Examen Técnico – QA Automation

# 1. Test Plan – Funcionalidades “Inicio de Sesión” y “Carrito de Compras”

Nombre del Proyecto: SauceDemo - Test Plan de Inicio de Sesión y Carrito de Compras  
  
Objetivo:  
Validar la funcionalidad de inicio de sesión y carrito de compras en https://www.saucedemo.com.  
  
Alcance:  
- Inicio de sesión con distintos perfiles de usuarios  
- Validación de errores de autenticación  
- Carrito de compras: agregar, eliminar   
- Checkout básico  
  
Características a probar:  
- Login: éxito, error, bloqueo, campos vacíos  
- Carrito: agregar/eliminar productos,navegación  
  
Criterios de Entrada:  
- Ambiente de testing disponible  
- Credenciales: standard\_user, locked\_out\_user   
- Contraseña común: secret\_sauce  
  
Criterios de Salida:  
- Casos ejecutados  
- Defectos reportados  
- Automatización en ejecución satisfactoria  
  
Tipos de Prueba:  
- Funcional, UI básica, Smoke, Regresión, Exploratoria  
  
Herramientas:  
- Manual: Navegador + Excel  
- Automatización: Cypress/Javascript  
  
Casos clave identificados:  
- Login exitoso, fallido, validación vacía  
- Agregar, eliminar productos, ir a checkout

# 2. Casos de Prueba – Login y Carrito

Se desarrollaron 6 casos de prueba en total:

• Login exitoso con usuario estándar  
• Login fallido con usuario bloqueado  
• Login - Validación de campos vacíos

• Completar compra de producto  
• Eliminar producto del carrito  
• Eliminar producto del carrito-Continuar con el proceso de compra

Todos están incluidos en el Excel adjunto en la carpeta del proyecto.

# 3. Suite de Smoke y Regresión

Suite de Smoke:

Login:  
- CP-001: Login exitoso

Shopping Cart:  
- CP-001: Completar compra de producto  
- CP-003: Eliminar producto del carrito-Continuar con el proceso de compra  
Justificación: Prueban flujo crítico del sistema.  
  
Suite de Regresión:  
- Incluye todos los casos creados (login y carrito)  
Justificación: Asegura que funcionalidades existentes sigan funcionando tras cambios.

# 4. Reporte de Defecto

Se adjunta el bug dentro de la carpeta del Proyecto.

# 5. Proceso para Pruebas de API

1. Analizar documentación de API (Swagger/Postman)  
2. Diseñar casos: positivos, negativos, límites, performance  
3. Herramientas: Postman, Rest Assured, Pytest, etc.  
4. Preparar entorno (tokens, datos, variables)  
5. Ejecutar y validar (status, estructura, tiempos)  
6. Reportar errores  
7. Automatizar en CI si aplica

# 6. Ejemplo de Test de API – Pytest + Requests

Test de login exitoso:

import requests  
  
def test\_login\_exitoso():  
 response = requests.post("https://api.saucedemo.com/login", json={  
 "username": "standard\_user",  
 "password": "secret\_sauce"  
 })  
 assert response.status\_code == 200  
 assert "token" in response.json()

Test de login inválido:

def test\_login\_invalido():  
 response = requests.post("https://api.saucedemo.com/login", json={  
 "username": "usuario\_fake",  
 "password": "clave\_incorrecta"  
 })  
 assert response.status\_code == 401  
 assert response.json()["error"] == "Invalid credentials"