



Universidad San Carlos De Guatemala
Centro Universitario Del Occidente
Manejo e Implementación de Archivos
Segundo Semestre
Sección: A
Inga. Bryan Monzon
Aux. Luis Cifuentes

Manual Tecnico

Commerce_GT

Nombre y carné:

Marco Pool Chiché Sapón 201832053

Quetzaltenango 28 de octubre 2025

Introducción

El presente documento técnico describe el proceso de desarrollo, la arquitectura, las herramientas utilizadas y el funcionamiento interno de la aplicación Commerce_GT, una plataforma web de comercio electrónico diseñada especialmente para usuarios residentes en Guatemala.

Commerce_GT ofrece una solución integral de e-commerce que incluye registro de usuarios, publicación de productos, control de inventario, gestión de pagos mediante tarjetas, calificaciones de productos, seguimiento de pedidos, administración logística y moderación de contenido.

La aplicación fue desarrollada utilizando una arquitectura cliente-servidor, con Spring Boot (Java 21) para el backend, Angular 17 para el frontend y PostgreSQL 17 como sistema gestor de base de datos. Además, incorpora autenticación segura mediante JWT (JSON Web Tokens), despliegue del cliente en Netlify y exposición temporal del servidor mediante Ngrok.

Este manual tiene como objetivo documentar los aspectos técnicos más relevantes del sistema, desde su estructura tecnológica hasta sus principales funcionalidades y configuraciones, con el fin de facilitar la comprensión, mantenimiento y futura expansión del proyecto.

Objetivos

Objetivo General

Desarrollar una aplicación web de comercio electrónico denominada Commerce_GT, que permita la interacción entre distintos tipos de usuarios (Común, Moderador, Logística y Administrador) para la compra, venta, gestión y distribución de productos en línea dentro del territorio guatemalteco.

Objetivos Específicos

1. Implementar la gestión de usuarios con diferentes roles y permisos, controlando el acceso mediante autenticación con JWT.
2. Desarrollar módulos funcionales para la venta, compra y calificación de productos, integrando un sistema de carrito de compras y pagos mediante tarjeta de crédito.
3. Diseñar un sistema de control administrativo que permita al administrador registrar empleados, generar reportes y gestionar las operaciones internas de la plataforma.
4. Implementar un módulo de moderación, que permita a los usuarios moderadores revisar y aprobar productos, así como aplicar sanciones a vendedores en caso de incumplimientos.
5. Desarrollar un módulo de logística, encargado del seguimiento y actualización del estado de los pedidos, garantizando la correcta entrega de los productos.
6. Integrar notificaciones automáticas por correo electrónico, para informar a los usuarios sobre el estado de sus pedidos o productos en tiempo real.
7. Diseñar y normalizar la base de datos en PostgreSQL bajo un modelo relacional de nivel 2, optimizando la integridad y consistencia de los datos.
8. Aplicar buenas prácticas de arquitectura de software, utilizando Spring Boot y Angular bajo un enfoque modular y desacoplado.
9. Desplegar la aplicación utilizando herramientas modernas como Netlify (frontend) y Ngrok (backend), garantizando su accesibilidad en entornos de prueba y demostración.

Tecnologías Utilizadas

Componente	Tecnología	Descripción
Lenguaje Backend	Java 21	Lenguaje principal para la API
Framework	Spring Boot 3.5.6	Gestión de dependencias, web, seguridad y persistencia
Seguridad	Spring Security + JWT	Autenticación basada en tokens
Base de Datos	PostgreSQL 17	Motor relacional
ORM	Hibernate / JPA	Mapeo objeto-relacional
Frontend	Angular 17	Framework SPA para el cliente web
Estilos	Bootstrap 5	Diseño responsivo
API Test	Postman	Pruebas de endpoints
Servidor temporal	Ngrok	Túnel HTTPS dinámico
Despliegue Frontend	Netlify	Hosting gratuito para SPA Angular

Dependencias Backend (Maven)

Archivo: pom.xml

Principales dependencias:

- spring-boot-starter-data-jpa
- spring-boot-starter-security
- spring-boot-starter-web
- postgresql
- lombok
- jjwt (api, impl, jackson) → autenticación JWT
- jakarta.validation y hibernate-validator → validaciones de entrada

Base de Datos

Gestor: PostgreSQL

Nombre: commerce

Principales Tablas:

- usuarios
- tarjetas
- categorias
- productos
- carritos, carrito_detalle
- pedidos, pedido_detalle
- sanciones
- ratings
- logistica

Relaciones clave:

- usuarios tiene relación 1:N con productos, tarjetas, pedidos, etc.
- productos pertenece a categorias.
- pedidos tiene detalles (pedido_detalle).
- carritos y productos están relacionados mediante carrito_detalle.
- usuarios sancionan a otros usuarios mediante sanciones.

Seguridad y Autenticación

- JWT (JSON Web Token) implementado para autenticación sin estado.
- Al iniciar sesión, se genera un *token* con información del usuario y tipo (COMUN, MODERADOR, LOGISTICA, ADMINISTRADOR).
- El token se envía en el header Authorization de cada petición.
- Se implementaron filtros para validar los tokens y restringir acceso a rutas protegidas.

Configuración del Entorno

Requisitos previos:

- Java 21
- Maven 3.9+
- Node.js 20+
- PostgreSQL 17
- Angular CLI 17+

Configuración del Backend

```
1 # Clonar el repositorio:
2
3 git clone https://github.com/marcochiche/commerce_gt.git
4 cd commerce_gt/backend
5
6
7 # Crear base de datos:
8
9 CREATE DATABASE commerce;
10
11
12 # Configurar archivo application.properties:
13
14 spring.datasource.url=jdbc:postgresql://localhost:5432/commerce
15 spring.datasource.username=postgres
16 spring.datasource.password=tu_contraseña
17 spring.jpa.hibernate.ddl-auto=update
18 spring.jpa.show-sql=true
19 jwt.secret=clave_secreta_commercegt
20
21
22 # Ejecutar el servidor:
23
24 mvn spring-boot:run
```

Configuración del Frontend

```
1 #Instalar dependencias:
2
3 npm install
4
5 #Ejecutar servidor de desarrollo:
6
7 ng serve
8
9
10 #Acceso:
11 http://localhost:4200
12
13 #Desplegar en Netlify subiendo la carpeta /dist/commerce-front.
```

Herramientas de Prueba y Despliegue

Herramienta	Uso
Postman	Pruebas de los endpoints REST
Ngrok	Exposición del backend local mediante HTTPS
Netlify	Hosting del frontend Angular
Terminal (bash)	Ejecución de comandos de Maven, Angular y Git

Conclusiones Técnicas

- La arquitectura Spring Boot + Angular + PostgreSQL garantiza modularidad, escalabilidad y seguridad.
- El uso de JWT permite una autenticación segura y sin sesiones persistentes.
- El despliegue mediante Netlify + Ngrok facilita la exposición temporal o permanente del sistema sin servidores dedicados.
- La estructura es fácilmente adaptable para un entorno de producción en AWS, Render, Railway u otros.

Roles y Funcionalidades

El sistema Commerce_GT está diseñado con una estructura de roles jerarquizada que permite mantener el control, la seguridad y la organización dentro de la plataforma. Cada rol posee funcionalidades específicas que se alinean con sus responsabilidades dentro del sistema.

Usuario Común

- Registro e inicio de sesión.
- Publicación de productos en venta (nombre, descripción, imagen, precio, stock, estado, categoría).
- Actualización y eliminación de sus productos.
- Visualización de productos aprobados por moderadores.
- Gestión de carrito de compras: agregar, eliminar, modificar cantidades, vaciar o pagar el carrito.
- Pago con tarjeta de crédito y almacenamiento seguro de tarjetas.
- Calificación y comentarios sobre productos adquiridos.
- Seguimiento de pedidos con estado (“en curso”, “entregado”).

Moderador

- Revisión, aprobación o rechazo de productos publicados por usuarios comunes.
- Gestión de sanciones a usuarios: registrar motivo, fecha y estado de cada sanción.
- Control de historial de sanciones para garantizar la calidad del comercio.

Usuario de Logística

- Visualización de todos los pedidos en curso.
- Actualización de fechas de entrega.
- Cambio del estado de los pedidos (“en curso”, “entregado”).

Administrador

- Registro y actualización de usuarios tipo Moderador, Logística y Administrador.
- Generación de reportes administrativos:
 - Top 10 productos más vendidos.
 - Top 5 clientes con mayores ganancias.
 - Top 5 vendedores más productivos.
 - Top 10 clientes con más pedidos.
 - Top 10 clientes con más productos a la venta.
 - Historial de sanciones y notificaciones.
- Envío automático de notificaciones por correo.
- Administración general del sistema y mantenimiento de usuarios.

Entradas y Salidas del Sistema

Entradas

Las entradas corresponden a los datos que el usuario proporciona o genera a través de las distintas interfaces del sistema:

- Datos de registro e inicio de sesión (correo, contraseña, nombre, etc.).
- Información de productos (nombre, descripción, precio, imagen, stock, categoría).
- Datos de pago (número de tarjeta, fecha de expiración, CVV, nombre del titular).
- Calificaciones y comentarios de productos.
- Información de pedidos (dirección, cantidad, fecha de entrega).
- Acciones administrativas (aprobaciones, sanciones, actualizaciones, reportes).

Salidas

Las salidas representan la información procesada que el sistema entrega al usuario:

- Confirmaciones de registro, compra o publicación.
- Listado de productos filtrados y disponibles.
- Estado actualizado de pedidos.
- Resultados de reportes generados.
- Correos electrónicos automáticos de notificación.
- Mensajes de éxito o error según la acción realizada.

Componentes del Sistema

El sistema Commerce_GT está compuesto por tres niveles principales, cada uno con sus propios componentes y responsabilidades:

1. Frontend (Cliente – Angular)

- Interfaz gráfica que permite la interacción del usuario.
- Formularios de registro, inicio de sesión, publicación y edición de productos.
- Módulos específicos para cada rol (usuario, moderador, logística, administrador).
- Gestión dinámica de vistas, rutas y autenticación mediante JWT.
- Desplegado públicamente en Netlify.

2. Backend (Servidor – Spring Boot)

- API REST encargada de la lógica de negocio.
- Controladores, servicios y repositorios organizados bajo arquitectura en capas.
- Manejo de autenticación y autorización con JWT.
- Gestión de productos, pedidos, pagos, sanciones y reportes.
- Exposición pública mediante Ngrok con protocolo HTTPS.

3. Base de Datos (PostgreSQL)

- Almacenamiento persistente y estructurado de la información.
- Relaciones entre usuarios, productos, pedidos, sanciones y notificaciones.
- Aplicación de una arquitectura de nivel 2, garantizando integridad y consistencia.

Conclusión

El desarrollo de la aplicación **Commerce_GT** representa una solución tecnológica completa para la gestión de comercio electrónico en Guatemala, integrando de manera efectiva los distintos procesos que intervienen en una plataforma de este tipo: registro de usuarios, publicación y venta de productos, pagos en línea, moderación, logística y administración.

Gracias a la implementación de una arquitectura basada en **Spring Boot** para el backend, **Angular** para el frontend y **PostgreSQL** como sistema gestor de base de datos, se logró construir un sistema escalable, modular y seguro. La integración de **JWT (JSON Web Token)** como mecanismo de autenticación garantiza la protección de la información y el control de accesos según los distintos roles definidos en el sistema.

Asimismo, el uso de herramientas como **Ngrok** para la exposición temporal del servidor y **Netlify** para el despliegue del cliente web permitió realizar pruebas en tiempo real, facilitando la validación de los servicios y la accesibilidad del sistema desde cualquier dispositivo.

Durante el desarrollo, se aplicaron buenas prácticas de programación, principios de arquitectura limpia y una estructura de base de datos normalizada, asegurando un proyecto robusto y mantenible a largo plazo.

En conclusión, **Commerce_GT** cumple con los objetivos propuestos al ofrecer una plataforma funcional, moderna y adaptable, capaz de mejorar la experiencia de compra y venta en línea dentro del contexto guatemalteco, consolidándose como una base sólida para futuras mejoras y ampliaciones del sistema.