

Projeto - Arquitetura de Objetos

O projeto consistirá no desenvolvimento do Back-End (com Spring) e do Front-End (com React e MUI) de uma parte de uma das aplicações definidas no fim deste documento. As aplicações serão distribuídas por sorteio entre os grupos. A aplicação deve acessar um banco de dados H2.

Além do desenvolvimento das aplicações, cada grupo terá que interagir com um outro grupo para fazer uma integração entre as aplicações.

Datas

O projeto será iniciado no dia 27/04 após o sorteio dos projetos entre os grupos.

A data de entrega do projeto é no dia 09/05, quando os grupos farão a apresentação do projeto.

As aulas dos dias 02/05 e 04/05 serão exclusivamente para o desenvolvimento dos projetos e para tirar dúvidas.

Entrega

A entrega será feita via GitHub, deverá existir dois repositórios por grupo, um para o FrontEnd e outro para o BackEnd, os repositórios podem ser públicos.

Correção

- 3 pontos para o desenvolvimento do Back-End
- 3 pontos para o desenvolvimento do Front-End
- 2 pontos para o desenvolvimento e documentação da API que será usada pelo outro grupo.
- 2 pontos para o desenvolvimento do cliente da API do outro grupo.

Alguns pontos que serão avaliados:

- Seguir o padrão REST na nomeação nas rotas
- Todas os BE devem ter um swagger
- Validação de erros
- Utilização de componentes do MUI
- Organização dos projetos

Para testar a aplicação sem depender do outro grupo, pode ser utilizada ferramentas que fazem o Mock de APIs, como por exemplo o Mookoon:

<https://mockoon.com/tutorials/getting-started/>

Apresentação

- Os grupos devem apresentar a parte 1 e 2 de cada um dos projetos conjuntamente, isso pode ser feito em apenas em uma máquina, ou cada aplicação pode ser executada separadamente e as chamadas feitas na rede.
- Caso a integração não funcione, cada grupo deve utilizar uma ferramenta de Mock para simular a parte do outro grupo, nesse caso os 2 pontos da integração serão perdidos.

Projetos

Projeto 1 - Aplicação de Viagens em Carros Compartilhados

Grupo 1

A primeira aplicação é para a utilização de carros compartilhados, como o Uber, o projeto será dividido em um duas partes, o primeiro grupo será responsável por cadastrar motoristas do aplicativo, contendo rotas para o cadastro do motorista e do veículo.

Devem existir rotas e telas para cadastro, edição, exclusão e listagem dos motoristas. Deve também existir uma rota que valida os dados de um motorista, indicando se ele pode ou não fazer viagens. (Utilizar um status do motorista para fazer esse controle)

A aplicação deve ter duas rotas, uma GET que retorna um motorista disponível, e outra POST ou PUT que atualiza o status do motorista para disponível novamente. Um motorista nunca deve poder fazer duas viagens ao mesmo tempo.

A aplicação deve ter uma tela que seleciona um motorista, ela lista todas as viagens daquele motorista (a rota será desenvolvida pelo outro grupo).

Alguns dados que todos os motoristas devem ter:

- Nome, CPF, Placa do Veículo, Modelo do Veículo, Preço da Viagem por Hora, Status de Utilização da aplicação (PENDENTE, LIBERADO, CANCELADO) e Status de Ocupação (DISPONIVEL, INDISPONIVEL).

Grupo 2

A segunda parte da aplicação deve possibilitar a criação e finalização de viagens, toda viagem deve ter um motorista, que deve ser retornado da aplicação do grupo 1. Toda viagem deve ter um endereço de origem e destino. Se a rota que retorna um motorista der erro ou não retornar nada a viagem deve ser cadastrada como ERRO.

Essa parte do projeto deve ainda ter uma rota que passado o identificador de um motorista, deve retornar todas as viagens cadastradas para aquele motorista.

Deve existir quatro telas nesta parte da aplicação, a de cadastro de uma viagem, a de confirmação/erro de uma viagem, a de finalização de uma viagem e uma listagem de todas as viagens realizadas com um botão de exclusão das viagens.

Alguns dados que todas as viagens devem ter:

- Data de Início, Origem, Destino, Horas da Viagem, Preço Total, Status (CONFIRMADO, ERRO, FINALIZADO) e o id do motorista.

Projeto 2 - Aplicação de Pedidos de Refeições

Grupo 1

A segunda aplicação é para o cadastro de pedidos, como o iFood, o projeto será dividido em um duas partes, o primeiro grupo será responsável por cadastrar os entregadores do aplicativo.

Devem existir rotas e telas para cadastro, edição, exclusão e listagem dos entregadores. Deve também existir uma rota que valida os dados de um entregador, indicando se ele pode ou não fazer entregas. (Utilizar um status do motorista para fazer esse controle)

A aplicação deve ter duas rotas, uma GET que retorna um entregador disponível, e outra que atualiza o status do entregador para disponível novamente. Um entregador nunca deve poder fazer duas viagens ao mesmo tempo.

A aplicação deve ter uma tela que selecionado um entregador, ela lista todas as viagens daquele entregador (a rota será desenvolvida pelo outro grupo).

Alguns dados que todos os entregadores devem ter:

- Nome, CPF, Tipo Veículo, Preço da Viagem por KM, Status de Utilização da aplicação (PENDENTE, LIBERADO, CANCELADO) e Status de Ocupação (DISPONIVEL, INDISPONIVEL).

Grupo 2

A segunda parte da aplicação deve possibilitar a criação e finalização de pedidos, toda viagem deve ter um entregador, que deve ser retornado da aplicação do grupo 1. Toda viagem deve ter um endereço de origem e destino.

Essa parte do projeto deve ainda ter uma rota que passado o identificador de um entregador, deve retornar todas as entregas cadastradas para aquele motorista.

Deve existir quatro telas nesta parte da aplicação, a de cadastro de um pedido, a de confirmação/erro de um pedido, a de finalização de uma entrega e uma listagem de todas as entregas realizadas com um botão de exclusão de pedidos.

Alguns dados que todas as entregas devem ter:

- Data de Início, Origem, Destino, Horas da Viagem, KMs da viagem, Preço Total, Status (CONFIRMADO, ERRO, FINALIZADO) e o id do entregador.

Projeto 3 - Aplicação de Aluguel de Bicicletas

Grupo 1

A terceira aplicação é para o cadastro de aluguel de bicicletas, o projeto será dividido em duas partes, o primeiro grupo será responsável por cadastrar as bicicletas.

Devem existir rotas e telas para cadastro, edição, exclusão e listagem das bicicletas. Deve também existir uma rota que atualiza o status de uma bicicleta, indicando se ela está em funcionando ou em manutenção, indicando se ela pode ser usada ou não. (Utilizar um status da bicicleta para fazer esse controle)

A aplicação deve ter duas rotas, uma GET que retorna uma bicicleta disponível, e outra que atualiza o status do bicicleta para disponível novamente. Uma bicicleta nunca pode ser utilizada mais de uma vez ao mesmo tempo

A aplicação deve ter uma tela que selecionada uma bicicleta, ela lista todas as vezes que aquela bicicleta foi utilizada. (a rota será desenvolvida pelo outro grupo).

Alguns dados que todas as bicicletas devem ter:

- Modelo, Tipo, Preço do Aluguel por Hora, Status de Utilização (LIBERADA, MANUTENCAO) e Status de Ocupação (DISPONIVEL, INDISPONIVEL).

Grupo 2

A segunda parte da aplicação deve possibilitar a criação e finalização de aluguéis de uma bicicleta, toda utilização deve ter uma bicicleta, que deve ser retornada da aplicação do grupo 1. Toda viagem deve ter um endereço de origem e destino.

Essa parte do projeto deve ainda ter uma rota que passado o identificador de uma bicicleta, deve retornar todos os aluguéis cadastrados para aquela bicicleta.

Deve existir quatro telas nesta parte da aplicação, a de cadastro de um aluguel, a de confirmação/erro de um aluguel, a de devolução de um aluguel e uma listagem de todas os aluguéis realizados com um botão de exclusão de alugueis.

Alguns dados que todas os alugueis devem ter:

- Data de Início, Origem, Destino, Horas da Viagem, KMs da viagem, Preço Total, Status (CONFIRMADO, ERRO, FINALIZADO) e o ID da bicicleta.