IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

Programação Web 2021

Construção de Biblioteca de videojogos (Website e Fórum)

Projeto Final
Época Normal e de Recurso

Índice

1 Introdução

O principal objetivo deste projeto é desenvolver, utilizando HTML, CSS e JavaScript, uma solução Web que permita a criação de **um website** com informação sobre videojogos e **um fórum** sobre o tema. **O website** deverá compreender todas as técnicas e seguir as boas práticas do desenvolvimento de código faladas nas aulas e **o fórum** deverá permitir aos utilizadores ter diversos papeis como gestores, criar tópicos, escrever respostas e comentários, participar em discussões ou avaliar determinados videojogos colocados previamente por outros utilizadores.

Existirá também a possibilidade de ser desenvolvido um extra para o projecto que consiste na criação de um pequeno jogo. Este mini-jogo deverá recorrer ao uso das tecnologias já mencionadas (HTML, CSS e *JavaScript*) para cumprir o seu objectivo.

A decisão pela escolha desta temática justifica-se pelo facto da indústria dos videojogos ser uma das indústrias mundiais com maior relevância em termos financeiros e ultrapassa já as indústrias do cinema e desporto dos EUA combinadas. (https://tek.sapo.pt/noticias/negocios/artigos/receitas-da-industria-de-videojogos-saltam-em-2020-e-ultrapassam-as-do-cinema-e-desporto). É também muito provável que em tempos de pandemia, e com o respectivo isolamento, tenha havido um aumento do "consumo" deste tipo de entretenimento impulsionando ainda mais as vendas.

2 Tecnologia e metodologia a utilizar

A aplicação Web deve ser desenvolvida segundo uma **metodologia *Single-Page Application* (SPA)** para a vertente de Website, ou seja, ter uma página em HTML5 com conteúdo estático e informativo presente apenas num ficheiro HTML. A aplicação do fórum, deverá utilizar um sistema de navegação entre páginas mais comum (navegação através de *hyperlinks*) em que o conteúdo vai sendo gerado e atualizado dinamicamente mediante a interação do utilizador com o sistema.

Num primeiro momento, o principal foco do projeto é o desenvolvimento de uma interface HTML/CSS que possibilite uma interação limpa e simples com as principais funcionalidades.

Seguidamente, pretende-se que o aluno explore as capacidades da linguagem *Javascript* relativas à manipulação de elementos do DOM e que adicione as funcionalidades de CRUD (*Create, Read, Update, Delete*) para lidar com os dados na vertente *client-side*, guardando esses dados em tempo real e recorrendo a listas de objetos em tecnologia *JavaScript*.

Ficando para uma última fase de implementação a ligação deste sistema a uma base de dados, com manipulação dos dados através de "roteamento" dos diversos pedidos efetuados pelos clientes (*browsers*), mediante a adoção de uma tecnologia *server-side*. Pretende-se que os alunos construam uma camada de acesso a dados recorrendo à tecnologia de serviços Web (Web Services), neste caso implementando a popular norma RESTful em detrimento de protocolos mais antigos RPC (*Remote Procedure Call*) ou SOAP (*Simple Object Access Protocol*). O servidor Web e respetivos serviços deverão

ser implementados em linguagem Node.js, sendo que o repositório de dados será criado recorrendo ao SGBD MySQL[1].

Devido ao constrangimento que advém do facto da matéria de suporte ao servidor ser lecionada numa fase já mais adiantada do semestre, o que também é natural que assim seja, cabe aos alunos criarem o seu conjunto de teste (objetos em *JavaScript* com dados por omissão) de modo a poderem iniciar o sistema já com uma base iniciada para interagir com o utilizador.

3 A Biblioteca de videojogos (Fórum)

Apresentam-se aqui, de forma resumida, as especificações e requisitos básicos que deverão ser consideradas pelos alunos de forma a implementarem a Biblioteca de videojogos.

Para uma gestão realista de um determinado tema, terá de existir um foco principal na adequação dos conceitos que o sistema pretende suportar. Como tal, as **possíveis entidades** a considerar aqui como exemplo serão as seguintes, **podendo existir outras,** dependendo das funcionalidades que se queira implementar:

- **Categorias**
- **Posts**
- **Comentários**
- **Utilizadores**

O sistema deverá permitir a criação, visualização, edição e eliminação de *Categorias* de videojogos. Para cada uma dessas categorias deverá ser possível criar *Posts* (com um título e uma descrição), **que estarão sempre associados a um videojogo específico**, e os respectivos *Comentários*, que vão estar sempre associados a esse *Post* (videojogo). Cada comentário deve ter associado diversas reacções (que podem ser representadas por emojis, se o programador assim o entender) e cada uma delas deve ser contabilizada de forma a mostrar um total (caso haja reacções repetidas). Os *Posts* devem ter a funcionalidade de *UpVoting*, permitindo que qualquer utilizador registado possa votar positiva ou negativamente sobre o tema. Deverá também existir uma opção específica para a indicar se um utilizador já jogou aquele videojogo. Ao visualizarem o *Post*, todos os utilizadores devem poder ter acesso a uma listagem de todos os que seleccionaram a opção de já ter jogado aquele videojogo. Para melhor definir esta relação, segue **um exemplo** de *Categorias* e *Posts:*

- *Categoria – MMOs (Massive Multiplayer Online Games)*
  - *Post (videojogo) – World of Warcraft*
  - *Post (videojogo) – RuneScape*
  - *Post (videojogo) - Star Wars: The Old Republic*

Na gestão dos *Utilizadores*, o sistema deverá permitir a criação, edição e remoção de utilizadores. A data de inscrição desse utilizador no fórum não deverá aparecer na listagem, aparecendo apenas o tempo ao qual já se encontra registado na plataforma (ex.: 15 minutos, 1 hora, 2 dias, 3 semanas, 5 meses, etc), que deverá ser calculada mediante a data de inscrição de cada utilizador. Devem existir

[1] Sugere-se o *software* MySQL Workbench como interface gráfica para se trabalhar com o repositório de dados.

diferentes tipos de utilizador com permissões distintas. É necessário criar no mínimo 3 tipos de utilizadores, **o gestor do fórum, o gestor de conteúdo e o visitante**. Cabe ao gestor de fórum as funções de gestão de utilizadores, moderação das categorias e controlo e configuração da plataforma. O gestor de conteúdos pode apenas criar e moderar os seus posts e o visitante poderá apenas comentar e reagir a esse post (podendo utilizar os emojis se essa funcionalidade estiver implementada).

Como referido anteriormente, o intuito desta parte do trabalho é desenvolver um website de navegação comum e para tal será necessário fazer a gestão de qual o conteúdo a mostrar mediante as opções de um menu principal que deve responder às necessidades de base da aplicação. O clique em cada opção de menu poderá proporcionará a criação de algum conteúdo através do DOM. Contudo, não é necessário que todo o conteúdo seja construído desta forma, podendo existir algumas partes onde se poderá optar pela escrita do código HTML na página e gerir as visibilidades desses conteúdos através de *JavaScript*. Um exemplo simples dessa gestão de visibilidades poderá ocorrer com os formulários para inserção dos dados das diversas entidades.

Em relação ao Layout, o sistema deve seguir as diretrizes indicadas nas aulas teóricas e laboratoriais que permitam ter uma página *Responsive*. Em termos de interfaces gráficas, **cada grupo deverá optar pela opção gráfica que achar mais adequada, desde que se mantenha fiel aos objetivos e requisitos de base propostos para este projeto, sendo esta alvo do processo de avaliação.**

4 O mini-jogo (Extra)

O objectivo deste mini-jogo é o de fazer o match entre algumas dicas visuais (imagens, ícones, etc) e o nome do respectivo jogo. Mediante o número de ligações correctas, será atribuída uma pontuação.

Para tornar este mini-jogo mais desafiante é pretendido que exista um contador de tempo decrescente em segundos (sendo esse valor definido pelo developer). Ao fim desse tempo, o utilizador deixará de poder ligar as dicas visuais ao título dos videojogos e deverá ser apresentada a pontuação conseguida até ao momento. Os valores a utilizar para a pontuação ficam a cargo do programador.

É pretendido recorrer à utilização da canvas API (https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API, https://www.w3schools.com/html/html5_canvas.asp) para que seja possível criar uma linha de ligação entre um elemento visual e o elemento de texto. Quando todas as ligações estiverem terminadas, deverá haver um botão que permitirá ser executado um código *JavaScript* de validação dos resultados e que vai determinar a pontuação final.

A componente visual desta aplicação ficará a cargo do aluno, mas será importante que não haja uma diferença estética muito acentuada em relação ao restante website.

5 Sugestões

As interfaces de gestão das diversas entidades podem seguir uma metodologia como a utilizada nos laboratórios dedicados a *JavaScript*. Conseguindo ter um código mais versátil, e tendo em consideração as propriedades existentes em cada tipo de objeto, é possível minimizar o trabalho para a gestão dessas entidades.

Outro aspeto a ter em atenção é que sempre que se pretende criar uma entidade, os campos do formulário devem estar limpos para que não perdurem dados inseridos numa criação/edição anterior (caso se opte por mostrar/esconder formulários pré-existentes no HTML).

Poder-se-á optar, ou não, por usar uma solução de *web framework* JavaScript para servidor como o framework "Express", explicado em aula dedicada ao tema.

O módulo MySQL para o Node.js terá de ser instalado com o intuito de criar conexões à base de dados MySQL previamente instalada. Aqui será necessário ter-se os dados de acesso ao servidor de base de dados.

Poder-se-á sentir a necessidade de usar extensões do Visual Studio Code para ajudar com JSON, SQL, etc. Os alunos poderão usar o que acharem conveniente para melhorar a sua produtividade e a qualidade do trabalho final.

6 Regras no desenvolvimento do projeto

6.1 Regras de implementação e codificação

O projeto deverá ser desenvolvido segundo as seguintes etapas e tecnologias:

- Etapa 1 - Criação de mockups e definição de requisitos do projecto
- Etapa 2 - HTML, CSS para definição da IU;
- Etapa 3 - JavaScript para interatividade e gerir entidades com geração de objetos por omissão;
- Etapa 4 - Construção de uma base de dados em SQL e de um WebService RESTful em Node.js, que irá interagir com a SPA através da tecnologia AJAX;
- Etapa Extra – Desenvolvimento do mini-jogo recorrendo a HTML, CSS e *Javascript*;

Os alunos deverão colocar em prática os conceitos fundamentais da programação para a Web, base de dados e do paradigma de programação orientada por objetos que aprenderam nas aulas teóricas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, **sendo deixado ao critério dos alunos qualquer aspeto de implementação que não seja referido no mesmo, devendo ser apresentado ao respetivo docente de laboratório e devidamente documentado.**

Obviamente que se pretende que a aplicação utilize um modelo suficientemente genérico que possibilite a sua utilização com as mais variadas categorias, posts e comentários. No entanto **bastará que o grupo mostre que a aplicação consegue lidar com pelo menos duas categorias, dois posts distintos, um comentário e a opção de UpVoting para cada um deles.**

6.2 Constituição de grupos

Cada projeto deverá ser elaborado por um grupo de dois alunos do mesmo docente de laboratório, podendo eventualmente ser elaborado individualmente (devendo ter aprovação prévia do respetivo docente de laboratório). Excepções a estas regras têm obrigatoriamente de ser avaliadas pelos docentes.

6.3 Submissões do projeto

O projeto deverá ser entregue até à data limite especificada por via exclusivamente eletrónica, utilizando a área dos trabalhos do respetivo docente de laboratório no Moodle. Existem dois momentos de entrega:

- **Fase 1:** até às **23:55 do dia 13 de Dezembro de 2021**. Um documento resumo que apresente:
  - as categorias, estrutura de posts e tipos de utilizadores;
  - os requisitos funcionais de base a implementar;
  - mockups correspondentes;
  - modelo relacional;
  - lista de **possíveis** frameworks/APIs a usar e com que objetivo (**não** é obrigatório o uso de qualquer framework);

- **Fase 2**: até às **23:55 do dia 24 de Janeiro de 2022**. Esta será a entrega final e deverá incluir o projeto completo, englobando todas as etapas (e se pretenderem a etapa extra). O que foi entregue na fase 1 poderá ter sido retificado para esta fase final. **Nota:** em **Época de Recurso** o projecto terá de ser entregue até às **23:55 do dia 28 de Fevereiro de 2021.**

Os materiais do projeto deverão incluir:

- Pasta do projeto pronta a ser aberta e executada em VSCode. Deverá ter na raiz dessa pasta:
  - Ficheiro `package.json` com a descrição do projeto, em particular as suas dependências em termos de módulos, permitindo o seu carregamento através de "`npm install`".
  - Ficheiro `app.js` com a implementação do *web server*;
  - Pasta `www` com os ficheiros de implementação do programa: ficheiros `*.html`, pasta `images` para conter as imagens/ícones, pasta `scripts` para conter os ficheiros JavaScript e pasta `styles` para conter os ficheiros CSS.
  - Ficheiro, na linguagem SQL, com todo o código necessário para criação de todas as tabelas necessárias à solução, modelo entidade-relação. **Deverá existir também a inserção de linhas com dados de exemplo que permitam o teste da aplicação**.
- **Os ficheiros JavaScript deverão ser documentados através de JSDoc**.
- Devem também incluir um ficheiro pdf com a documentação da API REST desenvolvida, indicando para cada cada endpoint:
  - Título e descrição curta do que o endpoint faz
  - URL e método
  - Parâmetros e dados recebidos pelo endpoint (se existirem)
  - Formato do resultado devolvido pelo endpoint caso tudo corra bem (status 200)
  - Erros que podem ser devolvidos (status HTTP e formato de resultado para cada erro)

  Neste site podem ver uma explicação de como fazer este tipo de documentação:
  https://bocoup.com/blog/documenting-your-api

Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado (formato ZIP ou RAR), cujo nome deverá seguir a seguinte estrutura: <*numAluno1*>_<*numAluno2*>.zip|rar. **Nota: o não cumprimento desta regra implica a dedução de 0.5 valores na nota final do projecto.**

7 Avaliação

7.1 Regras de Avaliação

- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do *software*), a estrutura do código criado segundo os princípios da programação orientada por objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento das linguagens envolvidas.
- Serão premiadas a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- Após a entrega haverá uma discussão oral para validação da nota final da componente Projeto.
- Os alunos que não comparecerem à discussão serão classificados com zero. Nesta discussão com os alunos poderá ser apurada a capacidade do aluno de produzir o código apresentado. A nota atribuída será zero nos casos em que essa capacidade não for demonstrada.
- Após a discussão, a nota do Projeto será atribuída individualmente a cada um dos elementos do grupo.
- A avaliação oral será realizada pelo respetivo docente de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos nessas condições, quer tenham sido o original ou a cópia.
- No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente a que se destina.

7.2 Critérios de Avaliação

1. Funcionalidades e operações essenciais	**40%**
2. Implementação técnica (utilização correta dos elementos HTML; definição e utilização de CSS; desenvolvimento do servidor, etc.**)**	**40%**
4. Documentação (bom estilo (identificadores, comentários, indentação); JSDoc, relatório; etc.**)**	**10%**
4. Extras e Bónus (utilização de bibliotecas, ferramentas ou tecnologia não lecionada e que faça sentido adicionando real valor ao projeto **desde que devidamente justificada pelo aluno**; novos requisitos/funcionalidades, realização da etapa extra; etc.**)**	**10%**

8 Anexos

8.1 Sistema de Gestão da Base de Dados (SGBD)

Apesar de não ser o principal foco da Unidade Curricular de Programação para a Internet, o sistema de gestão de base de dados irá ser uma componente da máxima importância no projeto e, como tal, terá o respetivo peso na nota final da segunda fase. E até porque em termos profissionais, a maior parte dos projetos de engenharia informática recorre naturalmente a um armazenamento de dados.

8.1.1 MySQL

Trata-se de um SGBD relacional que usa SQL (Structured Query Language), sendo o seu código fonte aberto aos utilizadores. Neste momento é dos sistemas mais usados do mercado e para isso contribui uma base de documentação relativamente completa. Outro dos motivos da escolha deste SGBD é o facto de ele ter sido utilizado nas anteriores unidades curriculares da licenciatura dedicadas ao estudo de "Bases de Dados". Poderá efetuar o download do servidor no seguinte URL: https://dev.mysql.com/downloads/installer/.

O procedimento de instalação é simples e intuitivo, contudo tenha atenção ao passo de escolha do "porto" de ligação ao SGBD, do nome do utilizador de acesso (por default costuma ser root) e a palavra-passe definida para esse mesmo utilizador. Esta informação será necessária para interagir com o SGBD.

Existe um aplicativo gráfico opcional na instalação (MySQL Workbench), que poderá ser usado pelos alunos de forma a tornar mais apelativa a criação das tabelas, campos, chaves primárias, chaves estrangeiras, índices, procedimentos e funções. Tendo em conta, os conhecimentos e experiência adquirida pelos alunos na utilização do MySQL Workbench na Unidade Curricular de Bases de Dados, a sua utilização pode revelar-se um meio simples e rápido para o desenvolvimento da base de dados de suporte necessária ao sistema da aplicação Web para gestão de videojogos.

8.2 Web Service

Um *Web Service*[2] é um qualquer serviço que está disponível pela internet, que usa uma linguagem de comunicação standard para troca de mensagens, e não está dependente de nenhum sistema operativo ou linguagem de programação.

8.2.1 Abordagem RESTful

Para o desenvolvimento deste projeto deverá recorrer à solução arquitetural RESTful. Apesar de se usar principalmente JSON para a troca de mensagens, é possível recorrer a outros *standards*, como por exemplo XML. Esta abordagem utiliza os métodos (verbos, códigos, recursos) HTTP para efetuar as tradicionais operações de CRUD (ver Tabela 1), não guardando estado entre pedidos.

[2] Para mais informação: https://en.wikipedia.org/wiki/Web_service

OPERAÇÕES	COMANDOS SQL	REST
CREATE	INSERT	POST
READ (ou RETRIEVE)	SELECT	GET
UPDATE	UPDATE	PUT e/ou PATCH
DELETE (ou DESTROY)	DELETE	DELETE

Tabela 1 - Relação entre "Operações CRUD", "Comandos SQL" e "Verbos HTTP".

8.2.2 Comunicação entre a SPA e o serviço RESTful

A arquitetura que se pretende implementar poder ser visualizada na Figura 1.

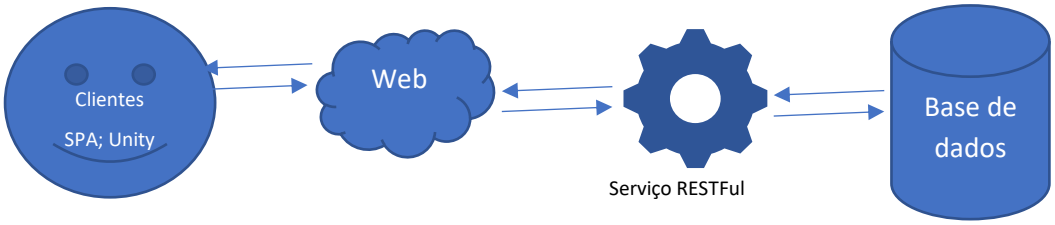

Figura 1 - Arquitetura.

O objetivo será que através do serviço RESTful se consiga a informação que irá iniciar os objetos das classes que já foram desenvolvidas anteriormente.

Caso tenha construído os objetos por omissão para testar a SPA numa fase anterior, terá de se remover a geração e criar-se as respetivas funções auxiliares recorrendo à tecnologia AJAX (*Asynchronous JavaScript and XML*), que será explicada em aulas futuras. Serão efetuados os pedidos necessários por parte do cliente ao serviço RESTful. Abaixo pode-se observar um exemplo de teste para verificar se é possível carregar o recurso uma determinada entidade.

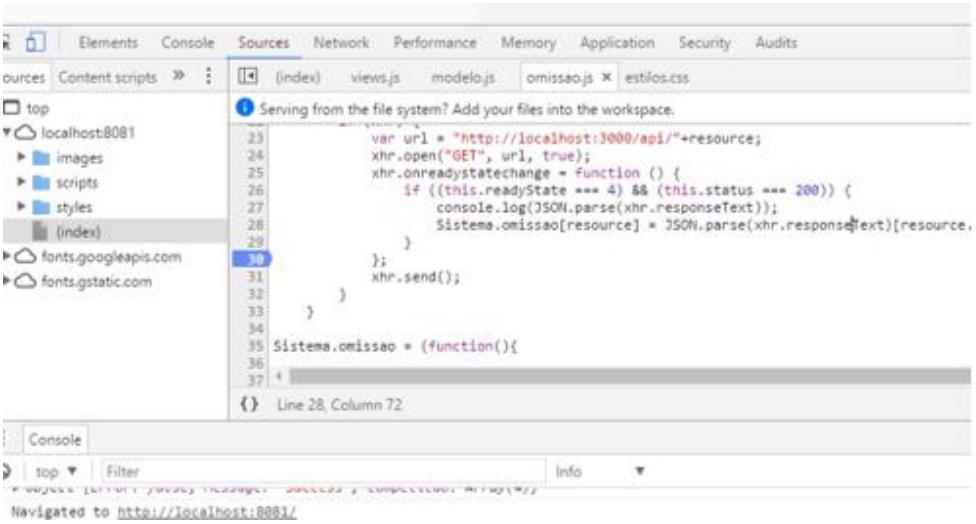

Figura 2 - Exemplo de consumo do serviço RESTful por parte do cliente (ambiente de debug do Chrome).