

**Estágio na Babel Ibérica, SA:
Contribuição para a implementação do Projeto
Emprego Público Online**

Diana Isabel Casteleiro Gomes

Relatório de Estágio para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Mário Freire
Co-orientador: Thiago Nolau

Covilhã, junho de 2023

Declaração de Integridade

Eu, Diana Isabel Casteleiro Gomes, que abaixo assino, estudante com número de inscrição M12160 do curso de 2º ciclo de Engenharia Informática da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridade da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, e que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assim assumo na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 12/06/2023.

Diana Isabel Casteleiro Gomes

Diana Isabel Casteleiro Gomes

Agradecimentos

A concretização deste curso e de todo o percurso académico não seria possível sem o apoio das pessoas que me acompanham, por isso, pretendo agradecer a todos os que me apoiaram e contribuíram para a realização desta etapa final na minha formação académica, o Mestrado em Engenharia Informática na Universidade da Beira Interior. Em primeiro lugar, agradecer à minha família, em especial à minha mãe e avó, por todo o apoio, esforço e dedicação que me deram ao longo de toda a minha vida. Em segundo lugar, agradecer a todos os professores que se cruzaram na minha vida académica e contribuíram para a minha progressão de conhecimento, em especial ao meu orientador, pela disponibilidade, simpatia e orientação. Quero agradecer também à empresa Babel, pela oportunidade de integração na empresa, assim como ao meu mentor Thiago Nolau da Babel pela compreensão, atenção, simpatia e ajuda. Sem esquecer todos os restantes colegas da Babel pelo suporte e auxílio. Por fim, agradecer aos meus amigos que me acompanharam sempre tanto a nível pessoal como profissional, por todo o companheirismo, entreaajuda, união e amizade.

Resumo

Este relatório visa documentar todo o processo de estágio que faz parte da unidade curricular Estágio, referente ao segundo semestre do segundo ano do Mestrado em Engenharia Informática na Universidade da Beira Interior. Neste documento podem ser encontradas informações detalhadas acerca do estágio, como, atividades propostas e realizadas, assim como as tecnologias e ferramentas disponíveis.

Este estágio passou por várias fases, desde a formação planeada pela empresa de várias tecnologias necessárias à aprendizagem da plataforma de *Outsystems* até à integração numa equipa para a implementação do projeto Emprego Público *Online*.

Palavras-chave

Estágio, Informática e *Outsystems*

Abstract

This report aims to document the entire internship process that is part of the Internship course, referring to the second semester of the second year of the Master's Degree in Computer Engineering at the University of Beira Interior. In this document you can find detailed information about the internship, such as proposed and carried out activities, as well as the technologies and tools available.

This internship went through several phases, from the training planned by the company of various technologies necessary for the learning of the *Outsystems* platform to the integration into a team for the implementation of the Emprego Público Online.

Keywords

Internship, Computer science and Outsystems

Índice

1	Introdução	1
1.1	Objetivos	1
1.2	Principais Desafios	1
1.3	Caracterização da Empresa	2
1.4	Formação	3
1.5	Descrição do Projeto	3
1.6	Planificação e Cronograma	3
1.7	Organização do Documento	6
2	Tecnologias Utilizadas	7
2.1	<i>Low-Code</i>	7
2.2	HTML/CSS	8
2.3	<i>Javascript</i>	8
2.4	<i>JQuery</i>	8
2.5	SQL	9
2.6	<i>.NET</i>	9
2.7	UI/UX	9
2.8	Ferramentas	10
2.8.1	<i>Microsoft Teams</i>	10
2.8.2	<i>Microsoft Azure DevOps</i>	10
2.8.3	<i>Figma</i>	11
3	Estado da Arte da Plataforma <i>Outsystems</i>	13
3.1	Metodologia	14
3.1.1	SCRUM	14
3.1.2	Kanban	14
3.2	Organização	15
3.3	Estrutura da Plataforma	15
3.3.1	<i>Service Studio</i>	15
3.3.2	Integration Studio	16
3.3.3	<i>LifeTime</i>	16
3.3.4	Service Center	16
3.4	<i>Platform Server</i>	16
4	Contextualização do Estágio	19
4.1	Abordagem da Proposta	19
4.2	Fases de Desenvolvimento	21
4.3	Planificação do Trabalho	21

5	Projeto: Emprego Público <i>Online</i>	23
5.1	Requisitos	23
5.1.1	Perfis	23
5.1.2	Requisitos Não Funcionais	24
5.1.3	Requisitos Funcionais	24
5.2	Desenvolvimento	28
6	Conclusão	31
	Bibliografia	32

Lista de Figuras

1.1	Logótipo da empresa.	2
1.2	Cronograma-Planificação do estágio.	5
2.1	<i>DevOps</i> : Ciclo de vida de uma aplicação.	11
3.1	Logótipo <i>Outsystems</i>	13
4.1	Diagrama representativo das etapas de desenvolvimento do <i>software</i> no <i>DevOps</i>	20
5.1	Desenvolvimento do ecrã "Lista de Perfis" no ambiente <i>Outsystems</i>	28
5.2	Base de dados e seus atributos referentes à "Lista de Perfis" no ambiente <i>Outsystems</i>	29
5.3	Desenvolvimento do ecrã "Criar/Editar Perfis" no ambiente <i>Outsystems</i>	29
5.4	Lógica do botão "Guardar".	30

Lista de Acrónimos

CSS Cascading Style Sheets

UBI Universidade da Beira Interior

HTML HyperText Markup Language

PHP Hypertext Preprocessor

SQL Structured Query Language

UI Interface do utilizador

UX Experiência do utilizador

REST Representational State Transfer

SOAP Simple Object Access Protocol

TI Tecnologias de Informação

SAP Systemanalysis Programmentwicklung

eSPap Entidade de Serviços Partilhados da Administração Pública

BEP Bolsa de Emprego Público

OML *Outsystems Modeling Language*

XML *Extensible Markup Language*

CC Cartão de Cidadão

CMD Chave Móvel Digital

Capítulo 1

Introdução

Este documento especifica o trabalho que foi realizado na unidade curricular de estágio em Engenharia Informática inserido no segundo ano do 2º ciclo em Engenharia Informática na Universidade da Beira Interior (UBI). Este estágio enquadra-se no desenvolvimento de uma aplicação com *low-code* através de *Outsystems*. É um estágio que consiste na realização de um projeto que pretende fazer a implementação do *site* do Emprego Público *Online*.

As seguintes subsecções apresentam melhor o que é pretendido, os objetivos e principais desafios que tenho com a realização deste estágio, uma breve caracterização da empresa onde se insere este projeto, a descrição do que envolve a implementação do *site* e a organização do documento onde será apresentado o que é tratado em cada capítulo.

1.1 Objetivos

Este estágio tem vários objetivos, entre os principais, integrar-me no mundo real do trabalho e formar-me na tecnologia utilizada maioritariamente pela empresa, a tecnologia *Outsystems*. A integração nesta empresa foi feita de forma gradual, com formações em diversas áreas informáticas, como linguagens de programação, de forma a culminar com mais bases para a aprendizagem da plataforma *Outsystems*. Durante esta integração foram ultrapassadas várias dificuldades, principalmente na fase do processo de aprendizagem em *Outsystems*. Esta fase inclui todos os cursos introdutórios à *Outsystems*, os próprios cursos de *Outsystems*, o projeto de aprendizagem individual e atualmente, o projeto do qual fui integrada do Emprego Público *Online*. Este primeiro contacto com o mundo do trabalho para além me trazer muito conhecimento e experiência, também me acrescenta responsabilidade e compromisso.

1.2 Principais Desafios

Um dos principais objetivos que tenho é corresponder à expectativa que a empresa tem em mim, assim como corresponder à exigência que é necessária no projeto para com a equipa. Outro dos desafios é crescer tanto a nível pessoal como profissional. A nível pessoal no sentido de ter ainda mais responsabilidade, compromisso, inteligência emocional, comunicação, rigor e critério comigo mesma, bem como com toda a equipa. A nível profissional no sentido de ser mais capacitada, competente, com maior pensamento crítico e autoconhecimento com o decorrer do tempo. A maior motivação deste estágio é também a ambição da aprendizagem da plataforma de desenvolvimento de *low-code*, a *Outsystems*.

1.3 Caracterização da Empresa

Este projeto de estágio em Engenharia Informática é realizado em colaboração com a Babel Ibérica, SA [1], uma empresa multinacional especializada em consultoria informática que tem o objetivo de ajudar grandes empresas nos seus desafios e processos de negócio digitais. A Babel Ibérica, SA foi fundada em 2003 e sua sede oficial é em Madrid, na Espanha. É uma empresa que trabalha maioritariamente no desenvolvimento de aplicações com *low-code*, porém também tem a parte do *traditional development*. Na área de *low-code* é usado o desenvolvimento de *software* através de *Outsystems*, enquanto na área do *traditional development* são usadas linguagens de programação como *Cascading Style Sheets (CSS)*, *JavaScript*, *HyperText Markup Language (HTML)*, *Python*, etc.

Para a concretização dos objetivos, a Babel atua sempre, individual e coletivamente, de forma honesta, respeitando as regras comerciais, fiscais, laborais e ambientais. Conta com mais de 2000 profissionais altamente especializados em tecnologias de ponta. Atua até então em 14 países (Portugal, Espanha, Marrocos, Costa Rica, Equador, Colômbia, Chile, México, El Salvador, Guatemala, Honduras, EUA, Panamá e República Dominicana) e em Portugal conta com dois escritórios, Proença-a-Nova e Lisboa.

A Babel consiste em prestar serviços tecnológicos dirigidos a grandes empresas e organismos públicos, nos mais diversificados setores para dar resposta às necessidades dos clientes. Os setores em que esses serviços se podem enquadrar são vários, como, administração tributária, banca, emprego e segurança social, energia e serviços públicos, governo digital, indústria e retalho, saúde e assistência, segurança e defesa, transportes e logística, seguros e telecomunicações.

Nesse sentido surge um projeto ligado ao setor do emprego, através da fusão da Babel com a empresa Entidade de Serviços Partilhados da Administração Pública (eSPap), que pretende a implementação do *site* do Emprego Público *Online*. A missão da eSPap é assegurar a obtenção de ganhos de eficácia e eficiência, através da utilização racional de recursos públicos comuns e da prestação de serviços partilhados, contribuindo para um Estado mais ágil e direcionado para o desenvolvimento sustentável do país.



Figura 1.1: Logótipo da empresa.

1.4 Formação

A empresa para todos os novos colaboradores tem preparado um plano de formação, bem definido, com estipulação aproximadamente de tempo e dificuldade para assim resultar num crescimento e integração gradual e equilibrada dos membros num projeto.

Este plano focou-se na revisão de uma lista de cursos de diversas linguagens de programação para ter como resultado um melhor desempenho na aprendizagem de raciocínio e manuseamento do programa de desenvolvimento de *software Outsystems*. Depois dessa revisão, realizei todos os cursos propostos pela empresa da *Outsystems*, alcançando desde os cursos mais básicos até aos intermédios da aprendizagem da plataforma *Outsystems*. Após a consumação de todos esses cursos, no âmbito do projeto individual em *Outsystems* efetuei várias atividades propostas pelo orientador na aplicação interna da empresa. Com o propósito destas formações todas concluído integrei uma equipa com diversos membros com experiência em *Outsystems* para desenvolver o projeto do Emprego Público *Online*.

1.5 Descrição do Projeto

Com a necessidade de melhorar o *site* já existente da Bolsa de Emprego Público (BEP) por parte da eSPap, nasceu este projeto com a Babel. Com a exigência que se coloca no melhoramento dos seus processos através da tecnologia, a produção e desenvolvimento de *Software* deste projeto é realizada através da plataforma *Outsystems* na área de *low-code* da Babel. Assim, o projeto Emprego Público *Online* designa-se na produção de um *site*, focando-se essencialmente na desmaterialização do Procedimento Concursal Comum para uma plataforma comum e interfuncional.

1.6 Planificação e Cronograma

A planificação da formação serviu para delinear aproximadamente a previsão de horas gastas e nível de dificuldade que cada fase deste estágio dispôs. Este plano iniciou-se com a revisão da maioria dos conceitos que necessitei para melhor compreensão e aprendizagem da plataforma *Outsystems*, como também para adaptação de novos métodos de trabalho imprescindíveis. Essa primeira fase da formação foi repartida em cursos, que corresponderam a linguagens de programação diferentes, como linguagens *HTML/CSS*, *JavaScript*, *JQuery*, *Structured Query Language (SQL)*, *.NET*, *Experiência do utilizador (UX)/Interface do utilizador (UI)*.

Seguidamente, após a conclusão dos cursos referidos anteriormente, realizei o curso de *Outsystems* que se dividiu em quatro cursos, o primeiro, *Traditional web developer*, o segundo, *Reactive web developer*, o terceiro, *Mobile developer* e o quarto, *Front-end developer*. Estes quatro cursos abrangeram toda a formação que é necessária para realizar uma aplicação de raiz com a plataforma de *Outsystems*. Para aprimorar todo o conhecimento que

recebi dos cursos de *Oustsystems* realizei algumas tarefas de complementação à plataforma interna já existente da empresa. No fim de tudo isto, como era expectável, fui integrada num projeto de umas das equipas da empresa e é esperado que no final da integração deste projeto real para um cliente, consiga desenvolver as minhas capacidades e me gradue mais habilitada e produtiva.

Este estágio decorre durante o 2º semestre do ano letivo 2022/2023, com início a 22 de fevereiro de 2023 e término a 9 junho de 2023. O horário do estágio é o praticado pela empresa, um horário condensado. São 9h diárias de segunda a quinta, das 09:00h às 13:00h - 14:00h às 19:00h e 4h na sexta feira, das 09:00h às 13:00h.

No cronograma apresentado a seguir, na figura 1.2, estão expostas as etapas pelas quais passei ao longo deste estágio e as que sucedem.

Numa primeira etapa, entre os meses de fevereiro e março, fiz os cursos introdutórios específicos das tecnologias que foram necessárias para uma melhor aprendizagem em *Oustsystems*.

Numa segunda fase, entre março e abril, fiz a formação na área de *Oustsystems* com as bases dos cursos introdutórios. Enquanto que os cursos introdutórios foram um aprofundar do conhecimento que já tinha do curso de Engenharia Informática, este foi uma novidade.

Numa terceira fase, entre abril e maio, usei o conhecimento adquirido e, individualmente, fiz algumas tarefas propostas de melhoramento e acréscimo na plataforma interna da Babel.

Quarta fase, entre maio e junho, fui integrada num projeto numa equipa da empresa, começando por fazer tarefas mais simples e caminhando para as mais complexas, de forma gradual.

Quinta e última fase, entre junho e setembro, será a duração prevista do projeto de desenvolvimento e implantação do novo site do Emprego Público *Online* até à sua conclusão.

Até então, todas as fases e atividades deste cronograma não sofreram qualquer desvio daquilo que estava planeado inicialmente.

2023

Planificação Estágio

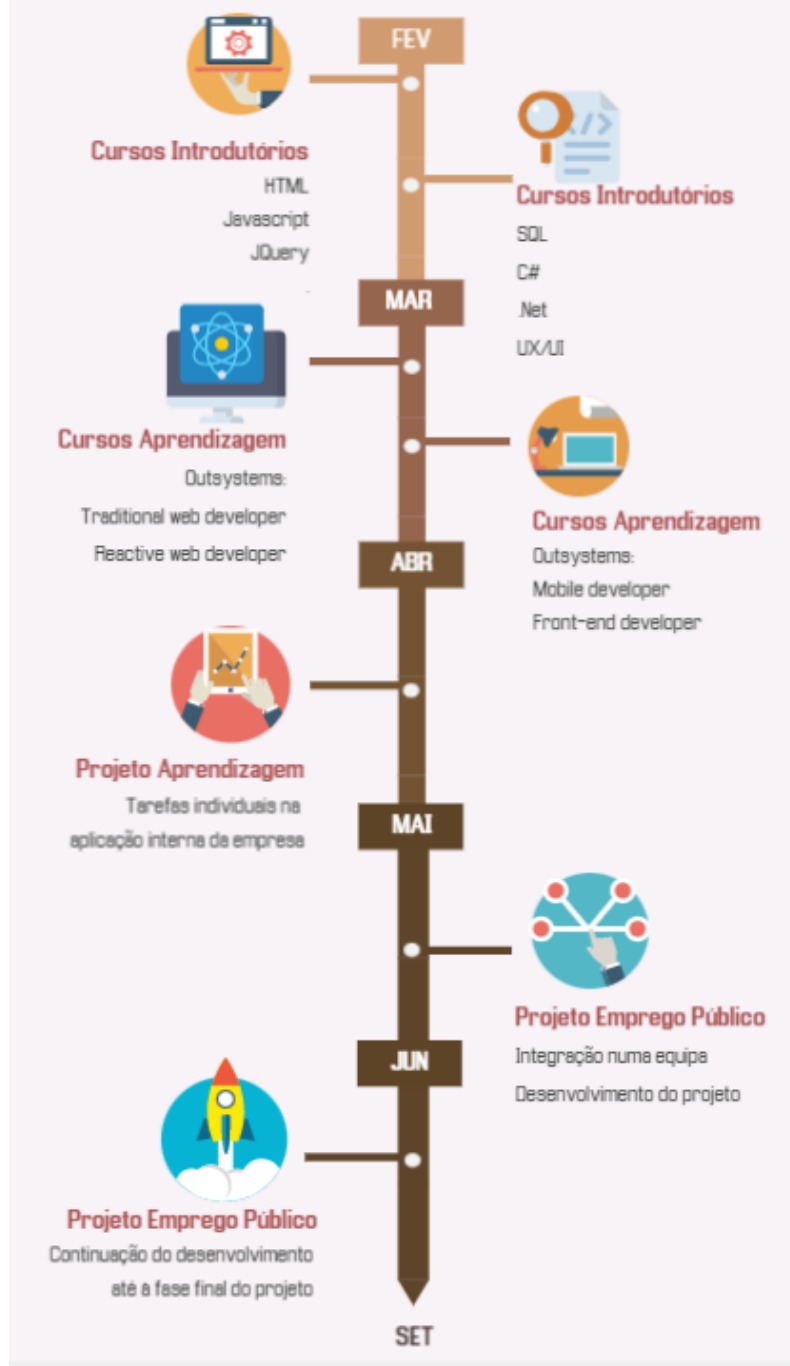


Figura 1.2: Cronograma-Planificação do estágio.

1.7 Organização do Documento

Este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo - **Introdução** - apresenta o projeto, contextualiza o mesmo, realizando uma breve caracterização da empresa onde este projeto está inserido. Descreve o projeto e os seus objetivos. Apresenta todas as etapas do plano de formação que foi apresentado, a descrição das atividades e a linha cronológica deste estágio. Por fim, expõe a respetiva organização do documento.
2. O segundo capítulo - **Tecnologias** - apresenta as tecnologias necessárias e utilizadas no processo de aprendizagem da plataforma *Oustsystems* e as ferramentas usadas diariamente para o desenvolvimento do projeto.
3. O terceiro capítulo - **Estado da Arte** - descreve as potenciais ferramentas de *Oustsystems* que integram este projeto.
4. O quarto capítulo - **Contextualização** - apresenta a proposta, o seu funcionamento, o seu propósito, as fases de desenvolvimento que dispõe e o processo completo desde o seu início até à fase atual.
5. O quinto capítulo - **Projeto: Emprego Público Online** - apresenta os requisitos da aplicação e o seu desenvolvimento na plataforma *Oustsystems*.
6. O sexto capítulo - **Conclusão** - faz uma revisão de todo o documento e expõe as principais conclusões.

Capítulo 2

Tecnologias Utilizadas

No processo de desenvolvimento de um *software* utilizam-se inúmeras tecnologias e ferramentas. Durante esse processo o *software* é dividido em duas camadas: *Front-end* e *Back-end* [2]. A camada *Front-end* é responsável por desenvolver por meio de código uma *interface* gráfica, normalmente com tecnologias *Web*: *HTML*, *CSS* e *Javascript*. Podemos dizer assim que é a parte visual com que o utilizador interage. Por outro lado, a camada *Back-end* é a parte não visível para o utilizador, é responsável em processar os dados e executar as ações que o *software* é suposto fazer, utilizando linguagens como *Python*, *C/C++*, *Java*, *Hypertext Preprocessor (PHP)*, entre outras. Durante o desenvolvimento do *software* podemos utilizar uma quantidade diferente de código, ou seja, utilizar *softwares* que usem pouco ou quase nenhum código (*low-code*) ou então utilizar mais código (*traditional development*) [3].

A plataforma *Ousystems* foi construída de forma a ser facilmente integrada noutros ambientes já existentes, sendo por isso, uma plataforma bastante ágil que funciona com *low-code*. Nesta plataforma pode-se usar *Ousystems* no *front-end* de forma a que este comunique com sistemas que já existam na *back-end*.

Neste estágio, a formação foi feita tanto em tecnologias *front-end* como *back-end*, que culminaram na formação da plataforma *Ousystems*. Antes disso, fiz revisão de tecnologias das quais já tenho bases do curso de Engenharia Informática, para realizar a aprendizagem do curso de *Ousystems* com sucesso. Segue-se uma breve introdução às tecnologias utilizadas durante todo este processo, assim como as ferramentas que uso diariamente.

2.1 Low-Code

Ferramentas de *low code* surgiram em virtude de dois fatores, permitir a criação de aplicações em pouco tempo e tornar esse processo menos democrático. O *low-code* traz também facilidade em termos de integração com as bases de dados. Os principais benefícios para a escolha desta ferramenta são, a economia de tempo (o que significa também redução de custos), a maior rapidez no tempo de construção da aplicação, a maior produtividade do tempo e a maior flexibilidade, pois permite aproveitamento de código já existente. Uma plataforma de *low-code* tem recursos como segurança, escalabilidade e visão geral da aplicação antes de ser executada e por estes motivos a maior parte das empresas opta por esta tecnologia. As empresas também tem a vantagem de assim reduzir pressão sobre os seus recursos de Tecnologias de Informação (TI) e aumentar a velocidade de lançamento da aplicação no mercado.

2.2 HTML/CSS

HTML é uma linguagem de marcação utilizada na construção de páginas na *Web*, ou seja, indica onde cada elemento do texto estará estruturado e onde será mostrado na página *web*. CSS é uma linguagem de folhas de estilo que é utilizada para definir como os documentos escritos na linguagem de marcação devem ser apresentados em termos de *layout*. Num cenário ideal, enquanto o HTML é utilizado para definir a estrutura e semântica do seu conteúdo, o CSS é utilizado para formatar esse mesmo conteúdo.

2.3 Javascript

Javascript [4] é uma linguagem de programação baseada em *scripts*, direcionada para aplicações *web*, de fácil execução. Pode ser usado para desenvolvimento *web*, desenvolvimento *mobile*, criação de sistemas de *back-end* e desenvolvimento de jogos. Algumas das vantagens desta linguagem de programação são:

- rapidez na leitura;
- versatilidade na linguagem;
- Sintaxe acessível;
- Não tem a necessidade de ser compilada;
- Ótima linguagem para iniciantes;
- Compatibilidade com uma grande variedade de plataformas e navegadores.

O *JavaScript* é uma das mais importantes tecnologias baseadas no *front-end* e, unindo-se ao trio HTML, CSS e PHP, formam um grupo de linguagens que abrangem praticamente todas as exigências do desenvolvimento de uma página completa, dinâmica e com boa performance.

2.4 JQuery

Jquery [5] é uma biblioteca livre que contém funções da linguagem de programação de *Javascript* que interage com páginas de HTML, desenvolvida para simplificar os *scripts* executados no navegador. Por ser uma biblioteca de *Javascript*, *JQuery* adiciona interatividade e enriquece a experiência do utilizador na internet. Abreviadamente as suas principais funcionalidades são:

- adiciona *scripts* e outros elementos, como animações;
- reduz códigos em *JavaScript*;
- cria uma sintaxe mais fluida;

- aumenta a compatibilidade entre códigos e navegadores;
- orienta a inspeção e a manipulação de documentos em HTML, XML ou XHTML.

Jquery é um *framework* simples e por todas estas razões é dos mais utilizados.

2.5 SQL

SQL [6] é uma linguagem de programação capaz de lidar com a pesquisa em bases de dados relacionais, baseados em tabelas. A programação SQL é usada para analisar e/ou executar tarefas em tabelas, principalmente através dos seguintes comandos: inserir (*insert*), pesquisar (*search*), atualizar (*update*) e excluir (*delete*). Porém também pode fazer coisas como escrever *queries* (comandos de consulta) com múltiplas informações. O principal objetivo do SQL é descomplicar o relacionamento direto do programador com as bases de dados, de maneira que possa manipular e/ou visualizar os dados com mais facilidade. SQL é usado em qualquer área que lide com tecnologia que tenha as suas próprias bases de dados, daí ser bastante solicitado.

2.6 .NET

O *.Net* [7] é um *framework* da *Microsoft* para desenvolvimento *Web* e como qualquer *framework* de desenvolvimento, fornece facilidades de reutilização e reaproveitamento de código. O *.NET* foi desenvolvido para disponibilizar um ambiente único para desenvolvimento e execução de sistemas e aplicações. Esta plataforma conta com recursos interessantes, como por exemplo a conexão com bases de dados e comunicações de rede e criptografia. O *.Net* tem inúmeras soluções codificadas para problemas mais frequentes dentro da programação, uma vez que o *framework* suporta diversas linguagens de programação como o *C Sharp* e o *Visual Basic*, bem como ferramentas de desenvolvimento integrado (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado (IDE)s), como o *Visual Studio*, que ajuda os programadores a produzir aplicações de alta qualidade.

2.7 UI/UX

UI e UX [8] são dois termos que vamos sempre ouvir na área da tecnologia, pois são dois aspectos muito importantes. A UI refere-se a tudo com a qual um utilizador possa interagir para usar um produto ou serviço digital. Isto inclui tudo, desde telas, botões, alternâncias, ícones, teclados, sons, luzes e outros tipos de elementos visuais. A UX refere-se a toda a interação que o utilizador tem com um produto, incluindo como se sente sobre a interação. Na medida em que a UI foca-se no design visual da aplicação, o UX foca-se na compreensão das necessidades do utilizador.

2.8 Ferramentas

2.8.1 *Microsoft Teams*

O *Microsoft Teams* [9] é uma aplicação da *Microsoft* que permite a cooperação entre todos os elementos da empresa. É uma aplicação que consegue integrar pessoas, conteúdos e ferramentas num só lugar, de forma remota. Com o uso da *Microsoft Teams* consigo comunicar com toda a minha equipa através de mensagens, chamadas de áudio, videochamadas e reuniões *online*. Para além disso, consigo partilhar arquivos e/ou documentos, armazenar ficheiros, ter acesso ao calendário que organiza os meus compromissos, assistir a palestras dadas por membros da empresa e quaisquer outros eventos.

2.8.2 *Microsoft Azure DevOps*

O *Microsoft Azure DevOps* [10] é uma plataforma que fornece recursos de controlo de versões, automação, compilação e testes que ajudam no ciclo de vida do desenvolvimento de um *software*. O termo *DevOps* é uma combinação de desenvolvimento (dev) de software e operações de TI (ops). O *Microsoft Azure DevOps* permite que funções anteriormente isoladas – desenvolvimento, operações de TI, engenharia da qualidade e segurança – atuem de forma coordenada e colaborativa para gerar produtos melhores e mais confiáveis. Ao adotar uma cultura de *DevOps* em conjunto com as práticas e ferramentas de *DevOps*, as equipas ganham a capacidade de responder melhor às necessidades dos clientes, aumentar a confiança nas aplicações que constroem e cumprir as metas empresariais mais rapidamente. As equipas que adotam a cultura, as práticas e as ferramentas de *DevOps* apresentam alto desempenho, criando produtos melhores, com mais rapidez, para maior satisfação do cliente. Esse aumento na colaboração e na produtividade também é essencial para cumprir metas empresariais como :

- Acelerar a colocação no mercado;
- Adaptar ao mercado e à concorrência;
- Manter a estabilidade e a confiabilidade do sistema;
- Melhorar o tempo médio de recuperação.

O *DevOps* influencia o ciclo de vida do aplicativo em todas as fases do planeamento, do desenvolvimento, da entrega e da operação. Cada fase depende das demais e elas não são específicas da função. Em uma verdadeira cultura de *DevOps*, cada função está envolvida de alguma forma em cada fase.

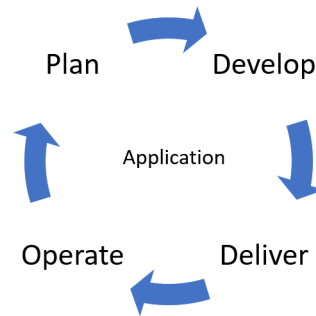


Figura 2.1: *DevOps*: Ciclo de vida de uma aplicação.

Na fase de planeamento, é definido e descrito os requisitos e as funcionalidades que o cliente quer para a aplicação. Uma das funcionalidades do *DevOps* é o *Backlog*. *Backlog* serve para planear o projeto adicionando nele *User Stories*. Um *User Story* é a unidade mais baixa de trabalho, que vai conter toda a informação detalhada dos requisitos da aplicação, o nível de prioridade que tem no projeto, uma estimativa de horas que serão gastas na realização dessas tarefas e que membro da equipa está a trabalhar nela. Um *Backlog* é representado por uma lista de *User Stories* ordenados pelo valor de negócio.

Na fase de desenvolvimento é realizada a gravação, testagem, revisão e integração do código pelos membros da equipa.

Na fase de entrega é realizado o processo de implementação nos ambientes de produção de maneira consistente e confiável.

A fase de operação envolve manter, monitorar e solucionar problemas da aplicação em ambiente de produção. Ao adotar as práticas de *DevOps*, é garantida a confiabilidade do sistema, reforçando a segurança e a governança. É possível também identificar os problemas antes que eles afetem a experiência do cliente e mitigar os problemas rapidamente quando ocorrem.

2.8.3 *Figma*

O *Figma* [11] é uma ferramenta de *design* e prototipagem de UI/UX que pode ser usada para criar sites, aplicações ou pequenos componentes de *interface* da aplicação. É um editor gráfico de vetor e prototipagem de projetos de *design* baseado principalmente no navegador *web*, com ferramentas *offline* adicionais para aplicações *desktop* como *GNU/Linux*, *macOS* e *Windows*. Com o uso do *Figma* consigo aceder aos *layouts* da aplicação, criados pela *UI/UX designer* da nossa equipa.

Capítulo 3

Estado da Arte da Plataforma *Outsystems*

Este capítulo apresenta e analisa diversas tecnologias de *Outsystems* atualmente disponíveis, assim como uma introdução à estrutura da plataforma.

Outsystems [12] é uma plataforma de desenvolvimento de *software* em *Low-code* que possibilita a criação de aplicações personalizadas de forma rápida e com pouco código, em comparação com as linguagens de programação convencionais. Nesta plataforma toda a programação é realizada de uma forma visual, construindo telas, fluxos lógicos, conexão com bases de dados e consumindo serviços Representational State Transfer (REST), Simple Object Access Protocol (SOAP) e Systemanalysis Programmentwicklung (SAP). Nas telas a programação é feita na base de arrastar e soltar. Nos fluxos lógicos são chamadas as páginas de forma cronológica que determinam o funcionamento da aplicação. Na conexão com as bases de dados, a plataforma está preparada para interagir com bases de dados como *Oracle*, *SqlServer* e *MySQL*, o que a torna uma ferramenta compatível com o ambiente existente na maioria das grandes empresas. Os serviços REST e SOAP permitem a comunicação entre aplicações. O serviço *REST* é um conjunto de princípios e definições necessários para a criação de um projeto com interfaces bem definidas, enquanto que, o serviço SOAP é um protocolo para troca de informações estruturadas numa plataforma descentralizada e distribuída. Enquanto que o serviço *REST* opera por meio de uma *interface* consistente para aceder aos recursos pedidos, o serviço *SOAP*, usa serviços e não dados. Além disso, opera por meio de diferentes *interfaces*. A SAP é um *software* de gestão empresarial, que permite que o cliente/empresa consiga adequar o sistema ao seu modelo de negócio, funcionando através de etapas, responsáveis por gerir os processos de cada área específica. Todas estas ferramentas são indispensáveis para trazer soluções na criação de aplicações com a *Outsystems*.

A principal diferença de *Outsystems* em relação a outras ferramentas semelhantes assenta na metodologia de desenvolvimento proposta e na flexibilidade apresentada. Segue-se uma introdução à plataforma *OutSystems* onde será abordada a metodologia de desenvolvimento que esta promove, os seus componentes e uma visão mais detalhada do componente responsável pela execução das aplicações desenvolvidas.



Figura 3.1: Logótipo *Outsystems*.

3.1 Metodologia

A plataforma *Outsystems* oferece uma ampla variedade de recursos e funcionalidades para ajudar os programadores a criar aplicações de alta qualidade, respeitando as tanto as necessidades tecnológicas como as necessidades de negócio das organizações. A *Outsystems* tem uma metodologia própria chamada *Agile Platform* que é baseada em metodologias ágeis que permite que sejam construídas aplicações com rapidez e com eficiência. Através de *Outsystems* é possível desenvolver projetos complexos, como, implementar soluções de *software* em ambientes *web* e/ou para dispositivos móveis, consumir e disponibilizar serviços *web* e efetuar integrações com códigos bases e de dados externos. Para além disso a plataforma oferece suporte a várias metodologias de desenvolvimento, incluindo o *SCRUM* e o *Kanban*.

3.1.1 SCRUM

O *Scrum* [13] é uma estrutura ágil de gestão de projetos que ajuda a estruturar e gerir o trabalho por meio do conjunto de valores, princípios e práticas. A definição do *Scrum* é baseada no empirismo e no pensamento enxuto. O empirismo diz que o conhecimento vem da experiência e que as decisões são tomadas com base no que é observado. O pensamento enxuto reduz o desperdício e concentra-se no essencial. Além disso, o *Scrum* traz vários benefícios para a gestão de projetos, como:

- Maior flexibilidade e adaptabilidade ao projeto e à equipa de trabalho;
- Colaboração e comunicação aprimoradas internamente;
- Maior transparência e visibilidade;
- Entrega mais rápida;
- Qualidade aprimorada;
- Maior satisfação do cliente.

O *Scrum* ajuda assim as equipas a responder rapidamente às mudanças e a lidar com os desafios que possam surgir ao longo do projeto.

3.1.2 Kanban

O *Kanban* [14] é um sistema visual de organização e gestão de trabalho que procura conduzir cada tarefa para um fluxo predefinido. É um método de organização e administração de projetos/tarefas que ajuda a equipa a visualizar o trabalho, maximizar a eficiência e a melhorar continuamente, adaptável à realidade de diferentes negócios. *Kanban* pretende assim aumentar a produtividade, organizando o fluxo de tarefas frente às demandas existentes. Este sistema traz vários benefícios, como:

- Maior dinamismo entre as pessoas;
- Promove um ambiente de colaboração;

- Maior visibilidade de fluxo;
- Maior velocidade de entrega;
- Alinhamento entre metas e execução;
- Melhor previsibilidade;
- Melhor gestão de equipas.

3.2 Organização

Os projetos que seguem a *OutSystems Agile Methodology* são compostos por uma sequência de iterações, denominadas de *Sprints*, cujo objetivo é criar uma versão funcional do sistema no final. Estas iterações, com duração de uma ou duas semanas, são compostas por atividades de análise, desenvolvimento e teste, com a finalidade que todas as funcionalidades do sistema final sejam terminadas. Assim, no final de todas as iterações consegue-se um sistema com todas as funcionalidades disponíveis, que foram sendo testadas, adaptadas e aprovadas paralelamente e progressivamente com o seu desenvolvimento. Com esta metodologia, a *OutSystems* promove uma adaptação constante às mudanças do mercado e às necessidades dos clientes, pois independentemente da fase do ciclo de vida das aplicações, novas funcionalidades podem ser facilmente adicionadas, erros corrigidos e *feedback* analisado, com riscos reduzidos e sem graves consequências para a aplicação.

Ao contrário do comum das ferramentas de desenvolvimento, esta plataforma possui um estilo de programação visual "drag'n'drop" em que é possível a construção de aplicações sem ter que se escrever qualquer linha de código. Desde o desenho do modelo de dados, criação de *interfaces*, definição de lógica de negócio ou instalação, tudo pode ser feito visualmente. Esta abordagem permite diminuir o desalinhamento existente entre o negócio e as TI, pois torna as aplicações mais fáceis de compreender, para os elementos do negócio e permite aos elementos das TI responder atempadamente às necessidades que lhes são apresentadas.

3.3 Estrutura da Plataforma

Esta plataforma é destinada ao desenvolvimento de aplicações para *Internet/Intranet* ou redes móveis e é composta por quatro componentes: *Service Studio*, *Integration Studio*, *LifeTime* e *Service Center*.

Estes componentes direcionam-se às fases de desenvolvimento, integração de sistemas, execução e monitorização das aplicações construídas. É apresentada de seguida uma descrição mais detalhada de cada um destes componentes.

3.3.1 *Service Studio*

O *Service Studio* é o componente/ambiente de desenvolvimento visual, integrado da *OutSystems*, destinado à criação, alteração e instalação das aplicações de *software web*, *Android* e *IOS* desenvolvidas.

Todo o processo de desenvolvimento é realizado neste componente, desde o desenho e criação do modelo de dados, desenho de *interfaces* e criação da lógica de negócio.

A instalação das aplicações é feita neste componente recorrendo ao processo denominado *1-Click-Publishing*, o qual verifica, guarda, efetua o *upload* no componente de execução, compila e instala a aplicação. No fim, se todo este processo decorrer sem erros obtém-se uma aplicação completamente executável.

3.3.2 Integration Studio

O *Integration Studio* é o componente de integração. Neste componente é possível fazer a integração com diversos outros sistemas. É usado para integrar a plataforma *Outsystems* em código externo, seja escrevendo código como novas ações ou importando código existente como ações e entidades. Permite uma integração perfeita entre diferentes aplicações de *software*, permitindo uma comunicação e troca de dados eficiente. Com este componente é possível criar novas extensões, verificar e publicar uma extensão e fazer conexão ao servidor da plataforma *Integration Studio*. Deste modo, para consumir um serviço *web* no *Integration Studio*, é necessário criar uma extensão com uma ação para cada método do serviço da *web*, de seguida criar um modelo de dados (estruturas) para representar os tipos de dados do serviço da *web* e por fim, importar o serviço da *web* no código da extensão.

Com este componente é concedida e facilitada a interacção das aplicações com os sistemas existentes.

3.3.3 LifeTime

O *LifeTime* é o componente de gestão do ciclo de vida das aplicações. É usado para implementar e gerir aplicações de diferentes ambientes, assegurando todo o processo desde o desenvolvimento até à conclusão das aplicações. Com este componente é simplificada e acelerada a gestão do ciclo de vida das aplicações.

3.3.4 Service Center

O *Service Center* é o componente de monitorização e gestão das aplicações. Este componente permite coordenar e controlar administradores de sistema, *managerse* equipas de gestão de operações. Com este componente é possível fazer a organização de todos os objectos necessários à execução, desde aplicações, serviços, adaptadores e quaisquer outros recursos.

Com o conjunto dos componentes descritos anteriormente a plataforma *OutSystems* aborda praticamente todos os factores necessários ao desenvolvimento de aplicações empresariais.

3.4 Platform Server

Para todos os componentes, mencionados em cima, funcionarem e se interligarem entre si, é necessário o *Platform Server*. *Platform Server* como o próprio nome indica é o servidor da plataforma *Outsystems*, por isso, é o componente central que orquestra todas as atividades

de tempo de execução, implementação e gestão de todas as aplicações. Esta plataforma é construída com base numa arquitetura de microserviços que permitem escalabilidade, tanto na vertical como na horizontal. Na *OutSystems* todas as aplicações dependem de arquiteturas e estruturas padrão e o *Platform Server* é o que torna isso possível. Assim, as principais vantagens do *Platform Server* passam por maximizar o desempenho e a segurança das aplicações, fornecer abertura para processos de entrega contínua, suportar inovação contínua e permitir aos programadores criarem aplicações rapidamente com facilidade.

A função do *Platform Server* passa por receber aquilo que foi realizado, após a ordem de publicação, proveniente do *Service Studio*, enviando um ficheiro com a definição completa e detalhada do que foi desenvolvido visualmente no componente de desenvolvimento. Este ficheiro é um ficheiro *Outsystems Modeling Language* (OML) que armazena código numa *sintaxe* semelhante ao *Extensible Markup Language* (XML). Após o *upload* do ficheiro com a definição da aplicação este é processado e utilizado num gerador de código responsável por gerar o todo o código da aplicação que anteriormente tinha sido desenvolvida visualmente. São geradas as classes, as *queries* SQL, os *ecrãs* e tudo o que é necessário à execução da aplicação. Uma vez gerado o código da aplicação, este é compilado e os resultados são instalados. A instalação corresponde a colocar os ficheiros criados nos respetivos locais. Qualquer erro que seja encontrado durante este processo é reportado, para ser corrigido. Após este processo a aplicação está pronta para executar. Qualquer pedido que lhe chegue vai utilizar os executáveis criados anteriormente que vão atuar quer sobre a base de dados, quer sobre outros ficheiros ou sistemas com que seja suposto a aplicação comunicar.

Capítulo 4

Contextualização do Estágio

Neste capítulo será descrito com maior pormenor o projeto do qual estou a integrar na Babel, que foi conhecido no capítulo 1.2.

Tendo como base a situação da necessidade de melhoramento do site já existente da BEP surgiu este projeto que pretende reformular e modernizar toda a plataforma de acesso às candidaturas do Emprego Público *Online*, bem como o acesso a toda a informação que isso exige. Daí surgiu uma proposta estável que estivesse alinhada com todas as necessidades, gerais e particulares, e evolução que esta plataforma necessita, assim como com os objetivos da *Outsystems*. Segue-se então a descrição da solução proposta, a descrição dos requisitos necessários ao funcionamento da aplicação e a toda a sua fase de desenvolvimento.

4.1 Abordagem da Proposta

Este projeto adota uma filosofia de desenvolvimento de *software Agile* que oferece facilidade em escolher metodologias e procedimentos que melhor se enquadram na equipa. Os valores e princípios de *software Agile* não ordenam ou prescrevem a forma como a equipa trabalha, mas procura ajudar a equipa a pensar e interagir de forma a conseguir agilidade. Agilidade é o processo de adaptar e melhorar a forma de trabalhar. Devido a esta capacidade de adaptação, *Agile* não é considerada uma metodologia. Assim, em vez de mencionar o que exatamente a equipa deve fazer e como, *Agile* oferece um conjunto de valores e princípios que a equipa pode adotar. Existem vários sistemas que ajudam as equipas a seguir esta filosofia e a equipa do qual integro utiliza especificamente a *DevOps*. Nesta equipa existem vários cargos: o *Project Manager*, o *Business Analyst* e a restante equipa de desenvolvimento.

O *Project Manager* é o responsável por fazer a gestão e organização da equipa de desenvolvimento que produz o *software*, enquanto que o *Business Analyst* é o responsável por definir e sequenciar os requisitos do *software*. Este processo de desenvolvimento está dividido em fases/eventos:

- **Planeamento do sprint:** reunião em que a equipa em conjunto define um objetivo, que será uma componente do sistema a desenvolver no próximo *sprint*, validada pelo *Business Analyst*;
- **Sprint:** período de tempo definido, normalmente entre 1 semana a 3 semanas, em que a equipa de desenvolvimento tenta fazer a componente definida no planeamento do *sprint*, no período de tempo definido. Caso a equipa falhe com a meta proposta pode passar as tarefas que faltam para o próximo *sprint*, definindo novamente um objetivo;
- **Daily:** reunião da equipa diária com a duração de cerca de 15 minutos onde a equipa se

coordena para alcançar o objetivo definido no *sprint*, isto faz com que os elementos da equipa sejam transparentes e se consigam adaptar a novas alterações e informações;

- **Avaliação do sprint:** reunião após o *sprint* em que a equipa apresenta ao cliente o que conseguiu desenvolver durante o *sprint* e receber algum feedback para melhorar e adaptar futuros *sprints*;
- **Retrospectiva do sprint:** reunião onde a equipa reflete e debate como correu o *sprint* para haver uma adaptação a futuros *sprints* de maneira a que o trabalho seja mais eficaz.

O desenvolvimento do sistema completo também está dividido em 3 componentes: o *backlog* do produto, o *backlog* do *sprint* e o incremento. O *backlog* do produto é uma lista de funcionalidades que provém do cliente e que o *Business Analyst* organiza e insere no sistema, ordenada pela sua importância. Por outro lado, o *Backlog* do *sprint* é um conjunto de funcionalidades presentes no *backlog* do produto, escolhidas pela equipa no planeamento do *sprint* para que essas funcionalidades sejam desenvolvidas durante esse *sprint*. Já o incremento é o resultado de um *sprint* onde as funcionalidades são integradas no sistema alvo/final.

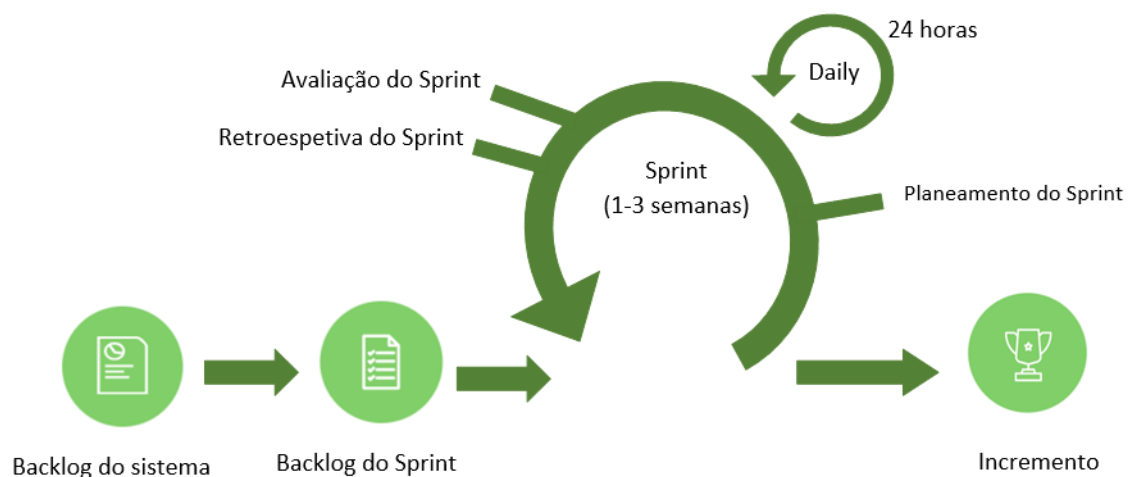


Figura 4.1: Diagrama representativo das etapas de desenvolvimento do *software* no *DevOps*.

O processo completo do desenvolvimento de *software* no *DevOps*, visível na figura anterior, passa pelas fases de levantamento inicial do *backlog* do produto que de seguida é acompanhado de ciclos. Estes começam com o planeamento do *sprint* onde são definidas as

funcionalidades que são desenvolvidas no *sprint*, constituindo assim o *backlog* do *sprint*. Seguidamente, inicia-se o *sprint* onde são produzidas as funcionalidades presentes no *backlog* do *sprint*. No decorrer do *sprint*, há reuniões diárias, as *daily* entre a equipa de desenvolvimento. No final, ao obter o incremento, há a avaliação do *sprint* em que há uma validação do incremento pelo cliente. Por fim, ocorre a retrospectiva do *sprint* onde a equipa reflete acerca deste e decide as melhorias para o próximo planeamento.

4.2 Fases de Desenvolvimento

O desenvolvimento deste projeto está dividido em fases, a fase do levantamento, a fase de planeamento, a fase de desenvolvimento, a fase de testagem e a fase de conclusão, melhor detalhadas a seguir:

1. **Fase 1 - Levantamento** - Participação no *onboarding* e integração na equipa de desenvolvimento, com o entendimento teórico geral do projeto e familiarização de componentes importantes do projeto;
2. **Fase 2 - Planeamento** - Período de organização, iniciado com o levantamento dos requisitos, com o objetivo de perceber o seu funcionamento.
3. **Fase 3 - Desenvolvimento** - Período de implementação das funcionalidades do projeto.
4. **Fase 4 - Testagem** - Período de análise e revisão do que foi desenvolvido, execução dos testes de segurança e a deteção de *bugs*;
5. **Fase 5 - Conclusão** - Versão final do projeto, com integração da aplicação no ambiente do utilizador.

4.3 Planificação do Trabalho

O projeto do Emprego Público *Online* iniciou-se com 2 semanas de reuniões, palestras e *workshops* onde foi apresentada a proposta e os vários componentes que o projeto requer, de forma a expor a visão e o propósito deste serviço. De seguida, houve a integração na equipa de desenvolvimento, analisando o estado do sistema, decisão das tecnologias a utilizar e divisão de tarefas. Seguidamente, consoante os requisitos existentes procedeu-se à implementação do projeto com o desenvolvimento na plataforma *Outsystems*, que é a fase atual. Esta fase estima-se que tenha a duração de 5 meses e é a mais crítica, pois para além de ser a fase de maior aprendizagem, é também a fase de potenciais dificuldades. Terminada a fase de desenvolvimento, é necessário testar aquilo que foi desenvolvido ao longo do tempo para finalizar o projeto e publicar a sua versão final para que o utilizador possa aceder.

Capítulo 5

Projeto: Emprego Público *Online*

A proposta do projeto do Emprego Público *Online* corresponde a uma nova plataforma comum e interfuncional, a partir da Desmaterialização do Procedimento Concursal Comum, onde pretende que sejam apresentadas, acedidas e solicitadas vagas de Emprego Público. Esta plataforma contribuirá para uma rigorosa política de recursos humanos na Administração Pública. Para a realização desta proposta foram acordados requisitos, requisitos esses que dão a conhecer que tipo de entidades a aplicação vai ter, como é que o *software* se vai comportar, assim como também ajuda a entender as características gerais da aplicação.

5.1 Requisitos

5.1.1 Perfis

Para esta proposta foram apresentados 14 perfis diferentes necessários para o desenvolvimento da aplicação, que serão todas as entidades em que é possível entrar na plataforma.

- Administrador eSPap - Acesso a administração e configuração do ambiente (Equipa Técnica);
- Administrador EP-On - Acesso a administração e configuração do sistema (Entidade Gestora do Emprego Público *Online*);
- *Helpdesk* - Apoio aos organismos e utilizadores nas funcionalidades. Consegue apenas simular, não insere ou altera nada na base de dados nem carrega, gera ou descarrega qualquer documento;
- Organismo/Administrador - Administrador no organismo. Configura o EP-On para o seu organismo;
- Organismo/RH - Recursos Humanos do organismo;
- Organismo/Responsável RH - Responsável de Recursos Humanos do organismo, que valida as tarefas do perfil RH;
- Organismo/Júri - Júri designado para o procedimento;
- Organismo/Júri Secretariado - Secretariado do júri para o procedimento;
- Organismo/Júri Consultor - Especialista que auxilia o júri do procedimento;
- Organismo/DMO - Dirigente Máximo do Organismo;

- Candidato/TFP - Candidato - Trabalhador em Funções Públicas, com vínculo de emprego público;
- Candidato/CR - Candidato - Cidadão Registado, sem vínculo de emprego público;
- Entidade Insetiva - Utilizador com acesso dado a um ou mais procedimentos em que pode consultar todo o sistema, relativamente aos procedimentos que lhe foram permitidos;
- Membro Governo - Utilizador com acesso ao *Power BI*, para consulta de dados em mapas.

5.1.2 Requisitos Não Funcionais

Os requisitos não funcionais surgem de maneira a percebermos os atributos de qualidade, as restrições ou os requisitos impostos ao sistema para melhor compreensão das características gerais do *software*. Deste modo, seguem alguns exemplos de requisitos não funcionais deste projeto, seguidos de uma breve descrição:

1. *Interfaces* e experiência de utilização - Deve ser facilmente identificável que o *website* correspondente ao “Emprego Público”, que devem ser seguidas as recomendações constantes do portal <http://usabilidade.gov.pt> e que a plataforma deve ser adaptativa, prevendo a sua utilização por dispositivos móveis.
2. Autenticação - A plataforma pressupõe uma autenticação forte por parte do requerente.
3. Entidades e Histórico - Todos os dados inseridos ou alterados nos modelos são replicados para uma entidade de histórico. Na entidade de histórico consegue-se ver todos os registos desde a sua criação, edição e eliminação. Todas as entidades com dados pessoais tem uma entidade de registo para consultas, onde regista que utilizador consultou e que versão da informação consultou.
4. Repositório Documental - Toda a documentação carregada pelos utilizadores do sistema deve ser enviada para um repositório;
5. Assinatura Digital - Deverá existir um módulo que permita assinar digitalmente toda a documentação produzida pelo EP-On ou carregada no EP-On, utilizando o Cartão de Cidadão (CC) ou Chave Móvel Digital (CMD).

5.1.3 Requisitos Funcionais

Os requisitos funcionais surgem de maneira a descreverem a maneira como a aplicação se deve comportar. Deste modo, seguem alguns exemplos, os mais abrangentes, de requisitos funcionais deste projeto, seguidos de uma breve descrição:

1. Registo do Utilizador
Perfis: Anónimo, TFP e CR;

Objetivo: Criar e gerir um registo de utilizador no sistema para entrar na plataforma

Requisitos:

- Criar um registo de utilizador;
- Consultar o registo.;
- Alterar registo;
- Cancelar registo;
- Validação de número de telemóvel;
- Validação de *email*;
- Envio da informação para a plataforma de Gestão de Utilizadores da eSPap;
- Atribuir o perfil de Trabalhador em Funções Públicas ou Cidadão Registado de acordo com a informação prestada na criação do registo.

2. Autenticação

Perfis: Anónimo;

Objetivo: Poder fazer autenticação como utilizador no sistema para entrar na plataforma

Requisitos:

- Autenticação com o CC, CMD ou *user e password* na plataforma;
- Escolher o perfil, se o utilizador tiver vários perfis associados;
- Terminar sessão.

3. Alteração de dados do utilizador

Perfis: Todos;

Objetivo: Como utilizador alterar dados para os manter atualizados e ser contactado, notificado e receber informações enviadas pelo sistema

Requisitos:

- Alteração de qualquer dado, exceto o *NIF* e o *login*;
- O sistema, se o utilizador mudar de número de telefone ou de endereço de *email*, tem de pedir ao sistema de autenticação para validá-los.
- O sistema tem de enviar os novos dados para o sistema de autenticação.

4. Gerir requisitos de admissão

Perfis: Administrador EP-On;

Objetivo: Como Administrador EP-On, indicar quais os requisitos de admissão que um candidato tem de cumprir, para permitir aos Organismos selecionarem quando estiverem a criar o procedimento concursal

Requisitos:

- Consultar uma lista de requisitos de admissão já inseridos na plataforma, filtrar essa lista pelo texto do requisito e consultar esses requisitos;
- Inserir ou alterar os requisitos de admissão que podem ser utilizados pelos Organismos na criação do procedimento concursal;
- Indicar qual o diploma legal que aprova o requisito de admissão.

5. Gerir métodos de seleção

Perfis: Administrador EP-On;

Objetivo: Como Administrador EP-On, criar e gerir parâmetros de avaliação, para que possam ser selecionados pelos Júris dos procedimentos, na definição dos parâmetros de avaliação de cada procedimento

Requisitos:

- Criar parâmetros obrigatórios e facultativos para cada tipo de procedimento e para cada tipo método de seleção;
- Definir a ponderação de cada método de seleção indicando um valor mínimo e máximo;
- Dar uma descrição ao método de seleção.;
- Indicar a estrutura do método de seleção;
- Introduzir orientação, tutoriais, guias, para os membros do júri tomarem consciência das decisões que estão a tomar;
- Consultar os parâmetros de avaliação por procedimento e método de seleção;
- Alterar métodos e ou parâmetros;
- Apagar métodos e ou parâmetros;
- O sistema tem de gerar versões dos parâmetros de avaliação sempre que forem feitas alterações;
- Criar métodos de seleção obrigatórios ou complementares.

6. Publicitação de oferta

Perfis: RH;

Objetivo: Como RH e depois da oferta ter sido publicada em Diário da República, publicitar a oferta de emprego no EP-On, para que seja disponibilizada aos interessados e para que sejam abertas as candidaturas

Requisitos:

- Indicação do N^o do Diário da República em que foi publicado o aviso;
- Publicitar a oferta;
- A oferta fica publicitada no dia útil seguinte às 00h00;
- Calcular a data limite da oferta com base nos dias que foram indicados na 'Identificação das necessidades de recrutamento', contando apenas dias úteis;

- Manter o formulário das candidaturas aberto para receber as candidaturas desde a data de publicitação até à data limite;
- Após a oferta ser publicitada, notificar os candidatos em bolsa para se candidatarem à oferta.

7. Detalhe da oferta

Perfis: Todos os perfis, incluindo não autenticado;

Objetivo: Para qualquer perfil, consultar o detalhe de uma oferta e dar a possibilidade de candidatura à mesma ou adicionar aos favoritos

Requisitos:

- Consultar toda a informação da oferta;
- Consultar ou descarregar a ata de métodos de seleção;
- Consultar ou descarregar qualquer documento associado à oferta;
- Adicionar a oferta às ofertas favoritas;
- Aceder ao formulário de candidatura ou Criação de CV (caso ainda não tenha sido criado) se a oferta ainda estiver ativa;
- Aceder à funcionalidade de criação de alerta, para ser notificado para futuras ofertas que sejam encontradas com o mesmo critério de pesquisa.

8. Candidatura às ofertas

Perfis: TFP; CR;

Objetivo: Como TFP ou CR, criar candidatura a uma oferta de emprego publicitada no EP-On, para poder ir aos métodos de seleção e iniciar funções no Organismo que criou a oferta

Requisitos:

- Criar a candidatura utilizando todos os dados já existentes no CV, incluindo documentos que estejam no CV;
- Indicar que se cumpre os requisitos de admissão identificados da oferta e inserir documentos comprovativos;
- Consultar a candidatura e fazer download dos documentos comprovativos que foram carregados;
- Editar candidatura e apagar documentos e carregar novos até ao fim do prazo de candidatura.
- Receber um email de confirmação quando submetida a candidatura com um PDF com os dados da candidatura;
- Eliminar a candidatura.

9. Análise das candidaturas e verificação dos requisitos de admissão

Perfil: Júri, Júri Consultor;

Objetivo: Como Júri ou Júri Consultor, validar as candidaturas e documentos comprovativos de forma a fazer correções ou excluir candidatos que não comprovem a informação prestada na candidatura

Requisitos:

- Consultar todos os campos da candidatura com os dados inseridos pelos candidatos e correções que tenham já sido feitas;
- Contactar os candidatos para solicitar esclarecimentos ou informação adicional quanto aos dados das suas candidaturas;
- Corrigir os dados dos campos da candidatura;
- Indicar que campos não são comprovados e o motivo porque não são comprovados;
- Inserir observações finais;
- Ir para a geração das atas para gerar a ata de admitidos e excluídos, com uma lista anexa com os candidatos que se encontrem nessas condições.

5.2 Desenvolvimento

Com os requisitos fechados, os *layouts* desenhados e a arquitetura da aplicação definida, podemos avançar para o desenvolvimento da aplicação, no ambiente de *Outsystems*. O exemplo que vou demonstrar refere-se à realização do ecrã de "Listas de Perfis".

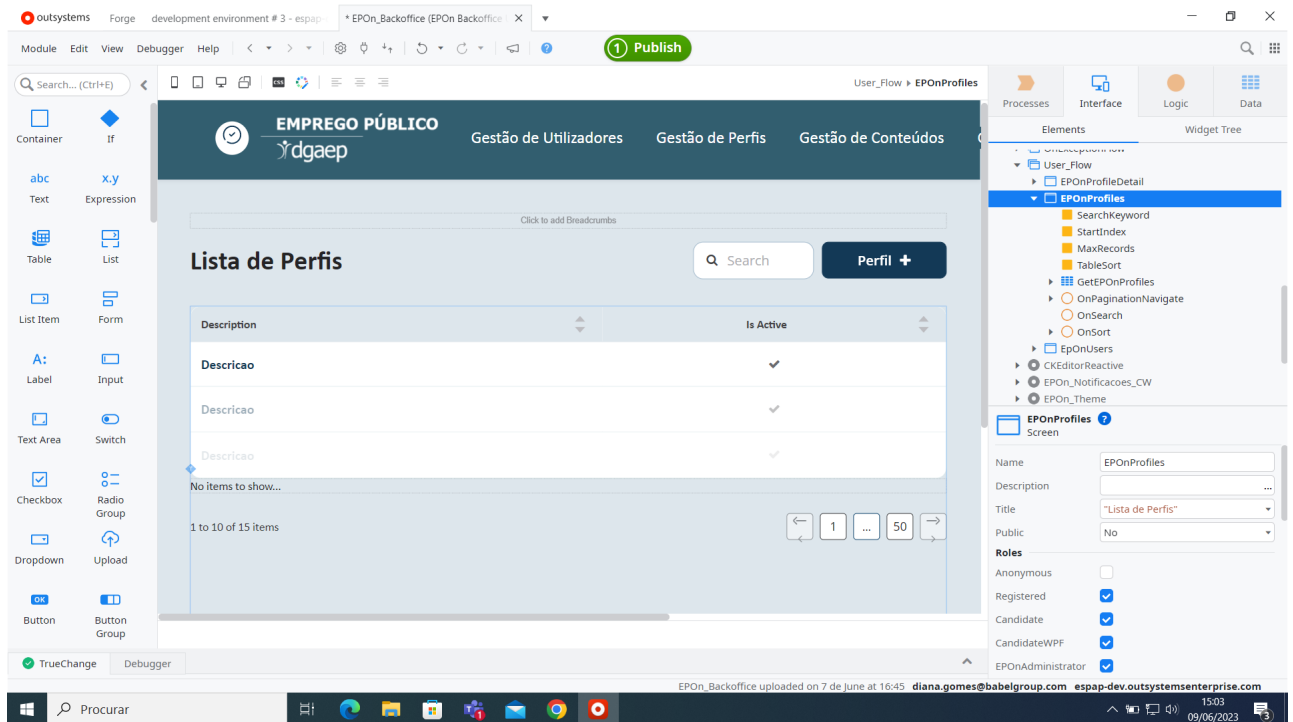


Figura 5.1: Desenvolvimento do ecrã "Lista de Perfis" no ambiente *Outsystems*.

A figura 5.1 mostra o ecrã que apresenta uma lista com todos os perfis registados na aplicação, um botão de pesquisa e um outro botão de criar/editar perfil. Na parte de *front-end*

deste ecrã o *layout* foi baseado no *mockup* já existente na aplicação *Figma* como já foi referido num dos capítulos anteriores. Na parte de *back-end* foi necessário criar uma base de dados com os atributos necessários, para alimentar a tabela que vai apresentar a descrição dos perfis existentes nela e se estão ou não ativos.

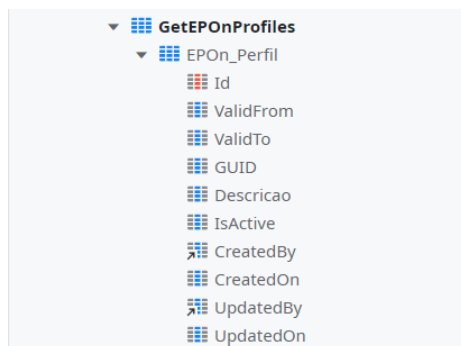


Figura 5.2: Base de dados e seus atributos referentes à "Lista de Perfis" no ambiente *Outsystems*.

Como podemos ver, na figura 5.2, este *Agreggate* tem vários atributos, entre os quais, o 'Id', que é a chave primária, a 'Descricao' que é a identificação de tipo de perfil e o 'IsActive' que vai ser o campo que nos informa se está ou não ainda ativo esse registo. A informação é mostrada na tabela da página do "Lista de Perfis". Logo depois, temos o botão de pesquisar, em que é usado essa base de dados para fazer a pesquisa do que escrito no *input* do botão de pesquisa. Essa pesquisa está a ser feita através da comparação daquilo que foi escrito no *input* do botão com o que existe na base de dados, através do atributo "Descricao". Já no botão "Perfil +", é associado um redirecionamento para outro ecrã. Esse ecrã será de criar ou editar perfis.

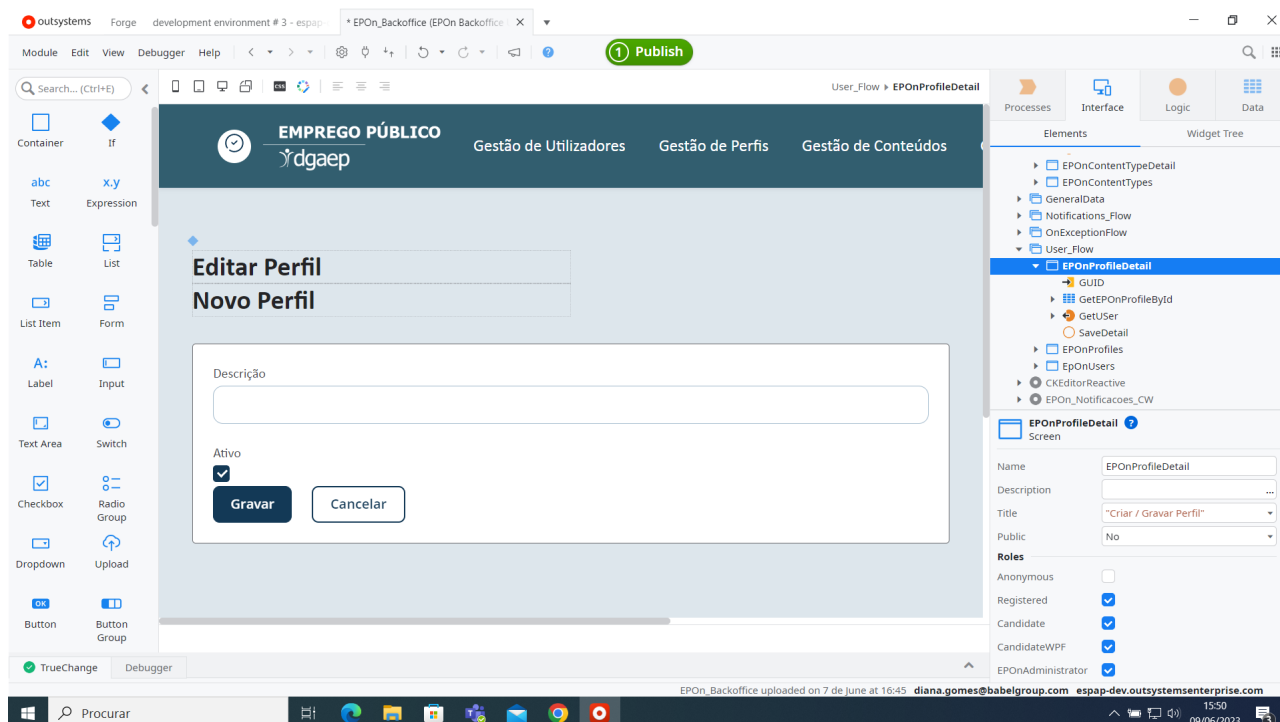


Figura 5.3: Desenvolvimento do ecrã "Criar/Editar Perfis" no ambiente *Outsystems*.

Na figura 5.3, para além de termos os mesmos atributos que no acrá anterior vamos ter 2

botões, um de gravar e outro de cancelar. No botão de gravar vamos ter uma lógica, que guarda o registo na base de dados.

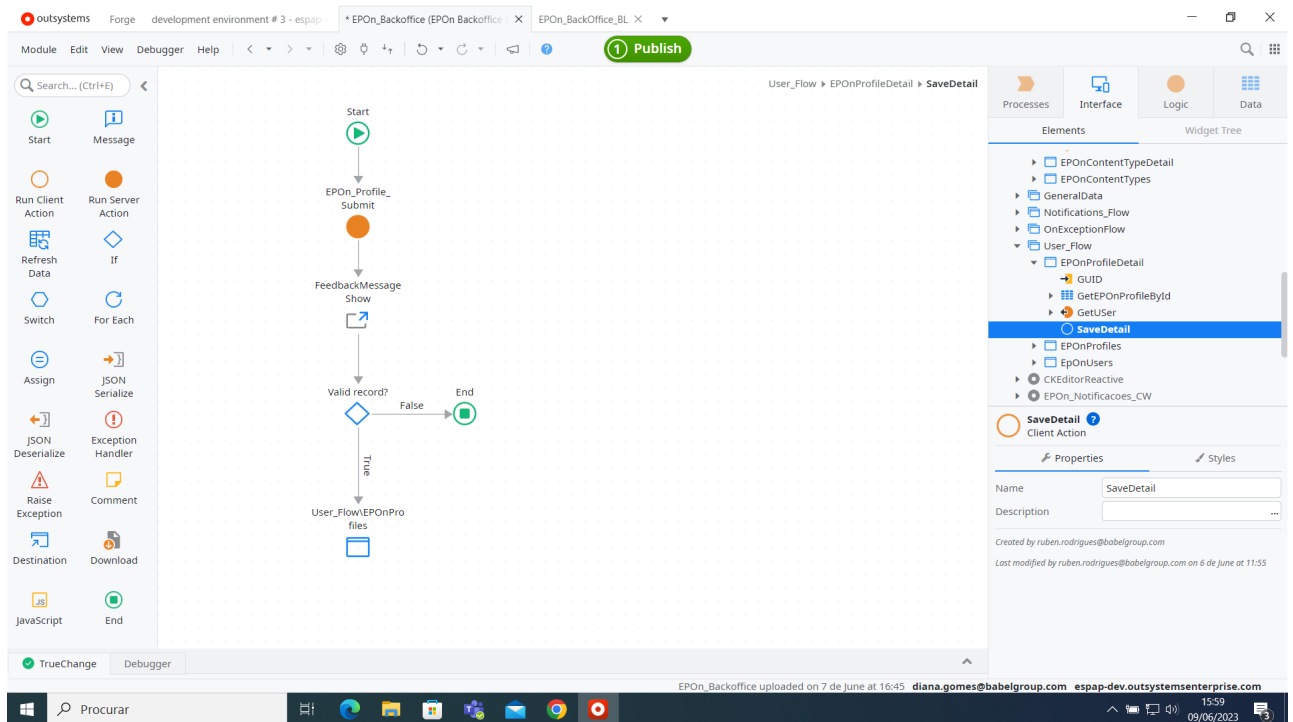


Figura 5.4: Lógica do botão "Guardar".

Nesta lógica, figura 5.4, temos uma ação, o "EpOnProfileSumit" que vai fazer ações de guardar e validar qualquer registo que tente ser submetido. Depois dessa validação, vamos ter um "FeedbackMessageShow", que será o aparecimento de uma mensagem em caso de sucesso com a informação de que o registo foi gravado. De seguida temos uma condição para saber se o registo é válido ou não. Em caso negativo a ação acaba e em caso afirmativo a ação retoma para o ecrã inicial "Lista de Perfis".

Este exemplo que foi dado é um exemplo simples mas que serviu para demonstrar parcialmente como ocorre o desenvolvimento de uma aplicação.

Capítulo 6

Conclusão

Neste documento refletiu-se todo o processo de aprendizagem durante este estágio inserido na área de *Outsystems* para integração numa equipa de desenvolvimento de um sistema. Atualmente, a tecnologia existe em quase tudo o que fazemos e em quase todos os produtos e serviços que consumimos e usamos, existindo uma necessidade constante de estarmos em contato com a informática. Neste mundo altamente tecnológico, a informática vem adquirindo cada vez mais relevância, tanto na vida das pessoas como nas empresas. Essa necessidade traduz-se em fatores fundamentais como o tempo de desenvolvimento e a qualidade das aplicações para o sucesso das empresas e a satisfação dos clientes. Surge disso, a plataforma *Outsystems*, que consegue construir aplicações com qualidade, eficiência, segurança, escalabilidade e adaptabilidade muito mais rápido, em comparação com as linguagens de programação mais convencionais. Esta plataforma facilita assim todo o processo de criação, manutenção e publicação de aplicações, *web* e *mobile*. Através da descrição anterior consegue-se perceber que uma aplicação em *OutSystems* não só é desenvolvida mais rapidamente que pelos processos tradicionais, como é bastante flexível a qualquer alteração que surja. Estes factores estão de acordo com as necessidades atuais das organizações que necessitam de adequar os seus sistemas o mais rapidamente possível às constantes alterações do mercado. Com isto não podia estar mais satisfeita em ter a oportunidade de trabalhar com algo que será o futuro pelo menos nos próximos anos.

O principal objetivo deste projeto era que fosse uma experiência enriquecedora tanto a nível profissional como pessoal e com o projeto Emprego Público *Online* isso cumpriu as minhas expectativas.

Bibliografia

- [1] Babel. [Online]. Available: <https://www.babelgroup.com/pt/> 2
- [2] Front-end vs back-end. [Online]. Available: <https://blog.anhanguera.com/o-que-e-front-end-e-back-end/> 7
- [3] Low-code vs traditional development. [Online]. Available: <https://www.monocubed.com/blog/low-code-vs-traditional-development/> 7
- [4] Javascript. [Online]. Available: <https://rockcontent.com/br/blog/javascript> 8
- [5] JQuery. [Online]. Available: <https://conceitos.com/jquery/> 8
- [6] L. Damas, *SQL*, ser. 14^a edição, 2007. [Online]. Available: <https://www.fnac.pt/SQL-Luis-Damas/a1241712> 9
- [7] .net. [Online]. Available: <https://dotnet.microsoft.com/pt-br/learn/dotnet/what-is-dotnet> 9
- [8] Ui/ux. [Online]. Available: <https://www.coursera.org/articles/ui-vs-ux-design> 9
- [9] Microsoft teams. [Online]. Available: <https://tecnoblog.net/responde/o-que-e-e-como-funciona-o-microsoft-teams/> 10
- [10] Microsoft azure devops. [Online]. Available: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-devops/> 10
- [11] Figma. [Online]. Available: <https://ebaonline.com.br/blog/o-que-e-figma-e-como-usar> 11
- [12] Outsystems. [Online]. Available: <https://www.outsystems.com/community/> 13
- [13] Scrum. [Online]. Available: <https://www.scrumportugal.pt/SCRUM/> 14
- [14] Kanban. [Online]. Available: <https://www.totvs.com/blog/negocios/kanban/> 14