

Relatório de Refatoração de Testes Automatizados

Disciplina: Testes de Software

Trabalho: Test Smell

Aluno: Rafael Marques Radieddine

Matrícula: 750376

1. Análise de Test Smells

Durante a análise da suíte de testes original (userService.smelly.test.js), foram identificados três

principais test smells.

1. Conditional Test Logic (Expect Condicional)

- Presença de condicionais dentro de testes (if, for).
- Risco: caminhos múltiplos mascaram falhas.

2. Exception Handling sem Verificação Adequada

- Uso de try/catch sem garantir falha caso a exceção não ocorra.
- Risco: falsos positivos.

3. Dependência Frágil de Formatação (Fragile Test)

- Testes que dependem rigidamente de strings formatadas.
- Risco: falhas falsas devido a pequenas mudanças cosméticas.

2. Processo de Refatoração

O arquivo foi refatorado de userService.smelly.test.js para userService.clean.test.js.

Antes (Smelly):

```
// O teste tem um loop e um if, tornando-o complexo e menos claro.
for (const user of todosOsUsuarios) {
  const resultado = userService.deactivateUser(user.id);
  if (!user.isAdmin) {
    // Este expect só roda para o usuário comum.
    expect(resultado).toBe(true);
    const usuarioAtualizado = userService.getUserById(user.id);
    expect(usuarioAtualizado.status).toBe('inativo');
  } else {
    // E este só roda para o admin.
    expect(resultado).toBe(false);
  }
}
});
```

Depois (Clean):

```
describe("deactivateUser", () => {  
  test("deve desativar um usuário comum com sucesso", () => {  
    // Arrange  
    const usuario = userService.createUser("Comum", "comum@teste.com", 30);  
  
    // Act  
    const resultado = userService.deactivateUser(usuario.id);  
  
    // Assert  
    expect(resultado).toBe(true);  
    expect(userService.getUserById(usuario.id).status).toBe("inativo");  
  });  
});
```

Decisões:

- Separação de cenários em testes independentes;
- Uso de expect().toThrow() para exceções;
- Expressões regulares para validações mais flexíveis.

3. Relatório da Ferramenta

Primeira execução do ESLint:

E:\Testes\test-smelly\test\userService.smelly.test.js

44:9 error Avoid calling expect conditionally jest/no-conditional-expect

77:3 warning Tests should not be skipped jest/no-disabled-tests

A ferramenta identificou automaticamente práticas incorretas e acelerou o processo de refatoração.

Após as correções, não foram encontrados erros.

Evidências:

```
PS E:\Testes\test-smelly> npm test test/userService.smelly.test.js

> test-smells-lab@1.0.0 test
> jest test/userService.smelly.test.js

PASS test/userService.smelly.test.js
UserService - Suite de Testes com Smells
  ✓ deve criar e buscar um usuário corretamente (3 ms)
  ✓ deve desativar usuários se eles não forem administradores (1 ms)
  ✓ deve gerar um relatório de usuários formatado
  ✓ deve falhar ao criar usuário menor de idade
  ○ skipped deve retornar uma lista vazia quando não há usuários

Test Suites: 1 passed, 1 total
Tests:      1 skipped, 4 passed, 5 total
Snapshots:  0 total
Time:       0.306 s, estimated 1 s
Ran all test suites matching /test/userService.smelly.test.js/i.
PS E:\Testes\test-smelly> npx eslint test/userService.clean.test.js
PS E:\Testes\test-smelly> npm test test/userService.smelly.test.js

> test-smells-lab@1.0.0 test
> jest test/userService.smelly.test.js

PASS test/userService.smelly.test.js
UserService - Suite de Testes com Smells
  ✓ deve criar e buscar um usuário corretamente (2 ms)
  ✓ deve desativar usuários se eles não forem administradores (1 ms)
  ✓ deve gerar um relatório de usuários formatado (1 ms)
  ✓ deve falhar ao criar usuário menor de idade
  ○ skipped deve retornar uma lista vazia quando não há usuários

Test Suites: 1 passed, 1 total
Tests:      1 skipped, 4 passed, 5 total
Snapshots:  0 total
Time:       0.333 s, estimated 1 s
Ran all test suites matching /test/userService.smelly.test.js/i.
PS E:\Testes\test-smelly> npx eslint test/userService.smelly.test.js

E:\Testes\test-smelly\test\userService.smelly.test.js
44:9  error  Avoid calling 'expect' conditionally  jest/no-conditional-expect
46:9  error  Avoid calling 'expect' conditionally  jest/no-conditional-expect
49:9  error  Avoid calling 'expect' conditionally  jest/no-conditional-expect
73:7  error  Avoid calling 'expect' conditionally  jest/no-conditional-expect
77:3  warning Tests should not be skipped  jest/no-disabled-tests

X 5 problems (4 errors, 1 warning)

PS E:\Testes\test-smelly> npx eslint test/userService.clean.test.js
PS E:\Testes\test-smelly>
```

4. Conclusão

A refatoração demonstrou que testes limpos aumentam a confiabilidade e a clareza do código. Ferramentas como o ESLint garantem padrões e previnem maus hábitos de codificação. Testes bem estruturados são essenciais para a sustentabilidade de projetos de software.