IFPI - ADS
POO 2023.2
Kleberson
Prof Ely Miranda

Atividade 08 – Exceções

1. Throw; try/catch e parâmetro opcional

```
//q1. método 1(lançar exceção com throw Erorr)
function ehInteiro(valor:number){
    if (valor < 0){
        throw new Error('Valor negativo')
    }
    return true
}

//q1. método 2(lançar exceção com try/catch)
function divide(a:number, b:number){
    try{
        if (b == 0){
            throw new Error('Divisão por 0')
        }
        return a / b
    }
    catch(erro:any){
        console.log(erro.message)
    }
}

//q1. método 3(usar parâmetro opcional para tratar tipo de argumento)
function nome_ou_numero(nome?:string, num?: number){
    try{
        if (nome){
            return nome
        }
        if (num){
            return num
        }
        if (nome && num){
            return `Nome: ${nome}\nNúmero: ${num}`
        }else{
            throw new Error('Nenhum argumento passado')
        }
    }catch(erro:any){
        console.log(erro.message)
    }
}

//console.log(divide(4, 2));
//console.log(divide(2, 0));
divide(4, 2)
divide(2, 0)

console.log(nome_ou_numero('joao'))
console.log(nome_ou_numero('', 1))
```

```
console.log(nome_ou_numero('joao', 2))
nome_ou_numero()

ehInteiro(-1)
```

2. O primeiro é que caso seja encontrada a exceção, o programa será abruptamente encerrado. O segundo é para casos em que o programa não pode ser tão flexível, ele vai permitir continuar executando. O terceiro é no caso de chamada com omissão de argumento, que dependendo do caso não pode retornar o esperado.

4. Retornou uma exceção dizendo que o valor a transferir é acima do saldo da conta debitada

5. Sim, pois ele retornando uma exceção, gera um impasse nos demais métodos que chamem o que gerou a exceção. Desta forma, a exceção é propagada aos métodos que precisam do que foi gerado a exceção.