

MAYO 16, 2024

# MAC-CENTER.COM Test Plan

Web interactions and Services

Johan Cifuentes

QA Automation Engineer

## Historial del documento

versión	fecha	autor	descripción
V1	16/05/2024	Johan Cifuentes	Creación documento

## Objetivo:

Este plan de prueba tiene la finalidad de asegurar la funcionalidad de la página web de mac-center.com como el apartado de compra de productos completa y registro e inicio de sesión. Además, se testearán los servicios de API que ofrece la pagina <https://developer.marvel.com/documentation/apiresults> como lo son de obtener personajes y comics por conjunto o individual

## Descripción página web:

- Compra de productos
  - Agregar varios productos al carrito
  - Llenar formularios
  - Generar orden de pago

## Descripción de las APIs:

- Leer (GET) servicio
  - Leer comics
  - Leer personajes

## Alcance:

Alcance para el plan de pruebas de mac-center.com

### Página web mac-center:

- En alcance:
  - Compra de varios productos seleccionados a través del Gherkin
  - Registro e inicio de sesión
- Fuera de alcance:
  - Todo lo que no está en alcance
  - Pruebas no funcionales

## Servicios API:

- En alcance:
  - Consulta de todos los personajes GET
  - Consulta de personaje por identificador GET
  - Consulta de todos los comics GET
  - Consulta de comics por identificador GET
- Fuera de alcance:
  - Todo lo que no está en alcance
  - Pruebas no funcionales

## Entorno de pruebas:

- Navegador Chrome
- Navegador Microsoft Edge
- Sistema operativo Windows 10 Pro

## Datos de prueba:

- Datos ingresados manualmente a través de los escenarios Gherkin
- Datos autogenerados utilizando librerías externas como Faker

## Estrategia de pruebas:

### Página web:

- Paso 1: Diseño de casos de prueba en lenguaje natural usando Gherkin utilizando pruebas de caja negra y partición de equivalencia
- Paso 2: Creación de los casos de prueba utilizando Gherkin y archivos .feature
- Paso 3: Automatizar los casos de prueba utilizando herramientas de automatización

### Servicios API:

- Paso 1: Diseño de casos de prueba en lenguaje natural usando Gherkin utilizando pruebas de caja negra y partición de equivalencia

- Paso 2: Creación de los casos de prueba utilizando Gherkin y archivos .feature
- Paso 3: Automatizar los casos de prueba utilizando serenity + screenplay + herramienta de automatización de API's directa de serenity

## Criterios de aceptación:

- Automatizar más del 80% de los casos de prueba.
- Pasar exitosamente más del 80% de los casos de prueba automatizados.
- Documentar más del 90% de los errores encontrados en el sistema y abrir el reporte correspondiente, incluyendo información detallada sobre el error.
- Completar la automatización dentro del plazo establecido, que será especificado en días, semanas o meses según lo acordado previamente.

## Herramientas de la prueba:

- Java (lenguaje de programación)
- Selenium (dependencia del del desarrollo)
- Serenity (framework)
- Rest Assured (dependencia del desarrollo)
- Postman
- IntelliJ Idea

## Calendario de pruebas

Nro.	Prueba	Fecha	Total, días
1	Creación plan de pruebas	15/05/2024	2
2	Diseño casos de prueba	15/05/2024	2
3	Automatización casos de prueba	15/05/2024	2
4	Reporte de pruebas	16/05/2024	2

# Entregables de la prueba

Los siguientes productos de trabajo serán entregados

Entregable	Descripción
Plan de pruebas	Detalles del objetivo de la prueba, el alcance, la estrategia y los entregables de la prueba
Diseños de caso de prueba	Diseños de caso de prueba realizados en archivo .feature que describe la secuencia de cada prueba
Matriz de riesgo	Matriz que resalta posibles riesgos y como mitigarlos en la fase del proyecto y producto
Automatización de las pruebas	Proyecto en Java con las pruebas automatizadas