

## **Relatório do Projeto CSS – Fase 2**

### **Trabalho realizado por:**

**Diogo Pinto      fc55179**

**Ivo Estrela      fc51051**

**Thiago Duarte   fc53636**

Explicação da arquitetura interna dos componentes: Aplicação WEB, Aplicação Desktop, e API REST.

**Aplicação WEB (*Server-side rendering*):** A componente web foi implementada utilizando o padrão *Model-View-Controller* (MVC). Por operar no contexto de *server-side rendering*, a aplicação gera páginas *HTML* do lado do servidor, preenchendo-as com os dados necessários antes de enviá-las ao cliente. Os *Controllers* recebem os pedidos, interagem com os *Services* necessários, que retornam dados (geralmente), e estes dados, juntamente com outras possíveis informações, são inseridos no(s) *Model(s)*. O *Model* é então utilizado para renderizar a *View* utilizando templates *Thymeleaf*. Estes templates, inserem dinamicamente os dados fornecidos pelo(s) *Model(s)* e retornam uma página *HTML*.

**Aplicação Desktop (*Client-side rendering*):** A componente *desktop* é uma aplicação *rich client* implementada utilizando o padrão *Model-View-Controller* (MVC). Operando no contexto de *client-side rendering*, a aplicação faz solicitações *HTTP* para a *API REST* exposta pelo servidor. Os dados recebidos (em formato *JSON*) por meio das respostas *HTTP*, são então preenchidos nas *Views* correspondentes (ficheiros *.fxml*), do lado do cliente. O *Controller*, nesse contexto, lida com as interações do usuário e atualiza o *Model* de acordo (através dos pedidos *HTTP* à *API REST* exposta), que em seguida, atualiza a respectiva *View*.

**API REST:** A componente *API REST* atua como uma interface entre o *backend* e a aplicação *desktop* (ou qualquer outra aplicação que siga a interface proposta pela *API RESTful*). Ela fornece *endpoints* que as aplicações remotas podem utilizar para interagir com o sistema. Foram implementados controladores (*Controllers*) para cada recurso utilizado pela *API*. Esses controladores lidam com os pedidos *HTTP* recebidos, comunicam-se com os serviços necessários e retornam respostas *HTTP* apropriadas (dados em formato *JSON*, se necessário, e *HTTP status code*) para a aplicação remota que fez o pedido.