Capítulo 5

Implementação

Neste capítulo será apresentada toda a parte da implementação, sendo ainda explicada a base de dados desenvolvida e a arquitetura da aplicação.

5.1 Modelo da Base de Dados

A utilização de uma base de dados neste tipo de aplicação é imprescindível por ser necessário guardar os dados de cada utilizador e as respetivas definições, documentos, faturas e notificações. Devido à existência de dependências entre entidades, foi criada uma base de dados relacional em OutSystems.

Com o intuito de organizar e visualizar estas dependências, foi criado um modelo Entidade-Associação em que inicialmente foram definidas as entidades e de seguida os respetivos atributos.

Uma Entidade é uma abstração para a descrição de objetos ou conceitos que possuam um conjunto de características comuns. Um Atributo é a característica comum aos objetos ou conceitos que a entidade retrata. Num conjunto de atributos de uma entidade é necessário eleger uma das suas chaves candidatas como chave principal, que se designa de chave primária.

Depois de definidas as entidades e atributos foram analisadas as associações que existiam entre as entidades, como por exemplo, a entidade Documento deverá estar relacionada à do Utilizador pois o Documento terá de ter um atributo referente ao Utilizador que a criou. Consequentemente dessas relações existem as chaves estrangeiras que são os atributos que "ligam" as duas entidades que no caso descrito anteriormente o Documento deverá ter um atributo que é o *UserId* para ter acesso ao identificador do utilizador que o inseriu.

Na Figura 5.1 será possível visualizar as relações entre as entidades e

também a chave primária de cada entidade, não foram representados os res-tantes atributos das entidades para facilitar a sua visualização.

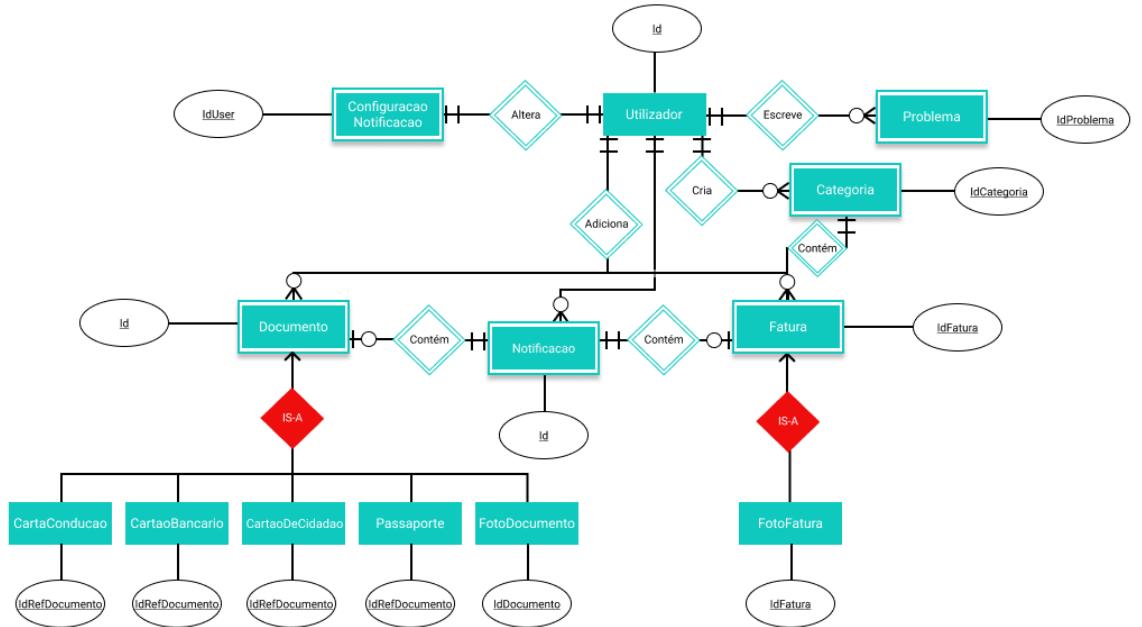


Figura 5.1: Modelo de Entidade-Associação Simplificado

Como é possível visualizar a entidade Utilizador tem relação com Documento, Fatura, Categoria, Problema, Configuração da Notificação e Notificação pois o utilizador poderá adicionar vários documentos e faturas, pode ainda adicionar categorias para associar às faturas, pode adicionar problemas com base na sua experiência com a aplicação, deverá ter associado uma configuração de notificação que irá guardar as preferências de notificações que pretende receber e por fim as notificações associadas com base na configuração que o utilizador definiu.

O documento tem também relacionado a si os vários tipos de documentos e ainda a FotoDocumento onde irão ser guardadas as fotos dos documentos.

Tal como no documento a fatura também terá associado FotoFatura onde serão guardadas as fotos das faturas.

Cada notificação terá relacionada a uma fatura ou a um documento.

5.1.1 Utilizador

A entidade Utilizador, apresentada na figura 5.2, neste caso corresponde à tabela já existente em OutSystems. Nesta entidade a chave primária é o Id que é o identificador do Utilizador.

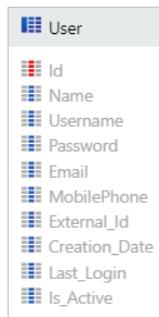


Figura 5.2: Tabela de Utilizadores

5.1.2 Documento

Como existem vários tipos de documentos pessoais, foi criada uma tabela Documento, representada na figura 5.3, que terá os dados comuns a todos estes. Depois cada tipo de documento terá a sua tabela com os seus dados mais específicos. Existem alguns atributos "ocultar" nos vários documentos que têm como objetivo guardar se o utilizador escolheu ocultar esse dado ou não. Os dados que poderão ser ocultados são os que se consideram ser confidenciais, como o número de cartão bancário ou o número do cartão de cidadão.

A entidade Documento tem como chave primária o Id que é o identificador do documento e como chave estrangeira o *UserId* que é o id do utilizador que adicionou o documento.

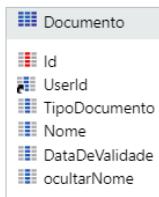


Figura 5.3: Tabela do Documento

Cartão de Cidadão

Na tabela Cartão de Cidadão, apresentada na figura 5.4, foram inseridos como atributos os dados que se encontram no mesmo, sendo ainda escolhidos alguns que não teriam de ser obrigatoriamente adicionados pelo utilizador ao inserir um novo documento, que foi o caso do Sexo, Nacionalidade, Filiação e Altura, pois são dados menos importantes e mais fáceis de memorizar. Nesta entidade a chave primária é também estrangeira, sendo o id da entidade Documento.

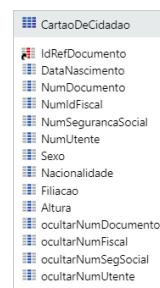


Figura 5.4: Tabela do Cartão de Cidadão

Carta de Condução

Os atributos da tabela Carta de Condução, apresentados na figura 5.5, correspondem aos dados presentes no documento físico que serão todos obrigatórios. Tal como no cartão de cidadão a chave primária é também estrangeira sendo o id correspondente à entidade Docuemtno.

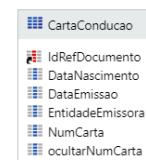


Figura 5.5: Tabela da Carta de Condução

Cartão Bancário

Na tabela Cartão Bancário todos os atributos representados na figura 5.6 terão de ser inseridos pelo utilizador ou extraídos pois são de uso obrigatório. Para além disso foram escolhidos estes atributos pois são os dados que se encontram disponíveis no Cartão Bancário. A chave primária e estrangeira é o id correspondente a este na tabela do documento.

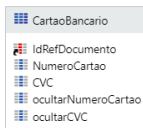


Figura 5.6: Tabela do Cartão Bancário

Passaporte

Na tabela Passaporte foram escolhidos os atributos com base nos dados que se encontram neste documento. Tal como aconteceu na tabela do Cartão de Cidadão foram escolhidos alguns atributos como sendo opcionais devido a não serem difíceis de memorização ou de escrever, como é o caso do Sexo, Altura e Nacionalidade. Todos os atributos são apresentados na figura 5.7. A chave primária e estrangeira é o id correspondente a este na tabela do documento.

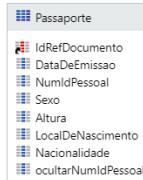


Figura 5.7: Tabela do Passaporte

Foto Documento

A tabela FotoDocumento, representada na figura 5.8, terá como objetivo guardar todas as fotos referentes a documentos que serão identificadas pelo id do Documento. Esta entidade tem como chave primária e estrangeira o id correspondente a este na tabela do documento.



Figura 5.8: Tabela da FotoDocumento

5.1.3 Fatura

Para guardar as faturas foi criada a tabela Fatura, representada na figura 5.9, cujos atributos são dados comuns em faturas de eletricidade, água, gás, etc. Alguns dados não serão obrigatórios de inserir, como no caso do período de faturação, e também existe a possibilidade de ocultar alguns dos dados como no caso dos documentos pessoais. A fatura tem como chave primária o identificador da fatura e como chaves estrangeiras o identificador do utilizador que adicionou a fatura e o identificador da categoria a que pertence.

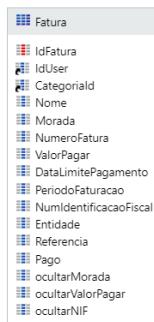


Figura 5.9: Tabela da Fatura

Foto Fatura

Esta tabela, representada na figura 5.10, será a responsável por guardar as fotos das faturas que serão identificadas pelo id da fatura correspondente. Nesta entidade tem-se como chave primária e estrangeira o identificador da fatura correspondente.



Figura 5.10: Tabela da FotoFatura

5.1.4 Categoria

Cada fatura terá de ser associada a uma categoria para se saber de que tipo de fatura se trata, por exemplo, se é uma fatura da Luz ou de Água, ajudando o utilizador na organização das mesmas.

Para isso foi criada esta tabela, representada na figura 5.11, em que apenas terá referência do id do Utilizador, que será a chave estrangeira, pois cada utilizador terá as suas categorias e o Nome que é o nome da Categoria. Como chave primária será o próprio identificador da categoria.

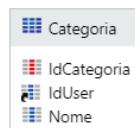


Figura 5.11: Tabela da Categoria

5.1.5 Problema

Esta tabela, visível na figura 5.12, será a responsável por guardar comentários que o utilizador tenha feito com base em problemas que tenha tido na sua utilização para poderem ser resolvidos posteriormente. O problema terá como chave primária o identificador do mesmo e como chave estrangeira o identificador do utilizador que adicionou um problema.

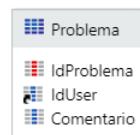


Figura 5.12: Tabela do Problema

5.1.6 Configuração de Notificação

Na Configuração de Notificação, representada na figura 5.13, serão guardadas apenas as escolhas do utilizador de que notificações quer receber e com que antecedência. Poderá escolher se pretende receber notificações de pagamentos a efetuar e/ou data de vencimento de documentos e escolher com que antecedência, em dias, pretende receber-las. Nesta tabela tem-se como chave primária e chave estrangeira o id do utilizador que tem essa configuração de notificações.

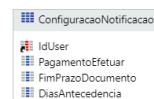


Figura 5.13: Tabela da ConfiguracaoNotificacao

5.1.7 Notificação

A tabela Notificação, representada na figura 5.14, terá como chave primária o seu id e como chaves estrangeiras o *UserId*, *idFatura* e *idDocumento* que vai referenciar o utilizador que receberá a notificação, e a que fatura ou documento se trata. Terá ainda os atributos tipo, dias e visualizado de modo a saber que tipo de documento ou fatura se trata a notificação, o número de dias que faltam e se já foi visualizada a notificação.

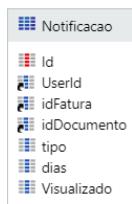


Figura 5.14: Tabela da Notificacao

5.2 Arquitetura

A aplicação está dividida por 3 módulos, como é possível visualizar na figura 5.15, sendo que no módulo do *End-User* é desenvolvida a parte gráfica da aplicação, o módulo *Core* onde irá ser construída a base de dados, a criação de processos, a lógica e a *Foundation* que é a parte lógica da aplicação que no caso será a utilização do *Form Recognizer* da Microsoft Azure para a extração dos dados.

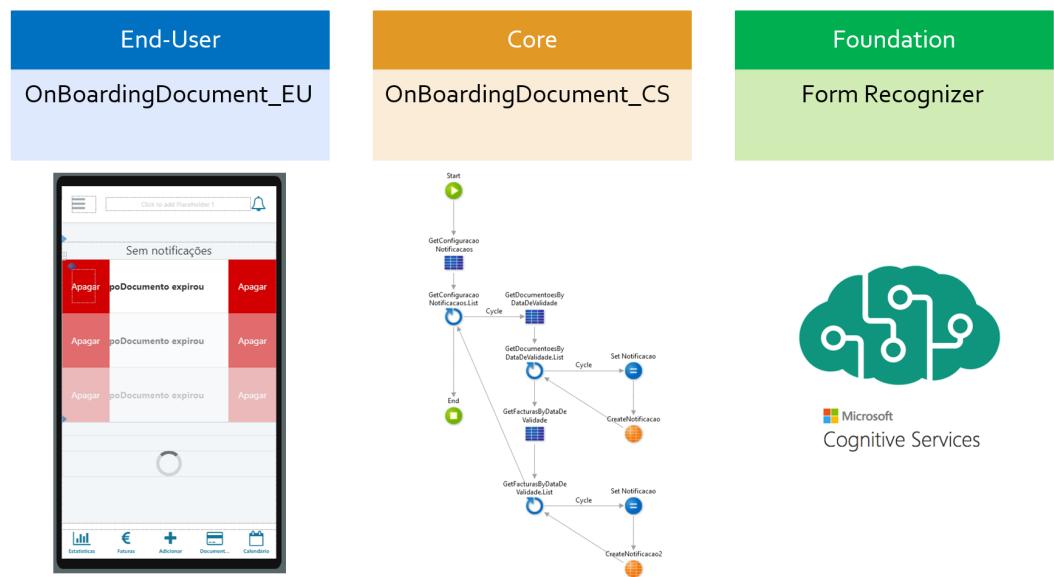


Figura 5.15: Arquitetura da Aplicação OnBoarding Document Capture

5.3 Implementação OutSystems

Nesta parte, para exemplificar a implementação da aplicação em OutSystems, irão ser explicados os principais componentes de um ecrã e de seguida uma tarefa de adicionar uma fatura, que engloba o uso da maioria dos componentes da plataforma tais como blocos, *widgets*, *client actions*, *server actions*, ler e escrever da base de dados e por último a extração de dados a partir de uma imagem.

Dentro de um ecrã, poderão existir os seguintes componentes:

- Variáveis: Poderão ser de inúmeros tipos, como simples inteiros, *strings* e booleanos, como também serem do tipo “Entidade de uma base de dados” (por exemplo uma variável do tipo Fatura, irá conter um identificador, um nome, uma morada, etc.), e muitos outros mais. Será apresentada na figura 5.16 um exemplo de uma variável do tipo Fatura.

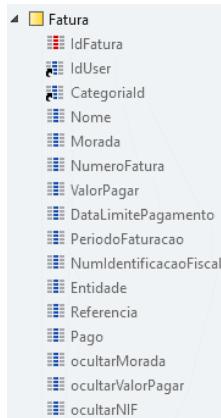


Figura 5.16: Variável do tipo Fatura

- Agregados: São utilizados para aceder a entradas de uma ou mais tabelas de uma base de dados, podendo filtrar os dados como se pretender, e aceder a estes no ecrã. Será ilustrado um exemplo de um agregado na figura 5.17

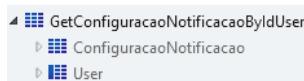


Figura 5.17: Agregado com duas tabelas

- *Client Actions*: São utilizadas para criar lógica na aplicação, podendo desempenhar uma ou mais funções no lado do cliente como por exemplo a validação de dados inseridos pelo utilizador ou invocar *server actions* (que permitem adicionar entradas em tabelas de bases de dados ou executar funcionalidades mais exigentes computacionalmente, no lado do servidor). Será mostrado um exemplo de uma *client action*, que valida os dados inseridos pelo utilizador e adiciona uma fatura à base de dados caso os dados estejam corretos na figura 5.18.

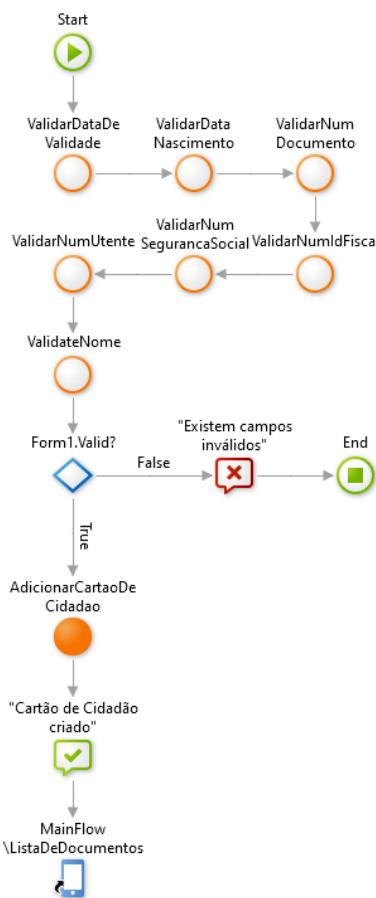


Figura 5.18: Exemplo de uma *client action*

Um dos objetivos da aplicação será a possibilidade de inserir os dados de uma fatura, sendo necessário a implementação de um ecrã na parte da interface no *MainFlow* (local onde estão todos os ecrãs da aplicação) denominado de AdicionarFaturas, como se pode ver na figura 5.19.



Figura 5.19: Ecrã de Adicionar Fatura

Neste ecrã as 3 componentes (variáveis, agregados e *client actions*) utilizadas poderão ser visualizadas na figura 5.20.

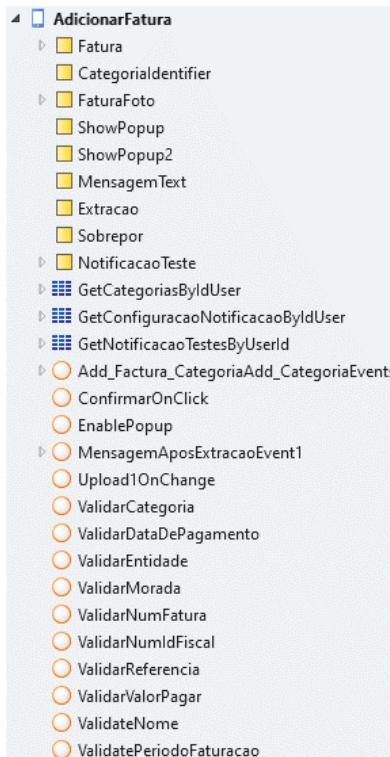


Figura 5.20: Componentes dentro de um ecrã

Tendo já todas as variáveis, agregados e *client actions* é possível implementar a interface do ecrã, utilizando blocos e *widgets*:

- Blocos: São partes de um ecrã que permitem implementar a sua própria lógica e serem utilizados em diferentes ecrãs. Neste caso foi criado um bloco que será reutilizado nos ecrãs de adicionar faturas e nos restantes ecrãs de adicionar documentos, que terá um cabeçalho que contém uma seta do lado esquerdo que permite ao utilizador voltar para um ecrã anterior, um título que será o nome do conteúdo que se está a adicionar (fatura, cartão de cidadão, carta de condução, passaporte ou cartão bancário) e por baixo do cabeçalho, espaço para colocar diferentes *widgets* que serão o conteúdo do ecrã, como é possível ver na figura 5.21.

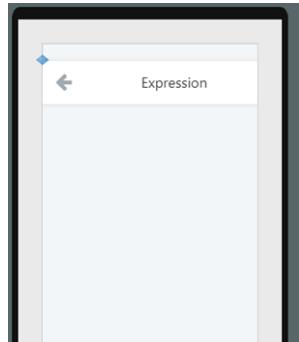


Figura 5.21: Bloco de um ecrã

Tal como um ecrã, este bloco contém também diferentes componentes que permitem a sua funcionalidade tais como eventos que são lançados no decorrer da aplicação que poderão chamar *client actions*, e *placeholders* que permitem reservar um espaço para se colocar texto quando o bloco for utilizado. Estas componentes poderão ser visualizadas na figura 5.22.

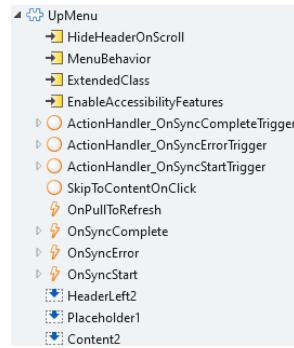


Figura 5.22: Componentes de um bloco

Este bloco irá ser adicionado à *Widget Tree* (árvore com diferentes hierarquias de *widgets* e elementos) do ecrã Adicionar Fatura, em que o “*HeaderLeft2*” contém a seta de voltar para trás, o “*Placeholder1*” contém o nome do conteúdo e o “*Content2*” irá conter o conteúdo específico do ecrã em questão. A estrutura do ecrã de adicionar fatura está representado na figura 5.23.

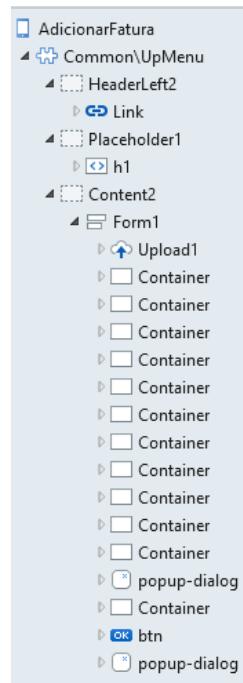


Figura 5.23: Widget Tree do ecrã ”Adicionar Fatura”

- *Widgets*: São funcionalidades já implementadas pela OutSystems que poderão ser adicionadas a um ecrã, como por exemplo um campo de *input*, um contentor para agrupar *widgets*, listas, formulários, *checkboxs*, *dropdowns*, entre muitos outros, sendo que em cada *widget* existe a possibilidade de poder alterar os seus atributos, como por exemplo um *input* de texto representado na figura 5.24.

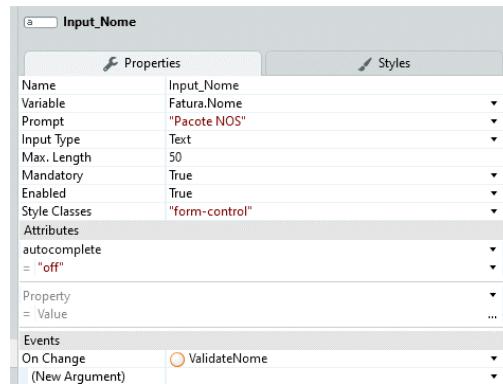


Figura 5.24: Atributos de um input widget

Estes *widgets* encontram-se no menu lateral esquerdo da interface, como se pode ver na figura 5.25.

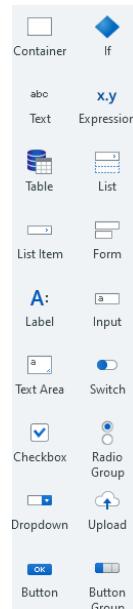


Figura 5.25: Exemplos de Widgets disponíveis

O ecrã de Adicionar Fatura é uma sequência de *widgets* de diferentes tipos, como imagens e campos de *input*, como se pode ver na figura 5.26.



Figura 5.26: Widgets do ecrã Adicionar Fatura

Quando um utilizador começa a inserir os dados nos campos de *input*, é sempre chamada uma *client action* que irá testar se estes estão corretos, chamando sempre uma *server action* que testa o *input* com uma *string regex* escrita pelo grupo, em que se estiver errado irá mostrar ao utilizador essa informação como se pode verificar de seguida na figura 5.27.

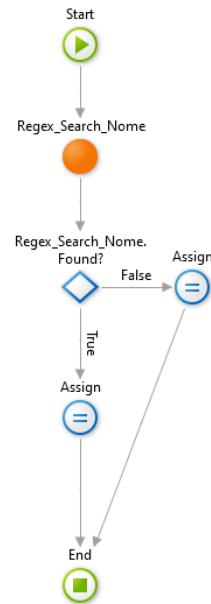


Figura 5.27: Validação de um campo

Todos os campos de *input* de texto foram implementados seguindo o mesmo raciocínio.

Ao inserir uma fatura também é possível escolher a que categoria esta pertence, selecionando uma opção de uma lista *dropdown*, representada na figura 5.28, que é preenchida automaticamente com um agregado.



Figura 5.28: *Dropdown* da lista de categorias

Caso não exista a categoria que o utilizador deseja, pode clicar na opção de “Criar nova categoria” em que irá aparecer um *popup* com a opção de poder adicionar uma nova. O *popup* está representado na figura 5.29.

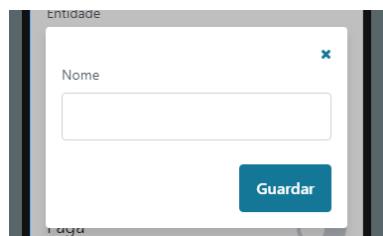


Figura 5.29: Popup de adicionar uma categoria

Quando o utilizador tiver finalizado a inserção dos dados da fatura nos campos respetivos, poderá selecionar o botão de “Criar” no final do ecrã, que irá chamar a seguinte *client action* demonstrada na figura 5.30.

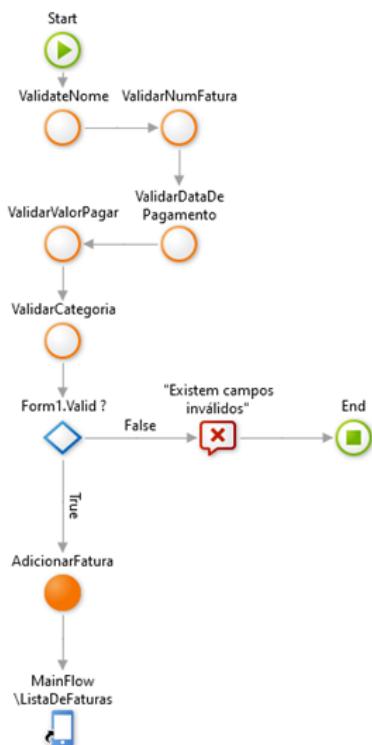


Figura 5.30: Adicionar uma fatura (client action)

Esta *client action* irá validar novamente os dados inseridos, e no caso de haver algum inválido irá alertar o utilizador com essa informação. Caso contrário irá chamar a *server action* (AdicionarFatura), que irá criar uma entrada na tabela da base de dados das faturas e na das fotografias das faturas, como está ilustrado na figura 5.31.



Figura 5.31: Adicionar uma fatura (server action)

Os restantes ecrãs foram implementados seguindo o mesmo processo e raciocínio.

5.3.1 Configuração e Criação de modelos

Para utilizar o *Form Recognizer* foi necessário primeiro treinar os modelos com algumas imagens de treino para depois quando for usado este modelo serem extraídos os dados mais facilmente.

Na ferramenta de rotulagem da amostra para criar um novo projeto é necessário fazer a configuração da conexão, carregando no botão correspondente como mostra na Figura 5.32.

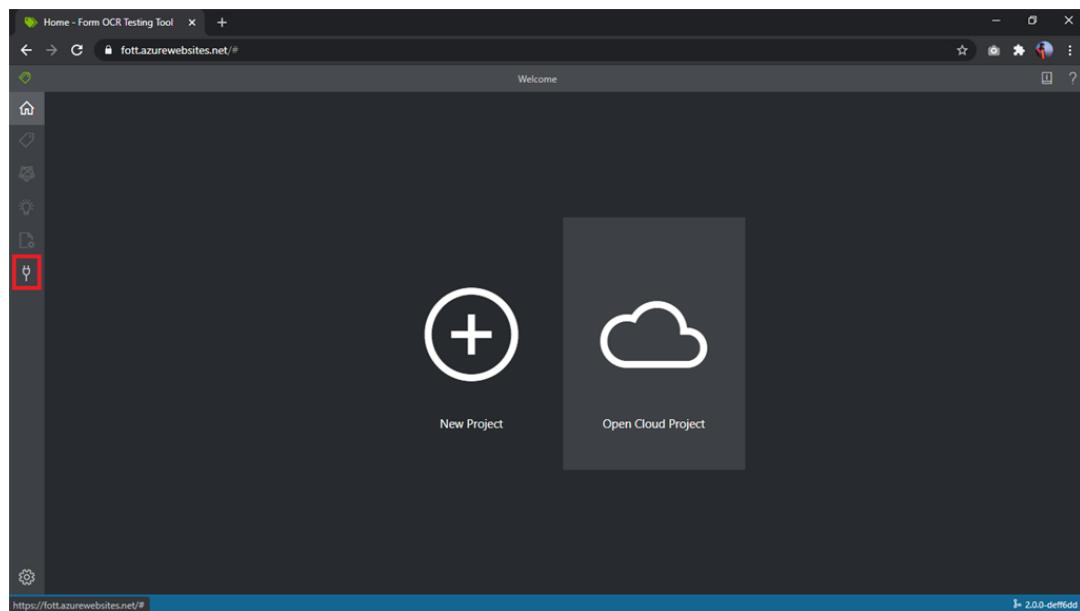


Figura 5.32: Selecionar Connections

Para a configuração é necessário inserir o nome e o *SAS URI*. O *SAS URI* é o *URL* de *SAS* do *BLOB* do contentor. Para aceder a este é necessário acecer ao Microsoft Azure e selecionar *documentcapturestorage1*, de seguida na secção armazenamento de dados seleccionar os contentores, e nestes existe o acesso a todas as pastas referentes aos treinos que se pretende realizar, como se pode ver na figura 5.33, podendo agora criar uma com o nome desejado.

Nome	Última modificação	Nível de acesso púb
\$logs	15/06/2021, 15:13:25	Privado
carta-conducao-training	22/06/2021, 17:56:28	Privado
carta-bancario-training	22/06/2021, 17:16:59	Privado
carta-cidadao-training	22/06/2021, 15:48:13	Privado
faturas-training	22/06/2021, 11:26:57	Privado
train-custom-postman	16/06/2021, 15:15:20	Privado

Figura 5.33: Lista de Contentores

Depois de criada, é possível selecionar na secção Definições a Assinatura de acesso partilhado em que será aqui gerado o *token* de *SAS*, como demonstrado na figura 5.34 e o *URL*, tendo assim o valor a meter no *SAS URI*.

Figura 5.34: Criação do Contentor

Assim, tem-se os dados suficientes para criar a conexão como ilustrado na figura 5.35.

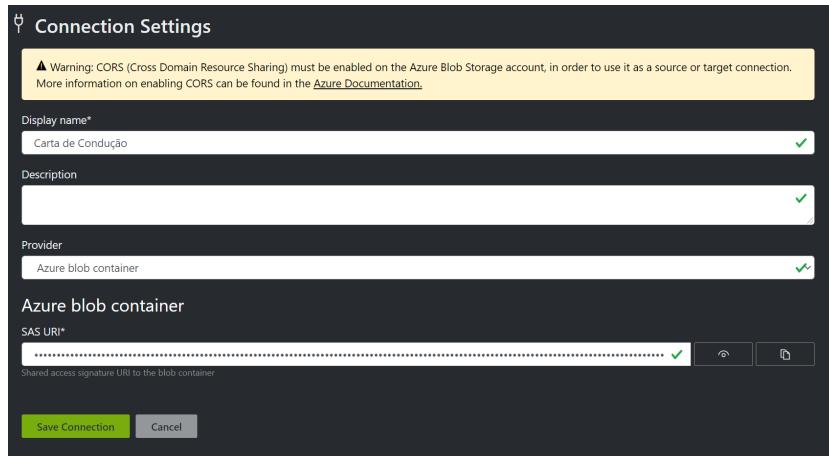


Figura 5.35: Dados para a criação da Conexão

Criada a conexão, será criado o projeto, como mostra a figura 5.36, em que para isso é necessário inserir um nome, selecionar o *source connection* criado, o *formrecognizer service URI* e a *API key* são adquiridos no módulo *DocumentCaptureForm* nas Chaves e Ponto Final em que o *URI* é o Ponto final e a *API key* a Chave 1.

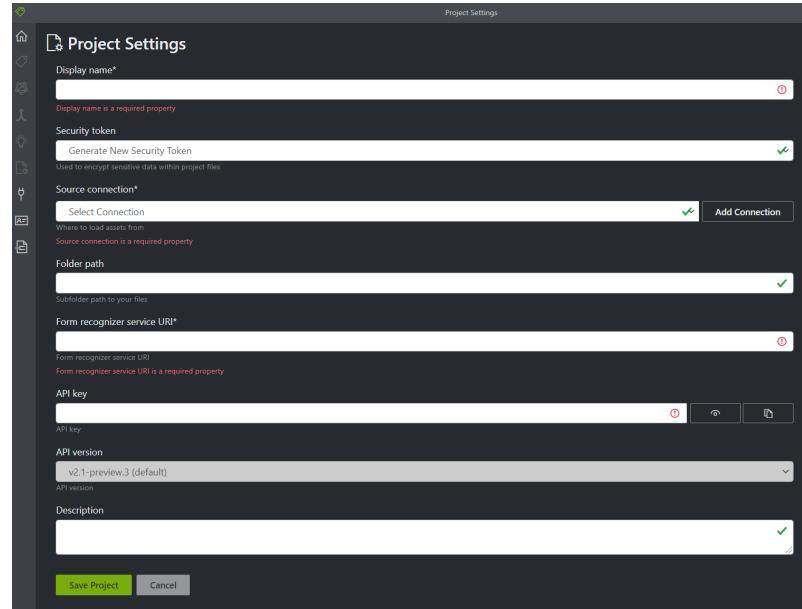


Figura 5.36: Criar Projeto

Na página do contetor de treino são adicionadas as imagens de treino que serão logo recebidas na ferramenta de rotulagem.

Ao abrir o projeto na ferramenta de rotulagem e selecionar o *icon* de *tag* aparecerá do lado esquerdo as imagens inseridas na página do contentor e no lado direito um campo para inserir as *tags*. Ao escolher uma imagem, esta é analisada e o seu texto aparecerá sublinhado, onde é possível de seguida selecionar o texto que se pretende e associá-lo a uma *tag*. Estas *tags* são personalizáveis e editáveis, podendo ser no caso de uma fatura o título, a morada, a entidade, a referência e o nome da companhia. Com isto, o modelo aprende que texto corresponde a que *tag*, e este processo irá ser repetido para todas as imagens de treino, em que quantas mais existirem, maior será a percentagem de acerto.

Para testar o modelo de treino, é possível clicar no botão Train onde irá aparecer a percentagem de acerto para cada *tag*, em que neste caso como exemplo foi treinado o modelo do cartão de cidadão como mostra a figura 5.37.



Figura 5.37: Resultados do treino de um modelo

Os modelos dos documentos obtiveram melhores resultados do que os modelos das faturas devido ao maior número de documentos que existem como exemplo e estes serem sempre iguais, em que no caso das faturas foram utilizadas imagens de diferentes empresas como da MEO, NOS e EDP, sendo estas todas diferentes o que dificultará altas percentagens de acertos nas *tags*.

5.3.2 Extração de dados

Ao introduzir uma imagem de um documento ou de uma fatura, a aplicação irá processá-la extraiendo os seus dados relevantes e colocando-os nos campos correspondentes. Será de seguida explicado o processo passo a passo desta funcionalidade, exemplificando-o introduzindo uma imagem de uma fatura.

Ao clicar no widget da figura 5.38 o utilizador poderá tirar uma fotografia a uma fatura ou poderá escolhê-la da sua galeria.

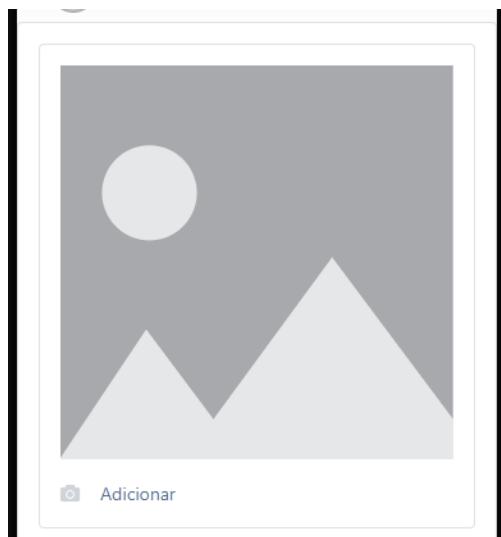


Figura 5.38: Fazer *upload* de uma imagem

Quando o *upload* for feito, irá ser executada a seguinte lógica em OutSystems:

- Preparar uma mensagem para mostrar ao utilizador caso este já tenha escrito dados nos campos, perguntando se deseja que os dados sejam sobrepostos com os da extração.
- Caso tenha dito que sim é chamada uma *server action* que irá receber esta imagem e um identificador do modelo de treino das faturas (*modelid*), e irá fazer uma chamada aos serviços cognitivos da Azure por um *URL* utilizando este *modelid*, retornando um resultado (chamado de *resultid*) que será um identificador desta extração. O *URL* construído para se poder obter o *resultid* utilizando uma *REST API* é o seguinte:

```
https://documentcaptureform.cognitiveservices.azure.com/formrecognizer/v2.1/custom/models/{modelId}/analyze?includeTextDetails=true
```

- O *resultid* é de seguida passado como parâmetro noutra *server action* que irá fazer uma chamada aos serviços cognitivos da Azure através de um *URL* com o *modelid* e o *resultid*, em que o resultado será um ficheiro *JSON* que é interpretado pela *API* e que irá preencher um conjunto de campos para depois serem utilizados, por exemplo no caso da fatura será o número da fatura, a entidade, a referência, etc. O *URL* para se obter os dados extraídos é o seguinte: <https://documentcaptureform.cognitiveservices.azure.com/formrecognizer/v2.1/custom/models/{modelId}/analyzeResults/{resultId}>

Sendo a ligação do Módulo do *Core* (onde se encontra toda a lógica da aplicação) com os serviços cognitivos da Azure, demonstrado no diagrama presente na figura 5.39.

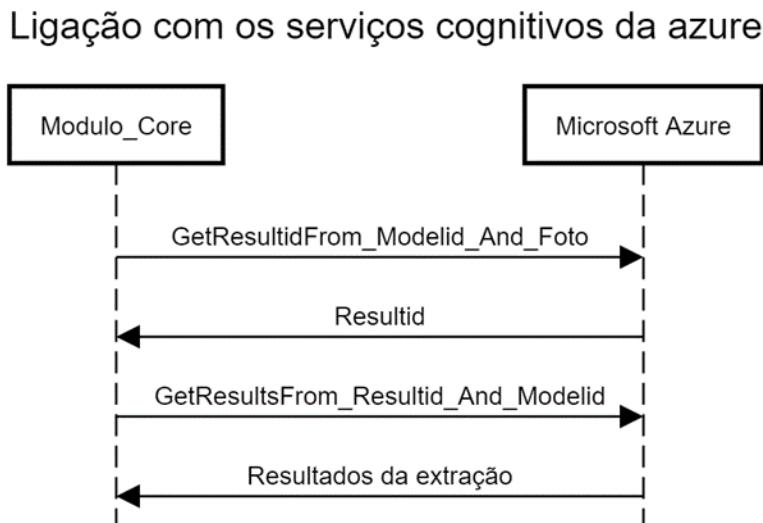


Figura 5.39: Ligação da aplicação com os serviços cognitivos da Azure

- Tendo os dados extraídos é possível adicionar a cada campo que não tenha sido ainda preenchido pelo utilizador, o respetivo valor extraído, em que de seguida irá aparecer ao utilizador os novos dados, podendo editá-los caso estejam errados.

Para exemplificar, foi feito o *upload* da fatura presente na figura 5.40.

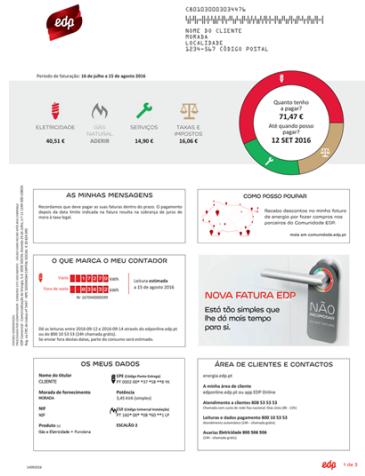


Figura 5.40: Fatura exemplo

Desta fatura foram extraídos os dados presentes na seguinte figura 5.41.

Morada	<input type="text" value="MORADA"/>
Número da fatura *	<input type="text" value="C801030003034476"/>
Valor a pagar (€) *	<input type="text" value="71,4"/>
Data limite de pagamento *	<input type="text" value="dd/mm/aaaa"/> <input type="button" value="..."/>
Período faturação	<input type="text" value="16 de julho a 15 de agosto 2016"/>

Figura 5.41: Dados extraídos de uma fatura

O *Form Recognizer* da Microsoft Azure conseguiu extrair corretamente a informação que se pretendia da imagem.

Capítulo 6

Resultados Obtidos

Ao ser desenvolvido o projeto foram realizados alguns testes à funcionalidade e usabilidade da aplicação.

6.1 Testes Funcionais

Os testes funcionais são definidos de acordo com os requisitos funcionais do *software*. Este teste é particularmente útil para revelar problemas como:

- Funções incorretas ou omitidas;
- Erros de interface;
- Erros de comportamento ou de desempenho.

Este tipo de testes foram utilizados em praticamente todos os ecrãs, serão explicados alguns exemplos de onde foram utilizados estes testes.

Para inserir um documento ou fatura manualmente foram precisas desenvolver várias funções para garantir que o utilizador seguiria a estrutura de cada dado e por isso nesta situação foi um ponto crucial o uso de testes funcionais para garantir que todas essas validações estariam corretamente implementadas.

Na Figura 6.1 todos os campos obrigatórios estão a dar erro pois são dados que tem de ser obrigatoriamente guardados para estarem associados ao Cartão de Cidadão.

The screenshot shows a mobile application interface for a 'Cartão de Cidadão'. At the top, there is a back arrow icon and the title 'Cartão de Cidadão'. Below the title are several input fields:

- Nome ***: The input field contains 'Nome completo' and has a red border, indicating it is invalid. Below the field, the error message 'Nome inválido.' is displayed.
- Sexo**: The input field contains 'M/F'.
- Altura (cm)**: The input field contains '160'.
- Nacionalidade**: The input field contains 'PRT'.
- Data de nascimento ***: The input field contains 'dd/mm/aaaa' and includes a date picker icon. Below the field, the error message 'Data de nascimento inválida.' is displayed.
- Nº ID Civil ***: The input field contains '15748345' and has a red border, indicating it is invalid. Below the field, the error message 'Número de documento inválido.' is displayed.
- Data de validade ***: The input field contains 'dd/mm/aaaa' and includes a date picker icon.

Figura 6.1: Validação dos campos obrigatórios no Cartão de Cidadão

Ao serem inseridos dados pelo utilizador serão ainda verificadas as validações de cada campo sobre que estrutura devem seguir. Por exemplo, o nome deverá apenas aceitar letras e espaços em que deverá ter um tamanho máximo de 50 caracteres, já o NºID Civil deverá ser apenas números e com um tamanho fixo de 8 dígitos.

Como se pode ver na figura 6.2 na primeira imagem os dados foram inseridos de forma inválida, alertando o utilizador para isso. Na segunda imagem os dados foram inseridos corretamente não tendo assim nenhum aviso de invalidez.

The figure consists of two side-by-side screenshots of a mobile application interface titled "Cartão de Cidadão".

Left Screenshot (Invalid Input):

- Nome ***: The input field contains "A3d Na", which is highlighted with a red border and followed by the error message "Nome inválido."
- Sexo**: The input field contains "M/F".
- Altura (cm)**: The input field contains "160".
- Nacionalidade**: The input field contains "PRT".
- Data de nascimento ***: The input field contains "dd/mm/aaaa" with a calendar icon.
- Nº ID Civil ***: The input field contains "ascv", which is highlighted with a red border and followed by the error message "Número de documento inválido."
- Data de validade ***: The input field contains "dd/mm/aaaa" with a calendar icon.

Right Screenshot (Valid Input):

- Nome ***: The input field contains "António Hugo".
- Sexo**: The input field contains "M/F".
- Altura (cm)**: The input field contains "160".
- Nacionalidade**: The input field contains "PRT".
- Data de nascimento ***: The input field contains "dd/mm/aaaa" with a calendar icon.
- Nº ID Civil ***: The input field contains "12345678".
- Data de validade ***: The input field contains "dd/mm/aaaa" with a calendar icon.

Figura 6.2: Testes aos campos Nome e N°ID Civil no Cartão de Cidadão

Na tarefa de ”Alterar palavra-passe” foi necessário verificar se a palavra-passe atual inserida pelo utilizador seria a mesma que atual e que as novas palavras-passe fossem exatamente iguais podendo conter letras, dígitos, alguns caracteres e ter o tamanho mínimo de 3.

Como se pode verificar na figura 6.3 na primeira imagem ao clicar no ”Confirmar” foi verificado que a palavra-passe não é igual à atual, na segunda a nova palavra-passe não tem o mínimo de caracteres como deve, na terceira ao ”Confirmar” foram comparadas as palavras-passe e não são compatíveis. Por fim na figura 6.4 ao adicionar os dados como esperado será alterada a palavra-passe.

The figure consists of three side-by-side screenshots of a web application interface for changing a password. Each screenshot shows a header with three horizontal bars and the text "Alterar palavra-passe" followed by a bell icon.

- Screenshot 1:** Shows the "Palavra-passe atual:" field containing "*****". Below it, a red error message says "Palavra-passe atual incompatível".
- Screenshot 2:** Shows the "Nova palavra-passe:" field containing a single dot ("."). Below it, a red error message says "Palavra-passe tem de ter no mínimo 3 caracteres".
- Screenshot 3:** Shows both the "Nova palavra-passe:" and "Confirmar nova palavra-passe:" fields containing "****". Below them, a red error message says "Palavras-passe incompatíveis".

Each screenshot features a teal "Confirmar" button at the bottom.

Figura 6.3: Testes da tarefa 'Alterar palavra-passe' - Erros

This screenshot shows the same password change interface as Figure 6.3, but with successful validation results. A green banner at the top displays a checkmark icon and the text "Palavra-passe alterada!" (Password changed!).

The input fields are identical to Figure 6.3, but now all are empty. The "Confirmar" button is present at the bottom.

Figura 6.4: Testes da tarefa 'Alterar palavra-passe' - Validação

Nas listas de documentos, faturas e na gestão de categorias o utilizador poderá fazer uma pesquisa para aceder com maior facilidade ao que deseja ver. Para isso foi preciso atualizar a lista que está a ser apresentada com base no texto escrito na barra de pesquisa, como é possível ver na figura 6.5.

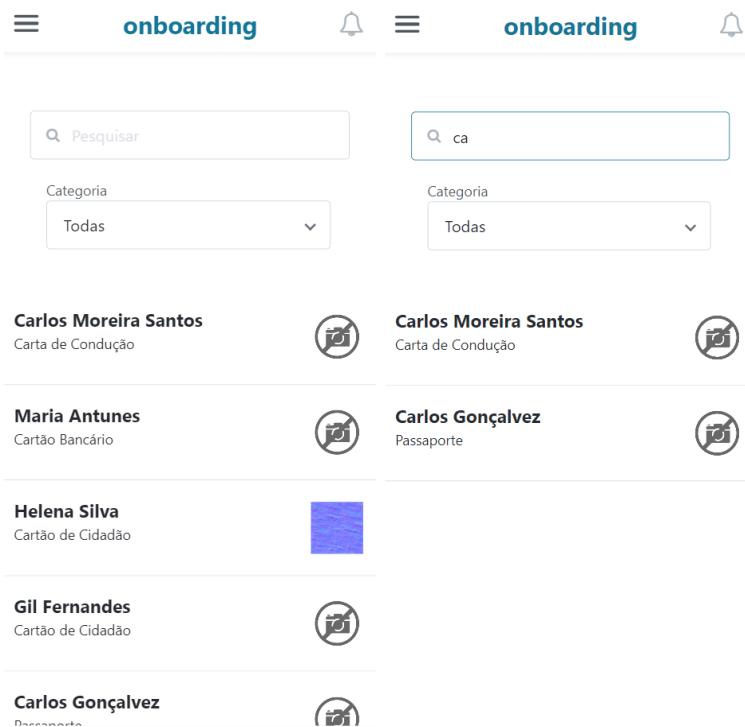


Figura 6.5: Teste à barra de pesquisa da lista de documentos

6.2 Testes de Usabilidade

Ao criar os primeiros ecrãs da aplicação foram feitos questionários de usabilidade para saber a opinião dos utilizadores relativamente ao aspeto gráfico da aplicação e as suas funcionalidades. Na figura 6.6 é possível ver um exemplo de uma pergunta feita ao utilizador e na figura 6.7 as respostas dadas pelos utilizadores.

Este será o ecrã inicial, onde dá a opção ao utilizador de poder entrar na sua conta ou caso não tenha seguir para o ecrã do registo. A imagem do círculo é apenas representativa e irá ser substituída pelo logo da aplicação.



Figura 6.6: Exemplo de uma pergunta do questionário aos utilizadores

Permitir entrar com o google/facebook/etc.
Borda no botão branco para se notar melhor onde acaba.
nada para dizer.
Para manter mais simples a introdução podia ser apresentada com um botão "!" no canto inferior.
Deveria aparecer logo o login em vez da tecla entrar. Quantos menos cliques, melhor.
Está ótimo
Adequado
Simples e eficaz
É legível e simples o que não faz subcarregar a visão do utilizador para o seu objetivo que é registrar-se, portanto não mudaria nada
Podia ter o nome e mudar as cores ou isso para não ser tão básico, mas tá giro
Fundo escuro
Mudar a cor Mudar / alterar logotipo
colocaria o entrar e o registar um pouco mais abaixo
Ok

Figura 6.7: Respostas à pergunta da Figura 6.6

Depois da aplicação toda implementada foram feitos testes de usabilidade. Os testes de usabilidade são realizados pelos utilizadores representativos do público-alvo, cada participante terá de realizar algumas tarefas e responder a um questionário sobre a facilidade que sentiu a realizá-las e sobre o que sente em relação à aplicação.

Primeiramente foram pedidos a 13 utilizadores a realizar 5 tarefas: Configurar Notificações, Aceder ao Calendário, ver página detalhada de uma fatura, apagar um documento e adicionar uma categoria. Através dos resultados das figuras 6.8, 6.9, 6.10, 6.11, 6.12 é possível verificar que num modo geral que os utilizadores conseguiram facilmente realizar as tarefas.

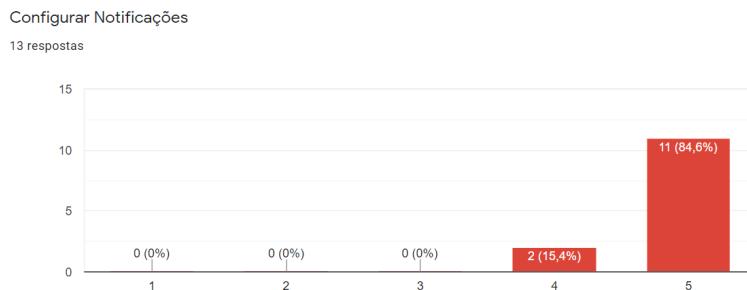


Figura 6.8: Resultados da tarefa ”Configurar Notificações” por parte dos utilizadores

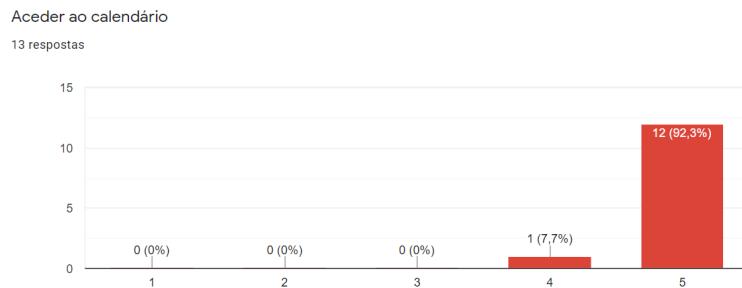


Figura 6.9: Resultados da tarefa ”Aceder ao calendário” por parte dos utilizadores

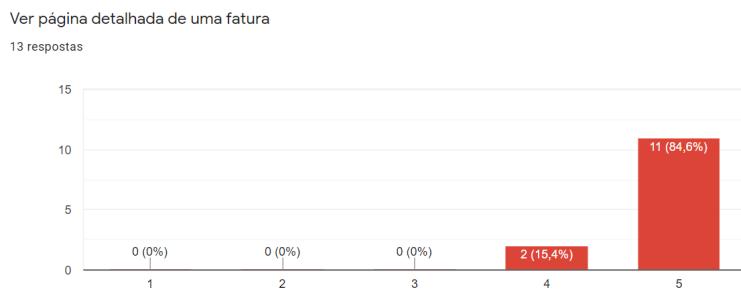


Figura 6.10: Resultados da tarefa ”Ver página detalhada de uma fatura” por parte dos utilizadores

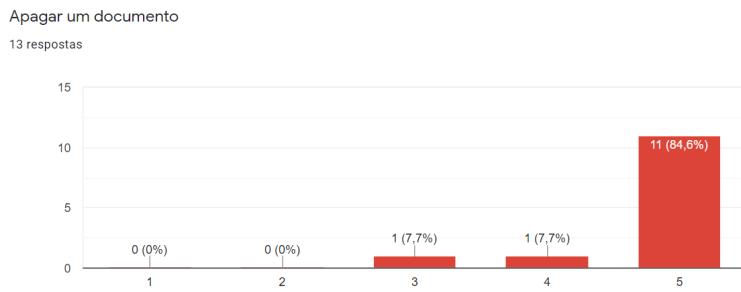


Figura 6.11: Resultados da tarefa ”Apagar um documento” por parte dos utilizadores

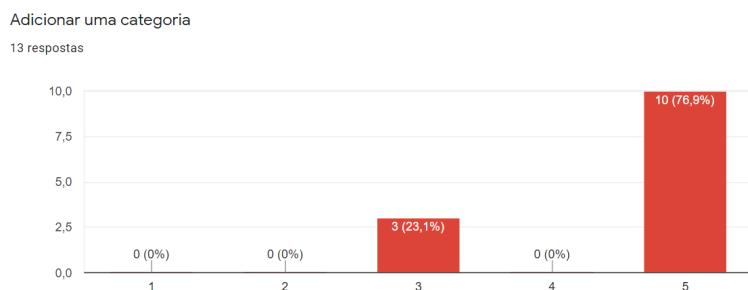


Figura 6.12: Resultados da tarefa ”Adicionar uma categoria” por parte dos utilizadores

De seguida foram feitas um conjunto de perguntas para a avaliação geral da aplicação e a partir das respostas dadas são retirados os valores de usabilidade da aplicação com base nas respostas dos utilizadores. Todas estas perguntas foram retiradas de uns exemplos que foram dados na unidade curricular Intereração Pessoa-Máquina.

Para calcular a pontuação de usabilidade é necessário seguir os seguintes passos:

1. Para cada pergunta ímpar subtrair 1 ao resultado obtido na resposta;
2. Para cada pergunta par subtrair ao 5 o resultado obtido na resposta;
3. Somar os valores obtidos e multiplicar por 2,5 (para converter os valores finais para um valor entre 0 e 100).

Assim sendo seguem-se as pontuações por cada utilizador e a média dos utilizadores.

Utilizador 1 = 100%

Utilizador 2 = 85%

Utilizador 3 = 97,5%

Utilizador 4 = 80%

Utilizador 5 = 100%

Utilizador 6 = 97,5%

Utilizador 7 = 85%

Utilizador 8 = 97,5%

Utilizador 9 = 100%

Utilizador 10 = 97,5%

Utilizador 11 = 85%

Utilizador 12 = 90%

Utilizador 13 = 80%

Média dos Utilizadores = 91,92 %

Tendo em conta a seguinte escala:

0% - 19% - Muito Insuficiente

20% - 49% - Insuficiente

50% - 69% - Suficiente

70% - 89% - Bom

90% - 100% - Muito Bom

É possível afirmar que a aplicação tem uma usabilidade muito boa com base na média dos utilizadores.

Por fim foi ainda pedido aos utilizadores para deixarem sugestões de melhorias da aplicação ou mesmo de erros que tenham sido encontrados, como se pode ver na figura 6.13. Foram feitas algumas mudanças na aplicação com intuito de melhorar a experiência do utilizador.

Título/logo da aplicação no topo das páginas principais para uma rápida identificação da app. Botão para marcar notificações como lidas.
No que toca às configurações de categorias e notificação, colocar "Gestão de ..."
Manter o menu do fundo do ecrã fixo
que haja o botão de retroceder em algumas páginas. Porque, quando vou para uma pagina, em vez de poder voltar à pagina anterior tenho de ir pelo caminho dos menus.
Nada a acrescentar.
A opção de criação de nova categoria (ou outro conceito) deve estar no header ou no footer das listas e nunca desaparecer (tipo no e-mail)
Para adicionar categoria, o campo poderia estar no topo do ecrã. Quando se apaga uma notificação, deveria actualizar o numero de notificações. Poderia ter a funcionalidade de apagar fatura quando se fizesse slide. No calendário, quando se clica no dia com ponto e aparece o documento, ao clicar na fatura de dia 1/07 não foi para o seu detalhe.

Figura 6.13: Sugestões dos Utilizadores

Capítulo 7

Conclusões

A versão final da aplicação desenvolvida em OutSystems neste projeto permite guardar faturas e documentos e ainda extrair dados de imagens como planeado, em que para isso foi imprescindível o estudo de diferentes tópicos como o Estado da Arte, que permitiu adquirir ideias para funcionalidades e aspeto gráfico da interface, os casos de utilização (*use cases*) que ajudaram na organização e na sequência de tarefas permitindo uma aplicação com uma boa usabilidade, e ainda o desenho de *wireframes* e *mockups* que deram uma ideia geral das funcionalidades de cada ecrã e permitiram um desenvolvimento mais rápido e organizado.

Desenvolver toda a aplicação foi um desafio pois durante a licenciatura não houve a oportunidade de utilizar nenhuma plataforma *low-code*. A utilização da plataforma OutSystems trouxe benefícios tais como uma maior flexibilidade, agilidade e otimização de tempo, tendo ainda permitido conhecer esta metodologia que é bastante diferente das que foram utilizadas ao longo do curso. Por outro lado, houve dificuldades ao utilizar o conector dos serviços cognitivos da Azure disponível na *forge* da OutSystems devido a um problema de versões, tendo optado por criar uma *REST API* que permitia a ligação entre a OutSystems e os serviços cognitivos.

A utilização da plataforma *Form Recognizer* da Azure facilitou o processo de integração de reconhecimento de texto em documentos, tendo sido uma mais-valia devido ao seu bom funcionamento e à ajuda que pode proporcionar ao utilizador, permitindo-lhe inserir os dados de forma mais rápida na aplicação. A principal desvantagem ao treinar os modelos foi a quantidade de documentos e faturas disponíveis para treino serem escassas, podendo diminuir a precisão da extração de dados corretos.

Ao longo de todo o desenvolvimento foram feitos questionários a utilizadores de modo a garantir um produto apelativo e que incluísse todas as funcionalidades que se pretendiam, tendo sido bastante importante também para

a deteção de erros, acrescendo de funcionalidades e melhorias. Muitos dos conhecimentos adquiridos na unidade curricular Interação Pessoa-Máquina (IPM) foram aplicados neste projeto nomeadamente nos *use cases*, *wireframes*, *mockups* e nos questionários. No desenvolvimento foi também essencial a aplicação de todos os conhecimentos adquiridos ao longo do curso nomeadamente a persistência, dedicação, resolução de problemas, trabalho em equipa e organização. Ao finalizar este projeto foram adquiridos novos conhecimentos relativamente a métodos de planeamento, a utilização da OutSystems para o desenvolvimento de uma aplicação para dispositivos móveis, a utilização de ferramentas de *ML* entre muitas outras.

Esta aplicação poderá ser melhorada em termos de interface gráfica adicionando outras cores ou animações, suportar mais tipos de documentos como cartões de identificação de outras nacionalidades para além da Portuguesa e a nível de extração de dados, adicionando aos modelos de treino mais documentos e faturas.

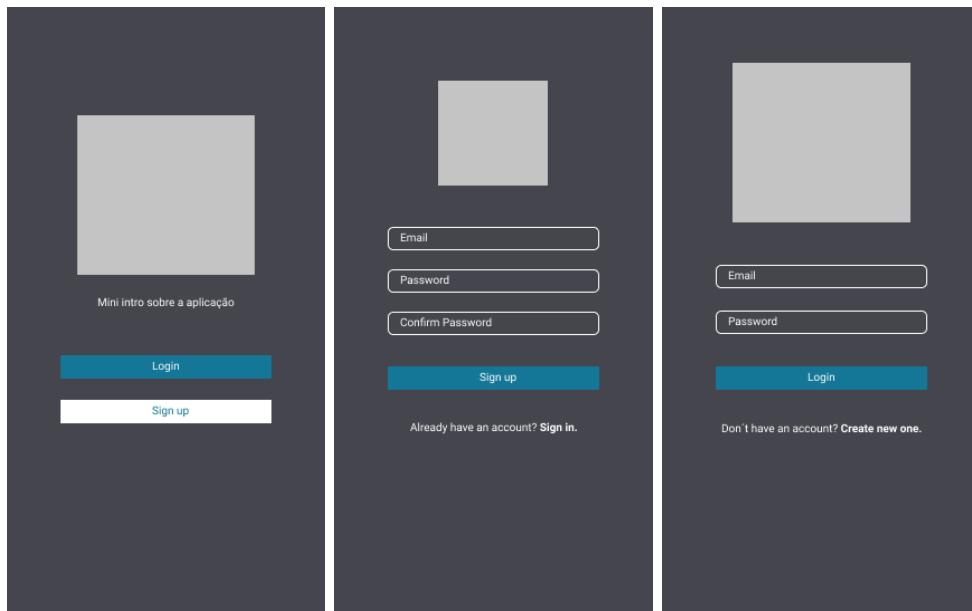
O resultado final foi bastante gratificante para o grupo tanto a nível de aspeto gráfico da aplicação e todas as suas funcionalidades como também a nível de conhecimentos adquiridos.

Bibliografia

- [1] Site da Payper, <https://www.payper.pt/pt/> e a página da PlayStore da Aplicação Payper, <https://play.google.com/store/apps/details?id=com.payper.app>
- [2] Site da Billdu, <https://www.billdu.com/> e a página da PlayStore da Aplicação Billdu, <https://play.google.com/store/apps/details?id=de.minirechnung>
- [3] Site da Expensify, <https://www.expensify.com/> e a página da PlayStore da Aplicação Expensify, <https://play.google.com/store/apps/details?id=org.me.mobiexpensifyg>
- [4] Site da BlinkID, <https://microblink.com/> e a página da PlayStore da Aplicação BlinkID, <https://play.google.com/store/apps/details?id=com.microblink.blinkidapp>
- [5] Site da IRISmart™, <https://www.irislink.com/PT/c2030/IRISmart-Security---ID-and-passport-recognition-software.aspx>
- [6] Site da Veryfi Receipts OCR & Expenses, <https://www.veryfi.com/> e a página da PlayStore da Aplicação Veryfi Receipts OCR & Expenses, <https://play.google.com/store/apps/details?id=com.iqboxyinc.iqboxy>
- [7] Site da CamScanner, <https://www.camscanner.com/> e a página da PlayStore da Aplicação CamScanner, <https://play.google.com/store/apps/details?id=com.intsig.camscanner>
- [8] Site de ajuda na criação de tabelas para latex, https://www.tablesgenerator.com/latex_tables#
- [9] Tutorial OutSystems - Becoming a Reactive Web Developer, <https://www.outsystems.com/training/paths/18/becoming-a-reactive-web-developer/>

- [10] Tutorial OutSystems - Becoming a Mobile Developer,
[https://www.outsystems.com/training/paths/1/
becoming-a-mobile-developer/](https://www.outsystems.com/training/paths/1/becoming-a-mobile-developer/)

Anexos



Screen 1: Adicionar conteúdo

Escolher ficheiro

- Nome: [Text input]
- Data de Nascimento: [Text input]
- Número do Documento: [Text input]
- Data de validade: [Text input]
- Número de identificação fiscal: [Text input]
- Número de segurança social: [Text input]
- Número de utente: [Text input]
- Adicionar mais dados

Screen 2: Escolher ficheiro

Nome: [Text input]
Morada: [Text input]
Número da fatura: [Text input]
Valor a pagar: [Text input]
Data limite de pagamento: [Text input]

Adicionar mais dados

Pago Não pago

Categoria

Screen 3: Payment Options

Cancel Done

Screen 1: Main menu (highlighted)

- Alterar password
- Notificações
- Categorias
- Relatar Problema
- Logout

Screen 2: Alterar password

Alterar password

Nova password: [Text input]
Confirmar nova password: [Text input]

Screen 3: Opções de notificação

Opções de notificação

Receber notificação de:

Pagamentos a efetuar
 Fim do prazo de documentos

Receber notificação com antecedência de:

1 dia
 7 dias
 14 dias
 30 dias

Screen 4: Payment Options

Cancel Done

Opções de categorias

Categorias de faturas

- Gás
- Apagar**
- Mudar nome
- Adicionar categoria...
- Done

Relatar um problema

Escreva aqui...

Faltam "x" dias para terminar o prazo de "nome"

Ver

Faltam "x" dias para terminar o prazo de "nome"

Ver

para terminar o prazo de

Ver **Apagar**

Cancel **Done** **Cancel** **Enviar**

Pesquisar

Categoria

All

Pago Não pago

Nome da fatura Categoria 1

Nome da fatura Categoria 2

fatura Pago

+ €

Pesquisar

Categoria

All

Nome do proprietário Passaporte

Nome do proprietário Carta de condução

Nome do proprietário Cartão Bancário

Nome do proprietário Cartão de cidadão

+ €

Month **Week** **Day**

JANEIRO 2016

DOM	SEG	TER	QUA	QUI	SEX	SAB
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Pressione longa para incluir uma nota
27 mar 2016: Páscoa (data comemorativa)

+ €

