

Tarjeta de presentacion

Escenarios de pruebas

- **Prueba de Renderizado:** Verificar que la tarjeta de presentación se renderiza correctamente en la interfaz de usuario.
- **Prueba de Datos:** Comprobar que los datos proporcionados a través de props se muestran correctamente en la tarjeta.
- **Prueba de Actualización de Datos:** Confirmar que la tarjeta se actualiza correctamente cuando se cambian los props.

Requisitos funcionales

- **Mostrar Nombre y Apellido:** El componente debe mostrar el nombre y apellido proporcionados a través de props.
- **Mostrar Profesión:** Debe mostrar la profesión proporcionada a través de props.
- **Mostrar Descripción:** Mostrar una breve descripción proporcionada a través de props.

Requisitos no funcioanales

- **Rendimiento:** La tarjeta debe renderizarse de manera rápida y eficiente.
- **Compatibilidad:** Debe ser compatible con diferentes navegadores y dispositivos.
- **Accesibilidad:** Debe ser accesible para personas con discapacidades, cumpliendo con estándares como WCAG.
- **Mantenibilidad:** El código debe estar bien estructurado y documentado para facilitar su mantenimiento.

- **Seguridad:** Debe seguir las mejores prácticas de seguridad para evitar vulnerabilidades.

Tipos de pruebas

- **Pruebas de Unidad:** Para probar funciones individuales del componente.
- **Pruebas de Integración:** Para verificar la interacción correcta entre el componente y otros componentes.
- **Pruebas de Aceptación del Usuario (UAT):** Para asegurarse de que el componente cumple con los requisitos del usuario final.
- **Pruebas de Regresión:** Para detectar cualquier regresión después de realizar cambios en el código.

Contador

Incrementar el contador al hacer clic en el boton

contador

Requisitos funcionales

Requisito: El contador debe aumentar en uno cada vez que se haga clic en el botón.

Escenario de prueba: Hacer clic en el botón y verificar que el contador se incremente correctamente.

Requisitos no funcionales

Rendimiento:

Requisito: El contador debe tener un rendimiento óptimo, incluso con múltiples clics rápidos.

Escenario de prueba: Evaluar si los usuarios comprenden fácilmente cómo usar el contador sin instrucciones adicionales.

- **Tipos de Pruebas:**

Pruebas de Unidad:

Estas pruebas se centran en verificar unidades individuales del código, como funciones o métodos.

Ejemplo: Probar la función que incrementa el contador para asegurarse de que aumente en uno correctamente.

Pruebas de Integración:

Estas pruebas evalúan cómo interactúan diferentes partes del sistema entre

Ejemplo: Verificar que el botón y el contador estén integrados correctamente para que el contador aumente cuando se haga clic en el botón.

Pruebas de Rendimiento:

Estas pruebas evalúan cómo responde el sistema bajo diferentes cargas y condiciones.

Ejemplo: Realizar clics rápidos repetidos para evaluar si el contador responde rápidamente sin retrasos significativos.

Pruebas de Interfaz de Usuario:

Estas pruebas se centran en evaluar la interfaz de usuario para asegurar que sea intuitiva y fácil de usar.

Ejemplo: Pedir a los usuarios que prueben el contador y proporcionar retroalimentación sobre su facilidad de uso.

Pruebas de Regresión:

Estas pruebas aseguran que los cambios recientes en el código no hayan introducido nuevos errores.

Ejemplo: Volver a ejecutar todas las pruebas después de realizar cambios en el código del contador para asegurarse de que siga funcionando correctamente.

Generador de colores

Requisitos funcionales

El componente debe cambiar el color de fondo de un cuadrado cada vez que se haga clic en un botón.

Selección de colores aleatorios: El cambio de color debe ser aleatorio, seleccionando colores de una paleta predeterminada o de manera completamente aleatoria.

Requisitos No Funcionales:

Rendimiento: El cambio de color debe ser instantáneo y sin retrasos perceptibles.

Accesibilidad: El componente debe ser accesible para personas con discapacidades visuales o de motricidad, cumpliendo con las pautas de accesibilidad web.

Responsive: La interfaz debe adaptarse a diferentes tamaños de pantalla, desde dispositivos móviles hasta pantallas de escritorio.

Tipos de pruebas

Pruebas de funcionalidad básica

descripción de la prueba

Se hace clic en el botón y se comprueba si el color de fondo del cuadrado cambia.

Pruebas de colores aleatorios

descripción

Se hace clic repetidamente en el botón y se verifica que el color de fondo del cuadrado cambie aleatoriamente en cada clic.

Pruebas de rendimiento

descripción

Se hacen clics rápidos y repetidos en el botón para verificar que el cambio de color sea instantáneo y sin retrasos perceptibles.

Prueba de seguridad

descripción

se intenta inyectar código malicioso a través del botón para verificar

requisitos funcionales para los emojis

Mostrar Emoji Aleatorio:

El componente debe mostrar un emoji aleatorio cada vez que se hace clic en el botón.

requisitos no funcionales para los emojis

Rendimiento:

El componente debe ser capaz de manejar múltiples solicitudes de clics simultáneos sin degradación significativa del rendimiento.

Usabilidad:

La interfaz debe ser intuitiva y fácil de usar, con retroalimentación clara para los usuarios sobre el resultado de su acción.

tipos de pruebas para los emojis

Pruebas de Funcionalidad:

Verificar que el componente muestra un emoji aleatorio después de hacer clic en el botón.

Pruebas de Rendimiento:

Evaluar el tiempo de respuesta del componente bajo diferentes cargas.

Pruebas de Usabilidad:

Obtener retroalimentación de los usuarios sobre la facilidad de uso y comprensión de la funcionalidad del componente.

Pruebas de Estrés:

Probar el comportamiento del componente bajo cargas extremas para identificar posibles puntos de fallo o cuellos de botella

Escenarios de pruebas para la adivinanza

Escenarios de Pruebas:

Generación de Número Aleatorio: Verificar que el componente genere un número aleatorio dentro de un rango especificado.

Interfaz de Usuario: Probar que la interfaz de usuario permita al usuario ingresar un número para adivinar.

Lógica de Adivinanza: Comprobar que el componente compare el número ingresado por el usuario con el número aleatorio generado y proporcione retroalimentación adecuada (por ejemplo, "demasiado alto", "demasiado bajo" o "correcto").

Finalización del Juego: Confirmar que el juego finalice correctamente una vez que el usuario haya adivinado el número correcto.

Requisitos funcionales para la adivinanza

- Generación de Número Aleatorio: El componente debe ser capaz de generar un número aleatorio dentro de un rango predefinido.
- Interfaz de Usuario: Debe proporcionar una interfaz que permita al usuario ingresar un número para adivinar.
- Finalización del Juego: El juego debe finalizar una vez que el usuario adivine correctamente el número.

Requisitos no funcionales para la adivinanza

- **Rendimiento:** El componente debe generar números aleatorios de manera eficiente para evitar retrasos perceptibles.
- **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de entender para el usuario.
- **Seguridad:** No se requieren requisitos de seguridad específicos para este componente.
- **Compatibilidad:** Debe ser compatible con los principales navegadores y dispositivos.

Tipos de pruebas para la adivinanza

Pruebas de Rendimiento: Para verificar que la generación de números aleatorios no afecte el rendimiento del componente.

Pruebas de Regresión: Después de realizar cambios en el componente, asegurarse de que las funcionalidades existentes, como la generación de números aleatorios y la lógica de adivinanza, sigan funcionando correctamente.

Pruebas Funcionales: Para verificar que todas las funciones del componente, incluida la generación de números aleatorios y la lógica de adivinanza, funcionen según lo esperado.

