

INFORME

TEST PLAN

HIBERUS UNIVERSITY

Nombre del proyecto	Desarrollo de software E2E con pruebas automatizadas.	Período del informe
Propietario del proyecto	Manuel Tormos Blanes	13/06/2024 - 19/06/2024
Preparado por	Manuel Tormos Blanes	

INTRODUCCION

Desarrollo de Software End-to-End con Pruebas Automatizadas

Este proyecto simula un proceso completo de desarrollo de software, desde la definición de criterios de aceptación basados en historias de usuario hasta la implementación de pruebas automatizadas en correlacion a los criterios de aceptación. Se pondrán en práctica conocimientos de pruebas de software y herramientas de automatización.

OBJETIVOS

OBJETIVOS DE CALIDAD

- Asegurar que todas las funcionalidades cumplan con los criterios de aceptación definidos.
- Garantizar la cobertura completa de pruebas automatizadas.
- Integrar pruebas en un entorno de CI para asegurar la calidad continua del software.
- Proveer reportes detallados de resultados de pruebas para un seguimiento eficiente.

PLANIFICACION

PROCESOS DE CALIDAD

- Revisión de Requisitos: Revisión de historias de usuario y criterios de aceptación para asegurar que sean claros y completos.
- Desarrollo Ágil: Desarrollo iterativo e incremental con revisiones constantes y retroalimentación continua.
- Pruebas Automatizadas: Implementación de pruebas automatizadas de API y UI utilizando Cucumber, Gherkin, Java, JUnit, RestAssured y Cypress.
- Integración Continua: Configuración de GitHub Actions para la ejecución automática de pruebas.

CRITERIOS DE ACEPTACION

Historia de usuario	Descripción	Criterios de aceptación
HU1: Búsqueda de Personajes	Como usuario, deseo poder buscar personajes de Rick and Morty por nombre para obtener información detallada sobre ellos, incluyendo su nombre, estado, ubicación, su imagen y los episodios en los que aparece. La búsqueda debe ser tolerante a nombres incompletos y ofrecer sugerencias si no se encuentra una coincidencia exacta.	<ol style="list-style-type: none">1.Entrada de Búsqueda2.Tolerancia a Nombres Incompletos.3.Resultados de Búsqueda.4.Información Detallada5.Mensajes de Error y Sugerencias.6.Rendimiento y Usabilidad.
HU2: Visualización de Detalles de Personaje	Como usuario, después de conocer el id de un personaje, deseo obtener la información detallada sobre él, incluyendo su nombre, estado, especie, genero, origen, locación, imagen y episodios en los que aparece	<ol style="list-style-type: none">1.Entrada de ID2.Visualización de información3.Validación de ID4.Rendimiento y Usabilidad5.Persistencia de Datos
	Como cliente, deseo poder agregar productos a mi carrito de compras desde cualquier página de la tienda. Al hacer clic en el botón “Agregar al Carrito“, el	<ol style="list-style-type: none">1.Al hacer clic en “Agregar al Carrito“ El producto se debe agregar al carrito.2.Debe mostrar un mensaje de confirmación indicando que el

HU3: Agregar Producto al Carrito.	<p>producto debe agregarse al carrito y se debe mostrar un mensaje de confirmación. El número de artículos y el total del carrito deben actualizarse automáticamente.</p>	<p>producto se ha agregado al carrito.</p> <p>3.El número de artículos en el carrito debe actualizarse automáticamente.</p> <p>4.El total del carrito debe actualizarse</p>
HU4: Ver y Editar Carrito	<p>Como cliente, deseo poder ver el contenido de mi carrito de compras en cualquier momento. En la página del carrito, debo poder ver una lista de los productos agregados, con sus nombres, imágenes, precios, cantidades y subtotal. También debo poder eliminar productos del carrito, actualizar las cantidades y ver el total actualizado.</p>	<p>1.Debo poder ver una lista de los productos en el carrito con sus nombres, imágenes,precios,cantidades y subtotal.</p> <p>2.Debo poder eliminar productos del carrito.</p> <p>3.Debo poder actualizar las cantidades de los productos en el carrito.</p> <p>4.El total del carrito debe actualizarse automáticamente después de cualquier</p>
HU5: Comprar producto	<p>Como cliente, deseo poder comprar un iPhone y un monitor Apple Cinema 30". Se debe mostrar un mensaje de confirmación indicando que los</p>	<p>1.Debo poder completar la compra de un iPhone y un monitor Apple Cinema 30".</p> <p>2.Se debe mostrar un mensaje de confirmación</p>

	productos se han comprado correctamente.	indicando que los productos se han comprado correctamente.
--	--	--

ACTUALIZACIONES DE ESTADO

Tarea o entregable	Propietario de la tarea	Notas	Estado
Gerkhin HU1,HU2,HU3,HU4,HU5	Manuel Tormos Blanes	HECHO	Finalizados los Gerkhin de las historias de usuario acorde a los CA.
Pruebas automatizadas de Backend	Manuel Tormos Blanes	EN CURSO	Programacion de pruebas automatizadas en Java con Junit, cucumber, maven.
Pruebas automatizadas de Frontend	Manuel Tormos Blanes	EN CURSO	Programadas pruebas automatizadas en JavaScript con Cypress, mochawesome, allure.
Configuración de GitHub Actions para la ejecución automatica de pruebas.	Manuel Tormos Blanes	ATASCADO	PTE. REALIZAR



HECHO

EN CURSO

ATASCADO

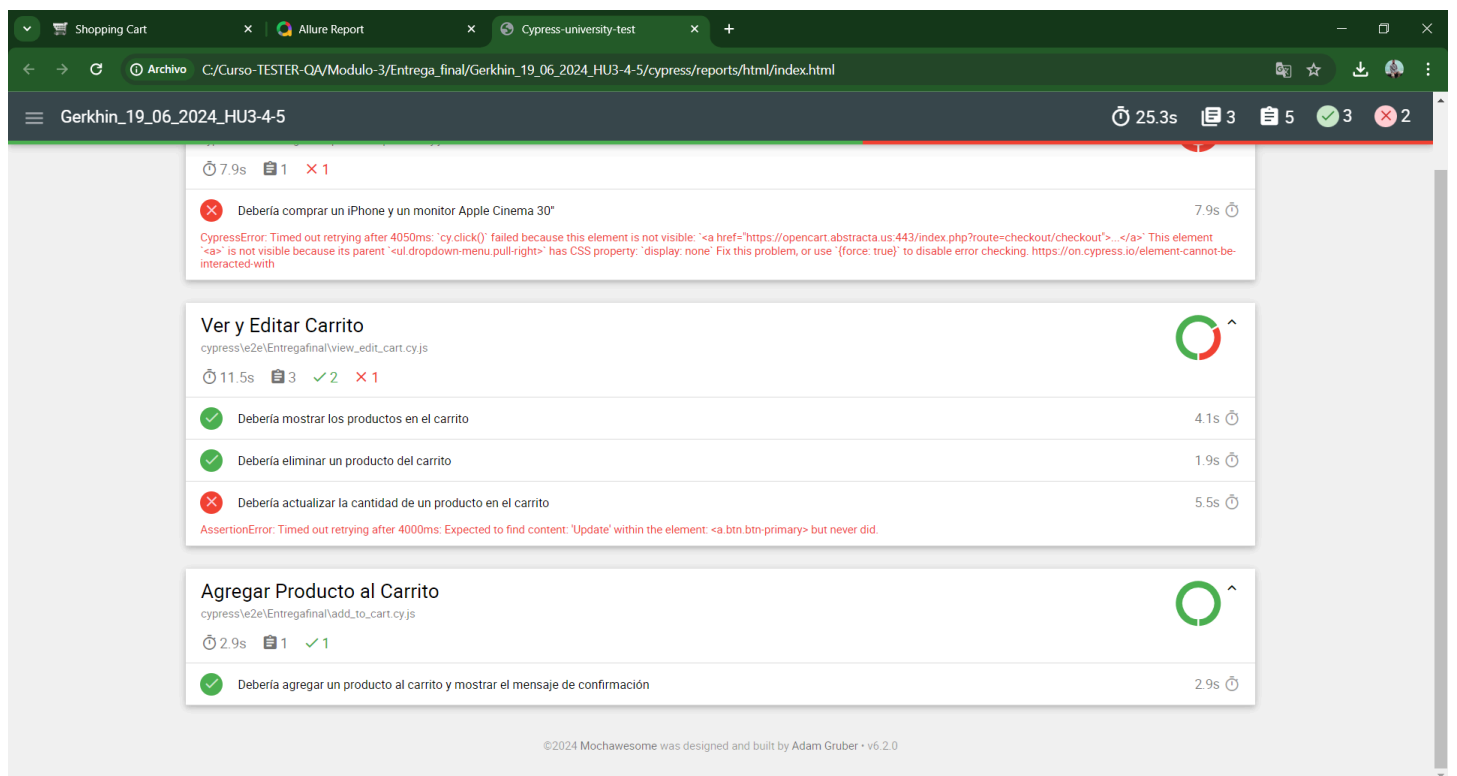
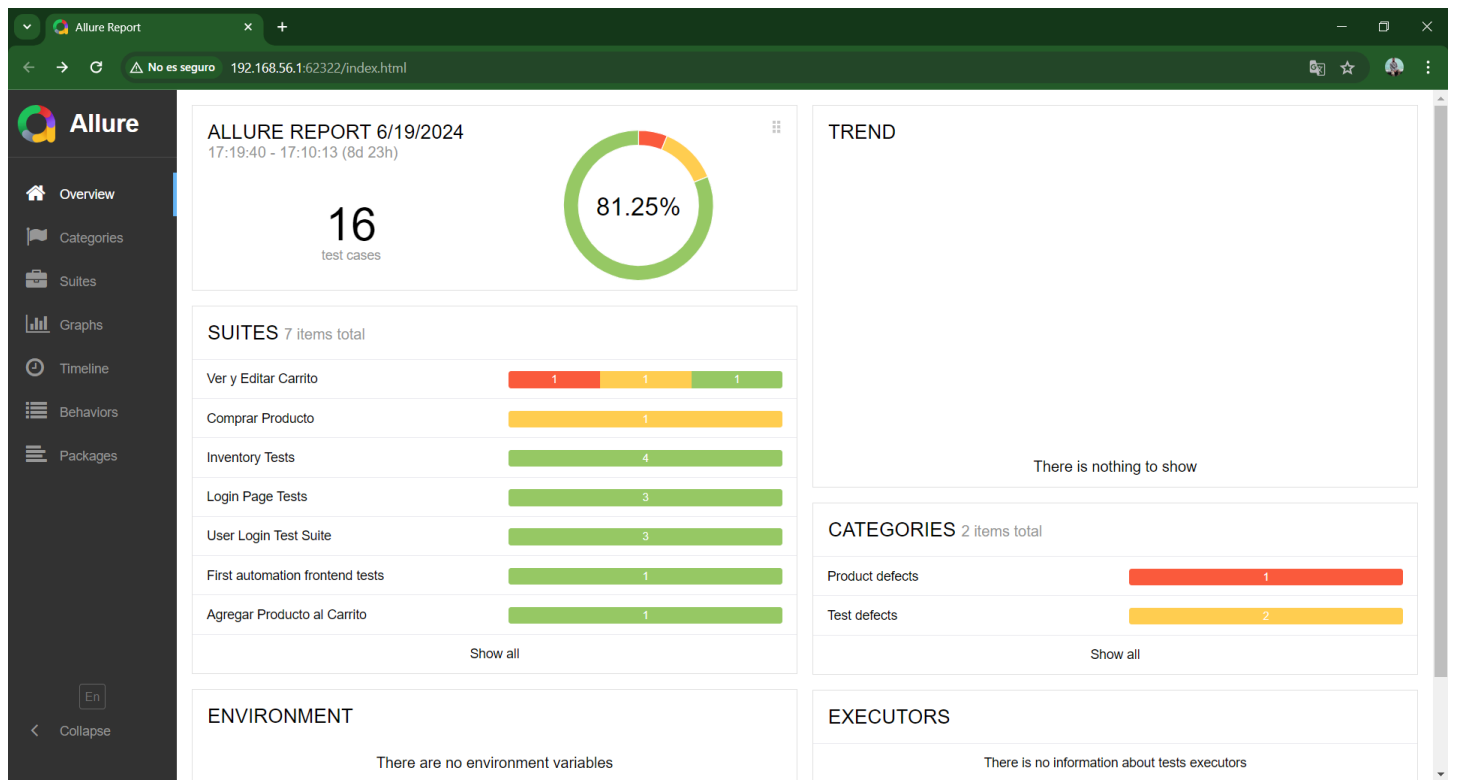
ARCHIVADO

PRÓXIMOS PASOS

Acciones

Tarea o entregable	Estado
Arreglar pruebas de opencart → HU4-HU5	Fallos en la ejecución de los Test, problemas con locators de los button de quitar del carrito y checkout. (Fallo en el button de quitar producto) Fallos en la ejecución de los Test, problemas con locators de los button de añadir el monitor de apple al carrito y realizar la compra. (Fallo en el checkout)
Arreglar pruebas de Rick&Morty → HU2	Fallo en la búsqueda de personajes con el nombre incompleto.

SNAPSHOTS TEST CON FALLOS (Frontend)



SNAPSHOTS TEST FUNCIONALES (Backend)

Gerkhin_17_06_2024_HU1-2 main

Project

- <> compiler.xml
- <> encodings.xml
- <> jarRepositories.xml
- <> misc.xml
- <> uiDesigner.xml
- <> vcs.xml
- <> workspace.xml
- src
 - test
 - java
 - Gerkhin_HU1_2_RickYMorty
 - runners
 - CharacterApiTest
 - testRunner
 - stepDefinitions
 - stepDefinitions
 - resources
 - Gerkhin_HU1_2_RickYMorty

CharacterApiTest.java

```
12
13 //RunWith(Cucumber.class)
14 //CucumberOptions(
15 //    features = {"src/test/resources/features"},
16 //    glue = {"com.vimalselvam.cucumber.stepdefinitions"},
17 //)
18
19 public class CharacterApiTest {
20
21     @BeforeAll
22     public static void setup() {
23         RestAssured.baseURI = "https://rickandmortyapi.com/api";
24     }
25
26     // GET request to search for characters by name
27     @Test
28     public void testSearchCharactersByName() {
29         given().
30             queryParam("name", "Rick Sanchez").
31             when().

```

Run CharacterApiTest

CharacterApiTest (3 sec 775 ms)

- testSearchCharacter 1 sec 917 ms
- testSearchPerformance 279 ms
- testSearchCharactersB 486 ms
- testGetCharacterDetails 397 ms
- testGetCharacterDetails 394 ms
- testDetailPerformance() 302 ms

Tests passed: 6 of 6 tests - 3 sec 775 ms

"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...

Process finished with exit code 0

Gerkhin_17_06_2024_HU1-2 > src > test > java > Gerkhin_HU1_2_RickYMorty > runners > CharacterApiTest 19:14 LF UTF-8 4 spaces