

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



EdScape - Sistema Web de Condecorações e Mini Jogos a Utilizar em Escape Room Educativos

Celina Mo Zheng

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Orientador: Prof. Armando Jorge Miranda de Sousa

Co-orientador: Prof. António Fernando Vasconcelos Cunha Castro Coelho

1 de Julho 2022

Resumo

O avanço na tecnologia promoveu o aparecimento de novos métodos de ensino, tal como os Escape Rooms educativos que são atividades lúdicas usadas como estratégia de aprendizagem ativa para aumentar a motivação dos alunos.

A *framework* “EdScape” permite criar este tipo de jogos (jogos de aventura e escape rooms educativos) com flexibilidade para atualizações frequentes por parte de um educador, inclusivamente fora do Unity3D.

Nesta dissertação são implementados: (i) 3 mini jogos para integração na *framework* do projeto “EdScape”; (ii) sistema *web* de condecorações, onde os estudantes poderão ter acesso às medalhas conquistadas, baseado na norma “Open Badges”; (iii) o sistema de registo em Base de Dados dos eventos do jogo.

Os mini jogos desenvolvidos incluem puzzles, correspondência de imagens a etiquetas e conjuntos de resposta fechada (perguntas e alternativas de respostas).

O sistema *web* de condecorações é genérico e permite ver as medalhas adquiridas. O código de visualização das medalhas pode ser copiado para integrar em sites próprios.

Também é implementada a funcionalidade de autenticação tanto para a *framework* como para o sistema *web*. Este tipo de atividades torna a aprendizagem mais divertida o que levará o aumento do interesse dos alunos pela matéria. As ferramentas utilizadas para a construção da *framework* são o C# dentro do Unity3D e a interface *web* para os Open Badges utiliza CSS, HTML, JavaScript e PHP.

O sistema desenvolvido nesta dissertação mostra ainda as informações recolhidas durante os jogos jogados e utiliza apenas recursos pré-existentes na FEUP.

O trabalho foi dividido nas seguintes etapas, primeiro identificar os requisitos necessários para o desenvolvimento do projeto. Em segundo lugar desenhar os *mockups*. Em terceiro lugar implementar os componentes. E por fim testar com um caso de estudo.

As 7 respostas obtidas pelos jogadores que testaram o caso de estudo indiciam que os mini jogos e as medalhas são úteis para o projeto e funcionam sem problemas. Os eventos dentro do jogo foram gravados e poderão vir a ser analisados de futuro.

Palavras-chave: *Escape room* educativo, *virtual escape room*, mini jogos, *open badges*, sistema de condecoração.

Abstract

The advance in technology has promoted the emergence of new teaching methods, such as educational room escapes that are playful activities used as active learning strategies to increase students' motivation.

The “EdScape” framework allows to create such games (adventure and educational escape rooms) with flexibility for frequent updates by an educator, even outside of the programming environment.

This dissertation implemented: (i) tree types of minigames; (ii) web system of awards, where students can have access to the medals they have won, based on the “open badges” standard; (iii) a database service that logs all relevant game events.

The minigames include puzzles, matching images and labels and closed question / answer.

The web interface for the awards system allows the users to see their medals or port code to incorporate into another web site. Authentication functionality is also implemented for both the framework and the web system. The tools used for the framework are C# within Unity3D and the web interface used CSS, HTML, JavaScript and PHP.

The work was divided into the following steps, starting with identifying the necessary requirements for project development. Second, design the *mockups*. Third, implement the components. Finally, the framework it is tested by creating a case study.

The results from the 7 users of the case study hint that the games and medals are usefull for the proeject and work without issues. Game events were logged and might be used in the future.

Keywords: Educational escape room, virtual escape room, mini games, open badges, badging system.

*“You should be glad that bridge fell down.
I was planning to build thirteen more to that same design”*

Isambard Kingdom Brunel

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivações e Objetivos	2
1.3	Enquadramento “EdScape” com outros trabalhos	2
1.4	Estrutura da Dissertação	2
2	Revisão Bibliográfica	3
2.1	<i>Escape Room</i>	3
2.1.1	Linha história	3
2.1.2	<i>Escape Room</i> Educativo	3
2.1.3	Jogos de aventura vs ER	5
2.2	Conceitos fundamentais	5
2.3	<i>Mozilla Open Badges</i>	6
2.3.1	Tipos de participantes	6
2.3.2	Caraterísticas	7
2.3.3	Conclusão	7
2.4	Ferramentas de aprendizagem	8
2.4.1	Google Formulários	8
2.4.2	LearningApps	9
2.5	Unity3D	11
2.5.1	Caraterísticas	11
2.5.2	Interface	12
2.5.3	Passos para a criação de um jogo	13
3	Projeto “EdScape”	15
3.1	Introdução	15
3.2	Análise	15
3.2.1	Atores	15
3.2.2	Requisitos	16
3.3	Arquitetura	18
3.3.1	Diagrama de caso de uso	18
3.3.2	Diagrama de sequência	19
3.3.3	Diagrama de implementação	20
3.4	Interfaces de mini jogos e mockups	21
3.4.1	Mini jogos	21
3.4.2	<i>Mockups</i>	23

4	Implementação	31
4.1	Unity3D	31
4.1.1	Sistema de autenticação	31
4.1.2	Mini jogos	34
4.1.3	Funcionamento	34
4.1.4	Ficheiro JSON	37
4.1.5	Inquérito	40
4.2	Base de dados	42
4.3	Aplicação <i>web</i>	43
4.3.1	Sistema de autenticação	43
4.3.2	Menu	44
4.3.3	Sistema aberto de medalhas	44
4.3.4	Página “Users”	47
4.3.5	Página “Game”	47
4.3.6	Outras Operações	48
5	Caso de Estudo	51
5.1	Contexto	51
5.2	Resultados	51
5.3	Sugestões	55
6	Conclusão e Trabalho futuro	57
6.1	Conclusões e contribuições	57
6.2	Trabalho futuro	58
A	API	59
A.1	Pedidos do Unity	59
A.1.1	Conta do Utilizador	59
A.1.2	Guardar Inquérito	60
A.1.3	<i>Log</i> e Medalhas	62
A.2	Medalhas (Web)	63
A.2.1	Criar nova medalha	63
A.2.2	Obter detalhes da medalha	64
A.2.3	Obter critério da medalha	64
A.2.4	Obter premiados da medalha	64
A.3	Utilizador (Web)	64
A.3.1	Conta	64
A.3.2	Verificar o código de verificação	66
A.3.3	<i>Upgrade/Update</i>	67
A.4	Página Game (Web)	67
A.4.1	Obter lista	67
A.4.2	Remover logs	68
A.5	<i>Download</i> das tabelas	68
B	Tabelas da Base de dados	69
B.1	Users	69
B.2	Issuer	69
B.3	Badge	70
B.4	Assertion	70

B.5	Survey	71
B.6	Survey_legacy	71
B.7	Game	71
B.8	Game_issuer	72
B.9	Game_log	72
C	Exemplos do ficheiro JSON dos mini jogos	73
D	Exemplo do ficheiro JSON do questionário	77
E	Ficheiro JSON do Caso de Estudo	79
F	Perguntas do Inquérito	87
	Referências	93

Lista de Figuras

2.1	As interações jogador-ator-mundo do jogo. [1]	5
2.2	Infraestrutura da <i>Mozilla Open Badges</i> [2]	7
2.3	Página <i>Google Forms</i>	8
2.4	Passos para criar uma App (Adaptado de [3])	10
2.5	Passos para criar uma coleção (Adaptado de [3])	10
2.6	Interface do editor de cena da Unity3D [4]	13
3.1	Relação entre os atores e a <i>framework</i>	16
3.2	Diagrama de casos de uso	19
3.3	Diagrama de sequência dos casos de usos “login” e “create account”	20
3.4	Diagrama de implementação	21
3.5	<i>Mockup</i> do mini jogo “Puzzle”	22
3.6	<i>Mockups</i> do mini jogo “Quiz”	22
3.7	<i>Mockup</i> do mini jogo “Label”	23
3.8	<i>Mockups</i> da Página Iniciar Sessão	23
3.9	<i>Mockups</i> da Página Criar Conta	24
3.10	<i>Mockups</i> da Página Recuperar Palavra-passe	25
3.11	<i>Mockups</i> da Página Código de Verificação	25
3.12	<i>Mockups</i> da página Nova Palavra-passe	26
3.13	<i>Mockups</i> do Menu das classes	26
3.14	<i>Mockup</i> da Página “Users”	27
3.15	<i>Mockups</i> da Página “Game”	28
3.16	<i>Mockups</i> da Página “Badges” de cada classe	29
3.17	<i>Mockups</i> da Página Detalhes da Medalha	30
3.18	<i>Mockup</i> da Página Criar Medalha	30
4.1	Cabeçalho dos mini jogos	34
4.2	Tempo a terminar	34
4.3	Significado dos botões	35
4.4	Exemplo de um “puzzle” implementado no caso de estudo	35
4.5	Exemplo de um “quiz” implementado no caso de estudo	36
4.6	Exemplo de um “label” implementado no caso de estudo	37
4.7	Painel de introdução do mini jogo - exemplo	37
4.8	Painel após sair do mini jogo se ganhar - exemplo	38
4.9	Painel de fim do mini jogo - exemplo	38
4.10	Classes de configuração do “Puzzle”	39
4.11	Classes de configuração do “Label”	40
4.12	Classes de configuração do “Quiz”	41

4.13	Classes de configuração do Questionário	42
4.14	Diagrama de Base de Dados	42
4.15	Implementação do menu	44
4.16	Código para obter lista de medalhas	45
4.17	Pedido Assíncrono	46
4.18	Código que verifica se conquistou a medalha-exemplo	47
4.19	Exemplo de uma filtragem	48
4.20	Botões de exportar dados	48
4.21	Exemplo de uma exportação	48
4.22	Código embutido	49
5.1	Respostas do inquérito inicial - antes do jogo	52
5.2	Respostas do inquérito final - após o jogo	54
5.3	Média das respostas	55
C.1	Exemplo do ficheiro JSON do mini jogo “Label”	74
C.2	Exemplo do ficheiro JSON do mini jogo “Puzzle”	75
C.3	Exemplo do ficheiro JSON do mini jogo “Quiz”	76

Lista de Tabelas

2.1	Aspetos positivos e negativos do uso da Google Formulários em ERE	9
2.2	Aspetos positivos e negativos do uso da LearningApps em ERE	11
2.3	Plataformas suportadas pela Unity [5, 6]	12
3.1	Requisitos do Administrador	16
3.2	Requisitos do Professor	17
3.3	Requisitos do Estudante	17
3.4	Requisitos da <i>Framework</i>	18

Abreviaturas

2D	Two Dimensional
3D	Three Dimensional
API	Application Programming Interface
BD	Base de Dados
CSV	Comma-Separated Values
ER	Escape Room
ERE	Escape Room Educativo
GF	Google Formulários
IDE	Ambiente de Desenvolvimento Integrado
JSON	JavaScript Object Notation
JWS	JSON Web Signature
LA	LearningApps
NPC	Non-Playable Character
OB	Open Badges
PDF	Portable Document Format
REG	Real Escape Games
UC	Unidade Curricular
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WYSIWYG	What You See is Shat you Get

Capítulo 1

Introdução

1.1 Contexto

Os avanços na tecnologia para além de terem impacto nas áreas da engenharia também tem influenciado cada vez mais a área da educação. Nos últimos tempos têm sido incrementadas as modalidades de ensino, em *B-learning*, uma derivada do *e-learning*, que é uma junção de formação à distância e formação presencial [7].

Um docente ao adotar o método de ensino-aprendizagem B-learning, a aula presencial decorrerá parcialmente numa plataforma *online* [7]. Esta mistura beneficiará o sucesso da aprendizagem, aumentará a participação dos estudantes e poderá despertar o seu interesse pela matéria.

Nos últimos anos, na área da educação, a aprendizagem ativa é da maior importância e esta tem substituído gradualmente o modelo pedagógico tradicional, devido às suas vantagens:

"[...] pesquisas recentes feitas por instituições [...] apontou índices menores de reprovação e notas mais altas" [8]

Pondo em prática a aprendizagem ativa, surgem algumas estratégias como debates, discussões, estudo em grupos e jogos educativos [9]. Entre estas, a estratégia jogos educativos é eventualmente a mais interessante e cativante para os estudantes devido às suas características e o gosto deles pelos jogos, destacando recentemente as salas de fuga educativos.

Os *Escape Rooms* ganharam importância nas atividades lúdicas, mas a utilização dela como estratégia de aprendizagem é limitada, na revisão sistemática realizada por [10], mencionou as limitações e desafios na implementação de *Escape Room* Educativo (ERE) em sala de aula foram: restrições como orçamento, disponibilidade de sala de aula e falta de tempo para preparar as aulas.

Tendo em conta o potencial para a área pedagógica, o projeto em que esta dissertação pertence pretende-se construir uma *framework* que permita criar um jogo educativo, sala de fuga virtual, utilizando a *ferramenta* Unity3D (definido na proposta), ultrapassando assim as limitações anteriormente referidas. Esta *framework* é direcionada principalmente aos professores em que estes possam utilizá-la nas suas aulas, e os estudantes possam utilizá-la fora das aulas para consolidar a matéria.

1.2 Motivações e Objetivos

A principal motivação deste trabalho é tentar criar uma ferramenta que ultrapasse as restrições das ferramentas de jogos educativos existentes que são muito limitadas aos modelos fornecidos, pelo que não é fácil ou até mesmo impossível de manipular conforme as necessidades e interesses dos professores e estudantes. Outra das razões são as restrições que o educante tem perante o ERE em salas de aula, referidas anteriormente.

Os objetivos desta dissertação são implementar as estruturas base dos mini jogos e dos inquéritos para integrar na *framework* do jogo e os conteúdos destes possam ser editáveis pelos educadores conforme as suas necessidade. Outro objetivo é implementar um sistema *web* de condecorações. Após construída a *framework* está será validada através de um caso de estudo definido em equipa.

1.3 Enquadramento “EdScape” com outros trabalhos

O “EdScape” é um projeto desenvolvido em equipa, ou seja, para além desta dissertação há outros trabalhos desenvolvidos em simultâneo visando a construir uma *framework* de um jogo do tipo *escape room* com finalidade educativa e esta poder ser usada pelos educadores.

Agradeço às minhas colegas de equipa, Ana Rita Fonseca Santos e Fátima Coelho Monteiro, pelo esforço e desempenho que mostraram durante o desenvolvimento do projeto, conseguindo assim testar os componentes desenvolvidos nesta dissertação no tempo pretendido.

O caso de estudo produzido no decurso deste trabalho de dissertação, denominado "Escape2Ethics101Test" pode ser jogado em:

<https://fe.up.pt/~efeupinho/jogo01>

1.4 Estrutura da Dissertação

A dissertação encontra-se dividida em seis capítulos:

- Capítulo 1 - é exposto o contexto, as motivações e os objetivos, o enquadramento desta dissertação com outros trabalhos, e a estrutura da dissertação.
- Capítulo 2 - é apresentado uma revisão bibliográfica dos temas relevantes para o projeto.
- Capítulo 3 - é feito uma análise dos atores e requisitos do projeto, é apresentado a arquitetura em diagramas de UML e os *mockups*.
- Capítulo 4 - é apresentada a implementação das componentes criadas no Unity3D e da plataforma *web*.
- Capítulo 5 - é apresentada o caso de estudo e os resultados obtidos.
- Capítulo 6 - é apresentada as conclusões e possíveis implementações e melhoramentos em trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

2.1 *Escape Room*

Este capítulo apresentará uma revisão dos conceitos fundamentais dos Escape Rooms Educativos e das temáticas necessárias para a apresentação do resto do trabalho.

2.1.1 Linha história

Antes do aparecimento da *Escape Room* (ER) já existiam os jogos *point and click* "em que o objetivo é sempre escapar de um cenário resolvendo seus puzzles" [11]. Foi a partir desses jogos o empresário Takao Kato em 2007 teve a iniciativa de investir nos *Real Escape Games* (REG), sendo estes considerados um subgênero dos jogos *point and click*.

ER consiste num jogo multi-jogadores no mundo real, onde estes se encontram fechados num espaço, e têm o objetivo de escapar num determinado tempo limite [12]. A mecânica do jogo está basicamente ligada a diferentes tipos de jogos, procura de objetos escondidos, quebra-cabeças, etc. Neste tipo de jogos desafiantes os participantes terão de utilizar a sua inteligência para encontrar as soluções dos mistérios [12] e trabalhar em equipa para ultrapassar os obstáculos que surgem. Durante o jogo os jogadores desenvolveram as suas habilidades e competências de raciocinar, trabalhar em equipa, a capacidade de obter informação e comunicação [13]. Os ERs do mundo virtual, jogos para telemóveis, é geralmente acompanhada por uma narrativa condutora que conduz o desenvolvimento da história e jogada por um único jogador.

O conceito ER apareceu pela primeira vez no mundo do entretenimento, como referido anteriormente, e que mais tarde foi adotado pelo setor da educação com as respetivas adaptações necessárias aos conteúdos curriculares [8], surgindo assim o conceito *Escape Room* Educativo.

2.1.2 *Escape Room* Educativo

Segundo Moura *et al.*, os *Escape Rooms* Educativos (ERE) são um recurso pedagógico cada vez mais utilizado pelos professores e usado dentro e fora das aulas. Ao ser um jogo educativo,

divertido e motivador, proporciona aos alunos uma imersão lúdica, despertando-lhes o entusiasmo pelos conteúdos curriculares [8], sendo estes mais facilmente adquiridos e memorizados.

Basicamente ERE cria um ambiente com gamificação que poderá ser em alguns casos um jogo que promove a ludificação, mas noutros casos considerados mais um jogo sério, isto dependendo da forma como está construída a narrativa, mas o importante é o encorajamento dos participantes na execução das tarefas, e que os motive e crie emoções.

Desenvolve as competências sociais de estarem em colaboração com os outros participantes do jogo.

Segundo Moura *et al.*, a ludificação da aprendizagem através do ERE pode trazer melhoria nos resultados das notas e ser utilizado em qualquer tipo de contexto de educação desde o ensino especial até ao ensino de adultos. A participação dos estudantes no ERE promove a colaboração e cooperação, o desenvolvimento de interpessoal de competências, espírito de equipa e até as notas dos estudantes [8].

ERE como ferramenta pode ser reutilizado [8], o que permite a poupança de tempo em criar ferramentas do zero, ou adaptado, por um ou mais docentes conforme o conteúdo a ensinar, ao longo do tempo.

2.1.2.1 Etapas para a criação de um ERE

Na criação de um ERE deve-se seguir os seguintes passos [8] :

1. Formular numa narrativa que enquadre no tema dos conhecimentos que pretendem ser transmitidos.
2. Escolher o(s) espaço(s) tendo em consideração ao número de participantes.
3. Criar um vídeo de introdução (opcional, mas é importante para despertar o interesse e melhor integração na história.
4. Preparar os conteúdos pedagógicos adequados a serem depois adaptados para o ambiente do jogo
5. Desenhar o cenário e os enigmas ou desafios, tendo em consideração o público-alvo e os objetivos educativos.
6. Estipular as regras do jogo, o tempo de duração e a formação de grupos.
7. Explorar as ferramentas que adequam ao tema, à narrativa e os desafios definidos.
8. Preparar o guião.
9. Desenvolver e implementar.

2.1.3 Jogos de aventura vs ER

Os jogos de aventura são um dos diversos géneros de categorias de jogos existentes. Neste tipo de jogos existe "uma estrutura rígida para o jogo"[1, tradução do autor], o jogador é apresentado por uma figura no ambiente do jogo e assume o papel protagonista de uma história sendo estas normalmente interessantes, longas e complexas [1, 14], a história é desvendada à medida que o jogador explora, resolve os desafios e avança no tempo.

ER são integrados na categoria de jogo de aventura, mas com certas diferenças. Os jogos de aventura podem ser individuais ou multi-jogadores e têm um lado mais de diversão enquanto os ER são multi-jogadores, os desafios obrigam a puxar mais pela inteligência e conhecimento dos jogadores para resolver os enigmas, valorizam a cooperação entre os elementos da equipa e tem um limite de tempo. Apesar das diferenças existentes o ER tem as características bases dos jogos de aventura, por exemplo, desvendar e encontrar as soluções dos mistérios/desafios com o objetivo de chegar ao final na história.

2.2 Conceitos fundamentais

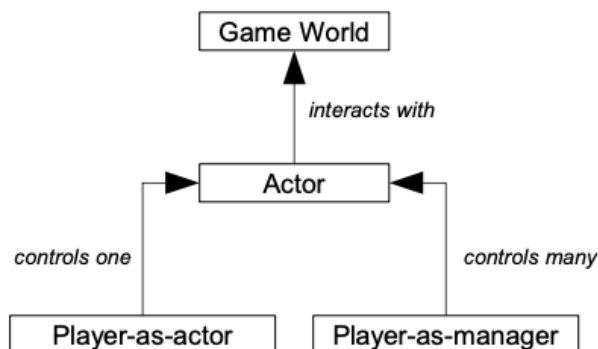


Figura 2.1: As interações jogador-ator-mundo do jogo. [1]

Nesta sessão pretende-se introduzir os conceitos básicos dos jogos digitais, expondo as suas definições.

- **Mundo do jogo** - é o ambiente onde o jogo acontece, contendo as regras e recompensas do jogo. Este pode ser caracterizado em diferentes tipos baseados em agentes referidos em [1].
- **Jogador** - é “uma inteligência controladora, humana ou artificial, que joga o jogo.” [1, tradução do autor].
- **Ator** - é a personagem ou identidade que existe no mundo dos jogos. Estes são controlados pelos jogadores e é a partir destes que o jogador consegue interagir com o mundo de jogo, como mostrado da figura 2.1.

- **Jogador como gerente** - é quando o jogador controla todos os atores, maior que um, para concretizar os objetivos do jogo, geralmente o jogo é do tipo jogo de gerenciamento [1].
- **Narrador** - será um ator que desempenha o papel de conduzir os jogadores pela história narrando-a e apresenta os desafios e as regras.
- **NPC (Non-Playable Character)** - é uma personagem (secundário ou figurante) do jogo e que não é controlada pelo jogador. O NPC pode desempenhar um papel de ajuda aos jogadores para alcançar o objetivo do jogo ou ser um adversário dificultando esse objetivo. [15]

2.3 Mozilla Open Badges

Atualmente, temos a possibilidade de aprender e adquirir conhecimentos em qualquer lugar, mas dificilmente somos reconhecidos pelas capacidades desenvolvidas fora da escola. Para ultrapassar essa dificuldade, em 2010 a *Mozilla Foundation* iniciou o projeto *Open Badges* (OB) com a colaboração da *MacArthur Foundation* [16]. Este projeto consiste em desenvolver um sistema que permita a emissão, coleta e exibição de crachás digitais.

"O Open Badges é o formato líder mundial para crachás digitais"[17, traduzido pelo autor]. Os OB são um tipo de crachá digital verificável que contém as informações sobre habilidades e conquistas, essa informação são anexadas a um arquivo na imagem da medalha. Um crachá digital é um emblema *online* que representa a aquisição de uma habilidade e pode ser compartilhados em perfis online [18], para educação ou emprego.

2.3.1 Tipos de participantes

OB podem ser emitidos, conquistados e geridos usando uma plataforma *Open Badges* certificada, por isso existem três grupos de participantes [2, 19]:

- **Emissores:** são organizações ou indivíduos que criam crachás e os disponibilizam aos ganhadores e os premiam. No contexto acadêmico, os educadores podem utilizar o Moodle visto que esta plataforma fornece uma ferramenta de criação de crachás [20].
- **Ganhadores:** são indivíduos ou grupos interessados em obter crachás em consideração às competências conquistadas. No contexto acadêmico, os estudantes podem importar as suas medalhas para o Moodle visto que esta plataforma disponibiliza essa funcionalidade.
- **Expositores:** são os responsáveis pela exibição dos crachás recebidos pelos ganhadores e disponibilizam a funcionalidade de verificação dos crachás. Esta verificação pode ser feita por dois métodos [21]:
 - Verificação de declarações hospedadas: é verificar se URI da declaração e *Badges Class* existem e assegurar que as origens do domínio coincidem ou são explicitamente permitidas.

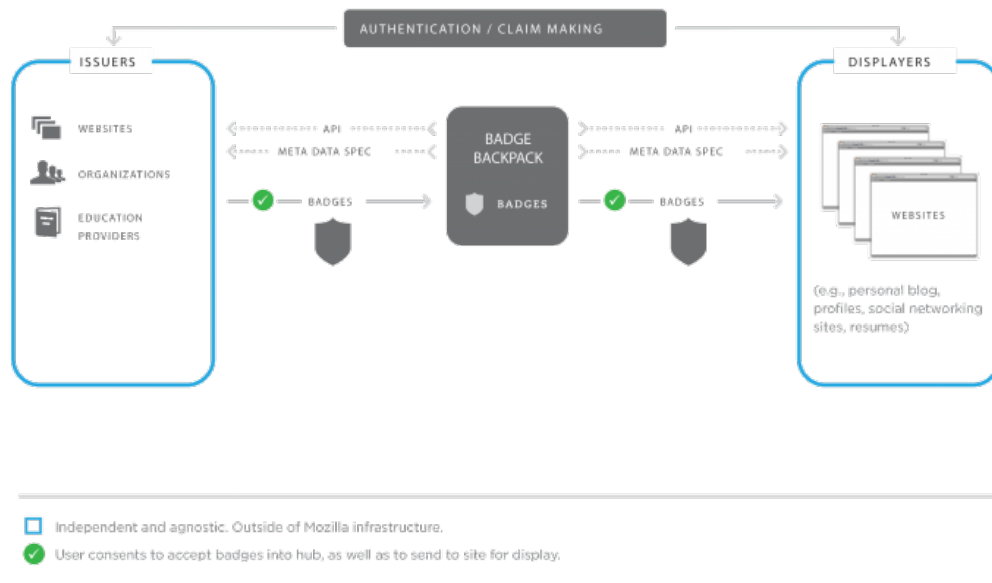


Figura 2.2: Infraestrutura da *Mozilla Open Badges* [2]

- Verificação de declarações assinadas digitalmente por intermédio de Jason Web Signature (JWS): utiliza a chave pública para verificar uma verificação.

2.3.2 Caraterísticas

Os OB são livres e abertas por não ser uma propriedade e ser um *software* livre que pode ser usada por qualquer organização interessada em criar, emitir e verificar as medalhas digitais [22]. São acumuláveis visto que é possível coletar crachás de diversas organizações e acumular para mostrar a história completa das conquistas obtidas durante a aprendizagem [22]. São transferíveis dado que os crachás colecionados podem ser compartilhadas, por exemplo, em *sites* de emprego e perfis de redes sociais [22]. São baseados em dados, pois os OB possuem metadados importantes embutidos no arquivo de imagem dos crachás[22] através do *script* da API [2] (figura 2.2). Cada OB contém o nome do emblema, a descrição, os critérios de receção, o nome do emissor e do destinatário, padrões, data de emissão e de expiração, evidências (URL) e rótulos. Portanto, "cada open badge emitido é único, pessoal e verificável"[20].

2.3.3 Conclusão

Com o aparecimento do *Mozilla Open Badges* contribuiu para obter o reconhecimento das coisas aprendidas e ensinadas aos outros, a verificação das competências e apresentação dos crachás verificados.

No contexto de uma unidade curricular (UC), os OB poderão ser integrados no ERE pelos docentes e emitidos (como certificados formais) conforme as conquistas feitas durante o jogo pelos aprendizados. Os OB adquiridos "terá um valor interno à UC, salvo se a instituição tiver regulamentado a emissão de badges para reconhecimento de UC"[20].

Para poder emitir OB em ambientes virtuais, como ERE virtuais, tem-se integrado de forma personalizada o código na própria plataforma, mas esta integração exigirá suporte de analistas de informática e uma equipa de desenvolvimento de sistemas [20].

2.4 Ferramentas de aprendizagem

Nesta secção serão abordadas algumas ferramentas de aprendizagem que possibilitam a criação de jogos simples de género ER, as suas características e componentes fornecidas para a sua realização.

2.4.1 Google Formulários

A Google Formulários (GF) é um aplicativo muito conhecido de gestão de pesquisas, lançado em 2018. Os utilizadores da Google podem utilizar a GF como uma ferramenta de recolha de informação através da elaboração de questionários, sondagens, formulários de registos e muitos mais [23]. As informações e os resultados do formulário são transmitidos automaticamente e guardados na página do respetivo formulário na secção Respostas (figura 2.3). Nessa secção disponibiliza um resumo das respostas (em gráficos), uma lista de respostas por pergunta e acesso ao formulário de cada indivíduo. Além disso, o GF também possui recursos de colaboração e compartilhamento.

Figura 2.3: Página Google Forms

Na área da educação a GF pode ser usada pelos docentes para a realização de testes *online*. Para isso é necessário ativar a opção questionário nas definições transformando o formulário em questionário, este permite indicar a resposta correta, atribuir pontuação às perguntas e ter classificação automática. Com esta ferramenta os docentes conseguem ter uma estatística dos resultados obtidos.

2.4.1.1 ERE

Com a Google Formulários podem ser criada um ERE digital baseando num diagrama de árvores. A ideia é criar um conjunto de perguntas/desafios e o estudante para conseguir "escapar" terá de acertar todas as questões para obter a chave final.

Na prática, para ter um ERE a funcionar o questionário tem que cumprir certas regras: cada secção convém ter apenas uma pergunta, se acertar recebe uma pista para a chave final e passa para a pergunta seguinte, caso contrário mantém-se na mesma pergunta e é dado uma dica.

Para a realização do jogo a GF fornece perguntas do tipo: resposta curta, resposta longa (não é adequado), escolhas múltiplas, caixas de verificação, pendente, escala linear, grelha de escolha múltipla.

2.4.1.2 Vantagens e Desvantagens

Tabela 2.1: Aspetos positivos e negativos do uso da Google Formulários em ERE

Vantagens	Desvantagens
Fornecer estatística	Não imersivo
Fácil de usar	Não cativante
Multiplataforma	Pouca diversidade de desafios(só perguntas)
Gratuito	

2.4.2 LearningApps

LearningApps (LA) é um aplicativo da Web 2.0, criado na Suíça pela *Bern University of Education* em cooperação com outras universidades [24], com objetivo de aumentar o uso do conteúdo de áudio e vídeo em sala de aula.

A LA é usado para apoiar os processos de aprendizagem e ensino com pequenos módulos interativos [3]. Esses módulos podem ser usados pelos docentes em materiais de aprendizagem, mas também pelos alunos para auto-estudo, criando os seus próprios módulos para consolidar uma matéria e repeti-la [24]. O objetivo é coletar blocos de construção reutilizáveis e disponibilizá-los para todos. Os blocos (designados por Apps, sendo estes mini jogos) não são adequados como lições ou tarefas completas, mas devem ser incorporados num cenário de ensino apropriado [3].

2.4.2.1 Modo de uso

A simplicidade de criar os módulos de aprendizagem é uma das vantagens que incentiva as pessoas a usar LA.

Na figura 2.4 temos os passos para a criação de uma App (bloco). Primeiro, ter uma ideia do conteúdo que a app vai conter. A seguir escolher o modelo mais adequado dos vários modelos disponibilizados pela LA, por exemplo, palavras-cruzadas, exercício de atribuições, escolhas múltiplas, a ferramenta fornece cerca de 20 modelos. Após escolhida o modelo é preenche-lo com o conteúdo e salvar. A App criada poderá ser compartilhada através de um *link* ou *QR code*, e o autor pode decidir se deseja pôr em público, caso ponha este ficará visível na galeria das Apps e poderá ser utilizado por outros utilizadores.



Figura 2.4: Passos para criar uma App (Adaptado de [3])

Na figura 2.5 é apresentada as etapas para a criação de uma coleção. Para isso é preciso primeiro coletar as Apps, blocos de mini-jogos, podendo estas ser criadas pelo utilizador ou usar as apps disponíveis da galeria. Após ter no meu “Stuff” as Apps necessárias, seleciono-as. Antes de salvar a coleção é possível adicionar uma imagem de fundo, uma descrição/dica inicial e um *feedback* no final depois de todas as apps estarem resolvidas.



Figura 2.5: Passos para criar uma coleção (Adaptado de [3])

2.4.2.2 Vantagens e Desvantagens

Tabela 2.2: Aspetos positivos e negativos do uso da LearningApps em ERE

Vantagens	Desvantagens
Fácil manipulação	Pouco imersivo
Diversidade de templates	Pouco cativante*
Permite vídeo, áudio	Não é tempo real
Gratuita	
20 idiomas	
Multiplataforma	

*para alunos do secundário e ensino superior

2.5 Unity3D

Unity, também conhecida por Unity3D, é um motor de jogos e ambiente de desenvolvimento integrado (IDE), criada pela *Unity Technologies*, para a construção de jogos 2D e 3D [25].

A primeira versão foi lançada em 2005. Inicialmente o motor só estava disponível para Mac OS X e só podiam ser implantadas em certas plataformas. A partir da versão 1.1 o Unity começou a suportar exportações para Windows e navegadores *web*. [25]

O Unity tem integrado o editor *MonoDevelop*, mas também permite ao utilizador escolher outro editor. Os *scripts*, inicialmente, podiam ser programados em C#, *UnityScript* (conhecido por *JavaScript*) e Boo [26]. No ano 2015 a linguagem Boo é removida do suporte com o lançamento da versão Unity 5 e em 2017 com o lançamento da Unity 2017 a linguagem *JavaScript* foi descontinuada [6]. C# passou a ser a linguagem de programação primária [6].

A versão mais recente é compatível com Windows e Mac OS X e pode ser exportado para uma maior diversidade de plataformas (tabela 2.3).

2.5.1 Caraterísticas

- Capacidade de criar jogos em 2D e 3D.
- Visualização instantânea - WYSIWYG ("What You See Is What You Get") [27].
- Apresenta uma interface intuitiva, com todas as secções de *scripts*, colisões de objetos, *assets* (texturas e gráficos do jogo) e outras funções muito bem organizadas [28].
- Tem uma boa arquitetura para a programação de *scripts* [29].
- Tem um editor de cenas integrado [29].
- Capaz de se importar diretamente modelos 3D, imagens e efeitos de áudio produzidos em ferramentas externas [29].

Tabela 2.3: Plataformas suportadas pela Unity [5, 6]

Dispositivos móveis	iOS Android BlackBerry Windows
Desktop	Windows Universal Windows Platform Mac OS Linux
Web	WebGL
Realidade Virtual	Windows Mixed Reality Google Cardboard Steam VR Playstation VR Oculus Rift Google's ARCore Magic Leap Microsoft HoloLens

- Unity Asset Store - permite encontrar assets facilmente, gratuitos ou pagos, e exporta-los de maneira fácil e rápida para o Unity [28].
- A simulação física usa a PhysX [29].
- Suporta a diversos formatos de arquivos [27].
- Linguagem de programação em C# (principal) e JavaScript (descontinuado) [27].
- O mesmo código pode ser utilizada para diversas plataformas [27].
- Disponibiliza a documentação, tutoriais e fórum de discussão [27, 28].
- Versão gratuita.

2.5.2 Interface

(A) **Toolbar** - Na barra de ferramentas estão presentes os recursos mais essenciais para o trabalho. À esquerda temos as ferramentas básicas de manipulação da janela cena e os objetos que ela contem. No centro estão os controles de *play*, *pause* e *step*. Nos botões à direita temos o acesso ao *Unity Cloud Services*, à conta do Unity, o menu de visibilidade de camada e o menu de layout do editor [5].

(B) **Hierarchy Window** - janela que representa a hierarquia de todos os objetos da cena em edição [29]. As janelas da cena e da hierquia são inerentemente relacionados, visto que "cada objeto na cena tem uma entrada na hierarquia" [5].

(C) **Game View** - visualização da aparência do jogo que será exibida quando ela estiver finalizada [5].



Figura 2.6: Interface do editor de cena da Unity3D [4]

(D) *Scene View* - é a janela principal que permite navegar, manipular e editar visualmente os elementos da cena. Dependendo do tipo de projeto a vista da cena pode ter uma perspectiva 2D ou 3D[5].

(E) *Inspector Window* - nesta janela tem-se acesso aos parâmetros do objeto selecionado [5].

(F) *Project Window* - janela que exibe a biblioteca de *assets* disponíveis no projeto [5]. A organização e manipulação da estrutura dos arquivos devem ser feitas somente dentro da Unity3D [29].

(G) *Status bar* - fornece notificações sobre os processos.

2.5.3 Passos para a criação de um jogo

- 1- Criar um novo projeto;
- 2- Procurar ou criar os *assets* e guardar-los;
- 3- Arrastar um *asset* para a janela de cena e configura-lo (textura, transformações, sombras, etc.);
- 4- Criar *scripts* e adicionar ao *asset* pretendido;
- 5- Fazer ajustes necessários.

Capítulo 3

Projeto “EdScape”

3.1 Introdução

O Projeto “EdScape”, é um projeto de equipa que consiste na implementação de uma *framework* que permita criar um jogo do tipo *escape room* educativo, e desenvolvimento de um sistema *web* de condecoração. Nesta dissertação é focada principalmente no desenvolvimento da base de alguns mini jogos para o jogo ERE, desenvolvimento de uma aplicação *web* para ter acesso às informações dos jogos (administrador e professores) e de um sistema aberto de medalhas (incluído na aplicação *web*). Neste sistema os professores podem criar medalhas e ter acesso à lista dos premiados de cada medalha, e os estudantes têm acesso às medalhas a que foram premiados.

3.2 Análise

3.2.1 Atores

Relativamente à *framework* foi considerada três atores: o *co-developer* informático, o educador e o estudante. Nesta dissertação tomei o papel de *co-developer* no desenvolvimento das componentes do projeto e o papel de educador na construção do caso de estudo.

- **Co-Developer Informático** - é a pessoa responsável pela implementação da *framework*, neste caso é desenvolvida em Unity3D, procura de *assets* e programação da página *web* para ter acesso às medalhas ganhas e emitidas, e informações dos jogos jogados pelos estudantes.
- **Educador** - é o gestor de conteúdos do jogo, ou seja, o responsável pela narrativa, conteúdos dos mini jogos, dicas, duração do jogo e dos mini-jogos, entre outros. O educador poderá ter de mexer no Unity3D, mas pouco, conforme as necessidades dos conteúdos que pretende integrar no jogo. O educador possivelmente irá alterar os ficheiros JSON para fazer alterações ao jogo.
- **Estudante** - é a pessoa que joga o jogo educativo, no *browser* ou no executável, e é avaliado.

A relação entre atores e a *framework* está demonstrado na figura 3.1. O *co-developer* implementa a *framework*, depois o educador cria o jogo final completando o conteúdo da *framework* a partir dos ficheiros JSON, e o estudante joga o jogo criado.

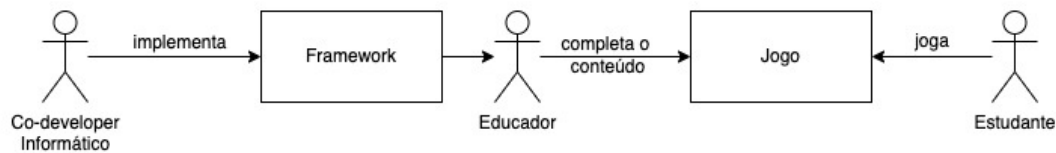


Figura 3.1: Relação entre os atores e a *framework*

Na base de dados, para distinguir os diferentes papéis dos utilizadores, na tabela *user* (figura 4.14) cada utilizador pode ser classificado por *student*, *issuer* ou *super user* (por ordem crescente de privilégios). O *student* é a classe predefinida para as contas acabadas de ser criadas e geralmente corresponde a um estudante. O *issuer* é um utilizador com privilégios intermédios que cria jogos e/ou medalhas, sendo que frequentemente corresponderá a um professor. A conta classificada por *super user* é o utilizador com o maior privilégio, o administrador.

3.2.2 Requisitos

Numa fase inicial do projeto foram definidos diversos requisitos para os três tipos de utilizador: administrador, professor e estudante, e também para a *framework*.

- Administrador:

Tabela 3.1: Requisitos do Administrador

Requisitos	Descrição
R_A01	ter acesso à lista de todos os utilizador
R_A02	ter acesso à lista de todos os <i>logs</i> dos jogos jogados
R_A03	ter acesso à lista de todas as medalhas e os premiados
R_A04	exportar a lista dos usuários, dos <i>logs</i> dos jogos jogados, das medalhas e dos premiados de uma certa medalha
R_A05	fazer <i>login</i>
R_A06	recuperar palavra passe
R_A07	passar uma conta da classe “student” para uma conta de classe “issuer”
R_A08	copiar o código embutido das medalhas

- Professor (gestor de conteúdos):

Tabela 3.2: Requisitos do Professor

Requisitos	Descrição
R_P01	adicionar novos <i>assets</i> ao jogo
R_P02	adicionar o contexto/narrativa do jogo
R_P03	criar uma nova versão do jogo
R_P04	criar medalhas (<i>Mozilla Open Badge</i>)
R_P05	exportar a lista dos <i>logs</i> dos jogos jogados, das medalhas e dos premiados de uma certa medalha
R_P06	criar conta e fazer <i>login</i>
R_P07	recuperar palavra passe
R_P08	copiar o código embutido das medalhas

- Estudante (jogador):

Tabela 3.3: Requisitos do Estudante

Requisitos	Descrição
R_E01	criar uma conta e fazer <i>login</i>
R_E02	recuperar palavra passe
R_E03	entrar na sala de espera
R_E04	conseguir visualizar o tempo restante do jogo
R_E05	conseguir visualizar o estado da equipa
R_E06	durante o jogo, registar uma saída intempestiva / falta de comunicações
R_E07	“perder vida” - pede ajuda e perde tempo
R_E08	receber uma classificação e ser premiado com medalhas
R_E09	copiar o código embutido das medalhas
R_E10	selecionar idioma

- *Framework*:

Tabela 3.4: Requisitos da *Framework*

Requisitos	Descrição
R_F01	interagir com o jogo (clicar, arrastar, rato, teclado, introduzir texto)
R_F02	ativar um evento quando está próximo do objeto
R_F03	incluir mini-jogos
R_F04	multi-jogadores
R_F05	<i>assets</i> de brilho e som

3.3 Arquitetura

Nesta secção é apresentada o *design* da arquitetura do projeto, recorrendo aos diagramas de UML como o diagrama de caso de uso, diagrama de sequência e diagrama de implementação. A UML é uma linguagem que permite ter uma representação do sistema de uma forma padronizada, ajudando a ter uma pré-visualização lógica e comunicação entre os objetos.

3.3.1 Diagrama de caso de uso

O diagrama de caso de uso ilustrado na figura 3.2 retrata as interações entre os atores (utilizadores) e o sistema desenvolvido.

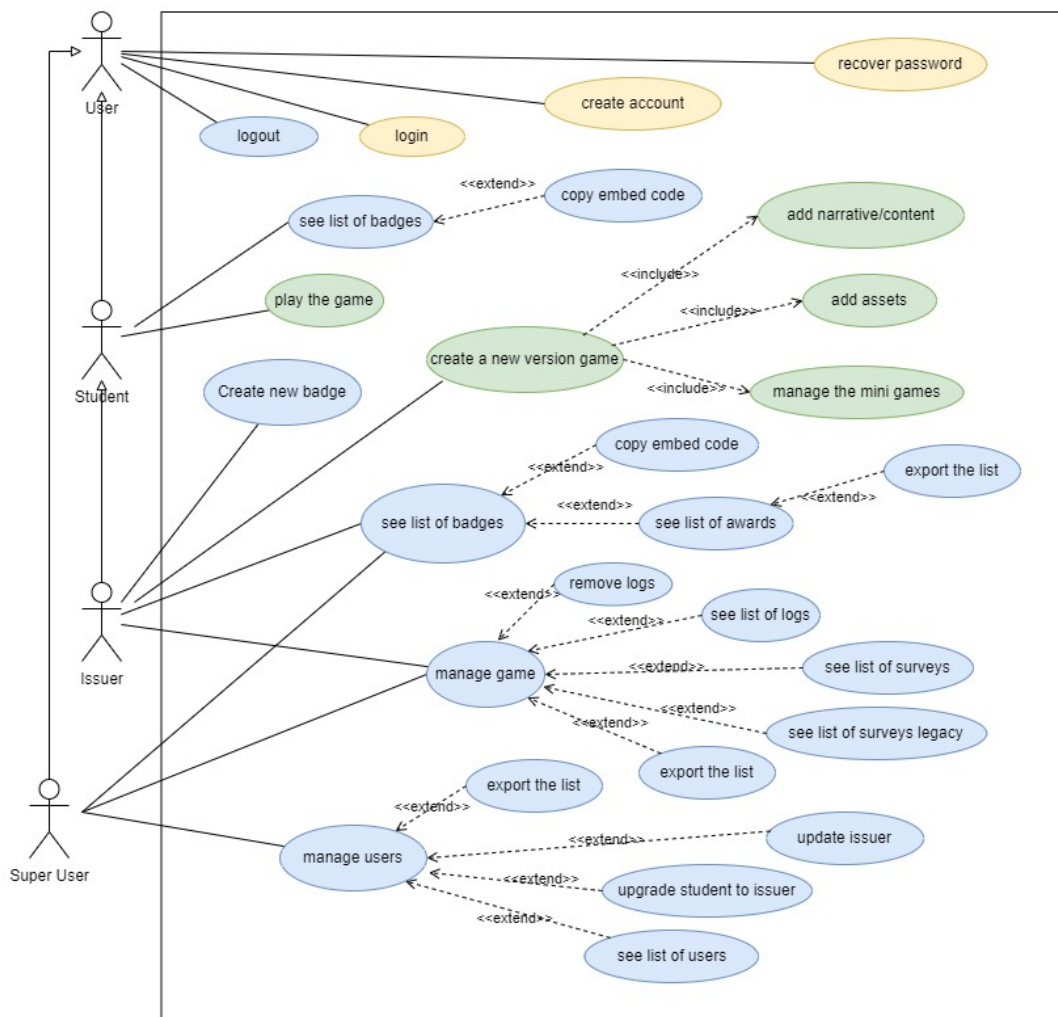


Figura 3.2: Diagrama de casos de uso

Na figura 3.2 podemos identificar três cores diferentes para três tipos de casos de uso:

- A azul são ações de interação com o sistema do aplicativo *web* das medalhas
- A verde são ações de interação com o sistema da *framework* do jogo
- A amarelo são ações de interação permitidos em ambos os sistemas

3.3.2 Diagrama de sequência

Um diagrama de sequência é um cenário de eventos, é usado para representar os detalhes, a lógica do processo e interação entre os objetos de um caso de uso.

Os casos de uso presentes no diagrama da figura 3.2 a azul a maioria são apenas processos de ler, atualizar ou inserir dados na base de dados e a verde são basicamente casos de uso de edição

dos ficheiros JSONs da *framework*. Por isso é feita uma demonstração do diagrama de sequência dos casos de uso “login” e “create account” no lado da *framework*.

No diagrama de sequência da figura 3.3 começa por criar uma conta e a seguir iniciar sessão. A partir do diagrama podemos observar os detalhes e perceber como funciona os dois processos.

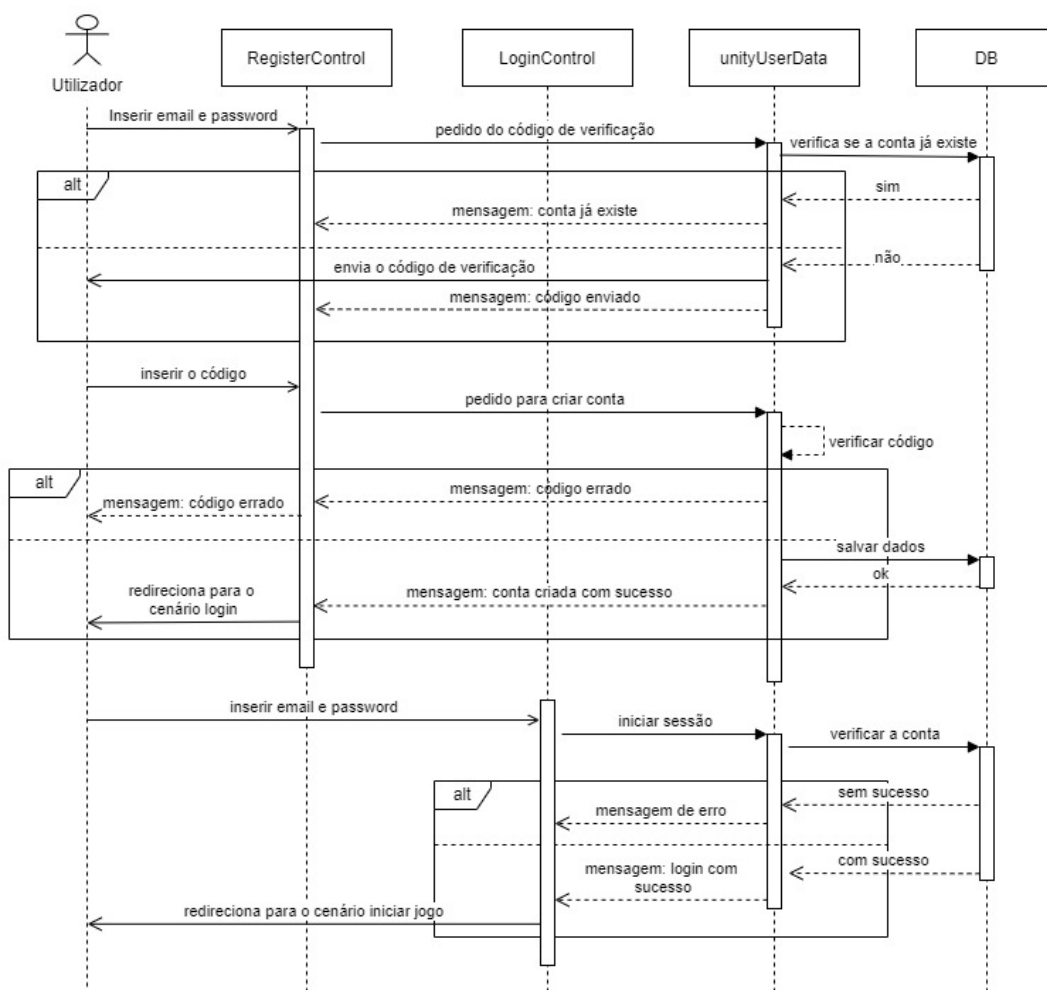


Figura 3.3: Diagrama de sequência dos casos de usos “login” e “create account”

3.3.3 Diagrama de implementação

Na figura 3.4 temos a representação do diagrama de implementação que mostra uma visão estática do funcionamento do sistema e esta é composta por nós (caixa tridimensional), componentes dos nós e o relacionamento entre nós.

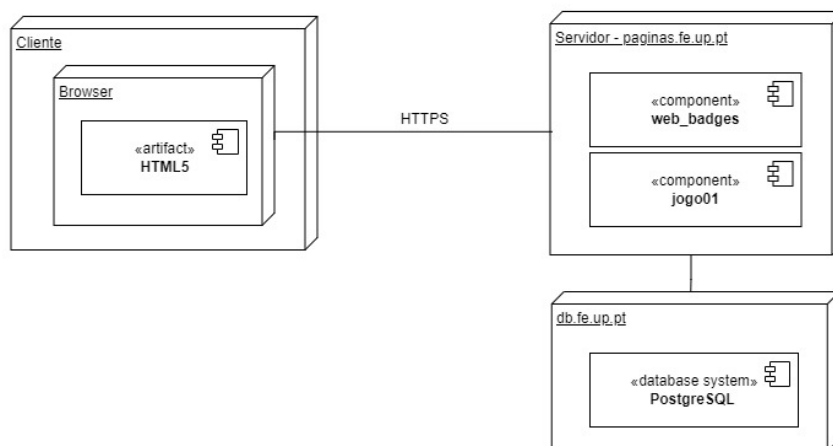


Figura 3.4: Diagrama de implementação

No diagrama temos um servidor *web*, um servidor de base de dados e um aparelho onde o utilizador visualiza a página, o *browser*. O Cliente para aceder ao servidor faz um pedido através do HTTPS do *brower*, podendo ser um pedido para a componente “web_badges” ou “jogo01” e o servidor retornará, respetivamente, a página do aplicativo *web* das medalhas ou a página do jogo.

3.4 Interfaces de mini jogos e mockups

3.4.1 Mini jogos

Como o tema do projeto é baseado em ERs, um aspeto importante e indispensável são os desafios presentes durante a atividade ER em que os participantes para escapar da sala têm que ultrapassar esses desafios.

No nosso projeto esses desafios são representados por mini jogos, por isso foi criado três tipos de mini jogos:

- “Puzzle”

Neste mini jogo no lado direito temos as peças do *puzzle* em posições aleatórias e no lado esquerdo temos o lugar em que as peças deveram ser postas. Para este mini jogo o educador precisa de indicar a imagem a utilizar no *puzzle* e outras informações de definição num ficheiro JSON que será abordado com mais detalhes na secção 4.1.4.1. Este tipo de mini jogo pode ser usado para apanhar um certo objeto, mas passando por um desafio primeiro e/ou para transmitir alguma informação importante a partir da imagem do *puzzle*. A ideia é que este jogo permita ativar a aprendizagem, estimular a curiosidade e a criação pode reforçar ideias a reter.

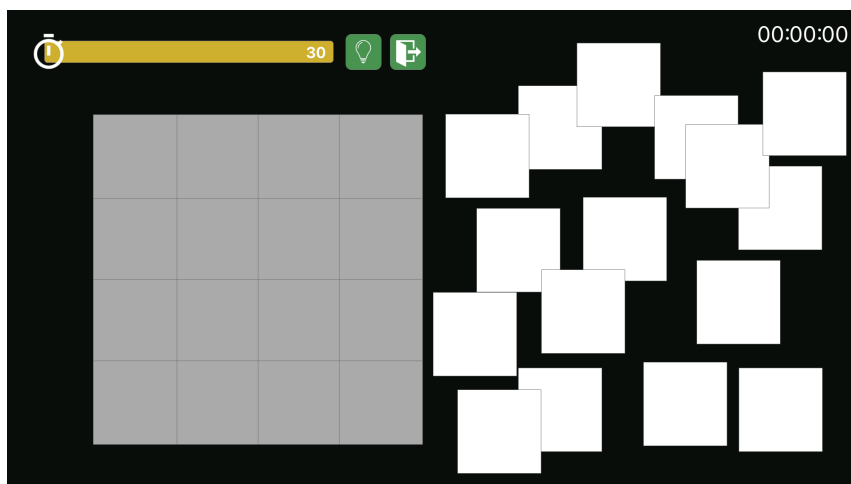


Figura 3.5: *Mockup* do mini jogo “Puzzle”

• “Quiz”

No “Quiz” temos um conjunto de perguntas com as suas respetivas opções de respostas (fig.3.6a), sempre que uma resposta é clicada surge um “feedback” (fig.3.6b). O educador precisará de escrever as questões e respostas num ficheiro JSON com uma determinada estrutura, esta será descrita com mais detalhes na secção 4.1.4.3.

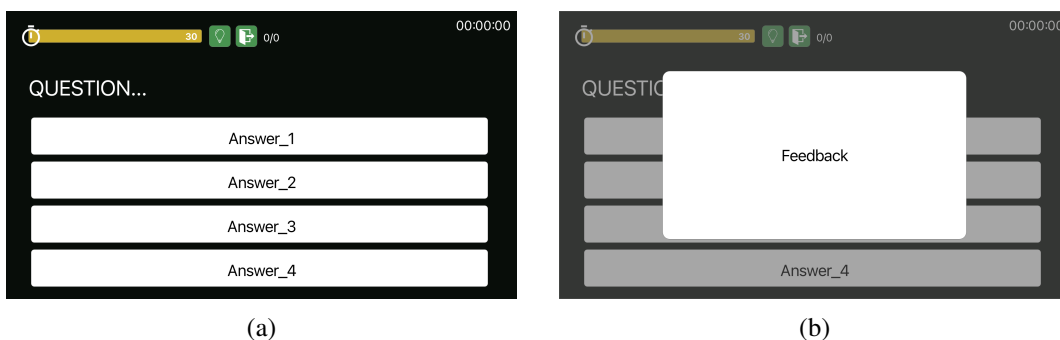
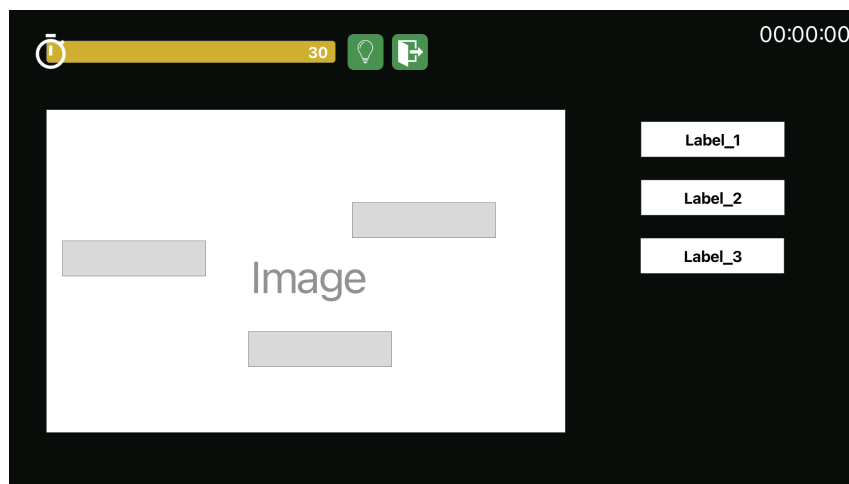


Figura 3.6: *Mockups* do mini jogo “Quiz”

• “Label”

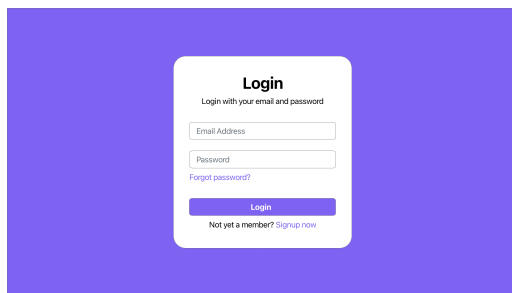
No mini jogo “Label” no lado direito temos as etiquetas para preencher os campos vazios (retângulos cinzentos) da imagem do lado esquerdo, podendo ser esta um jogo de correspondência ou preenchimento de espaços de uma frase. Neste mini jogo o educador terá de fornecer a imagem, as posições dos campos vazios e os nomes das etiquetas num ficheiro JSON, a estrutura do ficheiro será descrita na secção 4.1.4.2.

Figura 3.7: *Mockup* do mini jogo “Label”

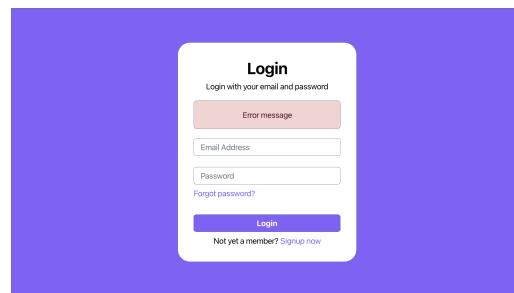
3.4.2 Mockups

Com intenção de criar uma interface intuitiva e simples para os utilizadores, foi criado os seguintes *mockups* para as diversas interfaces usando o Adobe XD, uma ferramenta de design de experiência do utilizador que ajuda a criar protótipos de interfaces para aplicativos moveis e da *web*.

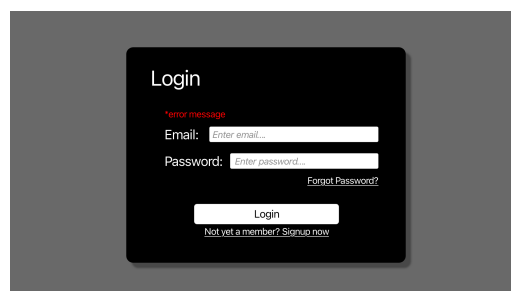
- Iniciar Sessão



(a) Web



(b) Web - mensagem de erro



(c) Unity3D

Figura 3.8: *Mockups* da Página Iniciar Sessão

Para a página Iniciar Sessão teve de se criar dois *mockups* distintos um para a plataforma *web* (fig.3.8a) e outra para o jogo (unity3D) (fig.3.8c). Tanto para entrar na página *web* ou no jogo é obrigatório autenticar-se.

Se o utilizador não tiver conta e quiser criar uma, ao clicar no botão “Signup now” é redirecionado para a página Criar Conta (fig.3.9).

Se o utilizador já tiver conta, mas se tiver esquecido da palavra-passe poderá recuperar-lá clicando no botão “Forgot Password” é redirecionado para a página Recuperar Palavra-passe (fig.3.10).

• Criar Conta

Para a página Criar Conta também foi criado dois *mockups* distintos. A grande diferença entre estas duas interfaces é que na plataforma *web* a criação da conta e o código de verificação estão em páginas distintas enquanto na interface do jogo estão contidos num único cenário.

Na página/cenário Criar Conta o utilizador pode ir para a página/cenário Iniciar Sessão clicando no botão “Login”.

(a) Web

(b) Web - mensagem de erro

(c) Unity3D

Figura 3.9: *Mockups* da Página Criar Conta

• Recuperar Palavra-passe

Para recuperar a palavra-passe o utilizador tem que introduzir o email da conta e clicar no botão “Continue” para fazer o pedido. Caso o email introduzido não exista é disparado a

aviso da figura 3.10b. Caso contrário é enviado o código para o respetivo email e redirecionado para a página Código de Verificação (fig.3.11a), mas no lugar de “verification code” temos “password reset otp”.

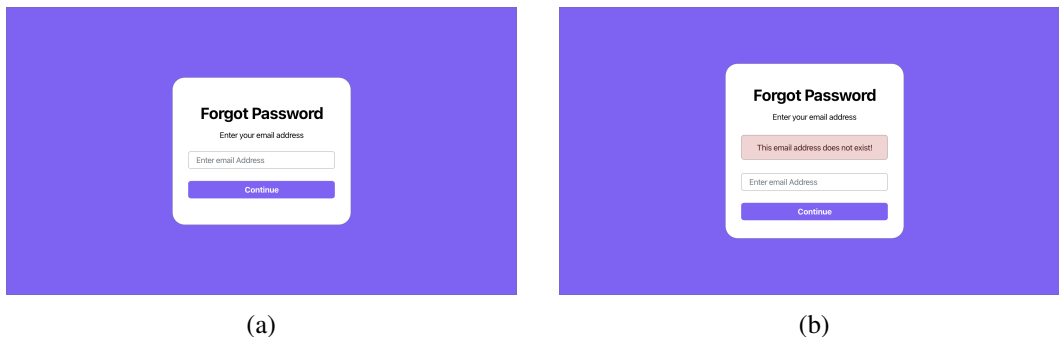


Figura 3.10: *Mockups* da Página Recuperar Palavra-passe

- **Código de Verificação**

O *mockup* desta página tanto pode ser usado para a verificação do código de verificação da conta criada como também para a verificação do código de verificação de recuperação da palavra-passe.

Se o código de verificação estiver correto, após clicar no botão “Submit”, no caso de verificação da conta é redirecionado para a página inicial do utilizador de classe “student” (fig.3.16c) enquanto no caso de verificação da recuperação da palavra-passe é redirecionado para a página Nova Palavra-passe (fig.3.12).

Se o código for incorreto é disparado o aviso de erro da figura 3.11b.

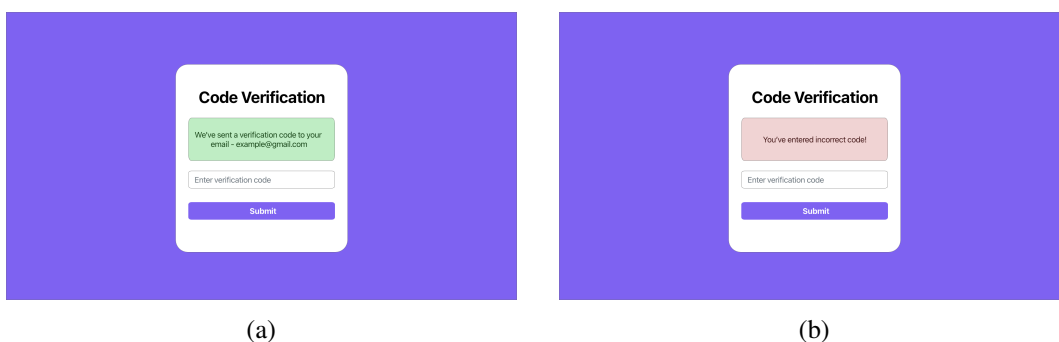


Figura 3.11: *Mockups* da Página Código de Verificação

- **Nova Palavra-passe**

Nesta página o utilizador preenche os campos “Create new password” e “Confirm your password” com a nova palavra-passe, depois clica no botão “Change” e este envia um pedido ao servidor. No servidor são feitas duas verificações:

- Palavras-passes coincidem

- Palavra-passe diferente da antiga

Se cumprir ambas as regras, é redirecionado para a página da figura 3.12c onde ao clicar em “Login now” é redirecionada para a página Iniciar Sessão (fig.3.8a).

Se falhar pelo menos uma das verificações é mostrada no campo “Error message” a descrição do respetivo erro.

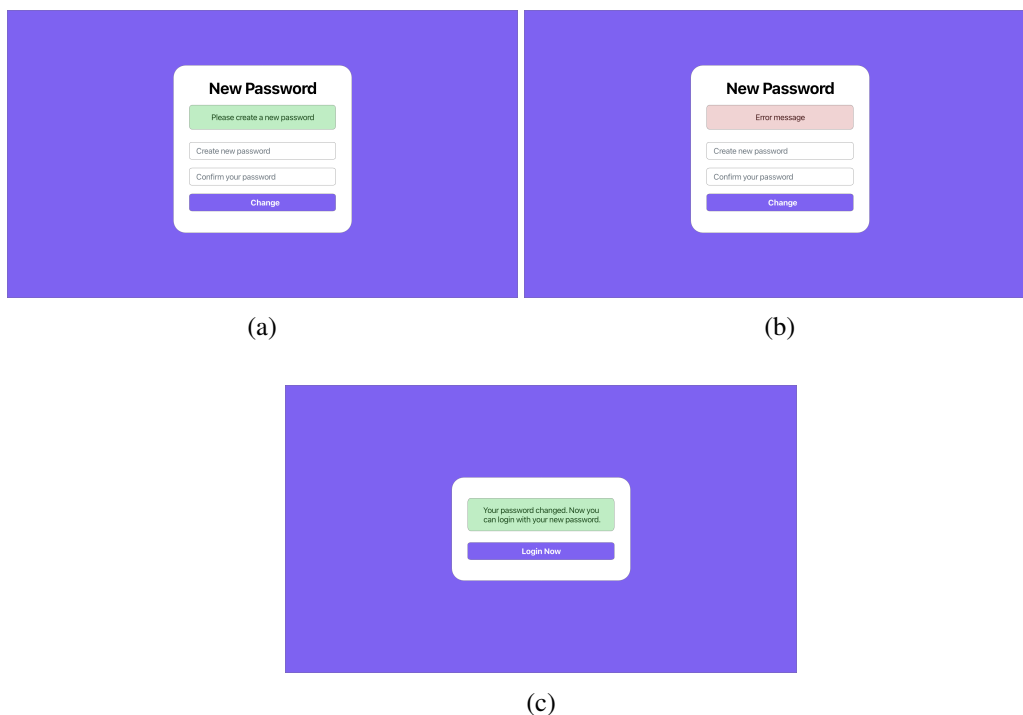


Figura 3.12: *Mockups* da página Nova Palavra-passe

• Menu

Foi criado três menus diferentes, um para cada classes. Inicialmente, quando uma conta é acabada de ser criada pertence à classe “student”, ou seja, terá o menu da fig.3.13c. Uma conta passará a ter o menu da fig.3.13b assim que o “super user” *upgrade* dessa conta para a classe “issuer”. A conta “super user” (administrador) é definida já por padrão e tem o menu da fig.3.13a. O conteúdo de cada item do menu é descrito mais adiante.

Edscape	Users	Badges	Game	Logout	(a) “super user”
Edscape		Badges	Game	Logout	(b) “issuer”
Edscape			Badges	Logout	(c) “student”

Figura 3.13: *Mockups* do Menu das classes

• Página “Users”

A página “User” é apenas acessível ao utilizador da classe “super user”. Nesta página o administrador tem acesso à lista de todas as contas e pode executar três ações:

- **Atualizar:** clicando no botão “Update” atualiza os dados do “issuer”;
- **Upgrade:** clicando no botão “Upgrade”, após ter os campos das colunas “Issuer Name” e “Issuer Url” preenchidos, passa uma conta da classe “student” para “issuer”;
- **Exportar:** clicando no botão “Export the table” exporta os dados da tabela para um ficheiro csv.

ID	Email	Status	Class	Issuer Name	Issuer Url
1	admin@gmail.com	verified	super user	admin	-
2	student1@gmail.com	verified	student	issuer or orga	url
3	teacher1@gmail.com	verified	issuer	name	https://sigarra.up.pt/...

Figura 3.14: Mockup da Página “Users”

• Página “Game”

A página “Game” pode ser acedida pelos “issuers” e pelo “super user”. Qualquer uma das imagens da figura 3.15 pode ter o menu do “issuer” (fig.3.13b) ou do “super user” (fig.3.13a).

Ações permitidas:

- **Selecionar o tipo:** na opção “Select type” o utilizar pode escolher a categoria de informação que quer visualizar. Há três hipóteses de escolha como podemos ver na figura 3.15c;
- **Selecionar versão:** na opção “Select a game version” o utilizador seleciona a versão do jogo. Se o utilizador for “super user” nesta opção aparecerá todas as versões do jogo presentes na base de dados, caso seja um “issuer” apenas aparecerá as versões a que pertence;
- **Atualizar:** clicando no botão “Show table” atualiza a tabela conforme as opções selecionadas;
- **Remover:** clicando no botão “Remove logs” remove todos os “logs” com mais de um ano;
- **Exportar:** clicando no botão “Export the table” exporta os dados da tabela para um ficheiro csv.

A tabela está predefinida para o tipo “Log” com todas as versões.



Figura 3.15: Mockups da Página “Game”

• Página “Badges”

A página “Badges” pode ser acessada por todos os utilizadores. É nesta página que se tem acesso à lista das medalhas premiadas (no caso do “student”) ou criadas (no caso do “issuer” e “super user”). A diferença entre as classes é que o “issuer” pode criar medalhas e os outros não.

Ações permitidas:

- **Copiar o código embutido:** clicando no *icon* de copiar copia o código para poder integrar a lista de medalhas noutras páginas;
- **Criar medalha:** clicando no botão “New Badge” é redirecionado para a página Criar Medalha (fig. 3.18), esta ação é apenas permitada os utilizadores de classe “issuer”;
- **Clicar na medalha:** clicando numa medalha é rediredionado para a página Detalhes da Medalha (fig.3.17);
- **Exportar:** clicando no botão “Export the table” exporta a lista das medalhas para um ficheiro csv.

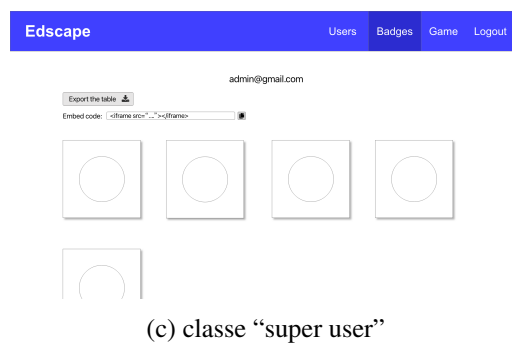


Figura 3.16: *Mockups* da Página “Badges” de cada classe

• Detalhes da Medalha

A página Detalhes da Medalha difere da classe “student”(fig.3.17a) para a classe “issuer”/“super user” (fig.3.17b). A grande diferença é no lado do “student”, temos as informações sobre a medalha, o email do premiado e a data que este recebeu a medalha enquanto no lado do “issuer”/“super user” para além dos dados a medalha temos uma lista dos estudantes premiados ao clicar no “Awards”.

Ações permitidas:

- **Voltar:** clicando no botão “Back” volta para a página anterior, ou seja, página “Badges” (fig.3.16);
- **Exportar:** clicando no botão “Export the table” exporta a lista dos estudantes premiados para um ficheiro csv. Apenas para as classes “issuer” e “super user”.

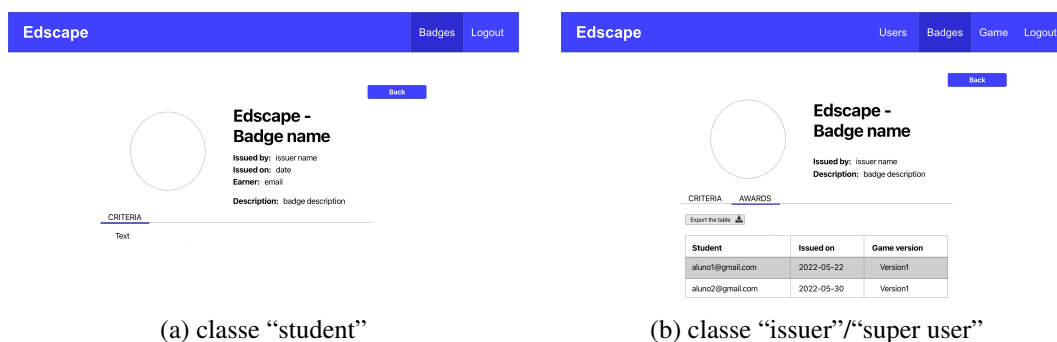


Figura 3.17: Mockups da Página Detalhes da Medalha

• Criar Medalha

Para criar uma medalha o “issuer” tem que preencher os campos nome, descrição, critérios e escolher uma imagem.

Ações permitidas:

- **Upload:** clicando no botão “Choose file” permite fazer *upload* de uma imagem do tipo png, jpg ou jpeg;
- **Voltar:** clicando no botão “Back” volta para a página anterior, página “Badges” (fig.3.16b);
- **Criar:** clicando no botão “Create” a medalha é criada e é redirecionada para a página “Badges” (fig.3.16b).

Figura 3.18: Mockup da Página Criar Medalha

Capítulo 4

Implementação

Nesta secção é apresentada as componentes implementadas no Unity3D (as interfaces de iniciar sessão e criar conta, os mini jogos e os questionários), a base de dados e o desenvolvimento da página web

4.1 Unity3D

4.1.1 Sistema de autenticação

Para poder saber a quem pertence as informações recolhidas do jogo houve o interesse de distinguir os jogadores daí o aparecimento da funcionalidade de iniciar sessão e criar conta.

As interfaces Iniciar Sessão e de Criar Conta são compostos por um conjunto de componentes: *Text*, *InputField* e *Button*. Estes são do tipo *TextMeshPro*, ou seja, as componentes redimensionam automaticamente conforme o tamanho da tela utilizada.

4.1.1.1 Criar Conta

Para criar uma conta é necessário ter os campos “Email”, “Password” e “Code” preenchidos (fig. [A.1.1.3](#)). Inicialmente o campo “Code” está desativo, para o ativar o utilizador tem que submeter os outros dois dados primeiro clicando no botão “Send code”, antes de enviar pedido (anexo [A.1.1.2](#)) ao servidor este verifica se estão preenchidos. Se o pedido retornar uma resposta de sucesso a conta é criada e dispara no campo “alert text”, a verde, o aviso de já ter enviado o código para o email. Após de ter todos os parâmetros preenchidos, clicando no botão “Signup” é enviado outro pedido ao servidor (anexo [A.1.1.3](#)). Se o “code” estiver correto a conta é verificada e redirecionado para o cenário Iniciar Sessão (fig. [3.8c](#)). Se ocorrer algum erro durante o processo de criação da conta por não cumprimento de alguma condição será demonstrada no campo “alert text” a vermelho com a respetiva descrição do erro.

Classe *RegisterControl*

A classe *RegisterControl* trata das ações que ocorrem na interface Criar Conta (fig.3.9c).

Atributos:

- ***email***: email introduzido pelo utilizador;
- ***password***: palavra-passe introduzida pelo utilizador;
- ***code***: código de verificação introduzida pelo utilizador;
- ***url***: contém o endereço para fazer o pedido de criar uma conta (API unityUserData)

Métodos:

- ***ReadEmail(string s)***: lê e guarda a *string* no atributo *email* sempre que termina a edição do *InputField* do email;
- ***ReadPassword(string s)***: lê e guarda a *string* no atributo *password* sempre que termina a edição do *InputField* da palavra-passe;
- ***ReadCode(string s)***: lê e guarda a *string* no atributo *code* sempre que termina a edição do *InputField* do código de verificação;
- ***SendCodeButton()***: é chamada quando se clica no botão “Send Code”, verifica se os campos “Email” e “Password” estão preenchidos se sim invoca o método *SendCode*, caso contrário mostra uma mensagem de erro;
- ***SendCode()***: faz um pedido do tipo *POST* para solicitar o código de verificação (anexo [A.1.1.2](#)) ao endereço do atributo *url*.
- ***RegisterButton()***: é chamada quando se clica no botão “Signup”, verifica se os todos os campos estão preenchidos se sim invoca o método *SignUp*, caso contrário mostra uma mensagem de erro;
- ***SignUp(string email, string password, string code)***: faz um pedido do tipo *POST* de criar conta (anexo [A.1.1.3](#)) ao endereço do atributo *url*.
- ***GoToLoginPanel()***: é chamado quando se no botão “Log in”. Redireciona para a interface Iniciar Sessão.

4.1.1.2 Iniciar Secção

Para iniciar sessão o utilizador após preencher os campos e clicar no botão “Login” é enviado um pedido de autenticação ao servidor (anexo A.1.1.1), se este retornar uma resposta de sucesso é redirecionada para o cenário inicial o jogo, caso contrário é exibido a mensagem de erro retornada no campo “error message” (fig.3.8c).

Classe *LoginControl*

A classe *LoginControl* trata das ações que ocorrem na interface Iniciar Sessão (fig. 3.8c).

Atributos:

- **email:** email introduzido pelo utilizador;
- **password:** palavra-passe introduzida pelo utilizador;
- **urlResetPassword:** contém o endereço para recuperar a palavra-passe (API formForgotPassword);
- **urlNewPassword:** contém o endereço para definir uma nova palavra-passe no caso dela já ter há mais de um ano (API formNewPassword);
- **urlLogin:** contém o endereço para fazer o pedido de login (API unityUserData).

Métodos:

- **ReadEmail(string s):** lê e guarda a *string* no atributo email sempre que termina a edição do *InputField* do email;
- **ReadPassword(string s):** lê e guarda a *string* no atributo *password* sempre que termina a edição do *InputField* da palavra-passe;
- **LoginButton():** é chamada quando se clica no botão “Login”, verifica se os campos estão todos preenchidos se sim invoca o método Login, caso contrário mostra uma mensagem de erro;
- **Login(string email, string password):** faz um pedido do tipo *POST* de iniciar sessão(anexo A.1.1.1) ao endereço do atributo *urlLogin*. Caso o retorno seja do tipo *success* é redirecionado para o início do jogo e no caso do *red-old* é aberto uma página web para o endereço do atributo *urlNewPassword*.
- **ResetPassword():** é chamado quando se clica no botão “Forgot Password?”. É aberto uma página web para o endereço do atributo *urlResetPassword*;
- **GoToRegisterPanel():** é chamado quando se no botão “Signup now”. Redireciona para a interface Criar Conta.

4.1.2 Mini jogos

A implementação dos mini jogos pode ser dividida em duas etapas: funcionamento e conteúdo (ficheiro JSON). E a interface destes é composta pela zona do jogo e o cabeçalho.

4.1.3 Funcionamento

Nesta subsecção será apresentada o funcionamento do cabeçalho e dos mini jogos criados.

4.1.3.1 Cabeçalho

No cabeçalho podemos encontrar o tempo do mini jogo, botão de ajuda, botão de saída, pontuação (em alguns) e o tempo global do jogo (fig.4.1).

O tempo do mini jogo é parametrizável pelo educador sendo configurado a partir do ficheiro JSON, sempre que o tempo chega aos 5 segundos a cor passa para vermelho como podemos observar na figura 4.2, esta mudança de cor serve como um impacto visual de que o tempo está acabar.



Figura 4.1: Cabeçalho dos mini jogos



Figura 4.2: Tempo a terminar

O botão de ajuda está pré-configurado. No “Puzzle” e no “Label” este botão serve para remover as peças que estão colocadas em posições erradas, enquanto no “Quiz” elimina uma resposta errada.

O botão de sair permite sair do mini jogo e voltar para a sala de jogo.

Em ambos os botões ao pôr o rato por cima aparece uma mensagem a indicar o que faz e a penalização sofrida (fig.4.3). Esta informação é definida no JSON do mini jogo.

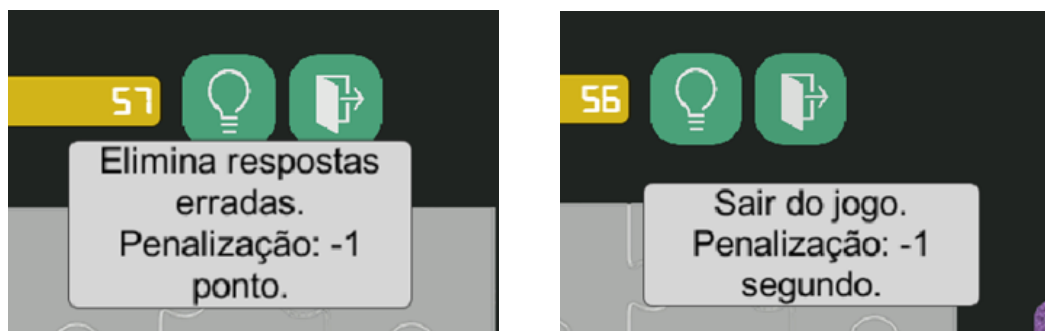


Figura 4.3: Significado dos botões

4.1.3.2 “Puzzle”

O “Puzzle” está definido para um tamanho 4x4 e funciona da seguinte maneira:

- antes de iniciar o mini jogo a imagem e as posições aleatórias das peças são configuradas conforme as informações definidas no ficheiro JSON.
- sempre que se pega numa peça e depois se larga é calculado o slot mais perto e se essa distância é inferior que 1 o centro da peça é ajustado ao centro do slot, caso contrário fica na posição onde se largou.
- quando é posto uma peça num slot já ocupado por outra peça, essa é substituída e volta para a sua posição inicial.
- o mini jogo termina quando o tempo acaba ou quando as peças estão todas no lugar certo ou quando se clicar no botão de sair.

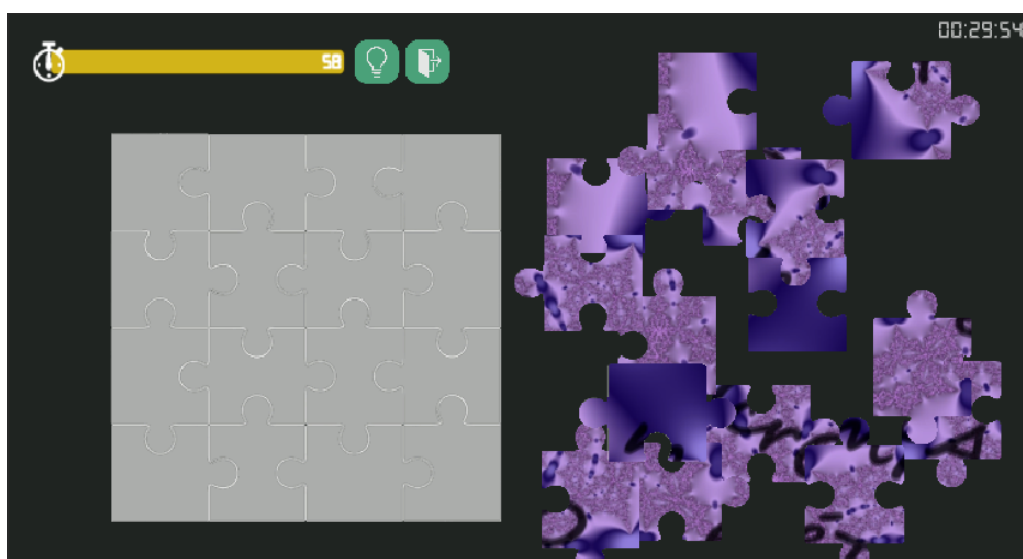


Figura 4.4: Exemplo de um “puzzle” implementado no caso de estudo

4.1.3.3 “Quiz”

O “Quiz” funciona da seguinte maneira:

- antes de iniciar o *quiz* é descarregado todas as perguntas e respostas de um ficheiro JSON.
- só se passa para a próxima questão quando acertar a resposta.
- sempre que se clica numa resposta é dado um *feedback*, sendo este definido também no ficheiro JSON.
- o mini jogo termina quando acertar todas as respostas ou quando o tempo termina ou quando se clicar no botão de sair.

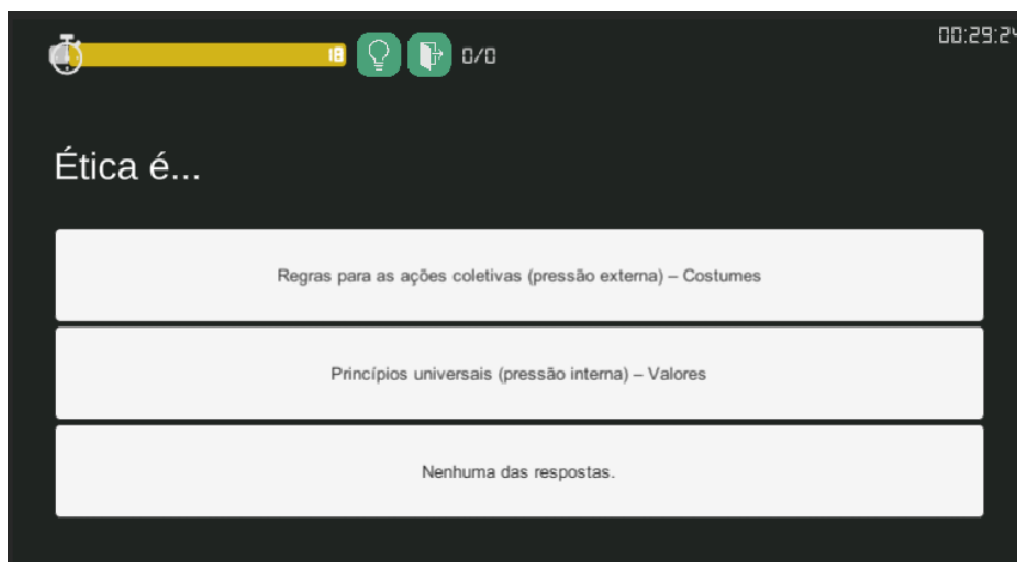


Figura 4.5: Exemplo de um “quiz” implementado no caso de estudo

4.1.3.4 “Label”

O “Label” funciona da seguinte maneira:

- antes de iniciar o mini jogo é carregado a imagem, posicionado os espaços em branco (*slots*) e criado a lista das etiquetas definidas no ficheiro JSON.
- sempre que se pega numa peça e depois se larga é calculado o slot mais perto e se essa distância é inferior que 1 o centro da peça é ajustado ao centro do slot, caso contrário volta para a posição inicial.
- quando é posto uma peça num slot já ocupado por outra peça, essa é substituída e volta para a sua posição inicial.
- o mini jogo termina quando o tempo acaba ou quando as peças estão todas no lugar certo ou quando se clicar no botão de sair.

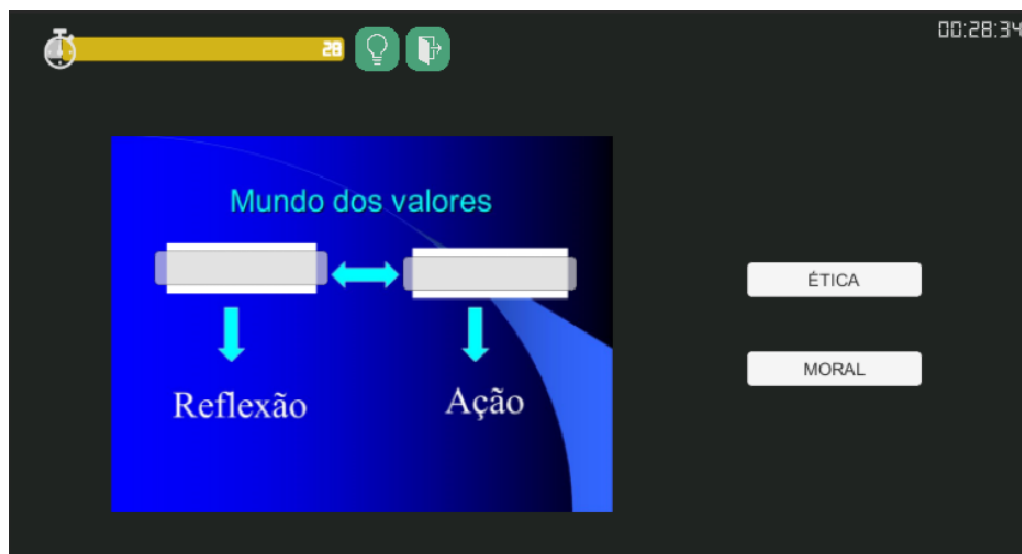


Figura 4.6: Exemplo de um “label” implementado no caso de estudo

4.1.4 Ficheiro JSON

Cada mini jogo tem o seu próprio ficheiro JSON onde está contido a informação. No ficheiro temos os parâmetros globais, obrigatório em todos mini jogos, e os parâmetros particulares, pertencentes a uma certa categoria de mini jogo. No anexo C está apresentado um exemplo de cada categoria de mini jogo.

Parâmetros Globais:

- **introTitle:** nome do título do mini jogo
- **introText:** introdução do mini jogo

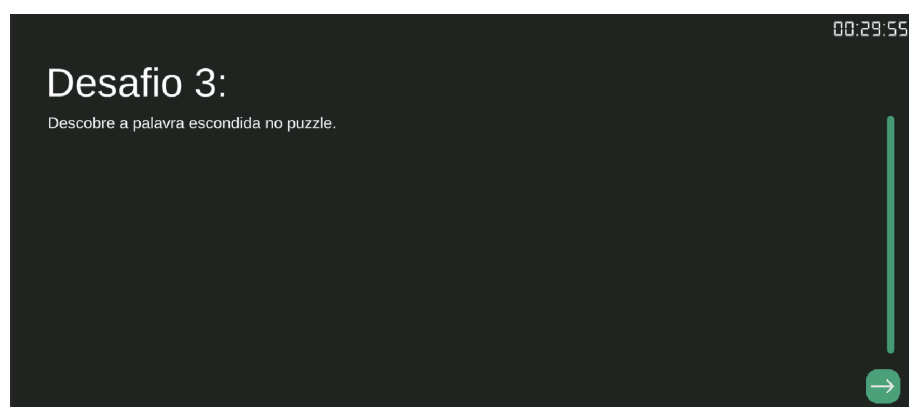


Figura 4.7: Painel de introdução do mini jogo - exemplo

- **finalIntroTextIfWin:** mensagem que aparece num panel após de sair do mini jogo quando termina com sucesso e antes de voltar para a sala do jogo.

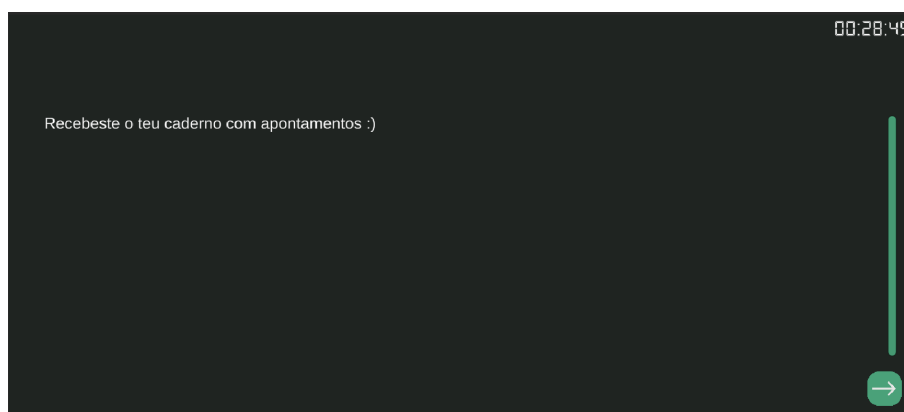


Figura 4.8: Painel após sair do mini jogo se ganhar - exemplo

- **endGameSucceededFeedbackText:** mensagem que aparece quando o jogador ganha o mini jogo sem ajuda
- **endGameSucceededWithHelpFeedbackText:** mensagem que aparece quando o jogador ganha o mini jogo com ajuda
- **endGameFailedFeedbackText:** mensagem que aparece quando o tempo do mini jogo termina

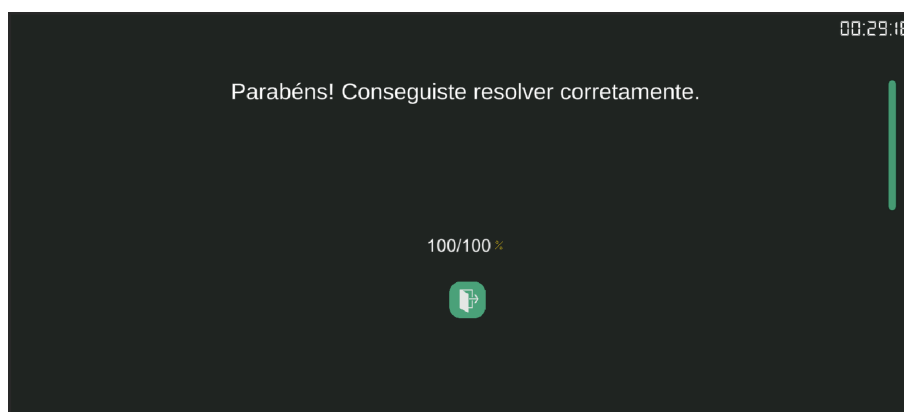


Figura 4.9: Painel de fim do mini jogo - exemplo

- **hintButtonFeedback:** mensagem que aparece quando o rato está por cima do botão de ajuda
- **exitButtonFeedback:** mensagem que aparece quando o rato está por cima do botão de sair do mini jogo
- **timeLimitInSeconds/gameTimer:** tempo do mini jogo em segundos
- **pointsRemovedForAskingHint:** pontuação descontada quando se pede ajuda

- **penaltyTimeFail:** tempo descontado (tempo global do jogo) quando o tempo do mini jogo termina
- **penaltyTimeHelpNeeded:** tempo descontado (tempo global do jogo) quando pede uma ajuda durante o mini jogo
- **penaltyTimeExit:** tempo descontado (tempo global do jogo) quando sai do mini-jogo

4.1.4.1 “Puzzle”

Parâmetros particulares:

- **imagePath:** caminho da imagem
- **puzzlePositions:** lista das posições das peças aleatórias, posição inicial das peças, tamanho da lista tem que ser igual ao número de peças do *puzzle*
 - **x:** coordenada x da posição da peça
 - **y:** coordenada y da posição da peça

Estes parâmetros são guardados nas variáveis das classes da figura 4.10.

É apresentado um exemplo na figura C.2.

```
public class PuzzleConfiguration : MiniGameData
{
    public string imagePath;
    public int gameTimer;
    public List<Position> puzzlePositions;
}

public class Position
{
    public float x;
    public float y;
}
```

Figura 4.10: Classes de configuração do “Puzzle”

4.1.4.2 “Label”

Parâmetros particulares:

- **imagePath:** caminho da imagem
- **pointsAddedForCorrectAnswer:** pontuação de cada resposta correta
- **puzzleItems:** lista dos nomes das etiquetas
- **puzzleSlots:** lista dos espaços vazios

- **correctItem:** nome da etiqueta correta para espaço
- **position:** posição dos slots, pedido as coordenadas x e y

Estes parâmetros são guardados nas variáveis das classes da figura 4.11.

É apresentado um exemplo na figura C.1.

```
public class ConfigGame : MiniGameData
{
    public string imagePath;
    public int pointsAddedForCorrectAnswer;
    public int pointsRemovedForAskingHint;
    public int timeLimitInSeconds;
    public List<puzzleItem> puzzleItems;
    public List<puzzleSlot> puzzleSlots;
}

public class puzzleItem
{
    public string name;
}

public class puzzleSlot
{
    public string correctItem;
    public Vector2 position;
}
```

Figura 4.11: Classes de configuração do “Label”

4.1.4.3 “Quiz”

Parâmetros particulares:

- **imagePath:** caminho da imagem
- **pointsAddedForCorrectAnswer:** pontuação de cada resposta correta
- **puzzleItems:** lista dos nomes das etiquetas
- **puzzleSlots:** lista dos espaços vazios
 - **correctItem:** nome da etiqueta correta para espaço
 - **position:** posição dos slots, pedido as coordenadas x e y

Estes parâmetros são guardados nas variáveis das classes da figura 4.12.

É apresentado um exemplo na figura C.3.

4.1.5 Inquérito

O inquérito é uma componente opcional no jogo. Este pode ser adicionado no início, no final do jogo ou em ambos conforme as suas necessidades do educador.

```
public class GameData : MiniGameData
{
    public RoundData[] allRoundData;
}

public class RoundData
{
    public string name;
    public int timeLimitInSeconds;
    public int pointsAddedForCorrectAnswer;
    public int pointsAddedForIncorrectAnswer;
    public QuestionData[] questions;
}

public class QuestionData
{
    public string questionText;
    public AnswerData[] answers;
}

public class AnswerData
{
    public string answerText;
    public bool isCorrect;
    public string feedback;
}
```

Figura 4.12: Classes de configuração do “Quiz”

O inquérito pode ser composto por perguntas de escolha múltipla e respostas abertas, sendo estas escritas num ficheiro JSON com a estrutura do exemplo do anexo D e depois são guardados nas variáveis das classes da figura 4.13.

Parâmetros do ficheiro JSON:

- **questions:** vetor de perguntas de escolha múltipla
 - **questionText:** pergunta da escolha múltipla
 - **answers:** vetor de respostas
- **openQuestions:** vetor de perguntas de resposta aberta

Modo de funcionamento:

- só avança para a pergunta seguinte quando a presente pergunta estiver respondida.
- sempre que muda de pergunta é inserida/atualizada a resposta da pergunta anterior na base de dados.
- só é possível submeter o inquérito após de ter respondido a tudo.

```

public class SurveyData
{
    public SurveyQuestions[] questions;
    public string[] openQuestions;
}

public class SurveyQuestions
{
    public string questionText;
    public string[] answers;
    public string selectedAnswer;
    public int selectedAnswerIdx;
}

```

Figura 4.13: Classes de configuração do Questionário

4.2 Base de dados

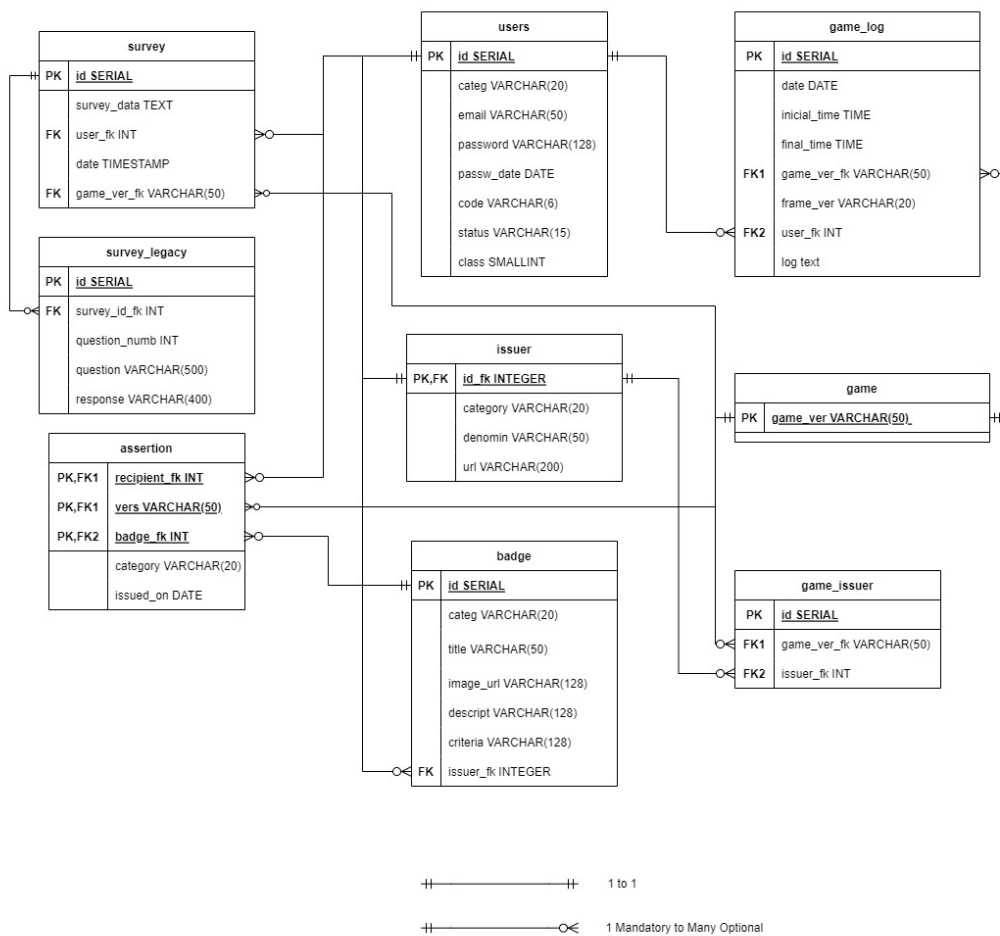


Figura 4.14: Diagrama de Base de Dados

A base de dados (BD) escolhida para o armazenamento e organização das informações recolhidas foi o PostgreSQL, por ser uma das BD disponibilizadas pela faculdade e por estar mais familiarizada.

Na figura 4.14 temos um diagrama de BD onde estão representadas as tabelas criadas e as relações entre elas. Em cada tabela temos o nome da tabela, os nomes e os tipos das colunas, e as restrições de ser uma chave primária e/ou estrangeira.

A partir deste diagrama e o anexo B é possível ter uma noção de como estão organizadas as informações.

4.3 Aplicação web

A aplicação *web* foi desenvolvida com o objetivo principal de desenvolver um sistema aberto de medalhas em que os estudantes poderão ter acesso as medalhas conquistadas nos jogos criados pela *framework* desenvolvida, e os educadores poderão criar medalhas e visualizar os premiados das medalhas emitidas por eles. Outra funcionalidade desta aplicação *web* é disponibilizar aos educadores as informações registadas durante os jogos jogados pelos estudantes e as respostas dos inquéritos.

Para poder ter acesso às informações disponibilizadas pela aplicação *web* houve a necessidade de criar um sistema de autenticação de modo a poder distinguir os utilizadores, mostrado assim apenas as informações que lhes pertencem.

As páginas da aplicação web são implementadas conforme os mockups apresentados na secção 3.4.2.

4.3.1 Sistema de autenticação

O utilizador para entrar na aplicação tem que se autenticar para isso tem que ter uma conta criada e verificada.

- **Criar Conta**

Para criar uma conta o utilizador tem que preencher os campos “Email Address”, “Password” e “Confirm Password” (fig.3.9a). Clicando no botão “Signup” é enviado o pedido ao servidor (anexo A.3.1.2) . No servidor são feitas duas verificações:

- Endereço da conta introduzida não existe;
- As palavras-passes coincidem.

Se validar ambas as condições a conta é criada com sucesso, é enviada uma mensagem para o email da conta criada com o código de verificação e redirecionada para a página Código de Verificação (fig.3.11a). Caso contrário é disparado uma mensagem de erro no campo “Error message” com a descrição do respetivo erro encontrado (fig.3.9b). Depois de verificar a conta esta está pronta para ser usada sendo redirecionada diretamente para a página inicial (fig.3.16c).

• Iniciar Sessão

O utilizador depois de ter uma conta criada pode iniciar sessão preenchendo os campos “Email” e “Password” (fig.3.8a). Ao clicar no botão “Login” este envia um pedido de autenticação ao servidor (anexo A.3.1.1). São feitas as seguintes verificações:

- Todos os campos estão preenchidos;
- Endereço da conta introduzida existe;
- Palavra-passe correta.

Se os dados introduzidos estiverem corretos, a conta é autenticada com sucesso e a página é redirecionada para a página inicial, em que esta difere de classe para classe (do utilizador). Se a autenticação der negativo, ou seja, alguma verificação falhou, é exibido uma mensagem de erro no campo “Error message” (fig.3.8b).

4.3.2 Menu

O menu da aplicação *web* depende da classe da conta autenticada como já foi mostrado nos *mockups* da figura 3.13.

Na figura 4.15 está apresentada o código implementado para os diferentes menus, neste código temos ativo (parâmetro *active*) a página inicial de cada classe de utilizador, podendo concluir que a página inicial da classe 2 é a página “Users” e para a classe 1 e 0 é a página “Badges”.

```
<div class="navbar">
  <div class="navbar-brand">Edscape</div>
  <?php
    if($user_class == 2){
      echo "<div class='topnav'>";
      echo "<a class='active topnav-btn' href='\"home.php\"'>Users</a>";
      echo "<a class='topnav-btn' href='\"listBadges.php\"'>Badges</a>";
      echo "<a class='topnav-btn' href='\"listGames.php\"'>Games</a>";
      echo "<a class='topnav-btn' href='\"../actions/actionLogout.php\"'>Logout</a>";
      echo "</div>";
    }else if($user_class == 1){
      echo "<div class='topnav'>";
      echo "<a class='active topnav-btn' href='\"listBadges.php\"'>Badges</a>";
      echo "<a class='topnav-btn' href='\"listGames.php\"'>Games</a>";
      echo "<a class='topnav-btn' href='\"../actions/actionLogout.php\"'>Logout</a>";
      echo "</div>";
    }else{
      echo "<div class='topnav'>";
      echo "<a class='active topnav-btn' href='\"listBadges.php\"'>Badges</a>";
      echo "<a class='topnav-btn' href='\"../actions/actionLogout.php\"'>Logout</a>";
      echo "</div>";
    }
  ?>
</div>
```

Figura 4.15: Implementação do menu

4.3.3 Sistema aberto de medalhas

O sistema aberto de medalhas está integrado na página “Badges” do menu.

Neste sistema é permitido listar as medalhas da conta autenticada, criar medalhas e atribuir medalhas.

4.3.3.1 Listar medalhas

Dependendo da classe da conta autenticada o método de obter a respetiva lista é diferente.

- Classe 2: o administrador tem direito ao acesso de todas as medalhas emitidas até ao momento.

Método: `getAllBadges()` - retorna a lista de todas as medalhas guardadas na tabela “badge” da BD.

- Classe 1: o educador tem acesso às medalhas que emitiu.

Método: `getBadgeByIssuerId($user_id)` - retorna a lista de medalhas em que o *id* do emissor da tabela “badge” da BD é igual ao parâmetro *user_id*.

- Classe 0: o estudante tem acesso às medalhas que conquistou.

Métodos: `getAssertionByRecipient($user_id)` - retorna a lista das medalhas em que o *id* do recebedor da tabela “assertion” da BD é igual ao parâmetro *user_id*. Com este retorno temos o *id* das medalhas conquistadas (*\$badge_id*).

Para poder ter acesso aos dados da medalha é chamada outro método:

`getBadgeById($badge_id)` - retorna uma linha com as informações da medalha com o *id* igual ao parâmetro *badge_id*.

```
if($class == 2){
    $result = getAllBadges();
}else if($class == 1){
    $result = getBadgeByIssuerId($user_id);
}else if($class == 0){
    $result = getAssertionByRecipient($user_id);
}
```

Figura 4.16: Código para obter lista de medalhas

Quando o utilizador clica numa das medalhas é redirecionada para a página de detalhes (fig.3.17) da respetiva medalha. Se a conta for da classe 0 a página mostra os dados da medalha, a data em que conquistou, o nome do emissor e o nome do premiado (fig.3.17a). Se a conta for da classe 1 ou 2 a página mostra os dados da medalha, o nome do emissor e a lista dos premiados (fig.3.17b). A alternância da informação entre a “Criteria” e “Awards” é implementada a partir do *Ajax*, em que ao clicar numa deles é enviada um pedido através do objeto *XMLHttpRequest* para o API (anexo A.2.3 e A.2.4).

```

var request = new XMLHttpRequest();
request.onreadystatechange = function() {
    if(this.readyState == 4 && this.status == 200){
        console.log("responseText:" + this.responseText);
        if(!criteria.classList.contains('active')){
            criteria.classList.add('active');
            awards.classList.remove('active');
        }
        content.innerHTML = this.responseText;
    }
}
request.open ("GET", url, true) //true => Asynchronous request
request.send();

```

Figura 4.17: Pedido Assíncrono

4.3.3.2 Criar medalha

A operação de criar uma medalha apenas os educadores (classe 1) têm permissão.

O formulário de criar uma medalha (fig.3.18) é acedido a partir do botão “New Badge” na página “Badges” (fig.3.16b). Para a medalha ser criada com sucesso é preciso de cumprir as seguintes restrições:

- o nome da medalha está limitado a 50 carateres
- a descrição e os critérios da medalha está limitado a 128 carateres
- obrigatório de escolher uma imagem
- obrigatório de preencher todos os campos

Método: `insertNewBadge($bg_name, $bg_image, $bg_description, $bg_criteria, $issuer_id)`

4.3.3.3 Atribuir medalhas

O processo de atribuição de medalhas é feito no final de cada jogo, em que cada medalha está associada a uma variável e o valor desta é recebida no envio do último *log* do jogo (anexo A.1.3.3).

Para verificar se uma medalha é atribuída, primeiro comparamos se o valor da variável é o que queremos, caso seja verificamos se a respetiva medalha já foi conquistada pelo jogador na presente versão do jogo, se ainda não tiver sido premiado é-lhe atribuída a medalha. Na figura 4.18 está representada um exemplo de código de atribuição de uma medalha.


```

if($isFinishedInTime == "true"){
    $result = getAssertionByRecipientAndBadge($student_id,"1",$game_ver);
    if(pg_numrows($result)<=0){
        $badgeswon = "1";
        if(insertNewAssertion($student_id,"1",$date,$game_ver)){
            $assertion = $assertion . "new 1,";
        }else{
            $assertion = $assertion . "faield 1,";
        }
    }
}

```

Figura 4.18: Código que verifica se conquistou a medalha-exemplo

Para o caso de estudo apresentado no capítulo 5, o sistema apenas foi implementado para três medalhas e cada uma delas está associado a um *id*:

- id=1 : medalha de terminar o jogo a tempo
- id=2 : medalha de terminar o jogo depois do tempo
- id=4 : medalha de não ter pedido ajuda durante o jogo

4.3.4 Página “Users”

A página “Users” (fig.3.14) foi implementada com o objetivo de o administrador poder gerar as contas dos utilizadores. Nesta página o administrador tem acesso a uma lista com todas as contas, pode exportar essa lista, atualizar as informações do emissor e fazer *upgrade* de uma conta passando da classe 0 (estudante) para classe 1 (emissor/educador).

A lista de utilizadores é obtida chamando o método `getAllUsers()` este faz um pedido à BD e retorna todas as contas armazenadas na tabela “users”. À frente de cada elemento da tabela tem um botão de atualização com funções diferentes, se a conta for da classe 1 então o botão é do tipo “Update” caso seja da classe 2 é do tipo “Upgrade”. Clicando num dos botões é feito um pedido ao API (anexo A.3.3) para fazer a respetiva operação.

4.3.5 Página “Game”

A página “Game” (fig.3.15) foi implementada para os administradores (classe 2) e os educadores (classe 1) terem acesso às informações recolhidas dos jogos jogados. Nesta página eles podem filtrar as informações mostradas na tabela, exportar a tabela e remover *logs*.

A tabela está predefinida para mostrar os *logs* de todas as versões do jogo no caso do administrador, e no caso dos educadores mostra todas as versões dos jogos que criou. A componente dos filtros é composto por dois filtros, um filtra a categoria de informação podendo ser *log*, *survey* ou *survey_legacy*, e o outro filtra a versão do jogo. Clicando no botão “show table” é enviado um pedido ao API (anexo A.4.1) e retorna o código HTML da nova tabela. Na figura 4.19 é mostrada um exemplo de uma tabela filtrada com a categoria *log* e a versão *Escape2Ethics101Test*.

log Escape2Ethics101Test show table

Export the table Remove logs

List of all logs (Escape2Ethics101Test)

Game Id	Framework Version	Date	Initial Time	Final Time	Student	Log
Escape2Ethics101Test	0.25	2022-06-02	22:27:56		98regil@gmail.com	[6/2/2022 10:27:56 PM] [Time] Time ended. [6/2/2022 10:27:56 PM] [Info/Instructions] Show text. [6/2/2022 10:27:59 PM] [Intro] Show text. [6/2/2022 10:28:01 PM] Game starts. [6/2/2022 10:28:03 PM] [Movement] Automatic movement started with (6.1, 0.0, 6.1) as final coordinates. [6/2/2022 10:28:03 PM] [Movement] Automatic movement started with (6.7, 0.0, 7.0) as final coordinates. [6/2/2022 10:28:06 PM] [Movement] Automatic movement started with (8.1, 0.8, 6.2) as final coordinates. [6/2/2022 10:28:06 PM] [Clues] Show clue after specific time if player does not own, Item: , clue: Procura a informação primeiro
Escape2Ethics101Test	0.25	2022-06-02	22:14:05	22:14:19	98regil@gmail.com	[6/2/2022 10:14:05 PM] [Time] Time ended. [6/2/2022 10:14:06 PM] [Info/Instructions] Show text. [6/2/2022 10:14:08 PM] [Intro] Show text. [6/2/2022 10:14:11 PM] Game starts. [6/2/2022 10:14:13 PM] [Movement] Automatic movement started to reach Door. [6/2/2022 10:14:16 PM] [Clues] Show clue after specific time if player does not own, Item: , clue: Procura a informação primeiro antes dos desafios. [6/2/2022 10:14:17 PM] [Item Door] Starts interacting with item. [6/2/2022 10:14:17 PM] [Item Door] Moved. [6/2/2022 10:14:19 PM] [Item Door] Made player escape. [6/2/2022 10:14:19 PM] Escapes

Figura 4.19: Exemplo de uma filtragem

O botão “remover logs” basicamente faz um pedido à BD para eliminar os *logs* do jogo com mais de um ano (anexo A.4.2).

4.3.6 Outras Operações

• Exportar Dados

Os utilizadores têm a possibilidade de exportar as informações das tabelas e lista de medalhas para um ficheiro CSV, clicando nos botões de “export” é enviado um pedido ao API do anexo A.5. Na figura 4.21 temos um exemplo de uma exportação da lista de medalhas para um ficheiro CSV.



Figura 4.20: Botões de exportar dados

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1																					
2																					
3																					
4																					
5																					
6																					
7																					
8																					
9																					

(a) Ficheiro CSV exportado

	A	B	C	D	E	F
1	Column1	Column2	Column3	Column4	Column5	Column6
2						
3	Id	Title	Image_url	Descript	Criteria	Issued_by
4	1	Edscape - Successfully escaped	badge1.png	This badge is awarded to people who attended "Escape" and escaped in set time.	This badge is awarded to people who escaped in set time.	Armando Jorge Sousa
5	2	Edscape - Escaped	badge1.png	This badge is awarded to people who attended "Escape" and escaped but after time.	This badge is awarded to people who escaped but after time.	Armando Jorge Sousa
6	3	Gestão de projetos	5fc68357e77385f9f472ffe_17jan2020-sharingandcaring-image1.png	Descrição da medalha- gestão de projetos.	Critérios para obter a medalha: 1. 2.	celina
7	4	Edscape - Without Hint	badge2.jpeg	This badge is awarded to people who attended the "Escape" and escaped without hint.	This badge is awarded to people who don't use hints in the game.	Armando Jorge Sousa
8						

(b) Ficheiro CSV importado para o excel

Figura 4.21: Exemplo de uma exportação

- **Código embutido**

Os utilizadores também têm a possibilidade de copiar o código embutido da lista de medalhas e adicionar no corpo de outro programa/página. Apenas precisa de clicar no *icon* copiar e o código fica gravada na área de transferência do sistema.

Embed code: 

Figura 4.22: Código embutido

Capítulo 5

Caso de Estudo

5.1 Contexto

Para testar o trabalho desenvolvido no Projeto “EdScape” foi criado um jogo / caso de estudo em ética para a engenharia. Este caso de estudo foi produto de uma colaboração da equipa do Projeto “EdScape”.

Primeiro criou-se um esboço de um guião do caso de estudo. A seguir, tendo em conta a esse guião criaram-se os ficheiros JSONs com as informações do jogo. Por fim, a *framework* percorre os ficheiros e demonstra o jogo criado. Após o jogo ser criado, versão “Escape2Ethics101Test”, foi testada por um grupo de estudante.

No anexo [E](#) está representado o ficheiro JSON que contém a narrativa, as dicas, as ligações aos inquéritos (ficheiro JSON, anexo [D](#)) e as informações dos objetos utilizados e a suas ligações a outros ficheiros JSONs (mini jogos ou PDFs). No anexo [C](#) é representada um exemplo do ficheiro JSON para cada tipo de mini jogo dos vários ficheiros criados.

O jogo “Escape2Ethics101Test” é composto por apenas uma sala onde o jogador pretende escapar dentro de um determinado tempo. Na sala tem espalhado diversos objetos em que uns contém desafios e outros contém PFDs com as matérias. O jogo termina quando o tempo acaba ou quando o jogador escapa da sala, este só consegue escapar caso tenha resolvido todos os desafios com sucesso.

5.2 Resultados

Os resultados deste caso de estudo foram exportadas a partir da aplicação *web*, desenvolvida nesta dissertação, após iniciar sessão com a conta educadora que criou o tal jogo.

A partir dos dados exportados da tabela dos *logs* podemos concluir que o jogo foi testado por 14 estudantes e que apenas metade deles jogaram até ao fim do jogo.

A seguir será feita uma análise das respostas dadas pelos estudantes nos inquéritos iniciais e finais do jogo sobre o tema, as perguntas dos inquéritos estão apresentadas no anexo [F](#).

No inquérito inicial é feita uma breve sondagem dos conhecimentos que os estudantes têm sobre alguns conceitos relacionados com a ética. Na figura 5.1 está representado, em gráfico, as respostas dos estudantes para cada pergunta. Podemos notar que todos os estudantes responderam ao inquérito inicial.



Figura 5.1: Respostas do inquérito inicial - antes do jogo

As respostas foram respondidas numa escala de 1 a 5 com os seguintes significados:

- 1 - Nunca ouvi falar e não sei nada sobre o assunto

- 2 - Já ouvi falar, mas não sei quase nada sobre o assunto
- 3 - Tenho apenas conhecimentos básicos sobre o assunto
- 4 - Tenho alguns conhecimentos, mas não domino o assunto
- 5 - Tenho muitos conhecimentos e domino o conceito

A partir dos gráficos é fácil de concluir que entre os 8 conceitos: ética, deontologia, moral, microética, ética utilitarista, macroética, ética das virtudes e ética deontológica, destacam-se a ética e a moral por ser os mais conhecidos, mas apenas estão na gama entre ter conhecimentos básicos e alguns conhecimentos.

No inquérito final é feita uma sondagem sobre os conhecimentos dos mesmos conceitos perguntados no inquérito inicial. Na figura 5.2 está representado os gráficos com as respostas relacionadas a cada conceito, é de notar de cada conceito é composto por duas perguntas, em que as barras a azul são perguntas sobre o grau de conhecimento após o jogo e as barras a laranja são perguntas sobre quanto o jogo ajudou a adquirir conhecimento sobre os respetivos conceitos. Podemos observar que apenas 7 estudantes responderam ao inquérito final como já foi referido anteriormente metade dos estudantes desistiram ao meio do jogo.

Escala dos itens a azul:

- 1 - Nunca ouvi falar e não sei nada sobre o assunto
- 2 - Já ouvi falar, mas não sei quase nada sobre o assunto
- 3 - Tenho apenas conhecimentos básicos sobre o assunto
- 4 - Tenho alguns conhecimentos, mas não domino o assunto
- 5 - Tenho muitos conhecimentos e domino o conceito

Escala dos itens a laranja:

- 1 - Nenhum – não se alteraram
- 2 - Muito pouco – sei quase o mesmo que antes
- 3 - Algum – aprendi algumas coisas relevantes
- 4 - Bastante – aprendi várias coisas relevantes
- 5 - Muito – aprendi muitas coisas que desconhecia

Nos gráficos da figura 5.2 destacam-se alguns casos em que o grau de conhecimento e o grau de aquisição diferenciação de 2 ou mais níveis isto poderá acontecer quando o jogador já tem conhecimentos sobre o assunto e as matérias introduzidas no jogo não são o suficiente para ele

absorver novos conhecimentos. Nestes casos o jogo pode ser usado para consolidar a matéria em vez aprender.

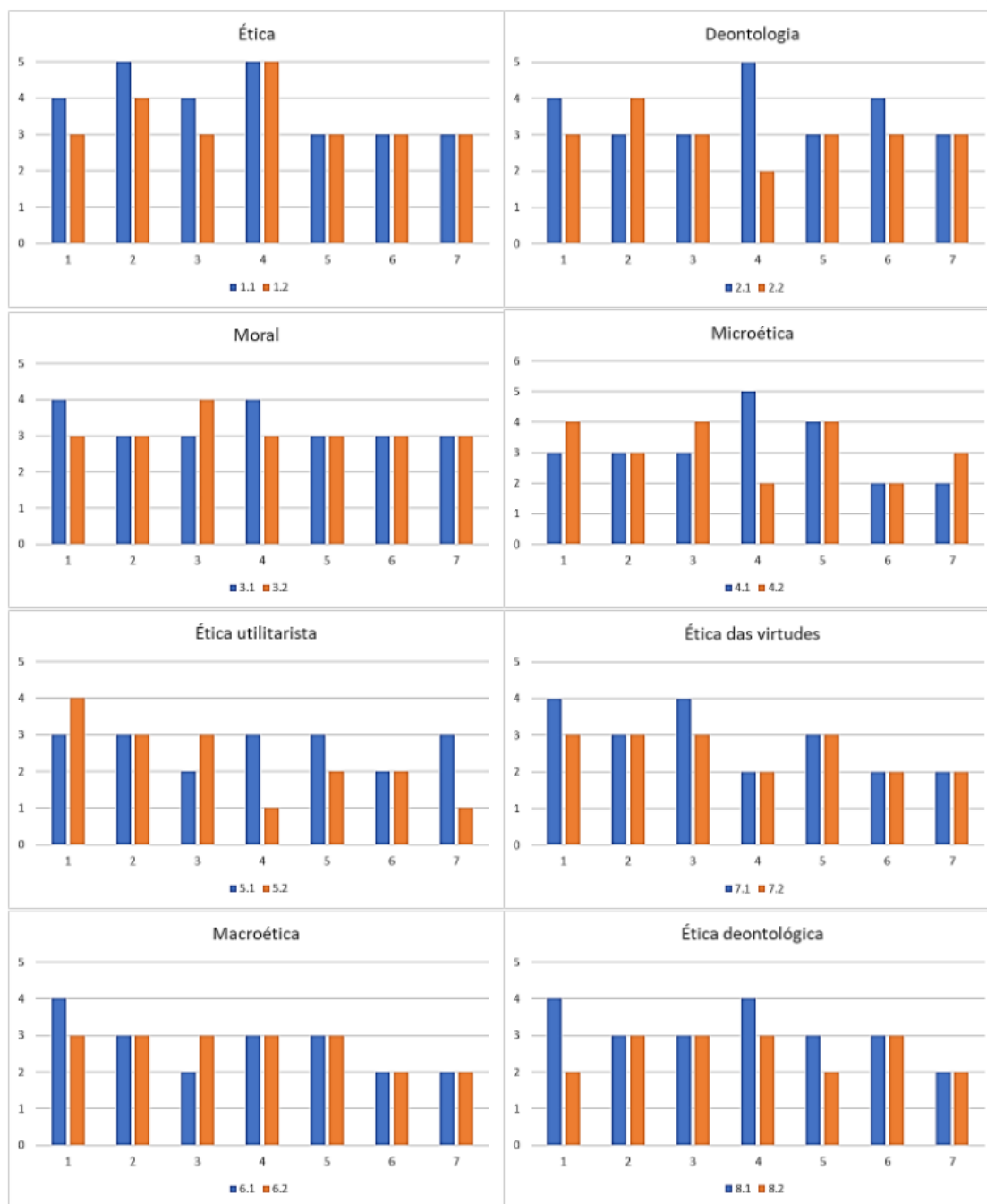


Figura 5.2: Respostas do inquérito final - após o jogo

Comparando os gráficos de média da figura 5.3 é possível observar que houve uma ligeira subida no grau dos conhecimentos dos conceitos podendo concluir que o jogo criado ajudou os estudantes, pouco ou algum, a adquirir os conhecimentos integrados no jogo, o que é um bom começo. Outro ponto que comprova esta conclusão são as respostas dadas às perguntas sobre o

quanto o jogo ajudou a absorver os conhecimentos (linha laranja do gráfico do inquérito final), é de notar que numa escala de 1 a 5 as médias foram acima de 2,5.

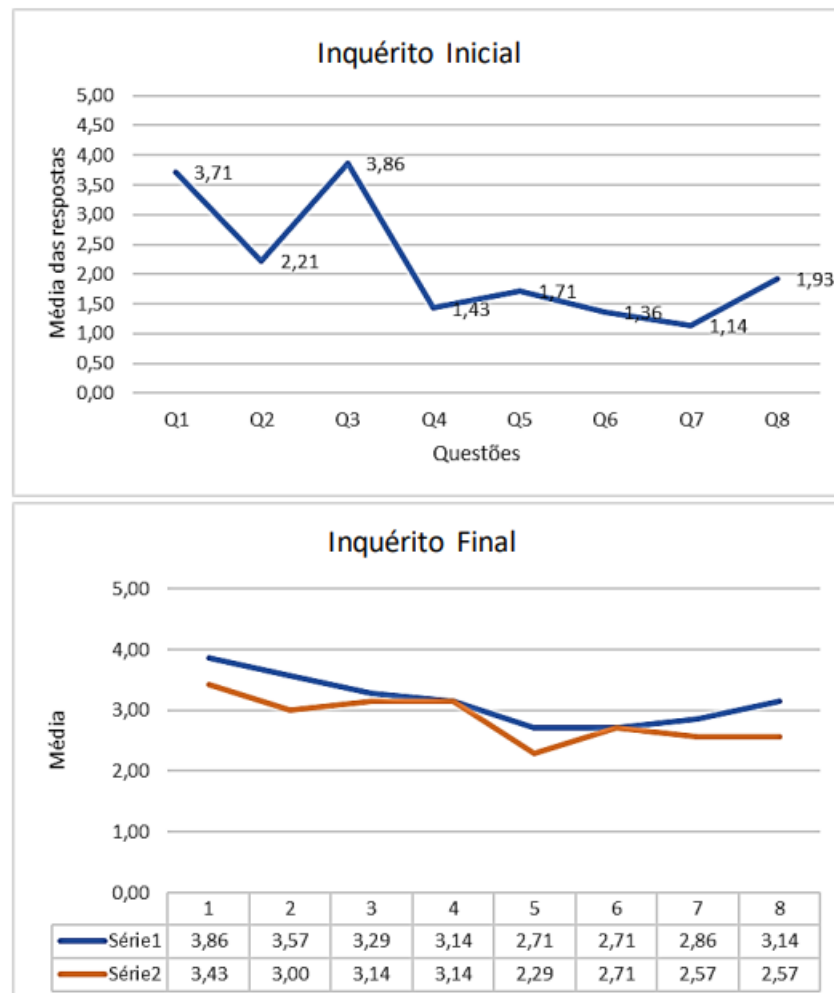


Figura 5.3: Média das respostas

5.3 Sugestões

Os estudantes após terem testado o jogo deram alguns *feedbacks* e sugestões de melhoria:

- gostaram da ideia das medalhas,
- gostaram de jogar e acharam uma forma interessante adquirir os conceitos,
- uns acharam que a diversidade dos mini jogos era um aspeto positivo outros acharam que deveria haver mais diversidade ainda,
- o botão para iniciar o mini jogo devia ser maior,

- melhorar os documentos: ser mais clara e organizada, em vez de *slides*, usar outros métodos de aprendizagem mais apelativo ao ser humano,
- a opção “ainda não sei” deveria estar em cada mini jogo e deveria ser seguida de uma mensagem que incentivasse a leitura dos documentos (por exemplo: “Na secretária há informações sobre este tópico”),
- ser mais desafiante.

Após revisão de todo este *feedback* temos pistas para admitir que os jogadores consideram a mecânica de todos os jogos adequada e intuitiva (pois não houve queixas nem sugestões de melhoria a nível da interface de nenhum dos mini jogos). Alguns comentários mencionam mesmo que os mini jogos são uma das partes positivas deste caso de estudo.

Capítulo 6

Conclusão e Trabalho futuro

6.1 Conclusões e contribuições

Ao longo deste documento, é apresentado uma introdução e o desenvolvimento de uma parte do projeto “EdScape”.

Nesta dissertação são implementados: (i) mini jogos; (ii) sistema de condecorações; (iii) o sistema de registo em Base de Dados dos eventos do jogo.

Os mini jogos desenvolvidos foram o "Puzzle", o "Quiz" e o "Label", apresentados na secção 4.1.2, implementados com a ferramenta Unity3D e integrados na *framework* do jogo como desafios. Para cada jogo, é apresentada a arquitetura, os *mockups* e a implementação.

O sistema de condecorações inclui autenticação de utilizadores e recebe medalhas do jogo. O sistema de autenticação e gestão de medalhas é utilizado dentro do jogo e pode ser acessível também via interface *web*. O desenvolvimento teve por base a norma “Open Badges”.

Foi ainda desenvolvido um sistema de registo persistente em base de dados de todos os eventos no jogo, o que permitirá futuramente analisar os desenvolvimentos no jogo.

O projeto foi validado com um caso de estudo e obteve-se resultados positivos dado que o jogo criado ajudou os estudantes a absorver conhecimentos sobre a matéria apresentada como podemos ver nos resultados obtidos no capítulo 5.

Esta dissertação contribui para o projeto “EdScape” tendo em conta que um jogo *escape room* é indispensável de ter desafios, ou seja, mini jogos. Outra contribuição foi o facto de o sistema de atribuição de medalhas ser um ponto atraente, enriquecendo a potencialidade do projeto, podendo ser uma motivação extrínseca para os estudantes, levando-os a jogar com o objetivo de ganhar as medalhas e em simultâneo estes absorvem conhecimentos sobre a matéria.

No âmbito da área da educação, espera-se que o projeto “EdScape” venha a contribuir como uma ferramenta que gera jogos que sejam uma ferramenta de ensino e aprendizagem ativa.

Grande parte da informação presente nos mini jogos é facilmente alterável, mesmo sem ter o Unity3D instalado, permitindo assim uma facilidade de utilização adicional (um educador pode rapidamente implementar pequenas alterações aos jogos gerados pelo projeto "EdScape").

O caso de estudo produzido no decurso deste trabalho de dissertação, denominado "Escape2Ethics101Test" pode ser jogado em:

<https://fe.up.pt/~efeupinho/jogo01>

O código-fonte do projeto EdScape é livremente acessível em:

<https://gitlab.com/asousa-feup/edscape>

6.2 Trabalho futuro

Em trabalhos futuros, deve ser melhorado o sistema de atribuição das medalhas, na forma como é feita a atribuição e a que versões de jogos as medalhas estão relacionadas. Para além desse melhoramento deve ser criado mais diversidades de mini jogos para os educadores poderem estar menos limitados nos poucos tipos de mini jogos existentes quando desenvolvem os desafios dos seus jogos. Quanto aos requisitos enumerados no capítulo 3, a sala de espera (R_E03), a possibilidade de ser jogado em multi-jogadores (R_J04) e receber uma classificação (R_E08) são requisitos que ficaram por implementar o que deve ser implementado em trabalhos futuros.

Anexo A

API

A.1 Pedidos do Unity

A.1.1 Conta do Utilizador

Método: POST /web_badges/unity/unityUserData.php

Nesta API, dependendo dos parâmetros de entrada são executadas funções diferentes.

A.1.1.1 Verificar a conta de Iniciar Sessão

Descrição: Verifica se o email existe na base de dados, caso exista confirma a palavra-passe se está correta e por fim verifica se a palavra-passe já tem há mais de um ano.

Parâmetros de entrada:

- login (string): indicador do tipo de ação, predefinido
- email (string): email do utilizador
- password (string): password do utilizador

Formato das respostas de retorno (string):

- |success|{mensagem}: passa a todas as verificações, retorna a mensagem de sucesso
- |red-old|{mensagem}: a palavra-passe tem mais de um ano, retorna a mensagem de aviso
- |red|{mensagem}: alguma verificação não se verifica, retorna com a respetiva mensagem de erro

A.1.1.2 Enviar código de verificação

Descrição: Antes de enviar o código é verificado se o email fornecido já foi criado ou não, se não é criado a conta com o estado “notverified” e enviado o código.

Parâmetros de entrada:

- check (string): indicador do tipo de ação, predefinido
- email (string): email da nova conta
- password (string): password da nova conta

Formato das respostas de retorno (string):

- |green|{mensagem}: código de verificação enviado para o endereço com sucesso, retorna mensagem de sucesso
- |red|{mensagem}: alguma verificação não se verifica, retorna com a respetiva mensagem de erro

A.1.1.3 Pedir para criar conta

Definição: Confere se código de verificação está correto, estando correto é atualizado o estado da conta para “verified”, ficando pronta para ser usada.

Parâmetros de entrada:

- signup (string): indicador do tipo de ação, predefinido
- email (string): email da nova conta
- password (string): password da nova conta
- code (string): código de verificação da conta

Formato das respostas de retorno (string):

- |green|{mensagem}: conta criada com sucesso, retorna mensagem de sucesso
- |red|{mensagem}: alguma verificação não se verifica, retorna com a respetiva mensagem de erro

A.1.2 Guardar Inquérito**A.1.2.1 Todas perguntas**

Método: POST `web_badges/unity/unitySurvey.php`

Descrição: Adicionar ou atualizar o inquérito na base de dados.

Parâmetros de entrada:

- first (bool): indica se é a primeira vez que guarda na base de dados

- `user` (string): email do jogador
- `survey` (string): texto com todas as perguntas e respostas do inquérito
- `game_version` (string): id único do jogo (limite de 50 caracteres)
- `issuers` (string): lista dos educadores que criaram o jogo, separados por uma vírgula
- `survey_id` (int): id único do inquérito, obrigatório quando o parâmetro `first=true`.

Nota: Quando `first=true` verifica-se primeiro se a versão do jogo e os `issuers` estão inseridos na base de dados, caso não estejam são adicionados primeiro.

Formato das respostas de retorno (string):

- `|success|{survey_id}|{mensagem}`: inquérito adicionado/atualizado na base de dados com sucesso, retorna o respetivo id e a mensagem de sucesso.
- `|failed|{mensagem}`: falha ao inserir o inquérito na base de dados, retorna a mensagem de erro

A.1.2.2 Uma pergunta

Método: POST `web_badges/unity/unitySurveyLegacy.php`

Descrição: Adiciona a pergunta e a resposta na base de dados ou atualiza a resposta da pergunta, cada pergunta está associada a um inquérito.

Parâmetros de entrada:

- `id` (int): id único da pergunta
- `numb` (int): número da pergunta que corresponde ao inquérito
- `survey_id` (int): o id único do inquérito
- `question` (string): pergunta (limite de 500 caracteres)
- `response` (string): resposta escolhida ou introduzida (limite de 400 caracteres)

Nota: Se `id=-1` então é executado a operação de inserir os dados na base de dados, caso contrário atualiza a resposta da respetiva pergunta.

Formato das respostas de retorno (string):

- `|success|{id}|{mensagem}`: pergunta inserida/atualizada com sucesso, retorna o respetivo id da pergunta e a mensagem de sucesso
- `|fail|{mensagem}`: falha na inserção/atualização da pergunta, retorna a mensagem de erro

A.1.3 Log e Medalhas

Método: POST `web_badges/unity/unityLogAndBadges.php`

A.1.3.1 Primeiro *log*

Descrição: Adicionar o *log* do jogo à base de dados

Parâmetros de entrada:

- user (string): email do jogador
- issuers (string): lista dos educadores que criaram o jogo, separados por uma vírgula
- log (string): texto com os registos do jogo
- date (date): data em que o jogo é jogado, formato YYYY-MM-DD
- initial_time (time): tempo em que o jogo começa, formato hh:mm:ss
- game_version (string): id único do jogo (limite de 50 carateres)
- framework_version (string): versão do *build* (limite de 20 carateres)

Formato das respostas de retorno (string):

- |success|{gameLog_id}|{mensagem}: *log* adicionado na base de dados com sucesso, retorna o id do *log* e a mensagem de sucesso
- |failed|{mensagem}: falha na adição do log na base de dados, retorna a mensagem de erro

A.1.3.2 Log Intermédio

Descrição: Atualizar o *log* do jogo na base de dados.

Parâmetros de entrada:

- gameLog_id (int): id único do *log*
- log (string): texto com os registos do jogo

Formato das respostas de retorno (string):

- |success|{gameLog_id}|{mensagem}: *log* com o respetivo id é atualizado na base de dados com sucesso, retorna o id do *log* e a mensagem de sucesso
- |failed|{mensagem}: falha na atualização do log na base de dados, retorna a mensagem de erro

A.1.3.3 Último *log*

Descrição: Atualizar o *log* do jogo na base de dados.

Parâmetros de entrada:

- gameLog_id (int): id único do *log*
- final_time (time): tempo a que terminou o jogo, formato hh:mm:ss
- log (string): texto com os registos do jogo
- isFinishedInTime (bool): indica se terminou o jogo a tempo (variável de critério de medalhas)
- withoutHint (bool): indica se utilizou ou não ajuda durante o jogo (variável de critério de uma medalha)
- date (date): data em que o jogo é jogado, formato YYYY-MM-DD
- game_ver (string): id único do jogo (limite de 50 carateres)
- user (string): email do jogador

Formato das respostas de retorno (string):

- |{url}|{mensagem}: retorna o endereço da página para ver as medalhas e a mensagem com outras informações

A.2 Medalhas (Web)

A.2.1 Criar nova medalha

Método: POST `web_badges/actions/actionNewBadges.php`

Requer autenticação: Sim

Parâmetros de entrada:

- badge_name (string): nome da medalha (limite de 50 carateres)
- description (string): breve descrição da medalha (limite de 128 carateres)
- criteria (string): critérios necessários para ganhar a medalha. (limite de 128 carateres)
- image_input (file): imagem da medalha

Retorno: Sem retorno.

A.2.2 Obter detalhes da medalha

Método: GET `web_badges/pages/badgeDetails.php?id={badge_id}&version={game_ver}`

Requer autenticação: Sim

Parâmetros de entrada:

- id (int): id único da medalha
- version (string): versão do jogo em que ganhou a medalha, obrigatório se a classe do utilizador for *student*.

Retorno: Sem retorno.

A.2.3 Obter critério da medalha

Método: GET `web_badges/webservices/getCriteriaOfBadge.php?id={badge_id}`

Requer autenticação: Sim

Parâmetros de entrada:

- id (int): id único da medalha

Retorno: retorna os critérios em *string* para o cliente *ajax*.

A.2.4 Obter premiados da medalha

Método: GET `web_badges/webservices/getAwardsOfBadge.php?id={badge_id}`

Requer autenticação: Sim

Parâmetros de entrada:

- id (int): id único da medalha

Retorno: retorna o código HTML (tabela com os premiados) para o cliente *ajax*.

A.3 Utilizador (Web)

A.3.1 Conta

Método: POST `web_badges/actions/controllerUserData.php`

Neste API dependo dos parâmetros de entrada é executado funções diferentes.

A.3.1.1 Iniciar Sessão

Requer autenticação: Não

Parâmetros de entrada:

- login (string): indicador do tipo de ação, predefinido
- email (string): email do utilizador
- password (string): password do utilizador

Retorno:

- se verificar que a palavra-passe tem há mais de um ano redireciona para a página de definir uma nova palavra-passe.
- se verificar que a conta ainda não foi verificada redireciona para a página de código de verificação.
- se passar a todas as verificações é redirecionado para a página inicial.
- se ocorrer algum erro ou não cumprimento de uma verificação é guardado na variável \$errors a mensagem de erro.

A.3.1.2 Criar Conta

Requer autenticação: Não

Parâmetros de entrada:

- signup (string): indicador do tipo de ação, predefinido
- email (string): email da nova conta do utilizador
- password (string): palavra-passe da nova conta do utilizador
- cpassword (string): confirmação da palavra-passe da nova conta do utilizador

Retorno:

- se verificar que a conta não existe e as palavras-passe coincidem é redirecionado para a página de código de verificação da conta.
- se ocorrer algum erro ou não cumprimento de uma verificação é guardado na variável \$errors a mensagem de erro.

A.3.2 Verificar o código de verificação

Requer autenticação: Sim **Parâmetros de entrada:**

- check (string): indicador do tipo de ação, predefinido
- otp (string): *one time password*, código para verificar a conta (limite de 6 carateres)

Retorno:

- se o código estiver certo é redirecionado para a página inicial.
- se ocorrer algum erro ou não cumprimento de uma verificação é guardado na variável \$errors a mensagem de erro.

A.3.2.1 Verificar o email de recuperação da palavra-passe

Requer autenticação: Não **Parâmetros de entrada:**

- check-email (string): indicador do tipo de ação, predefinido
- email (string): email da conta que pretende recuperar a palavra-passe

Retorno:

- se verificar que a conta existe é enviado para o email um código de recuperação e redirecionado para a página do código de recuperação.
- se ocorrer algum erro ou não cumprimento de uma verificação é guardado na variável \$errors a mensagem de erro.

A.3.2.2 Verificar o código de recuperação

Requer autenticação: Não **Parâmetros de entrada:**

- check-reset-otp (string): indicador do tipo de ação, predefinido
- otp (string): *one time password*, código para recuperar a palavra-passe (limite de 6 carateres)

Nota: É necessário do email guardado na variável de sessão - \$_SESSION['email'].

Retorno:

- se o código estiver correto é redirecionado para a página de alterar a palavra-passe.
- se ocorrer algum erro ou não cumprimento de uma verificação é guardado na variável \$errors a mensagem de erro.

A.3.2.3 Alterar Palavra-passe

Requer autenticação: Não **Parâmetros de entrada:**

- change-password (string): indicador do tipo de ação, predefinido
- password (string): nova palavra-passe
- cpassword (string): confirmação da nova palavra-passe da nova

Nota: É necessário do email armazenado na variável de sessão - `$_SESSION['email']`. **Retorno:**

- se verificar que a palavra-passe diferir da antiga e as palavras-passe introduzidas se coincidem é redirecionado para uma página com a mensagem de que foi alterado com sucesso.
- se ocorrer algum erro ou não cumprimento de uma verificação é guardado na variável `$errors` a mensagem de erro.

A.3.3 Upgrade/Update

Método: POST `web_badges/actions/actionUpgradeUser.php`

Requer autenticação: Sim

Parâmetros de entrada:

- id (int): id único do utilizador autenticado
- issuer_name (string): nome do utilizador ou da organização
- url (string): perfil da media social da entidade, seja individual ou institucional. Deve ser um URL/URI acessível via HTTP
- submit (string): indica se executa a operação *update* ou *upgrade*

Retorno: retorna uma mensagem, guardada na variável `$alert`.

A.4 Página Game (Web)

A.4.1 Obter lista

Método: GET `web_badges/webservices/getGameList.php?userId={id}&type={tipo}&version={versão}`

Requer autenticação: Sim

Parâmetros de entrada:

- userId (int): id único do utilizador autenticado

- type (string): categoria da informação da tabela
- version (string): versão do jogo

Retorno: retorna o código HTML (tabela com a lista de informações) para o cliente *ajax*.

A.4.2 Remover logs

Método: GET `web_badges/webservices/removeLogs.php`

Descrição: Remove todos os logs da base de dados guardados há mais de um ano.

Requer autenticação: Sim

Parâmetros de entrada: sem argumentos

Retorno: sem retorno

A.5 Download das tabelas

Método: GET `web_badges/actions/actionDownloadCSV.php`

Requer autenticação: Sim

Parâmetros de entrada:

- download (string): indica que tabela é para instalar num ficheiro CVS
- email (string): email do utilizador (quando *download=badge*)
- id (int): id único da medalha (quando *download=award*)
- type (string): categoria dos dados a mostrar (quando *download=game*)
- version (string): versão do jogo (quando *download=game*)
- userEmail (string): email do utilizador (quando *download=game* e *download=badge*)

Retorno: download do ficheiro csv

Anexo B

Tabelas da Base de dados

B.1 Users

Descrição: Tabela que armazena todas as contas dos utilizadores, sendo distinguidos a partir da variável classe.

Estrutura:

- id SERIAL
 - category VARCHAR(20) DEFAULT 'user'
 - email VARCHAR(50) NOT NULL UNIQUE
 - password VARCHAR(128) NOT NULL
 - passw_date DATE NOT NULL
 - code VARCHAR(6) DEFAULT 0
 - status VARCHAR(15) DEFAULT 'notverified' CHECK (status = 'notverified' OR status='verified')
 - class SMALLINT DEFAULT 0 CHECK (class=0 OR class=1 OR class=2)
- PRIMARY KEY (id)

Notas: class=0 é estudante; class=1 é educador (*issuer*); class=2 é administrador (*superuser*).

B.2 Issuer

Descrição: Tabela que estende da tabela *user*, lista de todas as contas de classe 1 (emissores).

Estrutura:

- id_fk INTEGER REFERENCES users(id)

- category VARCHAR(20) DEFAULT 'issuer'
 - denomin VARCHAR(50) NOT NULL
 - url VARCHAR(200) NOT NULL
- PRIMARY KEY(id_fk)

Notas: *denomin* é o nome da entidade ou da organização que pretende que apareça nas medalhas atribuídas; *url* é o *link* do perfil de mídia social.

B.3 Badge

Descrição: Tabela que armazena as medalhas criadas pelos emissores.

Estrutura:

- id SERIAL
 - categ VARCHAR(20) DEFAULT 'badge'
 - title VARCHAR(50) NOT NULL
 - image_url VARCHAR(128) NOT NULL
 - descript VARCHAR(128) NOT NULL
 - criteria VARCHAR(128) NOT NULL
 - issuer_fk INTEGER NOT NULL REFERENCES issuer(id_fk)
- PRIMARY KEY(id)

B.4 Assertion

Descrição: Tabela que contém a lista dos premiados pelas medalhas.

Estrutura:

- category VARCHAR(20) DEFAULT 'assertion'
 - recipient_fk INT NOT NULL REFERENCES users(id)
 - badge_fk INT NOT NULL REFERENCES badge(id)
 - issued_on DATE NOT NULL
 - vers VARCHAR(20) NOT NULL REFERENCES game(game_ver)
- PRIMARY KEY(recipient_fk, badge_fk, vers)

Notas: como uma medalha pode ser reutilizada para várias versões do jogo e o utilizador pode recebê-la em diferentes versões daí a necessidade da variável *vers*.

B.5 Survey

Descrição: Tabela que armazena as perguntas e respostas do inquérito num só texto.

Estrutura:

- id SERIAL
 - survey_data TEXT NOT NULL
 - user_fk INT NOT NULL REFERENCES users(id)
 - date TIMESTAMP NOT NULL
 - game_ver_fk VARCHAR(50) NOT NULL REFERENCES game(game_ver)
- PRIMARY KEY(id)

B.6 Survey_legacy

Descrição: Tabela que armazena as respostas do inquérito pergunta a pergunta.

Estrutura:

- id SERIAL
 - survey_id_fk INT NOT NULL REFERENCES survey(id)
 - question_num INT NOT NULL
 - question VARCHAR(500) NOT NULL
 - response VARCHAR(400) NOT NULL
- PRIMARY KEY(id)

B.7 Game

Descrição: Tabela que armazena as versões dos jogos.

Estrutura:

- game_ver VARCHAR(50) NOT NULL
- PRIMARY KEY(game_ver)

B.8 Game_issuer

Descrição: Tabela que armazena os educadores que criaram uma certa versão do jogo.

Estrutura:

- id SERIAL
 - game_ver_fk VARCHAR(50) NOT NULL REFERENCES game(game_ver)
 - issuer_fk INT NOT NULL REFERENCES issuer(id_fk)
- PRIMARY KEY(id)

B.9 Game_log

Descrição: Tabela que armazena todos os registos de informação ocorridas durante o jogo numa variável de texto.

Estrutura:

- id SERIAL
 - date DATE NOT NULL
 - initial_time TIME NOT NULL
 - final_time TIME NOT NULL
 - game_ver_fk VARCHAR(50) NOT NULL REFERENCES game(game_ver)
 - frame_ver VARCHAR(20) NOT NULL
 - user_fk INT NOT NULL REFERENCES users(id)
 - log text NOT NULL
- PRIMARY KEY(id)

Anexo C

Exemplos do ficheiro JSON dos mini jogos

Nos exemplos a baixo os textos a:

- azul - nomes das variáveis do código, não alterar;
- vermelho/verde - informação definida pelo educador, editável.

```

{
  "introTitle": [{"text": "Desafio 1:"}],
  "introText": [{"text": "Arrasta para o espaço correto cada corrente ética de acordo com o seu conceito/definição."}],

  "finalIntroTextIfWin": [{"text": "Recebeste o livro :)"}],

  "imagePath": "Escape2Ethics101Test/desafio_1.png",

  "endGameSucceededFeedbackText": [{"text": "Parabéns! Conseguiu resolver corretamente."}],
  "endGameSucceededWithHelpFeedbackText": [{"text": "Parabéns! Conseguiu resolver corretamente com ajuda."}],
  "endGameFailedFeedbackText": [{"text": "Opss..."}],

  "hintButtonFeedback": [{"text": "Elimina respostas erradas.\nPenalização: -1 ponto."}],
  "exitButtonFeedback": [{"text": "Sai do jogo.\nPenalização: -1 segundo."}],

  "timeLimitInSeconds": 90,

  "pointsRemovedForAskingHint": 1,

  "penaltyTimeFail": 1,
  "penaltyTimeHelpNeeded": 1,
  "penaltyTimeExit": 1,

  "pointsAddedForCorrectAnswer": 10,

  "puzzleItems":
  [
    {"name": "Ética das virtudes"},
    {"name": "Ética utilitarista"}
  ],
  "puzzleSlots":
  [
    {
      "correctItem": "Ética das virtudes",
      "position": {"x": -276.5, "y": 55}
    },
    {
      "correctItem": "Ética utilitarista",
      "position": {"x": -276.5, "y": -25}
    }
  ]
}

```

Figura C.1: Exemplo do ficheiro JSON do mini jogo “Label”

```

{
  "introTitle": [{"text": "Desafio 3:"}],
  "introText": [{"text": "Descobre a palavra escondida no puzzle."}],

  "finalIntroTextIfWin": [{"text": "Recebeste o teu portátil :)"}],

  "imagePath": "Escape2Ethics101Test/desafio_3.jpg",

  "endGameSucceededFeedbackText": [{"text": "Parabéns! Conseguieste resolver corretamente."}],
  "endGameSucceededWithHelpFeedbackText": [{"text": "Parabéns! Conseguieste resolver corretamente com ajuda."}],
  "endGameFailedFeedbackText": [{"text": "Opss..."}],

  "hintButtonFeedback": [{"text": "Elimina respostas erradas.\nPenalização: -1 ponto."}],
  "exitButtonFeedback": [{"text": "Sai do jogo.\nPenalização: -1 segundo."}],

  "gameTimer": 60,

  "pointsRemovedForAskingHint": 1,

  "penaltyTimeFail": 1,
  "penaltyTimeHelpNeeded": 1,
  "penaltyTimeExit": 1,

  "puzzlePositions":
  [
    {"x": 4.00, "y": -4.82 },
    {"x": 2.20, "y": 1.98 },
    {"x": 4.51, "y": 4.11 },
    {"x": 6.63, "y": 0.00 },
    {"x": 6.59, "y": 1.57 },
    {"x": 9.11, "y": 4.24 },
    {"x": 5.56, "y": 2.21 },
    {"x": 9.6, "y": -1.86 },
    {"x": 10.30, "y": -2.87},
    {"x": 8.00, "y": -4.89 },
    {"x": 3.90, "y": 3.02 },
    {"x": 1.44, "y": 0.15 },
    {"x": 3.55, "y": -1.00 },
    {"x": 3.31, "y": -2.94 },
    {"x": 1.72, "y": -5.00 },
    {"x": 5.78, "y": -3.43 }
  ]
}

```

Figura C.2: Exemplo do ficheiro JSON do mini jogo “Puzzle”

```

{
  "introTitle": [{"text": "Desafio 5:"}],
  "introText": [{"text": "Escolhe a opção correta."}],

  "finalIntroTextIfWin": [{"text": "Recebeste o teu caderno com apontamentos :)"}],

  "endGameSucceededFeedbackText": [{"text": "Parabéns! Conseguieste resolver corretamente."}],
  "endGameSucceededWithHelpFeedbackText": [{"text": "Parabéns! Conseguieste resolver corretamente com ajuda."}],
  "endGameFailedFeedbackText": [{"text": "Opss..."}],

  "hintButtonFeedback": [{"text": "Elimina uma resposta errada.\nPenalização: -1 ponto."}],
  "exitButtonFeedback": [{"text": "Sai do jogo.\nPenalização: -1 segundo."}],

  "pointsRemovedForAskingHint": 1,

  "penaltyTimeFail": 1,
  "penaltyTimeExit": 1,
  "penaltyTimeHelpNeeded": 1,

  "allRoundData": [
    {
      "timeLimitInSeconds": 20,
      "pointsAddedForCorrectAnswer": 10,
      "pointsAddedForIncorrectAnswer": -1,
      "questions": [
        {
          "questionText": "Microética é...",
          "answers": [
            {
              "answerText": "Ética aplicada ao agir de cada individuo nas suas ações públicas ou privadas.",
              "isCorrect": true,
              "feedback": "Correto."
            },
            {
              "answerText": "Ética associada com as consequências para o coletivo / comum do somatório das ações individuais.",
              "isCorrect": false,
              "feedback": "Errado. Tenta novamente."
            }
          ]
        },
        {
          "questionText": "Macroética é...",
          "answers": [
            {
              "answerText": "Ética aplicada ao agir de cada individuo nas suas ações públicas ou privadas.",
              "isCorrect": false,
              "feedback": "Errado. Tenta novamente.."
            },
            {
              "answerText": "Ética associada com as consequências para o coletivo / comum do somatório das ações individuais.",
              "isCorrect": true,
              "feedback": "Correto."
            }
          ]
        }
      ]
    }
  ]
}

```

Figura C.3: Exemplo do ficheiro JSON do mini jogo “Quiz”

Anexo D

Exemplo do ficheiro JSON do questionário

Os textos a azul são nomes das variáveis do código, não alterar. Os textos a vermelho são informações definidas pelo educador, editável.

```
{
  "questions":
  [
    {
      "questionText": "1. Relativamente ao conceito de <color=#49A179>\\"ética\\"\</color>: \n1.1.
Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos
sobre o conceito de <color=#49A179>\\"ética\\"\</color>, é:",
      "answers":
      [
        "1-Nunca ouvi falar e não sei nada sobre o assunto",
        "2-Já ouvi falar, mas não sei quase nada sobre o assunto",
        "3-Tenho apenas conhecimentos básicos sobre o assunto",
        "4-Tenho alguns conhecimentos, mas não domino o assunto",
        "5-Tenho muitos conhecimentos e domino o conceito"
      ]
    },
    {
      "questionText": "1.2. Qual o efeito do jogo Escape2Ethics101Test para os meus
conhecimentos sobre o conceito de <color=#49A179>\\"ética\\"\</color> (após jogar o jogo)? ",
      "answers":
      [
        "1-Nenhum - não se alteraram",
        "2-Muito pouco - sei quase o mesmo que antes",
        "3-Algum - aprendi algumas coisas relevantes",
        "4-Bastante - aprendi várias coisas relevantes",
        "5-Muito - aprendi muitas coisas que desconhecia"
      ]
    }
  ],
  "openQuestions":
  [
    "12. Quais os aspetos que considera <color=#49A179>melhores</color> no jogo?",
    "13. Quais os aspetos que considera <color=#49A179>piores</color> no jogo?"
  ]
}
```


Anexo E

Ficheiro JSON do Caso de Estudo

Neste anexo é apresentado o ficheiro JSON do jogo criado para testar a *framework* desenvolvida.

```
{
  "gameId": "Final_Fatima_Game_15_06_2022",
  "creationDate": "23/05/2022 00:00:00",
  "issuerEmail": "*****@hotmail.com",

  "infoIntroductionTitle": [{"text": "Instruções"}],
  "infoIntroductionText": [{"text": "\nControlos:\n    • Rodar a câmara: Botão direito do rato + Cursor do rato.\n    • Mudar de tipo de câmara: Tecla 'C'.\n    • Movimento: Teclas 'AWSD', Setas ou Botão esquerdo do rato clicando em uma posição da sala.\n    • Interagir com os objetos na sala: Os objetos aparecem realçados ao passar o cursor do rato, para interagir com eles clica com o Botão esquerdo do rato.\n\nJogo:\n    • O tempo está no canto superior direito.\n    • No canto inferior direito podes pedir dicas mas cada uma está associada a 10 segundos de penalização no tempo...\n    • Cada objeto no jogo pode ser um meio para conseguires o que é preciso para espares, podendo conter informação relevante ou um mini-jogo. Se conseguires resolver corretamente os desafios automaticamente apanhas esse mesmo objeto :) \n    • Errar ou sair de um desafio é penalizador 1 segundo, mas não te impossibilita de sair da sala, porque podes sempre voltar a tentar! \n    • Podes reler esta informação novamente clicando no botão ajuda no canto superior esquerdo.\n\nSugestão:\n    • Resolve o jogo aproveitando para estudar e, assim, escaparás para o sucesso!"}],

  "introductionTitle": [{"text": "Introdução"}],
  "introductionText": [{"text": "\nEstás sozinho numa sala de estudo e acabaste de receber uma mensagem de um colega, que te lembrou que daqui a 30 min tens uma avaliação à Unidade Curricular de 'Ética na Engenharia'. Com a pressa e o vento de uma janela aberta, os livros caem, os papeis espalham-se pela sala, a porta fecha-se com estrondo e ouve-se o trinco a fechar a porta à chave.\n\nApenas resolvendo os 8 desafios que permitem recolher os 8 objetos espalhados pela sala conseguirás abrir a porta com alguma matéria já estudada e só dessa maneira terás sucesso no teste.\n\nTens tempo suficiente para resolver os jogos, estudar e sair!\n\n"}],
```

```
"startText": [{"text": "O tempo começa quando permires o botão."}],

"initialTime":1800.0,
"clueObjs":
[
  {
    "showClueAfterTime":5.0,
    "message": [{"text": "Procura a informação primeiro antes dos desafios :)"}],
    "destroyAfterShowing": true
  },
  {
    "showClueAfterTime":10.0,
    "message": [{"text": "Lê os documentos que estão em cima da mesa onde estavas a estudar!"}],
    "destroyAfterShowing": true
  },
  {
    "showClueAfterTime":20.0,
    "message": [{"text": "Procura objetos em todas as mesas!"}],
    "destroyAfterShowing": true
  },
  {
    "itemIdNotOwned":"BookOpen",
    "showClueAfterTime":20.0,
    "message": [{"text": "O livro que tenho andado a estudar?"}],
    "penalization": 10
  },
  {
    "itemIdNotOwned":"Calculator",
    "showClueAfterTime":20.0,
    "message": [{"text": "Mmmm... eu tinha uma máquina de calcular..."}],
    "penalization": 10
  },
  {
    "itemIdNotOwned":"Laptop",
    "showClueAfterTime":20.0,
    "message": [{"text": "É melhor levar o meu portátil!"}],
    "penalization": 10
  },
  {
    "itemIdNotOwned":"PenPencilEraser",
    "showClueAfterTime":20.0,
    "message": [{"text": "Vou precisar de material de escrita..."}],
    "penalization": 10
  },
  {
    "itemIdNotOwned":"Notepad",
    "showClueAfterTime":20.0,
```

```

        "message": [{"text": "É melhor procurar o meu caderno de apontamentos para
estudar!"}],
        "penalization": 10
    },
    {
        "itemIdNotOwned": "Flashlight",
        "showClueAfterTime": 20.0,
        "message": [{"text": "Se falhar a energia dá jeito ter uma lanterna por
perto!"}],
        "penalization": 10
    },
    {
        "itemIdNotOwned": "HardDrive",
        "showClueAfterTime": 20.0,
        "message": [{"text": "Preciso do meu disco com backups. Mmmm... aquele balde do
lixo..."}],
        "penalization": 10
    },
    {
        "itemIdNotOwned": "Phone",
        "showClueAfterTime": 20.0,
        "message": [{"text": "Mmmm... mas onde é que está o meu telemóvel?"}],
        "penalization": 10
    }
],
"interactiveObjs":
[
    {
        "id": "BookOpen",
        "initialPosition": [8.259, 0.8124999, 4.687],
        "initialRotation": [0, 60.558, 0],
        "miniGame":
        {
            "id": "LabelGame",
            "pathOptions": [{"path": "Escape2Ethics101Test/desafio_1_label.json"}]
        }
    },
    {
        "id": "Calculator",
        "initialPosition": [8.097339, 0.818, 6.064828],
        "initialRotation": [0, 46.648, 0],
        "miniGame":
        {
            "id": "LabelGame",
            "pathOptions": [{"path": "Escape2Ethics101Test/desafio_2_label.json"}]
        }
    },
    {
        "id": "Laptop",

```

```

    "initialPosition": [0.717, 0.818, 4.279532],
    "initialRotation": [0, -90, 0],
    "miniGame": {
      "id": "Puzzle",
      "pathOptions": [{"path": "Escape2Ethics101Test/desafio_3_puzzle.json"}]
    },
    {
      "id": "PenPencilEraser",
      "initialPosition": [5.092, 0.84, 8.202991],
      "initialRotation": [0, 210.261, 0],
      "miniGame": {
        "id": "Puzzle",
        "pathOptions": [{"path": "Escape2Ethics101Test/desafio_4_puzzle.json"}]
      },
      {
        "id": "Notepad",
        "initialPosition": [6.154306, 0.821, 8.15561],
        "initialRotation": [0, 0, 0],
        "miniGame": {
          "id": "Quiz",
          "pathOptions": [{"path": "Escape2Ethics101Test/desafio_5_quiz.json"}]
        },
        {
          "id": "Flashlight",
          "initialPosition": [0.895, 0.851, 6.506],
          "initialRotation": [0, 90, -90],
          "miniGame": {
            "id": "Quiz",
            "pathOptions": [{"path": "Escape2Ethics101Test/desafio_6_quiz.json"}]
          },
          {
            "id": "HardDrive",
            "initialPosition": [3.452, 0, 8.334],
            "initialRotation": [0, 150.528, 0],
            "miniGame": {
              "id": "Quiz",
              "pathOptions": [{"path": "Escape2Ethics101Test/desafio_7_quiz.json"}]
            },
            {

```

```

    "id": "Phone",
    "initialPosition": [3.961, 0.017, 8.337],
    "initialRotation": [83.253, 270, 90],
    "miniGame":
    {
        "id": "LabelGame",
        "pathOptions": [{"path": "Escape2Ethics101Test/desafio_8_label.json"}]
    }
},
{
    "id": "Door",
    "initialPosition": [9, 0, 0.86],
    "initialRotation": [0, -90, 0],
    "finalRotation": [0, 0, 0],
    "unlockItemsIds": ["BookOpen", "Calculator", "Laptop", "PenPencilEraser", "Notepad",
"Flashlight", "HardDrive", "Phone"],
    "messageSucceeded": [{"text": "Escapaste!"}],
    "messageLocked": [{"text": "Antes de sair preciso das minhas coisas para fazer
o exame..."}],
    "winsGame": true
},
{
    "id": "TrashBin",
    "initialPosition": [3.444, 0, 8.455],
    "initialRotation": [-90, 0, -189.311],
    "finalPosition": [2.755, 0, 8.455]
},
{
    "id": "TrashCan",
    "initialPosition": [3.966, 0.027, 8.399],
    "initialRotation": [-89.98, 0, 0],
    "finalPosition": [4.556, 0.005, 8.399]
},
{
    "id": "LegalPad",
    "initialPosition": [4.717636, 0.816, 8.142496],
    "initialRotation": [0, 0, 0],
    "miniGame":
    {
        "id": "PDFViewer",
        "pathOptions": [{"path": "Escape2Ethics101Test/ética das virtudes 2.pdf"}]
    }
},
{
    "id": "Book_2",
    "initialPosition": [0.353, 0.8125001, 6.57],
    "initialRotation": [0, 112.107, 0],
    "miniGame":
    {

```

```

        "id": "PDFViewer",
        "pathOptions": [{"path": "Escape2Ethics101Test/ética das virtudes 3.pdf"}]
    },
    {
        "id": "Book_1",
        "initialPosition": [0.77, 0, 5.59],
        "initialRotation": [-90, 58.209, 0],
        "miniGame": {
            "id": "PDFViewer",
            "pathOptions": [{"path": "Escape2Ethics101Test/ética deontológica.pdf"}]
        }
    },
    {
        "id": "Monitor",
        "initialPosition": [0.57, 0.8124999, 7.304887],
        "initialRotation": [0, -90, 0],
        "miniGame": {
            "id": "PDFViewer",
            "pathOptions": [{"path": "Escape2Ethics101Test/ética do dever 2.pdf"}]
        }
    },
    {
        "id": "NoteGreen",
        "initialPosition": [8.404337, 0.704, 4.017],
        "initialRotation": [0, -12.687, 0],
        "miniGame": {
            "id": "PDFViewer",
            "pathOptions": [{"path": "Escape2Ethics101Test/ética e moral 1.pdf"}]
        }
    },
    {
        "id": "NoteYellow",
        "initialPosition": [0.575008, 0.829, 3.722453],
        "initialRotation": [0, 0, 0],
        "miniGame": {
            "id": "PDFViewer",
            "pathOptions": [{"path": "Escape2Ethics101Test/ética e moral 2.pdf"}]
        }
    },
    {
        "id": "NoteRed",
        "initialPosition": [0.916, 0.831, 8.168],
        "initialRotation": [0, -25.021, 0],
        "miniGame":
    
```

```

{
  "id": "PDFViewer",
  "pathOptions": [{"path": "Escape2Ethics101Test/ética e moral 3.pdf"}]
},
{
  "id": "BooksVertical",
  "initialPosition": [5.253, 0, 0.337],
  "initialRotation": [0.95, -94.081, 0],
  "miniGame": {
    "id": "PDFViewer",
    "pathOptions": [{"path": "Escape2Ethics101Test/ética utilitarista.pdf"}]
  }
},
{
  "id": "BooksHorizontal",
  "initialPosition": [7.197, 1.528, 8.358],
  "initialRotation": [0, 179.717, 0],
  "miniGame": {
    "id": "PDFViewer",
    "pathOptions": [{"path": "Escape2Ethics101Test/ética virtudes 1.pdf"}]
  }
},
],
"congratulationsText": [{"text": "Parabéns! Conseguiu sair.\nAgora corre para a sala da avaliação.\nBoa sorte!"}],
"timeEndedText": [{"text": "O teu tempo acabou! Já estás atrasado..."}],
"endAfterSurveyText": [{"text": "Obrigado por participar! :)"}],

"initialSurveyPath": [{"path": "Escape2Ethics101Test/surveyInitialPT.json"}],
"finalSurveyPath": [{"path": "Escape2Ethics101Test/surveyFinalPT.json"}]
}

```


Anexo F

Perguntas do Inquérito

INQUÉRITO INICIAL

Questões: (coloque um X na opção correta)

1. A opção que melhor descreve os meus conhecimentos sobre o conceito de “ética”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

2. A opção que melhor descreve os meus conhecimentos sobre o conceito de “deontologia”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

3. A opção que melhor descreve os meus conhecimentos sobre o conceito de “moral”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

4. A opção que melhor descreve os meus conhecimentos sobre o conceito de “microética”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

5. A opção que melhor descreve os meus conhecimentos sobre o conceito de “ética utilitarista”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

6. A opção que melhor descreve os meus conhecimentos sobre o conceito de “macroética”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

7. A opção que melhor descreve os meus conhecimentos sobre o conceito de “ética das virtudes”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

8. A opção que melhor descreve os meus conhecimentos sobre o conceito de “ética deontológica”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

INQUÉRITO FINAL

Questões: (coloque um X na opção correta)

1. Relativamente ao conceito de “ética”:

1.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “ética”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

1.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “ética” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Alguns – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

2. Relativamente ao conceito de “deontologia”

2.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “deontologia”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

2.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “deontologia” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Alguns – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

3. Relativamente ao conceito de “moral”

3.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “moral”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

3.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “moral” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Algum – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

4. Relativamente ao conceito de “microética”

4.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “microética”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas alguns conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

4.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “microética” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Algum – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

5. Relativamente ao conceito de “ética utilitarista”

5.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “ética utilitarista”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas alguns conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

5.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “ética utilitarista” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Algum – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

6. Relativamente ao conceito de “macroética”

6.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “macroética”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas alguns conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

6.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “macroética” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Algum – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

7. Relativamente ao conceito de “ética das virtudes”

7.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “ética das virtudes”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

7.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “ética das virtudes” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Algum – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

8. Relativamente ao conceito de “ética deontológica”

8.1. Após jogar o jogo Escape2Ethics101Test a opção que melhor descreve os meus conhecimentos sobre o conceito de “ética deontológica”, é:

Nunca ouvi falar e não sei nada sobre o assunto	Já ouvi falar, mas não sei quase nada sobre o assunto	Tenho apenas conhecimentos básicos sobre o assunto	Tenho alguns conhecimentos, mas não domino o assunto	Tenho muitos conhecimentos e domino o conceito

8.2. Qual o efeito do jogo Escape2Ethics101Test para os meus conhecimentos sobre o conceito de “ética deontológica” (após jogar o jogo)?

Nenhum – não se alteraram	Muito pouco – sei quase o mesmo que antes	Algum – aprendi algumas coisas relevantes	Bastante – aprendi várias coisas relevantes	Muito – aprendi muitas coisas que desconhecia

Referências

- [1] E A A Gunn, B G W Craenen, e E Hart. A taxonomy of video games and AI. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 2009.
- [2] Badges/Onboarding-Issuer - MozillaWiki. Acedido em 2022-02-20. URL: <https://wiki.mozilla.org/Badges/Onboarding-Issuer>.
- [3] LearningApps - blocos de aprendizagem interativos e multimídia. Acedido em 2022-02-20. URL: <https://learningapps.org/>.
- [4] Unity Technologies. Unity - Manual: interface do Unity. Acedido em 2022-02-21. URL: <https://docs.unity3d.com/Manual/UsingTheEditor.html>.
- [5] Unity Technologies. Unity - Manual: Requisitos do sistema para Unity 2022.1 (alfa). Acedido em 2022-02-21. URL: <https://docs.unity3d.com/2022.1/Documentation/Manual/system-requirements.html#console>.
- [6] Unity (motor de jogo). Acedido em 2022-02-21. URL: http://stringfixer.com/pt/Unity_3D.
- [7] O que é o ensino em b-learning?, Fevereiro 2020. Section: Formação. URL: <https://www.e-konomista.pt/b-learning/>.
- [8] Adelina Moura e Idalina Lourido Santos. Escape room educativo: reinventar ambientes de aprendizagem. *Aplicações para dispositivos móveis e estratégias inovadoras na educação*, páginas 107–115, 2020.
- [9] 5 maneiras de levar a aprendizagem ativa para as salas de aula, Outubro 2018. URL: <https://blog.wpensar.com.br/pedagogico/aprendizagem-ativa/>.
- [10] Panagiotis Fotaris e Theodoros Mastoras. Escape rooms for learning: A systematic review. Em *Proceedings of the European Conference on Games Based Learning*, páginas 235–243, 2019.
- [11] Claudio Santiago. A história por trás dos jogos de escape the room - Blog - Escape Time, Setembro 2017. URL: <https://escapetime.com.br/br/blog/a-historia-por-tras-dos-jogos-de-escape-the-room>.
- [12] Escape Room. Acedido em 2022-02-20. URL: <https://conceitos.com/escape-room/>.
- [13] Thiago Benasi. Escape Room - Conheça tudo sobre esse jogo, Julho 2019. Section: Escape Game. URL: <https://www.adrena.me/blog/escape-game/escape-room-conheca-tudo-sobre-esse-jogo/>.

- [14] José Ahirton Batista Lopes Filho. Uma nova era para os jogos point and click: Como a realidade virtual pode dar nova vida aos jogos de aventura.
- [15] Yoppy Sazaki, Hadipurnawan Satria, e Muhammad Syahroyni. Comparison of a and dynamic pathfinding algorithm with dynamic pathfinding algorithm for npc on car racing game. Em *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, páginas 1–6. IEEE, 2017.
- [16] Open Badges History | IMS Open Badges. Acedido em 2022-02-19. URL: <https://openbadges.org/about/history>.
- [17] Emblemas abertos do IMS. Acedido em 2022-02-20. URL: <https://openbadges.org/>.
- [18] Tharindu R Liyanagunawardena, Sandra Scalzavara, e Shirley A Williams. Open badges: A systematic review of peer-reviewed published literature (2011-2015). *European Journal of Open, Distance and E-learning*, 20(2):1–16, 2017.
- [19] Mozilla Open Badges – Great. Acedido em 2022-02-19. URL: <https://gr-eat.eu/mozilla-open-badges/>.
- [20] Gabriel G. Hornink. IDEAdigital #26 : Uso dos open badges no contexto académico, Novembro 2020. Acedido em 2022-02-21. URL: <https://idea.uminho.pt/pt/ideadigital/entradas/Paginas/entrada26.aspx>.
- [21] Build | IMS Open Badges. Acedido em 2022-03-13. URL: <https://openbadges.org/build>.
- [22] Porquê Open Badges? | Ajuda de Open Badges. Acedido em 2022-02-20. URL: <https://support.mozilla.org/pt-PT/kb/porque-open-badges>.
- [23] Formulários do Google – criar e analisar inquéritos, gratuitamente. Acedido em 2022-02-20. URL: <https://www.google.com/forms/about/>.
- [24] Learningapps.org: multimediale, interaktive lernbausteine erstellen einfach gemacht.
- [25] John K Haas. A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE*, página 32, 2014.
- [26] Linguagens - Unity 3D. Acedido em 2022-02-21. URL: <http://unity3d.wikidot.com/linguagens>.
- [27] Características - Unity 3D. Acedido em 2022-02-21. URL: <http://unity3d.wikidot.com/caracteristicas>.
- [28] Unity – Guia Completo sobre a Game Engine [2018], Janeiro 2017. URL: <https://producaodejogos.com/unity/>.
- [29] Erick Baptista Passos, José Ricardo da Silva Jr, Fernando Emiliano Cardoso Ribeiro, e Pedro Thiago Mourão. Tutorial: Desenvolvimento de jogos com unity 3d. Em *VIII Brazilian Symposium on Games and Digital Entertainment*, páginas 1–30, 2009.