

# Documentação Técnica - CRUD para Gestão de Produtos em uma Ótica

## Introdução

Este documento descreve a implementação de um CRUD (Create, Read, Update, Delete) em Java para a gestão de produtos em uma loja virtual de ótica. A aplicação permitirá a adição, visualização, edição e remoção de produtos.

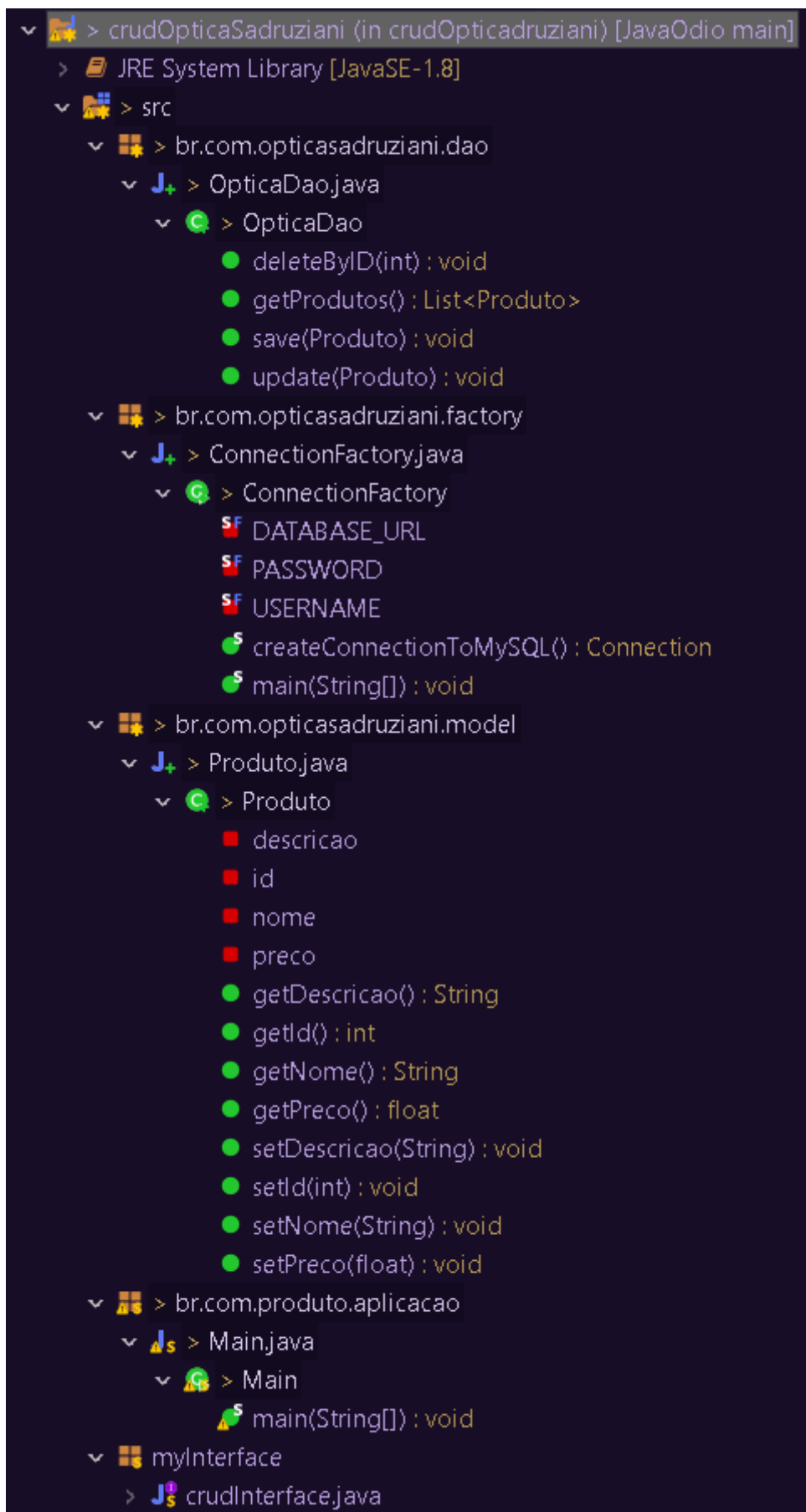
## Requisitos do Sistema

- JDK (Java Development Kit) instalado (versão recomendada: Java 8 ou superior).
- Ambiente de desenvolvimento Java (Eclipse, IntelliJ, etc.).
- Conhecimentos básicos de Java e orientação a objetos.
- SGDB (Sistema Gerenciador de Banco de Dados) instalado (DBeaver, SQL Server Management Studio, etc.).
- XAMPP

## Estrutura do Projeto

O projeto será organizado da seguinte forma:

```
crudOpticaSadruziani/  
└──src/  
    ├──br.com.opticasadruziani.dao/  
    │   ├──OpticaDao.java  
    ├──br.com.opticasadruziani.factory/  
    │   ├──ConnectionFactory.java  
    ├──br.com.opticasadruziani.model/  
    │   ├──Produto.java  
    ├──br.com.produto.aplicacao/  
    │   ├──Main.java  
    ├──myInterface/  
    └──crudInterface.java
```



## Descrição das Classes

### 1. OpticaDao.java

A classe `OpticaDao` é responsável por gerenciar a persistência dos produtos, utilizando da conexão com o banco de dados para manipular o mesmo. Nela estão inclusos os métodos para:

- Adicionar um novo produto.
- Buscar um produto por ID.
- Atualizar informações de um produto.
- Remover um produto.

## 2.1 Método *Save*

O método `Save` é utilizado para cadastrar um novo produto na loja na base de dados.

## 2.2 Método *getProduto*

É utilizado para recuperar os dados de um produto da base de dados através do seu ID e armazena-los em uma lista por onde nos é permitido visualizar as informações de um produto.

## 2.3 Método *update*

O método *update* possibilita recuperar os dados de um produto da base de dados através do seu ID e realizar edição das informações deste produto existente.

## 2.4 Método *delete*

Este método permite a remoção de um produto da loja realizando uma busca na base de dados através do seu ID.

## 2. ConnectionFactory.java

A classe `ConnectionFactory` é responsável por estabelecer a conexão com o banco de dados MySQL. Ela utiliza a biblioteca JDBC para facilitar a comunicação com o banco de dados.

Nessa classe estão sendo armazenados o `login`, a `senha` e a `URL` que serão utilizados para estar criando a conexão com o banco de dados da loja, além do método *createConnectionToMySQL* que é responsável por realizar esta conexão.

## 3. Produto.java

A classe `Produto` representa um item disponível na loja. Ela possui os seguintes atributos:

- `int id` - Identificador único do produto.
- `String nome` - Nome do produto.
- `String descricao` - Descrição do produto.
- `float preco` - Preço do produto.

## 4. Main.java

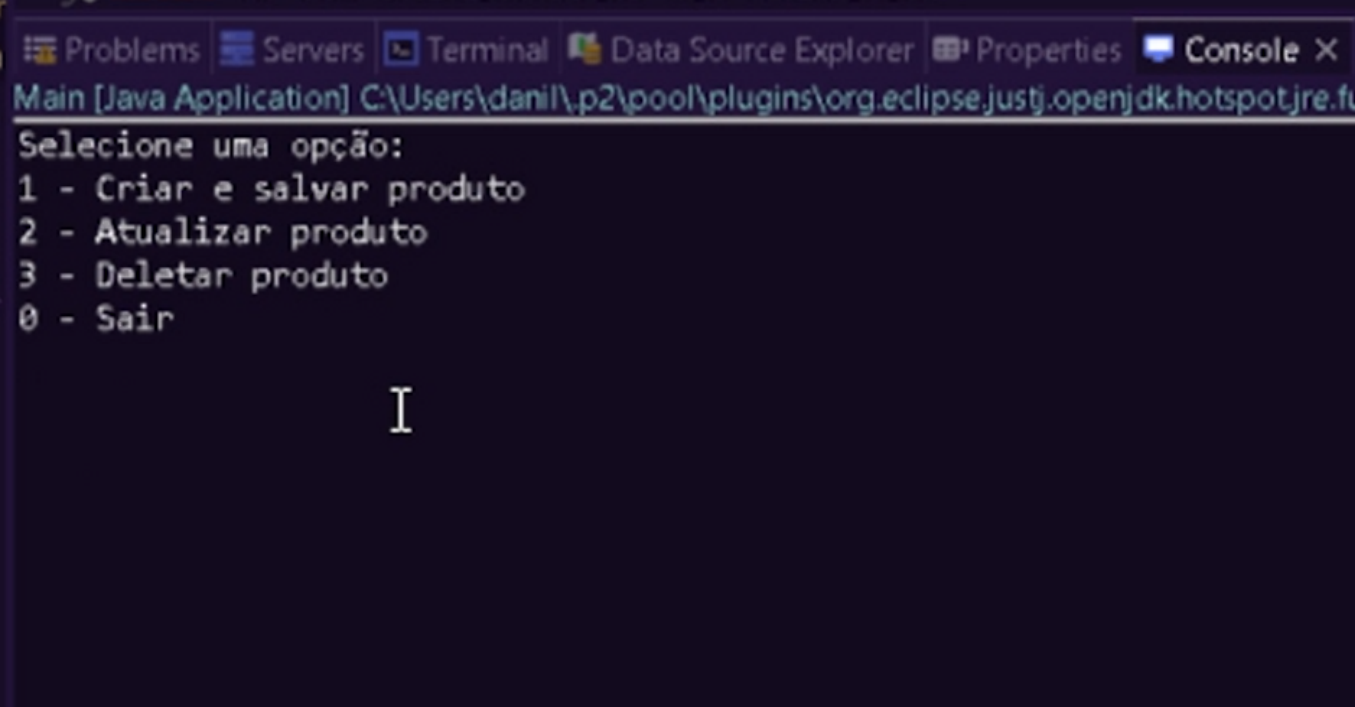
A classe principal da aplicação pode ser chamada de `Main.java`. Esta classe contém o método `main` onde a execução do programa é iniciada.

Aqui, podemos criar instâncias das classes de gestão de produtos e interagir com o sistema.

## Execução

Ao iniciar o programa será perguntado ao usuário qual ação ele deseja realizar, exibindo uma lista de operações que podem estar sendo executadas, como:

- Criar e salvar produto
- Atualizar produto
- Deletar produto
- Sair



Caso o usuário decida pela primeira ação, para *Criar* um novo produto, será então solicitado o nome, a descrição e o preço do novo produto que será adicionado ao banco de dados:



Caso o usuário decida por *Atualizar* um produto existente, primeiro será solicitado o ID do produto, e posteriormente será solicitado o novo nome, a nova descrição e o novo preço que será imposto ao produto:

```
Selecione uma opção:
1 - Criar e salvar produto
2 - Atualizar produto
3 - Deletar produto
0 - Sair
2
Informe o ID do produto a ser alterado:
3
Nome do produto:
Novo oculos de teste
Descrição do produto:
Descrição para o oculos
Preço do produto:
550
Salvo
```

Caso o usuário decida *Deletar* um produto, então será solicitado para que informe o ID do produto que deseja excluir e então após informado a ação será realizada:

```
Problems Servers Terminal Data Source Explorer Properties Console X
Main [Java Application] C:\Users\daniil\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Selecione uma opção:
1 - Criar e salvar produto
2 - Atualizar produto
3 - Deletar produto
0 - Sair
3
Informe o ID do produto que deseja deletar:
3
Deletado
```

E por fim, ao selecionar a opção *Sair* a execução do programa será encerrada.

## Conclusão

Esta documentação oferece uma visão abrangente da implementação de um sistema CRUD em Java, destinado à gestão de produtos em uma loja virtual de ótica. Cada classe é cuidadosamente projetada para desempenhar um papel específico, alinhando-se de forma coesa com os princípios fundamentais da programação orientada a objetos.

Ao possibilitar operações cruciais para a administração de produtos, esta aplicação proporciona uma base robusta, facilitando a integração de funcionalidades adicionais e permitindo a expansão do sistema de acordo

com as necessidades evolutivas da loja virtual. Com essa estrutura bem definida, os desenvolvedores têm a flexibilidade necessária para incorporar novos recursos e adaptar o sistema conforme as demandas do mercado e dos usuários.