



Trabajo de Grado

DESARROLLO MÓDULOS DE GESTIÓN DE PRODUCTOS, ESTADÍSTICAS,
EXPOSICIÓN DE API PARA INTEGRACIONES Y SEGURIDAD PARA
GARANTIZAR UNA MIGRACIÓN EXITOSA A LA NUEVA VERSIÓN DE LA
PLATAFORMA CONTENTO
(Modalidad pasantía)

JOSE DAVID SUÁREZ CARDONA
202067548
jose.david.suarez@correounivalle.edu.co

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas

Universidad del Valle
Sede Tuluá

29 de noviembre de 2025

DESARROLLO MÓDULOS DE GESTIÓN DE PRODUCTOS, ESTADÍSTICAS,
EXPOSICIÓN DE API PARA INTEGRACIONES Y SEGURIDAD PARA
GARANTIZAR UNA MIGRACIÓN EXITOSA A LA NUEVA VERSIÓN DE LA
PLATAFORMA CONTENTO
(Modalidad pasantía)

Director:

José Luis Unás Gómez.

Ingeniero de Sistemas, Especialista en Seguridad Informática
Coordinador del programa de Ingeniería de Sistemas
jose.unas@correounivalle.edu.co

Supervisor:

Christian Alexis Castro Correa

Ingeniero de Sistemas

Director Tecnológico en Contenido LATAM
tecnologia@contentolatam.com

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas

Universidad del Valle
Sede Tuluá

29 de noviembre de 2025

Índice

1. Introducción	10
1.1. Descripción de la empresa	11
1.1.1. Descripción general	11
1.1.2. Misión	11
1.1.3. Visión	11
1.1.4. Valores organizacionales	11
1.1.5. Razón social	11
1.1.6. Tipo de empresa	11
1.1.7. Cargos de la empresa	12
1.1.8. Número de empleados	12
1.1.9. Organigrama de la empresa	12
1.1.10. Mapa de procesos de la empresa	13
1.1.11. Descripción de funciones a desarrollar el estudiante en la empresa.	16
1.1.12. Director de proyecto de grado.	16
1.1.13. Supervisor y jefe.	16
1.2. Planteamiento del problema	17
1.2.1. Descripción del problema	17
1.2.2. Árbol de problemas	18
1.2.3. Formulación del problema	19
1.3. Alcance del proyecto	20
1.3.1. Objetivo general	20
1.3.2. Objetivos específicos	20
1.3.3. Objetivos esperados	21
1.4. Justificación	22
2. Marco referencial	23
2.1. Marco conceptual	23
2.1.1. JavaScript (JS)	23
2.1.2. Scrum	23
2.1.3. TypeScript (TS)	23
2.1.4. React	23
2.1.5. NestJS	24
2.1.6. Node.js	24
2.1.7. Git	24
2.1.8. Github	24
2.1.9. Jira	24
2.1.10. Microsoft Teams	24
2.1.11. Material-UI (MUI)	24
2.2. Antecedentes	25
3. Metodología	26
3.1. Metodología de Gestión de Actividades	26
3.2. Metodología de Desarrollo de Software	26
3.3. Metodología de Investigación	26
4. Ingeniería de desarrollo	28

4.1.	Requerimientos del Sistema	28
4.1.1.	Objetivo	28
4.1.2.	Módulo de Gestión de Productos (GP)	28
4.1.3.	Módulo de Estadísticas (ES)	31
4.1.4.	Exposicion de Api para integraciones (EAI)	33
4.1.5.	Requerimientos No Funcionales (RNF)	34
4.2.	Diseño de la Arquitectura de Software	36
4.2.1.	Objetivo	36
4.2.2.	Arquitectura General e Integración	36
4.2.3.	Estructura Modular y Organización del Código	36
4.2.4.	Diseño del Modelo de Datos	37
4.2.5.	Patrones de Diseño Implementados	38
4.3.	Gestión de Productos	39
4.3.1.	Objetivo	39
4.3.2.	Funciones Principales	39
4.3.3.	Creación de Productos (POST /app/:appId/product)	40
4.3.4.	Obtener Productos de una Tienda (GET /app/:appId/products)	41
4.3.5.	Actualizar Productos (PATCH /app/:appId/product/:id)	42
4.3.6.	Eliminar Productos (DELETE /app/:appId/product/:id)	42
4.3.7.	Importar Productos desde Archivo Excel (POST /app/:appId/product/import)	43
4.3.8.	Exportar productos en archivo de excel (POST /app/:appId/product/export)	45
4.3.9.	Obtener Detalles de un Producto (GET /app/:appId/products/:productId)	45
4.3.10.	Reflexión	46
4.4.	Seguridad de la Aplicación	47
4.4.1.	Objetivo	47
4.4.2.	Funciones Principales	47
4.4.3.	Endpoint para Realizar Login (POST /auth/login)	48
4.4.4.	Guard para Proteger Rutas del Servidor (Access Token)	49
4.4.5.	Guard para validar permisos de un usuario	50
4.4.6.	Configuración para limitar peticiones	51
4.4.7.	Validaciones de DTOs con Class-Validator	52
4.4.8.	Reflexión	52
4.5.	Módulo de Estadísticas	53
4.5.1.	Objetivo	53
4.5.2.	Funcionalidades Principales	53
4.5.3.	Obtener promedio de ventas (GET /statistics/average/:appId)	54
4.5.4.	Obtener segmentación geográfica de clientes (GET /statistics/geo/:appId)	54
4.5.5.	Obtener estadísticas de visitas (GET /statistics/views/:appId)	55
4.5.6.	Obtener órdenes recientes (GET /statistics/recent/orders/:appId)	55
4.5.7.	Obtener métodos de pago utilizados (GET /statistics/method-payments/:appId)	56
4.5.8.	Obtener categorías más vendidas (GET /statistics/top-categories/:appId)	56
4.5.9.	Obtener valor promedio de tickets (GET /statistics/tickets/:appId)	57
4.5.10.	Obtener productos más vendidos (GET /statistics/top-products/:appId)	57
4.5.11.	Reflexion	58
4.6.	Exposicion de api para integraciones (Bepay webhook)	59
4.6.1.	Objetivo	59
4.6.2.	Procesamiento de Webhook de Pagos (pasarela BePay)	59
4.6.3.	Funciones clave del webhook	59

4.6.4. Reflexión	60
4.7. Pruebas del Sistema	61
4.7.1. Objetivo de las Pruebas	61
4.7.2. Metodología de Pruebas	61
4.7.3. Cobertura y Resultados	61
4.7.4. Evidencia de Pruebas	61
4.7.5. Reflexion	63
5. Conclusiones	64
5.1. Sobre la Arquitectura y Escalabilidad	64
5.2. Sobre la Gestión de Productos	64
5.3. Sobre la Seguridad y Control de Acceso	64
5.4. Sobre la Capacidad Analítica	64
5.5. Conclusión General	64
Referencias Bibliográficas	65

Índice de figuras

1.	Organigrama organizacional.	12
2.	Mapa de proceso comercial y vinculación comercial.	13
3.	Bienvenida y programación de activación de local.	14
4.	Capacitación local virtual.	15
5.	Árbol de problemas del presente proyecto.	18
6.	Diagrama de marco de trabajo Scrum	23
7.	Diagrama de arquitectura general del sistema e integración de servicios. .	36
8.	Estructura modular del proyecto en el entorno de desarrollo.	37
9.	Diagrama Entidad-Relación (DER) de la plataforma.	38
10.	Creación de producto con imágenes y variantes.	40
11.	Formulario extendido de creación de producto con múltiples atributos. . .	40
12.	Listado de productos con paginación.	41
13.	Búsqueda de productos por slug.	41
14.	Actualización de datos de un producto existente.	42
15.	Eliminación lógica de un producto (soft delete).	43
16.	Intento de importación con formato inválido.	43
17.	Plantilla de formato Excel para carga masiva.	44
18.	Importación exitosa de productos desde archivo válido.	44
19.	Endpoint para obtener archivo de excel con informacion de productos . .	45
20.	Vista de excel ya exportado	45
21.	Visualización completa de los detalles de un producto.	46
22.	Detalle extendido del producto con variantes y precios.	46
23.	Proceso de inicio de sesión mediante autenticación JWT.	48
24.	Guard que valida el token de acceso antes de procesar la solicitud.	49
25.	Guard que valida permisos requeridos de un endpoint.	50
26.	Configuración del módulo con limitación de peticiones por ventana de tiempo.	51
27.	DTO con validaciones de campos obligatorios para creación de productos.	52
28.	Endpoint para obtener el promedio de ventas de una tienda.	54
29.	Endpoint para obtener la distribución geográfica de los clientes de una tienda.	55
30.	Endpoint para obtener estadísticas de visitas de una tienda.	55
31.	Endpoint para consultar las órdenes más recientes de una tienda.	56
32.	Endpoint para obtener estadísticas sobre el uso de métodos de pago. . . .	56
33.	Endpoint para obtener el top de categorías con mayor número de ventas.	57
34.	Endpoint para calcular el valor promedio por transacción (ticket promedio).	57
35.	Endpoint para consultar los productos más vendidos de una tienda. . . .	58
36.	Se procesa transacción y se pasa a estado pendiente	60
38.	Resultado de pruebas generadas con jest	62
37.	Reporte de resultados de pruebas generadas con Jest	62
39.	Resultado de pruebas generadas con jest	63

Índice de tablas

1.	Tabla de objetivos esperados	21
2.	Antecedentes de trabajos relacionados	25
3.	RF-GP-01: Crear producto	28
4.	RF-GP-02: Obtener productos	29
5.	RF-GP-03: Actualizar producto	29
6.	RF-GP-04: Eliminar producto	30
7.	RF-GP-05: Obtener detalles de producto	30
8.	RF-GP-06: Importar productos desde Excel	31
9.	RF-ES-07: Obtener distribución geográfica	31
10.	RF-ES-08: Promedio de ventas	32
11.	RF-ES-09: Obtener visitas y variación	32
12.	RF-ES-10: Métodos de pago	32
13.	RF-ES-11: Obtener productos top	33
14.	RF-EAI-01: Procesamiento Webhook Pagos	33
15.	RNF-01: Rendimiento	34
16.	RNF-02: Seguridad	34
17.	RNF-03: Mantenibilidad	34
18.	RNF-04: Interoperabilidad	35

Resumen

El comercio electrónico en Colombia ha tenido un crecimiento significativo, lo que presenta una oportunidad y un desafío para plataformas como Contenido. Este proyecto de grado tiene como objetivo mejorar la plataforma a través de la implementación de módulos clave como la gestión de productos, estadísticas para la optimización de decisiones, seguridad robusta y la exposición de APIs para facilitar integraciones. La necesidad de esta modernización surge debido a las limitaciones técnicas de la versión actual, que afectan la competitividad de Contenido en un mercado digital en constante cambio.

La metodología empleada para el desarrollo es el marco de trabajo ágil Scrum, que asegura la entrega continua de valor mediante sprints de dos semanas, con prácticas como Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective. Además, se utilizará una arquitectura modular y escalable para el backend con tecnologías como NestJS y TypeScript.

Se espera que la nueva versión de la plataforma Contenido esté optimizada en términos de funcionalidad, seguridad y escalabilidad, ofreciendo una plataforma más eficiente y adaptable a las necesidades del mercado. La implementación de los nuevos módulos mejorará la experiencia del usuario, facilitará la integración con otros sistemas y permitirá una gestión más efectiva de la información. Estas mejoras contribuirán al fortalecimiento de la plataforma, asegurando su competitividad y sostenibilidad en un entorno digital en constante evolución.

PALABRAS CLAVE: E-commerce, Plataformas Tecnológicas, Experiencia de Usuario, Desarrollo Backend.

Abstract

E-commerce in Colombia has had a significant growth, which represents an opportunity and a challenge for platforms such as Contenido. This degree project seeks to improve the platform through the implementation of key modules such as product management, statistics for decision optimization, robust security and the exposure of APIs to facilitate integrations. The need for this modernization arises due to the technical limitations of the current version, which affect Contenido's competitiveness in an ever-changing digital market.

The methodology used for the development is the agile Scrum framework, which ensures continuous delivery of value through two-week sprints, with practices such as Sprint Planning, Daily Scrums, Sprint Review and Sprint Retrospective. In addition, a modular and scalable architecture will be used for the backend with technologies such as NestJS and TypeScript.

The new version of the Contenido platform is expected to be optimized in terms of functionality, security and scalability, offering a platform that is more efficient and adaptable to market needs. The implementation of the new modules will improve the user experience, facilitate integration with other systems and enable more efficient information management. These enhancements will help strengthen the platform, ensuring its competitiveness and sustainability in a constantly evolving digital environment.

KEYWORDS: E-commerce, Technology Platforms, User Experience, backend Development.

1. Introducción

Contento es una empresa tecnológica creada con el propósito de brindar soluciones integrales que optimicen los procesos clave dentro de las empresas. Su enfoque se centra en digitalizar los negocios tradicionales, mejorando tanto su gestión operativa como su estrategia empresarial. A través de herramientas tecnológicas accesibles, la empresa Contento busca facilitar la transición al mundo digital, permitiendo a las empresas ser más eficientes, competitivas y adaptarse mejor a las demandas del mercado actual. Este trabajo de grado se centrará en contribuir a la migración tecnológica de la plataforma, un proceso esencial para modernizar su infraestructura y responder a las crecientes exigencias del mercado actual.

Durante el desarrollo de las pasantías, se abordará el diseño e implementación de módulos clave como la gestión de productos, orientada a facilitar la administración eficiente de catálogos e inventarios, estadísticas, que proporcionarán información estratégica basada en datos para la toma de decisiones, seguridad, garantizando la protección de la información sensible y el acceso confiable, exposición de APIs, que permitirá integrar la plataforma con otros sistemas, ampliando su capacidad de interoperabilidad.

La migración tecnológica no solo busca mejorar el rendimiento y la funcionalidad de la plataforma Contento, sino también posicionarla como una herramienta más versátil y robusta, capaz de atender de manera integral las necesidades de sus clientes. Para el estudiante, esta experiencia representa una oportunidad única de aplicar sus conocimientos académicos en un entorno laboral real, adquirir nuevas habilidades relacionadas con la tecnología y la gestión de proyectos, y contribuir activamente al fortalecimiento competitivo de la empresa en un mercado en constante evolución.

1.1. Descripción de la empresa

1.1.1. Descripción general

Contento es una empresa enfocada en desarrollar herramientas digitales diseñadas para apoyar a las empresas comerciales que buscan reinventarse, crecer y avanzar a través de la interconexión, la automatización y el acceso a datos en tiempo real.

Su objetivo es contribuir al crecimiento de las empresas colombianas mediante soluciones tecnológicas en comercialización, estableciendo alianzas estratégicas con entidades que comparten el compromiso de mejorar el bienestar de los comerciantes en Colombia.

1.1.2. Misión

Transformar locales físicos en auténticas experiencias de ventas virtuales.

1.1.3. Visión

Ser referentes en llevar herramientas listas conectadas a acciones, estrategias y alianzas innovadoras que impacten de manera activa y relevante en el desarrollo tecnológico, económico y cultural de Colombia.

1.1.4. Valores organizacionales

- **Ágiles:** Nos motivan los retos y nos adaptamos a los cambios para desarrollar soluciones y estrategias oportunas y ajustadas a la necesidad.
- **Empáticos:** Escuchamos, entendemos y nuestro corazón para brindar satisfacción a necesidades de quienes forman parte de nuestro ecosistema.
- **Transformadores:** Soñamos en grande, por eso trabajamos con pasión abriendo puertas que generan cambios impactando en lo tecnológico, económico y cultural de Colombia.

1.1.5. Razón social

CONTENTO LATAM SAS

1.1.6. Tipo de empresa

Jurídica - Microempresas SAS (sociedad acciones simplificadas)

1.1.7. Cargos de la empresa

Doce (12) cargos 1

1.1.8. Número de empleados

Dieciocho (18) empleados 1

1.1.9. Organigrama de la empresa

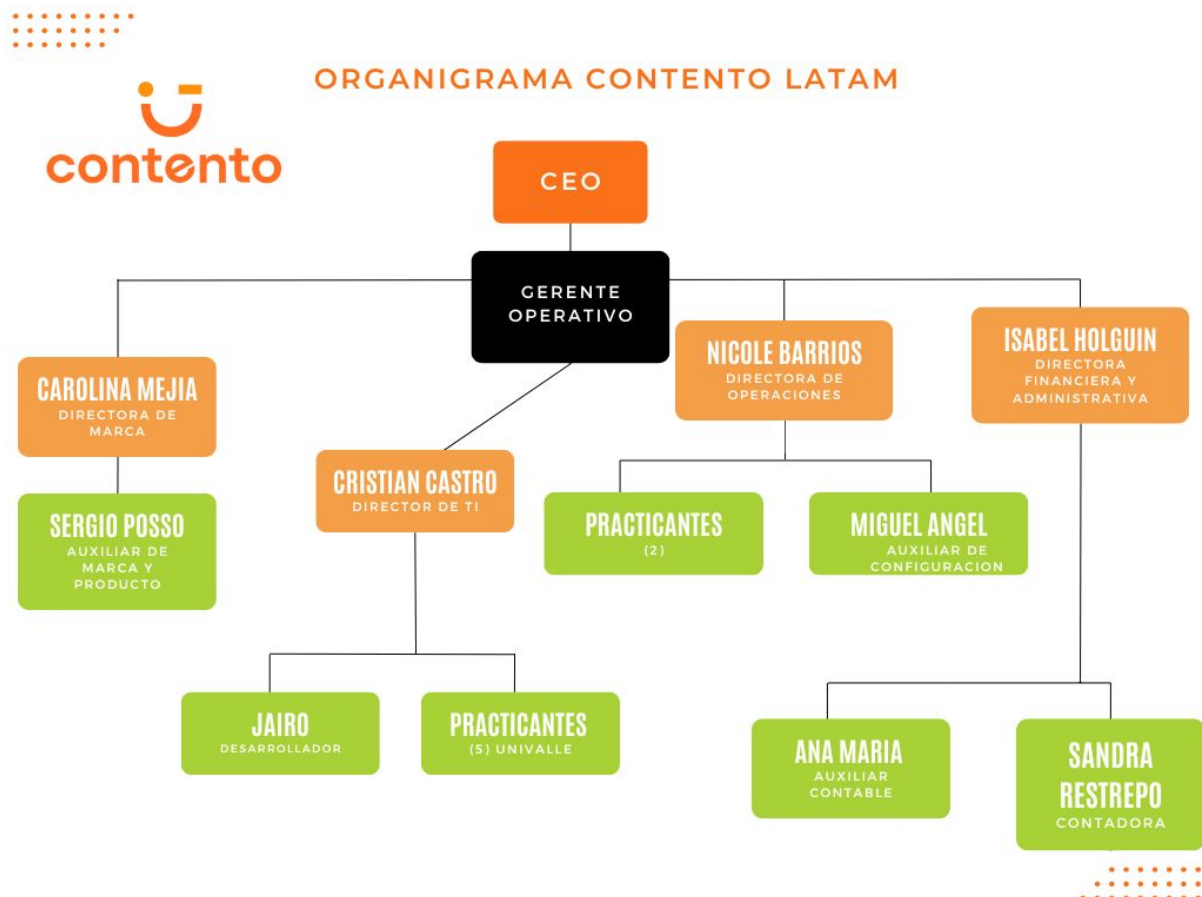


Figura 1: Organigrama organizacional.

Fuente: Directora general de finanzas (Isabel Cristina Holguín Prado).

1.1.10. Mapa de procesos de la empresa

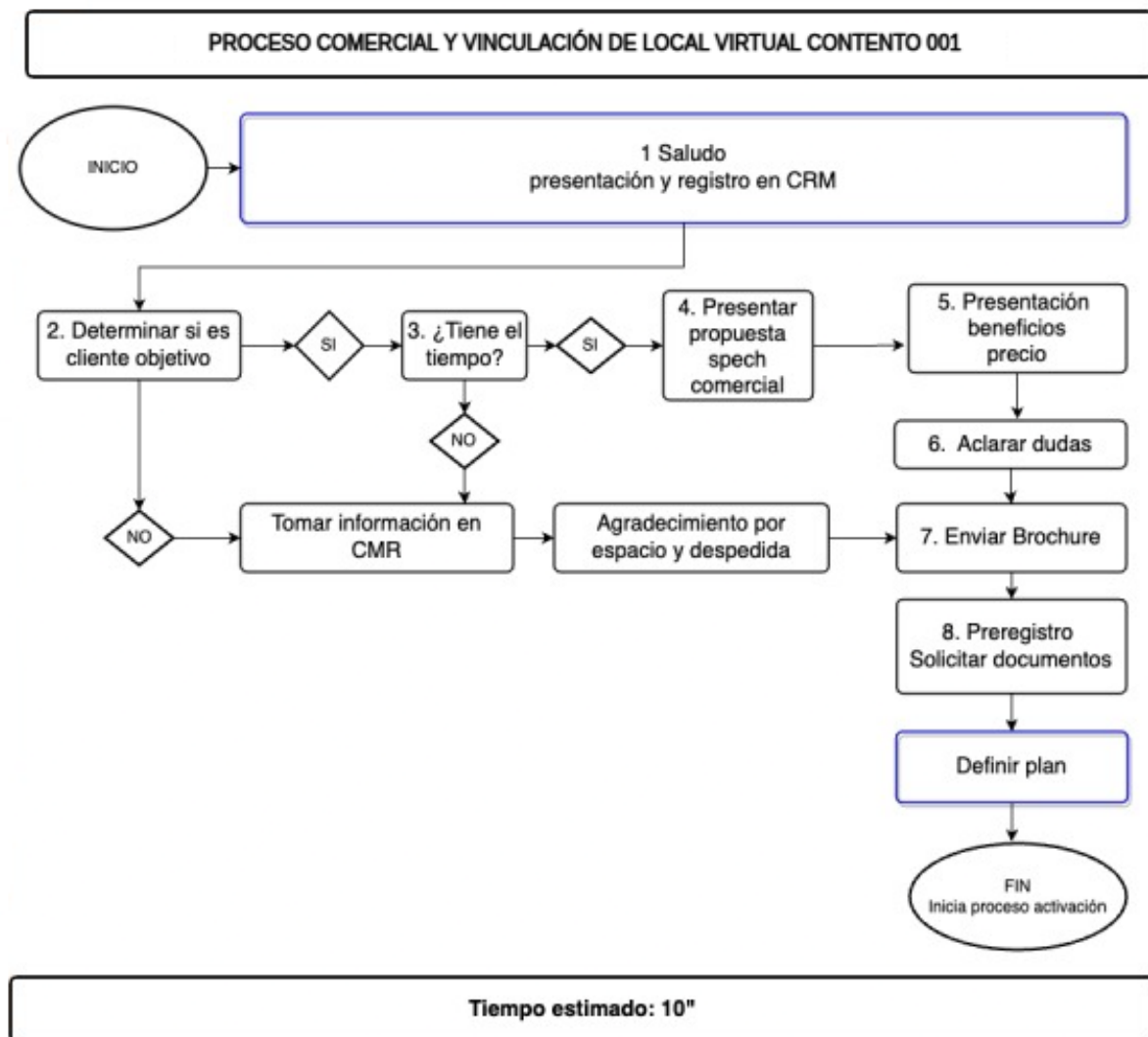


Figura 2: Mapa de proceso comercial y vinculación comercial.
Fuente: Directora general de finanzas (Isabel Cristina Holguín Prado).

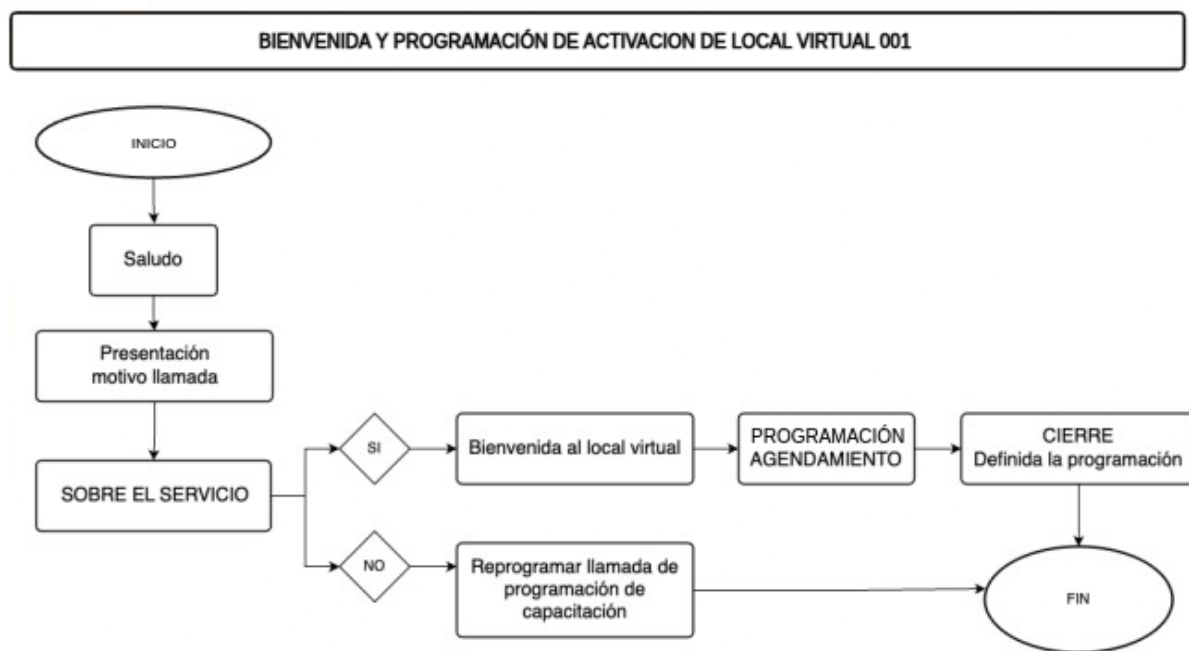


Figura 3: Bienvenida y programación de activación de local.
 Fuente: Directora general de finanzas (Isabel Cristina Holguín Prado).

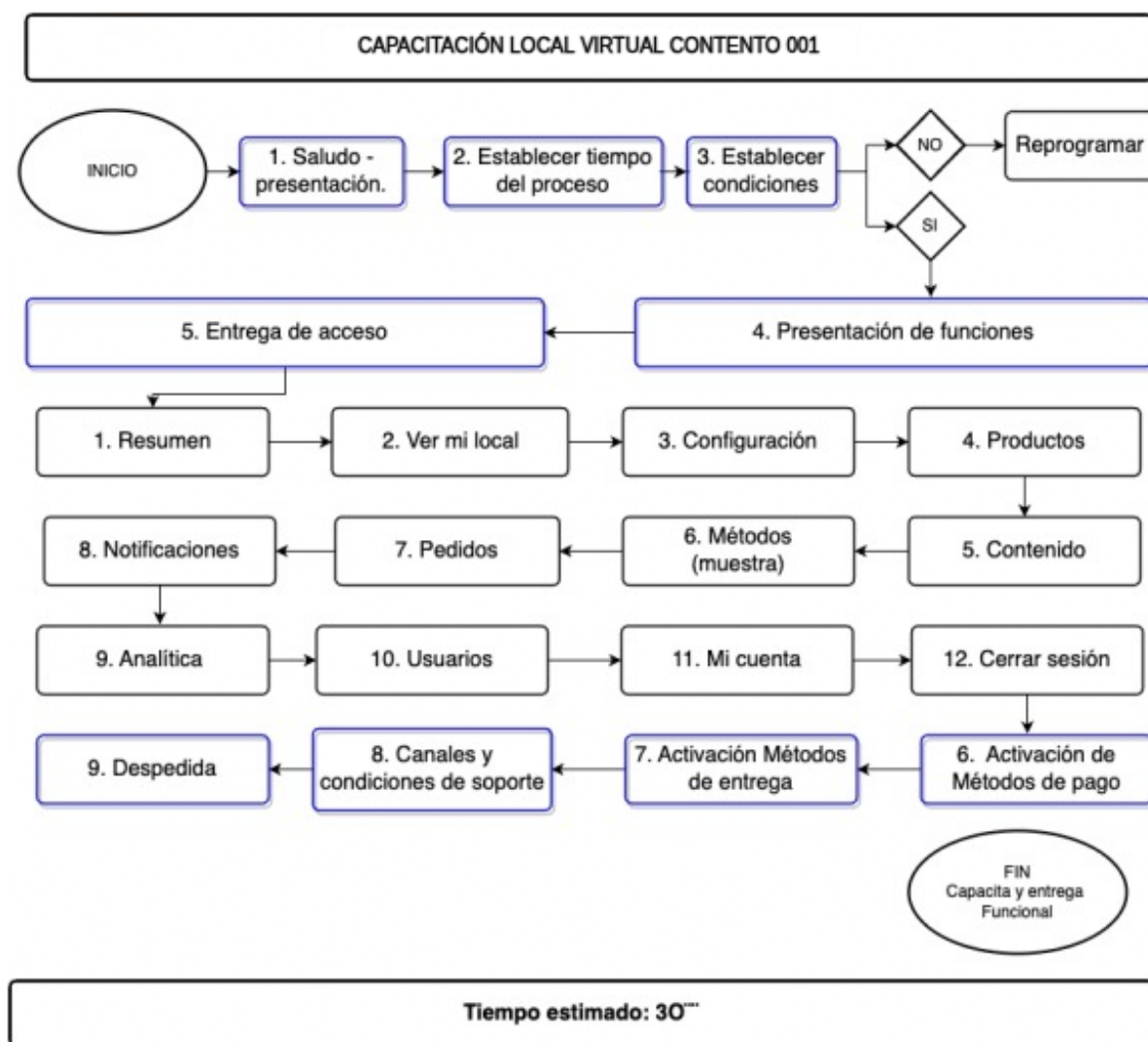


Figura 4: Capacitación local virtual.
Fuente: Directora general de finanzas (Isabel Cristina Holguín Prado).

1.1.11. Descripción de funciones a desarrollar el estudiante en la empresa.

1. Desarrollar módulos de la plataforma Contenido, específicamente en backend, a través de actividades de planificación, análisis, diseño y programación. Los módulos a desarrollar incluyen la gestión de productos, estadísticas, exposición de API para integraciones y la implementación de un sistema robusto de seguridad.
2. Realizar tareas operativas relacionadas con el desarrollo y mantenimiento de los módulos backend de Contenido, siguiendo las mejores prácticas y manteniéndose actualizado con las últimas tecnologías para garantizar la calidad del código y la escalabilidad de la plataforma.
3. Colaborar estrechamente con el equipo de desarrollo para asegurar una migración exitosa a la nueva versión de la plataforma, con un enfoque en la mejora continua y la implementación de nuevas funcionalidades que satisfagan las necesidades de los usuarios y clientes de Contenido.

1.1.12. Director de proyecto de grado.

Nombre: José Luis Unás Gómez. (Ingeniero de Sistemas, Especialista en Seguridad Informática)

Cargo: Coordinador del programa de Ingeniería de Sistemas

Correo: jose.unas@correounivalle.edu.co

Función: Asesorar al estudiante, orienta la investigación, brinda retroalimentación y asegura que el trabajo cumpla con los estándares académicos.

1.1.13. Supervisor y jefe.

Nombre: Christian Alexis Castro Correa. (Ingeniero de Sistemas)

Cargo: Director tecnológico

Correo: tecnologia@contentolatam.com

Función: Brinda orientación técnica y académica, asegura el cumplimiento de los requisitos del proyecto de grado, y revisa el avance para garantizar la calidad y correcta implementación de los módulos.

1.2. Planteamiento del problema

1.2.1. Descripción del problema

El comercio electrónico en Colombia ha experimentado un notable crecimiento en los últimos años, consolidándose como un pilar clave de la economía nacional. En 2022, se registraron ventas en línea por un total de 55,17 billones de pesos, representando un incremento del 38,4 % frente a 2021 y un impresionante 94,1 % en comparación con 2020. **tendencias'ecommerce'2022** Este avance ha sido impulsado por estrategias como los días sin IVA y el fortalecimiento de plataformas digitales, lo que ha promovido categorías como moda 52 % y belleza y cuidado personal 24 % como las más transadas en Social Commerce.

En este contexto, La empresa Contenido enfrenta la necesidad de evolucionar su plataforma para responder a las demandas actuales del mercado digital. Aunque su infraestructura tecnológica ha permitido consolidarse como una herramienta útil para las empresas, las limitaciones técnicas de la versión actual restringen su capacidad para ofrecer funcionalidades avanzadas, como la gestión eficiente de productos, la integración con otros sistemas mediante APIs, la generación de estadísticas en tiempo real y la implementación de medidas robustas de seguridad y autenticación.

Estos retos no solo afectan la competitividad de Contenido, sino también la de sus clientes, quienes dependen de una solución tecnológica confiable para gestionar sus operaciones. Por lo tanto, resulta imperativo modernizar la plataforma, garantizar su interoperabilidad y optimizar los módulos existentes para atender las crecientes demandas de un mercado digital en constante evolución.

1.2.2. Árbol de problemas

RAÍZ DEL PROBLEMA: La plataforma Contenido enfrenta dificultades técnicas y operativas significativas, que limitan su capacidad para competir de manera efectiva en el mercado global del comercio electrónico.

Estas limitaciones son especialmente relevantes dado al crecimiento del comercio digital en Colombia y el mundo.

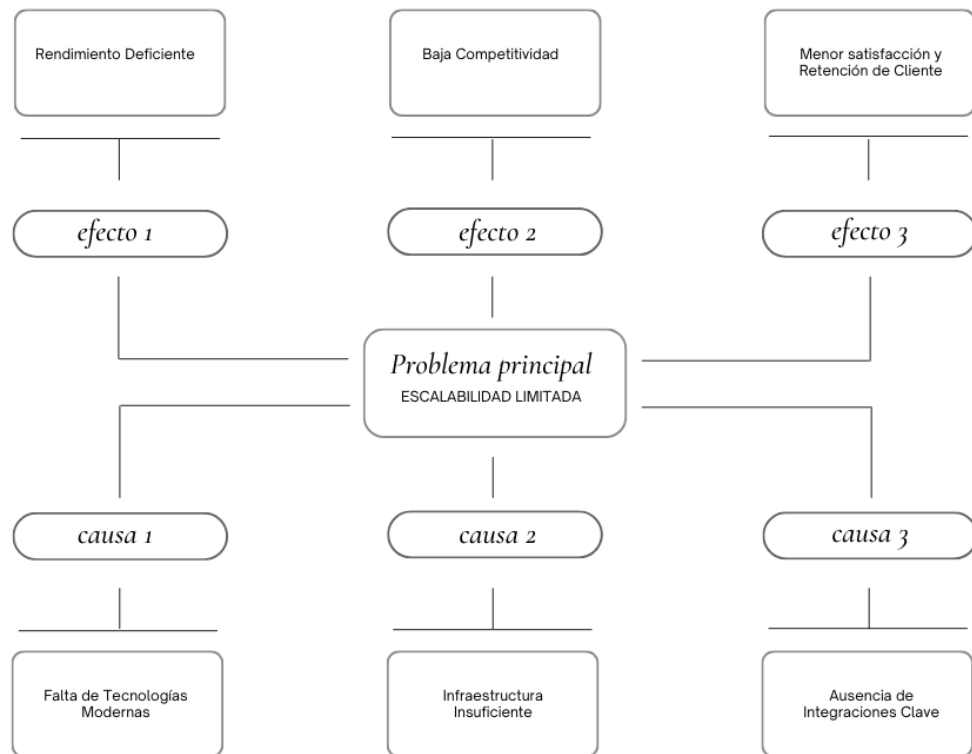


Figura 5: Árbol de problemas del presente proyecto.
Fuente: Elaboración propia.

1.2.3. Formulación del problema

La nueva versión de la plataforma Contenido requiere la implementación de módulos clave como gestión de productos, estadísticas, exposición de API y seguridad. Sin embargo, la falta de una integración eficiente de estos módulos podría afectar la experiencia del usuario y la eficiencia operativa del sistema. Por lo tanto, es necesario desarrollar una solución que permita la implementación e integración adecuada de estos componentes, garantizando su correcto funcionamiento y alineación con las necesidades del negocio.

1.3. Alcance del proyecto

Este proyecto de grado busca mejorar la nueva versión de la plataforma Contenido mediante el desarrollo de módulos en el backend para gestión de productos, estadísticas, seguridad y exposición de APIs. Se analizarán los requerimientos funcionales y no funcionales para garantizar calidad y alineación con las necesidades del negocio.

El desarrollo seguirá un enfoque ágil con Scrum, asegurando entregas iterativas y mejoras continuas. Se diseñará una arquitectura backend modular y escalable utilizando NestJS y TypeScript, optimizando el rendimiento y la interoperabilidad. Además, se implementarán pruebas unitarias e integrales para validar el funcionamiento y la seguridad del sistema.

Al finalizar, la plataforma contará con una infraestructura moderna, eficiente y segura, permitiendo una mejor gestión operativa, mayor protección de datos y facilidad de integración con otros sistemas, fortaleciendo su competitividad en el mercado digital.

1.3.1. Objetivo general

Desarrollar e integrar los módulos de gestión de productos, estadísticas, exposición de API y seguridad en la nueva versión de la plataforma Contenido, con el propósito de optimizar la experiencia del usuario y asegurar la eficiencia operativa mediante soluciones tecnológicas adecuadas.

1.3.2. Objetivos específicos

1. Analizar los requerimientos funcionales y no funcionales de los módulos a desarrollar, con el fin de garantizar que cumplan con las expectativas del negocio y los estándares de calidad de la plataforma Contenido.
2. Diseñar la arquitectura backend de los módulos a desarrollar, asegurando una estructura escalable, mantenible y alineada con las necesidades de la plataforma Contenido.
3. Desarrollar e integrar un módulo de gestión de productos y estadísticas, junto con la exposición de API, permitiendo una administración eficiente del catálogo, el análisis detallado del rendimiento de la plataforma y la interoperabilidad con otros sistemas externos, mejorando la experiencia del usuario y la toma de decisiones estratégicas.
4. Fortalecer la seguridad de la plataforma mediante la implementación de un sistema de autenticación y validación robusto, garantizando la protección de la información de los usuarios y la empresa.
5. Realizar pruebas unitarias e integrales para asegurar el correcto funcionamiento de los módulos desarrollados, garantizando su fiabilidad y cumplimiento de los requisitos establecidos.

1.3.3. Objetivos esperados

Objetivo	Objetivo Esperado
Analizar los requerimientos funcionales y no funcionales para garantizar su alineación con las expectativas del negocio y los estándares de calidad.	Documento con la especificación detallada de los requerimientos, validación con las partes interesadas y alineación con los estándares de calidad de la plataforma.
Diseñar la arquitectura backend asegurando una estructura escalable, mantenible y alineada con las necesidades de la plataforma.	Definición de una arquitectura modular con documentación técnica, diagramas de estructura y plan de integración con los sistemas existentes.
Desarrollar e integrar un módulo de gestión de productos, estadísticas y exposición de API para mejorar la administración del catálogo, el análisis de datos y la interoperabilidad con otros sistemas.	Implementación de los módulos con sus respectivas funcionalidades, optimización del análisis de datos y desarrollo de APIs para facilitar integraciones externas.
Fortalecer la seguridad mediante un sistema de autenticación y validación robusto que proteja la información de usuarios y la empresa.	Implementación de mecanismos de autenticación segura, mitigación de vulnerabilidades y cumplimiento de estándares de seguridad.
Realizar pruebas unitarias e integrales para garantizar el correcto funcionamiento y cumplimiento de los requisitos de los módulos desarrollados.	Desarrollo y ejecución de pruebas unitarias e integrales, documentación de resultados y corrección de errores para asegurar la fiabilidad del sistema.

Tabla 1: Tabla de objetivos esperados

1.4. Justificación

El presente trabajo de grado tiene como objetivo aplicar los conocimientos adquiridos durante el proceso formativo en el área de desarrollo de software, enfocándose en la mejora y expansión de la nueva versión de Contenido. Este proyecto, representa la transición de un entorno académico a un entorno profesional, permitiendo al estudiante adquirir experiencia práctica en tecnologías avanzadas aplicadas al desarrollo de plataformas tecnológicas.

El estudiante se encargará de añadir nuevas funcionalidades y módulos a Contenido V2, lo cual contribuirá a mejorar la competitividad y la experiencia del usuario. Además, se aplicará un marco de trabajo ágil como Scrum, lo que permitirá gestionar de manera eficiente el desarrollo de las nuevas características y asegurar una entrega continua de valor.

Este trabajo de grado también brindará la oportunidad de ampliar el perfil profesional del estudiante, mediante la colaboración directa con el equipo de desarrollo y el uso de tecnologías modernas. Con la integración de nuevas funcionalidades, se mejorará la plataforma, fortaleciendo la posición de la empresa en el mercado y asegurando un producto más completo y de alta calidad para sus usuarios.

2. Marco referencial

2.1. Marco conceptual

2.1.1. JavaScript (JS)

JavaScript es un lenguaje de programación ampliamente utilizado para crear aplicaciones web interactivas y dinámicas. Permite la manipulación de elementos en el navegador, gestión de eventos, y comunicación en tiempo real, siendo esencial para aplicaciones frontend modernas.`mozilla.js`

2.1.2. Scrum

Scrum es un marco de gestión de proyectos ágil que ayuda a los equipos a estructurar y gestionar su trabajo a través de un conjunto de valores, principios y prácticas. Al igual que un equipo de rugby (de donde recibe su nombre) que se entrena para el gran partido, el scrum anima a los equipos a aprender a través de las experiencias, a organizarse mientras trabajan en un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente. [1]

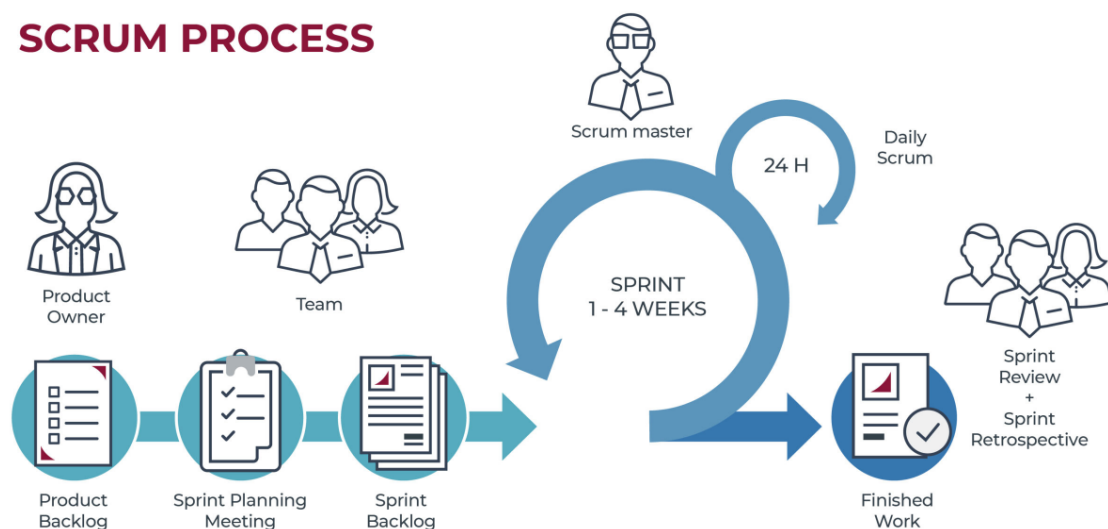


Figura 6: Diagrama de marco de trabajo Scrum

Fuente: <https://ausum.cloud/scrums-metodologia-agil-mas-popular-en-empresas/>

2.1.3. TypeScript (TS)

TypeScript es una versión de JavaScript con tipado estático, lo que permite detectar errores durante el desarrollo antes de que el código se ejecute. Facilita la escritura de código más seguro, legible y escalable, especialmente en proyectos grandes o colaborativos.[2]

2.1.4. React

React es una biblioteca de JavaScript para construir interfaces de usuario basadas en componentes reutilizables. Es eficiente para desarrollar aplicaciones de una sola página, donde

el rendimiento y la experiencia del usuario son cruciales. React permite actualizaciones rápidas de la interfaz sin recargar la página completa.[3]

2.1.5. NestJS

NestJS es un framework de desarrollo backend que utiliza TypeScript. Está basado en Node.js y es adecuado para crear aplicaciones escalables y de alto rendimiento. Su arquitectura modular permite organizar el código de manera estructurada y facilita el mantenimiento y la extensión de aplicaciones complejas.[4]

2.1.6. Node.js

Node.js es un entorno de ejecución para JavaScript del lado del servidor. Permite desarrollar aplicaciones rápidas y escalables, especialmente aquellas que manejan muchas operaciones de entrada y salida, como aplicaciones en tiempo real o servicios web.**NodeJS**

2.1.7. Git

Git es un sistema de control de versiones que permite gestionar el código fuente y realizar un seguimiento de los cambios realizados en el proyecto. **Git**

2.1.8. Github

GitHub es una plataforma que permite a los desarrolladores almacenar, gestionar y colaborar en proyectos de código. Facilita el control de versiones, las revisiones de código y el trabajo en equipo de manera eficiente. **GitHub**

2.1.9. Jira

Jira es la mejor herramienta de gestión de proyectos ágiles que utilizan los equipos para planificar, supervisar, publicar y ofrecer soporte de software de primer nivel con total seguridad. Es la única fuente de información de todo el ciclo de vida de desarrollo, ya que permite a los equipos autónomos moverse con rapidez dentro del contexto y, al mismo tiempo, permanecer conectados con una estrategia empresarial más amplia.**jira**

2.1.10. Microsoft Teams

Microsoft Teams es una plataforma de colaboración que integra herramientas de comunicación como chat, videollamadas y gestión de archivos. Facilita la interacción y coordinación entre equipos de desarrollo, especialmente en entornos de trabajo remoto.**Microsoft Teams**

2.1.11. Material-UI (MUI)

MUI es una biblioteca de componentes de interfaz de usuario basada en el diseño de Material Design de Google. Proporciona componentes predefinidos y personalizables, lo que acelera el desarrollo de interfaces visualmente atractivas y consistentes.**MaterialUI**

2.2. Antecedentes

Nombre del Trabajo	Autor	Aporte e Innovación
Migración de módulos en un ERP empresarial basado en Java a frameworks modernos	Mateo Baena Chavarriaga	La migración de módulos a microservicios es un componente fundamental para la arquitectura de la nueva plataforma de Contenido. El enfoque en la modernización de la infraestructura, utilizando frameworks modernos, se alinea con la necesidad de mejorar la escalabilidad y flexibilidad de la plataforma, un aspecto clave de mi trabajo de grado.
Diseño e implementación de un modelo de gestión de inventarios para mejorar el manejo de entradas y salidas de dotación y documentación para la empresa Acciones&Servicios	Wilmer Ferney Gelvez Ramirez	Este proyecto es relevante para el módulo de gestión de productos, específicamente en lo que respecta a la optimización de procesos de manejo y control de inventarios. La implementación de soluciones prácticas que mejoren la eficiencia operativa se pueden aplicar a los módulos de gestión de productos y estadísticas en la nueva plataforma de Contenido.
Towards Interoperability of APIs - an LLM-based approach	René Lehmann	La propuesta de mejorar la interoperabilidad entre APIs con el uso de modelos de lenguaje se puede aplicar a la exposición de APIs en la nueva plataforma de Contenido, facilitando la integración con otros sistemas. Este enfoque innovador de superar las diferencias semánticas y sintácticas es crucial para garantizar la conectividad fluida entre módulos y servicios externos.

Tabla 2: Antecedentes de trabajos relacionados

3. Metodología

3.1. Metodología de Gestión de Actividades

Para garantizar una organización eficiente del trabajo, el desarrollo de la nueva plataforma de Contenido sigue el marco de trabajo ágil Scrum. Este enfoque permite la entrega continua de valor y la adaptación a los cambios del mercado digital.

El equipo de desarrollo está compuesto por seis integrantes, distribuidos en los roles de Product Owner, Scrum Master y Scrum Team. La planificación y ejecución de tareas se estructura en sprints de dos semanas, siguiendo estas prácticas:

1. Sprint Planning: Selección de tareas del Product Backlog, estimadas en Sprint Points utilizando la secuencia de Fibonacci.
2. Daily Scrums: Reuniones cortas (martes y jueves) para evaluar el progreso y resolver impedimentos.
3. Sprint Review: Presentación de avances y validación de funcionalidades implementadas.
4. Sprint Retrospective: Evaluación del sprint para identificar mejoras en los procesos de desarrollo.

Este enfoque garantiza que cada funcionalidad desarrollada cumpla con los requerimientos del proyecto y responda a las necesidades del mercado.

3.2. Metodología de Desarrollo de Software

El desarrollo de la nueva plataforma de Contenido se basa en una arquitectura modular y escalable, con un fuerte enfoque en la seguridad y la integración con otros sistemas. Para ello, se emplean las siguientes estrategias:

1. Desarrollo Backend: Implementación de nuevos módulos en el backend para mejorar la gestión de productos, la generación de estadísticas y la seguridad de la plataforma.
2. Exposición de APIs: Creación de endpoints que permitan la integración con sistemas externos, facilitando la interoperabilidad con otras plataformas.
3. Seguridad y Autenticación: Uso de tokens, middlewares y validaciones avanzadas para fortalecer el acceso a la plataforma y proteger la información.
4. Optimización del Rendimiento: Aplicación de buenas prácticas en el procesamiento de datos para garantizar eficiencia y escalabilidad.

El desarrollo seguirá principios de buenas prácticas en programación, garantizando código limpio, mantenible y alineado con los estándares de la industria.

3.3. Metodología de Investigación

Dado que el proyecto no solo implica el desarrollo de software, sino también la modernización de una plataforma existente, se adopta un enfoque de investigación aplicada. Este proceso incluye:

1. Análisis del estado actual de la plataforma: Identificación de limitaciones técnicas y necesidades de mejora.
2. Revisión de literatura: Exploración de tendencias tecnológicas, metodologías y estándares de seguridad aplicables al comercio electrónico.
3. Evaluación comparativa: Análisis de otras plataformas similares para identificar mejores prácticas y oportunidades de innovación.
4. Validación de resultados: Pruebas funcionales y recopilación de retroalimentación para asegurar que las nuevas funcionalidades cumplen con los objetivos planteados.

Este enfoque permite fundamentar las decisiones técnicas y garantizar que las soluciones implementadas respondan a las demandas del mercado digital en constante evolución.

4. Ingeniería de desarrollo

4.1. Requerimientos del Sistema

4.1.1. Objetivo

El objetivo de esta fase fue analizar y documentar los requerimientos funcionales y no funcionales de los módulos a desarrollar de la nueva versión de la plataforma *Contento*, con el fin de garantizar que su desarrollo respondiera a las expectativas del negocio y cumpliera con los estándares de calidad definidos.

A partir de este proceso se establecieron los comportamientos esperados, restricciones técnicas y criterios de validación para cada módulo —**gestión de productos, estadísticas, seguridad y exposición de API**— los cuales guiaron el diseño e implementación del backend.

A continuación, se detallan los requerimientos funcionales y no funcionales en formato de Historias de Usuario.

4.1.2. Módulo de Gestión de Productos (GP)

Tabla 3: RF-GP-01: Crear producto

ID HU	RF-GP-01	
Título	Crear producto	
Como	Usuario - Plataforma	
Quiero	Poder crear un producto en la plataforma	
Descripción	Se debe poder crear un producto en la plataforma para que el negocio exponga su catálogo de productos a los clientes finales.	
Prioridad	Alta	
Criterio	Evento	Resultado
Creación exitosa	El usuario ingresa todos los datos válidos (sku, nombre, precio...).	El sistema valida y crea el producto respondiendo 201 Created.
Creación fallida	El usuario digita datos inválidos.	El sistema devuelve error 400 informando el fallo.
Seguridad	Usuario sin permisos.	El sistema devuelve 401/403.

Tabla 4: RF-GP-02: Obtener productos

ID HU	RF-GP-02	
Título	Obtener productos	
Como	Usuario - Plataforma	
Quiero	Poder obtener un listado paginado de productos con filtros	
Descripción	Se debe poder consultar el catálogo de productos por diferentes filtros (slug, sku, id) y paginación para mostrarlo a los clientes.	
Prioridad	Alta	
Criterio	Evento	Resultado
Listado pagina- do	Se solicitan productos con pa- ge=1 y limit=50.	El sistema responde 200 OK con datos y metadatos.
Filtros de búsqueda	Se filtra por texto en slug/nombre.	El sistema retorna coinciden- cias parciales (nombre) y exac- tas (SKU).
Parámetros inválidos	Se envía formato incorrecto.	El sistema devuelve 400 Bad Re- quest.

Tabla 5: RF-GP-03: Actualizar producto

ID HU	RF-GP-03	
Título	Actualizar producto	
Como	Usuario - Plataforma	
Quiero	Poder actualizar parcialmente un producto existente	
Descripción	Se debe permitir modificar uno o más campos de un producto, in- cluyendo la gestión de imágenes.	
Prioridad	Alta	
Criterio	Evento	Resultado
Actualización exitosa	Se envía body válido con campos parciales.	El sistema responde 200 OK con el producto actualizado.
Unicidad	Se intenta duplicar SKU o Slug.	El sistema devuelve 409 Con- flict.
Gestión de imágenes	Agregar o eliminar imágenes.	El sistema actualiza las relacio- nes correctamente.

Tabla 6: RF-GP-04: Eliminar producto

ID HU	RF-GP-04	
Título	Eliminar producto	
Como	Usuario - Plataforma	
Quiero	Eliminar un producto del catálogo (lógica)	
Descripción	Se debe permitir eliminar un producto conservando sus datos históricos (soft delete).	
Prioridad	Media	
Criterio	Evento	Resultado
Eliminación exitosa	Se ejecuta DELETE sobre ID existente.	El producto queda inactivo y oculto (200 OK).
Conservación	Consulta en BD tras eliminación.	El registro físico persiste, manteniendo integridad.

Tabla 7: RF-GP-05: Obtener detalles de producto

ID HU	RF-GP-05	
Título	Obtener detalles de producto	
Como	Usuario - Plataforma	
Quiero	Poder obtener el detalle completo por identificador	
Descripción	Endpoint para consultar todos los atributos, relaciones e imágenes a partir de un identificador único.	
Prioridad	Alta	
Criterio	Evento	Resultado
Detalle exitoso	Consulta con ID, Slug o SKU válido.	El sistema responde 200 OK con el detalle completo.
Precedencia	Múltiples identificadores enviados.	Prioridad: ID > SKU > Slug.
Inexistente	Identificador no encontrado.	El sistema responde 404 Not Found.

Tabla 8: RF-GP-06: Importar productos desde Excel

ID HU	RF-GP-06	
Título	Importar productos desde Excel	
Como	Usuario - Plataforma	
Quiero	Poder crear productos en lote desde un archivo Excel	
Descripción	Carga masiva mediante archivo .xlsx, validando datos mínimos requeridos por fila.	
Prioridad	Alta	
Criterio	Evento	Resultado
Importación exitosa	Carga de archivo válido.	El sistema responde 201 e incluye productos generados.
Errores de formato	Archivo corrupto o columnas faltantes.	El sistema responde 400 con detalle del error.
Reglas de negocio	Precios o cantidades negativas.	Se reporta error específico por fila.

4.1.3. Módulo de Estadísticas (ES)

Tabla 9: RF-ES-07: Obtener distribución geográfica

ID HU	RF-ES-07	
Título	Obtener distribución geográfica	
Como	Usuario - Plataforma (Análisis)	
Quiero	Ver la distribución geográfica de los clientes	
Descripción	Obtener clientes agrupados por ubicación (país, ciudad) con conteos y porcentajes.	
Prioridad	Media	
Criterio	Evento	Resultado
Consulta exitosa	Solicitud de informe.	Responde 200 OK con ubicaciones y porcentajes.
Sin datos	Tienda sin clientes.	Retorna arreglo vacío (no error 404).

Tabla 10: RF-ES-08: Promedio de ventas

ID HU	RF-ES-08	
Título	Promedio de ventas	
Como	Usuario - Plataforma (Análisis)	
Quiero	Consultar el promedio de ventas y su variación	
Descripción	Obtener promedio actual comparado con el periodo anterior.	
Prioridad	Media	
Criterio	Evento	Resultado
Consulta exitosa	Solicitud de métricas.	200 OK con promedio actual, anterior y % variación.

Tabla 11: RF-ES-09: Obtener visitas y variación

ID HU	RF-ES-09	
Título	Obtener visitas y variación	
Como	Usuario - Plataforma (Análisis)	
Quiero	Consultar el conteo de visitas y su variación	
Descripción	Comparativo de tráfico actual vs periodo anterior.	
Prioridad	Media	
Criterio	Evento	Resultado
Consulta exitosa	Solicitud de conteo.	200 OK con <code>currentCount</code> , <code>prevCount</code> y <code>change</code> .

Tabla 12: RF-ES-10: Métodos de pago

ID HU	RF-ES-10	
Título	Métodos de pago	
Como	Usuario - Plataforma (Análisis)	
Quiero	Consultar el uso de distintos métodos de pago	
Descripción	Desglose de transacciones por método de pago y evolución.	
Prioridad	Media	
Criterio	Evento	Resultado
Consulta exitosa	Solicitud de estadísticas.	200 OK con arreglo de métodos y sus métricas.
Integridad	Verificación de datos.	La suma de porcentajes de uso es 100 %.

Tabla 13: RF-ES-11: Obtener productos top

ID HU	RF-ES-11	
Título	Obtener productos top	
Como	Usuario - Plataforma (Análisis)	
Quiero	Consultar listado de productos con mayores ventas	
Descripción	Ranking de productos por unidades vendidas o monto total.	
Prioridad	Alta	
Criterio	Evento	Resultado
Consulta exitosa	Solicitud de ranking.	200 OK con listado paginado ordenado por rendimiento.

4.1.4. Exposicion de Api para integraciones (EAI)

Tabla 14: RF-EAI-01: Procesamiento Webhook Pagos

ID HU	RF-EAI-01	
Título	Procesamiento de Webhook de Pagos (BePay)	
Como	Pasarela de pagos BePay	
Quiero	Notificar a Contento sobre el estado final de una transacción	
Descripción	Recepción y procesamiento de notificaciones de pago para actualizar facturas e inventario.	
Prioridad	Alta	
Criterio	Evento	Resultado
Seguridad (Firma)	Recepción de petición.	Se valida firma de seguridad. Si falla, retorna 401/403.
Recepción de datos	Firma válida.	Se interpreta correctamente el payload (ID, estado).
Actualización	Transacción aprobada.	Se actualiza factura y se ajusta inventario.
Notificación	Proceso finalizado.	Se envía correo al cliente y log de auditoría.

4.1.5. Requerimientos No Funcionales (RNF)

Tabla 15: RNF-01: Rendimiento

ID HU	RNF-01	
Título	Rendimiento	
Como	Sistema Backend	
Quiero	Mantener un tiempo de respuesta óptimo	
Descripción	Las operaciones críticas deben tener tiempos de respuesta bajos.	
Prioridad	Alta	
Criterio	Evento	Resultado
Medición del P95	Se miden operaciones críticas.	Tiempo de Respuesta < 500 ms.

Tabla 16: RNF-02: Seguridad

ID HU	RNF-02	
Título	Seguridad (Auth)	
Como	Sistema Backend	
Quiero	Garantizar acceso seguro a la información	
Descripción	Implementación exclusiva mediante JWT y RBAC.	
Prioridad	Alta	
Criterio	Evento	Resultado
Auditoría	Revisión de endpoints.	Todos protegidos por JWT y permisos.

Tabla 17: RNF-03: Mantenibilidad

ID HU	RNF-03	
Título	Mantenibilidad	
Como	El Equipo de Desarrollo	
Quiero	Asegurar que el código sea fácil de probar	
Descripción	Arquitectura modular y cobertura de pruebas mínima.	
Prioridad	Alta	
Criterio	Evento	Resultado
Cobertura	Reporte de pruebas.	Cobertura $\geq$ 80 %.

Tabla 18: RNF-04: Interoperabilidad

ID HU	RNF-04	
Título	Interoperabilidad	
Como	El API para Integraciones	
Quiero	Facilitar la conexión estandarizada	
Descripción	Documentación OpenAPI (Swagger) accesible.	
Prioridad	Alta	
Criterio	Evento	Resultado
Documentación	Revisión de Swagger.	Documentación 100 % actualizada.

Reflexión

La identificación y documentación de los requerimientos funcionales y no funcionales permitió establecer una visión clara del sistema antes de su desarrollo, garantizando la alineación entre las necesidades del negocio y la arquitectura técnica. Este proceso fue fundamental para definir los endpoints, validar las reglas de negocio y asegurar que cada respondiera a los objetivos de la plataforma *Contento*. Además, esta documentación sirvió como referencia directa durante las etapas de diseño, codificación y pruebas, reduciendo riesgos e incrementando la calidad del producto final.

4.2. Diseño de la Arquitectura de Software

4.2.1. Objetivo

Para cumplir con los objetivos de escalabilidad y mantenibilidad, se diseñó una arquitectura basada en el patrón de **Monolito Modular** utilizando el framework **NestJS**. Esta elección permite consolidar la lógica de negocio, la seguridad y el acceso a datos en una única unidad de despliegue, manteniendo al mismo tiempo una estricta separación interna de componentes basada en los dominios del negocio.

El diseño prioriza la cohesión funcional y la claridad estructural. A continuación, se detallan los componentes fundamentales de la arquitectura implementada.

4.2.2. Arquitectura General e Integración

A nivel macro, el sistema opera como un núcleo orquestador centralizado. El backend expone una API RESTful que actúa como punto de entrada único para los clientes (Frontend React), gestionando la comunicación a través del protocolo HTTP/S.

Como se observa en la Figura 7, el servidor integra internamente los módulos funcionales del negocio y gestiona la comunicación asíncrona con servicios externos críticos, separando claramente la infraestructura propia (Base de Datos, Almacenamiento) de los servicios de terceros (Pasarela de Pagos BePay).

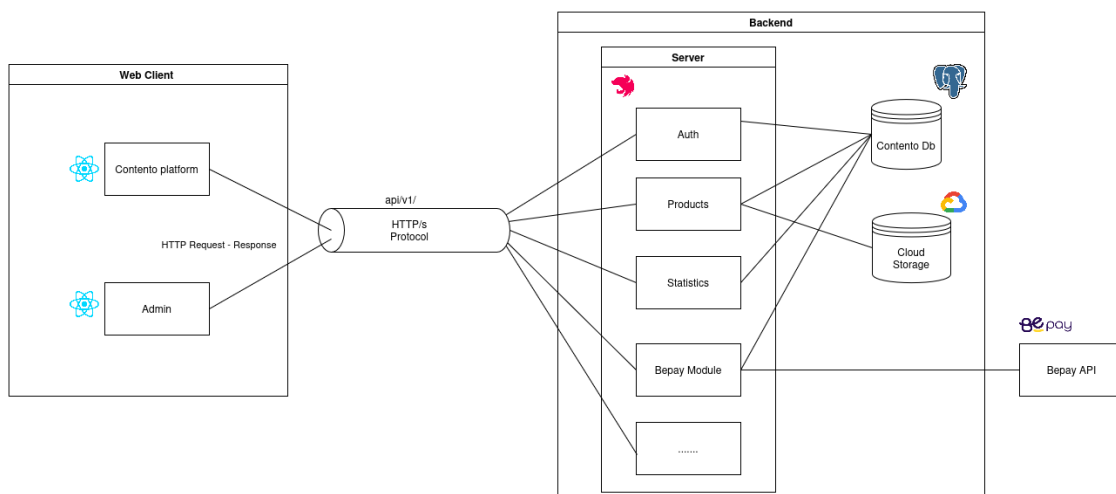


Figura 7: Diagrama de arquitectura general del sistema e integración de servicios.

4.2.3. Estructura Modular y Organización del Código

Uno de los requerimientos no funcionales clave fue asegurar la mantenibilidad y escalabilidad del código. Para ello, se adoptó la arquitectura modular nativa de NestJS. El proyecto se estructura separando claramente las responsabilidades en módulos independientes (Auth, Products, Statistics, Common), lo que permite que cada equipo de desarrollo trabaje en funcionalidades específicas sin generar conflictos en el código base.

La Figura 8 evidencia la organización del directorio `src`, donde se destacan los módulos de negocio y las carpetas transversales de configuración y utilidades.

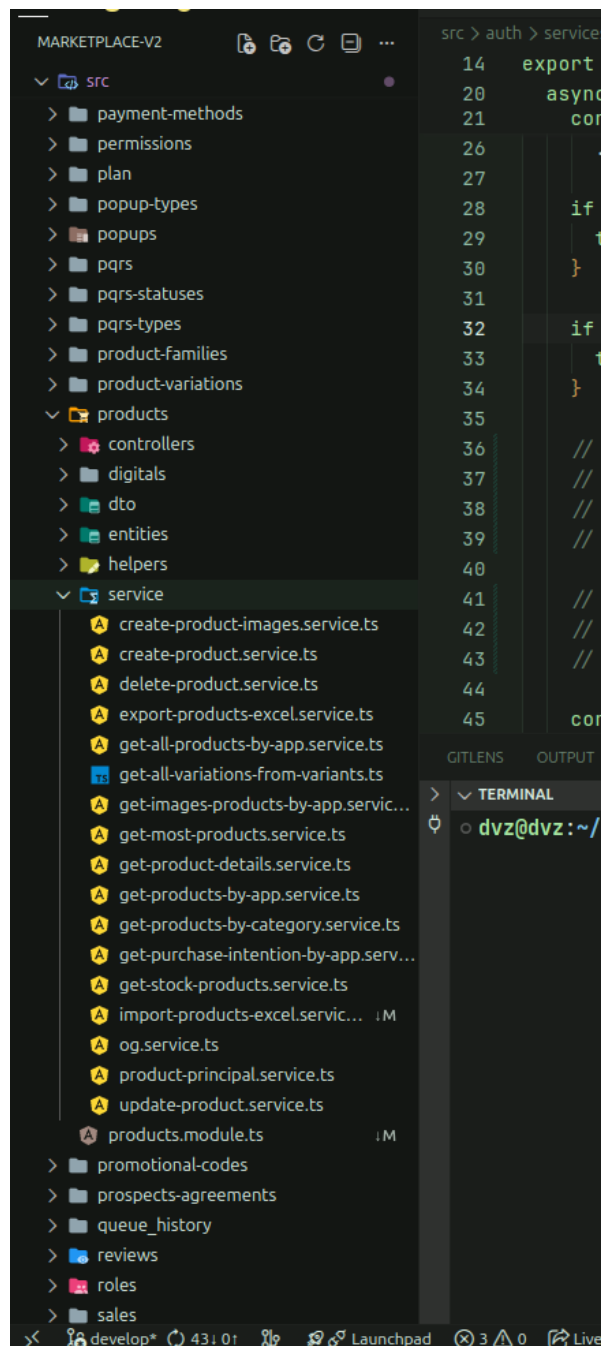


Figura 8: Estructura modular del proyecto en el entorno de desarrollo.

4.2.4. Diseño del Modelo de Datos

Para garantizar la integridad y persistencia de la información, se seleccionó un modelo de base de datos relacional implementado en **PostgreSQL**. Esta elección responde a la necesidad de manejar relaciones complejas entre productos, variantes, pedidos y usuarios de manera eficiente.

El diseño del esquema de base de datos sigue las formas normales para evitar redundancia.

La gestión de la base de datos se realiza a través de **TypeORM**, lo que permite una abstracción orientada a objetos de las entidades del sistema. En la Figura 9 se presenta el Diagrama Entidad-Relación (DER) que soporta los módulos de gestión de productos y estadísticas.

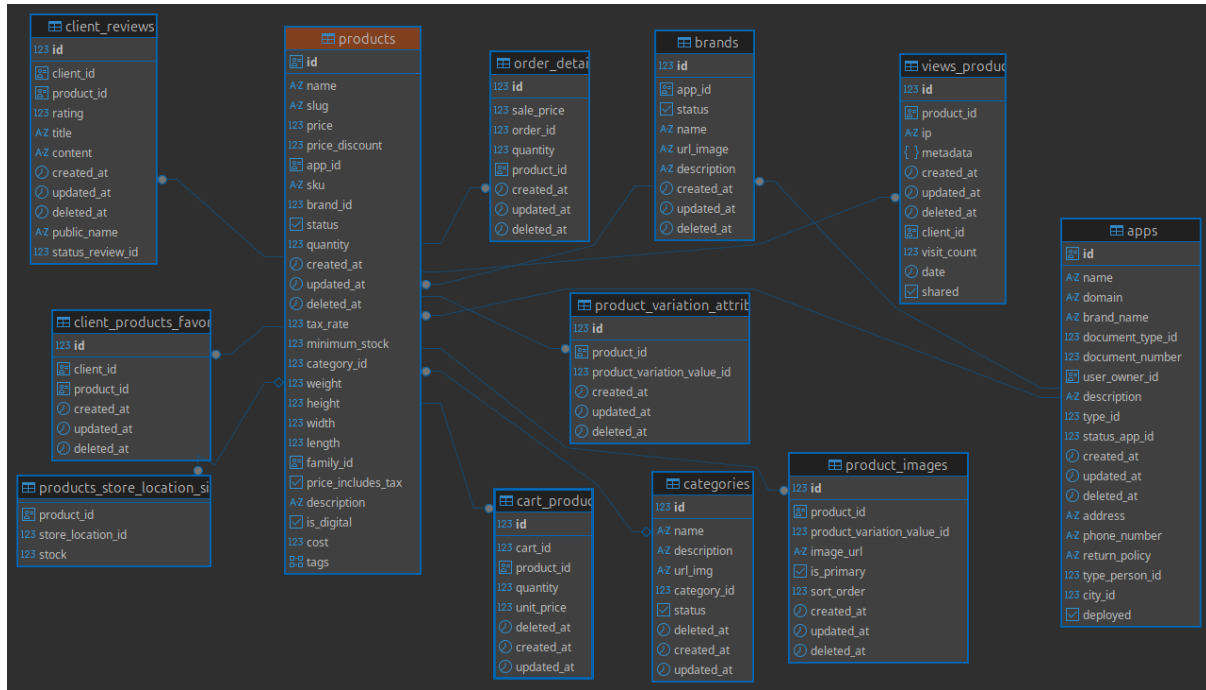


Figura 9: Diagrama Entidad-Relación (DER) de la plataforma.

4.2.5. Patrones de Diseño Implementados

Para resolver problemas comunes de desarrollo y estandarizar la construcción del software, se aplicaron los siguientes patrones de diseño, los cuales son verificables en el código fuente presentado en las secciones posteriores:

- **Inyección de Dependencias (Dependency Injection):** Utilizada para gestionar la creación y vinculación de servicios dentro de los módulos, reduciendo el acoplamiento entre clases.
- **Data Transfer Objects (DTO):** Se implementaron DTOs junto con la librería `class-validator` para asegurar que los datos que ingresan a la API cumplan con los formatos requeridos antes de ser procesados por la lógica de negocio 27.
- **Guards y Decoradores:** Se utilizó el patrón de interceptor mediante *Guards* 24 personalizados para manejar la autenticación y autorización (RBAC) de manera transversal, protegiendo los endpoints críticos sin ensuciar la lógica de los controladores.
- **Repository Pattern:** A través de TypeORM, se abstraizó la capa de acceso a datos, permitiendo realizar operaciones CRUD sin escribir consultas SQL directas en el código de negocio.

4.3. Gestión de Productos

4.3.1. Objetivo

El módulo de gestión de productos centraliza y automatiza las operaciones de creación, consulta, actualización y eliminación del catálogo en la plataforma Contenido. Está orientado al rol de administrador y garantiza un manejo eficiente, consistente y trazable de la información, manteniéndola organizada, actualizada y sincronizada con el resto de funcionalidades del sistema.

4.3.2. Funciones Principales

- **Creación de productos:** Registra nuevos productos con atributos como nombre, descripción, precio, stock, categoría, imágenes entre otros.
- **Consulta de productos por tienda:** Devuelve un listado paginado y filtrable de productos.
- **Actualización parcial:** Permite modificar campos específicos sin sobrescribir información no incluida en la petición. Se utiliza `PartialType` en los DTOs para garantizar actualizaciones seguras y controladas.
- **Eliminación:** Permite eliminar un producto de tal manera que marca el registro con una bandera (*soft delete*) en lugar de su eliminación física.
- **Importación masiva desde Excel:** Permite cargar un archivo `.xlsx` (vía `multipart/form-data` campo `file`) para crear productos en lote. Valida columnas obligatorias (por ej., `sku`, `nombre`, `precio`, `stock`, `categoría`).
- **Exportar productos a Excel:** Permite exportar productos en un archivo `.xlsx` donde se muestra toda la información relacionada con los productos de una tienda.
- **Consulta de detalle de producto:** Retorna la información completa de un producto específico, identificado por su `id`, `slug` o `sku`. La respuesta incluye todos los detalles del producto.

4.3.3. Creación de Productos (POST /app/:appId/product)

Esta funcionalidad permite registrar nuevos productos dentro de la plataforma. Incluye campos para nombre, descripción, precios, imágenes y variantes.

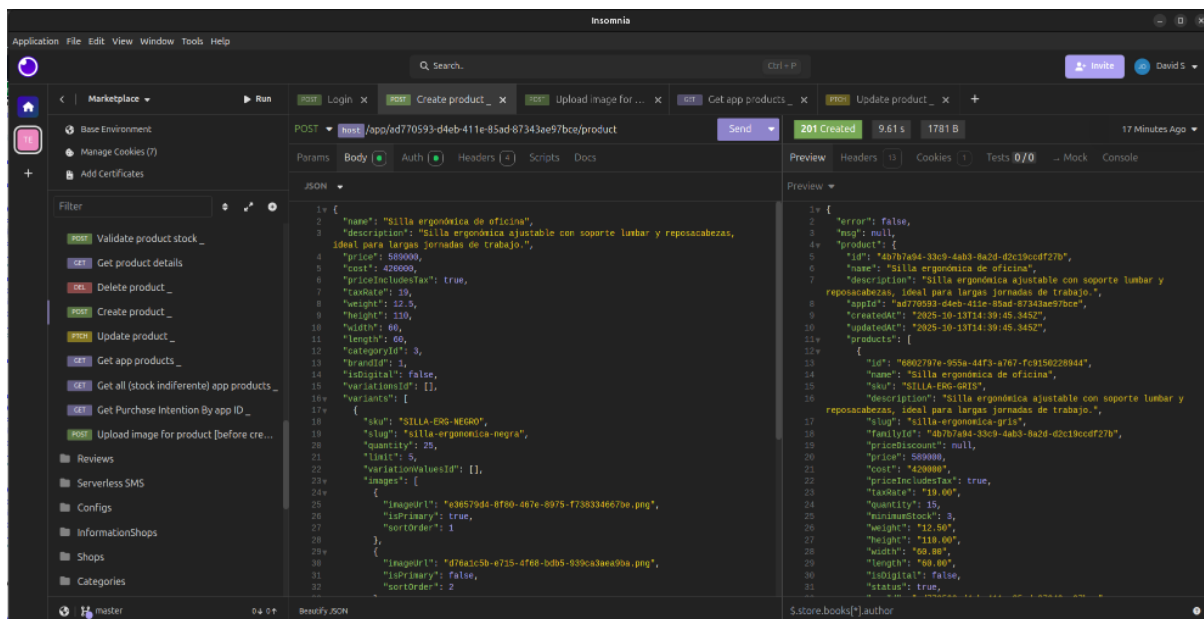


Figura 10: Creación de producto con imágenes y variantes.

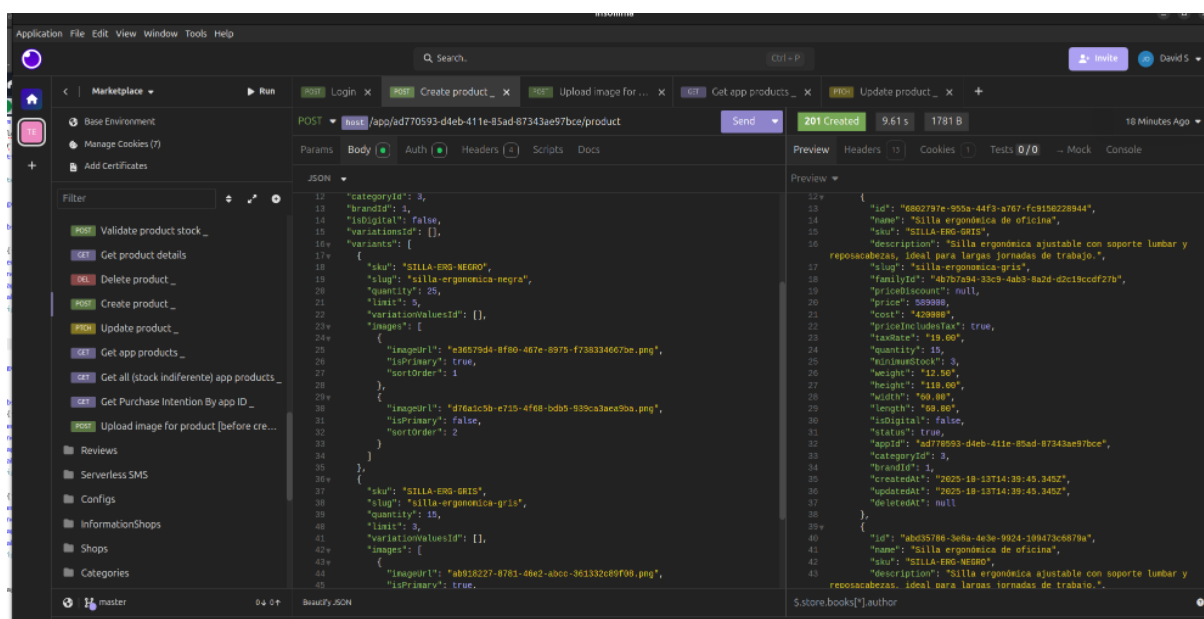


Figura 11: Formulario extendido de creación de producto con múltiples atributos.

4.3.4. Obtener Productos de una Tienda (GET /app/:appId/products)

Permite listar los productos registrados en una tienda determinada. Soporta paginación, filtros por slug, sku o categoría.

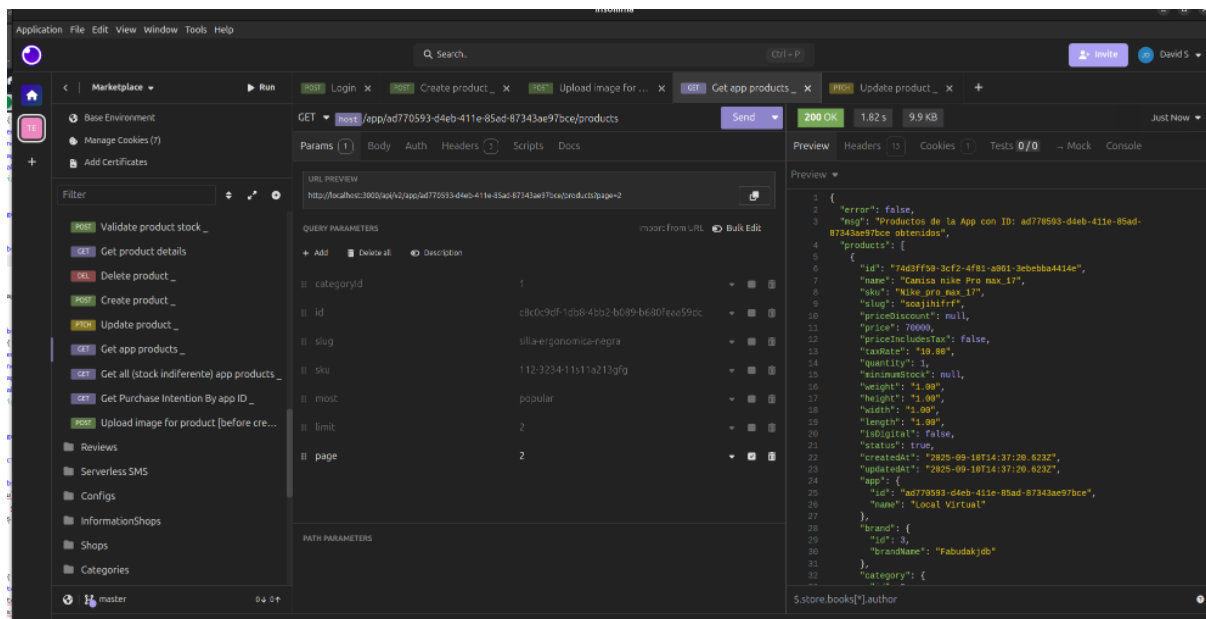


Figura 12: Listado de productos con paginación.

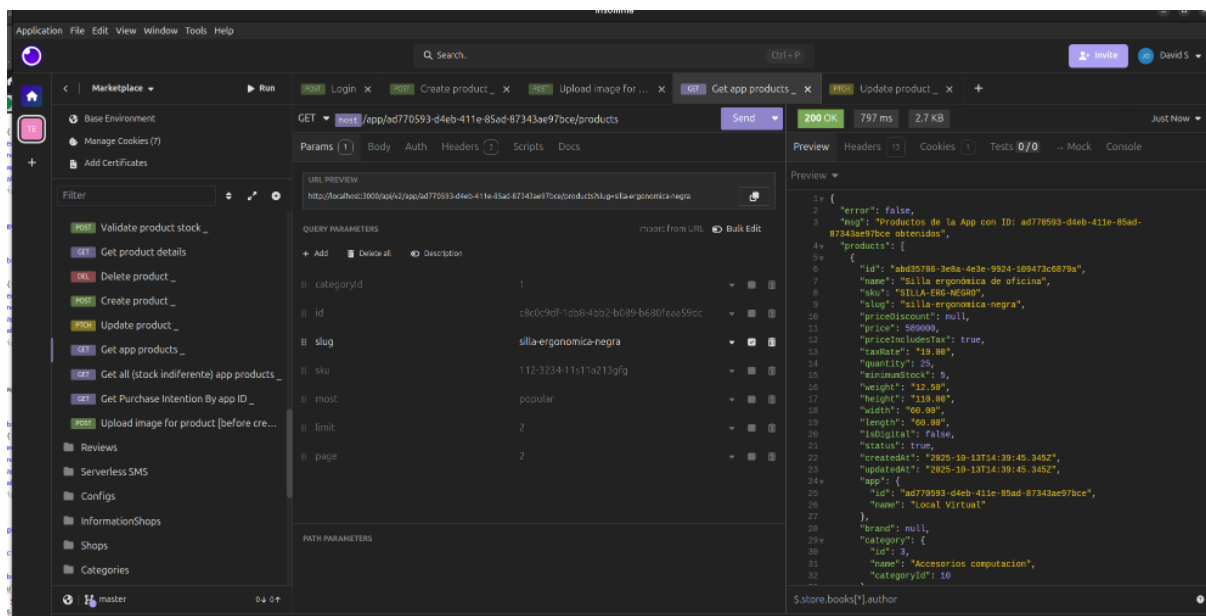


Figura 13: Búsqueda de productos por slug.

4.3.5. Actualizar Productos (PATCH /app/:appId/product/:id)

Esta operación permite modificar parcialmente los datos de un producto sin sobrescribir los campos no enviados en la solicitud.

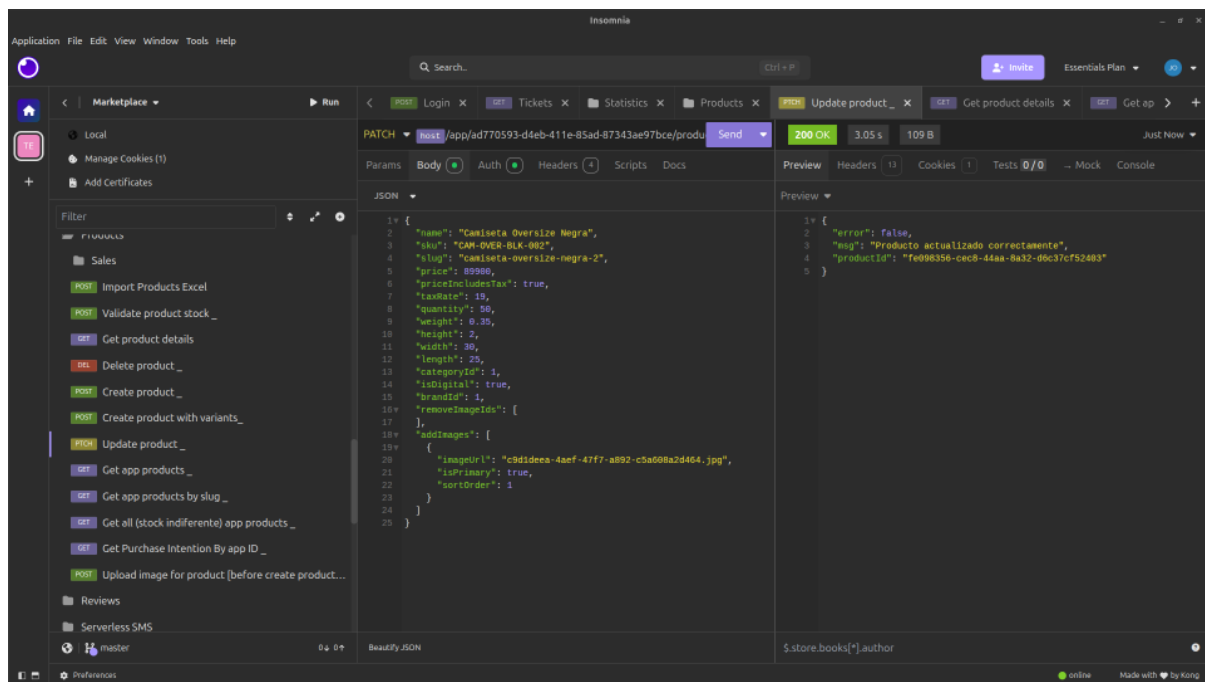


Figura 14: Actualización de datos de un producto existente.

4.3.6. Eliminar Productos (DELETE /app/:appId/product/:id)

El sistema implementa un mecanismo de eliminación lógica que marca el producto como inactivo en lugar de borrarlo físicamente. Esto garantiza la trazabilidad de la información y evita la pérdida de relaciones con otras entidades (como pedidos o reportes).

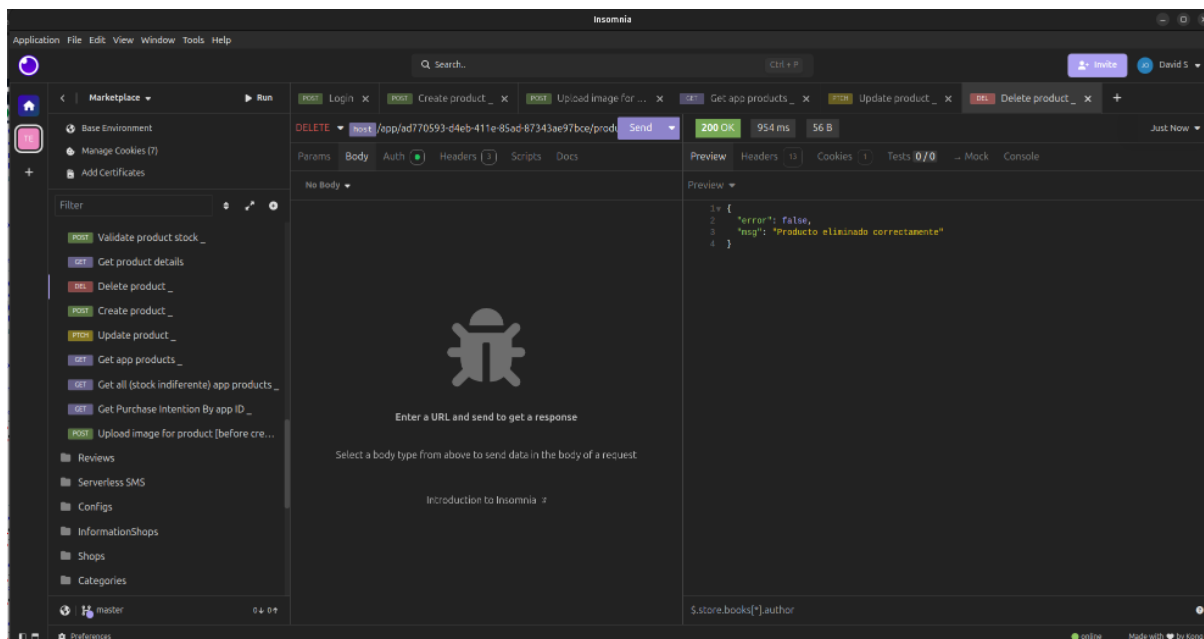


Figura 15: Eliminación lógica de un producto (soft delete).

4.3.7. Importar Productos desde Archivo Excel (POST /app/:appId/product/import)

Esta herramienta facilita la carga masiva de productos mediante archivos `.xlsx`. El sistema valida formato, columnas obligatorias y consistencia de datos antes de realizar la inserción en lote. Los errores de formato se reportan de forma detallada al usuario.

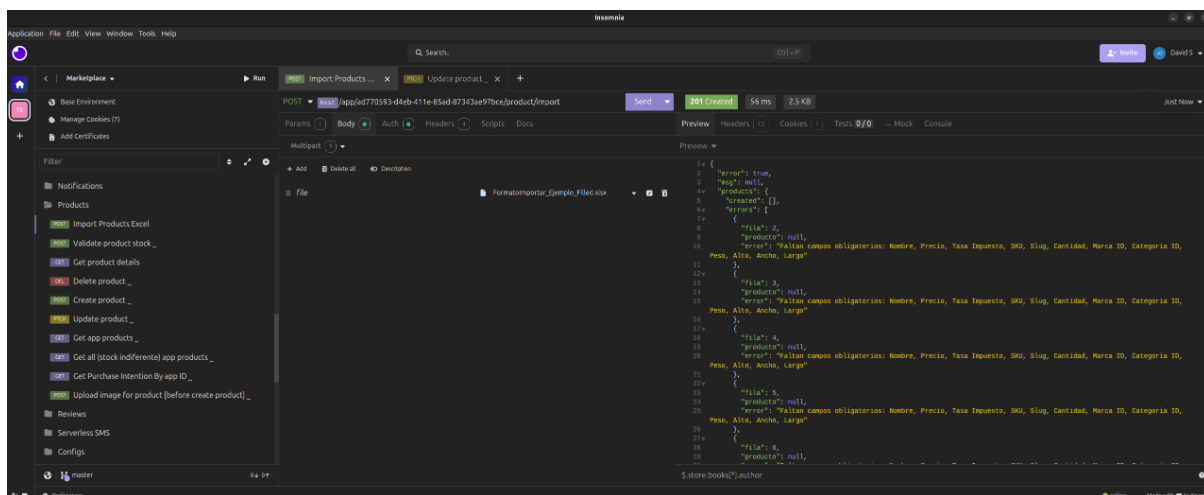


Figura 16: Intento de importación con formato inválido.

4.3.8. Exportar productos en archivo de excel (POST /app/:appId/product/export)

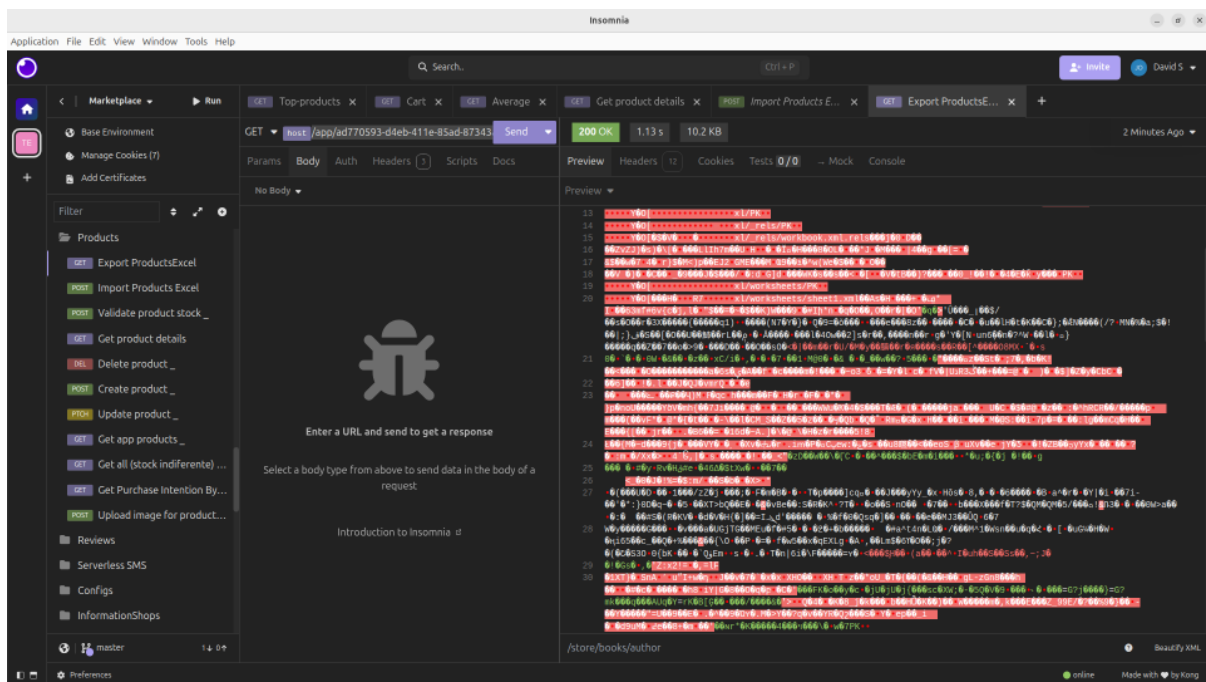


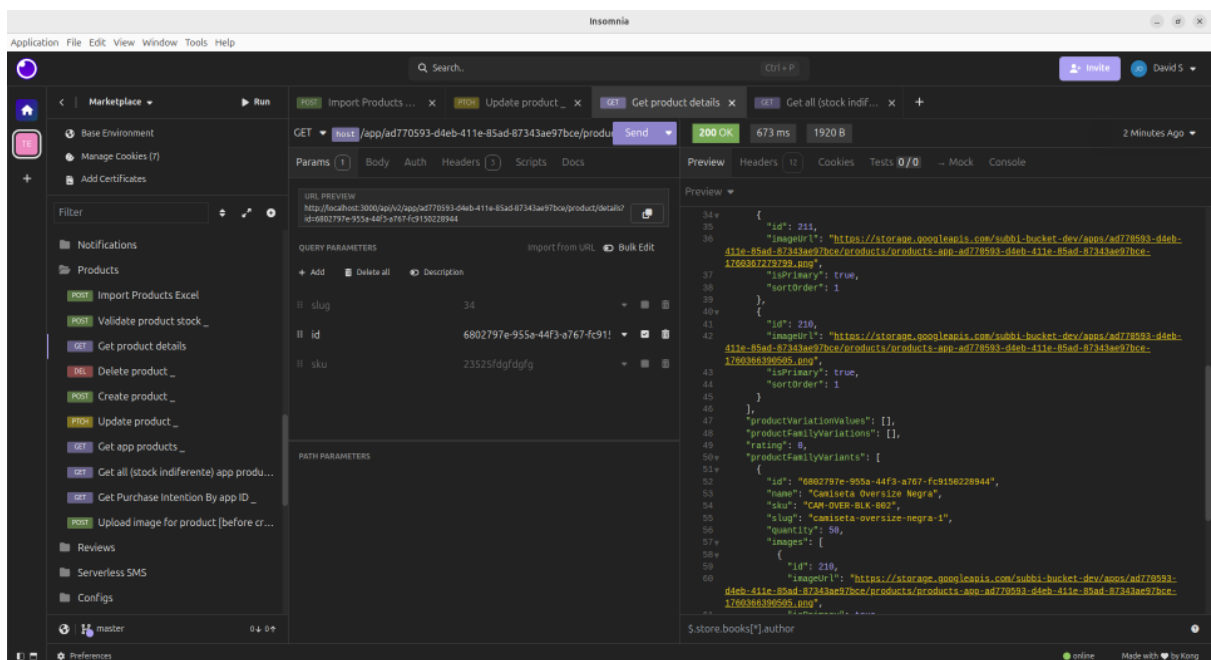
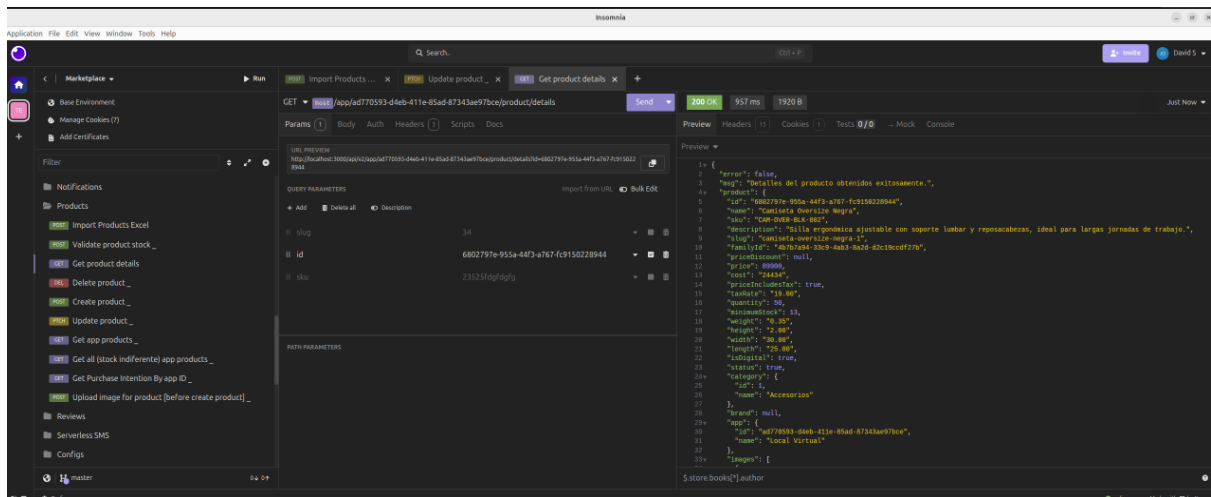
Figura 19: Endpoint para obtener archivo de excel con informacion de productos

products.xlsx — LibreOffice Calc										
File Edit View Insert Format Styles Sheet Data Tools Window Help										
Calibri 11pt B U I A Z Color Background Image Link Table Formula Sort Filter AutoSum										
A1										
	A	B	C	D	E	F	G	H	I	J
1	ID	Nombre	SKU	Descripción	Slug	Precio	Precio Descuento	Tasa Impuesto	Cantidad	Marca
2	74d3f505-3c72-4f81-a061-3ebecb64414e	Camisa Nike Pro max 17	Nike_pro_max_17	COMEDOR-4-PUESTOS ejemplo2	soajihfrf	70000		10.00	1	Fabudaskjib
3	8567c725-52bc-4a16-a916-e60c7031b064	Camiseta	COMEDOR-4-PUESTOS prueba1	comedor-4-puestos	comedor-4-puestos	43455		10.00	100	
4	e1808c31-463e-46de-ae7a701705713490		GORAR_TQP	miranda descripción	soajihfrfs	70000		10.00	1	Fabudaskjib
5	49a96d9-801e-4b02-979c-f2c303030303	Producto Físico (válido)	SKU-781	Fisica prueba nassiva 6	comedor4771	49000		10.00	1	Fabudaskjib
6	7610503c-ff9c-4272-b8bc-6942072567	asdfasdf asdf asdf asd	31c2	cp fasd fasd fasd fasd fcp	12332	120000		0.00	12	
7	d46c9d8-60bd-4c2c-7211-4083c73b4649	Pantalón level 5	level5 20	cnvls 20	cnvls3y9fy4	80000		10.00	0	Fabudaskjib
8	420e8dd4-e1a-4642-b17a-0787d4b7fab0	Comedor para comer	COMEDOR-8-PUESTOS comedor	comedor-8-puestos	comedor-8-puestos	43455		10.00	100	
9	bed4f447-e447-4d5a-b63b-e31c5f7f255a	GORAR	GORAR_Gris	soajihfrf	soajihfrf	70000		10.00	1	Nike
10	406446c-7a64-49d2-476b-b255a16d1b91	Nombre del producto con	SKU-54877	cp-Descripción del producto -strong-Esta es prueba del producto con	prueba-de-front	10000		0.00	1	Nike
11	6802797e-955a-44f3-a767-c1f50228944	Camiseta oversize negra	CAM-OVER-BLK-002	Silla ergonomica ajustable con soporte lumbar	camiseta-oversize-negra-1	89900		19.00	50	
12	8dfc3eb3-5d32-4656-8777-abe5d6317130	Zapatos bonitos baratos	SKU009991	unos zapatos bien bonitos	zapatos-lindos-baratosos	123999		0.00	123	Adidas
13	30f06f6e-76f1-46a8-ae42-af2f8eb0a30a79	asdf asdf asdf asdf	123c12	cp fasd fasd fasd fasd fasd fcp	com4	120000		0.00	10	
14	5278698f-4d0d-45e3-3449c-955c2b89522a	Puma	PUMA_RETRO	centra4	centra4	250000		0.00	0	Fabudaskjib
15	70453f2c-3d0f-49bc-9e34-0a3ab00b1f7a	Prueba de front 1	PruebaFront124	prueba-de-front	prueba-de-front	1299991		0.00	101	Adidas
16	4925940-c929-4a65-963e-50f1c5224ae7	Importado	SKU-test	Prueba Importacion	slug-test	200000		21.00	10	
17	75245207-2a6e-4832-404d-6e0e9d9f5b09	producto digital	SKU-DIG123	cp-Esta dertmjj descripción de producto -strong-slug-digital	slug-digital	200000		0.00	200	Adidas
18	18744505-274d-461a-4a45-023bde953aaf	Camisa Blannennon	123	cp-fasd asf fasd fasd fasd asf fcp	bolafas fasd fasd fasd	20000		0.00	2317	Fabudaskjib
19	4d37f128-4b1f-41c9-8693-364e4a40c	Prueba de productooos	qqeq12e	cp-esta es una -strong-prueba de producto1234k-daeq-dasf	1234k-daeq-dasf	120000		0.00	12	Fabudaskjib
20	5034f52e-2c33-45ad-ab04-549d9c1768b1	asdfasdfa	2112	cp-fasdfsasdfsadf fcp	cp-fasdfsasdfsadf	222222		0.00	22	
21	52823149-4ee3-4baf-aba7-3cb6f9d4188a	Camisa Nike Pro max	Nike_pro_max	soajihfrf	soajihfrf	70000		10.00	1	Fabudaskjib
22	2e5afdf7-453b-42a8-829a-b3c38337087	Producto digital	dwqet2dm12	cp-fsfsasda dsadas -strong-fasdfsasdfsadf	dad.asdfs.f	120000		0.00	12	Fabudaskjib
23	21847656-2d30-469f-4ac7-4dwa323117c1	ddadadweda digital	23132	cp-asdf asdf asdf fcp	ddadadweda	333333		0.00	22	
24	c8b1c40e-7944-49bc-9f31-5bd6d0e69e9	Importado	Importado-10	si tiene ejemplos ya	importado-101	200000		21.00	10	
25	20a40c3d-eaef-430d-9a8b-074f10aa706	Producto Demo 5	SKU-4005	Producto de prueba generado automáticamente	comedor-producto-demo-5	426992		0.05	144	Zapatos
26	7470341f-418b-4a59-895d-6075a0c8761f	Pantalon level 5	level5	cnvls 20	cnvls3y9fy4	80000		10.00	10	Fabudaskjib
27	b05514d3-7dab-4139-913c-c023a39c469a	GORAR	GORAR_ROJA	necesito dos de esto	comedor-4-puestos	70000		10.00	1	Fabudaskjib
28	30b36362-74e7-4aee-8706-947f7c31161c	sofasdfsasf sd fasdfsdf	e22223d0dddd	cp-strong-fasd asdf asdf -strong- fasd fasd fasd fasd fasd fcp	comedor-4-puestos	43455		10.00	20	Adidas
29	f1c12b4c-75f8-45f6-bd5-4fd79947c1c6	Comedor para comer	COMEDOR-8-PUESTOS comedor	comedor-7-puestos	comedor-7-puestos	43455		10.00	100	
30	217f72a3-1093-40a2-844e-5a182caec498	Comedor para comer	COMEDOR-4-PUESTOS comedor	comedor-4-puestos	comedor-4-puestos	43455		10.00	100	
31	14740374-9764-4d4e-bd6c-01ac3cb26dd9	asdfasdf f asdfm	123c3eqwefrr	cp-fasd asdf asdf fcp ffffasaa-cp	fasdf-asdf-fasd-fff	12000		10.00	120	Adidas
32	33aee0ef-00f5-4aa1-bda7-b95f068039ea	dsdfsda	dydwq22	cp-fasdfsasf-cp	fasdfsadf-asdf	120000		0.00	12	
33	09fa1b34-d5f7-42f8-8b37-4a5d0c8f2070	Comedor para comer	COMEDOR-4-PUESTOS comedor	comedor-4-puestos	com	43455		10.00	100	
34	86e0e121-47cd-49ae-beb7-19a9b711b7d9	GORAR	GORAR_Gris	soajihfrftr8t5Y4	70000		10.00	0.00	-3	Nike
35	a4e85270-45ee-4d8b-8902-e2889604360c	Camisa Nike Pro max	Nike_pro_max_16	soajihfrf343	70000		10.00	0.00	1	Fabudaskjib
36										
37										
38										
39										
40										
41										

Figura 20: Vista de excel ya exportado

4.3.9. Obtener Detalles de un Producto (GET /app/:appId/products/:productId)

Permite consultar toda la información relacionada con un producto específico, incluyendo atributos, variantes, precios, imágenes y relaciones con categorías o marcas.



4.3.10. Reflexión

Este módulo fue clave para entender la importancia de estructurar bien los datos y pensar en las necesidades reales de quienes gestionan los productos. Más allá del código, permitió conectar la lógica del backend con la experiencia del usuario final, haciendo que administrar y actualizar los productos sea un proceso ágil, claro y confiable.

4.4. Seguridad de la Aplicación

4.4.1. Objetivo

Garantizar la protección de los datos sensibles y el acceso controlado a los recursos de la plataforma, mediante la implementación de mecanismos robustos de autenticación, autorización, cifrado y validación de datos.

4.4.2. Funciones Principales

- **Autenticación con JWT:** Se implementó un sistema de autenticación basado en tokens JSON Web Token (JWT), que permite a los usuarios iniciar sesión de forma segura y mantener sesiones autenticadas sin exponer credenciales en cada solicitud.
- **Control de acceso por permisos (NestJS Guards):** La autorización se gestiona mediante *guards* personalizados de NestJS, en combinación con decoradores como `@Permissions('CREATE-PRODUCT')`. Esto permite restringir el acceso a funcionalidades críticas según los permisos asignados a cada usuario.
- **Cifrado de contraseñas (bcrypt):** Las contraseñas de los usuarios se almacenan de forma segura utilizando el algoritmo bcrypt, que aplica hashing con sal aleatoria. Esto garantiza que, incluso ante una posible filtración de la base de datos, las contraseñas no sean legibles.
- **Validación de datos de entrada (DTOs y Pipes):** Para prevenir vulnerabilidades como inyecciones SQL o ataques XSS, se aplican validaciones estrictas mediante DTOs definidos con `class-validator`. Los Pipes de NestJS se encargan de interceptar y rechazar cualquier dato que no cumpla con las reglas establecidas, asegurando la integridad del sistema.
- **Rate Limiting:** Se incorporó una política de limitación de solicitudes para mitigar ataques de fuerza bruta. Esto impide múltiples intentos fallidos de inicio de sesión en un corto periodo de tiempo, bloqueando temporalmente peticiones sospechosas.
- **Configuración de CORS (Cross-Origin Resource Sharing):** Se configuró una política de CORS que restringe las peticiones hacia la API únicamente desde dominios autorizados cuando el acceso proviene de un navegador. Esta medida evita que aplicaciones web externas no confiables puedan interactuar con los recursos protegidos de la plataforma, reforzando la seguridad frente a solicitudes de origen cruzado (*Cross-Origin*) o intentos de acceso no permitidos desde interfaces web.

4.4.3. Endpoint para Realizar Login (POST /auth/login)

El proceso de autenticación inicia con el envío de credenciales válidas al endpoint /auth/login. El sistema valida los datos y genera un token JWT que contiene la información del usuario y su nivel de permisos. Este token se incluye en los encabezados de las solicitudes posteriores, asegurando un acceso continuo y seguro sin reautenticación constante.

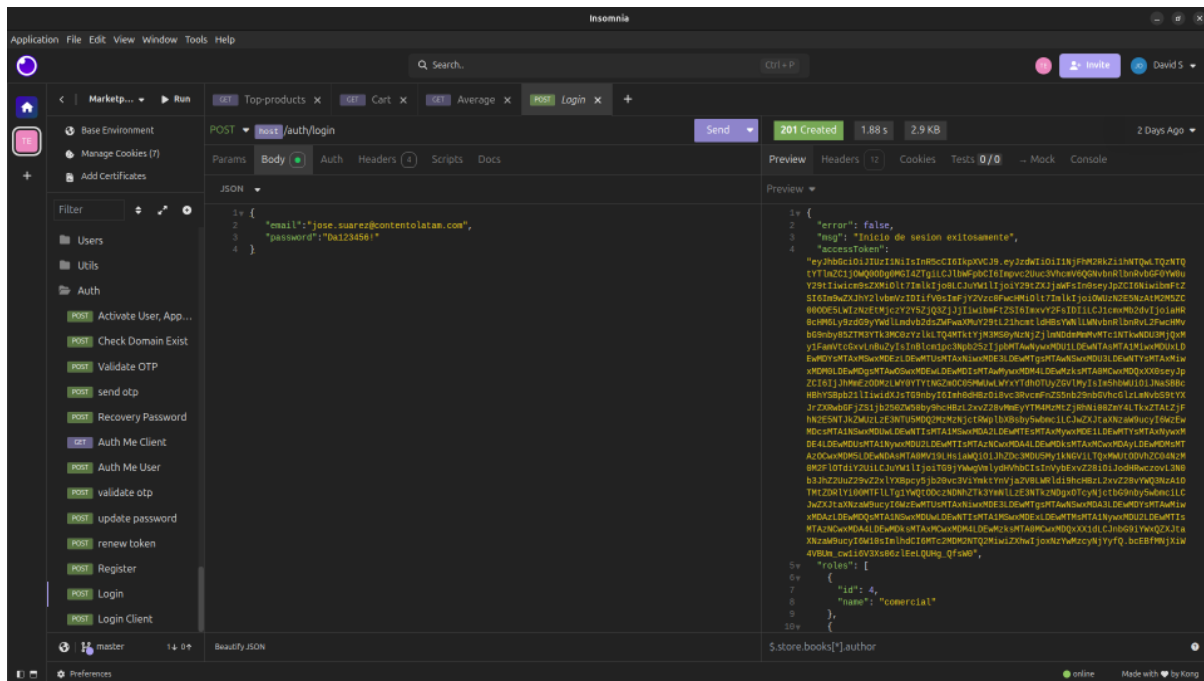


Figura 23: Proceso de inicio de sesión mediante autenticación JWT.

4.4.4. Guard para Proteger Rutas del Servidor (Access Token)

Los guards de NestJS se utilizan para interceptar solicitudes entrantes y verificar la validez del token JWT antes de acceder a un endpoint protegido.

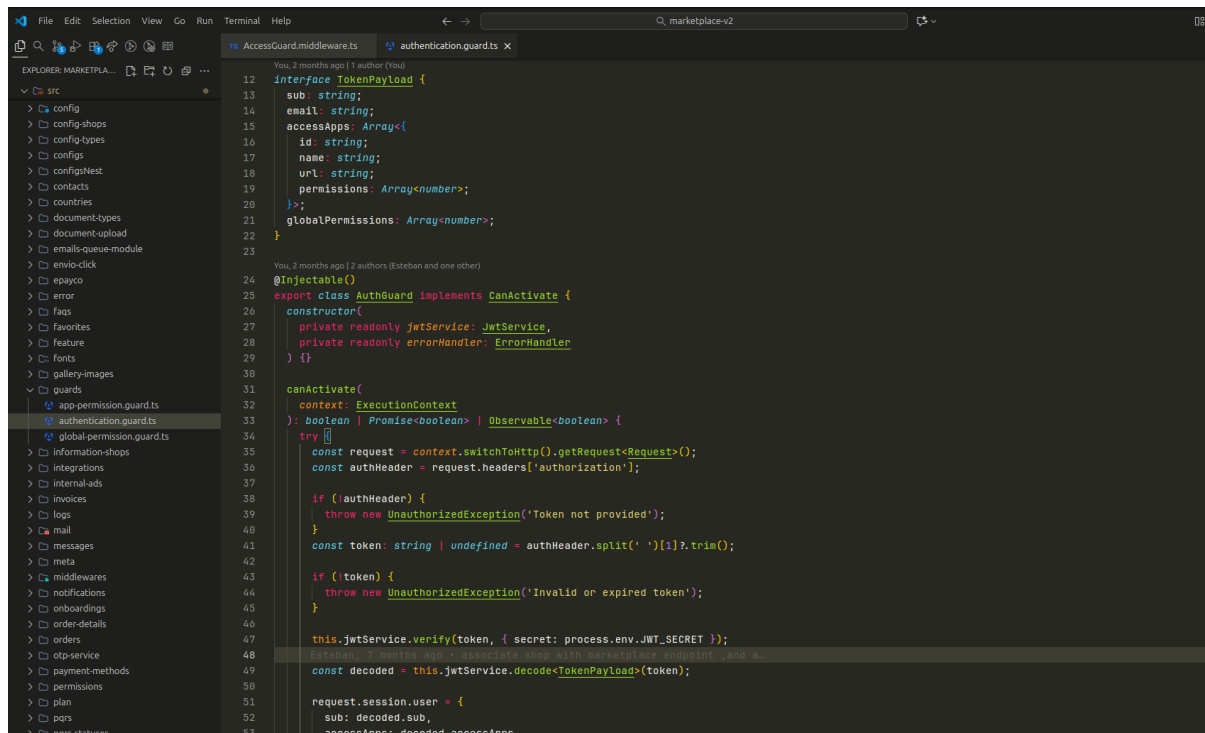


Figura 24: Guard que valida el token de acceso antes de procesar la solicitud.

4.4.5. Guard para validar permisos de un usuario

El `AppPermissionGuard` aplica control de acceso granular por aplicación (Tiendas) y permisos específicos. Toma el *user session* desde la petición y verifica que el usuario tenga acceso a la aplicación objetivo (`appId`) y los permisos requeridos por el endpoint.

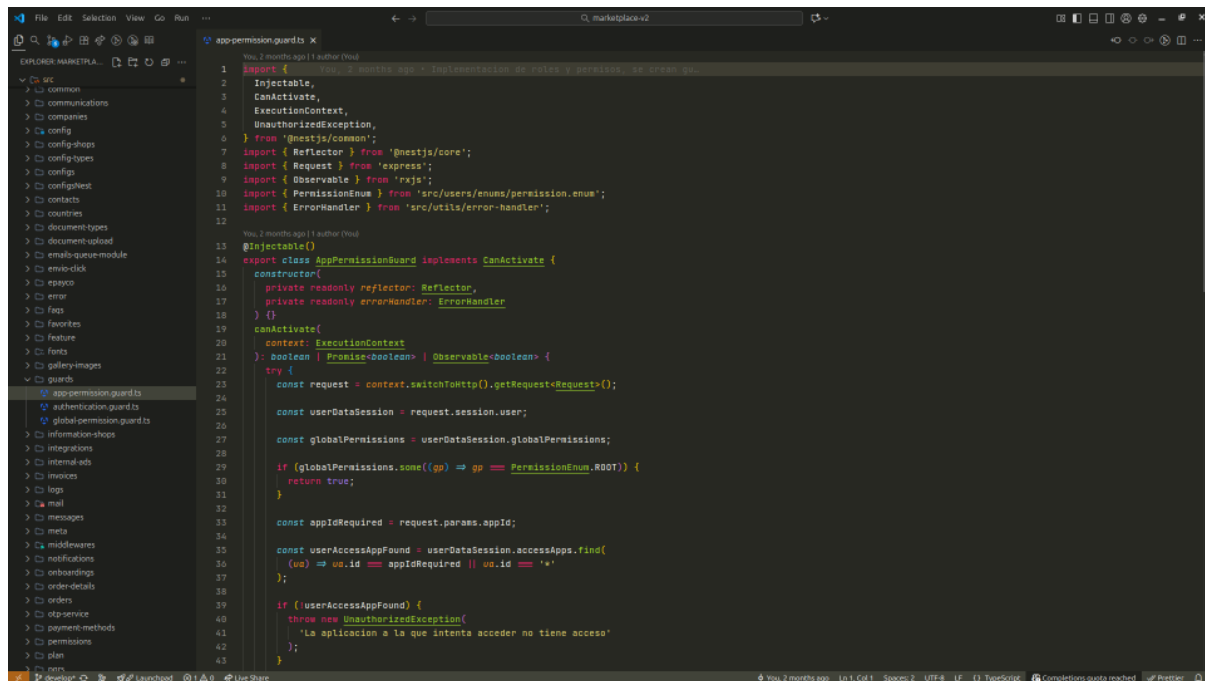
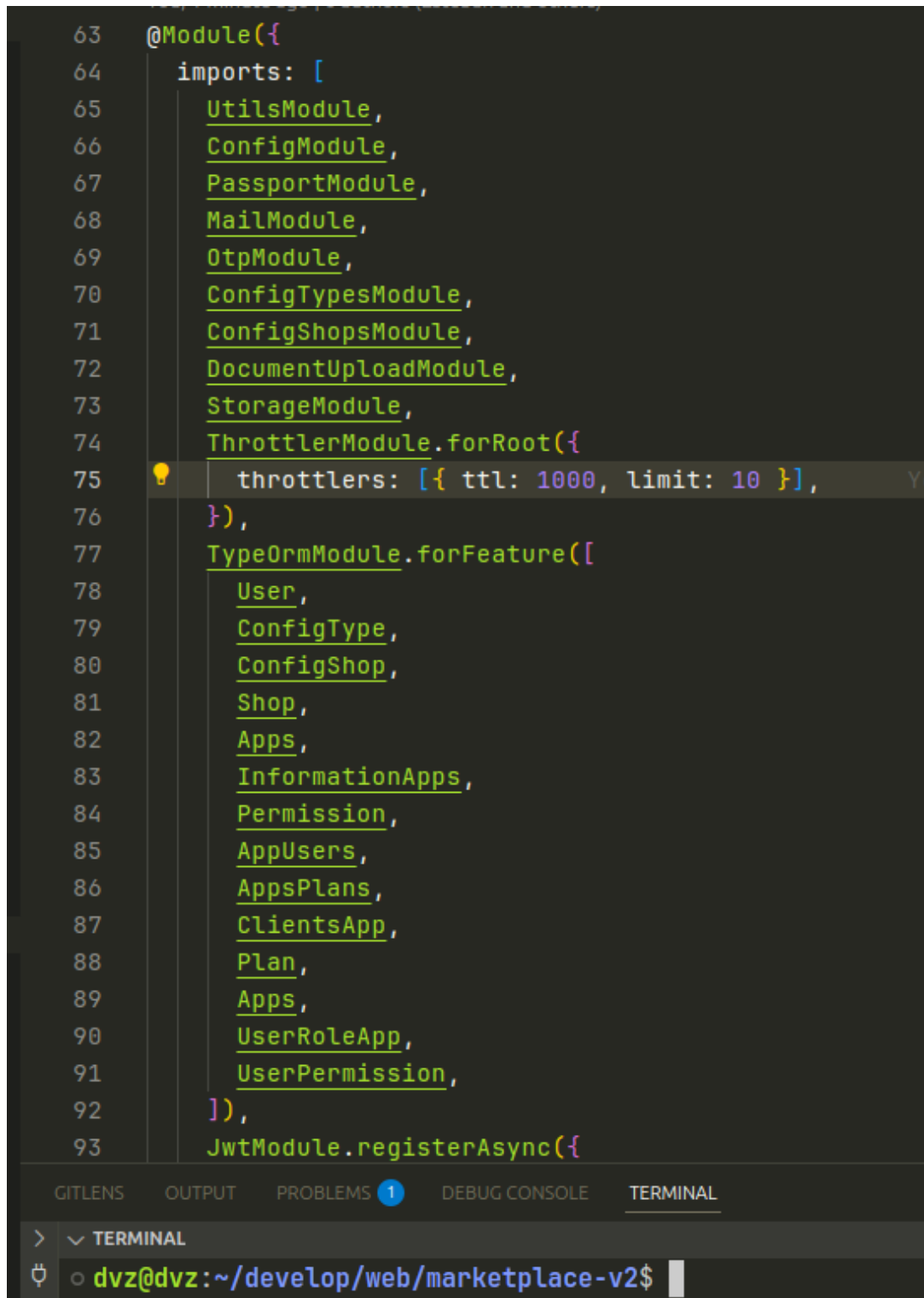


Figura 25: Guard que valida permisos requeridos de un endpoint.

4.4.6. Configuración para limitar peticiones

Esta configuración implementa un control de *rate limiting* por módulo, utilizando una ventana de tiempo en la cual se permite únicamente un número máximo de peticiones. Si un cliente excede dicho límite, las solicitudes adicionales son bloqueadas temporalmente, protegiendo así la aplicación frente a abusos, ataques de denegación de servicio (DoS) y consumo excesivo de recursos.



```
63 @Module({
64   imports: [
65     UtilsModule,
66     ConfigModule,
67     PassportModule,
68     MailModule,
69     OtpModule,
70     ConfigTypesModule,
71     ConfigShopsModule,
72     DocumentUploadModule,
73     StorageModule,
74     ThrottlerModule.forRoot({
75       throttlers: [{ ttl: 1000, limit: 10 }],
76     }),
77     TypeOrmModule.forFeature([
78       User,
79       ConfigType,
80       ConfigShop,
81       Shop,
82       Apps,
83       InformationApps,
84       Permission,
85       AppUsers,
86       AppsPlans,
87       ClientsApp,
88       Plan,
89       Apps,
90       UserRoleApp,
91       UserPermission,
92     ]),
93     JwtModule.registerAsync({
```

GIT LENS OUTPUT PROBLEMS 1 DEBUG CONSOLE TERMINAL

> ▾ TERMINAL

dvz@dvz:~/develop/web/marketplace-v2\$

Figura 26: Configuración del módulo con limitación de peticiones por ventana de tiempo.

4.4.7. Validaciones de DTOs con Class-Validator

Las validaciones se aplican en el nivel de DTOs utilizando el paquete `class-validator`, asegurando que los datos recibidos cumplan con las reglas definidas

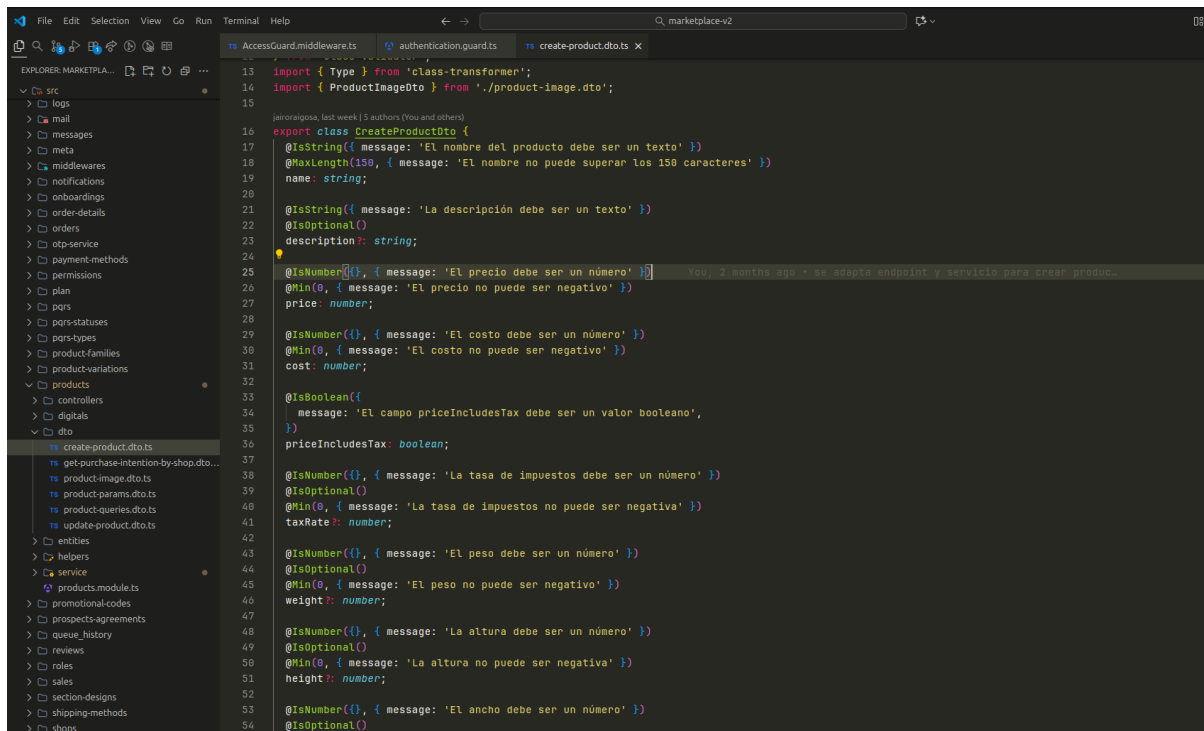


Figura 27: DTO con validaciones de campos obligatorios para creación de productos.

4.4.8. Reflexión

La implementación de estos aspectos fue esencial para garantizar la confianza en la plataforma. Más allá de aplicar técnicas como autenticación, cifrado o validaciones, este proceso permitió fortalecer la arquitectura del sistema y entender cómo cada medida contribuye a proteger tanto a los usuarios como a la empresa.

4.5. Módulo de Estadísticas

4.5.1. Objetivo

El módulo de estadísticas está diseñado para centralizar, analizar y exponer información estratégica sobre el comportamiento general de la plataforma. Su propósito es ofrecer al administrador una visión integral del rendimiento comercial y de la interacción de los usuarios, facilitando la toma de decisiones basadas en datos reales. A través del procesamiento de métricas clave —como ventas, visitas, productos más vistos, métodos de pago y comportamiento del cliente— el módulo transforma la información dispersa en indicadores claros, útiles y accionables.

4.5.2. Funcionalidades Principales

- **Análisis de ventas:** Permite obtener métricas agregadas, promedios y conteos de ventas por período, con posibilidad de aplicar filtros por tienda, fecha u otros parámetros relevantes.
- **Estadísticas de comportamiento de usuarios:** Presenta información sobre visitas, conversiones, pedidos recientes y patrones de compra.
- **Rendimiento por producto y categoría:** Identifica los productos y categorías con mayor actividad o volumen de ventas.
- **Análisis de métodos de pago:** Proporciona estadísticas sobre la frecuencia y preferencia de los distintos medios de pago utilizados por los usuarios.
- **Segmentación geográfica:** Muestra la distribución de clientes y ventas por ubicación geográfica.
- **Valor promedio de tickets:** Calcula el gasto promedio por transacción.

4.5.3. Obtener promedio de ventas (GET /statistics/average/:appId)

Esta operación permite calcular el promedio de ventas en un rango de fechas determinado.

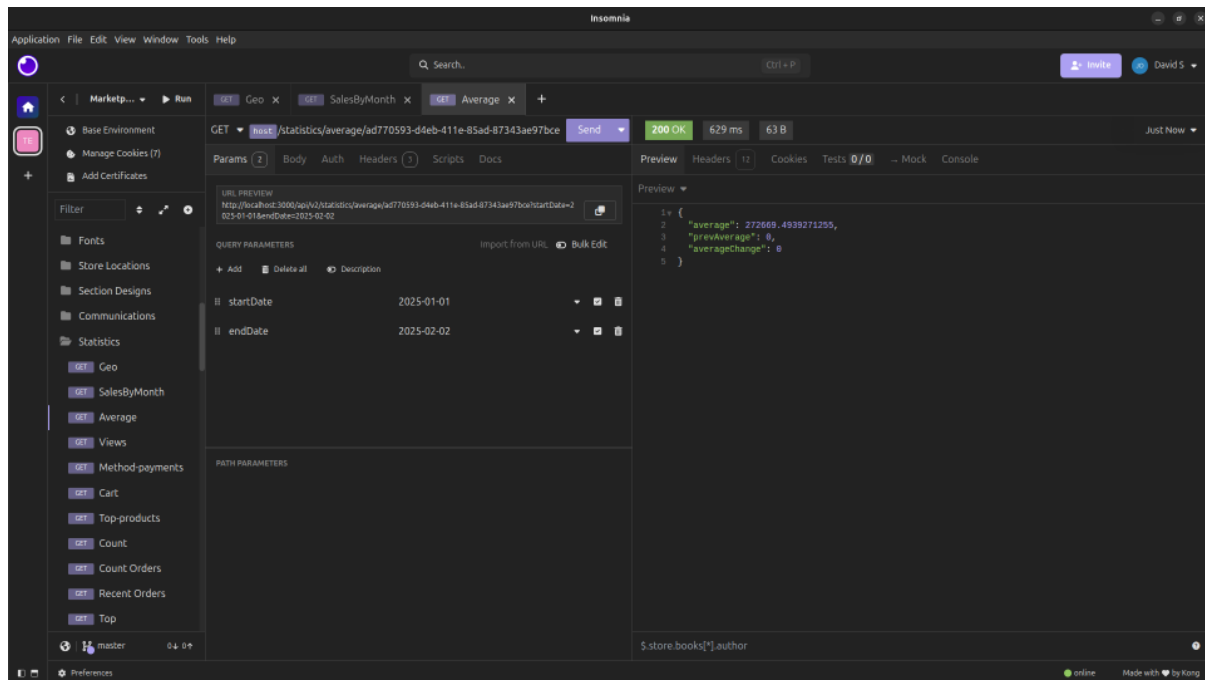


Figura 28: Endpoint para obtener el promedio de ventas de una tienda.

4.5.4. Obtener segmentación geográfica de clientes (GET /statistics/geo/:appId)

Este endpoint analiza la localización de los clientes registrados o compradores activos, generando mapas de calor o diagramas de densidad. Facilita la identificación de zonas de mayor demanda y regiones potenciales para expansión comercial.

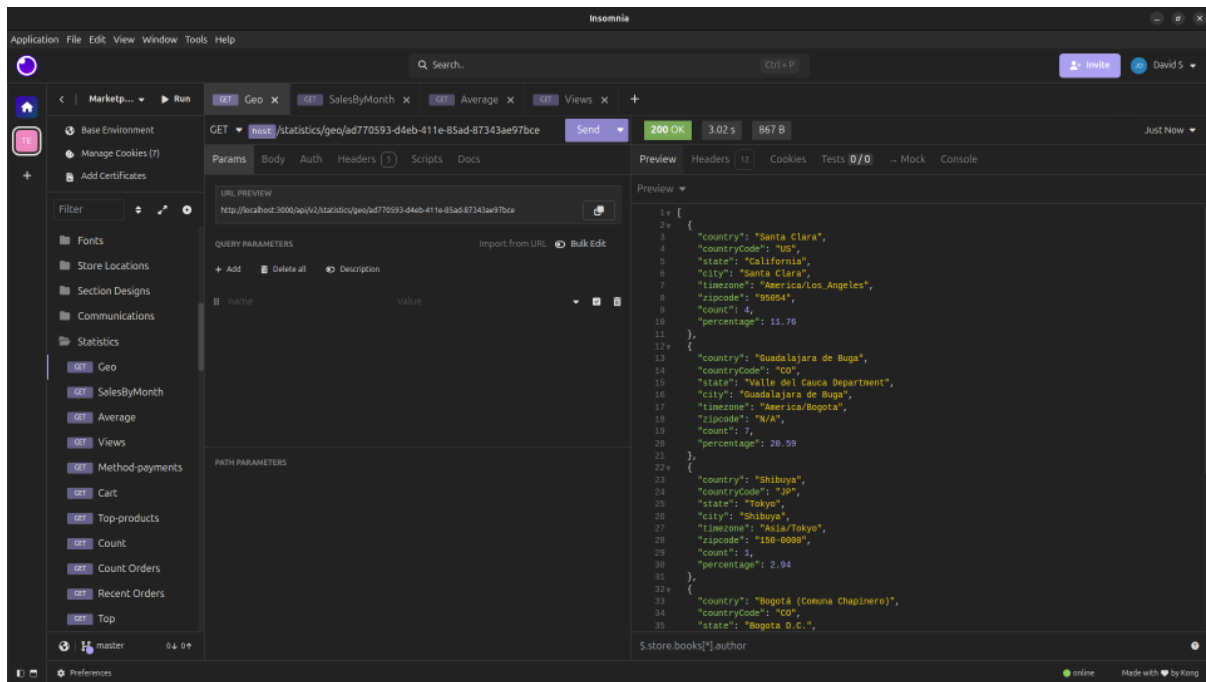


Figura 29: Endpoint para obtener la distribución geográfica de los clientes de una tienda.

4.5.5. Obtener estadísticas de visitas (GET /statistics/views/:appId)

Permite monitorear el tráfico general de la plataforma, incluyendo número de visitas únicas

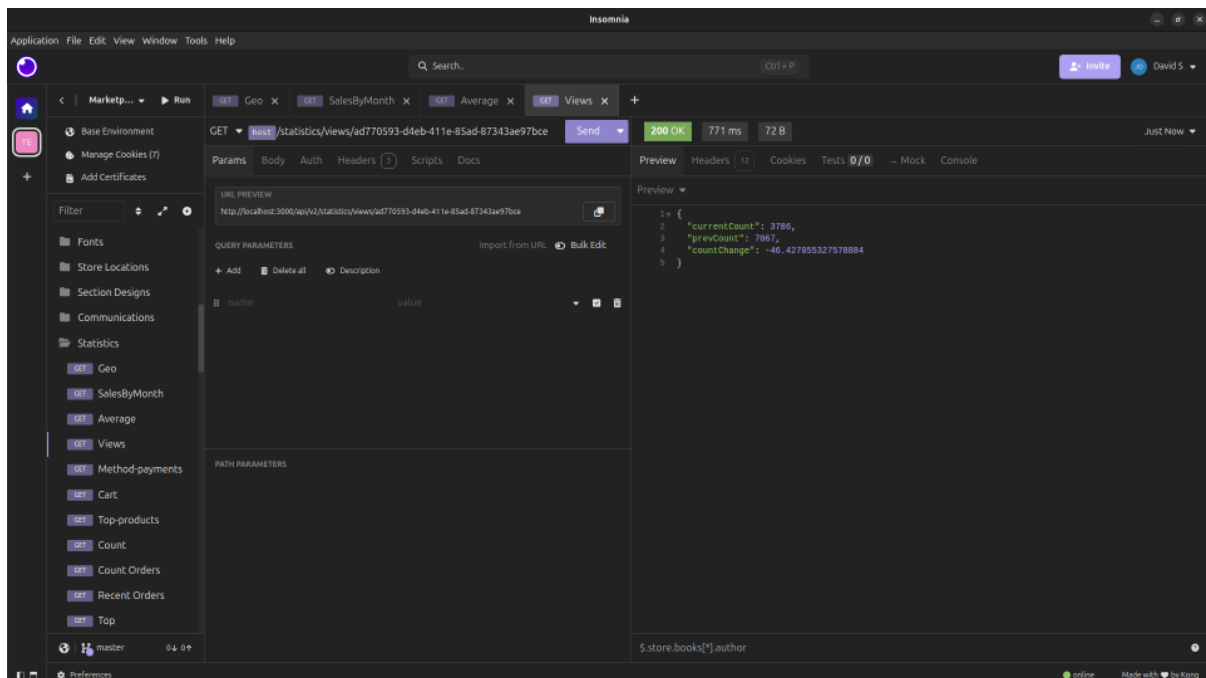


Figura 30: Endpoint para obtener estadísticas de visitas de una tienda.

4.5.6. Obtener órdenes recientes (GET /statistics/recent/orders/:appId)

Proporciona un resumen cronológico de las últimas órdenes generadas en la tienda

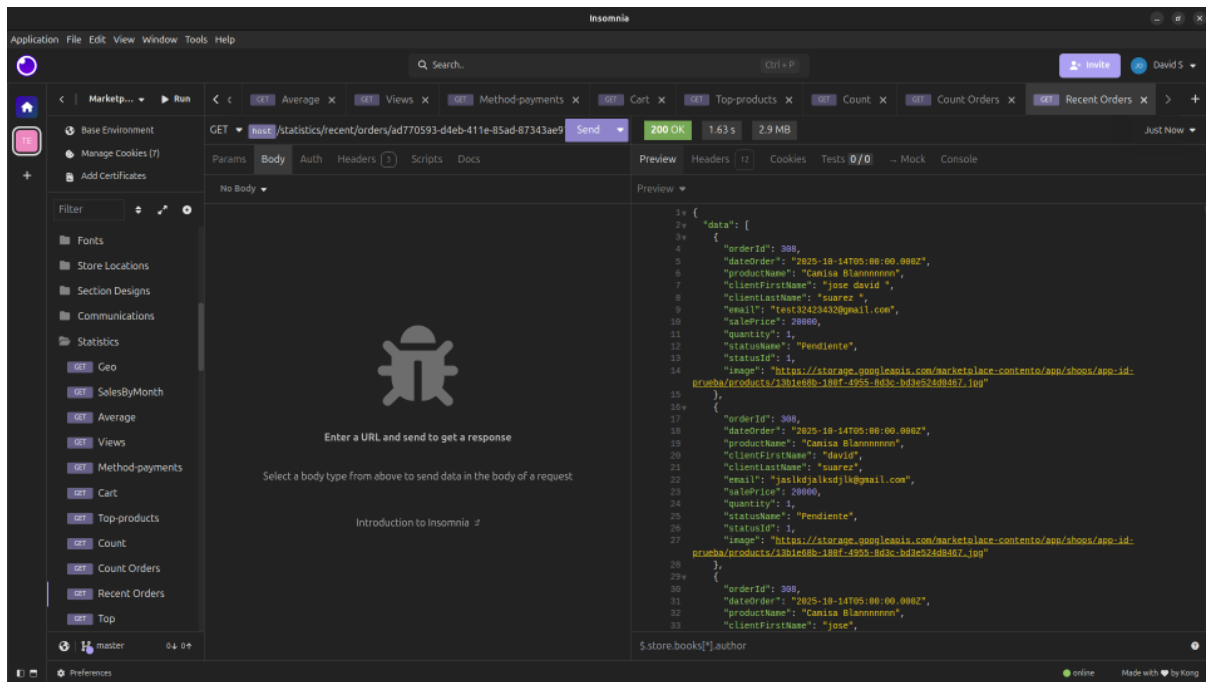


Figura 31: Endpoint para consultar las órdenes más recientes de una tienda.

4.5.7. Obtener métodos de pago utilizados (GET /statistics/method-payments/:appId)

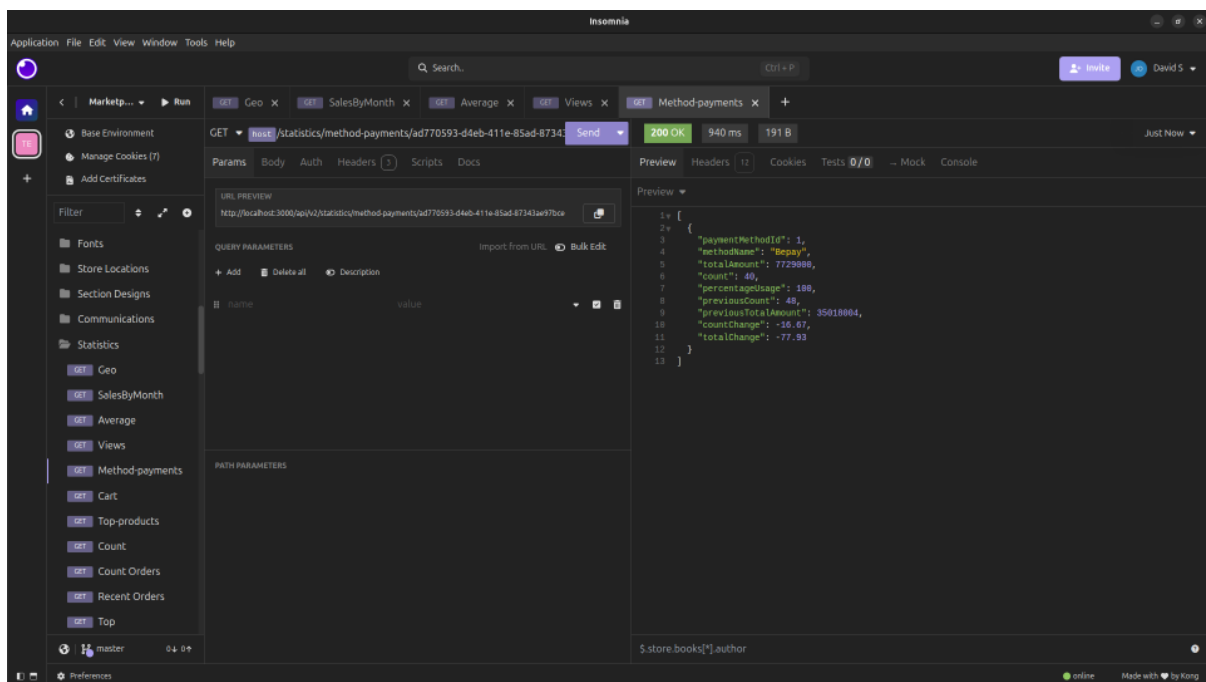


Figura 32: Endpoint para obtener estadísticas sobre el uso de métodos de pago.

4.5.8. Obtener categorías más vendidas (GET /statistics/top-categories/:appId)

Este servicio genera un ranking de las categorías con mayor participación en ventas.

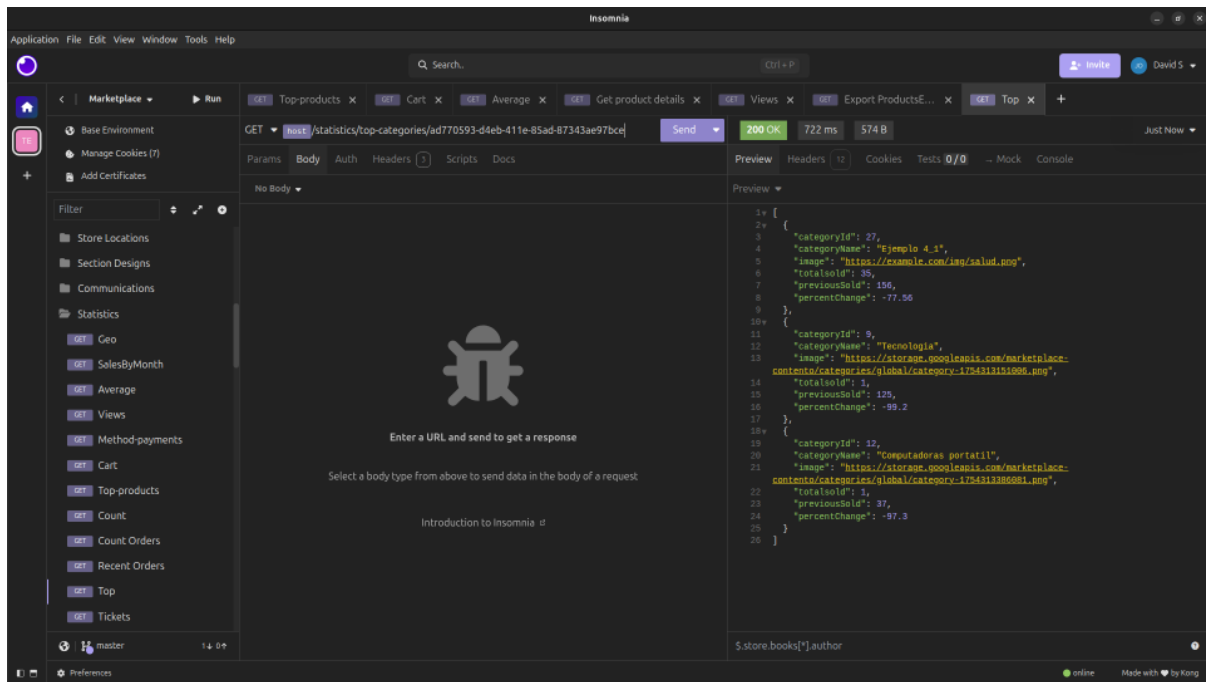


Figura 33: Endpoint para obtener el top de categorías con mayor número de ventas.

4.5.9. Obtener valor promedio de tickets (GET /statistics/tickets/:appId)

Calcula el importe medio gastado por cada cliente en una transacción.

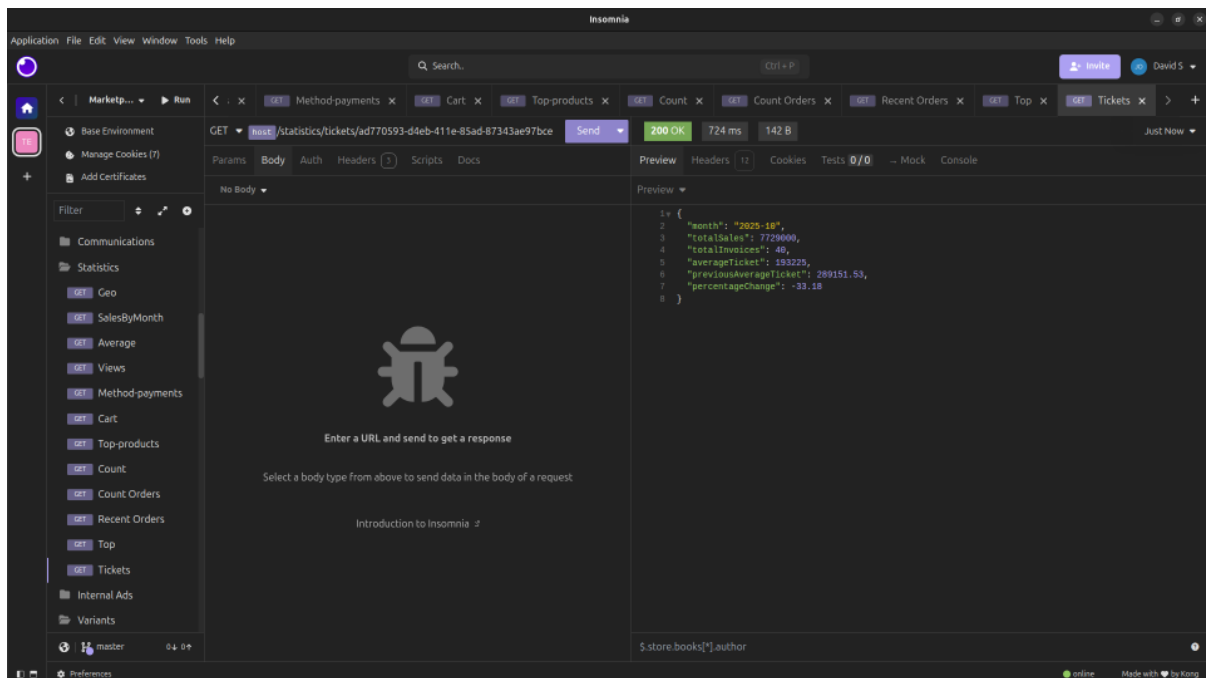


Figura 34: Endpoint para calcular el valor promedio por transacción (ticket promedio).

4.5.10. Obtener productos más vendidos (GET /statistics/top-products/:appId)

Genera un listado ordenado de los productos con mayor número de ventas.

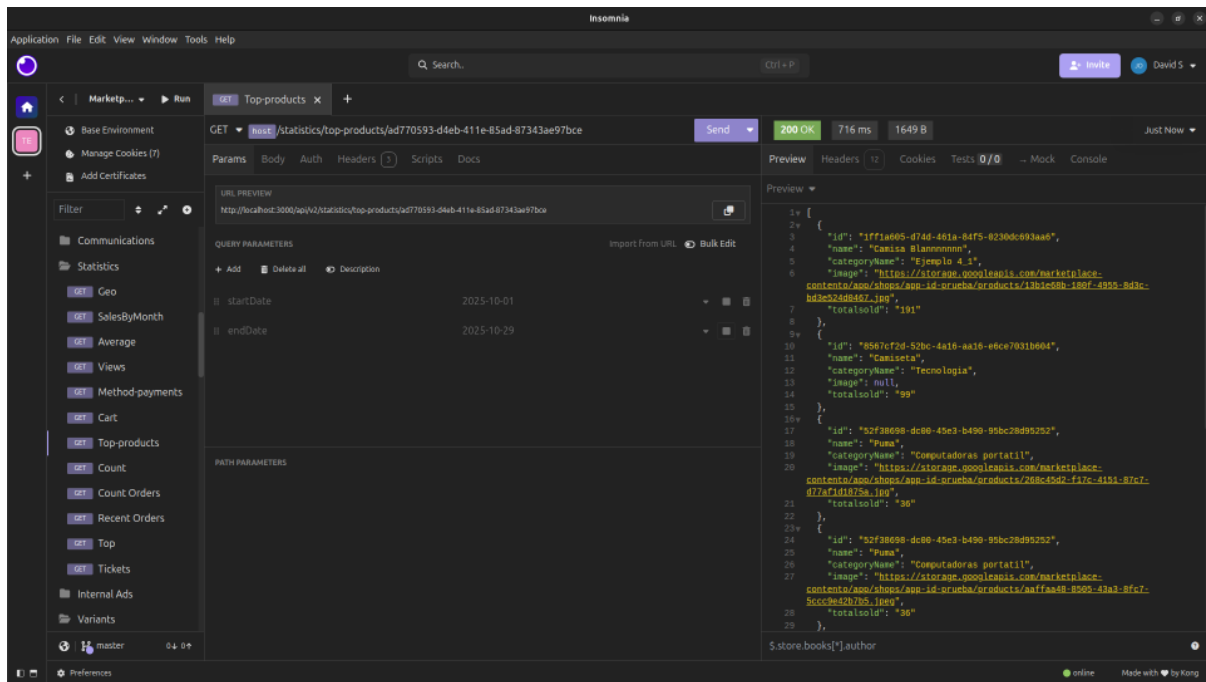


Figura 35: Endpoint para consultar los productos más vendidos de una tienda.

4.5.11. Reflexion

El módulo de estadísticas constituye una herramienta fundamental para la gestión estratégica de la plataforma. Su capacidad para convertir datos operativos en información analítica aporta claridad, eficiencia y control al proceso de toma de decisiones. Gracias a sus servicios y visualizaciones, es posible monitorear el desempeño comercial en tiempo real y anticipar tendencias que impulsen la mejora continua del negocio.

4.6. Exposición de API para integraciones (Bepay webhook)

4.6.1. Objetivo

Esta funcionalidad tiene como propósito facilitar la interoperabilidad entre la plataforma Contenido y servicios externos mediante la exposición de una API basada en webhook. En particular, permite que terceros —como la pasarela de pagos BePay— puedan notificar a la plataforma sobre eventos relevantes, como el estado de una transacción, y activar flujos internos de manera automatizada. Esto incluye acciones como la validación de pagos, el ajuste de inventario y el envío de notificaciones a los clientes, mejorando así la eficiencia operativa.

4.6.2. Procesamiento de Webhook de Pagos (pasarela BePay)

Este servicio recibe y procesa notificaciones enviadas por la pasarela de pagos BePay, las cuales informan sobre el estado de las transacciones realizadas por los usuarios. El sistema interpreta el contenido de la notificación, busca la factura asociada, valida su estado y, si es válido, actualiza la información correspondiente, permitiendo responder en tiempo real ante pagos aprobados, rechazados, pendientes o fallidos.

4.6.3. Funciones clave del webhook

- **Recepción de datos:** Interpreta el payload enviado por BePay, incluyendo ID de transacción, método de pago, estado y metadatos.
- **Validación de factura:** Busca la factura relacionada con la transacción, verificando que exista, no esté eliminada y no haya sido procesada anteriormente.
- **Actualización de estado:** Cambia el estado de la factura según el resultado de la transacción (aprobada, rechazada, pendiente o fallida).
- **Gestión de inventario:** Actualiza la cantidad de productos vendidos según el detalle de la orden, y lanza notificaciones si se detectan niveles bajos de inventario.
- **Notificación al cliente:** Envía un correo automático al comprador informando el estado de su transacción y datos relevantes.
- **Registro de actividad:** Genera un log detallado del evento recibido, permitiendo trazabilidad y auditoría de la operación.

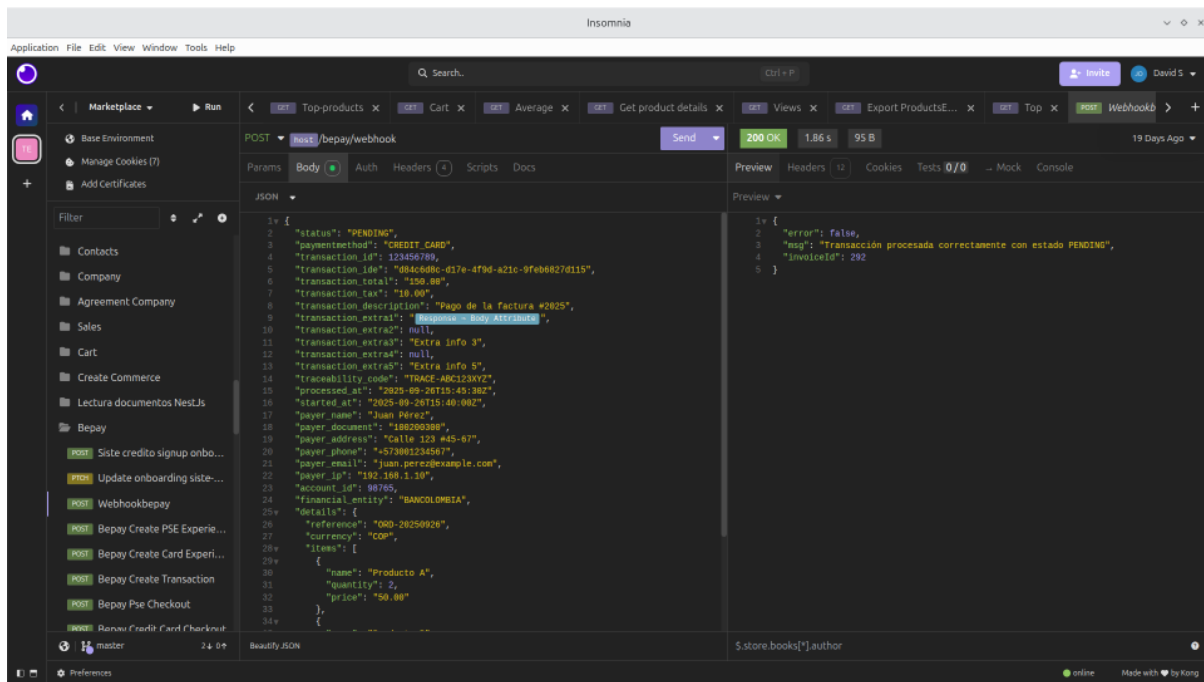


Figura 36: Se procesa transacción y se pasa a estado pendiente

4.6.4. Reflexión

Este módulo representa un componente esencial en la integración de Contenido con el ecosistema digital externo. Su capacidad de reaccionar ante eventos en tiempo real, como los generados por pasarelas de pago, Gracias a este diseño, se logra automatizar procesos críticos como la confirmación de pagos, la actualización de inventario y la comunicación con los clientes, todo ello sin intervención manual, mejorando significativamente la experiencia del usuario final y la eficiencia del sistema

4.7. Pruebas del Sistema

4.7.1. Objetivo de las Pruebas

El propósito de esta etapa fue garantizar que los módulos desarrollados funcionaran de manera correcta, estable y coherente con los requerimientos definidos para la plataforma. Las pruebas buscaron comprobar que la lógica de negocio respondiera adecuadamente ante distintos escenarios, validando tanto la estructura de los datos como el comportamiento general de los servicios implementados.

4.7.2. Metodología de Pruebas

Se aplicaron dos tipos de pruebas complementarias: **unitarias** y **end-to-end (E2E)**.

Las pruebas unitarias se centraron en evaluar de forma aislada los servicios, controladores y repositorios, asegurando que cada componente ejecutara su función correctamente. Para ello, se utilizó el framework **Jest**, que permite simular dependencias mediante mocks y verificar el flujo interno de ejecución sin necesidad de conectarse a la base de datos real.

Por otro lado, las pruebas end-to-end (E2E) permitieron validar el funcionamiento integral del sistema, comprobando la interacción completa entre módulos, controladores y la base de datos PostgreSQL. Estas pruebas se realizaron utilizando **Supertest** junto con el entorno de **Jest**, enviando peticiones HTTP reales a la aplicación NestJS y verificando las respuestas, los códigos de estado y la estructura de los datos retornados.

4.7.3. Cobertura y Resultados

Las pruebas se enfocaron principalmente en los siguientes módulos:

- **Gestión de productos:** Verificación de la creación, actualización, eliminación lógica e importación masiva desde archivos Excel.
- **Estadísticas:** Validación de consultas agregadas sobre ventas, visitas, productos más vendidos y comportamiento de los usuarios.
- **Seguridad:** Pruebas sobre el flujo de autenticación con JWT, validación de permisos por rol y protección de endpoints.

4.7.4. Evidencia de Pruebas

La figura 37 muestra un ejemplo del reporte generado por Jest, en el cual se evidencian los resultados de las pruebas ejecutadas, los casos exitosos y la cobertura alcanzada por cada módulo del sistema.

File	ExecTime	Status	Action
Rest/auth/auth.guard.spec.ts	00:14.199	All Passed: 4	
AuthGuard > should allow access with valid token	00:00.056	✓ Passed	
AuthGuard > should throw if token is missing	00:00.009	✓ Passed	
AuthGuard > should throw if token is malformed	00:00.004	✓ Passed	
AuthGuard > should throw if token verification fails	00:00.007	✓ Passed	
Rest/auth/global-permission-guard.spec.ts	00:01.073	All Passed: 5	
GlobalPermissionGuard > retorna false si no hay metadata de permisos en el handler	00:00.009	✓ Passed	
GlobalPermissionGuard > permite el acceso si el usuario tiene alguno de los permisos requeridos	00:00.002	✓ Passed	
GlobalPermissionGuard > lanza UnauthorizedException si el usuario no tiene los permisos requeridos	00:00.031	✓ Passed	
GlobalPermissionGuard > OR lógico: varios permisos requeridos, usuario con uno de ellos	00:00.003	✓ Passed	
Rest/statistics/view-client-geographic.service.spec.ts	00:15.965	All Passed: 3	
GeographicClientServices > filtra IPs locales/privadas, agrega por país/ciudad y calcula porcentaje	00:00.044	✓ Passed	
GeographicClientServices > usa filtro Between cuando se proporciona la fecha (YYYY-MM)	00:00.013	✓ Passed	
GeographicClientServices > cuando metadata falta o es inválida, esas vistas no se integran en el mapa	00:00.012	✓ Passed	
Rest/statistics/buy-visit.service.spec.ts	00:10.550	All Passed: 2	
GetBuyAndVisitService > calcula vistas y ventas para un año completo (12 meses) con variaciones porcentuales	00:00.051	✓ Passed	
GetBuyAndVisitService > respeta el año actual generando meses hasta el mes corriente	00:00.011	✓ Passed	

Figura 38: Resultado de pruebas generadas con jest

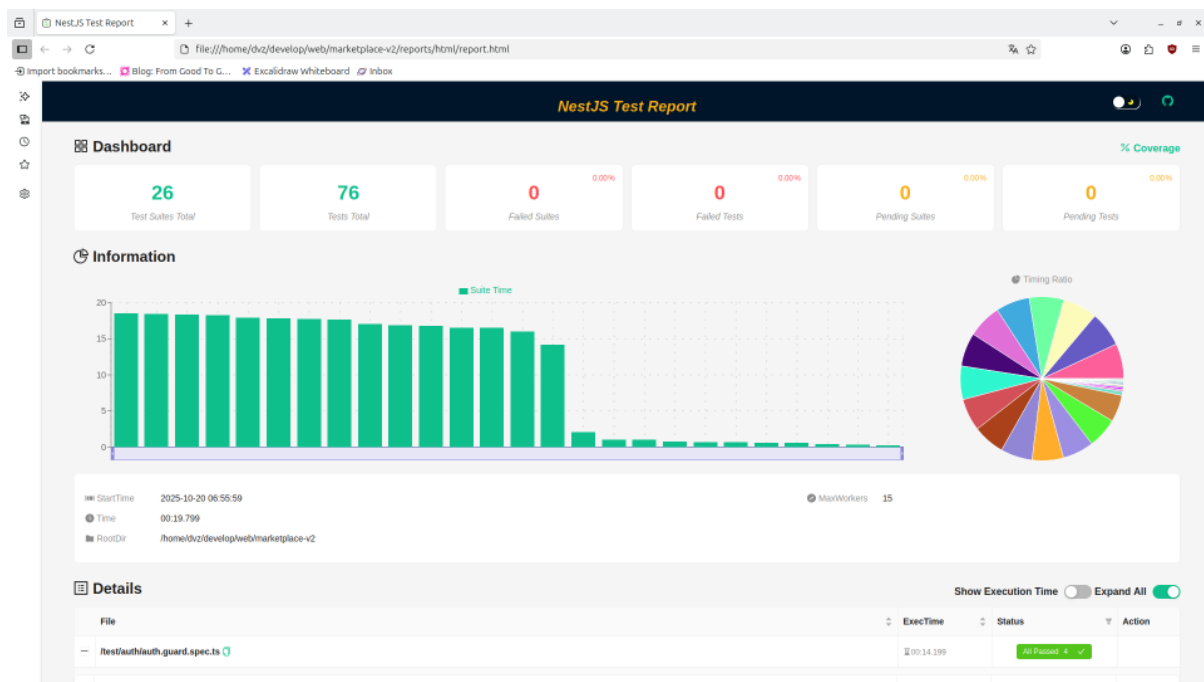


Figura 37: Reporte de resultados de pruebas generadas con Jest

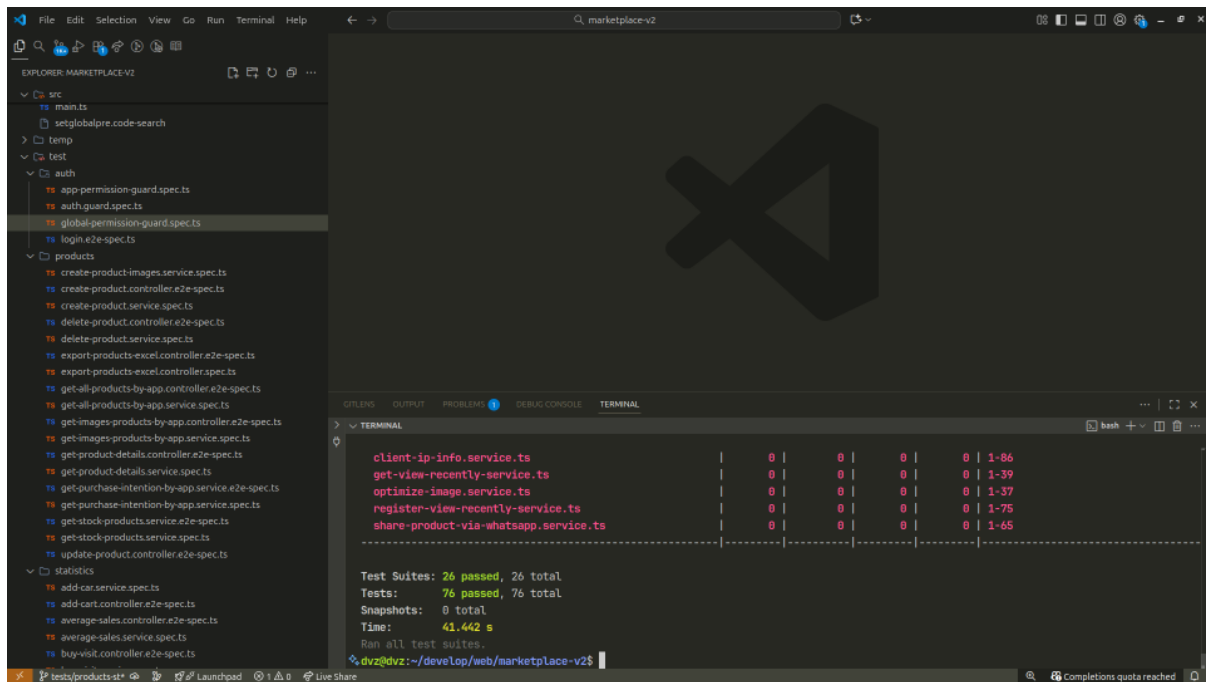


Figura 39: Resultado de pruebas generadas con jest

4.7.5. Reflexion

La implementación de pruebas unitarias y end-to-end permitió verificar que cada componente del sistema cumpliera con su propósito, tanto de forma individual como integrada. Gracias a esta estrategia, fue posible detectar errores tempranamente, validar los flujos principales de la aplicación y asegurar que futuras modificaciones no afecten el funcionamiento general.

5. Conclusiones

El desarrollo de este proyecto permitió cumplir con el objetivo de modernizar el backend de la plataforma Contenido. Tras finalizar las fases de diseño, implementación y pruebas, se obtuvieron las siguientes conclusiones técnicas y operativas:

5.1. Sobre la Arquitectura y Escalabilidad

La decisión de utilizar una **Arquitectura Monolítica Modular** en NestJS demostró ser la más adecuada para el estado actual del proyecto. Esta estructura permitió organizar el código por dominios de negocio (como Productos y Ventas) sin introducir la complejidad de gestión que requieren los microservicios. Esto facilitó el trabajo de desarrollo y dejó un código ordenado que el equipo podrá mantener y escalar fácilmente en el futuro.

5.2. Sobre la Gestión de Productos

El desarrollo del módulo de gestión de productos permitió establecer un control estricto sobre la información del catálogo, asegurando que todos los datos ingresados cumplan con las reglas de negocio antes de ser procesados. Esta estandarización, sumada a la implementación de la eliminación lógica (*soft delete*) y la carga masiva, garantizó la integridad del inventario y la preservación del historial de ventas, solucionando los problemas de inconsistencia presentes en la versión anterior.

5.3. Sobre la Seguridad y Control de Acceso

La seguridad de la plataforma mejoró significativamente al implementar un control estricto en los endpoints. Mediante el uso de **Guards personalizados**, se centralizó la validación de los tokens JWT y los roles de usuario (RBAC), asegurando que ninguna operación crítica pueda ejecutarse sin los permisos adecuados.

5.4. Sobre la Capacidad Analítica

El módulo de estadísticas comprobó que el backend puede procesar y transformar datos transaccionales en información útil sin degradar el rendimiento del sistema. Gracias a la optimización de las consultas en PostgreSQL, fue posible generar reportes complejos, como la segmentación geográfica de clientes, lo cual aporta valor directo para la toma de decisiones del negocio.

5.5. Conclusión General

El proyecto cumple con los requerimientos planteados, entregando una plataforma que resuelve las limitaciones técnicas de la versión anterior. La migración a TypeScript y al ecosistema NestJS ha proporcionado al equipo de desarrollo un entorno de trabajo estandarizado y tipado, lo que reduce la deuda técnica y deja a la empresa Contenido preparada para soportar el crecimiento de sus operaciones en el comercio electrónico.

Referencias Bibliográficas

- [1] Atlassian. “Qué es Scrum?” Visitado 22 de dic. de 2024. dirección: <https://www.atlassian.com/es/agile/scrum>.
- [2] Microsoft. “TypeScript: JavaScript with syntax for types,” visitado 22 de dic. de 2024. dirección: <https://www.typescriptlang.org/>.
- [3] Meta Platforms, Inc. “React – Una biblioteca de JavaScript para construir interfaces de usuario,” visitado 18 de dic. de 2024. dirección: <https://react.dev/>.
- [4] K. Myśliwiec. “NestJS – A progressive Node.js framework,” visitado 18 de dic. de 2024. dirección: <https://nestjs.com>.
- [5] Cámara Colombiana de Comercio Electrónico. “Informe del comercio electrónico en 2022 y perspectivas 2023,” visitado 20 de oct. de 2025. dirección: <https://www.ccce.org.co/>.
- [6] T. Buitrago y J. G. Restrepo. “Origen y evolución del E-commerce en el mundo,” Pymas, visitado 27 de nov. de 2024. dirección: <https://www.pymas.com.co/ideas-para-crecer/mundo-pyme/infografia-evolucion-comercio-electronico>.
- [7] PowerData. “Migración de datos: definición, desafíos y mejores prácticas,” visitado 18 de dic. de 2024. dirección: <https://www.powerdata.es/migracion-de-datos>.
- [8] PostgreSQL Global Development Group. “PostgreSQL: The world’s most advanced open source relational database,” visitado 18 de dic. de 2024. dirección: <https://www.postgresql.org>.
- [9] OpenJS Foundation. “Node.js Documentation,” visitado 22 de dic. de 2024. dirección: <https://nodejs.org/en/docs/>.
- [10] L. Torvalds. “Git - Sistema de Control de Versiones,” visitado 22 de dic. de 2024. dirección: <https://git-scm.com>.
- [11] GitHub, Inc. “GitHub - Plataforma de Desarrollo Colaborativo,” visitado 22 de dic. de 2024. dirección: <https://github.com>.
- [12] Atlassian. “¿Qué es Jira?” Visitado 22 de dic. de 2024. dirección: <https://www.atlassian.com/es/software/jira>.
- [13] Microsoft. “Microsoft Teams,” visitado 22 de dic. de 2024. dirección: <https://www.microsoft.com/es-co/microsoft-teams/group-chat-software>.
- [14] MUI. “Material UI: The React component library you always wanted,” visitado 18 de dic. de 2024. dirección: <https://mui.com/>.
- [15] TypeORM. “TypeORM: Data Mapper and Active Record ORM for Node.js.” Referencia clave para la capa de persistencia, visitado 20 de oct. de 2025. dirección: <https://typeorm.io/>.
- [16] Auth0. “Introduction to JSON Web Tokens (JWT).” Estándar utilizado para la autenticación, visitado 20 de oct. de 2025. dirección: <https://jwt.io/introduction>.
- [17] OWASP Foundation. “OWASP API Security Top 10.” Base para las prácticas de seguridad implementadas, visitado 20 de oct. de 2025. dirección: <https://owasp.org/API-Security/>.
- [18] SmartBear Software. “Swagger OpenAPI Specification.” Estándar para la documentación de la API, visitado 20 de oct. de 2025. dirección: <https://swagger.io/specification/>.
- [19] K. Myśliwiec. “NestJS Documentation - Guards.” Documentación oficial sobre la implementación de la capa de seguridad, visitado 20 de oct. de 2025. dirección: <https://docs.nestjs.com/guards>.

- [20] D. F. Ferraiolo, D. R. Kuhn y R. Chandramouli, “Role-Based Access Control (RBAC): Features and Motivations,” National Institute of Standards y Technology (NIST), inf. téc., 1995, Estándar de la industria para el control de acceso. visitado 20 de oct. de 2025. dirección: <https://csrc.nist.gov/pubs/proceedings/1995/rolebased-access-control-features-and-motivations-paper>.
- [21] M. B. Jones, J. Bradley y N. Sakimura, “JSON Web Token (JWT) - RFC 7519,” Internet Engineering Task Force (IETF), inf. téc., mayo de 2015, Especificación técnica estándar para tokens de seguridad. visitado 20 de oct. de 2025. dirección: <https://datatracker.ietf.org/doc/html/rfc7519>.