



Trabajo de grado

DESARROLLO MÓDULOS DE AUTENTICACIÓN, GESTIÓN DE MARCAS,
GESTIÓN DE DATOS COMUNES Y EXPERIENCIA DE PAGO PARA LA
MIGRACIÓN A LA NUEVA VERSIÓN DE LA PLATAFORMA CONTENTO
(Modalidad pasantía)

Oscar Fernando Rivera Pardo

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

Universidad del Valle
Sede Tuluá

9 de febrero de 2026

DESARROLLO MÓDULOS DE AUTENTICACIÓN, GESTIÓN DE MARCAS,
GESTIÓN DE DATOS COMUNES Y EXPERIENCIA DE PAGO PARA LA
MIGRACIÓN A LA NUEVA VERSIÓN DE LA PLATAFORMA CONTENTO
(Modalidad pasantía)

Oscar Fernando Rivera Pardo
Codigo : 202067730
oscar.pardo@correounivalle.edu.co

Director:
José Luis Unás Gómez

Supervisor: Christian Alexis Castro Correa

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

Universidad del Valle
Sede Tuluá

9 de febrero de 2026

Índice

1. Introducción	8
1.1. Descripción de la empresa	9
1.1.1. Descripción general	9
1.1.2. Misión	9
1.1.3. Visión	9
1.1.4. Valores organizacionales	9
1.1.5. Razón social	9
1.1.6. Tipo de empresa	9
1.1.7. Número de cargos	9
1.1.8. Número de empleados	9
1.1.9. Organigrama de la empresa	10
1.1.10. Mapa de procesos de la empresa	11
1.1.11. Descripción de funciones a desarrollar el estudiante en la empresa.	13
1.1.12. Director de proyecto de grado.	14
1.1.13. Supervisor y jefe.	14
1.2. Planteamiento del problema	15
1.2.1. Descripción del problema	15
1.2.2. Árbol de problemas	16
1.2.3. Formulación del problema	17
1.3. Alcance del proyecto	18
1.3.1. Objetivo general	18
1.3.2. Objetivos específicos	18
1.3.3. Objetivos esperados	19
1.4. Justificación	20
2. Marco referencial	21
2.1. Marco conceptual	21
2.1.1. JavaScript (JS)	21
2.1.2. Scrum	21
2.1.3. TypeScript (TS)	21
2.1.4. React	21
2.1.5. NestJS	22
2.1.6. Node.js	22
2.1.7. Git	22
2.1.8. Github	22
2.1.9. Jira	22
2.1.10. Microsoft Teams	22
2.1.11. Material-UI (MUI)	22
3. Metodología	23
4. Ingeniería de desarrollo	24
4.1. Módulo de Autenticación	24
4.2. Módulo de gestión de marcas	28
4.2.1. Objetivo del Módulo	28
4.3. Módulo de gestión de datos comunes	33

4.3.1. Objetivo del Módulo	33
4.4. Módulo de pasarela de pago (BePay)	45
4.4.1. Objetivo del Módulo	45
4.5. Reflexión Final	52
Referencias Bibliográficas	53

Índice de figuras

1.	Organigrama organizacional.	10
2.	Mapa de proceso comercial y vinculación comercial.	11
3.	Bienvenida y programación de activación de local.	12
4.	Capacitación local virtual.	13
5.	Árbol de problemas del presente proyecto.	16
6.	Diagrama de marco de trabajo Scrum	21
7.	Petición y respuesta del endpoint de autenticación (Contenido V2)	25
8.	Prueba del endpoint de registro en Insomnia (Contenido V2)	26
9.	Prueba del endpoint de recuperación de contraseña (Contenido V2)	27
10.	Ejemplo del correo de recuperación de contraseña recibido por el usuario (Contenido V2)	27
11.	Prueba del endpoint de creación de marca (Contenido V2)	29
12.	Prueba del endpoint de obtención de marcas (Contenido V2)	30
13.	Prueba del endpoint de borrado de marca (Contenido V2)	31
14.	Prueba del endpoint de actualización de marca (Contenido V2)	32
15.	Registro automático de logs durante el flujo de checkout (Contenido V2) .	34
16.	Prueba del endpoint de creación de ciudad (Contenido V2)	35
17.	Prueba del endpoint de obtención de ciudades (Contenido V2)	36
18.	Prueba del endpoint de creación de departamento (Contenido V2)	37
19.	Prueba del endpoint de obtención de departamentos (Contenido V2) . . .	38
20.	Prueba del endpoint de creación de país (Contenido V2)	39
21.	Prueba del endpoint de obtención de países (Contenido V2)	40
22.	Prueba del endpoint de creación de estado de factura (Contenido V2) . . .	41
23.	Prueba del endpoint de actualización de estado de factura (Contenido V2)	42
24.	Prueba del endpoint de eliminación de estado de factura (Contenido V2) .	43
25.	Prueba del endpoint de obtención de estados de factura (Contenido V2) .	44
26.	Prueba del endpoint de creación de transacción en BePay (Contenido V2)	46
27.	Prueba del endpoint de creación de transacción PSE en BePay (Contenido V2)	47
28.	Prueba del endpoint de creación de transacción con tarjeta de crédito en BePay (Contenido V2)	48
29.	Prueba del endpoint de guardado parcial del Onboarding, pendiente de completar campos requeridos (Contenido V2)	49
30.	Prueba del endpoint de registro y envío de Onboarding completo hacia BePay (Contenido V2)	50
31.	Ejemplo de estructura de campos booleanos enviados en formato aceptado por BePay (Y/N) (Contenido V2)	51

Resumen

El comercio electrónico en Colombia ha experimentado un crecimiento notable en los últimos años, lo que presenta tanto una oportunidad como un desafío para plataformas tecnológicas como Contenido. El proyecto de grado tiene como objetivo mejorar la plataforma mediante la implementación de nuevos módulos y funcionalidades, como la gestión de productos, que facilitará la administración eficiente de catálogos e inventarios; la administración de tiendas virtuales, optimizando la gestión de configuraciones y datos de las tiendas; la configuración de interfaces de usuarios finales, proporcionando una experiencia intuitiva y personalizada para los clientes; y la optimización de métodos de pago, que garantizará transacciones ágiles y seguras.

El principal objetivo es optimizar la experiencia del usuario y garantizar que la plataforma siga siendo competitiva en un entorno digital en constante evolución. Para lograrlo, se utilizará el marco de trabajo ágil Scrum, que permitirá entregar avances continuos y asegurar que cada módulo y funcionalidad cumpla con los requisitos establecidos.

Este proyecto no solo permitirá aplicar y ampliar los conocimientos adquiridos durante la formación académica, sino también contribuir al fortalecimiento y éxito de Contenido en un mercado altamente competitivo, mejorando su capacidad para satisfacer las necesidades de los usuarios y administradores de la plataforma.

PALABRAS CLAVE: E-commerce, Optimización, Seguridad, Interfaz, Integración.

Abstract

E-commerce in Colombia has experienced remarkable growth in recent years, presenting both an opportunity and a challenge for technological platforms like Contenido. The goal of this thesis project is to enhance the platform by implementing new modules and functionalities, such as product management to enable efficient catalog and inventory administration; virtual store management, optimizing store configuration and data management; user interface configuration for end-users, providing an intuitive and personalized experience for customers; and payment method optimization to ensure fast and secure transactions.

The main objective is to optimize the user experience and ensure the platform remains competitive in an ever-evolving digital environment. To achieve this, the Scrum agile framework will be used, enabling continuous progress delivery and ensuring that each module and functionality meets the established requirements.

This project will not only allow the application and expansion of the knowledge gained during academic training but also contribute to strengthening and ensuring the success of Contenido in a highly competitive market by improving its ability to meet the needs of users and platform administrators.

KEYWORDS: E-commerce Optimization, Security, Interface, Integration.

1. Introducción

Contento es una plataforma tecnológica diseñada para ofrecer soluciones integrales que optimizan procesos clave en empresas, abarcando tanto la gestión operativa como la estrategia empresarial. 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 este trabajo de grado, se abordará el diseño e implementación de módulos clave para la nueva versión de la plataforma Contento, enfocados en optimizar la experiencia de los usuarios finales y mejorar la gestión operativa. Esto incluirá la gestión de productos, facilitando la administración de catálogos e inventarios de manera eficiente y escalable; la administración de tiendas virtuales, permitiendo a los administradores gestionar configuraciones, productos y datos de las tiendas con facilidad; la configuración de interfaces de usuario, garantizando una experiencia intuitiva y personalizada para los clientes finales; y finalmente, el módulo de métodos de pago, que optimizará el flujo de transacciones para garantizar procesos seguros y ágiles.

La migración tecnológica no solo busca mejorar el rendimiento y la funcionalidad de 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.

1.1. Descripción de la empresa

1.1.1. Descripción general

Contento es una empresa que busca desarrollar herramientas digitales motivadas a sumar en empresas de comercialización que quieren reinventarse, crecer y avanzar con la interconectividad, la automatización y los datos en tiempo real.

Buscamos contribuir en el crecimiento de empresas colombianas con nuestras soluciones tecnológicas de comercialización mediante alianzas con entidades que también quieren ver a los comerciantes colombianos contentos.

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. Número de cargos

Doce (12) cargos

1.1.8. Número de empleados

Dieciocho (18) empleados

1.1.9. Organigrama de la empresa

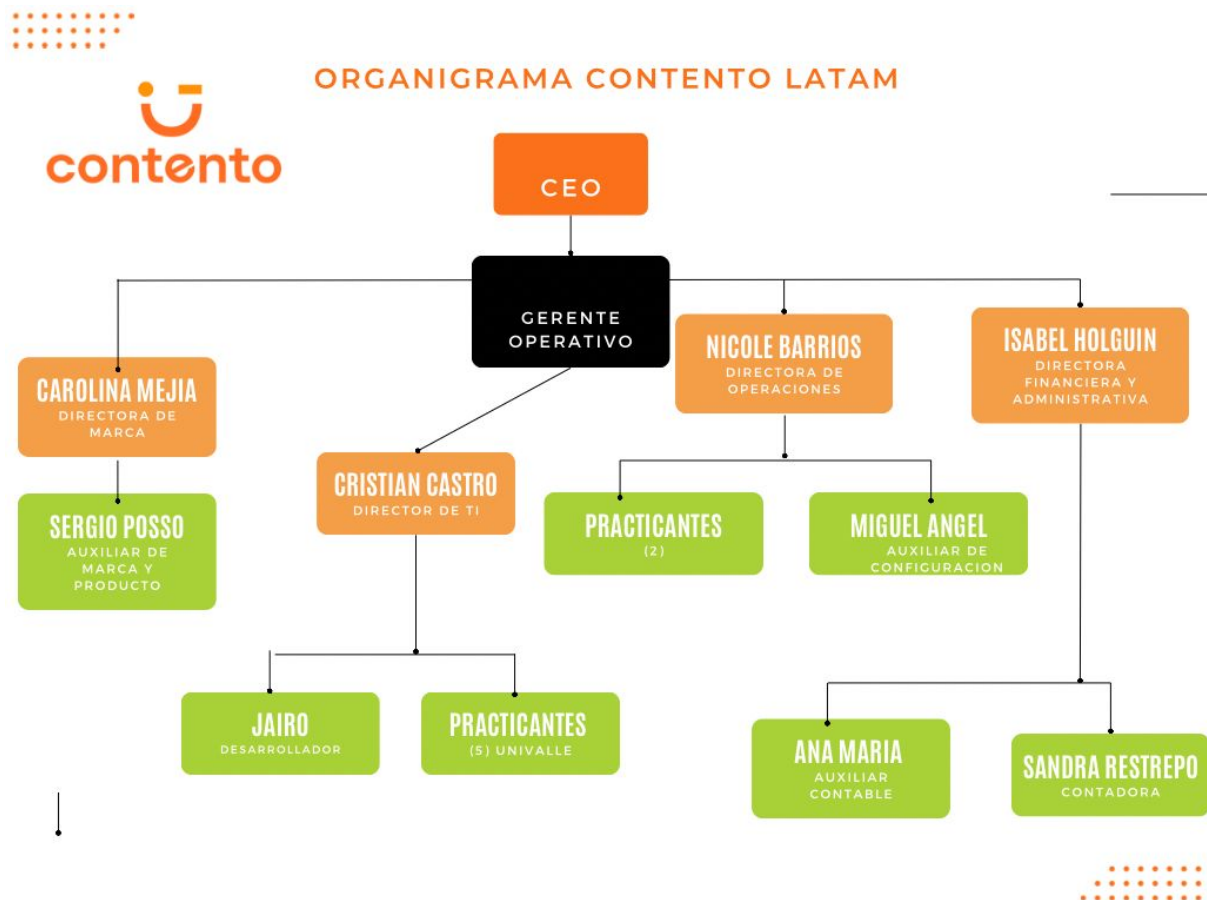


Figura 1: Organigrama organizacional.

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

1.1.10. Mapa de procesos de la empresa

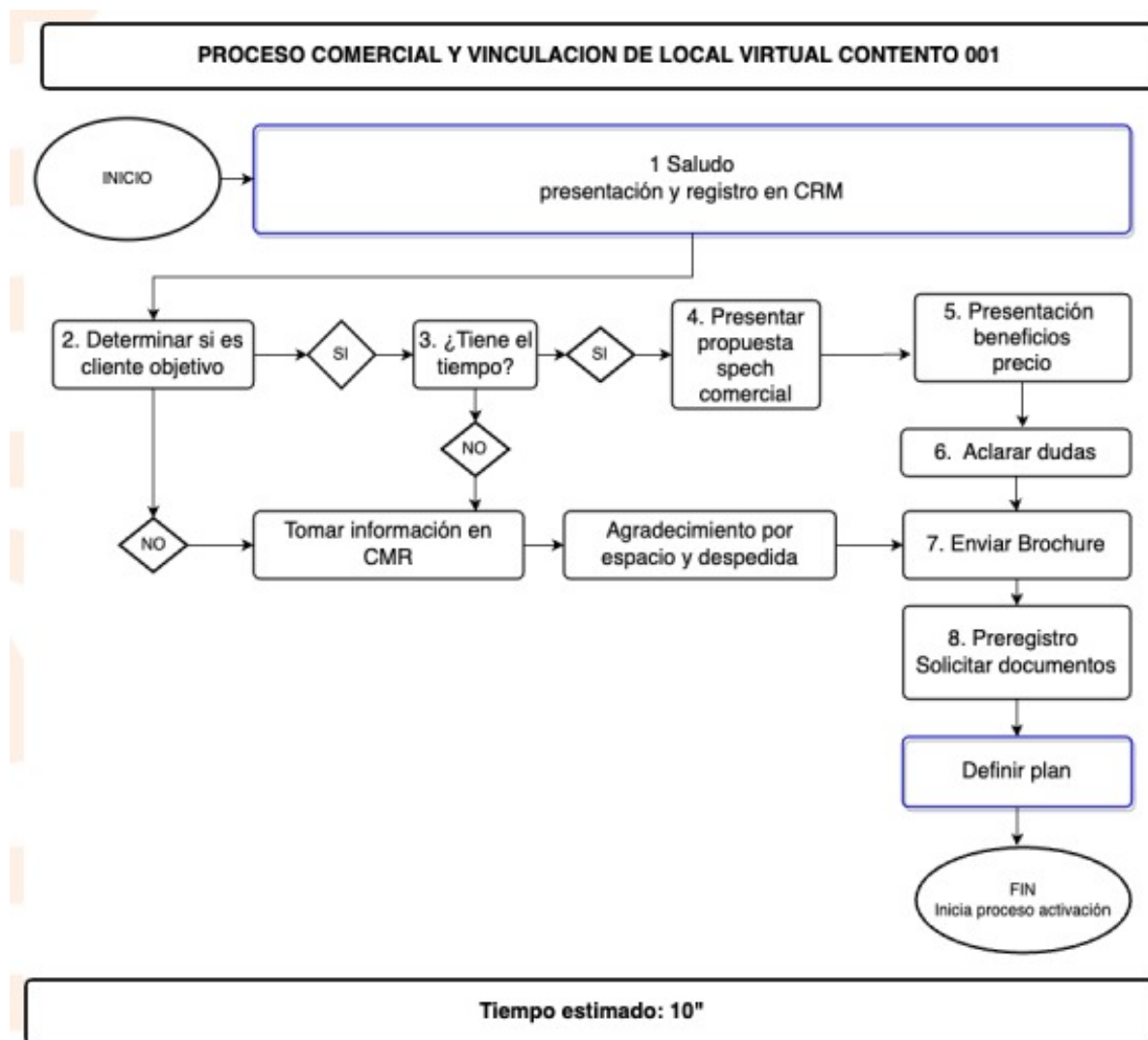


Figura 2: Mapa de proceso comercial y vinculación comercial.
Fuente: Directora general de finanzas (Isabel Cristina Holguín Prado).

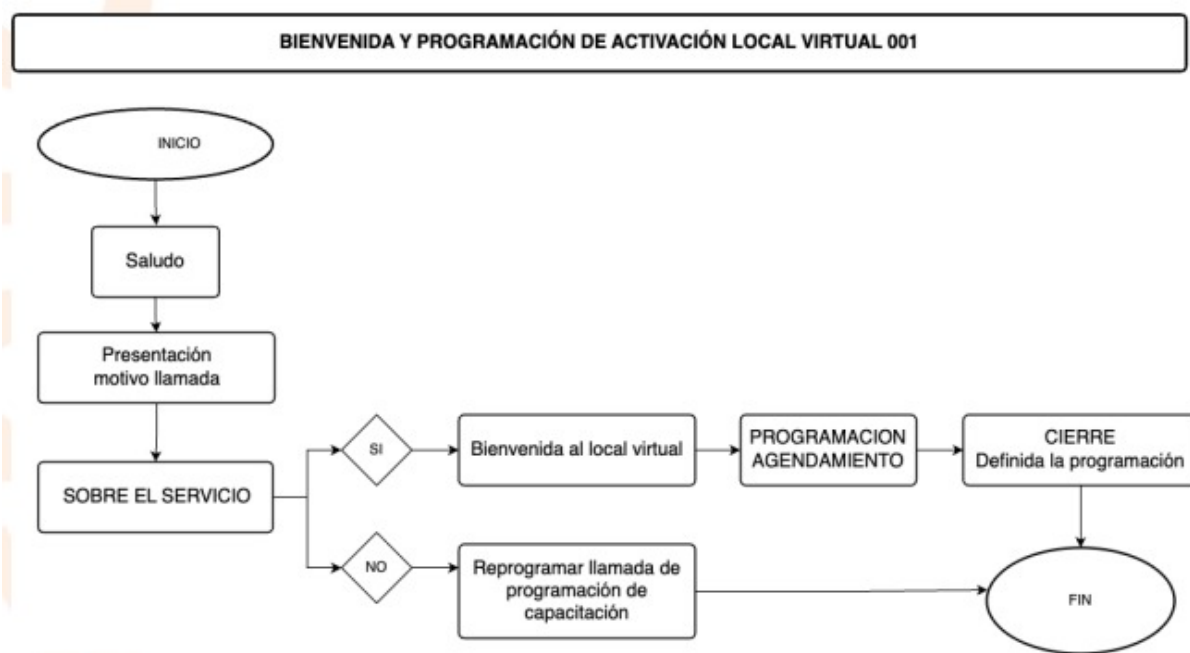


Figura 3: Bienvenida y programación de activación de local.
 Fuente: Directora general de finanzas (Isabel Cristina Holguín Prado).

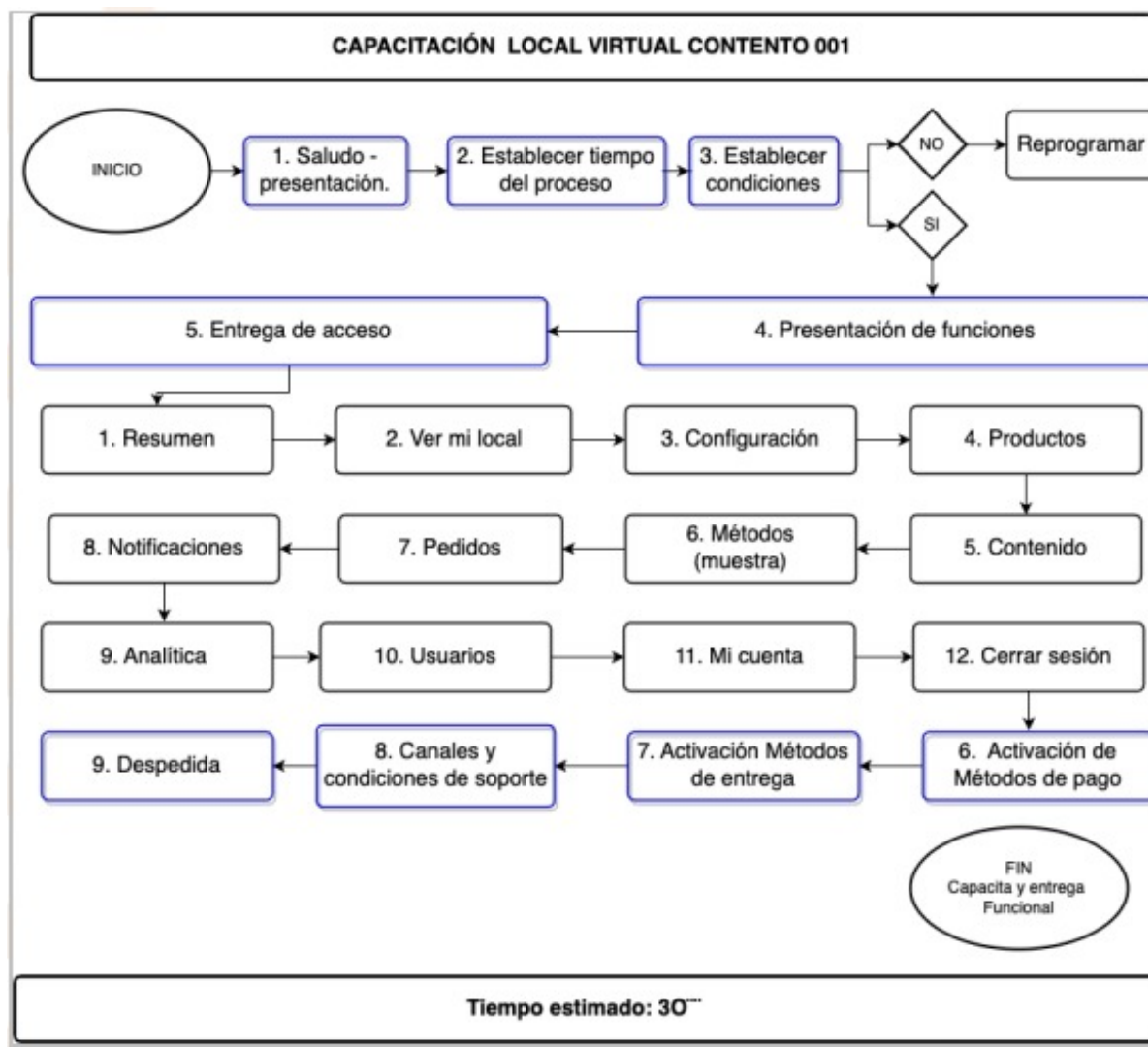


Figura 4: Capacitación local virtual.

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

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

1. **Desarrollo backend de la plataforma *Contento*:** Participar en el diseño, implementación y optimización de módulos del lado del servidor, abarcando aspectos como la gestión de datos y creación de servicios.
2. **Mantenimiento y mejora continua:** Ejecutar tareas de mantenimiento correctivo y evolutivo sobre los componentes backend existentes, aplicando buenas prácticas de programación y control de versiones.
3. **Colaboración en la arquitectura del sistema:** Contribuir en la planificación y diseño de nuevas funcionalidades, asegurando la correcta integración con la arquitectura general del sistema.
4. **Trabajo en equipo bajo metodologías ágiles:** Participar activamente en las dinámicas del equipo de desarrollo utilizando la metodología ágil *Scrum*, asistiendo a reuniones de planificación, seguimiento y revisión de sprints, con el fin de mantener una entrega continua de valor al proyecto.

5. **Investigación y aplicación de nuevas tecnologías:** Analizar y aplicar herramientas, librerías y frameworks modernos del ecosistema backend que permitan mejorar el rendimiento, la seguridad y la mantenibilidad del sistema.

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.[1] 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, 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, la versión actual presenta limitaciones en aspectos clave del backend, como la gestión centralizada de datos, la administración eficiente de marcas, la autenticación segura de usuarios y la optimización de los procesos de pago. Estas limitaciones afectan la escalabilidad, seguridad y mantenibilidad del sistema, dificultando su capacidad para adaptarse a las necesidades de crecimiento y modernización de sus clientes.

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.

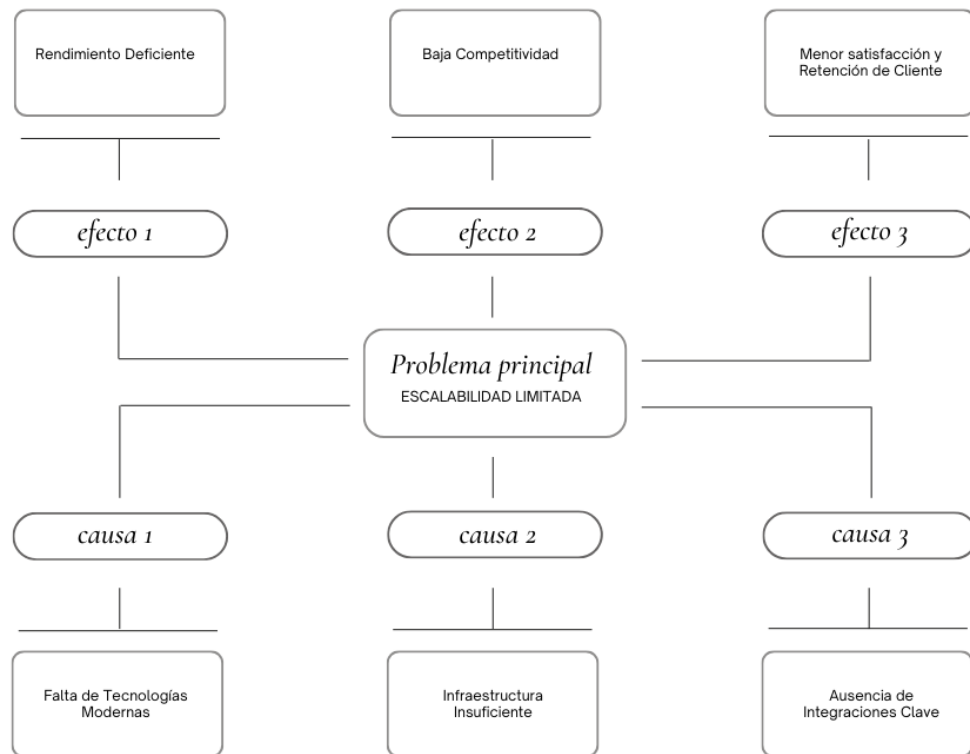


Figura 5: Árbol de problemas del presente proyecto.
Fuente: Elaboración propia.

1.2.3. Formulación del problema

¿Cómo migrar los módulos y sus funcionalidades al nuevo lenguaje, y desarrollar, cuando sea necesario, nuevas funcionalidades en los módulos de autenticación, gestión de marcas, gestión de datos comunes y experiencia de pago en la nueva versión de la plataforma Contenido, garantizando su funcionamiento y mantenibilidad durante la transición desde la versión anterior y la adopción de nuevas tecnologías?

1.3. Alcance del proyecto

1.3.1. Objetivo general

Diseñar, desarrollar e integrar módulos backend clave en la nueva versión de la plataforma mediante la implementación de endpoints, servicios y mecanismos de validación que soporten la autenticación de usuarios, la gestión de datos comunes, la administración de marcas y el flujo de pagos. Estas implementaciones fortalecen la arquitectura del sistema al mejorar su escalabilidad, mantenibilidad y confiabilidad operativa.

1.3.2. Objetivos específicos

- **1.** Diseñar e implementar un módulo de autenticación mediante endpoints y lógica backend que garantice un acceso seguro, validación de credenciales y protección de la información de los usuarios.
- **2.** Desarrollar un módulo de gestión de datos comunes que centralice información reutilizable a través de servicios backend y validaciones estandarizadas, facilitando su administración e interoperabilidad entre módulos.
- **3.** Implementar un módulo de gestión de marcas mediante operaciones CRUD seguras que permitan su administración estructurada y trazable dentro del sistema.
- **4.** Integrar el flujo de pagos mediante la implementación de endpoints y servicios que permitan la comunicación segura con la pasarela de pago, garantizando la correcta gestión y seguimiento de transacciones.

1.3.3. Objetivos esperados

Objetivo	Objetivo Esperado
Crear un módulo backend de autenticación que garantice un acceso seguro y confiable a la plataforma, fortaleciendo la seguridad.	Implementación de endpoints de autenticación robusto con validación de credenciales.
Diseñar e implementar un módulo backend de gestión de datos comunes que permita centralizar y organizar de manera eficiente la información compartida en la plataforma.	Desarrollo de endpoints con estructura de datos estandarizada.
Desarrollar un módulo backend de gestión de marcas que optimice la administración y categorización de las mismas, mejorando la organización y accesibilidad de la información.	Implementación de endpoints para la creación, edición, eliminación y consulta de marcas.
Optimizar la experiencia de pago mediante la implementación de un sistema backend que facilite transacciones fluidas y confiables, mejorando la satisfacción del usuario y la eficiencia operativa.	Integración de un endpoints para una experiencia de pago seguro con validación de transacciones, manejo de errores y registros de actividad, garantizando un proceso ágil y confiable.

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 aplicarán el 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.[2]

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. [3]

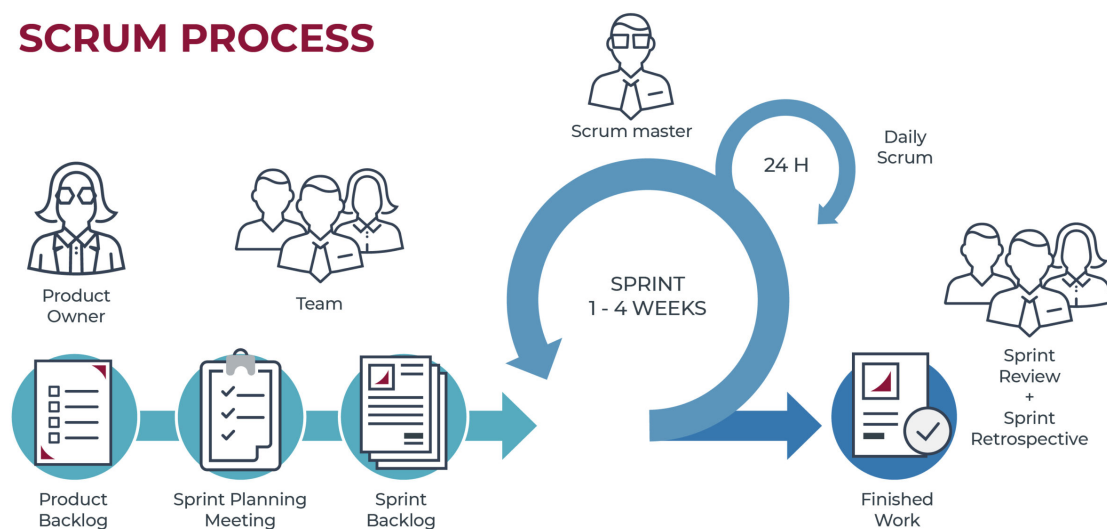


Figura 6: Diagrama de marco de trabajo Scrum

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.[4]

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.**react'2024**

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.[5]

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.[6]

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. [7]

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. [8]

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.[9]

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.[10]

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**

3. Metodología

En Contenido, se utiliza el marco de trabajo ágil Scrum para organizar y gestionar el desarrollo de la plataforma. El equipo de desarrollo está compuesto por 6 personas, incluyendo ingenieros y practicantes, distribuidos en los roles de Product Owner, Scrum Master y Scrum Team. Este marco de metodología ágil permite al equipo adaptarse de forma flexible a los cambios y entregar valor continuamente.

Los sprints en Contenido tienen una duración de dos semanas. En la Sprint Planning, se seleccionan las tareas del Product Backlog, y cada tarea es estimada en términos de Sprint Points utilizando la secuencia de Fibonacci. Esto ayuda a asignar puntos de esfuerzo de manera proporcional a la complejidad de cada tarea, asegurando que las tareas sean alcanzables dentro del tiempo del sprint.

Las Daily Scrums se realizan los martes y jueves para evaluar el progreso, discutir obstáculos y mantener la alineación del equipo con los objetivos del sprint. Estas reuniones son breves y flexibles, siendo programadas por el Scrum Máster según las necesidades del equipo.

El Sprint Backlog consiste en el conjunto de tareas seleccionadas para un sprint específico. Durante el sprint, el equipo trabaja en estas tareas, ajustando el esfuerzo si es necesario para alcanzar los objetivos. Al final del sprint, se lleva a cabo una Sprint Review para evaluar los resultados y planificar los siguientes pasos.

4. Ingeniería de desarrollo

4.1. Módulo de Autenticación

Objetivo del Módulo

El objetivo de este módulo es diseñar e implementar la lógica de autenticación del sistema mediante la construcción de endpoints seguros que gestionen el acceso de los usuarios al marketplace. Este módulo abarca el desarrollo de servicios backend responsables de procesos críticos como el inicio de sesión, registro de usuarios y recuperación de contraseñas.

La implementación incluye mecanismos de seguridad orientados a la protección de credenciales, validación de datos y control de acceso, garantizando la integridad y confidencialidad de la información. Además, el módulo contempla la integración de técnicas de cifrado de contraseñas y manejo de errores, asegurando una experiencia de autenticación confiable tanto para compradores como para vendedores.

De esta manera, el módulo no solo habilita la interacción segura de los usuarios con la plataforma, sino que también establece una base estructurada para la gestión de identidad, validaciones y flujos de autenticación dentro de la arquitectura del sistema.

- Implementar endpoints para el inicio de sesión mediante correo electrónico y contraseña.
- Desarrollar servicios de registro de usuarios con validación de campos obligatorios.
- Construir el flujo de recuperación de contraseñas mediante envío de correo.
- Gestionar respuestas y mensajes de error ante intentos fallidos de autenticación.
- Aplicar hashing de contraseñas (bcrypt) para garantizar la seguridad de las credenciales en la base de datos.

Login (POST /auth/login)

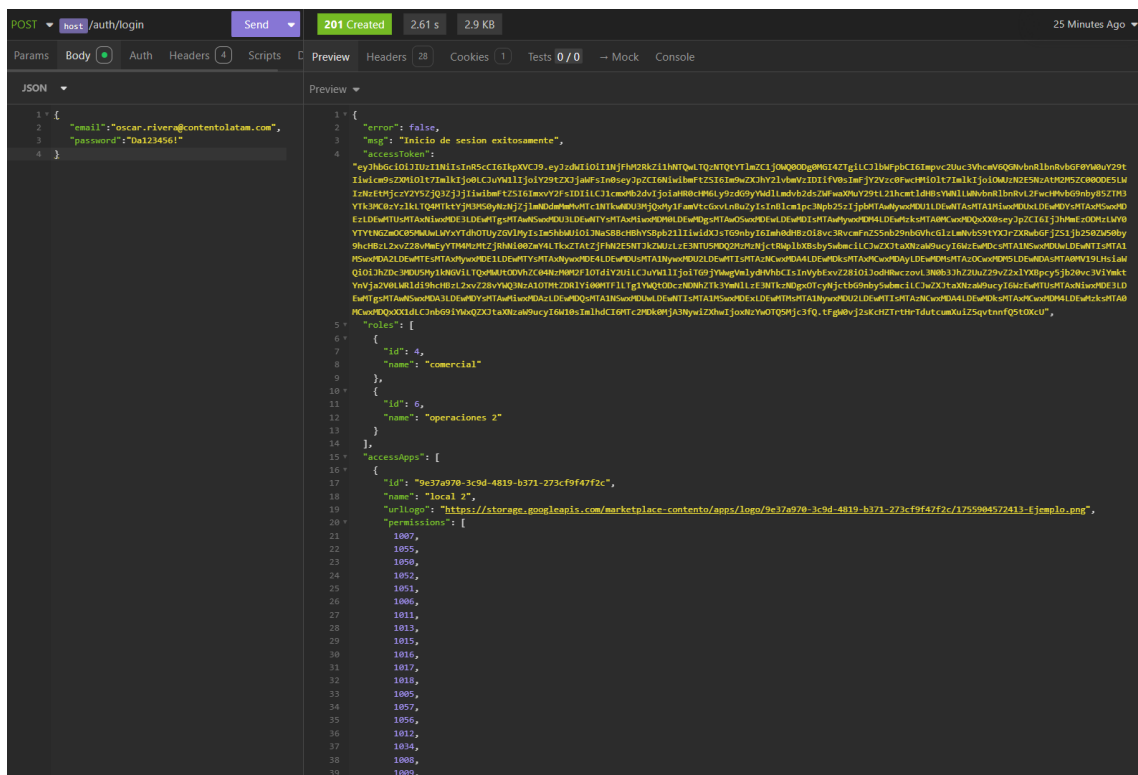


Figura 7: Petición y respuesta del endpoint de autenticación (Contenido V2)

La imagen muestra la prueba realizada desde Insomnia al endpoint de autenticación de la plataforma Contenido V2. Este endpoint recibe las credenciales del usuario (correo electrónico y contraseña), valida su existencia en la base de datos y, en caso de ser correctas, genera un token JWT que permite mantener la sesión activa de forma segura.

El token firmado contiene la información básica del usuario, permisos y un tiempo de expiración predefinido, garantizando la autenticación y autorización adecuadas dentro del sistema. En caso de que las credenciales no sean válidas, el servicio responde con el mensaje de error correspondiente.

Registro - Endpoint

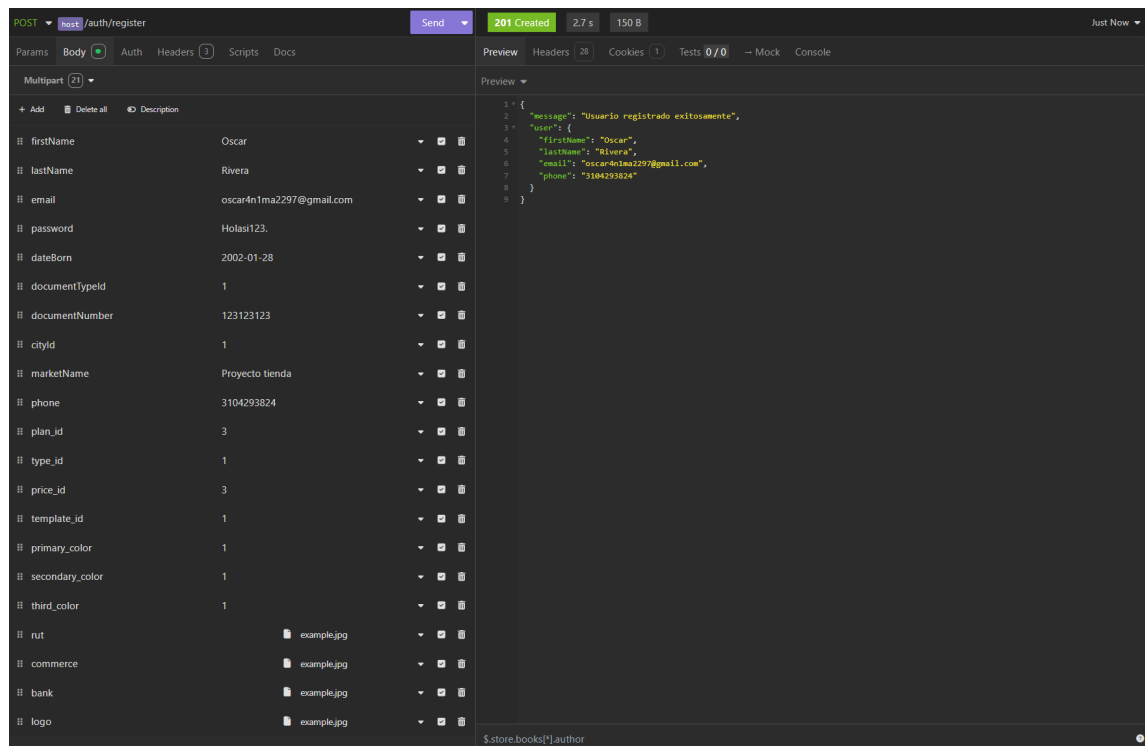


Figura 8: Prueba del endpoint de registro en Insomnia (Contenido V2)

La imagen muestra la prueba del endpoint de registro de la plataforma Contenido V2 realizada mediante Insomnia. En esta solicitud se envían los datos necesarios para crear un nuevo comercio y su usuario administrador, incluyendo la información de contacto, credenciales y datos de configuración inicial.

El endpoint recibe la información enviada en formato JSON y realiza las validaciones correspondientes, como la verificación de la existencia previa del correo electrónico, la coincidencia de contraseñas y la disponibilidad del nombre de la tienda. Una vez superadas las validaciones, se ejecuta el proceso de registro, almacenando los datos en la base de datos y retornando una respuesta satisfactoria con la información del usuario creado y los detalles del comercio asociado.

Recuperación de contraseña - (POST /auth/recovery-password)

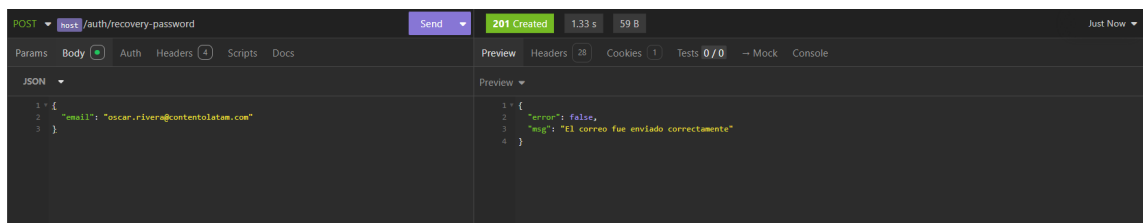


Figura 9: Prueba del endpoint de recuperación de contraseña (Contenido V2)

La imagen muestra la prueba del endpoint encargado del proceso de recuperación de contraseña en la plataforma Contenido V2. A través de esta solicitud, el usuario envía su correo electrónico registrado para iniciar el proceso de restablecimiento de contraseña. El sistema valida la existencia del usuario y, en condiciones normales, genera un código OTP temporal que es enviado al correo asociado para continuar con el procedimiento de recuperación.

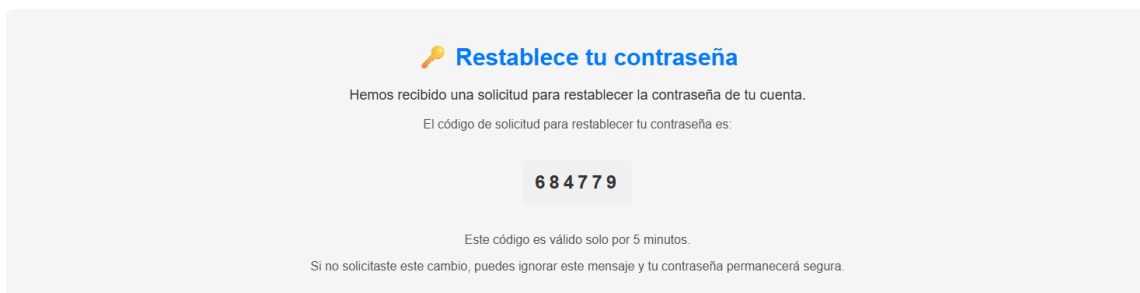


Figura 10: Ejemplo del correo de recuperación de contraseña recibido por el usuario (Contenido V2)

La imagen muestra el ejemplo del correo que recibiría el usuario en un entorno de producción, el cual contiene el código OTP necesario para restablecer su contraseña de manera segura.

4.2. Módulo de gestión de marcas

4.2.1. Objetivo del Módulo

El objetivo de este módulo es diseñar e implementar la gestión de marcas del marketplace mediante el desarrollo de endpoints y lógica backend que permitan su administración de forma estructurada, segura y escalable. Este módulo soporta la organización del catálogo de productos al facilitar la creación, actualización, consulta y desactivación de marcas asociadas a una aplicación específica.

La implementación contempla servicios encapsulados que gestionan operaciones CRUD, validaciones de datos y mecanismos de control de acceso, garantizando la integridad de la información y la correcta interacción con la base de datos. Asimismo, se incorporan estrategias de eliminación lógica y registro de trazabilidad para mantener la consistencia del sistema.

De esta manera, el módulo establece una capa funcional que permite administrar las marcas como entidades clave dentro de la arquitectura del marketplace, asegurando control, mantenimiento y escalabilidad del catálogo.

- Implementar endpoints para la creación de marcas con validación de datos y asociación a una aplicación específica.
- Desarrollar servicios para la actualización de información de marcas existentes.
- Construir mecanismos de consulta para listar marcas activas por aplicación.
- Implementar eliminación lógica (soft delete) para preservar integridad referencial y trazabilidad.
- Gestionar el registro automático de timestamps de creación, actualización y desactivación.

Creación de marca - (POST /app/appId/brands)

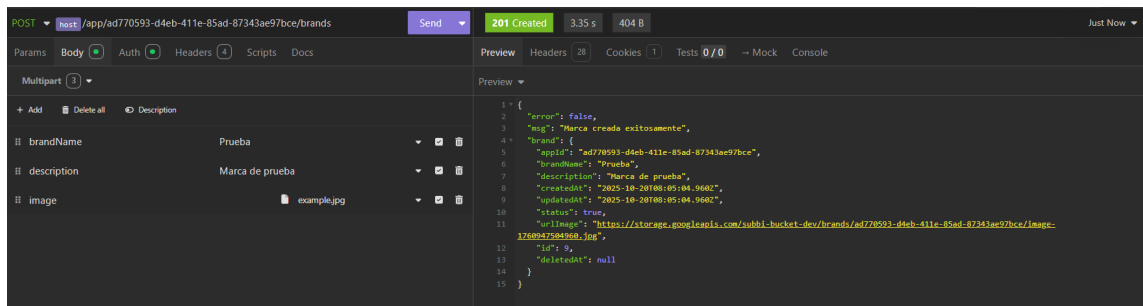


Figura 11: Prueba del endpoint de creación de marca (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint de creación de marca, donde se envían nombre y descripción junto con la imagen correspondiente. Como respuesta, el endpoint devuelve un mensaje de éxito confirmando la creación de la marca, acompañado de la información registrada.

Obtención de marcas - (GET /app/appId/brands)

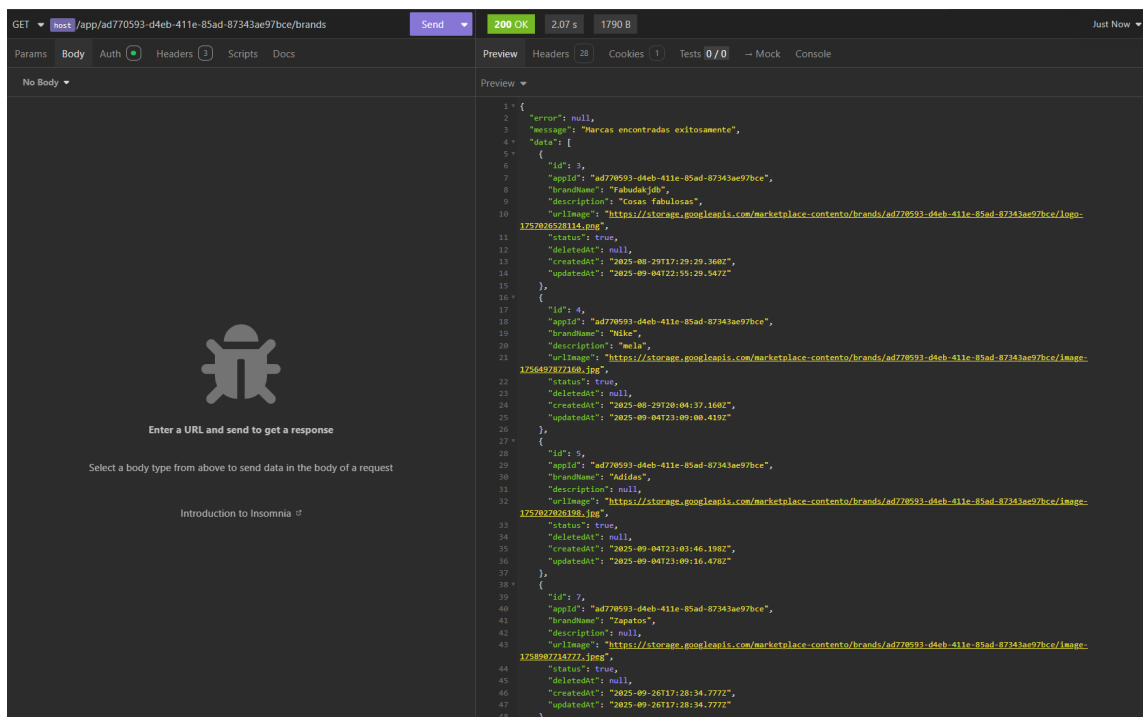


Figura 12: Prueba del endpoint de obtención de marcas (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de obtener todas las marcas registradas en una aplicación específica. En la petición se envía el identificador de la aplicación, y el sistema retorna todas las marcas asociadas a ella.

Como respuesta, el endpoint devuelve un arreglo con la información de cada marca, incluyendo su nombre, descripción, imagen, fecha de creación y estado actual. Esta funcionalidad permite listar de manera eficiente todas las marcas pertenecientes a una aplicación sin necesidad de realizar búsquedas individuales.

Borrado de marca - (DELETE /app/appId/brands/brandId)

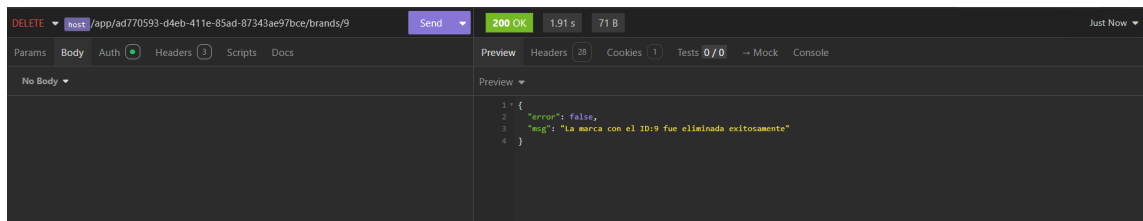


Figura 13: Prueba del endpoint de borrado de marca (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de eliminar una marca existente. Para esta operación, se envía el identificador correspondiente de la marca que se desea eliminar.

Como respuesta, el endpoint devuelve un mensaje confirmando que la marca fue eliminada correctamente. Este proceso realiza un borrado lógico, es decir, la marca no se elimina definitivamente de la base de datos, sino que se actualiza su campo de eliminación con la fecha actual, manteniendo así la integridad de los datos en el sistema.

Actualización de marca - (PATCH /app/appId/brands/brandId)

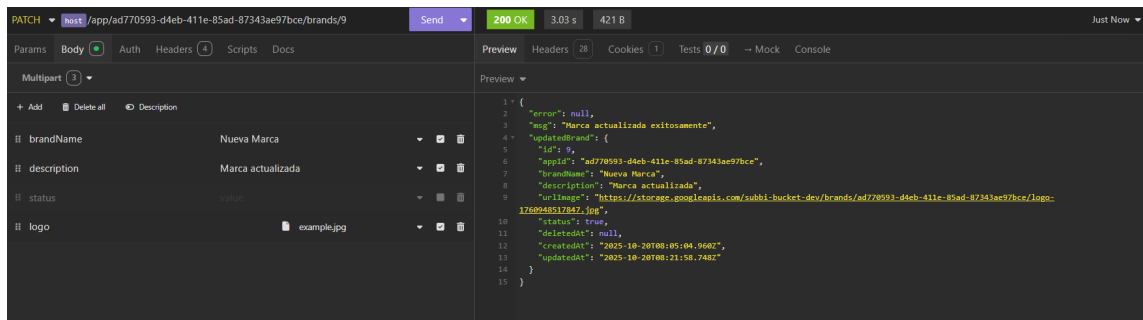


Figura 14: Prueba del endpoint de actualización de marca (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de actualizar la información de una marca existente. En la petición se envían los datos modificados, como el nombre, descripción o imagen, junto con el identificador correspondiente a la marca.

Como respuesta, el endpoint devuelve un mensaje indicando que la marca fue actualizada correctamente. Este proceso valida previamente que la marca exista y que pertenezca a la tienda asociada, garantizando que solo los usuarios con los permisos adecuados puedan realizar modificaciones. En caso de no incluir una nueva imagen, se conserva la imagen previamente registrada.

4.3. Módulo de gestión de datos comunes

4.3.1. Objetivo del Módulo

El objetivo de este módulo es diseñar e implementar la gestión centralizada de datos comunes del sistema mediante el desarrollo de endpoints y servicios backend que permitan administrar información compartida entre distintos componentes de la plataforma. Estos datos corresponden a entidades reutilizables como categorías, tipos de documento, países, ciudades y parámetros de configuración que sirven como base operativa para múltiples módulos del ecosistema.

La implementación contempla la creación de servicios estructurados que gestionan operaciones de registro, actualización y consulta, incorporando validaciones estandarizadas y mecanismos de persistencia que garantizan la consistencia de la información. Asimismo, el módulo promueve la reutilización de datos y la interoperabilidad entre componentes, reduciendo la duplicación de registros y facilitando el mantenimiento del sistema.

De esta manera, el módulo establece una capa de servicios común que soporta la arquitectura del marketplace, asegurando accesibilidad controlada, trazabilidad de cambios y escalabilidad funcional.

- Implementar endpoints para la creación y actualización de registros comunes con validaciones estandarizadas.
- Desarrollar mecanismos de consulta eficiente de datos reutilizables por parte de otros módulos.
- Centralizar la lógica de administración de información compartida para evitar duplicaciones.
- Gestionar la trazabilidad mediante el registro de fechas de creación, actualización y eliminación lógica.

Servicio de registro de logs - (CreateLogService)

El sistema cuenta con un servicio auxiliar denominado **CreateLogService**, encargado de registrar eventos relevantes dentro de la plataforma, brindando trazabilidad y control sobre las operaciones ejecutadas en los distintos módulos. Este servicio forma parte del módulo de **datos comunes**, ya que su funcionalidad puede ser utilizada por cualquier componente del sistema, desde procesos internos hasta integraciones externas.

El servicio recibe un objeto del tipo **CreateLogDto**, que contiene información sobre el tipo de log, una descripción del evento, la URL asociada y datos adicionales opcionales. Su implementación utiliza transacciones de base de datos para garantizar la integridad del registro y almacenar la información de manera estructurada en la entidad **Log**, la cual mantiene relación con la tabla **LogType**, encargada de categorizar los diferentes tipos de eventos (por ejemplo: *Bepay*, *Transacciones*, *Errores de autenticación*, etc.).

Cada registro conserva detalles como la descripción del suceso, un campo adicional en formato JSON que permite incluir información dinámica o respuestas completas de APIs externas, la URL del proceso que lo originó y las fechas de creación y actualización.

Este servicio resulta especialmente útil para:

- Registrar acciones críticas del sistema, como la creación de transacciones o el envío de datos a pasarelas externas.
- Mantener un historial completo de operaciones con fines de auditoría o depuración.
- Centralizar el manejo de logs, evitando la duplicación de lógica en diferentes módulos.
- Asegurar la consistencia y trazabilidad de los eventos mediante una arquitectura unificada.

Gracias a este enfoque, el sistema dispone de un mecanismo robusto para el seguimiento y análisis de eventos, reforzando la transparencia operativa y facilitando la detección temprana de errores o inconsistencias dentro de los procesos internos y externos.

id	log_type_id	description	extra_column1	created_at	updated_at	deleted_at	url
[PK] integer	integer	character varying (255)	text	timestamp with tz	timestamp with tz	timestamp with tz	text
1	3629	Transacción PSE BEPAY creada exitosamente	{\"payload\":{\"direccion\":\"Calle 123\",\"apellidos\":\"Gomez\",\"ciudad\":\"21551\",\"pais\":\"48\",\"company_terms\"...	2025-10-20 0...	2025-10-20 ...	[null]	/bepay/ad770593-d4eb-411e-85ad-87343ae97bce/checkout/pse
2	3629	Transacción PSE BEPAY creada exitosamente	{\"payload\":{\"direccion\":\"Calle 123\",\"apellidos\":\"Gomez\",\"ciudad\":\"21551\",\"pais\":\"48\",\"company_terms\"...	2025-10-20 0...	2025-10-20 ...	[null]	/bepay/ad770593-d4eb-411e-85ad-87343ae97bce/checkout/pse

Figura 15: Registro automático de logs durante el flujo de checkout (Contenido V2)

Creación de ciudad - (POST /cities)

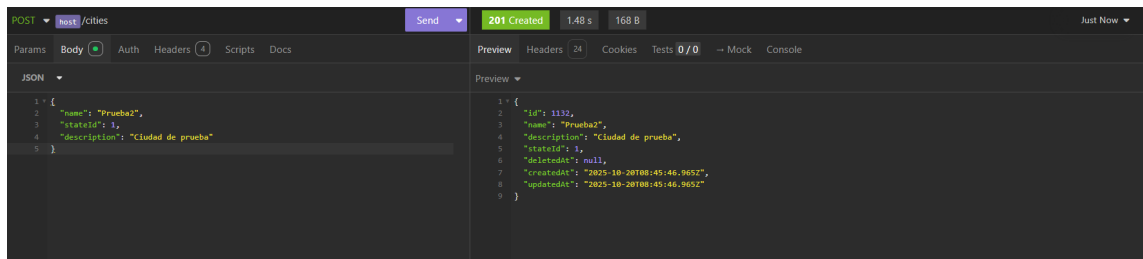


Figura 16: Prueba del endpoint de creación de ciudad (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de registrar una nueva ciudad dentro del sistema. En esta prueba se envían los datos correspondientes al nombre de la ciudad y su país asociado. Como respuesta, el sistema confirma la creación exitosa del registro, devolviendo un mensaje de éxito junto con los datos de la ciudad creada, incluyendo su identificador único, nombre y fecha de creación.

Este endpoint permite mantener actualizado el catálogo de ciudades disponibles en la plataforma, sirviendo como base para otros módulos que requieren asociar ubicaciones geográficas, como tiendas, marcas o direcciones de usuarios.

Obtención de ciudades - (GET /cities)

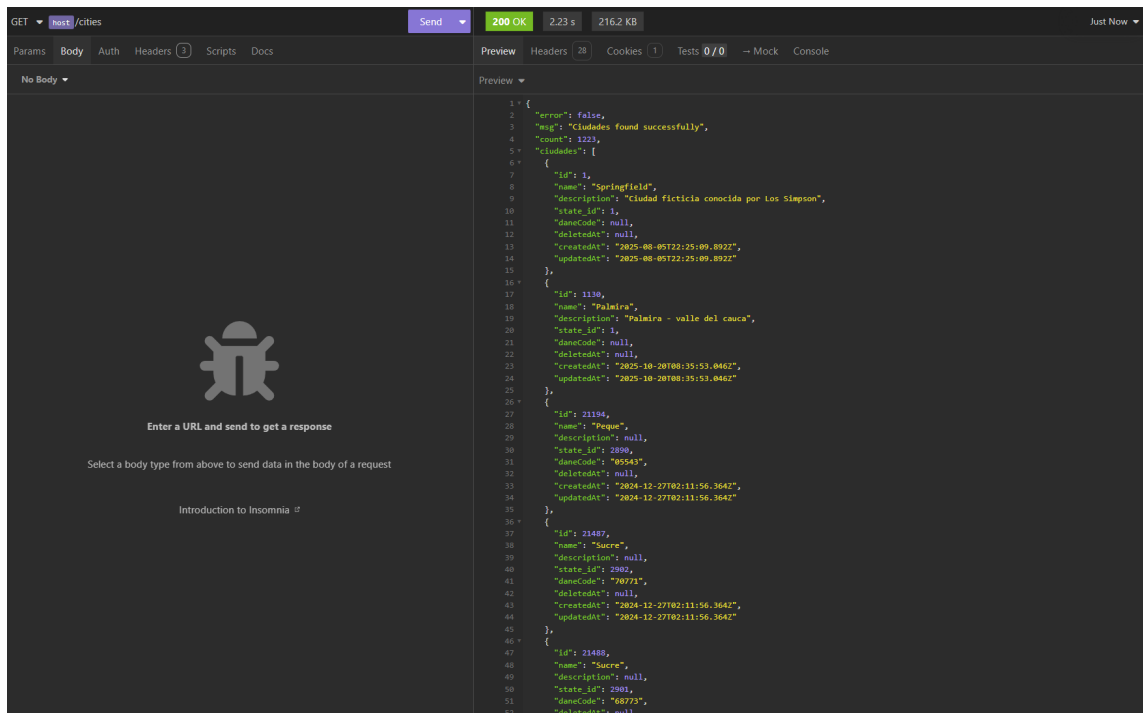


Figura 17: Prueba del endpoint de obtención de ciudades (Contenido V2)

La imagen muestra la solicitud al endpoint encargado de listar todas las ciudades registradas en el sistema. La respuesta obtenida desde Insomnia contiene un arreglo con los registros existentes, donde se detallan campos como el identificador, nombre de la ciudad, país asociado, estado y fechas de creación o actualización.

Este endpoint facilita la consulta general de ciudades, permitiendo que otros módulos de la plataforma consuman esta información de manera centralizada para mantener la coherencia de los datos geográficos utilizados en todo el ecosistema.

Creación de departamento - (POST /states)

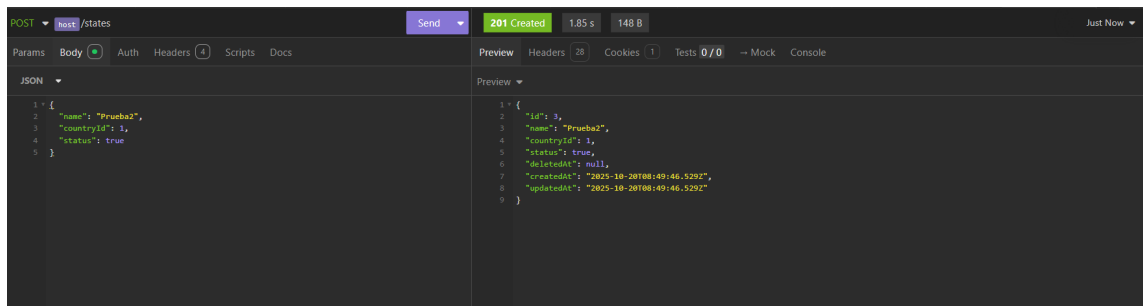


Figura 18: Prueba del endpoint de creación de departamento (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de registrar un nuevo departamento dentro del sistema. En esta prueba se envían los datos correspondientes al nombre del departamento y su país asociado. Como respuesta, el sistema confirma la creación exitosa del registro, devolviendo los datos del departamento creado, incluyendo su identificador único, nombre, estado activo y fechas de creación y actualización.

Este endpoint permite mantener actualizado el catálogo de departamentos disponibles en la plataforma, sirviendo como una capa intermedia entre países y ciudades. Su correcta gestión garantiza una estructura geográfica coherente para las entidades que dependen de esta información, como usuarios, direcciones o comercios.

Obtención de departamentos - (GET /states)

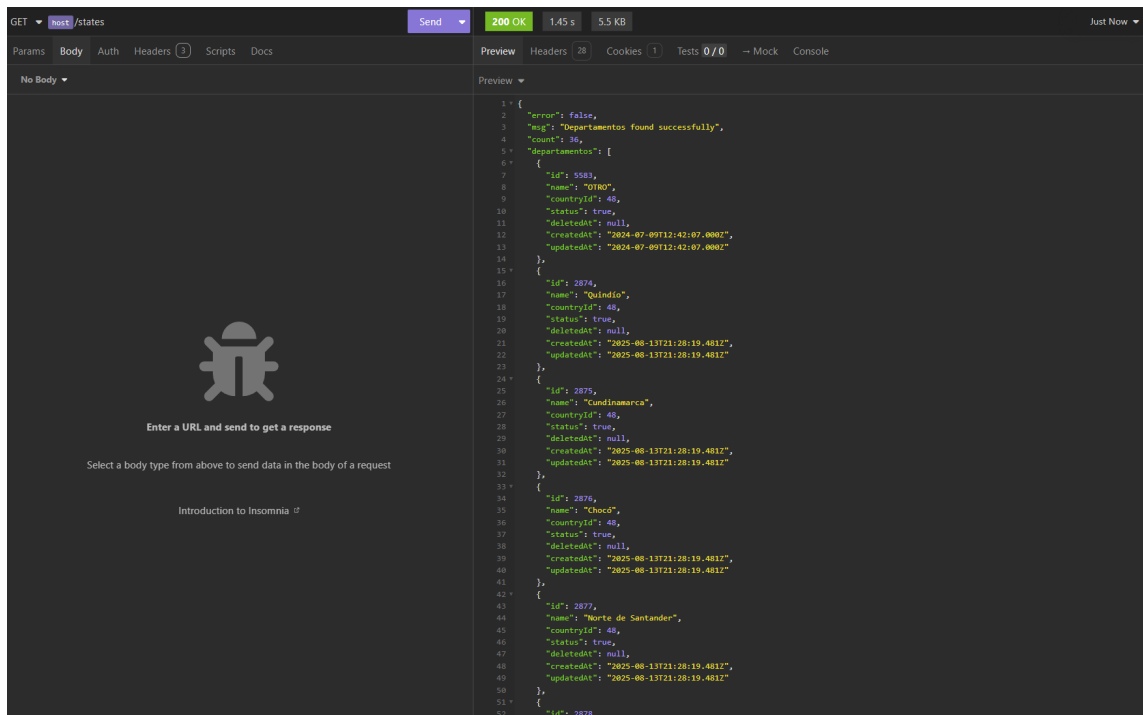


Figura 19: Prueba del endpoint de obtención de departamentos (Contenido V2)

La imagen muestra la solicitud al endpoint encargado de listar todos los departamentos registrados en el sistema. La respuesta obtenida desde Insomnia contiene un arreglo con los registros existentes, detallando campos como el identificador, nombre del departamento, país asociado, estado, y las fechas de creación y actualización.

Este endpoint facilita la consulta general de departamentos o provincias disponibles en la base de datos, permitiendo que otros módulos, como los de ciudades o direcciones, consuman esta información de manera centralizada y consistente dentro del ecosistema de la plataforma.

Creación de país - (POST /countries)

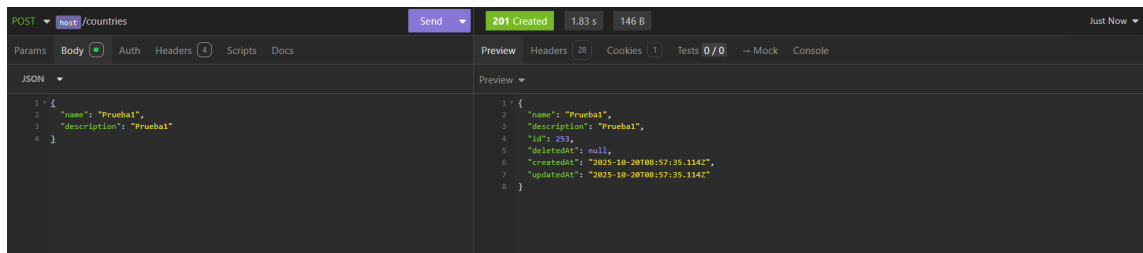
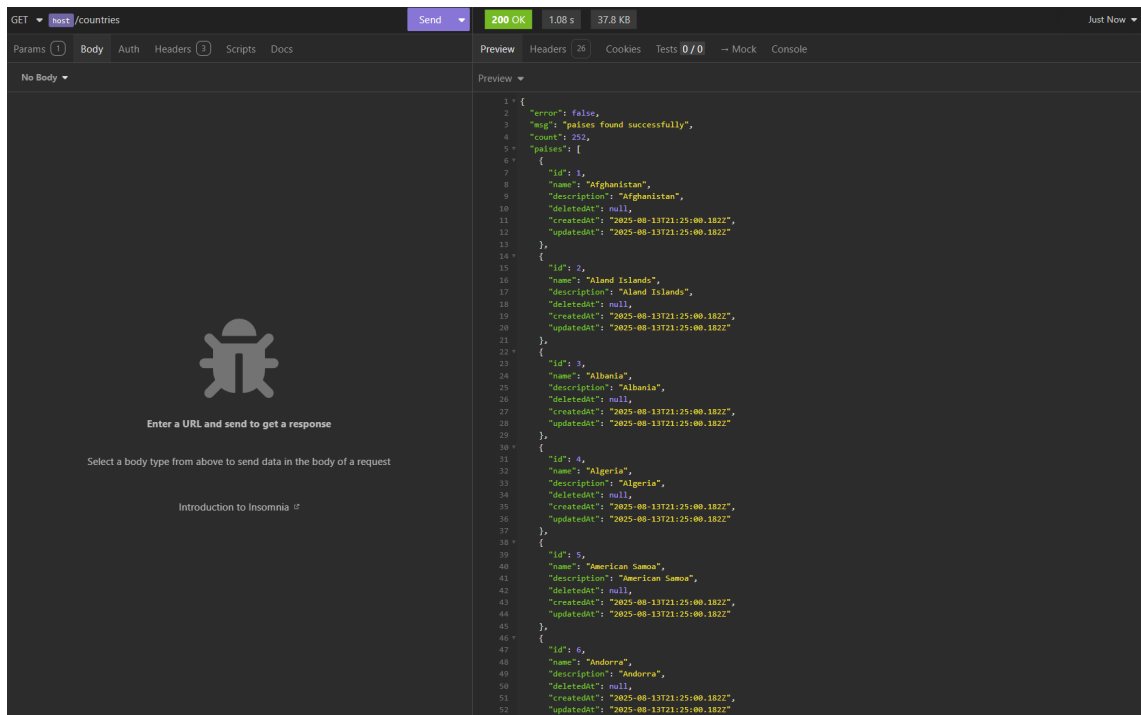


Figura 20: Prueba del endpoint de creación de país (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de registrar un nuevo país dentro del sistema. En esta prueba se envían los datos correspondientes al nombre y descripción del país que se desea agregar. Como respuesta, el sistema confirma la creación exitosa del registro, devolviendo los datos del país creado, incluyendo su identificador único, nombre, descripción y las fechas de creación y actualización.

Este endpoint permite mantener actualizado el catálogo de países disponibles en la plataforma, funcionando como la base geográfica sobre la cual se estructuran los departamentos y ciudades. Su correcta administración garantiza una jerarquía territorial coherente y facilita la gestión de información dependiente, como direcciones, zonas de envío o ubicación de comercios.

Obtención de países - (GET /countries)



Creación de estado de factura - (POST /common/status-invoices)

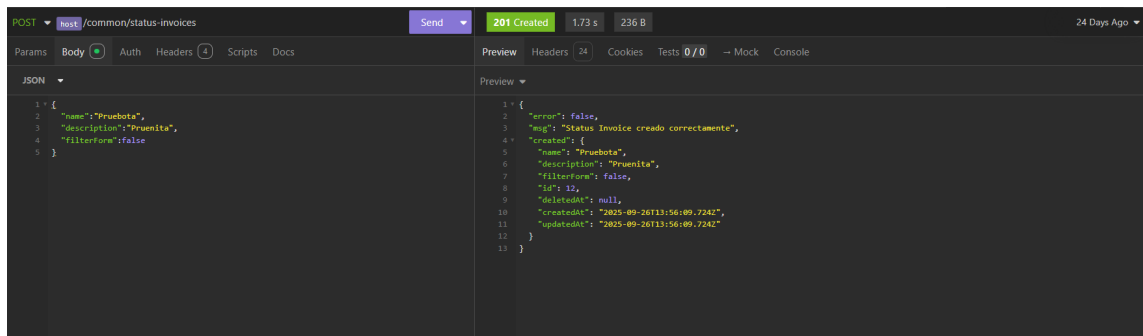


Figura 22: Prueba del endpoint de creación de estado de factura (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint encargado de registrar un nuevo estado de factura dentro del sistema. En esta prueba se envían los datos correspondientes al nombre, descripción y configuración del estado. Como respuesta, el sistema confirma la creación exitosa del registro, devolviendo los datos del estado creado, incluyendo su identificador único, nombre, descripción y las fechas de creación y actualización.

Este endpoint permite mantener actualizada la tabla de estados de facturas, asegurando que el sistema de gestión contable y administrativa cuente con las categorías necesarias para el ciclo de vida de las facturas (por ejemplo: pendiente, pagada, cancelada, anulada, etc.).

Actualización de estado de factura - (PATCH /common/status-invoices/:id)

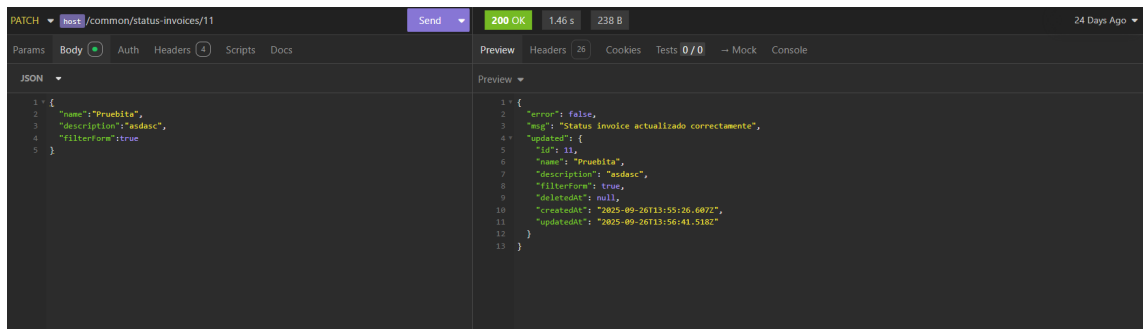


Figura 23: Prueba del endpoint de actualización de estado de factura (Contenido V2)

La imagen muestra la solicitud al endpoint encargado de modificar los datos de un estado de factura existente. En este caso, se envían los nuevos valores del nombre, descripción o campo `filterForm`. Como respuesta, el sistema devuelve la información actualizada del registro junto con un mensaje de éxito que confirma la operación.

Este endpoint resulta útil para mantener los estados de facturación actualizados y coherentes con las necesidades del negocio, permitiendo ajustes sin necesidad de eliminar o recrear registros.

Eliminación de estado de factura - (DELETE /common/status-invoices/:id)

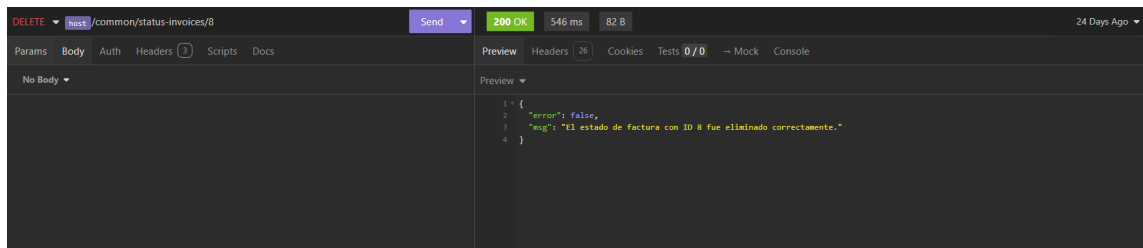


Figura 24: Prueba del endpoint de eliminación de estado de factura (Contenido V2)

La imagen muestra la solicitud realizada al endpoint encargado de eliminar un estado de factura del sistema. El borrado se realiza de forma lógica, estableciendo una marca temporal en el campo `deletedAt`, con el fin de mantener trazabilidad y no afectar la integridad referencial de las facturas que hayan utilizado dicho estado.

Como respuesta, el sistema confirma la eliminación lógica del registro, devolviendo un mensaje de confirmación y la fecha en que se aplicó la operación.

Obtención de estados de factura - (GET /common/status-invoices)

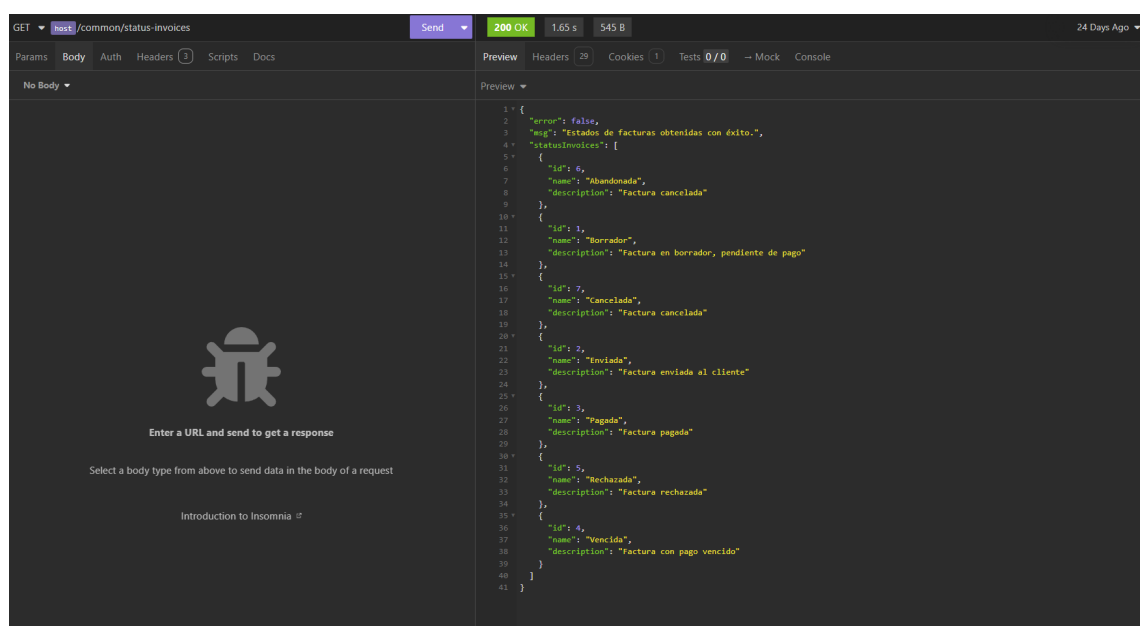


Figura 25: Prueba del endpoint de obtención de estados de factura (Contenido V2)

La imagen muestra la solicitud al endpoint encargado de listar todos los estados de factura registrados en el sistema. La respuesta obtenida desde Insomnia contiene un arreglo con los registros existentes, incluyendo campos como el identificador, nombre, descripción, estado del filtro, y las fechas de creación o actualización.

Este endpoint permite consultar de forma centralizada todos los estados de facturación definidos, siendo utilizado por otros módulos del sistema como el de facturas, reportes y paneles administrativos.

4.4. Módulo de pasarela de pago (BePay)

4.4.1. Objetivo del Módulo

El objetivo de este módulo es diseñar e implementar la integración del sistema con la pasarela de pago **BePay** mediante el desarrollo de endpoints y lógica backend que permitan gestionar de forma segura el flujo de transacciones electrónicas dentro de la plataforma. Este módulo actúa como una capa de comunicación entre la aplicación y el servicio externo de pagos, asegurando la correcta creación, validación y seguimiento de las operaciones financieras.

La implementación contempla servicios responsables de generar órdenes de pago, procesar respuestas del proveedor externo y actualizar el estado de las facturas asociadas, incorporando mecanismos de validación, manejo de errores y protección de datos sensibles. Estas funciones garantizan la integridad de la información y la confiabilidad del proceso de pago.

De esta manera, el módulo establece un flujo transaccional controlado que permite integrar servicios financieros externos dentro de la arquitectura del marketplace, asegurando trazabilidad, seguridad y consistencia en la gestión de pagos.

- Implementar endpoints para la creación y gestión de transacciones mediante la API de BePay.
- Facilitar la creación de transacciones desde el entorno de la aplicación.
- Gestionar el registro y trazabilidad de cada transacción dentro del sistema.

Creación de transacción en BePay - (POST /bepay/app/:appId/checkout/transaction)

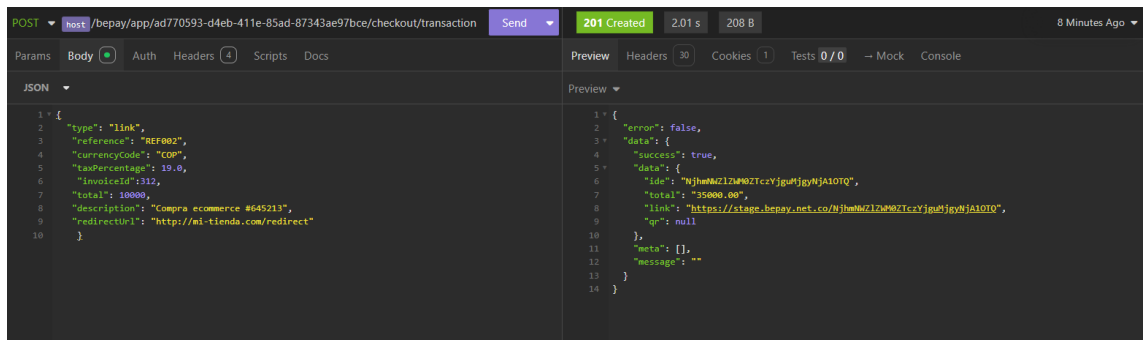


Figura 26: Prueba del endpoint de creación de transacción en BePay (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint responsable de crear una nueva transacción a través de la pasarela de pago **BePay**. Este proceso forma parte del flujo de checkout de la aplicación y permite generar una transacción segura, vinculada a una factura específica dentro del sistema.

Durante la prueba, se envía un cuerpo de solicitud con la información requerida para el registro de la transacción, incluyendo el identificador de la factura, el tipo de operación, el valor total, la moneda utilizada, la descripción de la compra y la URL de redirección tras el pago.

El sistema verifica que la factura exista y que la aplicación asociada al parámetro `appId` sea válida. Luego, obtiene un token de acceso interno mediante el servicio de autenticación de BePay y construye el *payload* con los datos correspondientes. Una vez validados los datos, el sistema realiza una solicitud HTTP al endpoint externo `/api/v1/checkout/transactions` de BePay, enviando los parámetros de la transacción en formato JSON.

Como respuesta, BePay devuelve un objeto con la información de la transacción creada, incluyendo un identificador interno, un código de referencia y la URL del enlace de pago. Esta información es retornada al cliente dentro de la respuesta del sistema, indicando un resultado exitoso.

Creación de transacción PSE en BePay - (POST /bepay/app/:appId/checkout/pse)

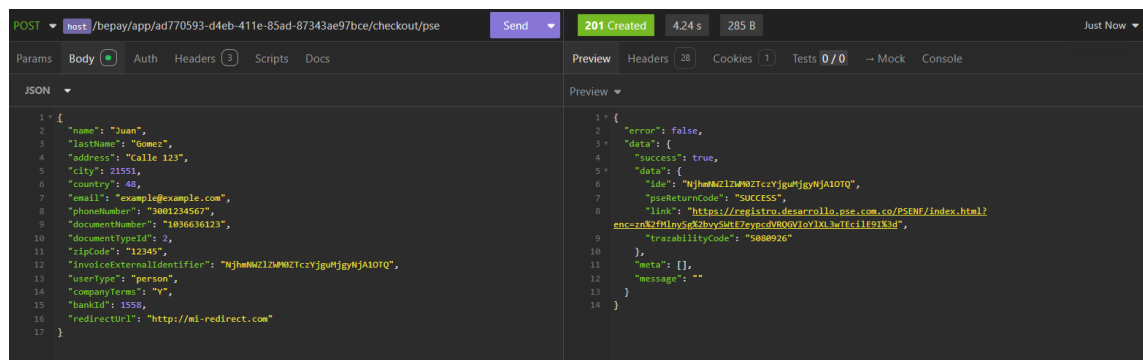


Figura 27: Prueba del endpoint de creación de transacción PSE en BePay (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint destinado a procesar pagos mediante el método **PSE** a través de la pasarela de pago **BePay**. Este servicio permite generar una transacción bancaria directa, enlazando los datos del usuario, la factura y la aplicación con la infraestructura de pagos de BePay.

En la solicitud, el cliente envía los datos del formulario de pago —como nombre, apellidos, tipo y número de documento, correo electrónico, banco seleccionado, dirección, país, ciudad, código postal y URL de redirección— junto con el identificador de la factura externa y la dirección IP del cliente. Además, se incluye en la cabecera el token de autenticación **Authorization-Bepay**, que valida la sesión con la API de BePay.

El sistema, al recibir la petición, realiza las siguientes operaciones:

- Obtiene el **access token** correspondiente a la aplicación mediante el servicio interno **GetAccessTokenInsideBepayService**.
- Construye el *payload* requerido por la API de BePay, respetando los campos esperados por el endpoint externo, incluyendo el identificador de la transacción y el token de seguridad con formato específico.
- Envía la solicitud HTTP a la ruta **/checkout/checkoutPse** de BePay.

En caso de éxito, BePay retorna la información de la transacción creada, incluyendo campos como el **pseReturnCode**, el enlace de pago (**link**) y el **trazabilityCode**, que permiten al usuario continuar el flujo de pago dentro del entorno bancario seguro.

El sistema registra en su bitácora (módulo de logs) tanto las operaciones exitosas como los errores mediante el servicio **CreateLogService**, almacenando el *payload*, la respuesta y los mensajes asociados al evento. Si la API de BePay responde con errores de autenticación o validación, el controlador gestiona las excepciones lanzando respuestas claras y estructuradas hacia el cliente.

Creación de transacción con tarjeta de crédito en BePay - (POST /bepay/app/:appId/checkout/credit-card)

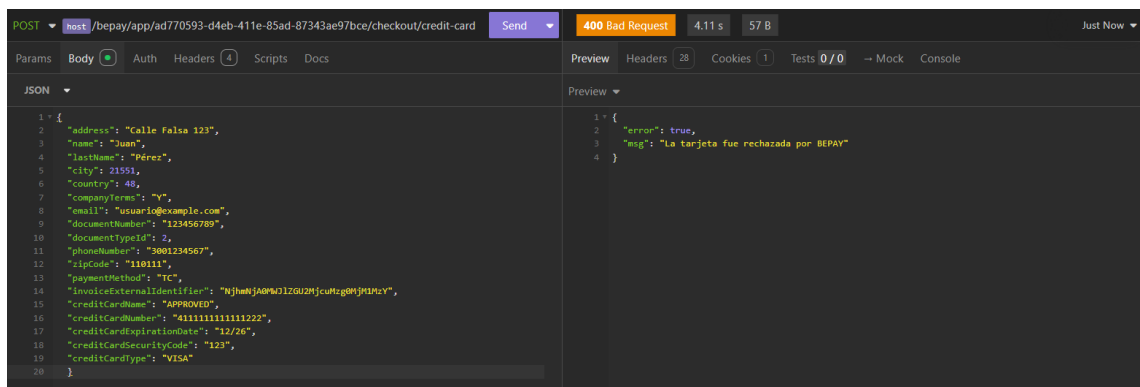


Figura 28: Prueba del endpoint de creación de transacción con tarjeta de crédito en BePay (Contenido V2)

La imagen muestra la solicitud realizada desde Insomnia al endpoint destinado a procesar pagos mediante **tarjeta de crédito** a través de la pasarela de pago **BePay**. Este servicio permite generar una transacción segura cifrando la información sensible de la tarjeta y enviándola a la API de BePay para su validación y procesamiento.

En la solicitud, el cliente envía los datos del formulario de pago —como nombre, apellidos, tipo y número de documento, correo electrónico, teléfono, dirección, país, ciudad, código postal y datos de la tarjeta (nombre del titular, número, fecha de expiración, código de seguridad y tipo de tarjeta)— junto con el identificador de la factura externa y la dirección IP del cliente. El sistema gestiona internamente la autenticación con BePay mediante un **access token** asociado a la aplicación, sin requerir un token explícito en la cabecera de la solicitud.

Al recibir la petición, el sistema ejecuta las siguientes operaciones:

- Obtiene el **access token** y el **accountId** correspondientes a la aplicación a través del servicio interno `GetAccessTokenInsideBepayService`.
- Solicita la **llave pública RSA** desde la API de BePay y cifra los datos de la tarjeta (número, fecha de expiración y código de seguridad) antes de enviarlos.
- Construye el *payload* esperado por BePay y realiza la llamada al endpoint externo `/checkout/checkoutCard`.

Si la transacción es aprobada, BePay devuelve los datos de la operación, incluyendo el estado y los identificadores generados. En caso de rechazo o error, el sistema interpreta la respuesta y lanza una excepción con un mensaje claro al cliente (por ejemplo: "La tarjeta fue rechazada por BEPAY").

Finalmente, el sistema registra cada evento mediante el servicio `CreateLogService`, almacenando tanto los datos sanitizados de la solicitud como la respuesta del proveedor. Esto permite mantener un historial detallado de los intentos de pago, incluyendo los errores de autenticación, validación o comunicación con la pasarela.

Registro y envío de Onboarding hacia BePay - (POST /bepay/onboardings)

El endpoint permite registrar los datos de **onboarding** de un comercio dentro del sistema y enviarlos a la pasarela **BePay** una vez que se cumplen todos los requisitos exigidos por dicha plataforma. El flujo está diseñado para que el cliente pueda realizar el proceso de alta **de forma progresiva**, guardando la información parcialmente sin perder avances entre sesiones o pasos.

En la solicitud, el cliente envía un formulario en formato **multipart/form-data**, compuesto por los campos de identificación del titular, datos de la empresa y los documentos requeridos (como RUT, cámara de comercio, contrato, certificación bancaria, fotos, firma, entre otros). El servicio recibe esta información, la almacena en la base de datos interna y valida si se cumple con la totalidad de los campos obligatorios para proceder con el envío a BePay.

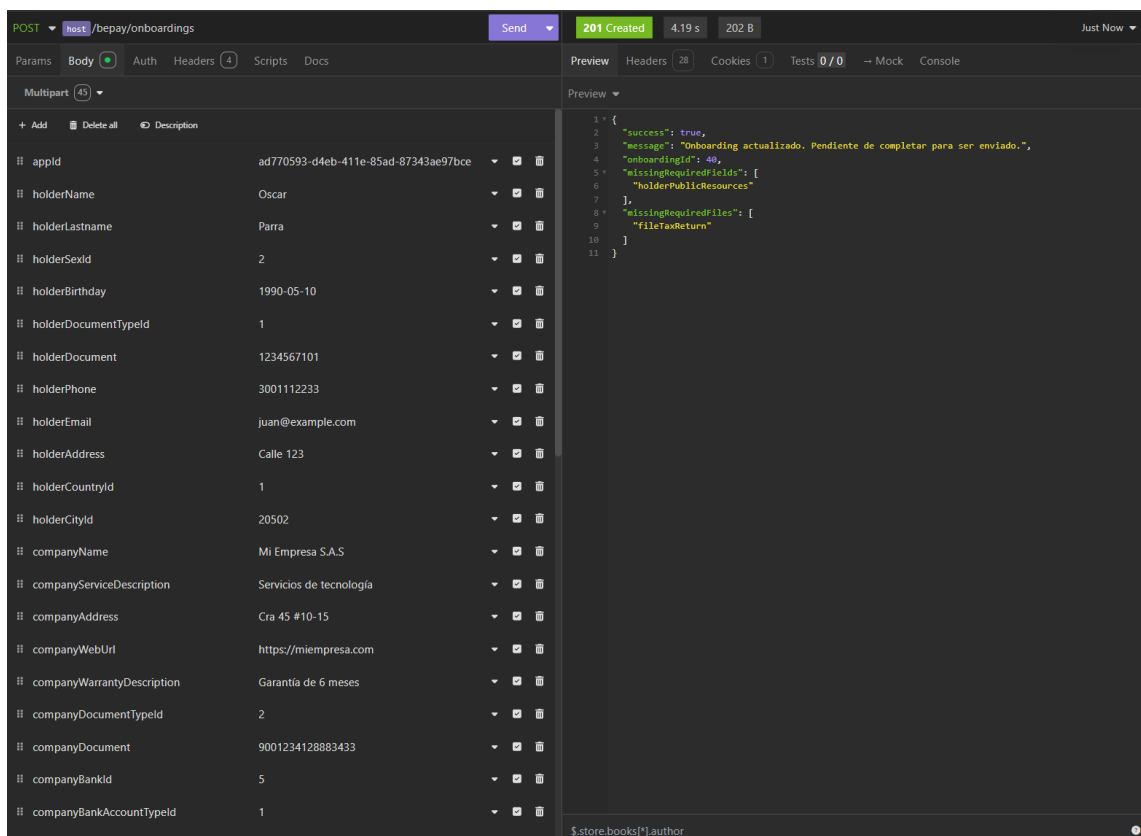


Figura 29: Prueba del endpoint de guardado parcial del Onboarding, pendiente de completar campos requeridos (Contenido V2)

Cuando **faltan campos o archivos requeridos**, los datos se guardan correctamente en la base de datos, pero no se realiza el envío a BePay. La respuesta del servicio indica el listado de campos y archivos faltantes, lo que permite que el cliente continúe su proceso posteriormente sin pérdida de información. Este enfoque facilita un *onboarding paso a paso*, pensado para usuarios que pueden completar su registro en diferentes momentos.

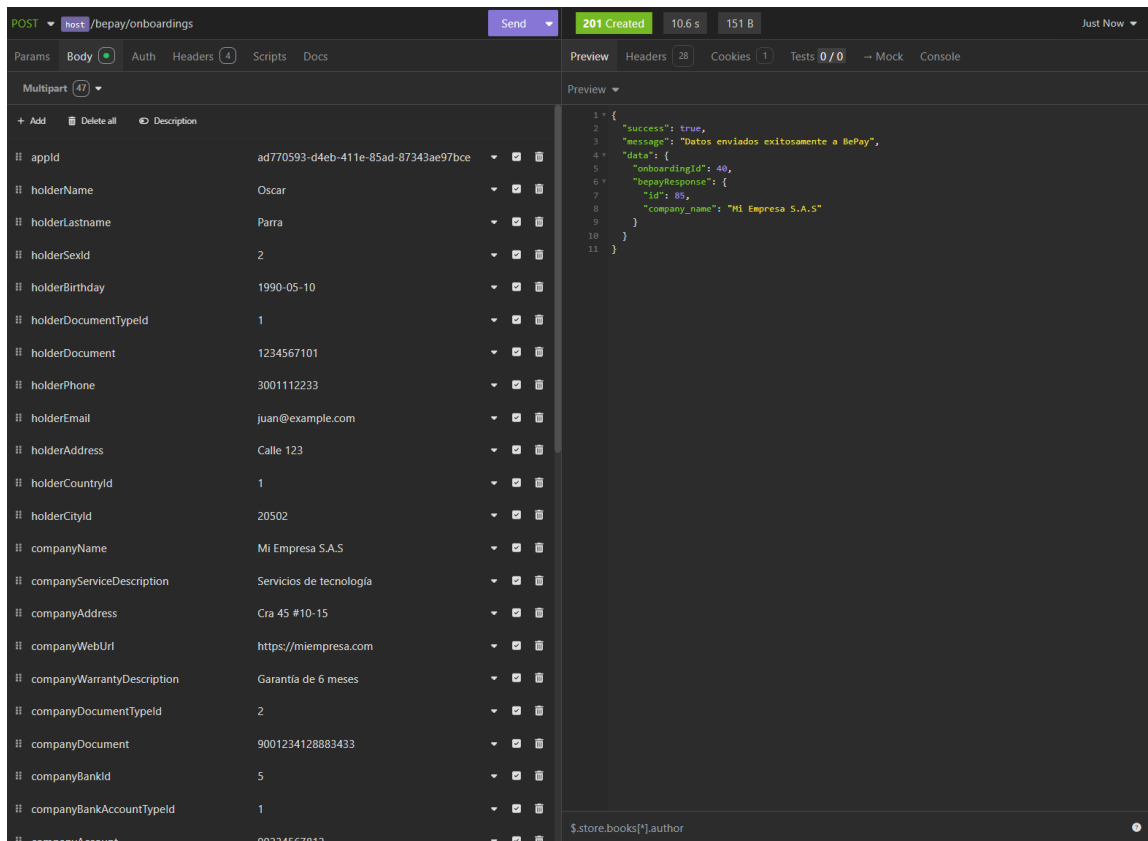


Figura 30: Prueba del endpoint de registro y envío de Onboarding completo hacia BePay (Contenido V2)

Cuando **todos los requisitos están completos**, el sistema obtiene el token de autenticación correspondiente a la aplicación, compila la información desde la base de datos y realiza el envío hacia la API de BePay (`/api/v1/Onboarding/signup`). Los datos y documentos se adjuntan en formato `multipart/form-data`, siguiendo la estructura exigida por BePay. Al completarse el envío, el estado del onboarding se actualiza como “enviado” y la respuesta del servicio incluye el resultado devuelto por BePay.

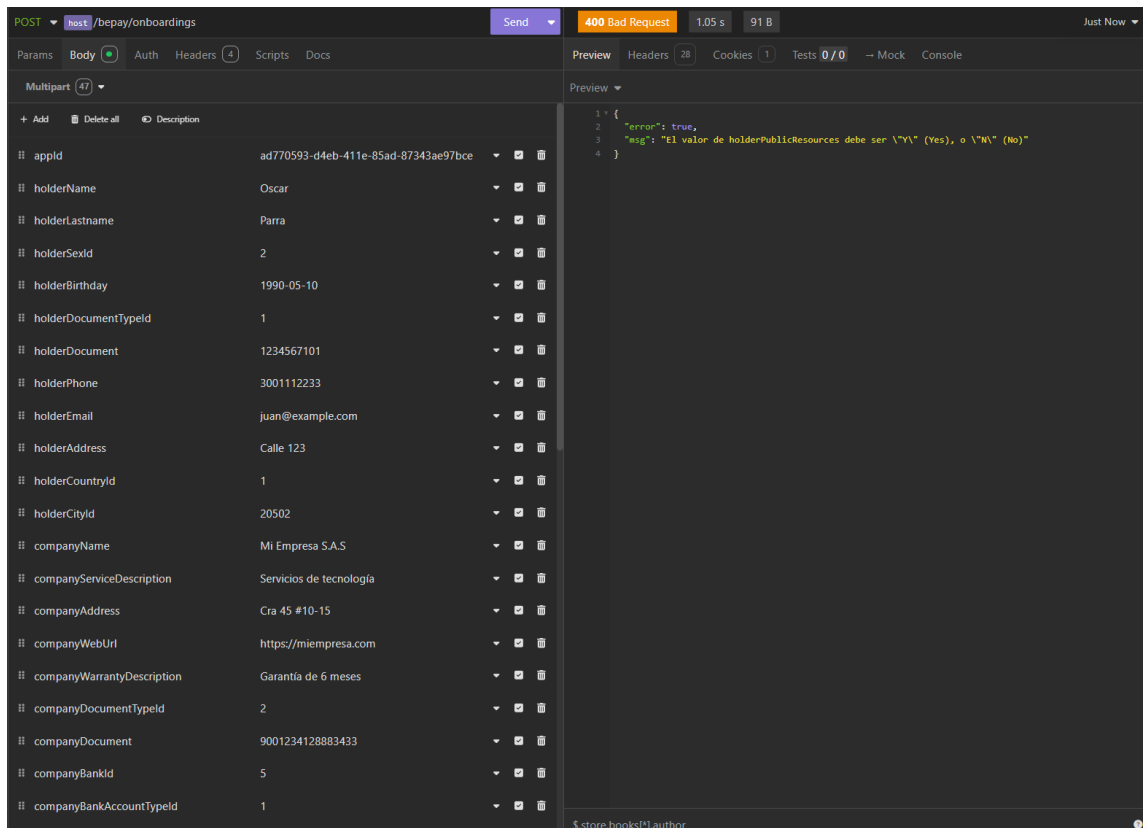


Figura 31: Ejemplo de estructura de campos booleanos enviados en formato aceptado por BePay (Y/N) (Contenido V2)

Algunos campos específicos, como `holderPep` o `holderPublicResources`, deben enviarse en el formato literal que BePay requiere, utilizando los valores "Y" o "N" para representar opciones booleanas. Este control garantiza la compatibilidad de los datos enviados y evita rechazos por formato inválido durante el procesamiento del onboarding.

Finalmente, tanto los guardados parciales como los envíos exitosos o con error son registrados mediante el servicio `CreateLogService`, manteniendo trazabilidad completa de cada evento: incluyendo el *payload*, la respuesta de BePay y el estado final del proceso. Este diseño asegura flexibilidad y robustez, permitiendo que cada cliente complete su proceso de registro de manera gradual, sin riesgo de pérdida de datos ni duplicaciones.

4.5. Reflexión Final

El desarrollo de los distintos módulos del sistema permitió comprender de manera profunda la relevancia de diseñar soluciones que no solo cumplan con los requerimientos inmediatos, sino que también se integren de forma coherente dentro de una arquitectura escalable y sostenible. Cada componente desde la autenticación hasta los procesos de onboarding e integración con servicios externos implicó analizar cómo la lógica de negocio, la persistencia de datos y la comunicación con terceros deben coexistir de forma segura, eficiente y mantenible.

Este proceso reforzó la importancia de aplicar principios sólidos de ingeniería de software, como la responsabilidad única, la reutilización de componentes y el manejo estructurado de errores. Asimismo, evidenció el valor de construir flujos flexibles que se adapten al usuario y a las necesidades reales del negocio, sin sacrificar la consistencia técnica ni la trazabilidad del sistema.

En definitiva, este desarrollo representó más que una implementación funcional: fue un ejercicio de diseño consciente, donde cada decisión técnica buscó sentar bases sólidas para la evolución futura del proyecto, manteniendo un equilibrio entre la complejidad tecnológica y la claridad en la experiencia de uso.

Referencias Bibliográficas

- [1] “INFORME DEL COMERCIO ELECTRÓNICO EN 2022 Y PERSPECTIVAS 2023,” visitado 20 de dic. de 2025. dirección: https://www.ccce.org.co/gestion_gremial/informe-del-comercio-electronico-en-2022-y-perspectivas-2023/.
- [2] M. D. Network, *JavaScript*, 2024. visitado 20 de oct. de 2025. dirección: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [3] atlassian. “Que es Scrum?” Visitado 20 de dic. de 2025. dirección: <https://www.atlassian.com/es/agile/scrum>.
- [4] Typescript. “Que es Typescript?” Visitado 20 de dic. de 2025. dirección: <https://www.typescriptlang.org/>.
- [5] NestJS. “NestJS – Un framework para crear aplicaciones del lado del servidor con Node.js y TypeScript,” visitado 20 de dic. de 2025. dirección: <https://nestjs.com>.
- [6] O. Foundation. “Node.js,” visitado 20 de dic. de 2025.
- [7] L. Torvalds. “Git - Sistema de Control de Versiones,” visitado 20 de dic. de 2025.
- [8] I. GitHub, *GitHub - Plataforma de Desarrollo Colaborativo*, <https://github.com>, 2008. visitado 20 de dic. de 2025.
- [9] atlassian. “¿Que es Jira?” Visitado 20 de dic. de 2025. dirección: <https://www.atlassian.com/es/software/jira/guides/getting-started/introduction#what-is-jira-software>.
- [10] atlassian. “¿Que es Microsoft Teams?” Visitado 20 de dic. de 2025. dirección: <https://support.microsoft.com/es-es/office/introducci%C3%B3n-a-microsoft-teams-b98d533f-118e-4bae-bf44-3df2470c2b12>.
- [11] C. C. de Comercio Electrónico. “Tendencias en Ecommerce para 2024 que impulsarán las ventas del sector retail,” visitado 20 de dic. de 2025. dirección: <https://www.ccce.org.co/noticias/tendencias-en-ecommerce-para-2024-que-impulsaran-las-ventas-del-sector-retail/>.
- [12] O. eCommerce. “Observatorio eCommerce,” visitado 20 de dic. de 2025. dirección: https://observatorioecommerce.mintic.gov.co/797/articles-338172_recurso_1.pdf.
- [13] T. Buitrago y J. G. Restrepo. “Origen y evolución del E-commerce en el mundo,” visitado 20 de dic. de 2025. dirección: <https://www.pymas.com.co/ideas-para-crecer/mundo-pyme/infografia-evolucion-comercio-electronico>.
- [14] “Migración de datos: definición, desafíos y mejores prácticas para afrontarla,” visitado 20 de dic. de 2025. dirección: <https://www.powerdata.es/migracion-de-datos>.
- [15] P. G. D. Group. “PostgreSQL: The world’s most advanced open source relational database,” visitado 20 de dic. de 2025. dirección: <https://www.postgresql.org>.
- [16] SmartBear Software. “Swagger OpenAPI Specification.” Estándar para la documentación de la API, visitado 20 de dic. de 2025. dirección: <https://swagger.io/specification/>.