



DESENVOLVIMENTO DE SISTEMAS

FLARRIELE SANTOS DE JESUS

JÚLIA DOS SANTOS CONCEIÇÃO

**Desenvolvimento de um CRUD em Java Utilizando Hibernate e MySQL**

*Relatório de Atividade Prática*

Camaçari

2025

FLARRIELE SANTOS DE JESUS  
JÚLIA DOS SANTOS CONCEIÇÃO

## **Desenvolvimento de um CRUD em Java Utilizando Hibernate e MySQL**

*Relatório de Atividade Prática*

Trabalho final elaborado para a disciplina Programação de Aplicativos, do curso de Desenvolvimento de Sistemas, ofertado pelo SENAI Camaçari, como parte dos requisitos para a obtenção da nota final do semestre.

Docente: Westn Melo de Souza

Camaçari

2025

## **1 ESTRUTURA DO SISTEMA**

O sistema foi desenvolvido em Java com o controle de dependências do Maven. Foi adicionado o Hibernate para interação com o banco de dados MySQL.

Foi criada uma interface para interação com o usuário via terminal na classe Main. Essa classe conhece as opções do sistema, exibindo ao usuário, e os métodos negociais que serão chamados a partir da escolha do usuário. A tabela Clientes do banco de dados é representada pela classe Cliente, que contém as anotações para identificá-la como entidade e para mapear as colunas com os atributos. A classe ClienteDAO é responsável pela interação com o banco de dados, contendo os métodos de CRUD da entidade Cliente.

O sistema é inicializado na classe Main. É exibido o menu ao usuário e ele poderá tomar ações de cadastro, leitura, atualização e exclusão de dados na tabela Clientes. Sempre que uma ação é concluída o programa retorna ao menu inicial, até que o usuário escolha por finalizar o sistema.

## **2 PASSOS PARA CONFIGURAR O HIBERNATE**

A configuração do Hibernate no projeto começou com a adição das dependências necessárias no arquivo pom.xml, incluindo o hibernate-core para o mapeamento objeto-relacional, o mysql-connector-java como driver JDBC, e o slf4j-simple para o gerenciamento de logs. Em seguida, foi configurado o arquivo hibernate.cfg.xml na pasta src/main/resources, este arquivo centraliza as informações de conexão: URL, usuário, senha, driver, e outras opções. Nele também foi especificada a estratégia hbm2ddl.auto como update durante a fase de desenvolvimento, permitindo que o Hibernate criasse ou atualizasse as tabelas automaticamente com base nas entidades. Para cada entidade, como a classe Cliente, foi necessário adicionar a tag <mapping class="..." /> neste arquivo. Por fim, para gerenciar a SessionFactory do Hibernate, a classe utilitária HibernateUtil.java foi desenvolvida, garantindo que as sessões com o banco de dados fossem abertas e fechadas de forma eficiente e segura.

### **3 PRINCIPAIS APRENDIZADOS E DIFICULDADES**

No desenvolvimento do projeto tivemos que aprender conceitos de configuração do ambiente Java: o controle das dependências com Maven, a instalação e definição no Eclipse do Java e do Maven, a adição de uma dependência no pom.xml e a interação com o banco de dados através do aplicativo Java.

Aplicar conceitos novos em um sistema desenvolvido do zero foi difícil. A dificuldade em formular as pesquisas e encontrar as soluções exatas para os desafios que surgiam foi uma constante, exigindo persistência e uma abordagem iterativa de tentativa e erro.