Aplicativo Swing em Java NetBeans - Cadastro de Pessoas

Visão Geral

Este é um aplicativo CRUD (Create, Read, Update, Delete) completo desenvolvido em Java usando Swing e SQLite para gerenciar cadastros de pessoas.

Componentes Principais

1. Classe Pessoa (Modelo)

- Representa a entidade pessoa com atributos: id, nome, idade e salário
- Possui construtor e métodos getters/setters
- Implementa o padrão JavaBean

2. PessoaDAO (Acesso a Dados)

- Classe responsável pela persistência no banco SQLite
- Métodos principais:
    - inserir(): Adiciona nova pessoa
    - listar(): Retorna todas as pessoas cadastradas
    - atualizar(): Modifica dados de uma pessoa existente
    - excluir(): Remove uma pessoa do banco

3. TelaPrincipal (Interface)

- Interface gráfica desenvolvida com Swing
- Componentes:
    - Campos de texto para nome, idade e salário
    - Tabela para exibir registros
    - Botões para operações CRUD e relatório

4. Conexão com Banco de Dados

- ConexaoSQLite: Gerencia a conexão com o SQLite
- CriarBanco: Cria a tabela pessoa no banco

Funcionalidades

Operações CRUD

1. **Inserir**:
    - Valida campos obrigatórios
    - Converte tipos numéricos
    - Exibe mensagens de erro/sucesso

2. **Atualizar**:
    - Requer seleção prévia na tabela
    - Atualiza dados da pessoa selecionada
3. **Excluir**:
    - Confirmação com detalhes do registro
    - Remove pessoa do banco
4. **Listar**:
    - Atualiza tabela automaticamente
    - Mostra todos os registros ordenados por ID

Relatório

- Gera estatísticas:
    - Média salarial e de idade
    - Maior salário
    - Soma total de salários
    - Lista completa de registros
- Exibe em nova janela com tabela formatada

Técnicas Utilizadas

1. **Padrão MVC (Model-View-Controller)**:
    - Model: Classe Pessoa
    - View: TelaPrincipal
    - Controller: PessoaDAO e métodos de ação
2. **Manipulação de Banco de Dados**:
    - SQLite como banco embutido
    - JDBC para operações SQL
    - PreparedStatement para segurança
3. **Componentes Swing**:
    - JTable com DefaultTableModel
    - JOptionPane para diálogos
    - JScrollPane para rolagem
    - Tratamento de eventos
4. **Validações**:

- Campos obrigatórios
- Tipos numéricos
- Confirmação para exclusão

Como Executar

1. Criar o banco executando CriarBanco.main()
2. Executar TelaPrincipal.main()
3. Usar a interface para gerenciar cadastros

Observações

- O código demonstra boas práticas como:
  - Separação de responsabilidades
  - Tratamento de exceções
  - Reutilização de código
  - Interface intuitiva

Guia de Estudo Didático: Aplicativo CRUD em Java Swing

Relacionando os Arquivos por Operação

Este guia explica como os arquivos Pessoa.java, PessoaDAO.java e TelaPrincipal.java se conectam em cada operação do CRUD (Create, Read, Update, Delete).

---

1. Estrutura Básica dos Arquivos

Arquivo	Função
Pessoa.java	Classe modelo que representa uma pessoa (atributos: id, nome, idade, salario).
PessoaDAO.java	Classe de acesso a dados (Database Access Object) que faz operações no banco SQLite.
TelaPrincipal.java	Interface gráfica (Swing) que interage com o usuário e chama os métodos do PessoaDAO.

---

2. Operações do CRUD

🔷 CREATE (Inserir Pessoa)

Fluxo:

1. **TelaPrincipal** (Interface) coleta os dados do usuário (nome, idade, salário).
2. **Pessoa** (Modelo) cria um objeto com esses dados.
3. **PessoaDAO** insere no banco de dados.

Conexão entre arquivos:
✅ **TelaPrincipal** → **Pessoa** (cria objeto) → **PessoaDAO** (insere no banco).

Trechos relevantes:

```
// TelaPrincipal.java (btnInserir)
String nome = txtNome.getText();
int idade = Integer.parseInt(txtIdade.getText());
double salario = Double.parseDouble(txtSalario.getText());

Pessoa novaPessoa = new Pessoa(0, nome, idade, salario); // Cria objeto Pessoa
PessoaDAO.inserir(novaPessoa); // Chama o DAO para inserir
```

```
// PessoaDAO.java (método inserir)
public static void inserir(Pessoa p) {
    String sql = "INSERT INTO pessoa (nome, idade, salario) VALUES (?, ?, ?)";
    // Executa no banco...
}
```

🔷 READ (Listar Pessoas)

Fluxo:

1. **TelaPrincipal** solicita a lista de pessoas.
2. **PessoaDAO** busca no banco e retorna uma List<Pessoa>.
3. **TelaPrincipal** exibe os dados na tabela (JTable).

Conexão entre arquivos:
✅ **TelaPrincipal** → **PessoaDAO** (busca dados) → **Pessoa** (modelo)
→ **TelaPrincipal** (exibe).

Trechos relevantes:

```java
// TelaPrincipal.java (atualizarTabela)
List<Pessoa> pessoas = PessoaDAO.listar(); // Busca dados via DAO
DefaultTableModel model = (DefaultTableModel) tabelaPessoas.getModel();
model.setRowCount(0);

for (Pessoa p : pessoas) {
    model.addRow(new Object[]{ p.getId(), p.getNome(), p.getIdade(), p.getSalario() });
}
```

```java
// PessoaDAO.java (método listar)
public static List<Pessoa> listar() {
    List<Pessoa> pessoas = new ArrayList<>();
    String sql = "SELECT * FROM pessoa ORDER BY id";
    // Retorna List<Pessoa>...
}
```

🔷 **UPDATE (Atualizar Pessoa)**

Fluxo:

1. **TelaPrincipal** permite selecionar uma pessoa na tabela.
2. O usuário edita os campos e clica em **Atualizar**.
3. **PessoaDAO** modifica o registro no banco.

Conexão entre arquivos:
✅ **TelaPrincipal** (seleciona e envia dados) → **Pessoa** (cria objeto atualizado) → **PessoaDAO** (executa UPDATE).

Trechos relevantes:

```java
// TelaPrincipal.java (btnAtualizar)
int id = (int) tabelaPessoas.getValueAt(linhaSelecionada, 0);
String nome = txtNome.getText();
int idade = Integer.parseInt(txtIdade.getText());
double salario = Double.parseDouble(txtSalario.getText());

Pessoa pessoaAtualizada = new Pessoa(id, nome, idade, salario);
PessoaDAO.atualizar(pessoaAtualizada);
```

```java
// PessoaDAO.java (método atualizar)
public static void atualizar(Pessoa p) {
    String sql = "UPDATE pessoa SET nome=?, idade=?, salario=? WHERE id=?";
    // Executa o UPDATE...
}
```

🔷 **DELETE (Excluir Pessoa)**

Fluxo:

1. **TelaPrincipal** permite selecionar uma pessoa na tabela.
2. O usuário clica em **Excluir** e confirma.
3. **PessoaDAO** remove o registro do banco.

Conexão entre arquivos:
✅ **TelaPrincipal** (seleciona ID) → **PessoaDAO** (executa DELETE).

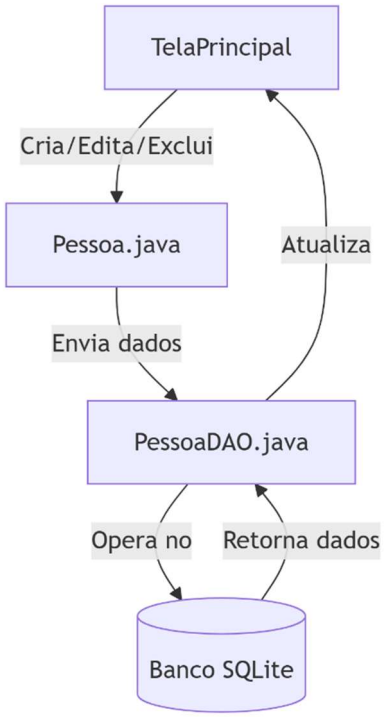
Trechos relevantes:

```
// TelaPrincipal.java (btnExcluir)
int id = (int) tabelaPessoas.getValueAt(linhaSelecionada, 0);
PessoaDAO.excluir(id); // Chama o DAO para excluir
```

```
// PessoaDAO.java (método excluir)
public static void excluir(int id) {
    String sql = "DELETE FROM pessoa WHERE id = ?";
    // Executa o DELETE...
}
```

---

3. Fluxo Geral do CRUD



---

4. Resumo das Conexões

Operação	TelaPrincipal (View)	Pessoa (Model)	PessoaDAO (Controller)
CREATE	Coleta dados do usuário	Cria objeto Pessoa	Insere no banco (INSERT)
READ	Exibe dados na tabela	-	Busca dados (SELECT)
UPDATE	Seleciona e edita dados	Atualiza objeto	Modifica no banco (UPDATE)
DELETE	Seleciona e confirma	-	Remove do banco (DELETE)

---

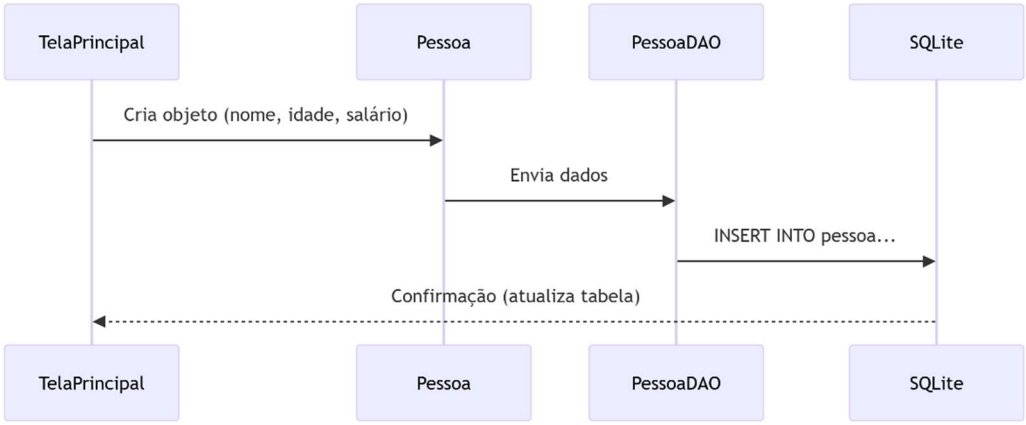
5. Como Estudar?

1. **Comece pelo** Pessoa.java (Modelo): Entenda como os dados são estruturados.
2. **Analise o** PessoaDAO.java: Veja como cada operação (CRUD) é feita no banco.
3. **Explore a** TelaPrincipal.java: Observe como ela interage com o usuário e chama o DAO.
4. **Teste cada operação separadamente**: Modifique um campo e veja como o banco é afetado.

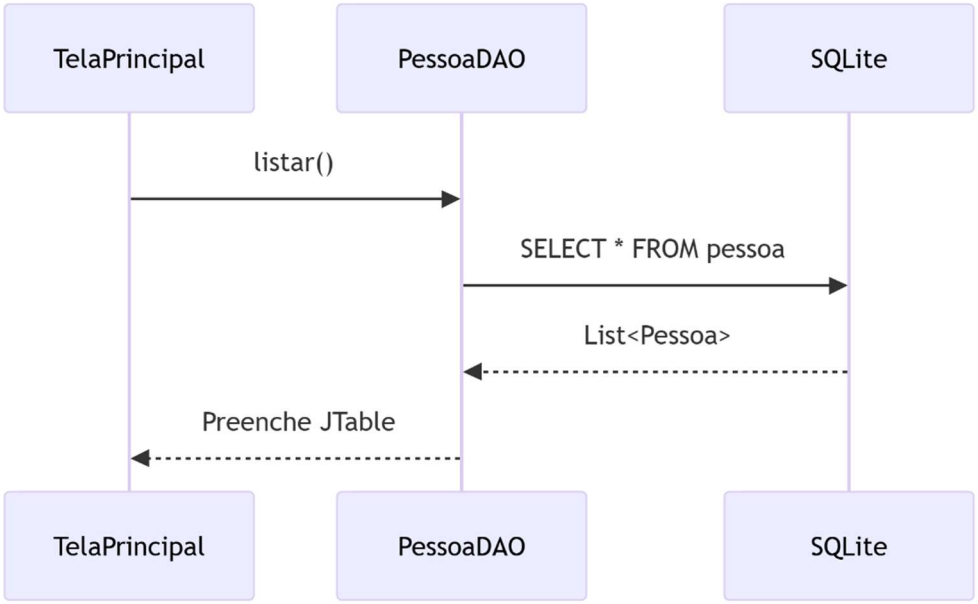
Este guia ajuda a entender **como o Swing (interface) se comunica com o banco de dados** usando **modelos (Pessoa) e operações (DAO)**. 🚀

📌 **Resumo Visual das Conexões (CRUD + Arquivos)**

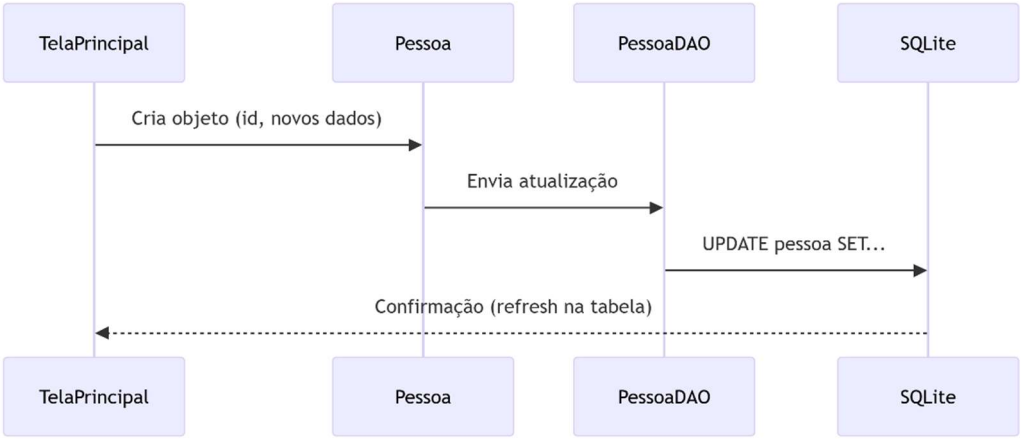
🎯 **CREATE (Inserir)**



🎯 READ (Listar)



🎯 UPDATE (Editar)



🎯 **DELETE (Excluir)**