

# Programação Avançada em Java

## Trabalho Prático 1

### Introdução

**Objetivo:** pretende-se com este projeto que os alunos sejam capazes de criar uma aplicação web HTML5, CSS3 e Javascript. Os alunos devem ser capazes de utilizar os elementos mais básicos da linguagem, bem como alguns aspetos mais avançados relacionados com HTML dinâmico. Os alunos deverão ainda utilizar o sistema de controlo de versões *Git*.

**Data de Entrega:** 5 de Fevereiro de 2023 (23H59)

**Grupos:** o projeto é realizado em grupos de **dois** elementos, constituição dos quais é definida pelos docentes.

### Objetivo

Neste projeto, os alunos deverão criar uma aplicação web usando HTML 5 (HyperText Markup Language 5), CSS3 (Cascade Style Sheets 3), e Javascript que tem como objetivo gerir listas de atividades (e.g., Microsoft To Do<sup>1</sup>). O layout da aplicação deve seguir um modelo convencional deste tipo de ferramentas. Deve existir um cabeçalho no topo da página ou do lado esquerdo no vertical, que irá conter um menu de navegação e outros elementos que poderão ser incluídos nas fases posteriores do projeto incremental. Deve, também, existir uma área principal, que será utilizada para colocar o conteúdo principal (i.e., formulário de login/registo, a lista detalhada de atividades, detalhes duma atividade), assim como outros possíveis elementos que poderão ser incluídos nas fases futuras do projeto incremental.

Os alunos deverão criar pelo menos 3 páginas diferentes interligadas entre si. Uma página de login que neste fase do projeto servirá apenas para o utilizador introduzir o seu nome, colocando esse no cabeçalho da aplicação. A segunda página que é uma página principal (i.e., *home page*) com a lista de atividades e uma terceira página que irá permitir editar/consultar detalhes de uma atividade específica. É importante garantir que o resultado final tem layout intuitivo e com boa apresentação visual. O conteúdo para esta parte do projeto pode ser estático com textos *dummy* (p.ex. <https://www.lipsum.com/>). Como requisito para esta parte da construção do site, para

---

<sup>1</sup> <https://todo.microsoft.com/>

além dos elementos HTML básicos que podem usar, deverão incluir elementos semânticos<sup>2</sup>.

Os alunos devem utilizar CSS3 como forma de definir o estilo do site, e.g. tipo de letra, tamanho, cor.

Toda esta parte do trabalho deverá estar guardada num repositório **Git**.

## Interatividade em JavaScript

Os alunos devem acrescentar algumas funcionalidades básicas que tornem a aplicação dinâmica e interativa. Para estas funcionalidades iremos recorrer de algum HTML + Javascript, sendo que os requisitos (funcionais e não funcionais) serão os seguintes:

1. deve ser possível adicionar, remover, completar e editar a lista de atividades;
2. deve existir um botão na página principal que permita adicionar uma nova atividade, assim como um campo de *input* através do qual o utilizador especifica a descrição da mesma (consultar as partes de Javascript sobre DOM<sup>3</sup> e Eventos<sup>4</sup>). Ao clicar no botão a nova atividade deve ser acrescentada à lista;
3. deve ser possível editar/consultar os detalhes de uma atividade. Ao clicar na atividade deve abrir uma nova página com o formulário através do qual o utilizador pode alterar, remover ou acrescentar detalhes. Nessa página deve existir um botão que permita guardar as alterações. A lista de atividades deve ser atualizada em conformidade (consultar HTML5 storage<sup>5</sup>). Nota: não será para utilizar JSPs;
4. caso sejam adicionadas itens à lista e o utilizador visite uma das páginas das atividades, ao voltar à página principal não deve perder informação das restantes atividades;
5. cada item da lista deve possuir um *check box*, ao clicar no qual esse item deve ser marcado como completo (e.g., strikethrough). As atividades concluídas devem passar para o fim da lista;
6. cada item da lista deve incluir um botão que permita eliminar a atividade respetiva;
7. na página de login (ou de Perfil) deve acrescentar a funcionalidade de poder guardar o nome do utilizador, sendo esses dados estarem visíveis no cabeçalho da aplicação em todas as páginas (deve recorrer ao localStorage).

---

<sup>2</sup> [https://www.w3schools.com/html/html5\\_semantic\\_elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp)

<sup>3</sup> [https://www.w3schools.com/js/js\\_html\\_dom\\_document.asp](https://www.w3schools.com/js/js_html_dom_document.asp)

<sup>4</sup> [https://www.w3schools.com/js/js\\_html\\_events.asp](https://www.w3schools.com/js/js_html_events.asp)

<sup>5</sup> [https://www.w3schools.com/html/html5\\_webstorage.asp](https://www.w3schools.com/html/html5_webstorage.asp)

## Pré-requisitos e Material de Referência

Estas funcionalidades deverão ser realizadas utilizando as seguintes ferramentas:

- Visual Studio Code (suporte à implementação HTML/CSS/JavaScript) opcional;

Existem várias referências na web que os alunos podem consultar sobre HTML5 e construção de páginas web. Para uma introdução aos conteúdos recomenda-se o tutorial do LearnToCodeAdvanced<sup>6</sup>. De forma geral, o W3Schools<sup>7</sup> é particularmente relevante pois contém tutoriais, tabelas de referências das linguagens e exemplos sobre HTML5, CSS e JavaScript.

As páginas do website devem ser **verificadas** usando o HTML Validator <sup>8</sup>e o CSS validator<sup>9</sup>.

## Entregáveis e avaliação

**Entregas:** os projeto deve ser guardado num **repositório Git**. Só será considerado para avaliação o código do **último commit** feito na data indicada na introdução do enunciado.

A avaliação é feita através de um conjunto de critérios que correspondem aos requisitos definidos para o projeto. Os alunos devem entregar a tabela de requisitos preenchida por quem fez/tomou responsabilidade (e.g., num README.md, Google sheets, Excel) junto no repositório *Git*.

**Nota importante:** cada elemento de grupo deve obrigatoriamente ser envolvido em pelo menos uma tarefa de cada grupo de requisitos (assinaladas com cores diferentes).

Podem consultar a grelha na figura da página seguinte ou através do link [Requisitos projeto 1](#)

---

<sup>6</sup> <https://learn.shayhowe.com/html-css/building-your-first-web-page/>

<sup>7</sup> <http://www.w3schools.com>

<sup>8</sup> [https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input)

<sup>9</sup> [https://jigsaw.w3.org/css-validator/#validate\\_by\\_input](https://jigsaw.w3.org/css-validator/#validate_by_input)

Grupo	Requisito
Interface - Frontend	Campos do Formulário para inserir informação da actividade
Interface - Frontend	Check box que serve para assinalar as actividades concluídas
Interface - Frontend	Campo de input para inserir descrição da actividade na página principal
Interface - Frontend	Um campo para introduzir o nome do utilizador e um botão para submeter os dados
Interface - Frontend	Botão para eliminar uma actividade
Interface Funcional - Frontend	Adição da actividade através de botão em Javascript
Interface Funcional - Frontend	Submissão do Formulário através de botão em Javascript
Interface Funcional - Frontend	Riscar a actividade concluída (strikethrough)
Interface Funcional - Frontend	Passar para o fim da lista a actividade concluída
Interface Funcional - Frontend	Eliminar um item da lista
Interface Funcional - Frontend	Actualizar o nome do utilizador visível no cabeçalho
Persistência - Frontend	Dados da Lista de actividades guardados corretamente em local storage
Persistência - Frontend	Nome do utilizador guardado em local storage
Não funcionais - Frontend	Elementos semânticos e a sua utilização correcta
Não funcionais - Frontend	Actualizar a lista de actividades sempre que há alterações nos dados

**Defesa:** As defesas terão uma duração de **45 minutos**, sendo necessário uma inscrição prévia na plataforma Infoestudante, na página da disciplina. Durante a defesa, todo o projeto deve estar a funcionar e deve ser executado sem erros. Se durante a defesa o código apresentar diferenças face ao entregue, será **descontado 25%** da nota. Também é importante garantir que cheguem à defesa preparados (i.e., código aberto, IDE aberto, web browser com a aplicação a funcionar). Caso contrário será aplicada uma **penalização de 10%**.

**Nota Importante:** qualquer componente implementado, mas não comprovado na defesa recebe **35% de penalização**.