



UNIVERSIDADE D
COIMBRA

Tiago Daniel Fonseca Pessoa

**DESENVOLVIMENTO DE SOFTWARE PARA
ATIVIDADES LÚDICO-DIDÁTICAS VIRTUAIS
SOBRE A UNIÃO EUROPEIA**

VOLUME 1

**Dissertação no âmbito do Mestrado em Engenharia Informática,
especialização em Engenharia de Software, orientada pelo
Professor Doutor Paulo Rupino da Cunha, do Departamento de
Engenharia Informática da Faculdade de Ciências e Tecnologia da
Universidade de Coimbra, pela Professora Doutora Maria Raquel
Freire e pela Professora Doutora Teresa Almeida Cravo, ambas da
Faculdade de Economia da Universidade de Coimbra, e
apresentada à Faculdade de Ciências e Tecnologia / Departamento
de Engenharia Informática.**

junho de 2021

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Desenvolvimento de software para atividades lúdico-didáticas virtuais sobre a União Europeia

Tiago Daniel Fonseca Pessoa

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software, orientada pelo Professor Doutor Paulo Rupino da Cunha, do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, pela Professora Doutora Maria Raquel Freire e pela Professora Doutora Teresa Almeida Cravo, ambas da Faculdade de Economia da Universidade de Coimbra, e apresentada à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

junho 2021



UNIVERSIDADE D
COIMBRA

Resumo

Nos últimos anos, a exposição de crianças e adolescentes a ecrãs cresceu exponencialmente. Os meios audiovisuais, nomeadamente, a internet, podem ser benéficos ao processo de aprendizagem, facultando uma forma de obter conhecimento, para além dos métodos tradicionais. Em particular, as plataformas lúdico-didáticas podem ajudar os seus visitantes mais jovens a aprender sobre determinados temas e expandir conceitos, utilizando o entretenimento como forma de aprendizagem enquanto jogam, desenvolvem atividades ou assistem a vídeos.

O Centro de Excelência Jean Monnet da Universidade de Coimbra tem como tema as relações da União Europeia com os vizinhos a Leste, contendo um conjunto alargado de iniciativas como: ensino e investigação; recomendações políticas; e ligação à comunidade. A proposta apresentada por esta entidade consistiu na necessidade de ser desenvolvido um software lúdico-didático que permita, de forma interativa, aproximar os mais jovens do projeto europeu, motivando, desde cedo, a familiaridade e o gosto por temas da União Europeia.

Neste relatório são apresentadas as várias etapas que levaram ao desenvolvimento de uma plataforma com atividades lúdico-didáticas, vocacionada para crianças entre os 6 e 9 anos, com o intuito de transmitir conhecimentos sobre a Europa e a União Europeia, expondo as atividades do Centro. Esta foi concebida no formato de uma plataforma web que aloja um conjunto de jogos e atividades lúdicas. Tanto a plataforma como os jogos foram desenvolvidos de raiz. Das atividades lúdicas, fizeram parte vídeos didáticos com tema da União Europeia e um livro que conta uma história infantil, ambos produzidos por equipas externas. Ainda, podem ser alojados trabalhos que conseguem ser submetidos a votações online por parte do público, promovendo a realização de competições. Para gerir o conteúdo presente na plataforma, e esta poder ser atualizada a qualquer altura, existe uma página de administração. Esta página permite ainda ver métricas recolhidas anonimamente de jogos, possibilitando um método de estudo sobre a eficácia destes.

Palavras-Chave

União Europeia, Plataforma lúdico-didática, Desenvolvimento Web, Desenvolvimento de Jogos, Crianças

Abstract

In the past few years, the exposure to screens of kids and teenagers has exponentially increased. Audiovisual methods, mainly, the internet, can be beneficial for the learning process, giving a way to obtain knowledge, beyond the traditional ways. Particularly, the platforms of a leisure and didactic nature can help the younger visitors acquire knowledge about specific themes and expand concepts, using entertainment as a way of learning while playing games, developing activities or watching videos.

The relationships with the European Union and its neighbours to the East, it's the theme of the Jean Monnet Centre of Excellence of the University of Coimbra. It has a wide spectrum of initiatives such as: education and investigation; political recommendations; and bond with the community. The proposal presented by this identity was to develop a platform of a leisure and didactic nature, that allows closeness of the younger ones to the European projects, in an interactive way. This will motivate, in an early age, the familiarity and taste for themes of the European Union.

In this report, it will be presented the phases that led to the development of a platform with a leisure and didactic nature, with a focus age between 6 to 9 years old. The intention was to provide knowledge about Europe and the European Union by exhibiting the activities of the Center of Excellence. This was created on a web platform format that combines a set of games and leisure activities. Both the platform and games were developed from scratch. Within the leisure activities, were included didactic videos with the European Union theme and a children's story book, both produced by external teams. It also can host works that can be submitted to online votes from the audience, promoting competitions. In order to manage and update at any time the content inside the platform, it was created an administration page. This page can also give insight about metrics gathered anonymously from games, providing the possibility to study the effectiveness of those games.

Keywords

European Union, Leisure and Didactic Platform, Web Development, Game Development, Kids

Índice

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.3	Estrutura do documento	2
2	Estado da arte	4
2.1	Conceitos	4
2.2	Análise de Software existente	5
2.2.1	Plataformas	5
2.2.2	Análise de conteúdos de plataforma	7
2.2.3	Conclusão do analisado	13
3	Metodologia, Planeamento e Gestão de Riscos	14
3.1	Metodologia	14
3.2	Planeamento	15
3.2.1	Primeiro Semestre	15
3.2.2	Segundo Semestre	16
3.3	Gestão de Riscos	17
3.3.1	Análise de Risco	17
3.3.2	Matriz de Risco	19
4	Requisitos	20
4.1	Personas	20
4.2	User Stories	23
4.3	Requisitos Funcionais	24
4.3.1	Casos de Uso	26
4.3.2	Protótipos de ecrã	27
4.3.3	Requisitos Não Funcionais	27
5	Arquitetura e Tecnologias	29
5.1	Tecnologias de Desenvolvimento	29
5.1.1	Front-end	29
5.1.2	Back-end	33
5.1.3	Tecnologias de Ambiente e Exploração	35
5.1.4	Licenças	35
5.1.5	Validação de Requisitos Não Funcionais	36
5.2	Arquitetura	36
5.2.1	Modelo Arquitetural	36
5.2.2	Modelação de Dados	40
6	Implementação e Resultados Finais	43
6.1	Configuração do Ambiente	43
6.1.1	Software externo	43
6.1.2	Instalação do software	44
6.2	Desenvolvimento dos jogos	44
6.2.1	Jogos desenvolvidos de raiz	44
6.2.2	Jogos desenvolvidos recorrendo a bibliotecas externas	50

6.3	Desenvolvimento da plataforma	52
6.3.1	Página Home	52
6.3.2	Página Games	52
6.3.3	Página Vídeos	53
6.3.4	Página Book	53
6.3.5	Página Poll	54
6.3.6	Página Login	55
6.3.7	Página Admin	55
6.4	Requisitos Implementados	61
7	Testes ao Software	62
7.1	Testes Funcionais	62
7.2	Testes Não Funcionais	64
7.2.1	Testes à Usabilidade	64
7.2.2	Testes à Compatibilidade	66
7.2.3	Testes à Escalabilidade	67
7.2.4	Testes à Segurança	67
8	Conclusão	68
8.1	Experiência Adquirida	68
8.2	Trabalho Futuro	68
8.3	Considerações Finais	69
	Referências	70
	Anexos	74
A	Casos de Uso	75
B	Protótipos de ecrã de Baixa Fidelidade	80
B.1	Protótipos Jogos	80
B.2	Protótipos Website	81
C	Licenças das Tecnologias Utilizadas	84
D	Tutorial de Instalação do Software	86
E	Requisitos Implementados	87
F	Testes Funcionais	89
F.1	Funcionalidades Gerais	89
F.2	Página de Jogos	90
F.3	Página de Vídeos	90
F.4	Página do Livro Infantil	90
F.5	Página de Votação	91
F.6	Página de Administração	92
G	Tarefas fornecidas para testar a Usabilidade	94
G.1	Tarefas para Adultos	94
G.2	Tarefas para Crianças	96

Acrónimos

CRUD Create Read Update Delete. 2, 6, 24, 38, 57

DEI Departamento de Engenharia Informática. 1, 35, 43

ER Entity-relationship. 40

JWT JSON Web Token. 34, 55, 60, 61, 67

NPM Node Package Manager. 31, 35

ODM Object Data Modeling. 33

PRONE Peace Relations, Ontologies and Narratives in Europe: EU and its Eastern Neighbours.
1

UC Universidade de Coimbra. 1, 2, 12, 14, 19, 21, 23, 35, 69

UE União Europeia. 1, 4–6, 8–12

Lista de Figuras

2.1	Espaço de Aprendizagem da plataforma [9]	5
2.2	Plataforma da National Geographic [12]	6
2.3	Exemplo do jogo Puzzle [13]	8
2.4	Exemplo do jogo de Colorir [14]	8
2.5	Exemplo do jogo de Sopa de Letras [15]	9
2.6	Exemplo do jogo Quiz [16]	9
2.7	Exemplo do jogo de Memória [17]	10
2.8	Exemplo do jogo de Mapas Interativos [18]	10
2.9	Exemplo do jogo de Palavras Cruzadas [20]	11
2.10	Exemplo da disposição dos vídeos que se pretende [21]	12
2.11	Exemplo de um livro infantil virtual [22]	12
2.12	Exemplo da disposição do conteúdo que se pretende [23]	13
3.1	Matriz de gestão dos riscos acima descritos	19
5.1	Documentos com diferentes objectos no campo config	34
5.2	Diagrama de Contexto	37
5.3	Diagrama de Contentores	37
5.4	Organização das diretorias no React	38
5.5	Organização das diretorias do Servidor	39
5.6	Diagrama do modelo de dados utilizado	40
5.7	Exemplo do documento de uma estatística correspondente ao jogo Quiz	41
6.1	Cenas presentes nos jogos desenvolvidos	44
6.2	Jogo de colorir desenvolvido	45
6.3	Jogo Puzzle desenvolvido	46
6.4	Jogo Sopa de Letras desenvolvido	48
6.5	Jogo Quiz desenvolvido	48
6.6	Jogo de Memória desenvolvido	49
6.7	Jogo de Mapas Interativos	50
6.8	Jogo de Palavras Cruzadas	51
6.9	Página inicial da plataforma	52
6.10	Página dos Jogos	53
6.11	Página dos Vídeos	53
6.12	Página do Livro Infantil	54
6.13	Página das votações	54
6.14	Página de login	55
6.15	Gerir Jogos	56
6.16	Gerir Vídeos	56
6.17	Gerir Votações	57
6.18	Formulário	57
6.19	Esquema de validação do formulário do Puzzle	58
6.20	Estatísticas de um jogo Quiz	59
6.21	Estatística de um jogo Puzzle	59
6.22	Alterar o perfil	60
B.1	Protótipos dos jogos	80

B.2	Protótipos dos jogos	81
B.3	Protótipos do Website	82
B.4	Protótipos do Website	83
D.1	Tutorial de configuração e instalação da plataforma	86
G.1	Exemplo de como fazer um jogo de Palavras Cruzadas	95

Lista de Tabelas

3.1	Planeamento Primeiro Semestre	15
3.2	Planeamento Segundo Semestre	16
3.3	Definição do Risco - R1	17
3.4	Definição do Risco - R2	18
3.5	Definição do Risco - R3	18
3.6	Definição do Risco - R4	19
4.1	Persona 1	21
4.2	Persona 2	22
4.3	Persona 3	23
4.4	User Stories	24
4.5	Requisitos Funcionais	26
4.6	Caso de uso UC-2	26
5.1	Tabela de comparação de motores e frameworks de jogos	30
5.2	Resumo da comparação entre React, Angular e Vue	31
7.1	Exemplo de um Teste Funcional	62
7.2	Exemplo de um Teste Adhoc	63
7.3	Estatísticas Puzzle - Esperado	63
7.4	Estatísticas Quiz - Esperado	64
7.5	Problema de usabilidade detetado - 1	66
7.6	Problema de usabilidade detetado - 2	66
A.1	Caso de uso UC-1	75
A.2	Caso de uso UC-3	75
A.3	Caso de uso UC-4	76
A.4	Caso de uso UC-5	77
A.5	Caso de uso UC-6	77
A.6	Caso de uso UC-7	78
A.7	Caso de uso UC-8	79
C.1	Tabela de Licenças das Tecnologias Utilizadas (front-end)	84
C.2	Tabela de Licenças das Tecnologias Utilizadas (back-end)	85
E.1	Requisitos Funcionais Implementados	88

Capítulo 1

Introdução

Esta dissertação, intitulada Desenvolvimento de software para atividades lúdico-didáticas virtuais sobre a União Europeia, descreve o trabalho efetuado durante o primeiro e o segundo semestre do ano letivo de 2020/2021, no âmbito do programa de Mestrado em Engenharia de Software da Universidade de Coimbra (UC) e de um estágio financiado pelo Centro de Excelência Jean Monnet da UC - Peace Relations, Ontologies and Narratives in Europe: EU and its Eastern Neighbours (PRONE). É em resposta aos objetivos deste projeto específico do Centro de Excelência, e fruto de uma articulação próxima entre o Departamento de Engenharia Informática (DEI) e a Faculdade de Economia da UC, que o produto final de criação de uma plataforma lúdico-didática sobre a União Europeia (UE) foi desenvolvido.

1.1 Motivação

O Centro de Excelência Jean Monnet da Universidade de Coimbra, criado no início do ano letivo de 2019-2020, tem como objectivo principal analisar a UE enquanto ator de paz e segurança, com enfoque nas suas relações com a Rússia e os países a Leste. No âmbito das suas atividades, o Centro pretende igualmente abrir-se à sociedade civil e às comunidades estudantis, desenvolvendo sinergias com várias instituições que permitam alargar o conhecimento geral sobre a Europa. Nesse sentido, foi proposto o desenvolvimento de um software para atividades lúdico-didáticas para crianças do ensino primário, entre os 6 e 9 anos, centrado em temas da UE.

Nos últimos anos, a exposição de crianças e adolescentes a ecrãs cresceu exponencialmente. Os meios audiovisuais, nomeadamente, a internet, podem ser benéficos ao processo de aprendizagem, facultando uma forma de obter conhecimento, para além dos métodos tradicionais. Em particular, as plataformas lúdico-didáticas podem ajudar os seus visitantes mais jovens a aprender sobre determinados temas, expandir conceitos, compreender eventos históricos, culturais, utilizando o entretenimento como forma de aprendizagem enquanto jogam, desenvolvem atividades ou assistem a vídeos. No contexto ainda incerto da atual pandemia, as plataformas online lúdico-didáticas têm vindo a ser ainda mais utilizadas pelo público mais jovem, pois permitem que o tempo que as crianças passam confinadas em casa e, conseqüentemente, em frente a ecrãs, seja aproveitado para adquirirem conhecimentos, de forma divertida e relaxada.

Pessoalmente, enquanto estudante, a participação num projeto desta importância permitiu, por um lado, pôr à prova todos os conhecimentos adquiridos ao longo do curso. Por outro, desenvolver jogos virtuais foi um desafio bastante interessante, pois é uma área que ainda hoje me acompanha, enquanto jogador, e, pode, nesse sentido, aproveitar a minha sensibilidade e experiência do ponto de vista de utilizador para criar uma plataforma o mais intuitiva possível. Além disso, saber que o projeto terá um impacto no mundo real e que vai permitir sensibilizar crianças para os temas da paz, da cooperação, da vizinhança e do conhecimento do outro, como pretende o Centro de Excelência, é uma grande motivação para realizar uma plataforma útil e agradável, que possa ser utilizada pelo maior número de pessoas.

1.2 Objetivos

O objetivo desta tese é desenvolver de raiz uma plataforma online lúdico-didática com atividades sobre a Europa e a UE, em língua portuguesa, vocacionada para crianças do 1º ciclo do ensino básico, entre os 6 e os 9 anos de idade. O propósito é expor os mais novos a estes temas, de uma maneira interativa e relaxada. Esta plataforma não pretende ser utilizada para efeitos de avaliação curricular formal em contexto de sala de aula. Não há, nesse sentido, um programa de objetivos de aprendizagem que tenha que ser cumprido e avaliado. O administrador da plataforma, a partir da sua criação, será um/a investigador/a associado/a ao Centro de Excelência Jean Monnet, que ficará responsável por oportunamente atualizar os seus conteúdos.

Esta plataforma irá conter um conjunto de jogos do tipo quebra-cabeças [1], vídeos, um livro virtual e uma secção onde se vão realizar votações de trabalhos manuais infantis. Terá ainda uma página de administração, que vai permitir gerir o conteúdo mencionado.

Os jogos serão desenvolvidos de raiz, numa linguagem web, permitindo uma fácil integração com a plataforma online. Serão independentes, com objetivos e interfaces diferentes, partilhando apenas em comum a cena gráfica a congratular caso o jogador ganhe ou a informar caso perca. No estado da arte, na subsecção 2.2, serão listados os jogos que vão fazer parte da plataforma e o motivo da sua escolha. Da parte do administrador, o objetivo será cada jogo ter parâmetros editáveis. Isto vai permitir atualizar e aumentar o número de jogos existentes, com diferentes níveis de dificuldades, a qualquer altura, com conteúdo variado sobre a União Europeia.

A página de vídeos vai conter vídeos produzidos pelo Centro Jean Monnet da Universidade de Coimbra, que vão estar alojados no seguinte canal de Youtube [2]. Pode também conter vídeos de outros canais. O propósito aqui será ter os vídeos organizados por categorias e prontos a serem reproduzidos nesta página da plataforma. A cargo da página de administração, vai ser possível fazer o Create Read Update Delete (CRUD) de cada categoria de vídeos e os respetivos vídeos associados.

A página do livro virtual irá conter um livro em formato digital, que vai ser possível folhear para ler a sua história. Este livro será produzido pela equipa de design [3], que irá fornecer as imagens das ilustrações que vão compor cada página. O objetivo será procurar as ferramentas web necessárias que permitam unir as imagens de modo a dar uma experiência ao utilizador de um livro digital.

Por fim, a página de votação vai conter várias categorias e, cada uma, vai conter diferentes trabalhos. Vai ser possível votar num ou mais trabalhos à escolha, através de um email e nome. O administrador vai poder encerrar uma votação e encontrar um vencedor, bem como fazer o CRUD das categorias e dos trabalhos presentes em cada categoria.

Sem esquecer que, devido ao intervalo de idades para o qual a aplicação vai ser desenvolvida, deve-se atender a um design e experiência de utilizador ajustado, bem como garantir a responsividade em vários ecrãs, de modo a poder ser utilizada na maioria dos dispositivos.

A expectativa é que o sistema fique alojado no ecossistema do Centro de Excelência Jean Monnet da UC.

1.3 Estrutura do documento

Esta dissertação está dividida em oito capítulos, que descrevo em seguida. Ao presente capítulo, que corresponde à Introdução e apresenta a motivação e os objetivos gerais da dissertação, segue-se:

- Capítulo 2 - *Estado da Arte*, onde são expostos conceitos relevantes a ter em conta no processo de desenvolvimento dos jogos e da plataforma, para o grupo de idade entre 6 e 9 anos, fruto da pesquisa realizada sobre este tema. É também efetuado um estudo do software existente ao nível de plataformas e jogos lúdico-didáticos.
- Capítulo 3 - *Planeamento, Metodologia e Gestão de Riscos*, onde é descrito o planeamento que foi seguido no primeiro e no segundo semestre, bem como qual a metodologia de desenvolvimento de software utilizada. Ainda neste capítulo é realizada a gestão de risco.

- Capítulo 4 - *Requisitos*, onde estão expostas as funcionalidades que a plataforma vai conter, bem como as suas qualidades. Podem ainda ser vistos os protótipos de ecrã desenvolvidos.
- Capítulo 5 - *Arquitetura e Tecnologias*, onde é descrita a arquitetura e quais as tecnologias adotadas para o desenvolvimento da plataforma (front-end e back-end) e como estas permitem a concretização dos requisitos anteriormente definidos.
- Capítulo 6 - *Implementação e Resultados Finais*, onde são descritas as principais decisões tomadas ao nível de implementação e é mostrado o resultado final da plataforma desenvolvida.
- Capítulo 7 - *Testes ao Software*, onde são descritos os testes realizados para avaliar o comportamento do software e também comprovar que está alinhado com o que foi acordado previamente nos requisitos.
- Capítulo 8 - *Conclusão*, onde são apresentadas considerações finais sobre o trabalho realizado, assim como o trabalho futuro.

Capítulo 2

Estado da arte

Este capítulo aborda conceitos fundamentais para a construção de um software que seja apelativo para o grupo de idade pretendido, entre os 6 e os 9 anos de idade. Toda a informação foi obtida por meio de pesquisa online e de artigos, pelo que será sempre mencionada a fonte. A pesquisa versou sobre o conceito de atividades lúdico-didáticas virtuais, assim como softwares relevantes já existentes que ajudaram no desenho do sistema desenvolvido.

2.1 Conceitos

O conceito de lúdico-didático virtuais refere-se a todo o conteúdo com o intuito de ensinar, sendo, simultaneamente, divertido e recreativo, e realizado por meio de um programa de computador (Software). No caso da plataforma desenvolvida no âmbito desta dissertação, estas atividades abordam o tema da Europa e, especificamente, a União Europeia (UE). A UE é uma união económica e política composta por 27 Estados-membros independentes situados na Europa e tem como objetivo promover a paz, oferecer segurança e liberdade, combater exclusão social e discriminação, respeitar a diversidade cultural e linguística, estabelecer uma união económica e monetária cuja moeda é o euro, entre outros [4].

O enfoque deste software não é explicitamente o ensino: isto é, não se pretende que as crianças garantidamente ganhem conhecimentos mensuráveis e que esta ferramenta seja um substituto do estudo formal. Contudo, alguns conceitos de ensino têm de ser tidos em conta no processo de desenvolvimento, para que a divulgação das atividades do Centro de Excelência Jean Monnet seja transmitida com sucesso para um grupo de idade mais novo. A pesquisa teve, por isso, em conta a intenção de se passar informação sobre o tema com grande sensibilidade relativamente à aptidão de um público infantil.

Segundo o artigo *The Importance of Didactic Computer Games in the Acquisition of New Knowledge* [5], entende-se por jogo didático uma ferramenta que serve o propósito de ensinar e tem o objetivo de resolver um problema. Para isso, são necessários certos conhecimentos e habilidades, existindo uma estratégia ou uma solução que permite aumentar a probabilidade de vencer. Jogos computacionais didáticos conseguem captar facilmente a atenção dos mais novos, mantendo-os focados e interessados por um período de tempo mais longo comparativamente ao que os métodos tradicionais conseguiriam alcançar, pois a concentração necessária para atingir um objetivo (ganhar o jogo) é mais intensa. Isto faz com que sejam ferramentas didáticas úteis e eficazes. Além disso, pelo facto de atualmente a maioria das crianças estarem familiarizadas com dispositivos móveis, como smartphones ou tablets, o artigo recomenda que os jogos didáticos tenham uma versão mobile, para permitir preencher os espaços livres de tempo curto. Este facto também permite abordar questões de acesso e exclusão digital, pois criando uma versão mobile permite uma maior utilização de estudantes de vários níveis socioeconómicos, que não têm um computador. A plataforma vai ser, assim, construída ponderando o valor da maior inclusão possível do público-alvo.

O artigo *Making Educational Computer Games “Educational”* [6] tem como objetivo revelar con-

siderações que devem ser tidas em conta no processo de criar jogos efetivamente interativos. A inclusão cautelosa de detalhes sedutores [7] em jogos consiste em elementos chamativos que são inseridos ao lado do conteúdo didático com a intenção de que o interesse das crianças por esses elementos torne o conteúdo educacional mais atraente e memorável. Pesquisas realizadas neste campo dizem que estes detalhes não funcionam como o esperado, levando a que as crianças expostas a estes elementos tendessem a se lembrarem mais da aparência destes do que do conteúdo educacional. A longo prazo, esta abordagem, pode, inadvertidamente, promover atitudes negativas por parte da criança, vendo o conteúdo educacional como o "trabalho" necessário para chegar à "diversão" e não à aprendizagem [8]. Outra questão importante no desenvolvimento de um jogo é o feedback quando o jogador erra na resposta ou solicita ajuda. Nas respostas erradas, a maneira mais eficaz da criança aprender é perceber onde e porquê errou, encaminhando-a na direção certa, isto é, da resposta errada para a resposta correta.

Das pesquisas efetuadas, estes conceitos e considerações foram os que se acharam mais relevantes e que se entendeu necessário estarem presentes na análise do software existente, para filtrar os tipos de jogos pretendidos, e no processo de desenvolvimento, para evitar cometer erros que pudessem comprometer a mensagem que se pretendia transmitir.

2.2 Análise de Software existente

Esta secção realiza a análise de software em duas partes. A primeira analisa plataformas que compilam atividades lúdico-didáticas, cujo objetivo é explorá-las e apontar os seus pontos positivos e negativos, assim como frisar a sua importância neste contexto. A segunda parte analisa os conteúdos que estão presentes na plataforma a desenvolver, nomeadamente os jogos, vídeos, livro e a página de votação. Para este efeito, procurou-se encontrar conteúdos com temas sobre a Europa e a UE. Não obstante, poderão analisar-se jogos aplicados a outras temáticas para identificar as suas lógicas de funcionamento e, assim, contribuir para os requisitos.

2.2.1 Plataformas

Primeiramente, pesquisou-se a existência de uma plataforma que tinha como tema a UE [9], que é uma secção do website oficial da UE [10], intitulada de Espaço de Aprendizagem, figura 2.1. Aqui existem uma série de jogos, concursos e livros de atividades, que permitem descobrir mais sobre a EU.



Figura 2.1: Espaço de Aprendizagem da plataforma [9]

Apesar do tema e do objetivo final desta plataforma oficial da UE ser semelhante ao que se pretende realizar, existem aspetos que se pretendem melhor. Os aspetos negativos observados, no contexto do que se pretende desenvolver, foram:

- A interface da plataforma não é muito intuitiva para crianças entre os 6 e os 9 anos, dado o excesso de texto e a falta de elementos como, por exemplo, botões.
- A combinação de jogos digitais com físicos exige a que estes últimos levem à necessidade de descarregar e imprimir uma imagem, não sendo jogável na plataforma. Isto leva a que o jogo não esteja disponível de imediato para se jogar.

Em termos dos aspetos positivos, constatou-se que os jogos existentes estão bem conseguidos em termos visuais, com boas interfaces de utilizador e jogabilidade, em particular os elementos que os constituem, como personagens, botões, imagens, entre outros.

Apesar da subjetividade desta apreciação, a expectativa para a plataforma a desenvolver foi a de procurar melhorar os aspetos mais negativos e acompanhar os positivos. Para isso, foi pesquisada uma plataforma que não contemplasse os aspetos negativos mencionados e que tivesse como audiência a idade pretendida. Estudou-se a National Geographic Kids, figura 2.2, que é uma revista infantil online pertencente à National Geographic Society. Esta sociedade lançou uma aplicação web oficial, que contém jogos, vídeos e atividades didáticas, como leituras de factos, centradas em animais e na natureza. O público-alvo situa-se entre os 6 e os 13 anos de idade [11].

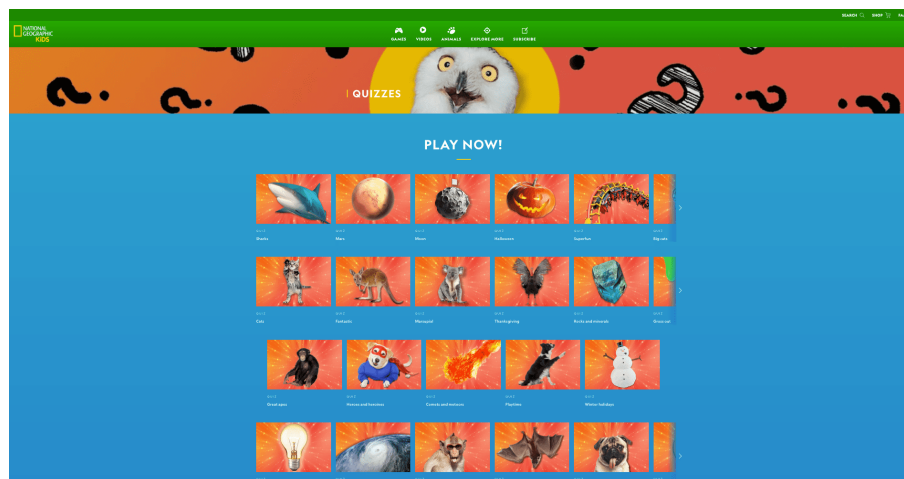


Figura 2.2: Plataforma da National Geographic [12]

Nesta figura 2.2, observa-se a presença de elementos intuitivos como botões generalistas (comando para jogos, play para vídeos, etc). Imagens apelativas para crianças, animadas, cantos arredondados e ilustradas. O texto é substituído maioritariamente por imagens e títulos mais chamativos. Tem também uma boa organização do conteúdo, que não combina diferentes temas, onde cada grelha corresponde apenas a um tema. Será dentro destas características mencionadas que se pretende o resultado final da plataforma a desenvolver.

Página de administração

Apesar de não ser possível aceder às páginas de administração das plataformas exploradas (figuras 2.1 e 2.2), supõe-se que exista uma página de administração dedicada a gerir os respetivos conteúdos, como atualizar jogos, imagens, texto, entre outros elementos da plataforma. Esta página é uma necessidade a que se pretende responder, pois permite que seja atualizada por um “não-programador”. A principal funcionalidade desta página é realizar o Create Read Update Delete (CRUD) ao conteúdo que o programador irá colocar editável.

2.2.2 Análise de conteúdos de plataforma

Nesta subsecção são analisados os diferentes tipos conteúdos que vão estar presentes na plataforma a desenvolver. Cada tipo terá a sua própria página.

Os jogos que vão estar na página de jogos são os seguintes:

- Puzzle
- Jogo de Colorir
- Jogo de Sopa de Letras
- Quiz
- Jogo de Memória
- Mapas Interativos
- Palavras Cruzadas

De seguida, será analisado um jogo para cada tipo, de modo a servir de base e expectativa final, explicando o motivo do jogo estar presente na plataforma. Será também analisada uma página de vídeos, um livro virtual e uma votação online, seguindo a mesma lógica de ideias de criar uma expectativa para o trabalho final e explicar o seu contexto.

Jogos

Os jogos que vão estar presentes na página de jogos serão do tipo quebra-cabeças, também conhecidos como jogos de puzzle [1]. Este foi um dos primeiros requisitos e expectativa para este trabalho a desenvolver, de acordo com os orientadores nas reuniões iniciais.

Este tipo de jogos leva os jogadores a focarem-se em desafios lógicos e conceituais para resolverem um determinado quebra-cabeças. Isto envolve um padrão de reconhecimento, de lógica, de entendimento do processo, entre outros desafios. Normalmente consistem num conjunto de regras onde o jogador manipula as “peças” no tabuleiro de modo a resolvê-lo com sucesso. Os objetivos mais comuns destes jogos são:

- Fazer correspondência entre objetos semelhantes.
- Visualizar e manipular objetos num tabuleiro.
- Memorizar padrões complexos.
- Mesmo jogo com vários níveis de dificuldade.
- Resolver um jogo num determinado espaço de tempo ou tentativas.
- Aplicar conhecimento do mundo real.

Manipulando um ou mais destes objetivos mencionados num determinado jogo é possível dificultá-lo, fazendo com que este se enquadre para uma idade mais velha. O jogo é o mesmo, apenas são alterados os seus parâmetros.

Seguem-se então os jogos exemplos que a plataforma vai conter, cada um explicando o seu objetivo, as necessidades iniciais de implementação que demonstra e qual a expectativa do que se pretende desenvolver, isto é, como se enquadra nas necessidades.

Puzzle



Figura 2.3: Exemplo do jogo Puzzle [13]

Análise do jogo da figura 2.3:

- **Objetivo** - Arranjar as peças de modo a formar a imagem. As peças têm um encaixe específico e formato.
- **Necessidades de implementação** - Será necessário um algoritmo que recorte uma imagem com determinados contornos de modo a formar uma peça e que os encaixes coincidam de modo a construir a imagem correta.
O administrador poderá colocar qualquer imagem como puzzle e criar níveis de dificuldades, reduzindo o tamanho das peças e/ou restringir o tempo para completar.
- **Expetativa do jogo a desenvolver** - Expor o jogador a qualquer imagem que tenha como tema a UE, como por exemplo as bandeiras dos países que a compõem.

Jogo de Colorir



Figura 2.4: Exemplo do jogo de Colorir [14]

Análise do jogo da figura 2.4:

- **Objetivo** - A finalidade é colorir as cores corretas da bandeira. Esta é fornecida em branco, apenas com contornos em negro e é dada a possibilidade de consultar a bandeira com as cores originais a qualquer momento, para ajudar.
- **Necessidades de implementação** - Será necessário um algoritmo que faça a coloração dos pixels dentro de limites definidos pelas linhas.
O administrador deverá poder inserir as diferentes bandeiras e as cores que pretende que sejam utilizadas para a coloração.
- **Expetativa do jogo a desenvolver** - A expetativa é realizar algo semelhante que coloque o jogador a colorir as bandeiras da UE ou outras imagens relacionadas.

Jogo de Sopa de Letras

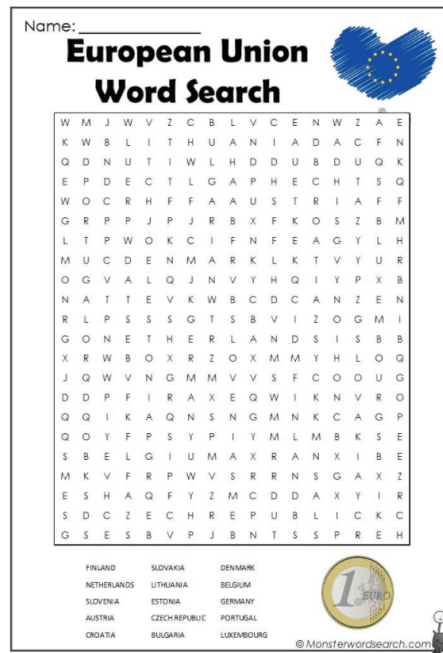


Figura 2.5: Exemplo do jogo de Sopa de Letras [15]

Análise do jogo da figura 2.5:

- **Objetivo** - O propósito é encontrar as palavras presentes na lista fornecida no meio de letras aleatórias.
- **Necessidades de implementação** - Algoritmo responsável por aleatoriamente colocar as palavras no tabuleiro e envolvê-las em letras aleatórias. O administrador deverá escolher o conjunto de palavras para encontrar e definir tamanho do tabuleiro, podendo dificultar colocando tempo para encontrar as palavras.
- **Expetativa do jogo a desenvolver** - Vai permitir ao jogador usar as suas capacidades visuais e lógicas para encontrar palavras relacionadas com a UE.

Jogo Quiz



Figura 2.6: Exemplo do jogo Quiz [16]

Análise do jogo da figura 2.6:

- **Objetivo** - A intenção é colocar ao jogador uma pergunta e uma série de respostas, onde apenas uma resposta é a correta. De seguida carrega na resposta que acha a mais acertada e o jogo dá a informação se esta está correta ou errada, apresentando uma justificação/correção. Várias perguntas podem ser realizadas.
- **Necessidades de implementação** - O administrador controla todos os parâmetros anteriormente mencionados, como quais as perguntas, respostas, tempo de resposta, etc.
- **Expetativa do jogo a desenvolver** - Vai ajudar a melhorar o conhecimento que o jogador tem sobre a UE.

Jogo de Memória



Figura 2.7: Exemplo do jogo de Memória [17]

Análise do jogo da figura 2.7:

- **Objetivo** - A intenção deste jogo é descobrir sempre dois pares iguais até todas as cartas estarem corretamente viradas.
- **Necessidades de implementação** - O administrador possa controlar que imagens formam os pares, o número de pares de cartas para adivinhar e o tempo disponível ou tentativas que tem para completar o jogo.
- **Expetativa do jogo a desenvolver** - Vai expor o jogador a imagens relacionadas com a UE.

Jogo de Mapas Interativos

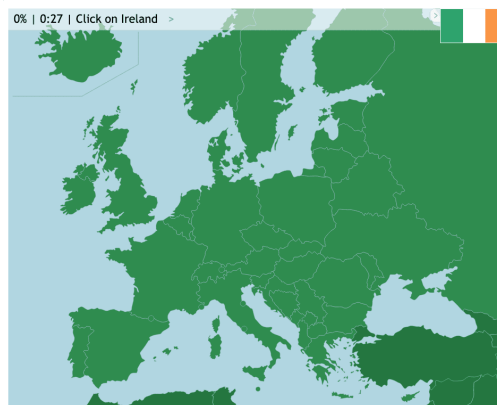


Figura 2.8: Exemplo do jogo de Mapas Interativos [18]

Análise do jogo da figura 2.8:

- **Objetivo** - Responder à pergunta colocada através de um clique no mapa no país correto.
- **Necessidades de implementação** - O administrador cria um conjunto de perguntas e assinala a resposta no mapa. Ao inspecionar o elemento da página, verificou-se neste website que o formato do mapa era SVG (imagem vetorial) [19].
Será para isso necessário integrar um mapa interativo, preferencialmente em formato SVG, e lógica adicional para sincronizar as perguntas, respostas e cliques do utilizador.
- **Expetativa do jogo a desenvolver** - Melhorar os conhecimentos geográficos do jogador em relação à UE.

Jogo de Palavras Cruzadas



Figura 2.9: Exemplo do jogo de Palavras Cruzadas [20]

Análise do jogo da figura 2.9:

- **Objetivo** - Preencher o espaço limitado com letras que vão formar palavra, para isso é dada uma pista ou realizada uma pergunta.
- **Necessidades de implementação** - Será necessário um algoritmo que permita cruzar palavras que contenham letras em comum de modo a formar um tabuleiro com todas as palavras. Associar à palavra perguntas/dicas que também são mostradas ao utilizador. Ter ainda input de texto para dispositivos móveis sem teclado.
O administrador poderá adicionar novas palavras ao tabuleiro.
- **Expetativa do jogo a desenvolver** - Ajudar a melhorar o conhecimento que o jogador tem sobre a UE, adivinhando uma palavra através de uma pista/pergunta.

Vídeos

Um dos requisitos inicialmente discutidos nas reuniões foi a presença de uma página de vídeos. O Centro Jean Monnet iria encarregar-se da sua produção e estes teriam de estar disponíveis na plataforma. Para isso, a melhor solução será criar uma página na plataforma, que permita reproduzi-los de maneira eficaz, não tendo de aceder a outros websites.

O seguinte exemplo é apresentado:

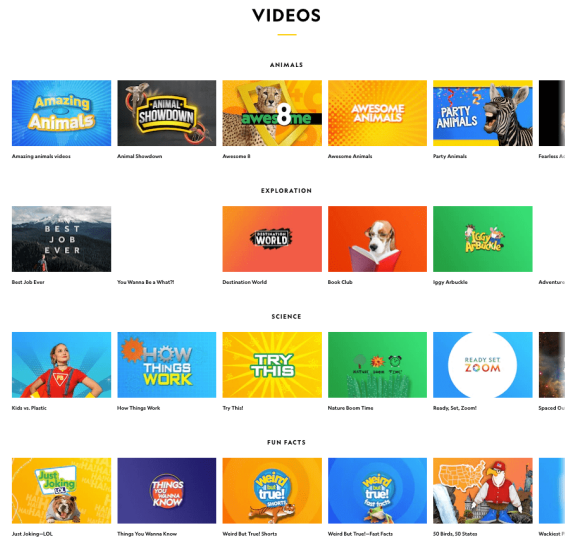


Figura 2.10: Exemplo da disposição dos vídeos que se pretende [21]

Na figura 2.10, que pertence à plataforma da National Geographic Kids (figura 2.2) existe uma secção de vídeos [21]. Esta é uma página independente e observam-se os vídeos organizados por categorias e em lista. Pretendem-se organizar as categorias de vídeos e vídeo de maneira semelhante ao apresentado na plataforma a desenvolver.

Livro Virtual

Um dos requisitos iniciais era a presença de um livro virtual com uma história sobre a UE. Este livro será produzido pela equipa de design [3], associada ao centro Jean Monnet da Universidade de Coimbra (UC) que irá fornecer, no futuro, cada página em formato de imagem. Pode-se observar, de seguida, um exemplo encontrado do que se pretende, na figura 2.11.

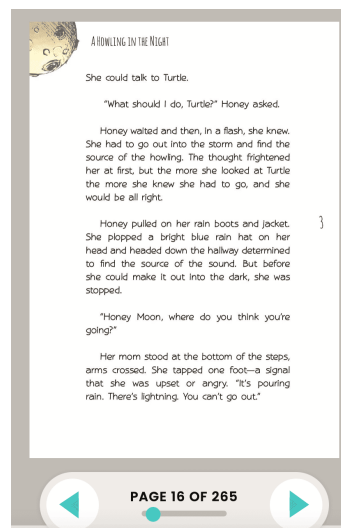


Figura 2.11: Exemplo de um livro infantil virtual [22]

O desafio será integrar um livro virtual na plataforma, recorrendo a uma ferramenta que o permita fazer. Será criada também uma página independente para o livro de modo a organizar o conteúdo.

Votação

Uma das funcionalidades da plataforma requerida pelos orientadores seria ter um espaço onde pudessem ser alojados trabalhos manuais realizados por crianças, como por exemplo pinturas e desenhos, por exemplo. Era importante os visitantes da aplicação, nomeadamente os adultos, poderem ver os diferentes trabalhos presentes nas várias categorias e, através do fornecimento de alguns dados pessoais, como email e nome, poderem votar neles. Com isto, criar-se-ia uma espécie de concurso onde o administrador poderia fechar uma votação e encontrar um vencedor. Posto isto, será criada uma página com esta finalidade, com o conteúdo disposto semelhante à página apresentada em 2.12.

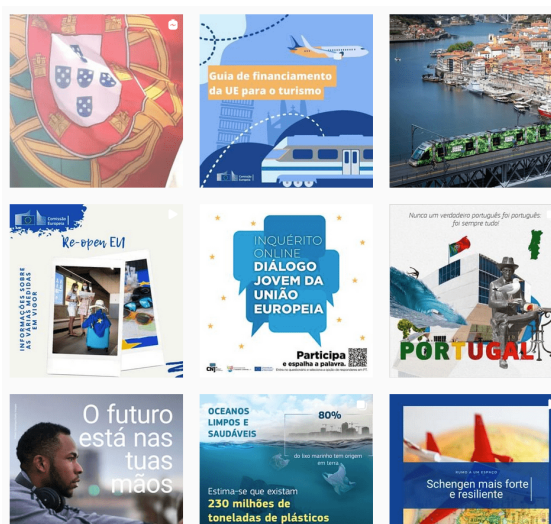


Figura 2.12: Exemplo da disposição do conteúdo que se pretende [23]

A ideia será semelhante à encontrada na figura 2.12, onde as categorias serão dispostas em forma de grelha e clicáveis. Ao clicar numa categoria, aparecem os trabalhos, novamente no mesmo formato, mas já com informações sobre estes. Após clicar num trabalho, é aberto um formulário para se realizar a votação.

2.2.3 Conclusão do analisado

O objetivo de cada jogo será igual ao respetivo jogo analisado, porém devem ser realizados esforços no desenvolvimento para que este tenha parâmetros editáveis para servir em várias idades e dificuldades. Conceitos como a inclusão de detalhes sedutores deve ser cautelosa e pode-se observar que nos jogos escolhidos para analisar estes não se sobressaiam ao objetivo e à mensagem que se pretendia transmitir. Também, em alguns jogos é possível observar um sistema de feedback que guia o jogador caso faça uma ação errada/inesperada. Por exemplo, no jogo de Mapas Interativos, se o jogador responder mal a uma questão, este indica o país correto. Ainda, uma funcionalidade em falta que se pretende que exista nos jogos a desenvolver é assegurar a responsividade nos diferentes ecrãs.

As plataformas desempenham um papel fundamental, pois permitem reunir e expor conteúdos de um tema num único lugar. É necessário que a plataforma a desenvolver contemple elementos que cativem a atenção do grupo-alvo, tais como botões universais, imagens arredondadas e infantis, entre outros mencionados. É essencial que esta reúna também os diferentes conteúdos de maneira clara e organizada, para facilitar a sua utilização. E, ainda, que permita que os seus conteúdos sejam editados por um administrador, deixando a mensagem que se quer passar a cargo deste e não do programador que desenvolveu o sistema, numa lógica de sustentabilidade da plataforma.

Capítulo 3

Metodologia, Planeamento e Gestão de Riscos

Neste capítulo é descrito o processo de desenvolvimento adotado. De seguida, vai estar presente o planeamento do primeiro e segundo semestre, contendo uma breve explicação das decisões tomadas. Ainda, uma secção dedicada à gestão de riscos, onde são expostos os riscos a que este projeto está sujeito e formas de os mitigar, caso aconteçam.

3.1 Metodologia

O processo de desenvolver software deve seguir umas das diferentes metodologias existentes de modo a dividir o trabalho em partes menores, facilitar o processo de adicionar alterações ou funcionalidades quando necessárias e ajudar na gestão do projeto [24].

Uma das metodologias mais adotada é Agile [25], que consiste na integração contínua de desenvolvimento e testes ao longo do projeto. Isto permite que sejam introduzidas novas alterações em conformidade com o feedback do cliente nas várias fases do projeto.

Agile é dividido em vários métodos e o escolhido foi SCRUM [26], que consiste dividir o trabalho em sprints, e cada sprint ter como função entregar partes do produto final. As reuniões com os orientadores tiveram lugar de duas em duas semanas, pelo cada um destes intervalos constituía uma sprint. As reuniões permitiram obter feedback ao que desenvolvido na sprint anterior e apontar que alterações deveriam ser realizadas para estarem presentes na próxima sprint. Não houve a necessidade de daily meetings, que é uma característica deste método, porque estava a ser desenvolvido por uma pessoa, porém foram tidas em contas outras, como:

- **Product Backlog** – são os requisitos funcionais que foram acordados com o *Product Owner* no início do projeto.
- **Product Owner** – foi a orientadora Prof.^a Doutora Teresa Almeida Cravo, que era a representante mais próxima das necessidades deste projeto no contexto do Centro de Excelência Jean Monnet da UC.
- **Sprint Backlog** – email enviado aos orientadores de duas em duas semanas, marcando a reunião de videoconferência que continha os seguintes 3 tópicos:
 - o que foi realizado desde a última reunião (**sprint review** [27]).
 - o que espera ter feito até à próxima (**sprint planning** [28]).
 - o que quer ver discutido nesta (esclarecimento de dúvidas existentes)

3.2 Planeamento

O planeamento vai permitir fazer uma estimativa da duração temporal de cada tarefa, ao longo do ano letivo. Foram realizados dois tipos de planeamentos, a expectativa, executada no início do semestre, e a realidade, produzida próxima do fim do semestre.

3.2.1 Primeiro Semestre

Tarefas principais realizadas no primeiro semestre, com duração aproximada de 15 semanas:

Tarefa	Descrição	Tempo Estimado (semanas)	Tempo Real (semanas)
1	Estado da arte	3	3
2	Especificação de requisitos	6	4.5
3	Arquitetura e tecnologias a utilizar	4	3.5
4	Escrita do relatório intermédio	2	4

Tabela 3.1: Planeamento Primeiro Semestre

1. **Estado da arte** - Esta tarefa consistiu no estudo de conceitos relevantes ao desenvolvimento de jogos, assim como na análise de plataformas e jogos existentes que iriam servir de inspiração e ponto de partida para a aplicação que se pretendia desenvolver.
2. **Especificação de requisitos** - As primeiras reuniões com os orientadores serviram para avaliar quais seriam as principais necessidades em termos de funcionalidades do software que se iria desenvolver no segundo semestre. Juntamente com o estudo já efetuado de software existente, documentaram-se e validaram-se os requisitos do sistema. Seguiu-se o desenho de protótipos de ecrã de baixa fidelidade e também a sua validação.
3. **Arquitetura e Tecnologias a utilizar** - Seguiu-se a tarefa do desenho da arquitetura do sistema, respeitando os requisitos anteriormente definidos. Devido a apresentarem dependências, a arquitetura com as tecnologias, foram documentadas em simultâneo.
4. **Escrita do relatório intermédio** - Consistiu na escrita de um documento onde estava reunido todo o trabalho realizado durante o primeiro semestre. Foram aproveitadas partes de documentos que eram apresentados nas reuniões, como a especificação de requisitos, tecnologias, etc.

3.2.2 Segundo Semestre

O segundo semestre foi ocupado maioritariamente pelo desenvolvimento da aplicação, tendo durado aproximadamente 23 semanas. Foi dividido nas seguintes tarefas:

Tarefa	Descrição	Tempo Estimado (semanas)	Tempo Real (semanas)
1	Correções defesa intermédia	1	1
2	Estudo e configuração	1	3
3	Desenvolvimento dos jogos	10	8
4	Desenvolvimento do website	3	6
5	Testes	1	1
6	Escrita do relatório final	4	4
7	Escrita do artigo científico	3	(Iniciado)

Tabela 3.2: Planeamento Segundo Semestre

1. **Correções defesa intermédia** - Neste período de tempo foram melhorados os defeitos apontados pelos júris no relatório entregue na primeira defesa. Também, se definiram tarefas para serem corrigidas mais tarde quando o avanço do projeto o permitisse.
2. **Estudo e configuração** - Esta tarefa consistiu, primeiramente no estudo de tecnologias que permitiam criar um ambiente de desenvolvimento. Em seguida, foram aplicados os conhecimentos para criar o sistema (front-end e back-end) e foi testado também o deploy numa máquina virtual, de modo a disponibilizar desde o início a aplicação publicamente para ir sendo testada.
3. **Desenvolvimento dos jogos** - Esta tarefa consistiu no desenvolvimento de jogos e a divisão de tarefas, ao longo das 8 semanas, foi a seguinte:
 - Jogo de colorir e puzzle - 2.5 semanas
 - Jogo Quiz e Sopa de letras - 3 semanas
 - Jogo de Memória - 1 semana
 - Jogo Mapas Interativos e Palavras Cruzadas - 1.5 semanas
4. **Desenvolvimento da plataforma** - Esta tarefa consistiu no desenvolvimento da plataforma, do website que ia alojar os jogos desenvolvidos e o restante conteúdo. A divisão de tarefas, ao longo das 6 semanas, foi a seguinte:
 - Back-end - 1 semana
 - Página de jogos, vídeos e livro - 1 semanas
 - Página de administração - 3.5 semanas
 - Página de votação - 0.5 semana
5. **Testes** - Neste período de tempo foram realizados testes funcionais à plataforma e testes de usabilidade em utilizadores reais, adultos e crianças, com o devido consentimento legal.
6. **Escrita do relatório final** - Escrita da dissertação e revisões finais.
7. **Escrita do artigo científico** - A escrita do artigo foi protelada até ao software ganhar a forma final e a dissertação que o explica estar completa. O ponto de partida para a escrita do artigo, com base neste documento, está assim, definido e necessita agora das sugestões dos orientadores para poder ser desenvolvido nas próximas semanas.

Ao longo do trabalho, foram alteradas prioridades de requisitos, em conformidade com as indicações do Centro de Excelência Jean Monnet, devido a determinadas estratégias de mitigação de riscos. Juntamente com a seleção de bibliotecas para ajudar no desenvolvimento, descritas na subsecção 5.1.1, foi necessário dedicar tempo ao estudo destas para poder aplicá-las no projeto, como é o caso dos formulários nas páginas de administração. Também, o estudo de CSS, componentes já predefinidas do Material UI [29] e outras, levaram a que o desenvolvimento do website demorasse mais tempo que o esperado.

3.3 Gestão de Riscos

Nesta secção, são identificados, avaliados e é definida uma estratégia preventiva para os riscos que poderão ter uma repercussão negativa no sucesso do projeto. Define-se por risco qualquer ação que possa ter impacto na entrega de uma tarefa ou num objetivo do projeto, atrasando-os ou impedindo a sua concretização.

Cada risco foi contextualizado temporalmente, ao longo do semestre, permitindo verificar se o risco eventualmente se tornou realidade.

3.3.1 Análise de Risco

Seguem-se os riscos identificados para o projeto que se vai realizar. Cada risco é identificado por (template disponibilizado em [30]):

- **Descrição** - Descreve resumidamente o problema.
- **Consequência** – Consequências se o risco se tornar realidade.
- **Probabilidade** – Probabilidade do risco acontecer ao longo do projeto. Avaliado numa escala de alta, média ou baixa.
- **Impacto** - Impacto que o risco tem no projeto se acontecer. Avaliado numa escala de baixo, médio, alto ou crítico.
- **Estratégia preventiva** – Plano para minimizar as consequências do risco. Existem 4 planos:
 - *Evitar* - mudar o plano para não enfrentar este risco.
 - *Transferir* - transferir o risco para outra entidade (outsourcing).
 - *Mitigar* - rever o plano para reduzir o impacto do risco.
 - *Aceitar* - assumir o impacto negativo que vai ter no projeto.

ID	R1
Descrição	Ausência de feedback por parte da equipa de design, no tempo previsto, sobre a estética das mockups realizadas.
Consequência	O resultado final em termos de design e experiência de utilizador será prejudicado, pois não é fornecida a ajuda profissional nesse aspeto.
Probabilidade	Alta
Impacto	Alto
Estratégia preventiva	Mitigar - O projeto avança com os protótipos de baixa fidelidade (mockups) realizados.

Tabela 3.3: Definição do Risco - R1

O risco R1 tornou-se realidade logo no início do segundo semestre, pois começou o processo de desenvolvimento e foi necessário começar a implementar os jogos e a plataforma. A sensibilidade

estética e os conhecimentos profissionais de design eram fundamentais para fazer evoluir os protótipos de ecrã de baixa fidelidade para algo mais consistente e alinhado com os objetivos de imagem do Centro de Excelência Jean Monnet. Na ausência desse apoio atempado, a probabilidade de ocorrência foi logo definida como alta, tendo sido, de imediato, ativada a estratégia preventiva de mitigação das consequências estéticas no produto final.

ID	R2
Descrição	Ausência de elementos gráficos – como as imagens do livro virtual, personagens, botões, fundos de imagem, tipo de letra, etc. – por parte da equipa gráfica.
Consequência	<ul style="list-style-type: none"> - Não é possível cumprir o requisito de ter um livro virtual produzido pela equipa de design na plataforma. - Funcionalidade de narrar a página é prejudicada, pois não existem as páginas definitivas. - Jogos e plataforma prejudicadas esteticamente.
Probabilidade	Média
Impacto	Alto
Estratégia preventiva	<p>Mitigar</p> <ul style="list-style-type: none"> - O programador encontra elementos contextualizados com a União Europeia em domínio livre na internet e utiliza-os para implementar as funcionalidades - Como o jogo de Tabuleiro e a Realidade Aumentada dependem das personagens e de um enquadramento com o livro virtual estes dois requisitos passam para prioridade <i>Won't have</i> e são substituídos pelos requisitos da página de administração.

Tabela 3.4: Definição do Risco - R2

Definiu-se a probabilidade de ocorrência do risco R2 como média, pois o fornecimento das imagens do livro virtual estava nos objetivos da proposta de dissertação. Devido a atrasos no fornecimento da história, por parte do Centro, e das imagens por parte da Equipa de Design Gráfico, a probabilidade deste risco foi aumentando ao longo do semestre. Este risco tornou-se realidade quando, no calendário previsto, estava definida uma tarefa que não podia ser realizada devido à falta de material. Nesta situação, foi ativada a estratégia preventiva de mitigação, de implementar as funcionalidades com conteúdo não definitivo, que possam posteriormente ser substituídas. Houve igualmente mudanças nas prioridades de requisitos (como na página de administração), de modo a dar continuidade ao trabalho.

ID	R3
Descrição	Conteúdo didático não é fornecido a tempo dos testes de usabilidade
Consequência	Os testes de usabilidade são prejudicados e poderão não ser realizados
Probabilidade	Baixa
Impacto	Alto
Estratégia preventiva	<p>Mitigar</p> <ul style="list-style-type: none"> - O programador substitui o conteúdo didático por <i>dummy data</i> relevante à União Europeia e realiza os testes de usabilidade com estes dados.

Tabela 3.5: Definição do Risco - R3

Definiu-se a probabilidade de ocorrência do risco R3 como baixa, pois o fornecimento do conteúdo didático era dependente do *Product Owner*, definido em 3.1, e era mantido um contato constante com esta entidade. Contudo, foi sempre um risco que se teve em consideração.

A probabilidade deste risco R3 voltou a ser analisada quando o software foi terminado e testado funcionalmente. Nesta fase, havia a necessidade de inserir conteúdo definitivo na plataforma para poderem ser realizados testes de usabilidade com crianças. Estes iriam permitir extrair conclusões relativo ao software desenvolvido. O material foi fornecido na data em que era necessário e o risco não se materializou.

ID	R4
Descrição	Não há resposta atempadamente para mudar o software do ambiente do DEI para o definitivo.
Consequência	Não é possível testar o requisito não funcional da escalabilidade.
Probabilidade	Média
Impacto	Baixo
Estratégia preventiva	Aceitar Aceita-se que não se vão testar o(s) requisito(s) em específico

Tabela 3.6: Definição do Risco - R4

A probabilidade do risco R4 voltou a ser avaliada quando o desenvolvimento do software se deu por terminado. Nesta data a entidade encarregue do IT do Centro de Excelência Jean Monnet da UC, ainda não tinha um prognóstico de quando poderia ser realizada a integração, pelo que se aceitou as consequências de deixar o software dentro do DEI e não realizar testes a determinados requisitos não funcionais. Contudo, foi realizado um tutorial para futuramente poder ser instalado noutras máquinas.

3.3.2 Matriz de Risco

A matriz de risco é um mecanismo que permite visualizar os riscos de maneira mais eficaz e rápida, permitindo ajudar na tomada de decisões. Foi utilizado o seguinte template [31].

		Impacto			
		Baixo	Médio	Alto	Crítico
Probabilidade	Baixa			R3	
	Média	R4		R2	
	Alta			R1	

Figura 3.1: Matriz de gestão dos riscos acima descritos

A figura 3.1 permite visualizar rapidamente quais os riscos que poderão afetar com maior impacto o sucesso do projeto se se tornarem realidade. As probabilidades definidas são as iniciais, como mencionadas nas tabelas dos riscos acima.

Pode-se observar na figura 3.1 que os riscos R2 e R1 são os que merecem mais atenção pois estão com uma cor próxima de vermelho. Segue-se o risco R3, que a qualquer momento pode transitar para a cor laranja, caso a sua probabilidade de ocorrência aumente. O risco R4, apesar de estar numa zona verde, sendo um risco, é necessário prestar-lhe atenção.

Assim sendo, os riscos R3, R2 e R1 foram sempre observados e foram definidas as respetivas estratégias de mitigação, para que o seu impacto no sucesso do projeto fosse minimizado e, nalguns casos, as prioridades fossem redefinidas ou substituídas, em conformidade com as preferências do Centro.

Capítulo 4

Requisitos

O presente capítulo serve para descrever as funcionalidades e características do sistema a desenvolver. O processo adotado para a recolha dos requisitos consistiu em quatro atividades principais [32]:

- Recolha de Requisitos - fase inicial de contacto com o projeto, onde foi analisado software já existente, como plataformas lúdico-didáticas infantis, realizadas reuniões com os orientadores e requisitos descritos em alto nível, como personas e user stories. Nestas etapas, nas reuniões era sempre dado o parecer do trabalho previamente realizado.
- Especificação de Requisitos - fase formal onde foi realizada a documentação dos requisitos, que se pode observar neste capítulo. Aqui, constam os requisitos funcionais e não funcionais, casos de uso e protótipos de ecrã.
- Verificação e validação de Requisitos (V&V) [33] - a verificação que assegura que o produto a desenvolver vai ao encontro dos requisitos definidos e a validação que certifica que o produto a desenvolver garante as necessidades do cliente. Mais uma vez, foram utilizadas as reuniões regulares para assegurar que tudo estava a ser especificado corretamente.
- Gestão de Requisitos - como o desenvolvimento do software foi realizado de maneira incremental, a cada duas semanas foram calendarizadas as reuniões com os orientadores e, em várias delas, foram realizadas demonstrações do que havia sido desenvolvido até ao momento, permitindo acompanhar regularmente o processo de desenvolvimento, o que foi também implicando modificações nos requisitos inicialmente propostos.

4.1 Personas

Atores e Cenários

Os atores que vão interagir com a plataforma classificam-se em dois tipos: visitante e administrador.

O visitante idealmente será uma criança entre os 6 e os 9 anos, porém qualquer pessoa que aceda à plataforma será considerada um visitante. As principais ações dos visitantes são: jogar os jogos disponíveis, ver os vídeos, ler o livro virtual infantil e ver/votar nos trabalhos disponíveis.

O administrador será um adulto que contenha as credenciais de acesso para aceder à página de administração. As principais ações deste ator são: gerir o conteúdo da plataforma e ver estatísticas disponíveis relevantes de jogos. Vai poder alterar as definições de todos os jogos existentes e, com isto, pode criar um novo, visualmente diferente, mas com o mesmo objetivo. Vai poder também gerir os vídeos e votações.

Definição das personas

As personas consistem numa personagem fictícia que representa os utilizadores ideais da plataforma. Esta descrição é útil numa fase inicial para considerar os objetivos e desejos dos utilizadores ajudando futuramente na tomada de decisões sobre as características do sistema, como na definição de requisitos ou desenho de protótipos. Pensou-se em 3 tipos de personas:


Persona	Visitante - Criança entre os 6 e 9 anos
Fotografia	
Nome fictício	Afonso Fonseca
Cargo	Estudante, frequenta o 1º clico do ensino básico
Dados demográficos	7 anos Naturalidade portuguesa Mora em Coimbra
Enquadramento tecnológico	Utiliza o iPad dos pais, quando permitem, para ler livros didáticos e jogos virtuais. É utilizador frequente do tablet.
Citação	“Mãe, com este jogo já aprendi muitas coisas sobre a União Europeia!”
Objetivo(s)	Aprender sobre a União Europeia, conhecendo quais as atividades do Centro de Excelência Jean Monnet da UC. Divertir-se e aprender ao mesmo tempo.
Cenário	Quando os pais lhe disponibilizarem o tablet com a plataforma aberta, o Afonso explora-a, jogando jogos, vendo os vídeos e lendo o livro. No final, adquire mais conhecimento sobre a União Europeia.

Tabela 4.1: Persona 1


Persona	Visitante - Adulto
Fotografia	
Nome fictício	Maria Fonseca
Cargo	Advogada
Dados demográficos	37 anos Casada Mãe de Afonso Costa Naturalidade portuguesa Mora em Coimbra Licenciada em Direito
Enquadramento tecnológico	Utiliza diariamente computador, tablet e internet para o seu trabalho e lazer
Citação	“Deixar o meu filho explorar esta nova realidade tecnológica e esta forma de adquirir conhecimentos, com responsabilidade”
Objetivo(s)	Ver o conteúdo da Plataforma Votar em trabalhos Abrir a plataforma no tablet para o seu filho Sugerir jogos para o seu filho jogar
Cenário	Pretende ver se o trabalho realizado pelo seu filho já está disponível na plataforma para poder votar nele. Apresentar a plataforma ao filho pela primeira vez. Ajudar o seu filho com qualquer dúvida nos jogos.

Tabela 4.2: Persona 2


Persona	Administrador - Adulto
Fotografia	
Nome fictício	Rosa Maria
Cargo	Professora associada ao Centro de Excelência Jean Monnet da UC
Dados demográficos	40 anos Casada Naturalidade portuguesa Mora em Coimbra
Enquadramento tecnológico	Encarregue da parte de divulgação das atividades do Centro nos meios de comunicação.
Citação	“Colocar as atividades do Centro de Excelência Jean Monnet da UC numa plataforma lúdico-didática infantil vai permitir transmitir a mensagem também aos mais novos!”
Objetivo(s)	Adicionar ou atualizar o conteúdo na plataforma. Monitorizar as estatísticas.
Cenário	Pretende adicionar novas perguntas ao jogo Quiz. Pretende ver estatísticas de um jogo de nível difícil e ver se está adequado ao público-alvo. Vai ver a métrica de percentagem de vitórias e se estiver baixa vai, de seguida, facilitar um parâmetro do jogo, com o objetivo de ficar mais fácil. Mais tarde, volta a observar e vê se a percentagem já subiu. Repete o processo até obter resultados satisfatórios. Pretende criar uma categoria de votação e inserir trabalhos de crianças que foram realizados no âmbito de uma atividade de divulgação do Centro. Pretende adicionar um novo vídeo que foi publicado recentemente pelo Centro.

Tabela 4.3: Persona 3

Com as personas anteriormente descritas é possível saber quais as necessidades fulcrais que o produto a desenvolver terá de conter para ser bem-sucedido. Também, torna possível imaginar a interação do utilizador com o produto, ajudando a criar histórias que permitem descrever as principais funcionalidades do sistema, como se pode observar de seguida na secção 4.2.

4.2 User Stories

São comumente utilizadas numa fase inicial na metodologia Agile, que é a escolhida, e permitem explicar uma funcionalidade do sistema no contexto do utilizador final, ajudando a perceber como esta entrega valor ao utilizador. Sendo um ponto de partida são definidas a alto nível, podendo depois serem divididas para dar mais detalhe e informações.

A sua principal utilidade neste projeto foi encorajar a pensar criticamente na melhor maneira de transformar uma história num requisito/funcionalidade da aplicação.

A estrutura é a seguinte: Enquanto [ator], posso [ação], para [motivo].

ID	Descrição
US-1	Enquanto visitante posso jogar um jogo disponível na página de jogos para aprender mais sobre a União Europeia.
US-2	Enquanto visitante posso ver um vídeo na página de vídeos para saber mais sobre a União Europeia.
US-3	Enquanto visitante posso ler o livro didático virtual na página do livro para ficar a conhecer uma história sobre a União Europeia.
US-4	Enquanto visitante posso ver todos os trabalhos disponíveis, divididos em categorias, para no final votar no(s) que mais gostei.
US-5	Enquanto visitante posso ver na plataforma informações relevantes sobre quem financia o projeto para saber a quem pertence.
US-6	Enquanto administrador posso realizar login e logout para aceder e sair, respetivamente, da página de administração da plataforma.
US-7	Enquanto administrador posso alterar as credenciais de acesso caso seja necessário.
US-8	Enquanto administrador posso realizar o CRUD aos jogos para poder atualizar a página de jogos.
US-9	Enquanto administrador posso realizar o CRUD de categorias de vídeos e dos vídeos que as vão compor para poder atualizar a página de vídeos.
US-10	Enquanto administrador posso realizar o CRUD de categorias de trabalhos e dos trabalhos que as vão compor para poder atualizar a página de votações.
US-11	Enquanto administrador posso encerrar uma votação para encontrar um vencedor.
US-12	Enquanto administrador posso ver estatísticas recolhidas anonimamente dos jogos para tirar conclusões à cerca destes.

Tabela 4.4: User Stories

Estas histórias de utilizador foram as principais necessidades tidas em conta para especificar com mais detalhe os requisitos funcionais apresentados de seguida.

4.3 Requisitos Funcionais

Os requisitos funcionais descrevem as funções que um determinado software deve desempenhar e que vão ser implementadas. Na tabela 4.5 vão ser apresentados os requisitos funcionais considerados para o sistema. Para cada requisito funcional é dado um identificador único, uma breve descrição e sua respetiva prioridade.

Vão ser priorizados segundo o método MoSCoW que é uma técnica utilizada em desenvolvimento de software para chegar a um consenso com o cliente sobre a importância dada à entrega de cada requisito. Apresenta a seguinte nomenclatura:

- **"M- Must have** - são fundamentais estarem implementados para o sucesso do projeto.
- **"S- Should have** - igualmente importantes, mas não é necessário estarem implementados na totalidade para o sucesso do projeto.
- **"C- Could have** - são desejáveis no projeto, mas não necessários, podem melhorar a experiência do utilizador ou satisfação do cliente, contudo só serão implementados se o tempo o permitir.
- **"W- Won't have** - são os requisitos considerados pelo cliente que não são apropriados para o momento e por isso só serão implementados futuramente.

ID	Descrição	Prioridade
FR-1	A plataforma deverá conter uma barra de funcionalidades com botões que redireciona para os Jogos, Vídeos, Livro e Votações	Must have
FR-2	A plataforma deverá conter uma secção no footer com informações relevantes sobre a instituição que financia o projeto	Must have
FR-3	A plataforma deverá conter uma página de jogos	Must have
FR-3.1	Os jogos, quando completados com sucesso, devem parabenizar o jogador. Quando completados com insucesso, devem perguntar se quer jogar novamente	Must have
FR-3.2	A página de jogos deverá conter jogos de Colorir	Must have
FR-3.3	A página de jogos deverá conter Puzzle's	Must have
FR-3.4	A página de jogos deverá conter Quiz's	Must have
FR-3.5	A página de jogos deverá conter jogos de Sopa de Letras	Must have
FR-3.6	A página de jogos deverá conter jogos de Palavras Cruzadas	Should have
FR-3.7	A página de jogos deverá conter jogos de Memória	Must have
FR-3.8	A página de jogos deverá conter jogos de Mapas Interativos	Should have
FR-3.9	A página de jogos deverá um jogo de Tabuleiro	Won't have
FR-3.10	A plataforma deverá conter Realidade Aumentada para as perguntas do Jogo de Tabuleiro	Won't have
FR-3.11	A página de jogos deverá ter jogos de Palavras Antónimas	Should have
FR-4	A plataforma deverá conter uma página de vídeos	Should have
FR-4.1	A página de vídeos deverá integrar vídeos do Youtube do canal Centro de Excelência Jean Monnet UC, bem como outros canais	Should have
FR-4.2	Os vídeos devem ser divididos em categorias	Should have
FR-5	A plataforma deverá conter uma página com o livro infantil	Must have
FR-5.1	A página deverá integrar as imagens do livro infantil fornecidas pela equipa de design [3]	Must have
FR-5.2	O utilizador deverá virar as páginas do livro realisticamente	Must have
FR-5.3	Cada página deverá ter um player de áudio para poder ser narrada	Should have
FR-5.4	Deverá informar o utilizador da página em que se encontra	Should have
FR-6	A plataforma deverá conter uma página de votação	Should have
FR-6.1	A página deverá listar as diferentes categorias de votações	Should have
FR-6.2	Cada trabalho deverá conter o seu título, a fotografia, uma breve descrição e o seu autor. Também, o número de votos atuais e o botão para votar	Should have
FR-6.3	Para votar no trabalho, o utilizador deverá inserir o seu nome e email num formulário	Should have
FR-7	A plataforma deverá ter uma página de autenticação do administrador	Must have
FR-8	A página de administrador deverá ter uma dashboard com as seguintes ações: - alterar as credenciais de acesso da conta de administrador - realizar logout - gestão do conteúdo da página - estatísticas dos jogos	Must have

ID	Descrição	Prioridade
FR-8.1	A secção de gestão de conteúdo da página deverá permitir as seguintes ações: - CRUD todos os jogos existentes - CRUD categorias de vídeo e vídeos - CRUD categorias de votação e trabalhos	Must have
FR-8.2	A secção de estatísticas deverá fornecer informações como: - jogos abertos - jogos finalizados - vitórias - respostas certas - percentagens relevantes - tempo de resposta	Must have

Tabela 4.5: Requisitos Funcionais

Como a metodologia do projeto é Agile, definida na secção 3.1, foram acrescentados novos requisitos e outros alterados ao longo do desenvolvimento do projeto.

Devido à ativação da estratégia preventiva de mitigação do risco R3, tabela 3.5, os requisitos FR-3.9 e FR-3.10 mudaram a sua prioridade de *Must have* e *Could have*, respetivamente, para *Won't have*. Todavia, para compensar, os requisitos FR-7, FR-8, FR-8.1 e FR-8.2 com prioridades *Should have* passaram para *Must have*.

Para esclarecer certos requisitos que não tenham ficado claros e ajudar a refletir na maneira mais eficiente de os implementar foram realizados casos de uso. Estes são apresentados abaixo.

4.3.1 Casos de Uso

Um caso de uso é uma descrição de como um ator utiliza uma funcionalidade do sistema para atingir um objetivo, permitindo detetar mais facilmente as funcionalidades que são necessárias implementar.

ID	UC-2 - Ver um vídeo
Caso de Uso	Este caso de uso começa quando o utilizador quer ver um vídeo e termina quando o utilizador acaba de vê-lo
Ator Primário	Visitante
Precondições	O utilizador tem a aplicação web aberta
Triggers	O utilizador carrega na opção “Vídeos” da barra presente no canto superior
Flow Básico	<ol style="list-style-type: none"> 1. O utilizador é redirecionado para a página de vídeos 2. O sistema apresenta os vários vídeos, em formato de lista, separados por categorias 3. O utilizador carrega num vídeo 4. O sistema reproduz o vídeo automaticamente 5. O utilizador vê o vídeo
Pós Condição	Nenhuma

Tabela 4.6: Caso de uso UC-2

Os restantes casos de uso realizados encontram-se no Anexo A.

4.3.2 Protótipos de ecrã

A fidelidade de um design refere-se ao nível de detalhe e funcionalidades que é colocado no desenho do produto.

Para este projeto, foram realizados protótipos de baixa fidelidade que permitiram colocar numa fase inicial as ideias num artefacto testável. Este tipo de prototipagem oferece vantagens como não ter a responsabilidade de colocar todas as funcionalidades no design e rapidamente se conseguir mudar de modo a ir ao encontro do produto desejado – ao contrário da prototipagem de alta fidelidade, no qual o design se assemelha ao produto final. Foram utilizadas as reuniões para ir recebendo feedback regular por parte dos orientadores.

As qualidades apontadas foram:

- Estavam alinhados com os objetivos anteriormente definidos.
- As principais funcionalidades estavam desenhadas corretamente.

Os defeitos:

- Precisariam de mais detalhe e cuidado em termos de design quando as interfaces fossem implementadas.

Foi colocada a possibilidade da equipa de design parceira do projeto [3], ajudando a transformar a fidelidade de baixo nível para alto. Contudo, foi definido o risco da tabela 3.3, antecipando a possibilidade das sugestões gráficas não serem entregues atempadamente.

Os protótipos desenvolvidos podem ser observados no anexo B.

4.3.3 Requisitos Não Funcionais

Os requisitos não funcionais são requisitos que não têm uma funcionalidade específica no sistema, são antes requisitos que ajudam a obter a solução desejada, especificando as qualidades que o sistema deve ter.

De seguida, vão ser apresentados os atributos de qualidade que melhor descrevem como o sistema se deve comportar. Cada um é seguido da sua definição e contexto no projeto.

- **Usabilidade** - Refere-se à qualidade da experiência do utilizador ao interagir com o software [34].

A UI do sistema deve ser facilmente interpretada por uma criança com mais de 6 anos, tendo em conta um design intuitivo, facilidade de aprender, eficiência a utilizar e a satisfação. Também, estas características devem ser tidas em conta na página de administração, mas para um adulto.

- **Portabilidade** - Define-se como a capacidade de transferir o software para ambientes diferentes, criando um programa executável, e este funcionar como o expectável.

Uma das necessidades iniciais apontadas foi o software poder ser instalado com facilidade noutra máquina sem a presença do programador para, por exemplo, ficar disponível para uma comunidade em específico. Como há várias possibilidades para o sistema onde poderá ser instalado, tem de ser escolhidas tecnologias que permitam esta abstração.

- **Compatibilidade** - Refere-se à capacidade de um determinado software poder operar com as mesmas funcionalidades em sistemas diferentes.

A plataforma deverá poder ser utilizada tanto num dispositivo móvel como no computador, com as mesmas funcionalidades, para permitir que seja acedida por um maior número de pessoas. Serão testados e validados os sistemas possíveis posteriormente, mas numa fase inicial foram definidos os mais utilizados:

Nos computadores os navegadores: Chrome, Safari e Firefox.

Nos dispositivos móveis os sistemas operativos e navegadores:

- iOS - Safari
- Android - Chrome

As versões serão depois especificadas no capítulo de testes.

- **Escalabilidade** - Relaciona-se com a capacidade do sistema operar dentro dos limites da normalidade quando o número de ligações aumenta.

Como dito anteriormente, a necessidade do software estar disponível ao público, leva a necessidades de ligações em simultâneo, foi definido um número de 100 ligações, que corresponde aproximadamente a duas turmas de alunos, número por alto exigido numa atividade de divulgação do Centro.

- **Segurança** - Define-se pela capacidade do sistema operar dentro dos limites da normalidade e proteger a sua integridade quando é atacado maliciosamente.

As necessidades encontradas a nível de segurança para a plataforma a desenvolver foram encriptar a password do administrador na base de dados e negar acessos a utilizadores não autenticados, tanto à página de administração como à base de dados.

- **Conformidade** - Todos os dados inseridos na plataforma por utilizadores devem estar em conformidade com as regras e leis General Data Protection Regulation (GDPR).

Como há recolha de dados como emails e nomes nas votações, dados sensíveis, estes devem ser protegidos se forem armazenados, que é o caso. Para isso, vão ser guardados encriptados na base de dados.

Os requisitos não funcionais passíveis de serem testados (usabilidade, compatibilidade, escalabilidade e segurança (autenticação não autorizada)) serão verificados e validados na fase de testes, secção 7.2. Os restantes, não mencionados, (portabilidade, segurança (encriptação) e conformidade) serão na secção das Tecnologias 5.1.5, no capítulo seguinte.

Capítulo 5

Arquitetura e Tecnologias

Este capítulo aborda, primeiramente, todas as tecnologias utilizadas para desenvolver o software, descrevendo-as e justificando a sua seleção. Posteriormente, é exposta a Arquitetura do sistema desenvolvida, através do modelo C4 [35]. Decidiu-se juntar estes dois tópicos pois apresentam dependências, a escolha das tecnologias influenciou a arquitetura e vice-versa, o que levou a serem pesquisadas em simultâneo.

5.1 Tecnologias de Desenvolvimento

Esta secção apresenta as tecnologias selecionadas para o front-end e back-end. Ainda, verifica as suas compatibilidades e os requisitos não funcionais que são validados.

5.1.1 Front-end

O termo Front-end refere-se à camada de apresentação, que tem por base converter dados numa interface gráfica, através de HTML, CSS e JavaScript, de modo aos utilizadores poderem ver e interagir com estes.

Jogos

Os motores e frameworks de jogos são ferramentas disponíveis para os programadores construírem um jogo de maneira fácil e rápida, sem a necessidade de implementarem tudo de raiz. Para isso, oferecem funções predefinidas que, por exemplo, ajudam a reproduzir música ou animações, interfaces gráficas com *drag and drop*, que auxiliam no sentido de não trabalhar com coordenadas na Canvas, elemento do HTML5 que permite a renderização dinâmica de gráficos [36], entre outras.

Foi explorada uma lista com essas ferramentas disponíveis [37] e foram selecionadas as que eram gratuitas, o seu foco de plataforma era a web (HTML5) e orientada a 2D. Os candidatos foram Phaser, GDevelop e Unity.

- Phaser [38] - é uma framework JavaScript gratuita de jogos 2D com foco para a web, HTML5, e dispositivos móveis. O único requisito para estar framework é o navegador suportar a tag HTML `<canvas>`, pelo que é compatível com a maioria. Dentro das suas funcionalidades destacam-se: exibição de imagens, sprite sheets (imagem que consiste em várias imagens mais pequenas, sprites, com o objetivo de criar uma animação) e tweens (animações predefinidas, como ease-in, ease-out); receber inputs e ter funções específicas para lidar com estes eventos; físicas de jogo, como colisões.
- GDevelop [39] - é um motor de jogo 2D gratuito para todas as plataformas, incluindo web, que tem como foco criar jogos para HTML5. Como disponibiliza uma interface para se

desenvolver o jogo, não programadores podem construir um, pois utiliza programação visual baseada em eventos. Estas são as suas principais funcionalidades.

- Unity [40] - é um motor de jogo que permite o desenvolvimento 2D e suporta WebGL, pelo plugin Unity Web Player, que tem compatibilidade com os navegadores mais populares, para mais tarde embutir o jogo em ambiente web. A linguagem utilizada para programas é o C# e disponibiliza uma interface que permite interagir com os objetos. É gratuito se o projeto não passar dos \$100K por ano de receita, ultrapassando este limite é necessário a versão Pro, com custos. Também, algumas funcionalidades só ficam disponíveis com esta versão.

Segue-se uma tabela comparativa, 5.1, com as tecnologias mencionadas anteriormente. Os parâmetros de comparação utilizados foram a linguagem utilizada, licença, prós e contras considerados relevantes para este projeto.

	Phaser [38]	GDevelop [39]	Unity [40]
Descrição	Framework de jogos HTML5 rápida e gratuita	Motor de jogo para não programadores	Motor de jogo popular para todo o tipo de jogos e plataformas
Linguagem	JavaScript	(programação visual baseada em eventos)	C#
Licença	MIT	MIT	Proprietária
Prós	Gratuita Linguagem de programação JavaScript (como todo o projeto) Boa documentação e exemplos Rápida curva de aprendizagem	Gratuita Interface gráfica	Interface gráfica Scripts para complementar jogos Possibilidade de criar jogos mais complexos e profissionais Popular e grande comunidade
Contras	Não ter interface gráfica	Limitada e pouco personalizável	Possibilidade de acarretar custos Curva de aprendizagem acentuada Loja com recursos para se utilizarem em projetos mas são pagos (Unity Asset Store)

Tabela 5.1: Tabela de comparação de motores e frameworks de jogos

Foi escolhida a framework Phaser, visto apresentar prós relevantes para o projeto, nomeadamente a linguagem JavaScript que, para desenvolvimento Web, é a mais utilizada. Também foi analisada a documentação desta ferramenta, *Phaser3-docs* [41], e, *Phaser Labs* [42], que consiste em exemplos de códigos das suas funcionalidades que podem ser testados em tempo real. Estas ferramentas mencionadas permitem acelerar o tempo de aprendizagem da framework. A única desvantagem encontrada foi a inexistência de uma interface gráfica, porém é colmatada com o conhecimento de técnicas de coordenadas e rotações na Canvas.

O Phaser pode também ser utilizada com TypeScript [43], que é uma linguagem construída no topo de Javascript. Adiciona a principal funcionalidade de *static typing* [44], que consiste no processo de verificar, por exemplo, se uma variável recebe o tipo correto de dados que foi especificado inicialmente. Isto é útil quando existem muitas variáveis e funções, que acontece nos desenvolvimentos de jogos, evitando bugs ao passar dados de tipos diferentes.

Interface de Utilizador

As ferramentas consideradas para criar as interfaces do utilizador que vão presentes na plataforma formam as seguintes: React, Vue e Angular [45]. Destas 3, todas são competentes para serem utilizadas no projeto, como se pode ver na tabela 5.2, mas apenas uma será escolhida.

	React [46]	Angular [47]	Vue [48]
Descrição	Biblioteca JavaScript mais popular	Framework completa	Framework JavaScript progressiva
Linguagem	JSX (JavaScript XML)	TypeScript	JavaScript
Licença	MIT	MIT	MIT
Prós	Estilo declarativo Abordagem baseada em componentes virtual DOM Grande comunidade Curva de aprendizagem rápida	Estilo declarativo Abordagem baseada em componentes real DOM Boa documentação	Estilo declarativo Abordagem baseada em componentes virtual DOM Boa documentação Curva de aprendizagem rápida
Contras	Dependência de bibliotecas externas	Curva de aprendizagem elevada	Menos popular

Tabela 5.2: Resumo da comparação entre React, Angular e Vue

Pela análise da tabela 4.2, pode-se verificar que as vantagens do React vão de encontro às necessidades do projeto, pelo que vai ser a tecnologia escolhida. O seu estilo declarativo e abordagem baseada em componentes vai permitir uma melhor leitura, organização e reaproveitamento de código, o que vai permitir construir uma plataforma com foco na longevidade e expansibilidade, à qual podem ser acrescentadas funcionalidades, sobre as já existentes, no futuro. A desvantagem de dependência de bibliotecas externas é colmatada pela grande comunidade, que estão constantemente a atualizar e lançar bibliotecas e, à partida, todas as necessidades que o projeto engloba são satisfeitas por este meio. Esta comunidade ainda permite aumentar a base de dados de tutoriais disponíveis online, o que ajuda na resolução de problemas e diminui curva de aprendizagem.

Ainda, como já existia background com React, esta foi a escolhida de modo a acelerar o processo de desenvolvimento.

Bibliotecas utilizadas no React

Seguem-se as bibliotecas mais relevantes que vão ser instaladas no React. Todas foram obtidas por pesquisa no Node Package Manager (NPM) [49] que é um repositório online open-source onde são publicadas Node.js packages. Estas depois podem ser instaladas, pela linha de comandos, no projeto que se está a desenvolver.

Para cada uma vai ser dada uma breve explicação do seu objetivo e qual a utilidade neste projeto.

- **Ajuda no Design**

- **material-ui: 0.20.2** - Disponibiliza um conjunto de componentes já com um design predefinido, muito semelhante ao que a Google [29] utiliza nas suas aplicações. Podem ser botões, cartas, modals, checkboxes, tabelas, etc. Vai ajudar a melhorar a experiência de utilizador no geral, principalmente da página de administração.
- **styled-components: 5.2.1** - Permite utilizar CSS na componente Javascript do React, retirando a necessidade de criar outros ficheiros externos. Vai tornar mais fácil e rápido o processo de dar estilo à parte visual da plataforma (elementos HTML) que não utilizem Material-UI [29].

- **Livro Virtual**

- **react-flip-page: 1.6.4** – Permite ter as animações que imitam o virar de uma página de um livro. Vai ser útil para o livro virtual didático.
- **react-audio-player: 0.14.0** – Permite reproduzir áudio, é dada a localização do ficheiro na pasta do projeto e este encarrega-se de colocá-lo em formato player. Vai ser útil para o utilizador conseguir ouvir a narração dos diferentes áudios que compõem a narração do livro virtual.

- **Pedidos ao servidor**

- **axios: 0.21.1** - Torna mais fácil enviar pedidos HTTP assíncronos aos REST endpoints da API do servidor.
- **react-query: 3.13.5** – Permite ao React, através do axios, realizar o pedido ao servidor, guardar essa informação numa cache, sincronizar e atualizar os dados na camada de apresentação em função do servidor. Vai ser útil nas operações CRUD, quando, por exemplo, é eliminado um jogo. O pedido é enviado ao servidor para eliminar, este elimina na base de dados e dá informação ao cliente do que foi feito, a biblioteca então encarrega-se de atualizar a cache, indo buscar informação mais atualizada. Ainda permite dar informações sobre o estado do pedido (loading, error, etc), útil para mostrar ao utilizador o estado atual.

- **Formulários**

- **react-hook-form: 7.4.0** - Permite a criação e validação de formulários. No React sempre que o utilizador interage com a UI, para esta apresentar as modificações tem de atualizar a componente. Como esta biblioteca está otimizada para estes casos indesejáveis que prejudicam a performance (atualizar sempre) e ainda fazer a validação dos inputs inseridos antes dos submeter, entre outras features, vai tornar o comportamento da aplicação mais previsível e fácil de manter, permitindo que não chegem dados inválidos ao servidor. Esta biblioteca vai ser utilizada em todos os formulários presentes na plataforma.
- **yup: 0.32.9** - Complementa a biblioteca de formulário anteriormente mencionada, permitindo construir um esquema de dados que é o desejado para o formulário aceitar. Por exemplo, permite dizer que o campo email vai receber uma string no formato email, e, se não for o caso, dar um erro mostrando que a string inserida não é um email. Ainda, é possível personalizar estas validações para por exemplo um input só ser válido se outro estiver preenchido.
É mostrado um exemplo de utilização real desta biblioteca na figura 6.19.

- **Rotas**

- **react-router-dom: 5.2.0** - Permite a criação de rotas (routes), que se traduzem em endereços URL's, e renderizar componentes condicionalmente. Estes endereços podem ser dinâmicos. Útil para construir páginas com diferentes endereços, colocar o ID de um determinado jogo no URL para ir buscar informação relativa a este, entre outros.

- **Jogos**

- **react-full-screen: 1.0.2** – Permite fazer um wrapper no qual todos os filhos desta component vão ficar em full-screen. Vai ser útil para se poderem jogar os jogos em ecrã inteiro e conseguir que os jogos funcionem responsivamente em todos os tamanhos de ecrãs.
- **react-simple-maps: 2.3.0** – Permite disponibilizar uma componente que contém um mapa em formato SVG da europa. Este mapa tem os países definidos com contornos e funções úteis como saber se o utilizador passou com o rato por cima (onHover), se clicou (onClick), etc.
- **@jaredreisinger/react-crossword: 2.3.1** – Esta biblioteca consiste em disponibilizar um tabuleiro onde são inseridas palavras através de coordenadas, permitindo construir um jogo de Palavras Cruzadas. Ainda, para cada palavra permite ter associada, além das coordenadas (número da linha e coluna), uma pista/pergunta, já com toda a experiência de utilizador implementada.

- **Outras**

- **react-scroll: 1.8.2** – Permite que a aplicação dê scroll automático para um elemento do HTML. Vai ser útil para melhorar a experiência de utilizador e redirecioná-lo para a localização desejada
- **react-color: 2.19.3** – Disponibiliza uma componente que permite selecionar uma determinada cor numa paleta de cores, em estilo Photoshop [50]. Vai ser utilizada para o administrador adicionar qualquer tipo de cor ao jogo de colorir.
- **react-player: 2.9.0** - Permite reproduzir vídeos do Youtube através do respetivo URL. (sem ser o link embedded).

5.1.2 Back-end

O back-end refere-se a toda a parte que não é visível pelo utilizador final, permitindo dar ao front-end, por meio de pedidos, conteúdo dinâmico. A escolha das tecnologias anteriores, efetuadas na secção 5.1.1, vão influenciar esta diretamente. Assim, tentou-se encontrar a mais compatível que, pelas pesquisas efetuadas [51], se demonstrou ser o Node combinado com a framework Express.

O Node (Node.js) é um ambiente de execução JavaScript que, aliado ao Express, que é uma framework complementar, permite executar código fora do web browser, no servidor. A execução de scripts permite enviar informações, em formato JSON, ao front-end, de modo a produzir-se conteúdo dinâmico.

A framework Express (Express.js) vai executar no servidor Node. A sua principal funcionalidade passa por acondicionar os pedidos e respostas HTTP. Permite o mapeamento dos endereços URL's em diversas funções no servidor, criando uma API. Isto possibilita que o front-end comunique com o back-end.

Para este projeto, há a necessidade de existir uma base de dados para guardar as credenciais do administrador, associar os endereços dos vídeos e dos jogos para embutir na plataforma, recursos de jogos como perguntas, entre outros.

De modo a tirar proveito das tecnologias anteriormente selecionadas, nomeadamente da stack MERN [52], explorou-se a base de dados não relacional orientada a documentos MongoDB. Esta permite armazenar documentos, objetos em formato JSON, e sua principal vantagem é a flexibilidade do conteúdo que os compõe. Tem foco para aplicações web e móveis, onde, por vezes, ocorrem mudanças com frequência e há a necessidade de esquemas (schemas) flexíveis [53].

Na ligação do Node.js com a MongoDB será utilizada a biblioteca Mongoose, que é uma ferramenta de Object Data Modeling (ODM). Permite funcionar em ambiente assíncrono, não enviando a resposta ao utilizador enquanto as operações na base de dados não estiverem terminadas e possibilita definir os objetos que cada documento pode receber, fornecendo um esquema de validação.

Concluindo, uma base de dados não relacional, como o caso do MongoDB, é a mais favorável para o projeto pois permite que sejam guardados, de maneira simples, dados em formato de objeto (documentos). Oferece vantagens como *nested objects*, como é mostrado nos dois screenshots a seguir, tirados da base de dados ao modelo de dados dos documentos do jogo de Memória e Puzzle, na figura 5.1.

```

    _id: ObjectId("60a7ac1a2b434401400ce1e8")
  > assets: Object
  > config: Object
    > timer: true
    > time_to_complete: 30
    > turn_speed: 3
    > destroy_card: true
    > total_images: 0
  > thumbnail: Object
    > description: "UE"
    > title: "3 Bandeiras UE"
    > game_ref_name: "memory"
    > __v: 0
    > difficulty: 0
  (a) Jogo de Memória

    _id: ObjectId("60a79ff64040ea012427e13c")
  > assets: Object
  > config: Object
    > pieces_size: 3
    > timer: false
    > background_puzzle_im...: true
    > piece_position_helper: false
    > move_pieces_freely: false
  > thumbnail: Object
    > description: "Puzzle com background"
    > title: "Puzzle Croácia 1"
    > game_ref_name: "puzzle"
    > __v: 0
    > difficulty: 0
  (b) Jogo Puzzle

```

Figura 5.1: Documentos com diferentes objectos no campo config

Pode-se observar que ambos utilizam o mesmo documento, apenas os campos que têm valor *Object*, como o *config*, podem receber objectos diferentes, pois não são especificados os campos que o objeto tem de conter, apenas que tem de ser do tipo objeto. De notar que esta metodologia levanta problemas que necessitam de atenção no front-end, de modo a aceder sempre ao documento desejado (para não aceder a um campo que não exista no documento). Também, o facto do software ser realizado de maneira incremental, é vantajoso ter esta flexibilidade no esquema de dados para sempre que for adicionado um novo parâmetro num objeto de um documento não ter de especificar essa alteração na base de dados.

Bibliotecas utilizadas no Node.js, no Back-end

• Autenticação

- **jsonwebtoken: 8.5.1** - É um método para permitir a autenticação, sem armazenar qualquer informação sobre o utilizador no back-end ou front-end, mas sim no browser do utilizador.
- **passport: 0.4.1** - É um middleware de autenticação para o Node. Recebe os pedidos do front-end relativos à autenticação e antes de serem tratados por uma função definida pelo programador, passam por este middleware. Aqui é decidido se a autenticação é válida ou não, por meio de uma estratégia já implementada. Foram consideradas duas estratégias:
 - * **passport-local: 1.0.0** – Autentica o utilizador pelo seu username e password. Útil para realizar login na plataforma para aceder à página de administração e, de seguida, como foi autorizado, criar um JSON web token com base nos seus dados para armazenar no browser (local storage).
 - * **passport-jwt: 4.0.0** – Autentica o utilizador pelo seu JSON Web Token (JWT), criado no passport-local, verificando se é válido ou não. Útil para mandar num request de um pedido para ver se o utilizador tem autorização para o pedido e confirmar a sua identidade.

• Encriptação

- **bcryptjs: 2.4.3** – Encripta uma string recebida pelo front-end. Vai possibilitar que a password do administrador e os emails dos utilizadores sejam guardados encriptados na base de dados.

• Ligação à base de dados

- **express: 4.17.1** – Referida anteriormente, nas tecnologias do back-end, mas agora complementada com a versão.
- **mongoose: 5.11.13** – Referida anteriormente, nas tecnologias do back-end, mas agora complementada com a versão.

- **Upload de imagens**

- **multer: 1.4.2** – É um middleware para tratar de um multipart/form-data, que é comumente utilizado para dar upload de ficheiros. Neste caso, irá ser para imagens. Permite que a informação passada seja validada e escrita no armazenamento (diretoria) do servidor.
- **uuid: 8.3.2** – Permite criar um UUID, um identificador. A cada imagem guardada no servidor vai ser dado um UUID, para facilitar a organização.

- **Outras**

- **dotenv: 8.2.0** – Permite utilizar um ficheiro .env para carregar variáveis ambiente o projeto. Assim, quando o projeto for para o github não há informação privada a ser exposta, como por exemplo key's utilizadas para a encriptação.
- **nodemon 2.0.7** – Permite que sempre que se edite qualquer ficheiro javascript pertencente ao servidor, este reinicie. Útil para desenvolvimento.

5.1.3 Tecnologias de Ambiente e Exploração

As tecnologias apresentadas a seguir não são obrigatórias para o desenvolvimento da plataforma, mas são um contributo para melhorar e agilizar o processo.

Foi utilizado um template [54] que juntava TypeScript com a framework Phaser, com as principais funcionalidades de gerar um servidor de desenvolvimento e outro de produção. No de desenvolvimento tinha a funcionalidade de live reloading, que permitia que o jogo atualizasse automaticamente (F5) se fosse detada alguma alteração nos seus ficheiros. No de produção, produzia uma build, todos os ficheiros eram comprimidos e otimizados, e posteriormente podiam ser copiados para uma pasta pública de um servidor para serem acedidos estaticamente.

O Docker [55] é um software instalado localmente na máquina que utiliza virtualização ao nível do sistema operativo para conceber software em packages, conhecidos como containers. Estes containers são independentes e isolados, permitindo que cada um tenha as suas bibliotecas, dependências e ficheiros de configuração. Uma instância de uma imagem é um container. Este serviço veio mitigar o problema “it works on my machine”, ter de configurar um determinado software em várias máquinas. O Docker permite que as imagens corram em qualquer máquina da mesma forma, possibilitando que o software seja instalado mais facilmente em qualquer uma, útil para futuros contribuidores do projeto.

Também, vai permitir simular a rede dos serviços necessários. Containers individuais para o front-end, back-end e base de dados. Estes vão ser interligados utilizando o Docker Compose [56] que é uma ferramenta que permite criar uma configuração de modo a inicializar todos containers de uma vez só, assegurando as dependências entre eles.

Para Deployment e Testes, o que vai permitir mudar do ambiente local para web, vai ser utilizada uma máquina virtual alojada no Departamento de Engenharia Informática (DEI) da UC, o serviço cloud2. Esta tecnologia vai permitir realizar os testes de usabilidade à plataforma, pois disponibiliza um endereço público para se aceder.

Para controlo de versão e repositório de todo o código desenvolvido vai ser utilizado o GitHub [57].

5.1.4 Licenças

As licenças das tecnologias a utilizar foram analisadas cautelosamente e não foram verificadas incompatibilidades que evidenciassem que não se poderiam utilizar em conjunto. Também, as bibliotecas obtidas por meio do NPM [49], que é um repositório open-source, não mostrou provas em contrário. Podem ser consultadas todas as licenças no anexo C. Contudo, todo o código realizado será disponibilizado num repositório público.

5.1.5 Validação de Requisitos Não Funcionais

A utilização da biblioteca *bcryptjs* no servidor, que utiliza o algoritmo *bcrypt* [58], vai permitir encriptar os dados sensíveis recebidos pelo utilizador para serem colocados neste formato seguro na base de dados. Deste modo, os requisitos não funcionais de segurança e conformidade vão ser satisfeitos.

A utilização do Docker como tecnologia de ambiente de desenvolvimento e produção vai permitir colmatar os problemas de portabilidade, como é explicado na subsecção acima 5.1.3. Assim sendo, o requisito não funcional de portabilidade será, à partida, cumprido.

5.2 Arquitetura

Para representar a arquitetura do sistema vai ser utilizado o modelo de visualização C4 [35].

5.2.1 Modelo Arquitetural

O modelo arquitetural C4 [35] tem como objetivo criar um modelo universal, onde é possível captar, de maneira fácil e rápida, a ideia que se pretende desenvolver, permitindo ao leitor aumentar ou diminuir a ampliação consoante a sua área de interesse.

É composto por quatro níveis:

- Diagrama de Contexto - é o nível 1, que dá um contexto geral sobre a arquitetura, mais concretamente os sistemas e quem interage com estes.
- Diagrama de Contentores - é o nível 2, que especifica os diferentes serviços contidos dentro do sistema a desenvolver, representado no nível 1.
- Diagrama de Componentes - é o nível 3, onde estão presentes as diferentes componentes de um determinado serviço do nível 2.
- Diagrama de Código - é o nível 4, que consiste em dar detalhes para a implementação de código, por meio de diagramas de classes, para uma componente representada no nível anterior.

Destes níveis, apenas vão ser representados os dois primeiro, pois o sistema a desenvolver em termos arquiteturais não tem muitas restrições e não apresenta complexidade que justifique uma ampliação maior (nível 3 e 4).

Porém, vai ser explicada a organização das diretorias, tanto no cliente como no servidor, que vai permitir dar uma ampliação nos contentores da figura 5.3, sem seguir necessariamente as regras do nível 3 do modelo C4. Este maior detalhe vai abordar aspetos mais relevantes que necessitam de atenção numa fase inicial, como, por exemplo, de que forma o servidor vai disponibilizar os jogos para serem acedidos posteriormente na interface, que páginas vão existir na plataforma, entre outros.

Diagrama de Contexto

O nível 1, que consiste num diagrama de contexto, dá um contexto geral sobre a arquitetura, mais concretamente sobre os seus sistemas e as interações que tem com o exterior. Permite ver a aplicação a desenvolver de uma maneira simples, onde a ampliação da arquitetura está mais distante e o detalhe não é importante. Foca-se nas pessoas, como os atores presentes nos requisitos 4.1 e nos sistemas de software.

A audiência pretendida com este diagrama da figura 5.2 são pessoas técnicas e não técnicas, pois apenas está representado o sistema e as suas interações mais básicas com os utilizadores e outros sistemas.

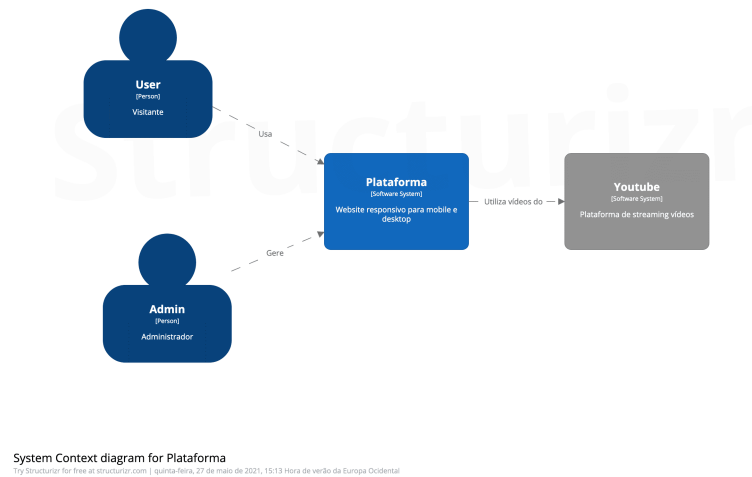


Figura 5.2: Diagrama de Contexto

Na figura 5.2, pode-se observar a azul mais claro, a plataforma como o software principal. Com esta interagem dois atores, o User que a explora e o Admin que a gere, representados a azul escuro.

- O User é o visitante da plataforma e explora as funcionalidades desta.
- O Admin é o administrador da plataforma e gere os conteúdos desta.
- O retângulo a cinzento representa um sistema externo, que não é necessário desenvolver. Este é o Youtube e o seu papel é disponibilizar URL's que permitam reproduzir os vários vídeos desejados.

Diagrama de Contentores

Segue-se o Diagrama de Contentores, nível 2, onde se realiza uma ampliação no sistema de software Plataforma do diagrama anterior, figura 5.2. Este nível é importante para dar um detalhe alto à arquitetura. São mencionadas as tecnologias a utilizar, como é realizada a comunicação entre contentores e como estes se relacionam. O objetivo passa por representar vários contentores. Um contentor é uma unidade que executa separadamente de outras. A figura 5.3, devido ao seu foco técnico, será útil para o programador que vai construir o software.

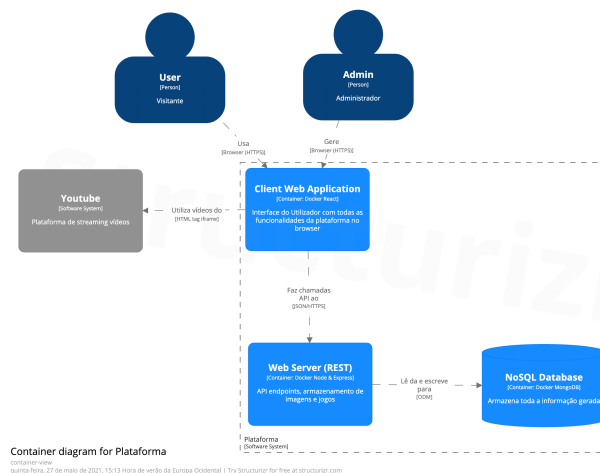


Figura 5.3: Diagrama de Contentores

De seguida, é apresentada uma breve explicação da função de cada contentor representando na figura 5.3.

- **Client Web Application** - Será o front-end da plataforma. Permite aos visitantes (User) e administrador (Admin) interagirem por meio de uma interface visual com as funcionalidades da plataforma.

Tecnologia: React, subsecção 5.1.1

- **Web Server (REST)** - Será o back-end da plataforma. Contempla toda a lógica a nível do servidor da plataforma, como o CRUD dos conteúdos, autenticação do administrador, upload de ficheiros, disponibilizar os jogos numa pasta pública, etc.

Disponibiliza uma API para a *Client Web Application* requisitar serviços REST, através de métodos HTTP, como o GET, POST e DELETE.

Tecnologia: Node & Express, subsecção 5.1.2.

- **NoSQL Database** - Será a base de dados não relacional da plataforma. Vai permitir para armazenar toda a informação gerada e consumida pela plataforma, como a inserção de trabalhos, votações, conteúdo dos jogos, entre outros.

O *Web Server (REST)* será responsável pela leitura e escrita na base de dados.

Tecnologia: MongoDB, subsecção 5.1.2.

Diretorias do Client Web Application (front-end)

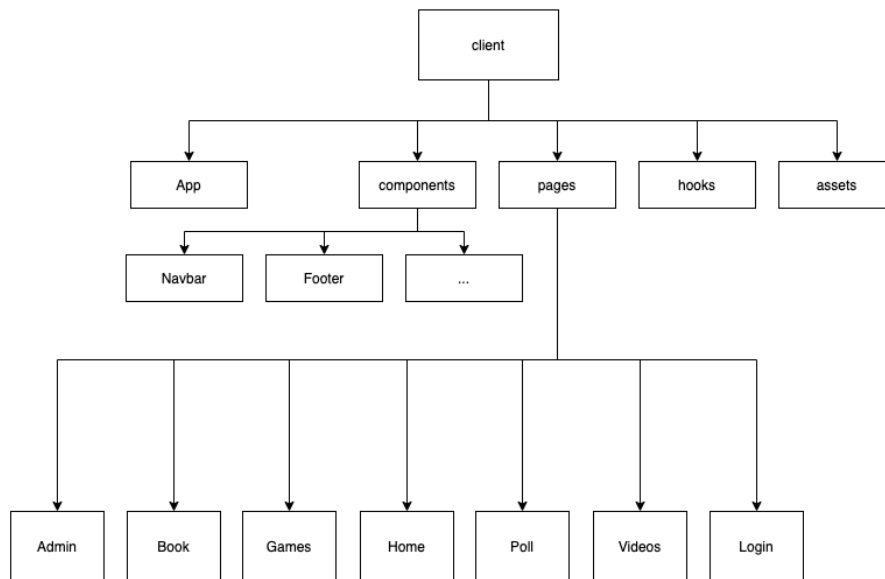


Figura 5.4: Organização das diretorias no React

Na figura 5.4 está presente o *client* que é a diretoria que engloba todo o código que vai permitir gerar o front-end. Está organizado pela seguinte estrutura:

- **App** - Em React, App é a componente que ocupa a posição mais elevada na hierarquia e nesta, através de rotas, são importadas as componentes que vão ser renderizadas nos respetivos URL's. (ex: no caminho `"/games"` será renderizada a página Games e todas as componentes que fazem parte desta).
- **components** - Pasta que contem todas as componentes que serão utilizadas no desenvolvimento das interfaces do utilizador. Uma componente representa uma parte da interface e pode ser reutilizada em mais que uma página (ex: botões, campos de texto, footer, barra de navegação, etc).

- **pages** - Pasta que contém as páginas da aplicação, onde nestas são importadas componentes da pasta *components*, de modo a construir uma interface.
- **hooks** - Pasta que contém código que permite obter dados do servidor, para reduzir código, não o repetindo.
Os *Hooks*, em React, permitem usar o estado e outras funcionalidades sem ter de criar uma classe. [59]
- **assets** - Pasta que contém conteúdos estáticos, como imagens, que podem ser utilizados nas interfaces.

Diretorias do Web Server (REST) (back-end)

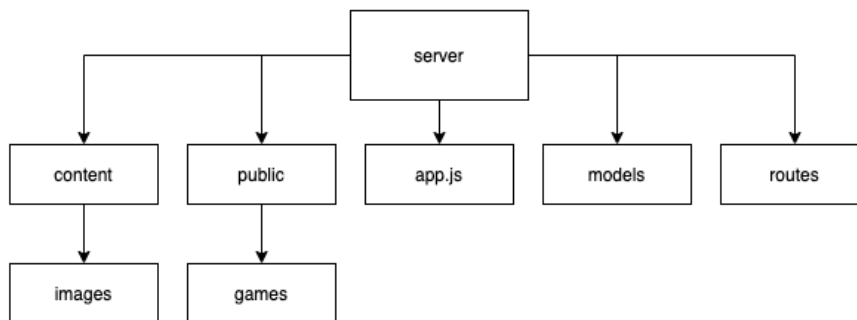


Figura 5.5: Organização das diretorias do Servidor

Na figura 5.5 está presente o *server* que é a diretoria do servidor, que engloba todo o código que vai permitir gerar a API para o front-end comunicar. Está organizado pela seguinte estrutura:

- **app.js** - Ficheiro Javascript que permite utilizar o Node em conjunto com o Express. Permite iniciar o servidor configurando as suas funcionalidades como a conexão com a base de dados, definição das pastas públicas do servidor, configurar middlewares, API endpoints, etc.
- **content** - Pasta pública criada para armazenar os uploads realizados de imagens pelo cliente. Estas são acedidas pelo endereço do servidor acrescentando à frente `"/content/images/(nome da imagem)"`.
- **public** - Pasta pública criada para se armazenarem os jogos. Estes irão ser manualmente colocados nesta diretoria pelo programador. São acedidos pelo endereço do servidor acrescentando à frente `"/games/(nome da pasta do jogo)/(nome do ficheiro .html do jogo)"`
- **models** - Pasta que contém os modelos dos dados que devem ser recebidos no servidor, e que devem estar de acordo com a estrutura definida para serem inseridos com sucesso na base de dados.

Cada modelo é explicado em detalhe na subsecção 5.2.2.

- **routes** - Pasta que permite definir vários ficheiros `.js` que correspondem a diferentes categorias de rotas. Por exemplo, permite definir um ficheiro onde vão estar organizadas todas as rotas relativas aos jogos, com métodos HTTP como:
 - GET - obter todos os jogos na base de dados, obter um jogo em específico pelo ID, obter um conjunto de jogos pelo seu tipo, etc.
 - POST - adicionar um jogo de um determinado tipo consoante a informação passada neste pedido, etc.
 - DELETE - eliminar um jogo pelo seu ID, etc.

Estas funções criadas em cada ficheiro permitem construir a API com o qual o front-end vai comunicar.

5.2.2 Modelação de Dados

Nesta subsecção é especificado o modelo de dados utilizado, modelo Entity-relationship (ER), em MongoDB, que se pode observar na figura 5.6, para a aplicação desenvolvida. Para cada entidade é realizada uma breve explicação sobre esta.

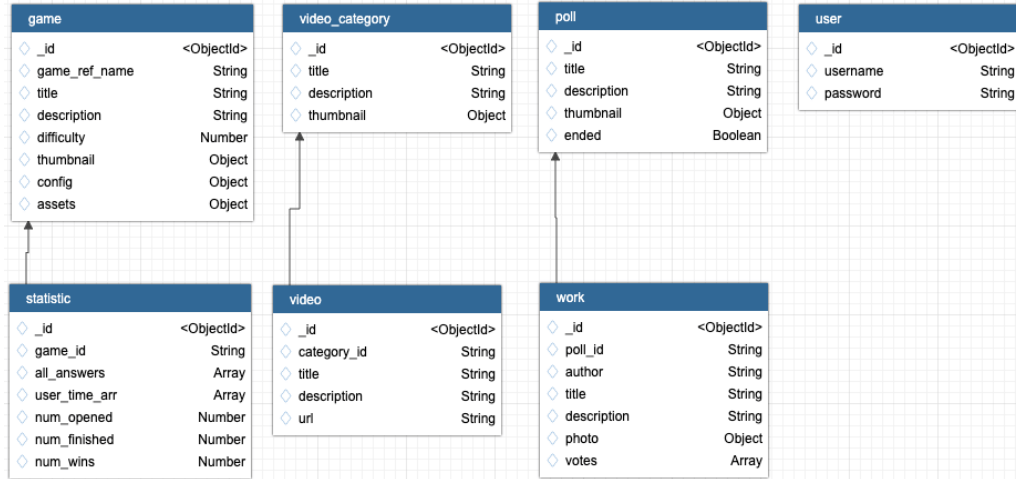


Figura 5.6: Diagrama do modelo de dados utilizado

Breve explicação das entidades da figura 5.6

O campo `_id`, presente em todos os documentos, é um identificador único criado pela base de dados para cada documento inserido.

- **game** – Documento que guarda informação sobre um jogo.
 - **game_ref_name** - Identificador do jogo, que pode ser:
 - * puzzle
 - * colorGame
 - * wordSearch
 - * quiz
 - * memory
 - * interactiveMaps
 - * crossWords
 - **title** - String que guarda o título do jogo.
 - **description** - String que guarda a descrição do jogo.
 - **difficulty** - Número que guarda a dificuldade do jogo.
 - * 0 - fácil
 - * 1 - médio
 - * 2 - difícil
 - **thumbnail** - Permite guardar um objeto com a localização do caminho da imagem no servidor.
 - **config** - Objecto único para cada tipo de jogo, cada um tem a sua própria configuração.
 - **assets** - Objecto único para cada tipo de jogo, cada um tem os seus próprios recursos (imagens, sons, etc).

Os objetos *config* e *assets*, anteriormente referidos, dizem respeito aos parâmetros editáveis de cada jogo, que podem ser consultados no capítulo de implementação, na secção 6.2, onde esse processo de escolha é descrito.

- **statistics** - Documento que guarda as estatísticas sobre um jogo
 - **game_id** - String com o ID do jogo ao qual pertence esta estatística.
 - **all_answers** - É ativado apenas para jogos de perguntas, como o quiz, onde cada elemento desse Array corresponde a um jogo completo respondido. Este jogo, é guardado num objeto que contem as respostas dadas ao jogo pelo utilizador. É apresentado um exemplo na figura 5.7 retirado da base de dados para clarificação.

```

_id: ObjectId("60b8fc7a67122102dcf25e19")
all_answers: Array
  0: Array
    0: Object
      correct: true
      user_time: 8
    1: Object
      correct: true
      user_time: 5
    2: Object
      correct: false
      user_time: 4
  1: Array
    0: Object
      correct: true
      user_time: 5
    1: Object
      correct: true
      user_time: 5
    2: Object
      correct: false
      user_time: 4
  2: Array
  3: Array
  4: Array
user_time_arr: Array
game_id: "60b8fc7a67122102dcf25e18"
game_ref_name: "quiz"
num_opened: 5
num_finished: 5
num_wins: 0
_v: 41

```

Figura 5.7: Exemplo do documento de uma estatística correspondente ao jogo Quiz

- **user_time_arr** - A este array é adicionado, a cada jogo finalizado, um número que corresponde ao tempo, em segundos, que o jogador gastou. Se o jogo não tiver tempo para completar, não é utilizado este campo.
- **num_opened** - Número de vezes que o jogo foi iniciado.
- **num_finished** - Número de vezes que o jogo foi finalizado (com vitória ou derrota).
- **num_wins** - Número de vezes que o jogo foi ganho.

Pode-se observar, pela figura 5.7 que foram abertos e jogados até ao final 5 jogos deste *Quiz*. Cada um é composto por 3 perguntas e foi guardado, para cada pergunta, se o utilizador acertou e quanto tempo gastou para responder. Também, não são contabilizadas as vitórias neste tipo de jogo.

Nota: _v(version key) - versão do documento

- **video_category** - Documento que guarda informação sobre uma categoria de vídeos.
 - **title** - String que guarda o título do categoria de vídeo.
 - **description** - String que guarda a descrição do categoria de vídeo.
 - **thumbnail** - Permite guardar um objeto com a localização do caminho da imagem no servidor.
- **video** - Documento que guarda informação sobre um vídeo.
 - **category_id** - String com o ID da categoria ao qual pertence este vídeo.
 - **title** - String que guarda o título do vídeo.
 - **description** - String que guarda a descrição do vídeo.
 - **url** - Permite guardar um endereço URL de um vídeo do Youtube.
- **poll** - Documento que guarda uma categoria de votação, onde nesta podem ser inseridos trabalhos.

- **title** - String que guarda o título da categoria de votação.
- **description** - String que guarda a descrição da categoria de votação.
- **thumbnail** - Permite guardar um objeto com a localização do caminho da imagem no servidor.
- **url** - Campo (true ou false) que permite encerrar uma votação.
- **work** - Documento que guarda dados sobre um trabalho.
 - **poll_id** - String com o ID da categoria de votação à qual este trabalho está associado.
 - **author** String que guarda o autor do trabalho.
 - **title** - String que guarda o título do trabalho.
 - **description** - String que guarda a descrição do trabalho.
 - **photo** - Permite guardar um objeto com a localização do caminho da imagem no servidor.
 - **votes** - Array que vai permitir guardar os emails encriptados (String) de quem votou no trabalho.
- **user** - Documento que guarda as credenciais do utilizador.
 - **username** - String que guarda o username.
 - **password** - String que guarda a password (encriptada).

Capítulo 6

Implementação e Resultados Finais

Este capítulo tem como finalidade fornecer os detalhes mais relevantes no processo de implementação bem como mostrar o que foi desenvolvido, o produto final.

Ao começar o desenvolvimento, foi ativada a estratégia preventiva do risco R1, da tabela 3.3. Consequentemente, as páginas das interfaces e os jogos foram desenvolvidos com base nos protótipos de baixa fidelidade realizados na fase de requisitos (subsecção 4.3.2), acarretando a consequência futura mencionada no risco. E, por este motivo, o resultado final é ligeiramente diferente dos seus protótipos.

6.1 Configuração do Ambiente

Na configuração do ambiente, criou-se a base para começar o desenvolvimento do software. A primeira tarefa foi o estudo, nomeadamente de como articular os diferentes contentores descritos na arquitetura, na figura 5.3.

Utilizou-se o docker para virtualizar estes contentores e o docker-compose para estes comunicarem na mesma rede. Criou-se um ambiente de desenvolvimento e outro de produção.

O ambiente de desenvolvimento tem características que permitem agilizar o desenvolvimento como recargar a página web caso detete alguma alteração no código, apresentação de erros, entre outros. O de produção, permite otimizar a performance, minimizando todo o código [60].

Foi também criado um script que permitia realizar *backup* à base de dados e outro para realizar *restore*. Isto permitia copiar a base de dados do ambiente de desenvolvimento para o de produção, reverter alterações, etc.

Depois de testar a rede, primeiramente com pings entre contentores e um teste simples de escrever na base de dados através do front-end passando pelo servidor web, seguiu-se para o desenvolvimento de jogos.

Posteriormente, foi requisitada uma máquina virtual no DEI (cloud2) para alojar o software, disponibilizando um endereço público, permitindo a cada reunião quinzenal com os orientadores receber um feedback mais prático, em termos de usabilidade, e também possibilitava a realização de experiências em ambientes diferentes, como tablets e telemóveis.

6.1.1 Software externo

De modo ao Centro conseguir colocar vídeos próprios na plataforma, foi assumida a responsabilidade de criar um email para criar um canal de Youtube. Foi utilizado este email também para criar uma conta GitHub para publicar todo o código desenvolvido. Todas as credenciais serão entregues no fim do projeto ao Centro.

6.1.2 Instalação do software

Como o risco R4 da tabela 3.6 se materializou, o software não migrou atempadamente para o seu ambiente definitivo. Para poderem ser realizadas alterações ao software futuramente, pode-se consultar no anexo D instruções que permitem a configuração e instalação do software realizado, que também constam no repositório do GitHub criado.

6.2 Desenvolvimento dos jogos

Esta secção permite clarificar os jogos que foram desenvolvidos de raiz e os jogos que foram realizados utilizando ajudas de bibliotecas já existentes. Permite igualmente conhecer a estrutura utilizada para desenvolver um jogo e todos os detalhes que foram tidos em conta no seu desenvolvimento.

De seguida será apresentado cada tipo de jogo presente na plataforma e fornecidos detalhes técnicos de implementação, decisões tomadas, limitações e o produto final obtido.

6.2.1 Jogos desenvolvidos de raiz

Os jogos desenvolvidos de raiz foram o Puzzle, Colorir, Sopa de Letras, Quiz e o de Memória.

Numa fase inicial os jogos foram desenvolvidos em JavaScript juntamente com a biblioteca Phaser, apenas num ficheiro. Isto levantou problemas a nível da manutenibilidade, como a dificuldade em adicionar novas funcionalidades e resolver bugs. Para contornar esta situação, foi realizado um refactoring do código para TypeScript juntamente com o template [54], referido na subsecção 5.1.3.

Os jogos foram desenvolvidos de maneira independente, isto é, cada um tinha o seu servidor, permitindo assim trabalhar simultaneamente em vários jogos.

Para organização no desenvolvimento dos jogos foram criadas as seguintes cenas ¹:

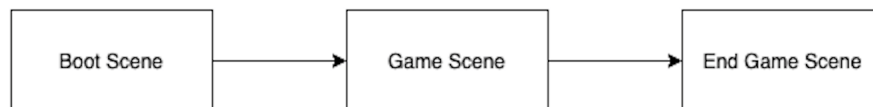


Figura 6.1: Cenas presentes nos jogos desenvolvidos

Cada cena tinha a seguinte função:

- **Boot Scene** - Obtém os dados através de uma chamada API ao endpoint do servidor que retorna os valores da base de dados para o jogo em específico. Depois de obter os dados, estes são carregados para o ambiente de jogo, chamando a Game Scene.

Nesta cena é enviada a informação à API do servidor a informar que o jogo foi iniciado.

- **Game Scene** - Onde é realizada toda a lógica que constrói o jogo. Quando o jogador perde ou ganha é chamada a End Game Scene.
- **End Game Scene** - Permite mostrar o feedback ao jogador informando se perdeu ou ganhou, bem como se pretende jogar outra vez.

Nesta cena são enviados dados à API do servidor com informações relevantes sobre a prestação do jogador.

¹Cena é a representação visual do código desenvolvido para um determinado ecrã do jogo. Este ecrã pode ser transitado para outro (outra cena)

Cada jogo contém um campo *config* e este é um objeto diferente para cada um, como se pode observar na figura 5.1, anteriormente mostrada na secção 5.1.2. Consoante os dados obtidos na boot scene, o jogo poderia variar a sua dificuldade bem como os seus conteúdos.

Os jogos foram desenvolvidos tendo em mente quais os parâmetros que faziam sentido variar e que poderiam dar outra experiência ao utilizador. Estes parâmetros são disponibilizados na página de administração para o administrador poder alterá-los, de modo a criar “novos” jogos. Também, são recolhidos dados (estatísticas) de forma anónima sobre como cada jogo é jogado e são enviados ao servidor para tratamento futuro.

Jogo de Colorir

O primeiro jogo desenvolvido foi o de Colorir. Para este jogo foi necessário estudar um algoritmo que colorisse todos os pixéis dentro de uma zona delimitada por pixéis mais escuros (traços). Foi utilizado o algoritmo Flood Fill [61]. A base de código utilizado foi encontrada em [62] e foram realizadas algumas adaptações para funcionar corretamente no caso de uso desejado (adaptar para funcionar com o Phaser, limite da cor dos pixéis de paragem, etc).

O objetivo não foi encontrar a melhor solução, mas sim uma que funcionasse corretamente, que foi o caso, pelo que não se tentou explorar outros algoritmos mais eficientes. O resultado final pode ser observado na figura 6.2.



Figura 6.2: Jogo de colorir desenvolvido

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Upload da imagem para pintar e a imagem pintada com uma sugestão de cores (podem ser pintadas outras cores).
- Poder escolher as cores disponíveis para colorir.
- Sensibilidade do algoritmo:
 - Pouco sensível - a cor no qual o algoritmo considera um traço e por isso para de colorir é uma tonalidade de preto mais escura. RGB(50, 50, 50). Caso de uso são imagens com traços escuros.
 - Muito sensível - a cor é uma tonalidade menos escura RGB(100, 100, 100). Caso de uso são imagem com traços mais acinzentados.

Limitações:

- O upload da imagem tem de respeitar as seguintes dimensões (Máx.Width: 500px, Máx.Height: 600px). Devido a bug quando se redimensionava a imagem, não se conseguia obter a cor coerente dos pixels da imagem, pois a textura da imagem não estava a ser redimensionada. Este bug foi apontado e colocado para resolver mais tarde, caso houvesse tempo.
- Não foi encontrada uma forma de prever o fim do jogo, pois:
 - Devido à liberdade para colorir, qualquer pixel da imagem pode ser colorido, não foi possível limitar certos pixels a certas tonalidades RGB. Devido a isto, também não é possível carregar uma cena que diga se o jogador ganhou ou perdeu.
 - Para contornar esta situação, as imagens que deverão ser inseridas neste jogo deverão ser imagens que permitam ser pintadas de várias formas, sem ter uma solução correta.

Como foi estudada até ao fim a possibilidade de tentar detetar o fim do jogo, o processo de refractoring foi sendo adiado, não tendo acontecido, pelo que o jogo se encontra ainda em Javascript.

Puzzle

O jogo Puzzle necessitou de ter peças idênticas ao tipo Jigsaw [63], semelhantes ao analisado na figura 2.3. Para isso foi necessário recortar as peças em diferentes formatos e, de modo a funcionar para qualquer tamanho de peças e de imagem do puzzle, houve a necessidade de ser realizado internamente no jogo, isto é, não fornecer as imagens já recortadas. O algoritmo consiste resumidamente nas seguintes etapas:

1. Redimensionar a imagem recebida para caber no frame.
2. Receber um parâmetro informando sobre o tamanho das peças do puzzle.
3. Construir peças através de retângulos e, para os encaixes, utilizar curvas de Bézier [64] para obter os cantos redondos. (formato de peça Jigsaw).
4. Explorar os “edge cases” para conseguir um tabuleiro onde as peças encaixem corretamente.
5. Cada peça vai ter como máscara o pedaço da imagem correspondente.
6. Evitar sobreposições ao posicionar aleatoriamente as peças nas bordas laterais do frame.
7. Verificar se a peça é posicionada no lugar correto no puzzle.

O resultado final pode ser observado na figura 6.3.



Figura 6.3: Jogo Puzzle desenvolvido

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Upload da imagem do puzzle.
- 4 tamanhos de peças (muito pequenas, pequenas, média ou grandes).
- Limite de tempo em segundos para acabar o puzzle.
- Possibilidade de ter ajudas como:
 - Imagem de fundo do puzzle resolvido.
 - Marca das peças no tabuleiro.
 - Mover as peças livremente pelo tabuleiro.

Sopa de Letras

O jogo Sopa de Letras necessitou de um algoritmo capaz de colocar as palavras aleatoriamente dispostas num tabuleiro, em várias direções, rodeadas de letras aleatórias. Para isso foram tidas em conta as seguintes considerações no processo de desenvolvimento:

1. Ordena as palavras por ordem decrescente de tamanho do array de palavras.
2. Recebe as direções nas quais as palavras são dispostas, que podem ser: down, right, right-down, left-down.
3. Inicializa o array de informação do tabuleiro que permite marcar/guardar as coordenadas das palavras, colocando todas as posições disponíveis.
4. Tenta colocar a palavra maior numa direção aleatória das direções recebidas e itera as coordenadas possíveis do tabuleiro, aleatoriamente, até conseguir colocar com sucesso a palavra fazendo as seguintes verificações:
 - Se a palavra está dentro do tabuleiro.
 - Se existe colisão com outras palavras e se esta é impossível (letras diferentes sobrepostas de duas palavras não pode acontecer).
5. Seguem-se duas possibilidades:
 - A palavra é colocada com sucesso e passa à seguinte do array de palavras ordenadas.
 - Quando coloca com sucesso, assinala no array de informação do tabuleiro as coordenadas ocupadas pela palavra e passa para a próxima do array ordenado de palavras.
 - A palavra não é colocada, é testada outra direção (aleatória) e o processo anterior é repetido (4.).
 - Se testar todas as direções e não for possível colocar, passa para a palavra seguinte do array ordenado de palavras;
6. No fim de tentar colocar todas as palavras, preenche as posições vazias do array de informação do tabuleiro com letras aleatórias do alfabeto.

O resultado final pode ser observado na figura 6.4.



Figura 6.4: Jogo Sopa de Letras desenvolvido

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Escolher uma ou mais das 4 direções das palavras disponíveis (down, right, right-down, left-down).
- Inserir as palavras para colocar no tabuleiro.
- Número de células horizontais e verticais (definir tamanho do tabuleiro).
- Tempo para terminar o jogo em segundos.

Quiz

O jogo de perguntas Quiz não demonstrou grandes necessidades algorítmicas, apenas sincronização de ações, como a personagem entrar, levar a pergunta e trazer a justificação, isto foi conseguido com a ajuda da framework Phaser utilizada.

Na figura 6.5 pode-se observar o resultado final obtido. Se o tempo para o jogador responder acabar, a pergunta é dada como errada, é mostrada a justificação e pode passar para a pergunta seguinte.

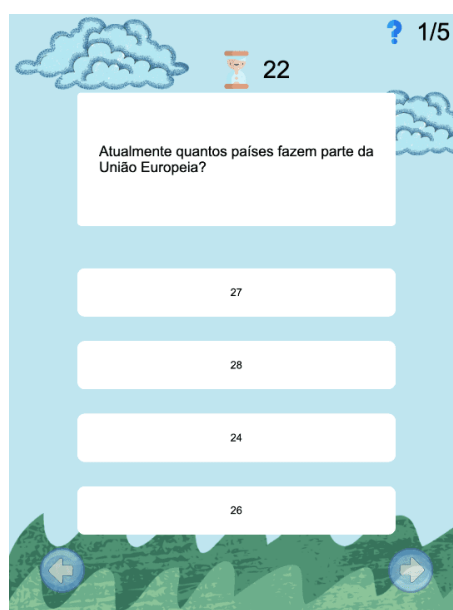


Figura 6.5: Jogo Quiz desenvolvido

No final do jogo, pode-se ver um resumo das perguntas, com número de perguntas acertadas e erradas. Ainda a possibilidade de ver as perguntas novamente e as respostas dadas.

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Inserir novas questões.
- Cada questão é constituída por:
 - Pergunta.
 - Quatro possíveis respostas.
 - Assinalar a resposta correta.
 - Justificação.
- Alterar a ordem pelas quais as questões podem aparecer no jogo.
- Tempo para responder às perguntas.

Limitações:

- O tempo é definido para todas as perguntas e não individualmente para cada.

Memória

O jogo de Memória necessitava de um algoritmo que colocasse as imagens viradas ao contrário, dispostas em modo galeria e, quando o utilizador clicasse numa, esta se virasse. Para isso foi necessário um efeito realista. Este efeito foi obtido por meio de uma distorção encolhendo no eixo do X, para dar o efeito de viragem. Aqui, a meio do efeito, troca-se a textura pela imagem que se quer mostrar ao utilizador e volta-se a distorcer o eixo X para o valor normal. É também efetuada uma ligeira distorção no eixo do Y, aumentando-o, para o efeito de levantar a carta. De resto, o jogo apenas permite virar duas cartas e, se estas forem o mesmo par, o utilizador é notificado que acertou, caso contrário, as cartas voltam-se a virar. O resultado final pode ser observado na figura 6.6.



Figura 6.6: Jogo de Memória desenvolvido

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Tempo em segundos para adivinhar todos os pares escondidos ou tentativas máximas que pode efetuar.
- 4 velocidades que as duas cartas viradas pelo utilizador ficam visíveis (muito rápido, rápido, normal ou lento).
- Eliminar o par no fim de descoberto.
- Número de pares de cartas, que podem ser 3, 6, 8 ou 10.

6.2.2 Jogos desenvolvidos recorrendo a bibliotecas externas

O jogo de Mapa Interativos e de Palavras Cruzadas foram desenvolvidos recorrendo a ajudas de bibliotecas externas, pois pelo estudo efetuado dificilmente se conseguiriam implementar de raiz. Isto será explicado mais ao pormenor nesta subsecção. Deste modo, não se recorreu ao Phaser, mas a Componentes do React já disponibilizadas pelas bibliotecas que utilizavam SVG's em vez da Canvas.

Mapas Interativos

O jogo de Mapas Interativos necessitava de um mapa em que o utilizador pudesse clicar para interagir com este. Descartou-se a possibilidade de utilizar a canvas, a framework Phaser, pois seria necessário desenhar um mapa com esta, o que estaria fora do alcance. Também, a possibilidade de utilizar uma imagem do mapa e o algoritmo Flood Fill, utilizado no jogo de colorir, figura 6.2, para definir um país, iria conter muitos “edge cases”, pois era necessário manualmente dizer que um determinado número de pixéis correspondia a um país, e isto para todas as possibilidades.

Assim, utilizou-se a biblioteca *react-simple-maps*, explicada em 5.1.1, que fornece um mapa que foca a Europa em SVG. Este formato de imagem já disponibiliza os contornos de cada país e, com a ajuda da biblioteca, permitia que cada clique do utilizador fosse traduzido num país.

Deste modo, o jogo foi desenvolvido em componentes React e foi criada uma lógica adicional, simultaneamente com as funcionalidades da biblioteca, que permitia dar uma experiência de utilizador sobre conhecimentos geográficos de países, respondendo a questões aleatórias sobre a localização destes. O resultado final pode ser observado na figura 6.7.

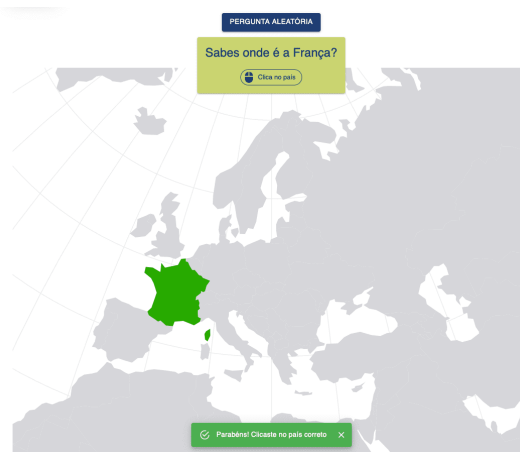


Figura 6.7: Jogo de Mapas Interativos

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Adicionar novas perguntas.
- Cada pergunta é composta pela questão e o país que corresponde à resposta correta.

Limitações:

- Como o jogo está sempre a gerar perguntas aleatórias, não é possível dar o jogo como terminado e, por este motivo, não contem cena final.
- Apenas foram traduzidos os códigos do países que pertencem à União Europeia, pelo que se for selecionado um país fora, é mostrada a mensagem de erro *Errado! Clícaste num país fora da União Europeia*. Não é apresentado, para este caso, o país que o jogador clicou.

Palavras Cruzadas

O jogo de Palavras Cruzadas necessitava de um algoritmo capaz de gerar um tabuleiro com palavras intercetadas. Pelo estudo efetuado, fazê-lo de raiz mostrou-se complexo e optou-se por não se ir por esse caminho, essencialmente devido ao consumo de tempo. Assim, procurou-se uma alternativa e a encontrada foi deixar a complexidade de criar o tabuleiro, com as palavras intercetadas, no administrador, isto é, este era responsável por indicar as coordenadas onde as palavras iriam ser colocadas (um exemplo de como criar um tabuleiro de palavras pode ser visto no anexo G.1, na figura G.1).

Assim, encontrou-se uma biblioteca que já realiza todo o trabalho de tabuleiro e experiência de utilizador, *react-crossword* analisada em 5.1.1 e adaptou-se para o caso de uso desejado, aproveitando algumas das funcionalidades disponíveis como preenchimento automático das palavras, reset do tabuleiro, input do teclado, entre outras. O resultado final pode ser observado na figura 6.8.

Capitais de Países da União Europeia



Horizontal Vertical

Figura 6.8: Jogo de Palavras Cruzadas

As preocupações a nível de parâmetros editáveis pelo administrador foram:

- Número variável de questões/pistas que se traduzem em palavras horizontais e verticais.
- Cada questão é composta por:
 - Número de questão.
 - Pista ou questão.
 - Resposta (corresponde à palavra que o utilizador adivinha no tabuleiro).
 - Número da linha (coordenada X).
 - Número da coluna (coordenada Y).

Limitações:

- Também não foi criado uma cena final para este jogo, pois não foi desenvolvido na framework Phaser, logo não se pôde aproveitar a Game End Scene, figura 6.1. Contudo, o tabuleiro de jogo transita para cor verde quando é completado com sucesso.
- Não é possível escrever palavras que tenham acentos, como CÃO. Terá de ser escrito CAO.

6.3 Desenvolvimento da plataforma

A plataforma foi desenvolvida em simultâneo com os jogos, pois existiu a necessidade de, para se realizarem apresentações nas reuniões quinzenais, haver organização de como o conteúdo era apresentado. Como dito anteriormente, na secção 5.1.1, foi escolhido React, para desenvolver as interfaces de utilizador.

Nesta subsecção vão ser abordadas as páginas que compõem a plataforma, referidas no diagrama da figura 5.4.

6.3.1 Página Home

Corresponde à página de entrada da plataforma, a que é mostrada por predefinição quando se acede, figura 6.9.



Figura 6.9: Página inicial da plataforma

6.3.2 Página Games

Página onde estão disponíveis os jogos para serem jogados. Para integrar os jogos desenvolvidos de raiz, foi utilizado um iframe que continha o endereço do caminho que o jogo ocupava na pasta pública do servidor. Os jogos que utilizaram bibliotecas externas e, consequentemente, foram desenvolvidos no React, não necessitaram desta integração por iframe. Foram integrados comportando-se como uma componente.

Os jogos estão divididos nos seus vários tipos (puzzle, memória, sopa de letras, etc) e, quando se clica num tipo, são mostradas as várias cartas que representam os jogos ordenados por dificuldade. Cada carta contém uma imagem referente a jogo, título, descrição e a sua dificuldade. Ao clicar no botão *Jogar*, o jogo é iniciado e o utilizador é redirecionado, tudo automaticamente. O resultado final pode ser visto na figura 6.10.

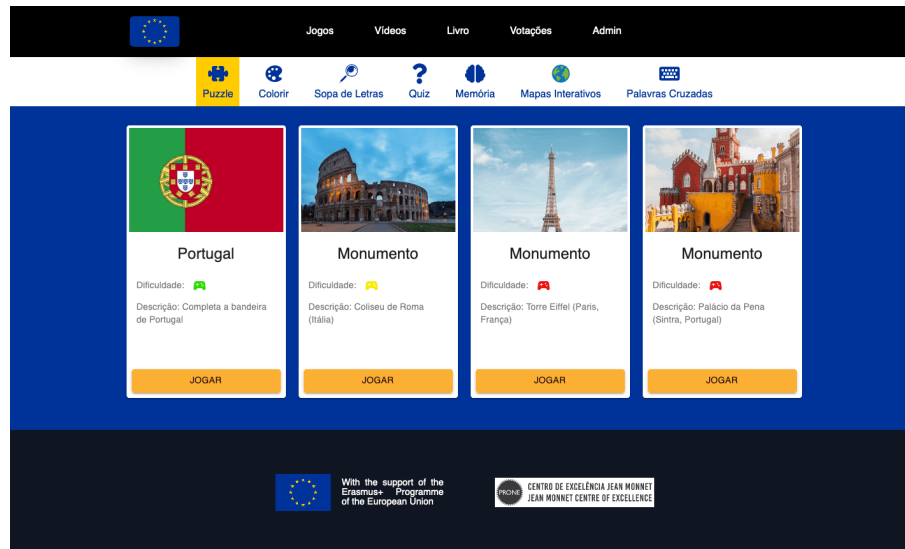


Figura 6.10: Página dos Jogos

6.3.3 Página Vídeos

O objetivo desta página é disponibilizar os vídeos em diferentes categorias. Para isso utilizaram-se várias listas que representam as diferentes categorias. Cada lista contém vídeos que estão alojados no Youtube e que se reproduzem automaticamente se clicados. A lista pode ser "rodada" para a direita e para a esquerda, de modo a irem surgindo os vídeos não estão na janela visível. Também, é possível ver todos os vídeos de uma categoria de forma isolada, carregando no botão *VER TODOS*. O processo de reprodução de um vídeo é semelhante ao anterior. O resultado final é mostrado na figura 6.11.

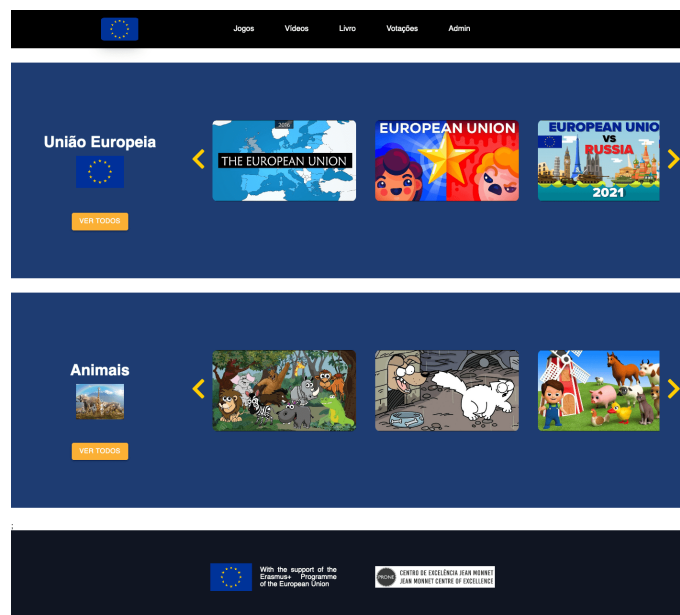


Figura 6.11: Página dos Vídeos

6.3.4 Página Book

O objetivo desta página é ter um livro que concede a experiência de utilização de um livro real, onde se podem folhear as páginas. Estas podem ser narradas e ver o número da página atual.

No desenvolvimento desta página, foi ativada a estratégia preventiva do risco R2, tabela 3.4. Os requisitos ficaram implementados, contudo o conteúdo da página é considerado *dummy data* e é proveniente da plataforma oficial da União Europeia, explorada na subsecção 2.2.1, como se pode ver na figura 6.12.

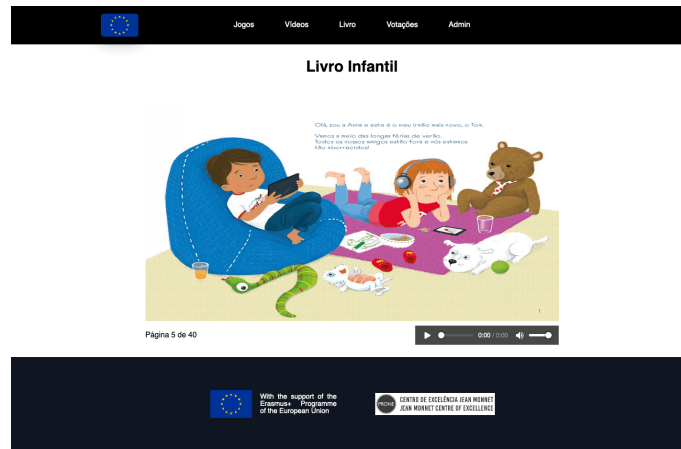


Figura 6.12: Página do Livro Infantil

6.3.5 Página Poll

O objetivo desta página era ter várias categorias de votações, onde cada categoria era clicável e continha trabalhos. Dentro de cada categoria, era possível votar num trabalho através de dados pessoais, com o nome e o email. Não era possível votar no mesmo trabalho duas vezes com o mesmo email. Para isso, sempre que o sistema recebe um voto, encripta o email e verifica se este está na lista de emails encriptados que já votaram. Se já estiver na lista, avisa o utilizador que já votou. Caso contrário, insere o email na lista e o voto é válido. Cada trabalho pode ser ampliado para ser visto com mais detalhe. O resultado implementado ser visto na figura 6.13.



Figura 6.13: Página das votações

6.3.6 Página Login

Esta página permite ao administrador autenticar-se perante a plataforma para aceder às funcionalidades de administração.

Não foi criado nenhum botão visível na plataforma para nenhuma criança, por engano, ser redirecionada para esta página. Para aceder a esta página, o utilizador vai à página inicial da plataforma e acrescenta `"/login"` à frente. Seguidamente, necessita de preencher o seguinte formulário presente na figura 6.14.

The screenshot shows the login page with a dark header containing the European Union flag and navigation links: Jogos, Vídeos, Livro, and Votações. Below the header is the title 'Login' and a form with two input fields: 'Username' and 'Password'. A blue button labeled 'ENTRAR' is positioned below the form. At the bottom of the page, there are logos for the Erasmus+ Programme of the European Union and the Jean Monnet Centre of Excellence.

(a) Página de login

The figure contains two screenshots of the login page. The left screenshot shows the form with empty 'Username' and 'Password' fields, each with a red border and a red error message below it: 'Username é obrigatório' and 'Password é obrigatória'. The right screenshot shows the form with 'admin' in the 'Username' field and '****' in the 'Password' field. A red error message 'Username ou Password incorreto(s)!' is displayed below the form. Both screenshots have a blue 'ENTRAR' button at the bottom. Below the screenshots are two identical footer sections with logos for the Erasmus+ Programme of the European Union and the Jean Monnet Centre of Excellence.

(b) Quando não são inseridos dados e é submetido o formulário

(c) Quando são inseridos dados inválidos

Figura 6.14: Página de login

Quando os dados inseridos são válidos, é adicionado um JWT [65], através da biblioteca `jsonwebtoken` mencionada na subsecção 5.1.2, com os dados encriptados (assinatura), à local storage do browser do utilizador e é redirecionado para a página de administração.

6.3.7 Página Admin

Esta página é intitulada de *Dashboard*. Quando o utilizador se autentica na plataforma com sucesso é redirecionado para esta página e fica disponível um botão na barra de navegação *Admin* que permite aceder a esta ação a qualquer momento.

Todos os endereços que só devem ser acedidos pelo administrador são protegidos. Estes endereços, que se traduzem em páginas, estão referidos nesta subsecção e sempre que são acedidos, o JWT presente no browser do utilizador, é decodificado e verifica-se a autenticidade de quem realizou o pedido. Se for válida, o utilizador pode aceder, caso contrário é redirecionado para a página

principal da plataforma 6.3.1.

Nesta página, o administrador consegue controlar vários aspetos da plataforma como:

- **Gerir o conteúdo**

Nesta secção da página, pode realizar alterações ao conteúdo presente nos:

- **Jogos**, figura 6.15

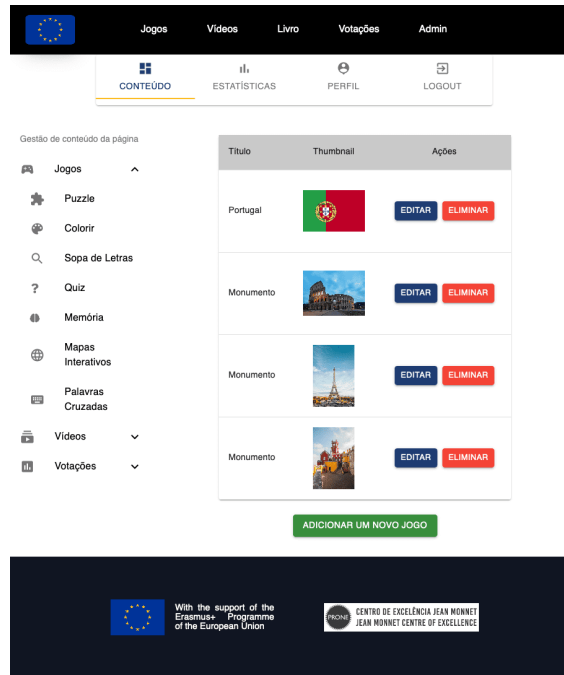
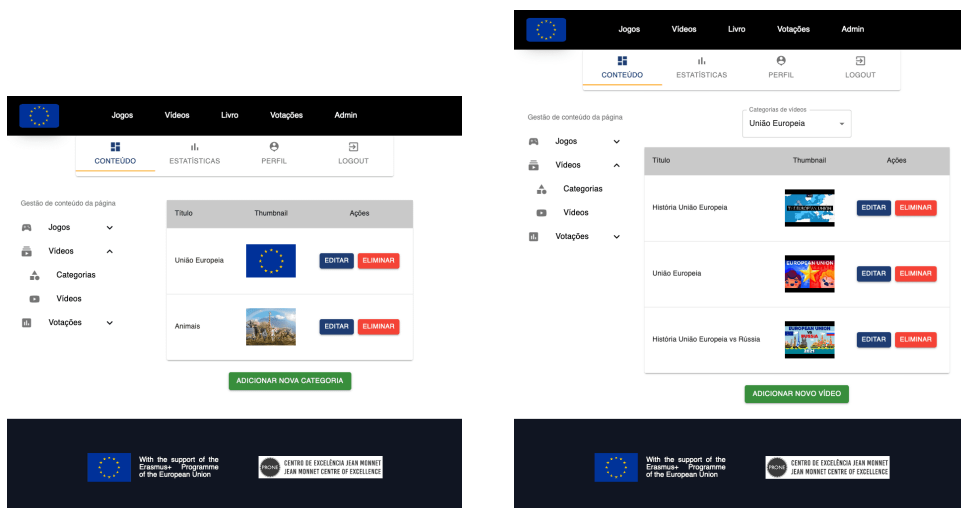


Figura 6.15: Gerir Jogos

- **Vídeos**, figura 6.16

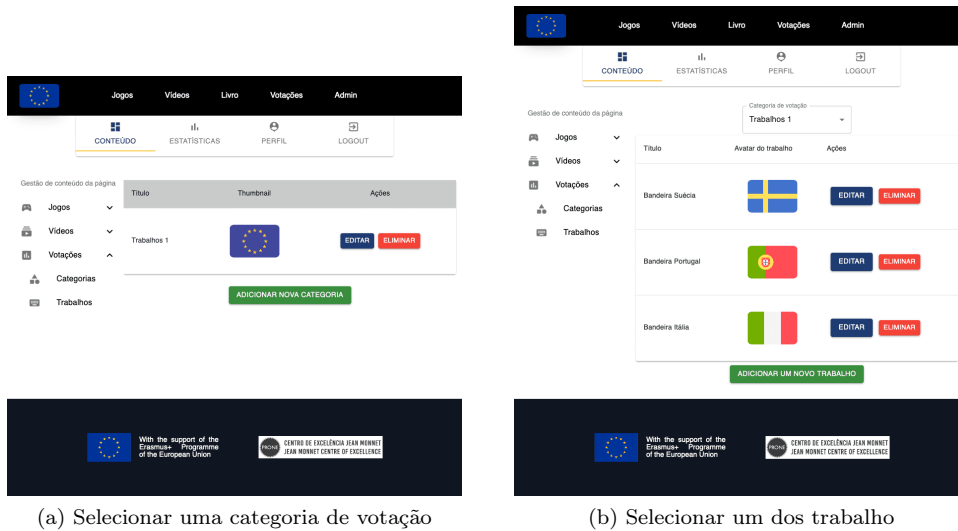


(a) Selecionar uma categoria de vídeo

(b) Selecionar um dos vídeos pertencente a uma categoria

Figura 6.16: Gerir Vídeos

- **Votações**, figura 6.17



(a) Selecionar uma categoria de votação

(b) Selecionar um dos trabalho

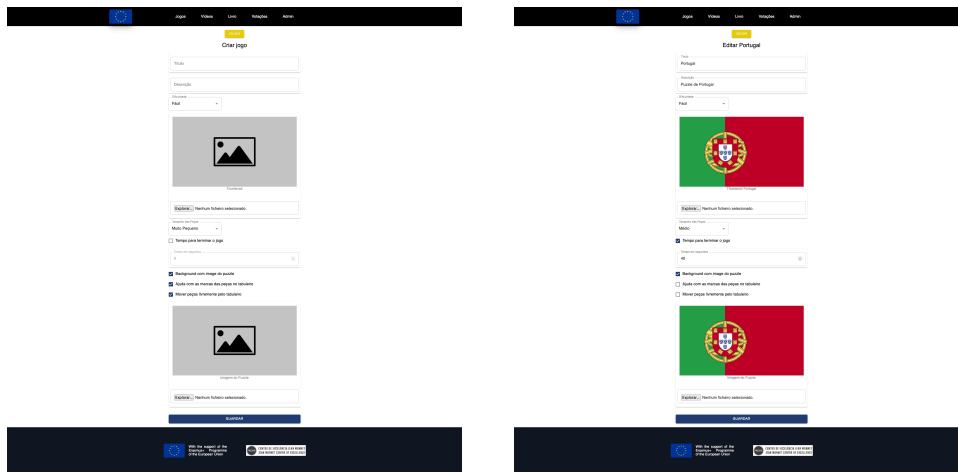
Figura 6.17: Gerir Votações

Formulários

Pode-se observar nas figuras acima um sistema que apresenta as funcionalidades designadas de CRUD. Aqui é possível ler, criar, atualizar e eliminar o conteúdo. Ora, para eliminar clica-se no botão *ELIMINAR* a vermelho. Para criar, no botão (*ADICIONAR UM NOVO ...*) a verde. E, por fim, para atualizar no botão de *EDITAR*.

Quando o administrador pretende criar ou atualizar um jogo é redirecionado para uma página que contem um formulário. Este já se encontra com os campos preenchidos se anteriormente clicou em editar, ou seja, o sistema lê os dados que estão na base de dados e apresenta-os. Encontra-se vazio se pretende criar conteúdo novo, porque ainda não há nada para ler.

De seguida, é apresentado um formulário sem e com dados, para exemplo, figura 6.18.



(a) Formulário do jogo Puzzle, opção *ADICIONAR UM NOVO JOGO*

(b) Formulário do jogo Puzzle, opção *EDITAR*

Figura 6.18: Formulário

O formulário da figura 6.18 apresenta um esquema de validação e este que tem como objetivo validar os dados antes de serem enviados ao servidor. Permite que a cada campo seja atribuído um input específico que deverá receber para ser válido. Cada formulário diferente tem o seu próprio esquema. De seguida é apresentado o esquema de validação utilizado para validar os campos do puzzle da figura 6.18.

```

export const getSchemaPuzzle = (createNew) => {
  return yup.object().shape({
    title: yup.string().required(geralNotEmpty),
    description: yup.string().required(geralNotEmpty),
    difficulty: yup.number().required(),
    thumbnail: createNew ? getRequiredFileSchema() : getOptionalFileSchema(),
    config: yup.object().shape({
      pieces_size: yup.number().required(),
      background_puzzle_image: yup.boolean(),
      piece_position_helper: yup.boolean(),
      move_pieces_freely: yup.boolean(),
      timer: yup.boolean(),
      time_to_complete: yup.number().when("timer", {
        is: true,
        then: yup
          .number()
          .min(0)
          .max(3 * 60) // 3 min
          .required(
            "0 tempo deve ser superior a 0 segundos e inferior a 200 segundos"
          ),
      }),
    }),
    assets: yup.object().shape({
      puzzle_image: createNew
        ? getRequiredFileSchema()
        : getOptionalFileSchema(),
    }),
  });
};

```

Figura 6.19: Esquema de validação do formulário do Puzzle

Na figura 6.19 podem-se observar os campos que o formulário da figura 6.18 permite preencher para, depois de validado, ser enviado para a base de dados para ser armazenado. Explicando resumidamente o que acontece no código, a função recebe um boolean *createNew* que pretende diferenciar entre criar ou editar o jogo, pois quando cria é necessário que as imagens como a thumbnail e a imagem do puzzle sejam de preenchimento obrigatórias. Contudo, quando é para editar, como as imagens já foram previamente inseridas, estas não são obrigatórias no preenchimento do formulário. Os restantes campos, definem-se com funções predefinidas oferecidas pela biblioteca *yup*, referida em tecnologias (subsecção 5.1.1).

- **Ver estatísticas de jogos**

Nesta secção da página, o administrador pode ver estatísticas dos jogos implementados de raiz, os presentes na subsecção 6.2.1, excepto o jogo de Colorir. Só se realizou para estes pois eram os únicos que continham cena final, e só aqui são enviados os dados para o servidor.

As estatísticas podem ser procuradas pelo nome do jogo e o seu título, esta forma demonstrou-se a mais rápida e eficiente em termos de encontrar o jogo desejado, comparativamente a colocar o respetivo ID.

Foram realizados dois tipos de estatísticas. Como o jogo Quiz engloba perguntas, seria interessante ver individualmente cada pergunta e ver a percentagem de vezes que foi acertada, por exemplo. Foram então recolhidos os seguintes parâmetros neste tipo de jogo:

Segue-se uma breve explicação dos parâmetros recolhidos presentes na figura 6.20:

- Questões - Todas as perguntas realizadas no jogo, para uma fácil e rápida visualização.
- Certas - Número de respostas certas recolhidas para cada pergunta.
- Erradas - Número de respostas erradas recolhidas para cada pergunta.
- Certas (%) - Percentagem de perguntas certas em relação ao número total de perguntas respondidas.

Permite tirar como conclusão se a pergunta que está a ser realizada é fácil, ao ponto de muitos jogadores a acertarem e consequentemente esta percentagem ser elevada. Também permite ver o contrário, se é difícil.

- Tempo (seg) - É realizada a média dos tempos recolhidos para cada pergunta (se o jogo tiver o tempo de resposta ativo).

Permite tirar como conclusão se o tempo dado para responder à pergunta é suficiente ou não.

- Abertos - Número de vezes que o jogo foi clicado na página de jogos e começou.
- Finalizados - Número de vezes que o jogo chegou ao fim, isto é, foi emitida a cena final onde aparece um resumo das perguntas que acertou e errou.

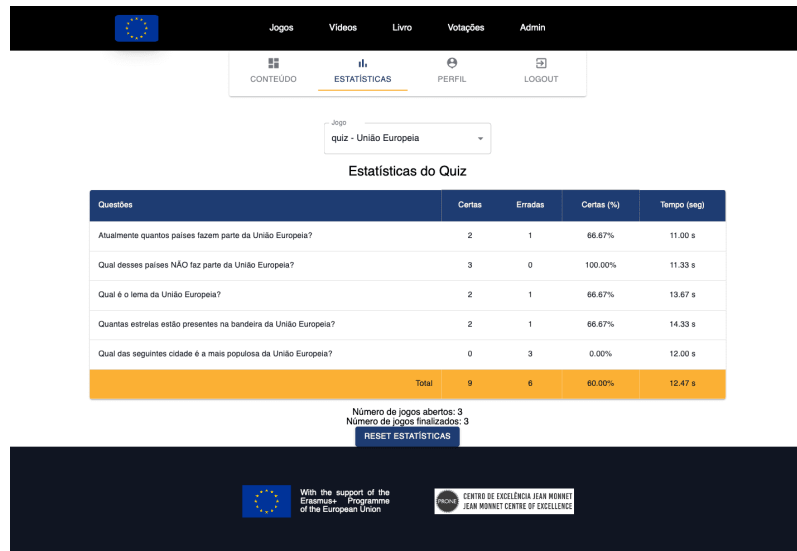


Figura 6.20: Estatísticas de um jogo Quiz

No final, é contabilizada a soma dos parâmetros anteriormente mencionados que permite ver o desempenho no geral de todas as perguntas. Permite tirar conclusões mais gerais, como se as perguntas realizadas no jogo são difíceis ou se é necessário aumentar o tempo disponível para responder.

Ainda, são contados o número de vezes que o jogo foi aberto e que foi finalizado. A principal conclusão que se consegue retirar com este parâmetro é que houve um motivo que não levou o jogador a chegar ao fim do jogo.

Os restantes jogos, como não tinham perguntas, tiveram de seguir outra abordagem, mais geral, como se pode ver na figura 6.21

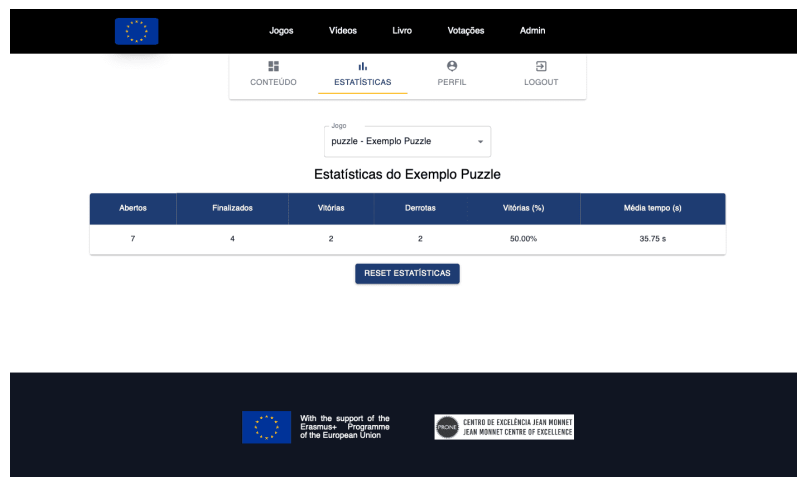


Figura 6.21: Estatística de um jogo Puzzle

Segue-se uma breve explicação dos parâmetros recolhidos presentes na figura 6.21:

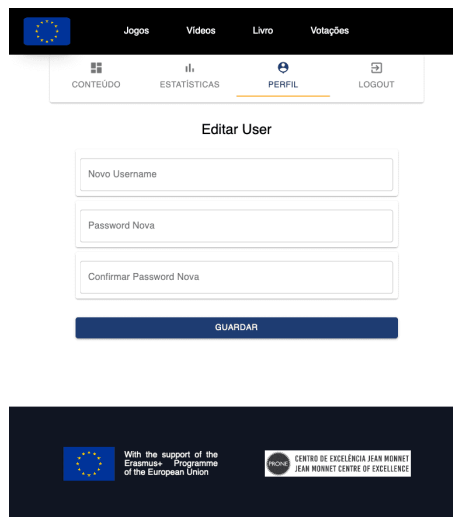
- Abertos - Número de vezes que o jogo foi clicado na página de jogos e começou.
- Finalizados - Número de vezes que o jogo chegou ao fim, isto é, foi emitida a cena final.
- Vitória - Número de vezes que o jogo chegou à cena final, informando que o jogador ganhou.
- Derrota - Número de vezes que o jogo chegou à cena final, informando que o jogador perdeu.

- Vitórias (%) - Percentagem de vitórias em relação ao total de jogos finalizados.
Permite tirar como conclusão se é necessário alterar algum parâmetro de modo a facilitar o jogo, se não tiver a percentagem de vitórias desejada pelo administrador.
- Média Tempo (s) - Se o jogo tiver a opção de ter um tempo limite para ser finalizado ativa, é realizada a média dos tempos recolhidos em cada jogo.
Permite tirar como conclusão se o tempo para finalizar um jogo é suficiente ou não.

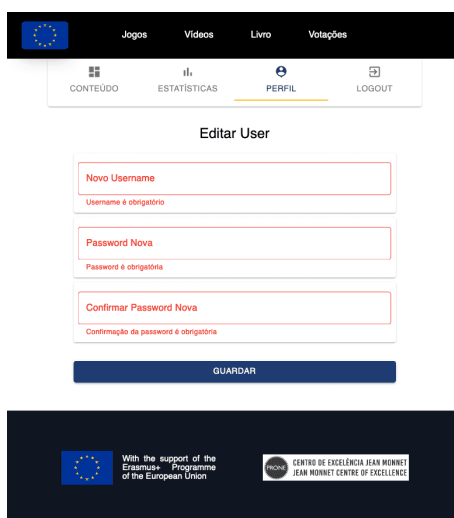
As estatísticas de cada jogo são passíveis de dar reset (botão *RESET ESTATÍSTICAS*) e sempre que se altera algum parâmetro de um jogo é realizado um reset automático para evitar dados incongruentes.

• **Alterar o perfil**

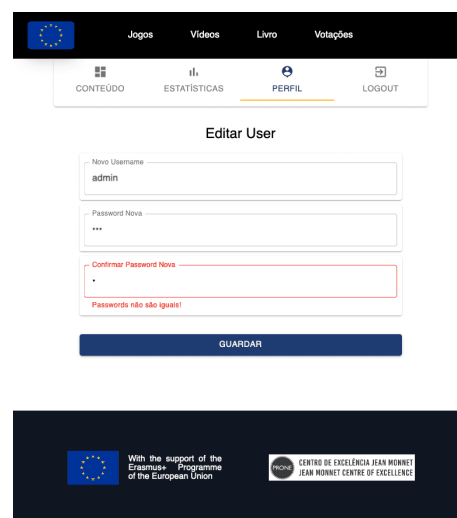
Nesta secção da página, o administrador pode alterar os seus dados de acesso, username e password, se necessitar. O que se pode ver na figura 6.22 é o formulário que necessita de preencher. Se preencher o formulário corretamente, com os dados novos é alterado o JWT do browser pelo mais recente, o que não requer que o utilizador se volte a autenticar com os novos dados para realizar ações na página de administração. Também, aparece uma mensagem a dizer que a ação foi realizada com sucesso.



(a) Formulário para alertar os dados



(b) Formulário com dados não preenchidos



(c) Formulário com as passwords novas incoerentes

Figura 6.22: Alterar o perfil

- **Realizar logout da plataforma**

Nesta secção da página, o administrador pode clicar no botão *LOGOUT*, presente na página de administração, como se pode ver, por exemplo, na figura 6.22. De seguida, o JWT é removido da local storage do browser e o utilizador é redirecionado para a página inicial da plataforma, não tendo mais acesso a esta página.

6.4 Requisitos Implementados

Os requisitos funcionais implementados podem ser consultados na tabela E.1 do anexo E.

Capítulo 7

Testes ao Software

O capítulo descreve os testes realizados ao software desenvolvido. Primeiramente, irão realizar-se testes funcionais que têm como objetivo verificar se os requisitos funcionais previamente acordados foram cumpridos. Também, testes não funcionais, pelo mesmo motivo. Ainda, testes de usabilidade com utilizadores reais para aferir se está adequado ao foco de idade pretendido e se tem as funcionalidades desejadas.

7.1 Testes Funcionais

Os principais objetivos dos testes funcionais são verificar as funcionalidades do software, nomeadamente testar os Requisitos Funcionais, presentes na tabela 4.5. Vão ser testados todos os que tenham prioridade *Must have*, pois são fulcrais estarem completos para o sucesso do projeto, bem como os de outras prioridades que se decidiram implementar ao longo do trabalho. (ver na tabela E.1).

Para cada requisito, foram desenvolvidos casos de teste que envolviam testar o comportamento das funcionalidades do software e observar a sua resposta. Para isso, definiu-se um caso de teste da seguinte forma:

ID	Identificador do teste
ID Requisito	Identificador do requisito ao qual pertence este caso de teste
Cenário	Breve descrição textual explicando no que consiste o teste
PASS/FAIL	PASS se o comportamento do software for o expectável FAIL se o comportamento do software for diferente do cenário esperado, seguido de uma breve explicação

Tabela 7.1: Exemplo de um Teste Funcional

De modo a se conseguir testar funcionalidades desenvolvidas em específico, que não foram cobertas pelos testes funcionais e que não constam nos requisitos funcionais, decidiu-se realizar testes Adhoc [66]. Este tipo de testes é realizado por alguém que conheça o sistema, utilizando a sua intuição. De modo a tornar o processo mais eficiente, será preenchido o seguinte template para cada teste deste tipo realizado.

ID	Identificador do teste
Cenário	Breve descrição textual explicando no que consiste o teste
PASS/FAIL	PASS se o comportamento do software for o expectável FAIL se o comportamento do software for diferente do cenário esperado, seguido de uma breve explicação

Tabela 7.2: Exemplo de um Teste Adhoc

Os testes funcionais realizados, juntamente com os Adhoc, podem ser consultados no anexo F. Segue-se um resumo:

- Um fator que levou à maioria dos testes documentados não conterem comportamentos inesperados foi serem realizados com antecedência, o que permitiu a correção de todos os que apresentaram FAIL, nomeadamente nos testes Adhoc gerados.
- Contudo, existem testes que apresentaram FAIL mas já se sabia previamente o motivo da falha, como se pode observar na explicação de cada teste falhado, no qual se fazem referências a limitações ou materializações de riscos anteriormente falados. Estes testes podem ser encontrados no anexo F, com os identificadores (ID):
 - FT-2-02
 - FT-4-02
 - FT-6-04

Testes às Estatísticas dos Jogos

Para testar que as estatísticas são corretamente apresentadas, primeiramente foi verificado que o jogo quando finalizava enviava os dados corretamente para o servidor. Foi, por exemplo, jogado um Puzzle que tinha 50 segundos para ser finalizado e foram gastos 20 segundos desse tempo. O valor esperado na base de dados seriam esses 20 segundos, juntamente com informação que o jogo foi aberto, finalizado e ganho. Foi verificado cada jogo, que tinha estatísticas disponíveis, da mesma forma.

Como esta informação era armazenada corretamente, para efeitos de rapidez, adicionaram manualmente dados na base de dados, para cada jogo.

• Teste Estatísticas Puzzle

Selecionou-se um jogo Puzzle com tempo para finalizar (máximo 50 segundos) para testar. Vai permitir concluir se a informação apresentada nas estatísticas, como a figura 6.21, está correta. Foram realizadas as seguintes ações:

1. Abrir o jogo 3 vezes;
2. Ganhar o jogo 2 vezes (tempo que demorou: 20s e 23s);
3. Perder o jogo 2 vezes (deixar o tempo acabar, 50s);

	Abertos	Finalizados	Vitórias	Derrotas	Vitórias (%)	Média Tempo (s)
Esperado	7	4	2	2	50%	35.75s

Tabela 7.3: Estatísticas Puzzle - Esperado

Pode-se observar que os valores são iguais aos da figura 6.21.

Para os restantes jogos, foi verificado se a escrita na base de dados era realizada de maneira correta, pois o algoritmo que era utilizado para fazer a percentagem de vitórias e média de

tempo era igual ao Puzzle. Como todos escreviam bem, pode-se concluir que as estatísticas para estes jogos (Puzzle, Memória e Sopa de Letras) são bem apresentadas.

- **Teste Estatísticas Quiz**

Selecionou-se um jogo Quiz que tinha 20 segundos de tempo para responder a cada pergunta. Vai permitir concluir se a informação apresentada nas estatísticas, como a figura 6.20, está correta. Foram realizadas as seguintes ações:

1. Jogo 1
 - Questão 1 - correta (tempo: 8s)
 - Questão 2 - correta (tempo: 11s)
 - Questão 3 - errada (tempo: 12s)
 - Questão 4 - correta (tempo: 16s)
 - Questão 5 - errada (tempo: 10s)
2. Jogo 2
 - Questão 1 - errada (tempo: 9s)
 - Questão 2 - correta (tempo: 14s)
 - Questão 3 - correta (tempo: 17s)
 - Questão 4 - correta (tempo: 10s)
 - Questão 5 - errada (tempo: 13s)
3. Jogo 3
 - Questão 1 - correta (tempo: 16s)
 - Questão 2 - correta (tempo: 9s)
 - Questão 3 - correta (tempo: 12s)
 - Questão 4 - errada (tempo: 17s)
 - Questão 5 - errada (tempo: 13s)

Questões	Certas	Erradas	Certas (%)	Tempo (seg)
Questão 1	2	1	66.67%	11 s
Questão 2	3	0	100%	11.33 s
Questão 3	2	1	66.67%	13.67 s
Questão 4	2	1	66.67%	14.33 s
Questão 5	0	3	0%	12 s
TOTAL	9	6	60.00%	12.47 s

Tabela 7.4: Estatísticas Quiz - Esperado

Pode-se observar que os valores são iguais aos da figura 6.20. Logo, os dados apresentados para os jogos Quiz estão corretos.

7.2 Testes Não Funcionais

Esta secção descreve os testes realizados aos requisitos não funcionais (presentes na subsecção 4.3.3), que eram passíveis de serem testados.

7.2.1 Testes à Usabilidade

Para este requisito, foram realizados testes em utilizadores reais nos focos de idades pretendidos, entre os 6 e 9 anos para os jogos e vídeos da plataforma e um adulto para a página de administração.

Testes com crianças

Com consentimento do encarregado de educação, o teste consistiu em expor a criança à plataforma, para realizar um conjunto de tarefas. O teste foi conduzido pelo adulto, que ia dizendo à criança o que fazer, de maneira imparcial, isto é, sem explicitamente dizer que botões tinha de carregar, apenas lendo as tarefas.

Gravaram-se, igualmente, vídeos pequenos, centrados apenas na interação da criança com o ecrã, para efeitos de estudo futuro. No final, pediu-se ao adulto que respondesse a algumas questões relativas ao desempenho da criança:

- Se conseguiu iniciar o jogo pedido (tipo de jogo e a dificuldade correta), sem a ajuda do adulto?
- Se conseguiu terminar o jogo sem ajudas externas?
- Se a dificuldade do jogo (fácil, médio, difícil) esteve adequada à criança?
- Se perdeu o interesse facilmente com algum jogo, não o acabando?
- Satisfação da criança de 0 a 10? (0 a 5 não satisfeita; 6-8 satisfeita; 9-10 muito satisfeita)

Testes com adultos

Para testar as funcionalidades da plataforma que não envolviam o input por parte de uma criança, como a página de administração e votar num trabalho foi dado a um adulto, que nunca utilizou a plataforma, um conjunto de tarefas e foi-lhe pedido que gravasse o seu ecrã ao realizá-las. Estes vídeos revelam se o utilizador completou as tarefas com sucesso e se seguiu o mesmo *flow* de alguém já com experiência na plataforma.

No final foi pedido que respondesse com o seu grau de satisfação numa escala de 0 a 10? (0 a 5 não satisfeito; 6-8 satisfeito; 9-10 muito satisfeito).

Lista de tarefas

As tarefas para testar a usabilidade fornecidas aos adultos e às crianças podem ser encontradas no anexo G.

Conclusões que se podem retirar destes testes

Segundo o artigo [67], com o feedback obtido, é possível retirar algumas conclusões acerca da usabilidade da plataforma como:

- Eficácia na utilização - se todas as tarefas foram completas.
- Satisfação na utilização - se o utilizador final se demonstrou satisfeito.
- Estudo das gravações - estudar o *flow* do utilizador ao realizar as tarefas.

Conclusões retiradas

- **Teste com Adulto**

A lista de tarefas foi a seguinte presente em G.1

- Eficácia de utilização - todas as tarefas foram completas (100%).
- Satisfação na utilização - 10.
- Estudo das gravações:

* Não foram observadas falhas nem comportamentos imprevistos no software. O *flow* seguido foi o mesmo de alguém com experiência.

- **Teste com Criança de 6 anos**

A lista de tarefas foi a seguinte presente em G.2, com dificuldade fácil devido à idade.

- Eficácia de utilização - todas as tarefas foram completas (100%).
- Satisfação na utilização - 9.
- Feedback do encarregado de educação e estudo das gravações:
 - * Conseguiu iniciar os jogos e terminá-los sem ajuda externa.
 - * A dificuldade esteve adequada e demonstrou interesse em finalizar o jogo.
 - * Problemas detetados descritos nas tabelas 7.5 e 7.6.

Contexto	Uma carta de um jogo é definida pela thumbnail, título, descrição, etc (pode ser visto na figura 6.10 um exemplo). No final, tem um botão para jogar e só este permite iniciar o jogo.
Problema	O jogo não é iniciado se carregar em outras partes da carta que não sejam o botão para jogar.
Causa	A criança intuitivamente carrega na thumbnail e na carta no geral, até ver que existe um botão a dizer jogar. Ou seja, faz vários cliques até conseguir acertar.
Solução	Colocar o botão de jogar mais saliente e maior, que faça com que se note mal a criança olhe para a carta. Colocar na carta um evento que permita iniciar o jogo se for clicada em qualquer parte.

Tabela 7.5: Problema de usabilidade detetado - 1

Contexto	Quando se carrega em jogar um jogo este surge no fundo página e é realizado um redirecionamento automático (autoscroll). Este processo foi definido com um delay de 1 segundo, para dar tempo do jogo já estar carregado quando o autoscroll acaba.
Problema	O redirecionamento foi várias vezes interrompido.
Causa	A criança, devido ao redirecionamento não ser imediato, demorar 1 segundo, fica a carregar no botão jogar e dá constantemente reset a esse delay.
Solução	Desativar as interações com a página nesse 1 segundo enquanto o autoscroll é realizado Dar um feedback claro que o jogo está a ser iniciado (colocar um fundo de página mais escuro, com um indicador de progresso [68]).

Tabela 7.6: Problema de usabilidade detetado - 2

7.2.2 Testes à Compatibilidade

Para testar a compatibilidade, foram realizados os seguintes testes:

- iOS e MacOS Safari - versão 14.1.1
 - Apenas foi detetado que o Fullscreen de um jogo não funcionava em telemóveis iPhone (iOS).
- MacOS Mozilla Firefox - versão 89.0.1

- Android e MacOS Chrome - versão 91

Em todos os testes realizados acima a plataforma funcionou como previsto, à exceção do Fullscreen para jogos em iPhone.

A responsividade de ecrã foi também testada nos dispositivos anteriormente mencionados e não foram detetadas falhas consideráveis que determinassem que certas funcionalidades não estavam disponíveis num determinado tamanho de ecrã.

O foco da aplicação foi para ecrãs de computadores e tablets. Não se descartaram os dos telemóveis, contudo existem jogos que ficam difíceis de se jogar em ecrãs pequenos, devido à precisão do clique que é necessário, por exemplo o jogo de Sopa de Letras, arrastar peças pequenas do Puzzle, entre outros. Apesar da opção Fullscreen conseguir otimizar todo o espaço do ecrã, estas dificuldades não se conseguiram colmatar.

7.2.3 Testes à Escalabilidade

Devido ao software ainda não ter migrado para o ambiente fora do DEI, não se testou este requisito. (Risco R4, tabela 3.6.)

7.2.4 Testes à Segurança

Nesta subsecção serão testados os acessos não autorizados a secções críticas da plataforma, nomeadamente à página de administração e base de dados.

Quando não existe um JWT válido na local storage do browser, sempre que o utilizador acede a um URL que corresponde à página de administração, é redirecionado para a página inicial da plataforma, não conseguindo entrar nesta página protegida.

Não é possível aceder à base de dados remotamente, apenas localmente, pois o contentor da base de dados é configurado no docker-compose exclusivamente para deixar o endereço externo 127.0.0.1 (localhost) aceder a este, que é o da máquina onde o docker está a correr. Logo, quem tiver o endereço da máquina onde o software está alojado, não consegue aceder à base de dados, pois não há nenhuma interface pública disponível para o fazer.

Capítulo 8

Conclusão

Neste capítulo será realizada uma retrospectiva do trabalho realizado, abordando aspetos como o que aprendi ao realizar este projeto, o que faltou fazer e as considerações finais.

8.1 Experiência Adquirida

Durante o primeiro semestre, o trabalho realizado focou-se essencialmente na investigação e na escrita. Todavia, foi possível aplicar uma grande parte de conhecimentos específicos do mestrado em Engenharia de Software, como a metodologia de desenvolvimento de software, planeamento, gestão de riscos, especificação de requisitos e desenho da arquitetura. Permitiu, portanto, rever os conceitos e aplicá-los num contexto real.

O segundo semestre foi ocupado pelo desenvolvimento, onde se realizou o maior processo de aprendizagem. Sem muita experiência prévia em desenvolvimento web, nomeadamente em tecnologias HTML e CSS, conseguiu-se desenvolver um sistema do zero até ao que se conhece hoje. Permitiu ganhar sensibilidade em áreas como *DevOps*, na parte de configuração dos ambientes de desenvolvimento, e *Full Stack Development*, na parte de toda a programação realizada, desde o servidor, *back-end*, até às interfaces de utilizador, *front-end*.

Foram também utilizadas bibliotecas externas, de modo a tornar mais rápido e ágil o desenvolvimento. Contudo, isto nem sempre acontece e com o uso de bibliotecas vem o estudo a fundo da sua documentação. Caso contrário, torna-se contra produtor, pois não se consegue extrair as funcionalidades ao máximo, ou até nem sequer usá-las como se pretende. Esta foi uma das lições aprendidas, estudar a documentação permite poupar tempo no futuro.

Este projeto vai agregar um grande peso no meu portefólio, que ainda se encontra numa fase embrionária, e mostrar que tenho conhecimentos e já alguma experiência nas tecnologias utilizadas, que são altamente valorizadas no mercado de trabalho atual, na área de Engenharia de Software.

8.2 Trabalho Futuro

A acrescentar ao trabalho que foi realizado, seria interessante os seguintes tópicos serem desenvolvidos futuramente:

- Trocar as imagens do livro infantil pelas definitivas.
- Ajustes no design em geral para ficar mais apelativo à idade pretendida.
- Testar as interfaces em mais dispositivos e assegurar que é sempre responsiva aos diferentes tamanhos de ecrã.
- Corrigir os defeitos de usabilidade já detetados nas interfaces.

- Aumentar o número de testes de usabilidade, principalmente com crianças de várias idades, para se conseguirem detetar mais defeitos e ver os que ocorrem com mais frequência. Isto vai permitir aumentar o grau de confiabilidade de modo a se poderem tirar conclusões convincentes.

8.3 Considerações Finais

Como em todos os projetos, existem dificuldades que têm de ser ultrapassadas da melhor maneira possível. Por esse motivo, definiram-se planos de mitigação para os riscos estabelecidos. As maiores dificuldades encontradas neste projeto prenderam-se com a equipa de design, que não permitiu que a plataforma tivesse um toque profissional, em termos estéticos, o que levou a um grande investimento de tempo na investigação de como deveriam parecer graficamente as interfaces para a idade pretendida, tanto dos jogos como da plataforma. Devido à inexperiência na área do design, não é garantido que estas estejam o mais parecidas ao que deveria ser o resultado final. Contudo, as funcionalidades acordadas estão implementadas, que foi a minha tarefa desde início e área de estudo.

No geral, o produto desenvolvido atendeu às expetativas iniciais, que consistiam em expor as funções do Centro, através de atividades interativas, como jogos e vídeos. Com esta plataforma, o Centro consegue colocar conteúdos dinâmicos, de forma rápida e simples, através da página de administração. Tem ainda à disponibilidade do administrador métricas que permitem adequar os jogos às diferentes idades, revelando se definiu corretamente os níveis de dificuldade. Tem ainda disponível um exemplo de um livro infantil virtual onde é possível virar as páginas realisticamente e um mecanismo de votação que permite alojar e votar em trabalhos.

É fornecido um tutorial de como instalar o software numa máquina para que este se consiga, sem a minha presença, migrar para outras máquinas depois de entregue e possa eventualmente ser melhorado no futuro. Espera-se que este software desenvolvido seja uma ferramenta utilizada pelo Centro de Excelência Jean Monnet da UC para expor as suas atividades aos mais novos, de maneira clara, simples e interativa.

Referências

- [1] Wikipedia. *Puzzle video game*. URL: https://en.wikipedia.org/wiki/Puzzle_video_game. (Acedido: 14-06-2021).
- [2] Centro de Excelência Jean Monnet UC. *Canal de Youtube do Centro*. URL: https://www.youtube.com/channel/UCQ32Dw0bU9oU_t6m01e1etQ/featured. (Acedido: 20-04-2021).
- [3] Oficina ID. *Website da empresa*. URL: <https://oficina-id.pt>. (Acedido: 20-04-2021).
- [4] europa.eu. *Goals and values of the EU*. (Acedido: 15-01-2021).
- [5] Veronika Stoffova. «The Importance of Didactic Computer Games in the Acquisition of New Knowledge». Em: *ICEEPSY 2016 : 7th International Conference on Education and Educational Psychology*. doi: 10.15405/epsbs.2016.11.70. Nov. de 2016.
- [6] Shalom M. Fisch. «Making Educational Computer Games “Educational”». Em: *IDC '05: Proceedings of the 2005 conference on Interaction design and children*. doi: 10.1145/1109540.1109548. Jun. de 2005.
- [7] Wikipedia. *Seductive details*. URL: https://en.wikipedia.org/wiki/Seductive%5C_details. (Acedido: 16-01-2021).
- [8] Garner R. Brown R. Sanders S. Menke D.J. «“Seductive details” and learning from text». Em: *K.A. Renninger, S. Hidi, A. Krapp (eds.), The role of interest in learning and development (pp. 239-254)*. 1992.
- [9] União Europeia. *Espaço de Aprendizagem*. URL: https://europa.eu/learning-corner/play-games_pt. (Acedido: 20-01-2021).
- [10] União Europeia. *Europa EU*. URL: https://europa.eu/european-union/index_pt. (Acedido: 20-01-2021).
- [11] National Geographic. *National Geographic Kids*. URL: <https://www.natgeokids.com/uk/kids-club/cool-kids/general-kids-club/national-geographic-kids-magazine/>. (Acedido: 23-06-2021).
- [12] National Geographic. *National Geographic Kids*. URL: <https://kids.nationalgeographic.com/games>. (Acedido: 20-01-2021).
- [13] Jigsawplanet. *Puzzles*. URL: <https://www.jigsawplanet.com/?rc=play&pid=232879b6b150&pieces=24>. (Acedido: 20-01-2021).
- [14] coloring-online. *Jogos de Colorir Online*. URL: <https://www.coloring-online.com/coloring-flags/asia/barein/index.php>. (Acedido: 20-01-2021).
- [15] monsterwordsearch. *Sopa de letras*. URL: <https://monsterwordsearch.com/geography-word-search/countries/european-union-word-search/>. (Acedido: 20-01-2021).
- [16] União Europeia - Learning corner. *Quiz*. URL: https://europa.eu/learning-corner/quiz/what/question_1_en. (Acedido: 20-01-2021).
- [17] União Europeia - Learning corner. *Jogo de Memória*. URL: https://europa.eu/learning-corner/match-the-flags_en. (Acedido: 20-01-2021).
- [18] Seterra. *Europe: Countries - Map Quiz Game*. URL: <https://online.seterra.com/en/vgp/3007>. (Acedido: 20-01-2021).
- [19] Wikipedia. *Scalable Vector Graphics*. URL: https://en.wikipedia.org/wiki/Scalable_Vector_Graphics. (Acedido: 20-01-2021).

-
- [20] Rachacuca. *Palavras Cruzadas*. URL: <https://rachacuca.com.br/palavras/palavras-cruzadas/54/>. (Acedido: 20-01-2021).
- [21] National Geographic. *National Geographic Kids*. URL: <https://kids.nationalgeographic.com/videos>. (Acedido: 20-01-2021).
- [22] FUNBRAIN. *Honey Moon: Dog Daze*. URL: <https://www.funbrain.com/books/honey-moon-dog-daze/page/16>. (Acedido: 20-01-2021).
- [23] Comissão Europeia. *Instagram da Comissão Europeia*. URL: https://www.instagram.com/comissao_europeia_pt/?hl=pt. (Acedido: 20-01-2021).
- [24] Wikipedia. *Software development process*. URL: https://en.wikipedia.org/wiki/Software_development_process. (Acedido: 2-03-2021).
- [25] Guru99. *Agile Methodology: What is Agile Software Development Model?* URL: <https://www.guru99.com/agile-scrum-extreme-testing.html>. (Acedido: 2-03-2021).
- [26] Wikipedia. *Scrum (software development)*. URL: [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)). (Acedido: 2-03-2021).
- [27] Scrum.org. *What is a Sprint Review?* URL: <https://www.scrum.org/resources/what-is-a-sprint-review>. (Acedido: 3-03-2021).
- [28] Scrum.org. *What is Sprint Planning?* URL: <https://www.scrum.org/resources/what-is-sprint-planning>. (Acedido: 3-03-2021).
- [29] material-ui. *React components for faster and easier web development. Build your own design system, or start with Material Design*. URL: <https://material-ui.com/components/buttons/>. (Acedido: 8-02-2021).
- [30] Jory MacKay. *7 Steps to Write a Risk Management Plan For Your Next Project (With Free Template!)* URL: <https://plan.io/blog/risk-management/>. (Acedido: 3-03-2021).
- [31] Steven Imke. *How to Develop a Risk Assessment Matrix For Business Success*. URL: <https://www.stevebizblog.com/how-to-develop-a-risk-matrix/>. (Acedido: 10-02-2021).
- [32] Geeks for Geeks. *Requirements Engineering Process*. URL: <https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/>. (Acedido: 5-11-2020).
- [33] Jeffrey Martin. *Verification Vs. Validation*. URL: <https://smartbear.com/blog/test-and-monitor/verification-and-validation-the-difference/>. (Acedido: 5-11-2020).
- [34] Usability.gov. *Usability Evaluation Basics*. URL: <https://www.usability.gov/what-and-why/usability-evaluation.html>. (Acedido: 4-03-2021).
- [35] Simon Brown. *The C4 model for visualising software architecture*. URL: <https://c4model.com/>. (Acedido: 5-12-2020).
- [36] Wikipedia. *Canvas element*. URL: https://en.wikipedia.org/wiki/Canvas_element. (Acedido: 5-12-2020).
- [37] Florian Rival (4ian). *List of game engines*. URL: https://en.wikipedia.org/wiki/List_of_game_engines. (Acedido: 5-12-2020).
- [38] Sagar Sharma. *Phaser - Desktop and Mobile HTML5 game framework*. URL: <https://phaser.io/>. (Acedido: 5-12-2020).
- [39] Florian Rival (4ian). *GDEVELOP - Game making for everyone*. URL: <https://gdevelop-app.com/>. (Acedido: 5-12-2020).
- [40] Florian Rival (4ian). *Unity - A plataforma líder para criação de conteúdo interativo em tempo real*. URL: <https://unity.com/pt>. (Acedido: 5-12-2020).
- [41] Sagar Sharma. *Phaser - Documentation*. URL: <https://photonstorm.github.io/phaser3-docs/>. (Acedido: 5-12-2020).
- [42] Sagar Sharma. *Phaser - Code examples*. URL: <https://labs.phaser.io/>. (Acedido: 5-12-2020).
- [43] Microsoft. *TypeScript*. URL: <https://en.wikipedia.org/wiki/TypeScript>. Acedido: 10-02-2021.

-
- [44] Wikipedia. *Type system*. URL: https://en.wikipedia.org/wiki/Type_system#STATIC. Acedido: 10-02-2021.
- [45] Sagar Sharma. *15 Top Front-End Technologies To Rule The Market in 2020 & Beyond*. URL: <https://www.credencys.com/blog/front-end-technologies/>. (Acedido: 5-12-2020).
- [46] Tomasz Grabski. *React JS Pros and Cons in 2020*. URL: <https://pagepro.co/blog/react-js-pros-and-cons-in-2020/>. (Acedido: 5-12-2020).
- [47] Chris Lojniewski. *React JS vs Angular – Which One To Choose in 2021?* URL: <https://pagepro.co/blog/react-vs-angular-comparison/>. (Acedido: 5-12-2020).
- [48] Jakub Dakowicz. *React vs Vue – A Quick Comparison (2020 Update)*. URL: <https://pagepro.co/blog/react-vs-vue-a-quick-comparison/>. (Acedido: 5-12-2020).
- [49] Isaac Z. Schlueter. *npm*. URL: <https://www.npmjs.com/>. (Acedido: 20-02-2020).
- [50] casesandberg. *React Color*. URL: <https://casesandberg.github.io/react-color/>. (Acedido: 28-03-2021).
- [51] SHANIKA WICKRAMASINGHE SAPNESH NAIK. *Most Compatible Frontend and Backend Framework Pairings*. URL: <https://blog.crowdbotics.com/most-compatible-frontend-backend-framework-pairings/>. (Acedido: 8-12-2020).
- [52] -. *What is the MERN Stack?* URL: <https://www.mongodb.com/mern-stack>. (Acedido: 28-03-2021).
- [53] Ronald Klazar. *Three Reasons To Use a NoSQL Document Store for Your Next Web Application*. URL: <https://medium.com/statuscode/three-reasons-to-use-a-nosql-document-store-for-your-next-web-application-6b9eabffc8d8>. (Acedido: 8-12-2020).
- [54] github:geocine. *Phaser 3 TypeScript Starter*. URL: <https://github.com/geocine/phaser3-rollup-typescript>. Acedido: 02-04-2021.
- [55] Solomon Hykes. *Docker*. URL: <https://www.docker.com/>. Acedido: 10-12-2020.
- [56] Solomon Hykes. *Docker Compose*. URL: <https://docs.docker.com/compose/>. Acedido: 10-12-2020.
- [57] Tom Preston-Werner. Chris Wanstrath. Scott Chacon. P J Hyett. *GitHub*. URL: <https://github.com/>. Acedido: 10-12-2020.
- [58] Wikipedia. *Bcrypt*. URL: <https://en.wikipedia.org/wiki/Bcrypt>. Acedido: 02-04-2021.
- [59] React. *React Hooks*. URL: <https://reactjs.org/docs/hooks-intro.html>. (Acedido: 18-06-2021).
- [60] React. *Optimizing Performance*. URL: <https://reactjs.org/docs/optimizing-performance.html>. (Acedido: 5-6-2021).
- [61] Wikipedia. *Flood Fill*. URL: https://en.wikipedia.org/wiki/Flood_fill. (Acedido: 5-2-2021).
- [62] William Malone. *Create a Paint Bucket Tool in HTML5 and JavaScript*. URL: <http://www.williammalone.com/articles/html5-canvas-javascript-paint-bucket-tool/>. (Acedido: 5-2-2021).
- [63] Wikipedia. *Peça Jigsaw*. URL: https://www.google.com/search?q=jigsaw+piece+wikipedia&tbm=isch&ved=2ahUKEwjugJWB9qDxAhU3AGMBHW2YBSkQ2-cCegQIABAA&oq=jigsaw+piece+wikipedia&gs_lcp=CgNpbWcQAZoECAAQEzoGCAAQCBAeUKItW0g2YPU3aABWAHAgAGeAYgB8AmSsclient=img&ei=cG7MYO6IA7eAjLsP7bCWyAI&bih=995&biw=1920&client=firefox-b-d#imgsrc=7Eh182s-0CcjoM. (Acedido: 5-2-2021).
- [64] Wikipedia. *Bézier curve*. URL: https://en.wikipedia.org/wiki/B%C3%A9zier_curve. (Acedido: 5-2-2021).
- [65] Wikipedia. *JSON Web Token*. URL: https://en.wikipedia.org/wiki/JSON_Web_Token. Acedido: 02-04-2021.
- [66] tutorialspoint. *Adhoc Testing*. URL: https://www.tutorialspoint.com/software_testing_dictionary/adhoc_testing.htm. (Acedido: 16-6-2021).
- [67] Justin Mifsud. *Usability Metrics: A Guide To Measuring The User Experience*. URL: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>. (Acedido: 16-6-2021).

- [68] Material UI. *Progress*. URL: <https://material-ui.com/components/progress/>. (Acedido: 16-6-2021).
- [69] MIT. *The MIT License (MIT)*. URL: <https://mit-license.org/>. (Acedido: 5-12-2020).
- [70] MongoDB Inc. *Server Side Public License*. URL: <https://www.mongodb.com/licensing/server-side-public-license>. (Acedido: 5-12-2020).

Anexos

Anexo A

Casos de Uso

ID	UC-1
Caso de Uso	Este caso de uso começa quando o utilizador quer ver um jogar um jogo e termina quando o utilizador acaba de jogá-lo
Ator Primário	Visitante
Precondições	O utilizador tem a aplicação web aberta
Triggers	O utilizador carrega na opção “Jogos” da barra presente no canto superior
Flow Básico	<ol style="list-style-type: none">1. O utilizador é redirecionado para a página de jogos2. O sistema apresenta os vários jogos, em formato de lista, separados por tipos3. O utilizador carrega numa categoria de jogos4. O sistema mostra em cartas todos os jogos disponíveis daquele tipo5. O utilizador carrega num jogo6. O sistema reproduz o jogo automaticamente e redireciona o utilizador para a janela de jogo7. O utilizador joga o jogo
Pós Condição	Nenhuma

Tabela A.1: Caso de uso **UC-1**

ID	UC-3 - Ler o livro infantil
Caso de Uso	Este caso de uso começa quando o utilizador quer ler o livro infantil e termina quando o utilizador acaba de lê-lo.
Ator Primário	Visitante
Precondições	O utilizador tem a aplicação web aberta
Triggers	O utilizador carrega na opção “Livro” da barra presente no canto superior
Flow Básico	<ol style="list-style-type: none">1. O utilizador é redirecionado para a página do livro digital2. O utilizador lê o livro "arrastando" as páginas na direção que deseja3. O sistema mostra a página atual no qual o utilizador se encontra e termina quando página atual = página final e permite que a página seja narrada na opção de reproduzir som
Pós Condição	Nenhuma

Tabela A.2: Caso de uso **UC-3**

ID	UC-4 - Votar num trabalho
Caso de Uso	Este caso de uso começa quando o utilizador quer votar num trabalho de uma votação que está a decorrer. Este caso de uso termina quando o utilizador submete a sua votação
Ator Primário	Visitante
Precondições	O utilizador tem a aplicação web aberta
Triggers	O utilizador carrega no ícone de "Votar"
Flow Básico	<ol style="list-style-type: none"> 1. O sistema redireciona o utilizador para a página de votação 2. O sistema mostra todas as categorias de votações que estão a decorrer 3. O utilizador escolhe dentro das possibilidades a categoria que pretende para ver os seus trabalhos 4. O sistema mostra ao utilizador os trabalhos relativos à categoria anteriormente selecionada 5. O utilizador escolhe o trabalho em que quer votar 6. O sistema abre uma janela com mais informações sobre o trabalho e o formulário para realizar a votação 7. O utilizador preenche o formulário 8. O utilizador submete o formulário
Flow Alternativo 1	<p>Do ponto 1. até ao 6. é igual</p> <p>8a. Depois de ver mais detalhes sobre o trabalho o utilizador decide votar noutro trabalho</p> <p>9a. O utilizador fecha a janela</p> <p>10a. O utilizador seleciona outro trabalho</p> <p>Volta ao ponto 7</p>
Flow Alternativo 2	<p>Do ponto 1. até ao 6. é igual</p> <p>8a. Depois de ver os trabalhos daquela categoria decide que quer ver os de outras. Para isso, carrega no botão de "Voltar"</p> <p>9a. O sistema redireciona o utilizador para o ponto 2.</p>
Flow de exceção	<p>Do ponto 1. até ao 7. é igual</p> <p>8b. O utilizador preenche o formulário com inputs inválidos</p> <p>9b. O sistema avisa o utilizador os campos que foram preenchidos incorretamente no formulário</p>
Pós Condição	<p>A votação é registada na base de dados</p> <p>Na página de votação, o utilizador é notificado que a sua votação foi realizada com sucesso</p> <p>O número de votos do trabalho incrementa (+1)</p>

Tabela A.3: Caso de uso **UC-4**

ID	UC-5 - Realizar login
Caso de Uso	Este caso de uso começa quando o utilizador quer entrar como administrador na plataforma Este caso de uso termina quando o utilizador é redirecionado para a página de administração
Ator Primário	Visitante
Precondições	O utilizador tem a aplicação web aberta
Triggers	O utilizador coloca à frente do endereço URL /login do endereço da página inicial da plataforma
Flow Básico	<ol style="list-style-type: none"> 1. sistema redireciona o utilizador para a página de login 2. O sistema apresenta um formulário com username e password para o utilizador preencher 3. O utilizador preenche e submete o formulário 4. O sistema autentica e autoriza o utilizador 5. O sistema redireciona o utilizador para a página de administração
Flow de exceção	<p>Do ponto 1. até ao 4. é igual</p> <p>5a. O sistema não autentica os dados inseridos e não autoriza o utilizador</p> <p>6a. O sistema apresenta uma mensagem a dizer que o username ou password estão incorreto(s)</p> <p>7a O utilizador fica na mesma página e vê a mensagem de insucesso.</p>
Pós Condição	Um JWT é colocado no local storage do browser de modo a autorizar o utilizador

Tabela A.4: Caso de uso **UC-5**

ID	UC-6 - Realizar logout
Caso de Uso	Este caso de uso começa quando o utilizador quer sair como administrador da plataforma Este caso de uso termina quando o utilizador é redirecionado para a página inicial da plataforma
Ator Primário	Admin
Precondições	O utilizador tem a aplicação web aberta e está autorizado
Triggers	O utilizador carrega no botão “Admin” para aceder à pagina de administração
Flow Básico	<ol style="list-style-type: none"> 1. O sistema redireciona o utilizador para a página de administração 2. O utilizador carrega no botão “LOGOUT” 3. O sistema redireciona o utilizador para a página inicial da plataforma
Pós Condição	JWT é retirado no local storage do browser de modo a não autorizar ações futuras do utilizador na página de administração.

Tabela A.5: Caso de uso **UC-6**

ID	UC-7 - Gerir conteúdo da plataforma
Caso de Uso	Este caso de uso começa quando o administrador quer alterar um determinado conteúdo da página dos disponíveis e termina quando a ação desejada for satisfeita.
Ator Primário	Admin
Precondições	O administrador tem a aplicação web aberta
Triggers	O utilizador carrega no botão "CONTEÚDO" e seleciona o que pretende modificar na lista disponível "Gestão de conteúdo da página".
Flow Básico	<ol style="list-style-type: none"> 1. O utilizador carrega no botão "Admin" para aceder à página de administração 2. O sistema redireciona o utilizador para a página de administração 3. O utilizador escolhe o tipo de conteúdo que pretende alterar (Jogos, Vídeos, Votação) 4. O sistema apresenta o tipo de conteúdo numa lista, com ações de criar, editar e eliminar 5. O utilizador pretende criar/editar conteúdo 6. O sistema redireciona para uma página onde tem um formulário para o utilizador preencher de modo a criar/editar novo conteúdo. Se selecionou editar os campos são preenchidos pelo conteúdo pre-existente na base de dados 7. O utilizador preenche e submete o formulário 8. O sistema valida o formulário, redireciona o utilizador para a página de administração (ponto 2 do flow)
Flow Alternativo 1	<p>Do ponto 1. até ao 4. é igual</p> <p>5a O utilizador pretende eliminar um determinado conteúdo e carrega no botão de eliminar</p> <p>6a O sistema retira o conteúdo que foi eliminado da tabela</p>
Flow Alternativo 2	<p>Do ponto 1. até ao 6. é igual</p> <p>8a. O administrador quer cancelar a alteração e carrega no botão "Voltar"</p> <p>9a. O sistema redireciona o utilizador para o ponto 3.</p>
Flow de exceção	<p>Do ponto 1. até ao 7. é igual</p> <p>O sistema não valida o formulário (porque algo foi preenchido incorretamente) e apresenta a mensagem de erro no(s) campo(s) do formulário</p>
Pós Condição	Criar/edita/elimina o conteúdo na base de dados

Tabela A.6: Caso de uso **UC-7**

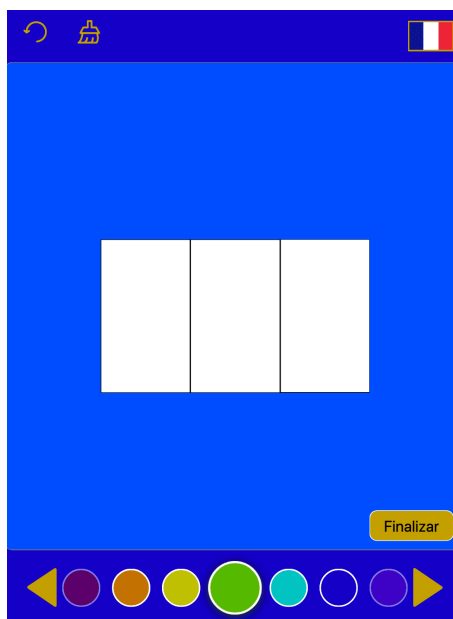
ID	UC-8 - Alterar credenciais de acesso
Caso de Uso	Este caso de uso começa quando o utilizador quer alterar as credenciais da conta de administrador da plataforma Este caso de uso termina quando o utilizador recebe feedback do sistema em relação ao estado da alteração
Ator Primário	Admin
Precondições	O utilizador tem a aplicação web aberta e está autorizado
Triggers	O utilizador carrega na opção “Livro” da barra presente no canto superior
Flow Básico	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Admin” para aceder à pagina de administração 2. O sistema redireciona o utilizador para a página de administração 3. O utilizador carrega no botão “PERFIL” 4. O sistema apresenta um formulário no qual o utilizador tem de preencher o novo username, password antiga, nova password e confirmar a nova password 5. O utilizador preenche estes campos do formulário e submete 6. O sistema notifica que os dados foram alterados com sucesso
Flow Alternativo 1	<p>Do ponto 1. até ao 5. é igual</p> <p>6a. O sistema não valida o formulário (porque algo foi preenchido incorretamente) e apresenta a mensagem de erro no(s) campo(s) do formulário</p>
Pós Condição	Edita as credenciais de acesso na base de dados e substitui o JWT antigo pelo novo.

Tabela A.7: Caso de uso **UC-8**

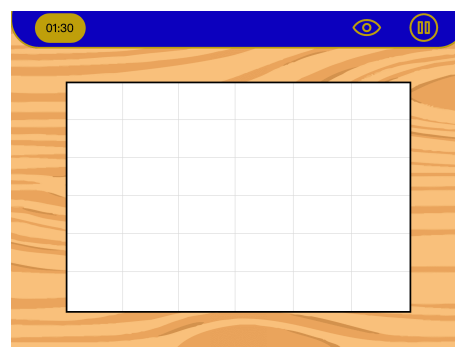
Anexo B

Protótipos de ecrã de Baixa Fidelidade

B.1 Protótipos Jogos



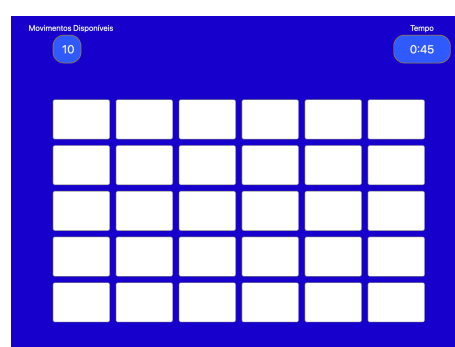
(a) Jogo de Colorir



(b) Jogo Puzzle

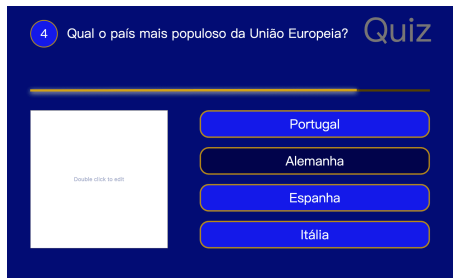


(c) Jogo de Sopa de Letras

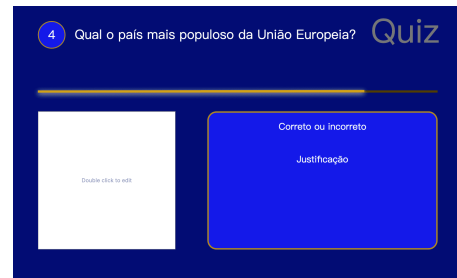


(d) Jogo de Memória

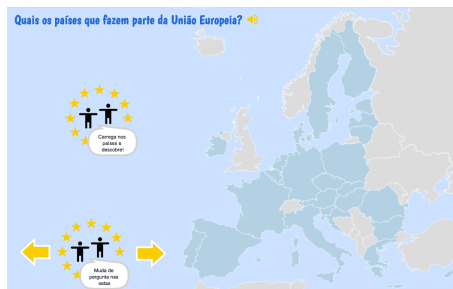
Figura B.1: Protótipos dos jogos



(a) Jogo de Quiz (com tempo)



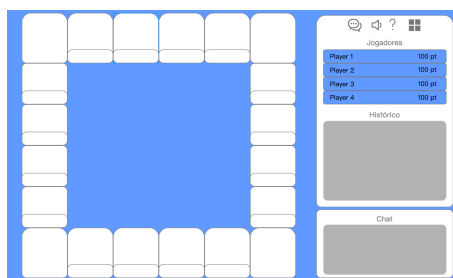
(b) Justificação a uma pergunta do jogo Quiz



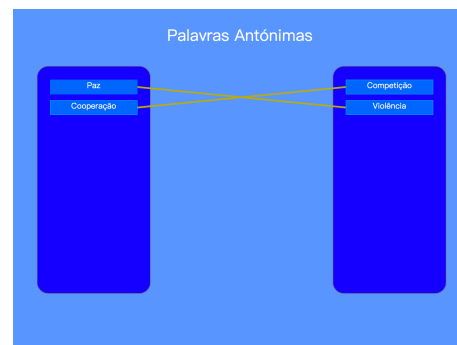
(c) Jogo de Mapas Interativos



(d) Jogo de Palavras Cruzadas



(e) Jogo de Tabuleiro



(f) Jogo de Palavras Antónimas

Figura B.2: Protótipos dos jogos

B.2 Protótipos Website

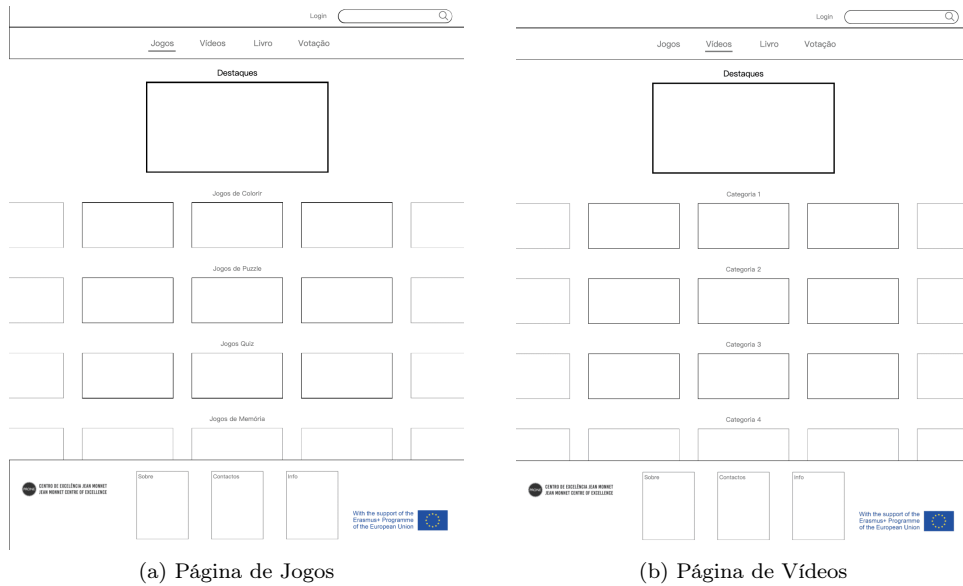


Figura B.3: Protótipos do Website

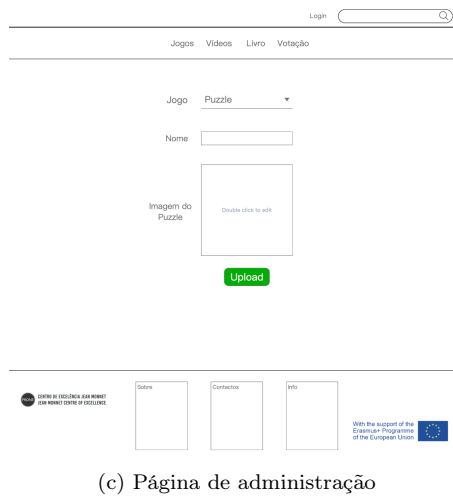


Figura B.4: Protótipos do Website

Anexo C

Licenças das Tecnologias Utilizadas

- Licença MIT [69]
- Licença SSPL [70]

Tecnologia	Versão	Licença
Phaser	3.55.2	MIT
React	17.0.1	MIT
material-ui	0.20.2	MIT
styled-components	5.2.1	MIT
react-flip-page	1.6.4	MIT
react-audio-player	0.14.0	MIT
axios	0.21.1	MIT
react-query	3.13.5	MIT
react-hook-form	7.4.0	MIT
yup	0.32.9	MIT
react-router-dom	5.2.0	MIT
react-full-screen	1.0.2	MIT
react-simple-maps	2.3.0	MIT
@jaredreisinger/react-crossword	2.3.1	MIT
react-scroll	1.8.2	MIT
react-color	2.19.3	MIT
react-player	2.9.0	MIT

Tabela C.1: Tabela de Licenças das Tecnologias Utilizadas (front-end)

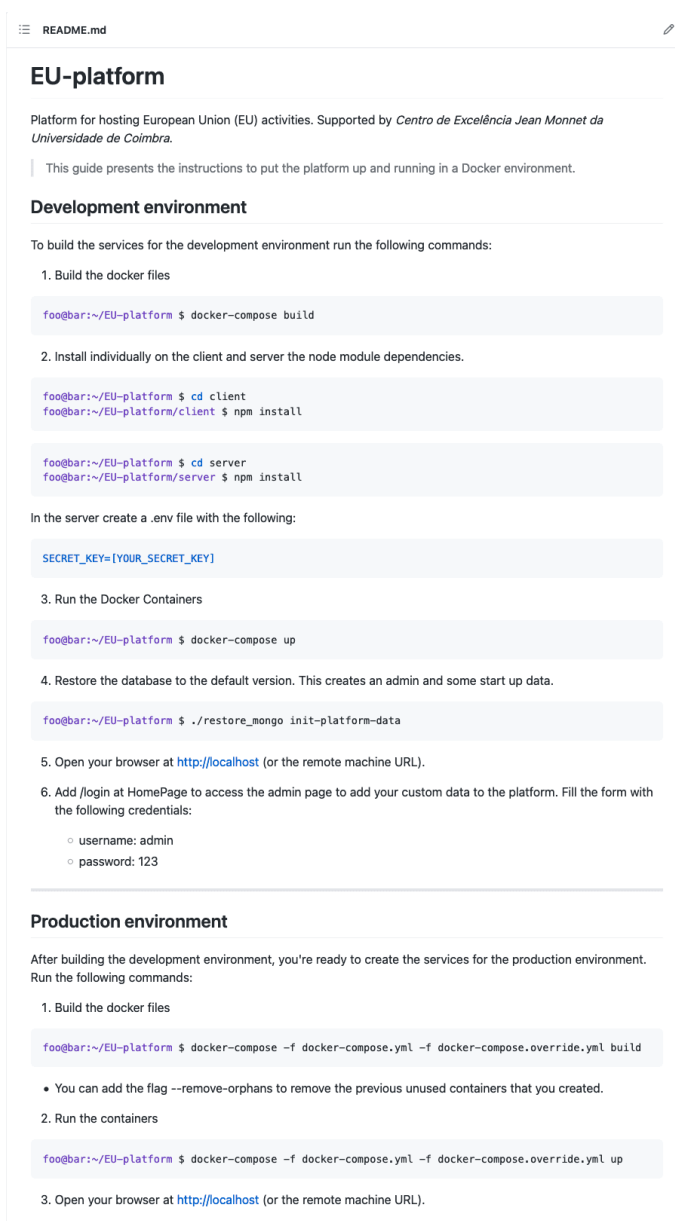
Tecnologia	Versão	Licença
Node	15	MIT
jsonwebtoken	8.5.1	MIT
passport	0.4.1	MIT
passport-local	1.0.0	MIT
passport-jwt	4.0.0	MIT
bcryptjs	2.4.3	MIT
express	4.17.1	MIT
mongoose	4.2	SSPL
mongoose	5.11.13	MIT
multer	1.4.2	MIT
uuid	8.3.2	MIT
dotenv	8.2.0	MIT
nodemon	2.0.7	MIT

Tabela C.2: Tabela de Licenças das Tecnologias Utilizadas (back-end)

Anexo D

Tutorial de Instalação do Software

Ficheiro *README.md* disponível na pasta do projeto.



README.md

EU-platform

Platform for hosting European Union (EU) activities. Supported by *Centro de Excelência Jean Monnet da Universidade de Coimbra*.

This guide presents the instructions to put the platform up and running in a Docker environment.

Development environment

To build the services for the development environment run the following commands:

1. Build the docker files

```
foo@bar:~/EU-Platform $ docker-compose build
```

2. Install individually on the client and server the node module dependencies.

```
foo@bar:~/EU-Platform $ cd client
foo@bar:~/EU-Platform/client $ npm install
```

```
foo@bar:~/EU-Platform $ cd server
foo@bar:~/EU-Platform/server $ npm install
```

In the server create a `.env` file with the following:

```
SECRET_KEY=[YOUR_SECRET_KEY]
```

3. Run the Docker Containers

```
foo@bar:~/EU-Platform $ docker-compose up
```

4. Restore the database to the default version. This creates an admin and some start up data.

```
foo@bar:~/EU-Platform $ ./restore_mongo_init-platform-data
```

5. Open your browser at <http://localhost> (or the remote machine URL).
6. Add `/login` at `HomePage` to access the admin page to add your custom data to the platform. Fill the form with the following credentials:
 - username: admin
 - password: 123

Production environment

After building the development environment, you're ready to create the services for the production environment. Run the following commands:

1. Build the docker files

```
foo@bar:~/EU-Platform $ docker-compose -f docker-compose.yml -f docker-compose.override.yml build
```

- You can add the flag `--remove-orphans` to remove the previous unused containers that you created.

2. Run the containers

```
foo@bar:~/EU-Platform $ docker-compose -f docker-compose.yml -f docker-compose.override.yml up
```

3. Open your browser at <http://localhost> (or the remote machine URL).

Figura D.1: Tutorial de configuração e instalação da plataforma

Anexo E

Requisitos Implementados

ID	Descrição	Prioridade	Implementado
FR-1	A plataforma deverá conter uma barra de funcionalidades com botões que redireciona para os Jogos, Vídeos, Livro e Votações	Must have	SIM
FR-2	A plataforma deverá conter uma secção no footer com informações relevantes sobre a instituição que financia o projeto	Must have	SIM
FR-3	A plataforma deverá conter uma página de jogos	Must have	SIM
FR-3.1	Os jogos, quando completados com sucesso, devem parabenizar o jogador. Quando completados com insucesso, devem perguntar se quer jogar novamente	Must have	SIM (*)
FR-3.2	A página de jogos deverá conter jogos de Colorir	Must have	SIM
FR-3.3	A página de jogos deverá conter Puzzle's	Must have	SIM
FR-3.4	A página de jogos deverá conter Quiz's	Must have	SIM
FR-3.5	A página de jogos deverá conter jogos de Sopa de Letras	Must have	SIM
FR-3.6	A página de jogos deverá conter jogos de Palavras Cruzadas	Should have	SIM
FR-3.7	A página de jogos deverá conter jogos de Memória	Must have	SIM
FR-3.8	A página de jogos deverá conter jogos de Mapas Interativos	Should have	SIM
FR-3.9	A página de jogos deverá um jogo de Tabuleiro	Won't have	NÃO
FR-3.10	A plataforma deverá conter Realidade Aumentada para as perguntas do Jogo de Tabuleiro	Won't have	NÃO
FR-3.11	A página de jogos deverá jogos de Palavras Antónimas	Should have	NÃO
FR-4	A plataforma deverá conter uma página de vídeos	Should have	SIM

* menos no jogo de Colorir, Palavras Cruzadas e Mapas Interativos

ID	Descrição	Prioridade	Implementado
FR-4.1	A página de vídeos deverá integrar vídeos do Youtube do canal Centro de Excelência Jean Monnet UC, bem como outros canais	Should have	SIM
FR-4.2	Os vídeos devem ser divididos em categorias	Should have	SIM
FR-5	A plataforma deverá conter uma página com o livro infantil	Must have	SIM
FR-5.1	A página deverá integrar as imagens do livro infantil fornecidas pela equipa de design [3]	Must have	NÃO
FR-5.2	O utilizador deverá virar as páginas do livro realisticamente	Must have	SIM
FR-5.3	Cada página deverá ter um player de áudio para poder ser narrada	Should have	SIM
FR-5.4	Deverá informar o utilizador da página em que se encontra	Should have	SIM
FR-6	A plataforma deverá conter uma página de votação	Should have	SIM
FR-6.1	A página deverá listar as diferentes categorias de votações	Should have	SIM
FR-6.2	Cada trabalho deverá conter o seu título, a fotografia, uma breve descrição e o seu autor. Também, o número de votos atuais e o botão para votar	Should have	SIM
FR-6.3	Para votar no trabalho, o utilizador deverá inserir o seu nome e email num formulário	Should have	SIM
FR-7	A plataforma deverá ter uma página de autenticação do administrador	Must have	SIM
FR-8	A página de administrador deverá ter uma dashboard com as seguintes ações: - alterar as credenciais de acesso da conta de administrador - realizar logout - gestão do conteúdo da página - estatísticas dos jogos	Must have	SIM
FR-8.1	A secção de gestão de conteúdo da página deverá permitir as seguintes ações: - CRUD todos os jogos existentes - CRUD categorias de vídeo e vídeos - CRUD categorias de votação e trabalhos	Must have	SIM
FR-8.2	A secção de estatísticas deverá fornecer informações como: - jogos abertos - jogos finalizados - vitórias - respostas certas - percentagens relevantes - tempo de resposta	Must have	SIM

Tabela E.1: Requisitos Funcionais Implementados

Anexo F

Testes Funcionais

F.1 Funcionalidades Gerais

ID	FT-1-01
ID Requisito	FR-1
Cenário	Botões presentes na barra de funcionalidades redirecionam para as respectivas páginas
PASS/FAIL	PASS

ID	FT-1-02
ID Requisito	FR-2
Cenário	Footer com as informações à cerca do Centro de Excelência está presente em todas as páginas da plataforma
PASS/FAIL	PASS

F.2 Página de Jogos

ID	FT-2-01
ID Requisito	FR-3, FR-3.1, FR-3.2, FR-3.3, FR-3.4, FR-3.5, FR-3.6, FR-3.7, FR-3.8.
Cenário	Existe uma página de jogos, todos iniciam e são jogáveis
PASS/FAIL	PASS

ID	FT-2-02
ID Requisito	FR-3.1
Cenário	Quando um jogo acaba deve ser mostrada uma cena final a parabenizar o jogador, se completado com sucesso. Caso contrário, a cena deve perguntar se quer jogar novamente
PASS/FAIL	Colorir - FAIL (motivo em Limitações na fig 6.2) Puzzle - PASS Quiz - PASS Sopa de Letras - PASS Memória - PASS Mapas Interativos - FAIL (motivo em Limitações na fig 6.7) Palavras Cruzadas - FAIL (motivo em Limitações na fig 6.8)

F.3 Página de Vídeos

ID	FT-3-01
ID Requisito	FR-4, FR-4.1, FR-4.2
Cenário	Existe uma página de vídeos, divididos em categorias e reproduzem-se automaticamente quando clicados. Estes vídeos são provenientes do Youtube.
PASS/FAIL	PASS

F.4 Página do Livro Infantil

ID	FT-4-01
ID Requisito	FR-5, FR-5.2, FR-5.3, FR-5.4
Cenário	Existe uma página do livro infantil, que vira as páginas realisticamente, reproduz áudio e informa o utilizador na página em que se encontra.
PASS/FAIL	PASS

ID	FT-4-02
ID Requisito	FR-5.1
Cenário	O livro infantil contém as páginas fornecidas pela equipa de design
PASS/FAIL	FAIL (motivo no Risco 2, tabela 3.4)

F.5 Página de Votação

ID	FT-5-01
ID Requisito	FR-6, FR-6.1
Cenário	Existe uma página de votação, com diferentes categorias de votação
PASS/FAIL	PASS

ID	FT-5-02
ID Requisito	FR-6.2
Cenário	Cada categoria tem trabalhos associados. Um trabalho é composto por título, fotografia, breve descrição, o seu autor, número de votos e botão que permite votar nele
PASS/FAIL	PASS

ID	FT-5-03
ID Requisito	FR-6.3
Cenário	Um utilizador pode votar num trabalho inserindo o seu nome e email
PASS/FAIL	PASS

Testes Ad Hoc à página

ID	AH-5-01
Cenário	Votar em mais que um trabalho diferente com o mesmo email
PASS/FAIL	PASS

ID	AH-5-02
Cenário	Não é possível votar em mais que um trabalho
PASS/FAIL	PASS

F.6 Página de Administração

ID	FT-6-01
ID Requisito	FR-7
Cenário	Existe uma página de autenticação do administrador
PASS/FAIL	PASS

ID	FT-6-02
ID Requisito	FR-8
Cenário	Existe uma página de administração que permite realizar as funcionalidades expectáveis (alterar perfil, realizar logout, ver estatísticas e gerir o conteúdo)
PASS/FAIL	Alterar Perfil - PASS Realizar Logout - PASS Gerir Conteúdo da Página - PASS Ver estatísticas de jogos - PASS

ID	FT-6-03
ID Requisito	FR-8.1
Cenário	A secção de gestão de conteúdo permite realizar as funcionalidades expectáveis (CRUD à página de jogos, vídeos e votação)
PASS/FAIL	CRUD jogos - PASS CRUD categorias de vídeos - PASS CRUD vídeos - PASS CRUD categorias de votação - PASS CRUD trabalhos - PASS

ID	FT-6-04
ID Requisito	FR-8.2
Cenário	São apresentadas as estatísticas para os jogos e as métricas expectáveis
PASS/FAIL	Colorir - FAIL (motivo em Limitações na fig 6.2) Puzzle - PASS Quiz - PASS Sopa de Letras - PASS Memória - PASS Mapas Interativos - FAIL (não implementado) Palavras Cruzadas - FAIL (não implementado)

Testes Ad Hoc à página

ID	AH-6-01
Cenário	O administrador não se consegue autenticar se as suas credenciais de acesso forem inválidas
PASS/FAIL	PASS

ID	AH-6-02
Cenário	O administrador não consegue alterar os seus dados se inserir a nova password diferente da confirmação da nova password
PASS/FAIL	PASS

ID	AH-6-03
Cenário	O administrador depois de realizar logout não está com permissões de administrador, isto é, não consegue voltar a aceder à página de administração
PASS/FAIL	PASS

ID	AH-6-04
Cenário	Eliminar uma categoria de vídeo elimina todos os vídeos
PASS/FAIL	PASS

ID	AH-6-05
Cenário	Eliminar uma categoria de votação elimina todos os trabalhos
PASS/FAIL	PASS

ID	AH-6-06
Cenário	As imagens (thumbnails) ou qualquer tipo de imagem associada são eliminadas automaticamente do servidor quando é eliminado o objecto em questão. Os objetos podem ser: jogos, categorias de vídeos, categorias de votação e trabalhos.
PASS/FAIL	PASS (em todos os objetos mencionados)

ID	AH-6-07
Cenário	As imagens (thumbnails) ou qualquer tipo de imagem associada são substituídas automaticamente pela mais recente (a antiga é eliminada do servidor) quando é alterado o objecto em questão. Os objetos podem ser: jogos, categorias de vídeos, categorias de votação e trabalhos.
PASS/FAIL	PASS (em todos os objetos mencionados)

ID	AH-6-08
Cenário	Eliminar uma categoria de votação elimina todos os trabalhos
PASS/FAIL	PASS

ID	AH-6-09
Cenário	Os formulários não podem ser submetidos com campos obrigatórios em branco
PASS/FAIL	PASS

ID	AH-6-10
Cenário	Os formulários apresentam o campo que foi preenchido incorretamente
PASS/FAIL	PASS

Anexo G

Tarefas fornecidas para testar a Usabilidade

G.1 Tarefas para Adultos

1. Adicionar um Puzzle com as seguintes características:

- (a) Título: Puzzle teste
- (b) Descrição: Teste de usabilidade
- (c) Dificuldade: Média
- (d) Thumbnail: Bandeira de Portugal
- (e) Tamanho de peças: Médias
- (f) Tempo para terminar o jogo: 40 segundos
- (g) Opções ativas:
 - background com imagem do Puzzle
 - mover peças livremente pelo tabuleiro

2. Editar as seguintes características no jogo de Colorir:

- Eliminar as cores todas que tem
- Adicionar as novas Cores:
 - Vermelho
 - Verde
 - Azul escuro
 - Amarelo
 - Azul claro
 - Roxo
 - Rosa
 - Cinzento

3. Editar as seguintes características da Sopa de Letras:

- (a) Eliminar as palavras todas que tem
- (b) Adicionar as palavras:
 - Teste
 - Para
 - Ver
 - Usabilidade

- (c) Número de células Horizontais: 15
 - (d) Número de células Verticais: 15
 - (e) Tempo para terminar o jogo: 10 segundos
4. Eliminar o Puzzle anteriormente criado
 5. Editar o jogo Palavras Cruzadas para conter a informação presente na imagem, figura G.1:

	1	2	3	4	5	6	7	8	9	10	11	12	13				
1											L						
2											I			Horizontal	(Linha, Coluna)	Vertical	(Linha, Coluna)
3											S			5. Capital da Eslováquia	(6, 4)	1. Capital de Portugal	(1, 11)
4	A			B							B			7. Capital da Espanha	(8, 1)	2. Capital da Grécia	(4, 2)
5	T		E					E			O			8. Capital da Bulgária	(13, 8)	3. Capital da Alemanha	(4, 5)
6	E	B	R	A	T	I	S	L	A	V	A					4. Capital da Suécia	(5, 9)
7	N		L					T			A					6. Capital da Polónia	(6, 12)
8	M	A	D	R	I	D		O			R						
9	S			N				C			S						
10								O			O						
11								L			V						
12								M			I						
13							S	O	F	I	A						

Figura G.1: Exemplo de como fazer um jogo de Palavras Cruzadas

6. Adicionar uma categoria de vídeo:
 - (a) Título: Categoria teste
 - (b) Descrição: Teste de usabilidade
 - (c) Imagem da Categoria: Bandeira de Portugal
7. Adicionar um vídeo já existente à categoria anteriormente criada
 - (a) Categoria de vídeos: Categoria teste
 - (b) Título: Vídeo teste
 - (c) Descrição: Teste de usabilidade
 - (d) Imagem da Categoria: link youtube (<https://www.youtube.com/watch?v=4VCYHTGjr-U>)
8. Eliminar a categoria de vídeos anteriormente criada
9. Verificar se o vídeo adicionado foi também eliminado
10. Criar uma categoria de votação com as seguintes características:
 - (a) Título: Categoria de votação teste
 - (b) Descrição: Teste de usabilidade
 - (c) Imagem da Votação: Bandeira de Portugal
11. Adicionar um trabalho à votação anteriormente criada com as seguintes características:
 - (a) Título: Trabalho teste 1
 - (b) Descrição: Teste de usabilidade
 - (c) Autor: Tester
 - (d) Imagem do trabalho: Bandeira de Croácia
12. Adicionar um trabalho à votação anteriormente criada com as seguintes características:
 - (a) Título: Trabalho teste 2
 - (b) Descrição: Teste de usabilidade
 - (c) Autor: Tester
 - (d) Imagem do trabalho: Bandeira de Espanha
13. Realizar uma votação no trabalho 2:
 - (a) Nome: Tester
 - (b) Email: tester@mail.com
14. Encerrar a votação anteriormente criada
15. Ver vencedor

G.2 Tarefas para Crianças

Pretendido do encarregado de educação:

1. Escolher a dificuldade do jogo consoante a idade da criança:
 - fácil - 6 anos
 - médio - 7-8 anos
 - difícil - 9 anos
 - *(Não é obrigatório seguir, pode aumentar ou baixar a dificuldade consoante as capacidades da criança)*
2. Gravar cada tarefa para tentar estudar o comportamento da criança ao interagir com o ecrã (vídeo não necessita de ser muito longo)
3. Em cada tarefa que a criança realizou escrever um pequeno comentário com sugestões, críticas, reação da criança, etc.
4. No final desenvolver num pequeno comentário em cada um dos seguintes tópicos:
 - Se conseguiu iniciar o jogo pedido (tipo de jogo e a dificuldade correta), sem a ajuda do adulto?
 - Se conseguiu reproduzir um vídeo?
 - Se conseguiu terminar o jogo sem ajudas externas?
 - Se a dificuldade do jogo (fácil, médio, difícil) esteve adequada à criança?
 - Se perdeu o interesse facilmente com algum jogo, não o acabando?
 - Satisfação da criança de 0 a 10? (0 a 5 não satisfeita; 6-8 satisfeita; 9-10 muito satisfeita)
 - Tarefas que não conseguiu fazer?

Tarefas da criança:

1. Jogar uma Sopa de Letras
2. Jogar um Puzzle
3. Jogar um jogo de Memória
4. Jogar um jogo Palavras Cruzadas
5. Reproduzir um vídeo qualquer

Este trabalho de investigação foi desenvolvido no âmbito do Centro de Excelência Jean Monnet PRONE - *Peace Relations, Ontologies and Narratives in Europe: EU and its Eastern Neighbours*, 611269-EPP-1-2019-1-PT-EPPJMO-CoE, Universidade de Coimbra. O apoio da Comissão Europeia à produção desta publicação não constitui um aval do seu conteúdo, que reflete unicamente o ponto de vista do autor, e a Comissão não pode ser considerada responsável por eventuais utilizações que possam ser feitas com as informações nela contidas.



CENTRO DE EXCELÊNCIA JEAN MONNET
JEAN MONNET CENTRE OF EXCELLENCE



With the support of the
Erasmus+ Programme
of the European Union

