

Orientações

1. Você receberá um zip contendo dois projetos: front (Angular) / back (Java);
2. Os projetos estão previamente configurados, a integração entre: front e back está funcional;
3. Os projetos sobem respectivamente nas portas: 4200 e 8080 (defaults)
4. Nos projetos constam classes de exemplos e uma capacidade (GET) implementada;
5. Após as alterações, suba os projetos em um GIT, que seja possível compartilhamento para avaliação.
6. Documente o passo a passo para execução da solução como completa, pode ser um README.md na raiz do diretório;

Front-End:

1. Importe o projeto na sua IDE de preferência.
2. Certifique-se que você possua o nodejs instalado em sua máquina. Versão maior ou igual a 16.17. (node -v no terminal)
3. Utilize o comando npm install --legacy-peer-deps para instalar todas as dependências do package.json
4. O projeto foi construído utilizando o framework Angular na versão 16.0 e já disponibilizamos a service PessoasService que faz o consumo do endpoint que lista todos os usuários que estão no banco de dados. Essa service está sendo chamada pelo app.component e os dados estão sendo apresentados no template. Fique à vontade para utilizá-la, ou não.
5. **Objetivo:** Criar uma tela que tenha um formulário e que faça o envio dos dados para o endpoint criado no projeto do backend. O formulário deverá conter pelo menos um select e deverá ter validações básicas de form (ex: preenchimento de campo obrigatório com mensagem de erro, regex para validação de e-mail ou CPF).
6. Criação de uma tela que exiba os dados de um usuário informando o CPF. Caso o CPF informado não seja encontrado, trate essa exceção.
7. Utilize algum framework css (ex tailwind, bootstrap) ou algum design system (Angular Material, PrimeNG) de sua preferência para criação do formulário e para tela de consulta.
8. Tenha pelo menos um componente ou service coberta 100% por testes unitários utilizando algum framework (karma, jasmine, jest).
9. As telas precisam ser responsivas.
10. Dados do formulário: nome, CPF, sexo, e-mail e telefone.
11. **Opcional:** A utilização de libs como rxjs, ngrx, ngxs serão consideradas como diferenciais. Fique à vontade para usá-las.

Back-End:

1. Importe o projeto na sua IDE de preferência;
2. Certifique-se de ter pelo menos uma JDK disponível com a versão 11 do Java, é a que será utilizada no projeto;
3. Certifique-se que seja possível compilar o projeto com o maven, é a ferramenta de gerenciamento de dependências que utilizaremos – antes de qualquer desenvolvimento, recomendamos o build o projeto: ***mvn clean install***
4. O projeto foi construído em Spring (Boot e Data) e já está configurado para subir um H2 (banco de dados em memória), no load inicial, um script já é executado criando o banco e com inserts defaults para facilitar sua compreensão (fique à vontade para explorar e alterar se necessário);
5. **Objetivo:** expor um endpoint que seja capaz de persistir os dados de uma pessoa, que será construído no projeto de front;
6. Crie uma **validação** que não seja possível a inserção de pessoa que já foi cadastrada, baseando-se no CPF (se atentar ao status code retornado);
7. Crie um teste unitário ao menos de uma classe construída (ideal que tenha cobertura de 100%);
8. **Opcional:** documente a API da forma que desejar. Entretanto o projeto já está previamente configurado para suportar o OpenAPI, use-o se achar pertinente;