

First week challenge

Les presentamos este **challenge** para poner en práctica todo lo que aprendimos durante la primera semana del **Reto!** El objetivo es que tengan la oportunidad de aplicar los conocimientos adquiridos sobre **APIs REST, Gherkin, BDD, XPath, Pruebas de performance**, etc.

El desafío se divide en tres partes: un reto de pruebas de **API**, uno de pruebas **end-to-end** y otro de **performance**. Cada uno está pensado para que puedan explorar distintas tecnologías y herramientas, pero también para que se diviertan!!

Pruebas de API

PokeAPI + StarWars API

El desafío es desarrollar un conjunto de pruebas automatizadas que interactúen con **dos APIs públicas**:

- PokeAPI - <https://pokeapi.co/>
- StarWars API (SWAPI) - <https://swapi.info/>

Objetivos

1. Elegí un framework de testing para APIs:

Puede ser pytest, mocha, supertest, etc.

2. Diseñar y ejecutar pruebas que incluyan:

- *Validación de status codes* y contenido esperado (por ejemplo, que un Pokémon o personaje exista)
- *Validación de esquemas* (campos obligatorios, tipos de datos)
- *Casos negativos* (IDs inexistentes, formatos inválidos)
- *Pruebas de integración entre APIs*:

■ Por ejemplo, compará altura promedio de Pokémon tipo “dragon” con la de los personajes de Star Wars raza “wookiee”.

- *Pruebas con sets de datos* de al menos 5 entradas dinámicas (ejemplo: probar varios Pokémon tipo fuego).

3. Documentar el trabajo:

- Un **README.md** con instrucciones para correr los tests.
- Bonus: incluir un reporte HTML (como mochawesome, pytest-html, etc).

Pruebas E2E

A - “Operación Grinch”

Automatizar un flujo completo de búsqueda y validación en [IMDb](#) o [TMDB](#), simulando el comportamiento de alguien buscando películas navideñas clásicas.

Objetivos

- Buscar la película "Home Alone" (Mi Pobre Angelito).
- Ingresar a la página de la película.
- Validar que:
 - Aparece el director "Chris Columbus"
 - La película tiene una calificación superior a 7.0
- Bonus: Guardar una captura de pantalla con las validaciones anteriores.

B - “Armando el arbolito”

Simular el flujo de compra de un usuario que entra a una tienda online de decoraciones navideñas.

Objetivos

1. Buscar "christmas decorations" o "adornos navideños".
2. Agregar al menos 2 productos distintos al carrito (como árbol de Navidad, luces, etc.).
3. Ir al carrito y validar:
 - Que estén los ítems correctos.
 - Que el precio total se calcule correctamente.
4. Bonus: Capturar una captura de pantalla del detalle del carrito.

Sugerencias de sitios:

- <https://www.christmastreeworld.com/>
- <https://www.christmaslightsetc.com/>
- <https://www.christmascentral.com/>

En ambas opciones

Para ambos casos, se debe de utilizar sintaxis Gherkin y Page Object Model

Tecnologías sugeridas:

- Behave + Playwright
- Cucumber + Selenium
- Cucumber + Cypress
- SpecFlow + Playwright

Pruebas de Performance

Operativo Invierno: InuMeth a prueba bajo presión

El **Instituto Nacional Uruguayo de Meteorología y Tecnología Humana (InuMeth)** está probando un nuevo servicio en línea que permite consultar el estado del tiempo en tiempo real. Ante la llegada de la Navidad y la necesidad de Papá Noel de planificar sus rutas de entrega de regalos considerando las condiciones climáticas, se busca simular el comportamiento de miles de personas y elfos del Polo Norte ingresando al sistema a la vez, especialmente consultando el clima en las ciudades más pobladas del país, con el objetivo de:

- Evaluar la capacidad de respuesta del servicio durante la temporada navideña
- Medir la resistencia del sistema bajo carga en fechas críticas
- Estimar cuánta población podría verse afectada en caso de caída o lentitud del sistema
- Garantizar que Papá Noel tenga información climática confiable para la noche del 24 de diciembre

Desafío

Simular un escenario de alta demanda sobre una API de clima pública (como OpenWeatherMap, WeatherAPI, etc.) **emulando el tráfico que recibiría InuMet** durante la época navideña, cuando tanto familias uruguayas como el equipo logístico de Papá Noel necesitan consultar el pronóstico para planificar celebraciones y entregas de regalos.

Objetivos

1. **Usar una herramienta de pruebas de performance**, como: JMeter, k6, Artillery, Gatling, etc.
2. **Diseñar y ejecutar el siguiente escenario:**
 - 100 usuarios virtuales concurrentes
 - Consultas cada 5 segundos durante 2 minutos
 - Ciudades objetivo:
 - **Más pobladas del país**: Montevideo, Salto, Paysandú, Las Piedras, Rivera, Maldonado, Tacuarembó
 - **Errores simulados**: nombres mal escritos o inexistentes ("Montefideo", "Paysandoom", "PoloNorte", "VillaNoel", etc.)
3. **Métricas esperadas**: Obtener métricas relevantes, ejemplos:
 - Tiempo de respuesta promedio
 - Tasa de errores (por ciudad o consulta)
 - Throughput (cantidad de peticiones por segundo)