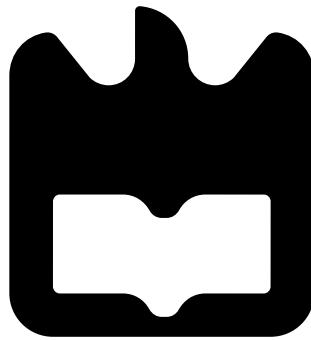




Catarina Madail
de Bastos

Plataforma de jogos multi*
(utilizador/papel/aplicação/nível/língua)





Catarina Madail
de Bastos

Plataforma de jogos multi*
(utilizador/papel/aplicação/nível/língua)

"Stand up and fight. Stand up
and see the sky turn bright.
Fight for a better day..."

— Turisas



**Catarina Madail
de Bastos**

**Plataforma de jogos multi*
(utilizador/papel/aplicação/nível/língua)**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor Joaquim Sousa Pinto, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor José Manuel Matos Moreira

Professor Auxiliar, Universidade de Aveiro

vogais / examiners committee

Professor Doutor Fernando Joaquim Lopes Moreira

Associado, Departamento de Inovação, Ciência e Tecnologia da Universidade
Portugalense

Professor Doutor Joaquim Manuel Henriques de Sousa Pinto

Professor Auxiliar, Universidade de Aveiro

agradecimentos / acknowledgements

Ao meu orientador, Professor Joaquim Sousa Pinto, o meu agradecimento pela oportunidade de realizar um trabalho numa área que tanto me fascina e com uma vertente tão importante para novas formas de aprendizagem. Agradeço também toda a disponibilidade, apoio e compreensão neste processo.

Aos meus pais, por todas as oportunidades que me deram e por todo o apoio e carinho, não só nesta fase, como em todo este percurso académico, que em muitos momentos pareceu impossível.

Ao Guilherme, que mais do que primo, é como se fosse o irmão que nunca tive. Que esta irmandade se mantenha para sempre.

Ao João, por toda a ajuda e paciência ao longo destes anos, nas mais diversas disciplinas e nesta última etapa.

À Raquel e à Tânia, pela amizade e força que me deram, sem elas não teria sido a mesma coisa. Foram as melhores amizades que fiz durante o meu percurso académico.

Ao Ricardo, que esteve sempre ao meu lado. Obrigada por seres o amigo com quem posso contar sempre.

À Mariana que em tão pouco tempo se tornou alguém tão importante na minha vida e que espero que seja para a vida.

À D. Fernanda, que em muitos momentos foi como uma mãe para mim. Por fim e não menos importante, a quem sempre me disse "Não acredites em ti, acredita em mim que acredito em ti", uma frase tonta, mas que me ajudou em muitos momentos em que só queria desistir.

Palavras Chave

jogos sérios, “joguificação”, desenvolvimento automatizado de aplicações móveis.

Resumo

Com a evolução tecnológica, novos conceitos foram introduzidos ao mundo. Se antigamente o significado de jogar passava por simples jogos de cartas e tabuleiro, hoje em dia é muito mais que isso. Os jogos eletrônicos passaram a ter uma importância significativa na vida de crianças e adultos. Com a crença que os jogos tomaram um papel importante na vida da maioria das pessoas, surge então o conceito de joguificação e jogos sérios.

Estes conceitos trouxeram toda uma nova abordagem ao ato de jogar, tornando possível a associação de um jogo a um processo de aprendizagem ou mesmo a uma simples aplicação com sistemas de recompensa. Estes sistemas de recompensa funcionam como elementos de jogos. Estudos foram feitos e provou-se a importância destes novos conceitos na vida das pessoas. Tornar aplicações joguificáveis, permite um maior foco e vontade de as usar, pois tem associado a si a diversão e competição. Assim o objetivo central desta dissertação é a implementação de uma plataforma de jogos multi*, isto é, uma aplicação onde é possível a realização de pequenos jogos com níveis e questões. Também com a possibilidade de ter jogos em diversos idiomas.

Esta plataforma deve ter em conta a simplicidade, facilidade de usar, consistência e futuras evoluções. Recorreu-se a SQL Server, Asp.NET, Javascript e HTML como principais linguagens de programação para a execução desta plataforma.

Keywords

serious games, gamification, automated development of mobile applications

Abstract

With the evolution of technology, new concepts were born. While the concept of playing used to be associated with simple card and board games, we can be sure that it means much more nowadays. Videogames have a significant role in the life of children and adults. With the belief that games have taken an important role in most people's lives, the concepts of gamification and serious games appear.

These concepts brought a brand new approach to the act of playing: we can associate a game to a learning process or even a simple app with a reward system. These reward systems function as game elements. Studies were made and they have shown the importance of these new concepts in people's lives.

The gamification of apps allow for a greater focus and desire to use them, as this associates the apps to concepts like entertainment and competition. Thus, the main goal of this essay is to implement a multi-game platform, an app where it is possible to make small games with levels and questions in several languages.

This platform must be simple, easy to use, consistent and adapt to future evolutions. The main programming languages for the making of this platform were SQL Server, Asp.NET, Javascript and HTML.

Conteúdo

Glossário	iv
Lista de Figuras	v
Lista de Tabelas	vii
1 Introdução	1
1.1 Motivação e Objetivos	1
1.2 Metodologia	1
1.3 Estrutura da dissertação	2
2 Estado da arte	3
2.1 Jogos e joguificação	3
2.1.1 Joguificação de estrutura e de conteúdo	4
2.2 Jogos sérios	4
2.3 Sistemas de joguificação	5
2.4 Arquitetura e Tecnologias disponíveis	6
2.4.1 Gestão de Bases de Dados	7
2.4.1.1 SQL Server	9
2.4.2 Backoffice	9
2.4.2.1 Asp.NET	10
2.4.2.2 ASP.NET WebAPI	11
2.4.3 Interface Móvel	11
2.4.3.1 Single Page Application	12
2.4.3.2 Frameworks de Javascript	13
3 Projeto e modelação do sistema	15
3.1 Contexto	15
3.2 Requisitos	15

3.2.1	Requisitos não funcionais	15
3.2.2	Requisitos Funcionais	16
3.2.2.1	Questões tipo	16
3.3	Visão Geral	17
3.3.1	Entidades e Atores	17
3.4	Descrição de tarefas	17
3.4.1	Criar jogo	18
3.4.2	Gerir jogos	18
3.4.3	Gerir utilizadores	18
3.4.4	Gestão de cargos	18
3.4.5	Ver tipos de questões	18
3.4.6	Gerir de Línguas	19
3.4.7	Listar e criar questões	19
3.4.8	Escolher um jogo	19
3.4.9	Jogar	19
3.4.10	Área de utilizador	19
3.5	Funcionalidades	19
3.5.1	Geral	20
3.5.2	Administração	22
3.5.3	Gestão de Jogos	23
3.6	Modelo Relacional	25
3.6.1	Jogo	25
3.6.2	Nível-Questões	26
3.6.3	Utilizador	27
4	Desenvolvimento do protótipo	29
4.1	Visão Geral	29
4.1.1	Área de trabalho	30
4.2	Tarefas	32
4.2.1	Backoffice	32
4.2.2	Interface do utilizador final	35
4.3	Resultados	37
5	Conclusão	39
5.1	Trabalho futuro	39
	Referências	41

A Anexo A	45
A.1 Descrição dos casos de uso	45
B Anexo B	52
B.1 Descrição detalhada dos atributos	52
C Anexo C	58

Glossário

AJAX Asynchronous JavaScript And XML.

API Application programming interface.

CSS Cascading Style Sheets.

HTML HyperText Markup Language.

JS JavaScript.

MER Modelo Entidade-Relação.

MVC Model-View-Controller.

PHP Hypertext Preprocessor.

SGBD Sistema de Gestão de Base de Dados.

SPA Single Page Application.

SQL Structured Query Language.

Lista de Figuras

2.1	Dashboard da plataforma NOVA	5
2.2	Aplicação QuizUp	6
2.3	Arquitetura genérica para suporte a aplicações web e multiplataforma	7
2.4	Papel de um SGBD	7
2.5	SQL Server 2017 logótipo	9
2.6	ASP.NET logótipo	10
2.7	Modelo MVC	11
2.8	Diferença entre o ciclo de páginas tradicionais e de uma SPA	13
3.1	Funcionalidades pertencentes à Sessão	21
3.2	Funcionalidades pertencentes à Conta	21
3.3	Funcionalidades pertencentes ao Jogo	22
3.4	Funcionalidades pertencentes à administração	23
3.5	Funcionalidades pertencentes à gestão de jogos	24
3.6	DER Jogos	26
3.7	DER Nível-Questões	26
3.8	DER Utilizadores	27
4.1	Protótipo da interação do sistema	31
4.2	Login	32
4.3	Menu Administrador	32
4.4	Menu do gestor de jogos	33
4.5	Lista de jogos	33
4.6	Lista de idiomas	34
4.7	Lista de utilizadores	34
4.8	Criar jogo	35
4.9	Login	35
4.10	Registo	36

4.11	Lista de jogos	36
4.12	Grupo de imagens	37
4.13	Grupo de imagens	37
C.1	Base de dados da plataforma	59

Lista de Tabelas

2.1	Diferenças entre Oracle e SQL Server	8
2.2	Diferenças entre Asp.NET e PHP	10
3.1	Tipos de perguntas suportadas em plataformas educacionais	17
3.2	Funcionalidades gerais	20
3.3	Funcionalidades de administração	22
3.4	Funcionalidades de Gestão de jogos	24
A.1	Descrição detalhada de Iniciar sessão	45
A.2	Descrição detalhada de Terminar sessão	45
A.3	Descrição de Registrar	46
A.4	Descrição de Editar Conta	46
A.5	Descrição de Ver pontuação	46
A.6	Descrição de Ver jogos	46
A.7	Descrição de Jogar	47
A.8	Descrição de Apagar jogo	47
A.9	Descrição de Apagar Utilizador	47
A.10	Descrição de Alterar cargo do utilizador	47
A.11	Descrição de Criar novo cargo	48
A.12	Descrição de Apagar cargo	48
A.13	Descrição de Editar cargo	48
A.14	Descrição de Adicionar língua	48
A.15	Descrição de Apagar língua	48
A.16	Descrição de Editar língua	49
A.17	Descrição de Adicionar um jogo	49
A.18	Descrição de adicionar um nível	49
A.19	Descrição de Adicionar questões	49
A.20	Descrição de Editar jogo	50

A.21 Descrição de Editar nível	50
A.22 Descrição de Editar questões	50
A.23 Descrição de Apagar jogo	50
A.24 Descrição de Apagar nível	51

1 | Introdução

Nesta dissertação vai ser apresentado o trabalho de desenvolvimento e pesquisa de uma plataforma de jogos, que se baseia nos conceitos de *jogos sérios* e *joguificação*. Este trabalho está direcionado para todo o tipo de áreas, tendo um especial interesse em ser usado como recurso no ensino.

1.1 Motivação e Objetivos

O aparecimento dos jogos despertou desde sempre o interesse de crianças e adultos. Ao longo dos anos foram surgindo novos conceitos que permitiram a sua evolução e lhes atribuíram um papel cada vez mais ativo no dia-a-dia, motivando a criação e adaptação de conceitos de jogos na sociedade atual.

Dois desses novos conceitos são a *joguificação* e os *jogos sérios* [1]. O objetivo da *joguificação* consiste em usar elementos de jogos (como por exemplo: pontos, emblemas, níveis, tabelas de classificações) em aplicações com objetivos completamente diversos e diferentes. Isto leva-nos ao conceito de *jogos sérios* que tem sido bastante usado na vertente de ensino e com resultados surpreendentes, tornando mais apelativa a aprendizagem. [1]

O objetivo central desta dissertação será apresentar o trabalho de desenvolvimento e pesquisa de uma plataforma de jogos, baseada no conceito de *jogos sérios* e *joguificação*. Hoje em dia torna-se crucial que este tipo de plataformas sejam possíveis de usar em qualquer tipo de dispositivo, seja um computador ou smartphone. A importância de ter uma plataforma que opera em diversos dispositivos relaciona-se com a necessidade atual de realização de várias tarefas em vários meios, quando em tempos existia um maior foco no computador.

1.2 Metodologia

Será realizada uma pesquisa descritiva com a finalidade de explorar os conceitos de *joguificação* e *jogos sérios*, partindo de uma revisão bibliográfica composta por alguns artigos pertinentes sobre o assunto. O objetivo após um estudo sobre os conceitos, será fazer uso dos mesmos na modelação da plataforma. Como objeto empírico, foram selecionadas 4 platafor-

mas joguificadas. Estas serão objeto de estudo por forma a encontrar pontos em comum entre os conceitos e ideias a implementar na plataforma de jogos multi*.

Partindo dos conceitos apresentados, será necessário um estudo das suas vantagens em ambientes reais. Para isto, será necessária uma pesquisa por artigos científicos que comprovem essas mesmas vantagens. Será também feita uma análise de várias tecnologias de forma a escolher as mais interessantes para a plataforma.

Após toda a análise cuidada de conceitos, exemplos e tecnologias, será feita a projeção e modelação do sistema, fazendo um levantamento de requisitos funcionais e não funcionais da plataforma, assim como de todos os casos de uso. Posteriormente será feito o desenvolvimento do protótipo.

1.3 Estrutura da dissertação

Esta dissertação está dividida em 5 capítulos, organizados da seguinte forma:

1. Introdução: é feita uma breve introdução sobre o que se pretende com esta plataforma e os conceitos a reter;
2. Estado da arte: neste capítulo fala-se sobre os conceitos teóricos adquiridos para a resolução deste projeto, exemplos de outros projetos onde são usados estes conceitos e é feita uma análise das soluções tecnológicas existentes;
3. Projeto e modelação do sistema: é apresentada uma visão geral de como se espera que seja o produto final, são enumeradas tarefas que farão parte da plataforma e são definidos os requisitos do sistema;
4. Desenvolvimento do protótipo: aqui é apresentada a interação do utilizador com a plataforma, indicando algumas das suas tarefas mais importantes. Faz-se uma análise dos resultados e das metas atingidas de acordo com o delineado em capítulos anteriores;
5. Conclusão: por fim é feita uma breve conclusão, definindo que objetivos foram cumpridos, a aprendizagem que houve com a resolução deste projeto e falando de possível trabalho futuro.

2 | Estado da arte

Este capítulo é o resultado de uma pesquisa relativa a conceitos teóricos a ser implementados na plataforma de jogos multi*, estudo de arquiteturas tecnológicas disponíveis e exemplos de sistemas de joguificados.

2.1 Jogos e joguificação

Um jogo trata-se de uma atividade estruturada com o envolvimento de jogadores. Os jogos tem objetivos traçados e que os jogadores têm de alcançar. Estes objetivos ou desafios costumam incluir vários mecanismos de tentativas caso o jogador falhe, e sistemas de recompensa por objetivos cumpridos. O fim do jogo é definido pela vitória de um ou vários jogadores.

Para melhor definir o conceito de jogo é importante conhecer algumas características importantes a que um jogo deve obedecer. A principal característica são as regras sobre as quais os jogos devem ser definidos. A segunda característica são os resultados, que devem ser quantificáveis e variáveis. Isto leva à terceira característica, o valor atribuído aos resultados pode variar, tanto de forma negativa como positiva e de acordo com ações do jogador. A quarta característica está relacionada com o esforço do jogador, que pode influenciar o resultado do jogo, sendo o objetivo os jogos serem desafiantes. Outra característica importante é manter o jogador motivado com o resultado final. Por fim, as consequências devem ser negociáveis, mantendo o interesse do jogador na possibilidade de mudar as consequências do seu próprio jogo. [2]

Os jogos têm sido populares desde sempre, incluindo todas as faixas etárias. O interesse do ser humano em jogos levou a estudos sobre a inclusão do conceito de jogos em atividades do dia-a-dia. Temos como exemplo, o estudo de disciplinas ou até competição entre colaboradores de uma empresa com o objetivo de melhorar a sua performance e ao mesmo tempo terem um sistema de recompensa. Isto leva-nos ao conceito de "joguificação".

Este conceito existe desde sempre, mas apenas em 2002 surgiu pela primeira vez o termo "gamification"(que significa "joguificação") pelo programador Nick Pelling[3]. Entretanto vários autores definiram a palavra e apesar de a definirem de formas distintas, o real conceito da palavra baseia-se no mesmo. De acordo com alguns artigos e até a própria definição do dicionário, "joguificação" define-se como o uso de técnicas de design de jogos que utilizam mecânicas e pensamentos orientados para enriquecer contextos diversos normalmente não relacionados com

jogos. Muitas aplicações estão "joguificadas" através de sistemas de recompensa e reputação com pontos, emblemas, níveis e tabelas de classificação.[1]

2.1.1 Joguificação de estrutura e de conteúdo

A "joguificação" pode ser dividida em dois tipos: de estrutura e de conteúdo.

No primeiro caso, diz-se que há "joguificação" estrutural quando são utilizados elementos de jogos numa aplicação - o objetivo da aplicação mantém-se o mesmo mas a adição de elementos de jogos permite motivar o utilizador. Por exemplo, o Foursquare é uma rede social com o objetivo do utilizador indicar onde se encontra, mas tem um aspecto "joguificado" ao permitir o utilizador ganhar um crachá por cada sitio por onde passa. [4]

No segundo caso também há utilização de elementos de jogo - mas o princípio é ser semelhante a um jogo. Nesta situação é fornecido o contexto e atividades que seriam usadas num jogo, mas num conteúdo de aprendizagem. Por exemplo o exército americano criou um "jogo" chamado America's Army [5], usado como uma ferramenta de recrutamento. É um jogo tático de tiro em primeira pessoa e para múltiplos jogadores. Este coloca os jogadores em cenários de combate pelo mundo e testa a sua capacidade de ser um bom soldado, permitindo saber se o jogador será um bom candidato à fase de recrutamento. [4]

2.2 Jogos sérios

Torna-se imperativo referir que o conceito de "jogos sérios" tem um objetivo educativo e de aprendizagem. Existem duas definições importantes relativas a "jogos sérios". A primeira é de Etienne Armand Amato (2007), que diz que "os "jogos sérios" podem ser definidos como videojogos utilitários, por outras palavras produtivos, cujo design procura trazer uma transformação nos seus utilizadores na linha da melhoria de capacidades (formação), adaptação ao ambiente (tratamento de fobias), compreensão de um fenómeno (educação), ou maior adesão a uma mensagem (promoção, publicidade, vídeo jogos ideológicos, também conhecidos como jogos políticos). [6]

A outra definição de acordo com Julian Alvarez (2007) diz que "um jogo sério procura combinar aspetos sérios como, mas não limitados a ensino, aprendizagem, comunicação ou demais informação com entretenimento vindo do jogo original. Tal associação tem como intenção elevar-se do mero entretenimento". [6]

A pergunta que se coloca é qual o verdadeiro benefício de jogos sérios na educação? Os "jogos sérios" incluem dois fatores de motivação importantes: tecnologia e diversão.

Thomas Malone e Mark Lepper (1987) referem que as 4 características que definem um ambiente motivador são: o desafio, a curiosidade, o controlo e a imaginação. Por outro lado, Csikszentmihalyi (1990) identifica o conceito do fluxo, também conhecido como experiência optimal, como um aspeto específico da atividade de jogo que provavelmente criará um ideal

de motivação. Num contexto de jogo é mais provável que o interesse pelo processo de aprendizagem seja o maior fator de motivação para quem está a aprender.[6]

2.3 Sistemas de joguificação

Numa breve pesquisa pode-se encontrar diversas aplicações que fazem uso dos conceitos de "joguificação" e "jogos sérios" - anteriormente já foram referidas algumas como o Foursquare e o America's Army. Não alargando muito o número de exemplos, há outras que importam referir dados os diferentes contextos.

Por exemplo, o NOVA é uma plataforma criada pela empresa Novabase que tem como principal objetivo o enriquecimento individual do colaborador, com vista a ter melhores resultados na empresa e com isso ter acesso a prémios. A plataforma NOVA tem um ambiente que remete a uma aventura especial, onde o "NOVA Master", isto é, o responsável por determinado projeto/colaborador, define objetivos que o colaborador tem que cumprir, existindo feedback diário. A cada 2 meses os colaboradores tem avaliações, onde se auto-avaliam e recebem a sua avaliação do gestor de projeto e de outros colaboradores desse mesmo projeto. Com isto vão ganhando pontos e crachás. A plataforma também tem desafios surpresa, que dão acesso a destinos de férias e conferências. De acordo com a Novabase, isto traz mais dinâmica à empresa, objetivos mais focados em missões e competição pelos melhores resultados, estimulando assim o espírito de equipa para que as missões de determinado projeto sejam bem sucedidas. Na Figura 2.1 é apresentada a dashboard da plataforma NOVA, onde o utilizador tem acesso a todas as informações importantes. [7]

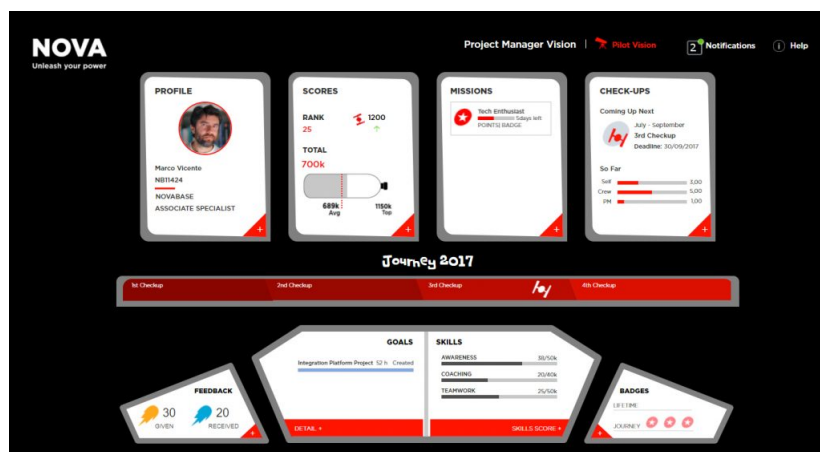


Figure 2.1: Dashboard da plataforma NOVA

Outro exemplo de certa forma mais ligado ao tema desta dissertação são jogos como o Quiz Up, onde o jogador pode escolher categorias para responder a questões, ganhando pontos e podendo desbloquear crachás de acordo com a categoria, e boosts para o jogo. Este tipo de jogo está mais ligado ao conceito de "jogos sérios" pela sua componente de aprendizagem. Apesar do objetivo ser mostrar conhecimentos, também é possível aprender com questões que

desconhecemos. Este jogo foi desenvolvido para Android, iOS e Windows Phone [8]

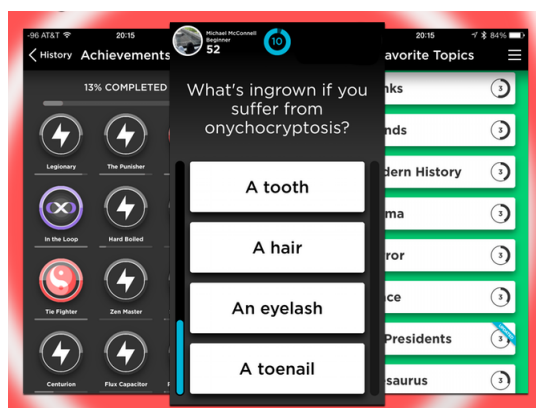


Figure 2.2: Aplicação QuizUp

Este último exemplo leva-nos a outro tópico muito importante, a necessidade de uma aplicação multiplataforma para dispositivos móveis. Esta aplicação pode ser usada em iOS, em Android e ainda tem um ranking online que pode ser consultado numa página web.

Ao criar uma aplicação multiplataforma, com base numa arquitetura SPA (Single Page Application), ou "aplicação de página única", podemos usá-la em qualquer dispositivo móvel e também numa página web no computador. A grande vantagem da aplicação multiplataforma, é que basta desenvolver apenas uma versão de código e após compilado, adapta-se a várias tipos e versões de sistemas operativos. Uma das ferramentas mais usadas para este efeito é o PhoneGap da Adobe Systems.

Basear-nos numa arquitetura SPA tem bastantes vantagens para além da enumerada em cima. Sempre que é necessário atualizar partes da interface, não é necessária atualizar a página inteira, apenas as partes necessárias. Este tipo de arquitetura usa AJAX para interagir com o servidor, templates HTML e uma infraestrutura MVC/MVVM (Model-View-Model/Model-View-ViewModel) e JavaScript.

A implementação da infraestrutura pode ser aplicada a várias plataformas JavaScript. Toda esta parte refere-se ao Front-end, a parte visível para os utilizadores da plataforma.

Isto leva a uma discussão sobre a arquitetura e tecnologias disponíveis para o desenvolvimento da plataforma de jogos multi*, que será abordada na secção seguinte.

2.4 Arquitetura e Tecnologias disponíveis

Na Figura 2.4 está representada uma arquitetura genérica para suporte a aplicações web e multiplataforma, onde existe uma base de dados que vai ser populada através do Backoffice. O Backoffice não é visto pelo utilizador final, mas vai ter tarefas bastante importantes que serão descritas mais tarde. Existe também uma Web API, que está ligada à base de dados. Esta API serve de ponte entre a base de dados e a aplicação mobile.

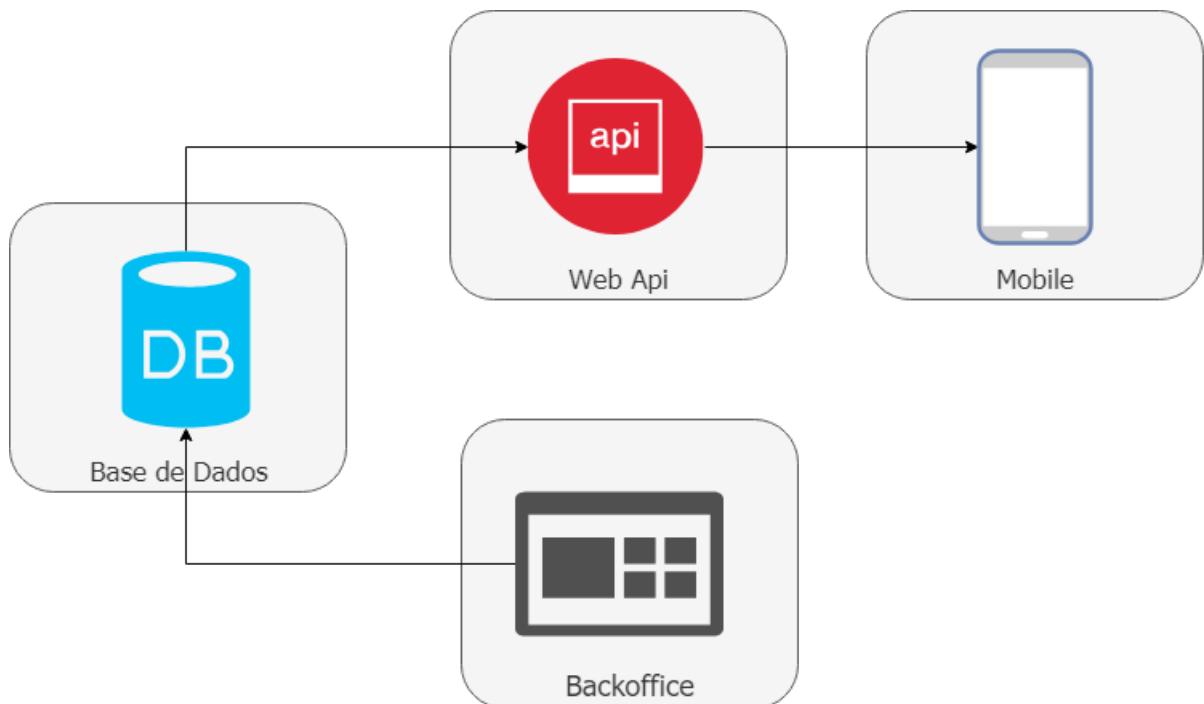


Figure 2.3: Arquitetura genérica para suporte a aplicações web e multiplataforma

2.4.1 Gestão de Bases de Dados

Um Sistema de Gestão de Base de dados (SGBD) é um programa ou conjunto de programas que gere manipula e pesquisa dados numa base de dados. Os dados são independentes deste sistema, sendo que este funciona como interface entre aplicações e os dados da base de dados [9]

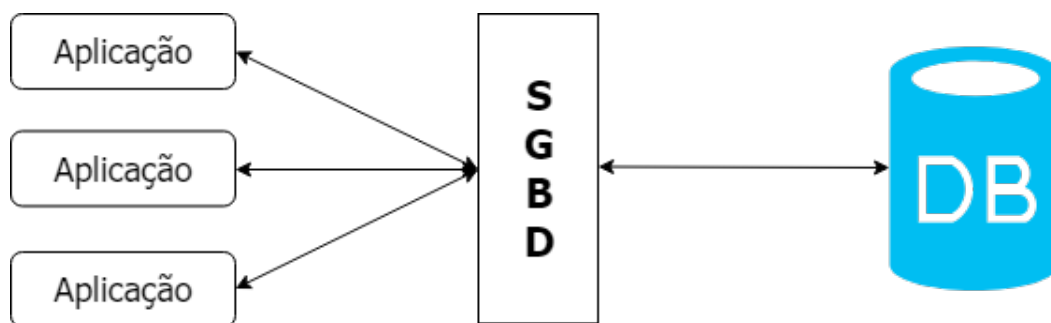


Figure 2.4: Papel de um SGBD

Ao utilizar um SGBD torna-se importante atingir alguns objetivos, sendo eles:

- a manutenção de dados armazenados (persistência);
- a capacidade de lidar com muitos dados (eficiência);

- a capacidade de recuperar de erros que possam existir (robustez);
- e ter um controlo de acessos e possibilidade de acessos múltiplos à base de dados, mantendo os dados coerentes. [10]

A utilização de um sistema de gestão de base de dados torna-se importante e proveitosa, pois tem diversas vantagens, tais como:

- Controlo de Redundância - Uma redundância numa base de dados ocorre quando os mesmos dados se encontram em locais diferentes, provocando inconsistências na base de dados. Ao fazer o controlo de redundância, não serão guardadas cópias de dados iguais, só acontecendo em casos pontuais onde é mesmo necessário;
- Restringir Acesso Não Autorizado - É importante que o SGBD tenha recursos que possibilitem aos utilizadores restrições ou autorizações de acesso, sendo que poderão ou não manipular os dados consoante a tarefa que se pretende executar. Isto é, certos utilizadores poderão ter qualquer tipo de acesso, outros só poderão ter acesso a leitura de dados, outros a escrita e assim sucessivamente;
- Consistência dos Dados - Sempre que é alterado um valor na base de dados, isso deve ir de um estado de consistência para outro estado de consistência na base de dados. Qualquer utilizador com autorização deve ter acesso a essa alteração;
- Cópias de Segurança e Recuperação de Erros - Os dados devem ser periodicamente guardados (backups), para que numa situação em que seja necessário fazer recuperação de erros, existam sempre cópias de segurança para esse efeito;
- Partilha de Informação - A base de dados pertence a todos os utilizadores, ainda que existam dados reservados a apenas alguns utilizadores.

Existem duas soluções de SGBD bastante populares que são: Oracle e SQL(Structured Query Language) Server da Microsoft. De forma a escolher a melhor ferramenta para a implementação desta forma, será apresentada a tabela 3.2 com prós e contras e fundamentar assim a escolha efetuada. [11–13]

	Oracle	SQL Server
Desenvolvimento	PL/SQL, Java	Transact SQL and .NET languages (Estas ferramentas facilitam o desenvolvimento)
Portabilidade	AIX, HP-UX, Linux, OS X, Solaris, Windows, z/OS	Windows, Linux
Segurança	Algumas vulnerabilidades	Mais seguro
Desempenho	O Oracle tem melhor desempenho que o SQL Server	
Custos	Um dos mais caros do mercado	Custo menor

Table 2.1: Diferenças entre Oracle e SQL Server

Ambas as ferramentas são bons Sistemas de Gestão de Base de Dados, cada um com os seus pontos fortes e pontos fracos. Mas dada a facilidade de desenvolvimento, menor custo e segurança, SQL Server será a ferramenta escolhida. Esta ferramenta também é mais indicada para pequenas e médias aplicações, visto que a sua integração com a linguagem que é usada é de mais fácil compreensão e adaptação.

2.4.1.1 SQL Server

Para o desenvolvimento da base de dados foi escolhido o SQL Server 2017. É um SGBD da empresa Microsoft bastante completo, em que o seu objetivo principal é armazenar e recuperar dados para outras aplicações.



Figure 2.5: SQL Server 2017 logótipo

O SQL Server 2017 permite criar aplicações em linguagens à escolha, nomeadamente: T-SQL, Java, C/C++, C#/VB.NET, PHP, Node.js, Python, Ruby. Já não está limitado apenas ao Windows, sendo possível ser usado em Linux ou Docker. Tem como grande referencia a escalabilidade, desempenho e disponibilidade para aplicações inteligentes.

É considerado também o que tem menos vulnerabilidades de segurança comparado com outros fornecedores de bases de dados, de acordo com o National Institute of Standards and Technology. [13]

2.4.2 Backoffice

O Backoffice é um sistema de gestão de conteúdos. Ajuda a alterar dados e conteúdos de uma forma fácil, sem exigir um conhecimento técnico por parte dos utilizadores com acesso ao Backoffice, que por norma têm o papel de administradores. [14]

Neste caso, o Backoffice vai permitir fazer as alterações à base de dados que serão importantes de ser vistas na interface móvel.

A criação do Backoffice vai-se basear em páginas web dinâmicas. Para isso é importante definirmos o que é uma página web dinâmica e qual será a plataforma que vamos usar para este fim.

Existem dois tipos de páginas: estáticas e dinâmicas. As páginas estáticas são simples e usam apenas HTML. Não têm muito mais do que links para outras páginas do mesmo tipo. Por sua vez, as páginas dinâmicas incluem funcionalidades que implicam a utilização de outras linguagens de programação, para além de HTML. Existem dois tipos de páginas dinâmicas: de cliente e de servidor. [15] Nas páginas web dinâmicas de cliente, os scripts são executados localmente, ou seja, o processamento é feito do lado do cliente. Por sua vez, nas páginas web dinâmicas de servidor, o processamento fica a cargo do servidor da página web. Neste último tipo de páginas são realizadas funções que não podem ser executadas no lado do cliente, por questões de segurança. As plataformas mais conhecidas para desenvolvimento web são ASP.Net e PHP. Por isso, foram estudadas as principais diferenças entre uma e outra de forma a escolher a melhor plataforma a usar. [16–18]

	Asp.NET	PHP
Escalabilidade	A escalabilidade não está dependente do Asp.NET ou PHP, mas sim das práticas e experiência do programador	
Performance	Boa performance, menos suscetível a falhas	Boa performance
Segurança	Bastante Seguro	Algumas vulnerabilidades
Desenvolvimento	Mais fácil e com escolha possível escolha de linguagem	Também de fácil aprendizagem.
Custos	Versão Core gratuita	Gratuito
Integração com SQL Server	Facilitada	Mais complexa

Table 2.2: Diferenças entre Asp.NET e PHP

2.4.2.1 Asp.NET

O ASP.NET é uma framework que permite criar sites e aplicações web. Também permite criar Web API's. As aplicações podem ser escritas em variadas linguagens, como C#, F# e Visual Basic. [19]



Figure 2.6: ASP.NET logótipo

O ASP.NET tem três frameworks disponíveis para aplicações web: Web Forms, ASP.NET

MVC, and ASP.NET Web Pages. Para esta plataforma foi escolhida a ASP.NET MVC.

MVC é a sigla para Model-View-Controller, um padrão de software poderoso capaz de construir páginas web dinâmicas. Permite uma separação dos problemas e total controlo para um desenvolvimento ágil e agradável do código. O "Model" refere-se ao conjunto de classes que modelam e suportam o projeto. As "Views" renderizam o conteúdo para o utilizador e transmitem ações do utilizador para o "Controller". Um "Controller" é um objeto que permite manipular o que é apresentado na "View", isto é, interage com o "Model" para apresentar dados na "View". De forma a percebermos melhor, podemos dizer que o "Controller" manipula a entrada enquanto a "View" manipula a saída. [20]

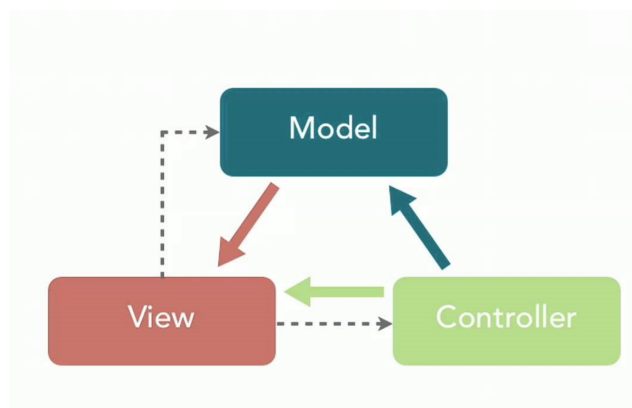


Figure 2.7: Modelo MVC

2.4.2.2 ASP.NET WebAPI

O ASP.NET Web API é uma framework que permite criar e disponibilizar serviços HTTP sobre .NET. Estes serviços serão consumidos por browsers e aplicações móveis.

Desta forma vamos ter Javascript, CSS e HTML do lado do cliente e ASP.NET Web API a disponibilizar serviços/dados do lado do servidor. Olhando para a figura 3.1, vemos que a Web API desta plataforma vai consumir os dados vindos da base de dados e mostrá-los na aplicação móvel. [21]

2.4.3 Interface Móvel

As aplicações móveis ganharam uma importância significativa na vida dos utilizadores. São usadas para realizar as mais distintas tarefas. Por esse motivo, torna-se importante garantir que a plataforma de jogos multi* tenha a possibilidade de ser usada como aplicação móvel. Existem ainda 3 opções: aplicações nativas, aplicações híbridas e aplicações web. É importante fazer esta distinção antes de fazer a escolha. [22]

- Aplicações nativas: são aplicações desenvolvidas para um sistema específico e criadas

com uma linguagem específica (por exemplo, Objective C e Swift para iOS e JAVA para Android). Este tipo de aplicações caracteriza-se por ter acesso aos recursos do dispositivo (exemplo: camara, gps,..)

- Aplicações híbridas: são imperceptíveis as diferenças entre este tipo de aplicação e as aplicações nativas, do ponto de vista do utilizador. Este tipo de aplicações são desenvolvidas com tecnologias HTML, CSS e Javascript. São depois alojadas dentro de uma aplicação nativa, utilizando uma web view. Essa web view é que permite o acesso a alguns recursos do dispositivo.

- Aplicações web: é feito o acesso através do browser.

Para esta plataforma foi escolhido realizar uma aplicação híbrida. É importante então explicar o conceito de SPA (Single Page Application) e a framework de Javascript usada, Durandal.

2.4.3.1 Single Page Application

SPA é a sigla para Single Page Application e é uma aplicação web que carrega uma página HTML e que vai fazendo updates dinâmicos nessa mesma página à medida que o utilizador interage com a mesma.

Usa AJAX e HTML5 para criar aplicações web fluidas e responsivas, sem estar constantemente a ser atualizada. Isto significa que muito do trabalho ocorre do lado do cliente através de Javascript. Para isto existem muitas frameworks de Javascript que torna fácil o processo de criar SPAs.

Numa aplicação web "tradicional", sempre que a aplicação chama o servidor, este renderiza uma nova página HTML. Isto provoca um atualização de toda a página no browser.

Numa SPA, depois da primeira atualização da página, todas as interações futuras com o servidor ocorrem através de chamadas AJAX. Estas chamadas retornam dados normalmente em formato JSON. Este tipo de aplicações usam dados JSON para atualizar a página dinamicamente, sem ter de atualizar toda a página. A Figura 3.5 descreve exatamente a diferença entre uma aplicação web "tradicional" e uma aplicação web que faz uso de SPA. [23]

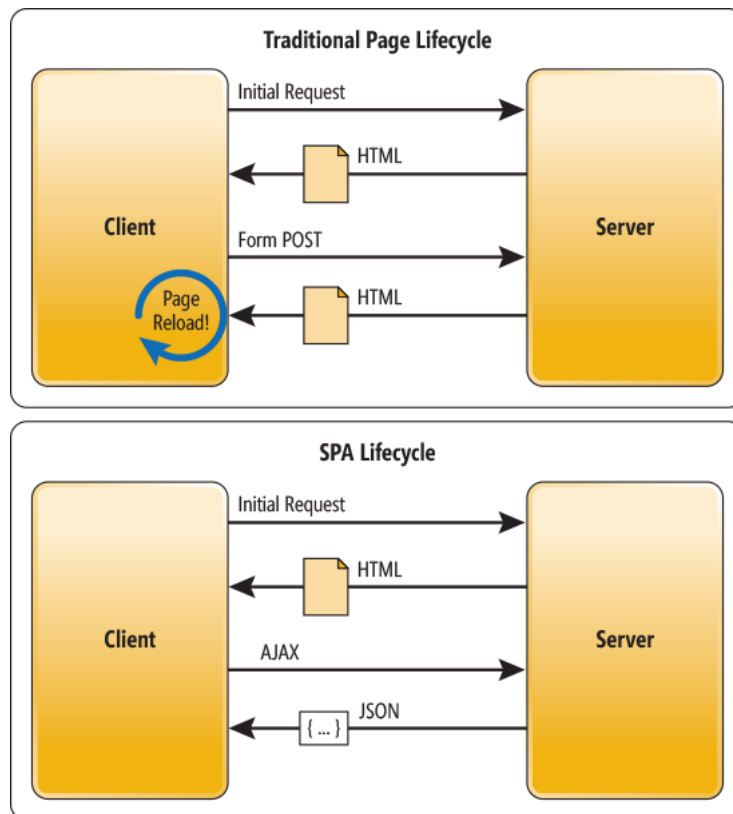


Figure 2.8: Diferença entre o ciclo de páginas tradicionais e de uma SPA

2.4.3.2 Frameworks de Javascript

Existem cada vez mais frameworks de Javascript disponíveis no mercado, sendo necessário escolher a mais viável de acordo com o projeto. Por este motivo foi feito um estudo de 2 frameworks (Durandal, Angular) e uma biblioteca (React). Importante referir que o Durandal é um pacote que integra bibliotecas jQuery, Knockout e Require.js.

Este estudo incidiu nos seguintes tópicos :

1. Curva de aprendizagem

- Durandal: várias dependências iniciais não facilitam uma aprendizagem tão fácil como Angular. Mas após ultrapassado o problema de dependências, os padrões e convenções usados por esta framework melhoram à medida que os projetos crescem. É considerado mais fácil de aprender que React.
- Angular: é mais fácil de começar comparativamente a Durandal, uma vez que não há dependências.
- React: usa menos abstrações que angular, mas pode demorar algum tempo a aprender as boas práticas desta biblioteca. [24,25]

2. Ligação de dados (Data binding) É das coisas mais importantes num produto web, pois permite a ligação de elementos HTML a objetos de dados javascript.

- Durandal: usa atributos de dados HTML5 para ligar elementos HTML a objetos de dados javascript. Isto permite que mais trabalho seja feito pela framework, em vez de exigir que se especifique onde os dados devem ser ligados no Javascript. Para esta ligação de dados o Durandal usa o KnockoutJS.
- Angular: A forma como os dados são usados pelo angular é mais simples que Durandal, no entanto a forma como se faz esta ligação pode acabar por afetar o desempenho da aplicação web.
- React: implementa uma ligação de dados unidirecional. Portanto, os dados são direcionados apenas de uma maneira. Essa abordagem facilita muito a depuração do projeto quando se trata de aplicações grandes. No entanto o Durandal, nomeadamente a biblioteca de Knockout.js é considerada melhor para ligação de dados que React, de acordo com as fontes citadas. [24, 25]

3. Arquitetura MVC

- Durandal: tem arquitetura MVC
- Angular: tem arquitetura MVC
- React: não tem arquitetura MVC [24, 25]

4. Escalabilidade

- Durandal: graças a dependências usadas na configuração no início de um projeto, a framework Durandal permite que seja mais escalável em projetos que vão crescendo, comparativamente a Angular.
- Angular: fácil de escalar.
- React: mais escalável que Durandal e Angular [26, 27]

Importante de referir que em relação à documentação, todas as frameworks e a biblioteca React são boas, no entanto React e Angular são mais populares - para além da boa documentação existem inúmeras fontes de informação.

Pesando os prós e contras enumerados a cima, a framework escolhida foi o Durandal. Durandal é uma framework de Javascript destinada a criar SPAs de forma simples. Uma vez que faz suporte a MVC, MVP e MVVM, não interessa o tipo de arquitetura de front-end que se pretende, que será possível realizar com Durandal.

A framework Durandal é um pacote que integra as bibliotecas jQuery, Knockout, Require.js. O RequireJS é a base, que usa uma estrutura modular de forma a ser possível existir escalabilidade na aplicação. Com isto o Durandal ajuda bastante na produtividade, mantendo práticas de programação sólidas. Durandal.js é bastante completo uma vez que usa o RequireJS para módulos, usa o Knockout.js para ligação de dados e o JQuery é utilizado para fazer a abstração dos elementos do browser.

3 | Projeto e modelação do sistema

Neste capítulo serão apresentados os requisitos funcionais e não funcionais, os atores do sistema, bem como as tarefas a incluir. Será realizada uma descrição de casos de uso e apresentado o modelo do domínio, que é basicamente a arquitetura base usada para suportar a consistência de dados que o sistema exige.

3.1 Contexto

O objetivo será criar uma plataforma de jogos multi*, esta plataforma serve para que múltiplos utilizadores com diferentes papéis, possam desenvolver múltiplos jogos. Estes jogos podem ser de multinível e multilíngue.

Cada jogo que tem N níveis, que por sua vez tem M tipos de perguntas diferentes. Utilizadores com papel de jogador, poderão mais tarde jogar qualquer jogo que esteja disponível na plataforma. Desta multiplicidade surge o nome da plataforma: multi*.

Esta plataforma que é categorizada como "jogos sérios", é uma ferramenta muito importante num contexto educacional e de aprendizagem, trazendo motivação através da contextualização de jogos na educação.

Para isto será criada uma plataforma online, possível de integrar em qualquer tipo de dispositivo móvel ou uma simples página web.

3.2 Requisitos

3.2.1 Requisitos não funcionais

Fazendo uma análise de requisitos, verificamos que há 5 essenciais:

- Portabilidade: possibilidade da aplicação ser usada em diversos tipos de dispositivos;
- Usabilidade: na perspetiva do utilizador, a interface deve ser simples e intuitiva;

- Escalabilidade: deve existir a possibilidade de integrar novas dinâmicas à plataforma, torná-la mais "joguificável" ou acrescentar novas ferramentas;
- Desempenho: qualquer aplicação deve ter uma resposta rápida, de forma a maximizar o desempenho dos utilizadores em relação à mesma;
- Segurança: os dados dos utilizadores devem ser confidenciais.

3.2.2 Requisitos Funcionais

Como requisitos funcionais temos:

- N utilizadores podem assumir M papéis;
- Cada utilizador possuirá N jogos com M permissões em cada jogo;
- Cada jogo suportará N níveis de M "tipos de perguntas" diferentes;
- Cada nível poderá ser desenvolvido em N línguas diferentes e um mesmo nível pode estar em M jogos distintos;
- Cada nível possui um identificador no jogo, um tipo de pergunta. A pergunta é a base de cada um dos níveis;
- Existência de uma mensagem de sucesso e uma mensagem de falha, deve ser apresentada consoante a resposta do utilizador. Também deve haver uma mensagem de ajuda, que contem pistas para o utilizador chegar à resposta;
- Lista com respostas à pergunta efetuada (conteúdo variável em função da pergunta).

Estes requisitos estão em função do que foi inicialmente descrito como objetivos da dissertação.

3.2.2.1 Questões tipo

Dado os requisitos funcionais desta plataforma, é importante definir os tipos de questões existentes e que fazem sentido incluir na plataforma multi*. A Tabela 3.1 sumariza os tipos de questões e a que grupo de perguntas pertencem.

Sendo que existe quatro grupos: "bullet questions"(onde se escolhe uma opção), "text questions"(onde a resposta é dada em forma de texto), "matching questions"(onde se faz correspondências entre listas) e por fim "ordering questions"(em que o objetivo é ordenar).

Tipos de questão	Grupo de perguntas
Múltipla escolha	Bullet Question
Verdadeiro/Falso	"
Escolha numa lista	"
Múltiplas corretas	"
Preenchimento de texto	Text Question
Alteração de resposta proposta	"
Resposta curta	"
Correspondência de listas	Matching Question
Ordenação	Ordering Question

Table 3.1: Tipos de perguntas suportadas em plataformas educacionais

3.3 Visão Geral

É crucial enumerar os atores e definir as tarefas que o sistema deve suportar. Assim permitimos que haja um avanço no desenvolvimento do sistema.

3.3.1 Entidades e Atores

Vamos começar por enumerar as entidades envolvidas na plataforma de jogos multi*:

- Administrador: Entidade que tem acesso ao Backoffice e pode alterar os papéis e fazer a administração de utilizadores. Tem acesso a tudo o que as outras entidades têm acesso.
- Editor de jogos: Entidade que pode criar, editar e apagar jogos e questões . Tendo apenas acesso aos seus próprios jogos e às próprias questões.
- Jogador: Entidade que apenas tem como papel jogar, os jogos fornecidos e autorizados na aplicação móvel.

Estas entidades, também podem ser descritas como atores, uma vez que todas contribuem de forma distinta para o sistema. Então temos 3 atores: Administrador, Editor de jogos e Jogador.

3.4 Descrição de tarefas

Nesta secção serão enumeradas e especificadas as principais tarefas do sistema, fazendo uma análise do seu funcionamento, os intervenientes, o contexto e objetivos a alcançar.

3.4.1 Criar jogo

Uma das principais tarefas desta plataforma, é a criação de jogos. A criação de um novo jogo, implica a criação de níveis e questões. Ao criar um novo jogo é necessário o preenchimento de todos os campos obrigatórios. Existe apenas um interveniente principal neste processo, apesar do administrador ter todos os privilégios para criação de jogos:

- Editor de jogos: cria o jogo, preenchendo todos os campos obrigatórios. De seguida cria um nível e as perguntas associadas a ele.

3.4.2 Gerir jogos

Esta tarefa permite editar e apagar jogos. Os intervenientes desta tarefa são dois:

- Administrador: Pode eliminar qualquer jogo, independentemente de quem o criou, isto é tem acesso a uma lista de todos os jogos existentes.
- Editor de Jogos: Pode editar e eliminar apenas os jogos que criou, a sua lista de jogos é restrita apenas aos jogos que criou.

3.4.3 Gerir utilizadores

Apenas o administrador tem acesso a esta tarefa, sendo que pode eliminar utilizadores e alterar os cargos deles.

3.4.4 Gestão de cargos

Esta é outra tarefa que pertence apenas ao administrador. Este pode criar cargos novos, editar os existentes ou apagar. É uma tarefa que não será usada muitas vezes, mas caso haja escalabilidade da plataforma, pode ser importante a existência de novos cargos. Na gestão de cargos, o administrador também tem uma lista de utilizadores, onde pode alterar os seus cargos.

3.4.5 Ver tipos de questões

De acordo com a tabela 3.1, existiram apenas esses tipos de questão na plataforma. Mas mais uma vez, caso haja escalabilidade da plataforma, pode ser adicionadas mais tipos de questões. No entanto, como é uma tarefa que pode vir a gerar erros na plataforma, só será possível ver os tipos de questões existentes.

3.4.6 Gerir de Línguas

A plataforma de jogos multi* é multilíngue, então o administrador deve manter uma lista de línguas atualizadas e ter a possibilidade de editar ou eliminar. Assim sempre que um jogo é criado, é também associada a língua em que foi criado o jogo. Por exemplo, se o editor de jogos criar um jogo onde todas as questões são em inglês, deve escolher a opção de inglês na escolha da língua.

3.4.7 Listar e criar questões

Terá como interveniente o editor de jogos. Este pode ver uma lista das questões e pode criar questões sem associar diretamente a um jogo, mais tarde estas questões podem ser associadas a jogos, ao editar um jogo ou criar um jogo novo.

3.4.8 Escolher um jogo

Os intervenientes principais desta tarefa são os jogadores, mas qualquer outro utilizador com outro cargo pode realizar esta tarefa. Terão ao seu dispor uma lista de jogos disponíveis, podendo escolher o que pretendem jogar.

3.4.9 Jogar

Mais uma vez é uma tarefa dos jogadores, apesar de qualquer utilizador independentemente do cargo poder executá-la. Estes vão percorrendo questões, onde podem ter ajudas/pistas para as mesmas e vão ganhando pontos por questão. Estas questões estarão divididas por níveis. A qualquer momento, pode existir alguma espécie de recompensa no jogo (em forma de imagem ou texto), de acordo com o planeado pelo editor do jogo.

3.4.10 Área de utilizador

Os intervenientes da área de utilizador são todos os atores. Sendo que para Administradores e Editores de jogos é uma área de utilizador comum, com os seus dados e de possível edição. Para o jogador, terá todas as características anteriores, mas ainda será possível consultar os pontos acumulados em jogos.

3.5 Funcionalidades

Feito a análise das tarefas que integram a plataforma, nesta secção pretende-se pormenorizar a descrição das funcionalidades da mesma. Para que seja de mais fácil compreensão esta

secção, optou-se por dividir o sistema em três blocos:

- Geral: funcionalidades que qualquer utilizador pode executar;
- Administração: funcionalidades executadas só por administradores;
- Gestão de jogos: funcionalidades executadas por editores de jogos;

Serão apresentadas as funcionalidades, por blocos lógicos com uma breve descrição e também será feita uma descrição sumária dos casos de uso (use cases). Num sistema de informação os casos de uso são uma peça fundamental, pois representam a interação do utilizador com o sistema e mostram os objetivos da plataforma [28]. Especificam ainda como se deve comportar o sistema, mostrando também alternativas caso haja algum erro. A descrição detalhada deste processo estará no Anexo A.

3.5.1 Geral

Neste bloco encontramos as funcionalidades gerais, que estão divididas em três funcionalidades globais: sessão, conta e jogo. A funcionalidade global "Jogo" está incluída neste bloco pois todos os utilizadores independentemente do cargo, têm acesso à aplicação móvel e podem usufruir dos jogos disponíveis. Podem ser consultadas todas as funcionalidades com respetiva descrição na Tabela 3.2

	Funcionalidades	Descrição
Sessão	Iniciar sessão	Autenticação do utilizador que lhe dará acesso às respetivas funcionalidades
	Terminar sessão	Termina autenticação no sistema
Conta	Registar	Registo de um novo utilizador, por defeito fica com o cargo de jogador só sendo possível alterar o cargo através do administrador
	Alterar password	Qualquer utilizador pode alterar a sua password
	Editar conta	Todos os dados associados à conta de utilizador são editáveis
	Ver pontuação	Todos os utilizadores que tenham jogado algum jogo na plataforma, podem consultar as suas pontuações
Jogo	Ver jogos	Listagem de todos os jogos disponíveis para jogar na plataforma
	Jogar	Após escolher o jogo, jogar o mesmo.

Table 3.2: Funcionalidades gerais

Relativamente à sessão, todos os atores têm acesso às funcionalidades descritas. A Figura 3.1 representa isso mesmo.

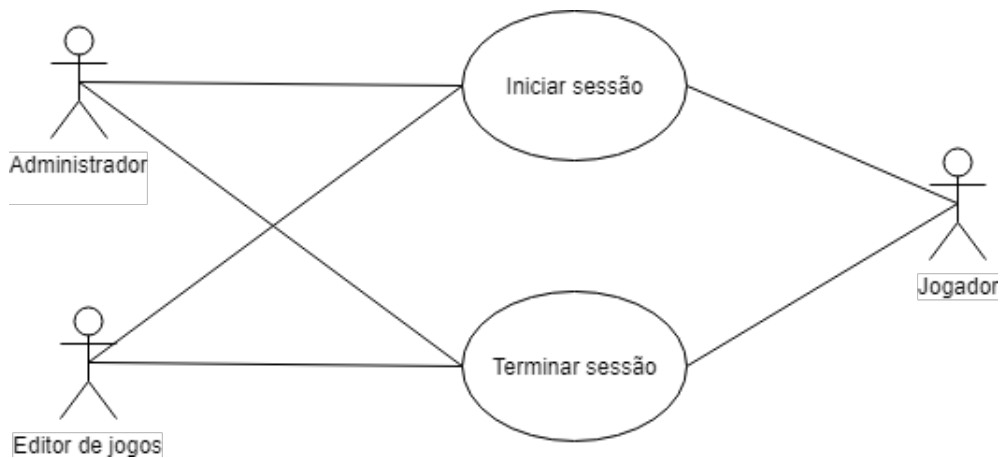


Figure 3.1: Funcionalidades pertencentes à Sessão

Em relação ao conjunto de funcionalidades da Conta, também todos os atores tem acesso a todas as funcionalidades, tal como descrito na Figura 3.2

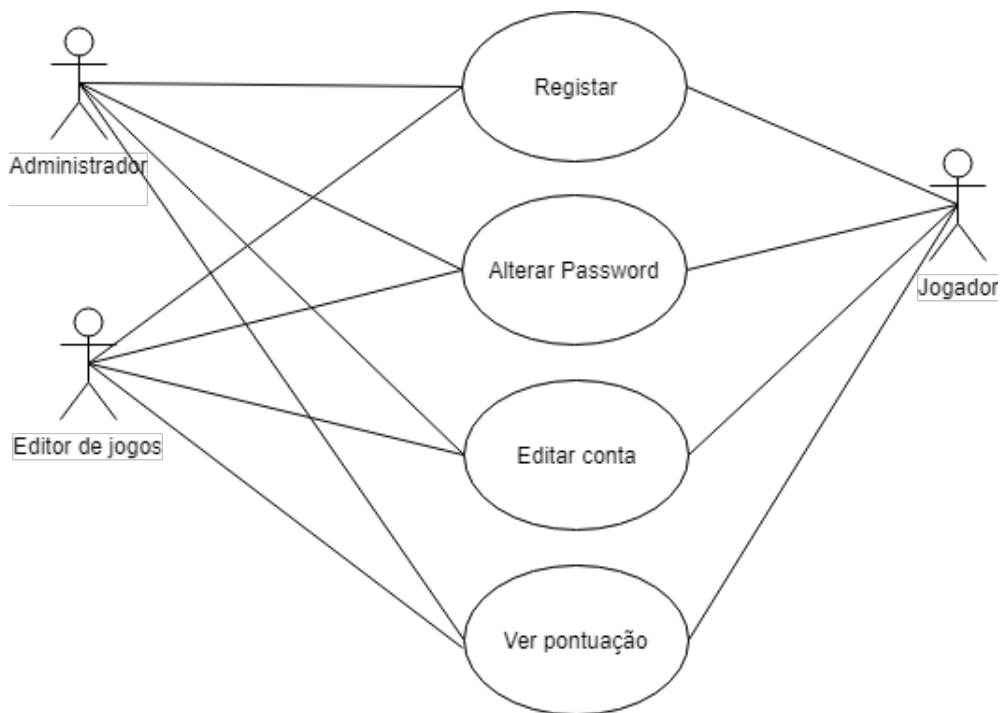


Figure 3.2: Funcionalidades pertencentes à Conta

Por último, o conjunto de funcionalidades "Jogo", também acessíveis por qualquer utilizador, embora seja mais específico para utilizadores com o cargo "Jogador". Este último também descrito através da Figura 3.3

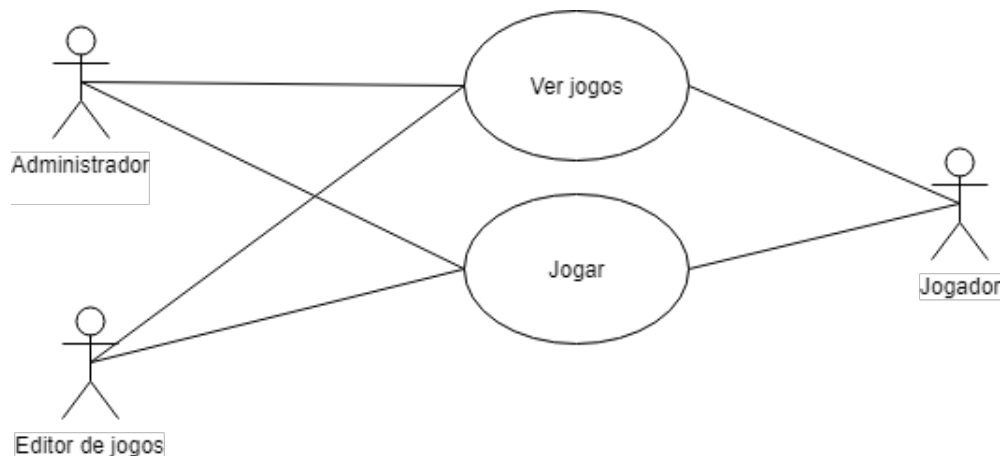


Figure 3.3: Funcionalidades pertencentes ao Jogo

3.5.2 Administração

Este bloco é composto pela funcionalidade Administração que inclui todas as tarefas relativas ao trabalho do administrador. Esta funcionalidade subdivide-se em 5 funcionalidades: Jogo, Utilizadores, Cargos, Língua, Questões, conforme enumerado e descrito na Tabela 3.3.

	Funcionalidades	Descrição
Jogo	Apagar jogo	O administrador pode apagar qualquer jogo da plataforma
Utilizadores	Apagar utilizador	Pode apagar qualquer utilizador
Cargos	Alterar cargo de utilizador	Por definição todos são jogadores, mas este cargo pode ser alterado
	Criar novo cargo	Permite criar mais cargos na plataforma
	Apagar cargo	Pode apagar qualquer cargo que exista
	Editar cargo	Edita o cargo, dando-lhe outro nome
	Ver cargos	Listagem dos cargos
Língua	Adicionar língua	Pode adicionar novos idiomas
	Apagar língua	Permite apagar um ou mais idiomas
	Editar língua	Edita qualquer nome e código de idioma
	Ver línguas	Listagem de todos os idiomas
Questões	Ver tipos de pergunta	Listagem de todos os tipos de perguntas

Table 3.3: Funcionalidades de administração

Todas estas funcionalidades são apenas permitidas ao(s) administrador(es) do sistema. A Figura 4.3 representa isso mesmo.

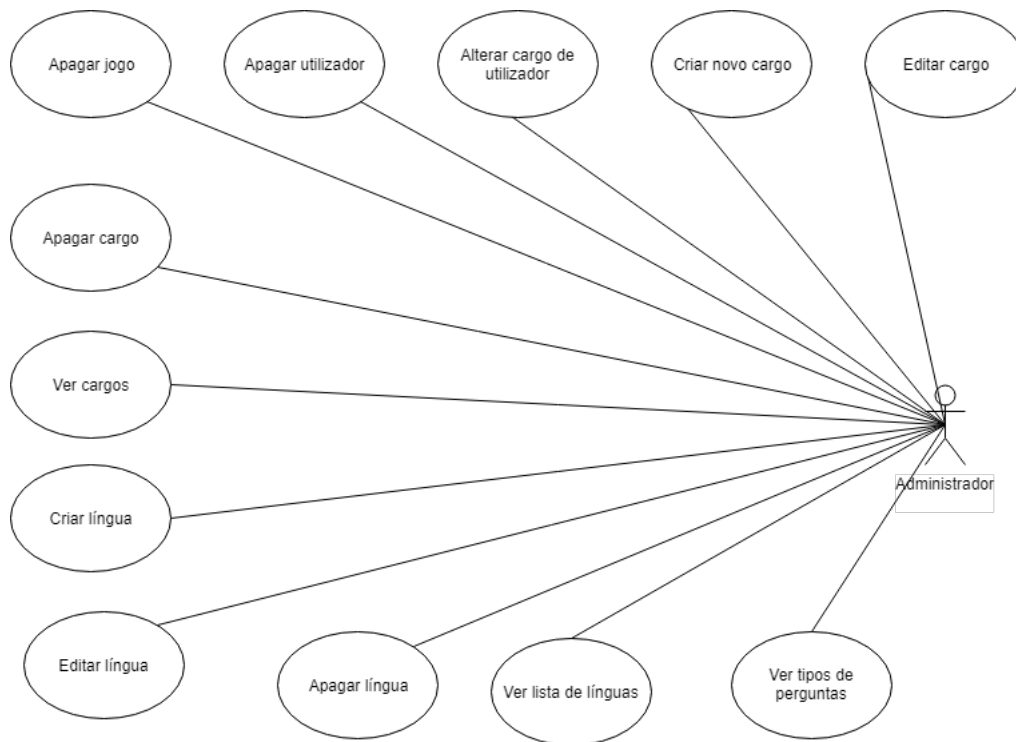


Figure 3.4: Funcionalidades pertencentes à administração

3.5.3 Gestão de Jogos

Por último, temos o bloco de Gestão de Jogos. As funcionalidades pertencentes a este bloco, são realizadas pelo ator Editor de Jogos. Na Tabela 3.4 estão descritas essas funcionalidades.

	Funcionalidades	Descrição
Gestão de jogos	Adicionar um jogo	É a primeira parte do processo de criar um jogo, dar-lhe o nome, descrição, imagem..
	Adicionar um nível	Adicionar níveis a jogos
	Adicionar questão	Adicionar questões a cada nível. Este pode ser um processo independente, onde são criadas várias questões e só mais tarde adicionadas ao jogo
	Editar jogo	Editar o jogo implica apenas mudar nome, descrição, etc
	Editar nível	Editar o nome e número do nível
	Editar questão	Permite editar o tipo de questão e seus parâmetros
	Apagar jogo	Apaga o jogo e respetivos níveis, as perguntas mantêm-se na lista de perguntas
	Apagar nível	Apaga o nível
	Apagar questão	Apaga uma questão
	Ver lista de questões	Permite ver uma lista de perguntas criada pelo editor
	Ver lista de jogos	Permite ver uma lista de jogos criados pelo editor

Table 3.4: Funcionalidades de Gestão de jogos

A Figura 3.5 descreve o referido anteriormente.



Figure 3.5: Funcionalidades pertencentes à gestão de jogos

3.6 Modelo Relacional

Após fazer o levantamento de requisitos necessários à construção da plataforma, pretende-se desenvolver o modelo conceptual que será usado para orientar o desenvolvimento do projeto. A abordagem mais conhecida é o modelo Entidade-Relação (MER), que é um modelo conceptual de dados de alto-nível. [10]

O MER é um método que descreve e define um sistema. Este sistema contém componentes que estão interligados através de relações. Estas relações caracterizam dependências e exigências entre as mesmas. Por sua vez, os componentes são definidos por um conjunto de propriedades. Este modelo é representado por um Diagrama Entidade-Relação (DER), que contém entidades, relações e atributos. As relações são algo que existe, as relações são entre duas ou mais entidades e os atributos definem entidades.

Posto isto, e uma vez que a junção de todas as funcionalidades da plataforma é complexa, a solução final vai ser apresentada de forma fracionada, respeitando as regras impostas, recorrendo a DER. Isto facilita-nos a percepção de todos os conceitos e as suas ligações.

Dividiu-se o DER em 3 pares. A primeira parte, "Jogo", ilustra o semi-diagrama no qual se descrevem todas as entidades relacionadas com o mesmo; "Nível-Questões" ilustra as relações diretas entre um Nível e as questões que podem existir; por último, "Utilizador" mostra todas as relações com o mesmo, explicando a importância de cada uma no sistema.

No Anexo B encontra-se uma descrição detalhada de cada atributo das várias relações. Os atributos estão apenas descritos neste anexo, uma vez que ficávamos com DER demasiado exaustivos, perdendo o propósito da sua explicação base.

3.6.1 Jogo

Para ter um Jogo bem definido e com suporte mais alargado, foram concebidas várias entidades, conforme ilustrado na Figura 3.6. Existe a entidade central, que é o Jogo. Esta por sua vez estabelece a atribuição de um id de jogo que identifica cada novo jogo que é criado na plataforma.

Surgiu então a necessidade de criar uma entidade Nível, pois faz parte da estrutura de alguns jogos a existência de níveis e foi algo definido nos requisitos do sistema. Foi criada também uma entidade "Estado de jogo", que é um booleano que define se o jogo está acabado ou não. A entidade Idioma de jogo, existe com o propósito de um jogo ter determinado idioma e assim ser definido para o utilizador final qual o idioma do jogo. Por fim, a entidade "Utilizador do Jogo", guarda pontuações obtidas dos jogos que forem jogados e mostra inícios e fins de sessões.

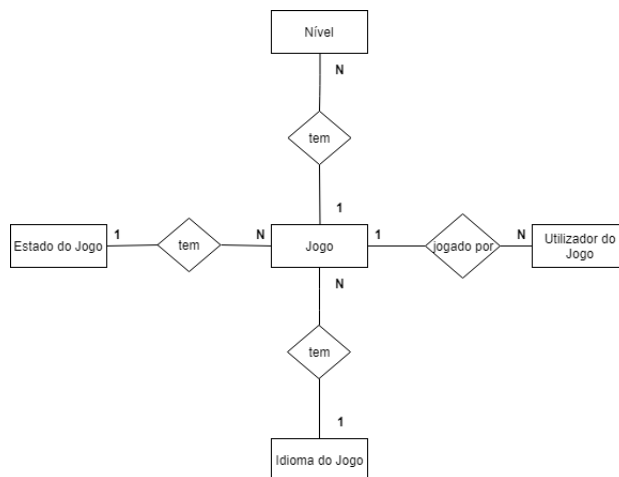


Figure 3.6: DER Jogos

3.6.2 Nível-Questões

Para que seja possível a existência de jogos, não basta haver níveis, é também necessário haver conteúdo nesses níveis e por isso foram criadas entidades de questões que podem estar presentes no nível. A entidade Tipo de Questões tem o atributo "questionTypeId" para garantir que a entidade nível vai buscar a questão à entidade certa.

Por sua vez, as entidades "Bullet Questions", "Matching Questions" e "Ordering Questions", são questões com várias opções e por isso tem uma entidade própria onde são definidas as suas opções para cada questão. A entidade "Bullet Questions" tem ainda a entidade "Tipo" para definir o tipo de "Bullet Question", pois existem vários tipos.

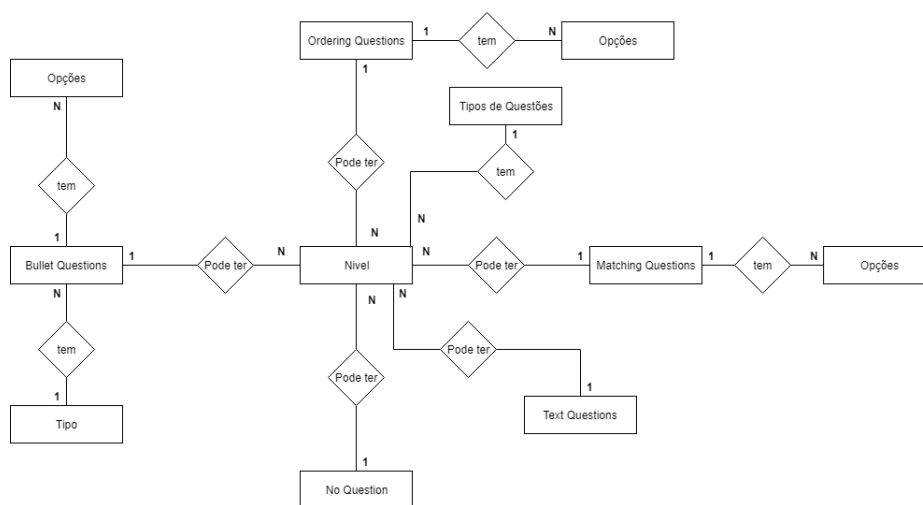


Figure 3.7: DER Nível-Questões

3.6.3 Utilizador

Por último, importa especificar o DER de utilizadores. Em que a entidade "Utilizador do Jogo" está ligada à entidade "Jogo" (Figura 3.6). Esta entidade serve para guardar inícios e fins de sessões, pontuações e jogos jogados dos utilizadores que usam a interface móvel. Esta entidade está relacionada com a entidade "Utilizador Genérico", pois é através da "Utilizador Genérico" que se faz o login, tanto no backoffice como na interface móvel. Também existe uma entidade "Papel", que serve para atribuir a cada utilizador um papel na plataforma.

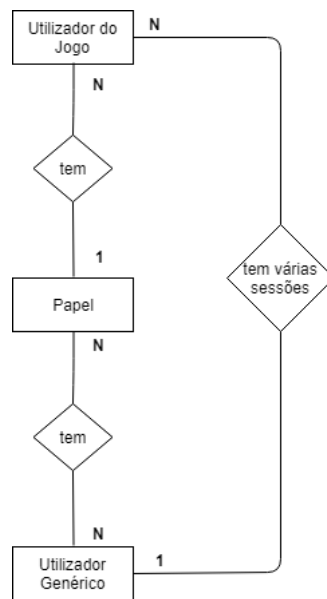


Figure 3.8: DER Utilizadores

4 | Desenvolvimento do protótipo

No capítulo anterior foram abordados os requisitos e tarefas inerentes à concretização desta plataforma. Torna-se agora importante definir como os atores se vão relacionar com os dados. Para isto desenvolveu-se uma interface dinâmica e com usabilidade, que suporta as tarefas analisadas anteriormente. Esta interface será agora discutida no presente capítulo.

4.1 Visão Geral

Após fazer o estudo das tecnologias disponíveis para o desenvolvimento desta plataforma no Capítulo 2, foram selecionadas as seguintes tecnologias:

- Para o desenvolvimento da base de dados foi escolhido o SQL Server 2017
- Para o backoffice foi escolhida a framework ASP.NET MVC
- Para a Web API foi escolhida ASP.NET Web API
- Para a interface móvel foi escolhida a framework Durandal.

São usadas também outras tecnologias, nomeadamente:

- É usado o AmplifyJS para utilização de armazenamento local;
- Bootstrap para tornar a aplicação mais responsiva;
- TouchSwipe que dá suporte à utilização das facilidades de arrastamento e multi-toque em dispositivos móveis;
- PhoneGap para criar aplicações em diversas plataformas (Android, iOS, etc). [29]

Para começar foi construída a base de dados em SQL. Para consultar o modelo da base de dados ver Anexo C. O segundo passo, foi construir a interface do Backoffice seguindo as normas do modelo MVC, descrito no Capítulo 2. De seguida construiu-se a Web API. Por último, construiu-se a aplicação móvel.

Para que a plataforma seja dinâmica e com usabilidade, foram estudados os métodos de avaliação heurística de Jakob Nielsen [30]. Este método obedece a algumas regras de forma a obter uma interface usável, que contribui bastante para a usabilidade da plataforma se forem seguidas durante a construção da mesma. O conjunto de heurísticas é o seguinte [30]:

- Visibilidade do estado do sistema: os utilizadores mantêm-se informados sobre os acontecimentos. Existe um feedback.
- Semelhança entre o sistema e o mundo real: O sistema deve falar a língua dos utilizadores, com palavras, frases e conceitos familiares para o mesmo. Deve seguir as convenções do mundo real, fazendo com que a informação pareça natural e com uma ordem lógica.
- Controlo e liberdade: Por vezes os utilizadores escolhem funções do sistema por engano e devem ter a oportunidade de abandonar essa tarefa caso o desejem.
- Prevenção de erros: A existência de mensagens de erro é importante, mas mais importante que isso é existir sempre a possibilidade de prevenir o erro.
- Consistência: Os utilizadores não devem ter de adivinhar que palavras, situações ou ações descritas de forma diferente, significam o mesmo. Por isso para a mesma tarefa, deve-se definir sempre as mesmas palavras.
- Reconhecimento: As instruções de uso e dados necessários devem ser visíveis e facilmente reconhecíveis, pois o utilizador precisa de recorrer à memória para executar ações.
- Flexibilidade e eficiência de uso: O sistema deve ter a possibilidade de atalhos para utilizadores mais experientes.
- Recuperar de erros: O sistema deve ter mensagens de erro claras, que identifiquem com precisão o problema.
- Minimalista: eliminar conteúdos irrelevantes ou que raramente são usados.
- Ajuda e documentação: O sistema deve ser intuitivo o suficiente para que a ajuda e documentação seja dispensada, no entanto é importante existir a possibilidade de obter ajuda ou ler a documentação.

Cumprindo estas heurísticas, contribui-se para a qualidade e usabilidade do produto final. Pode ser um processo complexo e que pode requerer a reescrita de partes da plataforma, mas ainda assim compensatório.

4.1.1 Área de trabalho

A interface da plataforma de jogos multi* foi projetada de forma a ter uma interação semelhante à mostrada na Figura 4.1. Está dividida em 5 pontos: Menu, Conta, Cabeçalho da página, Área de trabalho e Rodapé.

No menu o utilizador tem acesso às funcionalidades permitidas de acordo com o seu cargo. Na conta pode gerir as suas informações pessoais. O cabeçalho da página, indica o nome da

página atual de trabalho. Por fim, o rodapé contém algumas informações sobre o produto desenvolvido.

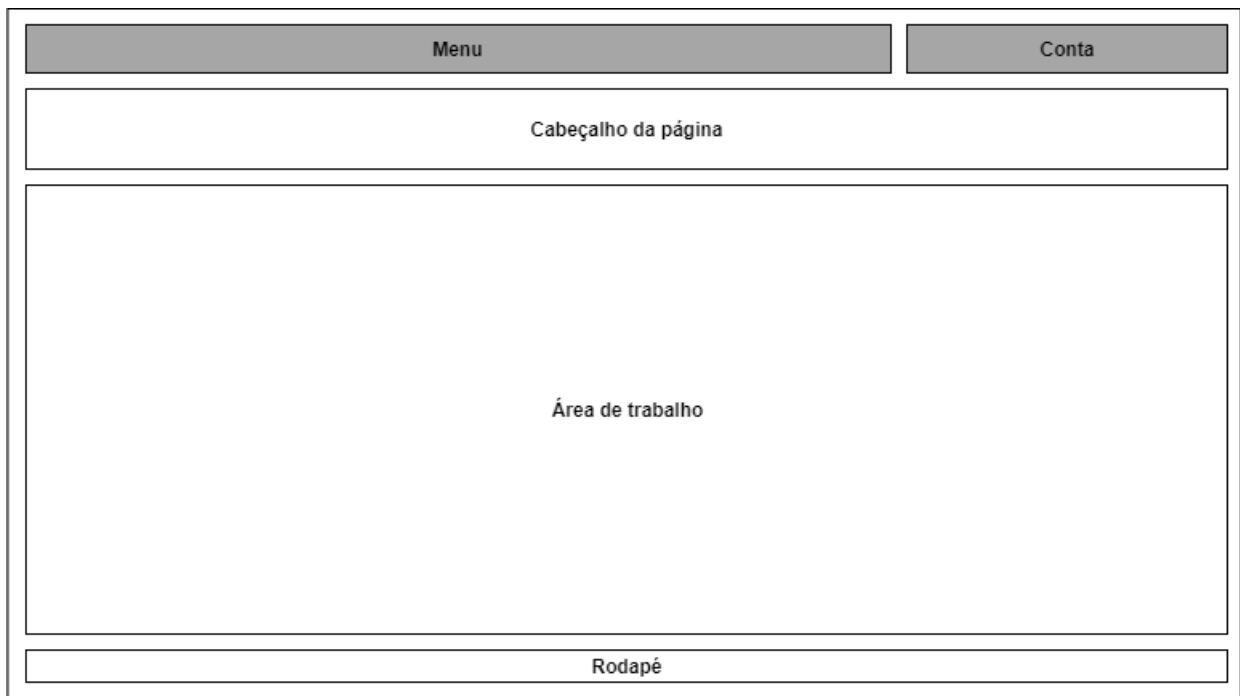


Figure 4.1: Protótipo da interação do sistema

Esta área de trabalho adapta-se facilmente às características do dispositivo - se for visualizada num smartphone, adaptar-se-á às dimensões do mesmo. Isto para dizer que tanto para o backoffice como para a aplicação móvel, a área de trabalho será semelhante à apresentada na Figura 4.1.

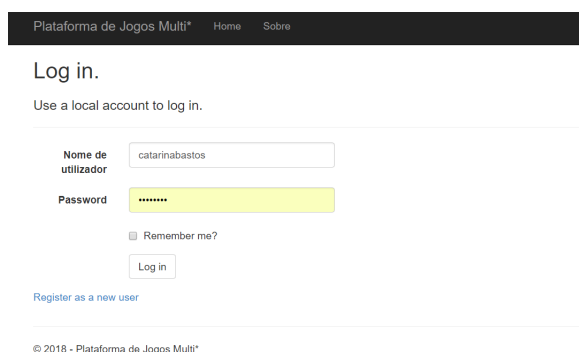
Importante de salientar que, para facilitar o uso da plataforma pelo utilizador, foram usadas ferramentas que poupam tempo e evitam erros. Para a validação e processamento de dados, foi usado JavaScript. A validação e retificação da informação introduzida pelos utilizadores antes de ser guardada é importante para que não existam erros e problemas de incompatibilidade. Esta validação é feita em tempo real. Um dos casos mais relevantes é o dos jogadores: é importante que haja validação dos dados que são inseridos. Por exemplo, se num jogo houver questões de matemática em que é necessário inserir números, é crucial que haja uma validação em tempo real que o jogador está a colocar números e não letras.

4.2 Tarefas

4.2.1 Backoffice

Nesta secção serão mostradas as tarefas principais do Backoffice, assim como imagens de como ficaram implementadas na plataforma.

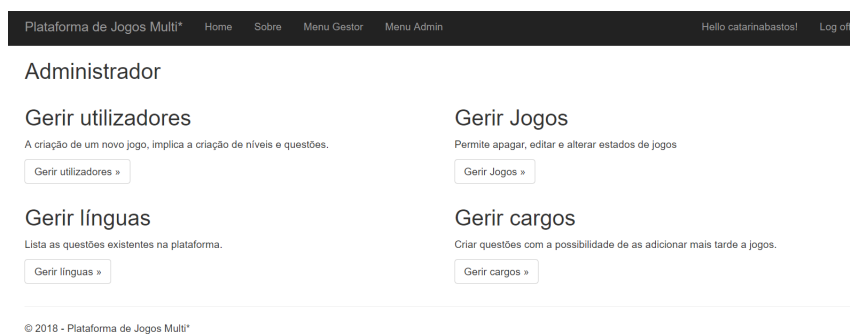
O Backoffice permite fazer login e registo de novos utilizadores.



The screenshot shows the login interface of the 'Plataforma de Jogos Multi*'. At the top, there is a dark navigation bar with the site name and links for 'Home' and 'Sobre'. Below this, the heading 'Log in.' is followed by the instruction 'Use a local account to log in.'. The form contains two input fields: 'Nome de utilizador' with the value 'catarinabastos' and 'Password' with masked characters. A 'Remember me?' checkbox is present below the password field. A 'Log in' button is at the bottom of the form. A link 'Register as a new user' is located below the login button. At the very bottom, the copyright notice '© 2018 - Plataforma de Jogos Multi*' is displayed.

Figure 4.2: Login

Importa mostrar o painel de administrador e o painel do gestor de jogos. Tal como mostrado na Figura 4.3, o administrador tem acesso a tarefas como: gerir utilizadores, gerir jogos, gerir as línguas possíveis de existir na plataforma e gestão dos cargos de cada utilizador.



The screenshot displays the administrator dashboard. The top navigation bar includes the site name, 'Home', 'Sobre', 'Menu Gestor', 'Menu Admin', and a user greeting 'Hello catarinabastos!' with a 'Log off' link. The main content area is titled 'Administrador' and features four management options, each with a brief description and a button: 'Gerir utilizadores' (description: 'A criação de um novo jogo, implica a criação de níveis e questões.'), 'Gerir Jogos' (description: 'Permite apagar, editar e alterar estados de jogos'), 'Gerir línguas' (description: 'Lista as questões existentes na plataforma.'), and 'Gerir cargos' (description: 'Criar questões com a possibilidade de as adicionar mais tarde a jogos.'). The copyright notice '© 2018 - Plataforma de Jogos Multi*' is at the bottom.

Figure 4.3: Menu Administrador

No menu de gestor de jogos, temos a possibilidade de criar um novo jogo (que implica a criação de níveis também), lista de questões, criar questões de forma autónoma à criação de um jogo e gerir os jogos (permitindo editar, apagar e ver detalhes dos jogos). Figura 4.4

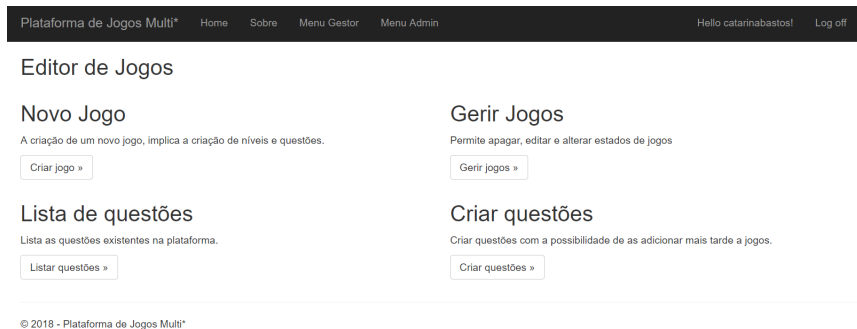








Figure 4.4: Menu do gestor de jogos

A plataforma tem também várias listas onde é possível ver o conteúdo de coisas importantes, como jogos, línguas existentes, entre outras. Todas as listas são similares, com ligação para a edição, detalhes e apagar determinado conteúdo da lista. Figura 4.5 Figura 4.6 Figura 4.7

Plataforma de Jogos Multi* Home Sobre Menu Gestor Menu Admin Hello catarinabastos! Log off

Index

[Criar novo jogo](#)

Name	Description	Game icon (url)	Game logo (url)	Language code	State name
Jogos da Nintendo	Tenta acertar no nome dos jogos, de acordo com as imagens que aparecem!			PT	Not Ready Editar Detalhes Apagar
História de Portugal	Quiz de			PT	Ready Editar Detalhes Apagar
Matemática	Contas de matemática			PT	Ready Editar Detalhes Apagar

© 2018 - Plataforma de Jogos Multi*

Figure 4.5: Lista de jogos

Language code	Language name	
PT	Portuguese	Edit Details Delete
EN	English	Edit Details Delete

Figure 4.6: Lista de idiomas

Email	PhoneNumber	UserName	ProfilePicture
joao@ua.pt		joao	Details Delete
admin@teste.com		admin	Details Delete
catarinabastos@ua.pt		catarinabastos	Details Delete
cenaxyzx@gmail.pt		cenaxyzx@gmail.pt	Details Delete
ci@ua.pt		catarinab	Details Delete
mariana@ua.pt		mariana	Details Delete
cat@ua.pt		cat@ua.pt	Details Delete
bern@ua.pt		bern@ua.pt	Details Delete
maria@ua.pt		mariaabla	Details Delete

Figure 4.7: Lista de utilizadores

Por fim, vemos como é possível criar jogos, todas as outras páginas onde se cria perguntas, níveis, línguas tem um aspeto semelhante à da Figura 4.8.

The screenshot shows the 'Create Game' form in the 'Plataforma de Jogos Multi*' web application. The form includes the following fields and options:

- Name:** A text input field.
- Description:** A text input field.
- languageid:** A dropdown menu currently set to 'PT'.
- Game icon (url):** A file selection button labeled 'Escolher ficheiro' with the text 'Nenhum fic...elecionado'.
- Game logo (url):** A file selection button labeled 'Escolher ficheiro' with the text 'Nenhum fic...elecionado'.
- stateid:** A dropdown menu currently set to 'Ready'.
- Create:** A button to submit the form.

Below the form, there is a link 'Voltar à lista' and a footer '© 2018 - Plataforma de Jogos Multi*'.

Figure 4.8: Criar jogo

4.2.2 Interface do utilizador final

Nesta secção será mostrado as tarefas principais da aplicação móvel, assim como imagens de como ficaram implementadas na plataforma.

A Figura 4.9 e a Figura 4.10 mostram o login e o registo na aplicação mobile.

The screenshot shows the login screen of the mobile application. The interface includes:

- A hamburger menu icon in the top right corner.
- The application title 'Plataforma de jogos multi*' with a small icon to the left.
- The heading 'Login'.
- A text input field for 'Username'.
- A text input field for 'Password'.
- A blue 'Submit' button.

Figure 4.9: Login

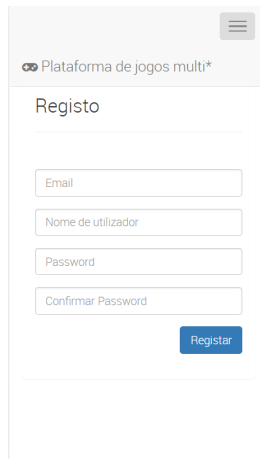


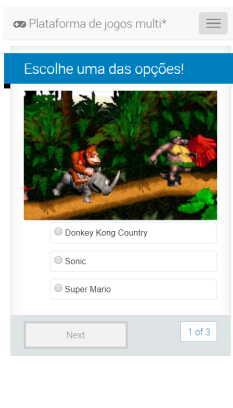
Figure 4.10: Registo

Depois é possível ver os jogos disponíveis para jogar na plataforma, tal como é mostrado na Figura 4.9

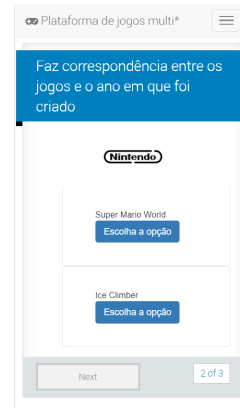


Figure 4.11: Lista de jogos

Após escolher o jogo, o utilizador inicia o jogo. Na Figura 4.12a, Figura 4.12b, Figura 4.13a é possível ver uma pergunta do tipo "Bullet Question", "Text Question" e "Matching Question".

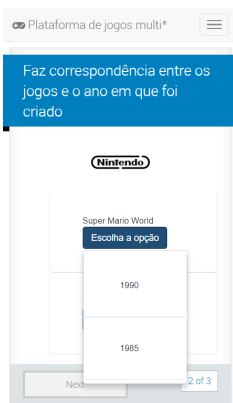


(a) Bullet Question

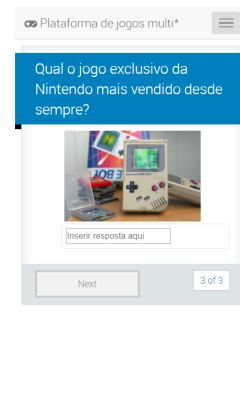


(b) Matching Question

Figure 4.12: Grupo de imagens



(a) Matching Question



(b) Text Question

Figure 4.13: Grupo de imagens

4.3 Resultados

Nesta secção faz-se uma análise dos resultados e das metas atingidas.

Relativamente a objetivos gerais traçados numa fase inicial desta dissertação pode-se afirmar que foram cumpridos. Definiu-se uma arquitetura tendo em conta o desempenho, a segurança e o fim a que se destina. As tarefas/processos foram implementados de acordo com o inicialmente proposto e criou-se uma plataforma simples, organizada e útil. A escolha das tecnologias também contribuiu para que não fosse exigido um estudo muito exaustivo, pois eram de aprendizagem intuitiva e com conhecimentos já adquiridos em disciplinas do curso de Engenharia de Computadores e Telemática. Isto ajudou a conseguir ir de encontro às necessidades do utilizador.

Do ponto de vista dos conceitos abordados nesta dissertação, "jogos sérios" e "joguificação", criou-se uma plataforma que cumpre os seus requisitos básicos. Sendo sobretudo uma plataforma de aprendizagem ou para mostrar conceitos adquiridos, temos toda uma estrutura que permite uma interação "joguificada", através de pontos e "recompensas" (imagens, texto, ...). Esta plataforma é bastante versátil, no sentido em que os jogos não têm de ter um tema definido, podendo abordar os mais vastos temas. Pode ser útil numa escola ou até num contexto do mercado de trabalho específico. Este projeto transporta de certa forma o conceito tradicional de uma plataforma de elearning, para uma plataforma joguificada, mais interativa e com mais tipos de questões envolvidas, sendo que maior parte das plataformas elearning têm limitações quanto aos tipos de questões existentes.

Outro ponto forte desta plataforma é a facilidade de integração com sistemas já existentes, seja para alojar num servidor ou pelo uso das tecnologias usadas (Visual Studio e SQL Server), o que permite a escalabilidade do sistema. As políticas de usabilidade implementadas nesta plataforma também podem ser consideradas um ponto forte, uma vez que torna fácil a manutenção do sistema através dos seus administradores.

5 | Conclusão

A necessidade de construir esta plataforma exigiu, numa primeira fase, o estudo da área de aplicações web. Com isto, aprofundou-se conhecimentos em várias linguagens de programação como ASP.Net, SQL, C#, HTML e Javascript. Foram usadas outras ferramentas como PhoneGap e a framework de Javascript, Durandal. Para o desenvolvimento da plataforma foram usadas duas ferramentas: Microsoft Visual Studio e SQL Management Studio. Estas ferramentas foram fáceis de usar, uma vez que já eram conhecidas, não necessitando de ser exploradas aprofundadamente.

Foi feita uma análise dos conceitos "jogos sérios" e "joguificação" e da sua importância num contexto de aprendizagem. Esta análise foi importante, no sentido de definir tarefas para a plataforma e aprender sobre os impactos dos "jogos sérios" na aprendizagem, estando provados os seus benefícios. Como vantagens desta plataforma temos uma centralização de jogos dado um real contexto: a diversão inerente na aprendizagem ou mostra de conhecimentos e a sensação de recompensa pelo facto de conseguir concluir níveis/jogos. A maior desvantagem, prende-se com o ceticismo que possa existir em relação ao conceito de "jogos sérios".

No desenvolvimento de todo este trabalho, investigação e implementações necessárias para atingir os objetivos propostos, surgiram fatores facilitadores e inibidores. Em relação aos fatores facilitadores, os conhecimentos já adquiridos em relação às linguagens e tecnologias de programação anteriormente adquiridos, ajudaram ao avanço da plataforma. Tal como o contributo científico e académico por parte de professores e colegas. O maior fator inibidor foi o de criar e estruturar um modelo de dados claro e abrangente, de forma a lidar eficazmente com o armazenamento da informação da plataforma.

5.1 Trabalho futuro

Para trabalho futuro seria útil a implementação de um sistema GPS que permitiria transportar o conceito de "jogos sérios" para o exterior, isto é, o utilizador poderia iniciar o jogo num ponto e ir seguindo um trajeto até acabar o jogo. Outro ponto importante, será a inserção de mais elementos de "joguificação", de forma a tornar ainda mais atrativa a plataforma. A inserção de estatísticas de jogos também seria útil para os utilizadores.

Por fim, e uma vez que não houve possibilidade de efetuar testes de usabilidade aprofundados, seria importante a sua realização para que após uma análise dos resultados obtidos

fosse possível uma melhorar a plataforma.

Referências

- [1] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM, 2011.
- [2] Jesper Juul. The game, the player, the world: Looking for a heart of gameness. *PLURAIIS-Revista Multidisciplinar*, 1(2), 2010.
- [3] Andrzej Marczewski. *Gamification: a simple introduction*. Andrzej Marczewski, 2013.
- [4] Two types of #gamification « karl kapp. <http://karlkapp.com/two-types-of-gamification/>. (Accessed on 2018).
- [5] The birth of gamification (history of gamification pt.2). <https://tinyurl.com/y7xg48uq>. (Accessed on 2018).
- [6] Houda Mouaheb, Ahmed Fahli, Mohammed Moussetad, and Said Eljamali. The serious game: what educational benefits? *Procedia-Social and Behavioral Sciences*, 46:5502–5508, 2012.
- [7] Novabase lança plataforma nova. <http://www.novabase.pt/pt/dp/novabase-lanca-plataforma-nova>. (Accessed on 2018).
- [8] Popular ios app quizup about to arrive on android | one click root. <https://www.oneclickroot.com/android-apps/popular-ios-app-quizup-about-to-arrive-on-android/>. (Accessed on 2018).
- [9] 7_ti2008-teoria_bd. <https://www.iseg.ulisboa.pt/aquila/getFile.do?fileId=8207&method=getFile>. (Acedido em 2018).
- [10] Ramez Elmasri and Shamkant Navathe. *Fundamentals of database systems*. Addison-Wesley Publishing Company, 2010.
- [11] Db-engines ranking - popularity ranking of database management systems. <https://db-engines.com/en/ranking>. (Acedido em 2018).
- [12] Oracle vs. sql server face-off: Microsoft sql cheaper, simpler than oracle db. <https://searchsqlserver.techtarget.com/tip/Oracle-vs-SQL-Server-faceoff-Microsoft-SQL-cheaper-simpler-than-Oracle-DB>. (Accessed on 10/24/2018).

- [13] Microsoft azure sql database provides unparalleled data security in the cloud with always encrypted | blog | microsoft azure. <https://tinyurl.com/ybsfk4lo>. (Accessed on 10/24/2018).
- [14] Desenvolvimento de backoffice / gestor de conteúdos. <https://www.pontocode.com/desenvolvimento-gestor-conteudos>. (Accessed on 10/25/2018).
- [15] What is a dynamic website? <https://www.computerhope.com/jargon/d/dynasite.htm>. (Accessed on 10/25/2018).
- [16] Php vs asp.net performance comparison. <https://www.comentum.com/php-vs-asp-net-comparison.html>. (Accessed on 10/25/2018).
- [17] Php vs asp.net: Top differences for beginners' understanding. <https://appinventiv.com/blog/php-vs-asp-net>. (Accessed on 10/25/2018).
- [18] Php: sqlsrv_connect - manual. <http://php.net/manual/en/function.sqlsrv-connect.php>. (Accessed on 10/25/2018).
- [19] Asp.net overview | microsoft docs. <https://docs.microsoft.com/pt-pt/aspnet/overview>. (Accessed on 10/25/2018).
- [20] John Deacon. Model-view-controller (mvc) architecture. *Online* [Citado em: 10 de março de 2006.] <http://www.jdl.co.uk/briefings/MVC.pdf>, 2009.
- [21] Asp.net mvc 5 | microsoft docs. <https://docs.microsoft.com/pt-pt/aspnet/mvc/mvc5>. (Accessed on 10/25/2018).
- [22] Opensoft. Quais as diferenças entre aplicações nativas, web e híbridas?, 2018.
- [23] Asp.net - single-page applications: Build modern, responsive web apps with asp.net. <https://msdn.microsoft.com/pt-pt/magazine/dn463786.aspx>. (Accessed on 10/25/2018).
- [24] Knockout vs react detailed comparison as of 2018 - slant. https://www.slant.co/versus/33/10513/~knockout_vs_react. (Accessed on 2018).
- [25] Durandal vs angular – a real world comparison – martijn boland. <https://blogs.taiga.nl/martijn/2013/12/19/durandal-vs-angular-a-real-world-comparison/>. (Accessed on 2018).
- [26] Which one is better for a scalable application, angular or react? - quora. <https://www.quora.com/Which-one-is-better-for-a-scalable-application-Angular-or-React>. (Accessed on 2018).
- [27] Angular vs react—the deal breaker – hacker noon. <https://hackernoon.com/angular-vs-react-the-deal-breaker-7d76c04496bc>. (Accessed on 2018).
- [28] Ivar Jacobson, Ian Spence, and Kurt Bittner. Use case 2.0: The guide to succeeding with use cases. *Ivar Jacobson International*, 49:50, 2011.

- [29] Renato S. Pinto Joaquim S. Pinto. Plataforma de jogos multi*. page 6, 2015.
- [30] 10 heuristics for user interface design: Article by jakob nielsen. <https://www.nngroup.com/articles/ten-usability-heuristics/>. (Accessed on 10/28/2018).

A | Anexo A

A.1 Descrição dos casos de uso

Caso de uso	Iniciar sessão
Breve descrição	Autenticação do utilizador que lhe dará acesso às respetivas funcionalidades.
Pré-condições	Ator tem de estar registado na plataforma.
Sequência típica	Preenchimento dos campos obrigatórios (email, password). Validação do login.
Sequência alternativa	Erro no login. Utilizador ainda não registado ou login errado.
Pós-condições	Início de sessão bem sucedido, utilizador tem acesso às suas respetivas funcionalidades na plataforma.

Table A.1: Descrição detalhada de Iniciar sessão

Caso de uso	Terminar sessão
Breve descrição	Termina autenticação na plataforma
Pré-condições	Ator tem de estar autenticado na plataforma
Sequência típica	Clicar em terminar sessão
Sequência alternativa	-
Pós-condições	Término da sessão bem sucedido. O utilizador não tem acesso às especificidades da plataforma

Table A.2: Descrição detalhada de Terminar sessão

Caso de uso	Registrar
Breve descrição	Registo de um novo utilizador, por defeito fica com o cargo de jogador só sendo possível alterar o cargo através do administrador
Pré-condições	-
Sequência típica	Preenchimento dos campos obrigatórios para o registo.
Sequência alternativa	Erro no preenchimento de algum campo
Pós-condições	Registo concluído. O utilizador tem acesso à plataforma como jogador. Este cargo pode ser alterado pelo administrador

Table A.3: Descrição de Registrar

Caso de uso	Editar conta
Breve descrição	Qualquer utilizador pode alterar dados da conta
Pré-condições	Estar autenticado
Sequência típica	Alterar dados já existentes, preenchendo de novo. Confirmar alteração
Sequência alternativa	Erro no preenchimento de algum campo
Pós-condições	A conta fica com dados alterados

Table A.4: Descrição de Editar Conta

Caso de uso	Ver pontuação
Breve descrição	Todos os utilizadores que tenham jogado algum jogo na plataforma, podem consultar as suas pontuações
Pré-condições	Estar autenticado e ter concluído algum jogo.
Sequência típica	O utilizador consulta nas suas informações pessoais o número de pontos que tem.
Sequência alternativa	O utilizador nunca jogou nenhum jogo, não tem qualquer informação de pontos
Pós-condições	-

Table A.5: Descrição de Ver pontuação

Caso de uso	Ver jogos
Breve descrição	Listagem de todos os jogos disponíveis para jogar na plataforma
Pré-condições	Estar autenticado na aplicação móvel
Sequência típica	É apresentada uma lista de jogos, que o jogador pode jogar.
Sequência alternativa	-
Pós-condições	Clicando num jogo, o utilizador pode jogá-lo

Table A.6: Descrição de Ver jogos

Caso de uso	Jogar
Breve descrição	Após escolher o jogo, jogar o mesmo
Pré-condições	Escolher um jogo da lista de jogos
Sequência típica	Acertar nas questões
Sequência alternativa	Ao errar na questão, procurar pela opção de ajuda (caso exista)
Pós-condições	Jogo terminado, o jogador acumula pontos e talvez uma recompensa dependendo de como o editor de jogos construiu o mesmo

Table A.7: Descrição de Jogar

Caso de uso	Apagar jogo
Breve descrição	O administrador pode apagar qualquer jogo da plataforma
Pré-condições	Escolher um jogo da lista de jogos
Sequência típica	Apagar esse jogo
Sequência alternativa	-
Pós-condições	Jogo eliminado da lista de jogos

Table A.8: Descrição de Apagar jogo

Caso de uso	Apagar utilizador
Breve descrição	Pode apagar qualquer utilizador
Pré-condições	Escolher um utilizador da lista de utilizadores da plataforma
Sequência típica	Apagar esse utilizador
Sequência alternativa	-
Pós-condições	Utilizador eliminado da plataforma. Não poderá fazer login

Table A.9: Descrição de Apagar Utilizador

Caso de uso	Alterar cargo do utilizador
Breve descrição	Por definição todos são jogadores, mas este cargo pode ser alterado
Pré-condições	Escolher um utilizador da lista de utilizadores da plataforma
Sequência típica	Alterar o cargo desse utilizador
Sequência alternativa	-
Pós-condições	Utilizador muda de cargo, pode ter acesso à edição de jogos ou administração. Em alternativa os privilégios do utilizador podem ser menores, no caso de ser um editor de jogos ou administrador e passar a ser apenas um jogador.

Table A.10: Descrição de Alterar cargo do utilizador

Caso de uso	Criar novo cargo
Breve descrição	Permite criar mais cargos na plataforma
Pré-condições	Ser administrador
Sequência típica	Cria um cargo novo
Sequência alternativa	-
Pós-condições	Existe mais cargos na plataforma

Table A.11: Descrição de Criar novo cargo

Caso de uso	Apagar cargo
Breve descrição	Pode apagar qualquer cargo que exista
Pré-condições	Ser administrador. Ver lista de cargos
Sequência típica	Apaga o cargo
Sequência alternativa	-
Pós-condições	-

Table A.12: Descrição de Apagar cargo

Caso de uso	Editar cargo
Breve descrição	Edita o cargo, dando-lhe outro nome
Pré-condições	Ser administrador. Ver lista de cargos
Sequência típica	Editar o nome do cargo
Sequência alternativa	-
Pós-condições	Cargo com novo nome

Table A.13: Descrição de Editar cargo

Caso de uso	Adicionar língua
Breve descrição	Pode adicionar novos idiomas
Pré-condições	Ser administrador
Sequência típica	Adiciona um idioma novo e código
Sequência alternativa	-
Pós-condições	Novo idioma na lista de idiomas disponíveis

Table A.14: Descrição de Adicionar língua

Caso de uso	Apagar língua
Breve descrição	Permite apagar um ou mais idioma
Pré-condições	Ser administrador. Ver lista de idiomas
Sequência típica	Apaga uma língua
Sequência alternativa	-
Pós-condições	-

Table A.15: Descrição de Apagar língua

Caso de uso	Editar língua
Breve descrição	Edita qualquer nome e código de idioma
Pré-condições	Ser administrador. Ver lista de idiomas
Sequência típica	Editar o nome do idioma e/ou código
Sequência alternativa	-
Pós-condições	Nome e/ou código alterado

Table A.16: Descrição de Editar língua

Caso de uso	Adicionar um jogo
Breve descrição	É a primeira parte do processo de criar um jogo, dar-lhe o nome, descrição, imagem
Pré-condições	Ser editor de jogos
Sequência típica	Preencher os campos obrigatórios para a criação de um jogo
Sequência alternativa	Erro caso algum campo obrigatório não seja preenchido
Pós-condições	Jogo criado. Pode avançar para a criação de níveis

Table A.17: Descrição de Adicionar um jogo

Caso de uso	Adicionar um nível
Breve descrição	Adicionar níveis ao jogo
Pré-condições	Ser editor de jogos, ter jogo criado
Sequência típica	Preencher os campos obrigatórios para a criação de um nível
Sequência alternativa	Erro caso algum campo obrigatório não seja preenchido
Pós-condições	Pode criar questões. Embora a criação de questões possa ser uma tarefa independente

Table A.18: Descrição de adicionar um nível

Caso de uso	Adicionar questões
Breve descrição	Adicionar questões a cada nível. Este pode ser um processo independente, onde são criadas várias questões e só mais tarde adicionadas ao jogo
Pré-condições	Ser editor de jogos
Sequência típica	Preencher os campos obrigatórios para a criação de uma questão
Sequência alternativa	Erro caso algum campo obrigatório não seja preenchido
Pós-condições	-

Table A.19: Descrição de Adicionar questões

Caso de uso	Editar jogo
Breve descrição	Editar o jogo implica apenas mudar nome, descrição, etc
Pré-condições	Ser editor de jogos
Sequência típica	Alterar os campos que quiser
Sequência alternativa	Erro caso algum campo obrigatório não seja preenchido
Pós-condições	Campos relativos à informação do jogo alterados

Table A.20: Descrição de Editar jogo

Caso de uso	Editar nível
Breve descrição	Editar o nome e número do nível
Pré-condições	Ser editor de jogos
Sequência típica	Alterar os campos que quiser
Sequência alternativa	Erro caso algum campo obrigatório não seja preenchido
Pós-condições	Campos relativos à informação do nível alterados

Table A.21: Descrição de Editar nível

Caso de uso	Editar questão
Breve descrição	Permite editar o tipo de questão e seus parâmetros
Pré-condições	Ser editor de jogos
Sequência típica	Alterar o tipo de questão e parâmetros associados
Sequência alternativa	Erro caso algum campo obrigatório não seja preenchido
Pós-condições	Campos relativos à questão alterados

Table A.22: Descrição de Editar questões

Caso de uso	Apagar jogo
Breve descrição	Apaga o jogo e respectivos níveis, as perguntas mantém-se na lista de perguntas
Pré-condições	Ser editor de jogos
Sequência típica	Apaga um jogo de uma lista de jogos
Sequência alternativa	-
Pós-condições	Ao apagar um jogo, os níveis também são apagados. Mantém apenas as questões na lista de questões

Table A.23: Descrição de Apagar jogo

Caso de uso	Apagar nível
Breve descrição	Apaga o nível
Pré-condições	Ser editor de jogos
Sequência típica	Apagar o nível à escolha dentro de um jogo
Sequência alternativa	-
Pós-condições	O nível é apagado as questões mantêm-se na lista de questões

Table A.24: Descrição de Apagar nível

B | Anexo B

B.1 Descrição detalhada dos atributos

Jogo (Game)

- gameId : número de identificação do jogo
- name : nome do jogo
- description : descrição do jogo
- languageId : número de identificação do idioma do jogo
- icon : ícone do jogo
- logo : logótipo do jogo
- stateId : número de identificação do estado do jogo

Nível (Level)

- levelId : número de identificação do nível
- gameId : número de identificação do jogo
- levelNumber : número do nível
- questionId : número de identificação da questão do nível
- questionTypeId : número de identificação do tipo de questão do jogo

Questões Tipo (QuestionType)

- questionTypeId : número de identificação do tipo de questão
- name : nome do tipo da questão

- description : descrição do tipo de questão
- tableName : nome da tabela correspondente

Estado do Jogo (GameState)

- stateId : número de identificação do estado do jogo
- stateName : nome do estado do jogo
- visible : booleano que indica se o jogo vai estar visível ou não

Idioma do Jogo (GameLanguage)

- languageId : número de identificação do idioma
- languageCode : código do idioma (segundo as normas internacionais)
- languageName : idioma

Utilizador do Jogo (UserGame)

- userId : número de identificação do utilizador
- roleId : número de identificação do papel do utilizador
- gameId : número de identificação do jogo
- dateIn : data de início de sessão
- dateOut : data de fim de sessão
- points : pontos acumulados na sessão

Papel (AspNetRole)

- Id : número de identificação do papel
- Name : nome do papel

Utilizador Genérico (AspNetUsers)

- Id : número de identificação do utilizador
- Email : email do utilizador

- PhoneNumber : número de telefone do utilizador
- UserName : nome que o utilizador escolhe para início de sessão
- ProfilePicture : foto de perfil do utilizador

Ordering Questions (OrderingQuestion)

- orderingQuestionId: número de identificação das questões de ordenação
- title : título da questão
- text : texto/pergunta
- points : pontos dados pela questão
- pictureUrl : URL da imagem da questão
- pictureDescription : descrição da imagem
- backgroundImageUrl : URL de imagem de fundo para a questão (útil para usar no frontend)
- backgroundImageDescription : descrição da imagem de fundo
- helpTitle : título da ajuda
- helpMessage : mensagem de ajuda
- helpUrl : link para ajuda
- failTitle : título da mensagem resposta errada
- failMessage : mensagem da resposta errada
- successTitle : título da mensagem de sucesso
- successMessage : mensagem sucesso

Opções (OrderingOption)

- optionId : número de identificação da opção
- orderingQuestionId : número de identificação da questão
- text : texto da opção
- value : valor que indica a ordem

Text Questions (TextQuestion)

- textQuestionId : número de identificação das questões de texto
- title : título da questão
- text : texto/pergunta
- points : pontos dados pela questão
- suggestion : sugestão de resposta
- answer : resposta correta
- pictureUrl : URL da imagem da questão
- pictureDescription : descrição da imagem
- backgroundImageUrl : URL de imagem de fundo para a questão (útil para usar no frontend)
- backgroundImageDescription : descrição da imagem de fundo
- helpTitle : título da ajuda
- helpMessage : mensagem de ajuda
- helpUrl : link para ajuda
- failTitle : título da mensagem resposta errada
- failMessage : mensagem da resposta errada
- successTitle : título da mensagem de sucesso
- successMessage : mensagem sucesso

Matching Questions (MatchingQuestion)

- matchingQuestionId : número de identificação das "matching questions"
- title : título da questão
- text : texto/pergunta
- points : pontos dados pela questão
- pictureUrl : URL da imagem da questão
- pictureDescription : descrição da imagem
- backgroundImageUrl : URL de imagem de fundo para a questão (útil para usar no frontend)
- backgroundImageDescription : descrição da imagem de fundo

- helpTitle : título da ajuda
- helpMessage : mensagem de ajuda
- helpUrl : link para ajuda
- failTitle : título da mensagem resposta errada
- failMessage : mensagem da resposta errada
- successTitle : título da mensagem de sucesso
- successMessage : mensagem sucesso

Opções (MatchingOption)

- optionId : número de identificação da opção
- matchingQuestionId : número de identificação da questão
- text : texto da opção
- result : resultado correspondente

Bullet Questions (BulletQuestion)

- bulletQuestionId : número de identificação das "bullet questions"
- title : título da questão
- text : texto/pergunta
- points : pontos dados pela questão
- pictureUrl : URL da imagem da questão
- pictureDescription : descrição da imagem
- backgroundImageUrl : URL de imagem de fundo para a questão (útil para usar no frontend)
- backgroundImageDescription : descrição da imagem de fundo
- helpTitle : título da ajuda
- helpMessage : mensagem de ajuda
- helpUrl : link para ajuda
- failTitle : título da mensagem resposta errada
- failMessage : mensagem da resposta errada

- successTitle : título da mensagem de sucesso
- successMessage : mensagem sucesso

Opções (BulletOption)

- optionId : número de identificação da opção
- bulletQuestionId : número de identificação da questão
- text : texto da opção
- result : resultado certo

Tipo (BulletType)

- bulletQuestionTypeId : número de identificação do tipo de "Bullet Question"
- description : descrição do tipo

No Question (NoQuestion)

- noQuestionId : número de identificação da "No Question"
- title : título da questão
- text : texto/pergunta
- pictureUrl : URL da imagem da questão
- pictureDescription : descrição da imagem
- backgroundImageUrl : URL de imagem de fundo para a questão (útil para usar no frontend)
- backgroundImageDescription : descrição da imagem de fundo
- helpTitle : título da ajuda
- helpMessage : mensagem de ajuda
- helpUrl : link para ajuda
- failTitle : título da mensagem resposta errada
- failMessage : mensagem da resposta errada
- successTitle : título da mensagem de sucesso
- successMessage : mensagem sucesso

C | Anexo C

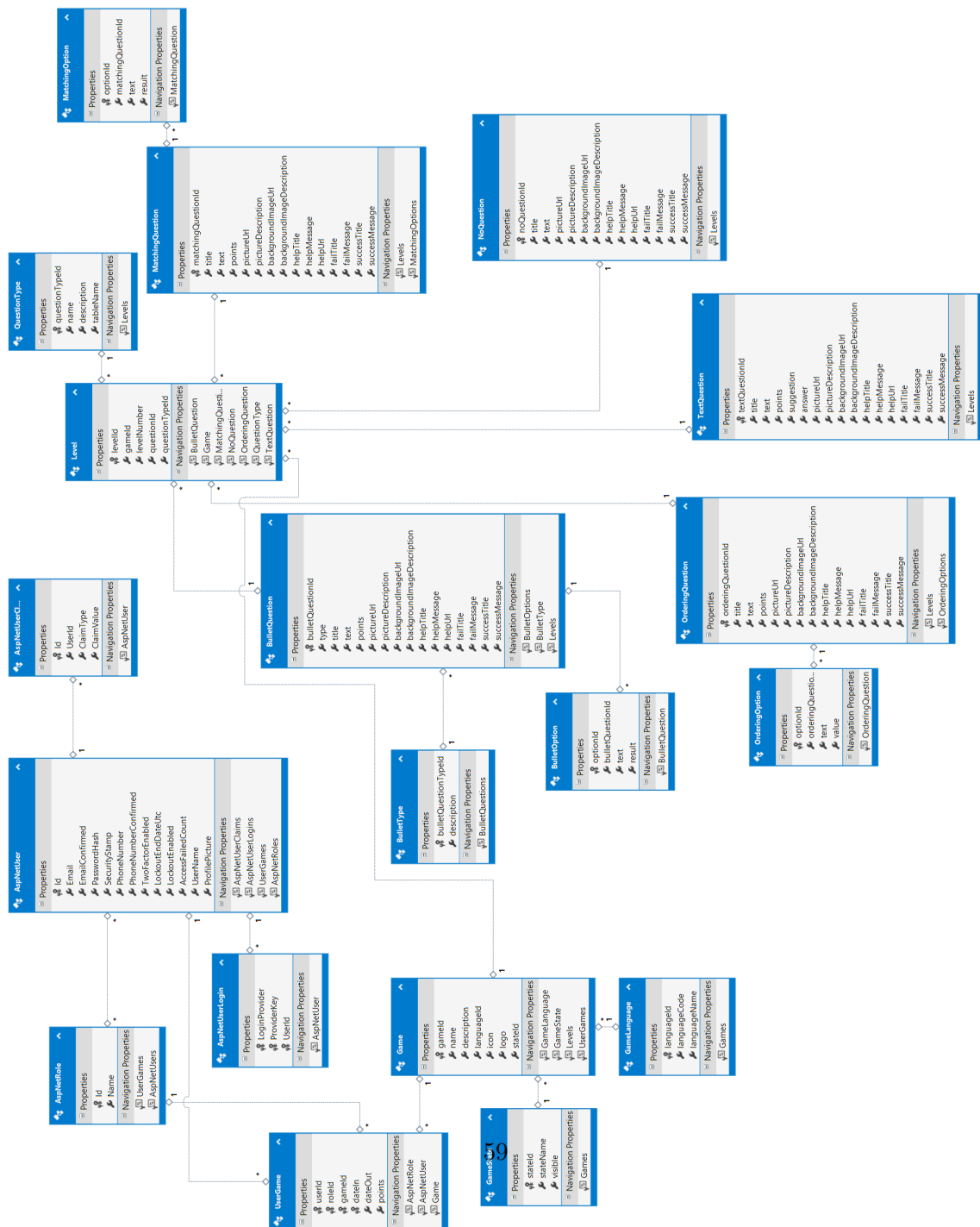


Figure C.1: Base de dados da plataforma