

# Frikomicks



<b>1.Introducción.....</b>	<b>3</b>
<b>2.Objetivos.....</b>	<b>4</b>
Objetivos Específicos.....	4
<b>3. Temporalización y coste de la aplicación.....</b>	<b>5</b>
<b>4.Funcionalidades de la aplicación.....</b>	<b>7</b>
2.2 Gestión de Productos.....	7
2.3 Carrito de Compras.....	8
2.4 Gestión de Subastas.....	9
2.5 Perfil de Usuario.....	9
2.6 Administración.....	9
2.7 Seguridad y Validaciones.....	10

# 1.Introducción

**Frikomics** es una innovadora plataforma web dedicada a la venta y gestión de cómics, creada con el objetivo de brindar a los amantes de las historietas una experiencia de usuario moderna, intuitiva y eficiente. A través de una interfaz amigable y funcionalidades bien estructuradas, los usuarios pueden explorar fácilmente un extenso catálogo de cómics de diversos géneros, autores y editoriales, lo que les permite descubrir nuevas historias o adquirir sus títulos favoritos de manera rápida y segura.

El sistema está diseñado para adaptarse a dos tipos principales de perfiles:

- **Usuarios normales**, quienes tienen acceso a funciones como navegación por el catálogo, búsqueda personalizada, registro de cuenta, administración de su historial de compras, y por supuesto, la posibilidad de adquirir productos de forma sencilla y confiable.
- **Administradores**, responsables de la gestión interna de la plataforma. Ellos pueden añadir, editar o eliminar cómics del catálogo, gestionar el inventario, actualizar precios y descripciones, así como supervisar y administrar la base de datos de usuarios registrados.

Gracias a su enfoque centrado en la usabilidad y en las necesidades tanto de compradores como de gestores, Frikomics se posiciona como una solución completa para el comercio electrónico de cómics, combinando tecnología, organización y pasión por el noveno arte.

## 2.Objetivos

Desarrollar una plataforma web robusta, escalable y segura para la gestión, visualización y comercialización de cómics, que permita diferenciar funcionalidades según el tipo de usuario (cliente o administrador), incorporando prácticas modernas de desarrollo full-stack, arquitectura limpia y persistencia de datos.

### Objetivos Específicos

1. **Diseñar e implementar una API RESTful con Laravel** que gestione las operaciones de usuarios, autenticación, productos (cómic), pedidos y roles, garantizando un sistema desacoplado y reutilizable.
2. **Establecer una estructura de control de acceso basada en roles** que permita restringir ciertas rutas o acciones según si el usuario es un cliente normal o un administrador, mediante middlewares personalizados.
3. **Desarrollar un sistema de autenticación y autorización** utilizando tokens (como Sanctum o Passport), garantizando la seguridad de las operaciones en el frontend y la validación en el backend.
4. **Integrar una base de datos relacional (MySQL)** que permita un manejo eficiente de la información del catálogo, usuarios, pedidos y stock, con relaciones definidas entre modelos y validaciones a nivel de base de datos (por ejemplo, constraints para el campo `role`).
5. **Permitir a los usuarios clientes** visualizar y filtrar cómics disponibles, registrarse e iniciar sesión, gestionar su perfil, y realizar pedidos de productos mediante una interfaz limpia y responsive.
6. **Permitir a los administradores** acceder a un panel privado donde puedan crear, editar, eliminar y listar cómics, así como ver y gestionar usuarios registrados y sus roles, mediante componentes de administración seguros y optimizados.
7. **Documentar todo el sistema a nivel funcional y técnico**, incluyendo diagramas de casos de uso, estructura de base de datos, flujos de autenticación y ejemplos de endpoints, para facilitar el mantenimiento, la ampliación y la evaluación del proyecto.

### 3. Temporalización y coste de la aplicación

#### 1. Analisis y preparacion:

Análisis de requisitos	10 horas
Diseño de casos de uso y modelo de datos	
Planificación del proyecto	

#### 2. Base de datos y entorno Docker:

Creación de Base de datos Mysql	10 horas
Dockerizacion de servicios	
Manual de instalación	

#### 3. Backend en Laravel

CRUD productos	8 horas
Carrito de compras	6 horas
Sistema de compras	5 horas
Subastas con WebSockets	8 horas
Seguridad en backend	6 horas

#### 4. Frontend en Angular

Visualización de productos	6 horas
Carrito e interfaz de compra	6 horas
Filtros y búsqueda	3 horas
Pujas en tiempo real	6 horas
Interfaz de publicación (admin)	6 horas
Estilo y diseño responsive	4 horas
Componente de historial de compras	6 horas

5. **Testing y pruebas Unitarias:** 6 horas.

6. **Documentación del proyecto:** 6 horas.

El tiempo estimado sería de unas 135 horas. Si tenemos en cuenta que el costo de la aplicación sería unos 30€ por hora nos quedaría la aplicación costaría unos 4050€.

## 4. Funcionalidades de la aplicación

- **F1. Registro de usuario**

- Descripción: Permite a un nuevo usuario registrarse en la plataforma mediante un formulario.
- Reglas:
  - El email debe ser único.

- **F2. Inicio de sesión**

- Descripción: Inicia sesión con email y contraseña.
- Reglas:
  - Verifica credenciales.
  - Devuelve token para sesiones autenticadas.

- **F3. Middleware de roles**

- Descripción: Controla el acceso a rutas protegidas.
- Reglas:
  - Usuario **admin** accede a recursos restringidos.
  - Usuario **normal** accede solo a funcionalidades estándar.

### 4.2 Gestión de Productos

- **F4. Crear producto (admin)**

- Campos: título, descripción, imagen, precio, categoría.
- Reglas:
  - Solo accesible por el rol administrativo.

- **F5. Editar producto (admin)**

- Descripción: Modifica datos de un producto existente.
- Reglas:
  - Solo accesible por el rol admin.

- **F6. Eliminar producto (admin)**
  - Descripción: Elimina un producto de la base de datos.
  - Reglas:
    - Solo accesible por el rol admin.
- **F7. Ver productos**
  - Descripción: Lista todos los productos.
  - Accesible por todos los usuarios (público y logueado).

### 4.3 Carrito de Compras

- **F8. Añadir al carrito**
  - Descripción: Añade un producto al carrito de usuario.
  - Reglas:
    - Debe estar logueado.
- **F9. Eliminar del carrito**
  - Descripción: Elimina producto individual del carrito.
- **F10. Checkout**
  - Descripción: Simula compra de productos.
  - Reglas:
    - Calcula total.
    - Limpia el carrito después.
    - Salta un mensaje de confirmación
    - El producto se mueve al historial de compras



## 4.4 Gestión de Subastas

- **F11. Crear subasta (admin)**
  - Descripción: Define un producto y tiempo límite.
- **F12. Participar en subasta**
  - Descripción: Los usuarios pueden realizar pujas en tiempo real.
  - Reglas:
    - Usa Websockets.
    - La mayor puja se actualiza dinámicamente.

## 4.5 Perfil de Usuario

- **F13. Ver perfil**
  - Descripción: Muestra datos del usuario actual.
- **F14. Editar perfil**
  - Campos: nombre, email, contraseña.
  - Validaciones aplicadas a cada campo.
- **F15. Eliminar cuenta**
  - Descripción: Elimina usuario y sus relaciones (carrito, pujas, etc.).
  - Requiere autenticación y confirmación.

## 4.6 Administración

- **F16. Dashboard del administrador**
  - Descripción: Visualización de productos, subastas y usuarios.
  - Permite acceso a edición/creación masiva.

## 4.7 Seguridad y Validaciones

- **F17. Validación de formularios**

- En frontend (Angular): con formularios reactivos.
- En backend (Laravel): mediante **Request** y validadores.

- **F18. Protección de rutas**

- Middleware de autenticación (**auth**) y autorización (**admin**).

- **F19. Gestión de errores**

- Devuelve mensajes estandarizados JSON.
- Códigos HTTP según tipo (422, 401, 403, 404...).