

# **Módulo de Automatización – Smoke Test de Inicio de Sesión (SauceDemo)**

## **Objetivo:**

Evaluar la capacidad del candidato para diseñar y automatizar un conjunto de pruebas básicas (Smoke Test) sobre un flujo real en web, usando buenas prácticas de automatización.

## **Aplicación a probar:**

<https://www.saucedemo.com/>

## **Tecnologías permitidas:**

- Framework: **Cypress, SerenityBdd o Selenium**
- Lenguaje: **JavaScript, Java o Python**

El candidato puede elegir la combinación que prefiera.

## **Alcance funcional del Smoke Test**

Automatizar un **conjunto mínimo de pruebas** para validar el inicio de sesión:

1. **Login exitoso con credenciales válidas**
2. **Login fallido con contraseña incorrecta**
3. **Validación de campos obligatorios**

# **Módulo Funcional**

En este módulo se debe realizar todo el proceso de testing que se realiza, donde van a crear escenarios de pruebas, casos de pruebas, reporte de bugs, van a poder señalar qué técnicas de pruebas y tipos de pruebas a utilizar

Producto ficticio: **MakersPay**, una billetera digital donde el usuario puede:

- Iniciar sesión.
- Ver su saldo.
- Enviar dinero a otro usuario usando su número de celular.

Requerimiento resumido:

Un usuario autenticado puede enviar dinero a otro usuario registrado usando su número de celular.

### **Reglas de negocio:**

- El monto mínimo por transacción es **\$5.000 COP**.
- El monto máximo por transacción es **\$2.000.000 COP**.
- El usuario **no puede** enviar más dinero del saldo disponible.
- No se permiten envíos a su mismo número de celular.
- Si la transacción es exitosa:
  - Se descuenta el saldo del remitente.
  - Se incrementa el saldo del destinatario.
  - Se registra el movimiento en el historial de ambos.
- Si la transacción falla, se muestra un mensaje de error claro y no se afecta el saldo.

## **Módulo API**

Realiza prueba funcional para la siguiente API: <https://reqres.in/api/>

- Realizar una solicitud POST al endpoint "/users"
- Enviar los siguientes datos en el cuerpo de la solicitud en formato JSON: json

Copy code

```
{"name": "Test User", "job": "Automation Engineer"}
```

- Verificar que la respuesta tenga un código de estado HTTP 201
- Extraer el ID del usuario recién creado de la respuesta y realizar una solicitud GET al endpoint "/users/{id}" donde "{id}" es el ID del usuario recién creado.
- Verificar que el código de estado HTTP de la respuesta sea 200 y que el nombre y el trabajo del usuario sean los mismos que los enviados en la solicitud POST.
- Se valorarán casos de prueba adicionales.

Criterio	Peso
Correcta implementación técnica	30%
Cobertura de los requisitos	25%
Buenas prácticas y legibilidad	20%
Uso adecuado de la herramienta	15%
Entregables claros y ejecutables	10%