

UNIVERSIDADE DO MINHO

LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Projeto - ComputationalMind
Desafios para Desenvolvimento Computacional

Inês Pires Presa (A90355) Ivo Miguel Gomes Lima (A90214)
Tiago André Oliveira Leite (A91694)
Tiago dos Santos Silva Peixoto Carriço (A91695)

16 de junho de 2022



Conteúdo

1	Introdução	1
2	Enunciado do Projeto	2
3	Concessão da Solução	3
4	Mockups	4
5	Base de Dados	7
5.1	Identificação e caracterização das entidades	7
5.1.1	Author	7
5.1.2	Player	7
5.1.3	Question	7
5.1.4	Content	7
5.1.5	Option	8
5.1.6	History	8
5.2	Identificação e caracterização dos relacionamentos	8
5.3	Modelo Lógico	8
6	Back-end	9
6.1	Ferramentas	9
6.2	Django	9
6.3	Objetivo	9
6.4	Desenvolvimento	10
7	Front-End	11
7.1	Ferramentas	11
7.2	React e TailWindCSS	11
7.3	Objetivo	11
7.4	Desenvolvimento	11
8	ComputationalMind	12
8.1	Página inicial (Sem sessão iniciada)	12
8.2	<i>Login / Sign Up</i>	13

8.3	Escolha de jogos para <i>Player</i> ou <i>Author</i>	13
8.4	Menu do <i>Author</i>	13
8.5	Página inicial (Com sessão iniciada)	13
9	Conclusão	15
A	Exemplos	16

Capítulo 1

Introdução

No âmbito da unidade curricular de Projeto da Licenciatura em Ciências da Computação, foi proposto o desenvolvimento de uma plataforma que permita inserir problemas/desafios para que os utilizadores possam, em modo jogo, responder a esses desafios. A plataforma deve ter uma interface de *back-office* (BO) para armazenar os problemas/desafios (nomeadamente o seu enunciado, opções de resposta, resposta certa, faixa etária, etc). A interface de *front-office* (FO) funciona em modo jogo para que os utilizadores possam responder aos desafios selecionados a partir do repositório criado no BO da plataforma. No modo jogo os utilizadores podem ter pontuações, prémios, ou outras formas de gamificação que os motivem e envolvam.

O presente relatório acompanha o processo de desenvolvimento do projeto.

Capítulo 2

Enunciado do Projeto

No âmbito do treino do Pensamento Computacional pretende-se criar uma plataforma que permita inserir problemas/desafios para que os utilizadores possam, em modo jogo, responder a esses desafios. A plataforma deve ter uma interface de *back-office* (BO) para armazenar os problemas/desafios (nomeadamente seu enunciado, opções de resposta, resposta certa, faixa etária, etc). Um exemplo do tipo de exercícios que se pretende pode ser consultado na *página da web* do bebras¹. A interface de *front-office* (FO) funciona em modo jogo para que os utilizadores possam responder aos desafios seleccionados a partir do repositório criado no BO da plataforma. No modo jogo os utilizadores podem ter pontuações, prémios, ou outras formas de gamificação que os motivem e envolvam.

¹http://bebras.dcc.fc.up.pt/problems/2021/problemas_09_10.pdf

Capítulo 3

Concessão da Solução

Para a elaboração de todo o projeto foi necessário recorrer a diversos recursos que ajudassem na realização do mesmo.

Antes de mais, o principal recurso utilizado foi a própria equipa de docentes que foi responsável por esclarecer e ajudar na tomada de decisões ao longo de todas as fases do projeto.

Inicialmente, com a escolha do tema foi necessário recolher uma série de artigos e questões com o intuito de nos familiarizar com os diferentes tipos de mídia que a plataforma teria de suportar e possibilitar a sua mesclação.

Após essa pesquisa, surgiu a necessidade de ser feita modelação da Base de Dados, junto com a esquemização da página através da criação de Mockups ¹. Na nossa visão estes passos representam um pouco os pilares e ideias sobre os quais o nosso projeto acenta, daí terem tomado uma parte significativa do nosso tempo.

De seguida, foram seleccionadas as *frameworks* para criação tanto da *parte de suporte* como da *interface frontal*, sendo elas o *Django* ² e o *Reactjs* ³, assim como o *TailWindCSS* ⁴. Nesta fase damos uma posição de destaque à fase de teste e adaptação com as ferramentas, que levaram a uma etapa de muitas correções dos eventuais erros e de ajustes por forma a que nos certificarmos que o produto final correspondia às nossas expetativas, e às do público-alvo.

¹<https://www.figma.com>

²<https://www.djangoproject.com>

³<https://reactjs.org>

⁴<https://tailwindcss.com>

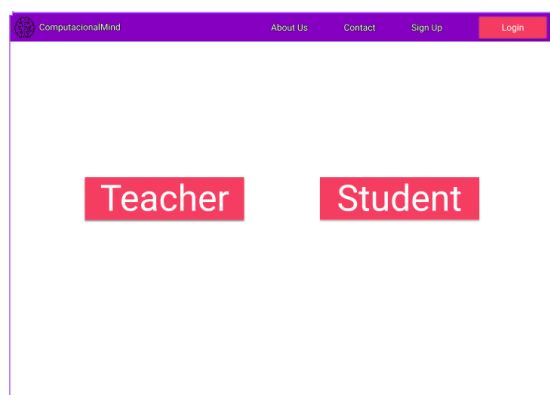
Capítulo 4

Mockups

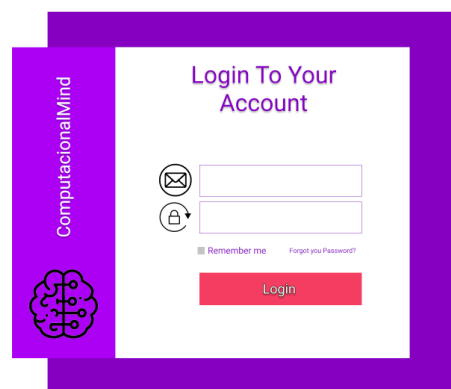
Durante o processo de criação do site foi-nos proposto a criação de uma maquete (*mockup*), que serviria como uma espécie de rascunho visual da forma como a *página da web* poderia ser apresentada, permitindo-nos testar como os vários elementos visuais funcionam em conjunto.

Como o *mockup* é um design estático, este não tem as funcionalidades oferecidas por um site vivo, por exemplo, não abrirá um *pop-up* quando se clica no botão *Login*, entre outros elementos. Fizemo-lo com o intuito de apresentar muitos dos elementos finais, da maneira como queríamos o *design*, mas como em todas as maquetes algumas ideias foram modificadas e eliminadas, devido a sugestões propostas pelos docentes da UC e para facilitar o processo de implementação.

Em última análise, percebemos que o *mockup* de uma página deve ser criado para um propósito específico em mente. Que ofereceu à equipa a chance de ver como o objetivo podia ser alcançado, para que pudesse ser trazido à vida por meio da utilização de padrões de marca e criatividade visual. Sendo que aparece no ponto médio do processo de *web design*.



(a) Página Principal



(b) Página de Login

ComputacionalMind

Sign Up

[I read and agree to Terms & Conditions](#)

Create

[Already have an Account? Login](#)

(a) Página de Criação de Conta

ComputacionalMind [About Us](#) [Contact](#) [Sign Up](#) [Login](#)

ComputacionalMind

Contact Us

Send

(b) Página de Contacto

ComputacionalMind [About Us](#) [Contact](#) [Sign Up](#) [Login](#)

Title

Content

+ Add Answer

Multiple Choice Short Answer

(c) Menu de Inserção de Jogo

ComputacionalMind

Title

Content

Option Option

Option Option

>

(e) Jogo de Escolha Múltipla

ComputacionalMind [About Us](#) [Contact](#) [Sign Up](#) [Login](#)

Id	Name	Creator	Difficulty
12	Memory Game	Inês Presa	Hard
98931	Ducks Game	Ivo Lima	Medium
42	Piramide Game	Tiago Leite	Easy
787762	Spot The Difference	Tiago Carriço	God

(d) Menu de Seleção de Jogo

ComputacionalMind

Title

Content

Short Answer

>

(f) Jogo de Resposta Curta

Congratulations



(a) Resposta certa

Never give up



(b) Resposta errada

Capítulo 5

Base de Dados

Para este projeto decidimos implementar uma Base de Dados Relacional que permita analisar e relacionar os dados do *Computational Mind* com vista a melhor o serviço, uma vez que esta deverá possibilitar a manipulação e consulta dos dados de uma forma ágil e segura. Este será um processo trabalhoso, mas que trará grandes frutos no futuro, uma vez que será substancialmente mais eficiente, especialmente na atualização, consulta e tratamento dos dados, o que permitirá alcançar todos os objetivos propostos pela equipa docente. Para além disso, será também um sistema mais fiável, visto que garantirá uma uniformização dos dados, garantindo que qualquer interveniente que pretenda consultar ou alterar os dados o fará de uma forma mais segura e controlada.

5.1 Identificação e caracterização das entidades

5.1.1 Author

Um autor é identificado por um **código**, devendo ter também uma referência do seu **nome**, *e-mail* e *password*.

5.1.2 Player

Um jogador é identificado por um **código**, devendo ter também uma referência do seu **nome**, **data de aniversário**, *e-mail* e *password*

5.1.3 Question

Uma questão é identificado por um **código**, devendo ter também uma referência ao **autor**, um **título**, um identificador do **tipo** (resposta curta, escolha múltipla ou verdadeiro e falso), uma **classificação**, a **dificuldade** e a **idade mínima**.

5.1.4 Content

Um conteúdo é identificado por uma referência à **questão**, uma **ordem** pela qual as questões aparecem, o **tipo** do conteúdo e os **links/imagens/vídeos** associados.

5.1.5 Option

Uma opção é identificado por um **código**, uma referência à **questão**, a **resposta** para serem feitas as verificações e a **opção correta**.

5.1.6 History

Um histórico é identificado por **código**, uma referência ao **jogador**, uma referência à **questão**, a **data** que o jogador respondeu à questão, as suas **respostas** e quais **acertou**.

5.2 Identificação e caracterização dos relacionamentos

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Question	1..N	tem	1	Author
Option	1..N	tem	1	Question
History	1..N	tem	1	Question
History	1..N	tem	1	Player
Content	1..N	tem	1	Question

5.3 Modelo Lógico

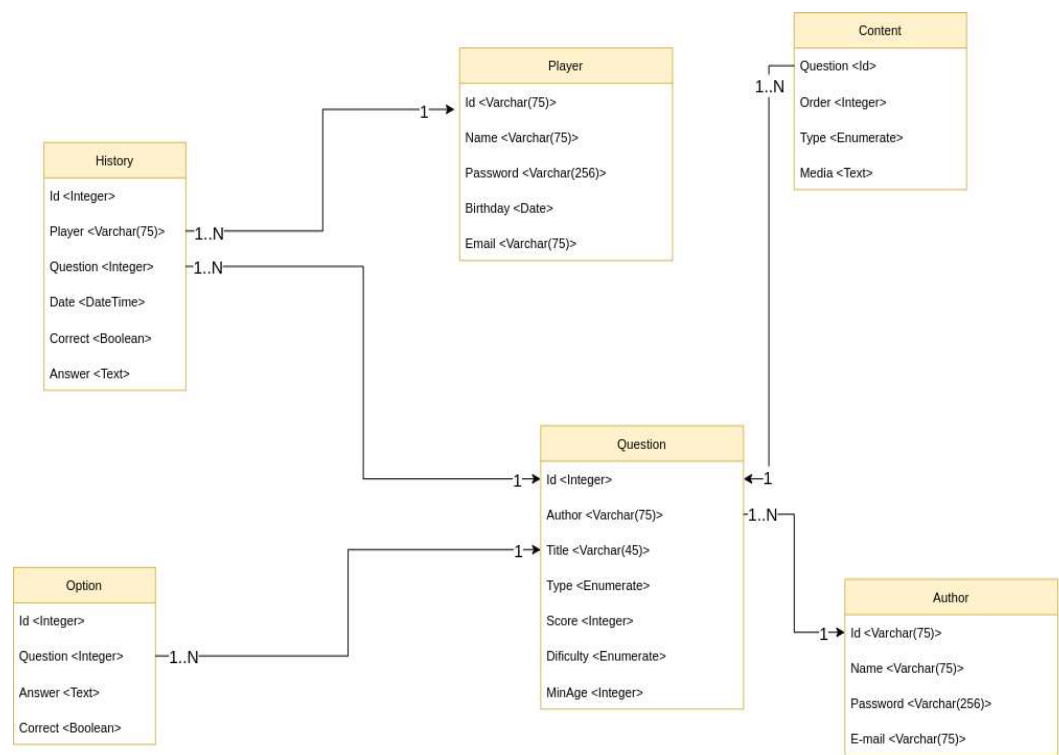


Figura 5.1: Esquema Lógico

Capítulo 6

Back-end

6.1 Ferramentas

Para o desenvolvimento da *parte secundária* utilizou-se *Django* e *Django REST Framework*.

6.2 Django

Django é uma *framework* gratuito e de código aberto, com o intuito de facilitar e acelerar o desenvolvimento *web*, escrito em Python, que utiliza o padrão *model-template-view* (MTV) e o princípio *don't repeat yourself* (DRY) através do aproveitamento máximo de código já feito. O *web server*, pode ajudar os desenvolvedores a produzir de forma eficiente um FO ou BO rico em recursos, seguro e escalável.

O *Django REST Framework* é um conjunto de ferramentas utilizado para construir *APIs* (Interfaces de Programação de Aplicações) para *Web*. Apesar do *Django* permitir fazer tanto *front-end* como *back-end*, a utilização de uma *REST API* permite desenvolver uma parte de suporte mais genérico, e torna possível desenvolver uma parte frontal para vários tipos de dispositivos no futuro. Ou seja, assim a aplicação torna-se mais escalável.

6.3 Objetivo

O *back-end* do projeto tem como objetivo fazer a ligação entre a Base de Dados e a *interface frontal*, tendo sido desenvolvido ainda uma *REST API* que será "consumida" pela mesma.

6.4 Desenvolvimento

No processo de desenvolvimento utilizou-se a base de dados previamente implementada no *MySQL Workbench* para se fazer a construção dessas mesmas entidades e relacionamentos no Django com o intuito de possibilitar a comunicação entre ambos foi necessário alterações o esquema mais concretamente a eliminação das chaves primárias compostas.

A comunicação entre o Django e a parte frontal através da REST API é feita com o uso de ficheiros em formato JSON. Para tal, o *Django* possui funções específicas para serializar e desserializar os dados. O uso dá-se da seguinte forma, o front-end faz uma pedido através da *REST API*, de seguida este pedido é analisado pelo Django e caso esteja correto os dados requeridos são copiados da base de dados, serializados e enviados em formato JSON para a interface frontal. A comunicação inversa também é possível mas os dados são desserializados e estando corretos, serão inseridos na base de dados.

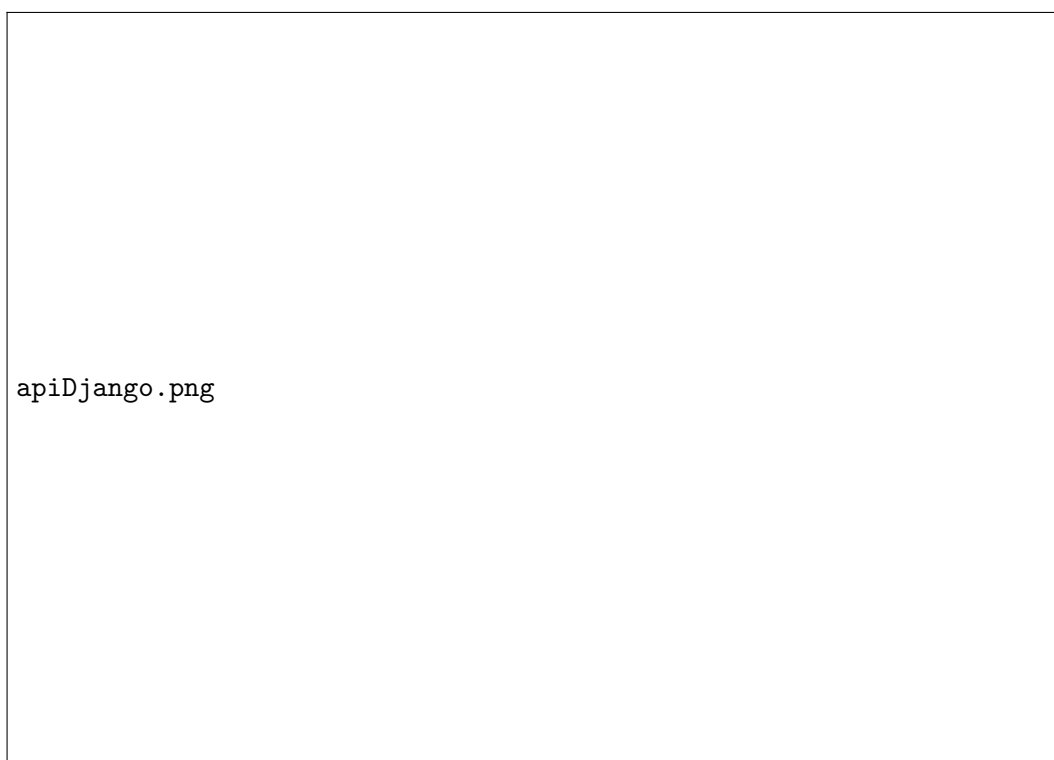


Figura 6.1: REST API

Capítulo 7

Front-End

7.1 Ferramentas

Para o desenvolvimento do *front-end* utilizou-se *React* e *TailWindCSS*.

7.2 React e TailWindCSS

React.js é uma biblioteca *JavaScript* de código aberto usada para construir o front-end para aplicativos e páginas web de página única. Sendo usado para lidar com a camada de visualização para os aplicativos da Web e móveis. O *React* também nos permite criar componentes de interface do usuário reutilizáveis para que os desenvolvedores criem grandes aplicativos da *Web* que possam alterar dados sem que seja necessário recarregar a página. O seu principal objetivo é ser rápido, escalável e simples. Este funciona apenas em interfaces de usuário no aplicativo, o que corresponde à exibição do modelo *Model-View-Controller* (MVC), podendo ser usado em conjunto com outras bibliotecas ou *frameworks JavaScript*, como *Angular JS* em MVC.

Tailwind CSS é basicamente um *framework CSS* utilitário para construir de forma rápida interfaces de utilizador personalizadas, sendo uma estrutura *CSS* altamente personalizável e de baixo nível que fornece todos os blocos de construção necessários para criar designs sob medida, sem que seja necessária a escrita de CSS como na abordagem tradicional.

7.3 Objetivo

O front-end é responsável por possibilitar a interação do utilizador com a página dentro de uma aplicação web e cobrindo as questões da política de privacidade.

7.4 Desenvolvimento

Cenas

Capítulo 8

ComputationalMind

Ao longo deste semestre desenvolvemos a página do *ComputationalMind* até que o produto final fosse semelhante ao que foi idealizado e apresentado nos capítulos anteriores. Em seguida, será demonstrado o funcionamento e onde se podem encontrar todas as funcionalidades implementadas.

8.1 Página inicial (Sem sessão iniciada)

Na página inicial consideramos que o utilizador pode ser um *Player* que apenas usufrui dos jogos disponibilizados na plataforma pelo *Author* que é a entidade que criou o jogo.

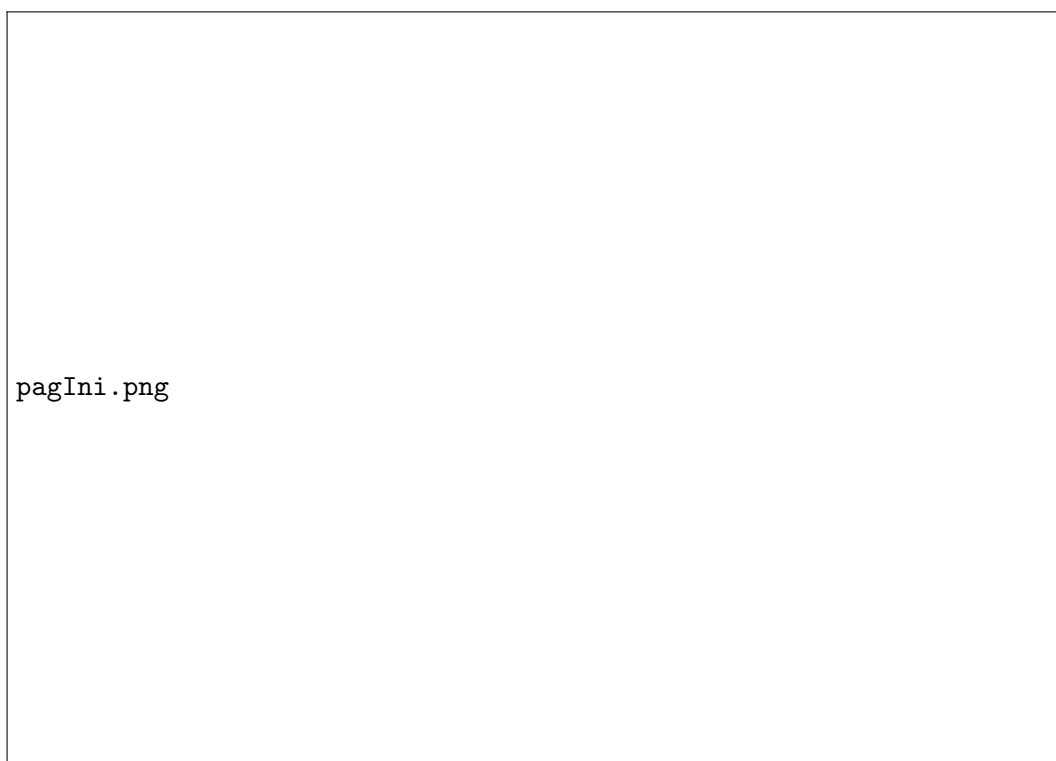


Figura 8.1: Página Principal

8.2 *Login / Sign Up*

Após a escolha de uma das opções disponibilizadas na página inicial os utilizadores serem redirecionados para a página de *Login / Sign Up*.

Nesta página é possível um utilizador autenticar-se ou efetuar o seu registo no site.

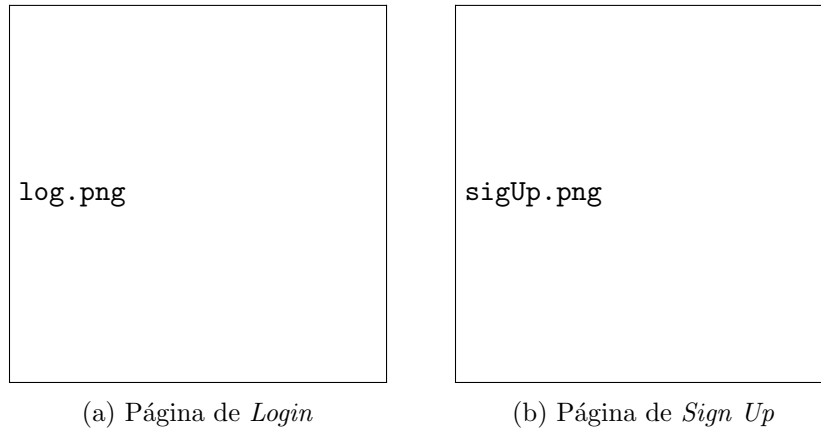


Figura 8.2: Página de *Login* e Registo

8.3 Escolha de jogos para *Player* ou *Author*

Em seguida, ambos os utilizadores são redirecionados para uma página onde se lista todos os jogos que estão disponíveis para jogar.

8.4 Menu do *Author*

O *Author* é um utilizador especial, diferindo de um utilizador normal pelo facto de existir uma secção chamada *Creat New Game* onde poderá criar os diferentes tipos de jogos referido nos capítulos anteriores.

8.5 Página inicial (Com sessão iniciada)

Quando um utilizador se regista ou autenticação, é redirecionado para a página principal da aplicação. Aqui é possível ver todas as opções de países e categorias disponíveis, bem como seleccioná-los de forma a filtrar as notícias. Além disto, consegue-se ver as trending news, e pode-se aceder a outras páginas da aplicação.

Demo da pagina, login, user, autor, apresentação das funcionalidades,...

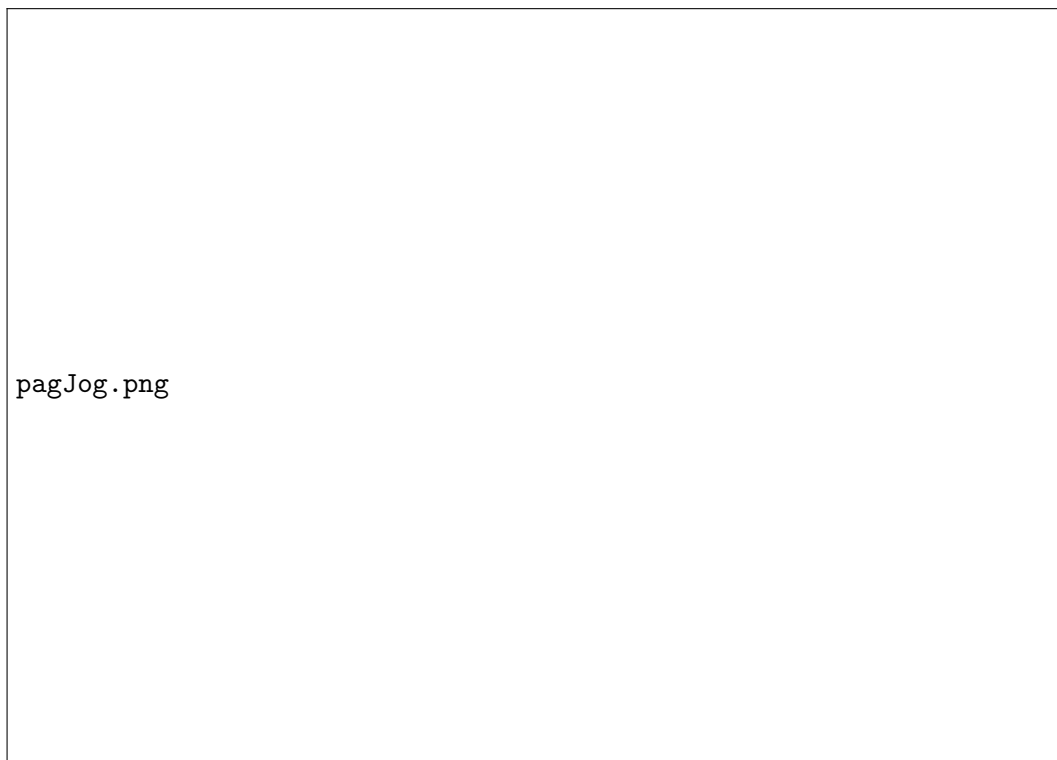


Figura 8.3: Página de Jogos

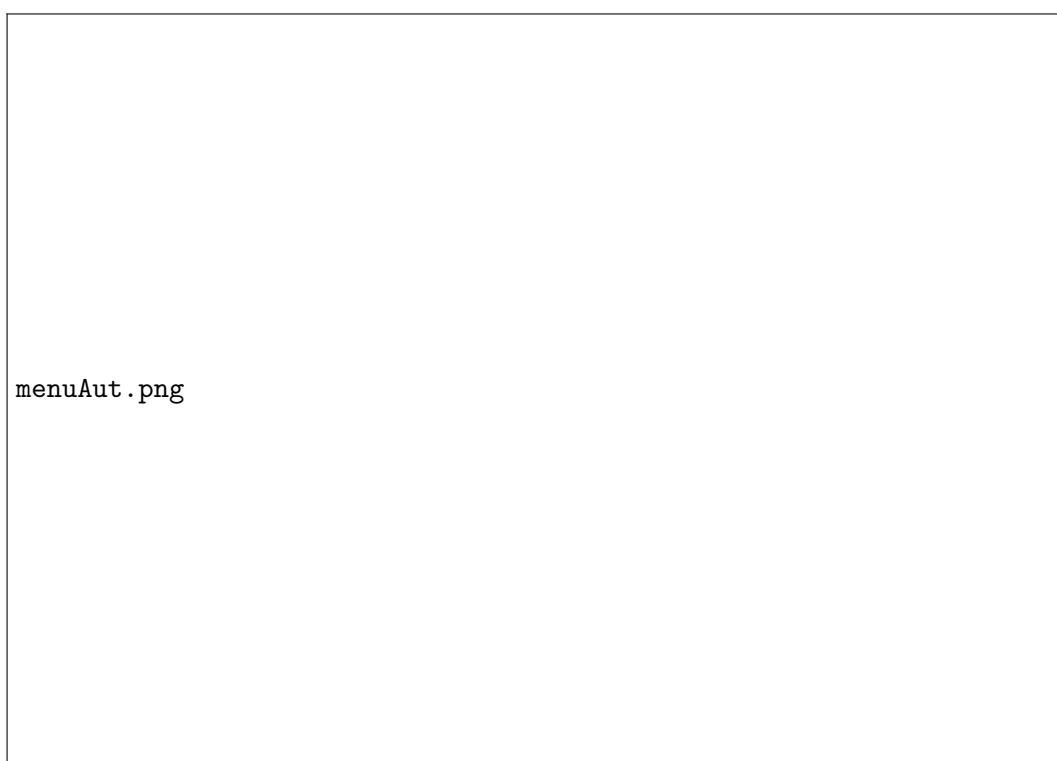


Figura 8.4: Menu do *Author*

Capítulo 9

Conclusão

No entanto, é necessário realçar que o projeto não se encontra totalmente finalizado, na medida em que algumas das funcionalidades planeadas acabaram por não ser implementadas na versão final. Desta forma, o projeto final é apenas uma “amostra” daquele que foi inicialmente idealizado estando presentes apenas as funcionalidades consideradas mais importantes.

Posto isto, as dificuldades sentidas consistiram essencialmente no facto de se estar a usar uma nova linguagem de programação e ferramentas de desenvolvimento de software completamente novas, e com as quais não estávamos familiarizados.

Desta forma, e apesar dos diversos obstáculos encontrados durante todo o processo, o grupo considera que concluiu o projeto com sucesso.

Como sugestão para trabalhos futuros, seria útil implementar todas as funcionalidades inicialmente planeadas e que não estão no produto final, além de ser também interessante desenvolver funcionalidades extra que tornassem a página *web* mais apelativa para o utilizador.

Apêndice A

Exemplos