



Título da prática:

Criação de aplicativo Java, com acesso ao banco de dados SQL Server através do middleware JDBC.

CAMPUS: Polo Planalto – BH – MG

DISCIPLINA: BackEnd sem banco não tem!

TURMA: 2025.1

SEMESTRE LETIVO: Primeiro Semestre (2025)

ALUNO: Bruno Ricardo Viana Venturelli

Matrícula: 202401226726

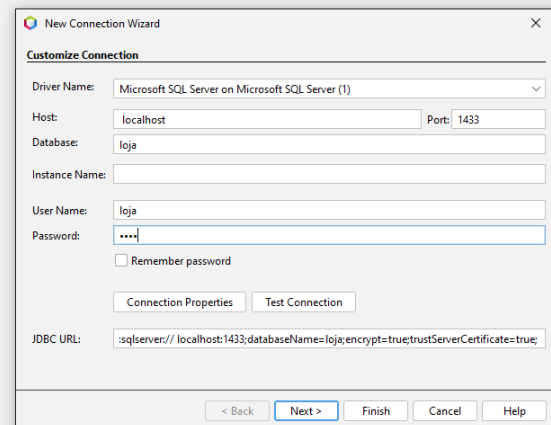
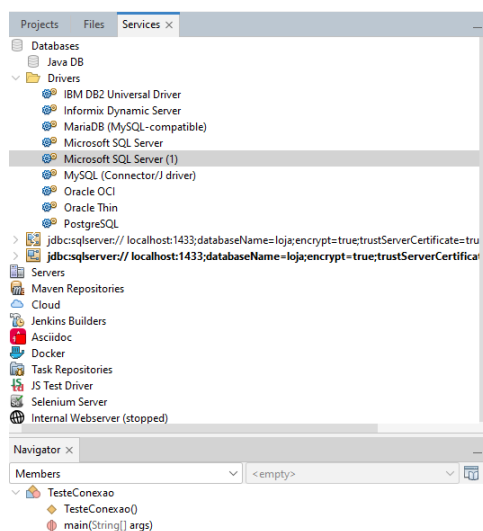
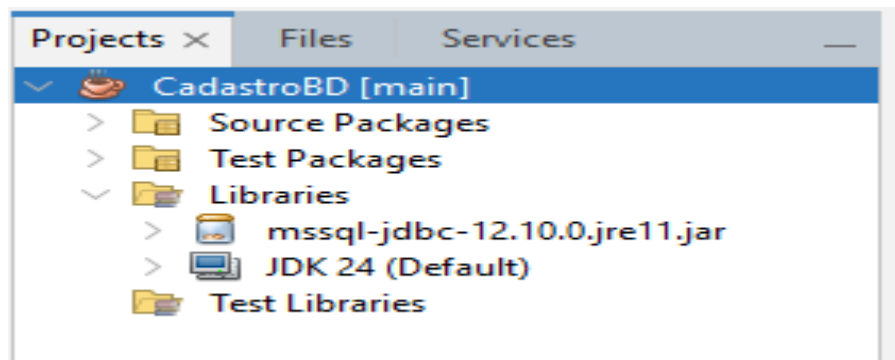
LINK DO MEU GITHUB:

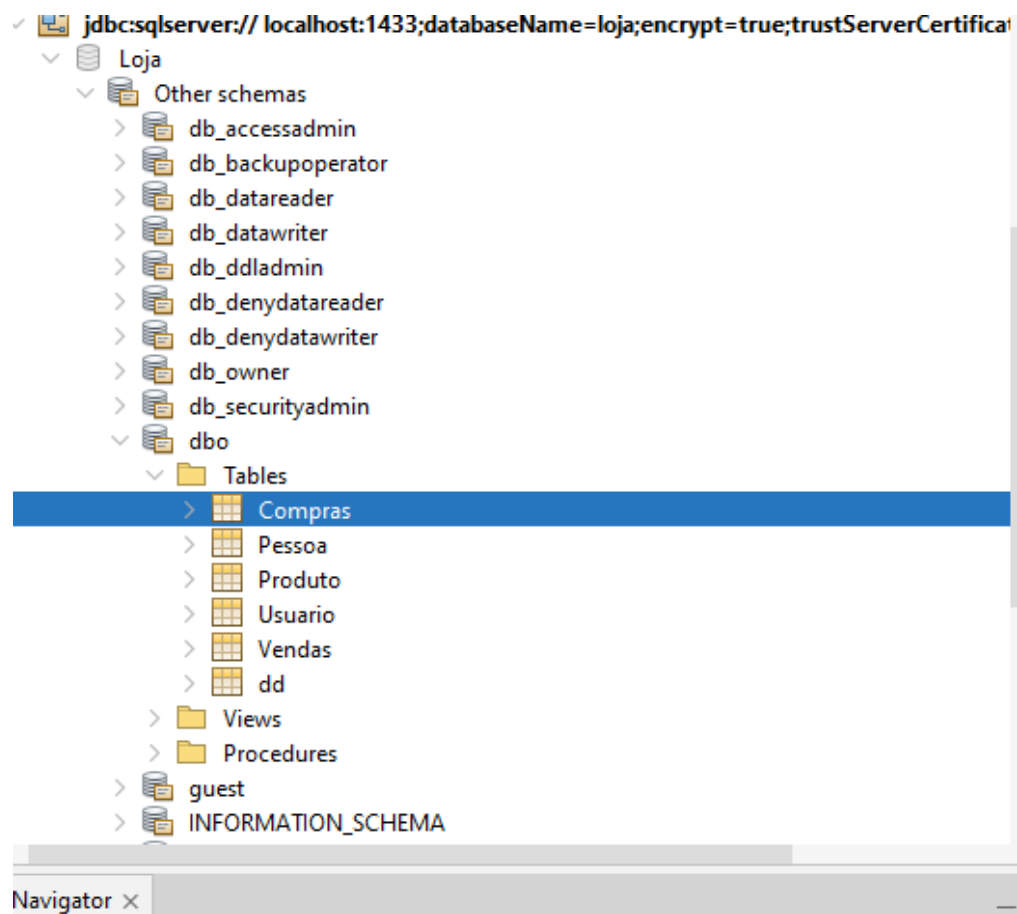
[https://github.com/DevBrN01/Trabalho_Pratico_M3_BackEnd_BancoD
ados](https://github.com/DevBrN01/Trabalho_Pratico_M3_BackEnd_BancoDados)

Objetivo da Prática

1. Implementar persistência com base no middleware JDBC.
2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
3. Implementar o mapeamento objeto-relacional em sistemas Java.
4. Criar sistemas cadastrais com persistência em banco relacional.

Criando projeto SQL Servers e NetBeans “Banco de Dados”





Views das tabelas “PRINT”:

Pesquisador de Objetos

Conectar

BRUNO-VENTURELLI\SQLEXPRESS (SQL Server 16.0.1000 - loja)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.Compras

dbo.dd

dbo.Pessoa

dbo.Produto

dbo.Usuario

dbo.Vendas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

Replicação

Gerenciamento

XEvent Profiler

SQLQuery4.sql - BRU...ESS.Loja (60)*

USE Loja;

SELECT * FROM dbo.Pessoa;

100 %

Resultados

Mensagens

	id	nome	logradouro	cidade	estado	telefone	email	cpf	cnpj
1	1	João da Silva	Rua das Flores, 100	São Paulo	SP	(11) 98765-4321	joao@email.com	123.456.789-00	NULL
2	2	Empresa X Ltda	Av. Brasil, 500	Rio de Janeiro	RJ	(21) 3333-4444	empresa@email.com	NULL	12.345.678/0001-90
3	3	João da Silva	Rua das Flores, 100	São Paulo	SP	(11) 98765-4321	joao@email.com	123.456.789-00	NULL
4	4	Empresa X Ltda	Av. Brasil, 500	Rio de Janeiro	RJ	(21) 3333-4444	empresa@email.com	NULL	12.345.678/0001-90
5	5	João da Silva	Rua das Flores, 100	São Paulo	SP	(11) 98765-4321	joao@email.com	123.456.789-00	NULL
6	6	Empresa X Ltda	Av. Brasil, 500	Rio de Janeiro	RJ	(21) 3333-4444	empresa@email.com	NULL	12.345.678/0001-90
7	7	João da Silva	Rua das Flores, 100	São Paulo	SP	(11) 98765-4321	joao@email.com	123.456.789-00	NULL
8	8	Empresa X Ltda	Av. Brasil, 500	Rio de Janeiro	RJ	(21) 3333-4444	empresa@email.com	NULL	12.345.678/0001-90
9	9	João da Silva	Rua das Flores, 100	São Paulo	SP	(11) 98765-4321	joao@email.com	123.456.789-00	NULL
10	10	Empresa X Ltda	Av. Brasil, 500	Rio de Janeiro	RJ	(21) 3333-4444	empresa@email.com	NULL	12.345.678/0001-90
11	11	João da Silva	Rua das Flores, 100	São Paulo	SP	(11) 98765-4321	joao@email.com	123.456.789-00	NULL
12	12	Empresa X Ltda	Av. Brasil, 500	Rio de Janeiro	RJ	(21) 3333-4444	empresa@email.com	NULL	12.345.678/0001-90

Pesquisador de Objetos

Conectar

BRUNO-VENTURELL\SQLEXPRESS (SQL Server 16.0.1000 - loja)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grafo
 - dbo.Compras
 - dbo.dd
 - dbo.Pessoa
 - dbo.Produto
 - dbo.Usuario
 - dbo.Vendas
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Repositório de Consultas
 - Service Broker
 - Armazenamento
 - Segurança
 - Objetos de Servidor
 - Replicação
 - Gerenciamento
 - XEvent Profiler

SQLQuery7.sql - BRU...ESS.Loja (loja (53))*

```
USE loja;  
SELECT * FROM dbo.Produto;
```

100 %

Resultados Mensagens

	idProduto	nome	quantidade	preco_venda
1	1	Notebook	10	2500.00
2	2	Mouse	50	30.00

Pesquisador de Objetos

Conectar

BRUNO-VENTURELL\SQLEXPRESS (SQL Server 16.0.1000 - loja)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grafo
 - dbo.Compras
 - dbo.dd
 - dbo.Pessoa
 - dbo.Produto
 - dbo.Usuario
 - dbo.Vendas
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Repositório de Consultas
 - Service Broker
 - Armazenamento
 - Segurança
 - Objetos de Servidor
 - Replicação
 - Gerenciamento

SQLQuery7.sql - BRU...ESS.Loja (loja (53))*

```
USE loja;  
SELECT * FROM dbo.Compras;
```

100 %

Resultados Mensagens

	idCompras	id_usuario	id_pessoa_juridica	id_produto	quantidade	preco_unitario	data_compra
1	1	2	2	1	5	2400.00	2025-05-14 00:18:18.723

Conectar

BRUNO-VENTURELL\SQLEXPRESS (SQL Server 16.0.1000 - loja)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.Compras

dbo.dd

dbo.Pessoa

dbo.Produto

dbo.Usuario

dbo.Vendas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Segurança

Objetos de Servidor

Replicação

Gerenciamento

XEvent Profiler

USE loja;

SELECT * FROM dbo.Usuario;









100 %

Resultados

Mensagens

	idUsuarios	Nome	Login	Senha
1	1	Administrador	admin	senha123
2	2	Operador Loja	loja	loja123
3	4	Administrador	admin1	senha123
4	5	Operador Loja	loja1	loja123

Pesquisador de Objetos

Conectar        

BRUNO-VENTURELL\SQLSERVER (SQL Server 16.0.1000 - loja)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grafo
 - dbo.Compras
 - dbo.dd
 - dbo.Pessoa
 - dbo.Produto
 - dbo.Usuario
 - dbo.Vendas
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Repositório de Consultas
 - Service Broker
 - Armazenamento
 - Segurança
 - Segurança
 - Objetos de Servidor
 - Replicação
 - Gerenciamento
 - XEvent Profiler

SQLQuery7.sql - BRU...ESS.Loja (loja (53))









```
USE loja;  
SELECT * FROM dbo.dd;
```

100 %

Resultados Mensagens

	idPessoa	nome	tipo_pessoa	cpf	cnj	telefone	email	endereco
1	1	João Silva	F	12345678901	NULL	(11) 98765-4321	joao@email.com	Rua A, 123
2	2	Empresa LTDA	J	NULL	12345678901234	(11) 3210-9876	empresa@email.com	Av. B, 456

Pesquisador de Objetos

Conectar        

BRUNO-VENTURELL\SQLSERVER (SQL Server 16.0.1000 - loja)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grafo
 - dbo.Compras
 - dbo.dd
 - dbo.Pessoa
 - dbo.Produto
 - dbo.Usuario
 - dbo.Vendas
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Repositório de Consultas
 - Service Broker
 - Armazenamento
 - Segurança
 - Segurança
 - Objetos de Servidor
 - Replicação
 - Gerenciamento
 - XEvent Profiler

SQLQuery7.sql - BRU...ESS.Loja (loja (53))

```
USE loja;  
SELECT * FROM dbo.vendas;
```

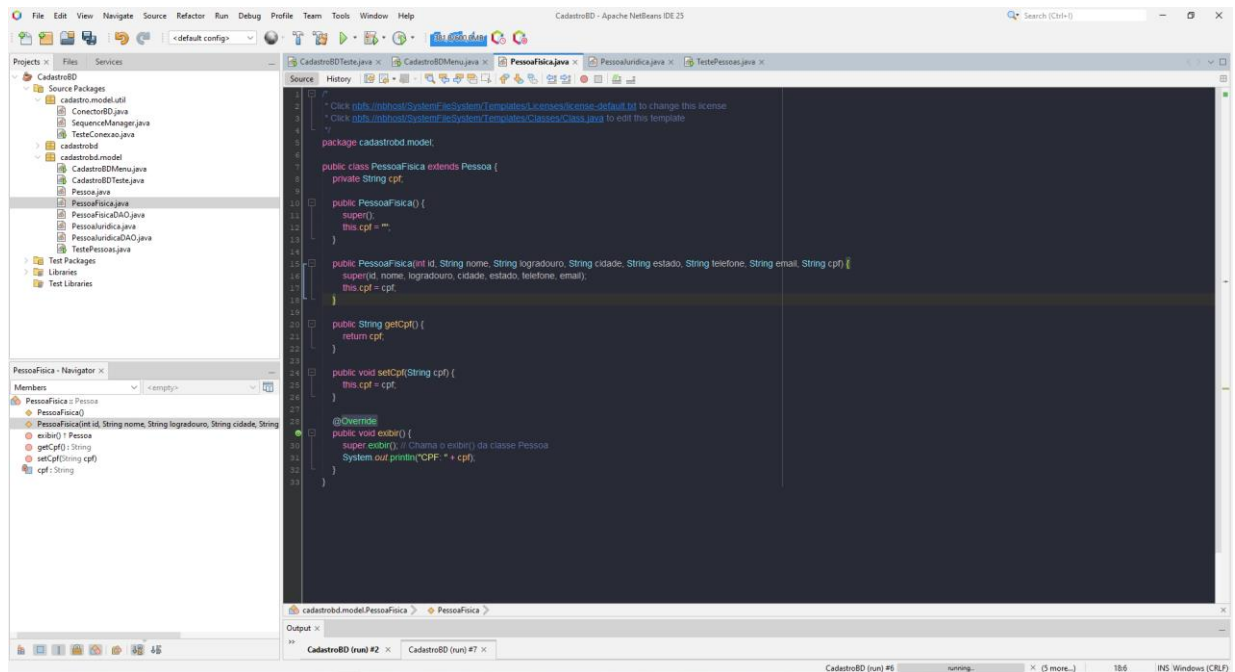
100 %

Resultados Mensagens

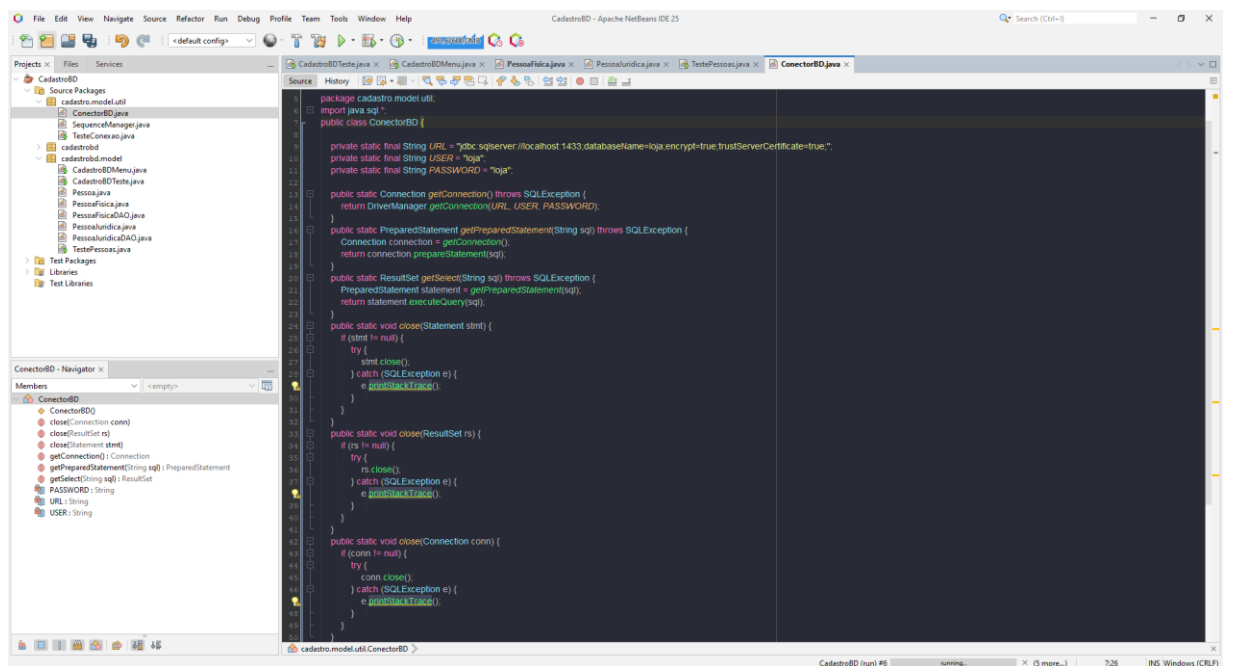
	idVendas	id_usuario	id_pessoa_fisica	id_produto	quantidade	preco_unitario	data_venda
1	1	2	1	2	2	35.00	2025-05-14 00:18:47.820

1º Procedimento | Mapeamento Objeto-Relacional DAO

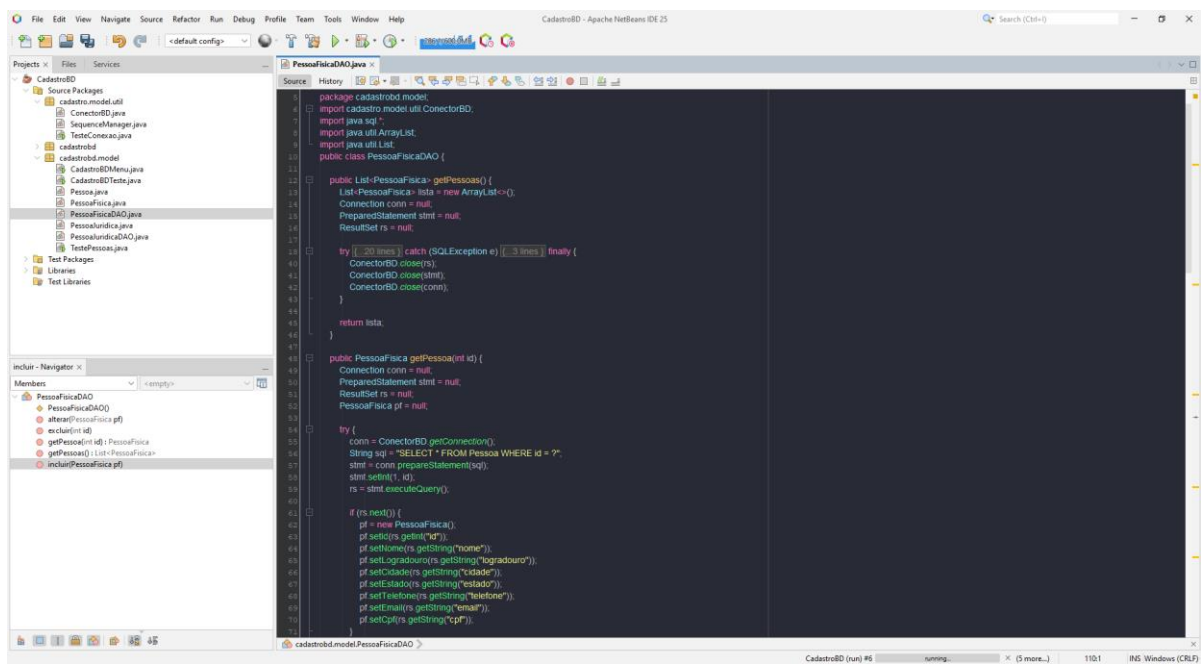
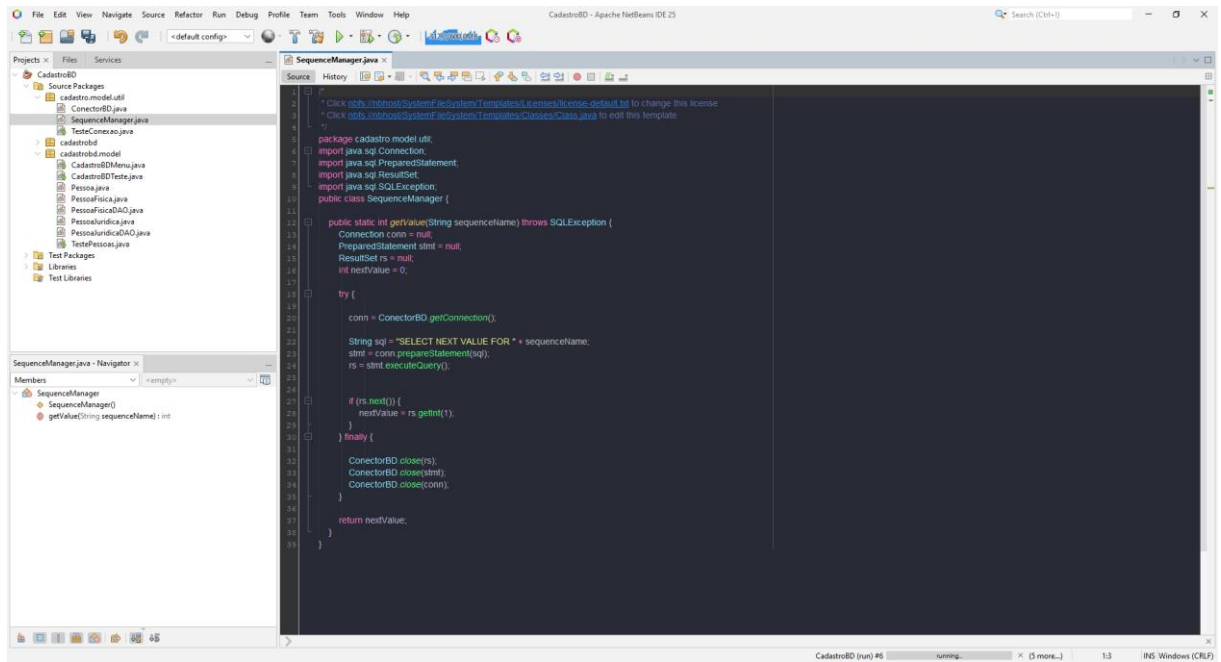
Códigos usados neste roteiro

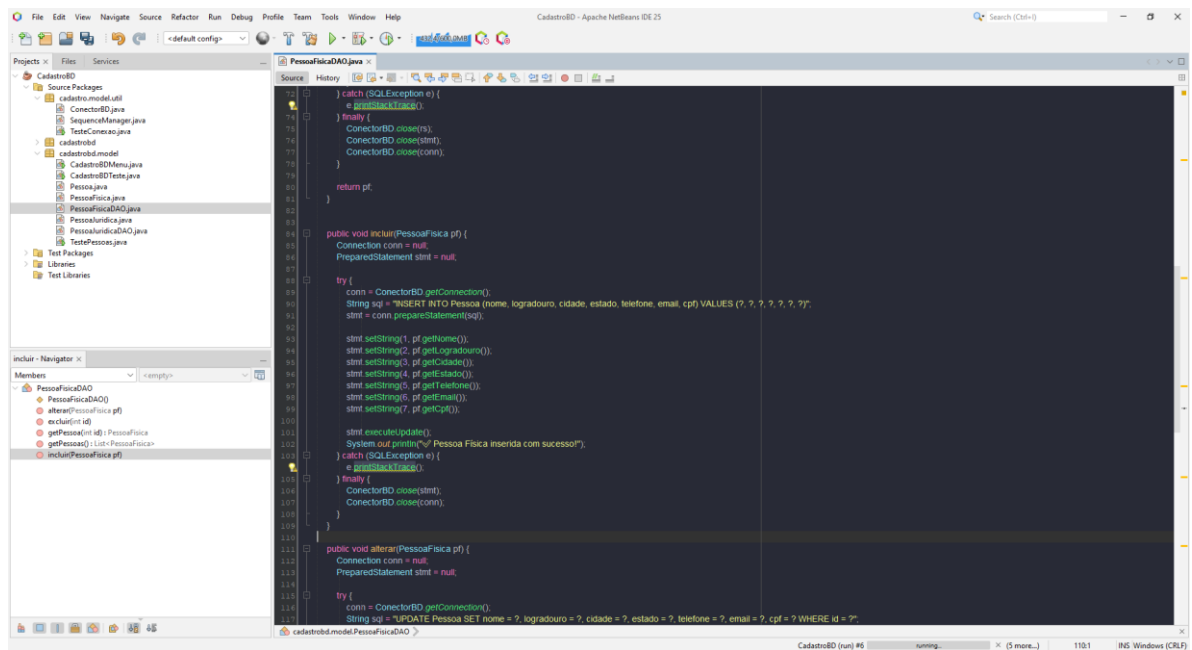


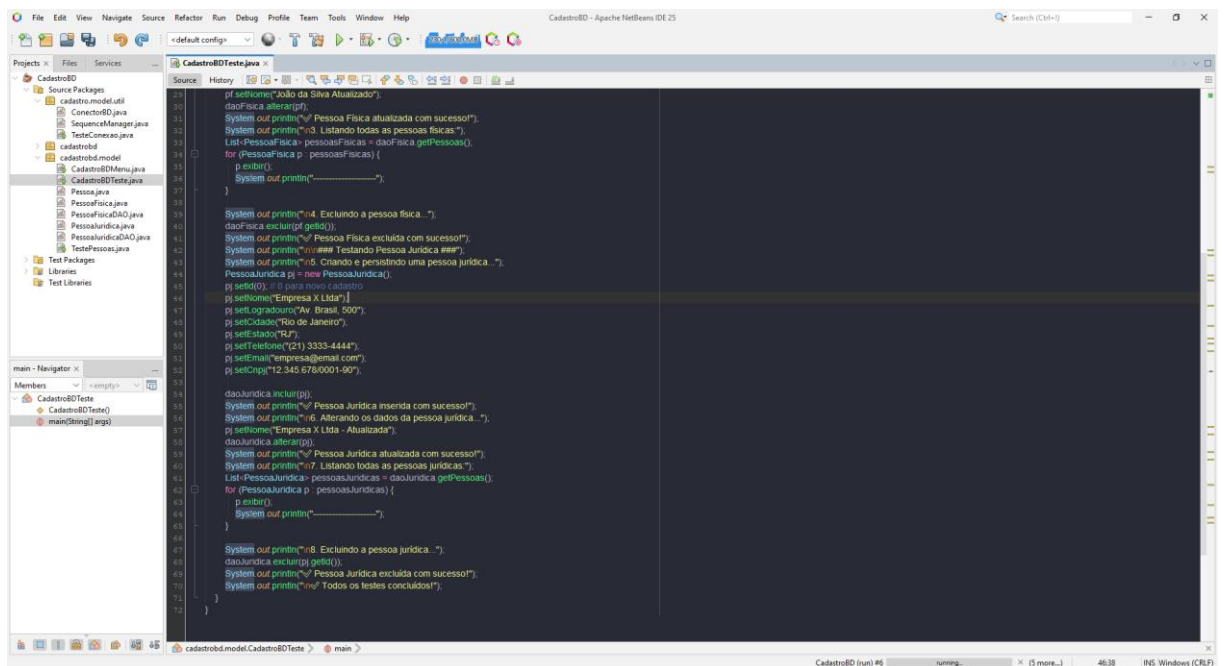
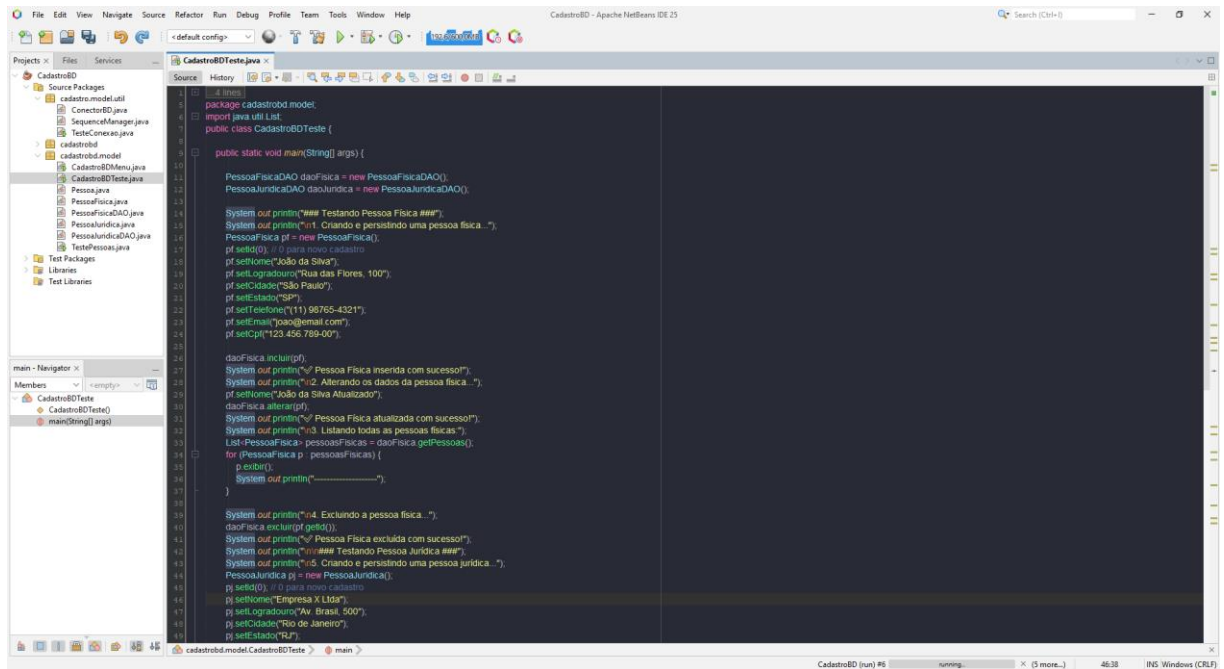
```
1 package cadastrobdt.model;
2
3 import java.util.*;
4
5 public class PessoaFisica extends Pessoa {
6     private String cpf;
7
8     public PessoaFisica() {
9         super();
10        this.cpf = "";
11    }
12
13    public PessoaFisica(int id, String nome, String logradouro, String cidade, String estado, String telefone, String email, String cpf) {
14        super(id, nome, logradouro, cidade, estado, telefone, email);
15        this.cpf = cpf;
16    }
17
18    public String getCpf() {
19        return cpf;
20    }
21
22    public void setCpf(String cpf) {
23        this.cpf = cpf;
24    }
25
26    @Override
27    public void exibir() {
28        super.exibir(); // Chama o exibir() da classe Pessoa
29        System.out.println("CPF: " + cpf);
30    }
31
32 }
```

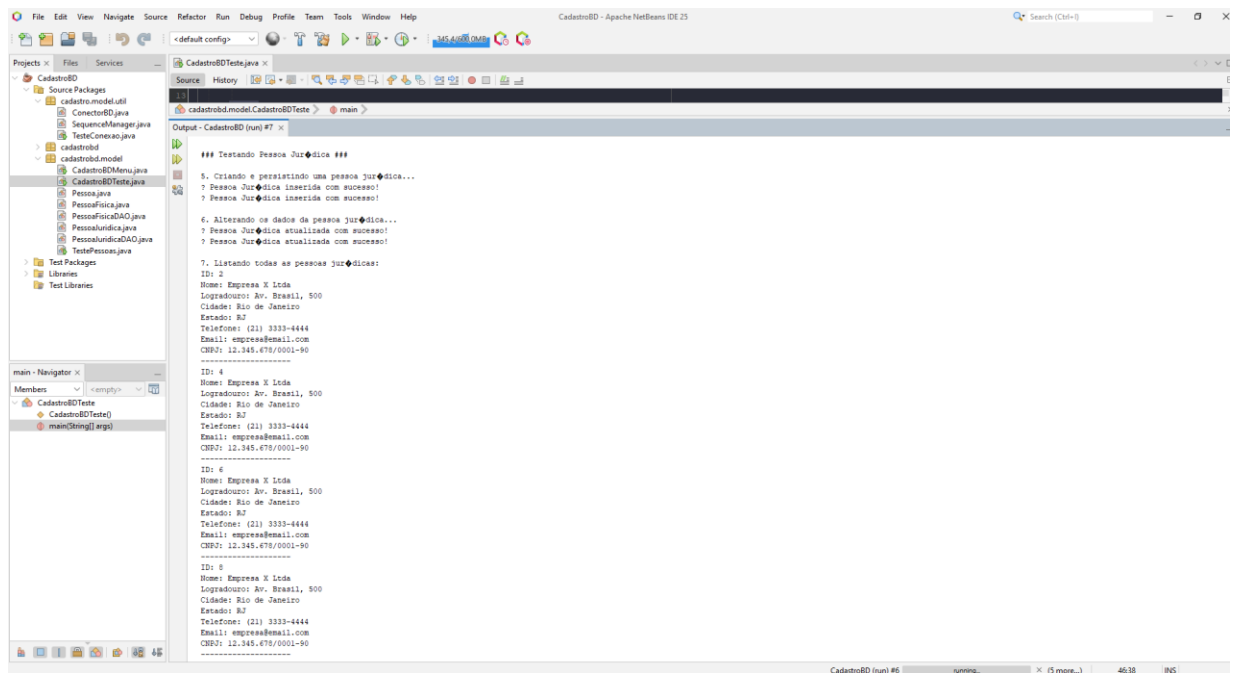
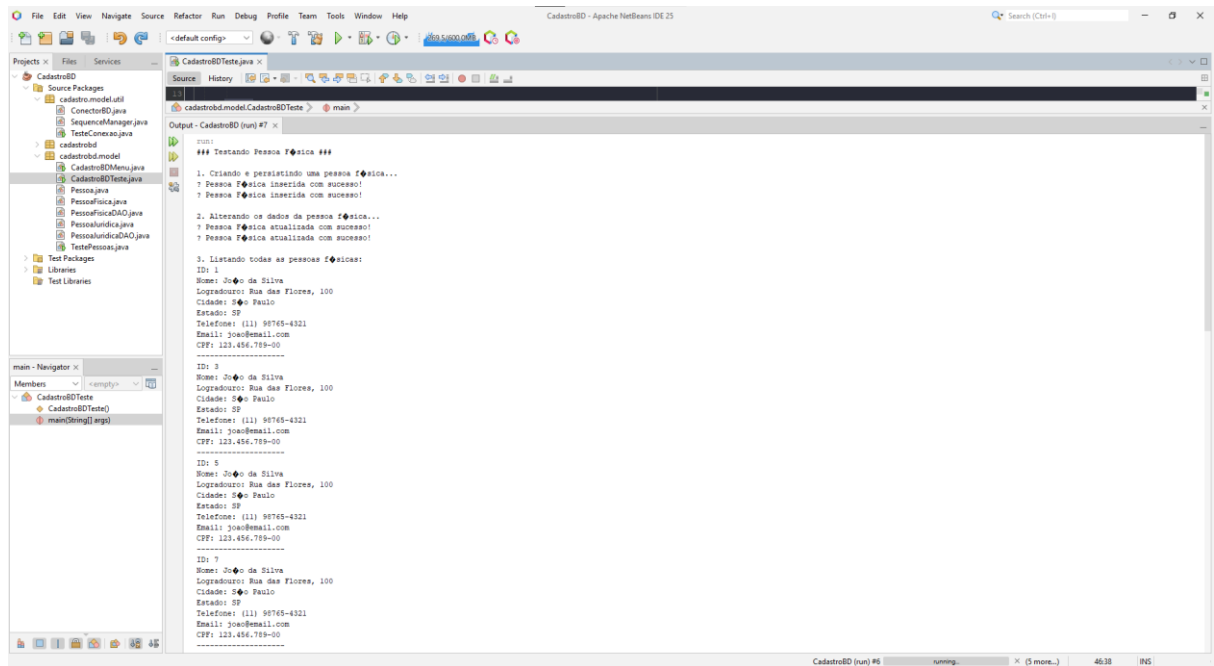


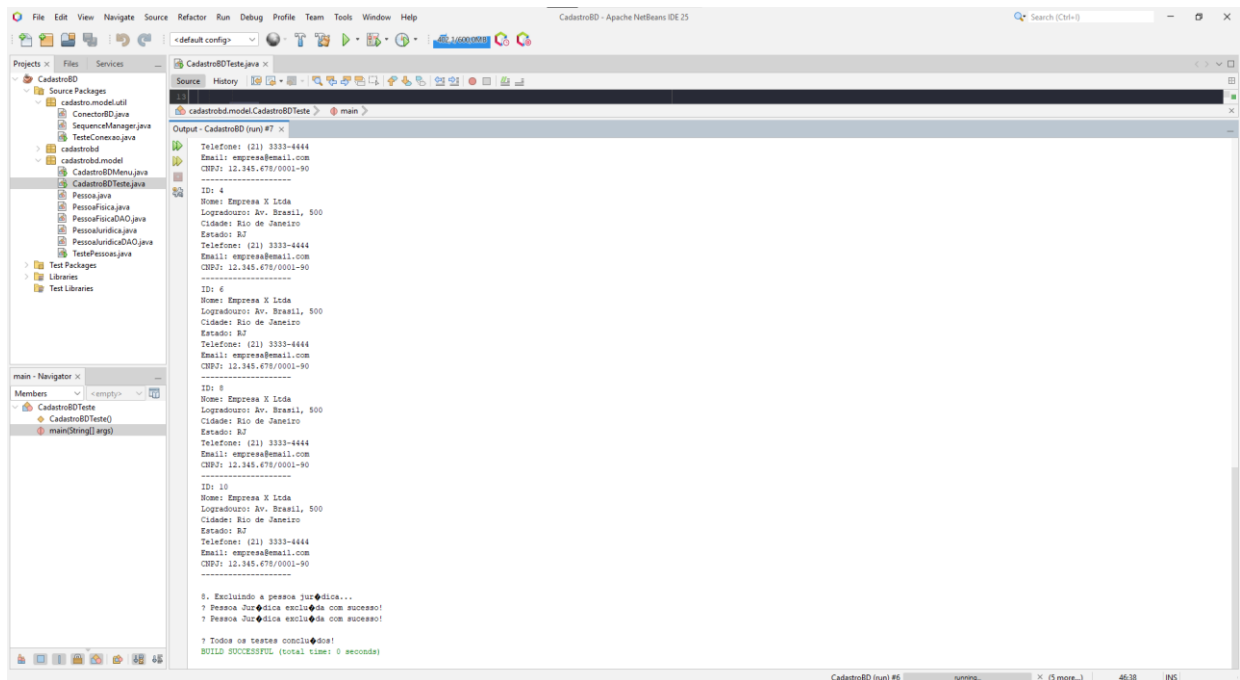
```
1 package cadastrobdt.model;
2
3 import java.sql.*;
4
5 public class ConectorBD {
6
7     private static final String URL = "jdbc:sqlserver://localhost:1433;databaseName=loja;encrypt=true;trustServerCertificate=true";
8     private static final String USER = "loja";
9     private static final String PASSWORD = "loja";
10
11     public static Connection getConnection() throws SQLException {
12         return DriverManager.getConnection(URL, USER, PASSWORD);
13     }
14
15     public static PreparedStatement getPreparedStatement(String sql) throws SQLException {
16         Connection connection = getConnection();
17         return connection.prepareStatement(sql);
18     }
19
20     public static ResultSet getSelect(String sql) throws SQLException {
21         PreparedStatement statement = getPreparedStatement(sql);
22         return statement.executeQuery(sql);
23     }
24
25     public static void closeStatement(Statement stmt) {
26         if (stmt != null) {
27             try {
28                 stmt.close();
29             } catch (SQLException e) {
30                 e.printStackTrace();
31             }
32         }
33     }
34
35     public static void closeResultSet(ResultSet rs) {
36         if (rs != null) {
37             try {
38                 rs.close();
39             } catch (SQLException e) {
40                 e.printStackTrace();
41             }
42         }
43     }
44
45     public static void close(Connection conn) {
46         if (conn != null) {
47             try {
48                 conn.close();
49             } catch (SQLException e) {
50                 e.printStackTrace();
51             }
52         }
53     }
54 }
```



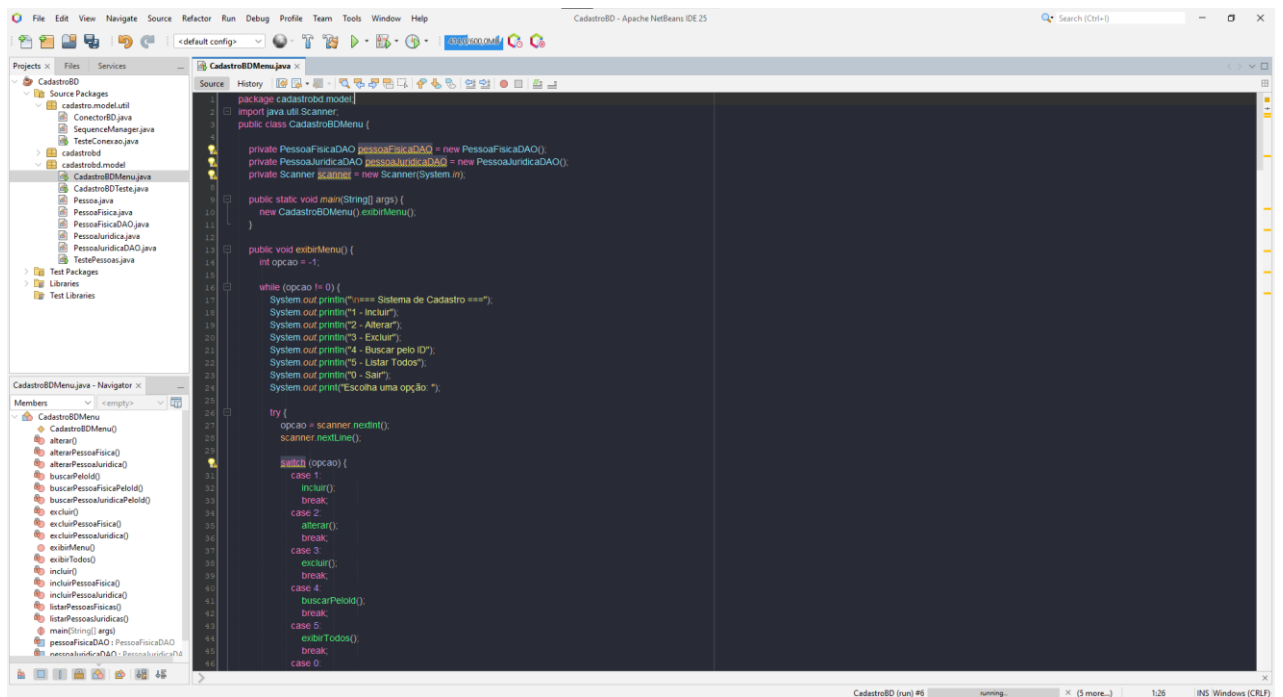


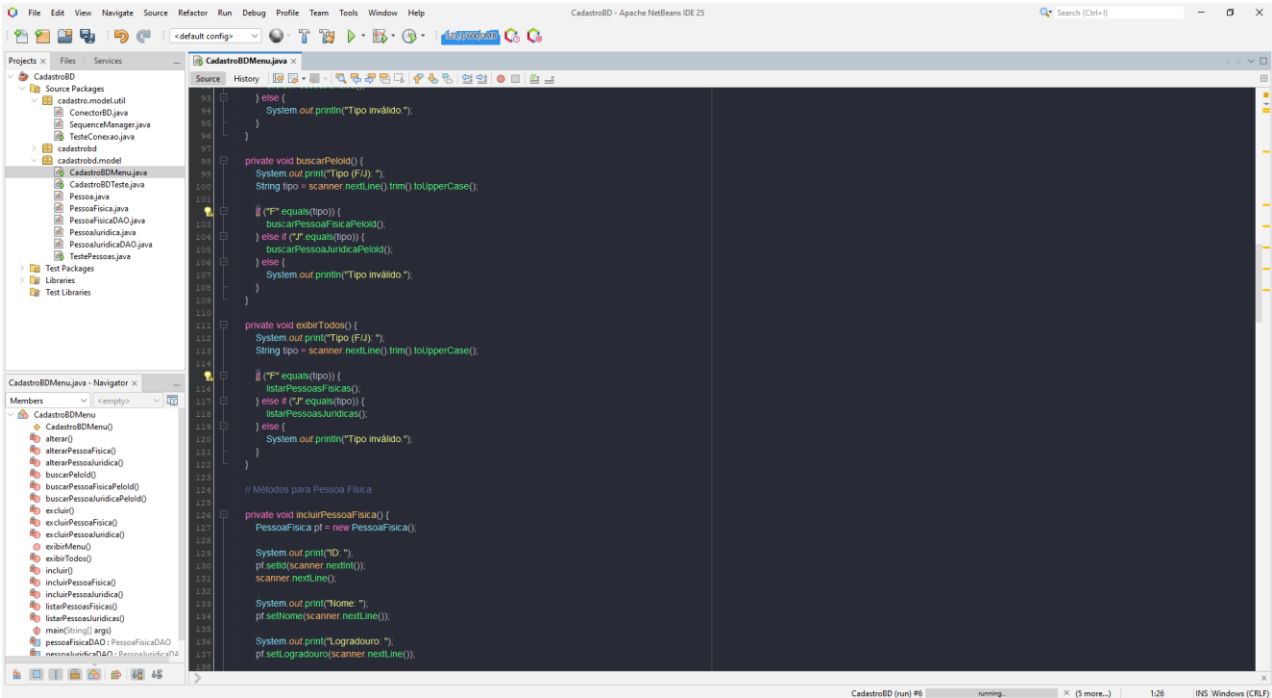


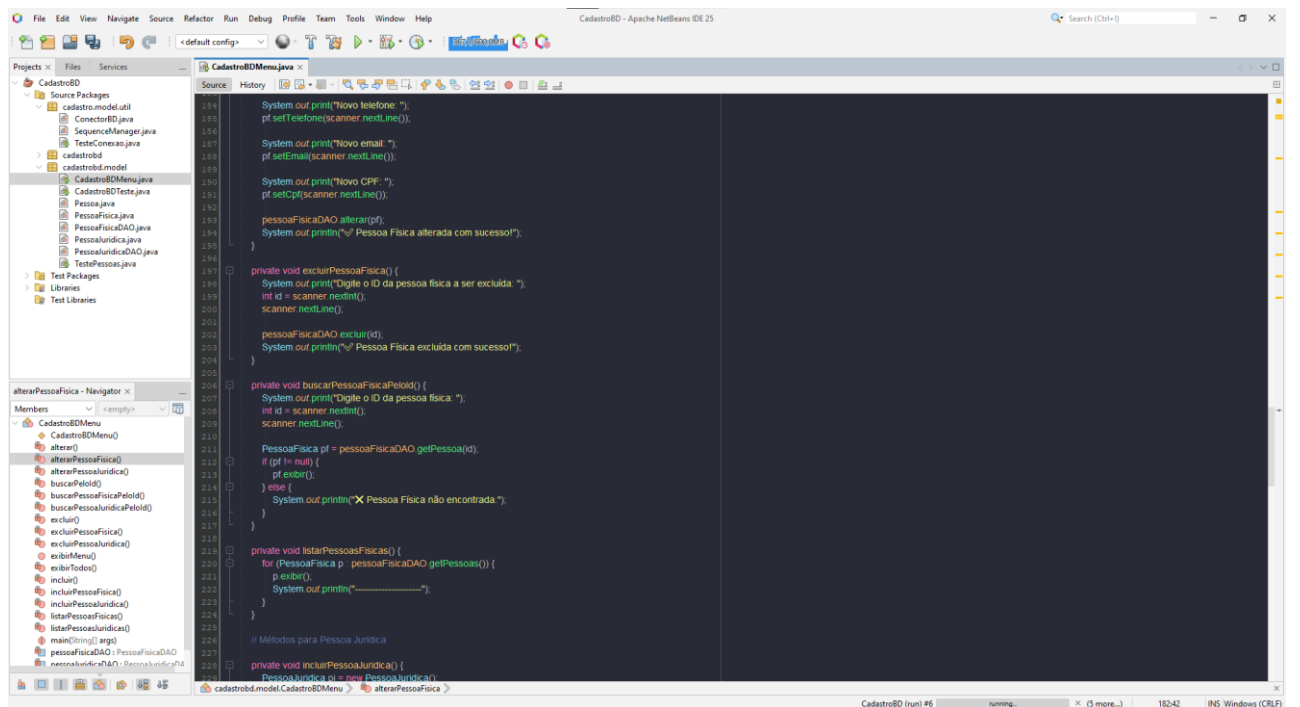
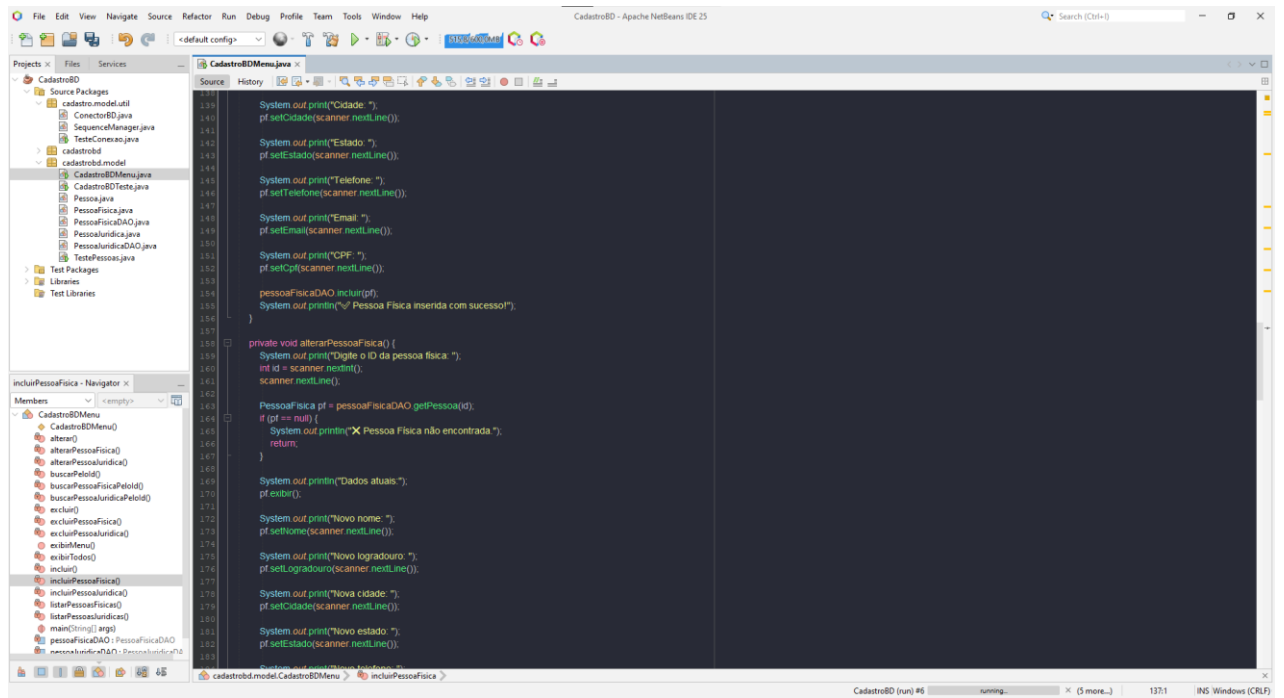


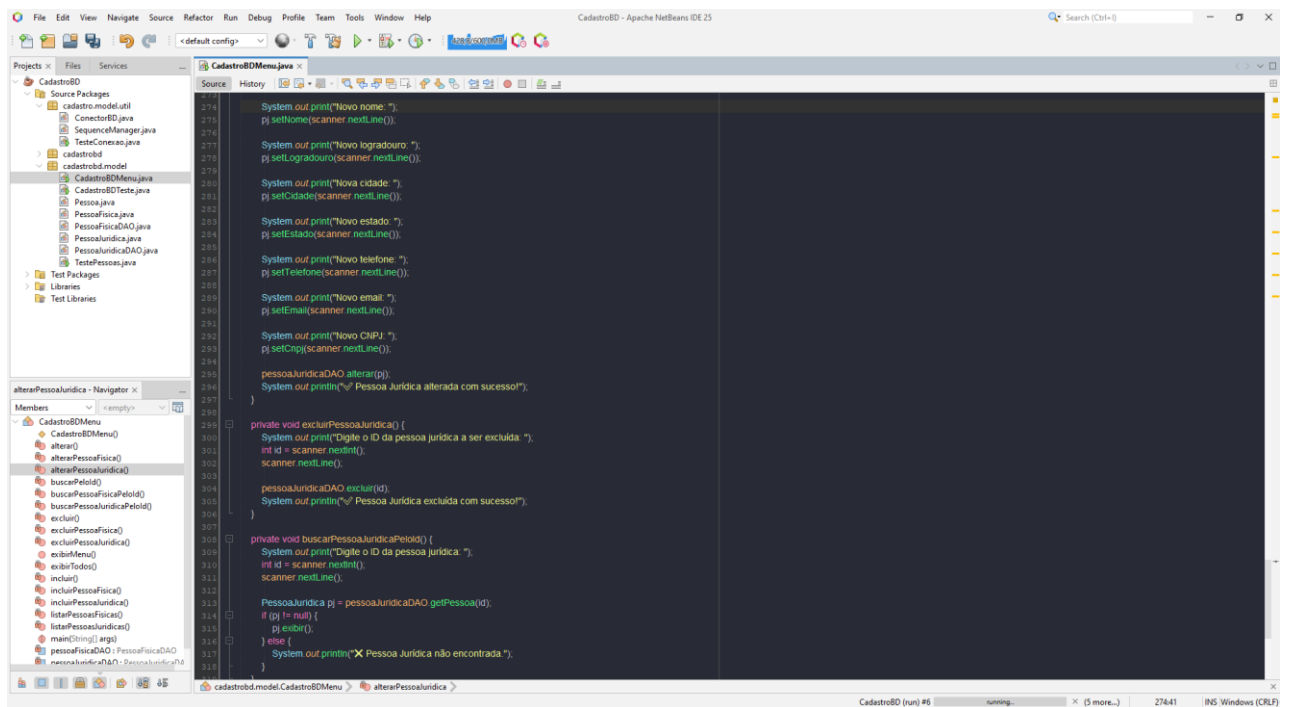
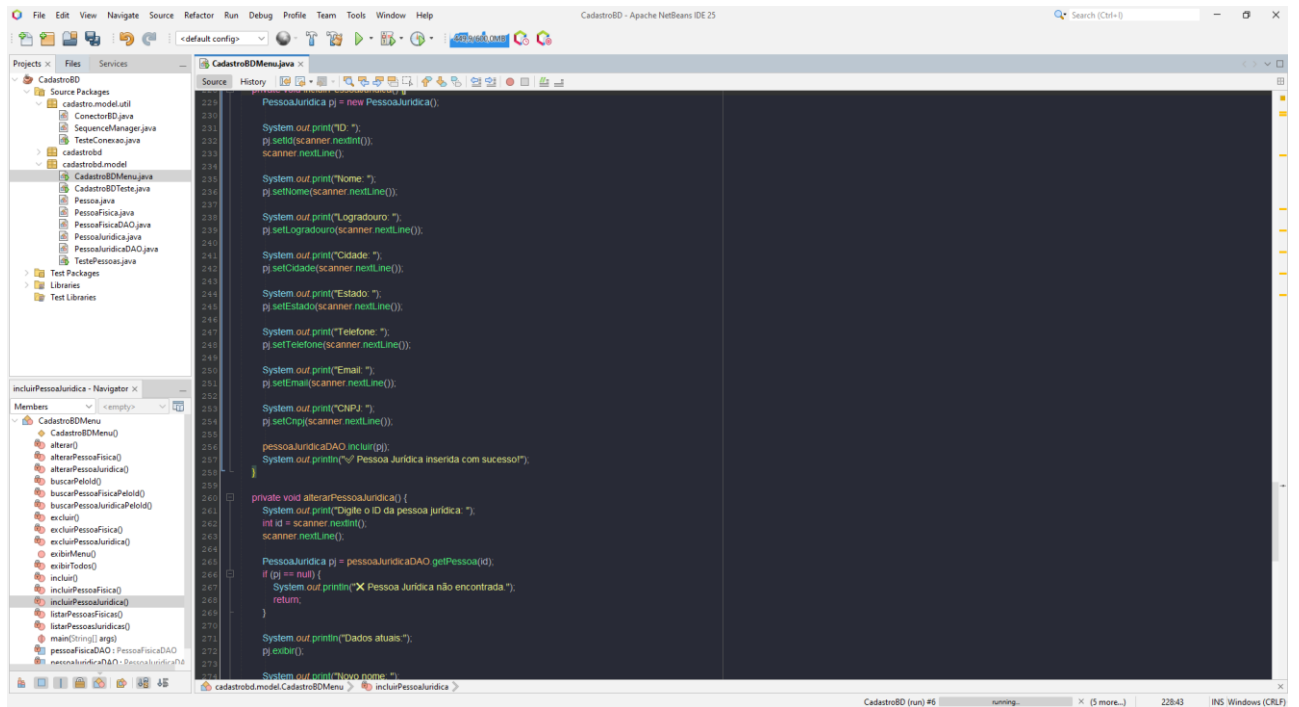
2º Procedimento | Alimentando a Base

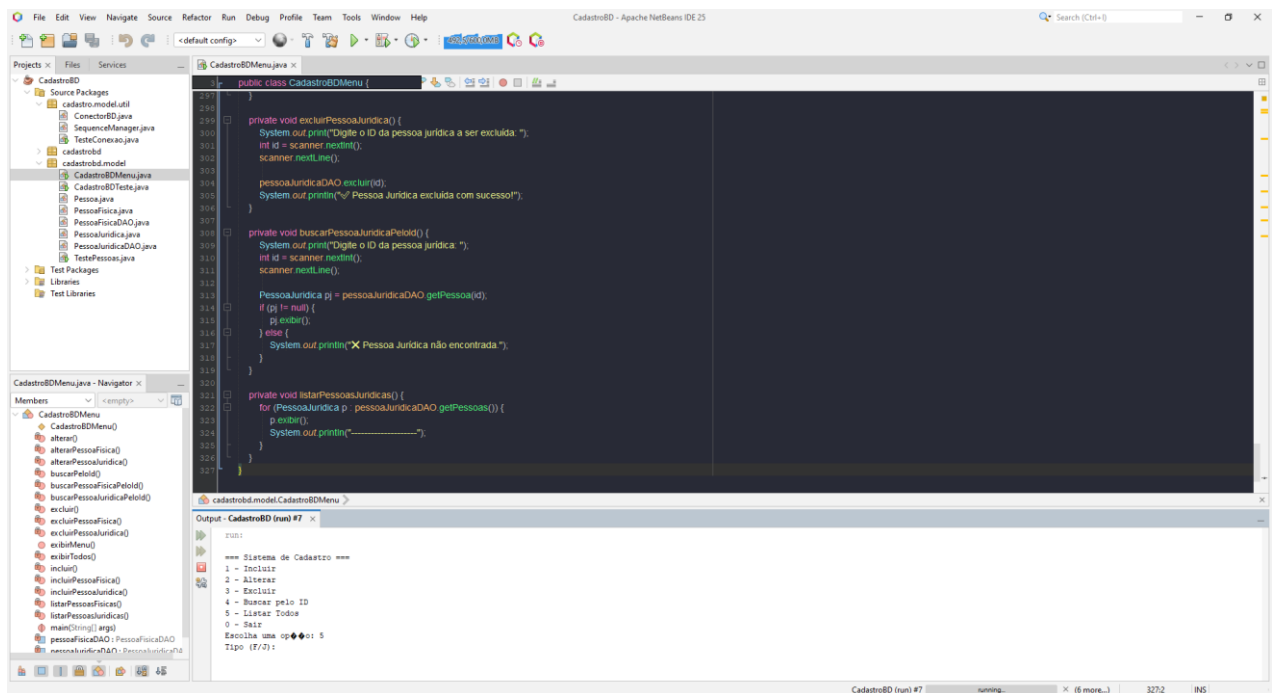
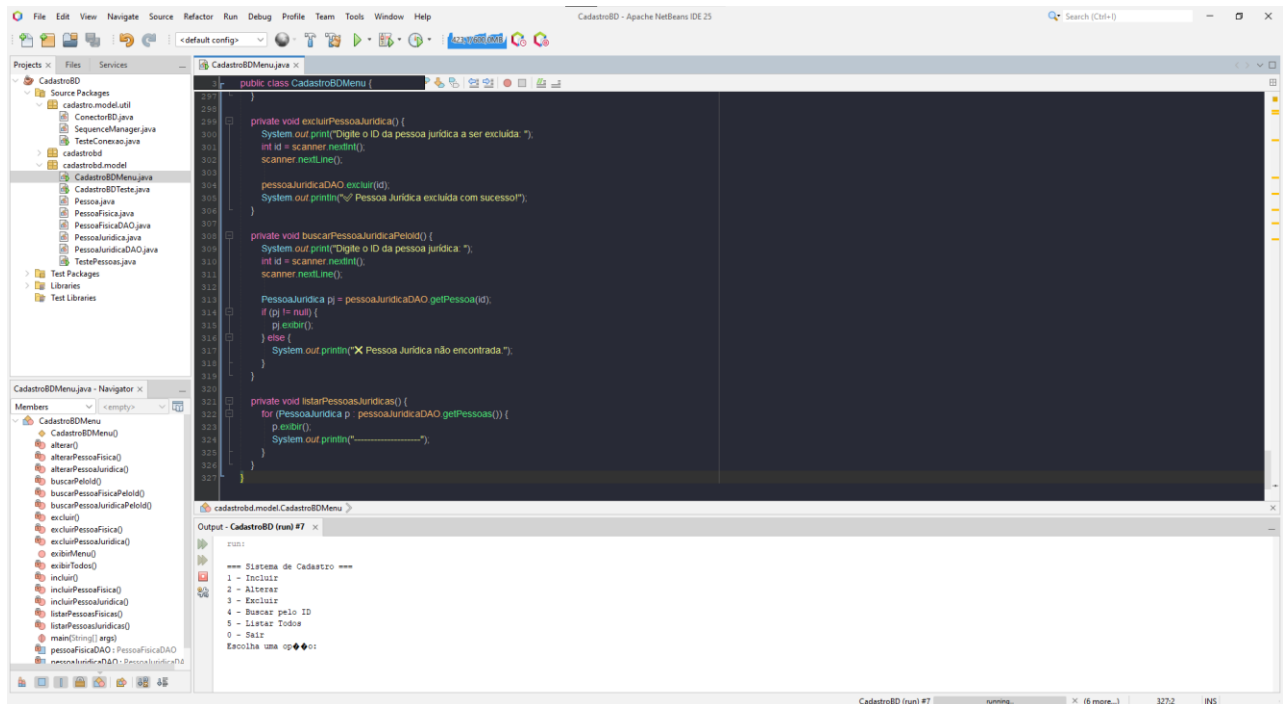
C digos usados neste roteiro

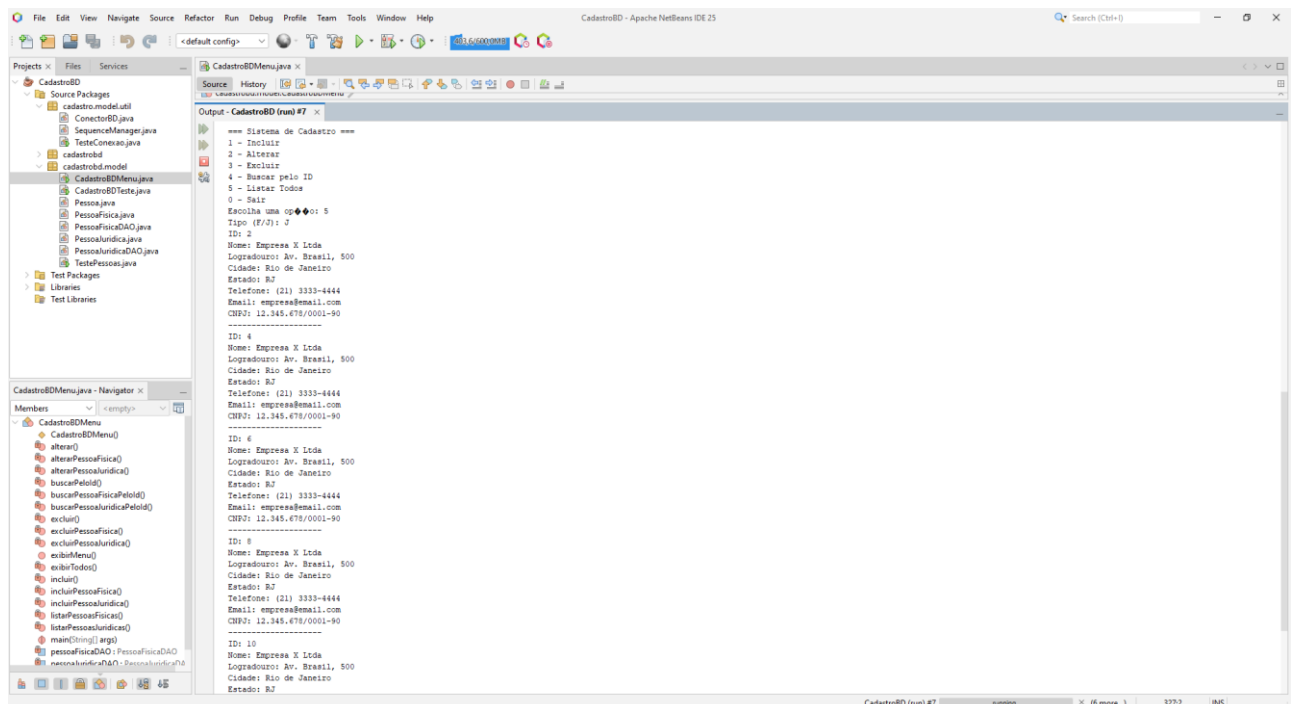
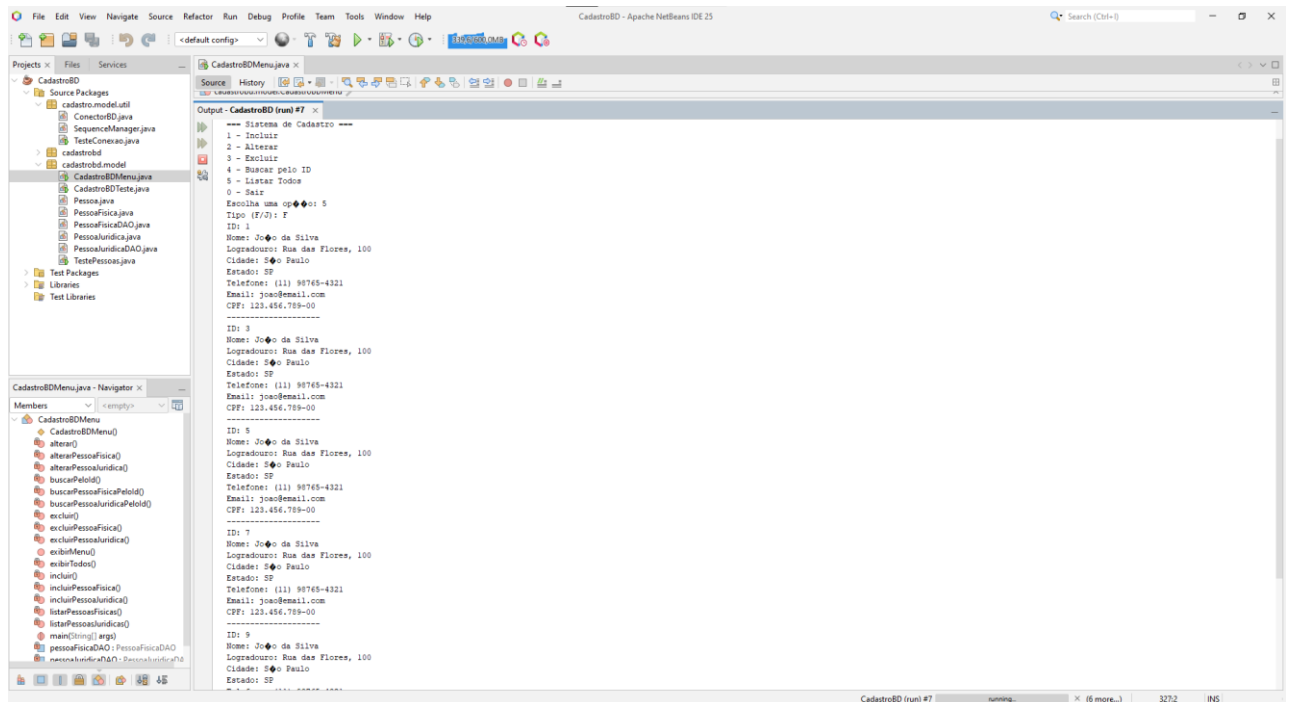












Conclusão do exercício, 1º Procedimento:

Qual a importância dos componentes de middleware, como o JDBC?

Os componentes de middleware, como o JDBC (Java Database

Connectivity), têm papel fundamental na comunicação entre a aplicação e o banco de dados. Eles atuam como uma ponte de integração, abstraindo detalhes técnicos e facilitando o desenvolvimento.

Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?

A principal diferença entre Statement e PreparedStatement no JDBC está em segurança, desempenho e facilidade de uso.

Como o padrão DAO melhora a manutenibilidade do software?

O padrão DAO (Data Access Object) melhora significativamente a manutenibilidade do software ao separar a lógica de acesso a dados da lógica de negócios.

Análise e conclusão:

Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?

Quando lidamos com herança em um modelo orientado a objetos (como em Java), mas usamos um banco de dados relacional (que não tem herança nativamente), precisamos simular esse comportamento com estratégias específicas de modelagem.

Conclusão do exercício, 2º Procedimento:

Qual a importância dos componentes de middleware, como o JDBC?

Os componentes de middleware, como o JDBC (Java Database Connectivity), têm papel fundamental na comunicação entre a aplicação e o banco de dados. Eles atuam como uma ponte de integração, abstraindo detalhes técnicos e facilitando o desenvolvimento.

Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?

O uso de operadores lambda no Java (introduzidos a partir do Java 8) trouxe uma forma muito mais concisa, expressiva e moderna de trabalhar com coleções, inclusive para imprimir valores de entidades..

Análise e conclusão:

Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

Métodos chamados diretamente a partir do método main precisam ser static porque o main também é um método estático e métodos estáticos só podem acessar diretamente outros métodos ou atributos que também sejam estáticos.