

Programação Avançada em Java

Trabalho Prático 4

Introdução

Objetivo: pretende-se com este projeto que os alunos façam uma extensão da aplicação implementada no projeto 3. Como no projeto anterior, deve ser utilizada persistência de dados do lado do servidor através de um base de dados MySQL. Devem também ser usados JPA e criados endpoints REST para a gestão dos dados. Devem ser capazes de criar uma aplicação web usando **ReactJS** e recorrer a tecnologias como JAX-RS e EJBs para desenvolver RESTful Web Services. Os alunos deverão ainda utilizar o sistema de controlo de versões Git, bem como a ferramenta de build Maven.

Data de Entrega: 26 de Março de 2023

Grupos: O projeto é realizado individualmente.

Especificação

Neste projecto pretende-se continuar a implementação da aplicação que permite gerir listas de actividades. O projecto foca-se maioritariamente na implementação de um Frontend, com ajustes pontuais na parte do Backend. O objectivo principal é implementar no Frontend todas as funcionalidades definidas no projecto 3 para as quais já foram desenvolvidas os endpoints necessários. Em caso de dúvida, é recomendado consultar o enunciado do projeto 3 para recordar as funcionalidades específicas.

Uma atividade deve ser caracterizada pelas seguintes propriedades:

- A categoria a qual pertence, título, uma descrição;
- Datas/horas:
 - Data/hora de criação (gravada automaticamente)
 - Data/hora de início (opcional, se não for introduzida pelo utilizador deve ser considerada a data/hora de criação);
 - Data/hora limite para a realização/deadline (opcional, o utilizador pode optar por não especificar a data e hora limite para realização da atividade. Neste caso a data de término fica indefinida);
 - Data/hora de conclusão da atividade (pode ser gravada automaticamente, quando o utilizador marca a atividade como concluída, ou pode ser inserida manualmente pelo utilizador).
- Estado (realizada ou não);

Os alunos são livres de adicionar propriedades extra conforme a necessidade.

Arquitetura do Sistema a Desenvolver

Deve ser seguida a arquitetura apresentada na Figura 1, cujos módulos são explicados em baixo.

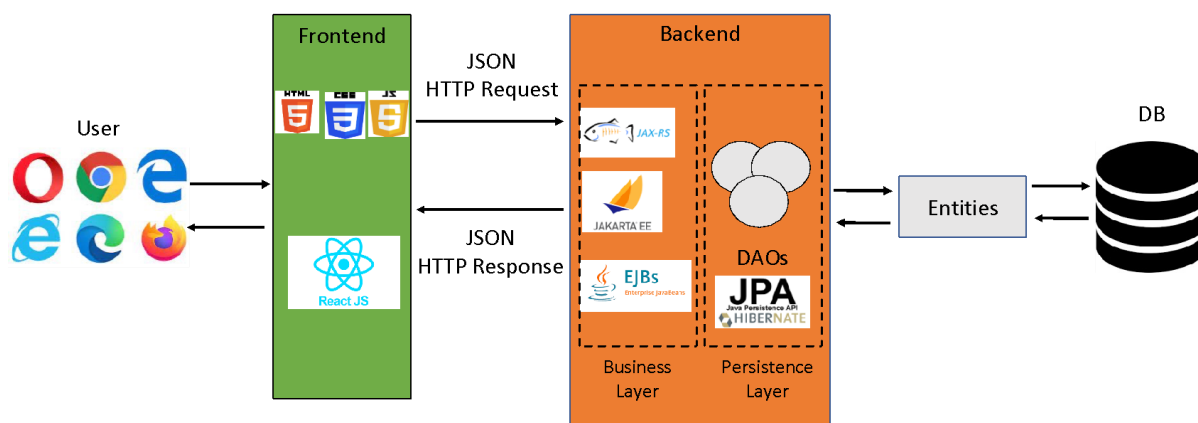


Figura 1. Arquitectura do sistema a implementar.

O módulo **Backend** corresponde a um projeto Maven que interage com uma base de dados MySQL usando Java Persistence API com Hibernate. O modelo de dados deve ser adaptado do projeto 3, no qual devem incluir todas as entidades necessárias e/ou alterações às entidades já existentes. É permitido escolher entre Criteria API e JPQL para fazer consultas aos dados. **Cada utilizador deve ter a capacidade de efetuar vários logins. Isto implica a necessidade de armazenar vários tokens para cada utilizador.**

O Backend deve expor uma API REST (completar a API implementada no projeto 3) que poderá ser consumida por outro sistema, neste caso pelo módulo Frontend. O **formato dos dados a serem trocados deve ser JSON.**

O módulo **Frontend** corresponde a um projeto que interage com o utilizador através de uma interface web organizada e usável, baseada em React.

Serviço REST

Os alunos são livres de adicionarem **novos** endpoints, se assim o desejarem. No entanto, devem procurar seguir as boas práticas de interfaces RESTful e procurar usar *status codes* adequados em todas as respostas produzidas. Tal como já foi referido anteriormente, todas as mensagens (tanto pedidos como respostas) REST devem usar o formato JSON.

Frontend

Neste projeto, os alunos devem completar a parte frontend do projeto 3 para ser

possível aceder a todas as funcionalidades mencionadas acima usando o browser.

Entrega do projeto e avaliação

Entregas: os projetos devem ser guardados num **repositório Git**. Só será considerado para avaliação o código do **último commit** feito na data indicada na introdução deste enunciado.

A avaliação é feita através de um conjunto de critérios que correspondem aos requisitos definidos para o projeto 4.

Podem consultar a grelha na figura da página seguinte ou através do link [Requisitos projecto 4 - alunos](#)

Defesa: As defesas terão uma duração de **30 minutos**. 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 os alunos cheguem à defesa preparados (i.e., código aberto no IDE, web browser com a aplicação pronta a mostrar). Caso contrário será aplicada uma **penalização de 10%**.

Nota importante: qualquer funcionalidade implementada mas não demonstrada na defesa recebe **35% de penalização**.

Grupo	Requisito
Interface Funcional - Frontend	Adicionar uma categoria de atividades (U2 do projeto 3)
Interface Funcional - Frontend	Alterar uma categoria de atividades (U3 do projeto 3)
Interface Funcional - Frontend	Apagar uma categoria de atividades (U4 do projeto 3)
Interface Funcional - Frontend	Adicionar atividades em cada categoria (U5 no projeto 3)
Interface Funcional - Frontend	Alterar atividades (U6 no projeto 3)
Interface Funcional - Frontend	Apagar atividade (U7 no projeto 3)
Interface Funcional - Frontend	Admin: alterar a informação do perfil de outros utilizadores (A1 no projeto 3)
Interface Funcional - Frontend	Admin: Adicionar um novo utilizador (A2 no projeto 3)
Interface Funcional - Frontend	Admin: Consultar lista de todos os utilizadores (A3 no projeto 3)
Interface Funcional - Frontend	Admin: Apagar um utilizador e as suas atividades do sistema (A4 no projeto 3)
Interface Funcional - Frontend	Admin: Consultar o perfil de um utilizador específico (A5 no projeto 3)
Interface Funcional - Frontend	Admin: Consultar lista de atividades de um utilizador (A6 no projeto 3)
Interface Funcional - Frontend	Admin: Alterar a role de um utilizador (user/admin) (A7 no projeto 3)
Interface Funcional - Frontend	Admin: Adicionar um admin no sistema (A8 no projeto 3)
Interface Funcional - Frontend	Admin: Consultar lista de todos os admins (A9 no projeto 3)
Interface Funcional - Frontend	Admin: Apagar um admin (A10 no projeto 3)
Persistência - Backend	Entidades atualizados
Requisitos funcionais - Backend	Endpoints atualizados
Requisitos funcionais - Backend	Utilizadores conseguem fazer vários logins simultaneamente