

### Kata: Encontrar bugs

En la empresa mi primera empresa tu eres la persona encargada de la calidad de los productos digitales. En mi primera empresa se ha realizado un gran refactor a los productos donde te piden el favor que antes que salgan producción garantices que los productos estén cumplidos con los requerimientos de calidad

Para ello tu como tester tienes que revisar la página web

<https://academybugs.com/find-bugs/> y encontrar la mayor cantidad de bugs posibles, reportarlos en la un Excel con la estructura de título, descripción del bug y según su experiencia catalogue los bugs por criticidad.

### Kata: API Automatización

El api para automatizar es la siguiente <https://fakestoreapi.com/docs> y lo que debes realizar con esa api es lo siguiente con los metodos GET, POST, PUT, DELETE:

- Consumo del api con estado exitoso
- Consumo del api con estado diferente a exito
- El body se debe de leer desde un json
- Se debe generar el reporte de las apis
- Se deben ejecutar en paralelo de 3 consumos de los servicios
- Se debe validar el código de respuesta que devuelva el estado del servicio
- Imprimir las respuestas de las apis

### **NOTA: ELEGIR UNA DE LAS DOS PARA REALIZAR LAS PRUEBAS DE PERFORMANCE**

### Kata: API con Gatling

El api para automatizar es la siguiente <https://fakestoreapi.com/docs> y lo que debes realizar con esa api es lo siguiente con los metodos GET, POST, PUT, DELETE:

- Enviar 3 peticiones con el modelo abierto (No se vale usar el de `nothingFor(duration)`)
- Enviar 2 peticiones con el modelo cerrado
- El body se debe de leer desde un json
- Generar los reportes

### Kata: API con Jmeter

El api para automatizar es la siguiente <https://fakestoreapi.com/docs> y lo que debes realizar con ese api es lo siguiente con los metodos GET, POST, PUT, DELETE:

- Crear un escenario de Pruebas de carga (Load testing) con su respectiva asersion, y grafiacas
- Crear un escenario de Pruebas de estrés (Stress testing) con su respectiva asersion, y grafiacas
- Crear un escenario de Pruebas de concurrencia (Concurrency testing) con su respectiva asersion, y grafiacas

Kata: Github actions

Crea tu propio pipeline o integracion CI/CD done sea funcional de la automatización en sus repositorios personales de git hub tiene que ser ese repositorio publico.

Tener presente que este pipeline o integracion CI/CD debe ser en leguaje .yaml con la estructura que se requiere en git hub, que ejecute el caso de prueba/ los casos de prueba en git hub actions, realizar lo minimo para la ejecucion de este.

El pipeline o integracion CI/CD se debe ejecutar en rama master con buenas practicas.

**NOTA: RECUERDA QUE DEBES TENER BUENAS PRACTICAS DE AUTOMATIZACION  
SE DEBE AUTOMATIZAR EN JAVA CON SELENIUM O SERENITY**