Departamento de Ciencias de la Computación (DCCO) Ingeniería en Tecnologías de la Información Metodología de Desarrollo de Software NRC 4617

Página Web SmartSteps

Prueba de Caja Negra

# Presentado por:

Director: Ing. Jenny Ruíz

Fecha: 19 de agostoo de 2023

**Prueba Caja Negra**

# Partición de clases equivalentes

# 

**Validación Botón siguiente y posterior de Carrusel de Imágenes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VARIABLE | CASO DE PRUEBA |  | ESTADO | REPRESENTANTE |
| prevAction | CP 1: leftPosition es mayor que 0. Debe verificar si la función cambia el estilo left del elemento track de acuerdo con la fórmula proporcionada y si se imprime el mensaje "entro 2" en la consola. | let prevAction = (leftPosition,slickWidth,track) => {      if(leftPosition > 0) {          console.log("entro 2")          track.style.left = `${-1 \* (leftPosition - slickWidth)}px`;      }  } | Valido | Posición actual |
| CP 2: leftPosition es igual a 0. Debe verificar si la función no cambia el estilo left del elemento track y si no se imprime el mensaje en la consola. | Valido | Retrocede el carrusel |
| nextAction | CP 3: leftPosition es menor que (trackWidth - listWidth). Debe verificar si la función cambia el estilo left del elemento track de acuerdo con la fórmula proporcionada. | let nextAction = (leftPosition,trackWidth,listWidth,slickWidth,track) => {      if(leftPosition < (trackWidth - listWidth)) {          track.style.left = `${-1 \* (leftPosition + slickWidth)}px`;      }  } | Valido | Posición actual |
|  | CP 4: leftPosition es igual a (trackWidth - listWidth). Debe verificar si la función no cambia el estilo left del elemento track.  Caso de prueba 5: leftPosition es mayor que (trackWidth - listWidth). Debe verificar si la función no cambia el estilo left del elemento track. | Valido | Avanza el carrusel |

**Conclusiones:**

* Estas pruebas pueden ayudar a identificar problemas de usabilidad y experiencia del usuario al evaluar cómo el software responde a las acciones del usuario sin entrar en detalles técnicos.
* Las pruebas de caja negra se utilizan principalmente para validar que el software cumple con los requisitos funcionales especificados. Estas pruebas evalúan si el software produce resultados correctos o si se comporta de acuerdo con las expectativas del usuario.
* Estas pruebas pueden revelar problemas que no son evidentes desde una perspectiva interna, lo que hace que sean una parte fundamental de la estrategia de aseguramiento de calidad.

**Bibliografía:**

Roble (2023, 6 octubre). *Pruebas de caja negra 2022 || Técnicas, muestras y tipos*. Ebooks Online. https://keepcoding.io/blog/que-son-las-pruebas-de-caja-negra/

*Qué es black box testing o pruebas de caja negra*. (2022, 1 diciembre). OpenWebinars.net. <https://openwebinars.net/blog/que-es-black-box-testing-o-pruebas-de-caja-negra/>

EcuRed. (s. f.). *Pruebas de caja negra - EcuRed*. <https://www.ecured.cu/Pruebas_de_caja_negra>