



Campus Universitário/Académico de Almada
Escola Superior de Tecnologia e Gestão Jean Piaget/Instituto Politécnico Jean Piaget
do Sul

UAW – Uma App para implementação de uma rede social Universitária

Discentes:

Cátia Sofia Victor Lopes Guerra, n.º 57150
Luís Manuel Nunes Ferreira, n.º 61102
Nuno Miguel Bernardo Simões, n.º 61123
Paulo Alexandre Gonzalez Henriques, n.º 38767
Ricardo Daniel Marques Novo, n.º 61014
Kael Loureiro Schossler de Sá, n.º 60291

Relatório final

Orientador: Professor Rui Pedro Patrão Carneiro
Coorientador: Professor Doutor Toacy Cavalcante de Oliveira
Licenciatura em Engenharia Informática

2023/2024

ÍNDICE

ÍNDICE DE FIGURAS.....	7
ÍNDICE DE ANEXOS.....	8
LISTA DE ABREVIATURAS.....	8
RESUMO	9
PALAVRAS-CHAVE	9
1. CAPÍTULO: INTRODUÇÃO	10
1.1. Introdução.....	10
1.2. Motivação	11
1.3. Objetivos do Projeto	12
1.4. Estrutura do Trabalho.....	13
2. CAPÍTULO: FUNDAMENTAÇÃO TEÓRICA E ANÁLISE COMPARATIVA.....	14
2.1. Fundamentação Teórica e Conceptual.....	14
2.1.1. Metodologia de desenvolvimento	14
2.1.2. Princípios de UI/UX Design.....	15
2.1.3. Análise Comparativa.....	15
3. CAPÍTULO: VISÃO MACRO DO PROJETO.....	15
3.1. Explicação da Solução Proposta.....	15
3.2. Requisitos do Sistema	17
3.2.1. Requisitos Funcionais	17
3.2.1.1 Gestão de utilizadores	17
3.2.1.2 Gestão de oportunidades	17
3.2.1.3 Administração do Sistema	18
3.2.2. Requisitos Não Funcionais.....	18
3.2.2.1 Desempenho	18
3.2.2.2. Segurança	18
3.2.2.3. Confiabilidade.....	19
3.2.2.4. Usabilidade	19
3.2.2.5. Manutenibilidade	19
3.3. Demonstração das Funcionalidades Globais.....	20

3.4. Casos de uso e Diagramas.....	20
3.4.1 Descrição Use Cases – Figura 1	21
3.4.1.1 Funcionalidade: Visualização de notícias após autenticação de utilizador	22
3.4.1.2 Funcionalidade: Pesquisa e visualização de Portfólios.....	23
3.4.1.3 Funcionalidade: Visualização e gestão de perfil.....	24
3.5. Tecnologia e Arquitetura do Sistema	25
3.5.1. Camada de Apresentação (Front-End) - Ver Anexo 2	25
3.5.2. Camada de Serviço (Back-End) - Ver Anexo 3	25
3.5.3. Camada de Dados (Banco de Dados) - Ver Anexo 4.....	26
3.5.4. Segurança - Ver Anexo 6	26
3.5.5. Infraestrutura e Deployment - Ver Anexo 7.....	26
3.6. Funcionalidades detalhadas.....	27
3.6.1. Splash / Aguarda carregamento – Figura	27
3.6.2. Início de Sessão - Figura 8, Anexo 2 – Front-End	27
3.6.3. Recuperação de Password – Figura 19, Anexo 2 – Front-End.....	28
3.6.4. Registo de Utilizadores: Empresa – Figura 20 Aluno – Figura 20, Anexo 2 – Front-End	28
3.6.5. Perfil – Empresa – Figura 21 Aluno – Figura 21, Anexo 2 – Front-End	28
3.6.6. Procedimento de Autenticação - Empresa – Figura 18 Aluno – Figura 18, Anexo 2 – Front-End	29
3.6.7. Welcome – (Empresa e Aluno) - Figura 22.....	29
3.6.8. Editar Perfil – Empresa – Imagem X Aluno – Imagem X, Anexo X	30
3.6.9. Portfólio - Aluno – Figura 12	30
3.6.10. Pesquisa – Empresa – Figura 11.....	31
3.7. Descrição Interface UI/UX	31
3.8. Análise	34
3.8.1. Análise de Soluções Concorrentes	34
3.8.1.1. LinkedIn	34
3.8.1.2. Facebook	34
3.8.1.3. Google Classroom	35
3.8.1.4. Microsoft Teams para Educação	35
3.8.2. Análise comparativa com Concorrente	35
3.8.2.1 Forças (Strengths).....	36
3.8.2.2 Fraquezas (Weaknesses)	36
3.8.2.3. Oportunidades (Opportunities).....	37
3.8.2.4. Ameaças (Threats).....	37
4. CAPÍTULO: CASO DE USO DEMONSTRATIVO.....	38
4.1. Caso de uso utilizador Empresa	38
5. CAPÍTULO: CONCLUSÃO	43
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	I
ANEXO 1 – DESIGN	II

Relatório de Design Gráfico UI/UX para Aplicação UAW	II
Brainstorming	II
Criação de Wireframes.....	III
Desenvolvimento de Mockups	III
Prototipagem e Testes de Usabilidade.....	III
Documentação e Entrega	III
Wireframes.....	IV
Mockups.....	VII
 ANEXO 2 – FRONT-END	 IX
 Tecnologia.....	 IX
 Visual Studio Code	 IX
 GitHub.....	 IX
 ReactJS.....	 IX
Componentes	IX
JSX (JavaScript XML)	IX
Virtual DOM	X
One-Way Data Binding	X
Hooks.....	X
 Vantagens do ReactJS	 X
Reutilização de Componentes.....	X
Desempenho	X
Comunidade e Ecossistema.....	X
Desenvolvimento Declarativo	XI
Ecossistema ReactJS	XI
 Estrutura	 XI
 Ecrãs	 XII
Público	XII
Splash / Aguarda carregamento.....	XII
Home	XIII
Início de Sessão	XIV
Recuperar Password.....	XV
Registo de Utilizador / Empresa	XV
Perfil	XVI
Autenticado	XVII
Welcome	XVII
Editar Perfil.....	XVIII
Portfólio.....	XX
Pesquisa.....	XXI
 ANEXO 3 – BACK-END	 XXII
 1. Introdução.....	 XXII

2. Tecnologias Utilizadas	XXII
2.1 Spring Boot.....	XXII
2.2 Maven	XXII
2.3 JWT (JSON Web Token).....	XXII
2.4 Comunicação com o Front-End.....	XXII
3. Estrutura do Projeto	XXIII
3.1 Visão Geral	XXIII
3.2 Estrutura	XXIII
4. Descrição dos Componentes.....	.XXIV
4.1 Configurações.....	.XXIV
4.2 Controladores (Controllers)XXIV
4.3 Modelos (Models)XXIV
4.4 Repositórios (Repositories).....	.XXIV
4.5 Serviços (Services).....	.XXV
5. Endpoints	XXV
5.1 AuthController	XXV
5.1.1 Login	XXV
5.1.2 Registo de Utilizador	XXVI
5.1.3 Registo de Empresa	XXVI
5.2 UAW_NewsFeedController	XXVII
5.2.1 Buscar todas as Notícias	XXVII
5.2.2 Buscar Notícia por ID.....	XXVII
5.2.3 Criar Notícia.....	XXVIII
5.2.4 Atualizar Notícia	XXVIII
5.2.5 Eliminar Notícia	XXVIII
5.3 UAW_PerfisController.....	XXIX
5.3.1 Buscar todos os Perfis	XXIX
5.3.2 Criar Perfil.....	XXXI
5.3.3 Atualizar Perfil	XXXI
5.3.4 Eliminar Perfil	XXXI
5.4 UserController.....	XXXII
5.4.1 Listar todos os Utilizadores	XXXII
5.4.2 Buscar Utilizador por ID.....	XXXII
5.4.3 Criar Utilizador	XXXII
5.4.4 Atualizar Utilizador	XXXIII
5.4.5 Deletar Utilizador	XXXIV
5.4.6 Pesquisar Utilizadores	XXXIV
5.5 Portfólio_Controller:	XXXIV
5.5.1 Criar Portfólio:	XXXIV
5.5.2 Vizualizar Portfólio específico:	XXXV
5.5.3 Salvar Portfólio:	XXXV
5.5.4 Listar Portfólio:	XXXVI
5.5.5 “Deletar” um portfólio:	XXXVII
6. Exemplos de Uso	XXXVIII
6.1 Processo de Login.....	XXXVIII
6.2 Processo de Registo	XXXIX

7. Glossário	XL
ANEXO 4 – BASE DE DADOS	XLI
Introdução.	XLI
Tabelas.....	XLI
uaw_utilizadores	XLI
uaw_perfis.....	XLIII
uaw_portofolio.....	XLV
uaw_axtiposutilizador	XLVII
uaw_axcursos	XLVII
uaw_axportofolio	XLVIII
Diagrama da Base de Dados.	XLIX
Conclusão.....	L
ANEXO 5 – SEGURANÇA	LI
Introdução	LI
Autenticação e Autorização	LI
JSON Web Token (JWT)	LI
Como funciona?.....	LI
Implementação Técnica.....	LII
Configuração de Segurança.....	LII
Principais Configurações:	LII
Criptografia de Senhas	LIII
Como funciona?.....	LIII
Implementação Técnica.....	LIII
Tratamento de Exceções	LIII
Principais Exceções Tratadas:	LIII
Comunicação Segura	LIV
Configuração de CORS	LIV
Conclusão	LIV
ANEXO 6 – TESTES	LV

ÍNDICE DE FIGURAS

Figura 1 - Use Case Sistema UAW	20
Figura 2 - Use Case Utilizador Empresa	22
Figura 3 - Visualização de notícias após autenticação de utilizador	22
Figura 4 - Pesquisa e visualização de Portfólios	23
Figura 5- Diagrama Utilizador Empresa	24
Figura 6 - Logótipo UAW	31
Figura 7 - Ecrã de Vista de Perfil de Utilizador no Front-End	32
Figura 8 - Utilização da Ferramenta de Figma para prototipagem de UI/UX	33
Figura 9 - Ecrãs “Splash”, “Home”, “Início de Sessão”	38
Figura 10 - Ecrã de Registo do tipo Empresa	39
Figura 11 - Ecrã "Welcome", após Autenticação, ação “Pesquisa”	39
Figura 12 - Ecrãs de Pesquisa.....	40
Figura 13 - Ecrã de "Portfólio" de um utilizador Aluno	41
Figura 14 - Ecrã de “Perfil” de Aluno	42
Figura 15 - Ecrã de Projeto de "Portfólio"	42
Figura 16 - Logotipo UAW	II
Figura 17 - Rascunho inicial	IV
Figura 18 - Wireframe.....	IV
Figura 19 - Esboço 1	V
Figura 20 - Esboço #2	VI
Figura 21 - Mockups.....	VIII
Figura 22 - Estrutura de Ficheiros do Front-End no Visual Studio Code	XI
Figura 23 - Ecrã Splash	XIII
Figura 24 - Ecrã Home.....	XIII
Figura 25 - Ecrã Início de Sessão	XIV
Figura 26 - Ecrã Recuperação de Password	XV
Figura 27 - Ecrãs de Registo de novo Utilizador, Aluno e Empresa.....	XVI
Figura 28 - Ecrãs de Visualização Pública de Perfil, Aluno e Empresa.....	XVII
Figura 29 - Ecrãs “Welcome”, após autenticação, Aluno e Empresa	XVIII
Figura 30 - Ecrã de Atualização de Perfil de Aluno	XIX
Figura 31 - Ecrã de Atualização de Perfil de Empresa	XIX
Figura 32 - Ecrã de Portfólio Público de Utilizador Aluno	XX
Figura 33 - Ecrãs de fases de Pesquisa para utilizador Empresa	XXI
Figura 34 - Estrutura de ficheiros do Back-End no IntelliJ	XXIII
Figura 35 - BPMN	XXXVIII
Figura 36 - BPMN #2	XXXIX
Figura 37 - Tabela Utilizadores	XLII

Figura 38 - Tabela Perfis	XLIV
Figura 39 - Tabela Portfólio	XLVII
Figura 40 - Tabela Auxiliar Tipo de Cursos	XLVIII
Figura 41 - Tabela Auxiliar Tipo de Portfólio.....	XLIX
Figura 42 - Diagrama Geral da Base de Dados	XLIX

ÍNDICE DE ANEXOS

Design – Anexo 1

Camada de Apresentação (Front-End) - Anexo 2

Camada de Serviço (Back-End) - Anexo 3

Camada de Dados (Banco de Dados) - Anexo 4

Segurança - Anexo 5

Testes – Anexo 6

LISTA DE ABREVIATURAS

- BPMN: *Business Process Model and Notation*
- HTTP: *HyperText Transfer Protocol*
- HTTPS: *HyperText Transfer Protocol Secure*
- JSON: *JavaScript Object Notation*
- JWT: *Jason web Token*
- REST: *Respresentational State Transfer*
- UAW: *U Are Welcome*
- UI / UX: *User Interface / User Experience*
- MVC – *Model View Controller*
- SWOT – *Strengths, Weaknesses, Opportunities, and Threats*

RESUMO

Este trabalho apresenta o desenvolvimento da plataforma UAW (U Are Welcome), uma aplicação móvel concebida para facilitar a interação entre alunos, empresas e instituições académicas. A UAW permite a criação de perfis detalhados, a partilha de portfólios e oferece um sistema de pesquisa eficaz. O projeto inclui uma análise comparativa da UAW com aplicações que poderão apresentar serviços concorrentes e demonstra como ela pode ser uma solução para as necessidades específicas no meio académico. No capítulo de fundamentação teórica, são discutidos as tecnologias utilizadas e os princípios de design de UI/UX, as funcionalidades da plataforma, os requisitos do sistema, e os casos de uso são detalhados, juntamente com diagramas ilustrativos. A visão futura do projeto inclui a implementação de notificações personalizadas, ferramentas de pesquisa avançada, e a promoção de eventos e workshops. A conclusão destaca a UAW como uma solução inovadora e essencial para a integração e desenvolvimento da comunidade académica, com potencial para expandir e transformar a interação e a colaboração entre os seus utilizadores. Esta plataforma visa maximizar as oportunidades educacionais e profissionais, estabelecendo um novo padrão para a conectividade no ambiente académico.

PALAVRAS-CHAVE

Plataforma Académica; Networking Profissional; Interação Universidade-Empresa; UAW (U Are Welcome); Portfólios Digitais; Comunicação Educacional; UI/UX Design; Tecnologias de Desenvolvimento; Oportunidades de Estágio e Emprego.

1. CAPÍTULO: INTRODUÇÃO

1.1. Introdução

No âmbito da Unidade Curricular de Projeto Integrador, da Licenciatura de Engenharia Informática, foi-nos proposto a realização de um Projeto em que os objetivos fossem alcançados de forma incremental, adequada e com o padrão de qualidade da nossa licenciatura, assim como aplicadas as competências adquiridas ao longo do nosso percurso académico.

Reunimos a equipa competente e desenvolvemos a ideia de criar um sistema de software inovador denominado por UAW – U ARE WELCOME.

O sistema tem como objetivo revolucionar e promover a interação e colaboração entre alunos, empresa e instituição académica.

Mediante as necessidades encontradas ao longo do nosso percurso, verificámos que existe uma grande falta de comunicação não formal entre a comunidade académica, como por exemplo a partilha de recursos e experiências entre os novos, atuais e ex-alunos da instituição. Essa ligação entre os diversos grupos poderia facilitar a partilha de matéria lecionada, informação regulamentar, atividades a decorrer, apontamentos, experiências vivenciadas, entre outros. Desta forma resultou a criação do utilizador Aluno.

De forma a promover a comunidade académica e a fomentar a sua presença no mercado de trabalho adicionámos o utilizador Empresa, que irá por sua vez ter acesso ao perfil e às informações dos utilizadores Alunos pertencentes à instituição, assim como aos seus portfólios, constituídos por materiais adicionados pelos mesmos: projetos desenvolvidos e currículum vitae.

A Instituição irá ser a entidade gestora da aplicação, administrando o sistema de software assim como os alunos e empresas registados no mesmo. Existe também a opção de contratar uma equipa de administração do sistema, os Alumínios.

Desta forma apresentamos o UAW, concebido para oferecer uma solução abrangente que vai além das redes sociais convencionais, promovendo uma comunidade virtual centrada no conhecimento e na aprendizagem colaborativa, oferecendo uma forma simples, segura e fidedigna às empresas para recrutar alunos nas suas áreas de interesse, complementando as necessidades de ambos no mundo laboral.

Mediante a nossa metodologia de trabalho direcionámos o nosso foco para as funcionalidades relacionadas com o mercado de trabalho, sendo que o protótipo demonstra o utilizador Empresa e as suas interações na aplicação.

Este documento apresenta a descrição geral do projeto e o seu conceito, assim como os concorrentes existentes, comparando as nossas funcionalidades como sendo únicas e úteis aos

atores envolvidos. Apresentamos exemplos da sua utilização, referindo o processo de desenvolvimento, ferramentas utilizadas, desafios encontrados e visão futura. Focando-se no objetivo do nosso projeto, a criação de uma interface web moderna e responsiva utilizando ferramentas de: Front-End, ReactJS, uma robusta camada de Back-End com Java Spring, e uma base de dados eficiente com MariaDB. Uma plataforma onde os utilizadores conectam-se e compartilham os seus conteúdos de forma segura e escalável, com uma experiência intuitiva e otimizada, garantindo ao mesmo tempo a integridade e segurança dos dados.

Além disso, damos a conhecer as vantagens e desvantagens presentes no sistema, face a um concorrente com funcionalidades semelhantes, tanto para os utilizadores finais quanto para potenciais parceiros e investidores. Por fim, exploraremos as oportunidades de crescimento e prospeção para o UAW, considerando o seu potencial impacto no cenário educacional e tecnológico.

1.2. Motivação

Com o advento da era digital, as redes sociais têm desempenhado um papel fundamental na conectividade e na disseminação de informações em diversos setores da sociedade. Reconhecendo a necessidade de uma plataforma específica para promover a interação e o compartilhamento de conhecimento dentro e fora do ambiente universitário, surge o projeto UAW, sendo mais do que uma simples rede social, é um sistema de software projetado exclusivamente para a comunidade universitária e empresas, visando fortalecer laços académicos, facilitar a colaboração entre entidades e enriquecer a experiência educacional e profissional.

A nossa principal motivação para a criação da aplicação UAW foi a experiência vivenciada ao longo do nosso percurso académico, pela falta de comunicação não formal na instituição académica, assim como a inexistência de acesso simples e transparente a oportunidades de estágio e a ofertas de emprego, tornando-se necessário recorrer a outras plataformas digitais para efetuar a pesquisa de oportunidades no mercado de trabalho.

Desta forma verificámos o quanto seria importante existir uma plataforma como o UAW ao nosso alcance, solucionando diversas lacunas na comunicação académica, sem depender de grupos de WhatsApp ou da plataforma institucional para que haja comunicação permanente e sem termo entre os alunos, assim como a visibilidade dos mesmos para com as empresas, ainda antes de os alunos terminarem os seus cursos.

A motivação para a escolha das tecnologias utilizadas baseia-se em vários fatores, tais como:

- **Popularidade e Comunidade:** ReactJS, Java Spring e MariaDB possuem comunidades vastas e ativas, proporcionando suporte contínuo e uma vasta quantidade de recursos e bibliotecas.
- **Eficiência e Desempenho:** ReactJS permite a criação de interfaces de utilizadores altamente eficientes e responsivas. Java Spring é conhecido pela sua robustez e escalabilidade, enquanto MariaDB oferece uma gestão de banco de dados rápido e seguro.
- **Experiência Pessoal:** O uso destas tecnologias permitirá a aplicação prática dos conhecimentos adquiridos durante o curso, consolidando a aprendizagem.

Outra das maiores motivações é o facto da sua aplicabilidade ter a capacidade de escalar, adquirir diversas funcionalidades e ir além da ideia inicial, como por exemplo adaptar-se a entidades organizacionais, interligar universidades, ou seja, qualquer meio que haja partilha de informação e diferentes pontos de acesso é o candidato ideal para o UAW, ligando todos de forma simples e eficaz!

Assim, com foco na inovação e um compromisso com a excelência educacional, o UAW representa uma promessa emocionante para transformar a experiência universitária, promovendo a colaboração e o compartilhamento de informações a um nível sem precedentes.

1.3. Objetivos do Projeto

O principal objetivo do projeto UAW é criar uma plataforma digital que promova uma interação contínua e eficaz entre os membros da comunidade académica e as empresas. Queremos fornecer uma solução que não só facilite a comunicação informal, mas também permita aos alunos apresentarem os seus portfólios de projetos de forma organizada e acessível. Desta forma, as empresas podem avaliar as competências dos alunos de maneira mais eficiente, facilitando o processo de recrutamento antes mesmo da conclusão dos seus cursos.

Outro objetivo importante é incentivar a colaboração entre alunos de diferentes áreas, criando um ambiente virtual onde possam partilhar conhecimentos e criar contatos. A UAW visa também fortalecer os laços entre os ex-alunos e as suas antigas instituições, promovendo o *networking* e a mentoria.

Através da UAW, pretendemos fornecer às empresas uma ferramenta simples, segura e confiável para recrutar talentos emergentes, enquanto damos aos alunos a oportunidade de se destacarem e de se prepararem melhor para o mercado de trabalho. A plataforma foi concebida para ser intuitiva e fácil de usar, com uma interface amigável que permite aos utilizadores navegarem e utilizarem todas as funcionalidades de forma eficiente.

1.4. Estrutura do Trabalho

No Capítulo II, será apresentada a fundamentação teórica e conceptual do projeto. Este capítulo irá explorar as bases teóricas que sustentam o desenvolvimento da plataforma UAW, incluindo os princípios de redes sociais aplicados ao contexto académico e as metodologias de design de interface e experiência do utilizador (UI/UX). Além disso, será realizada uma análise das soluções concorrentes, destacando as suas funcionalidades, pontos fortes e fracos, com o intuito de posicionar a nossa solução no mercado.

O Capítulo III abordará a visão macro do projeto, explicando em detalhe a nossa solução, UAW, descrevendo as suas funcionalidades, o fluxo de trabalho e as tecnologias utilizadas, assim como os requisitos funcionais e não funcionais do sistema criado, que por sua vez satisfazem as necessidades do sistema para atingir o objetivo pretendido. Demonstramos as suas funcionalidades globais, apresentando também os casos de uso e os diagramas que ilustram as principais interações do utilizador escolhido para demonstração: Utilizador Empresa. Seguindo-se a apresentação da arquitetura e tecnologias utilizadas no seu desenvolvimento, assim como também uma análise detalhada de algumas das funcionalidades mais importantes e uma visão geral da interface UI/UX.

Complementamos este capítulo com a análise de soluções concorrentes e a análise comparativa, realizando uma comparação direta com o concorrente LinkedIn, recorrendo à ferramenta de análise SWOT (Forças, Fraquezas, Oportunidades e Ameaças) para identificar os pontos chave do projeto e avaliar a sua viabilidade e potencial de sucesso no mercado académico e profissional, face ao concorrente em questão.

No Capítulo IV, faremos uma demonstração prática de um caso de uso do sistema, apresentando um exemplo de interação do utilizador Empresa com a UAW, ilustrado com capturas de ecrã e respetiva legenda associada à descrição da funcionalidade utilizada, descrita no capítulo III.

Por fim, no Capítulo V, apresentaremos a Conclusão e Visão Futura do trabalho realizado. Esta secção refletirá sobre os resultados obtidos, os desafios enfrentados e as aprendizagens adquiridas ao longo do desenvolvimento do projeto. Serão também discutidas as perspetivas e ideias futuras para a evolução da plataforma UAW, acompanhadas de alguns exemplos.

2. CAPÍTULO: FUNDAMENTAÇÃO TEÓRICA E ANÁLISE COMPARATIVA

2.1. Fundamentação Teórica e Conceptual

2.1.1. Metodologia de desenvolvimento

O desenvolvimento do UAW seguiu a estrutura *DevOps*, que integra desenvolvimento (*Dev*) e operações (*Ops*) para melhorar a colaboração entre a equipa. A estrutura de desenvolvimento *DevOps* abrange várias fases: requisitos, design, desenvolvimento, implementação e testes.

Requisitos: Nesta fase, foram identificados e documentados os requisitos do sistema, para garantir que todas as necessidades fossem contempladas. Para a UAW, isso incluiu funcionalidades como a criação de perfis, compartilhamento de portfólios, sistema de pesquisa, segurança e escalabilidade da plataforma.

Design: A fase de design envolveu a criação de diagramas de arquitetura do sistema, design de interfaces de utilizador (UI) e experiência do utilizador (UX). O objetivo foi garantir que a estrutura do sistema fosse robusta e escalável, e que a interface do utilizador fosse intuitiva e responsiva.

Desenvolvimento: Durante o desenvolvimento, a equipa utilizou o GitHub para controle de versão e organização do código. O desenvolvimento foi orientado a componentes para promover a reutilização de código e facilitar a manutenção. As tecnologias escolhidas foram Spring Framework para o Back-End e ReactJS para o Front-End, aproveitando as suas respetivas capacidades de robustez e eficiência.

Implementação: A implementação do projeto foi simulada como se a universidade fosse o Instituto Piaget, porém, a aplicação está preparada para ser implementada em qualquer universidade ou organização.

Testes: Os testes foram conduzidos em várias etapas, incluindo testes unitários, de integração e de sistema, para garantir que todos os componentes funcionassem corretamente de forma isolada e integrada. Ferramentas de automação foram utilizadas para realizar testes contínuos, identificando e corrigindo bugs rapidamente.

2.1.2. Princípios de UI/UX Design

O design da *UI/UX* seguiu princípios que garantem uma experiência do utilizador intuitiva, eficiente e agradável. Foram considerados fatores como a facilidade de navegação, consistência visual, feedback imediato, e acessibilidade. O objetivo foi criar uma interface que facilitasse a interação dos utilizadores com a plataforma, promovendo uma experiência positiva e envolvente.

2.1.3. Análise Comparativa

Para posicionar a UAW no mercado, foi realizada uma análise comparativa com outras aplicações concorrentes. Foram avaliadas as funcionalidades, pontos fortes e fracos dessas soluções para identificar oportunidades de melhoria e diferenciação para a UAW. As funcionalidades únicas e o foco nas necessidades específicas da comunidade académica destacam a UAW como uma solução inovadora e essencial.

3. CAPÍTULO: VISÃO MACRO DO PROJETO

3.1. Explicação da Solução Proposta

O UAW (U Are Welcome) é uma plataforma digital projetada para transformar a interação e colaboração entre alunos, empresas e instituições académicas. A aplicação oferece uma interface moderna e intuitiva, integrando diversas funcionalidades para atender às necessidades dessa comunidade. Através da UAW, os utilizadores podem conectar-se, compartilhar conhecimento e experiências, e facilitar a transição dos alunos para o mercado de trabalho, promovendo um ambiente colaborativo e enriquecedor.

A ideia central da UAW é criar um ecossistema onde a comunicação e a colaboração sejam simplificadas e eficazes. A plataforma permite que os alunos criem perfis, onde podem destacar as suas competências, experiências e realizações académicas e profissionais. Estes perfis funcionam como um currículo digital, acessível a empresas e instituições que procuram talentos emergentes.

A possibilidade de criação de perfis é um dos pontos chave desta App. Os utilizadores podem criar perfis pessoais, incluindo informações sobre a sua formação académica, competências, projetos e experiências profissionais. Estes perfis são vitais para que as empresas e instituições possam identificar e recrutar talentos que se alinhem com as suas necessidades específicas de contratação.

Mas a criação de perfis, só por si não nos pareceu suficiente e por isso desenvolvemos também a possibilidade do aluno partilhar o seu portfólio de trabalhos realizados e curriculum vitae. Esta funcionalidade é especialmente útil para estudantes de áreas como design, engenharia e

tecnologia, onde a demonstração prática de competências é essencial, mas seguramente será bastante útil em qualquer área de estudos que o aluno esteja a desenvolver.

Para que seja possível tirar o melhor resultado prático das informações publicadas, a plataforma inclui ferramentas de pesquisa avançada que permitem às empresas encontrar candidatos com habilidades específicas e aos alunos encontrar oportunidades de estágio e emprego adequadas aos seus perfis. A pesquisa pode ser filtrada por diversos critérios, como área de estudo, localização, tipo de oportunidades, entre outros.

Atualmente, a ferramenta oferece ainda um sistema de divulgação de eventos, workshops, feiras de emprego e outras informações que as instituições académicas considerem relevantes, tornando toda a informação significativa neste ambiente propício à partilha e comunicação entre toda a comunidade académica. Os alunos passam a ter conhecimento dos eventos de forma mais rápida e fácil, enquanto as empresas e a instituição podem promover os seus próprios eventos, aumentando a visibilidade e a participação, aumentando o alcance do conteúdo dos envolvidos.

A UAW tem assim como objetivo promover um ambiente onde a colaboração é incentivada e ao mesmo tempo promove também a transição dos alunos para o mercado de trabalho fornecendo ferramentas e recursos que ajudam os estudantes a organizarem as suas carreiras.

A aplicação UAW é construída utilizando tecnologias modernas que garantem uma experiência de utilizador fluída e responsiva. O uso de frameworks, Front-End e Back-End robustos, permite a integração de várias funcionalidades sem comprometer o desempenho e garantindo a escalabilidade. Sendo a segurança dos dados dos utilizadores uma prioridade, a UAW foi construída tendo na sua base o uso de protocolos de segurança avançados, implementados para proteger informações sensíveis.

Por fim, o design da sua interface foi concebido com o cuidado necessário para que fosse possível ao utilizador ter uma experiência visualmente atraente, centrada no utilizador e com uma navegação intuitiva. Cores vibrantes e uma tipografia clara garantem que a aplicação seja não só funcional, mas também agradável de usar. A consistência visual é mantida através de um sistema de design coeso, que facilita a navegação e a utilização de todas as funcionalidades.

Por essas razões, concebemos a UAW como uma solução inovadora que tem o potencial de transformar a maneira como alunos, empresas e instituições académicas interagem. Com uma gama de funcionalidades que promovem a colaboração, a partilha de conhecimento e a preparação para o mercado de trabalho, o UAW posiciona-se como uma ferramenta indispensável para a comunidade académica. A combinação de uma interface intuitiva, funcionalidades robustas e um foco claro na experiência do utilizador, garantem que a plataforma não só atende às necessidades atuais, mas também está preparada para evoluir e crescer com as necessidades futuras da educação e do mercado profissional.

3.2. Requisitos do Sistema

Para garantir que o sistema UAW – U ARE WELCOME atenda às necessidades dos utilizadores e funcione de forma eficiente e segura, é essencial definir os requisitos do sistema. Estes requisitos estão divididos em Funcionais e Não Funcionais, cobrindo todas as funcionalidades e aspectos técnicos necessários para o sucesso do projeto.

3.2.1. Requisitos Funcionais

3.2.1.1 Gestão de utilizadores

Registo e Autenticação:

- Alunos, empresas e administradores devem poder criar contas.
- O sistema deve suportar autenticação por e-mail e senha.

Perfis de Utilizadores:

- Alunos devem poder criar e editar perfis com informações pessoais, académicas e profissionais.
- Empresas devem poder criar e editar perfis com informações sobre a organização, áreas de interesse e oportunidades disponíveis.
- Administradores devem poder visualizar e gerir perfis de todos os utilizadores.

Portfólios de Alunos:

- Os Alunos devem poder adicionar, editar e remover projetos, CVs e outros materiais ao seu portfólio.
- As Empresas devem poder visualizar os portfólios de alunos.

3.2.1.2 Gestão de oportunidades

Ofertas de Emprego e Estágios:

- Empresas devem poder publicar ofertas de emprego e estágio.
- Alunos devem poder procurar e candidatar-se a essas ofertas.

Sistema de Correspondência:

- Implementação de um sistema que sugira vagas de acordo com o perfil e interesses dos alunos.
- Notificações de novas oportunidades relevantes.

3.2.1.3 Administração do Sistema

Gestão de Utilizadores:

- Administradores devem poder aprovar, suspender ou excluir contas.
- Gestão de permissões e roles dos utilizadores (aluno, empresa, administrador).

Monitoramento e Relatórios:

- Monitoramento da atividade na plataforma (login, publicações, interações).

Configurações da Plataforma:

- Administradores devem poder configurar parâmetros do sistema, como política de privacidade, termos de uso, etc.

3.2.2. Requisitos Não Funcionais

3.2.2.1 Desempenho

Tempo de Resposta:

- A plataforma deve responder a qualquer solicitação em até 2 segundos.

Escalabilidade:

- O sistema deve suportar até 10.000 utilizadores simultâneos, com capacidade de escalabilidade horizontal.

3.2.2.2. Segurança

Proteção de Dados:

- Dados dos utilizadores devem ser criptografados em repouso e em trânsito.
- Implementação de políticas de privacidade rigorosas em conformidade com o GDPR.

Autenticação e Autorização:

- Uso de OAuth 2.0 para gestão de autenticação e autorização.
- Implementação de controle de acesso baseado em roles (RBAC).

3.2.2.3. Confiabilidade

Disponibilidade:

- A plataforma deve ter uma disponibilidade de 99.9%, com planos de contingência e recuperação de desastres.

Backup:

- Realização de backups diários dos dados do sistema.

3.2.2.4. Usabilidade

Interface Intuitiva:

- A interface deve ser intuitiva, fácil de navegar e acessível para todos os utilizadores.
- Design responsivo para suportar dispositivos móveis e desktop.

Acessibilidade:

- Conformidade com os padrões WCAG 2.1 para garantir acessibilidade a utilizadores com deficiência.

3.2.2.5. Manutenibilidade

Código Modular:

- O sistema deve ser desenvolvido de forma modular para facilitar a manutenção e futuras atualizações.

Documentação:

- Documentação completa do código, APIs e guia de utilizador.

A definição dos requisitos funcionais e não funcionais é fundamental para o desenvolvimento de uma plataforma robusta e eficiente como a UAW. Esses requisitos garantem que todas as funcionalidades necessárias sejam contempladas, enquanto aspectos como desempenho, segurança, confiabilidade, usabilidade e manutenibilidade são assegurados, resultando em uma solução que atende às necessidades da comunidade académica e empresarial de forma eficaz.

3.3. Demonstração das Funcionalidades Globais

Perfil de Utilizadores Personalizado:

- **Aluno:** Permite que os alunos criem perfis detalhados, incluindo informações pessoais, partilhando o curriculum vitae, portfólios de projetos, certificações, conquistas académicas e experiências extracurriculares. Os perfis são focados tanto em aspectos académicos quanto profissionais. Podendo também realizar a função de pesquisa mediante os seus interesses.
- **Empresa:** Empresas podem criar perfis para exibir informações sobre as suas atividades, cultura corporativa e oportunidades de estágio ou emprego. Elas têm acesso a uma base de dados de perfis de alunos, facilitando a pesquisa por candidatos adequados.
- **Instituição Académica:** A instituição atua como administradora da plataforma, gerindo o conteúdo e os utilizadores. Pode compartilhar informações institucionais, regulamentações, eventos e oportunidades de desenvolvimento académico.

3.4. Casos de uso e Diagramas

Demonstramos como a solução proposta funciona na prática, através do Estudo de Caso, ou seja, a apresentação de um exemplo prático do Utilizador Empresa.

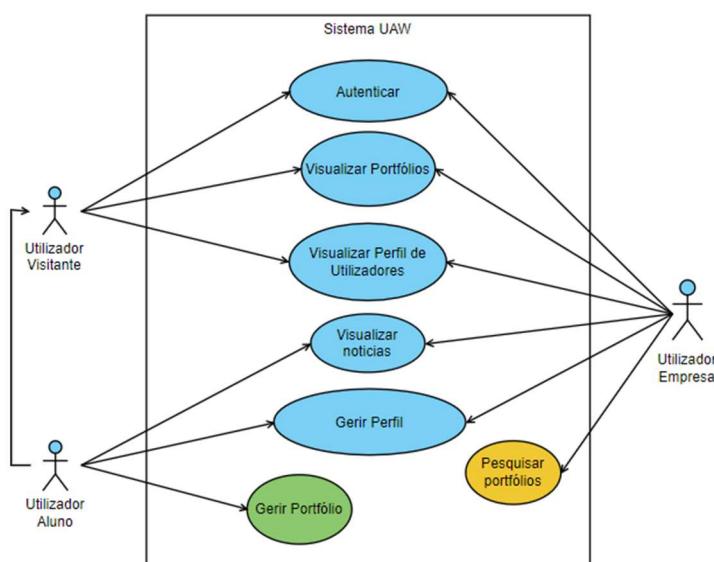


Figura 1 - Use Case Sistema UAW

3.4.1 Descrição Use Cases – Figura 1

- **Visualização de portfólios:** Qualquer utilizador consegue abrir o portfólio de um aluno, para visualizar os projetos que o aluno desenvolveu.
- **Visualização de notícias:** Qualquer utilizador consegue ver as notícias sobre a Universidade.
- **Autenticação:** Ação de início de sessão para aceder á aplicação.
- **Visualização de perfis de utilizadores:** Qualquer utilizador consegue abrir e ver os dados disponibilizados no perfil de um Utilizador Aluno ou Utilizador Empresa.
- **Preenchimento de perfil:** Página onde o utilizador Aluno e o utilizador Empresa conseguem preencher com os seus dados pessoais e fazer uma breve introdução sobre quem são e o que fazem.
- **Gerir Portfólio:** Esta ação é uma característica comprehensiva que permite o Utilizador Aluno demonstrar a sua capacidade académica através de projetos extracurriculares ou até projetos desenvolvidos no decurso das disciplinas da universidade. Esta funcionalidade é feita para o utilizador Aluno conseguir apresentar sistematicamente as suas capacidades e experiências obtidas no desenvolvimento de projetos, melhorando assim o perfil profissional do utilizador Aluno.
- **Pesquisa de Portfólio:** Esta ação é exclusiva para o utilizador Empresa, permite uma pesquisa eficiente de perfis de utilizadores Alunos baseado em *Tags* mais procuradas, ou na escrita de texto livre no campo de procura. Esta funcionalidade facilita a criação de conexões entre os alunos e potenciais empregadores.

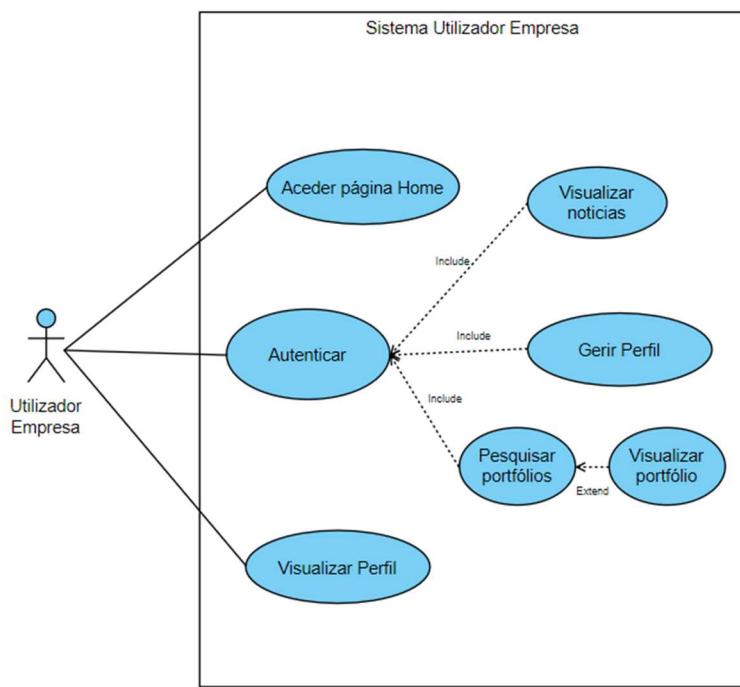


Figura 2 - Use Case Utilizador Empresa

O Use Case do "Utilizador Empresa" (Figura 2) permite que o mesmo aceda à página inicial, visualizando as notícias institucionais, gerir e visualizar os seus perfis, e pesquisar portfólios. É obrigatório autenticar-se para aceder às funcionalidades restritas, assegurando que apenas utilizadores registados possam gerir informações guardadas no perfil do utilizador. Esta abordagem garante a segurança e personalização da experiência do utilizador dentro do sistema.

3.4.1.1 Funcionalidade: Visualização de notícias após autenticação de utilizador

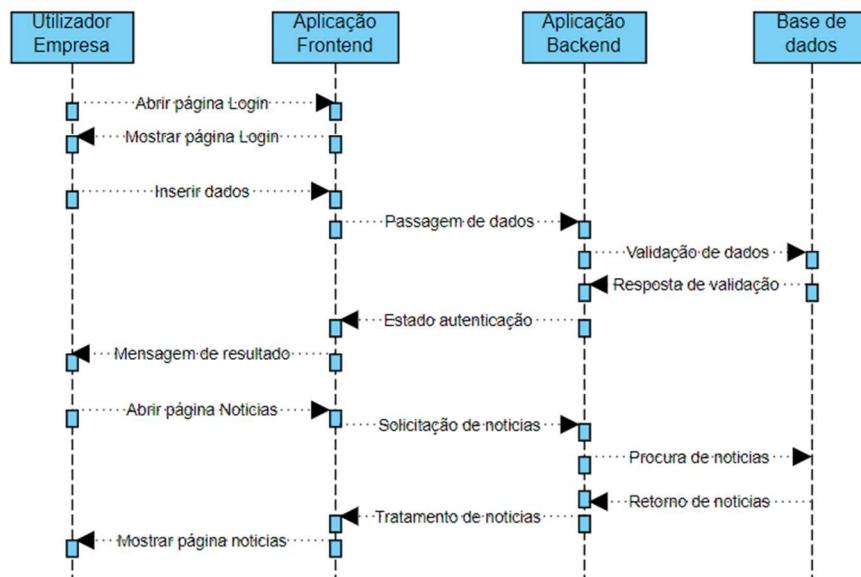


Figura 3 - Visualização de notícias após autenticação de utilizador

O diagrama de sequência (Figura 3) mostra o processo de autenticação e visualização de notícias no sistema "Utilizador Empresa". Inicialmente, o utilizador abre e insere dados na página de login através do Front-End, que passa esses dados para o Back-End. O Back-End valida os dados com a base de dados e retorna o estado de autenticação. Após a autenticação, o utilizador solicita a visualização de notícias, com o Front-End encaminhando esta solicitação para o Back-End. O Back-End consulta a base de dados para obter as notícias e retorna os dados para o Front-End, que, por sua vez, exibe as notícias ao utilizador. Este processo garante a autenticação segura e a apresentação de informações atualizadas ao utilizador.

3.4.1.2 Funcionalidade: Pesquisa e visualização de Portfólios

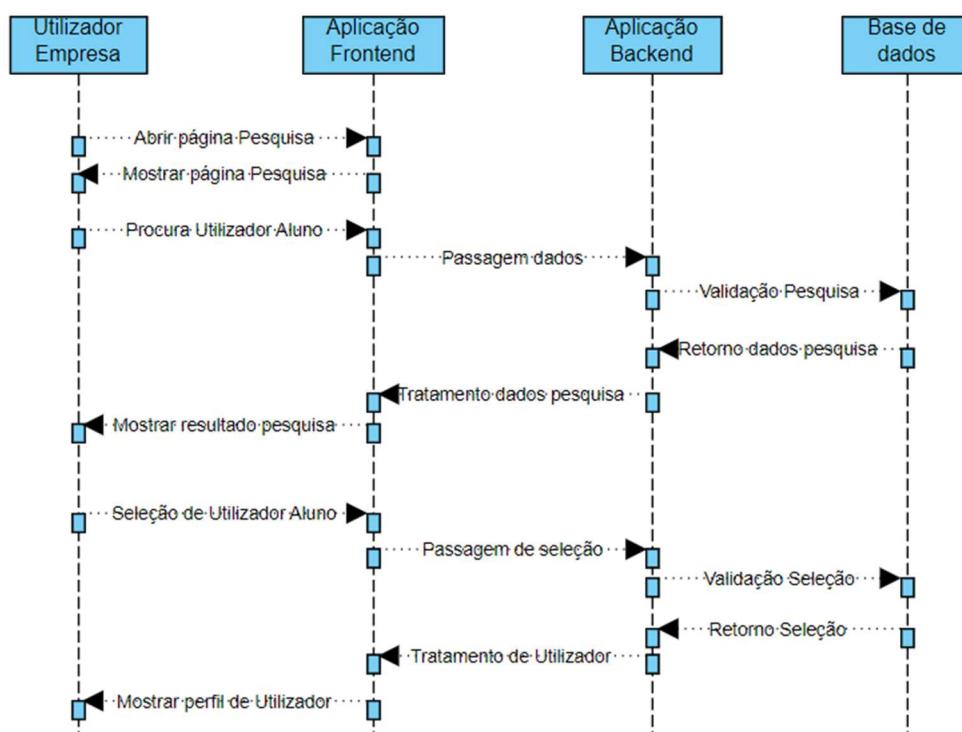


Figura 4 - Pesquisa e visualização de Portfólios

O diagrama de sequência (Figura 4) mostra o processo de pesquisa e seleção de utilizadores no sistema "Utilizador Empresa". Primeiro, o utilizador abre a página de pesquisa e insere os critérios para procurar um Utilizador Aluno no Front-End. O Front-End passa os dados da pesquisa para o Back-End, que valida a pesquisa na base de dados e retorna os Utilizadores encontrados para o Front-End. Em seguida, o utilizador seleciona um Utilizador Aluno dos resultados da pesquisa. O Front-End encaminha esta seleção para o Back-End, que valida a seleção e retorna os dados do utilizador selecionado. Finalmente, o Front-End exibe o perfil do utilizador selecionado. Este

processo assegura a pesquisa e a visualização detalhada dos perfis de utilizadores, proporcionando uma interação eficiente e segura com a base de dados.

3.4.1.3 Funcionalidade: Visualização e gestão de perfil

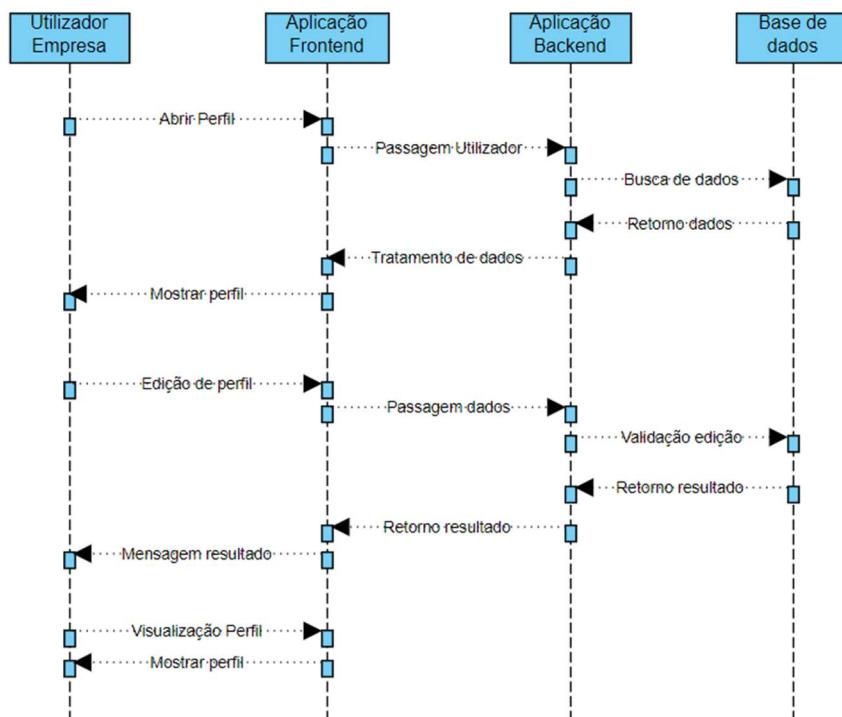


Figura 5- Diagrama Utilizador Empresa

O diagrama de sequência (Figura 5) ilustra o processo de abertura, edição e visualização do perfil do utilizador no sistema "Utilizador Empresa". Primeiro, o utilizador abre a página de perfil através do Front-End, que passa os dados do utilizador para o Back-End. O Back-End procura os dados na base de dados e retorna a informação para o Front-End, que mostra o perfil ao utilizador. Para editar o perfil, o utilizador insere as alterações no Front-End, que envia esses dados ao Back-End para validação e atualização na base de dados. Após a validação e atualização, o Back-End retorna o resultado da edição para o Front-End, que mostra uma mensagem de sucesso ao utilizador. Finalmente, o utilizador pode visualizar o perfil como Utilizador externo. Este processo assegura que os dados do perfil são atualizados corretamente e refletidos de forma precisa ao utilizador.

3.5. Tecnologia e Arquitetura do Sistema

A arquitetura do sistema UAW foi baseado na arquitetura MVC (*Model View Controller*), um padrão de arquitetura de software que divide a aplicação em três camadas: manipulação dos dados (*model*); interação do utilizador (*view*); camada de controle (*controller*). De forma a garantir robustez, escalabilidade e segurança, além de proporcionar uma experiência de utilizador fluída e responsiva. Abaixo, apresentamos uma descrição detalhada dos principais componentes e camadas da arquitetura:

3.5.1. Camada de Apresentação (Front-End) - Ver Anexo 2

- **Tecnologia Utilizada: ReactJS**
- **Descrição:** A interface de utilizador do UAW é construída com ReactJS, uma biblioteca JavaScript popular desenvolvida pelo Facebook, para a construção de interfaces de utilizador interativas e dinâmicas. ReactJS permite o desenvolvimento de componentes reutilizáveis e uma renderização eficiente, garantindo uma experiência de utilizador suave e responsiva em diferentes dispositivos (desktop e mobile).
- **Funcionalidades:**
 - **Interface Responsiva:** Design adaptativo para diferentes tamanhos de ecrã e dispositivos.
 - **Componentes de Interface:** Elementos interativos como formulários, menus de navegação, painéis de controle e *dashboard*.
 - **Gerenciamento de Estado:** Uso de bibliotecas como por exemplo *Redux* para gerir o estado da aplicação de forma centralizada.

3.5.2. Camada de Serviço (Back-End) - Ver Anexo 3

- **Tecnologia Utilizada: Java Spring Framework**
- **Descrição:** O Back-End do UAW é implementado usando o *Spring Framework*, uma plataforma robusta para o desenvolvimento de aplicações Java. O Spring proporciona uma arquitetura modular e flexível, facilitando a implementação de serviços *RESTful* e garantindo a alta escalabilidade da aplicação, podendo suportar a implementação empresarial da mesma.
- **Funcionalidades:**
 - **APIs RESTful:** Exposição de end-points para operações *CRUD* (*Create, Read, Update, Delete*) e outras funcionalidades da aplicação.
 - **Autenticação e Autorização:** Implementação de segurança baseada em JWT (JSON Web Tokens) para proteger os recursos e dados da aplicação.
 - **Gerenciamento de Sessões:** Controle de sessões de utilizadores e autenticação segura.

- **Escalabilidade:** Permite com que um grande fluxo de dados seja tratado ao mesmo tempo, permitindo a implementação do sistema com maior facilidade em larga escala.
- **Anotações:** Spring é utilizado por grandes empresas por ser possível definir níveis de acesso facilmente, simplificar operações básicas de programação, gestão da base de dados no próprio código, configurações de segurança, *Cloud*, Base de dados, entre outros, facilitada, ambientes de testes seguro e qualidade do código requisitado.

3.5.3. Camada de Dados (Banco de Dados) - Ver Anexo 4

- **Tecnologia Utilizada:** MariaDB
- **Descrição:** A camada de dados do UAW utiliza MariaDB, um sistema de gestão de banco de dados relacional que oferece alta performance, segurança e confiabilidade. MariaDB é uma escolha popular para aplicações web devido à sua compatibilidade com MySQL e capacidade de escalar conforme necessário.
- **Funcionalidades:**
 - **Modelagem de Dados:** Estruturação de tabelas para armazenar dados de utilizadores, perfis, projetos, interações e mais.
 - **Consultas Otimizadas:** Uso de índices e otimização de consultas para garantir respostas rápidas e eficientes.
 - **Backup e Recuperação:** Implementação de estratégias de backup e recuperação de dados para garantir a integridade e a disponibilidade.

3.5.4. Segurança - Ver Anexo 6

- **Descrição:** A segurança é um componente transversal da arquitetura do sistema, abrangendo todas as camadas.
- **Funcionalidades:**
 - **Controles de Acesso:** Implementação de políticas de controle de acesso para diferentes tipos de utilizadores (alunos, empresas, administradores).
 - **Monitoramento e Auditoria:** Ferramentas de monitorização para deteção de atividades suspeitas e auditoria de ações dos utilizadores.

3.5.5. Infraestrutura e Deployment - Ver Anexo 7

- **Descrição:** A infraestrutura do UAW é projetada para suportar a alta disponibilidade e escalabilidade, utilizando serviços de *cloud computing*.
- **Funcionalidades:**
 - **Deployment em Nuvem:** Hospedagem em provedores de *cloud* como AWS, Azure ou Google Cloud.

- **Balanceamento de Carga:** Distribuição de carga de trabalho entre múltiplos servidores para otimizar o desempenho.

A arquitetura do UAW é projetada para suportar uma plataforma robusta, segura e escalável, capaz de atender às necessidades específicas da comunidade académica. A combinação de uma interface de utilizador moderna com um Back-End robusto e um banco de dados eficiente proporciona uma base sólida para a expansão futura e a integração de novas funcionalidades.

3.6. Funcionalidades detalhadas

Conforme os Use Cases apresentados no Tópico 3.4., detalhamos as suas funcionalidades, mencionando as figuras e os anexos referentes.

3.6.1. Splash / Aguarda carregamento – Figura

Este ecrã é o primeiro a ser visível quando se acede à aplicação ou, quando o tempo de carregamento da página para a qual o utilizador está a navegar, devido a questões de comunicação com os servidores da UAW é mais demorada do que o esperado.

Este ecrã é apenas composto por uma mensagem de boas-vindas e uma vídeo animação com o símbolo “U” da marca UAW a ser revelado.

Num mundo ideal este ecrã dificilmente seria visível pelos utilizadores, no entanto foi desenvolvido para dar principalmente um feedback visual ao utilizador de que está a haver interação.

3.6.2. Início de Sessão - Erro! A origem da referência não foi encontrada.

O seu acesso pode ser feito diretamente através de botões de ação de navegação, como exemplificado no ecrã “Home”, ou através do acesso a uma área autenticada, caso ainda não tenha sido feita a autenticação do utilizador previamente.

Neste ecrã são apresentados o logo da UAW, campos de preenchimento, a opção de recuperação de palavra-passe, ir para o ecrã de registo e por último, opções de autenticação de terceiros.

É feita também uma pré-validação dos dados, ou seja, se os campos estão vazios, se são do tipo pretendido, exemplo, email ter o formato de email, e só após esta validação é possível ser feito o pedido de verificação de autenticação ao Back-End.

3.6.3. Recuperação de Password – Figura 26, Anexo 2 – Front-End

É composto por elementos simples, o logo da UAW, caixa de texto para inserção do email e botão de recuperação. Existe também a possibilidade de o utilizar voltar ao ecrã de Início de Sessão e tentar novamente a autenticação.

A utilidade desta funcionalidade é ser possível ao utilizador iniciar o procedimento de recuperação de password, desde que para isso se recorde do email que foi utilizado no momento do registo inicial, bem como tenha acesso a esse email para poder concluir o procedimento de recuperação.

3.6.4. Registo de Utilizadores: Empresa – Figura 20 | Aluno – Figura 20, Anexo 2 – Front-End

É feita a introdução de um “componente personalizado” ReactJS, o “Header”, este componente não é uma biblioteca, foi desenvolvido, precisamente devido à necessidade de ser ou não necessário o mesmo estar visível, ou eventuais necessidades futuras comuns a diversos ecrãs ou mesmo à aplicação na sua totalidade. No sentido da premissa do ReactJS, reutilização de código, o “Header” foi desenvolvido com esse intuito em mente.

Para além do “Header”, temos um formulário que permite ao utilizador o preenchimento de alguns campos. Esses campos podem ser diferentes, consoante se trate de um registo de utilizador aluno ou de um utilizador empresa.

Mais uma vez consoante o tipo de registo, existe uma ligeira alteração visual, nomeadamente nas cores, de modo a alertar o utilizador para a ação que está a executar.

No final existe também a integração com plataformas terceiras, para ser feito o registo de uma conta na aplicação UAW.

Está também disponível a possibilidade de navegar para o ecrã de Início de Sessão caso o utilizador já tenha credências de acesso registadas na plataforma.

Sempre que a quantidade de informação ou elementos visíveis, de certa forma seja superior à área visível vertical, os ecrãs da plataforma permitem o *scroll* vertical.

3.6.5. Perfil – Empresa – Figura 21 | Aluno – Figura 21, Anexo 2 – Front-End

O ecrã do “Perfil” é possível de ser visualizado quer o utilizador esteja ou não autenticado. Nele é possível consultar o perfil do utilizador ou empresa que se pretende mediante a utilização do link correto para aceder ao mesmo, nomeadamente “/profile/{username}” por exemplo que irá apresentar o perfil do utilizador {username}.

A diferença deste ecrã entre o utilizador estar ou não autenticado, tem a ver maioritariamente com um novo componente “Footer” que à semelhança do que vimos anteriormente para o “Header” é um componente/área que pode não estar visível em todos os ecrãs. No caso o “Footer” por defeito nas áreas não autenticadas irá sempre ter apenas visível um botão de ação que remete para o ecrã de Início de Sessão, permitindo ao visitante se possa autenticar. No caso de já estar autenticado, o “Footer” irá ter outro tipo de opções, que irá depender do tipo de utilizador autenticado.

Neste ecrã de “Perfil” é possível consultar as informações de um determinado utilizador Aluno ou Empresa, registados na plataforma. No caso do Aluno a melhor comparação é a semelhança com um cartão virtual, no caso da Empresa é semelhante à ficha de dados da empresa, para consulta.

3.6.6. Procedimento de Autenticação - Empresa – Figura 25 | Aluno – Figura 25, Anexo 2 – Front-End

Procedimento tecnológico, o Front-End comunica com o Back-End no momento da autenticação, com o propósito de validar se o utilizador está registado, forneceu os dados corretos e se pode aceder. Em caso afirmativo, é feito um *hand-shake* entre o Front-End e o Back-End, onde este último fornece uma *Cookie JWT*, que vem encriptado e contém no seu interior 3 dados fundamentais para os diversos ecrãs das áreas “Autenticados”, que são o ID, *Username* e o Tipo de Utilizador, no caso ID corresponde ao ID do utilizador na base de dados, o *Username*, o que o utilizador escolheu no momento do registo e o Tipo de Utilizador corresponde se é Aluno ou Empresa.

3.6.7. Welcome – (Empresa e Aluno) - Figura 29

O ecrã desta funcionalidade é possível de ser visualizado apenas se tiver sido feita uma autenticação válida. O utilizador, independentemente do seu tipo após a autenticação é redirecionado para este ecrã para poder ser informado em certa medida das novidades/notícias/eventos/etc. que a instituição tenha em divulgação.

Os dados apresentados aqui, são da responsabilidade da Instituição Universitária, no entanto são disponibilizados em colaboração com a equipa de BackOffice da UAW, que mediante plataforma externa, que sai do escopo deste documento, efetua o carregamento na Base de Dados. Por sua vez a aplicação UAW recorrer ao Back-End para apresentar os mesmos ao utilizador autenticado.

É possível interagir com a publicação, caso a notícia/evento/etc., tiver um URL associado, clicar na área da mesma, onde o utilizador será redirecionado para esse mesmo URL.

Dependendo também do tipo de multimédia da notícia/evento/etc., pode ser reproduzido um vídeo ou uma imagem.

Este ecrã, à semelhança de todos os ecrãs de autenticação, tem na sua composição visual os componentes de “Header” e “Footer”.

Neste caso o “Footer” é composto por 3 opções de Ação, voltar ao “Welcome” (este ecrã), visualizar o “Portfólio” do utilizador Aluno autenticado, editar a sua ficha de “Perfil” na aplicação ou caso seja um utilizador do tipo Empresa a ação “Portfólio”, é substituída pela ação “Pesquisa”.

Visualmente, a ação a que corresponde o ecrã ativo (a ser visitado) a mesma estará realçada com a cor laranja.

3.6.8. Editar Perfil – Empresa – Imagem X | Aluno – Imagem X, Anexo X

Aqui é permitido ao utilizador Aluno ou Empresa alterar os dados que foram inseridos no momento da sua inscrição, bem como complementar dados adicionais.

Dependendo do tipo de utilizador haverá campos que não estarão visíveis.

O campo *Username* não é possível ser alterado, pois devido à sua especificidade para apresentação de certos ecrãs públicos (sem autenticação necessária), poderia gerar incoerência de dados ou gerar páginas não encontradas.

À semelhança de outros ecrãs, se a informação a ser apresentada for superior à área de visualização vertical é aplicado um *scroll* vertical.

3.6.9. Portfólio - Aluno – Figura 13

Neste ecrã é possível ao utilizador Aluno autenticado gerir o seu “Portfólio” na plataforma. Este “Portfólio” será possível de ser consultado publicamente quer por outros Alunos quer por Empresas.

A gestão do “Portfólio”, novas entradas, remoção e edição é feita mediante o clique ou *touch* sobre o respetivo ícone apresentado no ecrã.

Também neste ecrã caso a informação verticalmente seja superior à área visível o mesmo irá permitir *scroll* vertical.

São também apresentadas algumas informações do “Perfil” do utilizador a que respeita o “Portfólio”, bem como a possibilidade de consultar o “Perfil” na totalidade mediante um botão de ação.

Em seguida é apresentada a lista de Projetos de Portfólio que o utilizador introduziu.

É apresentada a opção de Voltar ao ecrã anterior, bem como os componentes “Header” e “Footer”.

3.6.10. Pesquisa – Empresa – Figura 12

Este ecrã é específico de utilizadores autenticados do tipo “Empresa”. Nele é possível efetuar uma pesquisa mediante termos livres ou opções de procura pré-definidas já disponíveis, por utilizadores do tipo Aluno que correspondam de alguma forma, quer pelo “Perfil” quer pelo “Portfólio” ao termo que está a ser utilizado como pesquisa.

O ecrã é composto também pelos componentes “Header” e “Footer”, e uma área central onde é possível fazer a escolha dos termos de pesquisa e apresentação de resultados.

Este ecrã requer autenticação, e se for tentado o seu acesso sem a mesma o utilizador é redirecionado para o ecrã de Login.

Na eventualidade de não existirem resultados a ser apresentados, o utilizador é informado mediante uma área de notificação.

3.7. Descrição Interface UI/UX

O sucesso de uma aplicação móvel, depende fortemente da eficácia do seu design gráfico UI/UX. Por essa razão, cada elemento gráfico foi escolhido e desenvolvido com todo o cuidado para garantir que a aplicação seja intuitiva, funcional e visualmente atraente para os seus utilizadores.



Figura 6 - Logótipo UAW

O logótipo da UAW (**Figura 6**) é a primeira impressão que os utilizadores têm da aplicação. Foi desenhado para transmitir uma sensação jovem, moderna e acolhedora, refletindo os valores da plataforma. A escolha de escala cinza e as formas no logótipo visam evocar uma sensação de

inovação e dinamismo, elementos cruciais para atrair a atenção dos estudantes universitários e mantê-los interessados na aplicação.

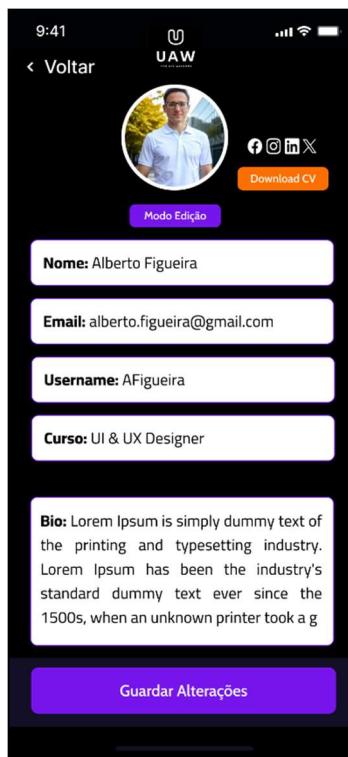


Figura 7 - Ecrã de Vista de Perfil de Utilizador no Front-End

A paleta de cores é fundamental para criar uma identidade visual coerente e agradável (**Figura 7**). Para a UAW, escolhemos cores que não só refletem a vitalidade e a energia da vida universitária, mas que também proporcionam um contraste adequado para garantir legibilidade e conforto visual. Cores vibrantes foram utilizadas para chamar a atenção para ações importantes, enquanto tons mais suaves foram usados para áreas de leitura e navegação, criando um equilíbrio visual agradável.

Selecionamos fontes modernas e limpas que facilitam a leitura em diferentes tamanhos de ecrã. A tipografia contribui para a identidade visual da UAW, transmitindo profissionalismo e acessibilidade, o que é fundamental para uma plataforma que busca conectar estudantes, empresas e instituições académicas.

Botões, ícones e outros elementos de interface foram desenhados para serem intuitivos e facilmente reconhecíveis. A utilização de ícones claros e consistentes ajuda os utilizadores a navegar pela aplicação de forma eficiente. Botões com feedback visual imediato garantem que os utilizadores saibam quando uma ação foi registada, melhorando a experiência de utilizador e reduzindo a frustração.

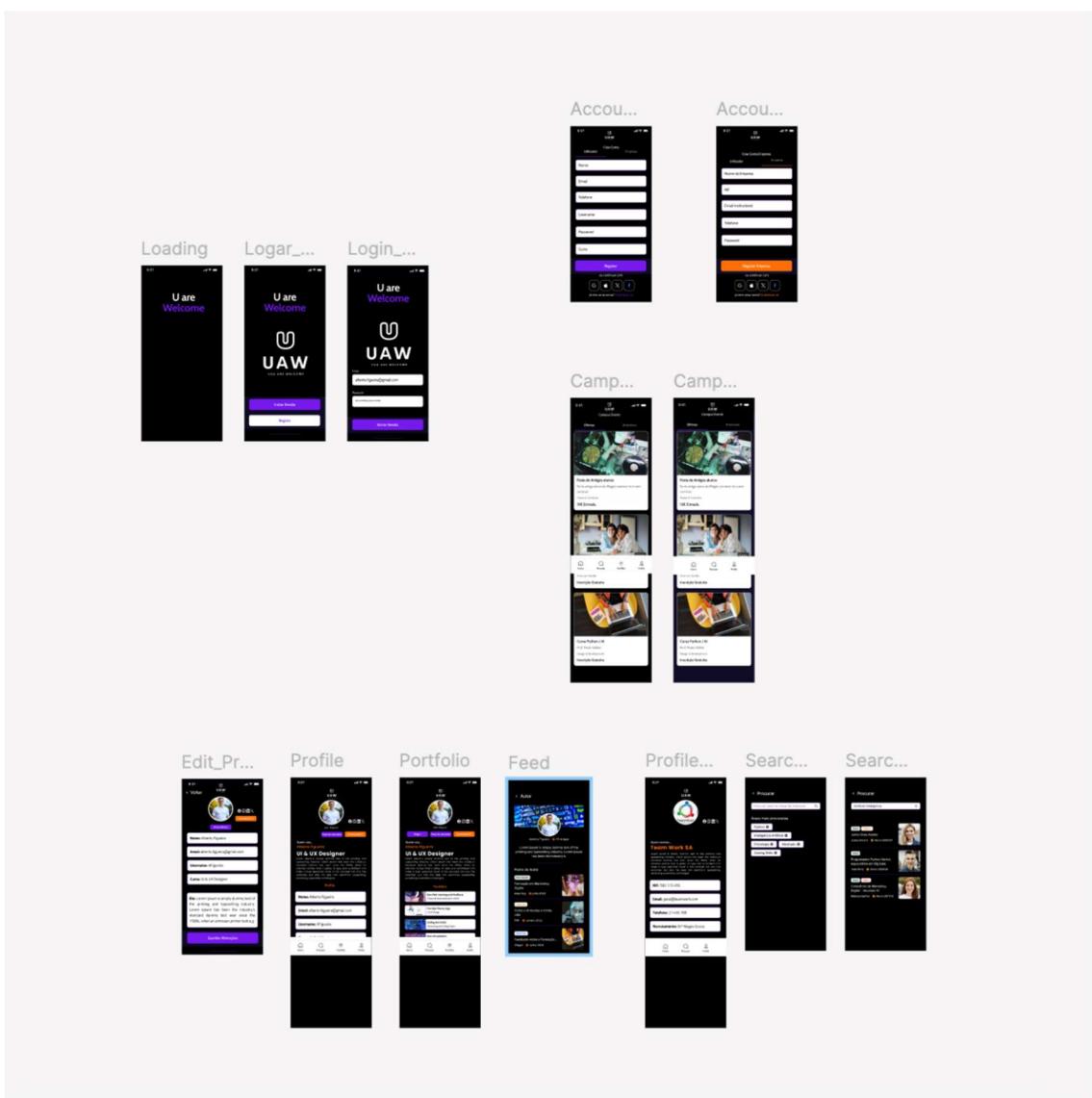


Figura 8 - Utilização da Ferramenta de Figma para prototipagem de UI/UX

Organizamos as informações de maneira clara e intuitiva, garantindo que os utilizadores possam encontrar o que procuram sem esforço. Este método também ajuda a manter a consistência visual em todas as páginas da aplicação, proporcionando uma experiência de utilizador coesa. Os protótipos interativos criados no Figma (**Figura 8**) foram fundamentais para testar e refinar a aplicação. Estes protótipos permitiram simular a navegação e as interações reais, fornecendo um ambiente seguro para identificar e corrigir problemas antes do lançamento e fornecerem insights valiosos para os refinamentos do design, assegurando que a aplicação atende às necessidades e expectativas dos utilizadores.

Cada elemento gráfico do UAW foi escolhido estratégicamente para contribuir para uma experiência de utilizador ótima. Desde o logótipo à paleta de cores, tipografia, componentes visuais e layout, cada decisão de design foi tomada para garantir que a aplicação seja não só funcional, mas também envolvente e intuitiva. Estes elementos gráficos desempenham um

papel crucial no sucesso do projeto, criando uma plataforma atraente e eficaz para a interação e colaboração dentro da comunidade académica.

3.8. Análise

3.8.1. Análise de Soluções Concorrentes

Apresentamos algumas soluções já existentes no mercado e que poderão ser mencionadas como concorrentes para a nossa solução, nas quais destacaremos os seus pontos fortes e os seus pontos fracos.

3.8.1.1. LinkedIn

Este talvez seja o produto que se apresenta como o concorrente mais óbvio. O LinkedIn destaca-se pelo seu foco no *networking* profissional, oferecendo uma plataforma para a construção de relações profissionais e oportunidades de emprego. A grande base de utilizadores profissionais e a integração com oportunidades de emprego são vantagens significativas. No entanto, este produto não é específico para estudantes universitários e falta-lhe ferramentas colaborativas específicas para educação, como por exemplo as notícias da instituição de ensino após a autenticação do utilizador e a interação entre empresa e instituição. A complexidade na interação social e a orientação de cariz profissional, podem não ser tão relevantes para a maioria dos estudantes universitários.

A solução LinkedIn não responde às necessidades específicas dos estudantes universitários e das instituições académicas de ensino superior.

3.8.1.2. Facebook

Esta plataforma possui uma enorme base de utilizadores, o que facilita a criação e manutenção de grupos com uma participação diversificada. A integração com outras funcionalidades do Facebook, como eventos, fotografias e mensagens, oferece uma experiência de utilização abrangente. Além disso, a interface é bastante intuitiva, uma vez que a maioria das pessoas já está familiarizada com o Facebook.

Realçamos, no entanto, mais uma vez, a falta de foco específico no ambiente universitário. A utilização desta plataforma para fins académicos, traduz-se em informação relevante misturada com todo o tipo de conteúdos irrelevantes resultando numa dispersão geral e negativa para os estudantes. A privacidade e segurança dos dados também são preocupações significativas, dado o histórico do Facebook em questões de privacidade.

3.8.1.3. Google Classroom

O Google *Classroom* é conhecido pela sua simplicidade de uso e pela fácil integração com o Google Drive, permitindo a partilha e gestão de ficheiros de forma eficiente. As ferramentas gratuitas para educação disponibilizadas pelo Google tornam-no uma opção acessível para muitas instituições de ensino.

Mas, o Google *Classroom* tem limitações nas funcionalidades de rede social, focando-se mais na gestão de tarefas e menos na interação social entre os utilizadores. Esta limitação pode reduzir a eficácia da plataforma em promover a comunidade de aprendizagem colaborativa que se pretende alcançar.

3.8.1.4. Microsoft Teams para Educação

O Microsoft Teams oferece uma robusta integração com o Microsoft Office 365, permitindo a utilização de ferramentas como Word, Excel e PowerPoint diretamente dentro da plataforma. As suas funcionalidades de colaboração, como chat, videoconferências e partilha de ficheiros, são particularmente úteis em contextos académicos. A presença de funcionalidades específicas para educação, como a integração com sistemas de gestão de aprendizagem (LMS), torna o Teams uma ferramenta poderosa.

É ainda a plataforma que já está implementada e em uso no Campus Universitário Jean Piaget, em que os alunos e professores utilizam para partilha de documentos e informações, complementando a plataforma Infoestudante.

Contudo, a complexidade de uso tem sido um obstáculo para alguns utilizadores, especialmente aqueles que não estão familiarizados com o ecossistema da Microsoft. A dependência de software proprietário e a curva de aprendizagem também podem ser barreiras significativas para a adoção generalizada. Assim como a limitação de acesso por questões financeiras dos alunos para com a universidade.

3.8.2. Análise comparativa com Concorrente

Depois da análise aos diversos produtos do mercado que se poderão apresentar mais próximos da solução UAW, e como tal com potencial para tornarem-se concorrentes diretos da nossa aplicação, realizámos uma análise comparativa recorrendo à ferramenta de análise SWOT (Forças, Fraquezas, Oportunidades e Ameaças), comparando a aplicação LinkedIn com o nosso UAW. É importante considerar as características e os objetivos de cada plataforma, especialmente no contexto de promover a interação entre alunos, empresas e instituições académicas.

3.8.2.1 Forças (Strengths)

LinkedIn:

- **Ampla Base de utilizadores:** Possui uma enorme rede global de profissionais e empresas, oferecendo uma vasta gama de conexões e oportunidades.
- **Reconhecimento de Marca:** É uma plataforma consolidada e amplamente reconhecida no mercado profissional.
- **Funcionalidades Avançadas:** Oferece recursos como recomendações, validações de competências, publicação de artigos e LinkedIn Learning.
- **Ferramentas de Recrutamento:** Possui funcionalidades robustas para recrutadores, como anúncios de emprego, pesquisa de candidatos e análise de dados.

UAW:

- **Foco Académico:** Projeto focado em conectar alunos, ex-alunos, empresas e a instituição académica, facilitando a comunicação e colaboração dentro dessa comunidade específica.
- **Customização para Instituições:** A instituição académica pode gerir e administrar o conteúdo e os utilizadores, garantindo a relevância e a adequação do material.
- **Engajamento e Colaboração:** Promove a troca de experiências, recursos e materiais entre os membros da comunidade académica.
- **Perfil Académico e Profissional:** Integra informações académicas com portfólios e Curriculum Vitae, facilitando o recrutamento especializado.

3.8.2.2 Fraquezas (Weaknesses)

LinkedIn:

- **Foco Corporativo:** Menor ênfase na comunidade académica e na interação entre alunos, o que pode limitar o envolvimento de estudantes e novos graduados.
- **Complexidade de Uso:** Pode ser intimidante para novos utilizadores devido à sua complexidade e variedade de funcionalidades.

UAW:

- **Base de Utilizadores Limitada:** Como é direcionado para uma comunidade académica específica, a sua base de utilizadores é mais limitada em comparação ao LinkedIn.
- **Visibilidade:** Menor reconhecimento de marca e visibilidade no mercado em comparação com uma plataforma global como o LinkedIn.
- **Recursos Limitados:** Pode ter menos recursos financeiros e tecnológicos para desenvolver funcionalidades avançadas e manutenção contínua.

3.8.2.3. Oportunidades (Opportunities)

LinkedIn:

- **Expansão Académica:** Potencial para integrar mais recursos educacionais e criar comunidades específicas para instituições académicas.
- **Novos Mercados:** Expansão em mercados emergentes e setores menos representados.
- **Inovações Tecnológicas:** Investimento em inteligência artificial e *machine learning* para personalização de conteúdo e recomendações.

UAW:

- **Nicho de Mercado:** Foco específico em instituições académicas, estudantes e empresas das áreas de interesse, permitindo uma personalização e adaptação mais profunda às necessidades educacionais e profissionais.
- **Parcerias com Instituições:** Possibilidade de firmar parcerias exclusivas com universidades e empresas para estágios e programas de recrutamento.
- **Desenvolvimento de Comunidade:** Oportunidade de criar uma comunidade sólida e engajada, focada na colaboração e no compartilhamento de conhecimento.

3.8.2.4. Ameaças (Threats)

LinkedIn:

- **Concorrência:** A competição com outras plataformas de networking e recrutamento, como Glassdoor, Indeed e outras redes sociais.
- **Regulamentação de Dados:** Mudanças nas leis de privacidade de dados podem afetar a maneira como a plataforma recolhe e utiliza informações dos utilizadores.

UAW:

- **Concorrência de Plataformas Consolidadas:** Concorrência com plataformas estabelecidas como LinkedIn, que já possuem uma vasta base de utilizadores e reconhecimento.
- **Sustentabilidade Financeira:** Desafios em garantir financiamento contínuo e sustentável para o desenvolvimento e manutenção da plataforma.
- **Adoção Limitada:** Risco de adoção limitada por parte dos alunos e empresas, especialmente se outras plataformas oferecerem soluções mais atrativas.

Como a sua principal Prova de Valor (*Unique Proof of Value*) o UAW apresenta uma proposta diferenciada e focada para um público específico, o que pode ser uma grande vantagem em termos de personalização e relevância. No entanto, enfrenta desafios significativos, especialmente em relação à visibilidade e sustentabilidade financeira. Por outro lado, o LinkedIn, sendo uma plataforma amplamente reconhecida e utilizada, tem a vantagem de uma

vasta rede de utilizadores e recursos, mas pode se beneficiar de uma maior integração com o mundo académico.

Ambas as plataformas têm oportunidades para crescimento e inovação, mas também enfrentam ameaças externas que podem impactar o seu sucesso e relevância no mercado.

4. CAPÍTULO: CASO DE USO DEMONSTRATIVO

4.1. Caso de uso utilizador Empresa

No sentido de apresentar a mais-valia da nossa solução, bem como realçar o elemento diferenciador referido anteriormente em relação à nossa concorrência, de seguida, iremos apresentar um caso de uso demonstrativo do procedimento de uma Empresa, externa à instituição académica, que tem como pressuposto a procura de novos Recursos Humanos, de forma a reforçar a sua Equipa.

A mesma teve conhecimento que a Instituição Universitária disponibilizava este serviço, à comunidade Académica e Empresarial, no sentido de contribuir ainda mais e melhor para o seu desenvolvimento profissional e estudantil.

Assim os próximos passos apresentados, com ecrãs de exemplificação, referem-se a todo o processo, ou seja, na prática são as interações que o utilizador “Empresa” pode realizar na plataforma UAW, com a instituição académica, onde a mesma é a instituição administradora que acolhe os utilizadores Alunos.

Ações realizadas pelo Utilizador Empresa

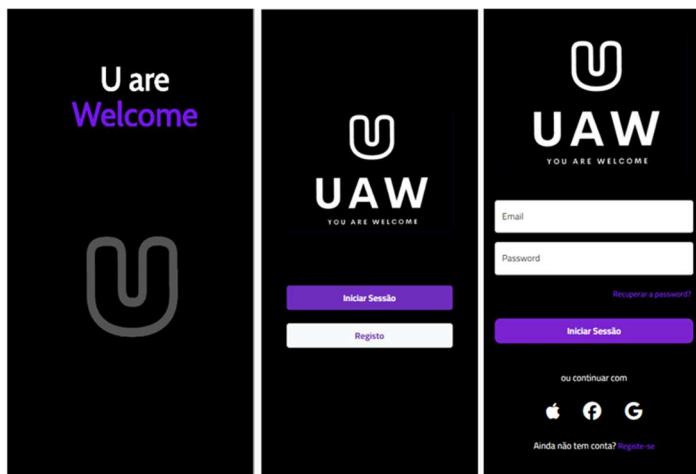


Figura 9 - Ecrãs “Splash”, “Home”, “Início de Sessão”

Como é possível verificar na **Figura 9**, a empresa ao aceder á aplicação UAW, é apresentado o ecrã de *Splash*, este ecrã em alguns casos pode não ser visível, caso a ligação à internet entre o dispositivo de acesso e os servidores da UAW seja rápida o suficiente.

Em seguida é apresentado o ecrã “Home”, neste ecrã o utilizador tem duas opções disponíveis: Caso já tenha dados de autenticação, escolha a opção “Iniciar Sessão”; caso seja a sua primeira visita, avança para o “Registo”. Esta segunda opção será a que iremos demonstrar.

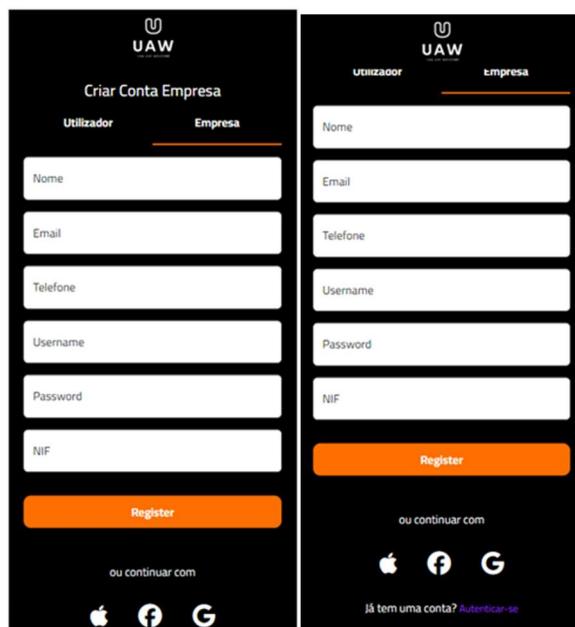


Figura 10 - Ecrã de Registo do tipo Empresa

Como é possível visualizar na **Figura 10**, o utilizador do tipo Empresa, tem um formulário com alguns campos necessários para efetuar o seu registo na plataforma. No caso os campos foram preenchidos e foi pressionado o botão “Registar”

Após o correto registo, o utilizador é notificado e encaminhado para o ecrã de “Início de Sessão” (**Figura 11**). Aí deverá preencher os campos email e password com os dados que utilização aquando do registo.

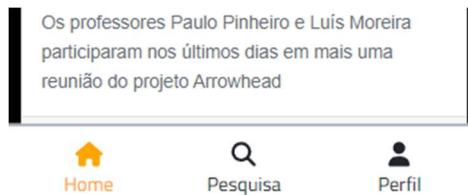


Figura 11 - Ecrã "Welcome", após Autenticação, ação “Pesquisa”

Após a autenticação é apresentado ao utilizador o ecrã “Welcome” (**Figura 11**), que não vamos descrever a sua funcionalidade, pois sai do foco desta demonstração, no entanto o utilizador iria escolher a opção “Pesquisa” presente na barra de ação no fundo a aplicação.

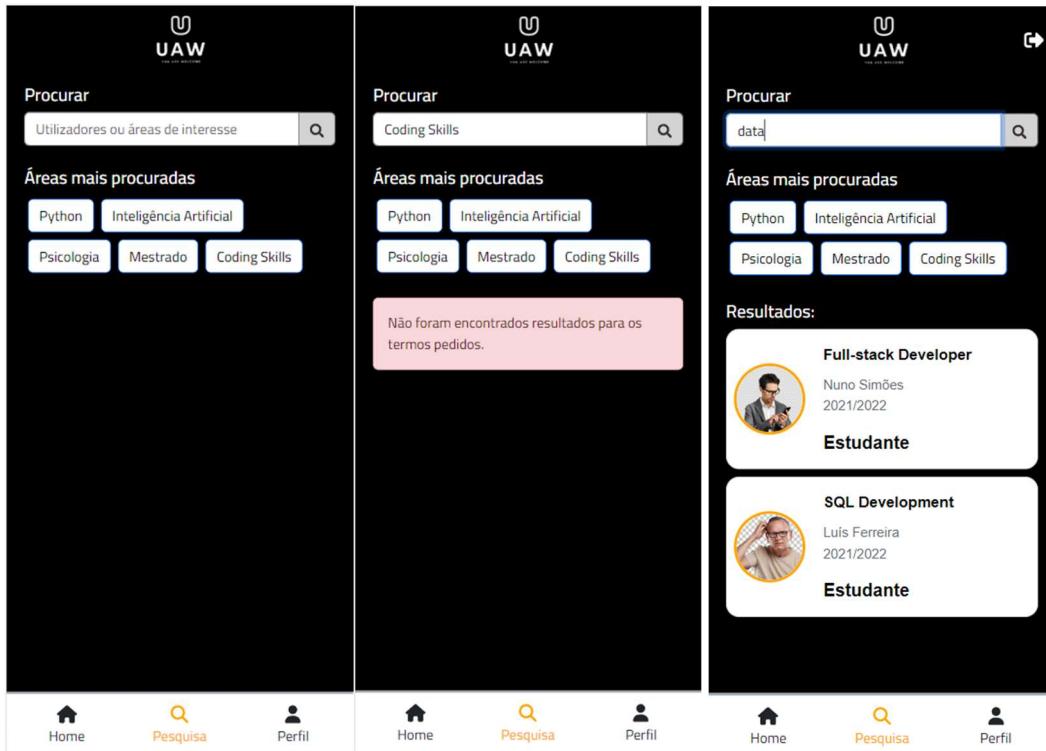


Figura 12 - Ecrãs de Pesquisa

Como é possível visualizar na **Figura 12**, o primeiro ecrã seria o estado inicial quando o utilizador chega ao mesmo, o segundo ecrã seria a notificação dada ao mesmo caso o termo que ele utilizou como pesquisa não retornasse qualquer resultado, e por último, havendo resultados que fizessem *match* com o termo de pesquisa, seria apresentada uma lista com os mesmos.

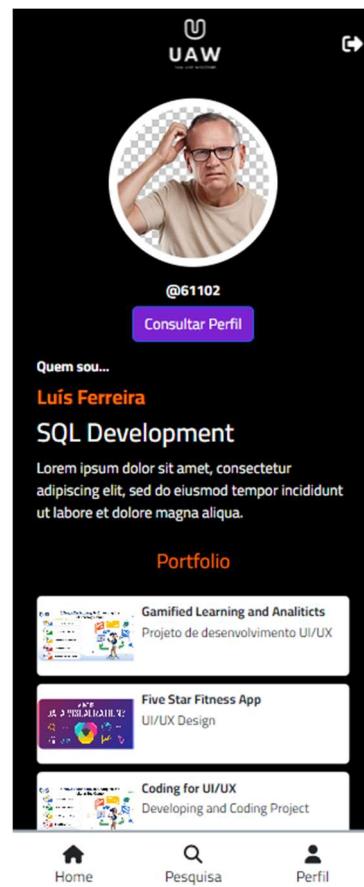


Figura 13 - Ecrã de "Portfólio" de um utilizador Aluno

É possível o utilizador clicar sobre o resultado de modo a poder ter acesso ao “Portfólio” do mesmo **Figura 13**.

Daqui é possível à Empresa visualizar o “Perfil” do aluno mediante o botão de ação presente para o efeito **Figura 14**.

Ou, visualizar mais em detalhe o “Portfólio” do aluno, escolhendo da lista de portfólios presentes e visualizar mais em detalhe o mesmo **Figura 15**.

UAW – Uma App para implementação de uma rede social universitária



Figura 14 - Ecrã de “Perfil” de Aluno

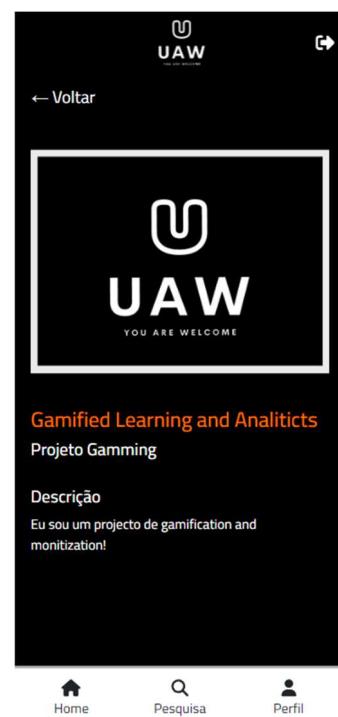


Figura 15 - Ecrã de Projeto de "Portfólio"

5. CAPÍTULO: CONCLUSÃO

A plataforma UAW (*U Are Welcome*) oferece uma solução inovadora para a interação e colaboração dentro da comunidade académica, com potencial para expandir e melhorar a experiência educacional e profissional dos utilizadores. O UAW destaca-se pela sua capacidade de integrar alunos, empresas e instituições académicas num ambiente digital coeso e funcional, facilitando a comunicação, a partilha de conhecimentos e a criação de oportunidades.

Para o futuro da UAW, identificamos várias funcionalidades adicionais que podem enriquecer ainda mais a experiência dos utilizadores e expandir o alcance da plataforma, como seja a implementação de um sistema de notificações personalizado que é fundamental para manter os utilizadores informados sobre novas oportunidades, mensagens recebidas e atualizações relevantes. Este sistema de notificações permitirá que cada utilizador ajuste as suas preferências para receber apenas informações que sejam pertinentes aos seus interesses e necessidades específicas. Por exemplo, um aluno poderá optar por ser notificado sobre novas ofertas de estágio na sua área de estudo, enquanto uma empresa pode preferir receber alertas sobre perfis de candidatos que correspondam aos seus requisitos de contratação.

Outra área que pretendemos desenvolver é a funcionalidade de calendarização e divulgação de eventos, workshops e feiras de emprego que se demonstra crucial para o desenvolvimento profissional dos alunos e a visibilidade das empresas. Através do UAW, os alunos poderão facilmente inscrever-se e participar em eventos que os ajudarão a desenvolver competências e a criar redes de contatos valiosas. Da mesma forma, as empresas e instituições académicas poderão promover os seus eventos, aumentando a participação e interligação entre a comunidade. Esta funcionalidade incluirá notificações de eventos e resumos pós-evento para que os participantes possam manter-se atualizados e refletir sobre as experiências adquiridas. As ferramentas de pesquisa avançada serão indispensáveis para maximizar a eficácia da plataforma. Empresas poderão utilizar filtros para encontrar candidatos com habilidades e experiências específicas, enquanto os alunos poderão pesquisar oportunidades de estágio e emprego que se alinhem com os seus objetivos profissionais. A pesquisa poderá ser refinada por critérios como área de estudo, localização, tipo de oportunidade, e outros parâmetros relevantes. Esta funcionalidade assegurará que tanto alunos quanto empregadores possam encontrar o que procuram de maneira rápida e eficiente, promovendo um *matching* mais preciso entre oferta e demanda.

Finalmente e para garantir uma comunicação fluida entre todos os utilizadores, o UAW incluirá um sistema de mensagens interno robusto. Este sistema permitirá a troca de mensagens privadas entre alunos, professores e representantes de empresas, facilitando a organização de entrevistas, discussões sobre projetos e outras interações necessárias. Além disso, os fóruns de discussão e grupos de interesse proporcionarão espaços onde os utilizadores poderão partilhar ideias, discutir temas relevantes e colaborar em projetos comuns. Esta funcionalidade promoverá um ambiente de aprendizagem colaborativa e de *networking*, essencial para o

crescimento profissional dos alunos e a interação eficiente entre as empresas e a comunidade académica.

A UAW, ao oferecer estas funcionalidades inovadoras, posiciona-se como uma plataforma essencial para a integração e desenvolvimento da comunidade académica. A interface intuitiva e as ferramentas avançadas de comunicação e pesquisa tornam o UAW uma solução completa para estudantes, empresas e instituições académicas que procuram maximizar suas interações e oportunidades.

Através de melhorias contínuas e da adição de novas funcionalidades, o UAW tem o potencial de se expandir ainda mais, adaptando-se às necessidades em constante evolução dos seus utilizadores. A visão futura da plataforma inclui o desenvolvimento de parcerias estratégicas com outras instituições e a exploração de novas tecnologias para melhorar a usabilidade e a segurança do sistema.

Em conclusão, o UAW não apenas preenche uma lacuna significativa no mercado académico e profissional, mas também estabelece um novo padrão para plataformas de interação e colaboração. Ao focar-se nas necessidades específicas dos seus utilizadores e ao promover um ambiente de aprendizagem e *networking*, a UAW está preparada para transformar a forma como alunos, empresas e instituições académicas se conectam e colaboram, contribuindo significativamente para o desenvolvimento educacional e profissional de todos os envolvidos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

1. React Documentation: <https://reactjs.org/docs/getting-started.html>
2. Spring Framework Documentation: <https://spring.io/projects/spring-framework>
3. MariaDB Documentation: <https://mariadb.org/documentation/>
4. LinkedIn: <https://pt.linkedin.com/>
5. Facebook: <https://www.facebook.com/>
6. Google Classroom: <https://classroom.google.com/>
7. Microsoft Teams: <https://teams.microsoft.com/>
8. MVC: <https://coodesh.com/blog/dicionario/o-que-e-arquitetura-mvc/>

ANEXO 1 – DESIGN

Relatório de Design Gráfico UI/UX para Aplicação UAW

O processo de design gráfico UI/UX desenvolvido para a criação de uma aplicação móvel de rede social voltada para estudantes universitários, foi dividido em várias etapas, utilizando as ferramentas Procreate e Figma para esboços iniciais e finalização do design, respetivamente.

O nome da aplicação, UAW, surgiu do brainstorming da equipa que está a trabalhar neste projeto e resulta das iniciais U (um jogo com a pronúncia em inglês, onde "U" e "you" têm a mesma forma de pronunciar) Are Welcome. A ideia é dar as boas-vindas a todos quantos queiram entrar nesta rede social.

A partir da sigla UAW, a equipa de design criou o logótipo que pretende transmitir uma sensação jovem, tecnológica e universitária.



Figura 16 - Logotipo UAW

A primeira fase do projeto envolveu uma pesquisa e planeamento detalhados. Foi realizada uma análise de mercado para entender as necessidades e preferências dos potenciais utilizadores, identificando as funcionalidades mais valorizadas em redes sociais universitárias. Além disso, foram estudadas tendências de design e melhores práticas em UI/UX.

Brainstorming

Com base nas informações obtidas na fase de pesquisa, foram realizadas sessões de brainstorming para gerar ideias e possíveis soluções de design. Durante estas sessões, foram discutidos os elementos essenciais como a navegação, estrutura da informação, e as principais funcionalidades da aplicação.

Criação de Wireframes

Utilizando o **Procreate**, foram criados esboços iniciais (*wireframes*) das principais janelas da aplicação. Esta ferramenta permitiu uma rápida iteração de ideias e a visualização das primeiras conceções do layout, fluxos de utilizador e interações básicas. Os wireframes serviram como base para discutir e refinar os conceitos antes de avançar para o design detalhado.

Desenvolvimento de Mockups

Após a aprovação dos *wireframes*, o passo seguinte foi desenvolver *mockups* de alta-fidelidade no Figma. Esta fase incluiu:

- Definição da Paleta de Cores: Escolha de uma paleta de cores que reflita a identidade da aplicação e seja atraente para o público-alvo.
- Tipografia: Seleção de fontes que garantam legibilidade e coerência estética.
- Componentes Visuais: Criação de botões, ícones, e outros elementos de interface.
- *Layout e Grid*: Organização das informações de forma clara e intuitiva, respeitando as proporções e alinhamentos.

Prototipagem e Testes de Usabilidade

Com os *mockups* desenvolvidos, foram criados protótipos interativos no Figma para simular a navegação e interações da aplicação. Estes protótipos foram utilizados em testes de usabilidade com o objetivo de corrigir e melhorar a aplicação.

Iteração e Refinamento

Com base no feedback dos testes de usabilidade, foram realizadas alterações no design. As modificações incluíram ajustes em componentes visuais, melhorias na navegação e aprimoramentos na experiência geral do utilizador. Esta etapa garantiu que a aplicação fosse não apenas funcional, mas também agradável e intuitiva de usar.

Documentação e Entrega

Por fim, todos os aspectos do design foram realizados detalhadamente, incluindo especificações de design, guia de estilo e diretrizes de implementação. A documentação completa foi entregue à equipa de desenvolvimento para garantir a consistência e qualidade na implementação final da aplicação. Além disso, o Figma permite exportar todo o código CSS e HTML para ReactJS, o que poupará tempo de desenvolvimento aos colegas encarregados de trabalhar no Front-End.

O uso combinado das ferramentas **Procreate** e Figma permitiu uma abordagem flexível e eficiente, desde os esboços iniciais até os *mockups* de alta-fidelidade e protótipos interativos.

Este relatório serve como registo detalhado das etapas de design e como guia para futuras iterações e melhorias na aplicação.

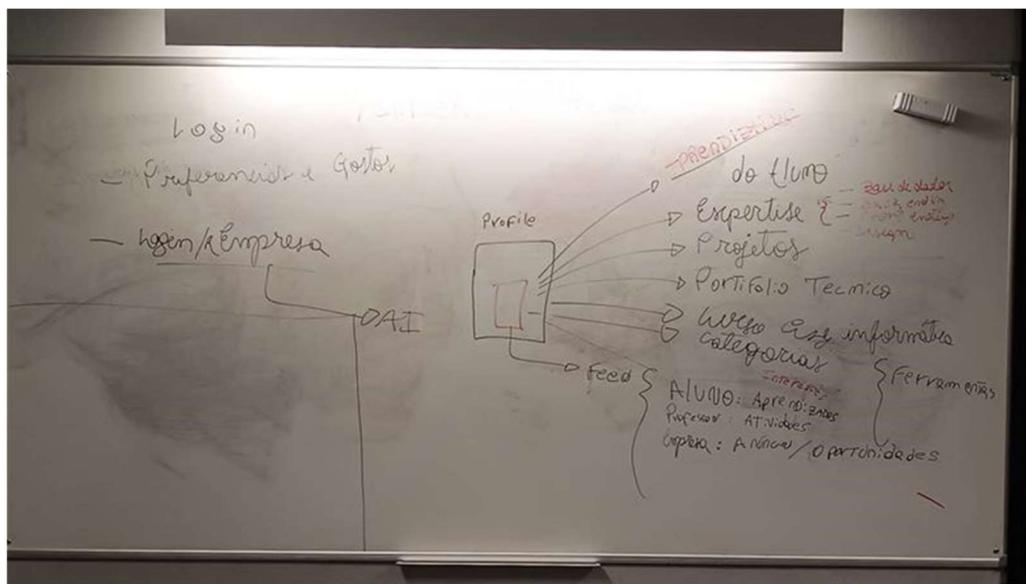


Figura 17 - Rascunho inicial

Wireframes

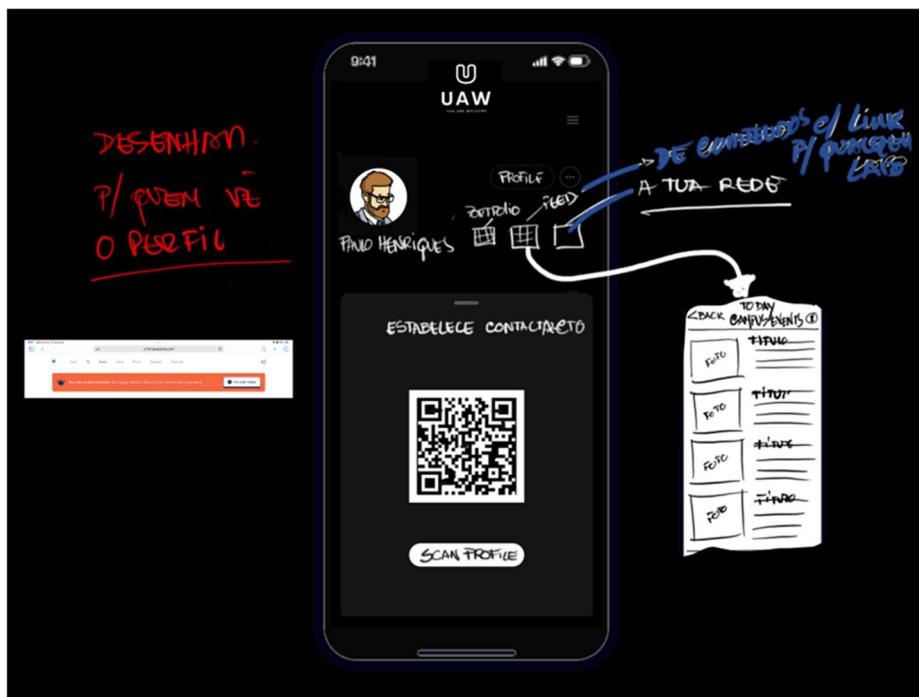


Figura 18 - Wireframe

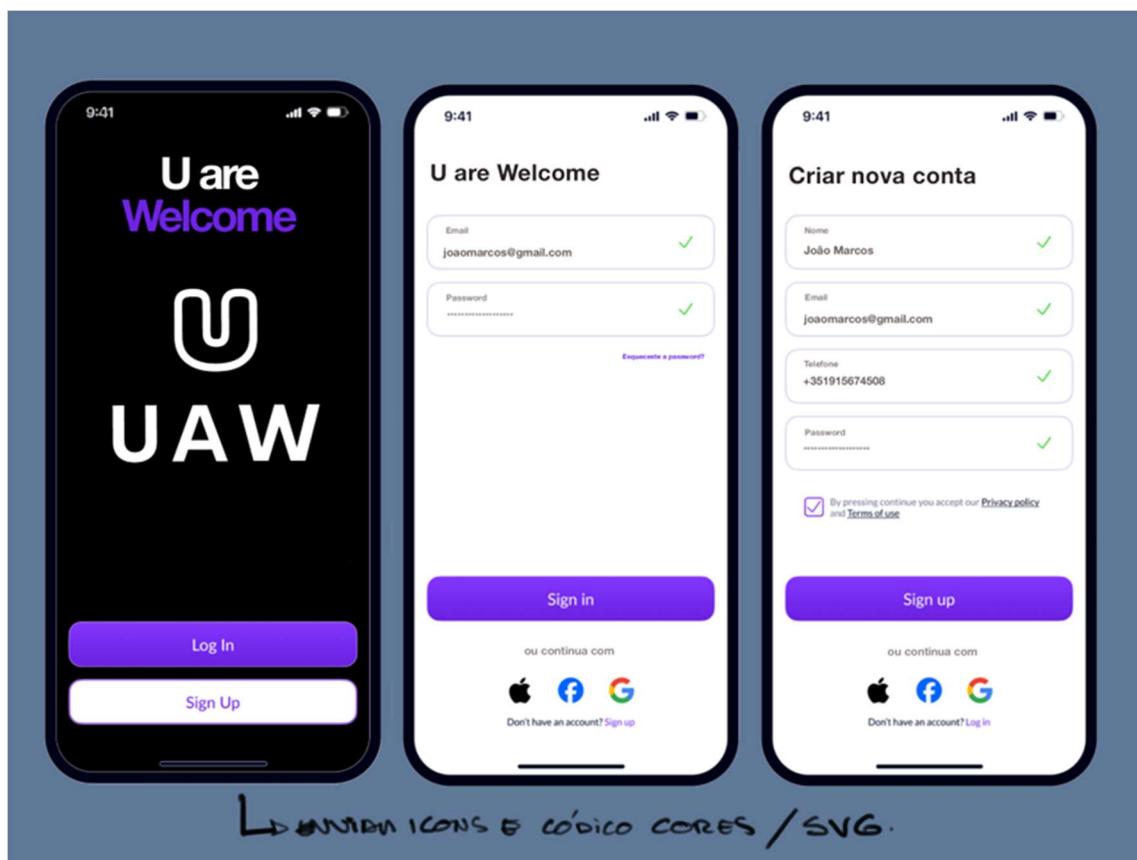
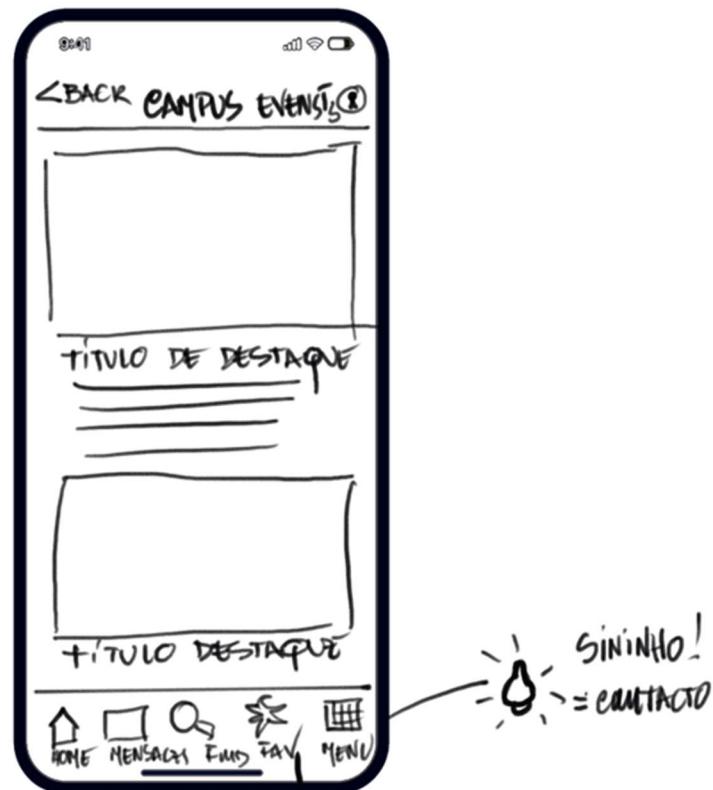


Figura 19 - Esboço 1

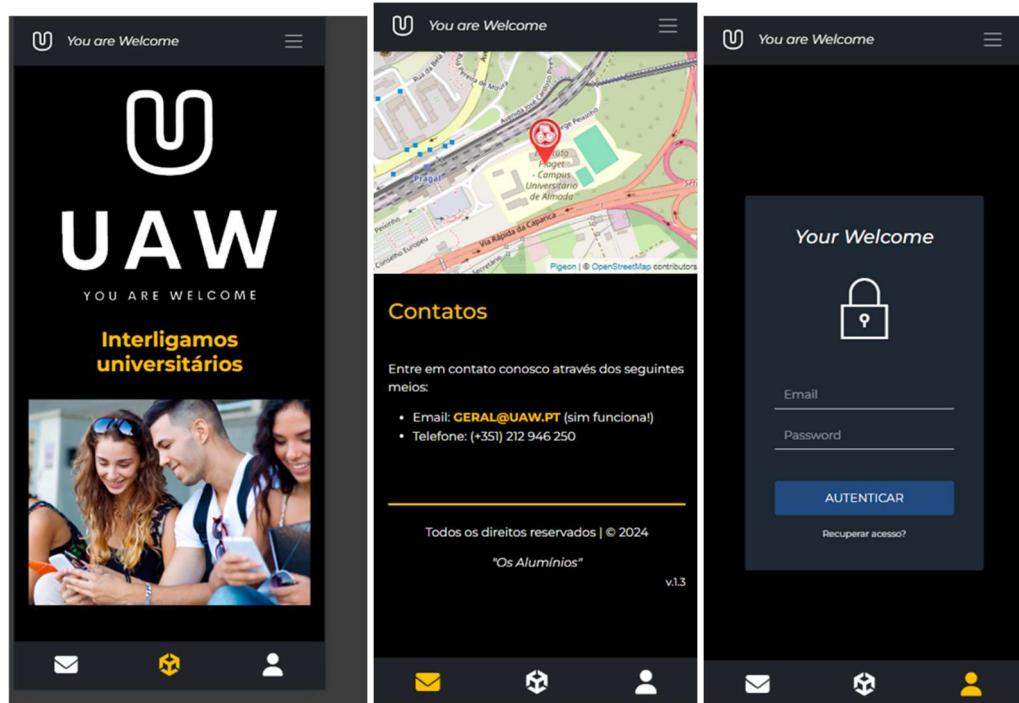


Figura 20 - Esboço #2

Mockups



Demo.uaw.pt



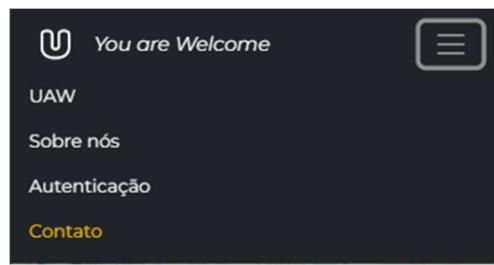


Figura 21 – Mockups

ANEXO 2 – FRONT-END

Tecnologia

Todo o Front-End foi desenvolvido na linguagem ReactJS, que é uma biblioteca JavaScript de código aberto desenvolvida pelo Facebook. É amplamente utilizada para a construção de interfaces de utilizador (UI) em aplicações web.

No caso do projeto integrador e para o objetivo que pretendemos atingir foi considerada a linguagem mais indicada para o efeito.

Visual Studio Code

Como ferramenta de desenvolvimento foi utilizado o Visual-Studio Code, uma ferramenta gratuita, mas com um potencial fantástico para este tipo de desenvolvimento.

A ferramenta permite também a instalação de alguns plug-ins de auxílio ao desenvolvimento e codificação, no caso a título de exemplo “Prettier” que permite que o código seja estruturado, e apresentado com elementos de realce visual no sentido de facilitar o desenvolvedor a coerência na programação.

GitHub

Foi utilizada também a ferramenta GitHub como repositório ao longo de todo o desenvolvimento do Front-End.

ReactJS

Componentes

O ReactJS é baseado no conceito de componentes, que são blocos reutilizáveis de código que representam partes da interface do utilizador. Cada componente pode ter seu próprio estado e lógica.

Existem dois tipos principais de componentes no ReactJS: funcionais (baseados em funções) e de classe (baseados em classes ES6).

JSX (JavaScript XML)

JSX é uma extensão de sintaxe que permite escrever código semelhante a HTML dentro do JavaScript. O JSX facilita a construção de componentes ReactJS de uma maneira que se assemelha a escrever HTML.

Virtual DOM

O ReactJS utiliza uma representação virtual da DOM (*Document Object Model*), chamada de Virtual DOM. Quando o estado de um componente muda, o Virtual DOM é atualizado primeiro. Em seguida, o ReactJS compara essa versão com a versão anterior e faz apenas as mudanças necessárias na DOM real, otimizando o desempenho.

One-Way Data Binding

O ReactJS segue o princípio de ligação de dados unidirecional. Isso significa que o fluxo de dados num aplicativo ReactJS é de cima para baixo, facilitando o rastreamento de mudanças no estado e na UI.

Hooks

Foi introduzido no ReactJS 16.8, os *Hooks* permitem aos programadores utilizarem o estado (*useState*) e outros recursos do ReactJS em componentes funcionais. Alguns dos *hooks* mais comuns são *useState*, *useEffect* e *useContext*.

Sempre que relevante serão indicadas mais informações sobre estes 5 pontos referidos, e com esclarecimentos adicionais se pertinente.

Vantagens do ReactJS

Reutilização de Componentes

Os componentes podem ser reutilizados em diferentes partes da aplicação, aumentando e melhorando a eficiência no desenvolvimento.

Desempenho

A utilização do Virtual DOM melhora significativamente o desempenho das atualizações da interface do utilizador.

Comunidade e Ecossistema

O ReactJS possui uma grande comunidade de desenvolvedores e um ecossistema rico em bibliotecas e ferramentas complementares.

Desenvolvimento Declarativo

A abordagem declarativa de construção de interfaces facilita a manutenção e leitura do código.

Ecossistema ReactJS

O ReactJS permite a utilização de diversas bibliotecas de auxílio ao desenvolvimento. No nosso projeto em específico foram utilizadas diversas bibliotecas, enumerando algumas, extra, base do próprio ReactJS:

- *React-Bootstrap*
- *React-router-dom*
- *Axios*
- *Js-Cookie*
- *jwt-Decode*
- *React-Icons*
- *React-FontAwesome*

As bibliotecas referidas foram sendo utilizadas na medida do necessário nos diversos ecrãs.

Estrutura

Como vimos o ReactJS tem uma estrutura própria a nível de pastas e nomenclaturas necessárias para o seu correto funcionamento. Assim apresentamos de seguida estrutura base do nosso projeto Front-End.

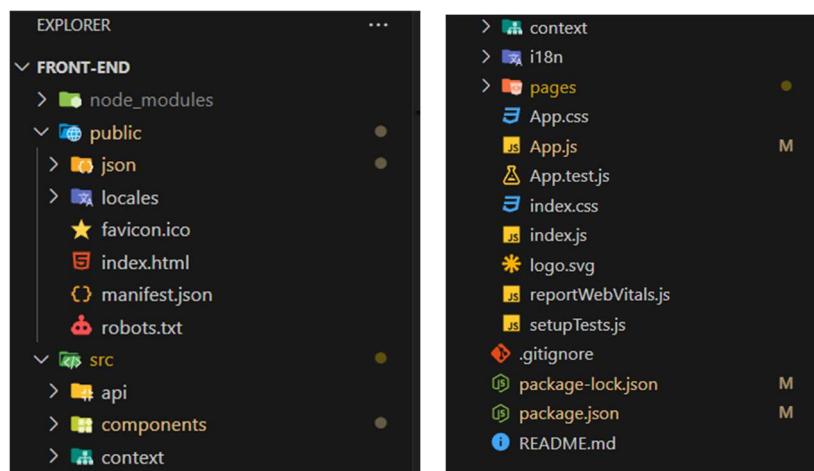


Figura 22 - Estrutura de Ficheiros do Front-End no Visual Studio Code

Ecrãs

Os ecrãs que são visíveis na aplicação e possíveis de interagir, foram adaptados e elaborados em ReactJS, com base nas especificações da equipa de design.

Foi dentro das possibilidades tecnológicas cumprir e fazer refletir, o desenho, comportamentos, UI e UX projetados para a realidade.

A aplicação ao nível de Front-End é composta pelos seguintes ecrãs, dividida em duas grandes áreas **Público** e **Autenticado**.

Público:

- *Splash / Aguarda carregamento*
- *Home*
- *Login*
- Recuperação de Palavra-passe
- Registo de Utilizador / Empresa
- Perfil

Autenticados:

- *Welcome*
- Editar Perfil
- Portfolio
- Pesquisa

Público

Splash / Aguarda carregamento

Este ecrã é o primeiro ecrã a ser visível quando se acede à aplicação, quando o tempo de carregamento da página para a qual o utilizador está a navegar é superior a 1seg, ou em última instância na interação com a API no tempo de espera da resposta da mesma estar a ser demorado.

Este ecrã é apenas composto por uma mensagem de boas-vindas e uma vídeo animação com o símbolo do “U” da marca UAW.

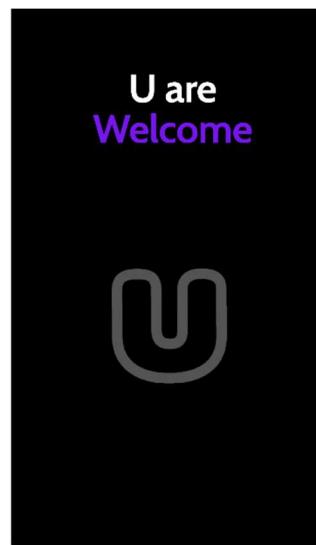


Figura 23 - Ecrã Splash

Home

Este é o ecrã base da aplicação, com isto refiro-me a não havendo um endereço específico, ou um endereço mal formatado, por defeito a aplicação retorna sempre este ecrã.

Neste ecrã podemos ver os elementos do logo bem como dois botões de ação: Iniciar Sessão e Registo.

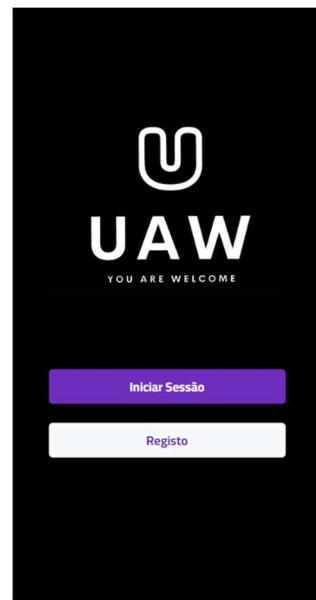


Figura 24 - Ecrã Home

Início de Sessão

Este ecrã é possível de ser acedido diretamente através de botões de ação de navegação, como exemplificado no ecrã Home, ou através do acesso a uma área autenticada, onde ainda não tenha sido feita a autenticação do utilizador.

Exemplo: Não estar autenticado e tentar aceder ao ecrã “Editar Perfil”, irá ser apresentado o ecrã de Início de Sessão ao utilizador.

Neste ecrã são apresentados o logo da UAW, campos de preenchimento, a opção de recuperação de palavra-passe, ir para o ecrã de registo e por último, opções de autenticação de terceiros.

É feita também uma pré-validação dos dados, ou seja, se os campos estão vazios, se são do tipo pretendido, exemplo, email ser um email mesmo, e só após esta validação é possível ser feito o pedido de verificação de autenticação à API.

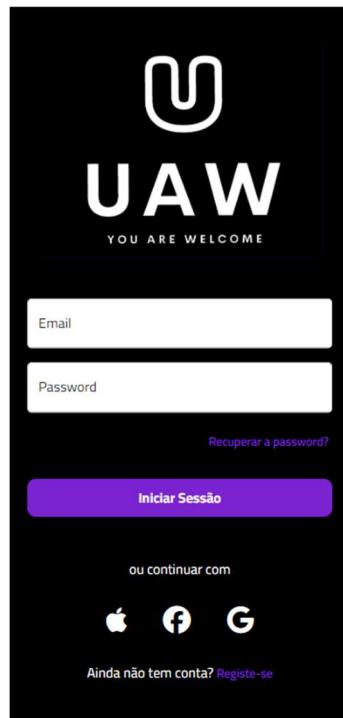


Figura 25 - Ecrã Início de Sessão

Recuperar Password

Este ecrã é composto por elementos simples, o logo da UAW, caixa de texto para inserção do email e botão de recuperação. Existe também a possibilidade de o utilizador tentar novamente a autenticação.

A utilidade deste ecrã é ser possível ao utilizador iniciar o procedimento de recuperação de password, desde que para isso se recorde do email que foi utilizado no momento do registo inicial, bem como tenha acesso a esse email para poder concluir o procedimento de recuperação.

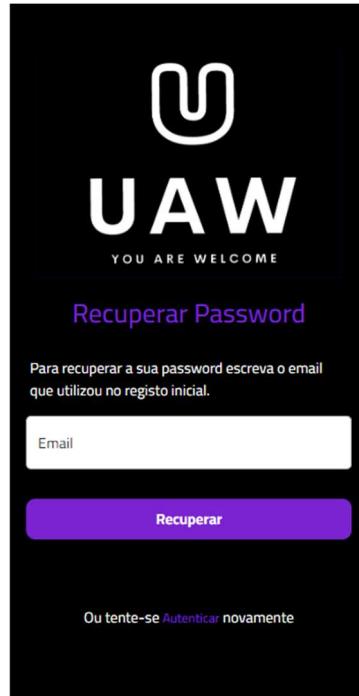


Figura 26 - Ecrã Recuperação de Password

Registo de Utilizador / Empresa

Neste ecrã é feita a introdução de um “componente” ReactJS, o “Header”, este componente não é uma biblioteca, foi desenvolvido, precisamente devido à necessidade de ser ou não necessário o mesmo estar visível, ou eventuais necessidades futuras comuns a diversos ecrãs ou mesmo à aplicação na sua totalidade. No sentido da premissa do ReactJS, reutilização de código, o Header foi desenvolvido com esse intuito em mente.

Para além do Header, temos um formulário que solicita ao utilizador o preenchimento de alguns campos. Esses campos podem ser diferentes, consoante se trate de um utilizador ou de uma empresa.

Mais uma vez consoante o tipo de registo, existe uma ligeira alteração visual, nomeadamente nas cores, de modo a alertar o utilizador para a ação que está a executar.

No final existe também a integração com plataformas terceiras, para ser feito o registo de uma conta.

E por último a possibilidade de navegar para o ecrã de login caso o utilizador já tenha credências de acesso registadas na plataforma.

Sempre que a quantidade de informação ou elementos de certa forma seja superior à área visível na vertical, os ecrãs da plataforma permitem o scroll vertical ao utilizador.

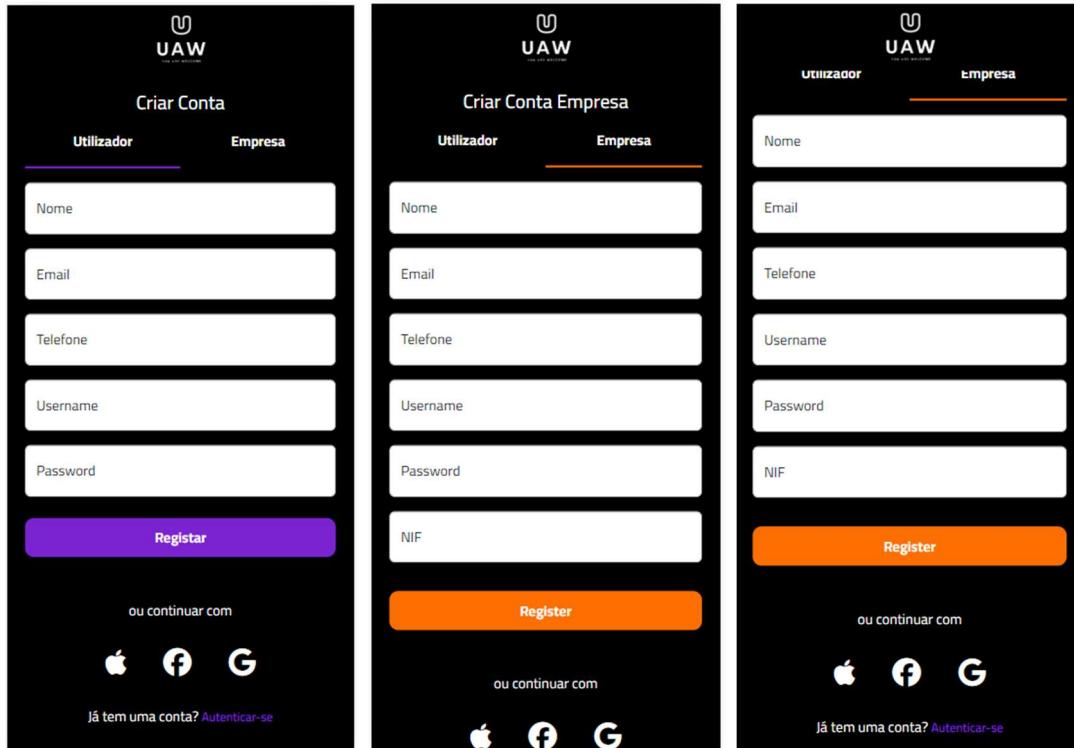


Figura 27 - Ecrãs de Registo de novo Utilizador, Aluno e Empresa

Perfil

Este ecrã é partilhado entre a área Pública e Autenticada. Nele é possível consultar o perfil do utilizador ou empresa que se pretende mediante a utilização do link correto para aceder ao mesmo, nomeadamente “/profile/61102” por exemplo que irá apresentar o perfil do utilizador 61102, de notar que 61102 não é o ID do utilizador, mas sim o seu *username*, no caso da instituição, Piaget, a mesma referência os alunos pelo seu número.

A diferença deste ecrã entre parte pública e a autenticada tem a ver maioritariamente com um novo componente “Footer” que à semelhança do que vimos anteriormente para o “Header” é uma área que pode não estar visível em todos os ecrãs. No caso o Footer por defeito nas áreas públicas irá sempre ter um botão de ação para que o visitante se possa autenticar, no caso de já estar autenticado, o Footer irá ter outro tipo de opções, que dependem do tipo de utilizador autenticado.

As informações apresentadas neste ecrã são apenas de consulta, uma espécie de cartão virtual para os utilizadores, ou uma ficha de empresa, se a consulta caiu sobre um perfil do tipo empresa.

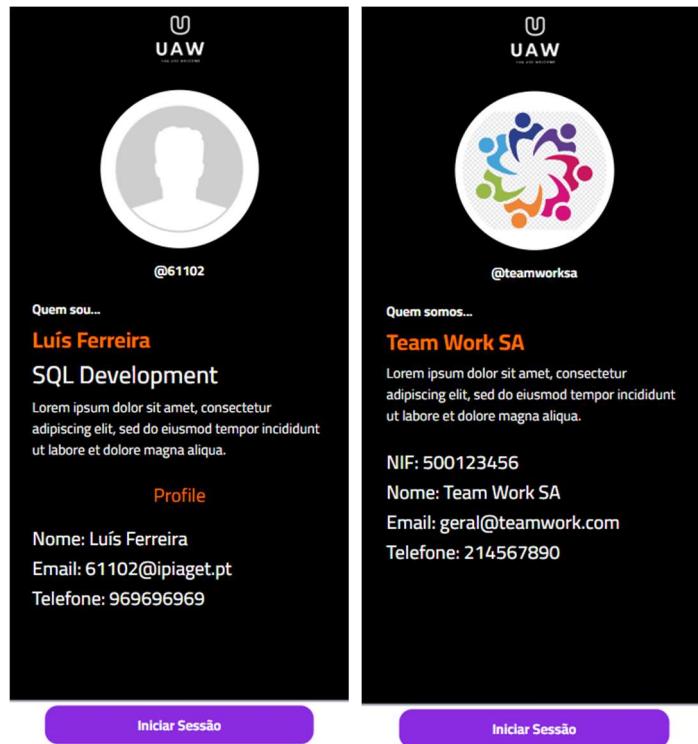


Figura 28 - Ecrãs de Visualização Pública de Perfil, Aluno e Empresa

Autenticado

A nível tecnológico, o Front-End comunica com a API no momento da autenticação, no sentido de validar se o utilizador pode aceder, e em caso afirmativo, é feito um *hand-shake* entre o Front-End e o Back-End, onde este último fornece uma *Cookie JWT*, que é encriptado e contém no seu interior 2 dados fundamentais para os diversos ecrãs da área “Autenticado”, que são o ID e uType, no caso ID corresponde ao id do utilizador na base de dados, e o uType corresponde ao seu tipo, utilizador ou empresa.

Welcome

Este ecrã é possível de ser visualizado apenas se tiver sido feita uma autenticação válida. No fundo o utilizador após a autenticação é redirecionado para este ecrã para poder ser informado em certa medida das novidades/notícias/eventos/etc. que a instituição tenha em divulgação.

Os dados apresentados aqui, não são da gestão ou responsabilidade da UAW, ou da aplicação, mas são fornecidos mediante uma API da própria instituição, onde aplicação apenas os divulga.

É possível se assim a notícia/evento/etc. tiver um URL associado, clicar na área da notícia o irá redirecionar o utilizador para esse mesmo URL.

Dependendo também do tipo de multimédia da notícia/evento/etc., pode ser reproduzido um vídeo ou uma imagem.

Este ecrã, à semelhança de todos os ecrãs de autenticação, tem na sua composição visual os componentes de Header e Footer.

Neste caso o *Footer* é composto por 3 opções de Ação, voltar ao *Welcome* (este ecrã), visualizar o Portfólio do utilizador autenticado, editar a sua ficha de Perfil na aplicação ou se for do tipo empresa pode fazer uma pesquisa.

Visualmente, cada opção estará realçada com a cor laranja, caso o utilizador esteja no ecrã correspondente à mesma.

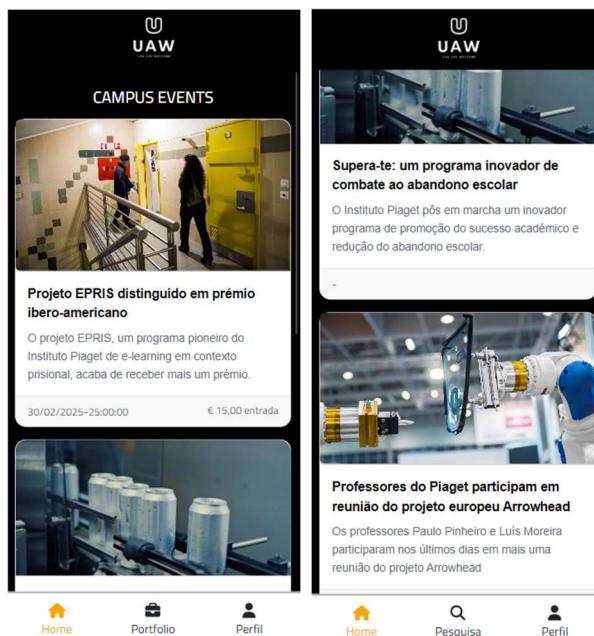


Figura 29 - Ecrãs “Welcome”, após autenticação, Aluno e Empresa

Editar Perfil

Aqui é permitido ao utilizador ou empresa alterar os dados que inseridos no momento da sua inscrição.

Dependendo do tipo de utilizador haverá campos que não estarão visíveis.

O campo *username* não é possível ser alterado, pois é o termo utilizado para apresentar por exemplo o Perfil publicamente.

À semelhança de outros ecrãs, se a informação a ser apresentada for superior à área de visualização é aplicado um scroll vertical.

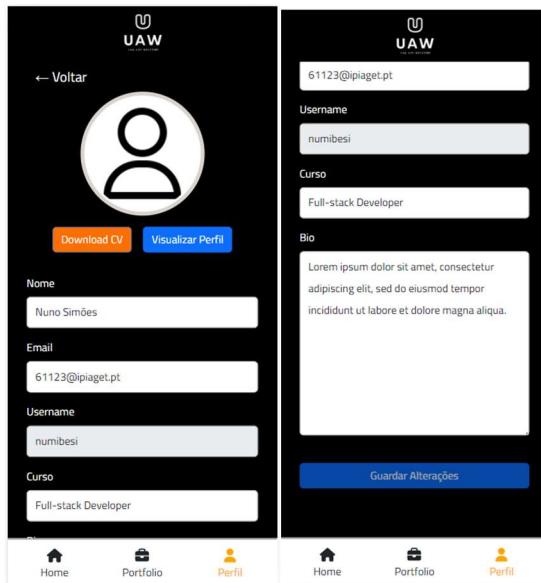


Figura 30 - Ecrã de Atualização de Perfil de Aluno

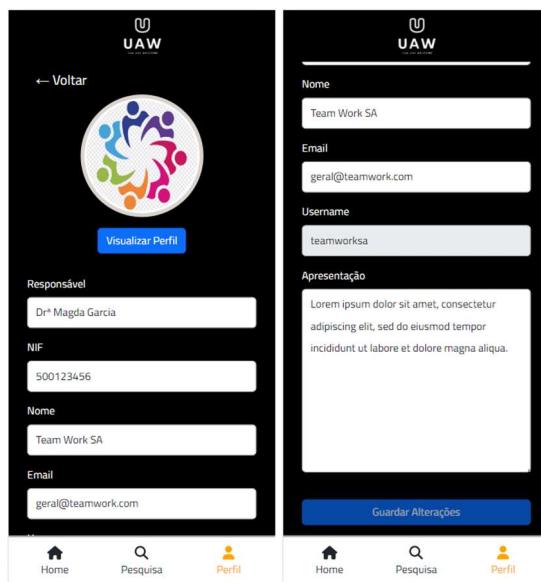


Figura 31 - Ecrã de Atualização de Perfil de Empresa

Portfólio

Neste ecrã é possível visualizar o Portfólio que o utilizador autenticado tem disponível para consulta por um outro utilizador do tipo Empresa.

A gestão do Portfólio, novas entradas, remoção e edição é feita mediante o clique ou touch sobre a entrada de Portfólio pretendida.

Também neste ecrã ele irá permitir *scroll* vertical se o número de elementos a apresentar ultrapassar a área vertical visível.

São apresentadas algumas informações do perfil neste ecrã, bem como a possibilidade de consultar o perfil mediante um botão de ação.

É apresentada a opção de voltar ao ecrã anterior, bem como o Header e Footer estão também eles presentes.

É apresentada a lista de projetos de Portfólio que o utilizador introduziu.

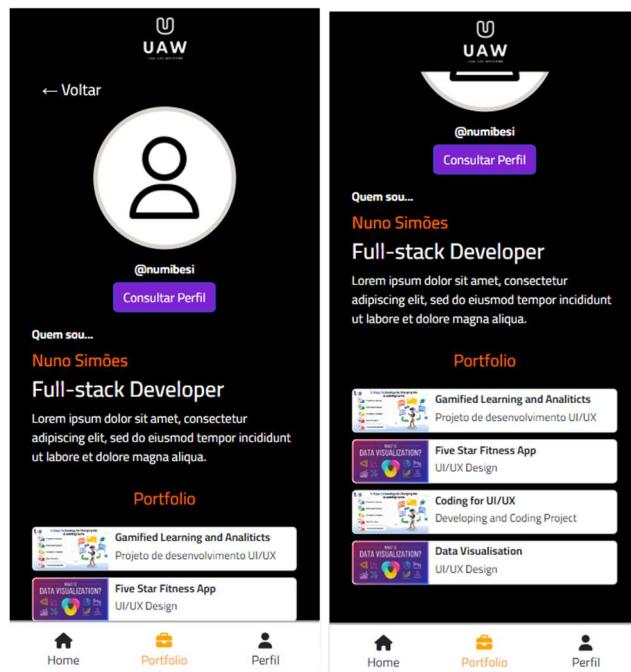


Figura 32 - Ecrã de Portfólio Público de Utilizador Aluno

Pesquisa

Este ecrã é específico de utilizadores autenticados do tipo “EMPRESA”. Nele é possível efetuar uma pesquisa mediante termos livres ou opções de procura pré-definidas, pela lista registos, de utilizadores do tipo “Utilizador” apenas de forma a pesquisar nos seus respetivos Perfis e Portfólios, se algum deles contém um termo igual ao da pesquisa introduzida no formulário.

O ecrã é composto pelo Header e Footer, e uma área central de pesquisa e apresentação de resultados.

Este ecrã se tentado aceder sem a autenticação efetuada redireciona para o ecrã de Login.

Caso não existam resultados a ser apresentados o utilizador é informado mediante uma área de notificação.

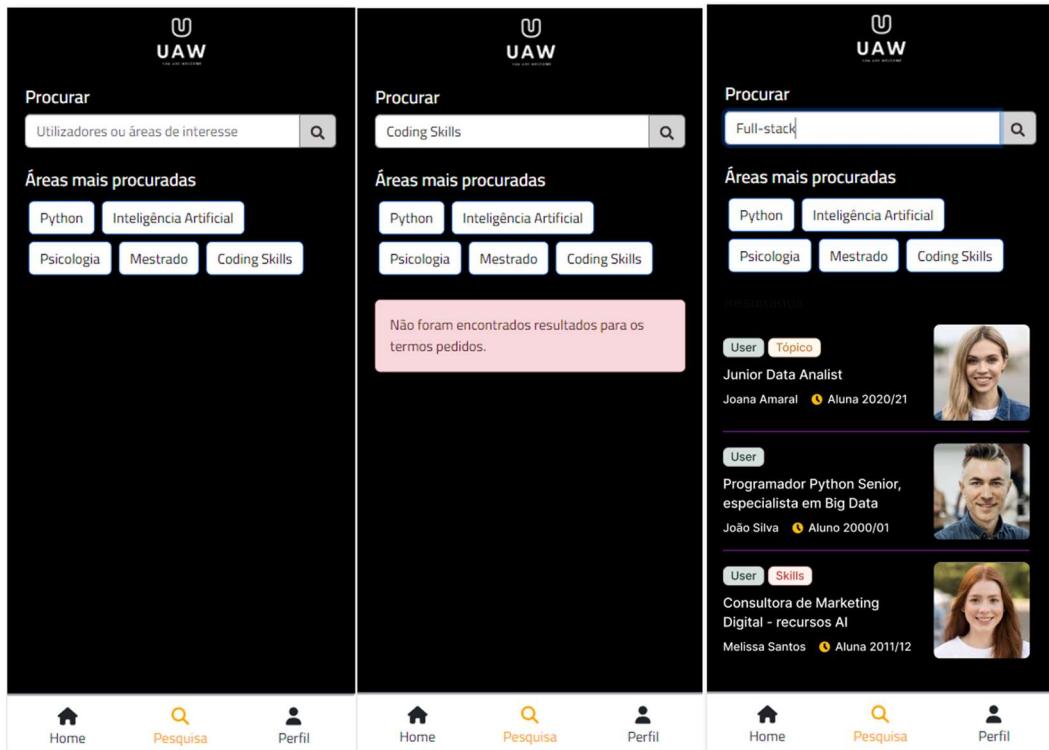


Figura 33 - Ecrãs de fases de Pesquisa para utilizador Empresa

ANEXO 3 – BACK-END

1. Introdução

Este documento tem como objetivo descrever o Back-End do projeto UAW, uma rede social que facilita a conexão entre empresas e alunos com base nas habilidades presentes em seus portfólios. O Back-End foi desenvolvido em Java utilizando o *Spring Boot*, e a comunicação com o Front-End é realizada por meio de end-points protegidos por *JWT (JSON Web Token)*.

2. Tecnologias Utilizadas

2.1 Spring Boot

Descrição: Spring Boot é um framework que simplifica o desenvolvimento de aplicações Java. Ele fornece um conjunto abrangente de funcionalidades para criar aplicações robustas e escaláveis com configuração mínima.

2.2 Maven

Descrição: Maven é uma ferramenta de gerenciamento de dependências e construção de projetos em Java. Ele facilita o gerenciamento de bibliotecas externas e a construção do projeto de forma eficiente.

2.3 JWT (JSON Web Token)

Descrição: JWT é um padrão aberto para autenticação e troca de informações de forma segura entre partes. Ele é amplamente utilizado para proteger comunicações entre o Front-End e o Back-End.

2.4 Comunicação com o Front-End

Descrição: A comunicação entre o Front-End e o Back-End é realizada por meio de end-points REST, utilizando JSON como formato de troca de dados. A autenticação é garantida por JWT, assegurando que apenas utilizadores autenticados possam aceder a recursos protegidos.

3. Estrutura do Projeto

3.1 Visão Geral

O Back-End do projeto UAW é organizado de maneira modular, facilitando a manutenção e a escalabilidade. A estrutura de pastas reflete os diferentes componentes da aplicação.

3.2 Estrutura

A estrutura de pastas de um projeto é fundamental para organizar o código de maneira lógica e eficiente, facilitando a manutenção e a escalabilidade da aplicação. No Back-End do projeto UAW, seguimos as melhores práticas de desenvolvimento Java com Spring Boot, organizando o código em pacotes (packages) específicos de acordo com suas responsabilidades. Aqui está uma visão geral da estrutura de pastas do projeto:

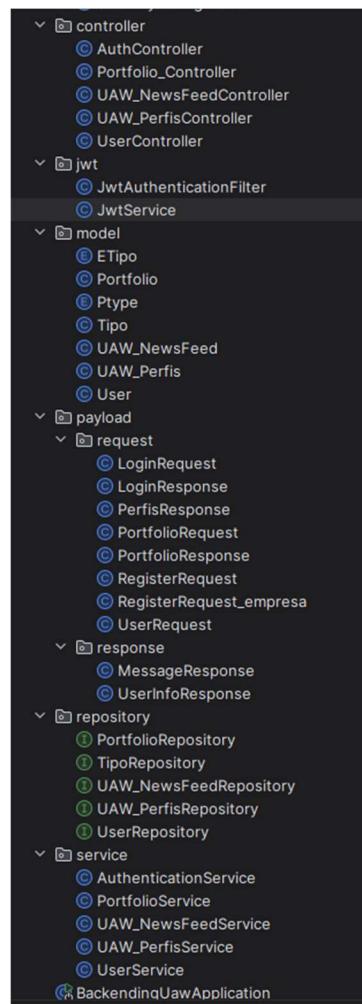


Figura 34 - Estrutura de ficheiros do Back-End no IntelliJ

4. Descrição dos Componentes

4.1 Configurações

Configurações de Segurança e Aplicação:

- SecurityConfiguration: Define as configurações de segurança, como autenticação JWT.
- ApplicationConfiguration: Configurações gerais da aplicação.
- GlobalExceptionHandler: Manipulador global de exceções, que lida com erros e retorna respostas apropriadas.

4.2 Controladores (Controllers)

Responsáveis por receber as requisições HTTP e direcioná-las aos serviços apropriados:

- AuthController: Controlador de autenticação.
- Portfólio_Controller: Controlador dos portfólios.
- UserController: Controlador de gerenciamento de utilizadores.
- UAW_NewsFeedController: Controlador do feed de notícias.
- UAW_PerfisController: Controlador dos perfis de utilizadores.
-

4.3 Modelos (Models)

Representam as entidades do domínio:

- User: Representa os utilizadores da aplicação.
- UAW_Perfis: Representa os perfis dos utilizadores.
- Ptype: Representa os tipos de portfólios possíveis.
- Portfólio: Representa o Portfólio dos utilizadores.
- UAW_NewsFeed: Representa as postagens no feed de notícias.
- Tipo: Gere os tipos de utilizadores e suas funções.
- ETipo: Define os tipos de utilizadores que podem existir na aplicação.
-

4.4 Repositórios (Repositories)

Responsáveis pela persistência dos dados:

- UserRepository: Repositório de utilizadores.
- PortfólioRepository: Repositório de todos portifólios dos utilizadores.
- TipoRepository: Repositório dos tipos de utilizadores.
- UAW_PerfisRepository: Repositório de perfis.
- UAW_NewsFeedRepository: Repositório de feed de notícias.

4.5 Serviços (Services)

Contém a lógica de negócios:

- AuthenticationService: Serviço de autenticação.
- PortfólioService: Serviço de gerenciamento dos portfólios.
- UserService: Serviço de gerenciamento de utilizadores.
- UAW_PerfisService: Serviço de gerenciamento de perfis.
- UAW_NewsFeedService: Serviço de gerenciamento de feed de notícias.

5. Endpoints

5.1 AuthController

A classe AuthController gerência os end-points relacionados com a autenticação de utilizadores, incluindo login e registo de novos utilizadores.

5.1.1 Login

- **Endpoint:** /api/auth/login
- **Método:** POST
- **Descrição:** Autentica um utilizador com base no email e senha fornecidos. Se as credenciais estiverem corretas, um token JWT é gerado e retornado ao utilizador.
- **Corpo da Requisição:**

```
{  
  "email": "string",  
  "password": "string"  
}
```

- **Resposta:**

```
{  
  "token": "string",  
  "expiresIn": "integer"  
}
```

5.1.2 Registo de Utilizador

- **Endpoint:** /api/auth/register
- **Método:** POST
- **Descrição:** Regista um novo utilizador no sistema.
- **Corpo da Requisição:**

```
{  
  "username": "string",  
  "firstName": "string",  
  "lastName": "string",  
  "email": "string",  
  "phone": "string",  
  "password": "string",  
  "tipo": "integer"  
}
```

- **Resposta:**

```
{  
  "id": "integer",  
  "username": "string",  
  "firstName": "string",  
  "lastName": "string",  
  "email": "string",  
  "phone": "string"  
}
```

5.1.3 Registo de Empresa

- **Endpoint:** /api/auth/register_empresa
- **Método:** POST
- **Descrição:** Regista uma nova empresa no sistema.
- **Corpo da Requisição:**

```
{  
  "username": "string",  
  "SNIF": "string",  
  "email": "string",  
  "phone": "string",  
  "password": "string",  
  "tipo": "integer"  
}
```

- **Resposta:**

```
{
    "id": "integer",
    "username": "string",
    "SNIF": "string",
    "email": "string",
    "phone": "string"
}
```

5.2 UAW_NewsFeedController

A classe UAW_NewsFeedController gerencia os end-points relacionados com as notícias (news feed) do sistema.

5.2.1 Buscar todas as Notícias

- **Endpoint:** /api/newsfeed
- **Método:** GET
- **Descrição:** Retorna todas as notícias do sistema.
- **Resposta:**

```
[
{
    "id": "integer",
    "title": "string",
    "content": "string"
}]
```

5.2.2 Buscar Notícia por ID

- **Endpoint:** /api/newsfeed/{id}
- **Método:** GET
- **Descrição:** Retorna uma notícia específica pelo ID.
- **Resposta:**

```
{
    "id": "integer",
    "title": "string",
    "content": "string"
}
```

5.2.3 Criar Notícia

- **Endpoint:** /api/newsfeed
- **Método:** POST
- **Descrição:** Cria uma nova notícia.
- **Corpo da Requisição:**

```
{  
    "title": "string",  
    "content": "string"  
}
```

- **Resposta:**

```
{  
    "id": "integer",  
    "title": "string",  
    "content": "string"  
}
```

5.2.4 Atualizar Notícia

- **Endpoint:** /api/newsfeed/{id}
- **Método:** PUT
- **Descrição:** Atualiza uma notícia existente pelo ID.
- **Corpo da Requisição:**

```
{  
    "title": "string",  
    "content": "string"  
}
```

- **Resposta:**

```
{  
    "id": "integer",  
    "title": "string",  
    "content": "string"  
}
```

5.2.5 Eliminar Notícia

- **Endpoint:** /api/newsfeed/{id}
- **Método:** DELETE
- **Descrição:** Deleta uma notícia pelo ID.
- **Resposta:** 204 No Content

5.3 UAW_PerfisController

A classe UAW_PerfisController gerencia os end-points relacionados com os perfis de utilizadores do sistema.

5.3.1 Buscar todos os Perfis

- **Endpoint:** /api/perfis
- **Método:** GET
- **Descrição:** Retorna todos os perfis de utilizadores do sistema.
- **Resposta:**

```
[  
  {  
    "idPerfil": "integer",  
    "idUtilizador": "integer",  
    "descricao": "string"  
  }  
]
```

9. Buscar Perfil por ID

- **Endpoint:** /api/perfis/{id}
- **Método:** GET
- **Descrição:** Retorna um perfil específico pelo ID.
- **Resposta:**

```
{  
  "id": 1,  
  "tipo": 1,  
  "icodigocurso": 2,  
  "ianocurso": "2021/2022",  
  "byfoto": "/images/61123.png",  
  "name": "Nuno Simões",  
  "username": "61123",  
  "email": "61123@ipiagget.pt",  
  "phone": "969696969",  
  "sbiografia": "Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor incididunt ut labore et  
dolore magna aliqua.",  
  "Portfólio": [  
    {  
      "id": 1,  
      "ptype": 1,  
      "title": "Gamified Learning and Analiticcts",  
    }  
  ]  
}
```

```

    "image": "/images/gamified.png",
    "smallDesc": "Projeto Gamming",
    "description": "Eu sou um projecto de gamification and
monitization!",
    "url": null,
    "tags": ["Gamified", "Learning", "Analitics", "UI",
"UX", "Coding Skills"]
},
{
    "id": 2,
    "ptype": 1,
    "title": "Five Star Fitness App",
    "image": "/images/bolsas.jpg",
    "smallDesc": "Aplicação de Desporto",
    "description": "",
    "url": null,
    "tags": ["Fitness", "App", "Five", "Star"]
},
{
    "id": 3,
    "ptype": 1,
    "title": "Coding for UI/UX",
    "image": "/images/coduiux.png",
    "smallDesc": "Developing and Coding UI/UX Projects",
    "description": "",
    "url": null,
    "tags": ["Coding", "Development", "Project", "UI", "UX"]
},
{
    "id": 4,
    "ptype": 1,
    "title": "Data Visualisation",
    "image": "/images/datavisualization.jpeg",
    "smallDesc": "UI/UX Design",
    "description": "",
    "url": null,
    "tags": ["Data", "Visualization", "UI", "UX", "Python"]
}
]
}

```

5.3.2 Criar Perfil

- **Endpoint:** /api/perfis
- **Método:** POST
- **Descrição:** Cria um perfil de utilizador.
- **Corpo da Requisição:**

```
{  
    "idUtilizador": "integer",  
    "descricao": "string"  
}
```

- **Resposta:**

```
{  
    "idPerfil": "integer",  
    "idUtilizador": "integer",  
    "descricao": "string"  
}
```

5.3.3 Atualizar Perfil

- **Endpoint:** /api/perfis/{id}
- **Método:** PUT
- **Descrição:** Atualiza um perfil existente pelo ID.
- **Corpo da Requisição:**

```
{  
    "descricao": "string"  
}
```

- **Resposta:**

```
{  
    "idPerfil": "integer",  
    "idUtilizador": "integer",  
    "descricao": "string"  
}
```

5.3.4 Eliminar Perfil

- **Endpoint:** /api/perfis/{id}
- **Método:** DELETE
- **Descrição:** Deleta um perfil pelo ID.
- **Resposta:** 204 No Content

5.4 UserController

A classe UserController gerencia os end-points relacionados com os utilizadores do sistema.

5.4.1 Listar todos os Utilizadores

- **Endpoint:** /api/user/list
- **Método:** GET
- **Descrição:** Retorna todos os utilizadores do sistema.
- **Resposta:**

```
[  
  {  
    "id": "integer",  
    "username": "string",  
    "firstName": "string",  
    "lastName": "string",  
    "email": "string",  
    "phone": "string"  
  }  
]
```

5.4.2 Buscar Utilizador por ID

- **Endpoint:** /api/user/get{id}
- **Método:** GET
- **Descrição:** Retorna um utilizador específico pelo ID.
- **Resposta:**

```
{  
  "id": "integer",  
  "username": "string",  
  "firstName": "string",  
  "lastName": "string",  
  "email": "string",  
  "phone": "string"  
}
```

5.4.3 Criar Utilizador

- **Endpoint:** /api/user
- **Método:** POST
- **Descrição:** Cria um novo utilizador.
- **Corpo da Requisição:**

```
{  
  "username": "string",
```

```

    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string",
    "password": "string"
}
```

- **Resposta:**

```
{
    "id": "integer",
    "username": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string"
}
```

5.4.4 Atualizar Utilizador

- **Endpoint:** /api/user/set{id}
- **Método:** PUT
- **Descrição:** Atualiza um utilizador existente pelo ID.
- **Corpo da Requisição:**

```
{
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string",
    "password": "string"
}
```

- **Resposta:**

```
{
    "id": "integer",
    "username": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string"
}
```

5.4.5 Deletar Utilizador

- **Endpoint:** /api/user/del{id}
- **Método:** DELETE
- **Descrição:** Deleta um utilizador específico pelo ID.
- **Resposta:** 204 No Content

5.4.6 Pesquisar Utilizadores

- **Endpoint:** /api/user/search
- **Método:** GET
- **Descrição:** Pesquisa utilizadores com base em critérios específicos, como nome de utilizador, email ou número de telefone.
- **Parâmetros da Query:**
 - username (opcional): Filtra utilizadores pelo nome de utilizador.
 - email (opcional): Filtra utilizadores pelo email.
 - phone (opcional): Filtra utilizadores pelo número de telefone.
- **Resposta:**
-
- ```
{
 "id": "integer",
 "username": "string",
 "firstName": "string",
 "lastName": "string",
 "email": "string",
 "phone": "string"
}
```

### 5.5 Portfólio\_Controller:

A classe Portfólio\_Controller gerência os portfólios dos utilizadores.

#### 5.5.1 Criar Portfólio:

- **Endpoint:** /api/Portfólio/new
- **Método:** POST
- **Descrição:** Cria um portfólio novo no sistema.
- **Corpo de Requisição:**

```
{
 "id": "integer",
 "idutilizador": "string",
 "ptype": "integer",
 "image": "String",
```

```

"smalldesc": "string",
"description": "string"
"url": "string"
"tags": "Array_String"
}

```

### **5.5.2 Vizualizar Portfólio específico:**

- **Endpoint:** /api/Portfólio/view/{id}
  - **Método:** GET
  - **Descrição:** Vizualiza um portfólio.
  - **Resposta:** Exemplo:
- ```

{
    "id": 1,
    "idUtilizador": 1"
    "ptype": 1,
    "title": "Gamified Learning and Analiticts",
    "image": "/images/gamified.png",
    "smallDesc": "Projeto Gamming",
    "description": "Eu sou um projecto de gamification and
monitization!",
    "url": null,
    "tags": ["Gamified", "Learning", "Analiticts", "UI", "UX", "Coding
Skills"]
}

```

5.5.3 Salvar Portfólio:

- **Endpoint:** /api/Portfólio/save/{id}
 - **Método:** POST
 - **Descrição:** Salva um portfólio novo no sistema.
 - **Resposta:** Exemplo:
- ```

{
 "id": 1,
 "idUtilizador": 1"
 "ptype": 1,
 "title": "Gamified Learning and Analiticts",
 "image": "/images/gamified.png",
 "smallDesc": "Projeto AI", // Parte alterada
 "description": "Eu sou um projecto de gamification and
monitization!",
}

```

```
"url": null,
"tags": ["Gamified", "Learning", "Analiticts", "UI", "UX", "Coding
Skills"]
}
```

#### 5.5.4 Listar Portfólio:

- **Endpoint:** /api/Portfólio/list/{username}
- **Método:** GET
- **Descrição:** Apresenta todos os portfólios de um utilizador.
- **Resposta:** Exemplo:

```
{
 "id": 1,
 "ptype": 1,
 "title": "Gamified Learning and Analiticts",
 "image": "/images/gamified.png",
 "smallDesc": "Projeto Gamming",
 "description": "Eu sou um projecto de gamification and
monitization!",
 "url": null,
 "tags": ["Gamified", "Learning", "Analiticts", "UI", "UX", "Coding
Skills"]
},
{
 "id": 2,
 "ptype": 1,
 "title": "Five Star Fitness App",
 "image": "/images/bolsas.jpg",
 "smallDesc": "Aplicação de Desporto",
 "description": "",
 "url": null,
 "tags": ["Fitness", "App", "Five", "Star"]
}
```

#### 5.5.5 “Eliminar” um portfólio:

- **Endpoint:** /api/Portfólio/delete/{id}
- **Método:** DELETE
- **Descrição:** Desativa um portfólio pelo seu id.
- **Resposta:** Somente a resposta http.

## 6. Exemplos de Uso

### 6.1 Processo de Login

O diagrama BPMN a seguir descreve o fluxo de autenticação de um utilizador no sistema UAW. O processo começa quando o utilizador insere suas credenciais de login no frontend. Esses dados são organizados num formato JWT e enviados em uma requisição POST para o backend.

- **Verificação de Segurança:** O backend verifica se a requisição é segura. Se não for, retorna um erro HTTP 401 (Unauthorized).
- **Validação do Request:** Caso a requisição seja segura, o backend prossegue para verificar se o corpo da requisição é válido. Se houver qualquer problema com os dados, retorna um erro HTTP 400 (Bad Request).
- **Consulta ao Banco de Dados:** Se os dados forem válidos, o backend realiza uma consulta no banco de dados para verificar a existência do utilizador. Se o utilizador não for encontrado, retorna um erro HTTP 401 (Unauthorized).
- **Geração do Token JWT:** Se o utilizador for encontrado, o backend gera um token JWT contendo as informações do utilizador e retorna um HTTP 200 (OK), permitindo o acesso do utilizador à aplicação.

Em caso de falha em qualquer uma das etapas, uma mensagem de erro apropriada é retornada para o utilizador, garantindo a segurança e a clareza do processo.

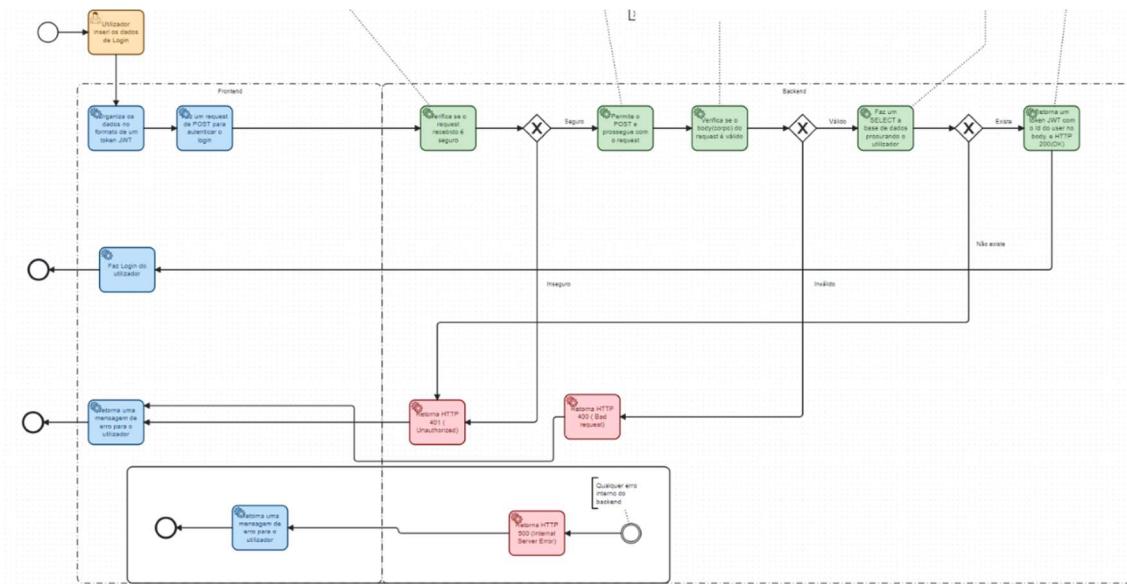


Figura 35 - BPMN

## 6.2 Processo de Registo

O diagrama BPMN a seguir ilustra o fluxo de registo de um novo utilizador no sistema UAW. O processo inicia quando o utilizador insere as suas informações de registo no frontend. Esses dados são organizados num formato JWT e enviados em uma requisição POST para o backend.

10. **Verificação de Segurança:** O backend verifica se a requisição é segura. Se não for, retorna um erro HTTP 401 (Unauthorized).
11. **Validação do Request:** Se a requisição for segura, o backend prossegue para verificar se o corpo da requisição é válido. Se os dados forem inválidos, retorna um erro HTTP 400 (Bad Request).
12. **Inserção no Banco de Dados:** Se os dados forem válidos, o backend insere as informações do novo utilizador no banco de dados. Caso haja qualquer erro durante essa operação, retorna um erro HTTP 500 (Internal Server Error).
13. **Criação do Perfil:** Após a criação do utilizador, o backend cria um perfil associado a ele. Se tudo ocorrer conforme esperado, o backend retorna um HTTP 201 (Created), confirmando o sucesso do registo.

Se ocorrer qualquer falha no processo, uma mensagem de erro correspondente é enviada ao utilizador, garantindo a integridade e a clareza do fluxo de registo.

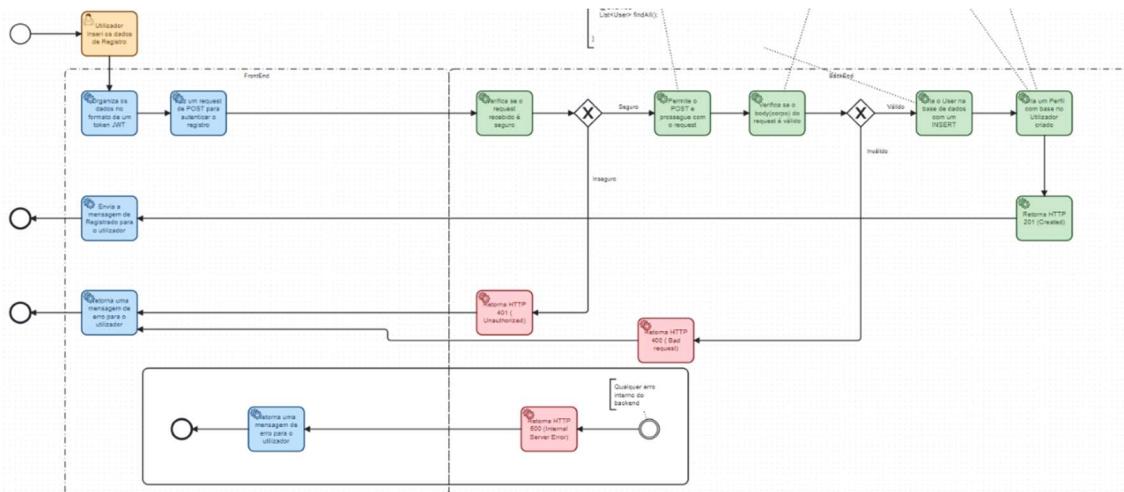


Figura 36 - BPMN #2

## 7. Glossário

- **AuthController:** Controlador responsável pelos endpoints relacionados à autenticação de utilizadores, incluindo login e registo.
- **Endpoint:** URL que representa um recurso no servidor e com a qual o cliente pode interagir.
- **ETipo:** Enumeração que define os tipos de utilizadores que podem existir na aplicação.
- **JWT (JSON Web Token):** Um padrão aberto para autenticação e troca de informações de forma segura entre partes. Amplamente utilizado para proteger comunicações entre o frontend e o backend.
- **Modelos (Models):** Representam as entidades do domínio dentro da aplicação.
- **Repositórios (Repositories):** Responsáveis pela persistência dos dados no sistema.
- **REST (Representational State Transfer):** Um estilo de arquitetura de software para sistemas distribuídos, como a web, que usa HTTP e JSON para troca de dados.
- **Spring Boot:** Um framework que simplifica o desenvolvimento de aplicações Java, fornecendo funcionalidades para criar aplicações robustas e escaláveis com configuração mínima.
- **Tipo:** Gere os tipos de utilizadores e suas funções na aplicação.
- **UAW\_NewsFeed:** Representa as postagens no feed de notícias.
- **UAW\_NewsFeedController:** Controlador que gerencia os endpoints relacionados com as notícias (news feed) do sistema.
- **UAW\_Perfis:** Representa os perfis dos utilizadores dentro do sistema.
- **UAW\_PerfisController:** Controlador que gerencia os endpoints relacionados com os perfis de utilizadores.
- **User:** Representa os utilizadores da aplicação.
- **UserController:** Controlador responsável pelos endpoints relacionados com o gerenciamento de utilizadores.

## ANEXO 4 – BASE DE DADOS

### Introdução.

O projeto UAW utiliza como sistema de gestor da base de dados o MariaDb ao qual se encontra associada a base de dados de nome “dbuaw\_sn”, esta base de dados é o suporte a todo o sistema em termos de gestão de dados e informação em sistema relativamente a uma instituição académica.

A estrutura aqui apresentada reflete a estrutura tipificada para suportar a versão corrente do projeto UAW que na sua essência é uma estrutura simplificada tendo em conta que a gestão em termos de registo e consulta de dados é executado via Java Spring utilizado no módulo de backend do projeto UAW.

A base de dados aqui apresentada é composta por 6 tabelas que irão gerir os dados e a informação do projeto.

### Tabelas

#### **uaw\_utilizadores**

Tabela onde são registados e consultados os utilizadores do sistema UAW, esta tabela é composta pelos seguintes oito campos:

- Idutilizador

Campo de tipo inteiro que possui como função a identificação de utilizador em sistema. Este campo é uma chave primária na tabela.

- sNome

Campo de tipo texto (varchar) que possui a função de guardar o nome ou designação do utilizador.

- sEmail

Campo de tipo texto (varchar) que possui a função de guardar o endereço de correio electrónico associado ao utilizador. Este campo não pode ter valores repetidos na tabela.

- sPhone

Campo de tipo texto (varchar) que possui a função de guardar um numero de telefone ou de contato para o utilizador.

- sPalavraPasse

Campo de tipo texto (varchar) que possui a função de guardar a cadeia de caracteres gerada pelo algoritmo de codificação utilizado no sistema de backend para encriptação da palavra-passe indicada pelo utilizador no momento do seu registo em sistema.

- **bEstado**

Campo de tipo booleano que possui a função de guardar o estado do utilizador em sistema, nomeadamente indicando se o utilizador está ativo em sistema (1) ou inativo (0).

- **iTipoUtilizador**

Campo de tipo inteiro que possui como função a tipificação do utilizador em sistema. Este campo é uma chave estrangeira que se relaciona com a chave primaria da tabela “uaw\_tiposutilizadores” através do seu campo iTipoUtilizador.

- **dtRegistro**

Campo de tipo de data e hora que tem como função guardar a data e a hora em que o utilizador foi registado em sistema

| <b>uaw_utilizadores</b> |              |
|-------------------------|--------------|
| ⌚ <b>idUtilizador</b>   | int(11)      |
| sNome                   | varchar(120) |
| * sEmail                | varchar(100) |
| sPhone                  | varchar(20)  |
| sPalavraPasse           | varchar(128) |
| bEstado                 | boolean      |
| iTipoUtilizador         | int(11)      |
| dtRegistro              | datetime     |

Figura 37 - Tabela Utilizadores

## **uaw\_perfis**

Tabela onde são registados os perfis dos utilizadores registados em sistema é composta pelos seguintes quinze campos:

- IdPerfil

Campo de tipo inteiro que possui como função a identificação de um perfil em sistema. Este campo é uma chave primária na tabela.

- IdUtilizador

Campo de tipo inteiro que identifica a que utilizador o corrente perfil está associado.

Este campo é uma chave estrangeira que se relaciona com a tabela “uaw\_utilizadores” através do seu campo idUtilizador, sendo também campo de valor único na corrente tabela.

- byfoto

Campo de tipo blob (array de bytes) que possui a função de guardar a fotografia do utilizador.

- byBackground

Campo de tipo blob (array de bytes) que possui a função de guardar uma imagem para papel de parede do perfil do utilizador.

- iCodigoCurso

Campo de tipo inteiro que identifica o curso de referência do utilizador.

Este campo é uma chave estrangeira da tabela “uaw\_axCursos” através do campo iCodigoCurso da tabela “uaw\_axCursos”

- iAnoCurso

Campo de tipo inteiro que identifica o ano de finalização do código de curso.

O valor deste campo deverá estar situado entre o valor do campo iAnoMinimo e IAnoMaximo do registo referenciado no campo iCodigoCurso da tabela “uaw\_axCursos” que se relaciona com o valor do corrente registo nesta tabela do campo iCodigoCurso.

- sGrauAcademico

Campo de tipo texto (varchar) que possui a função de guardar o grau académico indicado pelo utilizador no registo do seu perfil.

- sMorada

Campo de tipo texto (varchar) que possui a função de guardar a morada indicado pelo utilizador no registo do seu perfil.

- iCP1 e iCP2

Campos de tipo inteiro que possui como função guardar as referências do código postal associado á morada, iCP1 refere-se aos primeiros quatro dígitos e o iCP2 refere-se aos seguintes três dígitos.

- sLocalidadeCP

Campo de tipo texto (varchar) que possui a função de guardar a localidade do código postal afecto á morada indicada.

#### dtRegisto

Campo de tipo de data e hora que tem como função guardar a data e a hora em que o perfil do utilizador foi registado em sistema

- dtUltimaAlteracao

Campo de tipo de data e hora que tem como função guardar a data e a hora em que da última alteração ao corrente perfil do utilizador.

| uaw_perfis        |               |
|-------------------|---------------|
| ✓ idPerfil        | int(11)       |
| ⌘ idUtilizador    | int(11)       |
| byFoto            | int?          |
| sBiografia        | varchar?      |
| byBackGround      | int           |
| iCodigoCurso      | int(11)?      |
| iAnoCurso         | int(11)?      |
| sGrauAcademico    | varchar(150)? |
| sMorada           | varchar(150)? |
| iCP1              | int(4)?       |
| iCP2              | int(3)?       |
| sLocalidadeCP     | varchar(100)? |
| sNIF              | varchar(9)?   |
| dtRegisto         | datetime      |
| dtUltimaAlteracao | datetime      |

Figura 38 - Tabela Perfis

## **uaw\_portofolio**

Tabela onde são registados os itens que constituem o Portfólio de um utilizador a tabela é constituída pelos seguintes 12 campos:

- jdPortofolio

Campo de tipo inteiro que possui como função a identificação de um item de Portfólio afeto a um utilizador. Este campo é uma chave primária na tabela.

- idUtilizador

Campo de tipo inteiro que identifica a que utilizador o corrente item de portofolio está associado.

Este campo é uma chave estrangeira que se relaciona com a tabela “uaw\_utilizadores” através do seu campo idUtilizador.

- iTipoPortofolio

Campo de tipo inteiro que que identifica o tipo do item de registo de Portfólio (se imagem, vídeo, etc..)

Este campo é uma chave estrangeira que se relaciona com a tabela “uaw\_axTipoPortofolio” através do seu campo iTipoPortofolio.

- sTitulo

Campo de texto (varchar) que recebe o título que o utilizador atribui ao seu registo de item de Portfólio.

- sBody

Campo de texto (varchar) que recebe o corpo do registo de item do Portfólio normalmente uma descrição.

- sEtiquetas

Campo de texto (varchar) que recebe as etiquetas de catalogação do item por parte do utilizador para contexto de pesquisa.

- sUrl

Campo de texto (varchar) que recebe o link de acesso para o documento (externo) ao qual o item de Portfólio se encontra associado.

- bEstado

Campo de tipo booleano que possui a função de guardar o estado do item de Portfólio, nomeadamente indicando se o item de Portfólio se encontra ativo sistema (1) ou inativo (0).

- **dtEstado**

Campo de tipo de data e hora que tem como função guardar a data e a hora em que o item de Portfólio ficou ativo ou inativo.

- **idLoginEstado**

Campo de tipo inteiro que identifica o utilizador que altera o estado do item de Portfólio.

Este campo é uma chave estrangeira que se relaciona com a tabela “uaw\_utilizadores” através do seu campo idUtilizador.

- **dtRegisto**

Campo de tipo de data e hora que tem como função guardar a data e a hora em que o item de portfolio foi registado em sistema

- **idLogin**

Campo de tipo inteiro que identifica o utilizador que executou o registo de item de Portfólio.

Este campo é uma chave estrangeira que se relaciona com a tabela “uaw\_utilizadores” através do seu campo idUtilizador.

## **uaw\_axtiposutilizador**

Tabela onde são registadas as tipificações de tipos de utilizador em sistema, composta pelos seguintes dois campos:

- **iTipoUtilizador**

Campo de tipo inteiro que possui como função a tipificação do utilizador em sistema. Este campo é uma chave primaria que se relaciona com a chave estrangeira da tabela “uaw\_utilizadores” através do campo iTipoUtilizador.

- **sDescTipoUtilizador**

Campo de texto (varchar) que recebe a designação atribuída ao código de tipo de utilizador.

| <b>uaw_portfolio</b> |               |
|----------------------|---------------|
| idPortfolio          | int(11)       |
| idUtilizador         | int(11)       |
| iTipoPortofolio      | int           |
| sTitulo              | varchar(100)  |
| sBody                | varchar(3000) |
| sEtiquetas           | varchar(500)  |
| sURL                 | varchar(250)  |
| bEstado              | boolean       |
| dtEstado             | datetime      |
| idLoginEstado        | int(11)       |
| dtRegisto            | datetime      |
| idLogin              | int(11)       |

Figura 39 - Tabela Portfólio

## **uaw\_axcursos**

Tabela onde são registados os cursos existentes da instituição académica em termos de designação assim como o ano de início do curso e de ano de fim da existência do curso na instituição, a tabela comporta os seguintes quatro campos:

- **iCodigoCurso**

Campo de tipo inteiro que possui como função de identificação em sistema de um curso da instituição académica ao qual a base de dados pertence.

Este campo é uma chave primária que se relaciona com a chave estrangeira da tabela “uaw\_perfis” através do campo iCodigoCurso.

- sDescCurso

Campo de texto (varchar) que recebe a designação do curso atribuído ao código identificador do curso

- iAnoMinimo

Campo de tipo inteiro que possui como função a identificação do ano em que o curso começou a ser lecionado na instituição académica.

- iAnoMaximo

Campo de tipo inteiro que possui como função a identificação do último ano em que o curso foi lecionado na instituição académica, em caso de ainda ser correntemente lecionado o valor a atribuir ao campo deverá ser um valor nulo.



Figura 40 - Tabela Auxiliar Tipo de Cursos

## **uaw\_axportofolio**

Tabela onde são registadas as tipificações de tipos de itens de Portfólio em sistema, composta pelos seguintes dois campos:

- iTipoPortofolio

Campo de tipo inteiro que possui como função de identificação em sistema de um curso da instituição académica ao qual a base de dados pertence.

Este campo é uma chave primaria que se relaciona com a chave estrangeira da tabela “uaw\_portofolio” através do campo iTipoPortofolio.

- SDescCurso

Campo de texto (varchar) que recebe a designação do tipo de Portfólio relativo ao código de tipificação de item de Portfólio.

| uaw_axtipoportfolio |              |
|---------------------|--------------|
| iTipoPortofolio     | int          |
| sDescTipoPortofolio | varchar(100) |

Figura 41 - Tabela Auxiliar Tipo de Portfólio

## Diagrama da Base de Dados.

O diagrama abaixo demonstra a definição da estrutura da base de dados assim como os relacionamentos existentes nas tabelas que constroem a base de dados.



Figura 42 - Diagrama Geral da Base de Dados

Como se pode observar existe um relacionamento do tipo de 1:1 no que se refere às tabelas *uaw\_perfil* com a tabela *uaw\_utilizadores* uma vez que só poderá existir um registo perfil para cada utilizador ou seja um utilizador não poderá ter dois perfis em sistema. Um segundo relacionamento também muito importante é o relacionamento de N:1 entre as tabelas *uaw\_Utilizadores* e *uaw\_Portofolio* que indicam que um utilizador pode possuir vários registo de Portfólio sendo esse conjunto de itens afetos ao utilizador que se entende como um Portfólio.

De notar que a tabela *uaw\_Portofolio* possui três relacionamentos com a tabela *uaw\_Utilizadores* para três campos diferentes que, no entanto, se reportam a identificação de um utilizador.

Existem depois outros dois relacionamentos de N:1 relativamente a tabelas auxiliares ou de tipificação com as tabelas principais, nomeadamente o relacionamento entre a tabela *uaw\_axtiposUtilizador* com a tabela *uaw\_Utilizadores* e a tabela *uaw\_axtipoportfolio* com a tabela *uaw\_Portofolio*.

Todos estes relacionamentos têm como objetivo garantir a consistência dos registo na base de dados.

No diagrama também estão identificados os tipos de campos e os respetivos tamanhos assim como a identificação se os campos são chave primaria da tabela através do símbolo (  ) ou se são indexados e de valor único através do símbolo (  ), é possível também identificar os campos que poderão ter valores nulos nas respetivas tabelas através do símbolo (  ),

## Conclusão.

A base de dados agora descrita corresponde a uma estrutura mínima para execução do projeto na situação atual e para uma instituição académica e que na sua simplicidade garante não só a consolidação de dados e da informação nela contida como também a correta implementação das regras de negócio subjacentes ao projeto.

Em termos futuros e com o desenvolvimento do projeto a estrutura aqui descrita garante também que se possa evoluir a estrutura com total transparência para os utilizadores do sistema.

O sistema prevê que em caso de futuras associações de instituições académicas cada instituição usufruirá de uma instância própria da designação

## ANEXO 5 – SEGURANÇA

### Introdução

Este documento descreve em detalhe as medidas de segurança implementadas no projeto integrador de engenharia informática. A segurança em sistemas de informação é de extrema importância, pois protege os dados contra acessos não autorizados, garante a integridade das informações e assegura a privacidade dos utilizadores. Neste projeto, focamo-nos em estabelecer um ambiente seguro para o armazenamento e transmissão de dados, utilizando várias técnicas e práticas recomendadas no campo da segurança da informação.

A autenticação de utilizadores é realizada utilizando tokens JWT (JSON Web Tokens), uma tecnologia moderna e segura que permite a verificação de identidade sem a necessidade de armazenar estado no servidor. Além disso, as senhas dos utilizadores são criptografadas usando o algoritmo BCrypt, garantindo que mesmo em caso de comprometimento do banco de dados, as senhas reais dos utilizadores permaneçam protegidas.

O Spring Security, uma poderosa biblioteca de segurança para aplicações Java, é amplamente utilizado para configurar e gerenciar a segurança do sistema. Este documento fornece uma visão abrangente das várias configurações e técnicas de segurança implementadas, com explicações claras e detalhadas que tornam possível a compreensão mesmo para leitores sem formação técnica.

### Autenticação e Autorização

#### JSON Web Token (JWT)

No nosso projeto, utilizamos JSON Web Tokens (JWT) para autenticação de utilizadores. O JWT é um padrão aberto que define uma maneira compacta e segura de transmitir informações entre as partes em formato JSON. As informações contidas no token são assinadas digitalmente, o que garante que possam ser verificadas e confiáveis.

#### Como funciona?

1. **Geração do Token:** Quando um utilizador faz login no sistema, suas credenciais são verificadas. Se forem válidas, o sistema gera um token JWT que contém informações sobre o utilizador, como seu ID e nome de utilizador.
2. **Assinatura:** Esse token é assinado digitalmente usando uma chave secreta armazenada no servidor. Essa assinatura garante que o token não possa ser alterado sem ser detectado.

3. **Envio ao Cliente:** O token JWT é então enviado ao cliente (por exemplo, o navegador web do utilizador).
4. **Autenticação em Requisições Futuras:** Em requisições subsequentes, o cliente envia o token JWT ao servidor. O servidor verifica a assinatura do token para garantir que é válido e, se for, permite que o utilizador acesse os recursos solicitados.

## Implementação Técnica

- **JwtAuthenticationFilter:** Este filtro intercepta todas as requisições HTTP para verificar a presença e a validade do token JWT. Se o token for válido, o utilizador é autenticado e pode acessar os recursos protegidos.
- **JwtService:** Serviço responsável por gerar, validar e extraír informações dos tokens JWT. Ele utiliza uma chave secreta configurada no arquivo de propriedades para assinar os tokens.

## Configuração de Segurança

Utilizamos o Spring Security, uma poderosa biblioteca de segurança para aplicações Java, para configurar e gerenciar a segurança do sistema.

### Principais Configurações:

- **Configuração de CORS (Cross-Origin Resource Sharing):** Permite controlar quais origens podem acessar os recursos do servidor, prevenindo requisições não autorizadas de outras origens.
- **Desabilitação de CSRF (Cross-Site Request Forgery):** CSRF é uma técnica de ataque que força um utilizador final a executar ações indesejadas em uma aplicação web na qual está autenticado. Para nossa API, desabilitamos CSRF porque estamos utilizando tokens JWT, que são imunes a este tipo de ataque.
- **Política de Sessão Stateless:** A aplicação não mantém estado de sessão no servidor. Cada requisição é tratada de forma independente, utilizando o token JWT para autenticação.
- **Manejo de Exceções:** Configura um ponto de entrada para tratamento de exceções de autenticação, garantindo que respostas apropriadas sejam enviadas em caso de falhas de segurança.

## Criptografia de Senhas

A segurança das senhas dos utilizadores é fundamental. Para garantir que as senhas sejam armazenadas de forma segura, utilizamos a criptografia com o algoritmo BCrypt. Este algoritmo incorpora um sal (salt) e realiza múltiplas iterações, tornando as senhas extremamente difíceis de serem quebradas por ataques de força bruta.

### Como funciona?

5. **Sal (Salt):** Um valor aleatório é adicionado à senha antes de ser criptografada. Isso garante que mesmas senhas iguais terão resultados criptografados diferentes.
6. **Múltiplas Iterações:** O algoritmo BCrypt realiza várias iterações de criptografia, aumentando o tempo necessário para gerar o hash da senha. Isso torna mais difícil e demorado para um atacante descobrir a senha original.

### Implementação Técnica

- **BCryptPasswordEncoder:** Bean que fornece a implementação do algoritmo BCrypt para criptografia de senhas.

## Tratamento de Exceções

Para garantir que a aplicação lide de maneira adequada com erros e exceções, implementamos um manipulador global de exceções. Este manipulador captura e trata diversas exceções, fornecendo respostas claras e seguras.

### Principais Exceções Tratadas:

- **BadCredentialsException:** Lançada quando as credenciais de autenticação (como usuário e senha) são inválidas.
- **AccountStatusException:** Lançada quando a conta do usuário está bloqueada ou inativa.
- **AccessDeniedException:** Lançada quando o usuário tenta acessar um recurso sem ter a devida autorização.
- **SignatureException:** Lançada quando a assinatura do token JWT é inválida.
- **ExpiredJwtException:** Lançada quando o token JWT expirou.

## Comunicação Segura

A comunicação entre o frontend e o backend é realizada de maneira segura utilizando tokens JWT e HTTPS. O HTTPS (HyperText Transfer Protocol Secure) garante que os dados transmitidos entre o cliente e o servidor sejam criptografados, prevenindo que sejam interceptados por terceiros.

## Configuração de CORS

Controlamos quais origens podem acessar os recursos do servidor através de uma configuração de CORS. Isso ajuda a prevenir requisições não autorizadas de outras origens, aumentando a segurança da aplicação.

## Conclusão

As medidas de segurança implementadas neste projeto garantem que as informações dos utilizadores sejam protegidas contra acessos não autorizados e que a integridade e confidencialidade dos dados sejam mantidas. Utilizando tecnologias modernas como JWT para autenticação e BCrypt para criptografia de senhas, aliadas a uma configuração robusta de segurança do Spring Security, conseguimos criar um ambiente seguro e confiável para os utilizadores.

Através destas práticas, asseguramos que o sistema é capaz de resistir a diversas ameaças, fornecendo uma base sólida para a proteção de dados e mantendo a confiança dos utilizadores na integridade do sistema. A segurança da informação é uma prioridade constante, e continuaremos a evoluir e melhorar as nossas práticas para enfrentar novos desafios e ameaças emergentes.

## ANEXO 6 – TESTES

### TESTES EM BACKEND:

- Testes:  
**Configuração Inicial:**
- Utilizamos @SpringBootTest para carregar o contexto do Spring Boot e @AutoConfigureMockMvc para configurar o MockMvc, que é utilizado para simular requisições HTTP.
- O método setup é utilizado para qualquer configuração que precise ser feita antes de cada teste (atualmente está vazio).
- **Testes de UserController:**
- **testCreateUser:** Verifica se a criação de um utilizador retorna um status OK.
- **testDeleteUser:** Verifica se a exclusão de um utilizador retorna um status OK.
- **test GetUser:** Verifica se a recuperação de um utilizador retorna um status OK e o tipo de conteúdo é JSON.
- **testListUsers:** Verifica se a listagem de utilizadores retorna um status OK e o tipo de conteúdo é JSON.
- **testUpdateUser:** Verifica se a atualização de um utilizador retorna um status OK.
- **Testes de AuthController:**
- **testRegister:** Verifica se o registo de um novo utilizador retorna um status OK.
- **testLogin:** Verifica se o login de um utilizador retorna um status OK.
- **testLogout:** Verifica se o logout de um utilizador retorna um status OK.