

Desafio Estágio

1. Instalações Necessárias

Antes de iniciar, assegure-se de que as seguintes ferramentas estejam instaladas em seu computador:

- **JDK 17**
- **Jasper Studio**: Versão 6.11.
- **IntelliJ**

2. Conteúdos a Estudar

Para se preparar adequadamente, estude os seguintes tópicos:

- **jQuery**
- **Java 17**
- **Hibernate 5**
- **Wicket 7**
- **Angular 14**
- **SQL (MySQL)**
- **Angular Material**
- **Bootstrap**
- **JasperReports**
- **Padrões SOLID e MVC**

3. Recursos Adicionais

Abaixo, você encontrará links úteis com conteúdos relevantes para os tópicos mencionados. *Não incluímos links para soluções específicas de problemas:*

- [Desenvolvimento WEB com Java e Apache Wicket](#)
- [Callicoder: Apache POI para Excel](#)
- [Cursos Udemy](#)
 - [Solicitar o Acesso](#)
- [Alura](#)
 - [Solicitar o Acesso](#)
 - <https://cursos.alura.com.br/formacao-java>
 - <https://cursos.alura.com.br/course/java-collections>
 - <https://cursos.alura.com.br/formacao-boas-praticas-java>

4. Estrutura do Projeto

Utilize apenas um repositório para o projeto. A estrutura deve conter um diretório para o projeto angular, um para o projeto wicket e um para o backend em Java (Utilizando o Spring para a integração com o Angular). Seguindo os conceitos do padrão MVC, dividindo o projeto em camadas.

Obs: Comece implementando tudo em Wicket e, posteriormente, migre para Angular.

4.1 Modelo de Dados

- **Cliente:**
 - Tipo de Pessoa (Física ou Jurídica)
 - CPF ou CNPJ
 - Nome (para CPF)
 - RG (para CPF)
 - Data de Nascimento (para CPF)
 - Razão Social (para CNPJ)
 - Inscrição Estadual (para CNPJ)
 - Data de Criação (para CNPJ)
 - E-mail
 - Ativo (Sim ou Não)
 - Endereços (Lista de Endereços - relação 1:N com Foreign Key)

- **Endereço:**
 - Logradouro
 - Número
 - CEP
 - Bairro
 - Telefone
 - Cidade
 - Estado
 - Endereço Principal (Sim ou Não)
 - Complemento

5. Objetivo Geral

Desenvolver um CRUD para Clientes e Endereços com a possibilidade de impressão de relatórios em PDF e Excel.

5.1 Dicas para o Desenvolvimento

- Mantenha a estrutura do código genérica e organizada, permitindo reutilização em projetos futuros.
- Implemente as regras de validação para campos obrigatórios no serviço.
- Assegure que cada método tenha um único objetivo claro.
- Utilize nomes significativos para classes, métodos e atributos.
- Comente o código sempre que possível.
- Realize commits diariamente antes de encerrar o trabalho.

6. Desafio Geral

6.1 Tecnologias a Utilizar

- **Frameworks:** Wicket e Angular.
- **Validações:** Implemente a validação de campos obrigatórios.
- **Visibilidade de Campos:** Campos que alteram a visibilidade conforme ações do usuário.
- **JavaScript/jQuery:** Utilize para cálculos, máscaras de valores, etc.

6.2 Arquitetura

- Utilize o padrão **MVC** (Modelo, Visão e Controlador) com separação clara entre eles.
- Generalizar componentes para reutilização.

6.3 Organização do Código

- Leia todo o desafio e inicie criando as classes de modelo do projeto.
- Todas as operações de editar, excluir, incluir e listar devem ser realizadas na mesma página.
- Utilize janelas modais (Modal Window) quando necessário.
- Implemente listagens com paginação.

7. Funcionalidades dos CRUDs

- Utilize um FeedbackPanel no Wicket e outra biblioteca JS no Angular (ex: Notify.js) para exibir mensagens importantes (ex: sucesso no cadastro, alteração, exclusão, erros, etc.).
- Sempre solicite confirmação para ações irreversíveis, como exclusões.
- Gere relatórios sobre Clientes cadastrados em PDF (Jasper) e Excel (Apache POI).
- Implemente a funcionalidade para criar múltiplos Clientes a partir de uma planilha Excel. O usuário deverá colocar a planilha em um local designado, e o sistema deve realizar o cadastro ou fornecer explicações em caso de erro.
- Realize as validações necessárias, como evitar o cadastro de Clientes com o mesmo CPF/CNPJ e impedir alterações em dados sensíveis.
- Execute testes unitários com JUnit para todos os CRUDs.
- As mensagens de aviso devem aparecer em um local flutuante na tela e desaparecem automaticamente após um tempo.

8. Desafios Específicos

8.1 Desafio - Componentes

- Utilize Bootstrap e os componentes do Wicket como Panel, TextField, NumberTextField, RadioGroup, CheckBox, ListView, Botão Link, Botão Submit.
- Utilize **AjaxRequestTarget** para atualizar campos.
- Mostre, no botão submit, que os valores colocados no formulário são armazenados automaticamente no objeto Java.

8.2 Desafio Jasper

- Estude como gerar relatórios em Jasper.
- No CRUD já criado, adicione um botão para gerar relatórios ao lado de cada linha, que inclua todas as informações cadastradas e outro botão para gerar de todos os registros listados.

8.3 Desafio Excel

- Implemente a exportação de dados para Excel, similar à geração de PDF. Adicione um botão ao lado dos outros.
- Implemente a importação de dados via Excel.

8.4 Desafio de Persistência

- Utilize Hibernate (versão 5) para persistência.
- Estruture seu projeto em MVC.
- Foque na persistência e back-end, criando o mais genérico e fácil de refatorar possível.
- Os CRUDs devem incluir classes separadas para Serviço e DAO.
- Implemente relacionamentos OneToMany e ManyToOne, e utilize consultas com Search (com.googlecode.genericdao.search), HQL e SQL puro.

8.5 Desafio ListView e Modal Window

- Crie uma tela de listagem de dados com filtros.
- Utilize ListView/PageableListView/DataView/DefaultView.
- Adicione um botão para inclusão, que abra uma ModalWindow com os campos de cadastro.
- Ao fechar ou salvar, atualize a página de listagem automaticamente.
- Cada item da lista deve conter botões para editar e excluir, sendo que a exclusão deve solicitar confirmação.