

Programação para a Internet 2020

Website dinâmico sobre Sustentabilidade com módulo de Gestão de Conteúdos



Projeto Final
Época Normal e de Recurso

Índice

1	Introdução.....	3
2	Fases e Tecnologia	3
3	O Gestor de conteúdos dinâmicos.....	4
4	Sugestões	4
5	Regras no desenvolvimento do projeto.....	5
5.1	Regras de implementação e codificação	5
5.2	Constituição de grupos	5
5.3	Submissões do projeto.....	6
6	Avaliação	6
6.1	Regras de Avaliação	6
6.2	Critérios de Avaliação	7
7	Anexo	8
7.1	Web Services.....	8
7.1.1	Abordagem RESTful.....	8
7.1.2	Comunicação entre a SPA e o serviço RESTful	8

1 Introdução

O principal objetivo deste projeto é desenvolver, utilizando HTML, CSS e JavaScript, uma solução Web sobre sustentabilidade. Pretende-se ter um Website dinâmico que apresente várias secções sobre a temática, podendo estas ser geridas por utilizadores administradores. Poderão existir outros utilizadores que sejam responsáveis apenas pelos conteúdos, ou parte deles. Por exemplo, o utilizador GC1 poderá ser o responsável pelos conteúdos da secção 1 enquanto que um utilizador GC2 poderá ter a responsabilidade de gerir a secção 2. Isto significa que poderá existir o conceito de comissão, onde cada membro terá as suas responsabilidades de gestão no Website. Pretende-se ainda que a solução Web ofereça a possibilidade de alguma análise de dados sobre projetos, tecnologias, etc., com visualização de estatísticas que possam ser interessantes para quem pretende entender como a temática da sustentabilidade tem vindo a evoluir nos anos mais recentes.

A decisão pela escolha desta temática justifica-se pelo facto de a sustentabilidade ser uma preocupação crescente, uma vez que os recursos naturais estão cada vez mais ameaçados em consequência da ação humana. As Nações Unidas dão uma enorme relevância a esta temática através dos 17 objetivos para um Desenvolvimento Sustentável¹. Destaca-se ainda que a tecnologia e as Ciências da Computação têm vindo a ser cada vez mais usadas em projetos um pouco por todo o Mundo para “atacarem” esta problemática.

2 Fases e Tecnologia

Preferencialmente, a solução Web deve ser desenvolvida tendo por base a **metodologia *Single-Page Application* (SPA)**, ou seja, ter uma página em HTML5 em que o conteúdo vai sendo gerado e atualizado dinamicamente mediante a interação do utilizador com o sistema.

Numa primeira fase, 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, numa segunda fase, pretende-se que o aluno adicione as funcionalidades de CRUD (*Create, Read, Update, Delete*) para lidar com os dados na vertente *client-side*, guardando os dados em tempo real e recorrendo a listas de objetos em tecnologia *JavaScript*. 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 interação com o utilizador.

Ficando para uma última fase 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), 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

¹ <https://sustainabledevelopment.un.org/>

linguagem Node.js, sendo que o repositório de dados será criado recorrendo, de preferência, a MongoDB².

3 O Gestor de conteúdos dinâmicos

Apresentam-se aqui, de forma resumida, as especificações e requisitos básicos que deverão ser considerados pelos alunos de forma a implementarem a aplicação Web para a gestão de conteúdos dinâmicos, subordinado ao tema Sustentabilidade.

Para uma gestão realista de um determinado tema, terá que existir um foco principal na adequação dos conceitos que o sistema pretende suportar. Como tal, deverão descobrir quais as **possíveis entidades a considerar** como, por exemplo, para suportar possíveis configurações do gestor de conteúdos ou estatísticas várias, entre muitas outras, dependendo das secções e funcionalidades que se queiram implementar. Assim, para além dos conteúdos estáticos relativos ao tema Sustentabilidade, a aplicação Web deverá permitir uma gestão de conteúdos dinâmicos, ao possibilitar a criação, visualização, edição e eliminação dos mesmos.

A aplicação já referida também deverá de efetuar a gestão dos Utilizadores, i.e., o sistema deverá permitir a criação, edição e remoção de Utilizadores. Convém referir que os Utilizadores deverão de ter bem definidos, através de níveis de acesso adequados aos papéis que vão desempenhar na aplicação e, por conseguinte, quais as tarefas que poderão realizar na mesma. Sugere-se a criação de um módulo de autenticação.

Como referido anteriormente, o intuito deste trabalho é desenvolver uma SPA 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 proporcionará a criação de conteúdo através do DOM. Contudo, não é necessário que todo o conteúdo seja construído através do DOM, 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.**

A decisão pela escolha desta temática justifica-se pelo facto de dar alguma continuidade ao trabalho efetuado pelos Alunos, ao criarem um conjunto de páginas estáticas sobre o tema sustentabilidade.

4 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

² <https://www.mongodb.com>

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).

Sugere-se a utilização de mongoose³ para ligação à base de dados. mongoose é um módulo do NodeJS desenvolvido para ligação ao MongoDB, fornecendo uma solução simples, baseada em esquemas para modelar os dados da aplicação.

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.

5 Regras no desenvolvimento do projeto

5.1 Regras de implementação e codificação

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

- Etapa 1 - HTML, CSS para definição da IU;
- Etapa 2 - JavaScript para interatividade e gerir entidades com geração de objetos por omissão;
- Etapa 3 - Construção de um repositório de dados em MongoDB e de uma API de WebServices RESTful em Node.js, que irá interagir com a SPA através da tecnologia AJAX;

Os alunos deverão colocar em prática os conceitos fundamentais da programação para a Internet, 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.**

5.2 Constituição de grupos

Cada projeto deverá ser elaborado por um grupo de dois alunos, podendo eventualmente ser desenvolvido individualmente (devendo ter aprovação prévia do respetivo docente de laboratório). Não serão permitidos grupos com mais do que dois alunos, ou grupos formados por alunos de turmas de laboratório que não tenham o mesmo docente.

³ <https://mongoosejs.com/>

5.3 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:

- **Análise inicial:** até às **23:59 do dia 02 de junho de 2020**. Um documento resumo para que o docente de laboratório possa ajudar no dimensionamento do projeto e que apresente:
 - a estrutura de secções,
 - os requisitos funcionais de base a implementar,
 - mockups correspondentes,
 - modelo de dados,
 - projeto preliminar - versão estática com alguma dinâmica,
 - e lista de possíveis frameworks/APIs a usar e com que objetivo.
- **Projeto final:** até às **23:59 do dia 03 de julho de 2020**. Esta será a entrega final e deverá incluir o projeto completo, englobando todas as etapas. O que foi entregue na fase de análise inicial poderá ter sido retificado para esta fase final.

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 `.vscode` com os ficheiros `launch.json` e `tasks.json`;
 - 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 do repositório de dados inicial.
- Os ficheiros JavaScript deverão ser documentados através de JSDoc.

Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado (formato ZIP, RAR ou outro), cujo nome deverá seguir a seguinte estrutura: ProjPI2020_<numAluno1>_<numAluno2>.zip|rar.

6 Avaliação

6.1 Regras de Avaliação

- A classificação da solução terá em conta a qualidade da programação (fatores de qualidade do *software*), a estrutura do código criado, 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, presencial ou através de Zoom/Teams, 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.

6.2 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; novos requisitos/funcionalidades, etc.)	10%

7 Anexo

7.1 Web Services

Um *Web Service*⁴ 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.

7.1.1 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.

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”.

7.1.2 Comunicação entre a SPA e o serviço RESTful

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

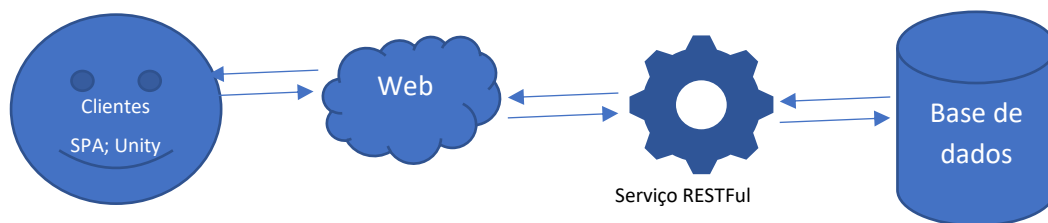


Figura 1 - Arquitetura.

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 que 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

⁴ Para mais informação: https://en.wikipedia.org/wiki/Web_service

para verificar se é possível carregar o recurso uma determinada entidade, que poderia ser *ProjetoDeSustentabilidade*.

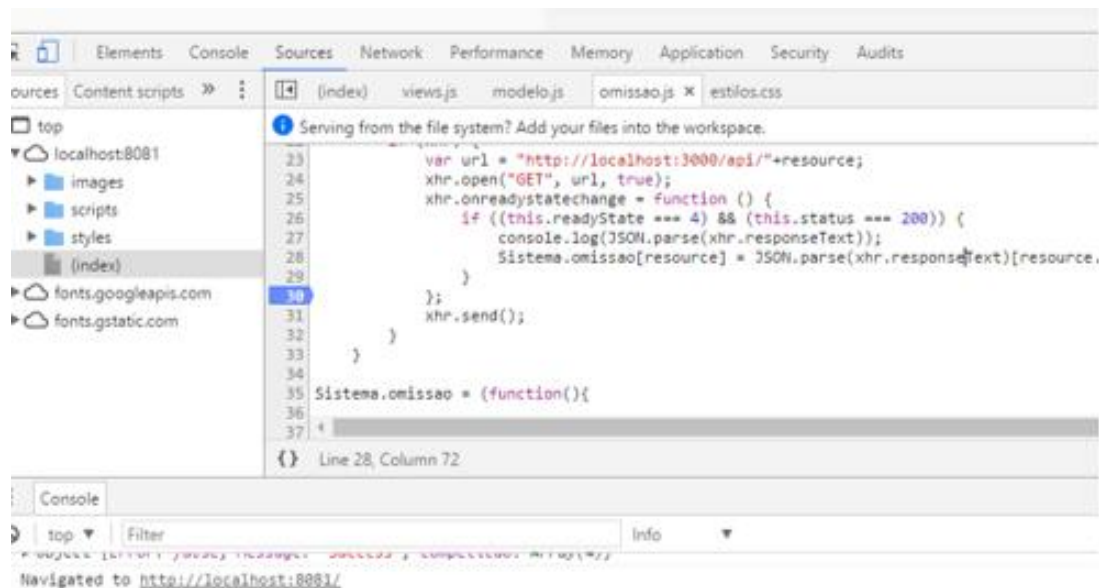


Figura 2 - Exemplo de consumo do serviço RESTful por parte do cliente (ambiente de debug do Chrome).