Atividade – Teste de Caixa Branca com Jest

Objetivo

Aplicar os conceitos de **teste de caixa branca**, com foco na análise de fluxo de controle, caminhos de execução e cobertura de testes.

Instruções Gerais

Para cada função abaixo:

1. **Leia e analise o código-fonte**.
2. **Desenhe o grafo de fluxo de controle** da função.
3. **Calcule a complexidade ciclomática**.
4. **Liste todos os caminhos lógicos independentes**.
5. **Implemente testes unitários com Jest** para cobrir cada caminho.
6. **Execute os testes com cobertura**:

```
npm run test:coverage nome_do_arquivo.test.js
```

7. **Garanta cobertura total** ou justifique os casos não testados.

Funções para Análise

1. `validaSenha(senha)`

```
function validaSenha(senha) {
  if (typeof senha !== 'string') return false;
  if (senha.length < 8) return false;
  if (!/[A-Za-z]/.test(senha)) return false;
  if (!/[0-9]/.test(senha)) return false;
  if (/\s/.test(senha)) return false;
  return true;
}
```

---

2. `classificaTemperatura(temp)`

```
function classificaTemperatura(temp) {
  if (temp < 15) return "Frio";
  else if (temp <= 25) return "Agradável";
  else return "Quente";
}
```

---

3. `calculaImposto(renda)`

```
function calculaImposto(renda) {
  if (typeof renda !== 'number' || renda < 0) return null;
  if (renda <= 2000) return 0;
  if (renda <= 5000) return renda * 0.15;
  return renda * 0.275;
}
```

---

4. `divideNumeros(a, b)`

```
function divideNumeros(a, b) {
  if (typeof a !== 'number' || typeof b !== 'number') throw new Error('Entrada
inválida');
  if (b === 0) throw new Error('Divisão por zero');
  return a / b;
}
```

---

5. `verificaIntervalo(n)`

```
function verificaIntervalo(n) {
  if (typeof n !== 'number') return false;
  return n >= 10 && n <= 20;
}
```
