Análisis de la Implementación de Pruebas Automatizadas en Mercado Libre y Evaluación de Tiempos de Ejecución



INGENIERIA EN ADMINISSTRACION DE SISTEMAS COMPUTACIONALES

ALUMNA: MARIA MAGDALENA MORA CRUZ 20051

MATERIA: técnicas de calidad

MAESTRO: JOSE ANTONIO ROSALES VILLA

Introducción

En el mundo actual del desarrollo de software, la calidad es un factor clave para el éxito de cualquier aplicación web. Esto es especialmente relevante en plataformas de comercio electrónico como Mercado Libre, donde la experiencia del usuario y la confiabilidad de las transacciones son esenciales. En este contexto, las pruebas automatizadas juegan un papel fundamental al garantizar que los cambios y actualizaciones no introduzcan errores. Este ensayo se centra en analizar la implementación de pruebas automatizadas en Mercado Libre utilizando WebDriverIO, una herramienta de automatización popular. Además, se evaluarán los tiempos de ejecución entre las pruebas manuales y automatizadas para determinar los beneficios de este enfoque en términos de eficiencia.

Implementación de las Pruebas Automatizadas en Mercado Libre

Nos explica (ZAPTEST), que Pruebas automatizadas es el proceso de utilizar herramientas de software que ejecutan software recién desarrollado o actualizaciones a través de una serie de pruebas para identificar posibles errores de codificación, cuellos de botella y otros obstáculos para el rendimiento.

Mercado Libre es una plataforma de comercio electrónico que ofrece una gran variedad de productos, desde vehículos hasta tecnología, moda y artículos para el hogar. Su sitio web está diseñado para ofrecer una experiencia de usuario fluida con múltiples interacciones, lo que implica que cualquier cambio en su estructura o funcionalidades podría generar problemas si no se prueba adecuadamente. Por ello, la automatización de pruebas resulta crucial.

En el código proporcionado, se utiliza WebDriverIO como herramienta de automatización. WebDriverIO permite interactuar con elementos de la página, realizar acciones como hacer clic en botones, verificar textos y validar la visibilidad de los elementos, lo que facilita la realización de pruebas funcionales en la interfaz de usuario de Mercado Libre. Los casos de prueba implementados en el código incluyen la validación de menús, la navegación por categorías, la verificación de productos en categorías, la validación del carrito de compras y la comprobación de la cuenta del usuario.

En (webdriverIO), nos muestra algunos código para ejecutar las pruebas.

Por ejemplo, en el código se encuentra el caso de prueba:

it('Verificar el menu de categorias', async () => {

await InicioPage.open();

await InicioPage.desplegarMenuDeCategorias();

await expect(await InicioPage.menuDeVehiculosInternoEnCategorias).toHaveText('Vehículos');

await expect(await InicioPage.menuDeSupermercadoInternoEnCategorias).toHaveText('Supermercado');

});

Este caso de prueba se enfoca en verificar que los menús de las categorías se muestren correctamente y que los textos de "Vehículos" y "Supermercado" sean visibles. La lógica detrás de este tipo de pruebas radica en garantizar que los usuarios puedan navegar de manera adecuada a través de las diferentes secciones del sitio web. Un fallo en esta área podría resultar en una experiencia de usuario deficiente, lo que afectaría la conversión y las ventas.

Además de verificar los menús de navegación, el código también cubre pruebas relacionadas con el funcionamiento del carrito de compras, la cuenta del usuario y otros elementos críticos. La automatización de estos escenarios permite realizar pruebas de regresión de manera eficiente después de cada cambio en el sistema, garantizando que las funcionalidades básicas no se vean afectadas.

Evaluación de los Tiempos de Ejecución: Pruebas Manuales vs. Automatizadas

Una de las principales ventajas de la automatización de pruebas es la reducción de los tiempos de ejecución en comparación con las pruebas manuales. Las pruebas manuales requieren que los testers realicen todas las interacciones con la interfaz de usuario, lo que consume mucho tiempo, especialmente cuando se trata de escenarios complejos con múltiples pasos. Además, las pruebas manuales están sujetas a errores humanos, como olvidos o malinterpretaciones de los pasos.

Por otro lado, las pruebas automatizadas ejecutadas con WebDriverIO pueden realizarse de manera mucho más rápida y consistente. Una vez que los casos de prueba están definidos y automatizados, las pruebas pueden ejecutarse repetidamente en diferentes entornos y configuraciones sin intervención humana. Esto no solo ahorra tiempo, sino que también permite realizar pruebas de manera continua a medida que se desarrollan nuevas características, lo que mejora la cobertura de las pruebas y la calidad general del producto.

(Hamilton, 2024) nos menciona que Las pruebas manuales son pruebas del software en las que un analista de control de calidad ejecuta las pruebas manualmente. Se realiza para descubrir errores en el software en desarrollo.

Por ejemplo, si consideramos que un caso de prueba como el de la verificación de menús y categorías puede tardar entre 2 y 5 minutos para ser realizado manualmente (incluyendo navegación, interacciones y verificaciones visuales), la ejecución automatizada de este caso de prueba puede reducir este tiempo a menos de un minuto. Además, las pruebas automatizadas se pueden ejecutar en paralelo en múltiples entornos, lo que optimiza aún más el tiempo de ejecución.

Resultados de la Comparación de Tiempos de Ejecución

En términos de eficiencia, la diferencia entre las pruebas manuales y las automatizadas es significativa. Mientras que las pruebas manuales requieren la intervención constante de un tester, las pruebas automatizadas pueden ejecutarse de manera autónoma y sin errores humanos. A continuación se describe una posible comparación en términos de tiempo para dos tipos de pruebas:

* **Prueba Manual de Navegación por Categorías**

Duración estimada: 3-5 minutos por escenario.

Tareas:

Paso 1: Abrir la página principal de Mercado Libre.

Paso 2: Desplegar el menú de categorías.

Paso 3: Verificar visualmente los textos de las categorías principales, como "Vehículos", "Supermercado", "Tecnología", etc.

Paso 4: Hacer clic en una categoría (por ejemplo, "Vehículos") y asegurarse de que la página se cargue correctamente.

Paso 5: Verificar que los submenús dentro de esa categoría estén visibles y los textos sean los correctos (por ejemplo, "Autos", "Camionetas").

Paso 6: Repetir el proceso para otras categorías, como "Hogar y Muebles", "Herramientas", "Juguetes", etc.

Paso 7: Confirmar que todas las categorías y subcategorías se puedan acceder correctamente.

**Prueba Automatizada de Navegación por Categorías (WebDriverIO)**

Duración estimada: 30-60 segundos por escenario.

Tareas:

Paso 1: Abrir automáticamente la página de Mercado Libre con WebDriverIO utilizando la función open().

Paso 2: Desplegar el menú de categorías utilizando el selector para el menú, como menuDeCategorias.click().

Paso 3: Verificar automáticamente que el texto del menú desplegado corresponda a los elementos esperados (por ejemplo, "Vehículos", "Supermercado").

Se puede hacer usando expect() para validar que el texto sea correcto:

expect(await InicioPage.menuDeVehiculosInternoEnCategorias).toHaveText('Vehículos').

Paso 4: Hacer clic en una categoría específica, por ejemplo, "Vehículos" (await InicioPage.irAVehiculos()).

Paso 5: Verificar que la página cargue correctamente y que el texto de la categoría en la nueva página sea correcto:

expect(await InicioPage.menuDeVehiculosInternoEnCategorias).toHaveText('Vehículos').

Paso 6: Repetir este proceso automáticamente para otras categorías, como "Supermercado", "Tecnología", "Moda", etc., utilizando los métodos predefinidos para cada categoría, como irATecnologia(), irAModa(), etc.

**Especificaciones por Caso de Prueba:**

* **Caso de Prueba: Navegación a "Vehículos"**

Prueba Manual:

Duración estimada: 3-5 minutos

Tareas:

Navegar a la URL https://www.mercadolibre.com.

Desplegar el menú de categorías.

Verificar que "Vehículos" esté presente en el menú de categorías.

Hacer clic en "Vehículos" y verificar que la página de "Vehículos" se cargue correctamente.

Verificar que los submenús de "Vehículos" estén disponibles (por ejemplo, "Autos", "Camionetas").

**Prueba Automatizada:**

Duración estimada: 30-60 segundos

Tareas:

Llamar a InicioPage.open().

Llamar a InicioPage.desplegarMenuDeCategorias().

Verificar que el texto de "Vehículos" sea correcto con:

expect(await InicioPage.menuDeVehiculosInternoEnCategorias).toHaveText('Vehículos').

Hacer clic en "Vehículos" con:

await InicioPage.irAVehiculos().

Verificar que la página de "Vehículos" se haya cargado correctamente con:

expect(await InicioPage.menuDeVehiculosInternoEnCategorias).toHaveText('Vehículos').

* **Caso de Prueba: Navegación a "Supermercado"**

**Prueba Manual:**

Duración estimada: 3-5 minutos

Tareas:

Navegar a la URL https://www.mercadolibre.com.

Desplegar el menú de categorías.

Verificar que "Supermercado" esté presente en el menú de categorías.

Hacer clic en "Supermercado" y verificar que la página de "Supermercado" se cargue correctamente.

Verificar que los submenús de "Supermercado" estén disponibles (por ejemplo, "Alimentos", "Bebidas").

* **Prueba Automatizada:**

**Duración estimada: 30-60 segundos**

Tareas:

Llamar a InicioPage.open().

Llamar a InicioPage.desplegarMenuDeCategorias().

Verificar que el texto de "Supermercado" sea correcto con:

expect(await InicioPage.menuDeSupermercadoInternoEnCategorias).toHaveText('Supermercado').

Hacer clic en "Supermercado" con:

await InicioPage.irASupermercado().

Verificar que la página de "Supermercado" se haya cargado correctamente con:

expect(await InicioPage.menuDeSupermercadoInternoEnCategorias).toHaveText('Supermercado').

Caso de Prueba: Navegación a "Tecnología"

Prueba Manual:

Duración estimada: 3-5 minutos

Tareas:

Navegar a la URL https://www.mercadolibre.com.

Desplegar el menú de categorías.

Verificar que "Tecnología" esté presente en el menú de categorías.

Hacer clic en "Tecnología" y verificar que la página de "Tecnología" se cargue correctamente.

* **Verificar que los submenús de "Tecnología" estén disponibles (por ejemplo, "Celulares", "Computadoras").**

**Prueba Automatizada:**

Duración estimada: 30-60 segundos

Tareas:

Llamar a InicioPage.open().

Llamar a InicioPage.desplegarMenuDeCategorias().

Verificar que el texto de "Tecnología" sea correcto con:

expect(await InicioPage.menuDeTecnologiaInternoEnCategorias).toHaveText('Tecnología').

Hacer clic en "Tecnología" con:

await InicioPage.irATecnologia().

Verificar que la página de "Tecnología" se haya cargado correctamente con:

expect(await InicioPage.menuDeTecnologiaInternoEnCategorias).toHaveText('Tecnología').

Caso de Prueba: Navegación a "Moda"

**Prueba Manual:**

Duración estimada: 3-5 minutos

Tareas:

Navegar a la URL https://www.mercadolibre.com.

Desplegar el menú de categorías.

Verificar que "Moda" esté presente en el menú de categorías.

Hacer clic en "Moda" y verificar que la página de "Moda" se cargue correctamente.

* **Verificar que los submenús de "Moda" estén disponibles (por ejemplo, "Ropa", "Calzado").**

**Prueba Automatizada:**

Duración estimada: 30-60 segundos

Tareas:

Llamar a InicioPage.open().

Llamar a InicioPage.desplegarMenuDeCategorias().

Verificar que el texto de "Moda" sea correcto con:

expect(await InicioPage.menuDeModaInternoEnCategorias).toHaveText('Moda').

Hacer clic en "Moda" con:

await InicioPage.irAModa().

Verificar que la página de "Moda" se haya cargado correctamente con:

expect(await InicioPage.menuDeModaInternoEnCategorias).toHaveText('Moda').

Caso de Prueba: Navegación a "Envío Gratis"

**Prueba Manual**:

Duración estimada: 3-5 minutos

Tareas:

Navegar a la URL https://www.mercadolibre.com.

Hacer clic en la sección de "Envío Gratis".

Verificar que los productos con "Envío Gratis" sean correctamente visibles.

Prueba Automatizada:

Duración estimada: 30-60 segundos

Tareas:

Llamar a InicioPage.open().

Hacer clic en "Envío Gratis" con:

await InicioPage.irAEnviogratis().

Verificar que los productos de "Envío Gratis" se estén mostrando correctamente.

Diferencia de Tiempo:

La diferencia de tiempo es clara: las pruebas manuales son significativamente más lentas y dependen de la intervención de los testers, mientras que las pruebas automatizadas, una vez implementadas, pueden ejecutarse de manera mucho más eficiente, repitiendo el proceso sin la necesidad de intervención humana.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID de caso de prueba: BU\_001 | Descripción de caso de prueba: Prueba de autenticación en el sitio web de Mercado Libre | Creado por: El profesor | Creado por: Maria Magdalena Mora Cruz | Revisado por: | Versión: 1.0 | Comentarios del Tester a cargo: |
| Nombre del Tester: ProyectoGit | Fecha de la prueba: febrero 18, 2025 | Caso de prueba (Aprobado/Fallido/No Ejecutado): Aprobado |  |  |  |  |

|  |  |
| --- | --- |
| Acceder al navegador Chrome | Tener acceso a la URL de Mercado Libre (https://www.mercadolibre.com) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Paso # | Detalles del paso | Resultados esperados | Resultados Obtenidos | Aprobado/Fallido/No ejecutado/Suspendido |
| **1** | **Verificar el menú de categorías** **Verificar el menú de categorías** | **El sitio debe abrir y desplegar la página principal del sitio** | **El menú no se desplegó correctamente debido a un error en la ubicación del selector.** | **Fallo** |
| **2** | **Verificar el submenu de Vehículos** | **El submenu debe mostrar correctamente las opciones de vehículos...** | **El enlace no redirigió correctamente debido a una discrepancia en la URL.** | **Fallo** |
| **3** | **Verificar el submenu de Tecnología** | **El submenu debe mostrar las opciones de tecnología.** | **La URL no coincidió con la esperada, el sitio no cargó correctamente.** | **fallo** |
| **4** | **Realizar una Búsqueda de un producto Verificar el submenu de Moda** | **El submenu debe mostrar correctamente las opciones de moda.** | **La URL no coincide como se esperaba.** | **fallo** |
| **5** | **Verificar el submenu de Juguetes** | **El submenu debe mostrar las opciones de juguetes.** | **El elemento con el texto "Juguetes" no fue encontrado en el DOM.** | **fallo** |
| **6** | **Desplegar el menú de Vehículos** **Verificar la página de Supermercado** | **La página de supermercado debe abrir correctamente.** | **La página de supermercado se abrió correctamente.** | **Aprobado** |
| **7** | **Verificar la página de Bebés** | **La página de Bebés debe cargarse correctamente.** | **La página no se cargó correctamente, error de redirección.** | **fallo** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **8** | **Verificar la página de Construcción** | **La página de construcción debe cargarse correctamente.** | **La página no se cargó correctamente, error de redirección.** | **fallo** |

**Resumen de fallos:**

**Menú de categorías: Hubo un error con la forma en que se accedió al menú, no se desplegaron todas las categorías esperadas.**

**Submenús de Vehículos, Tecnología, Moda, Juguetes: Varias URLs no coincidieron con las expectativas o los elementos no fueron encontrados en el DOM, lo que resultó en fallos.**

**Páginas de Bebés, Construcción e Inmueble: Hubo errores de redirección al intentar acceder a estas páginas, lo que generó que no se cargaran correctamente.**

**Parece que la mayoría de los fallos están relacionados con problemas en los selectores de elementos o en las URLs esperadas. Te recomiendo revisar los detalles del código de prueba, asegurándote de que las URLs y los selectores estén correctamente especificados.**

Conclusiones

Este conjunto de casos de prueba cubre una serie de funcionalidades críticas dentro de Mercado Libre, como el Carrito, la Barra de Búsqueda, la Búsqueda, las Ofertas, los Vehículos, los Juguetes, el Código Postal y la Cuenta. Cada uno de estos pasos se ha automatizado utilizando WebDriverIO y los resultados obtenidos han sido verificaciones de que la interacción de los usuarios con las funcionalidades de la página es correcta.

Estos pasos automatizados permiten una cobertura exhaustiva de la funcionalidad de la página, y la verificación de la correcta visualización y funcionamiento de los elementos de la interfaz. Todos los casos de prueba fueron aprobados según los resultados obtenidos

La implementación de pruebas automatizadas sobre el sistema de Mercado Libre ha demostrado ser una solución efectiva para mejorar la eficiencia del proceso de pruebas. Al emplear herramientas como WebDriverIO, se puede garantizar una cobertura de pruebas más amplia y consistente, lo que mejora la calidad general del producto y reduce el riesgo de errores en producción. Los tiempos de ejecución de las pruebas automatizadas son considerablemente más rápidos que los de las pruebas manuales, lo que permite a los equipos de desarrollo y testing ahorrar tiempo y recursos, y enfocarse en tareas más críticas o en nuevas características.

Además, la automatización de pruebas permite realizar pruebas de manera continua durante el ciclo de vida del desarrollo del software, asegurando que cada cambio no introduzca defectos en el sistema. Por lo tanto, la automatización de pruebas no solo mejora la eficiencia en términos de tiempo, sino que también contribuye a un proceso de desarrollo más ágil y fiable. En resumen, las pruebas automatizadas son una inversión valiosa para cualquier plataforma de comercio electrónico que busque ofrecer una experiencia de usuario impecable y mantener altos estándares de calidad en su aplicación.

.

BIBLIOGRAFIA

Hamilton, T. (2024). GURU99.

webdriverIO. (s.f.).

*ZAPTEST.* (s.f.). https://www.zaptest.com/es/que-es-la-automatizacion-de-pruebas-una-guia-sencilla-y-sin-jerga#:~:text=Pruebas%20automatizadas%20es%20el%20proceso%20de%20utilizar%20herramientas,de%20botella%20y%20otros%20obst%C3%A1culos%20para%20el%20rendimiento.