

Contribuição para a Implementação do Projeto Transportes Públicos Flexíveis, Versão Final Após Defesa

Francisco Miguel Galvão Pina

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

Orientador: Prof. Doutor João Muranho
Orientador da Latitудde: Engenheiro João José Teles Gouveia

agosto de 2024

Declaração de Integridade

Eu, Francisco Miguel Galvão Pina, que abaixo assino, estudante com número de inscrição M13010 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 quaisquer das variedades de Fraude Académica, e que aqui declaro conhecer, e que em particular atendi à exigida referência de frases, extratos, imagens e outras formas de trabalho intelectual, e assim assumo na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 02/08/2024.

Francisco Miguel Galvão Pina

Agradecimentos

Muitas pessoas marcam-nos para a vida; umas ajudam-nos a construir quem somos e outras desafiam-nos a concretizar os nossos sonhos. Trilhar este caminho não foi fácil e só foi possível com o auxílio, conselhos e amizade de várias pessoas.

Gostaria de deixar um grande bem-haja ao meu orientador, Professor Doutor João Muranho, por todo o apoio, disponibilidade e partilha de conhecimentos. As suas opiniões e ensinamentos contribuíram decisivamente para levar a “bom porto” a conclusão deste relatório.

Quero também expressar a minha gratidão ao meu co-orientador, Eng. João Gouveia, cuja orientação e apoio foram fundamentais durante o estágio. O seu conhecimento e experiência foram inestimáveis para o meu desenvolvimento profissional e pessoal.

Agradeço profundamente às pessoas que fizeram parte da minha equipa durante o estágio. A colaboração e o espírito de equipa que partilhámos foram cruciais para o sucesso do nosso projeto. O vosso suporte e dedicação foram, por mim, muito apreciados.

Seguidamente, quero também agradecer aos meus colegas de curso e amigos mais próximos, pelo apoio, solidariedade e companheirismo demonstrados.

Por último, mas não menos importante, um agradecimento à minha família, especialmente aos meus pais e irmão, pelas suas palavras de motivação nos momentos mais difíceis. Eles foram, desde o primeiro instante, o meu suporte para chegar até aqui. Sem o seu apoio incondicional, carinho, força e amor, não seria possível alcançar este objetivo.

A todos, muito obrigado.

Resumo

Este relatório visa documentar todo o processo de estágio realizado na unidade curricular de Estágio, referente ao segundo semestre do segundo ano do Mestrado em Engenharia Informática na Universidade da Beira Interior. O documento detalha as diversas fases do estágio, incluindo as atividades propostas e realizadas, bem como as tecnologias e ferramentas utilizadas.

O estágio decorreu na empresa Latitudde - Digital Enablers, LDA, e envolveu a minha contribuição para a implementação do Projeto Transportes Públicos Flexíveis. Este projeto abrangeu, desde a formação inicial, planeada pela empresa, pontos necessários para a aprendizagem de várias tecnologias, até à integração numa equipa de desenvolvimento dedicada à implementação de soluções tecnológicas para o referido projeto.

Palavras-chave

Angular, .NET, TypeScript, SQL

Abstract

This report aims to document the entire internship process undertaken as part of the Internship course, during the second semester of the second year of the Master's degree in Computer Engineering at the University of Beira Interior. The document details the various phases of the internship, including the proposed and completed activities, as well as the technologies and tools employed.

The internship took place at Latitude - Digital Enablers, LDA, and involved contributing to the implementation of the Flexible Public Transport Project. This project encompassed initial training organized by the company, necessary for learning various technologies, and the subsequent integration into a development team dedicated to implementing technological solutions for the project.

Keywords

Angular, .NET, TypeScript, SQL

Índice

Agradecimentos	iv
Resumo	v
Abstract	vii
Índice	ix
Lista de Figuras	xi
Lista de Acrônimos	xiii
1 Introdução	1
1.1 Motivação e Expectativas	1
1.2 Principais Desafios	2
1.3 Objetivos	3
2 Empresa	5
2.1 Introdução	5
2.2 História da Empresa	5
2.3 Missão e Valores	6
2.4 Formação	6
2.5 Espaço Físico	7
3 Planeamento do Estágio	9
3.1 Introdução	9
3.2 Cronograma	9
3.3 Etapas do Planeamento	10
4 Tecnologias Utilizadas	11
4.1 Vertentes de Desenvolvimento	11

4.2	Linguagens	11
4.2.1	(HTML)	11
4.2.2	<i>Cascading Style Sheets (CSS)</i>	12
4.2.3	C#	13
4.2.4	<i>JavaScript/TypeScript</i>	14
4.2.5	<i>Structured Query Language (SQL)</i>	15
4.3	<i>Frameworks</i>	16
4.3.1	<i>Angular</i>	16
4.3.2	<i>.NET</i>	17
4.3.3	<i>SCRUM</i>	18
4.4	Ferramentas	19
4.4.1	<i>Microsoft Teams</i>	19
4.4.2	<i>GitLab</i>	20
4.4.3	<i>GitHub Desktop</i>	21
4.4.4	<i>Figma</i>	22
5	Projeto Final	25
5.1	Introdução	25
5.2	Objetivos	25
5.3	Metodologia	26
5.4	Equipa	26
5.5	Trabalho Desenvolvido	27
6	Conclusão	31
6.1	Conclusões Principais	31
6.2	Trabalho Futuro	31
6.3	Palavras Finais	31
	Bibliografia	33

Lista de Figuras

2.1	Logótipo Latitudde	5
2.2	Espaço físico da Latitudde	8
4.1	Logótipo Angular	17
4.2	Logótipo .NET	18
4.3	Logótipo Microsoft Teams	20
4.4	Logótipo GitHub Desktop	21
5.1	Modelo de Dados.	27
5.2	Privilégios de cada utilizador (exemplo, tabela não completa).	28
5.3	Endpoints associados aos utilizadores	28
5.4	Nesta figura podemos observar a página utilizadores do <i>backoffice</i> da aplicação. Cada utilizador pode ter um ou mais <i>roles</i> , podemos também obter detalhes relevantes sobre cada utilizador, como o contacto, Email, Morada e Nome. Temos uma caixa de pesquisa, onde podemos pesquisar por palavras até encontrarmos o utilizador pretendido. Existe também um botão de filtro para podermos filtrar os utilizadores por <i>role</i> . Por fim, temos um botão para criar um novo utilizador que irá abrir um formulário como podem ver na figura 5.5. Ao clicarmos em cima de um utilizador uma janela será aberta (ver figura 5.6) onde serão mostradas mais informações, assim como, dois botões para editar e eliminar o utilizador selecionado.	29
5.5	Nesta figura podemos observar o formulário de criação de um novo utilizador. O formulário encontra-se já completo e como podemos observar há certas palavras que têm um asterisco à frente, isto significa que o botão de concluído só é clicável se o utilizador completar esses campos, são campos obrigatórios, caso contrário não será possível concluir a criação do novo utilizador.	30
5.6	Nesta figura podemos ver mais informações relativas ao utilizador selecionado, assim como o seu histórico de pedidos. É possível editar o utilizador e eliminá-lo. É importante referir que este eliminar é um <i>Soft Delete</i> , isto é, o utilizador não é eliminado totalmente da base de dados, apenas fica inativo.	30

Acrónimos

CSS	<i>Cascading Style Sheets</i>
DCL	<i>Data Control Language</i>
DDL	<i>Data Definition Language</i>
DML	Data Manipulation Language
DQL	<i>Data Query Language</i>
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
SQL	<i>Structured Query Language</i>
UBI	Universidade da Beira Interior
UI	<i>User Interface</i>
UX	<i>User Experience</i>
XML	Extensible Markup Language

Capítulo 1

Introdução

1.1 Motivação e Expectativas

No último ano do Mestrado em Engenharia Informática na Universidade da Beira Interior (UBI), surge a oportunidade crucial de escolher entre realizar uma dissertação ou um estágio profissional. Enquanto a dissertação permite uma investigação aprofundada numa área específica, o estágio oferece uma aplicação prática dos conhecimentos adquiridos, proporcionando uma imersão direta no mercado de trabalho. Para mim, a decisão de optar pelo estágio foi clara e motivada por várias razões significativas.

Desde o início dos meus estudos, sempre procurei uma forma de combinar a teoria com a prática, valorizando a aplicação real dos conceitos aprendidos. A escolha pelo estágio na Latitude - Digital Enablers, LDA, no qual contribuí para a implementação do Projeto de Transportes Públicos Flexíveis no Município de Castelo Branco, alinou-se perfeitamente com este objetivo. Este projeto não só exigia uma forte base técnica, mas também oferecia a oportunidade de impactar positivamente a comunidade, um aspeto que considero extremamente gratificante.

A minha principal motivação para este estágio foi a oportunidade de trabalhar numa área de grande relevância tecnológica. O desenvolvimento de uma plataforma *web* para a gestão de serviços de transporte público flexível é um desafio que combina a aplicação tecnológica com a necessidade de soluções práticas e eficientes para problemas reais. Este projeto permitiu-me explorar e aprofundar conhecimentos em tecnologias modernas, como *.NET* para o *back-end* e *Angular* para o *front-end*, ampliando assim as minhas competências técnicas e profissionais.

Além disso, entrar num ambiente profissional dinâmico como o da Latitude - Digital Enablers, LDA, proporcionou-me uma visão prática e abrangente das exigências e expectativas do mercado de trabalho atual. Trabalhar numa equipa dedicada e enfrentar desafios diários contribuiu significativamente para o meu crescimento pessoal e profissional, ensinando-me a importância da colaboração, comunicação e adaptabilidade.

As minhas expectativas para este estágio eram elevadas, mas também realistas. Esperava adquirir não só competências técnicas, mas também desenvolver *soft skills* essenciais, como a capacidade de trabalhar em equipa, gestão de tempo e resolução de problemas. A experiência superou as minhas expectativas, oferecendo-me oportunidades de aprendizagem e crescimento contínuo.

Em resumo, o estágio representou uma fase de transição entre o ambiente acadêmico e o mundo profissional, permitindo-me aplicar, testar e expandir os meus conhecimentos num contexto real e dinâmico. Esta experiência foi, sem dúvida, um passo importante para a minha carreira, preparando-me para futuros desafios e oportunidades na área da Engenharia Informática.

1.2 Principais Desafios

Ao longo do estágio na Latitude - Digital Enablers, LDA, enfrentei diversos desafios que foram cruciais para o meu desenvolvimento como engenheiro informático. Cada desafio representou uma oportunidade de aprendizagem e crescimento, tanto a nível técnico como pessoal.

Um dos principais desafios foi a adaptação a novas tecnologias e *frameworks*. Trabalhar com *.NET* no *back-end* e *Angular* no *front-end* exigiu uma aprendizagem intensiva e rápida, pois estas ferramentas eram novas para mim. A necessidade de dominar estas tecnologias num curto espaço de tempo foi exigente, mas extremamente enriquecedora. Através de pesquisa contínua, prática e orientação por parte dos colegas mais experientes, consegui adquirir as competências necessárias para contribuir eficazmente para o projeto.

Outro desafio significativo foi a integração numa equipa de desenvolvimento já estabelecida. Adaptar-me aos métodos de trabalho, processos de desenvolvimento e dinâmicas de equipa foi essencial para garantir uma colaboração eficiente. Aprendi a importância da comunicação clara e frequente, bem como a necessidade de ser proativo na resolução de problemas e na partilha de ideias. Este ambiente colaborativo foi fundamental para a minha adaptação e sucesso no projeto.

Além disso, a implementação do Projeto de Transportes Públicos Flexíveis envolveu a resolução de problemas complexos relacionados com a gestão de dados e a otimização de processos. Desenvolver uma plataforma que pudesse lidar eficientemente com a variabilidade e flexibilidade dos serviços de transporte público exigiu uma abordagem criativa e analítica. Trabalhei na identificação de requisitos, modelagem de dados e desenvolvimento de funcionalidades que assegurassem a robustez e escalabilidade da plataforma.

A gestão de tempo e prioridades foi outro desafio constante. Conciliar as tarefas do estágio com outras responsabilidades académicas e pessoais exigiu uma organização rigorosa e uma capacidade de adaptação constante. Aprendi a definir prioridades, a gerir prazos e a manter um equilíbrio saudável entre trabalho e vida pessoal.

Por fim, a necessidade de garantir a qualidade e a segurança da plataforma desenvolvida foi um desafio importante. Implementar práticas de desenvolvimento seguro, realizar testes exaustivos e assegurar a conformidade com as melhores práticas de segurança foram aspetos cruciais do projeto. Esta experiência aprofundou o meu entendimento sobre a

importância da segurança em aplicações *web* e a necessidade de práticas de desenvolvimento rigorosas.

Em resumo, os desafios enfrentados durante o estágio foram diversos e exigentes, mas cada um deles contribuiu significativamente para o meu desenvolvimento como profissional. Estas experiências prepararam-me melhor para futuros desafios na minha carreira e reforçaram a minha paixão pela Engenharia Informática.

1.3 Objetivos

O estágio na Latitude - Digital Enablers, LDA, foi uma oportunidade valiosa para definir e alcançar uma série de objetivos pessoais e profissionais. Estabeleci metas claras desde o início do estágio, alinhadas com as necessidades do projeto e as expectativas da empresa.

Em primeiro lugar, o principal objetivo era contribuir de forma significativa para a implementação bem-sucedida do Projeto de Transportes Públicos Flexíveis. Isso envolveu o desenvolvimento de uma plataforma *web* funcional e robusta, capaz de gerir eficientemente os serviços de transporte público na região de Castelo Branco. O meu objetivo era não apenas cumprir as tarefas atribuídas, mas também encontrar soluções inovadoras e eficazes para os desafios específicos do projeto.

Além disso, o estágio proporcionou-me uma oportunidade para expandir e aprofundar os meus conhecimentos em tecnologias modernas de desenvolvimento de software, como *.NET* e *Angular*. O objetivo era adquirir competências sólidas nestas áreas, aplicando as melhores práticas de engenharia de software e garantindo a qualidade e a escalabilidade da solução desenvolvida.

Outro objetivo importante era desenvolver e aprimorar habilidades profissionais essenciais, como trabalho em equipa, comunicação eficaz e resolução de problemas. A integração numa equipa de desenvolvimento dinâmica e colaborativa proporcionou-me uma oportunidade valiosa para aprender com os colegas mais experientes, compartilhar conhecimentos e contribuir para um ambiente de trabalho positivo e produtivo.

Além dos objetivos técnicos e profissionais, também tinha objetivos pessoais durante o estágio. Queria desafiar-me a sair da minha zona de conforto, assumir responsabilidades e enfrentar novos desafios com confiança e determinação. Também aspirei em manter um equilíbrio saudável entre trabalho e vida pessoal, priorizando o bem-estar físico e mental durante este período intenso.

Em resumo, os objetivos do estágio foram multifacetados, abrangendo tanto aspetos técnicos como pessoais. Estabelecer metas claras e específicas desde o início foi essencial para orientar o meu trabalho e garantir um progresso consistente ao longo do estágio.

Capítulo 2

Empresa

2.1 Introdução

A Latitudde[Lat24] - Digital Enablers (logótipo na Figura 2.1) é uma subsidiária do grupo RIT, uma empresa multinacional especializada em projetos de transformação digital. Com um corpo profissional de mais de 400 colaboradores e escritórios distribuídos por quatro países (Portugal, Chile, Peru e Nova Zelândia), a Latitudde - Digital Enablers opera em projetos internacionais em várias partes do globo.



Figura 2.1: Logótipo Latitudde

Os especialistas da Latitudde - Digital Enablers primam pela excelência, seguindo padrões de alta qualidade resultantes da formação ministrada pelos Centros de Excelência do grupo.

A sede desta empresa do grupo localiza-se na Rua Cidade da Covilhã, 3º, Casa da Moagem, 6230-346 Fundão.

2.2 História da Empresa

A Latitudde - Digital Enablers é uma empresa recente, tendo sido constituída em janeiro de 2021. Integrando-se no grupo RIT, presente na região da Beira Interior desde 2015, a Latitudde - Digital Enablers oferece duas principais divisões: Engenharia de Software e *Low-Code*.

Na divisão de Engenharia de Software, os projetos são abordados de várias maneiras, incluindo desenvolvimento de *software*, adaptação de *software* e garantia de qualidade. Para atender às necessidades específicas de cada projeto, a divisão é subdividida em várias equipas.

A divisão de *Low-Code* também é organizada para satisfazer as necessidades específicas de cada projeto, utilizando tecnologias específicas, como *Appian* e *OutSystems*.

O objetivo principal da Latitudde - Digital Enablers é integrar-se plenamente no mundo empresarial, adaptando-se a novas formas de trabalho e enfrentando desafios constantes. A empresa procura sempre utilizar a tecnologia mais recente disponível, alinhada com as expectativas e objetivos dos seus clientes.

2.3 Missão e Valores

A missão da Latitudde é destacar-se como um exemplo de excelência na área da engenharia de software, capacitando os seus parceiros para forjar um horizonte tecnológico próspero. Sob o lema "*Digital Enablers*", a empresa almeja ser a principal escolha confiável para seus parceiros, fornecendo uma ampla gama de opções para ajudá-los a alcançar os seus objetivos.

A Latitudde busca estabelecer parcerias sólidas e duradouras, sendo reconhecida como uma empresa confiável e inovadora em soluções tecnológicas. Através do comprometimento com a excelência, dedicação constante, responsabilidade e consideração, a Latitudde visa promover o sucesso dos seus parceiros, contribuindo para um horizonte tecnológico próspero.

A empresa cultiva uma cultura robusta de aprendizagem e desenvolvimento em equipa, onde a colaboração é valorizada. Esse ambiente é propício para que indivíduos em estágios iniciais possam dar os primeiros passos e crescer profissionalmente, como foi o meu caso.

A Latitudde preza pela excelência em cada projeto realizado, procurando superar expectativas e entregar soluções de qualidade superior. O comprometimento é uma característica essencial, evidenciando a dedicação da equipe em alcançar os objetivos estabelecidos. A responsabilidade é tratada com seriedade, garantindo a entrega dos projetos dentro dos prazos e com eficácia. Além disso, a Latitudde promove um ambiente de respeito, onde cada integrante da equipa é valorizado e as suas contribuições são reconhecidas.

Uma das minhas convicções pessoais foi algo que pude constatar na Latitudde. Acredito que ajudar os outros a crescer também impulsiona nosso próprio desenvolvimento, contribuindo para o progresso e beneficiando não apenas o indivíduo, mas a coletividade como um todo.

2.4 Formação

A empresa implementou um abrangente plano de formação para os novos membros da equipa, conhecido como *7 Summits*. Este plano, inspirado na conquista dos 7 maiores picos do mundo, destina-se a abordar áreas específicas da programação, cada uma repre-

sentada por um "cume" diferente.

O *7 Summits* foi meticulosamente elaborado para garantir uma integração gradual e bem-sucedida dos novos membros na equipa. Abrangendo uma vasta gama de tecnologias, o plano oferece caminhos personalizados, adaptados às habilidades e interesses individuais. Para os interessados em Engenharia de Software, há a oportunidade de se especializar em *front-end*, *back-end* ou ambos, através da formação em *Full-Stack*. Além disso, há a possibilidade de seguir a formação em *Low-Code*, concentrando-se em tecnologias como *Appian* e *OutSystems*.

Este programa de formação proporciona uma oportunidade única para os novos membros adquirirem habilidades e conhecimentos valiosos, essenciais para o seu sucesso e crescimento na empresa. A Latitudde compromete-se a apoiar o desenvolvimento profissional dos seus colaboradores, especialmente durante as primeiras fases dessa jornada.

Além do programa de formação, todos os novos recrutados recebem um *kit* de boas-vindas, que inclui um computador portátil para facilitar o trabalho, uma mochila para transporte, uma garrafa de água de vidro e materiais de escrita para fazer anotações necessárias. Essas iniciativas demonstram o compromisso da Latitudde em criar um ambiente de trabalho produtivo e acolhedor para os seus colaboradores.

2.5 Espaço Físico

Durante o meu estágio na Latitudde, tive o privilégio de frequentar o escritório localizado no Fundão, conforme ilustrado na figura 2.2. Embora tivesse a opção de frequentar outros locais pelo país, optei por este devido à sua proximidade geográfica com a UBI.

O espaço físico está situado na Casa da Moagem, onde também estão localizadas outras duas empresas do grupo, RedinessIt e Ritain. O escritório da Latitudde está localizado no piso térreo e é um espaço de trabalho aberto. Essa disposição apresenta vantagens, pois cria um ambiente comunitário onde colegas estão sempre disponíveis para oferecer ajuda em caso de dúvidas ou bloqueios.



Figura 2.2: Espaço físico da Latitudde

Capítulo 3

Planeamento do Estágio

3.1 Introdução

No período anterior ao início do estágio, foi elaborado pela empresa um planeamento para garantir a máxima organização e eficácia durante todo o processo.

Embora já possuísse uma base sólida em conceitos de Engenharia de Software, adquirida ao longo do meu percurso académico em Engenharia Informática, reconheci a necessidade de uma adaptação e aprofundamento desses conhecimentos, bem como a familiarização com novos conceitos e métodos de trabalho.

Portanto, o estágio foi estruturado de forma a incluir momentos de formação técnica e metodológica e períodos dedicados à aplicação prática desses conhecimentos em projetos concretos. Essa abordagem que permite proporcionar uma experiência abrangente e enriquecedora, que permita não só a consolidação dos conhecimentos teóricos, mas também o desenvolvimento de competências práticas essenciais para o exercício da profissão.

3.2 Cronograma

No período anterior ao estágio, foi estabelecido um cronograma detalhado para orientar as atividades ao longo do período correspondente ao 2º semestre, com uma duração determinada entre 5 de fevereiro de 2024 e 12 de junho de 2024.

Os horários seguidos foram os padrões da empresa, com uma jornada de 8 horas diárias, das 09:00h às 18:00h.

O estágio foi dividido em duas fases.

Durante a fase inicial, foi priorizada a adaptação às novas tecnologias e metodologias de trabalho. Isso foi alcançado através da participação em diversos cursos, formações e obtenção de certificações pré-determinadas, realizadas principalmente na plataforma *Udemy*.

Na fase de integração na equipa de projeto interno, o objetivo foi a inserção numa equipa de projeto da empresa. Nesta fase, os estagiários, incluindo eu, foram integrados num projeto real, onde puderam aplicar os conhecimentos adquiridos e contribuir de forma significativa para o cumprimento do plano estabelecido.

Valores como responsabilidade, rigor e compromisso foram fundamentais para o sucesso desses projetos, bem como a capacidade de estabelecer boas relações e interagir de forma eficaz com os outros membros da equipa.

3.3 Etapas do Planeamento

O planeamento do estágio foi dividido em duas etapas, cada uma com os seus objetivos e focos específicos.

Na primeira etapa do planeamento, os estagiários, onde eu me incluo, dedicaram-se à realização de cursos na plataforma *Udemy*, abrangendo diversas áreas de tecnologia relevantes para o estágio. Estes cursos incluíram *Angular*, *.NET*, *HTML* e bases de dados relacionais. O principal objetivo desta etapa foi criar uma base sólida de conhecimento, proporcionando aos estagiários uma formação abrangente em tecnologias essenciais para o desenvolvimento de software. Além disso, procurou-se estabelecer um equilíbrio entre as áreas de *front-end* e *back-end*, garantindo que os estagiários adquirissem competências tanto em desenvolvimento de interfaces de utilizador como em lógica de programação e manipulação de dados.

Já na segunda etapa do planeamento, o foco foi o desenvolvimento da Plataforma Tecnológica para a Gestão dos Serviços de Transporte Público Flexível no Município de Castelo Branco. Este projeto foi concebido e iniciado do zero pelos membros que ingressaram na academia *7 Summits*. Nesta etapa, apliquei os conhecimentos adquiridos nos cursos anteriores para projetar, desenvolver e implementar uma solução tecnológica inovadora. O projeto envolveu uma série de desafios técnicos e práticos, incluindo a integração de diferentes tecnologias, a análise de requisitos, o desenvolvimento de funcionalidades e a realização de testes.

Capítulo 4

Tecnologias Utilizadas

4.1 Vertentes de Desenvolvimento

No desenvolvimento de software, é necessário utilizar uma variedade de tecnologias, uma vez que várias camadas compõem a arquitetura de software, todas elas fundamentais para a construção de aplicações.

A camada de *back-end* [Wik24a], por exemplo, é responsável pela lógica do negócio e pelas operações relacionadas com bases de dados. Aqui, são comuns linguagens como *PHP*, *Ruby*, *Java*, *Python* e *C#*, que recebem os pedidos do *front-end*, processam-nos e devolvem as respostas que serão apresentadas ao utilizador final.

Já o *front-end* [Wik24b] é a parte visível da aplicação ao utilizador final, onde se apresenta a interface gráfica e se processam as interações do utilizador. Nesta área, linguagens como *HTML*, *CSS*, *JavaScript* e *Angular* são frequentemente utilizadas.

A abordagem *full-stack*, por sua vez, envolve tanto o *back-end* quanto o *front-end*, permitindo ao programador desenvolver aplicações completas sem depender de equipas separadas. Essa versatilidade permite assumir a responsabilidade por todas as fases da aplicação.

No estágio em questão, considerado *full-stack*, é crucial abordar tecnologias de ambas as áreas. Por isso, foram seleccionadas duas tecnologias *open-source* atuais, com amplo reconhecimento e suporte da comunidade: *Angular* para o *front-end* e *.NET* para o *back-end*.

4.2 Linguagens

4.2.1 HTML

HTML [Wik24c] é a linguagem fundamental para a criação e estruturação de páginas *web*. Esta linguagem de marcação define a estrutura básica de um website, permitindo a organização de conteúdos como textos, imagens, links e outros elementos multimédia.

A principal função do *HTML* é marcar o conteúdo de uma página *web* através de uma série de *tags*. Cada *tag* tem um propósito específico. Por exemplo, a *tag* `<h1>` é usada para definir um título principal, enquanto a *tag* `<p>` é utilizada para parágrafos. Estas *tags*

não só organizam o conteúdo mas também o tornam acessível aos motores de pesquisa, facilitando a indexação e a recuperação de informação.

Uma das grandes vantagens do HTML é a sua simplicidade e facilidade de aprendizagem. Mesmo para quem está a iniciar-se no desenvolvimento *web*, o HTML é uma linguagem acessível e intuitiva. Além disso, é suportada por todos os navegadores, garantindo que as páginas criadas sejam visualizadas corretamente em qualquer dispositivo.

O HTML não atua sozinho. Normalmente, é complementado por CSS e *JavaScript*. O CSS é responsável pela apresentação visual, permitindo estilizar os elementos HTML com cores, margens, fontes, entre outros. Já o *JavaScript* adiciona interatividade, permitindo a criação de funcionalidades dinâmicas que melhoram a experiência do utilizador.

Nos últimos anos, o HTML evoluiu consideravelmente. A versão mais recente, *HTML5*, trouxe melhorias significativas, incluindo novas *tags* e atributos que permitem maior flexibilidade e capacidade. Com *HTML5*, é possível incorporar diretamente elementos multimédia como vídeo e áudio, sem necessidade de *plugins* adicionais, o que simplifica o desenvolvimento e enriquece a experiência do utilizador.

Em suma, o HTML é uma ferramenta essencial para qualquer desenvolvedor *web*. É a base sobre a qual se constroem as páginas e aplicações *web*, oferecendo a estrutura necessária para criar conteúdos acessíveis e organizados. Para um estudante de Engenharia Informática, dominar o HTML é o primeiro passo para explorar o vasto universo do desenvolvimento *web*.

4.2.2 CSS

CSS [w3s24b] é uma linguagem de estilos utilizada para descrever a apresentação de documentos HTML ou Extensible Markup Language (XML). Através do CSS, é possível controlar a aparência visual dos elementos de uma página *web*, separando o conteúdo (HTML) do *design*. Isso permite uma maior flexibilidade e um design mais consistente e atraente.

A principal função do CSS é definir estilos para elementos HTML. Esses estilos incluem propriedades como cores, fontes, margens, espaçamentos, alinhamentos, e muitas outras características visuais. Por exemplo, com CSS, é possível definir que todos os títulos `<h1>` de uma página devem ser exibidos em uma determinada cor e tamanho de fonte, ou que os parágrafos `<p>` devem ter um certo espaçamento entre si.

Uma das grandes vantagens do CSS é a sua capacidade de aplicar estilos de maneira eficiente a múltiplas páginas *web*. Ao utilizar folhas de estilo externas, é possível definir um conjunto de regras CSS por um único ficheiro e aplicá-las a todas as páginas de um site. Isso não só facilita a manutenção e atualização do *design*, como também assegura uma aparência uniforme em todo o site.

CSS também permite a criação de *layouts* complexos. Através de técnicas como *Flexbox* e

CSS *Grid*, é possível organizar os elementos de uma página de forma precisa e responsiva, adaptando-se a diferentes tamanhos de ecrã e dispositivos. Estas técnicas são essenciais no design moderno, onde a experiência do utilizador deve ser consistente em *smartphones*, *tablets* e computadores.

Além das propriedades básicas, o CSS inclui funcionalidades avançadas como animações e transições, que adicionam interatividade e dinamismo às páginas *web*. Por exemplo, é possível criar efeitos, onde um elemento muda de aparência ao ser apontado pelo cursor, ou animações que alteram gradualmente as propriedades de um elemento ao longo do tempo.

O *CSS3*, a versão mais recente, introduziu várias novas funcionalidades e propriedades que ampliam as capacidades do CSS. Com *CSS3*, é possível criar efeitos visuais sofisticados, como sombras, gradientes, e transformações 3D, sem a necessidade de gráficos ou *scripts* adicionais.

Em resumo, o CSS é uma ferramenta fundamental para o *design web*. Ele permite separar a estrutura do conteúdo do design visual, proporcionando flexibilidade, consistência e eficiência na criação de páginas *web*. Para um estudante de Engenharia Informática, dominar o CSS é essencial para desenvolver *websites* modernos e visualmente apelativos.

4.2.3 C#

C# [w3s24a] (pronunciado "C-sharp") é uma linguagem de programação moderna, orientada a objetos, desenvolvida pela *Microsoft* como parte da sua plataforma *.NET*. Desde a sua criação, C# tem sido amplamente utilizada para desenvolver uma vasta gama de aplicações, desde software *desktop* e aplicações *web* até jogos e serviços empresariais.

Uma das principais características do C# é a sua sintaxe clara e concisa, que é semelhante a outras linguagens populares como *C++* e *Java*, tornando-a acessível para programadores familiarizados com essas linguagens. Além disso, C# incorpora muitas funcionalidades avançadas que ajudam a simplificar o desenvolvimento e aumentar a produtividade, como a gestão automática de memória, tipos de dados seguros e a possibilidade de usar programação assíncrona para melhorar o desempenho das aplicações.

A plataforma *.NET*, com a qual C# está intimamente integrada, oferece um vasto conjunto de bibliotecas e ferramentas que facilitam o desenvolvimento de aplicações robustas e escaláveis. Através do *.NET*, os programadores C# têm acesso a uma rica biblioteca de classes que fornecem funcionalidades prontas para uso, como manipulação de ficheiros, acesso a bases de dados, desenvolvimento de interfaces gráficas de utilizador e comunicação em rede.

Outra vantagem significativa do C# é a sua capacidade de ser usada para desenvolver aplicações multi-plataforma. Com a introdução do *.NET Core* (agora conhecido como *.NET*

5 e versões subsequentes), os desenvolvedores podem criar aplicações que funcionam em *Windows*, *macOS* e *Linux*, ampliando o alcance e a flexibilidade dos seus projetos.

C# também é amplamente utilizado no desenvolvimento de jogos, graças ao *Unity*, um dos motores de jogo mais populares do mundo. *Unity* usa C# como sua linguagem de *script* principal, permitindo que os desenvolvedores criem jogos 2D, 3D e realidade virtual para uma variedade de plataformas, incluindo PC, consolas e dispositivos móveis.

Além disso, a linguagem C# suporta a programação assíncrona através da utilização de palavras-chave como *async* e *await*. Este recurso é particularmente útil no desenvolvimento de aplicações que dependem de operações de entrada/saída, como chamadas a *APIs web* ou acesso a bases de dados, pois permite que essas operações sejam realizadas de forma não bloqueante, melhorando a responsividade e o desempenho da aplicação.

Em termos de segurança e robustez, o C# oferece vários mecanismos para prevenir erros comuns de programação. A linguagem inclui suporte para exceções e gestão de erros, tipos de dados seguros que ajudam a evitar problemas como *null references*, e mecanismos de controlo de acesso que permitem definir a visibilidade de classes e membros.

Em resumo, C# é uma linguagem de programação poderosa e versátil que oferece uma ampla gama de funcionalidades para o desenvolvimento de aplicações modernas e eficientes. Para estudantes de Engenharia Informática, dominar C# e a plataforma *.NET* abre muitas oportunidades no mercado de trabalho, desde o desenvolvimento de aplicações empresariais até à criação de jogos inovadores.

4.2.4 **JavaScript/TypeScript**

JavaScript[Moz24] é uma das linguagens de programação mais amplamente utilizadas no desenvolvimento *web*, essencial para a criação de páginas *web* interativas e dinâmicas. Originalmente desenvolvido para adicionar interatividade aos *websites*, *JavaScript* evoluiu significativamente ao longo dos anos e, atualmente, é uma linguagem completa que pode ser utilizada tanto no lado do cliente (*front-end*) quanto no lado do servidor (*back-end*).

Uma das principais características do *JavaScript* é a sua capacidade de manipular o Documento de Objetos de uma página *web*, permitindo que os desenvolvedores alterem o conteúdo e o estilo das páginas em resposta às ações do utilizador. Além disso, *JavaScript* suporta a programação assíncrona através do uso de *callbacks*, promessas (*Promises*) e, mais recentemente, *async/await*, facilitando a execução de operações não bloqueantes, como chamadas a *APIs web*.

TypeScript é um superconjunto de *JavaScript* que adiciona tipagem estática opcional à linguagem. Desenvolvido pela *Microsoft*, *TypeScript* foi criado para resolver alguns dos problemas associados ao desenvolvimento de grandes aplicações *JavaScript*, como a falta

de tipagem e o suporte limitado para desenvolvimento orientado a objetos.

Com *TypeScript*, os desenvolvedores podem definir tipos para variáveis, funções e objetos, o que ajuda a prevenir muitos erros comuns que só seriam detetados em tempo de execução no *JavaScript*. A tipagem estática também facilita a leitura e manutenção do código, tornando-o mais compreensível e menos propenso a erros. Além disso, o suporte para classes, interfaces e herança em *TypeScript* permite uma melhor organização do código e a implementação de padrões de design orientados a objetos.

Outra vantagem significativa de *TypeScript* é a sua compatibilidade total com *JavaScript*. Qualquer código *JavaScript* válido é também código *TypeScript* válido, o que facilita a adoção gradual de *TypeScript* em projetos existentes. O compilador *TypeScript* transpõe o código *TypeScript* para *JavaScript* puro, garantindo que ele possa ser executado em qualquer ambiente que suporte *JavaScript*.

TypeScript é particularmente popular em grandes projetos de desenvolvimento *front-end*, especialmente em combinação com *frameworks* como *Angular*, que foi escrito em *TypeScript*. A adoção de *TypeScript* em projetos de grande escala proporciona benefícios como a refatoração mais segura do código, melhor suporte a ferramentas de desenvolvimento (como *IntelliSense* e auto-completação) e uma detecção de erros mais eficiente durante o desenvolvimento.

Em resumo, *JavaScript* continua a ser uma linguagem fundamental no desenvolvimento *web*, essencial para criar experiências de utilizador interativas e dinâmicas. *TypeScript*, por outro lado, oferece uma poderosa extensão a *JavaScript*, proporcionando maior segurança e robustez no desenvolvimento de aplicações complexas.

4.2.5 SQL

O SQL[Wik24d] é uma linguagem de programação essencial para gerir e manipular bases de dados relacionais. Utilizada amplamente para executar operações de criação, leitura, atualização e eliminação de dados (conhecidas como operações *CRUD*), SQL é uma ferramenta fundamental para qualquer desenvolvedor que lida com grandes volumes de dados.

Uma das funcionalidades mais importantes do SQL é a *Data Definition Language* (DDL), que inclui comandos como *CREATE*, *ALTER* e *DROP*, usados para definir e modificar a estrutura das bases de dados e das suas tabelas. Estes comandos são essenciais para estabelecer a arquitetura inicial de uma base de dados, permitindo a criação de tabelas, índices e outras estruturas.

A *Data Manipulation Language* (DML) é outro componente crucial, com comandos como *INSERT*, *UPDATE*, *DELETE* e *SELECT*. Estes comandos permitem manipular os dados existentes nas tabelas, possibilitando a inserção de novos dados, a atualização de dados existentes, a eliminação de dados desnecessários e a consulta de dados específicos.

Além disso, a *Data Query Language* (DQL), representada principalmente pelo comando *SELECT*, é utilizada para consultar e recuperar dados específicos de uma base de dados. Este comando é extremamente poderoso e permite a criação de consultas complexas que podem filtrar, ordenar e agrupar dados de diversas formas.

A *Data Control Language* (DCL) inclui comandos como *GRANT* e *REVOKE*, que são utilizados para gerir permissões de acesso aos dados. Estes comandos são fundamentais para garantir a segurança e a integridade dos dados, permitindo que apenas utilizadores autorizados possam realizar determinadas operações.

O SQL também permite a realização de junções (*joins*) para combinar dados de várias tabelas de forma eficiente. As funções agregadas, como *SUM*, *AVG*, *COUNT*, *MAX* e *MIN*, permitem realizar cálculos sobre os dados diretamente nas consultas. Além disso, o SQL suporta subconsultas (*subqueries*), que são extremamente úteis para operações complexas que requerem a consulta de dados de forma hierárquica.

A gestão de transações, através de comandos como *BEGIN TRANSACTION*, *COMMIT* e *ROLLBACK*, garante a integridade dos dados, permitindo agrupar várias operações num único bloco transacional. Isso assegura que todas as operações dentro de uma transação sejam concluídas com sucesso ou que, em caso de falha, todas as operações sejam revertidas, mantendo a consistência dos dados.

Dominar SQL é crucial para qualquer estudante de Engenharia Informática, sendo uma competência altamente valorizada no mercado de trabalho. A sua importância na gestão de dados e sistemas de informação torna-o uma ferramenta indispensável para desenvolvedores que pretendem trabalhar com bases de dados relacionais e sistemas de gestão de informação.

4.3 Frameworks

4.3.1 Angular

Angular[Ang24] é uma estrutura de desenvolvimento *front-end* mantida pelo *Google*, usada para construir aplicações *web* dinâmicas e interativas. Baseado em *TypeScript*, uma extensão da linguagem *JavaScript*, o *Angular* oferece uma ampla gama de funcionalidades e ferramentas para facilitar o desenvolvimento de aplicações robustas e escaláveis.

Uma das características distintivas do *Angular* é a sua arquitetura baseada em componentes. As aplicações *Angular* são construídas a partir de componentes reutilizáveis, cada um encapsulando a sua própria lógica, interface de utilizador e estilos. Isso promove a modularidade e facilita a manutenção e evolução do código.

Além dos componentes, o *Angular* também oferece um sistema de injeção de dependências poderoso e uma camada de serviços para a partilha de lógica entre componentes. Isso



Figura 4.1: Logótipo Angular

permite separar as preocupações e manter um código limpo e organizado.

Angular utiliza o conceito de *data binding* bidirecional para manter a sincronização entre os modelos de dados e a interface de utilizador. Isso significa que qualquer alteração nos dados é automaticamente refletida na interface de utilizador e vice-versa, simplificando o desenvolvimento e eliminando a necessidade de código redundante.

Outra característica fundamental do Angular é o seu sistema de rotas, que permite a criação de aplicações de página única, com navegação dinâmica e gestão de estados. Isso oferece uma experiência de utilizador fluida e responsiva, sem a necessidade de recarregar a página a cada interação.

O *Angular* é amplamente utilizado na indústria devido à sua escalabilidade, desempenho e robustez. Com uma vasta comunidade de desenvolvedores e uma documentação abrangente, o *Angular* é uma escolha popular para o desenvolvimento de aplicações web modernas e complexas.

4.3.2 .NET

.NET [cod24] é uma *framework* de desenvolvimento de software, desenvolvido pela *Microsoft*, que oferece uma vasta gama de ferramentas e bibliotecas para criar aplicações *web*, *desktop* e *móveis*. Baseado na linguagem de programação *C#*, o *.NET* é conhecido pela sua versatilidade, desempenho e segurança.

Uma das principais vantagens do *.NET* é a sua abordagem de desenvolvimento orientada a objetos e baseada em componentes. Isso permite aos desenvolvedores criar aplicações modulares e escaláveis, reutilizando código e componentes em diferentes projetos.

O *.NET* oferece suporte a uma variedade de linguagens de programação, incluindo *C#*,



Figura 4.2: Logótipo .NET

VB.NET e *F#*, proporcionando aos desenvolvedores a flexibilidade de escolher a linguagem que melhor se adapta às suas necessidades e preferências.

Além disso, o *.NET* oferece um conjunto abrangente de bibliotecas de classe e *APIs* para lidar com uma ampla gama de tarefas de desenvolvimento, como manipulação de dados, segurança, redes e muito mais. Isso simplifica o desenvolvimento de aplicações complexas e acelera o tempo de lançamento no mercado.

O *.NET* também inclui o *ASP.NET*, uma *framework* para desenvolvimento de aplicações *web*, e o *Xamarin*, uma plataforma para criar aplicações móveis multiplataforma. Isso permite aos desenvolvedores criar aplicações para diferentes plataformas, como *Windows*, *macOS*, *iOS* e *Android*, compartilhando grande parte do código-fonte entre elas.

Com uma vasta comunidade de desenvolvedores e suporte ativo da *Microsoft*, o *.NET* é uma escolha popular para empresas de todos os tamanhos que buscam criar aplicações modernas e escaláveis. A sua compatibilidade com várias plataformas e a sua ampla gama de recursos tornam-no uma ferramenta poderosa para o desenvolvimento de software em diversos cenários.

4.3.3 SCRUM

SCRUM[SCR24] é uma *framework* ágil amplamente utilizada para o desenvolvimento e gestão de projetos de *software*, focada em melhorar a eficiência e flexibilidade das equipas de desenvolvimento. A sua abordagem iterativa e incremental permite entregar produtos de alta qualidade de forma rápida e eficiente, respondendo às mudanças e requisitos dos clientes.

A *framework SCRUM* é composta por três papéis principais: o *Product Owner*, que maximiza o valor do produto e gere o *backlog*; o *Scrum Master*, que facilita o processo e

remove impedimentos; e a Development Team, que é auto-organizada e responsável por criar os incrementos do produto. Estes papéis trabalham em conjunto para assegurar a entrega contínua de valor.

O *SCRUM* utiliza eventos regulares e *time-boxed* para criar uma cadência previsível e minimizar reuniões desnecessárias. As principais reuniões incluem a *Sprint Planning*, onde se define o trabalho para a *sprint*; o *Daily Scrum*, uma reunião diária para sincronizar as atividades; a *Sprint Review*, onde se apresenta o que foi desenvolvido e se recolhe *feedback*; e a *Sprint Retrospective*, que se foca na melhoria contínua do processo.

Os artefactos do *SCRUM*, como o *Product Backlog*, *Sprint Backlog* e *Increment*, garantem transparência e clareza no progresso do trabalho. O *Product Backlog* é uma lista ordenada de tudo o que pode ser necessário no produto, o *Sprint Backlog* contém os itens selecionados para a *sprint* e um plano para alcançar os objetivos, e o *Increment* é o resultado de todos os itens completados, que deve estar num estado utilizável.

O *SCRUM*, com a sua estrutura clara e ciclos de *feedback* frequentes, ajuda as equipas a entregar valor continuamente e a adaptar-se rapidamente às mudanças, proporcionando um ambiente de desenvolvimento mais produtivo e colaborativo.

4.4 Ferramentas

4.4.1 Microsoft Teams

O *Microsoft Teams* é uma plataforma de comunicação e colaboração desenvolvida pela *Microsoft*, criada para que equipas e organizações possam trabalhar em conjunto de forma eficiente e produtiva. Esta ferramenta integra várias funcionalidades, como *chat*, reuniões, chamadas e colaboração em documentos, num único ambiente.

Uma das funcionalidades principais do *Microsoft Teams* é o *chat* em tempo real, que permite trocar mensagens instantâneas individualmente ou em grupos. Os utilizadores podem partilhar ficheiros, *emojis*, reações e até mesmo integrar aplicações de terceiros nas conversas.

Além do *chat*, o *Teams* oferece uma ampla gama de recursos para reuniões virtuais, incluindo videoconferências, chamadas de áudio e partilha de ecrã. Os utilizadores podem agendar e participar facilmente em reuniões, convidar participantes externos e colaborar em documentos durante as sessões.

Outro aspeto importante do *Microsoft Teams* é a capacidade de colaborar em documentos em tempo real através do *Microsoft 365*. Os utilizadores podem criar, editar e partilhar documentos do *Word*, *Excel*, *PowerPoint* e outros aplicativos diretamente no *Teams*, facilitando a colaboração em projetos e documentos partilhados.



Figura 4.3: Logótipo Microsoft Teams

Além disso, o *Teams* integra-se com uma variedade de aplicações e serviços de terceiros, permitindo o acesso a ferramentas e informações adicionais, sem sair do ambiente do *Teams*. Isso inclui integrações com serviços de armazenamento na nuvem, ferramentas de produtividade, sistemas de gestão de projetos e muito mais.

Com uma interface intuitiva e uma ampla gama de recursos, o *Microsoft Teams* é uma ferramenta poderosa para equipas e organizações que procuram melhorar a comunicação, colaboração e produtividade no local de trabalho. A sua integração com outros produtos e serviços da *Microsoft* torna-o uma escolha popular para empresas que já utilizam o ecossistema *Microsoft 365*.

4.4.2 *GitLab*

O *GitLab* é uma plataforma de desenvolvimento de software colaborativo baseada em *Git*. É uma ferramenta poderosa que oferece um conjunto abrangente de funcionalidades para gerir o ciclo de vida do desenvolvimento de software, desde o controlo de versões até à integração contínua e à entrega contínua.

Uma das características distintivas do *GitLab* é que ele oferece tanto uma versão auto-gerida (*self-hosted*) quanto uma versão baseada na *cloud*, proporcionando flexibilidade aos utilizadores para escolherem a opção que melhor se adapta às suas necessidades.

No *GitLab*, os utilizadores podem criar e hospedar repositórios *Git* para os seus projetos de software, permitindo o controlo de versões eficiente e o trabalho colaborativo entre equipas. Além disso, o *GitLab* oferece uma série de ferramentas para gestão de projetos, incluindo acompanhamento de problemas (*issue tracking*), gestão de tarefas, *wikis* e muito mais.

Uma das funcionalidades mais poderosas do *GitLab* é a integração contínua e a entrega contínua. Os utilizadores podem configurar *pipelines* automatizados que executam testes de código, construções automatizadas e implantações contínuas, permitindo uma abor-

dagem ágil e eficiente ao desenvolvimento de software.

O *GitLab* também inclui funcionalidades de segurança integradas, como limpeza de segurança automatizadas, revisões de código e controlo de acesso granular, ajudando a garantir a qualidade e a segurança do código em todos os estágios do ciclo de vida do desenvolvimento.

Com uma interface intuitiva e uma vasta gama de funcionalidades, o *GitLab* é uma ferramenta essencial para equipas de desenvolvimento de software que procuram melhorar a colaboração, a eficiência e a qualidade do código. A sua abordagem aberta e extensível permite integrar facilmente com outras ferramentas e serviços, tornando-o uma escolha popular para projetos de todos os tamanhos.

4.4.3 *GitHub Desktop*

O *GitHub Desktop* é uma aplicação de interface gráfica (*Graphical User Interface* (GUI)) que facilita o trabalho com o *Git* e o *GitHub*. Desenvolvida pela *GitHub, Inc.*, a aplicação permite aos utilizadores gerirem facilmente os seus repositórios *Git* diretamente a partir do ambiente de trabalho, sem a necessidade de recorrerem a comandos de linha de comando.



Figura 4.4: Logótipo GitHub Desktop

Com o *GitHub Desktop*, os utilizadores podem realizar várias tarefas relacionadas com o controlo de versão *Git* e a colaboração no *GitHub*. Podem clonar repositórios do *GitHub* para o seu ambiente local com apenas alguns cliques, permitindo-lhes trabalhar nos projetos sem terem que lidar com comandos. Além disso, podem criar novos *branches*, fazer alterações nos ficheiros e adicionar essas alterações com mensagens descritivas, tudo através de uma interface intuitiva.

A aplicação também facilita a gestão de *branches* e solicitações de *pull*, permitindo aos uti-

lizadores criar, visualizar e alternar entre *branches*, bem como criar e rever solicitações de *pull*, tudo de forma simplificada. Além disso, o *GitHub Desktop* torna mais simples a sincronização das alterações locais com o *GitHub*, permitindo aos utilizadores enviar as suas alterações para o repositório remoto e receber as alterações feitas por outros colaboradores.

Por fim, os utilizadores podem visualizar o histórico de alterações do projeto, incluindo os *commits*, *branches* e *merges*, para acompanhar o progresso do desenvolvimento. O *GitHub Desktop* é uma ferramenta útil para programadores que preferem uma interface gráfica intuitiva para trabalhar com *Git* e *GitHub*. Ao simplificar as tarefas comuns de controlo de versão e colaboração, ajuda os utilizadores a serem mais produtivos no desenvolvimento de software.

4.4.4 *Figma*

O *Figma* é uma ferramenta poderosa de design e prototipagem que permite aos utilizadores criar interfaces de utilizador (*User Interface* (UI)) e experiências de utilizador (*User Experience* (UX)) de alta qualidade de forma colaborativa e eficiente. Funciona completamente na *cloud*, o que significa que os utilizadores podem aceder e trabalhar nos seus projetos a partir de qualquer dispositivo com ligação à internet, sem a necessidade de instalar software adicional.

Com o *Figma*, os utilizadores podem criar *designs* de interfaces de utilizador para *websites*, aplicações móveis, produtos digitais e muito mais. A aplicação oferece uma ampla gama de ferramentas de design, incluindo ferramentas de desenho vetorial, edição de texto, estilos de camada, componentes reutilizáveis e muito mais. Os utilizadores podem criar *designs* responsivos e adaptáveis, garantindo uma experiência consistente em diferentes dispositivos e tamanhos de ecrã.

Além disso, o *Figma* facilita a colaboração em tempo real entre equipas, permitindo que múltiplos utilizadores trabalhem no mesmo projeto simultaneamente. Os utilizadores podem partilhar *designs* com colegas de equipa e clientes, solicitar *feedback* e realizar revisões de forma eficiente. A funcionalidade de comentários integrada permite uma comunicação clara e contextual diretamente dentro da aplicação.

Uma das características mais poderosas do *Figma* é a capacidade de criar protótipos interativos e simulações de experiência de utilizador. Os utilizadores podem criar *links* entre diferentes *frames* para simular fluxos de navegação, transições de página e interações de elementos de interface. Isso permite que os designers testem e iterem rapidamente sobre os seus *designs*, antes de os implementarem no produto final.

Em resumo, o *Figma* é uma ferramenta versátil e eficaz para designers de UI/UX, oferecendo uma combinação única de funcionalidades avançadas, colaboração em tempo real e acessibilidade em qualquer lugar. É amplamente utilizado na indústria de design digital

por equipas de todas as dimensões, desde pequenas *startups* até grandes empresas.

Capítulo 5

Projeto Final

5.1 Introdução

Este capítulo aborda o desenvolvimento do projeto principal realizado durante o estágio na Latitude - Digital Enablers, LDA. Este projeto envolveu a criação de uma plataforma tecnológica para a gestão dos serviços de transporte público flexível no município de Castelo Branco. A plataforma foi desenvolvida com o objetivo de melhorar a eficiência e a qualidade dos serviços de transporte público, proporcionando uma solução inovadora e acessível tanto para os utilizadores finais como para os gestores de transporte.

Neste capítulo, serão detalhadas as fases do projeto, desde a sua conceção inicial até à implementação. Serão abordados os desafios enfrentados, as tecnologias utilizadas, as metodologias adotadas, e os resultados alcançados. Além disso, será feita uma reflexão sobre a experiência adquirida e o impacto do projeto tanto no contexto académico quanto profissional.

A introdução deste capítulo visa contextualizar o leitor sobre a importância e a relevância do projeto, fornecendo uma visão geral das atividades realizadas e dos objetivos atingidos. Ao longo deste capítulo, será possível compreender o processo de desenvolvimento do projeto e as contribuições significativas que ele trouxe para a minha formação.

5.2 Objetivos

Com a minha integração na equipa, o meu principal objetivo foi acompanhar o ritmo estabelecido, desenvolvendo software de qualidade dentro dos prazos estipulados. Paralelamente, procurei aprimorar as minhas competências de trabalho em equipa.

Ao longo do projeto, deparei-me com diversos desafios e obstáculos. Aprendi a lidar com prazos apertados, a dar prioridade às tarefas, a comunicar de forma eficaz e a colaborar de maneira produtiva com os membros da equipa. Estas experiências foram cruciais para o meu crescimento profissional, preparando-me para enfrentar futuros desafios na área.

5.3 Metodologia

Na metodologia adotada para o projeto final, seguimos uma abordagem ágil, baseada na *framework Scrum*. Esta metodologia permitiu uma gestão eficiente do projeto, com ciclos de desenvolvimento iterativos e incrementais.

Inicialmente, foram definidos os requisitos do projeto em colaboração com os *stakeholders*, estabelecendo assim o *backlog* do produto. Este *backlog* foi dividido em *sprints* de curta duração, geralmente de duas semanas, durante as quais foram implementadas funcionalidades específicas.

Cada *sprint* começou com uma reunião de planeamento, onde foram selecionadas as tarefas a serem realizadas e definidas as metas para o *sprint*. Durante o *sprint*, a equipa realizou reuniões diárias de acompanhamento, conhecidas como *daily*s, para discutir o progresso, identificar possíveis obstáculos e ajustar o plano, se necessário.

No final de cada *sprint*, realizou-se uma revisão, onde foram apresentadas as funcionalidades desenvolvidas aos *stakeholders* para *feedback*. Além disso, foi realizada uma retrospectiva para avaliar o desempenho da equipa e identificar áreas de melhoria.

Esta abordagem ágil permitiu uma resposta flexível às mudanças nos requisitos do projeto e uma entrega contínua de valor ao cliente ao longo do desenvolvimento do produto.

5.4 Equipa

No contexto do projeto dos Transportes Públicos Flexíveis, a nossa equipa é composta por cinco membros, juntamente com três membros experientes já presentes na equipa. O *Tech Leader*, que desempenha o papel de co-orientador, também está connosco para nos orientar e liderar o desenvolvimento do projeto.

Dos três membros experientes, um deles está encarregado do design, *mockups* e outras atividades relacionadas com a interface do utilizador. Os outros dois membros contribuem bastante, oferecendo apoio tanto no desenvolvimento do *front-end* como do *back-end*. Além disso, desempenham um papel fundamental na revisão e aprovação do código desenvolvido pelos estagiários.

Durante o estágio, todos nós tivemos a oportunidade de trabalhar em ambas as áreas, *front-end* e *back-end*. No entanto, ao longo do processo, alguns de nós descobrimos uma afinidade maior por uma das áreas. No meu caso, desenvolver no *back-end* despertou em mim um interesse particular e uma preferência pela parte técnica do projeto.

5.5 Trabalho Desenvolvido

Durante o desenvolvimento deste projeto, a nossa equipa começou a criar uma aplicação totalmente nova, a partir do zero. Foi fornecido um documento abrangente da Câmara Municipal de Castelo Branco, intitulado "Aquisição e Operacionalização de uma Plataforma Tecnológica para a Gestão dos Serviços de Transporte Público Flexível no Município de Castelo Branco". Esse documento serviu como ponto de partida, delineando as especificações técnicas e os requisitos essenciais para o projeto.

O nosso primeiro passo foi analisar detalhadamente o documento fornecido, discutindo e debatendo ideias para determinar a melhor abordagem para o desenvolvimento da aplicação. Estas sessões de *brainstorming* foram fundamentais para alinhar a nossa visão e estabelecer os objetivos do projeto de forma clara e precisa.

Uma das primeiras tarefas que enfrentei foi a elaboração do modelo de dados da aplicação, disponível na figura 5.1. Trabalhámos em estreita colaboração para definir a estrutura fundamental que sustentaria toda a lógica do sistema. Identificámos cuidadosamente os principais intervenientes envolvidos na aplicação, atribuindo a cada um deles diferentes papéis e privilégios (figura 5.2). Esta etapa exigiu uma análise minuciosa e detalhada para garantir que o modelo fosse robusto o suficiente para atender às diversas necessidades dos utilizadores.

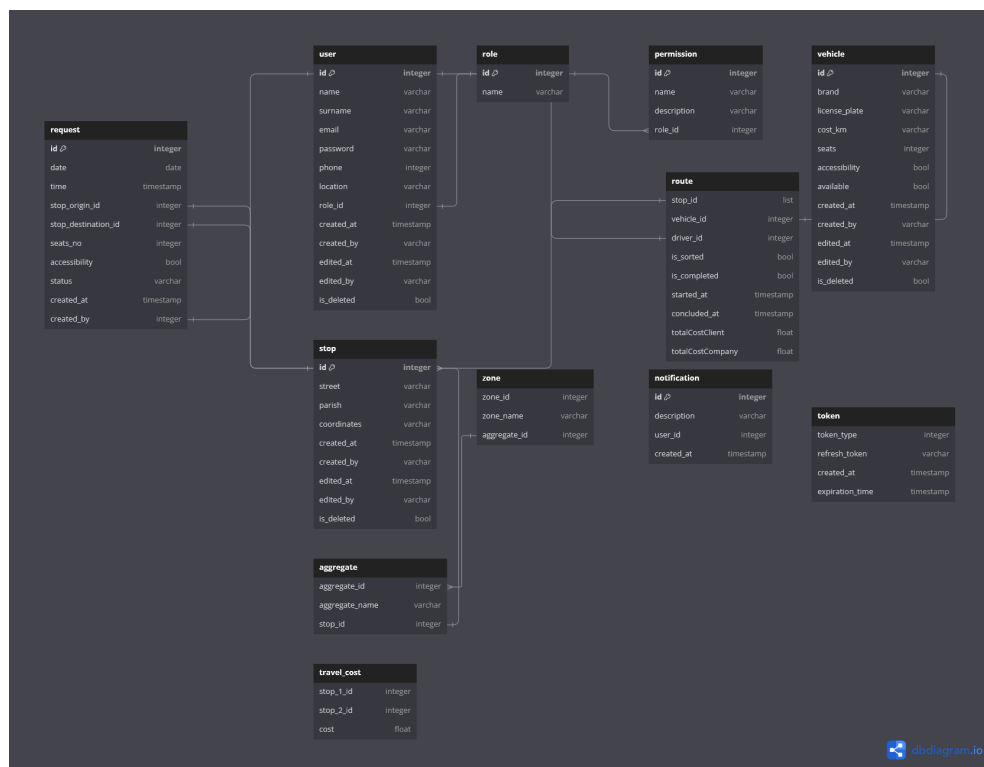


Figura 5.1: Modelo de Dados.

Com o modelo de dados em mãos, avançámos para a criação das *user stories*. Detalhámos

	Create Account	Edit/Delete Own Account	Add Other Users	Edit Other Users	Delete Other Users	Add Vehicle	Edit Vehicle	Delete Vehicle	Add/Edit/Delete Zone	Add/Edit/Delete Stop
Guest (no login required)	X									
Customer (login required)		X								
Driver (login required)										
Operator (login required)			X	X			X			
Admin (login required)	X	X	X	X	X	X	X	X	X	X

Figura 5.2: Privilégios de cada utilizador (exemplo, tabela não completa).

os requisitos funcionais e técnicos de cada página e funcionalidade da aplicação, assegurando-nos de que cada aspeto do projeto fosse cuidadosamente planeado e documentado. Paralelamente, uma colega dedicou-se ao desenvolvimento dos *mockups*, criando representações visuais vívidas e precisas da interface da aplicação.

No âmbito do desenvolvimento do *back-end*, concentrei-me na implementação das funcionalidades relacionadas aos utilizadores. Desenvolvi os endpoints necessários para realizar as operações de criação, leitura, atualização e eliminação de dados, bem como para lidar com a autenticação e autorização dos utilizadores através do *Identity .NET* (figura 5.3). Esta fase exigiu uma atenção meticulosa aos detalhes e uma compreensão profunda dos requisitos do sistema.

Method	Endpoint	Description
POST	/api/v1/Users/create-user	Creates an User.
GET	/api/v1/Users/get-user/{userId}	Get an User by his ID.
PUT	/api/v1/Users/update-user	Edits an User.
DELETE	/api/v1/Users/delete-user/{userId}	Deletes an User.
DELETE	/api/v1/Users/delete-account	Deletes Own Account.
GET	/api/v1/Users/get-all-users	Gets the users by page and size.
GET	/api/v1/Users/getProfile/{userId}	Get an User by his ID.
PUT	/api/v1/Users/updateProfile	Edits an User.

Figura 5.3: Endpoints associados aos utilizadores

Após a conclusão da implementação do *back-end*, concentramo-nos no desenvolvimento do *front-end* da aplicação. Utilizando as minhas habilidades em HTML, CSS e *TypeScript*, trabalhei na construção de páginas dinâmicas e responsivas (figura 5.4) que oferecessem uma experiência de utilizador intuitiva e agradável. Durante esse processo, enfrentámos vários desafios técnicos, mas com determinação e trabalho em equipa, conseguimos superá-los com sucesso.

Além disso, ao longo do projeto, participámos em reuniões regulares de acompanhamento com os nossos orientadores e a equipa técnica da empresa, onde apresentávamos os progressos alcançados. Discutimos os desafios encontrados e recebemos orientações valiosas para melhorar continuamente o nosso trabalho.

Neste momento, continuo envolvido no projeto, focando-me na implementação da fun-

The screenshot shows a web application interface for managing users. At the top, there is a navigation menu with items: Pedidos, Utilizadores (active), Motoristas, Veiculos, Paragens, Zonas, Agregados, and a user profile icon. The main header is 'Utilizadores'. Below it, there is a search bar labeled 'Procurar' and a blue button '+ Novo Utilizador'. A 'Filtro' button is located on the left side. The main content is a table with the following data:

Nome	Morada	Email	Contacto	Função
Cristiano Ronaldo	Madeira	cr7@gmail.com	967654321	Operador , Motorista
Rafael Leão	Aveiro	lionking@gmail.com	967556532	Cliente , Motorista
Bernardo Silva	Lisboa	silva@gmail.com	963636333	Admin
Bruno Fernandes	Covilhã	bruninho@gmail.com	967676767	Cliente , Operador , Motorista
João Palhinha	Porto	palhinha@gmail.com	966666666	Admin , Operador
Vitinha	Alvoco da Serra	vitinha@gmail.com	967674888	Operador , Motorista
Nuno Mendes	Cortes de Baixo	nuninho@gmail.com	123456789	Motorista

Figura 5.4: Nesta figura podemos observar a página utilizadores do *backoffice* da aplicação. Cada utilizador pode ter um ou mais *roles*, podemos também obter detalhes relevantes sobre cada utilizador, como o contacto, Email, Morada e Nome. Temos uma caixa de pesquisa, onde podemos pesquisar por palavras até encontrarmos o utilizador pretendido. Existe também um botão de filtro para podermos filtrar os utilizadores por *role*. Por fim, temos um botão para criar um novo utilizador que irá abrir um formulário como podem ver na figura 5.5. Ao clicarmos em cima de um utilizador uma janela será aberta (ver figura 5.6) onde serão mostradas mais informações, assim como, dois botões para editar e eliminar o utilizador seleccionado.

cionalidade de notificações da aplicação. Estou a explorar novas tecnologias e técnicas para criar uma solução eficaz e eficiente, que atenda às necessidades dos utilizadores e contribua para o sucesso global do projeto.

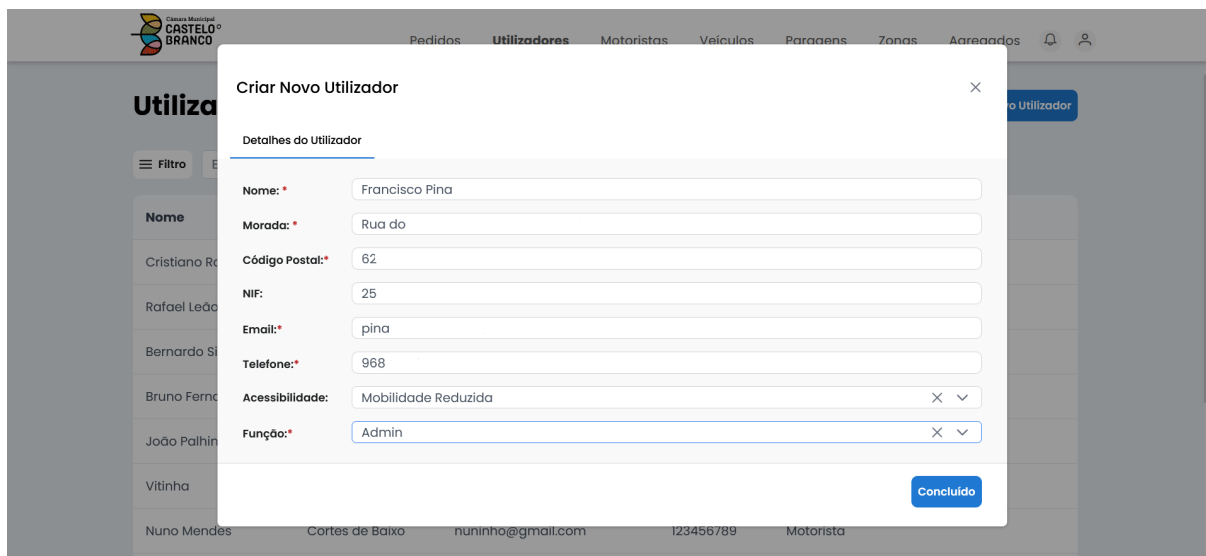


Figura 5.5: Nesta figura podemos observar o formulário de criação de um novo utilizador. O formulário encontra-se já completo e como podemos observar há certas palavras que têm um asterisco à frente, isto significa que o botão de concluído só é clicável se o utilizador completar esses campos, são campos obrigatórios, caso contrário não será possível concluir a criação do novo utilizador.

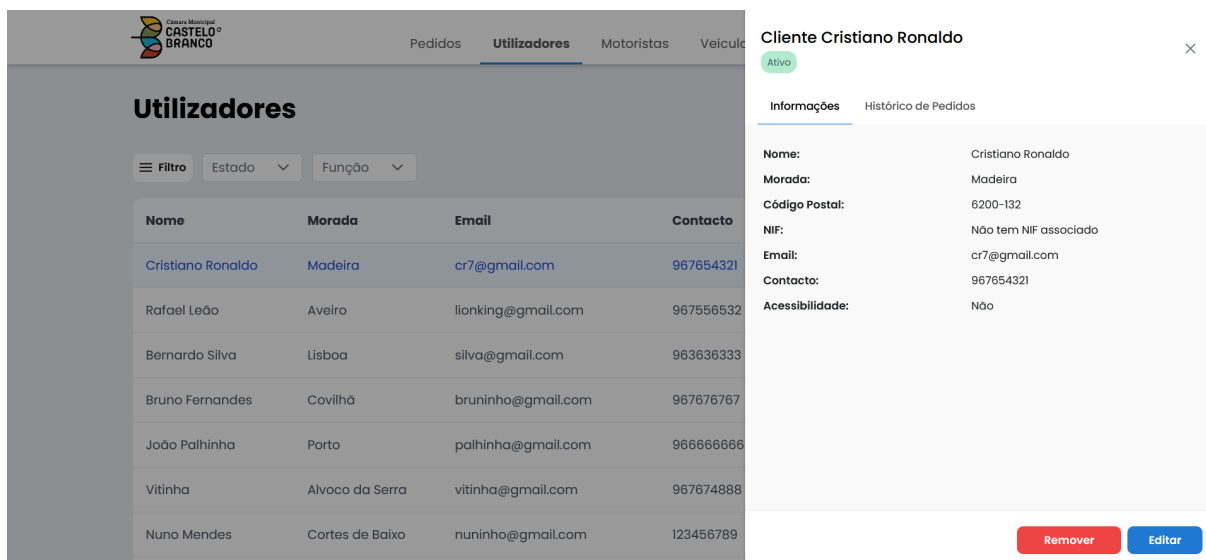


Figura 5.6: Nesta figura podemos ver mais informações relativas ao utilizador selecionado, assim como o seu histórico de pedidos. É possível editar o utilizador e eliminá-lo. É importante referir que este eliminar é um *Soft Delete*, isto é, o utilizador não é eliminado totalmente da base de dados, apenas fica inativo.

Capítulo 6

Conclusão

6.1 Conclusões Principais

Ao chegarmos ao término desta fase do projeto, várias conclusões importantes emergiram, refletindo não apenas as conquistas alcançadas até agora, mas também os desafios enfrentados e as lições aprendidas ao longo desta jornada.

Destaca-se, sobretudo, a eficácia da colaboração em equipa, um elemento fundamental para superar os obstáculos técnicos e manter um progresso consistente. Além disso, a aplicação prática dos conhecimentos teóricos adquiridos, durante o percurso académico, demonstrou a sua relevância e importância no campo da engenharia de software. A interação próxima com os *stakeholders* revelou-se crucial, permitindo-nos compreender e atender às necessidades e expectativas dos utilizadores finais.

6.2 Trabalho Futuro

Embora tenhamos alcançado marcos significativos até este ponto, reconheço que há muito trabalho a ser feito para concluir plenamente este projeto. Olhando para o futuro, identificam-se diversas áreas de trabalho que podem contribuir para a melhoria contínua da aplicação e para a expansão das suas funcionalidades. Uma dessas áreas reside na otimização do desempenho da aplicação, identificando e resolvendo possíveis pontos fracos e aprimorando a eficiência global. Além disso, poderiam ser exploradas novas funcionalidades e melhorias na interface do utilizador, visando aperfeiçoar ainda mais a experiência do utilizador. A integração de tecnologias emergentes e a adaptação às mudanças nas necessidades dos utilizadores também devem ser consideradas no desenvolvimento futuro da aplicação.

6.3 Palavras Finais

Ao encerrar esta fase do projeto, gostaria de expressar a minha gratidão pela oportunidade de participar neste estágio. Foi uma experiência enriquecedora e gratificante, que me proporcionou aprendizagens valiosas e um crescimento pessoal e profissional significativo. Agradeço aos membros da equipa, aos *stakeholders* e a todos os envolvidos neste

projeto pelo sua ajuda, empenho e colaboração. Estou entusiasmado com as oportunidades futuras que este projeto abre e ansioso para aplicar as lições aprendidas em futuros desafios e projetos. Que este projeto seja apenas o primeiro passo de uma jornada de sucesso contínuo na área da engenharia informática.

Bibliografia

- [Ang24] Angular. Angular, 2024. Acessado em: 11 de junho de 2024. Available from: <https://v17.angular.io/guide/what-is-angular>. 16
- [cod24] codecademy. .net, 2024. Acessado em: 11 de junho de 2024. Available from: <https://www.codecademy.com/article/what-is-net>. 17
- [Lat24] Latitudde. Latitudde, 2024. Acessado em: 11 de junho de 2024. Available from: <https://latitudde.com/>. 5
- [Moz24] Mozilla. Javascript, 2024. Acessado em: 11 de junho de 2024. Available from: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. 14
- [SCR24] SCRUM. Scrum, 2024. Acessado em: 11 de junho de 2024. Available from: <https://www.scrum.org/resources/what-scrum-module>. 18
- [w3s24a] w3schools. Csharp, 2024. Acessado em: 11 de junho de 2024. Available from: https://www.w3schools.com/cs/cs_intro.php. 13
- [w3s24b] w3schools. Css, 2024. Acessado em: 11 de junho de 2024. Available from: https://www.w3schools.com/css/css_intro.asp. 12
- [Wik24a] Wikipedia. back-end, 2024. Acessado em: 11 de junho de 2024. Available from: https://pt.wikipedia.org/wiki/Front-end_e_back-end. 11
- [Wik24b] Wikipedia. front-end, 2024. Acessado em: 11 de junho de 2024. Available from: https://pt.wikipedia.org/wiki/Front-end_e_back-end. 11
- [Wik24c] Wikipedia. Html, 2024. Acessado em: 11 de junho de 2024. Available from: <https://en.wikipedia.org/wiki/HTML>. 11
- [Wik24d] Wikipedia. Sql, 2024. Acessado em: 11 de junho de 2024. Available from: <https://en.wikipedia.org/wiki/SQL>. 15