

Prueba de validación técnica

Teoría

Contexto general

La agencia está desarrollando un **dashboard de métricas digitales** que permite a los usuarios autenticados visualizar resultados de campañas, filtrar información por fecha o categoría, y exportar reportes.

El equipo está finalizando una nueva versión del módulo de filtrado, y antes de hacer merge a `develop` requiere validar **flujo completo de login → carga de métricas → aplicación de filtros → validación de resultados**.

El objetivo de esta prueba es **evaluar tu capacidad para diseñar, estructurar y pensar como QA de producto digital**, combinando razonamiento funcional, automatización y control de calidad en entornos CI/CD.

Objetivo de la prueba

Diseñar una propuesta de validación de calidad para este escenario, abarcando:

1. Estrategia general de pruebas.
 2. Diseño de casos de prueba representativos.
 3. Propuesta de automatización.
-

Instrucciones generales

- El ejercicio debe ser **escrito** (no requiere código ejecutable).
 - Puedes usar pseudocódigo, estructura YAML, o ejemplos sintéticos.
 - Enfócate en demostrar **criterio técnico, claridad, y priorización de pruebas**.
-

1. Estrategia general de pruebas (máx. 1 página)

Describe cómo abordarías la validación del flujo descrito. Incluye:

- Tipos de pruebas que considerarías, metodología que usarías para su redacción y aplicabilidad al caso señalado.
- Cómo estructurarías los ambientes de prueba.
- Qué aspectos validarías antes del merge.
- Cómo manejarías evidencias, reportes y comunicación de resultados.

2. Casos de prueba representativos

Define los casos de prueba representativos, usa la estructura que consideres oportuna para claridad, contextualización y entendimiento de los casos diseñados y su finalidad.

3. Propuesta de automatización (flujo crítico)

Elije uno de los casos anteriores

Explica:

- Qué herramienta/framework usarías tanto para automatización como para reportes, justifica la elección.
 - Cómo estructurarías la prueba para el caso concreto.
-

Práctica

Prueba de automatización UI + API

La siguiente prueba tiene como objetivo evaluar tus habilidades prácticas en automatización de pruebas, tu capacidad para analizar flujos funcionales, estructurar casos de prueba y desarrollar scripts automatizados con buenas prácticas.

Deberás realizar automatizaciones sobre el sitio <https://automationexercise.com/>.

Este sitio web de e-commerce libre de producción permite realizar pruebas tanto de interfaz de usuario (UI) como de API, y dispone de endpoints documentados para testing.

Objetivo general

Automatizar casos de prueba UI y API, demostrar que puedes integrar ambos dominios (front end y back end) en una estrategia de automatización, presentar un código limpio, bien estructurado y documentado.

Entregables esperados

1. Repositorio público en GitHub o archivo ZIP con la estructura de tu proyecto de automatización.
 - **Recomendación de arquitectura:** Se valorará positivamente si el proyecto implementa una arquitectura modular o una estructura equivalente que permita mantener el código limpio, reutilizable y escalable.
2. README explicativo.
3. Scripts de automatización que cubran los casos solicitados (UI + API).

1.1 Casos de prueba UI

	☰ Flujo	☰ Objetivo	☰ Casos
1	Inicio de sesión	Validar autenticación positiva y negativa.	<ol style="list-style-type: none">1. Login exitoso<ul style="list-style-type: none">• Navegar a <i>Signup/Login</i>• Ingresar correo y contraseña válidos• Validar saludo "Logged in as {username}"2. Login fallido por contraseña incorrecta3. Login fallido por usuario inexistente4. Verificar que los mensajes de error aparezcan correctamente
2	Opcional: Data-driven testing	Agregar una prueba de login negativo basada en datos externos (CSV, JSON o una lista parametrizada). La prueba debe ejecutar múltiples combinaciones de usuarios/contraseñas inválidas y validar los mensajes de error correspondientes.	<ul style="list-style-type: none">• usuario válido + contraseña inválida• correo con formato inválido• usuario inexistente• ambos campos vacíos
3	Agregar productos al carrito	Validar interacción y actualización de UI.	<ol style="list-style-type: none">1. Seleccionar un producto y hacer clic en Add to cart2. Validar el modal de confirmación3. Navegar al carrito y verificar:<ul style="list-style-type: none">• Nombre del producto• Cantidad• Precio4. Actualizar la cantidad del producto5. Eliminar el producto del carrito6. Validar que el carrito quede vacío
4	Checkout (sin pago real)	Comprobar flujo completo hasta página final.	<ol style="list-style-type: none">1. Desde carrito → Proceed to Checkout2. Validar dirección del usuario cargada3. Añadir comentario4. Hacer clic en Place Order (no procesa pago real)5. Validar mensaje o navegación final

5	Logout	Validar cierre de sesión	<ol style="list-style-type: none"> 1. Clic en Logout 2. Validar redirección a página de Login 3. Intentar acceder a MyAccount sin sesión → debería bloquear o pedir login
---	--------	--------------------------	--

1.2 Casos de prueba API

Mediante la API pública que tiene el sitio: https://automationexercise.com/api_list

≡ Flujo	≡ Endpoint	≡ Casos
1 Login API	POST /login	<ol style="list-style-type: none"> 1. Login exitoso → validar <code>responseCode = 200</code> 2. Login con password inválido → <code>responseCode = 404</code> 3. Login con email inexistente → <code>responseCode = 404</code> 4. Validar estructura del JSON (mensaje, código, estado)
2 Productos	GET /productsList	<ol style="list-style-type: none"> 1. Validar que el endpoint responde 200 2. Validar que existan productos en el array 3. Verificar estructura de un producto (id, name, price) 4. Validar que los IDs no sean duplicados
3 Crear usuario	POST /createAccount	<ol style="list-style-type: none"> 1. Creación exitosa con email nuevo 2. Intento de creación con email ya existente → debe fallar 3. Validar estructura de campos requeridos
4 Validación cruzada UI + API		<ul style="list-style-type: none"> • Hacer login por UI • Capturar nombre del usuario en UI • Validar contra valor retornado por <code>/getUserDetailByEmail</code>
5 Casos API	<p>Agregar al menos un caso adicional enfocado en escenarios negativos o de borde.</p> <p>Este punto permite evaluar tu capacidad de anticipar errores y validar robustez del backend.</p>	<ul style="list-style-type: none"> • Validar comportamiento enviando parámetros vacíos o incorrectos. • Envío de request sin los headers requeridos (si aplica). • Validar estructura del error retornado por la API. • Validar tiempos de respuesta cuando corresponda (ej. asegurar que responde < X ms).