COPY: Testing Unitario con Vite y React

Answers

1. ¿Cuál es el comando correcto para instalar las dependencias necesarias para las pruebas unitarias?

- [ ] A. npm install --save-dev vitest @testing-library/react @testing-library/jest-dom
- [ ] B. npm install vitest @testing-library/react @testing-library/jest-dom
- [ ] C. npm install --save vitest @testing-library/react @testing-library/jest-dom
- [ ] D. npm install --dev vitest @testing-library/react

Answer: npm install --save-dev vitest @testing-library/react @testing-library/jest-dom (A)

Se debe usar el comando con la opción --save-dev para instalar las dependencias de desarrollo.

---

2. ¿Qué opción se debe incluir en el archivo vite.config.js para establecer el entorno de prueba?

- [ ] A. environment: 'node'
- [ ] B. environment: 'jsdom'
- [ ] C. environment: 'react'
- [ ] D. environment: 'dom'

Answer: environment: 'jsdom' (B)

El entorno 'jsdom' es el que se utiliza para simular un DOM en las pruebas.

---

3. ¿Qué archivo opcional se debe crear para la configuración de pruebas?

- [ ] A. src/testConfig.js
- [ ] B. src/configTests.js
- [ ] C. src/initTests.js
- [ ] D. src/setupTests.js

Answer: src/setupTests.js (D)

El archivo src/setupTests.js se usa para inicializar la configuración de pruebas.

---

4. ¿Qué biblioteca se utiliza para realizar las aserciones en las pruebas?

- [ ] A. @testing-library/jest-dom
- [ ] B. React Testing Library
- [ ] C. Vitest
- [ ] D. Jest

Answer: @testing-library/jest-dom (A)

La biblioteca '@testing-library/jest-dom' se utiliza para realizar aserciones en pruebas con Jest.

---

5. ¿Dónde se recomienda crear la carpeta para las pruebas unitarias?

- [ ] A. En el directorio raíz del proyecto
- [ ] B. Dentro de la carpeta src
- [ ] C. Cerca del componente que se quiere probar
- [ ] D. En una carpeta llamada tests

Answer: Cerca del componente que se quiere probar (C)

Se sugiere crear la carpeta __tests__ o un archivo de pruebas cerca del componente que se quiere probar.

---

6. ¿Cuál de las siguientes opciones no es un comando válido para ejecutar pruebas con Vitest?

- [ ] A. npm run vitest
- [ ] B. npx vitest
- [ ] C. npm run test:watch
- [ ] D. npm test

Answer: npm run vitest (A)

El comando correcto para ejecutar pruebas no incluye 'run' antes de 'vitest'.

---

7. ¿Qué debe asegurarse al importar componentes en las pruebas?

- [ ] A. Que el componente tenga una función de prueba
- [ ] B. Que el componente se llame correctamente
- [ ] C. Que la ruta de importación sea correcta
- [ ] D. Que se importe el componente con un alias

Answer: Que la ruta de importación sea correcta (C)

Es muy importante establecer bien las rutas en la importación de los componentes para evitar errores.

---

8. ¿Qué línea de código es necesaria para renderizar un componente dentro de una prueba?

- [ ] A. render(<HolaMundo />)
- [ ] B. render(<HelloWorld />)
- [ ] C. render(HelloWorld)
- [ ] D. render(HelloWorld())

Answer: render(<HelloWorld />) (B)

Se debe usar la sintaxis JSX para renderizar correctamente el componente en la prueba.

---

9. ¿Por qué es necesario envolver el componente Navbar con BrowserRouter?

- [ ] A. Porque Navbar solo funciona con componentes de clase.
- [ ] B. Porque BrowserRouter mejora el rendimiento del componente.
- [ ] C. Porque NavLink requiere un contexto de enrutamiento.
- [ ] D. Porque Navbar no renderiza ningún enlace.

Answer: Porque NavLink requiere un contexto de enrutamiento. (C)

NavLink necesita un contexto de enrutamiento para funcionar correctamente.

---

10. ¿Qué función cumple 'describe' en los tests?

- [ ] A. Agrupar varias pruebas relacionadas.
- [ ] B. Ejecutar una prueba individual.
- [ ] C. Buscar elementos en el DOM.
- [ ] D. Renderizar componentes en el DOM.

Answer: Agrupar varias pruebas relacionadas. (A)

'describe' es utilizada para agrupar pruebas bajo un mismo bloque.

---

11. ¿Qué comprueba el segundo test sobre los enlaces de navegación?

- [ ] A. Que los enlaces no están presentes en el DOM.
- [ ] B. Que los enlaces no contienen texto.
- [ ] C. Que cada enlace muestra el texto correcto.
- [ ] D. Que los enlaces están estilizados correctamente.

Answer: Que cada enlace muestra el texto correcto. (C)

El segundo test asegura que el texto de cada enlace sea el esperado.

---

12. ¿Cuál es el propósito de 'screen.getByLabelText' en los tests?

- [ ] A. Modificar propiedades de estilo en los elementos.
- [ ] B. Comprobar que un elemento esté ausente del DOM.
- [ ] C. Obtener elementos que tienen un atributo aria-label especificado.
- [ ] D. Buscar elementos que contienen texto específico.

Answer: Obtener elementos que tienen un atributo aria-label especificado. (C)

'screen.getByLabelText' permite buscar elementos por su atributo aria-label.

---

13. ¿Cómo se verifica que un elemento está presente en el DOM?

- [ ] A. Usando expect(...).toHaveClass()
- [ ] B. Usando expect(...).toBeVisible()
- [ ] C. Usando expect(...).toBeInTheDocument()
- [ ] D. Usando expect(...).toBeDefined()

Answer: Usando expect(...).toBeInTheDocument() (C)

La afirmación expect(...).toBeInTheDocument() verifica la presencia de un elemento en el DOM.

---

14. ¿Qué clase se espera que no tenga el enlace de 'Home' cuando no está activo?

- [ ] A. text-primary-background
- [ ] B. text-link-inactive
- [ ] C. text-active-link
- [ ] D. text-primary-foreground

Answer: text-primary-foreground (D)

Se espera que el enlace 'Home' no tenga la clase 'text-primary-foreground' cuando no está activo.

---

15. ¿Qué verificación se realiza en el tercer test?

- [ ] A. Que todos los enlaces están visibles.
- [ ] B. Que el enlace activo aplica la clase de estilo correcta.
- [ ] C. Que no haya enlaces activos.
- [ ] D. Que el enlace 'Chatbot' contiene el texto correcto.

Answer: Que el enlace activo aplica la clase de estilo correcta. (B)

El tercer test comprueba que el enlace activo tenga la clase de estilo correcta.

---

16. ¿Qué se espera verificar con el uso de 'expect(...).not.toHaveClass()'?

- [ ] A. Que un elemento contenga una clase específica.
- [ ] B. Que un elemento esté presente en el DOM.
- [ ] C. Que un elemento esté oculto en el DOM.
- [ ] D. Que un elemento no contenga una clase específica.

Answer: Que un elemento no contenga una clase específica. (D)

Se verifica que un elemento no tenga una clase específica aplicando el método 'not'.
