



Universidade Federal de Uberlândia
Faculdade de Computação
API Rest com Java e Spring Boot
Conceitos abordados: Inversão de Controle, Injeção de Dependência,
Arquitetura em Camadas – Prof. Daniel A. Furtado
Atividade Individual

- 1) Baixe e instale o **IntelliJ IDEA** versão 2025.3.2+ (<https://www.jetbrains.com/idea/download>).
- 2) Baixe e instale o **MySQL Community** seguindo os passos no final do tutorial a seguir: <https://furtado.prof.ufu.br/site/teaching/PPI/apache-php-mysql-tutorial-instalacao.pdf>. Abra o MySQL Workbench e crie um novo banco de dados para o trabalho. Se estiver usando um computador do laboratório (sem permissão de instalação do MySQL), pode-se utilizar como alternativa o **H2 Database**, já integrado no Spring Boot.
- 3) Descompacte o projeto de exemplo (<https://furtado.prof.ufu.br/site/teaching/DW2/Intro-Rest-loC.zip>) e abra a pasta no IntelliJ. Se necessário, aguarde pelas atualizações. Se solicitado, autorize a configuração automática do OpenJDK.
- 4) Clique no botão "m" de Maven na barra lateral direita e posteriormente na seta de seleção do botão de atualização do Maven e escolha **Reload all Maven Projects** para que todas as dependências do projeto sejam baixadas e configuradas.
- 5) Acesse o arquivo de **propriedades** da aplicação (demo → src → main → resources → application.properties) e ajuste o **usuário** do MySQL, a **senha** e o **nome** do banco de dados criado anteriormente. Se não estiver usando o MySQL, comente todas as linhas relacionadas a esse sistema e descomente aquelas relacionadas ao H2 Database.

```
spring.application.name=demo
spring.h2.console.enabled=true

# Descomente para usar o banco de dados H2 (em memória) ao invés do MySQL
# spring.datasource.url=jdbc:h2:mem:testdb
# spring.jpa.hibernate.ddl-auto=update

# Conexão com o MySQL - Substitua dw2 na URL pelo nome do seu banco
spring.datasource.url=jdbc:mysql://localhost:3306/dw2?serverTimezone=UTC
spring.datasource.username=ppi
spring.datasource.password=ppi

# Driver do MySQL
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

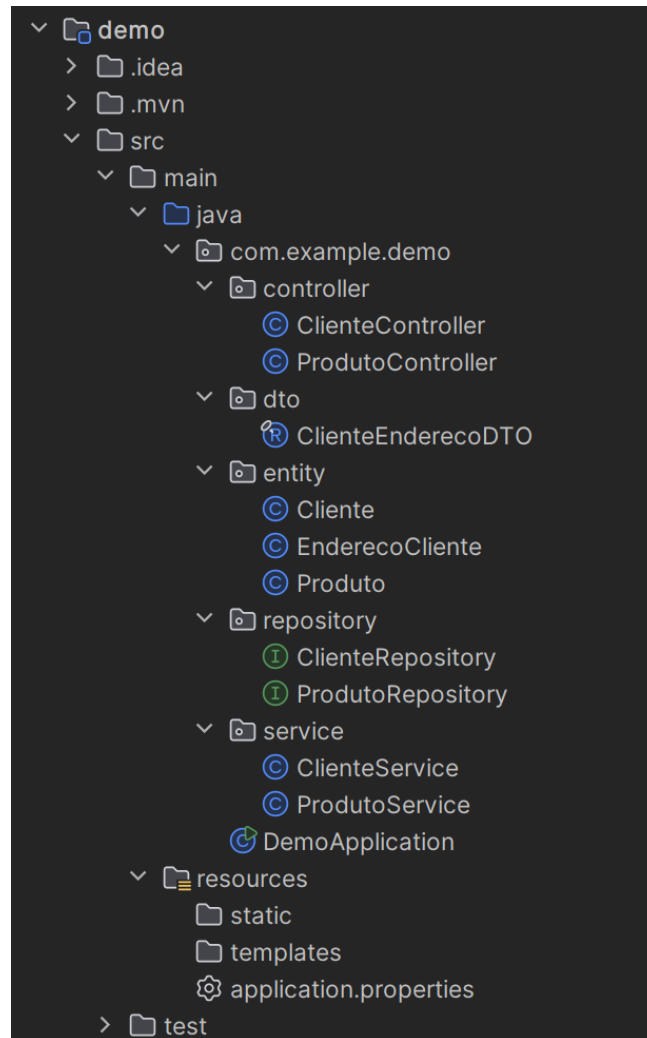
# O Hibernate irá criar as tabelas do MySQL automaticamente, caso não existam.
spring.jpa.hibernate.ddl-auto=update

# Exibe as instruções SQL executadas no console (útil para depuração)
spring.jpa.show-sql=true

# Mantem a conexão com o banco aberta até finalizar a requisição (view/JSON é gerado)
spring.jpa.open-in-view=true
```

- 6) Na estrutura do projeto (painel à esquerda), abra o arquivo **pom.xml** e observe as dependências do projeto.

- 7) Para acompanhar o estado do banco de dados MySQL diretamente no IntelliJ, clique no ícone de banco de dados no painel lateral direito, escolha New (+) → Data Source → MySQL e forneça os parâmetros de conexão conforme banco criado anteriormente.
- 8) Navegue pela estrutura de classes e interfaces da aplicação. Analise os arquivos correspondentes às múltiplas camadas da aplicação (**controllers, services, repositories**). Analise também os arquivos correspondentes às entidades **Cliente, EnderecoCliente** e **Produto**.



- 9) Execute a aplicação, abra o navegador e digite **localhost:8080/produtos** para acessar a lista de produtos (inicialmente vazia) da API. Abra um cliente HTTP (por exemplo, utilizando a extensão **Talend API Tester** no Chrome) e faça uma requisição **POST** para o endereço acima inserindo no corpo um objeto em JSON (ex. {"nome":"Produto1", "preco":"100"}). Observe o código de status (201) e o corpo da resposta HTTP. Cadastre mais um produto e registre um print da resposta HTTP (status e corpo) apresentada pelo cliente HTTP.
- 10) Acesse novamente **localhost:8080/produtos** no navegador e confira os produtos cadastrados. Busque um produto específico: **localhost:8080/produtos/codigoDoProduto**.
- 11) Verifique as tabelas do banco de dados utilizando o ambiente do próprio IntelliJ (configurado no passo 7).
- 12) Faça uma alteração no preço do segundo produto utilizando o método PUT no cliente HTTP. Se necessário, analise o código Java para descobrir como realizar a alteração.
- 13) Realize uma busca de produto pelo **nome** utilizando a barra de endereços do navegador. Faça também uma busca por todos os produtos com preço inferior a um valor fornecido. Tais funcionalidades já estão disponíveis na API. Analise os códigos correspondentes.

- 14) Exclua o segundo produto utilizando o método DELETE no cliente HTTP. Explique detalhadamente o fluxo de execução da operação dentro da API, incluindo desde a recepção da requisição no controlador até a exclusão de fato na tabela do banco de dados.
- 15) Verifique os comandos SQL executados automaticamente pelo **Hibernate** decorrentes das operações anteriores. Eles são apresentados na janela de console (Run Tool Window) do IntelliJ.
- 16) Utilize novamente o cliente HTTP para cadastrar dois **clientes** utilizando a API. Analise o código Java para descobrir como realizar esse cadastro. Confirme os dados cadastrados acessando **localhost:8080/clientes**. Explique o fluxo de execução da operação dentro da API, incluindo desde a recepção da requisição no controlador até a inserção dos dados nas tabelas do banco de dados.
- 17) Implemente a funcionalidade de exclusão de cliente pelo ID.
- 18) Implemente a funcionalidade de busca de cliente pelo CPF.
- 19) Adicione na API a capacidade de gerenciamento de **Fornecedores** (CRUD completo). Cada fornecedor deve ter um **Id**, um **nome** e um **cnpj**.
- 20) Crie uma página HTML de nome **index.html** na pasta **src/main/resources/static**. A página deve possuir um campo textual para digitação do código do fornecedor. Assim que o código for digitado, os dados do respectivo fornecedor devem aparecer “instantaneamente” logo abaixo (em campos ou em 's). Utilize a API Fetch com async/await.

Entrega

Compacte a pasta raiz do projeto em conjunto com as imagens (prints solicitados) e respostas dos exercícios (respostas.txt). Envie o arquivo compactado pelo Sistema Acadêmico de Aplicação de Testes (SAAT) até a data e hora limite indicada pelo professor.

Sobre Eventuais Plágios

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.