



## CASOS DE PRUEBAS

### TC (Test Cases)

<b>Nombre del documento</b>	Plan de Pruebas
<b>Nombre del fichero:</b>	<i>ISST-TC-V1.0.docx</i>
<b>Versión:</b>	<i>1.0</i>
<b>Fecha de entrega:</b>	<i>6/5/2024</i>
<b>Caso de estudio:</b>	AMaaS-Print3D-Marketplace de fabricación aditiva como servicio
<b>Entrega</b>	<i>Sprint 3</i>

**Autores/as:**

*Sergio Jiménez Sánchez(Developer), Álvaro Sánchez Martínez(Developer)*

**Revisor/a:**

*Silvia Rodríguez Hernández (Developer)*

## INDICE

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>2</b>
1.1	Propósito del documento	2
1.2	Definiciones, acrónimos y abreviaturas	2
1.3	Materiales de referencia	2
1.4	Estructura del documento	2
<b>2</b>	<b>PLAN DE PRUEBAS</b>	<b>2</b>
2.1	Visión global de las pruebas	2
2.2	Recursos, personal y productos de las pruebas	3
2.3	Control y seguimiento de las pruebas	3
2.4	Métricas, herramientas y entorno de pruebas	3
2.5	Calendario de pruebas	3
<b>3</b>	<b>ESPECIFICACIÓN CASOS DE PRUEBA</b>	<b>4</b>
3.1	Caso de prueba 1	2
3.1.1	<i>Descripción</i>	2
3.1.2	<i>Recursos</i>	3
3.1.3	<i>Precondiciones</i>	3
3.1.4	<i>Postcondiciones</i>	3
3.1.5	<i>Flujo de eventos</i>	3
3.1.6	<i>Resultado</i>	
3.2	Caso de prueba 2	4
3.2.1	<i>Descripción</i>	4
3.2.2	<i>Recursos</i>	4
3.2.3	<i>Precondiciones</i>	5
3.2.4	<i>Postcondiciones</i>	5
3.2.5	<i>Flujo de eventos</i>	5
3.2.6	<i>Resultado</i>	
3.3	Caso de prueba 3	5
3.3.1	<i>Descripción</i>	5
3.3.2	<i>Recursos</i>	5
3.3.3	<i>Precondiciones</i>	5
3.3.4	<i>Postcondiciones</i>	6
3.3.5	<i>Flujo de eventos</i>	6
3.3.6	<i>Resultado</i>	
3.4	Caso de prueba 4: Pruebas de historias de usuario con Selenium	6
3.4.1	<i>Prueba 1</i>	6
3.4.2	<i>Prueba 2</i>	6

## INTRODUCCIÓN

### 1.1 Propósito del documento

En este documento de pruebas se va a establecer un conjunto de criterios y escenarios de prueba que permitan validar el funcionamiento básico y las características principales del producto. Estas pruebas están diseñadas para verificar que el MVP cumple con los requisitos mínimos y las expectativas del usuario.

### 1.2 Definiciones, acrónimos y abreviaturas

- **Pruebas de caja blanca:** Pruebas que consideran la estructura interna del módulo.
- **Pruebas de caja negra:** Pruebas que no consideran la estructura interna del módulo.
- **Pruebas de aceptación:** Pruebas que testean el sistema en su conjunto frente a los requisitos.
- **MVP:** Producto mínimo viable
- **JSON (JavaScript Object Notation):** Formato de texto ligero usado para estructurar datos y transmitirlos de un sistema a otro.

### 1.3 Materiales de referencia

Documentos proporcionados por la propia asignatura ISST del Grado de Tecnologías y Servicios de Telecomunicación de la ETSIT

### 1.4 Estructura del documento

Primero, se detallan las pruebas realizadas con la herramienta Postman, incluyendo tanto las rutas definidas como las pruebas llevadas a cabo para estudiar cómo funcionan los controladores, que sirven como punto de entrada a la API desde el frontend al backend. Por último, se presenta un flujo de acción desarrollado y grabado utilizando Selenium.

## 2 PLAN DE PRUEBAS

### 2.1 Visión global de las pruebas

Se han llevado a cabo una serie de pruebas de caja blanca y de caja negra tanto unitarias como de sistema, para comprobar la funcionalidad de la aplicación. Estas pruebas se han realizado partiendo de los casos de uso acordados.

## 2.2 Recursos, personal y productos de las pruebas

Inicialmente se ha organizado una reunión para dejar claras las funcionalidades que se desean lograr con la aplicación. A partir de esto los developers han desarrollado pruebas utilizando la herramienta Postman para acceder a las rutas que funcionan como puntos de entrada de la API. Por último, los ingenieros de pruebas han ejecutado flujos de navegación en la aplicación web interactiva para obtener trazas, a partir de las cuales se ha generado un script utilizando Selenium. Todas estas actividades han sido realizadas por los ingenieros de pruebas para garantizar la calidad y el funcionamiento correcto de la aplicación.

## 2.3 Control y seguimiento de las pruebas

En lo que respecta a las pruebas se ha compartido los archivos JSON generados durante su ejecución. Las pruebas de sistema se llevaron a cabo después de verificar que cada componente individual funcionaba correctamente.

## 2.4 Métricas, herramientas y entorno de pruebas

Por un lado, se han llevado a cabo pruebas unitarias mediante Postman para funcionalidades aisladas. Por otro lado, gracias a Selenium IDE y Selenium WebDriver ha sido posible la grabación y reproducción de los flujos de navegación asociados a cada caso de uso.

## 2.5 Calendario de pruebas

Durante el desarrollo de la aplicación, se ha buscado seguir una metodología de integración continua, por lo que se han ido haciendo pruebas constantemente durante el desarrollo mediante Postman. Por tanto, antes de dar por finalizada una parte del código, se han realizado pruebas de validación comprobando las diversas llamadas a la API.

Fecha	Pruebas
01/05/2024	Reunión para definir casos de prueba
04/05/2024	Pruebas finales relacionadas con registro y impresoras
05/05/2024	Pruebas finales relacionadas con pedidos

### **3 ESPECIFICACIÓN CASOS DE PRUEBA**

#### **3.1 Caso de prueba 1**

##### **3.1.1 Descripción**

Se han hecho peticiones POST en relación al registro de usuarios, añadiendo credenciales inválidas para comprobar que da error a la hora de registrarse

##### **3.1.2 Recursos**

Para la implementación de esta prueba se ha hecho uso de la herramienta Postman

##### **3.1.3 Precondiciones**

No hay

##### **3.1.4 Datos de entrada**

Para el registro del usuario: username demasiado corto, password demasiado corta, email invalido, role invalido

##### **3.1.5 Postcondiciones**

Se realiza la comprobación de que cuando el usuario y sus atributos son inválidos da error y no permite continuar

##### **3.1.6 Flujo de eventos**

1. Usuario realiza el registro

##### **3.1.7 Resultados**

Ver archivo TestAMaaS.postman\_collection.json

#### **3.2 Caso de prueba 2**

##### **3.2.1 Descripción**

Este caso de prueba se centra en peticiones GET, POST, PUT y DELETE relacionadas con las impresoras por parte de un usuario fabricante, se van a realizar peticiones para añadir impresoras, ver impresoras, modificar impresoras y eliminarlas.

##### **3.2.2 Recursos**

En la implementación de esta prueba se ha hecho uso de la herramienta Postman.

##### **3.2.3 Precondiciones**

El usuario debe de estar registrado en la plataforma.

Para las peticiones GET, PUT y DELETE debe haber creada al menos una impresora

##### **3.2.4 Datos de entrada**

Cookie de sesión.

Para la creación de una impresora: modelo, material, máximo de ancho, largo y alto, velocidad de impresión, coste de material, coste de operación, imagen de la impresora y dirección

### **3.2.5 Postcondiciones**

Se debe comprobar que tras el inicio de sesión y creación de una impresora esta se quede guardada en la base de datos y se pueda ver en la pestaña de mis impresoras

### **3.2.6 Flujo de eventos**

1. Usuario inicia sesión en la web
2. Usuario añade una nueva impresora
3. Usuario consulta la información de su impresora
4. Usuario cierra sesión

### **3.2.7 Resultado**

Archivo TestAMaaS.postman\_collection.json

## **3.3 Caso de prueba 3**

### **3.3.1 Descripción**

Este caso de prueba se centra en peticiones GET, PUT y POST para generar un pedido, ver los pedidos generados y ver los pedidos recibidos.

### **3.3.2 Recursos**

Para la implementación de esta prueba se ha hecho uso de la herramienta Postman

### **3.3.3 Precondiciones**

El usuario diseñador debe estar registrado.

El usuario fabricante debe estar registrado.

En el caso de la generación del pedido o de la vista de pedidos generados, el usuario diseñador ha de estar logueado y debe haber una impresora añadida por el fabricante.

En el caso de la vista de pedidos recibidos, el usuario fabricante, propietario de la impresora destinataria, ha de estar logueado y debe haber algún pedido ya generado, si no, no habrá pedidos.

### **3.3.4 Datos de entrada**

Para hacer el pedido hay que introducir los campos de: file, material, color y acabado, escala, cantidad, aceptadoPor, pedidoPor y pagoId, y tener la cookie de sesión del usuario que genera el pedido o del usuario propietario de la impresora a la que se asigna el pedido .

### **3.3.5 Postcondiciones**

El pedido debe quedar guardado en la base de datos y le debe aparecer tanto al diseñador como al fabricante

### 3.3.6 Flujo de eventos

1. Usuario se registra
2. Usuario inicia sesión
3. Usuario sube su pedido
4. Usuario genera pedido

### 3.3.7 Resultado

Archivo TestAMaaS.postman\_collection.json

## 3.4 Caso de prueba 4: Pruebas de historias de usuario con Selenium

Se prueban las historias de usuario con selenium para verificar que funcionan.

### 3.4.1 Prueba 1

#### Descripción de pasos seguidos:

Generar usuario epepe-Loguear usuario epepe-Seleccionar impresora de bpepe-Generar pedido-Crear impresora-Modificar impresora-Eliminar impresora-Desloguear usuario epepe

#### Archivo de grabación:

UserstoriescreateUserSigninGenerateorderCreatePrinterModifyPrinterDeletePrintersignOutTest.java

### 3.4.2 Prueba 2

#### Descripción de pasos seguidos:

Loguear usuario bpepe-Ver pedidos para imprimir (para la impresora de bpepe)-seleccionar un pedido-descargar el archivo-marcas como completado-Desloguear usuario bpepe

#### Archivo de grabación:

UserstoriesLogearbpepeVerOrdenesRecibidasdescargarArchivoCompletarOrdendesloguearbpepeTest.java