

1. Package: principal

Class: Main

- **Descrição:** Método Main responsável por iniciar o sistema.

2. Package: telas

Class: TelaInicial

- **Método:** `chamarTelaMenuInicial()`
- **Descrição:** Inicia a tela inicial com todos os atributos necessários como JFrame, JPanel, JLabel, JTextField e JButton. Utiliza GridLayout, BufferedImage, ImageIcon e BorderLayout para melhorar o design. Adiciona o frame ao painel e chama a classe `ControllerTelaInicial`. Esta tela recebe a opção do usuário para realizar o cadastro de correntista.

Class: TelaSecundaria

- **Método:** `chamarTelaMenuSecundario()`
- **Descrição:** Configura a tela secundária com JFrame, JPanel, JLabel, JTextField e JButton, além de GridLayout e BufferedImage. Adiciona o frame ao painel e chama a classe `ControllerTelaSecundaria`. Esta tela permite ao usuário Cadastrar, Listar, Editar e Deletar correntistas, além de encerrar o programa e voltar à tela anterior.

Class: TelaCadastroCorrentista

- **Método:** `chamarTelaCadastroCorrentista()`
- **Descrição:** Configura a tela de cadastro com JFrame, JPanel, JLabel, JTextField e JButton, além de GridLayout e BufferedImage. Adiciona o frame ao painel e chama a classe `ControllerTelaCadastroCorrentista`. Contém um método `Override actionPerformed` que preenche automaticamente campos relacionados ao endereço ao digitar o CEP. Esta tela coleta os dados de cadastro e possui um botão para voltar à tela anterior.

Class: TelaEditarCorrentista

- **Método:** `ControllerTelaEditarCorrentista()`
- **Descrição:** Configura a tela de edição com JFrame, JPanel, JLabel, JTextField e JButton, além de GridLayout e BufferedImage. Adiciona o frame ao painel e chama a classe `ControllerTelaEditarCorrentista`. Contém um método `Override actionPerformed` que preenche automaticamente campos relacionados ao endereço ao digitar o CEP. Esta

tela coleta novos dados do usuário e possui um botão para voltar à tela anterior.

3. Package: controller

Class: ControllerTelaInicial

- **Descrição:** Controlador da tela inicial, adiciona atributos JFrame, JTextField e JButton para o construtor. Contém método `Override actionPerformed` para dar funcionalidade às opções inputadas pelo usuário, utilizando if/else e switch/case. Direciona o usuário ao menu secundário ou encerra o programa.

Class: ControllerTelaSecundaria

- **Descrição:** Controlador da tela secundária, adiciona atributos JFrame, JTextField e JButton para o construtor. Contém método `Override actionPerformed` para dar funcionalidade às opções inputadas pelo usuário, utilizando if/else, switch/case e try/catch. Direciona o usuário para a tela de cadastro, lista de correntistas, edição ou exclusão, ou encerra o programa.

Class: ControllerTelaCadastroCorrentista

- **Descrição:** Controlador da tela de cadastro, adiciona atributos JFrame, JTextField e JButton para o construtor. Contém método `Override actionPerformed` para salvar no banco de dados as informações inputadas pelo usuário ao clicar no botão 'Enviar', usando if/else. Exibe mensagem de sucesso e limpa os campos.

Class: ControllerTelaEditarCorrentista

- **Descrição:** Controlador da tela de edição, adiciona atributos JFrame, JTextField e JButton para o construtor. Contém método `Override actionPerformed` para salvar no banco de dados as novas informações inputadas pelo usuário ao clicar no botão 'Enviar', usando if/else. Exibe mensagem de sucesso e fecha a tela de edição.

4. Package: entidade

Class: Correntista

- **Descrição:** Classe abstrata com atributos como Nome, CPF, CEP, Logradouro, Bairro, Cidade, UF, Email, Quantidade de transações, Valor de anuidade. Contém métodos para calcular limite de saque e limite de crédito, dependendo do tipo de correntista. Possui métodos getter e setter para cada atributo.

Class: CorrentistaBasico | CorrentistaPremium

- **Descrição:** Herda de **Correntista**, reutilizando todos os atributos e métodos da classe base. **CorrentistaBasico** calcula o Limite de Saque e **CorrentistaPremium** calcula o Limite de Crédito.

Class: Cep

- **Descrição:** Representa um endereço no formato do código postal brasileiro (CEP). Contém atributos relacionados ao endereço, além de métodos getter e setter. Facilita a manipulação e o acesso aos dados de endereço.

5. Package: service

Class: BuscarCep

- **Descrição:** Permite buscar informações de endereço a partir de um CEP, consultando um serviço web (viacep.com.br) e convertendo a resposta JSON em um objeto **Cep**.

6. Package: persistencia

Class: FabricaConexao

- **Descrição:** Responsável por estabelecer uma conexão com um banco de dados MySQL. Encapsula a lógica de conexão, incluindo tratamento de exceções e gestão de credenciais e URL do banco de dados.

Class: DaoCorrentista

- **Descrição:** Realiza operações de CRUD (Create, Read, Update, Delete) no banco de dados para a entidade **Correntista**. Utiliza a **FabricaConexao** para gerenciar conexões e prepara comandos SQL. Trata exceções e garante fechamento adequado dos recursos de conexão e comando SQL.

Class: DaoCorrentistaBasico | DaoCorrentistaPremium

- **Descrição:** Realizam operações de salvamento e recuperação de correntistas no banco de dados. Utilizam a **FabricaConexao** para gerenciar conexões e executam comandos SQL. Tratam exceções e garantem fechamento adequado dos recursos de conexão e comandos SQL.

7. Package: repositorio

Class: CorrentistaBasicoRepositorio

- **Descrição:** Interface que define um contrato para operações de persistência de correntistas básicos. Declara métodos para salvar um correntista básico e listar todos os correntistas básicos do repositório.

Class: CorrentistaBasicoRepositorioImp

- **Descrição:** Implementação concreta da interface `CorrentistaBasicoRepositorio`. Fornece a lógica específica para as operações de persistência definidas pela interface.

Class: CorrentistaPremiumRepositorio

- **Descrição:** Interface que define um contrato para operações de persistência e recuperação de dados relacionados a correntistas premium. Declara métodos para salvar um `CorrentistaPremium` e listar todos os `CorrentistaPremium` do repositório.

Class: CorrentistaPremiumRepositorioImp

- **Descrição:** Implementação concreta da interface `CorrentistaPremiumRepositorio`. Fornece a lógica específica para as operações de persistência definidas pela interface.

8. Package: arquivo

Class: ManipuladorArquivo

- **Descrição:** Encapsula a lógica para criar um arquivo PDF detalhando informações de correntistas, oferecendo uma solução robusta para geração e visualização de relatórios no formato PDF.

Class: RodapeTemplate

- **Descrição:** Classe auxiliar que estende `PdfPageEventHelper` do iText, uma biblioteca Java para manipulação de arquivos PDF. Adiciona um rodapé padronizado em todas as páginas do documento PDF gerado.