**Index.js**

import {Cliente} from "./Cliente.js"

import {ContaCorrente} from "./ContaCorrente.js"

const cliente1 = new Cliente("Ricardo", 11122233309);

const cliente2 = new Cliente("Alice", 88822233309);

console.log(cliente2.cpf);

const contaCorrenteRicardo = new ContaCorrente(1001, cliente1);

contaCorrenteRicardo.depositar(500);

const conta2 = new ContaCorrente(102, cliente2);

let valor = 200;

contaCorrenteRicardo.transferir(valor, conta2);

console.log(ContaCorrente.numeroDeContas);

**Cliente.js**

export class Cliente {

nome;

\_cpf;

get cpf(){

return this.\_cpf;

}

constructor(nome, cpf){

this.nome = nome;

this.\_cpf = cpf;

}

}

**ContaCorrente.js**

import { Cliente } from "./Cliente.js";

export class ContaCorrente {

    static numeroDeContas = 0;

    agencia;

    \_cliente;

    //#saldo=0 campo privado https://github.com/tc39/proposal-class-fields#private-fields

    \_saldo = 0;

    set cliente(novoValor){

        if(novoValor instanceof Cliente){

            this.\_cliente = novoValor;

        }

    }

    get cliente(){

        return this.\_cliente;

    }

    get saldo(){

        return this.\_saldo;

    }

    constructor(agencia, cliente){

        this.agencia = agencia;

        this.cliente = cliente;

        ContaCorrente.numeroDeContas += 1;

    }

    sacar(valor){

        if (this.\_saldo >= valor){

            this.\_saldo -= valor;

            return valor;

        }

    }

    depositar(valor){

        if(valor <= 0){

            return;

        }

        this.\_saldo += valor;

    }

    transferir(valor, conta){

        const valorSacado = this.sacar(valor);

    }

}