**O que é um teste unitário?**

Teste unitário é uma verificação feita em um trecho específico de código, visando uma atenção mais detalhada e focada para facilitar a visualização de erros. Esses testes são parte do projeto, fazendo com que se integre de tal forma com a rotina de desenvolvimento que não seja preciso retirar os testes para proceder com algumas partes da codificação.

Texto

Descrição gerada automaticamenteFigura 1 - Exemplo de teste unitário feito em Golang

Fonte: Autoria Própria, 2024.

**O que faz um Teste Unitário um Teste Unitário?**

Um teste unitário, diferente de testes mais gerais, é focado em uma parte específica do código, como uma função ou, dependendo do tamanho, de um arquivo mais específico. Além disso, os testes unitários normalmente possuem um padrão codificável por quem programa, diferenciando-se de testes informais feitos comumente por pessoas.

**Qual é o Propósito dos Testes Unitários?**

Os testes unitários são utilizados como testes prévios antes de integrar uma parte do sistema a outra para facilitar a identificação de erros, reduzir os custos em testes manuais e utilizar o mesmo teste para rastrear bugs, diminuindo a reescrita de código.

**Quais São os Benefícios dos Testes Unitários?**

Além da redução de possíveis erros que é o foco da maioria dos testes, os testes unitários trazer diversos benefícios como a construção de um sistema mais robusto e, consequentemente, com uma maior qualidade, podem servir como pontos importantes da documentação do software e a facilitação de processos de desenvolvimento complexos, por permitir a implementação de Desenvolvimento Orientado a Testes (TDD).

**Como Escrever Testes Unitários (Usando as Melhores Práticas)**

Como em toda prática de Clean Code, é importante nomear com cuidado os métodos utilizados para o teste permitindo uma maior legibilidade do código, além de utilizar de mensagens de erros e de status ok claras para permitir o debug do código com maior facilidade. Por fim, os códigos de teste devem ser bem escritos a ponto de que funcionem com tamanha precisão para que garantam que o código só passe em caso positivo, caso contrário, seria necessário um código de teste para o código de teste.

**Pesquise 5 Ferramentas para Técnicas de Testes Unitários**

**Texto

Descrição gerada automaticamenteJUnit (Java)**

Figura 2- Exemplo de teste unitário usando JUnit

Fonte: Autoria Própria, 2024.

**Pytest (Python)**

Texto

Descrição gerada automaticamenteFonte: Autoria Própria, 2024.

Figura 3 - Exemplo de teste unitário usando Pytest

**Texto

Descrição gerada automaticamenteJest (Javascript)**

Figura 4 - Exemplo de teste unitário usando Jest

Fonte: Autoria Própria, 2024.

**NUnit (C#)**

Figura 5 - Exemplo de teste unitário usando NUnit

Texto

Descrição gerada automaticamenteFonte: Autoria Própria, 2024.

**PHPUnit**

**Texto

Descrição gerada automaticamente**Figura 6 - Exemplo de teste unitário usando PHPUnit

Fonte: Autoria Própria, 2024.