

Capítulo 1: Introdução ao Sistema Bancário

- **1.1. Visão Geral**
 - Apresentação do sistema bancário desenvolvido em TypeScript.
 - Objetivo do sistema e as principais funcionalidades.
-

Capítulo 2: Classe `ContaBancaria`

- **2.1. Descrição da Classe**
 - Explicação sobre o que a classe `ContaBancaria` representa no sistema.
 - Propriedades principais: `Id`, `Nome`, `Email`, `Senha`, `Saldo`, `Histórico`.
- **2.2. Método `GerarExtrato`**
 - **Descrição:** Este método gera um extrato detalhado da conta bancária.
 - **Funcionamento:** Ele percorre o histórico de transações e compila as informações em um formato de relatório.
 - **Exemplo de Uso:** `conta.GerarExtrato();`
- **2.3. Método `SaldoAtual`**
 - **Descrição:** Retorna o saldo atual da conta.
 - **Funcionamento:** Acessa a propriedade `Saldo` da classe e retorna.
 - **Exemplo de Uso:** `conta.SaldoAtual();`

Capítulo 3: Classe `ContaCorrente`

- **3.1. Descrição da Classe**
 - Explicação sobre o que a classe `ContaCorrente` representa e como ela estende a `ContaBancaria`.
- **3.2. Método `DepositarCorrente`**
 - **Descrição:** Permite realizar depósitos na conta corrente.
 - **Funcionamento:** O método adiciona o `valor` depositado ao saldo da conta e registra a transação no histórico.
 - **Exemplo de Uso:** `conta.DepositarCorrente(500);`
- **3.3. Método `SacarCorrente`**
 - **Descrição:** Realiza o saque de dinheiro da conta corrente.
 - **Funcionamento:** O método verifica se há saldo suficiente e, se houver, deduz o valor solicitado e atualiza o histórico.
 - **Exemplo de Uso:** `conta.SacarCorrente(200);`
- **3.4. Método `TransferirCorrente`**
 - **Descrição:** Transfere fundos para outra conta bancária.
 - **Funcionamento:** O método deduz o valor da conta de origem e adiciona à conta de destino, registrando a transação em ambos os históricos.

- **Exemplo de Uso:** `conta.TransferirCorrente(300, outraConta);`

Capítulo 4: Classe `ContaPoupanca`

- **4.1. Descrição da Classe**

- Explicação sobre o que a classe `ContaPoupanca` representa e como ela estende a `ContaBancaria`.

- **4.2. Método `AplicarJuros`**

- **Descrição:** Aplica juros ao saldo da conta poupança.
- **Funcionamento:** O método calcula os juros com base em uma taxa pré-definida e incrementa o saldo da conta.
- **Exemplo de Uso:** `contaPoupanca.AplicarJuros();`

- **4.3. Método `DepositarPoupanca`**

- **Descrição:** Permite realizar depósitos na conta poupança.
- **Funcionamento:** Similar ao método de depósito na conta corrente, mas registrado sob o contexto da conta poupança.
- **Exemplo de Uso:** `contaPoupanca.DepositarPoupanca(1000);`

- **4.4. Método `SacarPoupanca`**

- **Descrição:** Realiza o saque de dinheiro da conta poupança.
- **Funcionamento:** Semelhante ao saque na conta corrente, com regras específicas para conta poupança.
- **Exemplo de Uso:** `contaPoupanca.SacarPoupanca(300);`

Capítulo 5: Funções Utilitárias

- **5.1. Função `criarConta`**

- **Descrição:** Função responsável por criar novas contas bancárias.
- **Funcionamento:** Solicita informações do usuário e retorna uma nova instância de `ContaCorrente` ou `ContaPoupanca`.

- **5.2. Função `EntrarNaConta`**

- **Descrição:** Função para autenticar o usuário e acessar sua conta existente.
- **Funcionamento:** Verifica o ID e a senha do usuário e retorna a conta correspondente.

Capítulo 6: Estrutura de Menus

- **6.1. Menu Inicial**

- **Descrição:** Explicação sobre o menu principal do sistema bancário.
- **Funcionamento:** Apresenta opções ao usuário, como criar conta, acessar informações, ou sair do sistema.
- **Exemplo:** Código do menu inicial.
- **6.2. Menu Bancário**
 - **Descrição:** Explicação sobre o menu dentro do sistema bancário.
 - **Funcionamento:** Oferece opções de operações bancárias como depósito, saque, transferência, etc.
 - **Exemplo:** Código do menu bancário.
-