



DESARROLLO DE SISTEMAS DE SOFTWARE BASADOS  
EN COMPONENTES Y SERVICIOS  
MÁSTER INGENIERÍA INFORMÁTICA

**Práctica 3: Desarrollo de app para acceder a un  
catálogo de productos, basado en la aplicación  
Web RESTful - Documentación Técnica**

---

**Autores**

Pablo Valenzuela Álvarez y Ana Lucía Vico Martínez

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

Granada, 17 Diciembre de 2024



## 1. Introducción

Esta documentación es una descripción técnica de la arquitectura de la aplicación de comercio electrónico desarrollada para dispositivos “Android”. Se va a hablar sobre la estructura de la aplicación, su arquitectura, el flujo de datos y la autenticación y autorización implementada.

## 2. Arquitectura de la Aplicación

La aplicación está estructurada bajo un modelo MVC (Modelo-Vista-Controlador), lo que facilita la separación de responsabilidades y el mantenimiento del código. La arquitectura se compone de los siguientes módulos:

- Modelo: Encargado de representar los datos y la lógica de negocio de la aplicación. Contiene la clase ‘Producto’.
- Vista: Representa la interfaz gráfica del usuario, con archivos0 “XML” y actividades “Java”.
- Controlador: Gestiona las interacciones del usuario y coordina la comunicación entre el Modelo y la Vista.

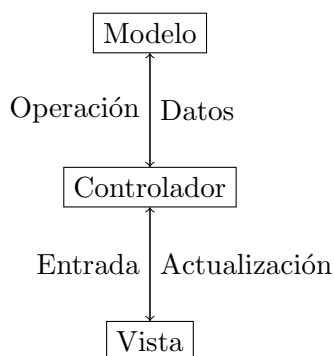


Figura 1: Diagrama de arquitectura MVC

### 3. Diagramas de Clases

Las principales clases de la aplicación son:

- **Producto:** Representa los productos del catálogo, con los atributos nombre y precio.
- **MainActivity:** Gestiona la visualización de los productos.
- **CartActivity:** Permite al usuario gestionar el carrito de compras (añadir o eliminar productos).
- **ThankYouActivity:** Confirma la compra y muestra el resumen de los productos seleccionados.
- **AdminActivity:** Da las funcionalidades para que los administradores gestionen los productos (añadir, editar y eliminar).
- **API REST:** Gestiona las operaciones relacionadas con los productos.
- **LoginActivity:** Autentica a los usuarios y les permite acceder a la aplicación.
- **AddProductActivity:** Pantalla destinada a la inserción de nuevos productos en el catálogo.
- **EditProductActivity:** Permite la edición de productos existentes en el catálogo.
- **MapActivity:** Muestra un mapa con las ubicaciones de las tiendas físicas disponibles.

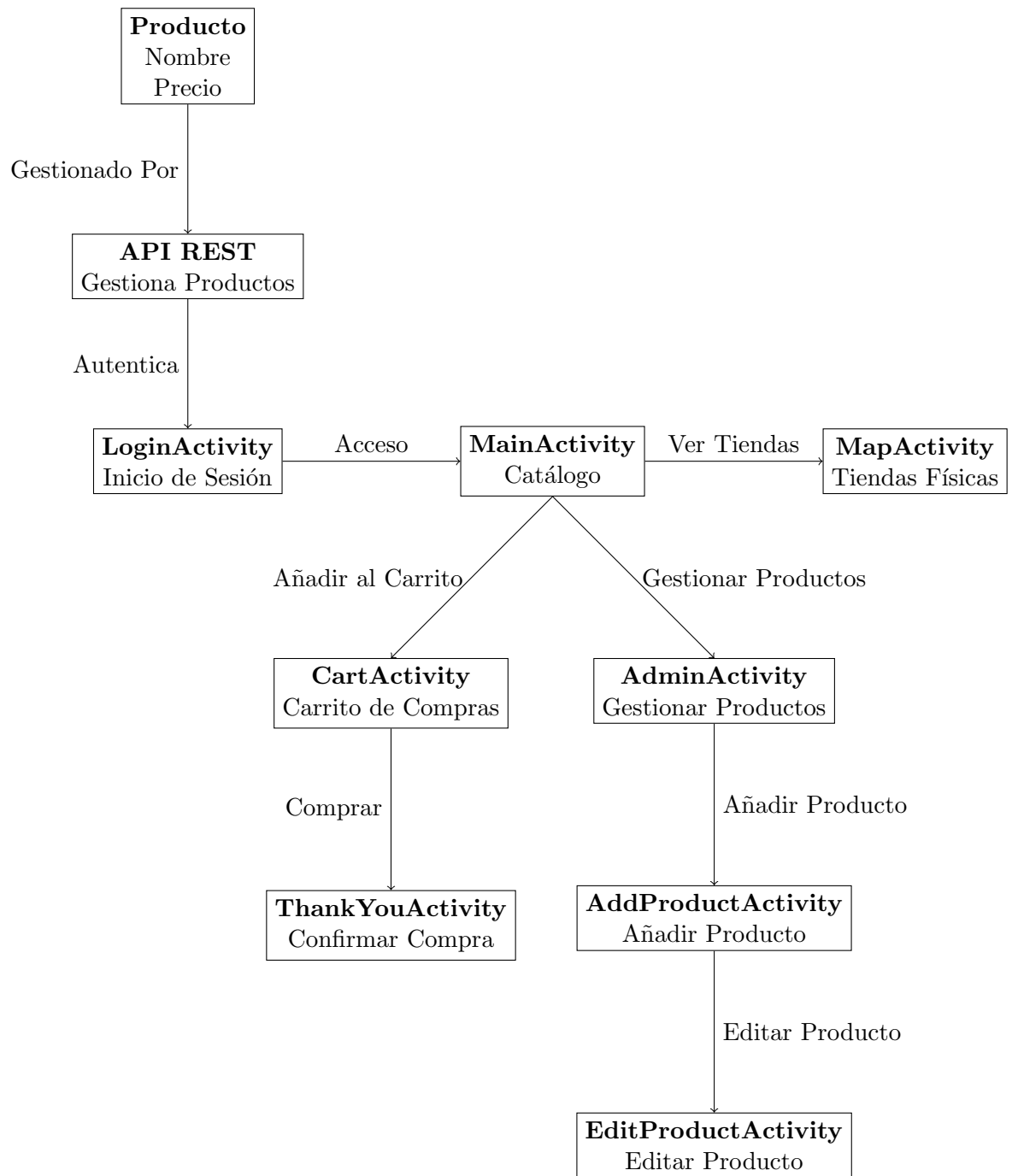


Figura 2: Diagrama de clases y actividades actualizado para la aplicación

## 4. Flujo de Datos

El flujo de datos en la aplicación sigue el siguiente esquema:

1. Interacción del usuario con la interfaz (Vista): El usuario inicia sesión, navega por los productos o realiza operaciones sobre el carrito.
2. Controlador: Las actividades actúan como intermediarios y realizan peticiones a la “API REST” local mediante “Retrofit”.
3. Modelo: La “API REST” responde con datos en formato “JSON”, que son procesados y convertidos a objetos Java mediante “Gson”.
4. Actualización de la Vista: Los datos procesados se muestran al usuario en tiempo real.

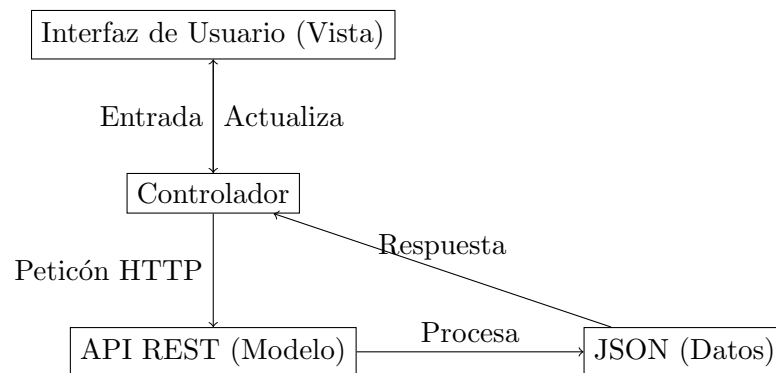


Figura 3: Flujo de datos en la aplicación

## 5. Autenticación y Autorización

La aplicación utiliza un sistema de autenticación y autorización para garantizar la seguridad en la gestión de productos.

### 5.1. Autenticación

- Al iniciar la aplicación, el usuario debe proporcionar sus credenciales (usuario y contraseña).
- Las credenciales se envían a la “API REST” mediante una petición “POST”.
- La “API” verifica las credenciales y devuelve una respuesta indicando si el usuario es válido.

### 5.2. Autorización

- Una vez autenticado, se asigna un rol al usuario (cliente o administrador).
- Los clientes solo pueden acceder a funciones como la vista de productos y la gestión del carrito.
- Los administradores tienen acceso a funcionalidades exclusivas como la edición, eliminación y creación de productos.

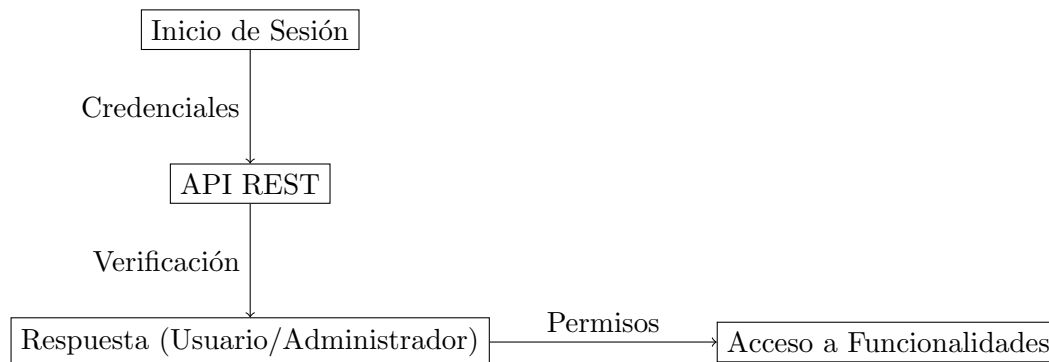


Figura 4: Proceso de autenticación y autorización

## 6. Conclusión

La presente documentación detalla los aspectos técnicos clave de la aplicación, incluyendo su arquitectura general, diagramas de clases, flujo de datos y el sistema de autenticación y autorización. Esta estructura permite una organización clara y separación eficiente de las tareas de cada archivo.

La integración con la **API REST** facilita la gestión de datos, dando un flujo constante y predecible entre la interfaz de usuario y el servidor. Además, los diagramas reflejan cómo cada componente interactúa, desde la solicitud de datos hasta su visualización en la aplicación.

Por último, el mecanismo de autenticación garantiza la seguridad en el acceso, diferenciando entre usuarios normales y administradores. Permitiendo la adaptación de la experiencia del usuario según su rol.

En definitiva, la aplicación presenta una solución funcional y escalable que responde a las necesidades del proyecto, con un diseño claro y una implementación eficiente de sus funcionalidades.