

Preguntas de selección múltiple:

- 1) ¿Cuál es el propósito principal de las pruebas unitarias?
 - a) Comprobar la integración de componentes
 - b) Encontrar errores de sintaxis
 - c) Validar el comportamiento individual de las unidades de código
 - d) Optimizar el rendimiento de la aplicación
- 2) ¿Cuál es la herramienta recomendada para escribir y ejecutar pruebas unitarias en Angular?
 - a) Karma
 - b) Protractor
 - c) Jasmine
 - d) Jest
- 3) ¿Qué comando se utiliza para ejecutar las pruebas unitarias en Angular?
 - a) ng run test
 - b) ng test
 - c) ng unit
 - d) ng run unit-tests
- 4) ¿Cuál es la función principal del decorador "@Component" en Angular?
 - a) Definir un servicio
 - b) Definir un módulo
 - c) Definir un componente
 - d) Definir una directiva
- 5) ¿Cuál es la sintaxis correcta para escribir una prueba unitaria en Jasmine?
 - a) describe('MyTest', () => { ... })
 - b) it('MyTest', () => { ... })
 - c) test('MyTest', () => { ... })
 - d) scenario('MyTest', () => { ... })
- 6) ¿Cuál es el propósito de la función "beforeEach" en Jasmine?
 - a) Ejecutar código antes de cada prueba
 - b) Ejecutar código antes de todas las pruebas
 - c) Ejecutar código después de cada prueba
 - d) Ejecutar código después de todas las pruebas
- 7) ¿Cuál es la función utilizada para realizar afirmaciones en las pruebas unitarias?
 - a) expect()
 - b) assert()
 - c) verify()
 - d) check()
- 8) ¿Cuál es la función utilizada para simular una interacción de usuario en una prueba unitaria en Angular?
 - a) spyOn()
 - b) fakeAsync()
 - c) tick()
 - d) triggerEvent()

- 9) ¿Cuál es la función utilizada para simular una respuesta HTTP en una prueba unitaria en Angular?
- a) `spyOn()`
 - b) `fakeAsync()`
 - c) `tick()`
 - d) `flush()`
- 10) ¿Cuál es el propósito de la extensión "spec.ts" en un proyecto de Angular?
- a) Contiene los archivos HTML de los componentes
 - b) Contiene los archivos CSS de los componentes
 - c) Contiene los archivos de prueba unitaria
 - d) Contiene los archivos de configuración del proyecto

Preguntas de verdadero o falso:

- 1. En las pruebas unitarias, se recomienda evitar el acceso a servicios y componentes reales.
- 2. Las pruebas unitarias en Angular se ejecutan en un navegador web real.
- 3. Las pruebas unitarias se centran en probar el comportamiento interno de una unidad de código.
- 4. Es posible probar directivas personalizadas en Angular utilizando pruebas unitarias.
- 5. Las pruebas unitarias no pueden verificar la apariencia visual de un componente.

Preguntas de respuesta corta:

- 1. ¿Cuál es la función utilizada para crear una instancia de un componente en una prueba unitaria en Angular?
- 2. ¿Qué es el "fixture" en una prueba unitaria de Angular?
- 3. Menciona dos métodos utilizados para realizar afirmaciones en Jasmine.
- 4. ¿Cuál es la diferencia entre "async" y "fakeAsync" en las pruebas unitarias de Angular?
- 5. ¿Qué es un "spy" en el contexto de las pruebas unitarias de Angular?

Preguntas de desarrollo:

- 1. Escribe una prueba unitaria que verifique si un componente de Angular tiene un título específico en su HTML.
- 2. Escribe una prueba unitaria que simule una llamada HTTP y verifique que se haya realizado correctamente.

3. Escribe una prueba unitaria para un componente que tenga una función "calcularSuma" que recibe dos parámetros y devuelve la suma de ellos. Verifica que la función retorne el valor esperado.
4. Escribe una prueba unitaria que verifique que un componente muestre un mensaje de error si una propiedad "error" es verdadera.
5. Escribe una prueba unitaria para una función "obtenerDato" de un servicio que hace una petición HTTP y devuelve un Observable. Verifica que la función retorne el dato esperado.