

Prueba técnica QA Automation

Mario Alejandro Benítez Orozco

Senior QA Automation

Medellín - Colombia

Table of Contents

SOLUCION PROPUESTA DE LA PRUEBA TÉCNICA	2
PARTE 1: EVALUACIÓN DE QA MANUAL	2
<i>Ejercicio #1: Escritura de casos de prueba.....</i>	<i>2</i>
<i>Ejercicio #2: Reporte de Defectos.....</i>	<i>4</i>
<i>Ejercicio #3: Propuesta de Mejoras en UX</i>	<i>6</i>
<i>Ejercicio #4: Ejercicio práctico</i>	<i>10</i>
PARTE 2: EVALUACIÓN DE QA AUTOMATIZADA	12
<i>Ejercicio #1: Implementación de Framework BDD.....</i>	<i>12</i>
<i>Ejercicio #2: Estructura de Automatización y Navegación en Netflix y Gmail</i>	<i>12</i>

SOLUCION PROPUESTA DE LA PRUEBA TÉCNICA

Teniendo en cuenta los requerimientos técnicos de la prueba, se desarrolla la prueba dejando a merced cada uno de los puntos indicados en este documento relacionados a través de enlaces donde se podrá revisar la propuesta de solución.

Parte 1: Evaluación de QA Manual

Ejercicio #1: Escritura de casos de prueba

01. Escenario de camino feliz.

- **CP01001:** Inicio de sesión exitoso con usuario válido.

Prioridad: Alta

Precondiciones:

- Usuario registrado en el sistema.
- Sistema en línea y operativo.

Datos de Entrada:

- Email: usuario.valido@dominio.com
- Contraseña: P@ssword123

Pasos de Ejecución:

1. Navegar a la página de inicio de sesión.
2. Ingresar el email válido.
3. Ingresar la contraseña válida.
4. Hacer clic en Iniciar sesión.

Resultado esperado:

- El usuario es redirigido al dashboard.
- Se muestra un mensaje de bienvenida.
- El token de sesión es generado y almacenado.

- **CP01002:** Inicio de sesión con opción "Recordar usuario".

Prioridad: Alta

Precondiciones:

- Usuario registrado en el sistema.
- Sistema en línea y operativo.

Datos de Entrada:

- Email: usuario.valido@dominio.com
- Contraseña: P@ssword123

Pasos de Ejecución:

1. Navegar a la página de inicio de sesión.
2. Ingresar el email válido.
3. Ingresar la contraseña válida.
4. Hacer clic en Iniciar sesión.

Resultado esperado:

- El usuario es redirigido al dashboard.
- Se muestra un mensaje de bienvenida.
- El token de sesión es generado y almacenado.

02. Escenario alternativo.

- **CP02001:** Contraseña incorrecta

Prioridad: Alta

Precondiciones:

- Usuario registrado en el sistema.
- Sistema en línea y operativo.

Datos de Entrada:

- Email: usuario.valido@dominio.com
- Contraseña: incorrecto123

Pasos de Ejecución:

1. Ingresar email válido.
2. Ingresar contraseña incorrecta.
3. Hacer clic en Iniciar sesión.

Resultado esperado:

- El sistema muestra: “Contraseña incorrecta. Intente nuevamente”
- No permite acceso.

- **CP02002:** Email no registrado

Prioridad: Alta

Precondiciones:

- Usuario registrado en el sistema.
- Sistema en línea y operativo.

Datos de Entrada:

- Email: no.registrado@dominio.com
- Contraseña: cualquier123

Pasos de Ejecución:

1. Ingresar email no registrado.
2. Ingresar contraseña.
3. Hacer clic en Iniciar sesión.

Resultado esperado:

- El sistema muestra: “Usuario no registrado”
- Acceso denegado.

03. Caso límite

- **CP03001:** Email al límite de longitud permitido

Prioridad: Baja

Datos de Entrada:

- Email: usuario.con.nombre.extremadamente.largo.para.prueba@dominio.com
- Contraseña: P@assword123

Pasos de Ejecución:

1. Ingresar email máximo permitido.
2. Ingresar contraseña.
3. Hacer clic en Iniciar sesión.

Resultado esperado:

- El sistema permite el login sin errores de validación por longitud.

Ejercicio #2: Reporte de Defectos

Basado en el contexto de la imagen dada, se identifican algunos hallazgos que deberían ser tenidos en cuenta para mejorar la experiencia y usabilidad por parte del usuario:

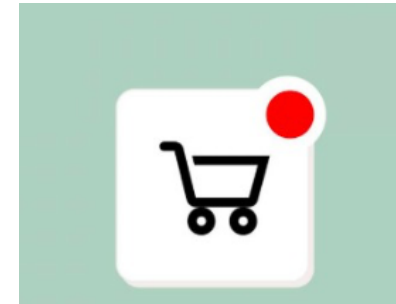
ID	Título del defecto	Descripción	Severidad	Impacto	Recomendación
DEF-001	Desalineación botón	El botón "Siguiente" no está alineado verticalmente con el campo de contraseña.	Baja	Estética	Alinear correctamente para consistencia visual.
DEF-002	Contraseña débil permitida	El sistema permite ingresar menos de 6 caracteres como contraseña.	Alta	Seguridad	Validar longitud mínima y complejidad de contraseña.
DEF-003	Falta icono de visibilidad	No hay opción para mostrar/ocultar contraseña ingresada.	Media	Usabilidad	Incluir icono de visibilidad (👁️).
DEF-004	Mensaje de error genérico	El sistema no distingue entre usuario no registrado y contraseña incorrecta.	Alta	Confusión de usuario	Proveer mensajes específicos.
DEF-005	Sin validación dinámica	El campo de contraseña no muestra feedback	Media	Usabilidad	Implementar validación en tiempo real.

		mientras se escribe.			
DEF-006	¿Sin link “Olvidaste tu contraseña?”	No hay enlace para recuperar la contraseña.	Media	Accesibilidad	Incluir enlace de recuperación.
DEF-007	Sin indicador de carga	No hay feedback visual al dar clic en “Siguiente”.	Baja	Percepción de lentitud	Mostrar spinner o barra de progreso.
DEF-008	Contraste bajo del botón	El botón “Siguiente” podría no cumplir WCAG para contraste.	Baja	Accesibilidad	Verificar y ajustar contraste.
DEF-009	Etiqueta poco visible	La etiqueta del campo contraseña está mal posicionada.	Baja	Usabilidad	Ajustar la posición y tamaño de la etiqueta.

Ejercicio #3: Propuesta de Mejoras en UX

Análisis imagen 1:

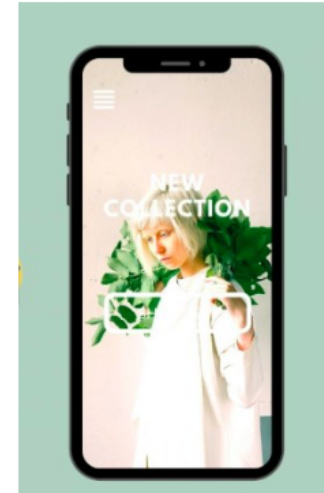
- El círculo rojo probablemente representa un indicador (notificación o cantidad de ítems).
- No hay número o contenido dentro del círculo rojo.
- El contraste entre el rojo y el fondo blanco es bueno, pero el recuadro se pierde un poco sobre el fondo verde claro.
- El propósito exacto del círculo rojo no es inmediatamente obvio.



Área UX	Problema Detectado	Impacto para el Usuario	Propuesta de Mejora
Claridad visual	El círculo rojo no tiene número o símbolo que indique su propósito (¿notificación? ¿cantidad?).	El usuario podría no entender si hay productos en el carrito o si es solo una alerta.	Incluir número dentro del círculo (por ejemplo, “1”, “2”, etc.) o un símbolo claro que indique su función.
Accesibilidad	El icono del carrito y el recuadro no tienen suficiente separación visual con el fondo verde claro.	Usuarios con baja visión o en pantallas con bajo contraste podrían no distinguir bien el icono.	Añadir borde o sombra al recuadro para que destaque sobre el fondo.
Consistencia	El círculo rojo podría interpretarse como error o alerta en lugar de cantidad.	Confusión sobre el estado: ¿es un error? ¿una notificación de producto?	Usar colores convencionales: rojo para error, azul/verde para notificación de cantidad; o acompañar el círculo con tooltip explicativo.
Usabilidad	No hay indicación de que el icono sea interactivo (no parece un botón).	El usuario podría no intuir que puede hacer clic.	Añadir hover effect, sombra o borde resaltado al pasar el cursor o al tocar.
Responsive	El tamaño del círculo rojo puede superponerse al icono en pantallas pequeñas.	Puede dificultar la identificación del icono en móviles.	Ajustar el tamaño del badge dinámicamente según el tamaño del dispositivo.

Análisis imagen 2:

- Una pantalla con el texto NEW COLLECTION en blanco.
- Imagen de una persona como parte de un banner publicitario.
- Icono de menú hamburguesa en la parte superior izquierda.
- Colores predominantes: claros (blanco, verde suave), fondo difuminado.

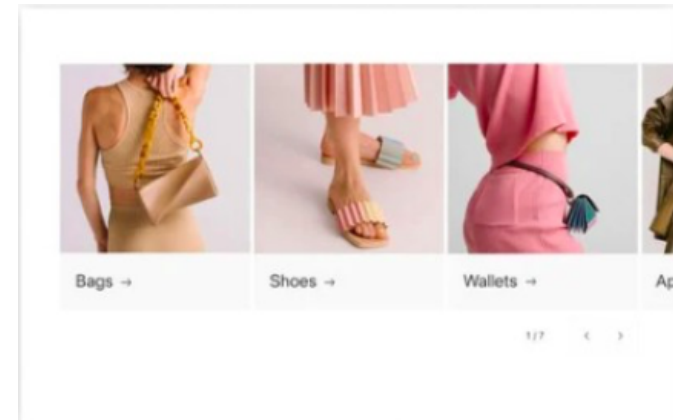


Área UX	Problema Detectado	Impacto para el Usuario	Propuesta de Mejora
Accesibilidad	El texto "NEW COLLECTION" tiene bajo contraste con el fondo de la imagen.	Dificultad para leer el texto, especialmente en condiciones de luz exterior o para personas con baja visión.	Añadir fondo semitransparente oscuro o sombra al texto para mejorar legibilidad.
Usabilidad	El ícono de menú hamburguesa es muy sutil y puede pasar desapercibido.	El usuario podría no encontrar fácilmente cómo navegar.	Aumentar el tamaño del icono y/o añadir texto ("Menú") o efecto hover/clic.
Llamado a la acción	No se observa un botón o link claro para interactuar con la colección.	El usuario no sabe cómo proceder (ver productos, comprar, explorar).	Incluir un botón destacado (por ejemplo: "Ver colección", "Shop now") con buen contraste y posición estratégica.
Consistencia visual	Elementos como el texto y el menú no destacan sobre el fondo artístico de la imagen.	La imagen publicitaria se ve bien, pero el mensaje y navegación se pierden.	Aplicar diseño con jerarquía visual: fondo opacado para imagen, texto y menú con mayor contraste y tamaño.

Responsive	El diseño podría no escalar bien en pantallas más pequeñas si todo depende de la imagen.	Riesgo de que los elementos se monten uno sobre otro en móviles pequeños.	Probar en distintos tamaños y ajustar posiciones dinámicas de elementos clave.
------------	--	---	--

Análisis imagen 3:

- Items visuales (Bags, Shoes, Wallets, entre otras).
- Texto descriptivo con flecha.
- Controles de navegación (1/7, flechas izquierda/derecha).



Área UX	Problema Detectado	Impacto para el Usuario	Propuesta de Mejora
Visibilidad del texto	Los títulos (Bags, Shoes, etc.) son pequeños y con bajo contraste sobre el fondo claro.	Puede ser difícil de leer, especialmente en pantallas pequeñas o con luz solar.	Aumentar tamaño de fuente y aplicar mayor contraste (color más oscuro o fondo semitransparente).
Navegación del carrusel	El paginador (1/7) y las flechas de navegación son poco visibles.	El usuario podría no notar que hay más productos para ver.	Incrementar tamaño de flechas, usar colores que resalten más, añadir sombra o borde.
Llamado a la acción	No hay botón o indicación para interactuar más allá de ver el ítem.	El usuario no sabe cómo proceder (ver detalles, agregar al carrito).	Añadir botón "Ver más", "Comprar ahora" o similar en cada tarjeta.
Feedback de interacción	No hay indicación visual de que los ítems son clicables.	El usuario podría no darse cuenta que puede hacer clic.	Implementar hover state (sombra, cambio de borde o color) o animación sutil.
Responsive	El diseño podría no escalar bien en móviles con pantallas pequeñas.	Elementos pueden solaparse o quedar muy juntos.	Realizar pruebas en dispositivos reales, ajustar márgenes dinámicamente.

Análisis imagen 4:

- Fondo fucsia brillante con promoción de electrodomésticos.
- Texto en varios tamaños y colores (amarillo, blanco, negro).
- Botón amarillo "¡Compra ahora!".
- Información de vigencia y términos en fuente muy pequeña.
- Fondo del sitio web queda oscurecido detrás del pop-up.



Área UX	Problema Detectado	Impacto para el Usuario	Propuesta de Mejora
Accesibilidad visual	Colores muy brillantes (fucsia, amarillo, blanco) juntos pueden generar fatiga visual.	Usuarios con sensibilidad visual o en ambientes de mucha luz pueden sentir incomodidad o no procesar bien el contenido.	Ajustar paleta de colores a tonos más suaves o balanceados. Verificar cumplimiento con normas WCAG.
Legibilidad	El texto de vigencia y términos es muy pequeño.	El usuario podría no leer condiciones importantes, afectando transparencia.	Aumentar tamaño mínimo de fuente o permitir ampliar en dispositivos móviles.
Control del usuario	No se observa un botón de cierre del pop-up.	El usuario podría frustrarse al no poder cerrar el anuncio y continuar navegando.	Incluir botón de cierre (X) visible y accesible, en la esquina superior derecha.
Usabilidad	El pop-up cubre gran parte del contenido y oscurece el fondo.	El usuario pierde el contexto de la página y podría abandonarla.	Reducir tamaño del pop-up o permitir moverlo / minimizarlo.
Accesibilidad	Posible falta de foco en teclado o screen readers para cerrar o interactuar.	Usuarios con discapacidad podrían no poder gestionar el pop-up.	Verificar navegación por teclado y etiquetado ARIA adecuado.

Ejercicio #4: Ejercicio práctico

Contexto actual

- El proceso de QA es principalmente manual, ejecutado por el equipo técnico con apoyo del equipo de Servicio al Cliente.
- Esto genera alta dependencia de recursos humanos, posibles errores por fatiga y menor velocidad en los ciclos de liberación.

Tecnologías en uso

- Backend / Infraestructura: Node.js, JavaScript, Postgres, GitLabCI, Servicios compartidos.
- Frontend: React (Web), Android (App).

Objetivo

Diseñar un plan de 3 meses para implementar herramientas y procesos de QA que:

1. Reduzcan la carga manual.
2. Aumenten la confiabilidad y la velocidad de las pruebas.
3. Se integren con el stack tecnológico existente.

Propuesta detallada de implementación

Mes	Tecnología/Proceso	Justificación	Valor agregado
1	Playwright (Web E2E automatizado) + integración en GitLabCI	Playwright se adapta perfectamente a React y permite pruebas rápidas y robustas. Integrarlo en CI asegura ejecución en cada pipeline.	Detectar fallas en el frontend en etapas tempranas, reducir esfuerzo manual en pruebas de regresión.
	Linters (ESLint, Prettier) + Unit testing backend (Jest)	Mejorar calidad de código y prevenir errores básicos antes de pruebas funcionales.	Prevención temprana de errores, mayor mantenibilidad del código.
2	Appium (Automatización app Android)	Permite pruebas E2E en la app, integrable en GitLabCI.	Cubrir regresiones en el app sin intervención manual, acelerar releases.
	Newman (Postman CLI) para pruebas de API integradas en CI	Validar servicios compartidos y backend de forma automatizada en los pipelines.	Asegura estabilidad de las APIs, reduce riesgo de fallos en producción.
3	Allure Reports + métricas en CI	Generar reportes claros y visuales para los distintos tipos de pruebas.	Mejora la comunicación de resultados y facilita la toma de decisiones.
	Pruebas de performance (k6)	Validar el rendimiento de los servicios clave bajo carga.	Asegura escalabilidad y calidad bajo demanda real.

Diagrama de Gantt propuesto (Visión general)

Lo siguiente es tener una base ya para un proyecto a mediano plazo donde se añadan más características a complementar el proyecto:



Prioridad de implementación

La prioridad está definida por:

- Impacto inmediato en calidad (Playwright, GitLabCI).
- Cobertura incremental: Web, luego API y mobile.
- Facilidad de integración con el stack actual.
- Valor que aporta a corto plazo al negocio.

Resumen de beneficios esperados

- Reducción del 50%-70% de la carga manual en pruebas regresivas.
- Mejora de la velocidad de los ciclos de integración y entrega.
- Reducción de defectos en producción por detección temprana.
- Visibilidad de resultados de QA para stakeholders.

Recomendaciones finales

- Iniciar capacitación interna en Playwright y Appium para asegurar apropiación.
- Asegurar que el pipeline de GitLabCI contemple stages para pruebas automatizadas.
- Definir un equipo responsable de la evolución y mantenimiento de los scripts de QA.

Parte 2: Evaluación de QA Automatizada

Ejercicio #1: Implementación de Framework BDD

Para este ejercicio se crea un proyecto con el nombre de **Cypress_Saucedemo_E2E_Funcional**

Características del proyecto

Este proyecto automatiza el flujo completo de compra en <https://www.saucedemo.com> usando Cypress, Gherkin (BDD) y el patrón Screenplay.

Tecnologías utilizadas

Cypress: framework de testing moderno y rápido.

Cucumber / Gherkin: sintaxis BDD para escenarios comprensibles.

Screenplay Pattern: patrón de diseño que organiza las pruebas en actores, tareas y preguntas.

Node.js + npm: gestor de dependencias.

Para más información ver el archivo README.md dentro de la carpeta del proyecto.

Ejercicio #2: Estructura de Automatización y Navegación en Netflix y Gmail

1. Automatizar el inicio de sesión en Netflix desde la página web.

Para este ejercicio se crea un proyecto con el nombre de **netflix_playwright_screenplay**.

Características del proyecto

Este proyecto es la demostración de uso del framework Playwright bajo la plataforma Web de Netflix, el cual, lo que busca es poder iniciar sesión y luego traerse los primeros 3 títulos de las películas de género Suspenso.

Importante tener en cuenta que se debe usar una cuenta activa para que la automatización funcione, adicional, modificar el profile que debe usar la automatización, esto se debe modificar y configurar en el archivo **credentials.json** dentro del directorio de **config**.

Tecnologías utilizadas

Playwright: framework de testing moderno y rápido similar a Cypress.

Screenplay Pattern: patrón de diseño que organiza las pruebas en actores, tareas y preguntas.

Node.js + npm: gestor de dependencias.

Para más información ver el archivo README.md dentro de la carpeta del proyecto.

2. Automatizar el inicio de sesión en Gmail para un dispositivo móvil

Para este ejercicio se crea un proyecto con el nombre de **gmail_appium_screenplay**.

Características del proyecto

Este proyecto tiene por objetivo demostrar la automatización mobile en Android para iniciar sesión de una cuenta de Gmail, teniendo en cuenta la versión y el dispositivo usado para este proceso.

Es relevante tener en cuenta que:

Para interactuar con una app nativa como Gmail en Android, Playwright no es la herramienta adecuada porque:

- Playwright está diseñado para automatización de navegadores web (aunque emule dispositivos móviles).
- No tiene soporte directo para interactuar con apps nativas (apk).

Por esa razón se usará una alternativa "cercana" a Playwright, en este caso, WebdriverIO, el cual, también usa nodeJS y el lenguaje es basado en Javascript.

Tecnologías utilizadas

Node.js >= 16

npm >= 8

Android Studio (para emulador y SDK tools)

Appium Server (GUI o `appium` por npm)

Para más información ver el archivo README.md dentro de la carpeta del proyecto.