

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SISTEMA PARA LA GESTIÓN Y CONTROL DE PEDIDOS EN LA EMPRESA PRIMA S.A.

DESARROLLO DE UN BACKEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

MATHÍAS AGUSTÍN TERÁN ALCÍVAR

mathias.teran@epn.edu.ec

DIRECTOR: ING. BYRON GUSTAVO LOARTE CAJAMARCA

byron.loarteb@epn.edu.ec

DMQ, julio 2025

CERTIFICACIONES

Yo, **MATHÍAS AGUSTÍN TERÁN ALCÍVAR** declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

MATHÍAS AGUSTÍN TERÁN ALCÍVAR

mathias.teran@epn.edu.ec

Certifico que el presente trabajo de integración curricular fue desarrollado por Mathías Agustín Terán Alcívar, bajo mi supervisión.

Ing. BYRON LOARTE

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

MATHÍAS AGUSTÍN TERÁN ALCÍVAR

DEDICATORIA

A mis queridos padres, quienes, con su amor incondicional, su sacrificio incansable y su apoyo constante, me brindaron las bases para alcanzar cada meta. Su fe en mí fue el motor que impulsó este camino.

A toda mi familia, por ser mi refugio y mi fuente de inspiración. Este logro, que hoy me llena de orgullo, es inseparable de su presencia y su aliento en cada paso de mi vida.

MATHÍAS AGUSTÍN TERÁN ALCÍVAR

AGRADECIMIENTO

Agradezco profundamente a mis padres, por ser el pilar fundamental en mi vida. Su confianza y aliento fueron mi mayor motivación para culminar este trabajo.

A mi familia, por su comprensión y ánimo en cada etapa de este proceso.

Mi especial gratitud a mi tutor de tesis, el Ing. Byron Loarte por su invaluable guía, paciencia y conocimiento, ya que su orientación fue clave para el desarrollo y la culminación exitosa de este proyecto. Además, este proyecto de Integración Curricular ha representado un desafío significativo que me ha permitido crecer profesional y personalmente.

Por último, extendiendo mi gratitud a todos quienes, de una u otra forma, contribuyeron con su conocimiento y acompañamiento para hacer realidad este proyecto.

MATHÍAS AGUSTÍN TERÁN ALCÍVAR

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VI
ABSTRACT	VII
1 Descripción del componente desarrollado	1
1.1 Objetivo general	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco teórico.....	3
2 METODOLOGÍA.....	6
2.1 Metodología de Desarrollo	6
Roles.....	7
Artefactos	8
2.2 Diseño de la arquitectura.....	10
Patrón arquitectónico.....	10
2.3 Herramientas.....	11
Librerías	12
3 RESULTADOS	14
Sprint 0 – Configuración del entorno de desarrollo.....	14
Sprint 1 – Codificación de las funcionalidades para el rol Administrador.	17
Sprint 2 – Codificación de las funcionalidades para el rol Vendedor.....	23
Sprint 3 - Pruebas.....	26
Sprint 4 - Despliegue.	29
4 CONCLUSIONES	31
5 RECOMENDACIONES.....	32
6 REFERENCIAS BIBLIOGRÁFICAS	33
7 ANEXOS.....	38
ANEXO I.....	39
ANEXO II.....	40
ANEXO III.....	67
ANEXO IV	68

RESUMEN

Promoción Internacional y Materiales PRIMA S.A., conocida como “La Casa del Pintor” es una empresa ecuatoriana dedicada a la comercialización de artículos de pintura y herramientas que enfrenta serios inconvenientes en su proceso comercial debido a una gestión de ventas completamente manual. Los vendedores registran pedidos en hojas de papel, lo que genera dificultades administrativas, técnicas, y una alta probabilidad de errores humanos como transcripciones incorrectas de códigos, cantidades o datos de clientes. Además, esta ineficiencia operativa provoca demoras, una carga administrativa excesiva y, fundamentalmente, una insatisfacción del cliente por la lentitud del servicio y los errores recurrentes en los pedidos.

Para abordar esta problemática, el presente proyecto propone el desarrollo de un backend enfocado en la gestión y control de pedidos, vendedores y productos para la empresa PRIMA S.A. Además, esta solución informática gestiona el registro, actualización, búsqueda y eliminación de dicha información, optimizando la consulta de disponibilidad de productos y ofreciendo un seguimiento en tiempo real del estado de cada pedido. Asimismo, el backend centraliza el manejo de la información y su disponibilidad a través de endpoints públicos y privados según roles, asegurando de esta manera, la confidencialidad e integridad de los datos con el objetivo de minimizar errores humanos y mejorar tanto la experiencia del cliente como la productividad de los vendedores.

El documento se encuentra estructurado de la siguiente manera el Capítulo 1, da una introducción a la empresa, su problemática, objetivos del proyecto, el alcance y el marco teórico respectivo. El Capítulo 2, detalla la técnica de investigación, la metodología de desarrollo ágil Scrum, el diseño de la arquitectura MVC, las librerías y herramientas que se han utilizado. Posteriormente, el Capítulo 3 presenta los entregables de cada sprint, desde la configuración del entorno, el desarrollo de funcionalidades para los roles de administrador y vendedor, hasta las pruebas y despliegue a producción. Por último, el Capítulo 4 se exponen las conclusiones del proyecto y finalmente, el Capítulo 5 las recomendaciones para futuras mejoras y mantenimiento del backend.

PALABRAS CLAVE: Backend, Gestión de Pedidos, API REST, Scrum, Nodejs, Express.js, MongoDB, PRIMA S.A.

ABSTRACT

Promoción Internacional y Materiales PRIMA S.A., known as “La Casa del Pintor”, is an Ecuadorian company dedicated to the commercialization of paint and tools that faces serious problems in its commercial process due to a completely manual sales management. Salespeople register orders on sheets of paper, which generates administrative and technical difficulties, and a high probability of human errors such as incorrect transcriptions of codes, quantities or customer data. In addition, this operational inefficiency leads to delays, excessive administrative burden and, fundamentally, customer dissatisfaction due to slow service and recurring order errors.

To address this problem, this project proposes the development of a backend focused on the management and control of orders, vendors and products for the company PRIMA S.A. In addition, this software solution manages the registration, updating, search and deletion of such information, optimizing the consultation of product availability and providing real-time monitoring of the status of each order. Likewise, the backend centralizes the management of information and its availability through public and private endpoints according to roles, thus ensuring the confidentiality and integrity of the data with the objective of minimizing human errors and improving both the customer experience and the salespersons' productivity.

The document is structured as follows: Chapter 1 gives an introduction to the company, its problems, project objectives, scope and the respective theoretical framework. Chapter 2 details the research technique, the Scrum agile development methodology, the design of the MVC architecture, the libraries and tools that have been used. Subsequently, Chapter 3 presents the deliverables of each sprint, from the configuration of the environment, the development of functionalities for the administrator and vendor roles, to the testing and deployment to production. Finally, Chapter 4 presents the conclusions of the project and finally, Chapter 5 presents the recommendations for future improvements and maintenance of the backend.

KEY WORDS: Backend, Order Management, API REST, Scrum, Nodejs, Express.js, MongoDB, PRIMA S.A.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Promoción Internacional y Materiales PRIMA S.A., conocida comercialmente como “La Casa del Pintor”, con sede en Ecuador, esta empresa centra su actividad en la comercialización de artículos de pintura y herramientas para trabajos de infraestructura en el hogar. No obstante, enfrenta varios inconvenientes en su proceso comercial, especialmente en la gestión de ventas, la cual se realiza totalmente de manera manual ya que los vendedores registran cada pedido del cliente en hojas de papel para su posterior procesamiento lo que ocasiona una serie de dificultades tanto administrativas como técnicas, además de una alta probabilidad de errores humanos, como la transcripción incorrecta de códigos de productos, cantidades o datos de clientes, lo que puede derivar en entregas equivocadas de los productos. Esta ineficiencia operativa, además de crear fricciones y demoras en la cadena de suministro interna, impone una carga administrativa excesiva sobre el equipo de ventas, desviando su tiempo de tareas productivas y, consecuentemente, impactando negativamente la satisfacción del cliente debido a la lentitud del servicio y los errores recurrentes en los pedidos.

En una era completamente digital, la implementación de tecnologías informáticas para el sector comercial y empresarial se ha vuelto esencial para conseguir una mayor efectividad en los procedimientos y, paralelamente, una gestión de datos más adecuada. Además, no solo facilita la automatización de tareas para los empleados, sino que también permite a los administradores llevar a cabo los procesos internos de la empresa de manera más efectiva, lo que resulta en una gestión más precisa y optimizada [1].

En base a lo mencionado anteriormente en este proyecto de Integración Curricular contempla la elaboración de un backend destinado a la empresa PRIMA S.A. una solución informática enfocada en la gestión de pedidos, vendedores y productos. Este backend se encarga de gestionar toda la información relacionada con el registro, actualización, búsqueda y eliminación de pedidos, vendedores y productos. Además, optimiza la consulta y disponibilidad de productos proporcionando un seguimiento en tiempo real del estado de cada pedido, con el objetivo de minimizar los errores humanos. Adicional a ello, el componente backend centraliza el manejo de toda la información, facilitando su acceso a través de una serie de endpoints públicos y privados, permitiendo a los diferentes roles acceder a ella asegurando así la confidencialidad e integridad de los datos. Con la implementación de esta solución, la empresa no solo se anticipa a las nuevas demandas del mercado ya que el definir nuevas directrices estratégicas como integrar una herramienta tecnológica como parte de sus procesos, no solo optimiza la experiencia del cliente durante la compra, sino también incrementa la productividad de cada uno de los vendedores en la empresa.

1.1 Objetivo general

Desarrollar el backend para la gestión y control de pedidos en la empresa PRIMA S.A.

1.2 Objetivos específicos

1. Definir los requerimientos para el componente backend.
2. Diseñar el modelo de base de datos en base a los requisitos.
3. Desarrollar cada uno de los endpoints (públicos y privados) para el componente backend.
4. Realizar pruebas para probar cada uno de los endpoints.
5. Desplegar el componente backend a un entorno de producción para su consumo.

1.3 Alcance

Definir claramente el alcance es vital en el desarrollo de software, ya que ayuda a prevenir retrasos, cambios inesperados o dificultades que puedan obstaculizar la consecución de las metas definidas [2]. En ese sentido, el alcance para este proyecto de Integración Curricular es desarrollar para la empresa PRIMA S.A. un componente backend que les permita en tiempo real y de forma organizada gestionar el proceso de pedidos, así como la consulta de clientes, la disponibilidad de productos y el monitoreo de cada pedido.

El primer paso para llevar el backend a producción es identificar claramente el problema, los objetivos y un análisis completo de todos los requisitos, empleando para ello Scrum, el cual es una metodología que permite implementar todo el proceso de desarrollo de forma organizada y en base a pequeños entregables asignados a cada miembro del equipo. En cuanto a la etapa de codificación, se usa Express.JS junto con una estructura MVC (Modelo-Vista-Controlador), lo que permite organizar de mejor manera el código y una fácil integración con Base de datos de tipo NoSQL. Posterior a la etapa de codificación, con el objetivo de constatar el funcionamiento correcto de todos los módulos, se ejecutan una serie de pruebas y como etapa final se despliega el backend a un entorno de producción, asegurando de esta manera un buen funcionamiento y una adecuada conexión con los otros componentes.

El backend opera con dos roles principales, y cada uno tiene funcionalidades distintas, las cuales se detallan a continuación.

Endpoints para el rol Administrador

- Inicio de sesión.
- Gestión completa de vendedores (crear, modificar, eliminar, listar).
- Visualización y seguimiento de pedidos.
- Administración de estado de pedidos.
- Generación de reportes de ventas.
- Gestión completa de productos y stock.
- Gestión completa de clientes (crear, modificar, eliminar, listar).

Endpoints para el rol Vendedor

- Inicio de sesión.
- Gestión de pedidos (crear, modificar, consultar).
- Consulta de disponibilidad de productos.
- Visualización del historial de pedidos.
- Actualización de estado de pedidos asignados.
- Gestión de clientes (Crear, Modificar, listar).

1.4 Marco teórico

Backend

Es el componente del sistema informático que opera en lado del servidor y que se encarga de gestionar las reglas del negocio, atender las solicitudes del cliente, así como la comunicación con la Base de datos. Además, el backend funciona en segundo plano, por lo que no es visible para el usuario final [3].

API REST

Son un conjunto de normas para la comunicación entre aplicaciones que operan sobre el protocolo HTTP, empleando los métodos estándar (GET, POST, PUT PATCH, DELETE) y códigos de estado para manipular los recursos de un servidor. Además, son importantes para el intercambio de información entre la capa de presentación (frontend) y la lógica del servidor (backend) [4].

MongoDB

Es un administrador para bases de datos de tipo NoSQL basado en documentos, en el cual la información se almacena en un formato binario denominado BSON. Además, se destaca por ofrecer un rendimiento rápido en operaciones de lectura y escritura, lo que la convierte en una opción ideal para aplicaciones que necesitan procesar grandes volúmenes de datos cuyos esquemas no son fijos [5].

JavaScript

Es frecuentemente utilizado en la programación de páginas web, volviéndolas interactivas; pero su alta popularidad y adaptabilidad permiten que se lo empiece a utilizar en el desarrollo de programas del lado del servidor, extendiendo de esta manera su campo de uso en diversas soluciones [6].

Node.js

Es un conjunto de módulos que logran ejecutar código de JavaScript en el lado del servidor; también permite programar diversos tipos de aplicaciones, ya que funciona con un mecanismo de entrada y salida que no bloquea y es reactivo a eventos. Sin duda estos puntos clave lo vuelven una alternativa adecuada y eficiente para aplicaciones escalables [7].

Json Web Token

Es un mecanismo que permite generar tokens de acceso en formato json, los cuales facilitan compartir información en aplicaciones. También los tokens contienen información necesaria que permiten identificar a un usuario, logrando así métodos de autenticación escalables y seguros [8].

ODM (Object Document Mapper)

Es un mecanismo que permite interactuar con bases de datos NoSQL, usando objetos en JavaScript. Además, su principal particularidad es que mapea documentos y los transforma a una serie de clases, facilitando usar operaciones CRUD (crear, leer, actualizar y eliminar) de una manera mucho más estructurada y orientada a objetos [9].

Express

Es un ambiente de trabajo que opera con Node.js y que es muy utilizado para desarrollar aplicaciones webs. Además, cuenta con una serie de atributos que aceleran y facilitan la creación de aplicaciones webs, aplicaciones backend, entre otros [10].

Insomnia

Es una aplicación de código abierto, diseñada para simplificar las pruebas y gestión de APIs. Además, ofrece una interfaz intuitiva que facilita el operar con solicitudes HTTP, REST, entre otros, clasificándose como un cliente REST [11].

Render

Es un servicio que trabaja en la nube y que está construido para poder optimizar la puesta en producción y el alojamiento de aplicaciones web, además, permite importar los proyectos directamente desde GitHub. Pero su enfoque está en la escalabilidad y en su facilidad de uso, convirtiéndolo en una opción atractiva para proyectos pequeños e incluso aplicaciones empresariales [12].

2 METODOLOGÍA

Como técnica de investigación el estudio de casos facilita el análisis profundo de un fenómeno utilizando para ello métodos cualitativos como cuantitativos. Además, gracias a estos métodos se logra una precisa recopilación y el procesamiento de la información buscando alcanzar una visión más nítida del objeto de estudio y de las alternativas viables para su solución [13].

Para este proyecto de Integración Curricular se ha seleccionado el estudio de casos como técnica de investigación, ya que la misma, simplifica el entendimiento del tema investigado mediante la recolección de información de múltiples orígenes. Es así como, esta técnica ha permitido recopilar de forma clara y precisa toda la problemática que presenta los vendedores respecto a la gestión de pedidos, clientes y productos, así como la implementación de tecnologías informáticas para el sector comercial y empresarial se han vuelto esenciales para potenciar la operatividad de los procesos. De este modo, se asegura que el desarrollo de este componente backend sea efectivo y esté directamente alineado con los objetivos que se han planteado y sobre todo que los resultados sirvan de base para futuras iniciativas similares.

2.1 Metodología de Desarrollo

Dentro del campo del desarrollo de software, una metodología opera como un marco de trabajo estructurado, que prescribe un conjunto específico de técnicas y procedimientos en la creación de sistemas informáticos. Además, una metodología es una decisión estratégica, ya que la misma debe estar orientada a lograr optimizar la organización y aumentar la eficiencia del proyecto; también debe lograr minimizar aquellos errores que pueden surgir a lo largo del ciclo del software [14].

Las metodologías ágiles son un pilar fundamental, ya que se destacan por tener un enfoque centrado en entregar continuamente productos funcionales, que son capaces de adaptarse de manera efectiva a los requerimientos del cliente. También la comunicación constante y la flexibilidad entre los involucrados permite agregar de manera oportuna nuevos requisitos contribuyendo de esta manera a obtener un producto final de mayor calidad y alineado a las necesidades del negocio [15].

Scrum como metodología ágil es ampliamente utilizada, se centra en entregar avances de manera incremental mediante ciclos de trabajo, también llamados Sprints. Además, se basa en la colaboración y en la adaptabilidad para una mejora continua mediante el uso de roles y artefactos que son elementos clave para poder organizar el trabajo [16]. Por lo antes dicho, se ha seleccionado Scrum para guiar el desarrollo del presente componente backend debido a su enfoque iterativo y flexibilidad ante modificaciones que puedan surgir

durante las fases de diseño, codificación, pruebas o despliegue tal y como se detalla a continuación su implementación.

Roles

Los roles son importantes en Scrum ya que otorgan responsabilidades para lograr un flujo ágil de trabajo. Además, cada rol posee un método de acción clara y complementaria [17]. De esta manera, se presenta a continuación los miembros que asumen cada uno de los roles durante el desarrollo del componente backend.

Product Owner

Es quien tiene la responsabilidad del producto desde la perspectiva del negocio y los diferentes elementos comerciales del proyecto. Además, su responsabilidad principal es definir claramente que se debe construir y priorizar cada una de las funcionalidades a través de una lista con el objetivo de maximizar el valor de entrega [17]. Por lo antes dicho, la **Tabla 2.1** contiene al encargado de asumir este rol en el desarrollo del backend.

Scrum Master

Actúa como un facilitador que tiene experticia en la metodología Scrum. Además, en lugar de ser un jefe tradicional, se comporta como un líder cortés que se esfuerza por asegurar que el equipo comprenda y emplee adecuadamente las buenas prácticas de Scrum. Asimismo, sus responsabilidades incluyen colaborar con el Product Owner, ayudar al equipo con asesorías, capacitaciones y mantener actualizado el tablero de trabajo [17]. Por lo antes dicho, la **Tabla 2.1** contiene al encargado de asumir este rol en el desarrollo del backend.

Development Team

Es un grupo de personas organizadas que se encargan de transformar cada requisito en una mini versión del producto final. Además, se organizan para determinar la mejor manera de realizar el trabajo dividiendo cada una de las tareas de manera eficiente y colaborando estrechamente para asegurar la calidad y el cumplimiento de los objetivos dentro de los plazos establecidos [17]. Por lo antes dicho, la **Tabla 2.1** contiene al encargado de asumir este rol en el desarrollo del backend.

Tabla 2.1 Designación de Roles.

ROL	INTEGRANTES
Product Owner	Lcda. Alexandra Remache.
Scrum Master	Ing. Byron Loarte, MSc.

Development Team	Sr. Mathías Agustín Terán Alcívar.
------------------	------------------------------------

Artefactos

Los artefactos de Scrum son componentes esenciales como parte del trabajo ágil, ya que los mismos facilitan la transparencia y estandarización de la información. Además, hacen posible que la totalidad del equipo mantenga una misma perspectiva sobre el estado de avance del proyecto garantizando de esta manera que el producto final cumpla con los requisitos definidos y aporte valor al cliente [18]. En ese sentido, se presenta los artefactos que se han utilizado para el registro de los requisitos de la empresa PRIMA S.A.

Recopilación de Requerimientos

Es un artefacto que permite documentar meticulosamente todos los requisitos y necesidades asociados al negocio. Además, establece una plataforma confiable para planificar y llevar a cabo el desarrollo del producto, asegurando de esta manera que cada funcionalidad esté alineada con los objetivos estratégicos del negocio [19]. La **Tabla 2.2** detalla el levantamiento de un requisito, mientras que la tabla en su versión extendida se encuentra disponible dentro del **ANEXO II**.

Tabla 2.2 Levantamiento de requerimientos para el componente backend.

ID - RR	ENUNCIADO DEL ITEM
RR-003	<p>Para el usuario vendedor se requiere generar endpoints para:</p> <ul style="list-style-type: none"> • Visualizar productos. • Gestionar clientes. • Gestionar órdenes de compra.

Historias de Usuario

Explica una funcionalidad que el software debe realizar utilizando una descripción breve por medio de un lenguaje sencillo y accesible. Además, su objetivo fundamental es comunicar claramente el valor o beneficio específico que esa característica aporta al usuario final mediante el uso del software [20]. La **Tabla 2.3** detalla la descripción de la funcionalidad de gestión de órdenes de compra, mientras que las demás tablas completas se encuentran disponibles dentro del **ANEXO II**.

Tabla 2.3 Historia de usuario para órdenes de compra.

HISTORIA DE USUARIO	
Identificador: HU-009	Usuario: Vendedor
Nombre historia: Gestionar órdenes de compra.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
Descripción: El vendedor utiliza varios endpoints, facilitados por el componente backend, con el fin de: <ul style="list-style-type: none"> • Registrar órdenes. • Visualizar órdenes. • Visualizar órdenes por ID. • Actualizar órdenes. 	
Observación: El uso de los endpoints descritos está limitado al rol de vendedor, y se implementan controles de acceso adicionales para que solo ellos puedan administrar la información referente a las órdenes de compra.	

Product Backlog

Se define como una colección sistemática de todos los requisitos que han sido recopilados previamente. Además, esta colección se presenta en un orden que refleja su prioridad y valor que aporta para el negocio [21]. La **Tabla 2.4** detalla el registro de un requisito con su estado, prioridad e iteración asignada, mientras que la tabla en su versión extendida se encuentra disponible dentro del **ANEXO II**.

Tabla 2.4 Product Backlog.

HISTORIA DE USUARIO	DESCRIPCIÓN DE LA HISTORIA	ITERACIÓN ASIGNADA	ESTADO	PRIORIDAD
HU-008	Gestionar clientes	2	Finalizado	Alta

Sprint Backlog

Este artefacto permite que los equipos puedan crear y gestionar sus planes de trabajo para cada iteración. Además, estos planes se descomponen en varias tareas específicas que deben cumplirse dentro del Sprint, con el objetivo de asegurar que el desarrollo de los objetivos definidos para esa iteración sea exitoso [22]. La **Tabla 2.5** detalla la descripción de la cuarta iteración, mientras que la tabla en su versión extendida se encuentra disponible dentro del **ANEXO II**.

Tabla 2.5 Sprint Backlog.

SPRINT BACKLOG						
SB	NOMBRE	COMPONENTE	HU	HISTORIA DE USUARIO	TAREA	PLAZO PREVISTO
SB-004	Despliegue del backend	<ul style="list-style-type: none">Despliegue en la plataforma Render				10H

2.2 Diseño de la arquitectura

En la construcción de aplicaciones de mediana y gran complejidad, la arquitectura proporciona el soporte fundamental para todos los componentes. Además, su elección se debe realizar de forma estratégica antes de escribir el código con el objetivo de evitar problemas de escalabilidad y es por esta razón que una buena arquitectura asegura que el sistema sea flexible, modular y capaz de adaptarse a cambios futuros, permitiendo a los equipos de desarrollo integrar nuevas funcionalidades sin afectar la estabilidad del sistema software [23]. Con respecto al componente backend el patrón que se ha establecido es el siguiente.

Patrón arquitectónico

Este patrón posee un enfoque estructural clave en el desarrollo de software, el cual descompone el sistema en tres componentes interconectados con responsabilidades distintas, por ejemplo, el Modelo se encarga de los datos, la vista se ocupa de como se muestra la información al usuario, y el controlador es quien dirige las respuestas a las acciones del usuario. De esta forma el sistema es modular y escalable para nuevas funcionalidades que se vayan integrando [24]. En la **Figura 2.1** se muestra la arquitectura que se ha utilizado para el componente backend con cada uno de sus elementos.

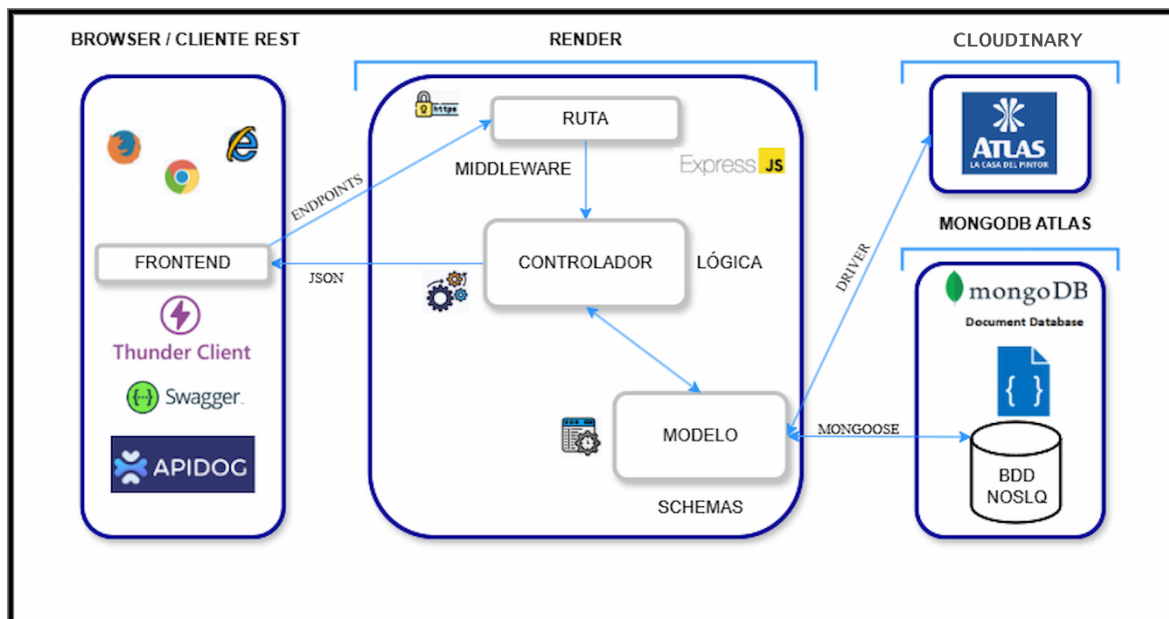


Figura 2.1 Arquitectura para el backend.

2.3 Herramientas

En el ámbito del desarrollo, las herramientas son un conjunto de elementos vitales que permiten aumentar la productividad y efectividad de los programadores, al crear y gestionar sistemas software. Además, son un abanico de recursos tecnológicos que está en constante expansión, ya que van desde entornos de desarrollo integrado hasta herramientas de automatización y de pruebas [25]. Por lo antes mencionado, la **Tabla 2.6** muestra un registro de las herramientas que se han usado para desarrollar el componente backend.

Tabla 2.6 Registro de herramientas.

HERRAMIENTA	JUSTIFICACIÓN
GitHub	Es una plataforma que permite gestionar de manera centralizada el código fuente del backend en la nube, junto con las capacidades de controlar versiones, resulta fundamental para trabajar de manera eficiente y organizada. Además ha permitido visualizar de manera sistemática y estructurada el código del componente backend [26].
Visual Studio Code	Gracias a la popularidad entre programadores y a su amplia disponibilidad de extensiones especializadas, ha permitido agilizar de manera efectiva el proceso de desarrollo y corrección de errores en el backend [27].

Insomnia	Ya que permite comunicarse con APIs externas a través de peticiones, la ha convertido en una herramienta fundamentalmente valiosa para llevar a cabo pruebas de manera fácil y efectiva en el componente backend [28].
Express	Tener el respaldo de una comunidad activa, ha facilitado el entendimiento de este Framework que ha logrado simplificar de manera significativa el proceso de creación del componente backend, permitiendo una rápida aplicación de funcionalidades [29].
Render	Desplegar el proyecto en un entorno de producción es sumamente importante. Además, existen varias herramientas que permiten realizar esta acción, pero Render ha permitido desplegar el proyecto backend de forma fácil y ágil [30].
MongoDB	La estructuración de datos en colecciones y documentos que emplea esta Base de datos NoSQL, hace que la lectura por parte del componente backend sea un proceso más directo y eficiente [31].

Librerías

Son módulos de código reutilizables que aíslan soluciones para tareas específicas y recurrentes. Además, aprovechan una gama de librerías que aceleran significativamente la creación de sistemas software, evitando realizar funcionalidades desde el inicio [32]. Gracias a lo mencionado anteriormente, la **Tabla 2.7** muestra las librerías principales que han sido usadas para el desarrollo del componente backend.

Tabla 2.7 Registro de librerías.

LIBRERÍA	DESCRIPCIÓN
Bcryptjs	Asegura criptográficamente las contraseñas [33].
Cors	Regula acceso a los endpoints de la API desde orígenes distintos al propio servidor backend [34].

Dotenv	Permite cargar de manera sencilla las variables de entorno [35].
Nodemailer	Permite que el servidor envíe correos electrónicos a los usuarios de forma sencilla y eficiente [36].
Mongoose	Permite trabajar con los datos alojados en MongoDB por medio de modelos [37].
JSON Web Token	Asegura la transmisión fiable de información [38].

3 RESULTADOS

A continuación, se presentan los resultados que se han conseguido en el proyecto, específicamente en lo que respecta al desarrollo del backend. Además, se detalla el funcionamiento y pruebas unitarias para cada uno de los endpoints, así como las tareas que dispone cada iteración con el objetivo de realizar entregables funcionales en cada avance del proyecto. En ese contexto, se detallan los entregables para cada Sprint.

Sprint 0 – Configuración del entorno de desarrollo.

La fase inicial para el desarrollo del componente backend requiere la configuración del entorno de trabajo y otras tareas asociadas con el objetivo prevenir inconvenientes en las etapas posteriores. En ese sentido, las tareas para este ciclo de trabajo abarcan lo siguiente:

- Establecer especificaciones a cumplir.
- Definir la arquitectura de carpetas y archivos para el proyecto.
- Modelado de la estructura de datos para las colecciones en MongoDB.
- Determinación de permisos por cada rol.

Establecer especificaciones a cumplir

El backend dispone de un conjunto de rutas públicas y privadas

Mediante el uso de endpoints (públicos y privados), el componente backend logra una administración más precisa de los datos. De esta manera, se restringe el acceso a ciertas funcionalidades e información, logrando con ello, que solo los usuarios con una sesión activa y el rol correspondiente puedan acceder a cada una de las rutas privadas.

El backend permite la creación de cuentas y la autenticación de usuarios

El componente backend, a través de endpoints, ofrece un sistema de autenticación para los diferentes roles de usuario, facilitando de esta manera, el acceso a los módulos pertinentes. Además, las credenciales del administrador, por ejemplo, son provistas por el equipo de desarrollo y una vez que el administrador inicia sesión con ellas, puede realizar la creación de cuentas para usuarios vendedores y posterior a ello autenticarse utilizando los datos de acceso enviados a su correo electrónico. Por último, este flujo de trabajo ha sido realizado bajo estrictos requisitos del dueño del negocio.

El backend permite gestionar vendedores

El backend expone una serie de endpoints que permiten al rol de administrador realizar operaciones CRUD (Crear, Leer/Listar, Actualizar, Borrar) y la modificación del estado sobre los registros de vendedores. Además, esta funcionalidad se implementa mediante el

uso de verbos HTTP convencionales por parte del administrador, facilitando así, un manejo eficiente de la información de los vendedores.

El backend permite gestionar pedidos

Existen endpoints dedicados a la gestión de pedidos por parte del vendedor, el cual le permite ejecutar operaciones CRUD (Crear, Leer/Listar, Actualizar, Eliminar) y la modificación del estado relacionado a un cliente en específico. De manera complementaria, el administrador puede realizar consultas sobre el conjunto de pedidos que han sido creados por los vendedores y efectuar cambios únicamente en el estado de los mismos.

El backend permite gestionar clientes

El backend provee endpoints para que el administrador pueda ejecutar el ciclo completo de operaciones (Creación, Lectura/Listado, Actualización, Eliminación) para los clientes. Además, posee otros endpoints que están diseñados para que el vendedor pueda visualizar los datos de los clientes, con el fin de facilitar la creación de órdenes de compra.

El backend permite gestionar productos

El backend expone endpoints para que el administrador ejecute operaciones CRUD completas (Crear, Leer/Listar, Actualizar, Eliminar) sobre productos. Además, otros endpoints para verificar cantidades en stock.

El backend permite visualizar estadísticas

Para el administrador, el backend ofrece endpoints particulares que muestran información estadística clave tales como número de vendedores, productos, clientes y órdenes de compra que se han realizado. Además, de mostrar información sobre las ventas individuales de cada vendedor y un listado de los vendedores con mayores ventas.

Definir la arquitectura de carpetas y archivos para el proyecto

El desarrollo de cada uno de los endpoints para el backend, así como el desarrollo de cada uno de los módulos por cada rol, se ha realizado usando el entorno de desarrollo Visual Studio Code y complementariamente, JavaScript como lenguaje de programación base, el patrón arquitectónico Modelo-Vista-Controlador (MVC) y el Framework Express. Esta combinación de herramientas tecnológicas proporciona una base sólida, pero sobre todo escalabilidad si se requiere implementar nuevas funcionalidades [29]. A continuación, en la **Figura 3.1** se presenta la organización de archivos y directorios.

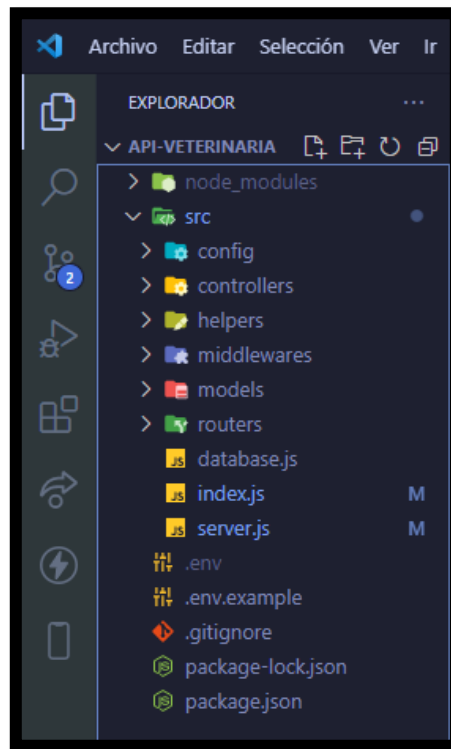


Figura 3.1 Esquema de archivos y directorios para el componente backend.

Modelado de la estructura de datos para las colecciones en MongoDB

Para manejar la estructura de datos (colecciones y documentos) del lado del servidor, se ha empleado MongoDB, el cual, como DBMS NoSQL, brinda soporte nativo para operaciones en tiempo real y el manejo eficiente de datos sin una estructura fija, lo cual es crucial para gestionar datos escalables, seguros y versátiles [5]. En la **Figura 3.2** se detalla cómo se han organizado las colecciones y en el **ANEXO II** se muestra en su totalidad la composición interna de los documentos.



Figura 3.2 Colecciones de la Base de datos (NoSQL).

Determinación de permisos por cada rol

El componente backend opera con 2 roles predefinidos y a cada rol se le han asociado privilegios y capacidades operativas específicas, garantizando así, un control efectivo de las funcionalidades y salvaguardar datos sensibles. Como consecuencia de esta configuración, la **Figura 3.3** presenta una descripción detallada de los permisos permitidos para cada usuario.



Figura 3.3. Roles de usuario.

Sprint 1 – Codificación de las funcionalidades para el rol Administrador.

El enfoque de este Sprint es la implementación de operaciones destinadas al perfil administrador, el cual incluye la gestión de vendedores, productos y clientes, así como los procesos de autenticación (inicio de sesión y recuperación de cuenta). En ese sentido, las tareas para este ciclo de trabajo abarcan lo siguiente:

- Creación de endpoints destinados a la funcionalidad de inicio de sesión.
- Creación de endpoints destinados a la funcionalidad de recuperar cuenta.
- Creación de endpoints destinados a la funcionalidad de vendedores.
- Creación de endpoints destinados a la funcionalidad de gestión de productos.
- Creación de endpoints destinados a la funcionalidad de clientes.

- Creación de endpoints destinados a la funcionalidad de órdenes de compra.
- Creación de endpoints destinados a la funcionalidad de visualización de analíticas.

Creación de endpoints destinados a la funcionalidad de inicio de sesión

El backend otorga varios endpoints y rutas específicas para que el usuario administrador pueda ingresar a los módulos asociados mediante el uso de credenciales de acceso las cuales se validan para todos los campos del formulario como se presenta en la **Figura 3.4** el correcto inicio de sesión del administrador, el resultado correspondiente a la prueba en la **Figura 3.5** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

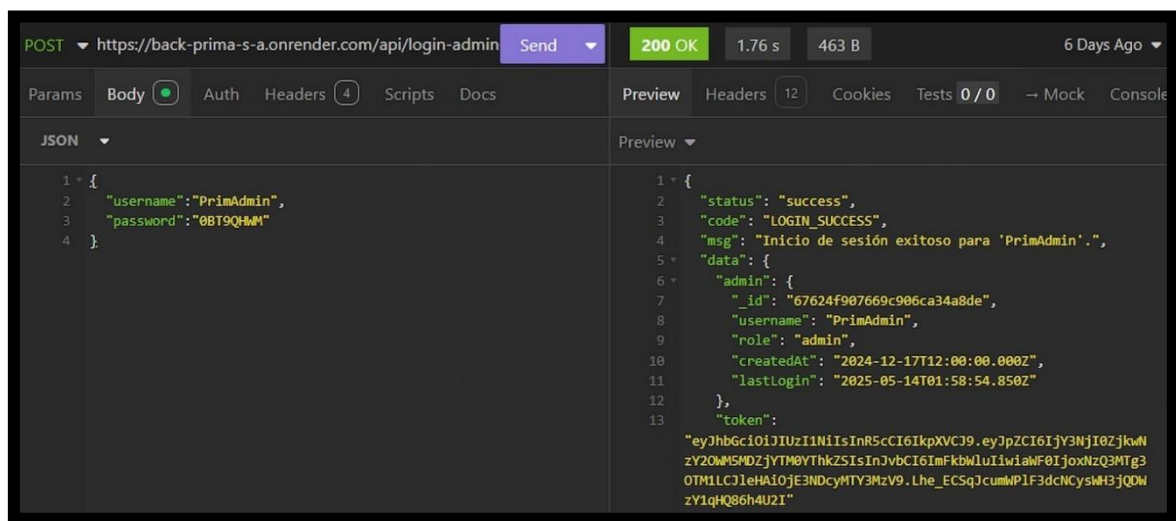


Figura 3.4 Inicio sesión del administrador.

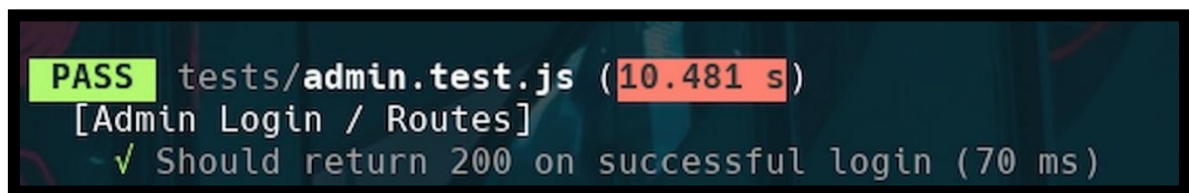


Figura 3.5 Test unitario – Inicio sesión del administrador.

Creación de endpoints destinados a la funcionalidad de recuperar cuenta

El componente backend provee un mecanismo para que el administrador recupere su contraseña a través de endpoints y rutas designadas. Además, este proceso se inicia cuando el administrador proporciona un nombre de usuario valido para que el backend genere una nueva contraseña y la cual sea enviada a la dirección de correo electrónico como se ilustra en la **Figura 3.6** el correo enviado, el resultado correspondiente a la prueba en la **Figura 3.7** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

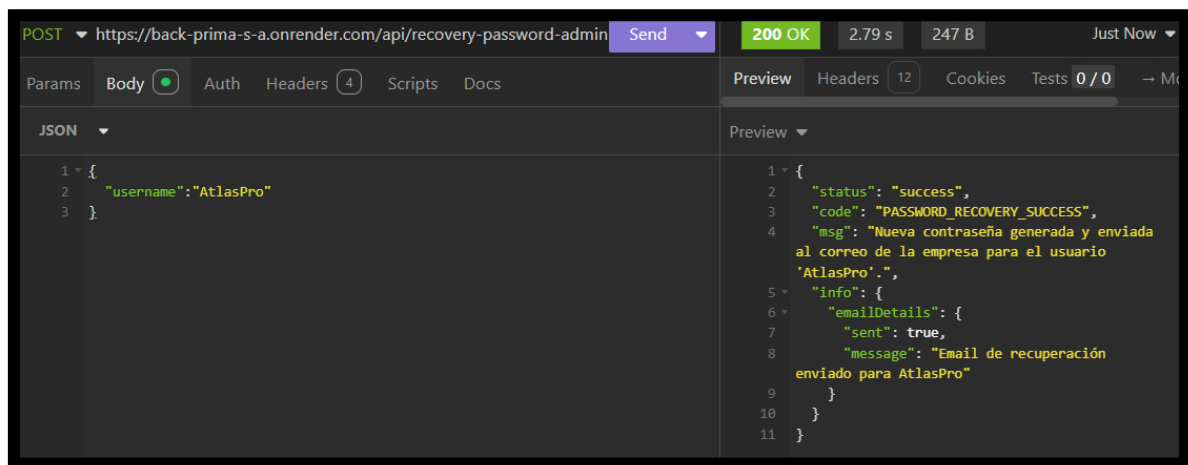


Figura 3.6 Recuperación contraseña.



Figura 3.7 Prueba unitaria - recuperación contraseña.

Creación de endpoints destinados a la funcionalidad de vendedores

El componente backend provee endpoints y rutas designadas para la gestión de vendedores por parte del administrador. Además, se implementa una validación de los datos de entrada previa a su procesamiento como se presenta en la **Figura 3.8** la inscripción de un nuevo vendedor, el resultado correspondiente a la prueba en la **Figura 3.9** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

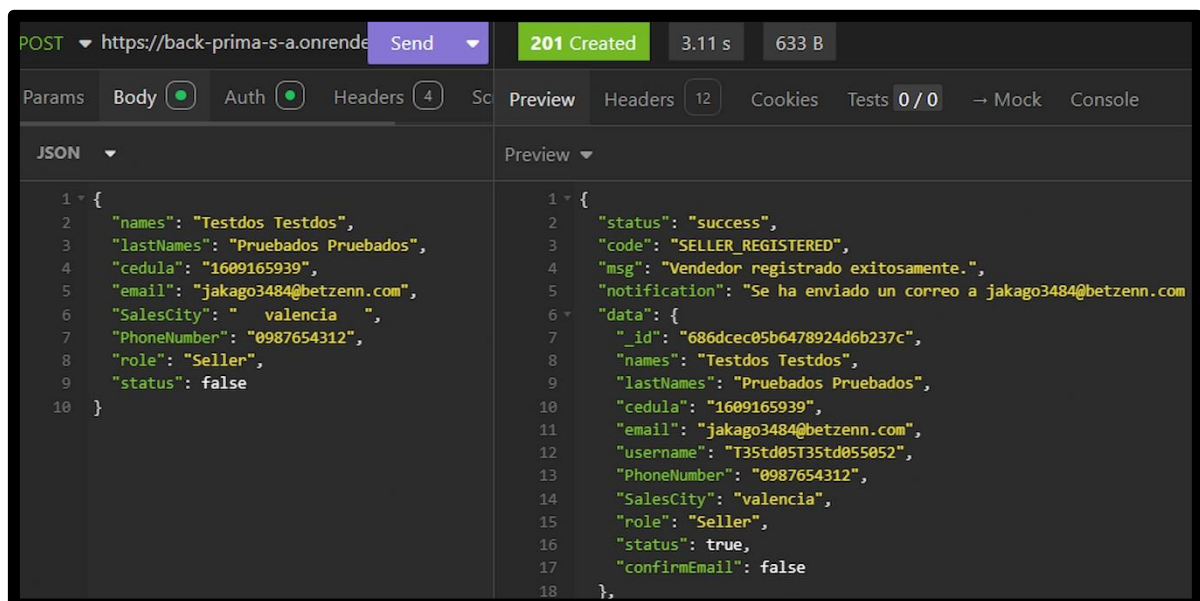


Figura 3.8 Registro de vendedor.

```
PASS tests/admin.test.js (7.103 s)
[Seller Registration / Routes]
  ✓ Should return 200 on successful registration (1793 ms)
  ✓ Should return 400 on empty fields (9 ms)
  ✓ Should return 400 on exist email (8 ms)
  ✓ Should return 400 on exist numberID (8 ms)
```

Figura 3.9 Test unitario – Registro de vendedor.

Creación de endpoints destinados a la funcionalidad de gestión de productos

El componente backend provee endpoints y rutas designadas para la gestión de productos por parte del administrador. Además, se implementa una validación de los datos de entrada previa a su procesamiento como se presenta en la **Figura 3.10** la inscripción de un nuevo producto, el resultado correspondiente a la prueba en la **Figura 3.11** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

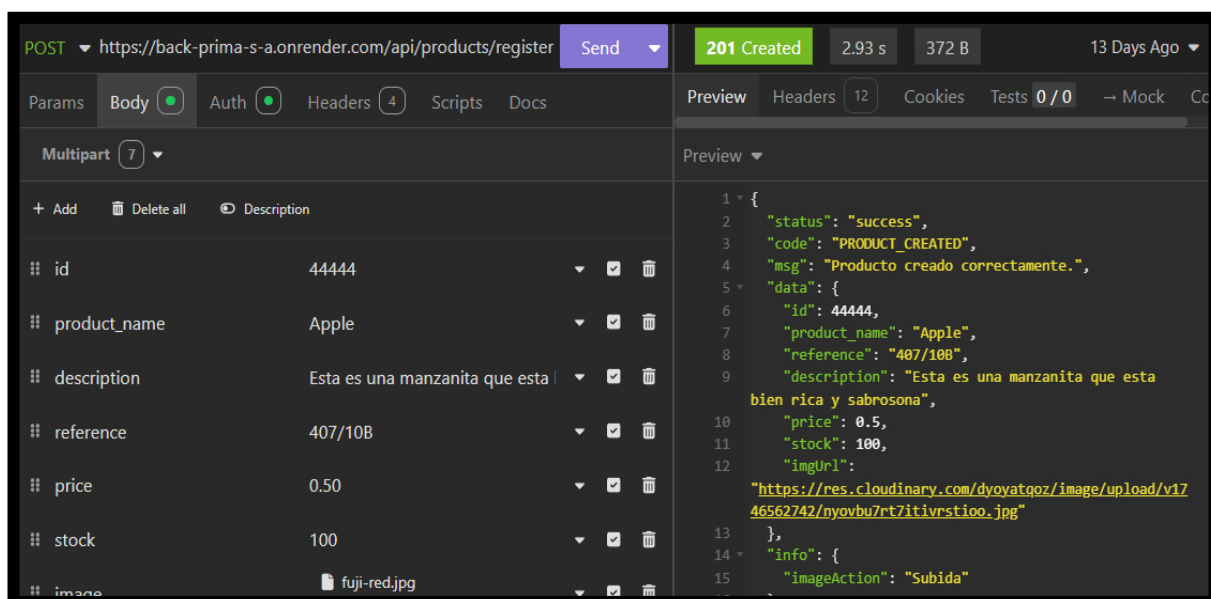


Figura 3.10 Registro de Producto.

```
PASS tests/products.test.js (6.461 s)
[Products Registration / Routes]
  ✓ Should return 201 on creating a product (1687 ms)
  ✓ Should return 409 on creating a product with an existing ID (1185 ms)
  ✓ Should return 400 on creating a product without an image (8 ms)
```

Figura 3.11 Prueba unitaria – Registro de Producto.

Creación de endpoints destinados a la funcionalidad de clientes

El componente backend está equipado con distintos endpoints que habilitan al administrador llevar a cabo un conjunto completo de operaciones sobre los datos de clientes. Además, se implementa una validación de los datos de entrada previa a su procesamiento como se presenta en la **Figura 3.12** la inscripción de un nuevo producto, el resultado correspondiente a la prueba en la **Figura 3.13** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

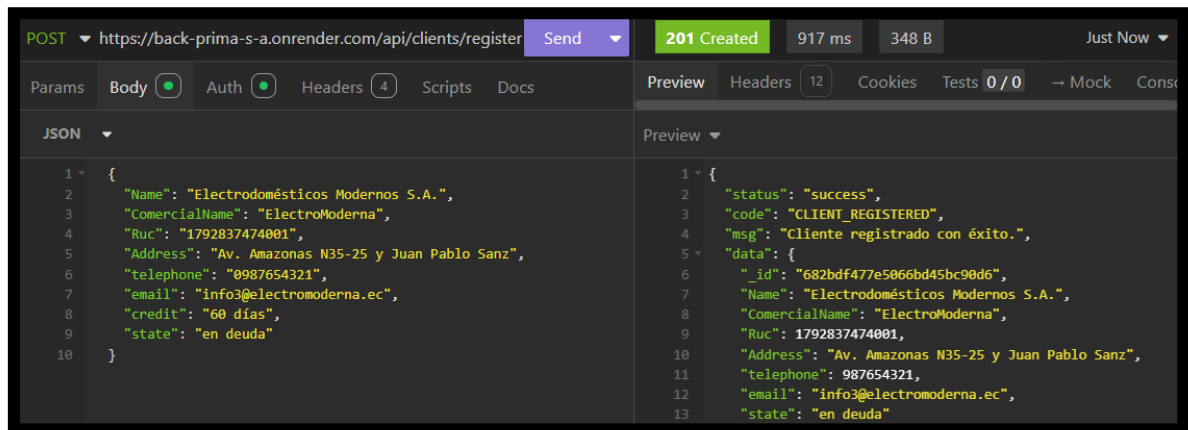


Figura 3.12 Registro de Cliente.

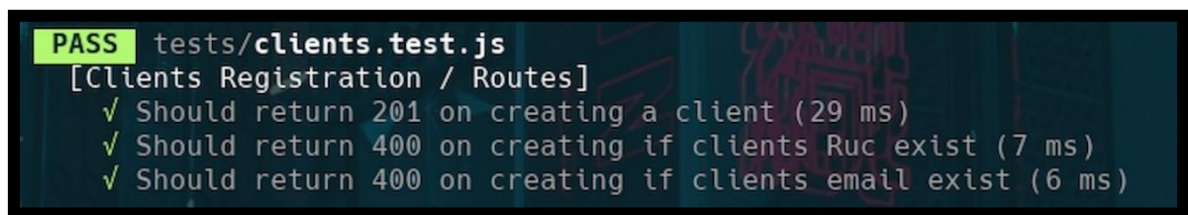


Figura 3.13 Test unitario – Registro de Cliente.

Creación de endpoints destinados a la funcionalidad de órdenes de compra

A través de endpoints específicos, el backend habilita al administrador la operación sobre órdenes de compra, las cuales incluye modificación del estado, anulación y la revisión de estas como se presenta en la **Figura 3.14** actualizar el estado de una orden de compra, el resultado correspondiente a la prueba en la **Figura 3.15** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

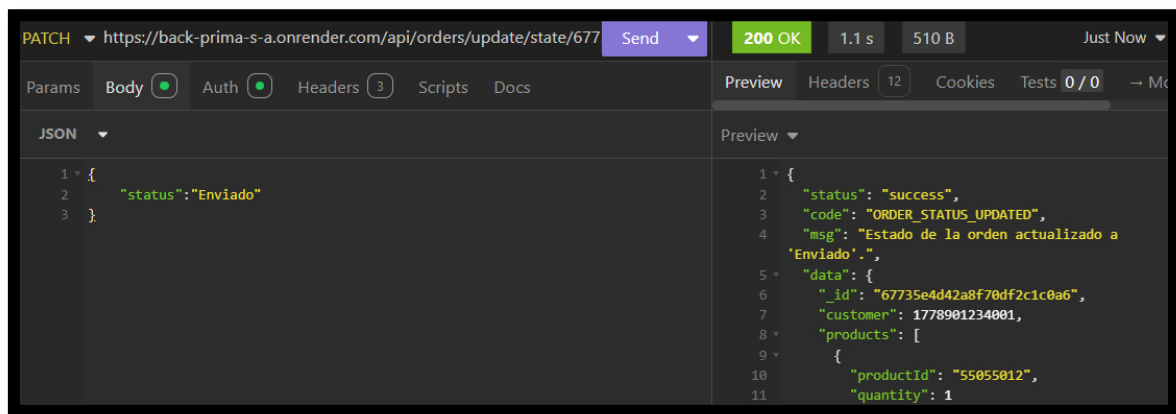


Figura 3.14 Actualización de estado.

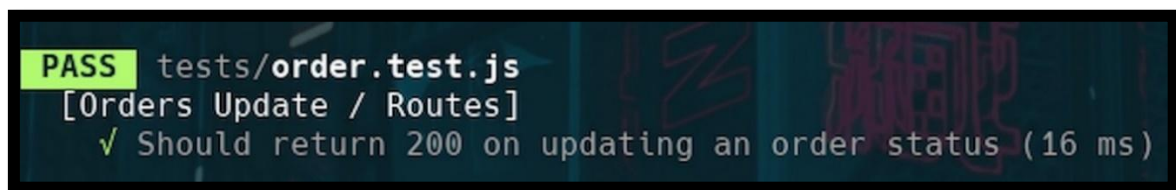


Figura 3.15 Test unitario - Actualización de estado.

Creación de endpoints destinados a la funcionalidad de visualización de analíticas

A través de diversos endpoints, el componente backend habilita al administrador la consulta de estadísticas clave, tales como como número de vendedores, productos, clientes y órdenes de compra que se han realizado como se presenta en la **Figura 3.16** ventas por vendedor, el resultado correspondiente a la prueba en la **Figura 3.17** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

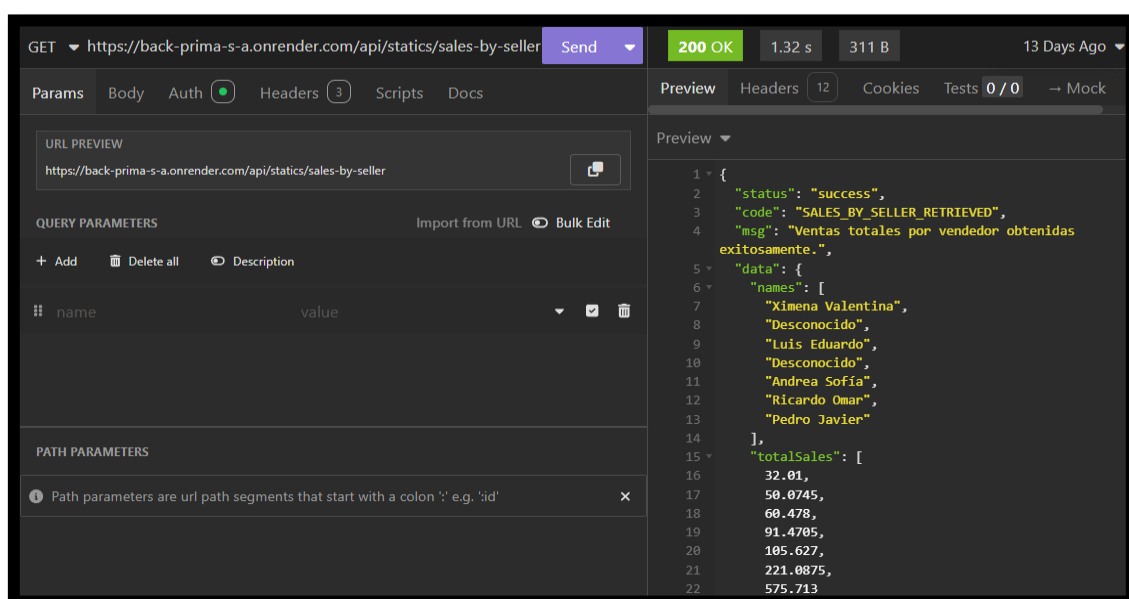


Figura 3.16 Visualización de ventas por vendedor.

```
PASS tests/stadistics.test.js
[STATICS] GET /statics/sales-by-seller
  ✓ should return 200 and sales by seller (22 ms)
  ✓ should return 401 if no token is provided (3 ms)
```

Figura 3.17 Test unitario - Visualización de ventas por vendedor.

Sprint 2 – Codificación de las funcionalidades para el rol Vendedor.

El enfoque de este Sprint es la implementación de operaciones destinadas al perfil vendedor, el cual incluye el acceso a la información de productos, la administración de clientes (con la excepción de la eliminación), las operaciones sobre órdenes de compra, así como los procedimientos de validación de identidad, incluyendo el Login y la recuperación de la cuenta. En ese sentido, las tareas para este ciclo de trabajo abarcan lo siguiente:

- El inicio de sesión del vendedor.
- Recuperación de la cuenta del vendedor a través de correo electrónico.
- Visualización de productos.
- Gestión de órdenes de compra.

Inicio de sesión del vendedor

El backend otorga varios endpoints y rutas específicas para que el usuario vendedor una vez que haya sido registrado por el administrador, pueda ingresar a los módulos asociados mediante el uso de credenciales de acceso las cuales se validan para todos los campos del formulario como se presenta en la **Figura 3.18** el correcto inicio de sesión del vendedor, el resultado correspondiente a la prueba en la **Figura 3.19** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

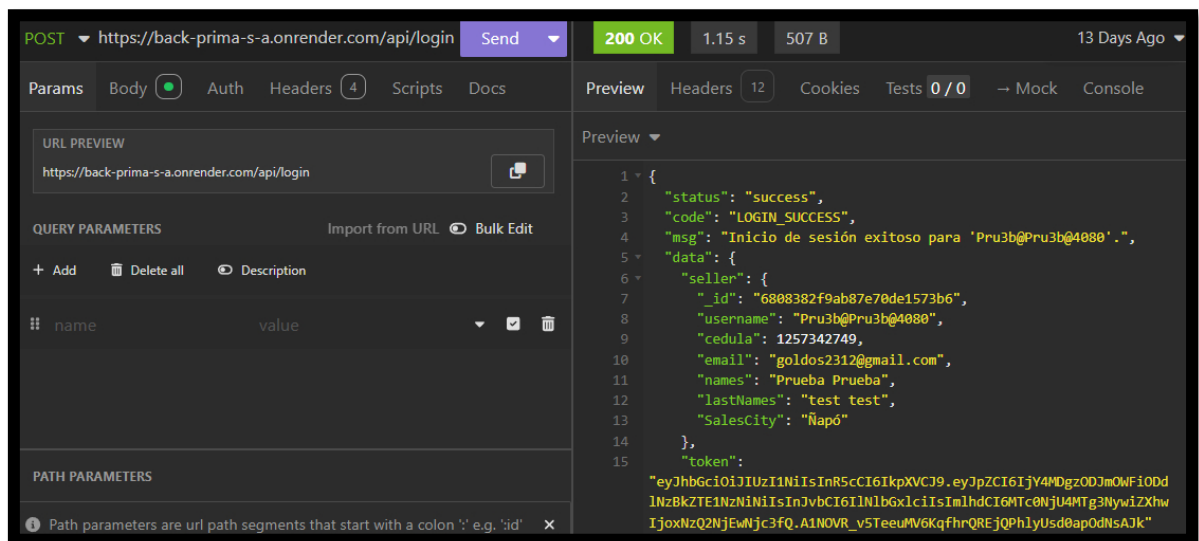


Figura 3.18 Inicio de sesión vendedor.

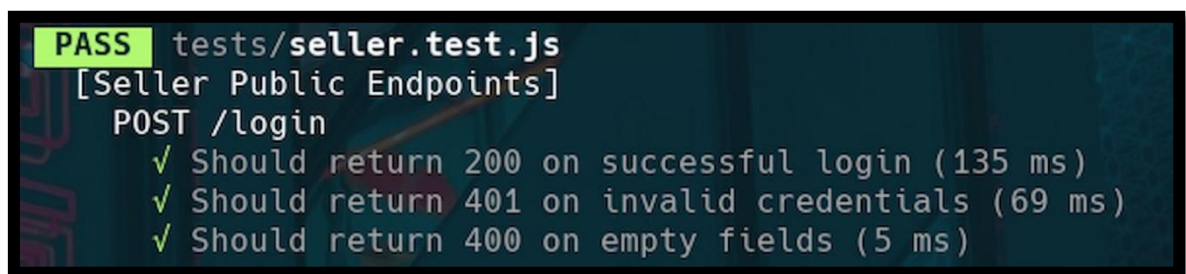


Figura 3.19 Test unitario – Inicio de sesión del vendedor.

Recuperación de la cuenta del vendedor a través de correo electrónico

El backend dispone de endpoints para la recuperación de cuenta del vendedor mediante su correo electrónico al cual se le envía un enlace para el restablecimiento y posterior a ello, ingresar con las nuevas credenciales de acceso como se ilustra en la **Figura 3.20** el correo enviado, el resultado correspondiente a la prueba en la **Figura 3.21** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

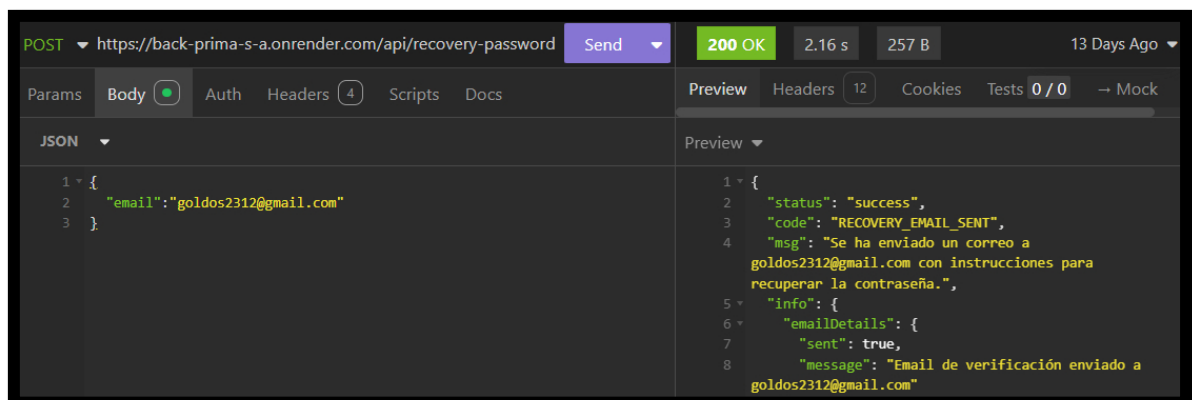


Figura 3.20 Recuperación de cuenta vendedor.

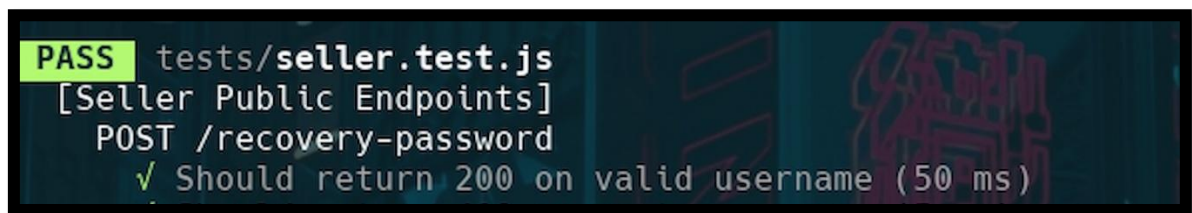


Figura 3.21 Test unitario- Recuperación de cuenta vendedor.

Visualización de productos

El componente backend está equipado con endpoints que otorgan al vendedor la capacidad de visualizar los productos como presenta la **Figura 3.22** la visualización de un producto, el resultado correspondiente a la prueba en la **Figura 3.23** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

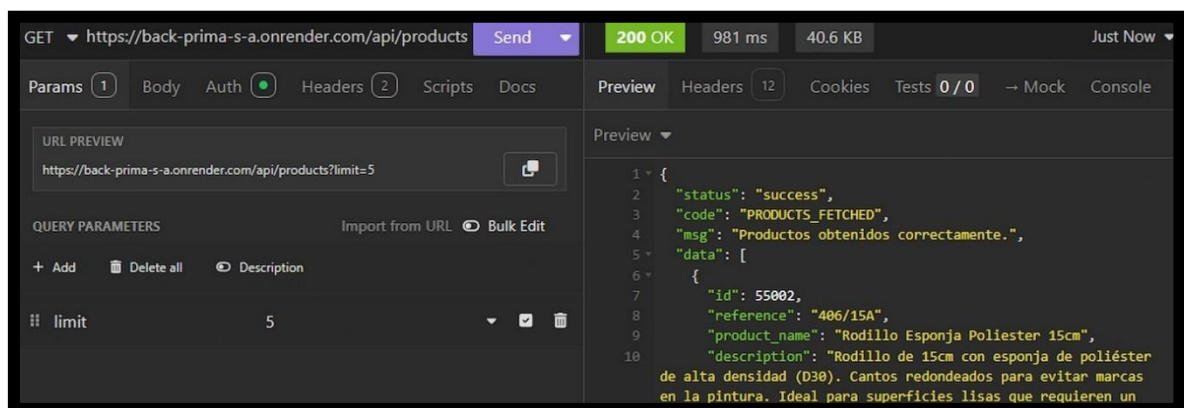


Figura 3.22 Visualización de producto.

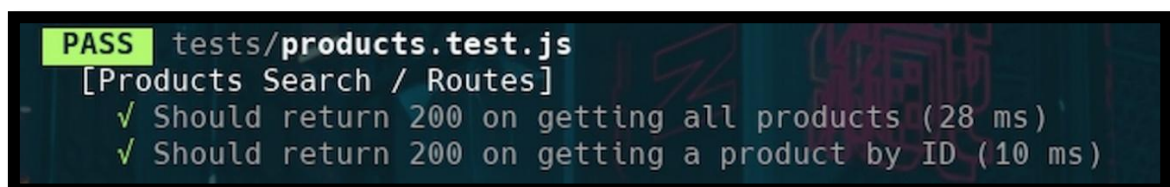


Figura 3.23 Test unitario – Visualización de producto.

Gestión de órdenes de compra

El componente backend habilita al usuario con rol vendedor realizar operaciones CRUD (Creación, Lectura, Actualización, Eliminación) sobre las órdenes de compra. Además, se implementa una validación de los datos de entrada previa a su procesamiento como se presenta en la **Figura 3.24** la elaboración de una orden de compra, el resultado correspondiente a la prueba en la **Figura 3.25** y una descripción de la funcionalidad completa se puede obtener desde el **ANEXO III**.

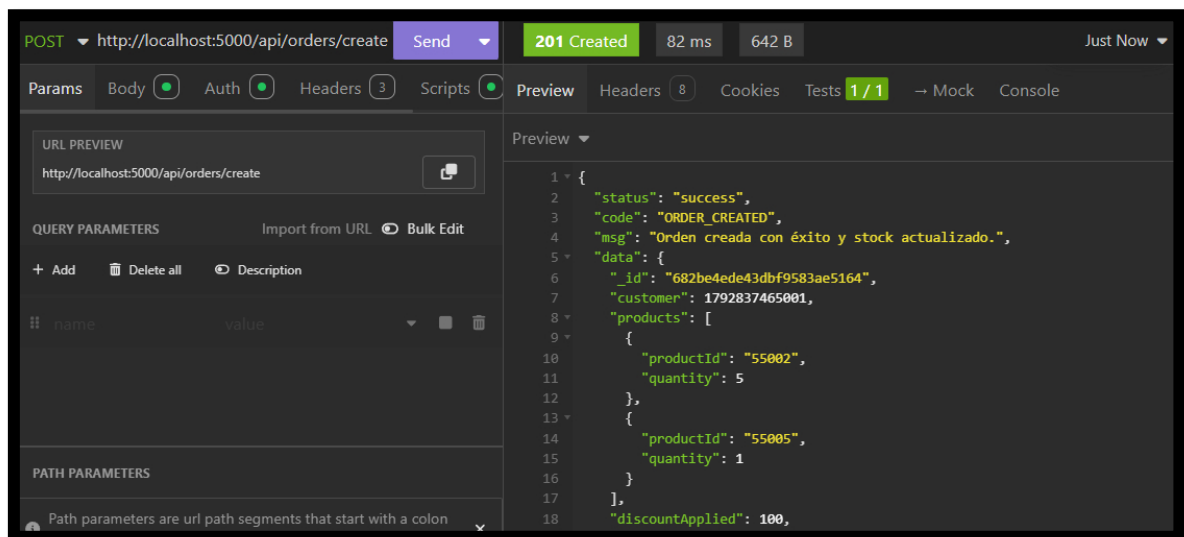


Figura 3.24 Registro de orden de compra.

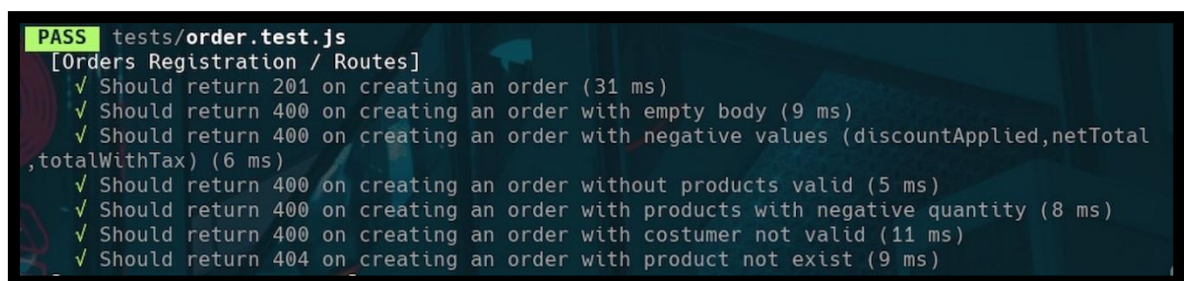


Figura 3.25 Prueba unitaria – Registro de orden de compra.

Sprint 3 - Pruebas.

Para este ciclo de trabajo se ha dedicado a la aplicación de un conjunto integral de pruebas con el objetivo de verificar la funcionalidad de cada uno de los endpoints. De esta manera, la asignación de tareas a realizar para este ciclo de trabajo son las siguientes.

- Tests unitarios.
- Tests de estrés.
- Tests de aceptación.

Tests unitarios

Este tipo de pruebas tienen la finalidad de determinar el comportamiento funcional de cada una de las unidades de código más importantes de una aplicación software. Asimismo, este enfoque de análisis agiliza la identificación de posibles errores en las etapas iniciales de la creación del software [39]. De esta forma, la presente prueba ha permitido validar el funcionamiento correcto de cada uno de los endpoints, y muestra de ello, se describe la

prueba usando el Framework JEST en la **Figura 3.26** y el resultado de la misma en la **Figura 3.27**. Para ver las pruebas unitarias completas ver el **ANEXO II**.

```
1 describe('[STATICS] GET /statics/count', () => {
2     it('should return 200 and entity counts', async () => {
3         const expected = 200
4         const response = await request
5             .get('/statics/count')
6             .set('Authorization', `Bearer ${ADMIN_TOKEN}`)
7         expect(response.status).toEqual(expected)
8         expect(response.body.data).toHaveProperty('products')
9         expect(response.body.data).toHaveProperty('orders')
10        expect(response.body.data).toHaveProperty('sellers')
11        expect(response.body.data).toHaveProperty('clients')
12    })
13
14    it('should return 401 if no token is provided', async () => {
15        const expected = 401
16        const response = await request.get('/statics/count')
```

Figura 3.26 Código de Prueba - Visualización de Estadísticas.

```
PASS tests/stadistics.test.js
[STATICS] GET /statics/count
  ✓ should return 200 and entity counts (20 ms)
  ✓ should return 401 if no token is provided (4 ms)
[STATICS] GET /statics/top-sellers
  ✓ should return 200 and top sellers (13 ms)
  ✓ should return 401 if no token is provided (3 ms)
[STATICS] GET /statics/sales-by-seller
  ✓ should return 200 and sales by seller (8 ms)
  ✓ should return 401 if no token is provided (4 ms)
```

Figura 3.27 Resultado prueba unitaria - Visualización de Estadísticas.

Ejecutar esta prueba unitaria ha sido de vital importancia para verificar que los endpoints que se han desarrollado se encuentren operativos y respondan según lo esperado con el objetivo de que cada usuario del backend no encuentre problemas al momento de ejecutarlo.

Test de estrés

Las pruebas de estrés tienen como objetivo someter a un sistema software a una gran cantidad de solicitudes y analizar su rendimiento y comportamiento. Además, el objetivo principal es descubrir posibles puntos débiles o áreas críticas donde el sistema comienza

a experimentar fallos y realizar correcciones oportunas [40]. Por lo tanto, esta prueba ha resultado fundamental para verificar la capacidad de respuesta del backend al ser sometido a un número elevado de solicitudes usando la herramienta JMeter, como se presenta en la **Figura 3.28**, el resultado de la misma en la **Figura 3.29** y en el **ANEXO II** contiene un desglose completo de las pruebas adicionales.

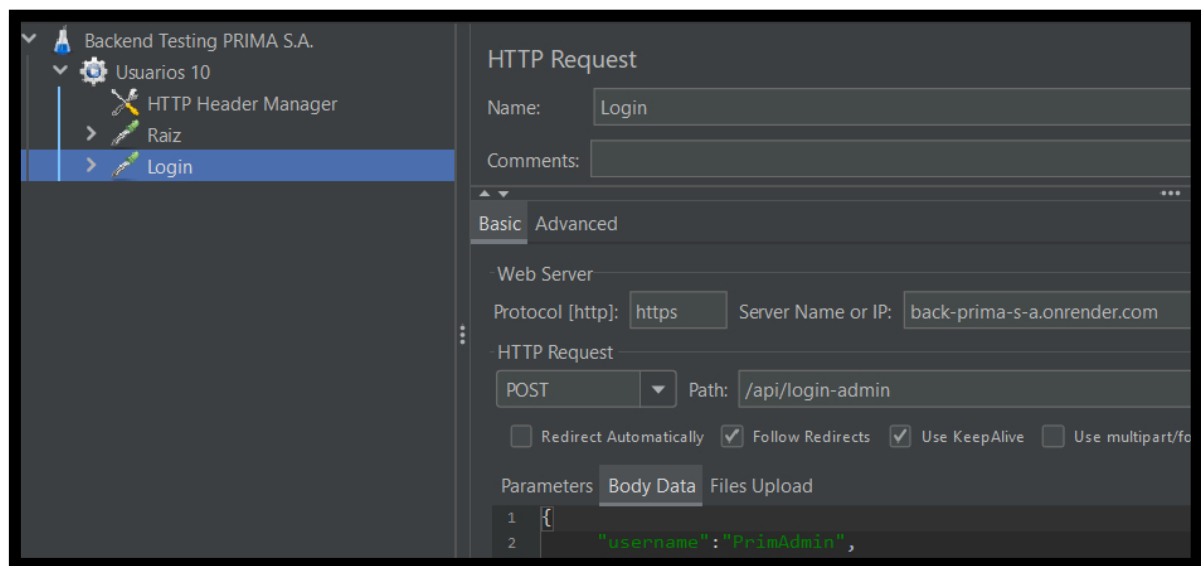


Figura 3.28 Implementación y configuración del Test de carga.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	10	1023	890	1136	65.96	0.00%	59.2/min	0.88	0.24	918.0
TOTAL	10	1023	890	1136	65.96	0.00%	59.2/min	0.88	0.24	918.0

Figura 3.29. Resultado del Test de carga.

Realizar esta prueba ha sido importante ya que se ha logrado entender de una forma clara cómo el backend utiliza sus recursos internos para responder a cada una de las peticiones. Además, los resultados aseguran que la estabilidad del backend sea segura y que no afecte al momento de su operación en producción.

Test de aceptación

El propósito de este tipo de pruebas es asegurar que todos los requerimientos se encuentren alineados con lo que el cliente espera, como paso previo a su aceptación formal. Además, no solo evalúan que el sistema funcione a nivel técnico, sino también que tenga un valor real y pueda cumplir con los objetivos del negocio [41]. De esta manera, ejecutar esta prueba ha sido importante para asegurar una plena satisfacción del cliente. En la **Tabla 3.1** se detalla un ejemplo de esta prueba y para visualizar todas las pruebas restantes se puede consultar el **ANEXO II**.

Tabla 3.1 Test de aceptación.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-010	(ID) historia de Usuario: HU006
Nombre: Visualizar estadísticas.	
Descripción: El componente backend proporciona un conjunto de endpoints que el administrador puede emplear para: <ul style="list-style-type: none"> • Presentar información general. • Visualizar el listado de vendedores registrados. • Visualizar las estadísticas de ventas semanales. 	
Descripción: Los endpoints usan un token JWT de autenticación para que solo los administradores puedan hacer uso estos endpoints.	
Instrucciones de funcionamiento: Para acceder y visualizar las estadísticas desde el backend se requiere: <ul style="list-style-type: none"> • Usar la URL del componente backend en un cliente Rest o en un navegador. • Iniciar sesión con las credenciales de administrador. • Verificar que la petición incluya el token de autorización. • Observar y analizar la información devuelta por los endpoints. 	
Resultado deseado: El backend entrega correctamente los datos de estadísticas generales, vendedores y ventas semanales únicamente a los administradores autenticados.	
Evaluación: Se ha verificado la funcionalidad del endpoint y la aprobación del cliente.	

La ejecución de esta prueba de aceptación ha permitido constatar la plena conformidad del cliente, como consecuencia directa de que todas las evaluaciones han arrojado resultados positivos.

Sprint 4 - Despliegue.

La culminación de este proyecto se centra en la puesta a producción del backend a través de un servicio de alojamiento en la nube. Además, esta estrategia de despliegue que es Render se caracteriza por su rapidez, la ausencia de configuraciones complejas y la facilidad para escalar nuevos recursos. El acceso a las funcionalidades del backend se realiza mediante la URL subsiguiente.

<https://back-prima-s-a.onrender.com/>

De igual manera, la documentación correspondiente al backend puede ser consultada a través del siguiente enlace web.

<https://docsapi.mathiast.me/>

Finalmente, en el **ANEXO II** se puede evidenciar que el dueño de la empresa “PRIMA S.A.” ha otorgado un certificado que confirma el cumplimiento de todos los requisitos para el negocio y funcionalidades que se han definido para el componente sistema web.

4 CONCLUSIONES

A continuación, se presentan las principales conclusiones que se han obtenido a lo largo del desarrollo del componente backend con la ayuda de la metodología Scrum.

- Gracias a un adecuado análisis y documentación de requisitos, se ha desarrollado una API RESTful robusta y completamente funcional, la cual está destinada a la gestión integral de las operaciones comerciales de PRIMA S.A. Además, este componente backend permite una administración centralizada de ventas, inventarios, productos, así como las relaciones con clientes y vendedores, constituyendo una solución integral para las necesidades de la empresa.
- La adopción de una arquitectura modular basada en la separación de responsabilidades ha sido un pilar fundamental para este proyecto, ya que no solo ha facilitado un desarrollo organizado y mantenible, sino que también ha sentado las bases para futuras expansiones y la integración de nuevas funcionalidades con un impacto mínimo en los módulos existentes.
- La elección de un modelo de datos NoSQL, el cual ha sido implementado con MongoDB y gestionado mediante el ODM Mongoose, ha demostrado ser una decisión acertada, ya que ha proporcionado flexibilidad para modelar las diversas entidades del negocio y sus interrelaciones, optimizando así el rendimiento de las operaciones CRUD y consultas complejas, además ofrece escalabilidad conforme evolucionan las necesidades de datos de la empresa.
- El diseño, validación, documentación e implementación de un conjunto de endpoints públicos y privados de tipo RESTful, ha garantizado una comunicación eficiente y controlada. Además, la autenticación robusta mediante JWT, junto con la autorización basada en roles, crea un entorno sólido que asegura la integridad de los datos sensibles y la continuidad de las funcionalidades vitales del componente backend.
- La ejecución de un conjunto integral de pruebas ha permitido verificar sistemáticamente la correcta operatividad de los diferentes módulos y endpoints del backend. Además, este riguroso proceso de pruebas ha sido esencial para la detección temprana de errores, la validación de la lógica de negocio y, en última instancia, asegurar la entrega de un producto de software de alta calidad, el cual se encuentra desplegado en producción y operativo para los empleados de la empresa.

5 RECOMENDACIONES

A continuación, se presentan las principales recomendaciones que deben tenerse en consideración y que pueden servir de insumo para futuros proyectos similares.

- Se recomienda adoptar un enfoque proactivo de revisión y mejora continua en el análisis del diseño de colecciones, documentos y la refactorización de consultas, ya que esto permite que el backend pueda escalar conforme aumenta la cantidad de datos que se vayan gestionado.
- Se recomienda fortalecer progresivamente las estrategias de seguridad del backend, más allá de la autenticación y autorización iniciales, mediante la implementación de mecanismos avanzados de gestión de tokens, auditorías de seguridad regulares y la constante actualización de dependencias para mitigar vulnerabilidades.
- Se recomienda ampliar la cobertura de pruebas con el objetivo de incrementar la confianza en la estabilidad y robustez del backend frente a diversas condiciones de uso y posibles cambios futuros.
- Se recomienda proporcionar siempre una documentación interactiva de la API, ya que esto facilita la exploración y el consumo de los endpoints por parte de los equipos de desarrollo u otros servicios, reduciendo la curva de aprendizaje y fomentando una integración más fluida.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] “Cómo la digitalización transforma los procesos empresariales”. Impacto TIC. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://impactotic.co/tecnologia/como-la-digitalizacion-transforma-los-procesos-empresariales/"](https://impactotic.co/tecnologia/como-la-digitalizacion-transforma-los-procesos-empresariales/) \o ["https://impactotic.co/tecnologia/como-la-digitalizacion-transforma-los-procesos-empresariales/"](https://impactotic.co/tecnologia/como-la-digitalizacion-transforma-los-procesos-empresariales/) <https://impactotic.co/tecnologia/como-la-digitalizacion-transforma-los-procesos-empresariales/>
- [2] “¿Qué es y cómo definir el alcance de un proyecto? Ejemplo y 5 pasos”. Appvizer. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/alcance-de-un-proyecto"](https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/alcance-de-un-proyecto) \o ["https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/alcance-de-un-proyecto"](https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/alcance-de-un-proyecto) <https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/alcance-de-un-proyecto>
- [3] Equipo editorial de IONOS. “¿Qué es el backend?” IONOS Digital Guide. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/"](https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/) \o ["https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/"](https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/) <https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/>
- [4] “¿Qué es una API de REST?” Red Hat - We make open source technologies for the enterprise. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://www.redhat.com/es/topics/api/what-is-a-rest-api"](https://www.redhat.com/es/topics/api/what-is-a-rest-api) \o ["https://www.redhat.com/es/topics/api/what-is-a-rest-api"](https://www.redhat.com/es/topics/api/what-is-a-rest-api) <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [5] “¿Qué Es MongoDB?” MongoDB. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://www.mongodb.com/es/company/what-is-mongodb"](https://www.mongodb.com/es/company/what-is-mongodb) \o ["https://www.mongodb.com/es/company/what-is-mongodb"](https://www.mongodb.com/es/company/what-is-mongodb) <https://www.mongodb.com/es/company/what-is-mongodb>
- [6] “¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS”. Amazon Web Services, Inc. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://aws.amazon.com/es/what-is/javascript/"](https://aws.amazon.com/es/what-is/javascript/) \o

"https://aws.amazon.com/es/what-is/javascript/" https://aws.amazon.com/es/what-is/javascript/

- [7] J. L. Flores. "Qué es NodeJS y para qué sirve | OpenWebinars". OpenWebinars.net. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://openwebinars.net/blog/que-es-nodejs/"](https://openwebinars.net/blog/que-es-nodejs/) \o "https://openwebinars.net/blog/que-es-nodejs/" https://openwebinars.net/blog/que-es-nodejs/
- [8] "CICS transaction server for z/os 6.x". IBM - United States. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://www.ibm.com/docs/es/cics-ts/6.x?topic=cics-json-web-token-jwt"](https://www.ibm.com/docs/es/cics-ts/6.x?topic=cics-json-web-token-jwt) \o "https://www.ibm.com/docs/es/cics-ts/6.x?topic=cics-json-web-token-jwt" https://www.ibm.com/docs/es/cics-ts/6.x?topic=cics-json-web-token-jwt
- [9] Top Java Full Stack Web Development Course in Bangalore | Guaranteed 100% Placement Support. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://www.dctacademy.com/blog/what-is-object-document-mapper-odm"](https://www.dctacademy.com/blog/what-is-object-document-mapper-odm) \o "https://www.dctacademy.com/blog/what-is-object-document-mapper-odm" https://www.dctacademy.com/blog/what-is-object-document-mapper-odm
- [10] "Express - infraestructura de aplicaciones web node.js". Express - Node.js web application framework. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://expressjs.com/es/"](https://expressjs.com/es/) \o "https://expressjs.com/es/" https://expressjs.com/es/
- [11] "The collaborative API development platform". The Collaborative API Development Platform - Insomnia. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://insomnia.rest/"](https://insomnia.rest/) \o "https://insomnia.rest/" https://insomnia.rest/
- [12] Byxn. "Los mejores sitios para desplegar tu aplicación web en 2024". DEV Community. Accedido el 6 de marzo de 2025. [En línea]. Disponible: [HYPERLINK "https://dev.to/bhyxen/los-mejores-sitios-para-desplegar-tu-aplicacion-web-en-2024-1hfh"](https://dev.to/bhyxen/los-mejores-sitios-para-desplegar-tu-aplicacion-web-en-2024-1hfh) \l "4-render-despliegue-simple-y-escalable" \o "https://dev.to/bhyxen/los-mejores-sitios-para-desplegar-tu-aplicacion-web-en-2024-1hfh#4-render-despliegue-simple-y-escalable" https://dev.to/bhyxen/los-mejores-sitios-para-desplegar-tu-aplicacion-web-en-2024-1hfh#4-render-despliegue-simple-y-escalable

- [13] M. en C. Oscar Elizalde. "El estudio de casos como método de investigación". lamalditatis. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.lamalditatis.org/post/el-estudio-de-casos-investigacion>
- [14] "Software development methodologies: Choose your approach! | abstracta". Blog about Software Development, Testing, and AI | Abstracta. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://abstracta.us/blog/development/software-development-methodologies-choose-your-approach/>
- [15] "¿Qué son las metodologías ágiles? - desarrollo de software". Freelancer Blog. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.freelancermag.com/blog/es/metodologias-agiles-desarrollo-software/>
- [16] "Scrum: Conceptos clave y cómo se aplica en la gestión de proyectos [2025] • Asana". Asana. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://asana.com/es/resources/what-is-scrum>
- [17] "Blog de cristina ramos vega". Blog de Cristina Ramos Vega. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://cristinaramosvega.com/los-roles-scrum/>
- [18] Artefactos Scrum: ¿qué son? ¿Para qué sirven? Mira ejemplos, <https://miro.com/es/agile/que-son-artefactos-scrum/> (accessed Apr. 1, 2025).
- [19] "Técnicas de recopilación de requisitos en ingeniería de software ágil - Visure Solutions". Visure Solutions. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://visuresolutions.com/es/guía-de-trazabilidad-de-gestión-de-requisitos/recopilación-ágil-de-requisitos/>
- [20] "¿Qué son las historias de usuario Scrum Agile?" Index German. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.scrumstudy.com/article/qu-son-las-historias-de-usuario-scrum-agile>
- [21] "Product backlog - what is it & how to create one". Collaboration software for software, IT and business teams. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.atlassian.com/agile/scrum/backlogs#:~:text=A%20product%20backlog%20is%20a,known%20what%20to%20deliver%20first.>
- [22] "Scrum sprint backlog creation steps". Index German. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.scrumstudy.com/article/scrum-sprint-backlog>

- [23] R. Murillo, «Patrones de arquitectura de software | Yapiko», Yapiko, 23 de junio de 2023. <https://yapiko.com/es/blog/patrones-arquitectura-software/>
- [24] “Ingeniería de software: Modelo vista controlador – devcodelight”. DevCodeLight. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://devcodelight.com/ingenieria-de-software-modelo-vista-controlador/>
- [25] Entornos de desarrollo integrados (IDE) | EducaOpen, <https://www.educaopen.com/digital-lab/blog/software/entornos-de-desarrollo-integrados> (accessed Apr. 3, 2025).
- [26] “Qué es GitHub y cómo empezar a usarlo”. ES Tutoriales. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.hostinger.com/es/tutoriales/que-es-github>
- [27] G. C. Cuadrado. “Visual Studio Code: Editor de código para desarrolladores | OpenWebinars”. OpenWebinars.net. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- [28] “Herramienta para pruebas de API: Reseña de insomnia - abstracta”. Blog de Desarrollo de Software, Testing e Inteligencia Artificial | Abstracta. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://es.abstracta.us/blog/insomnia-herramienta-pruebas-api/>
- [29] “¿Qué es express.js? Todo lo que debes saber”. Kinsta®. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://kinsta.com/es/base-de-conocimiento/que-es-express/>
- [30] kanezi. “Hosting apps on render.com”. Medium. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://medium.com/@kanezi.com/hosting-apps-on-render-com-15edbd18490>
- [31] “¿Qué Es MongoDB?” MongoDB. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.mongodb.com/es/company/what-is-mongodb>
- [32] Introducción a las librerías de programación: ¿Qué son y cómo , <https://generaciondigital.es/que-es-una-libreria-en-programacion/> (accessed Apr. 3, 2025).

- [33] “Qué es bcrypt - términos y definiciones de ciberseguridad”. VPN Unlimited - Fast & Secure VPN service. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.vpnunlimited.com/es/help/cybersecurity/bcrypt>
- [34] “Intercambio de recursos de origen cruzado (CORS) - HTTP | MDN”. MDN Web Docs. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://developer.mozilla.org/es/docs/Web/HTTP/Guides/CORS>
- [35] “Dotenv/README-es.md at master · motdotla/dotenv”. GitHub. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://github.com/motdotla/dotenv/blob/master/README-es.md>
- [36] E. E. P. Gomez. “Cómo usar nodemailer para enviar correos electrónicos desde tu servidor node.js”. freeCodeCamp.org. Accedido el 3 de abril de 2025. [En línea]. Disponible: <https://www.freecodecamp.org/espanol/news/como-usar-nodemailer-para-enviar-correos-electronicos-desde-tu-servidor-node-js/>
- [37] “Qué es mongoose - Blog de Código Facilito,” CódigoFacilito. <https://codigofacilito.com/articulos/que-es-mongoose>
- [38] Magaña, L. M. L. (2020, 17 de enero). Qué es Json Web Token y cómo funciona | OpenWebinars. OpenWebinars.net. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- [39] “¿Qué son las pruebas unitarias?: explicación de las pruebas unitarias en AWS”. Amazon Web Services, Inc. Accedido el 6 de mayo de 2025. [En línea]. Disponible: <https://aws.amazon.com/es/what-is/unit-testing/>
- [40] Tamushi. “Pruebas de estrés de software: ¿qué son y para qué sirven?” Empresa de pruebas de software | Testing IT. Accedido el 6 de mayo de 2025. [En línea]. Disponible: <https://www.testingit.com.mx/blog/pruebas-de-estres-de-software>
- [41] Tamushi. “Pruebas de aceptación de software, ¿Cuándo y por qué son necesarias?” Empresa de pruebas de software | Testing IT. Accedido el 7 de mayo de 2025. [En línea]. Disponible: <https://www.testingit.com.mx/blog/pruebas-aceptacion-software>

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha utilizado para el desarrollo del backend, los cuales se encuentran detallados de la siguiente manera:

- **ANEXO I.** Resultado del programa anti-plagio Turnitin.
- **ANEXO II.** Información extra del componente backend.
- **ANEXO III.** Manual de usuario (video).
- **ANEXO IV.** Credenciales de acceso y despliegue.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio Turnitin.

F_AA_236

CERTIFICADO DE ORIGINALIDAD TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 20 de julio de 2025

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado DESARROLLO DE UN BACKEND asociado al DESARROLLO DE SISTEMA PARA LA GESTIÓN Y CONTROL DE PEDIDOS EN LA EMPRESA PRIMA S.A. elaborado por el estudiante MATHÍAS AGUSTÍN TERÁN ALCÍVAR de la carrera en DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta antiplagio "TURNITIN" para la revisión de originalidad del documento escrito producto del Trabajo de Integración Curricular indicado.

- El documento escrito tiene un índice de similitud del 09%.

Respecto al uso de herramientas de Inteligencia Artificial en el desarrollo del Trabajo de Integración Curricular, se certifica:

- El documento escrito tiene un porcentaje de uso de herramientas de Inteligencia Artificial menor al 20%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



Loarte Cajamarca Byron Gustavo
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos

ANEXO II

Recopilación de requerimientos

Todos los requisitos que se han establecido previamente al desarrollo del proyecto se encuentran registrados en la **Tabla 1** .

Tabla 1 Levantamiento de Requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ITEM
Backend	RR-001	Para el usuario administrador y vendedor se requiere generar endpoints para: <ul style="list-style-type: none">• Iniciar y cerrar sesión.• Recuperar contraseña.• Confirmar cuenta.
	RR-002	Para el usuario administrador se requiere generar endpoints para: <ul style="list-style-type: none">• Gestionar productos.• Gestionar vendedores.• Gestionar clientes.• Gestionar órdenes de compra.• Visualizar estadísticas.
	RR-003	Para el usuario vendedor se requiere generar endpoints para: <ul style="list-style-type: none">• Visualizar productos.• Gestionar clientes.• Gestionar órdenes de compra.

Historias de Usuario

Las Historias de Usuario se encuentran detalladas a continuación, las mismas que han sido creadas una vez que se ha finalizado con la fase de levantamiento de requerimientos. De esta manera, se han realizado 9 Historias de Usuario, las cuales se detallan a continuación, desde la **Tabla 2** hasta la **Tabla 9**.

Tabla 2 Historia de usuario para gestionar cuenta.

HISTORIA DE USUARIO	
Identificador: HU-001	Usuario: Administrador y Vendedor
Nombre historia: Gestionar cuenta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador y vendedor para: <ul style="list-style-type: none">• Iniciar y cerrar sesión.• Recuperar contraseña.• Confirmar Cuenta.	
Observación: Todos los endpoints mencionados en la descripción son accesibles para los usuarios con rol administrador y vendedor. Además, se garantiza que las acciones sean seguras permitiendo que los datos personales sean encriptados antes de almacenarse en la base de datos.	

Tabla 3 Historia de usuario Gestión de productos.

HISTORIA DE USUARIO	
Identificador: HU-002	Usuario: Administrador
Nombre historia: Gestionar productos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para: <ul style="list-style-type: none">• Crear productos.	

- Visualizar productos.
- Visualizar productos por código.
- Actualizar productos.
- Eliminar productos.

Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de los productos.

Tabla 4 Historia de usuario Gestión de vendedores.

HISTORIA DE USUARIO	
Identificador: HU-003	Usuario: Administrador
Nombre historia: Gestionar vendedores	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Registrar vendedores. • Visualizar vendedores. • Visualizar vendedores por ID. • Actualizar vendedores. • Eliminar vendedores. 	
<p>Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de los vendedores.</p>	

Tabla 5 Historia de usuario Gestión de clientes.

HISTORIA DE USUARIO	
Identificador: HU-004	Usuario: Administrador
Nombre historia: Gestionar clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Visualizar clientes. • Visualizar clientes por ID. • Eliminar clientes. 	
<p>Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de los clientes. Por último, los administradores no registran ni actualizan la información de los clientes debido a que no interactúan con los mismos.</p>	

Tabla 6 Historia de usuario Gestión de órdenes de compra.

HISTORIA DE USUARIO	
Identificador: HU-005	Usuario: Administrador
Nombre historia: Gestionar órdenes de compra	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Visualizar órdenes. • Visualizar órdenes por ID. • Actualizar estado de las órdenes. • Eliminar órdenes. 	

Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de las órdenes de compra.

Tabla 7 Historia de usuario Visualización de estadísticas.

HISTORIA DE USUARIO	
Identificador: HU-006	Usuario: Administrador
Nombre historia: Visualizar estadísticas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para: <ul style="list-style-type: none"> • Presentar información de forma general. • Visualizar vendedores. • Visualizar ventas semanales. 	
Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan visualizar estadísticas relevantes para el negocio y que sirva para una adecuada toma de decisiones.	

Tabla 8. Historia de usuario Visualización de productos.

HISTORIA DE USUARIO	
Identificador: HU-007	Usuario: Vendedor
Nombre historia: Visualizar productos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:	

<ul style="list-style-type: none"> • Visualizar productos. • Visualizar productos por código.
<p>Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol vendedor. Además, se implementa controles de acceso para garantizar que solo los vendedores puedan visualizar la información de los productos.</p>

Tabla 9. Historia de usuario Gestión de clientes.

HISTORIA DE USUARIO	
Identificador: HU-008	Usuario: Vendedor
Nombre historia: Gestionar clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable (s): Mathías Terán	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Registrar clientes. • Visualizar clientes. • Visualizar clientes por ID. • Actualizar clientes. 	
<p>Observación: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol vendedor. Además, se implementa controles de acceso para garantizar que solo los vendedores puedan gestionar la información de los clientes a excepción de eliminar clientes que esa funcionalidad lo realiza el rol administrador.</p>	

Product Backlog

Cada módulo tiene una prioridad, dependiendo del impacto de este dentro del componente, de esta manera en la **Tabla 10** se encuentra el ciclo según la prioridad.

Tabla 10 Product Backlog.

PRODUCT BACKLOG				
ID-HU	DESCRIPCIÓN DE LA HISTORIA	ITERACIÓN ASIGNADA	ESTADO	PRIORIDAD
HU-001	Gestionar cuenta	1	Finalizado	Alta
HU-002	Gestionar productos	1	Finalizado	Alta
HU-003	Gestionar vendedores	1	Finalizado	Alta
HU-004	Gestionar clientes	1	Finalizado	Alta
HU-005	Gestionar órdenes de compra	1	Finalizado	Alta
HU-006	Visualizar estadísticas	1	Finalizado	Alta
HU-007	Visualizar productos	2	Finalizado	Alta
HU-009	Gestionar órdenes de compra	2	Finalizado	Alta

Sprint Backlog

Para el desarrollo completo del proyecto se han realizado en 5 Sprints, los cuales se encuentran delimitados en la **Tabla 11**.

Tabla 11 Sprint Backlog.

ELABORACIÓN DEL SPRINT BACKLOG						
ID-SB	NOMBRE	MÓDULO	HU	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB-000	Configuración del ambiente de desarrollo	N/A	N/A	N/A	<ul style="list-style-type: none"> Recopilación y definición de requerimientos. Diseño y creación de la Base de datos. Diseño de arquitectura RESTful para el Backend en base a los requerimientos. 	70 H
SB-001	Diseño y codificación de endpoints para el módulo administrador	Módulo perfil	HU-001	Gestionar cuenta	<ul style="list-style-type: none"> Implementación de los endpoints para poder iniciar sesión, cerrar sesión y recuperar contraseña. Validación de ingreso de información por parte del administrador al consumir los endpoints. Consulta de datos en la Base de Datos. 	60H
		Módulos administradores	HU-002	Gestionar productos		
			HU-003	Gestionar vendedores		
			HU-004	Gestionar clientes		
			HU-005	Gestionar órdenes de compra		
			HU-006	Visualizar estadísticas		

					<ul style="list-style-type: none"> • Verificación de la integridad de los datos requeridos. • Modelamiento e implementación de endpoints para que el administrador realice acciones tales como gestionar vendedores, gestionar productos, gestionar clientes, gestionar órdenes de compra y visualizar estadísticas. • Registro de información 	
SB-002	Diseño y codificación de endpoints para el módulo vendedores	Módulo perfil	HU-001	Gestionar cuenta	<ul style="list-style-type: none"> • Implementación de los endpoints para poder iniciar sesión, cerrar sesión y recuperar contraseña. • Validación de ingreso de información por parte del vendedor al consumir los endpoints. • Consulta de datos en la Base de Datos. • Verificación de la integridad de los datos requeridos. • Modelamiento e implementación de endpoints para que el vendedor realice 	40H
		Módulos vendedores	HU-007	Visualizar productos		
			HU-008	Gestionar clientes		
			HU-009	Gestionar órdenes de compra		

					acciones tales como visualizar productos, gestionar clientes y gestionar órdenes de compra. <ul style="list-style-type: none">Registro de información.	
SB-003	Pruebas del backend	<ul style="list-style-type: none">Pruebas unitarias.Prueba de estrés.Pruebas de aceptación.				30H
Documentación		<ul style="list-style-type: none">Trabajo de Integración Curricular.Anexos.				30H
TOTAL						240H

Elaboración de documentos por cada colección

A continuación, se presentan los documentos por cada colección los cuales van desde la Figura 1 hasta la Figura 5.

```
_id: ObjectId('67624f907669c906ca34a8de')
username : "PrimAdmin"
password : "$2a$10$62qq9f8ynenBv47vVskKneg0exD0oIBD6fS28Fe8lB1BNdGoTIUK."
role : "admin"
status : true
chances : 3
lastLogin : 2025-06-26T16:01:21.195+00:00
createdAt : 2024-12-17T12:00:00.000+00:00
updatedAt : 2025-06-26T16:01:21.285+00:00
```

Figura 1 Documento para administradores.

```
_id: ObjectId('683cec1b100e47b83878b40f')
Name : "ElectroModerna"
ComercialName : "Electrodomésticos Modernos S.A."
Ruc : "1792829654001"
Address : "Av. Amazonas N35-25 y Juan Pablo Sanz"
telephone : "0987654321"
email : "infojobs132@electromoderna.ec"
state : "en deuda"
```

Figura 2 Documento para clientes.

```
_id: ObjectId('683cfaaa6c0c4f69b2a255aa')
customer : "1792829654001"
seller : ObjectId('683cf9516c0c4f69b2a25582')
products : Array (2)
discountApplied : 15
netTotal : 12.68
totalWithTax : 14.58
credit : "Crédito 30 días"
status : "Pendiente"
comment : "Lorem Ipsum is simply dummy."
registrationDate : 2025-06-02T01:13:14.542+00:00
lastUpdate : 2025-06-03T04:02:04.690+00:00
createdAt : 2025-06-02T01:13:14.544+00:00
updatedAt : 2025-06-03T04:02:04.691+00:00
```

Figura 3 Documento para ordenes.

```
_id: ObjectId('681a71b1fd283578e9be6c01')
id : 55002
reference : "406/15A"
product_name : "Rodillo Esponja Poliéster 15cm"
description : "Rodillo de 15cm con esponja de poliéster de alta densidad (D30). Canto..."
price : 1.82
stock : 90
imgUrl : "https://res.cloudinary.com/dnp9gpo8w/image/upload/v1733881757/ROD_POL_..."
```

Figura 4 Documento para productos.

```
_id: ObjectId('683cf9516c0c4f69b2a25582')
names : "María Fernanda"
lastNames : "Gómez Pérez"
cedula : "1027491024"
email : "mafer@gmail.com"
SalesCity : "Quito"
PhoneNumber : "0995644186"
role : "Seller"
status : true
token : "o7qizisidi"
confirmEmail : true
password : "$2a$10$PuXmUl6S0tnJdlDcgfEfE0srK/5vMJkG2Biqu612AjDLylsei9glq"
username : "M@r1@F3rn@nd@8533"
createdAt : 2025-06-02T01:07:29.794+00:00
updatedAt : 2025-06-25T01:38:45.873+00:00
__v : 0
```

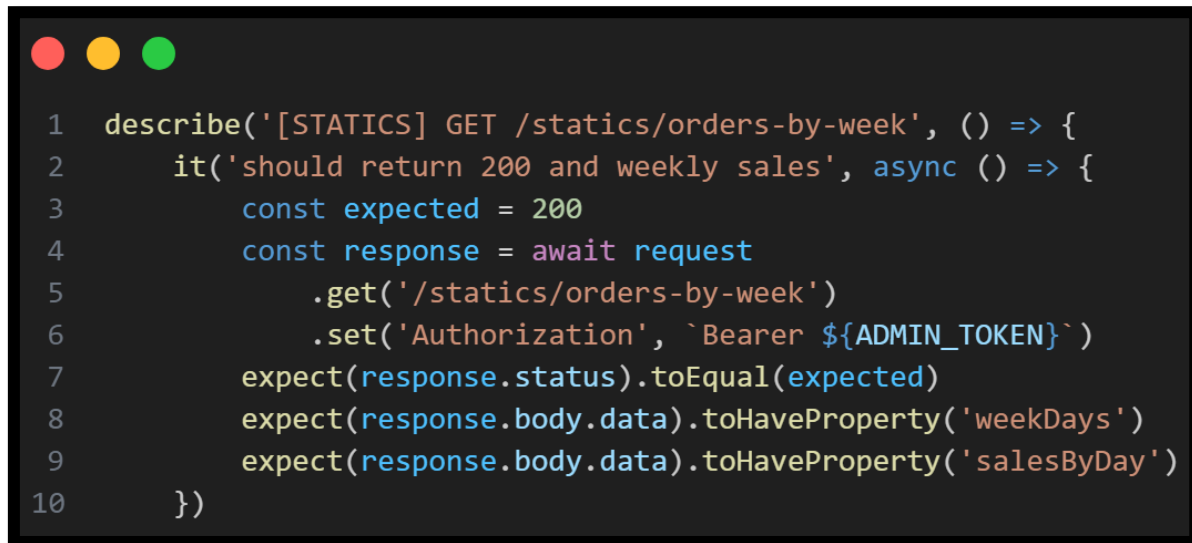
Figura 5 Documento para vendedores.

Pruebas

Al concluir la etapa de desarrollo, se procede a realizar pruebas pertinentes para asegurar la calidad del backend.

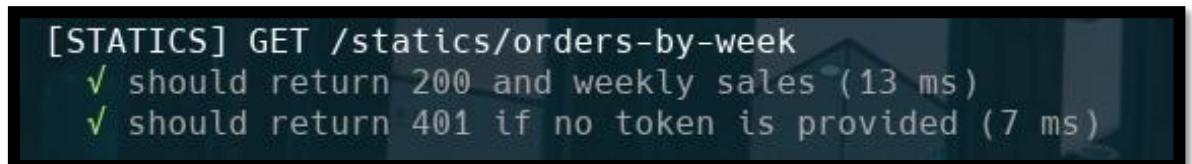
Pruebas unitarias

A continuación, se presentan las pruebas unitarias restantes las cuales van desde la **Figura 6** hasta la **Figura 17**.



```
1 describe('[STATICS] GET /statics/orders-by-week', () => {
2   it('should return 200 and weekly sales', async () => {
3     const expected = 200
4     const response = await request
5       .get('/statics/orders-by-week')
6       .set('Authorization', `Bearer ${ADMIN_TOKEN}`)
7     expect(response.status).toEqual(expected)
8     expect(response.body.data).toHaveProperty('weekDays')
9     expect(response.body.data).toHaveProperty('salesByDay')
10  })
}
```

Figura 6 Visualización de ordenes semanales.



```
[STATICS] GET /statics/orders-by-week
✓ should return 200 and weekly sales (13 ms)
✓ should return 401 if no token is provided (7 ms)
```

Figura 7 Prueba Unitaria de Visualización de ordenes semanales.



```
1 describe('[Seller Public Endpoints]', () => {
2   describe('POST /login', () => {
3     it('Should return 200 on successful login', async () => {
4       const response = await request
5         .post('/login')
6         .send({
7           username: process.env.SELLER_USER,
8           password: process.env.SELLER_PASSWORD
9         })
10    })
11  })
12 }
```

Figura 8 Inicio de sesión de vendedor.

```
[Seller Public Endpoints]
POST /login
  ✓ Should return 200 on successful login (136 ms)
  ✓ Should return 401 on invalid credentials (67 ms)
  ✓ Should return 400 on empty fields (5 ms)
```

Figura 9 Prueba unitaria de Inicio de sesión vendedor.

```
1 describe('[Products Deletion / Routes]', () => {
2   it('Should return 200 on deleting a product', async () => {
3     // Arrange
4     const expected = 200
5     // Act
6     const response = await request
7       .delete('/products/delete/44444')
8       .set('Authorization', `Bearer ${ADMIN_TOKEN}`)
9     // Assert
10    expect(response.status).toEqual(expected)
11  })
}
```

Figura 10 Eliminación de productos.

```
[Products Deletion / Routes]
  ✓ Should return 200 on deleting a product (404 ms)
  ✓ Should return 404 on deleting a product with unexisting ID (9 ms)
```

Figura 11 Prueba unitaria de Eliminación de productos.

```
1 describe('[Orders Update / Routes]', () => {
2   it('Should return 200 on updating an order', async () => {
3     // Arrange
4     const expected = 200
5     // Act
6     const response = await request
7       .patch('/orders/update/683cfaaa6c0c4f69b2a255aa')
8       .set('Authorization', `Bearer ${SELLER_TOKEN}`)
9       .send({
10         "products": [
11           {

```

Figura 12 Actualización de órdenes.

```
[Orders Update / Routes]
✓ Should return 200 on updating an order (16 ms)
✓ Should return 400 on updating an order with empty body (6 ms)
✓ Should return 400 on updating an order without products array (5 ms)
✓ Should return 400 on updating an order with products incomplete (6 ms)
✓ Should return 400 on updating an order with products with negative values (6 ms)
✓ Should return 404 on updating an order with invalid id (6 ms)
✓ Should return 200 on updating an order status (15 ms)
✓ Should return 400 on updating an order with invalid status (12 ms)
✓ Should return 400 on updating an order without stock (5 ms)
✓ Should return 404 on updating an order status with non exist id (5 ms)
```

Figura 13 Prueba Unitaria de Actualización de órdenes.

```
1 describe('[Clients Delete / Routes]', () => {
2     it('Should return 200 on deleting a client', async () => {
3         // Arrange
4         const expected = 200
5
6         // Act
7         const response = await request
8             .delete(`/clients/delete/${CLIENT_ID}`)
9             .set('Authorization', `Bearer ${ADMIN_TOKEN}`)
10
11         // Assert
12         expect(response.status).toEqual(expected)
13     })
14 }
```

Figura 14. Eliminación de clientes.

```
[Clients Delete / Routes]
✓ Should return 200 on deleting a client (12 ms)
✓ Should return 400 on deleting a client if id not exist (7 ms)
✓ Should return 400 on deleting a client if id not valid (5 ms)
```

Figura 15. Prueba Unitaria de Eliminación de clientes.

```

1 describe('[Admin Login / Routes]', ()=>{
2     it('Should return 200 on successful login', async () => {
3         // Arrange
4         const expected = 200
5
6         // Act
7         const response = await request
8             .post('/login-admin')
9             .send({
10             username: process.env.ADMIN_USER,
11             password: process.env.ADMIN_PASSWORD
12         })
13
14         // Assert
15         expect(response.status).toEqual(expected)
16     })

```

Figura 16. Inicio de sesión de administrador.

```

[Admin Login / Routes]
✓ Should return 200 on successful login (70 ms)
✓ Should return 400 on invalid login (5 ms)
✓ Should return 400 on empty fields (3 ms)
✓ Should return 400 on non-existent user (4 ms)
✓ Should return 400 on blocked account (4 ms)
✓ Should return 200 on successful password recovery (3472 ms)
✓ Should return 400 on empty username field (4 ms)

```

Figura 17. Prueba Unitaria de Inicio de sesión administrador.

Pruebas de estrés

A continuación, se presentan las pruebas de rendimiento restantes las cuales van desde la Figura 18 hasta la Figura 19.

Backend Testing / PRIMA S.A.		Summary Report									
Usuarios 10		Name: Summary Report									
HTTP Header Manager		Comments:									
Raiz		Write results to file / Read from file									
View Results Tree		Filename: <input type="text"/> <input type="button" value="Browse..."/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>									
Summary Report											
Graph Results											
Login											
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes	
Raiz	100	17375	473	50963	16329.19	0.00%	1.6/sec	1.40	0.52	867.0	
TOTAL	100	17375	473	50963	16329.19	0.00%	1.6/sec	1.40	0.52	867.0	

Figura 18: Prueba de carga.

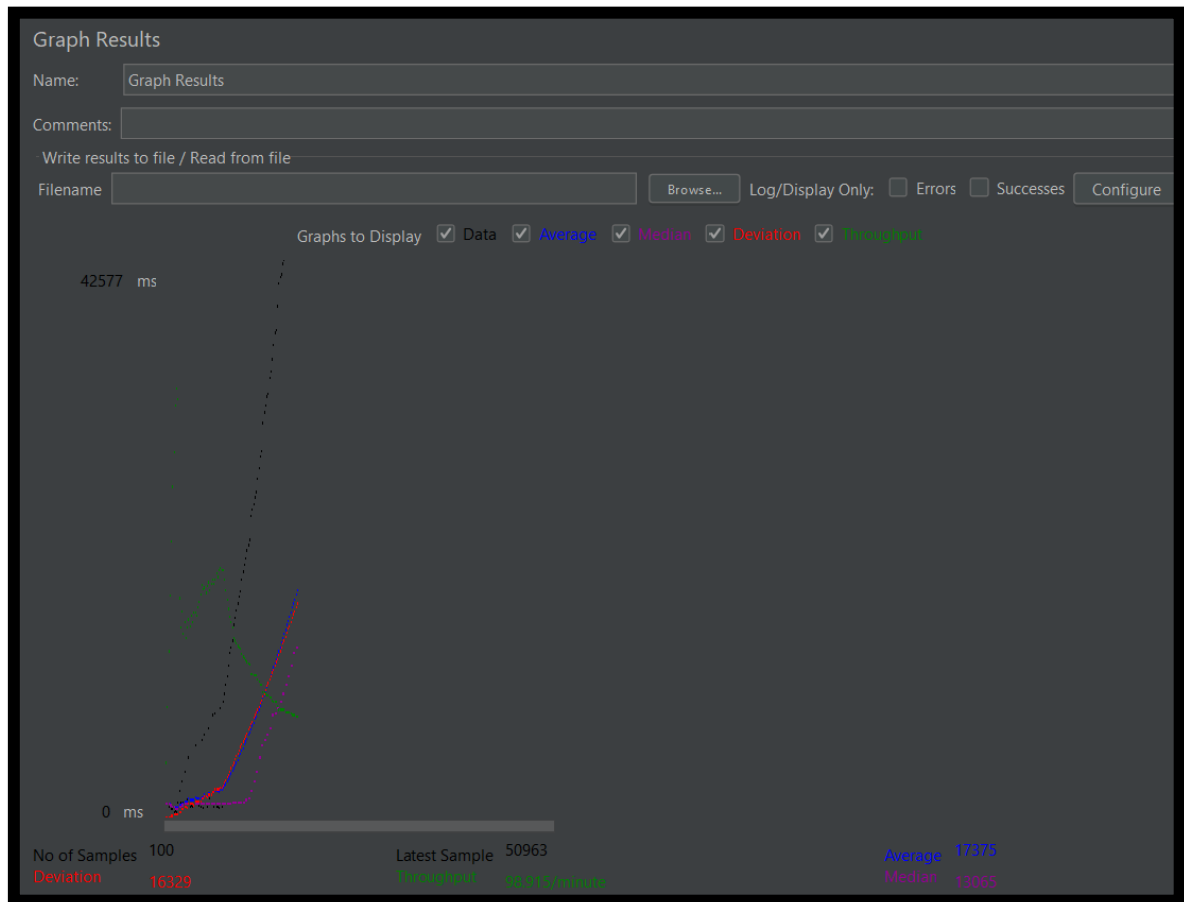


Figura 19. Grafo de Prueba de Carga.

Pruebas de aceptación

A continuación, se presentan las pruebas de aceptación restantes las cuales van desde la **Tabla 12** hasta la **Tabla 20**.

Tabla 12 Prueba de Aceptación de estadísticas.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-006	(ID) historia de Usuario: HU006
Nombre: Visualizar estadísticas	
Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para: <ul style="list-style-type: none"> • Presentar información de forma general. • Visualizar vendedores. • Visualizar ventas semanales. 	
Descripción: Todos los endpoints mencionados en la descripción son accesibles	

únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan visualizar estadísticas relevantes para el negocio y que sirva para una adecuada toma de decisiones.
<p>Instrucciones de funcionamiento:</p> <p>Para acceder y visualizar las estadísticas desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de administradores. • Incluir token de autorización en la petición. • Observar y analizar la información devuelta por los endpoints.
<p>Resultado deseado:</p> <p>El backend entrega correctamente los datos de estadísticas generales, vendedores y ventas semanales únicamente a los administradores autenticados.</p>
<p>Evaluación:</p> <p>Se ha verificado la funcionalidad del endpoint y la aprobación del cliente.</p>

Tabla 13 Prueba de aceptación gestionar cuenta.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-001	(ID) historia de Usuario: HU001
Nombre: Gestionar cuenta	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador y vendedor para:</p> <ul style="list-style-type: none"> • Iniciar y cerrar sesión. • Recuperar contraseña. • Confirmar Cuenta. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles para los usuarios con rol administrador y vendedor. Además, se garantiza que las acciones sean seguras permitiendo que los datos personales sean encriptados antes de almacenarse en la base de datos.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar la cuenta desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Realizar las acciones de iniciar sesión, cerrar sesión, recuperar contraseña y 	

<p>confirmar cuenta según corresponda.</p> <ul style="list-style-type: none"> • Para acciones que requieran credenciales, proporcionar credenciales válidas para usuarios con rol administrador o vendedor. • Verificar que los datos personales sean encriptados antes de almacenarse en la base de datos (mediante inspección de la base de datos o logs, si es accesible).
<p>Resultado deseado:</p> <p>El backend permite a los usuarios con rol administrador y vendedor iniciar y cerrar sesión, recuperar contraseña y confirmar su cuenta de forma segura, garantizando la encriptación de los datos personales antes de su almacenamiento.</p>
<p>Evaluación:</p> <p>Se ha verificado la funcionalidad de gestión de cuenta del endpoint y la aprobación del cliente.</p>

Tabla 14 Prueba de aceptación Gestionar productos.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-002	(ID) historia de Usuario: HU002
Nombre: Gestionar productos	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Crear productos. • Visualizar productos. • Visualizar productos por código. • Actualizar productos. • Eliminar productos. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de los productos.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar productos desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de administradores. • Incluir token de autorización en la petición. 	

<ul style="list-style-type: none"> • Ejecutar las operaciones de crear, visualizar (todos y por código), actualizar y eliminar productos. • Verificar que solo los usuarios con rol administrador puedan realizar estas operaciones.
<p>Resultado deseado:</p> <p>El backend permite al administrador crear, visualizar (todos y por código), actualizar y eliminar productos de forma segura, garantizando que solo los administradores puedan acceder a estas funcionalidades.</p>
<p>Evaluación:</p> <p>Se ha verificado la funcionalidad de gestión de productos del endpoint y la aprobación del cliente.</p>

Tabla 15 Prueba de aceptación Gestionar vendedores.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-003	(ID) historia de Usuario: HU003
Nombre: Gestionar vendedores	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Registrar vendedores. • Visualizar vendedores. • Visualizar vendedores por ID. • Actualizar vendedores. • Eliminar vendedores. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de los vendedores.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar vendedores desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de administradores. • Incluir token de autorización en la petición. • Ejecutar las operaciones de registrar, visualizar (todos y por ID), actualizar y 	

<p>eliminar vendedores.</p> <ul style="list-style-type: none"> • Verificar que solo los usuarios con rol administrador puedan realizar estas operaciones.
<p>Resultado deseado:</p> <p>El backend permite al administrador registrar, visualizar (todos y por ID), actualizar y eliminar vendedores de forma segura, garantizando que solo los administradores puedan acceder a estas funcionalidades.</p>
<p>Evaluación:</p> <p>Se ha verificado la funcionalidad de gestión de vendedores del endpoint y la aprobación del cliente.</p>

Tabla 16 Prueba de aceptación Gestionar clientes.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-004	(ID) historia de Usuario: HU004
Nombre: Gestionar clientes	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Crear Clientes. • Actualizar Clientes. • Visualizar clientes. • Visualizar clientes por ID. • Eliminar clientes. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de los clientes.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar clientes desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de administradores. • Incluir token de autorización en la petición. • Ejecutar las operaciones de registrar, visualizar (todos y por ID), actualizar y 	

<p>eliminar clientes.</p> <ul style="list-style-type: none"> • Verificar que solo los usuarios con rol administrador puedan realizar estas operaciones.
<p>Resultado deseado:</p> <p>El backend permite al administrador registrar, visualizar (todos y por ID), actualizar y eliminar clientes de forma segura, garantizando que solo los administradores puedan acceder a estas funcionalidades.</p>
<p>Evaluación:</p> <p>Se ha verificado la funcionalidad de gestión de clientes del endpoint y la aprobación del cliente.</p>

Tabla 17 Prueba de aceptación Gestionar órdenes de compra.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-005	(ID) historia de Usuario: HU005
Nombre: Gestionar órdenes de compra	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el administrador para:</p> <ul style="list-style-type: none"> • Visualizar órdenes. • Visualizar órdenes por ID. • Actualizar estado de las órdenes. • Eliminar órdenes. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol administrador. Además, se implementa controles de acceso para garantizar que solo los administradores puedan gestionar la información de las órdenes de compra.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar órdenes de compra desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de administradores. • Incluir token de autorización en la petición. • Ejecutar las operaciones de visualizar (todas y por ID), actualizar el estado y eliminar órdenes. • Verificar que solo los usuarios con rol administrador puedan realizar estas 	

operaciones.
Resultado deseado: <p>El backend permite al administrador visualizar (todas y por ID), actualizar el estado y eliminar órdenes de compra de forma segura, garantizando que solo los administradores puedan acceder a estas funcionalidades.</p>
Evaluación: <p>Se ha verificado la funcionalidad de gestión de órdenes de compra del endpoint y la aprobación del cliente.</p>

Tabla 18 Prueba de aceptación Visualizar productos.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-007	(ID) historia de Usuario: HU007
Nombre: Visualizar productos (Vendedor)	
Descripción: El componente backend entrega varios endpoints que son utilizados por el vendedor para: <ul style="list-style-type: none"> • Visualizar productos. • Visualizar productos por código. 	
Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol vendedor. Además, se implementa controles de acceso para garantizar que solo los vendedores puedan visualizar la información de los productos.	
Instrucciones de funcionamiento: <p>Para visualizar productos desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de vendedores. • Incluir token de autorización en la petición. • Ejecutar las operaciones de visualizar todos los productos y visualizar productos por código. • Verificar que solo los usuarios con rol vendedor puedan realizar estas operaciones. 	
Resultado deseado: <p>El backend permite al vendedor visualizar todos los productos y productos por código de</p>	

forma segura, garantizando que solo los vendedores puedan acceder a estas funcionalidades.

Evaluación:

Se ha verificado la funcionalidad de visualización de productos del endpoint y la aprobación del cliente.

Tabla 19 Prueba de aceptación gestionar clientes.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-008	(ID) historia de Usuario: HU008
Nombre: Gestionar clientes (Vendedor)	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el vendedor para:</p> <ul style="list-style-type: none"> • Registrar clientes. • Visualizar clientes. • Visualizar clientes por ID. • Actualizar clientes. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol vendedor. Además, se implementa controles de acceso para garantizar que solo los vendedores puedan gestionar la información de los clientes a excepción de eliminar clientes que esa funcionalidad lo realiza el rol administrador.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar clientes desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de vendedores. • Incluir token de autorización en la petición. • Ejecutar las operaciones de registrar, visualizar (todos y por ID) y actualizar clientes. • Verificar que solo los usuarios con rol vendedor puedan realizar las operaciones permitidas. 	
<p>Resultado deseado:</p> <p>El backend permite al vendedor registrar, visualizar (todos y por ID) y actualizar clientes de forma segura, garantizando que solo los vendedores puedan acceder a estas</p>	

funcionalidades y que la operación de eliminación de clientes no esté disponible o sea denegada para este rol.

Evaluación:

Se ha verificado la funcionalidad de gestión de clientes del endpoint y la aprobación del cliente.

Tabla 20 Prueba de aceptación Gestionar órdenes de compra.

TEST DE ACEPTACIÓN “PRIMA S.A”	
(ID) Designación: TA-009	(ID) historia de Usuario: HU009
Nombre: Gestionar órdenes de compra (Vendedor)	
<p>Descripción: El componente backend entrega varios endpoints que son utilizados por el vendedor para:</p> <ul style="list-style-type: none"> • Registrar órdenes. • Visualizar órdenes. • Visualizar órdenes por ID. • Actualizar órdenes. 	
<p>Descripción: Todos los endpoints mencionados en la descripción son accesibles únicamente para los usuarios con rol vendedor. Además, se implementa controles de acceso para garantizar que solo los vendedores puedan gestionar la información de las órdenes de compra.</p>	
<p>Instrucciones de funcionamiento:</p> <p>Para gestionar clientes desde el backend se requiere:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador o Cliente Rest. • Iniciar sesión en el endpoint de autenticación de vendedores. • Incluir token de autorización en la petición. • Ejecutar las operaciones de registrar, visualizar (todas y por ID), y actualizar órdenes. • Verificar que solo los usuarios con rol vendedor puedan realizar estas operaciones. 	
<p>Resultado deseado:</p> <p>El backend permite al vendedor registrar, visualizar (todas y por ID) y actualizar órdenes de compra de forma segura, garantizando que solo los vendedores puedan acceder a</p>	

estas funcionalidades.

Evaluación:

Se ha verificado la funcionalidad de gestión de órdenes de compra del endpoint y la aprobación del cliente.

Certificado de cumplimiento de requisitos por parte de PRIMA S.A.



Quito, 04 de julio de 2025

CERTIFICADO

Yo, Elizabeth Alexandra Remache Leiva con CI. 1710471697, Administradora de la Empresa PROMOCIÓN INTERNACIONAL Y MATERIALES PRIMA S.A., certifico:

Que los Srs., **Mathías Agustín Terán Alcívar con CI. 1752003523** y **Paúl Sebastián Cabrera Cruz con CI. 1751431964**, estudiantes de la carrera de Desarrollo de Software en la Universidad EPN, llevaron a cabo su trabajo de integración curricular mediante la implementación de un "sistema automatizado para gestión de ventas en PRIMA SA", el mismo que cumple con los requerimientos establecidos y las funcionalidades definidas a lo largo del proceso del desarrollo.

Extiendo este certificado para que los interesados hagan uso de este documento como creyeren conveniente.

ATENTAMENTE,

Lcda. Alexandra Remache

PRIMA SA.

Av. Shyris N38-11 y el Telégrafo

Telf. 02 2433 841 – 02 2433 843 WathsApp 099 202 8374

contabilidad@prima.com.ec – atlas@prima.com.ec

ANEXO III

A continuación, para visualizar el Manual de Usuario del backend se debe digitar la siguiente URL.

<https://youtu.be/wnkshX29QgA>

En donde se explica de forma clara y sencilla las diversas funcionalidades del backend, así como cada uno de los perfiles que forman parte de este componente.

ANEXO IV

A continuación, se detallan las credenciales para el acceso al backend, junto con el vínculo al repositorio de GitHub que alberga el código íntegro y las instrucciones de instalación en la sección README.

Credenciales para el ingreso al back:

Para ingresar al backend en producción, se puede hacer a través de la URL siguiente:

<https://back-prima-s-a.onrender.com/>

De igual manera, la documentación correspondiente al backend puede ser consultada a través del siguiente enlace web.

<https://docsapi.mathiast.me/>

Credenciales para el perfil Administrador:

- Usuario: PrimAdmin
- Contraseña: 0BT9QHWM

Credenciales para el perfil Vendedor:

- Usuario: M@r1@F3rn@nd@8533
- Contraseña: Qwerty\$12

Repositorio del backend

El proyecto se encuentra en un repositorio de GitHub, al cual se puede acceder a través del enlace siguiente:

https://github.com/PaulCabreraEPN/BACK-PRIMA_S.A