Relatório de Testes de funções com jest

**Funções de soma simples**

const soma1 = (positivo1 , positivo2) => {

    return positivo1 + positivo2

}

console.log(soma1(2, 3))

const soma2 = (positivo , negativo) => {

    return positivo + negativo

}

console.log(soma1(2, -3))

const soma3 = (negativo1 , negativo2) => {

    return negativo1 + negativo2

}

console.log(soma1(-2, -3))

describe('teste de somas', () => {

    it('deve somar dois positivos', () => {

        const esperado = 5;

        const retornado = soma1(2, 3);

        expect(retornado).toBe(esperado);

    })

    it('deve somar um positivo e um negativo', () => {

        const esperado = -1;

        const retornado = soma2(2, -3);

        expect(retornado).toBe(esperado);

    })

    it('deve somar dois negativos', () => {

        const esperado = -5;

        const retornado = soma3(-2, -3);

        expect(retornado).toBe(esperado);

    })

})

* Nos três testes os valores retornados se mostraram ser o mesmo do esperado, respectivamente: 5, -1 e -5

**Funções de subtração**

const subtracao1 = (positivo1 , positivo2) => {

    return positivo1 - positivo2

}

console.log(subtracao1(2, 3))

const subtracao2 = (positivo , negativo) => {

    return positivo - negativo

}

console.log(subtracao2(2, -3))

const subtracao3 = (negativo1 , negativo2) => {

    return negativo1 - negativo2

}

console.log(subtracao3(-2, -3))

describe('teste de subtrações', () => {

    it('deve subtrair dois positivos', () => {

        const esperado = -1;

        const retornado = subtracao1(2, 3);

        expect(retornado).toBe(esperado);

    })

    it('deve subtrair um positivo e um negativo', () => {

        const esperado = 5;

        const retornado = subtracao2(2, -3);

        expect(retornado).toBe(esperado);

    })

    it('deve subtrair dois negativos', () => {

        const esperado = 1;

        const retornado = subtracao3(-2, -3);

        expect(retornado).toBe(esperado);

    })

})

* Nos três testes os valores retornados se mostraram ser o mesmo do esperado, respectivamente: -1, 5 e 1
* Tive que ter cuidado aqui porque eu quase esqueci das regras subtração quando se trata de números negativos

**Funções para verificar se um número é par**

const P\_I\_positivo = (par , impar) => {

    return [par % 2 === 0, impar % 2 === 0]

}

console.log(P\_I\_positivo(4, 3))

const P\_I\_negativo = (par , impar) => {

    return [par % 2 === 0, impar % 2 === 0]

}

console.log(P\_I\_negativo(-4, -3))

const P\_I\_zero = (zero) => {

    return zero % 2 === 0

}

console.log(P\_I\_zero(0))

describe('par ou impar', () => {

    it('Os números positivos devem retornar par(true) e impar(false)', () => {

        const esperado = [true, false]

        const retornado = P\_I\_positivo(4, 3)

        expect(retornado).toEqual(esperado);

    })

    it('Os números negativos devem retornar par(true) e impar(false)', () => {

        const esperado = [true, false]

        const retornado = P\_I\_negativo(-4, -3)

        expect(retornado).toEqual(esperado);

    })

    it('O número zero deve retornar par(true)', () => {

        const esperado = true

        const retornado = P\_I\_zero(0)

        expect(retornado).toEqual(esperado);

    })

})

* Os três testes foram um pouco diferentes porque é necessário fazer um array para retornar múltiplos valores e mudar toBe para toEqual
* Os três testes foram um sucesso e retornaram os valores esperados:

[true, false], [true, false] e true