

Projeto Final

Validação e Verificação de Sistemas

Professor Lucian José Gonçalves

1. Escreva um plano de teste no formato Markdown (.md) e versione no repositório do seu projeto.

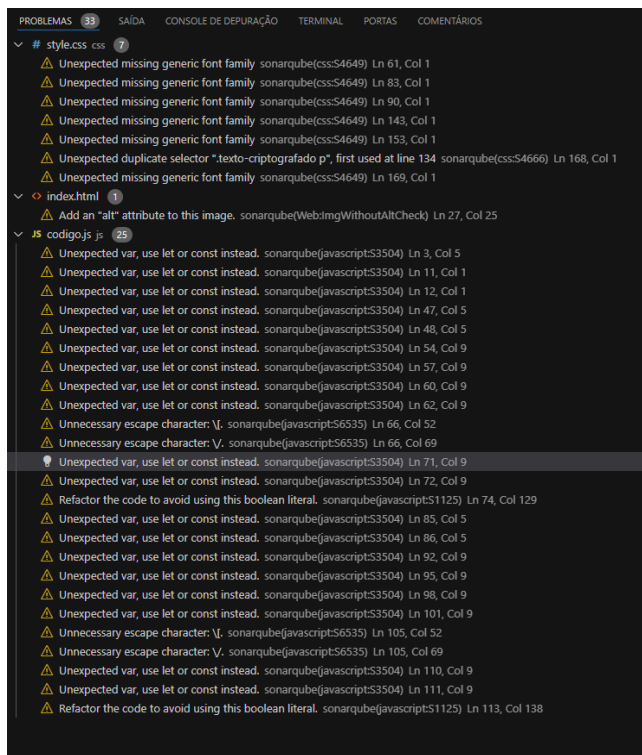
- **Github**, <https://github.com/joaobolasell/decode.git>

```
git init
git add .
git commit -m "Primeiro Commit"
git remote add origin
https://github.com/joaobolasell/decode.git
git push -u origin master

git add .
git commit -m "mensagem"
git push
```

2. Implemente uma verificação estática no projeto (checkstyle, pmd e sonarlint)

- SonarQube no Visual Studio Code



- **ESLint: Ferramenta de análise estática para JavaScript e TypeScript**
- Certifique-se de que você tem o Node.js instalado no projeto, Instale o ESLint e Inicie a configuração

```
npm install eslint -save-dev
npx eslint --init
```

- Execute o teste

```
npx eslint .\js\codigo_refatorado_para_testes.js --fix
npx eslint .
```

```
PS C:\Users\USER2\Desktop\decode> npx eslint .\js\codigo_refatorado_para_testes.js --fix
Warning: React version not specified in eslint-plugin-react settings. See https://github.com/
C:\Users\USER2\Desktop\decode\js\codigo_refatorado_para_testes.js
  1:10  error  'copiarTexto' is defined but never used  no-unused-vars
 43:1   error  'module' is not defined                  no-undef

✖ 2 problems (2 errors, 0 warnings)
```

3. Implemente um conjunto de testes unitários

- **Jest: Framework para testes em JavaScript**
- Certifique-se de que você tem o Node.js instalado no projeto e instale o Jest

```
npm init -y
npm install jest --save-dev
```

- Adicione um script no arquivo package.json para rodar o Jest

```
"scripts":{
  "test": "jest" }
```

- Isso permite executar os testes com o comando

```
npm test
```

- O arquivo com os testes unitários está na pasta

```
decode/___tests___/unit/codigo_refatorado_para_testes.js
```

```
PS C:\Users\USER2\Desktop\decode> npm test

> decode@1.0.0 test
> jest

FAIL tests_/unit/codigo_refatorado_para_testes.test.js
  Testes de criptografia
    ✓ Criptografa uma mensagem simples (3 ms)
    ✓ Criptografa texto com todas as vogais
    ✓ Criptografa texto com letras e números (4 ms)
    ✗ Criptografa texto com vogais repetidas (1 ms)
    ✗ Criptografa texto longo (4 ms)
    ✓ Criptografa texto vazio (1 ms)
  Testes de descriptografia
    ✓ Descriptografa uma mensagem simples (1 ms)
    ✓ Descriptografa texto com todas as vogais criptografadas
    ✓ Descriptografa texto com letras e números
    ✗ Descriptografa texto com vogais repetidas (1 ms)
    ✗ Descriptografa texto longo (2 ms)
    ✓ Descriptografa texto vazio (1 ms)
  Testes de casos extremos e validação
    ✓ Texto sem vogais (1 ms)
    ✗ Descriptografa texto parcialmente criptografado (1 ms)
  Testes de comportamento com entradas inválidas
    ✓ Criptografa texto com caracteres especiais permanece inalterado
    ✓ Descriptografa texto com caracteres especiais permanece inalterado (1 ms)

  ● Testes de criptografia > Criptografa texto com letras e números

    expect(received).toBe(expected) // Object.is equality

    Expected: "oberla123"
    Received: "oberla123"

    16 |     test("Criptografa texto com letras e números", () => {
    17 |       // Verifica se a função criptografa corretamente texto com números
    18 |       expect(criptografaTexto("ola123")).toBe("oberla123");
    19 |     });
    20 |
    21 |     test("Criptografa texto com vogais repetidas", () => {

    at Object.toBe (tests_/unit/codigo_refatorado_para_testes.test.js:18:44)

  ● Testes de criptografia > Criptografa texto com vogais repetidas

    expect(received).toBe(expected) // Object.is equality

    Expected: "baibaibaiba"
    Received: "baibaiba"

    59 |   });
    60 |
    61 |   test("Descriptografa texto longo", () => {
    62 |
    63 |     at Object.toBe (tests_/unit/codigo_refatorado_para_testes.test.js:58:52)
    64 |
    65 |     ● Testes de descriptografia > Descriptografa texto longo
    66 |
    67 |     expect(received).toBe(expected) // Object.is equality
    68 |
    69 |     Expected: "este é um teste para verificar como o sistema funciona com textos longos"
    70 |     Received: "este é um testeteste para verificar como o sistema funciona com o textetober longoberbos"

    63 |     const texto = "enterstenter é ufantentententer painersi verimesrificaien
    64 |     ter comoberober ober sistentermas funcioberna comober tentextoberber loberngoberberbos";
    65 |     const esperado = "este é um teste para verificar como o sistema funciona
    66 |     com textos longos";
    67 |     expect(descriptografaTexto(texto)).toBe(esperado);
    68 |   });
    69 |
    70 |   test("Descriptografa texto vazio", () => {
    71 |
    72 |     at Object.toBe (tests_/unit/codigo_refatorado_para_testes.test.js:65:44)
    73 |
    74 |     ● Testes de casos extremos e validação > Descriptografa texto parcialmente criptografado
    75 |
    76 |     expect(received).toBe(expected) // Object.is equality
    77 |
    78 |     Expected: "ola"
    79 |     Received: "ola"

    82 |     test("Descriptografa texto parcialmente criptografado", () => {
    83 |       // Verifica se a função de descriptografia consegue lidar com um texto p
    84 |       expect(descriptografaTexto("oberlaimes")).toBe("olai");
    85 |     });
    86 |
    87 |   });

    at Object.toBe (tests_/unit/codigo_refatorado_para_testes.test.js:84:51)

Test Suites: 1 failed, 1 total
Tests: 6 failed, 10 passed, 16 total
Snapshots: 0 total
Time: 0.661 s, estimated 1 s
Ran all test suites.
PS C:\Users\USER2\Desktop\decode>
```

4. Implemente testes de componentes (API) e/ou testes de sistema (Selenium ou Cypress)

- **Cypress:** Teste de Sistema
- Instale e Inicie para criar a estrutura básica

```
npm install cypress -save-dev
npx cypress open
```

- Crie o arquivo de testes e execute.

```
../cypress/e2e/decode.cy.js
npx cypress run
```

●

5. Configure um ambiente de integração contínua de sua escolha, por exemplo, Github Actions

- **Github Actions:**

```
.github/workflows/ci.yml
```