



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

PRÁCTICA 3

Documentación técnica

Autores

Sergio Hervás Cobo y Adriano García Giralda



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, Diciembre de 2024

Índice

1. Introducción	2
2. Documentación Técnica	2
2.1. Arquitectura del Sistema	2
2.2. Diagrama de Clases	3
2.3. Diagramas de secuencia	3
2.4. Flujo de Datos	6
2.5. Autenticación y Autorización	6

1. Introducción

Esta documentación describe la arquitectura técnica de la práctica desarrollada para la asignatura de DSS. El sistema está compuesto por dos partes principales que trabajan de manera integrada:

- **Aplicación móvil:** Implementada en Android Studio utilizando Kotlin. Ofrece una interfaz de usuario para interactuar con las funcionalidades del sistema, como la navegación por el catálogo de productos, la gestión del carrito y la ubicación de tiendas cercanas.
- **Backend:** Implementado en Java utilizando el framework Spring Boot (P1). Proporciona una API REST para la comunicación con la aplicación móvil y un sistema de plantillas Thymeleaf para la aplicación web. Este último permite renderizar páginas dinámicas, gestionar la interfaz web y manejar la autenticación y autorización para usuarios y administradores.

2. Documentación Técnica

2.1. Arquitectura del Sistema

El sistema sigue una arquitectura cliente-servidor, donde la aplicación móvil actúa como cliente que consume las APIs RESTful expuestas por el backend. La aplicación móvil está organizada utilizando una estructura tradicional basada en **Activities** y **Layouts**, mientras que el backend emplea principios de diseño modular.

- **Backend:**
 - Implementa un patrón de arquitectura RESTful para garantizar la separación de responsabilidades y escalabilidad.
 - Las APIs están organizadas en controladores (Controllers) y gestionadas mediante servicios (Services).
 - Utiliza Spring Security para implementar autenticación y autorización basadas en roles.
 - La persistencia de datos se realiza con JPA y una base de datos H2 embebida.
- **Aplicación móvil:**
 - Organizada con una estructura jerárquica donde cada **Activity** representa una funcionalidad principal, vinculada a su correspondiente **Layout XML**.

- Utiliza **Intents** para la navegación entre pantallas y la transmisión de datos.
- Integra Retrofit para el consumo de las APIs REST y Gson para la serialización de datos.

2.2. Diagrama de Clases

Se ha realizado el siguiente diagrama de clases. Por su gran extensión, se deja como fichero en la entrega para una mejor visualización.

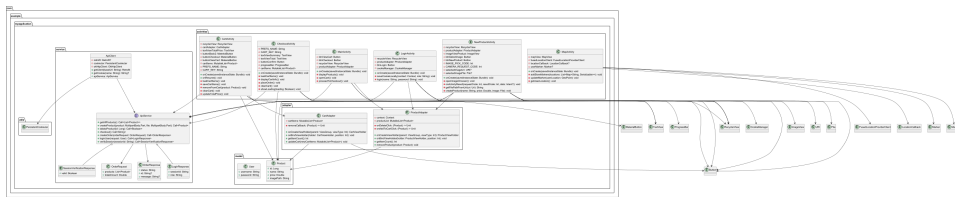


Figura 1: Diagrama de clases

2.3. Diagramas de secuencia

El primer diagrama ilustra el flujo de interacción de un administrador en una aplicación de gestión de productos para crear un nuevo producto. El administrador abre la aplicación, lo que provoca que la pantalla principal haga una solicitud para obtener todos los productos al servicio de API. El servicio de API envía una solicitud GET al controlador de productos, que devuelve una lista de productos al servicio de API, el cual la envía de vuelta a la pantalla principal. La lista de productos se muestra al administrador. Luego, el administrador decide crear un nuevo producto, lo que abre un formulario en la pantalla de creación de productos. El administrador llena los detalles del producto y envía el formulario. La pantalla de creación de productos envía los detalles al servicio de API para crear el producto, lo que provoca una solicitud POST al controlador de productos. El controlador confirma que el producto fue creado, y el servicio de API informa a la pantalla de creación de productos. Finalmente, la pantalla de creación de productos actualiza la lista de productos en la pantalla principal, y el administrador ve la lista actualizada.

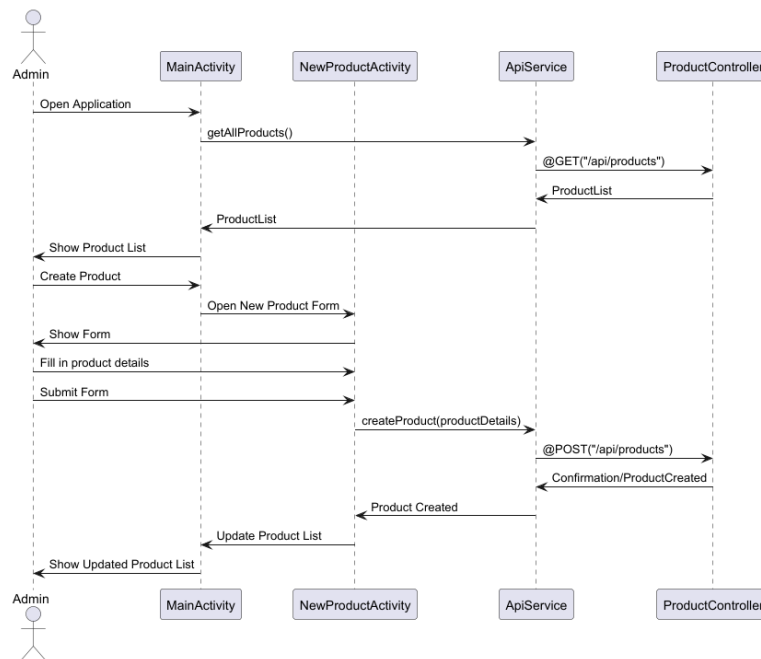


Figura 2: Diagrama de secuencia para crear producto

El segundo diagrama describe el flujo de interacción de un administrador en una aplicación de gestión de productos para borrar uno de estos. El administrador abre la aplicación, lo que provoca que la pantalla principal haga una solicitud para obtener todos los productos al servicio de API. El servicio de API envía una solicitud GET al controlador de productos, que devuelve una lista de productos al servicio de API, el cual la envía de vuelta a la pantalla principal. La lista de productos se muestra al administrador. Luego, el administrador selecciona un producto para eliminar. La pantalla principal solicita al servicio de API que elimine el producto, lo que provoca que se realice una solicitud DELETE al controlador de productos. El controlador confirma la eliminación, y el servicio de API informa a la pantalla principal sobre el éxito de la operación. Finalmente, la pantalla principal muestra un mensaje de éxito al administrador.

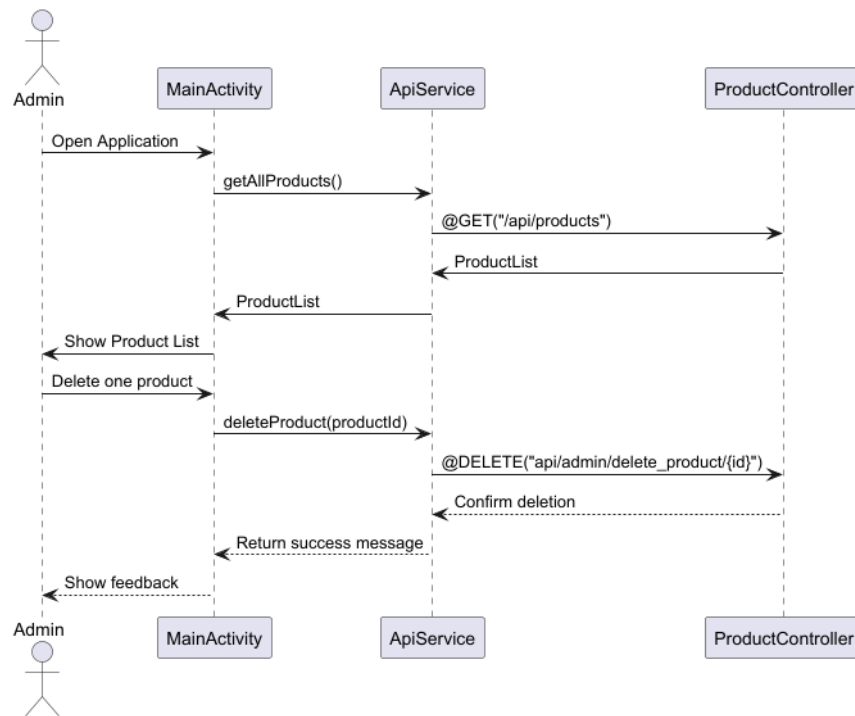


Figura 3: Diagrama de secuencia para eliminar producto

El último diagrama ilustra el flujo de interacción de un usuario en una aplicación de compras. El usuario comienza cargando los productos en la pantalla principal, luego abre el carrito de compras donde se muestran los artículos añadidos. Al hacer clic en “Checkout”, se abre la pantalla de pago, mostrando la factura y las opciones de pago. El usuario confirma el pago, y la aplicación envía la confirmación al servicio de API, que realiza una solicitud POST al controlador de pedidos. Este último procesa el pago y devuelve una respuesta exitosa al servicio de API, que a su vez informa a la pantalla de pago sobre el éxito. Finalmente, la pantalla de pago notifica al carrito de compras y el usuario ve un mensaje de éxito en la pantalla del carrito.

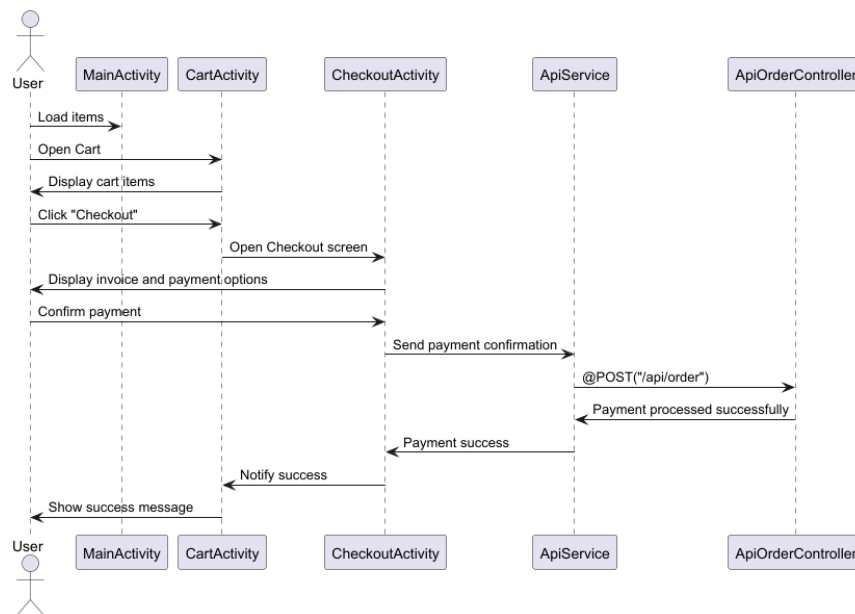


Figura 4: Diagrama de secuencia para comprar

2.4. Flujo de Datos

El flujo de datos entre el cliente y el servidor sigue estos pasos:

1. Solicitud desde la aplicación móvil: El cliente envía peticiones HTTP al backend (e.g., `GET /api/products` para obtener productos).
2. Procesamiento en el backend: El servidor procesa las solicitudes, accede a la base de datos y devuelve respuestas en formato JSON.
3. Respuesta al cliente: La aplicación móvil recibe los datos, los procesa y los muestra en la interfaz gráfica mediante RecyclerView u otros componentes.

2.5. Autenticación y Autorización

La autenticación y autorización se implementan de la siguiente manera:

- Autenticación:
 - Los usuarios se autentican utilizando un formulario de inicio de sesión en la aplicación Android.
 - Se genera la cookie `JSESSIONID` para mantener la sesión del usuario. Esto se realiza añadiendo un filtro a la configuración del `WebConfigSecurity` de la aplicación SpringBoot.

- Para que el usuario no tenga que volver a autenticarse cada vez que cierra y abre la aplicación, se guarda la cookie en memoria del dispositivo móvil con SharedPreferences.
- Autorización:
 - Los roles de usuario (administrador y cliente) se manejan mediante Spring Security.
 - Los administradores tienen acceso a operaciones avanzadas como la gestión de productos, mientras que los clientes solo pueden interactuar con el catálogo y realizar compras. Se han protegido las rutas de esta forma para que, de esta forma, si ocurre algún problema de visibilidad en la aplicación o alguna persona externa intenta entrar a la app mediante herramientas como Postman, necesite autorización. Sin embargo, se ha controlado para que la aplicación no muestre los botones propios de un administrador al usuario normal, y de esta forma garantizamos que protegemos dos veces accesos no autorizados