**Título:** Sistema Bancário em JavaScript e Node.js

**Descrição do Projeto:**

Este documento descreve a implementação de um sistema bancário simples utilizando JavaScript e Node.js. A aplicação inclui funcionalidades para criar contas bancárias, depositar e sacar valores, e aplicar juros ou rendimento.

**Requisitos da Atividade:**

* **Configuração do Ambiente**
* **Implementação das Classes**
* **Interação com o Usuário**
* **Documentação**

**Código:**

**1. Código da Classe Conta**

javascript

Copiar código// conta.js  
class Conta {  
 constructor(titular, saldo = 0) {  
 this.titular = titular;  
 this.saldo = saldo;  
 }  
  
 depositar(valor) {  
 this.saldo += valor;  
 console.log(`Depósito de R$${valor} realizado. Saldo atual: R$${this.saldo}`);  
 }  
  
 sacar(valor) {  
 if (valor > this.saldo) {  
 console.log('Saldo insuficiente.');  
 } else {  
 this.saldo -= valor;  
 console.log(`Saque de R$${valor} realizado. Saldo atual: R$${this.saldo}`);  
 }  
 }  
  
 consultarSaldo() {  
 console.log(`Saldo atual: R$${this.saldo}`);  
 }  
}  
  
module.exports = Conta;

**2. Código da Classe ContaCorrente**

javascript

Copiar código

// contaCorrente.js  
const Conta = require('./conta');  
  
class ContaCorrente extends Conta {  
 constructor(titular, saldo, juros) {  
 super(titular, saldo);  
 this.juros = juros;  
 }  
  
 aplicarJuros() {  
 const aumento = this.saldo \* this.juros;  
 this.depositar(aumento);  
 console.log(`Juros aplicados. Saldo atual: R$${this.saldo}`);  
 }  
}  
  
module.exports = ContaCorrente;

**3. Código da Classe ContaPoupanca**

javascript

Copiar código

// contaPoupanca.js  
const Conta = require('./conta');  
  
class ContaPoupanca extends Conta {  
 constructor(titular, saldo, rendimento) {  
 super(titular, saldo);  
 this.rendimento = rendimento;  
 }  
  
 aplicarRendimento() {  
 const aumento = this.saldo \* this.rendimento;  
 this.depositar(aumento);  
 console.log(`Rendimento aplicado. Saldo atual: R$${this.saldo}`);  
 }  
}  
  
module.exports = ContaPoupanca;

**4. Código do Arquivo Principal (index.js)**

javascript

Copiar código

// index.js  
const readline = require('readline-sync');  
const ContaCorrente = require('./contaCorrente');  
const ContaPoupanca = require('./contaPoupanca');  
  
let contas = [];  
  
function criarConta() {  
 const tipo = readline.question('Qual tipo de conta deseja criar? (1 - Corrente, 2 - Poupança): ');  
 const titular = readline.question('Nome do titular: ');  
 const saldoInicial = parseFloat(readline.question('Saldo inicial: '));  
  
 if (tipo === '1') {  
 const juros = parseFloat(readline.question('Taxa de juros (%): ')) / 100;  
 const conta = new ContaCorrente(titular, saldoInicial, juros);  
 contas.push(conta);  
 console.log('Conta Corrente criada com sucesso!');  
 } else if (tipo === '2') {  
 const rendimento = parseFloat(readline.question('Taxa de rendimento (%): ')) / 100;  
 const conta = new ContaPoupanca(titular, saldoInicial, rendimento);  
 contas.push(conta);  
 console.log('Conta Poupança criada com sucesso!');  
 } else {  
 console.log('Tipo de conta inválido.');  
 }  
}  
  
function selecionarConta() {  
 const titular = readline.question('Nome do titular da conta: ');  
 const conta = contas.find(conta => conta.titular === titular);  
  
 if (!conta) {  
 console.log('Conta não encontrada.');  
 return null;  
 }  
  
 return conta;  
}  
  
function menu() {  
 while (true) {  
 console.log('\n1. Criar Conta');  
 console.log('2. Depositar');  
 console.log('3. Sacar');  
 console.log('4. Consultar Saldo');  
 console.log('5. Aplicar Juros/Rendimento');  
 console.log('6. Sair');  
  
 const opcao = readline.question('Escolha uma opção: ');  
  
 switch (opcao) {  
 case '1':  
 criarConta();  
 break;  
 case '2':  
 const contaDeposito = selecionarConta();  
 if (contaDeposito) {  
 const valor = parseFloat(readline.question('Valor do depósito: '));  
 contaDeposito.depositar(valor);  
 }  
 break;  
 case '3':  
 const contaSaque = selecionarConta();  
 if (contaSaque) {  
 const valor = parseFloat(readline.question('Valor do saque: '));  
 contaSaque.sacar(valor);  
 }  
 break;  
 case '4':  
 const contaConsulta = selecionarConta();  
 if (contaConsulta) {  
 contaConsulta.consultarSaldo();  
 }  
 break;  
 case '5':  
 const contaAplicar = selecionarConta();  
 if (contaAplicar instanceof ContaCorrente) {  
 contaAplicar.aplicarJuros();  
 } else if (contaAplicar instanceof ContaPoupanca) {  
 contaAplicar.aplicarRendimento();  
 } else {  
 console.log('Operação não permitida para este tipo de conta.');  
 }  
 break;  
 case '6':  
 console.log('Saindo...');  
 return;  
 default:  
 console.log('Opção inválida.');  
 }  
 }  
}  
  
menu();

**Como Executar a Aplicação**

1. **Clone o Repositório**:

bash

Copiar código

git clone <https://github.com/OthonSoares/App-de-Banco-.git>

1. **Instale as Dependências**:

Navegue até o diretório do projeto e instale as dependências:

bash

Copiar código

cd sistema-bancario  
npm install

1. **Execute a Aplicação**:

bash

Copiar código

node index.js

1. **Interaja com a Aplicação**:

Siga as instruções fornecidas no menu da aplicação para criar contas, depositar, sacar e consultar saldos.