

Programação Avançada em Java

Trabalho Prático: Projeto 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: 4 de fevereiro de 2024 (23H59)

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

Conceito

Neste projeto os alunos deverão desenvolver uma aplicação web para gerir um quadro *Scrum*¹, inspirado nas metodologias ágeis de desenvolvimento de *software*. Assim, o *Scrum* é uma *framework* ágil que visa facilitar a colaboração e a entrega contínua de produtos, ajudando as equipas a organizar e acompanhar o progresso das tarefas. Este quadro é geralmente composto por colunas que representam as diferentes etapas do progresso como "*To Do*" (A Fazer), "*Doing*" (Em Andamento) e "*Done*" (Feito). Cada coluna contém cartões que representam tarefas específicas a serem realizadas.

O quadro *Scrum* tem o propósito de oferecer transparência, facilitando a visualização do trabalho em andamento por toda a equipa, identificando possíveis obstáculos e promovendo uma comunicação eficiente dentro da equipa.

Para além do quadro *Scrum*, outro elemento fundamental nas metodologias ágeis é o momento de retrospectiva², que desempenha um papel crucial no processo de melhoria contínua. A retrospectiva é uma prática recorrente no final de cada *sprint* (ciclo de desenvolvimento), onde a equipa reúne para refletir sobre o trabalho realizado. Este momento proporciona um momento de avaliação do que funcionou bem, permite identificar áreas de melhoria e definir ações para implementar mudanças positivas no próximo ciclo de desenvolvimento. Durante a retrospectiva,

¹ <https://www.forbes.com/advisor/business/what-is-a-scrum-board/>

² <https://www.scrumportugal.pt/sprint-retrospective/>

todos os membros da equipa são motivados a partilhar quais foram as suas experiências, desafios enfrentados e sucessos. Este ambiente de abertura e colaboração é essencial para construir confiança dentro da equipa.

Ao desenvolver esta aplicação, os alunos terão a oportunidade de criar uma ferramenta interativa que simula um ambiente empresarial, proporcionando uma experiência prática na implementação de conceitos *Scrum*, replicando o dinamismo e colaboração que acontece em muitas equipas de desenvolvimento de software.

Objetivo

Neste projeto, os **grupos de 2** deverão criar uma aplicação *web* usando as tecnologias identificadas na introdução, que tem como objetivo gerir um quadro *Scrum*. O **grupo 3**, para além do quadro *scrum*, deve implementar uma página para registo das retrospectivas. O layout da aplicação deve seguir um modelo convencional deste tipo de ferramentas, demonstrado na Figura 1:

1. Existir um cabeçalho (*Header*) o topo da página ou do lado esquerdo (*Aside*) na vertical, que irá conter um menu de navegação e outros elementos que poderão ser incluídos nas fases posteriores do projeto incremental;
2. Existir uma área principal (*Main*), que será utilizada para colocar o conteúdo principal (i.e., formulário de *login*/registo, o quadro *Scrum*, página de retrospectivas, detalhes duma atividade), assim como outros possíveis elementos que poderão ser incluídos nas fases futuras do projeto incremental;
3. Existir um rodapé (*footer*) para incluir informações referentes ao projeto que acharem necessárias.

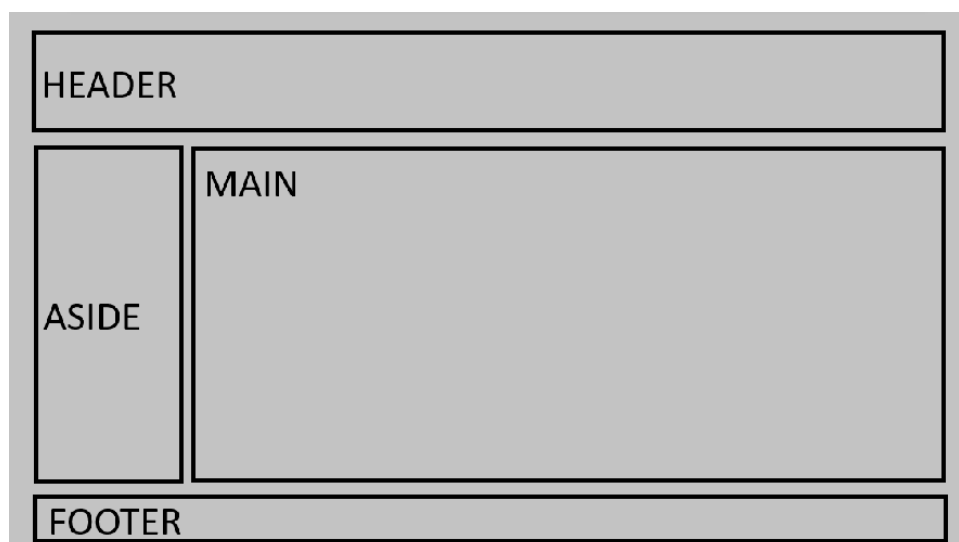


Figura 1 - Proposta de Layout

Os alunos devem criar pelo menos três páginas interligadas entre si. Uma página de *login* que, nesta fase, servirá apenas para o utilizador inserir o seu nome, exibindo-o

no cabeçalho da aplicação. A segunda página será a página principal (i.e., home page) com o quadro *Scrum* que contém as colunas "*To Do*", "*Doing*" e "*Done*", e a terceira página irá permitir a edição/consulta dos detalhes de uma tarefa específica. O conteúdo pode ser estático, nesta fase, utilizando textos fictícios (por exemplo, <https://www.lipsum.com/>), no entanto, é importante garantir que o resultado tem um *layout* intuitivo e boa apresentação visual. Como requisito para esta fase da construção da aplicação *web*, para além dos elementos *HTML* básicos que podem usar, deverão incluir elementos semânticos³.

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

Interatividade em JavaScript

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

Todos os grupos:

1. Deve ser possível adicionar, remover, e mover as tarefas no quadro *Scrum*.
2. Deve existir um botão na página principal para adicionar uma nova tarefa ao quadro. Uma tarefa é composta por um título e descrição. Para isto, devem existir campos de input através dos quais o utilizador insere os dados relativos ao título e a descrição (consultar as partes de *Javascript* sobre *DOM*⁴ e *Eventos*⁵) Ao criar uma tarefa, esta deve ser adicionada à coluna "*To Do*".
3. Edição e consulta dos detalhes de uma tarefa. Ao clicar na tarefa, deve abrir uma nova página com as informações relativas a esta, permitindo ao utilizador alterar, eliminar ou adicionar detalhes à tarefa. Essa página deve ter um botão para guardar as alterações, atualizando o quadro *Scrum* (consultar *HTML5 Storage*⁶).
4. Quando são adicionadas novas tarefas ao quadro, e o utilizador visita a página de detalhes de uma dada tarefa, posteriormente, ao regressar à página principal, não deve perder informações das restantes atividades que estavam no *board*;
5. Cada tarefa no quadro deve ter a capacidade de ser movida entre as colunas "*To Do*", "*Doing*" e "*Done*".

³ https://www.w3schools.com/html/html5_semantic_elements.asp

⁴ https://www.w3schools.com/js/js_html_dom_document.asp

⁵ https://www.w3schools.com/js/js_html_dom_events.asp

⁶ https://www.w3schools.com/html/html5_webstorage.asp

6. Na página de *login* (ou de perfil), deve ser possível guardar o nome do utilizador, permanecendo esse dado sempre visível no cabeçalho da aplicação em todas as páginas (utilizando *localStorage*).

Grupo de 3 alunos:

Devem criar um menu extra, no menu descrito no ponto 1) da secção Objetivo, que permita aos membros da equipa, no fim de cada ciclo de desenvolvimento, realizar o momento de retrospectiva. Esta nova página deve agregar e mostrar todos os momentos de retrospectiva realizados, servindo como um histórico. Cada momento de retrospectiva é realizado preenchendo os seguintes dados:

1. Data da realização da retrospectiva;
2. Membros presentes;
3. Comentários: Um campo de entrada de texto que permite aos membros da equipa adicionar comentários sobre a *sprint* recente. Cada comentário deve ser associado a uma categoria, como "Pontos Positivos", "Desafios" ou "Sugestões de Melhoria".

Todo o trabalho deverá estar guardado num repositório *Git*.

Pré-requisitos e Material de Referência

Estas funcionalidades deverão ser realizadas utilizando as seguintes ferramentas:

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

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 LearnToCodeAdvanced⁷. De forma geral, o W3Schools⁸ é particularmente relevante pois contém tutoriais, tabelas de referências das linguagens e exemplos sobre *HTML*, *CSS* e *JavaScript*.

As páginas do website devem ser **verificadas** usando o *HTML Validator*⁹ e o *CSS validator*¹⁰.

Entregáveis e avaliação

⁷ <https://learn.shayhowe.com/html-css/building-your-first-web-page/>

⁸ <http://www.w3schools.com>

⁹ https://validator.w3.org/#validate_by_input

¹⁰ https://jigsaw.w3.org/css-validator/#validate_by_input

Entregas: o 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, apresentados na Figura 2.

Os alunos devem entregar a tabela de requisitos preenchida por quem fez/teve 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 tabela de requisitos na figura da página seguinte ou através do link [Requisitos projeto 1](#)

Categoria	Requisito	Grupos
Interface - Frontend	Incluir campos para inserir informações relativas ao Login de um utilizador (<i>username</i> e <i>password</i>), e um botão para efetuar o <i>login</i>	Todos
Interface - Frontend	Incluir campos de input para permitir a inserção dos detalhes das tarefas (título e descrição) e um botão para guardar	Todos
Interface - Frontend	Incluir um campo para introduzir o nome do utilizador	Todos
Interface - Frontend	Botão para eliminar uma tarefa	Todos
Interface - Frontend	Incluir menu no header para aceder à página de Retrospectiva	Grupo de 3 alunos
Interface - Frontend	Incluir campos de input para preencher os detalhes do momento de retrospectiva (data, membros presentes, comentários)	Grupo de 3 alunos
Interface Funcional - Frontend	Submissão do formulário de <i>login</i> através de botão em <i>Javascript</i>	Todos
Interface Funcional - Frontend	Adição da tarefa através de botão em <i>Javascript</i>	Todos
Interface Funcional - Frontend	Mover entre <i>boards</i> as tarefas (<i>To do - Doing - Done</i>)	Todos
Interface Funcional - Frontend	Eliminar uma tarefa	Todos
Interface Funcional - Frontend	Actualizar o nome do utilizador visível no cabeçalho	Todos
Interface Funcional - Frontend	Submissão do formulário de retrospectiva com todos os campos através de um botão em <i>Javascript</i>	Grupo de 3 alunos
Persistência - Frontend	Dados das tarefas guardados corretamente em <i>local storage</i>	Todos
Persistência - Frontend	Nome do utilizador guardado em <i>local storage</i>	Todos
Não funcionais - Frontend	Elementos semânticos e a sua utilização correcta	Todos
Não funcionais - Frontend	Actualizar a lista de atividades sempre que há alterações nos dados	Todos

Figura 2- Requisitos do Projeto



Defesa: As defesas terão uma duração de **40 minutos**, sendo necessário uma inscrição prévia na plataforma Inforestudante, 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%**. Os alunos terão 5 minutos para apresentar a sua aplicação a funcionar. O restante tempo é para análise do código implementado.

Nota Importante: qualquer componente implementado, mas que os alunos demonstrem não conseguir explicar, na íntegra, o seu funcionamento, recebe **50% de penalização**.