Nome: Luiz Eduardo Mendonça Marrano
Nome da Atividade: Diferença Entre Testes Unitários e Funcionais

---

Link do repositório:
https://github.com/LuizEduardoMM/Teste_Software_2024_Marrano_Eduardo/tree/master

1. Pergunta Selecionada:

Pergunta: What is the difference between unit tests and functional tests?
Descrição: Esta pergunta discute as diferenças entre testes unitários e testes funcionais, fornecendo analogias e explicações sobre cada tipo de teste.

2. Reproduzindo o Problema e Adotando a Resolução:

Problema Reproduzido:

Para entender as diferenças entre testes unitários e testes funcionais, é importante observar o seguinte:

- **Testes Unitários:** Focados em verificar partes específicas do código (unidades), como funções ou métodos, para garantir que cada um funcione como esperado isoladamente. São realizados em nível de código e não consideram a interação com outras partes do sistema.
- **Testes Funcionais:** Avaliam o sistema como um todo, verificando se os requisitos funcionais são atendidos. São realizados em nível de sistema e garantem que o software funcione conforme esperado do ponto de vista do usuário final.

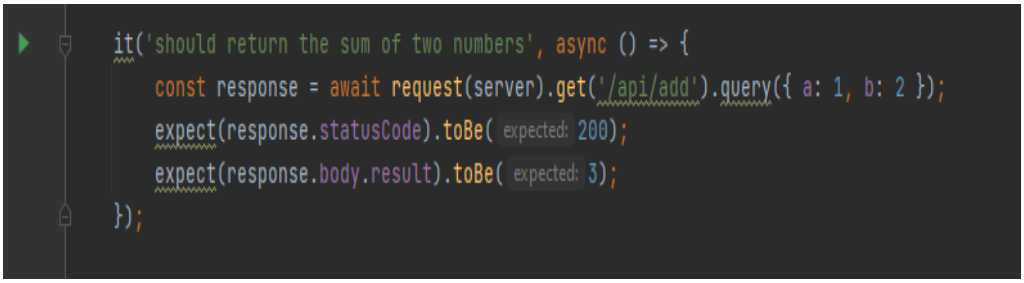
Etapas Seguidas:

1. **Reproduzir o Problema:**
    - Acesse um projeto que tenha implementações de testes unitários e funcionais.
    - Execute os testes para verificar se há falhas e se as descrições dos testes correspondem às funcionalidades.

```
PS C:\Users\Eduardo\WebstormProjects\TesteSoftware_2024_Eduardo_Marrano> npm test

> testesoftware_2024_eduardo_marrano@1.0.0 test
> jest

 PASS  tests/app.test.js
 PASS  tests/index.test.js

Test Suites: 2 passed, 2 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        0.709 s, estimated 1 s
Ran all test suites.
```

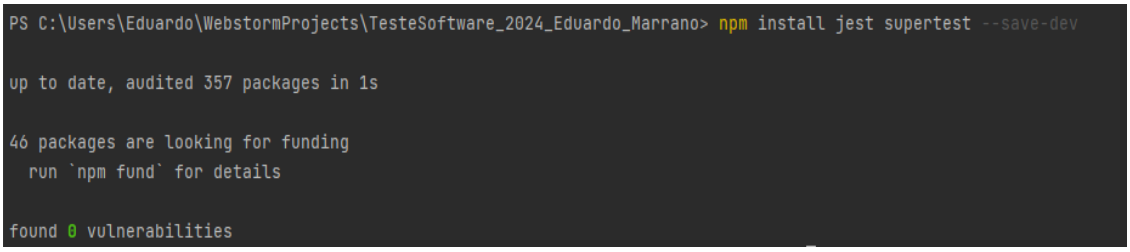
2. **Adotar a Resolução:**
   - **Testes Unitários:** Verifique se cada unidade de código funciona isoladamente e se a cobertura de testes é adequada para cada função/método.

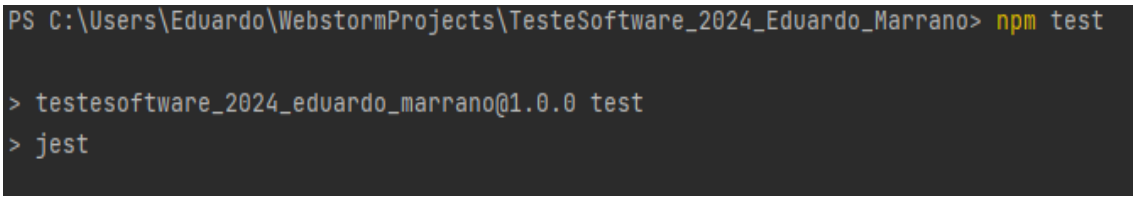
   - **Testes Funcionais:** Execute cenários de teste que simulam o comportamento do usuário para garantir que o sistema funcione conforme esperado.

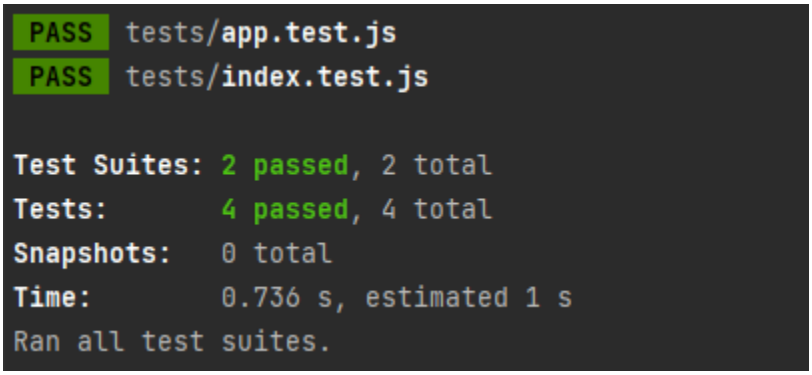


Tutorial Passo a Passo:

1. **Configuração do Ambiente:**

- Instale as ferramentas necessárias, como Jest para testes unitários e ferramentas específicas para testes funcionais, conforme o ambiente de desenvolvimento escolhido.
- 

2. **Criação dos Testes:**
   - Escreva testes unitários para funções ou métodos individuais, garantindo que cada um execute corretamente suas responsabilidades.
   - Escreva testes funcionais que simulem interações do usuário com o sistema e verifique se o software atende aos requisitos funcionais.
3. **Execução dos Testes:**
   - Utilize comandos como npm test para rodar os testes e verifique os resultados para garantir que todos os testes unitários e funcionais passem.
   - 

4. **Análise dos Resultados:**
   - Revise os resultados dos testes para identificar falhas e fazer ajustes no código conforme necessário.
   - 

Conclusão:

A diferenciação entre testes unitários e funcionais é crucial para garantir que tanto as partes individuais do código quanto o sistema como um todo estejam funcionando corretamente. Testes unitários asseguram a funcionalidade interna, enquanto testes funcionais garantem que o software atenda às expectativas do usuário.