

# Plan de Pruebas

## “Tumangamandapio”



Asignatura: Desarrollo Web Ágil de API  
y SPA

Maestro: David Antonio Soberanis Ramírez

**Elaborado por:**

Avila Pacheco David de Jesús  
Echeverría León Educador Leonel  
Pantoja Vázquez Rodrigo Jesús  
Urtecho Quintal Rodrigo  
Vázquez Flores Jorge Martín

Fecha de entrega: 23 de mayo de 2022

# Índice

<b>Índice</b>	2
<b>Introducción</b>	3
<b>Objetivos</b>	3
<b>Alcance</b>	3
<b>Entorno y configuración de pruebas</b>	3
4.1. Requisitos de hardware	3
4.2. Requisitos ambientales	4
4.3. Base de datos de pruebas	4
4.4. Criterios de aprobación/rechazo	5
4.5. Criterios de suspensión	5
<b>Estrategías de pruebas</b>	5
5.1. Pruebas	5

# 1.Introducción

Tumangamandapio es una página web de compras de productos de entretenimiento (mangas, animes y comics), donde podrías ver cuales son los mejores mangas, animes y comics del momento, y comprarlos para sean enviados directos a tu casa.

## **Funciones principales:**

- Consultar los mejores cómics del momento.
- Consultar los mejores mangas del momento.
- Consultar los mejores animes del momento.
- Iniciar sesión
- Registrarse
- Validar email
- Enviar comentarios
- Añadir productos a tu carrito
- Comprar productos.

## 2.Objetivos

- Identificar errores del productos
- Validar y verificar que el producto no tenga errores.
- Entregar un producto con un máximo de 1% de fallas.
- Entregar los módulos principales

## 3.Alcance

Abarcar la funcionalidad de los módulos por separado y en conjunto, así como los requerimientos definidos del sistema. Este documento será para los testers para conocer como realizar las pruebas consideradas para este sistema, y poder cumplir con los objetivos.

## 4.Entorno y configuración de pruebas

### 4.1. Requisitos de hardware

Requisitos mínimos:

Tarjeta gráfica: Intel HD 4000 en PC; Intel Iris Pro 5200.

Procesador: Core i3-3225 de 3.3 GHz o AMD Sempron 1.8 GHz.

Memoria: 4 GB de RAM.

SO: Windows 7/8/10 de 64 bits o Mac OS Mojave 10.14.6 en adelante.

## 4.2. Requisitos ambientales

Conexión: Conectividad a internet constante.

Navegador web: Chrome

Servidor web: Xampp

Base de datos: MySQL

IDE/Editor de código: Visual Studios

Herramienta de testing: Karma

## 4.3. Base de datos de pruebas

### Api de los productos disponibles de la tienda

Los datos que se usan para las pruebas:

id	type	name	img	price	stockQty	created_date	update_date
----	------	------	-----	-------	----------	--------------	-------------

### API de top comics

Se usan los datos que proporcione la API en ese momento, ya que estos están en constante cambio y no se puede controlar cuáles serán los datos a utilizar. Pero los campos que se usan son:

title	url	img
-------	-----	-----

### API de mangas

Se usan los datos que proporcione la API en ese momento, ya que estos están en constante cambio y no se puede controlar cuáles serán los datos a utilizar. Pero los campos que se usan son:

title	url	img
-------	-----	-----

### API de top animes

Se usan los datos que proporcione la API en ese momento, ya que estos están en constante cambio y no se puede controlar cuáles serán los datos a utilizar. Pero los campos que se usan son:

title	url	img
-------	-----	-----

#### 4.4. Criterios de aprobación/rechazo

Los criterios de aprobación es que todas las pruebas resulten con éxito, ninguna de las pruebas puede quedarse con fallido. Por lo que todos los casos contrarios de no tener completos las pruebas o encontrar errores no serán aceptados los resultados.

#### 4.5. Criterios de suspensión

Si en el momento de la aplicación de la medición se encuentra algún error, éste será suspendido y se reportará, y este se reanudará una vez que el error se haya solucionado.

En caso de que los dispositivos/programas dejen de funcionar, la medición será suspendida, y ésta se reanudará cuando estos hayan sido arreglados o reemplazados.

### 5. Estrategías de pruebas

Para la ejecución de las pruebas se utilizará la herramienta de Angular: Karma.

Para esto se utiliza el archivo con terminación '.spec.ts' que se genera al crear los componentes o servicios de Angular, en este irá toda la configuración para hacer las pruebas de cada módulo.

Para ejecutar las pruebas se usará el comando ng test, en donde abrirá una página web en la cual se encontrarán los resultados de las pruebas de todo el proyecto, las cuales saldrán en verde las exitosas (specs) y en rojas las fallidas (failure).

#### 5.1. Pruebas

##### **Registro de usuarios**

###### **Módulo**

Componente register (archivos: html y ts).

###### **Aplicación y datos**

Los datos que se usarán son:

correo: [ed\\_leo1411@hotmail.com](mailto:ed_leo1411@hotmail.com)

usuario: ednover

contraseña: ednover123

confirmación: ednover123

Los datos irán en la instrucción:

`component.validateForm.controls[nombredelcampo].setValue(valor)`

###### **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se agregó con éxito el usuario.

##### **Inicio de sesión**

## **Módulo**

Componente login (archivos: html y ts)

### **Aplicación y datos**

Los datos que se usarán son:

correo: [tilin@gmail.com](mailto:tilin@gmail.com)

contraseña: etesech

Los datos irán en la instrucción:

`component.validateForm.controls[nombredelcampo].setValue(valor)`

### **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se encontró al usuario y podrá acceder a la página web.

## **Validar email (API externa)**

### **Módulo**

Componente api1 (archivos: html y ts)

### **Aplicación y datos**

Los datos que se usarán son:

correo: [tilin@gmail.com](mailto:tilin@gmail.com)

Los datos irán en la instrucción:

`component.sendEmail(correo);`

### **Resultado**

El resultado esperado es que la función devuelve false, haciendo referencia a que el correo ya existe, por lo que es un correo válido.

## **Consultar comics (API externa)**

### **Módulo**

Componente api2 (archivos: html y ts)

### **Aplicación y datos**

Al ser una API, y solo es petición ya determinada en el servicio, no es necesario mandar ningún dato.

### **Resultado**

El resultado esperado es que la lista de comicsRecibidos sea diferente de null, haciendo referencia a que se pudo realizar la petición y que si devolvió lo consultado.

## **Consultar mangas (API externa)**

### **Módulo**

Componente api3 (archivos: html y ts)

### **Aplicación y datos**

Al ser una API, y solo es petición ya determinada en el servicio, no es necesario mandar ningún dato.

### **Resultado**

El resultado esperado es que la lista de MangasRecibidos sea diferente de null, haciendo referencia a que se pudo realizar la petición y que si devolvió lo consultado.

## **Consultar animes (API externa)**

### **Módulo**

Componente api4 (archivos: html y ts)

### **Aplicación y datos**

Al ser una API, y solo es petición ya determinada en el servicio, no es necesario mandar ningún dato.

### **Resultado**

El resultado esperado es que la lista de AnimesRecibidos sea diferente de null, haciendo referencia a que se pudo realizar la petición y que si devolvió lo consultado.

## **Consultar catálogo tienda**

### **Módulo**

Componente tienda (archivos: html y ts)

### **Aplicación y datos**

Al ser una consulta a la base de datos de los artículos existentes, no es necesario mandar ningún dato y ya se encuentra todo configurado en el archivo.

### **Resultado**

El resultado esperado es que la lista de ArticulosRecibidos sea diferente de null, haciendo referencia a que se pudo realizar la petición y que si devolvió lo consultado.

## **Enviar comentarios (API externa)**

### **Módulo**

Componente api5 (archivos: html y ts)

### **Aplicación y datos**

Los datos que se usarán son:

correo: [ed\\_leo1411@hotmail.com](mailto:ed_leo1411@hotmail.com)

nombre: Eduardo Echeverria

comentarios: La página me gusta y es interesante.

Los datos irán en la instrucción:

`component.validateForm.controls[nombredelcampo].setValue(valor)`

### **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se ha enviado el correo con éxito.

## **Eliminar producto (CRUD)**

### **Módulo**

Componente crud-catalogo (archivos: html y ts)

## **Aplicación y datos**

Los datos que se usarán son:

type: anime

name: Gintama

img: <https://cdn.myanimelist.net/images/anime/10/73274.jpg>

price: 2068.30

stockQty: 3

Los datos irán en la instrucción:

`component.datosProducto.controls[nombredelcampo].setValue(valor)`

Y en la función 'expect' y el componente crud-catalogo llamará a la función `eliminarProducto(component.datosProducto)`.

## **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se ha eliminado el producto con éxito.

## **Agregar producto (CRUD)**

### **Módulo**

Componente crud-catalogo (archivos: html y ts)

### **Aplicación y datos**

Los datos que se usarán son:

type: anime

name: Gintama

img: <https://cdn.myanimelist.net/images/anime/10/73274.jpg>

price: 2068.30

stockQty: 3

Los datos irán en la instrucción:

`component.datosProducto.controls[nombredelcampo].setValue(valor)`

Y en la función 'expect' y el componente crud-catalogo llamará a la función `agregarProducto(component.datosProducto)`.

## **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se ha eliminado el producto con éxito.

## **Editar producto (CRUD)**

### **Módulo**

Componente crud-catalogo (archivos: html y ts)

### **Aplicación y datos**

Los datos que se usarán son:

type: anime

name: Gintama

img: <https://cdn.myanimelist.net/images/anime/10/73274.jpg>

price: 2068.30

stockQty: 3



Los datos irán en la instrucción:

```
component.datosProducto.controls[nombre del campo].setValue(valor)
```

Y en la función 'expect' y el componente crud-catalogo llamará a la función editarProducto(component.datosProducto).

### **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se ha eliminado el producto con éxito.

## **Verificar tipo de cuenta**

### **Módulo**

Componente tienda (archivos: html y ts)

### **Aplicación y datos**

Al ser un método que solo es para determinar el tipo de cuenta en la que se encuentra, al momento de estar en la tienda, no se necesita enviar nada y ya todo está configurado en el archivo spec.

### **Resultado**

El resultado esperado es que la función devuelve verdadero, haciendo referencia a que se ha accedido con la cuenta adecuada.