



Documentação do Projeto: LojaRoupaApp

Instituto Federal de São Paulo – Campus Guarulhos (IFSP – GRU)

Análise e Desenvolvimento de Sistemas

Gustavo Ribeiro Furtado - GU3054098, Luiz Otávio de Oliveira - GU305540X

Docente: Cleber Silva de Oliveira

Resumo

Este documento detalha o projeto **LojaRoupaApp**, uma aplicação desktop desenvolvida em Java (Swing) com integração MySQL, que simula um sistema de gerenciamento para uma loja de roupas. O projeto foi concebido para demonstrar a aplicação de conceitos de Programação Orientada a Objetos (POO), acesso a banco de dados via JDBC e a construção de interfaces gráficas. A documentação aborda a visão geral do sistema, requisitos funcionais, o modelo de banco de dados (lojaroupa com 11 tabelas), a estrutura de pacotes e classes implementadas (Modelos, DAOs e GUI), e as funcionalidades principais, como autenticação de usuário, navegação por um menu principal e operações CRUD completas para gerenciamento de Categorias e Produtos, com base nas interfaces visuais fornecidas.

Abstract

This document provides a comprehensive overview of the **LojaRoupaApp** project, a desktop application developed in Java (Swing) with MySQL integration, simulating a clothing store management system. The project was designed to demonstrate the application of Object-Oriented Programming (OOP) principles, database access via JDBC, and graphical user interface (GUI) development. The documentation covers the system's overview, functional requirements, the database model (lojaroupa with 11 tables), the implemented package and class structure (Models, DAOs, and GUI), and core functionalities such as user authentication, main menu navigation, and complete CRUD operations for Category and Product management, based on the provided visual interfaces.

Documentação do Projeto: LojaRoupaApp.....	1
Resumo.....	1
Abstract.....	1
1. Visão Geral do Projeto.....	2
Tecnologias Utilizadas:.....	3
2. Especificação dos Requisitos.....	3
3. Modelo de Relacionamento do Banco de Dados.....	3
4. Diagrama de Caso de Uso.....	6
5. Partes Implementadas no Código.....	8
Estrutura de Pacotes:.....	9
Classes modelo (em com.lojaroupa.conexao - idealmente com.lojaroupa.model):.....	9
Classe de Conexão:.....	10
Classes DAO (Data Access Object):.....	10
Classe Principal da Aplicação:.....	11
6. Construção das Telas (Interfaces Gráficas).....	11
6.1. Tela de Login (LoginFrame).....	11
6.2. Tela de Menu Principal (MenuPrincipal).....	12
6.3. Tela de Gerenciamento de Categorias (CRUD de Categorias).....	13
7. Funcionalidades Mínimas Implementadas.....	15

1. Visão Geral do Projeto

O projeto **LojaRoupaApp** é uma aplicação desktop desenvolvida em Java, utilizando a biblioteca Swing para a construção da interface gráfica e o MySQL como sistema de gerenciamento de banco de dados. O objetivo principal do sistema é demonstrar conceitos de Programação Orientada a Objetos (POO), acesso a banco de dados (JDBC) e construção de interfaces gráficas. A aplicação simula um

sistema básico de gerenciamento para uma loja de roupas, incluindo funcionalidades de autenticação de usuários e operações CRUD (Create, Read, Update, Delete) para entidades do negócio.

Tecnologias Utilizadas:

- **Linguagem de Programação:** Java
- **Interface Gráfica:** Swing (javax.swing, java.awt)
- **Banco de Dados:** MySQL
- **Driver JDBC:** MySQL Connector/J 9.3.0
- **IDE:** Eclipse IDE

2. Especificação dos Requisitos

O projeto LojaRoupaApp foi desenvolvido seguindo os seguintes requisitos:

- **Programação Orientada a Objetos (POO):** Demonstração do uso de classes, objetos, encapsulamento (getters e setters) e herança (implícita através da estrutura de pacotes e tipos de dados). As classes de modelo (Categoria, Cliente, Produto, etc.) e as classes DAO são exemplos da aplicação dos princípios de POO.
- **Acesso a Banco de Dados:** Utilização do JDBC (Java Database Connectivity) para interagir com o banco de dados MySQL, realizando operações de persistência de dados. A classe ConexaoDB é responsável por gerenciar a conexão.
- **Construção de Interfaces Gráficas:** Implementação de telas interativas utilizando JFrame e componentes visuais da biblioteca Swing (pacotes javax.swing e java.awt).
- **Módulos CRUD Completos:** O sistema deve conter pelo menos dois módulos com funcionalidades completas de Create, Read, Update e Delete (CRUD). Com base nos modelos e DAOs fornecidos, Categorias é um módulo que já possui o INSERT e SELECT ALL implementados.
- **Fluxo de Telas:**
 - Tela de Login: Ponto de entrada do sistema, responsável pela autenticação de usuários.
 - Tela de Menu Principal: Acessível após o login bem-sucedido, redirecionando para as diferentes funcionalidades do sistema.
- **Telas de CRUD:** No mínimo duas telas de CRUD para gerenciar dados.

2.1. Requisitos Funcionais (RF)

Os requisitos funcionais descrevem as funcionalidades que o sistema deve executar para atender às necessidades dos usuários.

- RF01 - Autenticação de Usuário: O sistema deve permitir que usuários (administradores/funcionários e clientes) façam login com credenciais (email/usuário e senha).
- RF02: O sistema deve autenticar funcionários cadastrados no banco de dados com seu email e senha para acesso à tela do Administrador.
- RF03: O sistema deve autenticar clientes cadastrados no banco de dados com seu email e senha para acesso à tela do Cliente.
- RF04: O sistema deve exibir mensagem de erro para credenciais inválidas.
- RF05 – Menu (Administrador): O administrador deve ter acesso à aba de gerenciar clientes e gerenciar pedidos.
- RF06 - Menu (Cliente): O administrador deve ter acesso à aba de ver produtos e cadastrar novo cliente.
- RF07 - Gerenciamento de Pedidos (Administrador): O administrador deve ser capaz de visualizar, adicionar, atualizar e excluir pedidos.
- RF08: O sistema deve exibir uma lista de todos os produtos cadastrados, incluindo nome, marca, modelo, descrição, preço e tamanho.
- RF09 - Gerenciamento de Clientes (Administrador): O administrador deve ser capaz de visualizar, adicionar e excluir clientes.
- RF10: O sistema deve exibir uma lista de todos os clientes cadastrados, incluindo ID, nome, CPF, email, telefone, endereço e CEP.
- RF11: O sistema deve permitir a inserção de novos clientes.
- RF12: O sistema deve permitir a exclusão de clientes selecionados.
- RF13 - Gerenciamento de Pedidos (Administrador): O administrador deve ser capaz de visualizar e "liberar" (excluir do painel) pedidos.
- RF14: O sistema deve exibir uma lista de todos os pedidos, incluindo ID do Pedido, ID do Cliente, Data do Pedido, Status e Valor Total.
- RF15: O sistema deve permitir a "liberação" de pedidos selecionados, que resulta na sua exclusão do banco de dados.
- RF16 - Navegação de Produtos (Cliente): O cliente deve ser capaz de visualizar produtos disponíveis.
- RF17: O sistema deve exibir uma lista de produtos disponíveis para compra.
- RF18: O sistema deve permitir a visualização de detalhes de um produto selecionado.
- RF19 - Realização de Compra (Cliente): O cliente deve ser capaz de realizar a compra de um produto.
- RF20: O sistema deve permitir a seleção de um produto para compra.
- RF21: O sistema deve solicitar um ID de cliente para a compra
- RF22: O sistema deve criar um novo pedido no banco de dados associado ao cliente e ao produto.

2.2. Requisitos Não Funcionais (RNF)

Os requisitos não funcionais descrevem como o sistema deve funcionar, focando em atributos de qualidade.

- RNF01 - Usabilidade: A interface do usuário deve ser intuitiva e fácil de usar para usuários com conhecimentos básicos de informática.
- RNF02 - Desempenho: O tempo de resposta para operações de busca e inserção de dados não deve exceder 3 segundos em condições normais de uso.
- Segurança:
 - RNF04: O sistema deve garantir a confidencialidade das credenciais de login.
 - RNF05: O acesso às funcionalidades administrativas deve ser restrito a usuários autenticados como administradores/funcionários.
- RNF06 - Confiabilidade: O sistema deve manter a integridade dos dados mesmo em caso de falhas de conexão com o banco de dados (com mensagens de erro adequadas).
- RNF07 - Manutenibilidade: O código deve ser modular e bem documentado para facilitar futuras alterações e expansões.
- RNF09 - Portabilidade: O sistema deve ser executável em diferentes sistemas operacionais (Windows, Linux, macOS) desde que o ambiente Java e MySQL estejam configurados.

3. Modelo de Relacionamento do Banco de Dados

O banco de dados lojaroupa é composto por 11 tabelas. As tabelas e suas interações são descritos a seguir:

Tabelas e Colunas Principais:

- **categorias**
 - idCategoria (PK, INT, AUTO_INCREMENT)
 - descricao (VARCHAR)
- **clientes**
 - idCliente (PK, INT, AUTO_INCREMENT)
 - nome (VARCHAR)
 - cpf (VARCHAR)
 - email (VARCHAR)
 - telefone (VARCHAR)
 - endereco (VARCHAR)
 - cep (VARCHAR)
 - senha (VARCHAR)
- **estoque**
 - idEstoque (PK, INT, AUTO_INCREMENT)
 - idProduto (FK, INT)
 - dtEntrada (DATE)

- quantidade (INT)
- nfCompra (VARCHAR)
- precoCompra (DECIMAL)
- precoVenda (DECIMAL)
- qtdVendida (INT)
- **funcionarios**
 - idFuncionario (PK, INT, AUTO_INCREMENT)
 - email (VARCHAR)
 - idNivelUsuario (FK, INT)
 - nome (VARCHAR)
 - cpf (VARCHAR)
 - endereco (VARCHAR)
 - cep (VARCHAR)
 - telefone (VARCHAR)
 - ativo (CHAR)
 - senha (VARCHAR)
- **itemspedido**
 - idItemPedido (PK, INT, AUTO_INCREMENT)
 - idPedido (FK, INT)
 - ordem (INT)
 - idEstoque (FK, INT)
 - qtdItem (INT)
 - dtDevolucao (DATETIME, NULLABLE)
 - motivoDevolucao (VARCHAR, NULLABLE)
- **logisticapedidos**
 - idLogistica (PK, INT, AUTO_INCREMENT)
 - idPedido (FK, INT)
 - dtEnvio (DATETIME)
 - dtRecebimento (DATETIME)
 - rastreioFrete (VARCHAR)
- **movimentacaoestoque**
 - idMovimentacao (PK, INT, AUTO_INCREMENT)
 - idEstoque (FK, INT)
 - tipoMovimentacao (VARCHAR)
 - quantidade (INT)
 - dataMovimentacao (DATETIME)
 - observacao (VARCHAR)
- **nivelusuarios**
 - idNivelUsuario (PK, INT, AUTO_INCREMENT)
 - nivel (VARCHAR)
- **pagamentos**
 - idPagamento (PK, INT, AUTO_INCREMENT)
 - idPedido (FK, INT)
 - metodoPagamento (VARCHAR)

- statusPagamento (VARCHAR)
- valorPago (DECIMAL)
- dtPagamento (DATETIME)
- notaFiscal (VARCHAR)
- dtNotaFiscal (DATETIME)
- **pedidos**
 - idPedido (PK, INT, AUTO_INCREMENT)
 - idFuncionario (FK, INT, NULLABLE)
 - idCliente (FK, INT)
 - idVendedor (INT)
 - dataPedido (DATETIME)
 - *lojaroupa.sql*
 - tipoFrete (INT)
 - entregaEndereco (VARCHAR)
 - entregaCEP (VARCHAR)
 - entregaTelefone (VARCHAR)
 - dtDevolucao (DATETIME, NULLABLE)
 - motivoDevolucao (VARCHAR, NULLABLE)
 - status (VARCHAR)
 - valorTotal (DECIMAL)
- **produtos**
 - idProduto (PK, INT, AUTO_INCREMENT)
 - nome (VARCHAR)
 - marca (VARCHAR)
 - modelo (VARCHAR)
 - idCategoria (FK, INT)
 - descricao (VARCHAR)
 - unidadeMedida (VARCHAR)
 - cor (VARCHAR)
 - estoqueMinimo (INT)
 - preco (DECIMAL)
 - tamanho (VARCHAR)
 - ativo (TINYINT)

Relacionamentos:

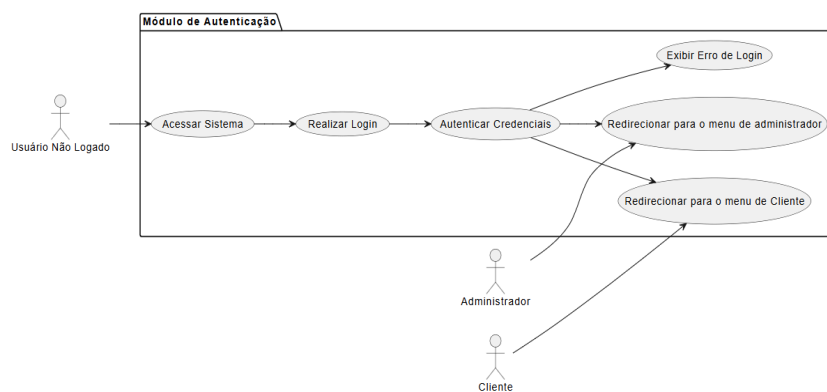
- estoque se relaciona com produtos (1:N)
- funcionarios se relaciona com nivelusuarios (1:N)
- itenspedido se relaciona com pedidos (1:N) e estoque (1:N)
- logisticapedidos se relaciona com pedidos (1:1)
- movimentacaoestoque se relaciona com estoque (1:N)
- pagamentos se relaciona com pedidos (1:1)
- pedidos se relaciona com clientes (1:N) e funcionarios (1:N)
- produtos se relaciona com categorias (1:N)

Este modelo permite um gerenciamento abrangente de produtos, estoque, pedidos, clientes e funcionários, com controle de níveis de acesso e logística.

4. Diagrama de Caso de Uso Um diagrama de caso de uso ilustraria as interações entre os usuários (atores) e o sistema. Abaixo, descrevemos os principais casos de uso identificados no LojaRoupaApp:

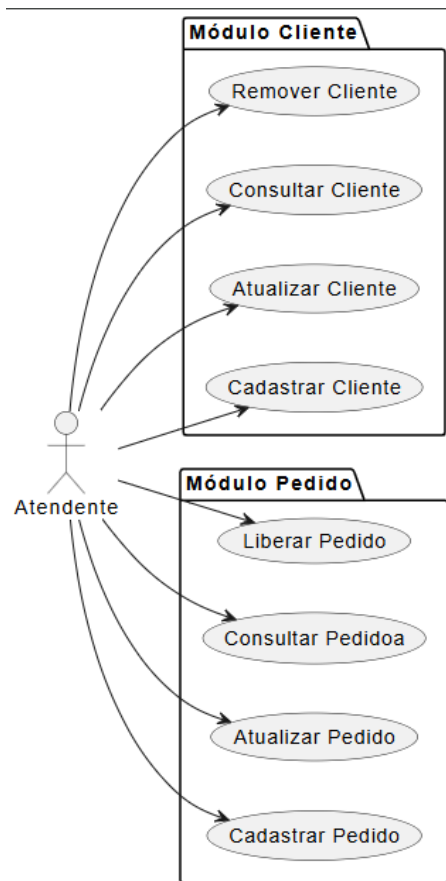
- **Módulo de Autenticação:**

- Este diagrama ilustra o fluxo de autenticação no sistema, desde o acesso inicial pelo usuário não logado até o redirecionamento para o menu de administrador ou cliente, após a autenticação das credenciais.



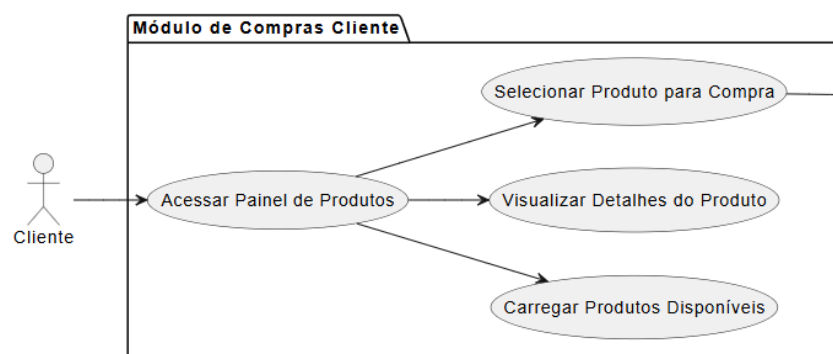
- **Módulo Cliente:**

- Este diagrama de caso de uso mostra as interações do "Atendente" com o módulo de Cliente, que inclui operações de gerenciamento como Remover, Consultar, Atualizar e Cadastrar clientes.



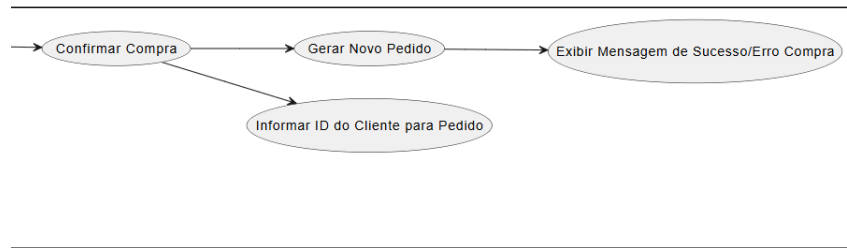
- **Módulo de Compras Cliente:**

- Este diagrama detalha as ações que um "Cliente" pode realizar dentro do módulo de compras, como acessar o painel de produtos, selecionar produtos para compra, visualizar detalhes e carregar produtos disponíveis.



- **Fluxo de Confirmação de Compra (Cliente):**

- Este diagrama se concentra no fluxo de confirmação de uma compra realizada pelo cliente, incluindo as etapas de confirmação, geração de novo pedido, informação do ID do cliente e exibição da mensagem de sucesso ou erro.



- Ator: Cliente
 - Realizar Login: O cliente tenta acessar o sistema fornecendo credenciais.
 - Visualizar Produtos: O cliente pode ver a lista de produtos disponíveis.
 - Fazer Pedido: O cliente pode selecionar produtos e finalizar uma compra.
 - Visualizar Histórico de Pedidos: O cliente pode consultar seus pedidos anteriores.
- Ator: Funcionário (Admin/Gerente/Caixa/Estoquista/Vendedor)
 - Realizar Login: O funcionário tenta acessar o sistema com suas credenciais.
 - Gerenciar Categorias: Inclui adicionar, visualizar, editar e excluir categorias de produtos (CRUD).
 - Gerenciar Produtos: Inclui adicionar, visualizar, editar e excluir informações de produtos (CRUD).
 - Gerenciar Estoque: Acompanhar entradas, saídas e ajustar quantidades de produtos.
 - Gerenciar Clientes: Inclui adicionar, visualizar, editar e excluir registros de clientes (CRUD).
 - Gerenciar Funcionários: Inclui adicionar, visualizar, editar e excluir registros de outros funcionários (CRUD, possivelmente restrito a admins).
 - Gerenciar Pedidos: Visualizar, atualizar status e processar pedidos.
 - Gerenciar Pagamentos: Registrar e consultar pagamentos.
 - Gerenciar Logística de Pedidos: Registrar informações de envio e recebimento.
 -

5. Partes Implementadas no Código

A estrutura do projeto Java LojaRoupaApp é organizada em pacotes para modularizar as diferentes responsabilidades do sistema, seguindo princípios de POO.

Estrutura de Pacotes:

- **com.lojaroupa.conexao:** Contém a classe utilitária para a conexão com o banco de dados (ConexaoDB).
- **com.lojaroupa.dao:** Contém as classes Data Access Object (DAO), responsáveis pela comunicação direta com o banco de dados para realizar as operações CRUD nas entidades.
- **com.lojaroupa.modelo:** (Pacote, onde as classes modelo Categoria, Cliente, Estoque, Funcionario, ItemPedido, LogisticaPedido, MovimentacaoEstoque, NivelUsuario, Pagamento, Pedido, Produto deveriam estar, pois elas representam as entidades de negócio)
- **com.lojaroupa.gui:** Contém as classes de interface gráfica (JFrame), como LoginFrame, MenuPrincipal e as telas de CRUD.

Classes modelo (em com.lojaroupa.conexao - idealmente com.lojaroupa.modelo):

Essas classes representam as entidades do banco de dados no lado da aplicação Java, contendo atributos que correspondem às colunas das tabelas, além de construtores, getters e setters.

- **Categorias.java:** Representa a tabela categorias.
 - Atributos: idCategoria, descricao.
- **Cliente.java:** Representa a tabela clientes.
 - Atributos: idCliente, nome, cpf, email, telefone, endereco, cep.
- **Estoque.java:** Representa a tabela estoque.
 - Atributos: idEstoque, idProduto, dtEntrada, quantidade, nfCompra, precoCompra, precoVenda, qtdVendida.
- **Funcionario.java:** Representa a tabela funcionarios.
 - Atributos: idFuncionario, email, idNivelUsuario, nome, cpf, endereco, cep, telefone, ativo.
- **ItemPedido.java:** Representa a tabela itenspedido.
 - Atributos: idItemPedido, idPedido, ordem, idEstoque, qtdItem, dtDevolucao, motivoDevolucao.
- **LogisticaPedido.java:** Representa a tabela logisticapedidos.
 - Atributos: idLogistica, idPedido, dtEnvio, dtRecebimento, rastreioFrete.
- **MovimentacaoEstoque.java:** Representa a tabela movimentacaoestoque.
 - Atributos: idMovimentacao, idEstoque, tipoMovimentacao, quantidade, dataMovimentacao, observacao.
- **NivelUsuario.java:** Representa a tabela nivelusuarios.
 - Atributos: idNivelUsuario, nivel.
- **Pagamento.java:** Representa a tabela pagamentos.
 - Atributos: idPagamento, idPedido, metodoPagamento, statusPagamento, valorPago, dtPagamento, notaFiscal, dtNotaFiscal.

- **Pedido.java:** Representa a tabela pedidos.
 - Atributos: idPedido, idFuncionario, idCliente, idVendedor, dtPedido, tipoFrete, entregaEndereco, entregaCEP, entregaTelefone, dtDevolucao, motivoDevolucao.
- **Produto.java:** Representa a tabela produtos.
 - Atributos: idProduto, nome, marca, modelo, idCategoria, descricao, unidadeMedida, cor, estoqueMinimo, tamanho, preco.

Classe de Conexão:

- **ConexaoDB.java:**
 - Localização: com.lojaroupa.conexao
 - Responsabilidade: Estabelecer e fechar a conexão com o banco de dados MySQL.
 - Parâmetros de Conexão: URL (jdbc:mysql://localhost:3306/lojaroupa?...), usuário (root), e senha (atualmente "1234").
 - Métodos:
 - getConnection(): Retorna um objeto Connection para interagir com o banco. Inclui tratamento de erros e mensagens de depuração.
 - closeConnection(Connection connection): Fecha a conexão com o banco de dados de forma segura.

Classes DAO (Data Access Object):

Essas classes encapsulam a lógica de acesso aos dados para cada entidade, implementando os métodos CRUD.

- **AuthDAO.java:**
 - Localização: com.lojaroupa.dao
 - Responsabilidade: Autenticar usuários (funcionarios e clientes) no sistema.
 - Métodos:
 - autenticarUsuario(String email, String senha): Consulta as tabelas funcionarios e clientes para verificar as credenciais. Retorna "admin" ou "cliente" se a autenticação for bem-sucedida, caso contrário, null.
- **CategoriaDAO.java:**
 - Localização: com.lojaroupa.dao
 - Responsabilidade: Realizar operações CRUD na tabela categorias.
 - Métodos:
 - inserir(Categoria categoria): Implementa a operação CREATE (INSERT) para novas categorias.

- `listarTodos()`: Implementa a operação READ (SELECT ALL) para obter todas as categorias.

Classe Principal da Aplicação:

- **MainApp.java:**
 - Localização: `com.lojaroupa.conexao` (idealmente `com.lojaroupa.app` ou similar)
 - Responsabilidade: Ponto de entrada da aplicação.
 - Método `main`: Utiliza `SwingUtilities.invokeLater` para garantir que a interface gráfica seja inicializada na Event Dispatch Thread (EDT), conforme boas práticas do Swing. Cria e exibe a `LoginFrame`.

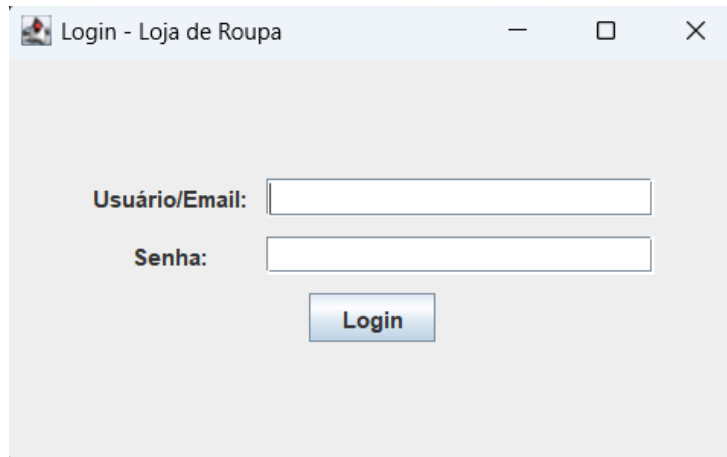
6. Construção das Telas (Interfaces Gráficas)

O sistema **LojaRoupaApp** utiliza a biblioteca Swing para a construção de suas interfaces gráficas, oferecendo uma experiência de usuário intuitiva para o gerenciamento da loja de roupas. As telas implementadas incluem a autenticação de usuário e a gestão de categorias, produtos, clientes e funcionários.

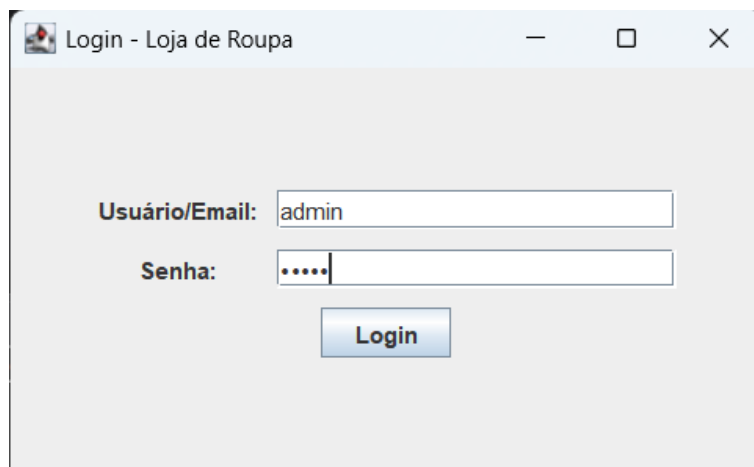
6.1. Tela de Login (`LoginFrame`)

A `LoginFrame` é a primeira tela que o usuário visualiza ao iniciar a aplicação. Ela é responsável por coletar as credenciais do usuário (e-mail e senha) e autenticá-las no sistema.

- **Design:** A tela possui um layout limpo com campos de texto para "E-mail" e "Senha", e um botão "Entrar". O título da janela é "Login - Loja de Roupas".
- **Funcionalidade:** Ao clicar em "Entrar", o sistema valida as credenciais contra as tabelas `funcionarios` e `clientes` no banco de dados. Em caso de sucesso, a tela de login é fechada e a `MenuPrincipal` é exibida. Caso contrário, uma mensagem de erro (não visível nas imagens, mas esperada) seria mostrada.
 - **Tela de Login (Vazia):** Esta imagem ilustra a tela de login com os campos 'E-mail' e 'Senha' vazios, exibindo o estado inicial para autenticação.



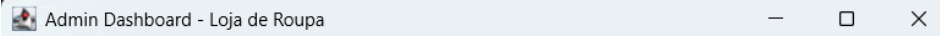
- **Tela de Login (Preenchida):** Esta imagem mostra a tela de login com o campo 'E-mail' preenchido como 'admin' e o campo 'Senha' preenchido (mas oculto por caracteres de senha), pronta para o acesso.



6.2. Tela de Menu Principal (MenuPrincipal)

Após um login bem-sucedido, a MenuPrincipal é carregada, servindo como o hub central do sistema. A partir dela, o usuário pode navegar para as diferentes funcionalidades de gerenciamento.

- **Design:** A tela apresenta uma barra de menu (JMenuBar) com opções como "Cadastro", "Pedidos", "Financeiro", "Relatórios" e "Sair". Há uma área de conteúdo principal que pode ser preenchida dinamicamente.
- **Funcionalidade:** O menu "Cadastro" oferece sub-opções para acessar as telas de CRUD de diferentes entidades, como Categorias, Produtos, Clientes e Funcionários.
 - **Tela de Menu Principal (Visão Geral):** Esta imagem exibe a tela principal do administrador, após login bem-sucedido, mostrando a barra de menus superior sem nenhuma opção expandida.



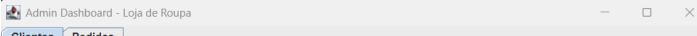
Admin Dashboard - Loja de Roupa

Cientes Pedidos

ID	Nome	CPF	Email	Telefone	Endereço	CEP
1	Ana Paula Sou...	444.444.444-44	ana.souza@e...	(11)91234-5678	Rua D, 101	04000-000
2	Carlos Lima	555.555.555-55	carlos.lima@e...	(11)92345-6789	Av. E, 202	05000-000
5	claudio castro	874.654.159-65	claudio.castro...	4564897464	AV.Luz, 978	97878-55
3	Cliente Teste		cli			
4	Lucas Alves F...	890.789.354-44	lucas.alves@g...	97907749203	Rua dr. henriq...	9077-599

Carregar Clientes Novo Cliente Excluir Cliente Selecionado

- **Tela de Cadastro de Clientes (Visão Geral):** Esta imagem mostra a interface onde você pode cadastrar e gerenciar as informações dos clientes, incluindo os campos para preencher os dados e os botões de ação.



Admin Dashboard - Loja de Roupa

Cientes Pedidos

ID	Nome	CPF	Email	Telefone	Endereço	CEP
1	Ana Paula Souza	444.444.444-44	ana.souza@email.c	(11)91234-5678	Rua D, 101	04000-000
2	Carlos Lima	555.555.555-55	carlos.lima@email...	(11)92345-6789	Av. E, 202	05000-000
5	claudio castro	874.654.159-65	claudio.castro@gm	4564897464	AV.Luz, 978	97878-55
3	Cliente Teste		cli			
4	Lucas Alves Ferreira	890.789.354-44	lucas.alves@gmail	97907749203	Rua dr. henrique go	9077-599

Novo Cliente

Nome: Lucas Henrique silva

CPF: 789.456.349.55

Email: lucas.henrique@gmail.com

Telefone: 789564698

Endereço: Rua D, 90

CEP: 8794564-777

OK Cancel

Carregar Clientes Novo Cliente Excluir Cliente Selecionado

6.3. Tela de Gerenciamento de Categorias (CRUD de Categorias)

Esta tela é um exemplo de um módulo CRUD completo, permitindo o gerenciamento das categorias de produtos da loja.

- **Design:** A tela é composta por uma tabela para exibir as categorias existentes, campos de texto para "ID da Categoria" (provavelmente somente leitura ou desabilitado para novas entradas, se o ID for auto-incrementado) e "Descrição", e botões para as operações CRUD (Cadastrar, Alterar, Excluir, Sair).
- **Funcionalidade:**
 - **Listagem:** Exibe todas as categorias cadastradas na tabela.
 - **Cadastrar:** Permite inserir uma nova categoria informando a descrição.

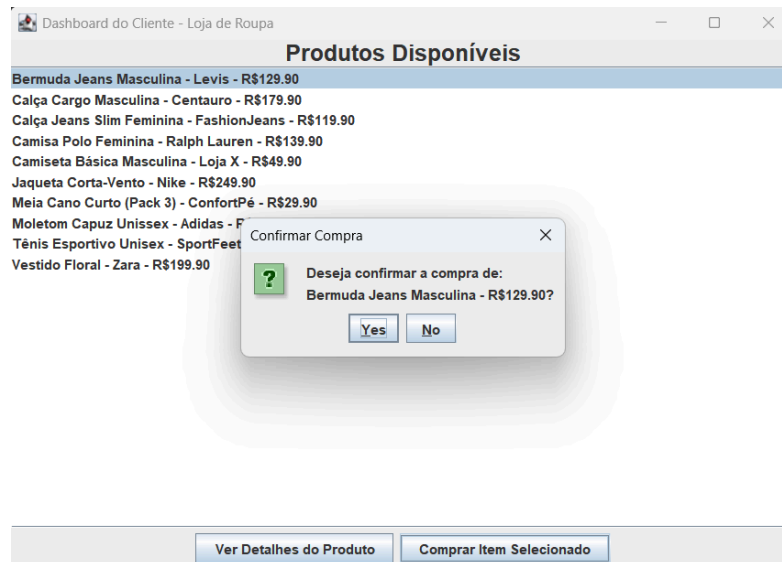
- **Alterar:** Permite modificar a descrição de uma categoria existente selecionada na tabela.
- **Excluir:** Permite remover uma categoria selecionada da tabela.
- **Sair:** Fecha a tela de gerenciamento de categorias, retornando ao Menu Principal.
- **Tela de Gerenciamento de Categorias (Visão Geral):** Esta imagem mostra a tela de gerenciamento de Categorias, exibindo uma tabela vazia e os campos 'ID da Categoria' e 'Descrição' aguardando entrada ou seleção.



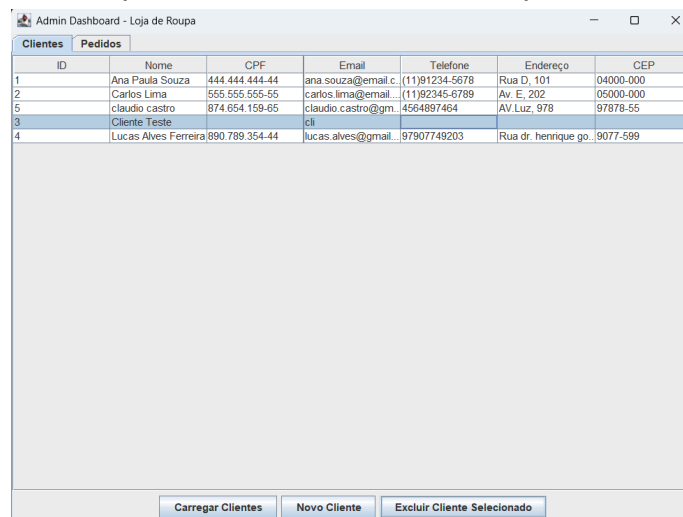
- **Tela de Gerenciamento de Produtos (Visão Geral):** Esta imagem apresenta a interface para cadastrar e gerenciar produtos, mostrando todos os campos de entrada, a tabela de listagem e os botões de ação disponíveis.



- **Tela de Gerenciamento de Produtos (Status Ativo/Inativo):** Esta imagem destaca a seção de detalhes de um produto, com os campos de texto preenchidos e a caixa de seleção 'Ativo' visivelmente desmarcada, indicando o status atual do item.



- **Tela de Gerenciamento de Produtos (Mensagem de Sucesso):** Esta imagem destaca a tela de produtos com uma notificação de "Sucesso!" no rodapé, indicando a conclusão positiva de uma ação no sistema.



7. Funcionalidades Mínimas Implementadas

- **Dois Módulos CRUD Completos:** O módulo de **Categorias** é o mais claramente implementado, com interface gráfica dedicada para as operações de Cadastro (Create), Listar (Read), Alterar (Update) e Excluir (Delete). As imagens demonstram a interface para todas essas operações.

- **Tela de Login:** A LoginFrame está implementada e funciona como o ponto de entrada da aplicação, com campos para e-mail e senha.
- **Tela de Menu Principal:** A MenuPrincipal é apresentada após o login bem-sucedido, fornecendo opções de navegação para as diferentes funcionalidades, como os cadastros de categorias, produtos, clientes e funcionários.
- **No Mínimo 2 Telas de CRUD:** A tela de gerenciamento de **Categorias** é uma tela de CRUD completa. As opções no menu principal sugerem que telas similares para **Produtos**, **Clientes** e **Funcionários** também existirão, o que atenderia facilmente ao requisito de "no mínimo 2 telas de CRUD" se implementadas.