

Anteproyecto: EcoTrade - Sistema Integral de Compraventa de Segunda Mano

1. Título del Proyecto

EcoTrade: Plataforma Multiplataforma de Gestión y Venta de Productos de Segunda Mano

2. Introducción

En el contexto actual, la economía circular ha cobrado una relevancia fundamental. La necesidad de reutilizar productos para reducir la huella de carbono, sumada a la búsqueda de ahorro económico por parte de los usuarios, ha disparado la demanda de mercados de segunda mano.

Este proyecto propone el desarrollo de un sistema informático centralizado llamado **"EcoTrade"**. La solución tecnológica se fundamenta en un backend robusto desarrollado con **Spring Boot**, el cual actuará como el núcleo lógico y de datos. Este núcleo alimentará simultáneamente a tres interfaces de usuario distintas: una web pública para el acceso universal, una aplicación de escritorio para la gestión administrativa intensiva y una aplicación móvil Android para la accesibilidad inmediata y geolocalizada.

3. Objetivos

Objetivo General

Diseñar e implementar un ecosistema de software escalable para la compraventa de artículos usados, garantizando la sincronización de datos en tiempo real entre un servidor web, clientes de escritorio y dispositivos móviles.

Objetivos Específicos

1. **Desarrollar el Backend (API RESTful):** Construir un servidor central con **Spring Boot** que gestione la lógica de negocio, la persistencia de datos (MySQL/PostgreSQL) y la seguridad (Spring Security con JWT).
2. **Implementar la Aplicación Web:** Crear un portal web (utilizando Spring MVC con Thymeleaf o un framework JS como React/Angular consumiendo la API) que permita a los usuarios registrarse, publicar anuncios y buscar productos.
3. **Desarrollar la Aplicación de Escritorio (PC):** Crear una aplicación (preferiblemente en **JavaFX** o Swing para mantener el ecosistema Java) destinada a la administración de la plataforma (moderación de anuncios, gestión de usuarios baneados, estadísticas de ventas) que consuma los datos del servidor.
4. **Desarrollar la Aplicación Android:** Programar una app nativa que permita a los usuarios subir fotos directamente desde la cámara, recibir notificaciones push de ventas y usar geolocalización para ver productos cercanos, conectándose a la API de Spring Boot.

4. Resultados Esperados

Al finalizar el proyecto, se obtendrán los siguientes productos entregables:

- **API RESTful desplegada:** Un servidor Spring Boot funcional documentado (con Swagger/OpenAPI) capaz de recibir peticiones HTTP de los distintos clientes.
- **Portal Web Funcional:** Una URL accesible donde los usuarios pueden realizar el ciclo completo de compraventa.
- **Cliente de Escritorio (.jar/.exe):** Un software instalable en Windows/Linux para el perfil "Administrador" del sistema, optimizado para gestión masiva de datos.
- **Aplicación Móvil (.apk):** Un archivo instalable para dispositivos Android con funcionalidades específicas de movilidad (cámara y ubicación).
- **Base de Datos Centralizada:** Un esquema relacional único que mantiene la integridad de los datos entre las tres plataformas.

5. Palabras Clave

Spring Boot, API REST, Java, Android, JavaFX, E-commerce, Economía Circular, MySQL, Seguridad Informática.

6. Tipos de Proyecto

Este trabajo se encuadra dentro de las siguientes categorías:

- **Desarrollo de Software Multiplataforma:** Integración de entornos Web, Desktop y Mobile.
- **Aplicación de Gestión Empresarial (ERP/CRM):** En su vertiente de administración de usuarios y catálogo.
- **Sistemas Distribuidos:** Arquitectura cliente-servidor mediante servicios web.

Detalles Técnicos Sugeridos

Para que el proyecto sea coherente con "Aplicación informática para PC" y "Android" usando Spring Boot, esta es la arquitectura recomendada:

Capa	Tecnología Sugerida	Función
Backend (Servidor)	Spring Boot + Spring Data JPA + Hibernate	Cerebro del sistema. Expone una API REST (JSON) que usan el PC y Android.
Base de Datos	MySQL o MariaDB	Almacenamiento único para todos.
Web (Frontend)	Spring Thymeleaf (SSR) o React/Vue	Interfaz para el navegador.
PC (Escritorio)	JavaFX	Cliente pesado que hace peticiones HTTP al servidor Spring Boot. Ideal para administradores.
Móvil	Android Nativo (Java/Kotlin)	Cliente móvil que hace peticiones HTTP (usando Retrofit/Volley) al servidor.