Fecha Entrega: 25/JUNIO/2025

Evaluación de QA Manual y Automatizada

Moisés Pinzón xiques

Parte 1: Evaluación de QA Manual

Ejercicio #1: Escritura de Casos de Prueba

Solución

para la solución, realice el diseño de varios escenarios

Camino Feliz 1

ID: GMAIL_LOGIN_001

Título: Inicio de sesión exitoso con correo y contraseña válidos
Precondición: El usuario posee una cuenta de Google activa.

Pasos:

1. **Abrir** `https://mail.google.com/`.

2. **Ingresar un correo electrónico válido (ej. `usuario@gmail.com`).**
3. **Pulsar Siguiente.**
4. **Ingresar la contraseña correcta.**
5. **Pulsar Siguiente / Iniciar sesión.**

Resultado esperado:
Se carga la bandeja de entrada de Gmail del usuario sin mensajes de error.

Camino Feliz 2

ID: GMAIL_LOGIN_002

Título: Inicio de sesión marcando “Mantener la sesión iniciada”
Precondición: El usuario tiene una cuenta válida y desea persistir la sesión.

Pasos:

1. **Navegar a `https://mail.google.com/`.**
2. **Escribir el correo electrónico válido.**
3. **Pulsar Siguiente.**
4. **Escribir la contraseña correcta.**
5. **Marcar la casilla “Mantener la sesión iniciada” (Stay signed in).**
6. **Pulsar Siguiente / Iniciar sesión.**
7. **Cerrar el navegador.**
8. **Volver a `https://mail.google.com/`.**

Resultado esperado:
Gmail abre la bandeja de entrada directamente (o solo pide la contraseña si la política de seguridad lo exige), mostrando que la sesión se mantuvo.

Escenario Alternativo 1

ID: TC_GMAIL_LOGIN_003

Título: Intento de inicio con contraseña incorrecta
Precondición: El correo está registrado pero la contraseña es inválida.

Pasos:

1. **Ir a `https://mail.google.com/`.**
2. **Introducir el email correcto.**
3. **Pulsar Siguiente.**
4. **Introducir una contraseña incorrecta.**
5. **Pulsar Siguiente / Iniciar sesión.**

Resultado esperado:
El sistema muestra el mensaje de error “Contraseña incorrecta.

Escenario Alternativo 2

ID: TC_GMAIL_LOGIN_004

Título: Intentar iniciar sesión sin rellenar campos
Precondición: Página de Gmail login abierta.

Pasos:

1. **Entrar a `https://mail.google.com/`.**

2. **Dejar el campo de correo vacío y pulsar Siguiente.**

3. **(Si el sistema avanza) Dejar la contraseña vacía y pulsar Siguiente / Iniciar sesión.**

Resultado esperado:

Aparecen mensajes de validación estilo “Introduce un correo electrónico” / “Introduce una contraseña” debajo de los campos vacíos.

Caso Límite

ID: GMAIL_LOGIN_005

Título: Email con formato inválido
Precondición: Página de Gmail login abierta.

Pasos:

1. **Escribir `usuario@correo` (sin dominio) en el campo de email.**

2. **Pulsar Siguiente.**

Resultado esperado:
Google muestra un aviso: “Introduce un correo electrónico válido” y no permite avanzar al campo de contraseña.

Ejercicio #2: Reporte de Defectos

Pantalla: Inicio de sesión en Gmail
Contexto asumido: El usuario ha ingresado un correo con error de dominio y una contraseña visible.

Lista de Defectos Detectados

---

1 -Contraseña visible en texto plano

Severidad: Alta

Descripción: El campo de contraseña permite ver los caracteres ingresados (`45555`) directamente.

Esperado: El campo de contraseña debe ocultar los caracteres mediante `type="password"`.

Impacto: Compromete la privacidad y seguridad del usuario en entornos públicos.

2- Correo mal escrito no es validado ni advertido

Severidad: Media-Alta

Descripción: El correo `test@gamil.com` contiene un error común de dominio (gamil en vez de gmail), pero el sistema permite avanzar sin advertencia.

Esperado: Mostrar sugerencia como “¿Quisiste decir test@gmail.com?” o impedir continuar.

Impacto: El usuario podría ingresar credenciales a una cuenta inexistente o incorrecta.

3- Falta de retroalimentación visual inmediata

Severidad: Media

Descripción: Al ingresar una contraseña corta o potencialmente incorrecta, el sistema no ofrece ningún tipo de validación en tiempo real.

Esperado: Validación visual como “la contraseña debe tener al menos 8 caracteres” o un ícono de advertencia.

Impacto: Mala experiencia de usuario (UX).

4- Falta opción de mostrar/ocultar contraseña

Severidad: Media

Descripción: No hay ningún icono o botón visible para alternar entre mostrar y ocultar la contraseña.

Esperado: Un icono tipo “ojo” para alternar visibilidad del campo.

Impacto: UX limitada, especialmente para usuarios con necesidades de accesibilidad o errores de tipeo.

5- Falta de información de ayuda o enlace para recuperar contraseña

Severidad: Media

Descripción: No hay un enlace visible a “¿Olvidaste tu contraseña?” en la imagen.

Esperado: Debe estar presente un enlace para recuperación de cuenta.

Impacto: El usuario puede frustrarse si no recuerda su contraseña.

Ejercicio #3: Propuesta de Mejoras en UX (30 minutos)

Imagen(1)

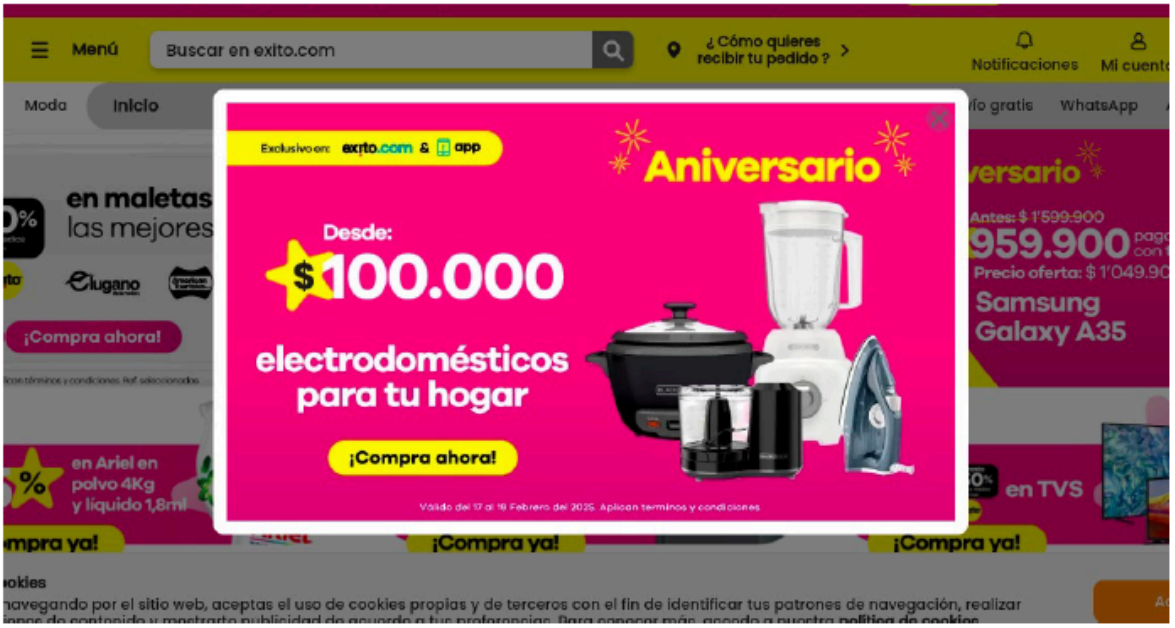


Imagen 1 – Popup promocional en Éxito.com

Problemas detectados:

1. **No hay botón de cierre visible en el popup**
   Impacto: El usuario no tiene una forma rápida de descartar la promoción si no le interesa.

   Sugerencia: Agregar un botón X mas visible en la esquina superior del popup.

2. **Contraste de colores puede afectar accesibilidad**

   Impacto: El texto blanco sobre fondo puede no cumplir con los niveles de accesibilidad.

   Sugerencia: Validar contraste con herramientas como Lighthouse o a11y color contrast checker.

3. **Falta de texto alternativo en imágenes del popup**

Impacto: Usuarios con discapacidad visual no entenderán el contenido promocional.

Sugerencia: Agregar `alt` descriptivo a cada imagen.

Imagen(2)

Imagen 2 – Ícono del carrito con punto rojo

Problemas detectados:

1. **Falta información sobre el punto rojo**
   Impacto: El usuario no sabe si es una notificación, un nuevo producto o un error.

   Sugerencia: Agregar un `tooltip` al pasar el cursor: “Tienes productos en el carrito” o “Nuevos productos disponibles”.

2. **Falta de contador numérico en el ícono del carrito**

   Impacto: No se muestra cuántos artículos hay en el carrito.

   Sugerencia: Incluir número dentro o junto al ícono.

Imagen(3)

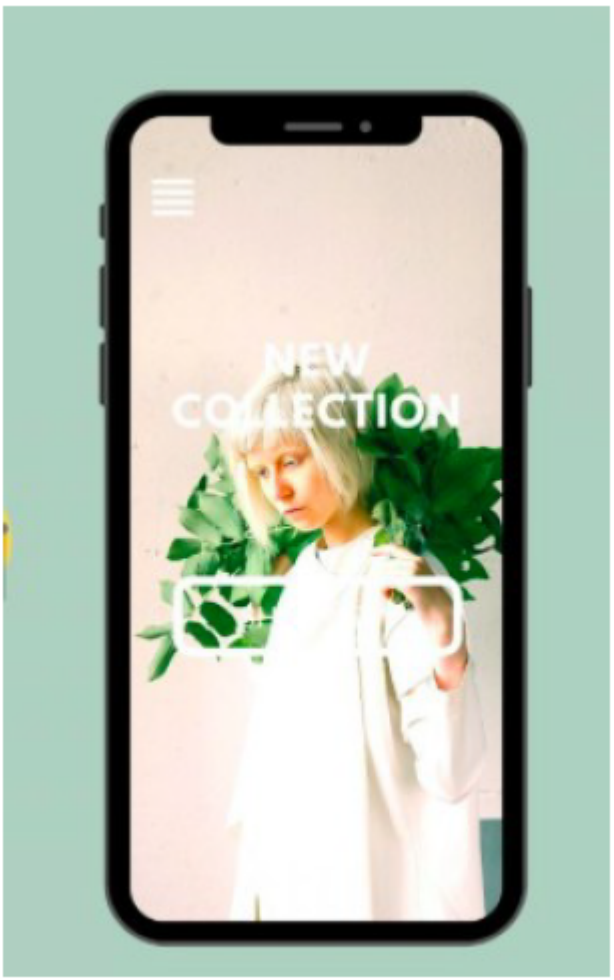


Imagen 3 – Página móvil con “New Collection”

Problemas detectados:

1. **Texto sobre imagen sin suficiente contraste**

   Impacto: “NEW COLLECTION” es difícil de leer.

   Sugerencia: Aplicar sombreado al texto o agregar fondo translúcido.

2. **Falta claridad en el botón de acción**

Impacto: El botón en la parte inferior es poco visible, no hay borde ni texto claro.

Sugerencia: Usar un botón con texto “Explorar” o “Ver colección”.

Imagen(4)

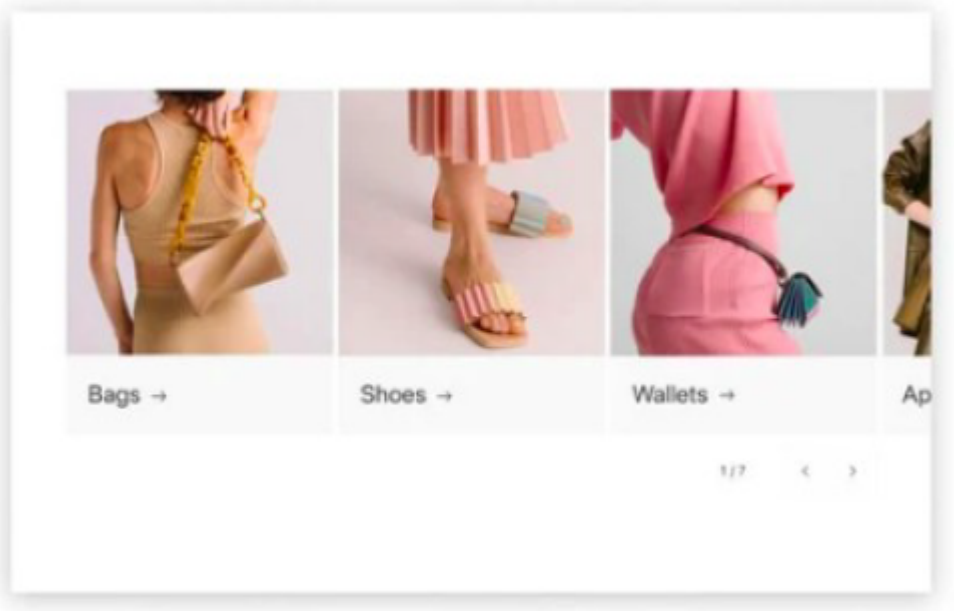


Imagen 4 – Categorías: Bags, Shoes, Wallets.

Problemas detectados:

1. **Falta de títulos accesibles o etiquetas ARIA**

   Impacto: No es claro para lectores de pantalla si son categorías o enlaces.

   Sugerencia: Usar `aria-label` o `role="link"` con descripción clara.

2. **No hay indicador visual de hover o selección**

   Impacto: UX se siente estática.

Sugerencia: Añadir animación o subrayado al pasar el mouse para indicar que son elementos clicables.

3. **Distribución inconsistente del contenido**

Impacto: La disposición de los textos y elementos no parece alineada uniformemente.

Sugerencia: Usar `Flexbox` o `Grid` para centrar vertical y horizontalmente.

Ejercicio #4: Ejercicio práctico

Pregunta

Teniendo en cuenta las tecnologías mencionadas,
¿qué tecnologías y procesos de QA

¿Qué propondrías implementar en los próximos 3 meses?

¿Cómo las priorizamos y por qué?

Respuesta:

Para fortalecer el proceso de QA durante los próximos 3 meses, propongo una estrategia progresiva enfocada en automatización, integración continua y mejora de la calidad desde etapas tempranas del desarrollo. Basándome en las tecnologías actualmente en uso, priorizaría lo siguiente:

1. Automatización de pruebas Alta prioridad - Mes 1 y 2

Tecnologías:

- **Playwright** o **Cypress** para pruebas end-to-end en aplicaciones React.
- **Jest** para pruebas unitarias en frontend (React).
- **Supertest** + **Jest/Mocha** para pruebas de integración en el backend Node.js.

Justificación:

Automatizar los flujos críticos login, pagos, formularios permite detectar errores regresivos sin depender de pruebas manuales extensivas. Dado que React y Node.js tienen muy buen soporte para estas herramientas, la curva de adopción sería baja.

---

2. Integración de pruebas en el pipeline CI/CD (Media-Alta prioridad - Mes 2)

Tecnologías:

> **GitLabCI** (ya en uso) para ejecutar pruebas automáticamente en cada merge request.
>
> **SonarQube** (opcional) para análisis estático de calidad del código y cobertura.

Justificación:

Integrar pruebas automatizadas en el pipeline asegura que cualquier cambio pase por validaciones antes de ser desplegado, reduciendo errores en producción y acelerando el ciclo de entrega.

3. Pruebas en dispositivos móviles (Media prioridad - Mes 2 y 3)

Tecnologías:

- **Appium** para pruebas automatizadas de la app Android.
- **Firebase Test Lab** como entorno escalable para pruebas en dispositivos reales.

Justificación:

La app Android debe validarse en distintos dispositivos. Appium permite reutilizar parte del conocimiento adquirido con herramientas web, y Firebase ofrece una forma rentable de validar en múltiples entornos.

---

4. Gestión de pruebas y colaboración (Media prioridad - A lo largo de los 3 meses)

Procesos y herramientas:

Implementar herramientas como **TestRail** o **Xray** (Jira plugin) para organizar casos de prueba manuales y automatizados.

Documentar criterios de aceptación desde el refinamiento con enfoque **BDD** (Behavior Driven Development), usando por ejemplo **Cucumber** en fases futuras.

Justificación:
Tener una fuente de verdad clara para QA mejora la trazabilidad, la colaboración con Customer Support y asegura cobertura de escenarios reales.

---

5. Monitoreo y feedback post-release (Baja prioridad inicial, escalar progresivamente)

Tecnologías:

- **Sentry** (frontend/backend) para capturar errores en tiempo real.
- **Posthog** o **Hotjar** para analizar comportamiento de usuarios en frontend.

Justificación:
Una buena estrategia de QA no termina en staging. Monitorear el comportamiento en producción ayuda a detectar patrones de fallo y áreas de fricción para iterar sobre la calidad del producto.

Resumen de priorización

Prioridad	Iniciativas	Justificación
Alta	Automatización E2E y pruebas unitarias	Reduce errores regresivos y acelera regresión.
Media-Alta	Integración con CI/CD	Detecta fallos antes del merge/deploy.
Media	Automatización móvil y gestión de pruebas	Mejora cobertura y trazabilidad.
Media-Baja	BDD y documentación con Customer Support	Mejora colaboración entre áreas y calidad de pruebas.
Baja	Observabilidad y monitoreo post-release	Cierra el ciclo de calidad con datos reales de uso.

Conclusión:

La estrategia se basa en automatizar lo repetitivo, prevenir en lugar de reaccionar, y acercar al equipo de QA con el desarrollo y soporte. Es progresiva, medible y alineada con el stack existente, sin introducir herramientas innecesarias.

Parte 2: Evaluación de QA Automatizada (8 Horas)

Visual Studio Code

Ejercicio #1: Implementación de Framework BDD

Utiliza cualquier framework de BDD para automatizar un escenario de inicio de sesión en una URL de tu elección.

Ejercicio #2:

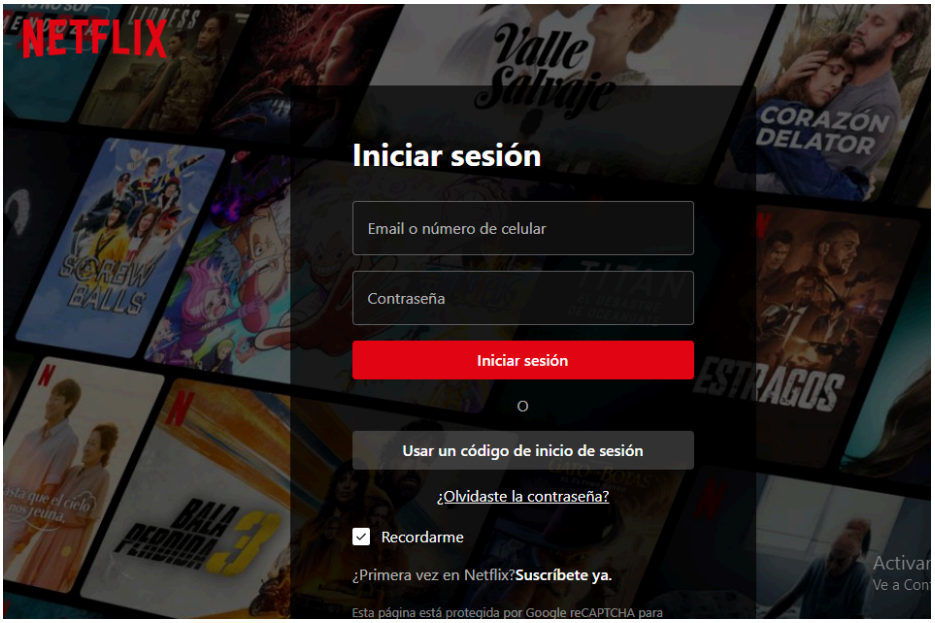
Estructura de Automatización y Navegación en Netflix y Gmail

Objetivo:

Crear un proyecto en Node.js con Playwright, siguiendo el patrón Screenplay, para automatizar los siguientes escenarios:

Proyecto Desarrollado

Repositorio: https://github.com/mpinzonxiqu/netflix-screenplay.git



https://www.netflix.com/co/login

1. Automatizar el inicio de sesión en Netflix desde la página web
   - ○ Navegar por la plataforma.
   - ○ Maximizar la pantalla.
   - ○ Capturar e imprimir los títulos de tres películas de suspenso.

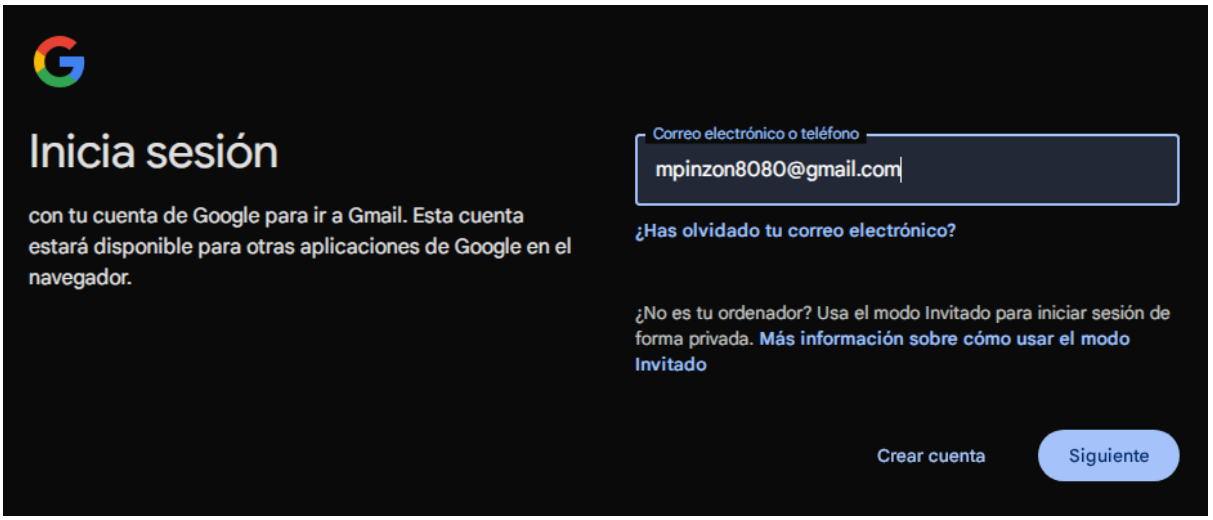
Escenario 1: Netflix (Web)
Navegar a www.netflix.com.
Maximizar la pantalla.
Extraer e imprimir:
- URL actual.
- Títulos de tres películas de suspenso.

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

Navegar en la aplicación móvil de Gmail.
Validar el mensaje de acceso exitoso.

Proyecto Desarrollado

Repositorio: https://github.com/mpinzonxiqu/gmail-playwright-screenplay.git



Escenario 2: Gmail (Móvil)
Acceder a la aplicación de Gmail desde un dispositivo móvil.
Capturar e imprimir el mensaje de inicio de sesión exitoso.