**Exercício Avaliativo Individual**

**ENTREGA SERÁ REALIZADA NO LMS CONFORME COMBINADO EM SALA DE AULA NO ITEM ENTREGAS FINAIS**

**Pergunta 1**

Qual framework é amplamente utilizado para testes unitários em JavaScript?

A) RSpec

B) Jest

C) PyTest

D) JUnit

**Pergunta 2**

O que é um Mock em testes unitários com JavaScript?

A) Uma função que retorna valores aleatórios.

B) Um tipo especial de teste que verifica a performance.

C) Um objeto simulado que imita o comportamento de objetos reais de forma controlada.

D) Uma ferramenta para compilar JavaScript em código nativo.

**Pergunta 3**

No contexto de TDD, qual é a sequência correta de passos?

A) Refatorar, Testar, Codificar

B) Codificar, Testar, Refatorar

C) Testar, Codificar, Refatorar

D) Testar, Refatorar, Codificar

**Pergunta 4**

O que diferencia BDD de TDD?

A) BDD não utiliza testes.

B) BDD foca em testes baseados no comportamento e na comunicação entre as partes interessadas.

C) TDD é exclusivamente para testes de UI.

D) BDD é uma técnica de desenvolvimento sem testes.

**Pergunta 5**

Qual é o propósito da Pirâmide de Testes?

A) Demonstrar a importância de testes manuais sobre testes automatizados.

B) Ilustrar a distribuição ideal de diferentes tipos de testes em um projeto de software.

C) Mostrar que testes de integração são mais importantes que testes unitários.

D) Indicar que todos os tipos de testes devem ter a mesma quantidade.

**Pergunta 6**

No padrão Triplo A (Arrange, Act, Assert) para testes unitários, o que significa "Arrange"?

A) Compilar o código antes de testar.

B) Configurar o ambiente de teste, preparando os dados de entrada.

C) Executar o teste.

D) Verificar o resultado do teste.

**Pergunta 7**

Por que é importante isolar o código de suas dependências em testes unitários?

A) Para tornar os testes mais lentos e detalhados.

B) Para garantir que o teste possa ser executado em qualquer ambiente.

C) Para testar cada unidade de forma independente e garantir a precisão dos resultados.

D) Para aumentar o acoplamento entre as unidades de código.

**Pergunta 8**

Qual método pode ser usado em Jest para criar um mock de uma função?

A) jest.createMock()

B) jest.fn()

C) mockFunction()

D) simulateFunction()

**Pergunta 9**

O que é Jest?

A) Uma linguagem de programação focada em testes.

B) Uma biblioteca para manipulação de DOM.

C) Um framework de teste JavaScript para testar todas as partes do seu aplicativo.

D) Um compilador de JavaScript para melhorar a performance.

**Pergunta 10**

Em BDD, o que são "cenários"?

A) Ambientes específicos onde o software será executado.

B) Descrições detalhadas de falhas conhecidas no software.

C) Exemplos específicos de como o software deve se comportar em determinadas situações.

D) Uma técnica de depuração para identificar a origem dos bugs.

**Pergunta 11**

Como o princípio Single Responsibility do SOLID pode impactar os testes unitários em JavaScript?

A) Torna os testes unitários mais complexos, pois requer mais classes e métodos para cada teste.

B) Facilita a escrita de testes unitários, pois cada módulo ou função tem apenas uma responsabilidade.

C) Torna os testes unitários desnecessários, pois o código será automaticamente correto.

D) Dificulta a automação dos testes unitários, pois exige múltiplas instâncias de objetos.

**Pergunta 12**

Qual é a principal diferença entre teste unitário e teste de integração?

A) Teste unitário avalia o comportamento de um usuário específico, enquanto o teste de integração verifica o software como um todo.

B) Teste unitário foca em verificar a precisão de uma única função ou método, enquanto o teste de integração verifica como diferentes partes do sistema trabalham juntas.

C) Teste de integração é realizado antes do teste unitário.

D) Teste unitário é apenas para front-end, e teste de integração é apenas para back-end.

**Pergunta 13**

O que significa "coverage" ou cobertura de testes?

A) A quantidade de testes que um desenvolvedor consegue escrever em um dia.

B) A porcentagem de código que é efetivamente testada por testes automatizados, indicando potencialmente áreas não testadas.

C) A capacidade de um teste de se conectar a bancos de dados externos.

D) A velocidade com que os testes são executados.

**Pergunta 14**

Em relação a testes unitários e princípios SOLID, qual das seguintes afirmações é verdadeira?

A) Aplicar os princípios SOLID pode tornar o código mais difícil de testar, devido à sua complexidade aumentada.

B) Os princípios SOLID não têm impacto na testabilidade do código.

C) Seguir os princípios SOLID pode melhorar a testabilidade do código, tornando-o mais modular e fácil de entender.

D) Os princípios SOLID são exclusivos para testes de integração e não afetam os testes unitários.