



CURSO: DESENVOLVIMENTO DE SISTEMAS

FERNANDA ALINE PEREIRA SANTOS

**DESENVOLVIMENTO DE UM SISTEMA CRUD COM JAVA, HIBERNATE
E MYSQL**

DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS

Camaçari – BA

27 Junho de 2025

FERNANDA ALINE PEREIRA SANTOS

**DESENVOLVIMENTO DE UM SISTEMA CRUD COM JAVA, HIBERNATE
E MYSQL**

DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS

Trabalho apresentado no curso de desenvolvimento de sistemas, da disciplina: PROGRAMAÇÃO ORIENTADA A OBJETOS, na instituição SENAI-BA/Camacari, sob orientação do professor Westn Melo de Souza, como trabalho avaliativo da disciplina em questão.

Camaçari – BA

27 Junho de 2025

1. Introdução

A evolução constante das tecnologias de desenvolvimento de software tem incentivado a utilização de frameworks e ferramentas que otimizam a criação de aplicações eficientes e robustas. Neste contexto, a presente atividade prática teve como objetivo a construção de um sistema CRUD (Create, Read, Update, Delete) utilizando a linguagem Java, o framework de persistência Hibernate e o banco de dados relacional MySQL.

O projeto simula o gerenciamento de clientes de forma simplificada, integrando os conceitos de orientação a objetos, mapeamento objeto-relacional e persistência de dados.

A atividade também buscou reforçar habilidades como organização de projetos com Maven, uso de boas práticas em camadas e controle de versão com Git.

2. Desenvolvimento

O desenvolvimento do sistema foi realizado no ambiente Eclipse, utilizando a estrutura de projeto Maven para organização das dependências. Inicialmente, foi criada a entidade `Cliente`, responsável por representar os dados dos clientes armazenados no banco de dados. Essa classe foi anotada com as anotações do JPA, como `@Entity`, `@Table`, `@Id` e `@GeneratedValue`, o que permitiu ao Hibernate realizar o mapeamento automático para a tabela correspondente.

Em seguida, foi implementada a classe `ClienteDAO`, responsável pelas operações de persistência. Essa classe encapsula os métodos salvar, listar, atualizar e excluir, utilizando transações controladas pela `Session` do Hibernate. A conexão com o banco foi configurada por meio da classe `HibernateUtil`, que instancia um `SessionFactory` a partir de um arquivo `hibernate.cfg.xml`, no qual foram especificadas as propriedades de acesso ao banco `sistema_clientes` e o dialeto correspondente ao MySQL 8.

Passos para configurar o Hibernate:

- ✚ Criação do arquivo `hibernate.cfg.xml` contendo os dados de conexão com o MySQL.
- ✚ Adição das dependências no `pom.xml` para Hibernate, JPA e MySQL.
- ✚ Implementação da classe `HibernateUtil` com a `SessionFactory`.

O arquivo hibernate.cfg.xml foi alocado na pasta src/main/resources, com as propriedades de conexão, dialeto, controle de esquema (hbm2ddl.auto = update) e o mapeamento da classe Cliente. O banco de dados foi criado no MySQL Workbench, por meio do comando CREATE DATABASE sistema_clientes, e, ao executar o sistema, o Hibernate gerou automaticamente a tabela cliente.

A interface do usuário foi construída em modo console, por meio da classe Main, que apresenta um menu com opções de cadastro, listagem, atualização e exclusão de clientes. O funcionamento foi validado por meio de testes com o Workbench, utilizando comandos SQL como SELECT * FROM cliente; para verificar os registros persistidos com sucesso.

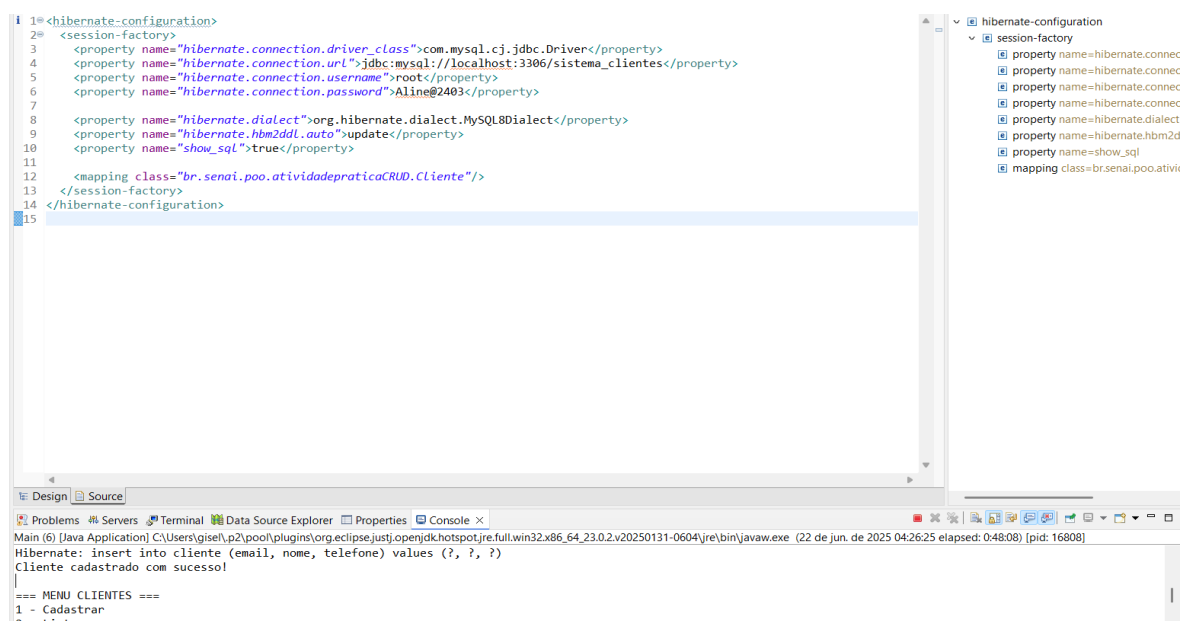
2.1 Integração com o Banco de Dados

No MySQL Workbench, foi criado o banco sistema_clientes com o comando:

CREATE DATABASE sistema_clientes;

Durante a execução da aplicação, a conexão foi bem-sucedida, e a tabela foi gerada automaticamente, contendo os campos id, nome, email e telefone.

Ao executar a classe Main.java, o sistema exibiu corretamente o menu de opções e permitiu cadastrar um cliente, visualizar os registros, editar dados e excluir entradas. Após o cadastro, foi feito um SELECT * FROM cliente; no Workbench, exibindo com sucesso os dados persistidos.



The screenshot displays an IDE with two main components. On the left, the 'Source' editor shows the content of hibernate.cfg.xml, which is an XML configuration file for Hibernate. It includes properties for the database connection (driver, URL, username, password), the dialect (MySQL8Dialect), and the hbm2ddl.auto setting (update). It also shows a mapping for the 'br.senai.poo.atividadepraticaCRUD.Cliente' class. On the right, the 'Console' window shows the output of the application. It starts with 'Hibernate: insert into cliente (email, nome, telefone) values (?, ?, ?)' and 'Cliente cadastrado com sucesso!'. Below this, a menu is displayed: '=== MENU CLIENTES ===' followed by '1 - Cadastrar'.

```
1<hibernate-configuration>
2<session-factory>
3  <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
4  <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/sistema_clientes</property>
5  <property name="hibernate.connection.username">root</property>
6  <property name="hibernate.connection.password">Aline@2403</property>
7
8  <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>
9  <property name="hibernate.hbm2ddl.auto">update</property>
10 <property name="show_sql">true</property>
11
12 <mapping class="br.senai.poo.atividadepraticaCRUD.Cliente"/>
13</session-factory>
14</hibernate-configuration>
15
```

```
Main (6) [Java Application] C:\Users\gise\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_23.0.2.v20250131-0604\jre\bin\javaw.exe (22 de jun. de 2025 04:26:25 elapsed: 0:48:08) [pid: 16808]
Hibernate: insert into cliente (email, nome, telefone) values (?, ?, ?)
Cliente cadastrado com sucesso!

=== MENU CLIENTES ===
1 - Cadastrar
```

Figura 1 - Saída no console após cadastro de cliente com sucesso via menu interativo

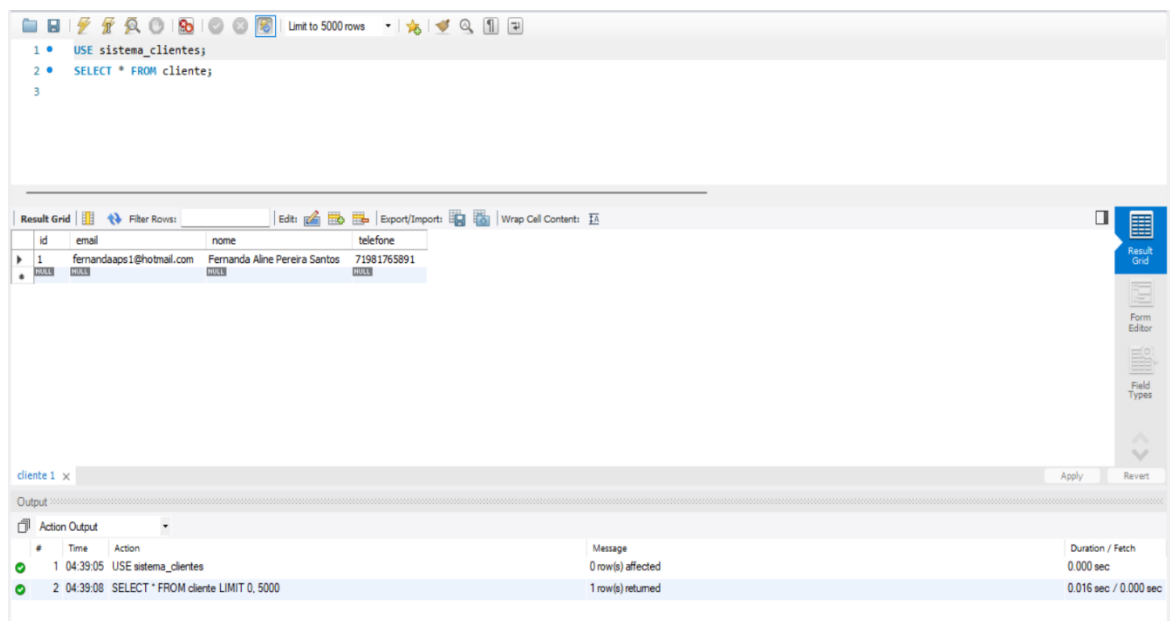


Figura 2 - Visualização dos dados persistidos exibidos no MySQL Workbench após cadastro

3. Testes práticos do sistema

As funcionalidades principais foram testadas com êxito por meio de um menu interativo via console, conforme demonstra a Figura 1.

Foram realizados os seguintes testes:

Funcionalidade	Ação Realizada	Resultado
Cadastro	Inserido cliente “Maria Jose de Souza ”	Registro saldo com sucesso (insert into cliente...)
Listagem	Exibida lista de clientes	Dois clientes listados corretamente
Atualização	Cliente com ID 2 atualizado	Dados alterados com sucesso (update cliente set...)
Exclusão	Cliente com ID 2 removido	Registro excluído (delete from cliente where id=?)

Tabela 1 - Funcionalidades testadas e resultados obtidos no sistema CRUD

Obs: Já havia um cliente cadastrado no primeiro teste (cliente Fernanda, ID =1)

As sequência de imagens abaixo mostram o resultado de cada teste realizado referente a cada funcionalidade, os testes foram feitos via console, e atualizados no MySQL workbench:

3.1 Cadastro e Listagem

Inserção do cliente ID = 2, e exibição da lista dos 02 clientes cadastrados:

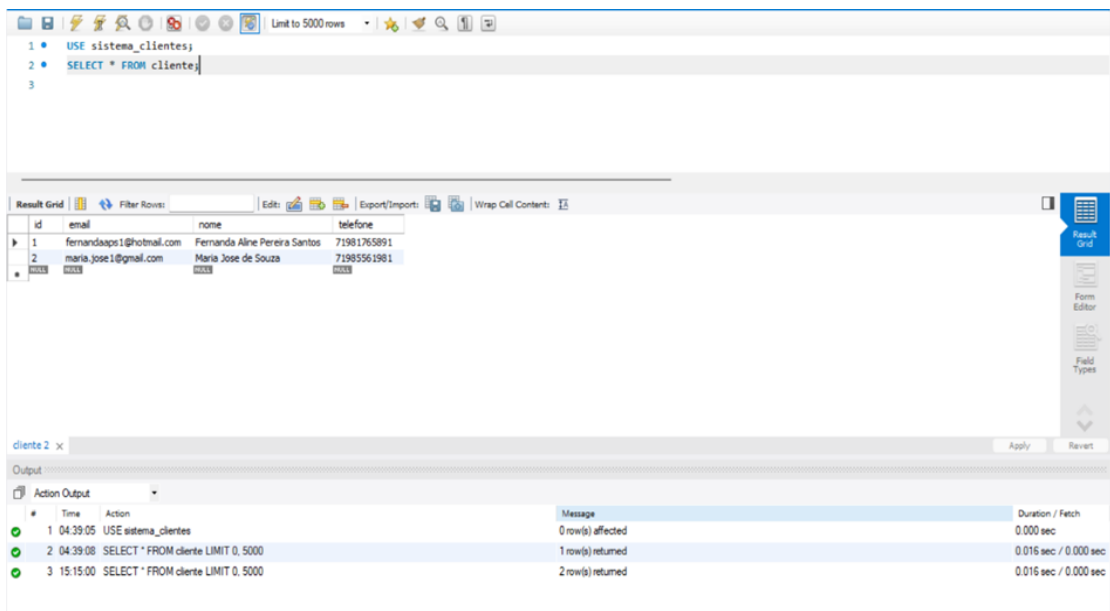


Figura 3 - Resultado após cadastro do cliente com ID = 2

3.2 Atualização

Alteração do nome do cliente com ID = 2:

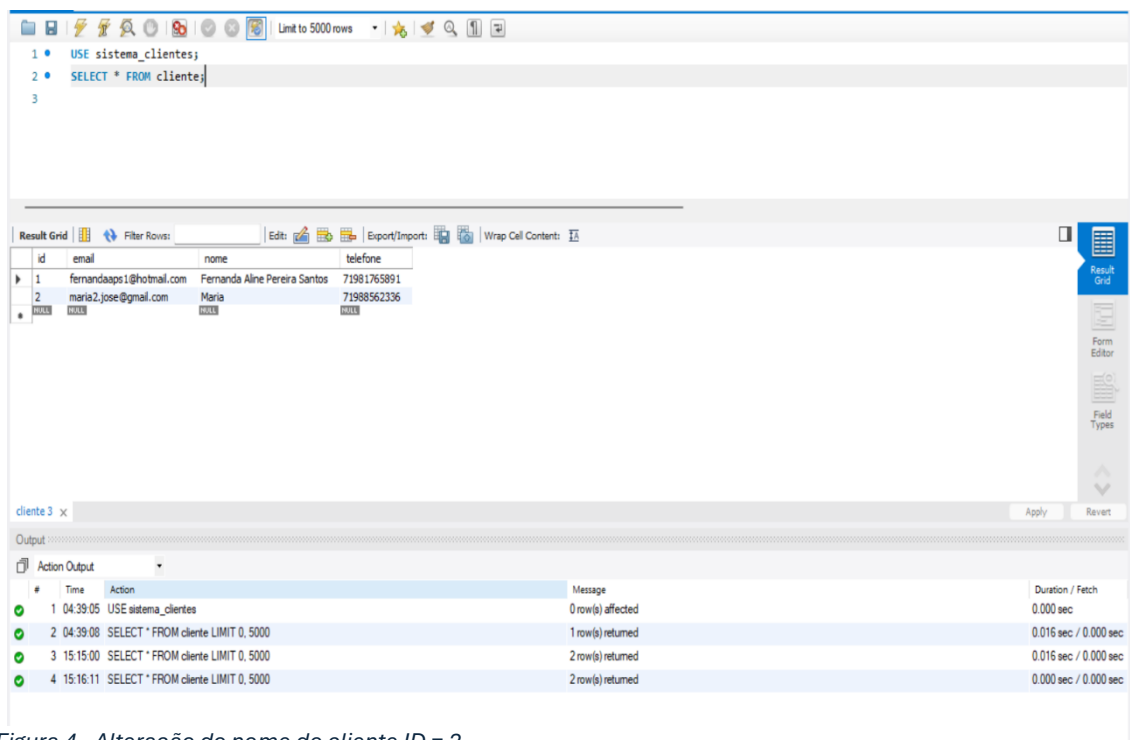


Figura 4 - Alteração do nome do cliente ID = 2

3.3 Exclusão

Exclusão do cliente com ID = 2:

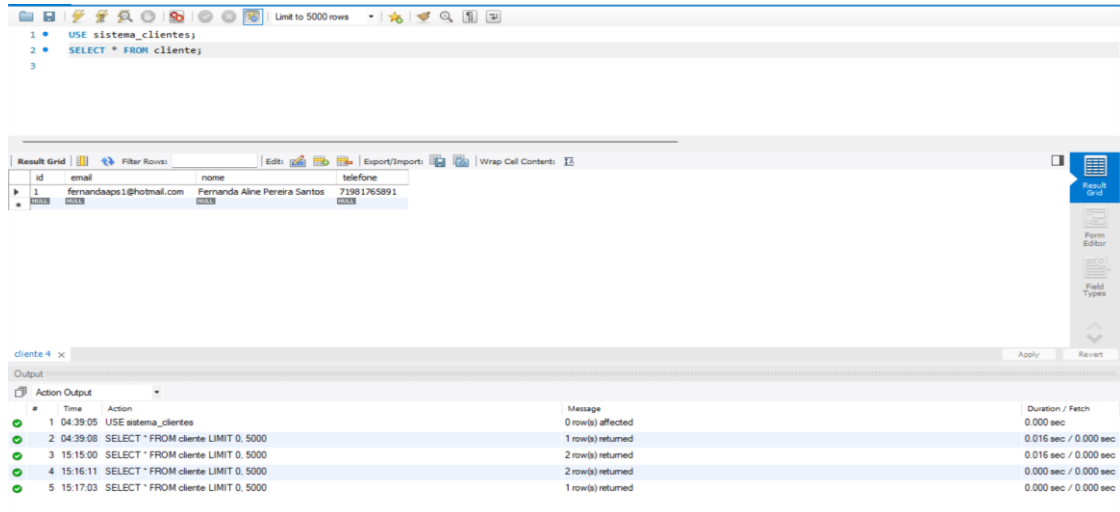


Figura 5 - Resultado após exclusão do cliente ID = 2

3.4 Histórico completo dos testes realizados exibidos no console:

=== MENU CLIENTES ===

1 - Cadastrar

2 - Listar

3 - Atualizar

4 - Excluir

0 - Sair

Escolha uma opção: 1

Nome: Maria Jose de Souza

Email: maria.jose1@gmail.com

Telefone: 71985561981

jun. 23, 2025 3:13:39 PM org.hibernate.Version logVersion

INFO: HHH000412: Hibernate ORM core version 5.6.15.Final

jun. 23, 2025 3:13:40 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager

<clinit>

INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}

jun. 23, 2025 3:13:40 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure

WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)

jun. 23, 2025 3:13:40 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL

[jdbc:mysql://localhost:3306/sistema_clientes]

jun. 23, 2025 3:13:40 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001001: Connection properties: {user=root, password=****}

jun. 23, 2025 3:13:40 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001003: Autocommit mode: false

jun. 23, 2025 3:13:40 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl\$PooledConnections <init>

INFO: HHH000115: Hibernate connection pool size: 20 (min=1)

jun. 23, 2025 3:13:41 PM org.hibernate.dialect.Dialect <init>

INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect

jun. 23, 2025 3:13:42 PM

org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl
getIsolatedConnection

INFO: HHH10001501: Connection obtained from JdbcConnectionAccess

[org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator\$ConnectionProviderJdbcConnectionAccess@177c41d7] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.

jun. 23, 2025 3:13:42 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator
initiateService

INFO: HHH000490: Using JtaPlatform implementation:
[org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

Hibernate: insert into cliente (email, nome, telefone) values (?, ?, ?)

Cliente cadastrado com sucesso!

=== MENU CLIENTES ===

1 - Cadastrar

2 - Listar

3 - Atualizar

4 - Excluir

0 - Sair

Escolha uma opção: 2

=== LISTA DE CLIENTES ===

Hibernate: select cliente0_.id as id1_0_, cliente0_.email as email2_0_, cliente0_.nome as nome3_0_, cliente0_.telefone as telefone4_0_ from cliente cliente0_

Cliente [id=1, nome=Fernanda Aline Pereira Santos, email=fernandaaps1@hotmail.com, telefone=71981765891]

Cliente [id=2, nome=Maria Jose de Souza, email=maria.jose1@gmail.com, telefone=71985561981]

=== MENU CLIENTES ===

1 - Cadastrar

2 - Listar

3 - Atualizar

4 - Excluir

0 - Sair

Escolha uma opção: 3

ID do cliente a atualizar: 2

Novo nome: Maria

Novo email: maria2.jose@gmail.com

Novo telefone: 71988562336

Hibernate: update cliente set email=?, nome=?, telefone=? where id=?

Cliente atualizado!

=== MENU CLIENTES ===

1 - Cadastrar

2 - Listar

3 - Atualizar

4 - Excluir

0 - Sair

Escolha uma opção: 4

ID do cliente a excluir: 2

Hibernate: delete from cliente where id=?

Cliente removido!

4. Uso do Git e GitHub no projeto

Durante o desenvolvimento do sistema de clientes em Java com Hibernate, utilizei a ferramenta de controle de versão **Git**, em conjunto com a plataforma **GitHub**, para realizar o versionamento do código-fonte.

O repositório foi criado remotamente no GitHub e vinculado ao projeto local através do Eclipse. Após configurar o repositório remoto e autenticação via token pessoal, realizei o **commit inicial** e utilizei o recurso de **Push** para publicar os arquivos diretamente pela IDE.

Esse processo facilitou o acompanhamento de mudanças, organização dos arquivos e garantia de backup do projeto.

Repositório público no GitHub: <https://github.com/CodeWithFernanda/sistema-clientes-hibernate.git>

Arquivos incluídos no primeiro commit:

NOME DO ARQUIVO	DESCRIÇÃO
.classpath	Configuração da estrutura de classes do eclipse
.gitignore	Listade arquivos/pastas ignoradas pelo Git
.project	Identificação e metadados do projeto no Eclipse
Cliente.java	Classe principal do modelo de cliente
ClienteDAO.java	Classe de aceso aos dados do cliente (DAO)
HibernateUtil.java	Classe utilitária para configurar o Hibernate
Main.java	Classe que executa/testa o sistema
Hibernate.cfg.xml	Arquivo de configuração do Hibernate
Pom.xml	Gerenciador de dependências e build do Maven

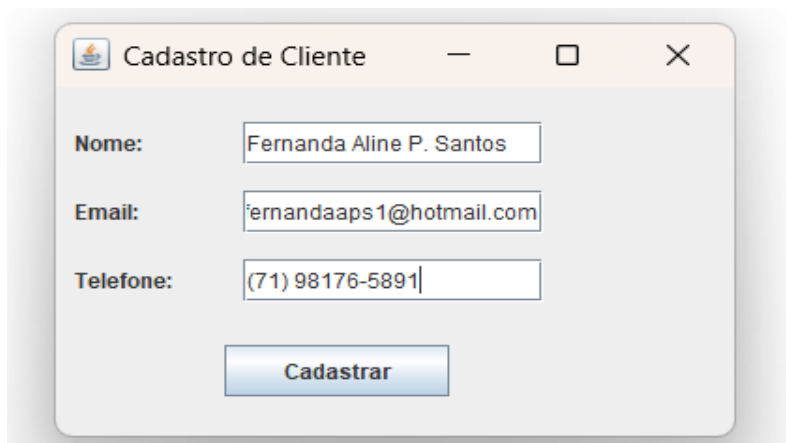
Tabela 2 - Arquivos commit

5. Desenvolvimento da Interface Gráfica (Front-End)

Após a validação das funcionalidades de persistência e regras de negócio no modo console, foi implementada uma interface gráfica utilizando a biblioteca **Java Swing**, visando tornar o sistema mais intuitivo e acessível.

5.1 Tela de Cadastro de Clientes

Essa tela permite inserir os dados de novos clientes (nome, e-mail e telefone) por meio de campos interativos e um botão "Cadastrar". Ao clicar, os dados são enviados ao banco de dados utilizando os métodos da classe ClienteDAO.




A janela "Cadastro de Cliente" possui um ícone de documento no canto superior esquerdo e botões de minimizar, maximizar e fechar no canto superior direito. Abaixo, há três campos de texto rotulados "Nome:", "Email:" e "Telefone:". O campo "Nome:" contém o texto "Fernanda Aline P. Santos". O campo "Email:" contém "fernandaaps1@hotmail.com". O campo "Telefone:" contém "(71) 98176-5891". Abaixo dos campos, há um botão azul com o texto "Cadastrar".

Figura 6 - Tela de cadastro

5.2 Tela de Listagem de Clientes

A segunda tela apresenta todos os clientes cadastrados em uma tabela (JTable), com carregamento automático dos dados ao abrir a interface. Essa funcionalidade facilita a visualização e conferência dos registros armazenados.



A janela "Lista de Clientes" possui um ícone de lista no canto superior esquerdo e botões de minimizar, maximizar e fechar no canto superior direito. Abaixo, há uma tabela com quatro colunas: "ID", "Nome", "Email" e "Telefone".

ID	Nome	Email	Telefone
1	Fernanda Aline Pereir..	fernandaaps1@hotm...	(71) 98176-5891
3	Luzinete Sampaio	luzinetes@hotmail.co..	(75) 93636-2500
4	Ana de Souza	ana.souza@hotmail.c..	(75) 99963-2175

Figura 7- Tabela com clientes listados

5.3 Tela de Edição e Exclusão

Essa interface exibe a lista de clientes e permite selecionar uma linha para editar ou excluir o registro. Os dados selecionados são carregados nos campos para alteração e atualização posterior via botão "Atualizar", ou remoção definitiva com o botão "Excluir".

Editar / Excluir Cliente

ID	Nome	Email	Telefone
1	Fernanda Aline Pereira ...	fernandaaps1@hotmail...	(71) 98176-5891
3	Luzinete Sampaio	luzinetes@hotmail.com	(75) 93636-2500
4	Ana de Souza	ana.souza@hotmail.com	(75) 99963-2175

Nome:

Email:

Telefone:

Atualizar

Excluir

Figura 8- Seleção de cliente para edição

Editar / Excluir Cliente

ID	Nome	Email	Telefone
1	Fernanda Aline Pereira ...	fernandaaps1@hotmail...	(71) 98176-5891
3	Luzinete Sampaio	luzinetes@hotmail.com	(75) 93636-2500
4	Ana de Souza	ana.souza@hotmail.com	(75) 99963-2175

Nome:

Email:

Telefone:

Atualizar

Excluir

Mensagem

Cliente atualizado com sucesso!

OK

Figura 10 - Confirmação da edição dos dados do cliente

Editar / Excluir Cliente

ID	Nome	Email	Telefone
1	Fernanda Aline Pereira Santos	fernandaaps1@hotmail.com	(71) 98176-5891
3	Luzinete Sampaio Gomes	luzinetes@hotmail.com	(75) 93636-2500
4	Ana de Souza	ana.souza@hotmail.com	(75) 99963-2175

Nome:

Email:

Telefone:

Atualizar

Excluir

Figura 9 - Evidência da alteração realizada

Editar / Excluir Cliente

ID	Nome	Email	Telefone
1	Fernanda Aline Pereira Santos	fernandaaps1@hotmail.com	(71) 98176-5891
3	Luzinete Sampaio Gomes	luzinetes@hotmail.com	(75) 93636-2500
4	Ana de Souza	ana.souza@hotmail.com	(75) 99963-2175

Nome:

Email:

Telefone:

Atualizar

Excluir

Figura 11- Seleção de cliente a ser excluído

ID	Nome	Email	Telefone
1	Fernanda Aline Pereira Santos	fernandaaps1@hotmail.com	(71) 98176-5891
3	Luzinete Sampaio Gomes	luzinetes@hotmail.com	(75) 93636-2500
4	Ana de Souza	ana.souza@hotmail.com	(75) 99963-2175

Nome:

Email:

Telefone:

Excluir

Confirmação

?

Tem certeza que deseja excluir este cliente?

Sim **Não**

Figura 12 – Confirmação do sistema

ID	Nome	Email	Telefone
4	Ana de Souza	ana.souza@hotmail.com	(75) 99963-2175

Nome:

Email:

Excluir

Mensagem

i

Cliente excluído com sucesso!

OK

Figura 13 - Confirmação de exclusão

ID	Nome	Email	Telefone
1	Fernanda Aline Pereira Santos	fernandaaps1@hotmail.com	(71) 98176-5891
3	Luzinete Sampaio Gomes	luzinetes@hotmail.com	(75) 93636-2500

Nome:

Email:

Telefone:

Atualizar

Excluir

Figura 14 - Evidência que a exclusão foi realizada

5.4 Tela de Menu Principal

Por fim, foi criada uma tela central que funciona como o ponto de entrada do sistema. Ela apresenta três botões: "Cadastrar Cliente", "Listar Clientes" e "Editar/Excluir Cliente", organizando as funcionalidades de forma clara e eficiente.

Sistema de Gerenciamento de Clientes

MENU PRINCIPAL

Cadastrar Cliente

Listar Clientes

Editar / Excluir Cliente

Figura 15 - Tela do menu principal com botões de navegação entre as telas

5.5 Commit de marco do front-end

Após a implementação das telas gráficas com Java Swing, realizei um novo commit identificando o marco de conclusão do front-end do sistema.

A mensagem utilizada seguiu o padrão de boas práticas com Conventional Commits, facilitando a leitura e o rastreo futuro: > > feat: conclusão do front-end até menu principal > > Implementadas as telas de cadastro, listagem e edição/exclusão de clientes, com navegação através do menu principal. > > .

Esse commit foi realizado pela IDE Eclipse, através da aba Git Staging, com os arquivos devidamente organizados e selecionados para versionamento. A ação foi seguida por um push para o repositório remoto no GitHub, garantindo a sincronização do projeto na nuvem e o registro formal da evolução.

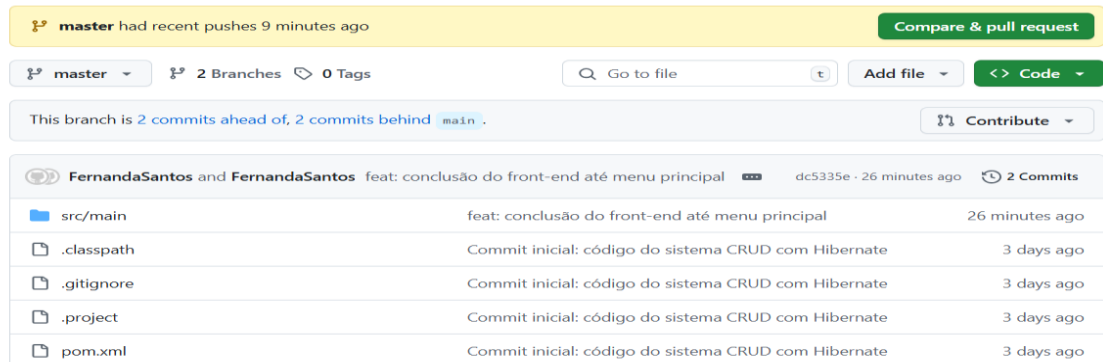


Figura 16 - Histórico de commits atualizado, pós inclusão do front-end.

6. Portabilidade e Configuração de Ambiente

O sistema foi desenvolvido com atenção à portabilidade, permitindo execução em diferentes computadores sem necessidade de alteração no código-fonte. Para isso, as informações sensíveis de conexão com o banco de dados (URL, usuário e senha) foram externalizadas por meio de um arquivo de configuração `dbconfig.properties`.
> > Essa abordagem permite que o sistema se adapte facilmente a diferentes ambientes (ex: MySQL ou MariaDB, com diferentes credenciais), bastando atualizar o conteúdo do arquivo `.properties` com os dados corretos do servidor local. Essa configuração é lida dinamicamente pela aplicação no momento da inicialização da conexão com o banco de dados.

7. Conclusão

A realização deste projeto proporcionou uma experiência prática significativa, consolidando conhecimentos essenciais sobre a integração entre programação orientada a objetos, persistência de dados com Hibernate e manipulação de banco de dados MySQL. A criação do sistema CRUD reforçou a compreensão do ciclo completo de desenvolvimento de aplicações, desde a modelagem das entidades até o armazenamento e recuperação dos dados.

Durante o processo, também houve uma imersão valiosa em práticas de versionamento com Git e GitHub, o que contribuiu para a organização do projeto e o acompanhamento da evolução do código. Dificuldades enfrentadas, como problemas iniciais de configuração de conexão e mapeamento entre as entidades, foram superadas com pesquisa e leitura de documentação, o que me ajudou ainda mais na compreensão do processo.

Um dos pontos de destaque foi a implementação de um mecanismo de configuração externa, por meio de um arquivo `dbconfig.properties`, que permite ao sistema ser executado em qualquer computador sem alterações no código-fonte. Essa adaptação representa uma aplicação prática de boas práticas de segurança e portabilidade, permitindo que o sistema se ajuste a diferentes ambientes de forma simples e eficiente.

O projeto final representa uma base sólida para desafios maiores no campo acadêmico e profissional, unindo teoria e prática com organização, clareza e atenção a detalhes importantes da engenharia de software.

8. Referencial Técnico

A seguir estão os principais materiais e documentações técnicas utilizados como referência durante a construção e configuração do sistema:

Documentação oficial do MySQL: <https://dev.mysql.com/doc/>

Documentação do Hibernate ORM: <https://hibernate.org/orm/documentation/7.0/>

Materiais didáticos da disciplina de Programação Orientada a Objetos – SENAI
Camaçari