

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN SISTEMA WEB Y APP DE E-COMMERCE PARA LA TIENDA SATURNINA

DESARROLLO DE UN BACKEND

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

DAVID SEBASTIAN BASANTES LUCERO

DIRECTOR: BYRON GUSTAVO LOARTE CAJAMARCA

DMQ, marzo 2024

CERTIFICACIONES

Yo, **DAVID SEBASTIAN BASANTES LUCERO** 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.

DAVID SEBASTIAN BASANTESLUCERO

david.basantes@epn.edu.ec

sebastian2405lucero@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **DAVID SEBASTIAN BASANTES LUCERO**, bajo mi supervisión.

Ing. BYRON LOARTE, MSc.

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.

DAVID SEBASTIAN BASANTES LUCERO

DEDICATORIA

A mi amada madre, Karina Lucero, quien ha sido mi fuente inagotable de amor, sabiduría y apoyo. Gracias por ser mi luz en los momentos oscuros y mi inspiración constante.

A mi querida hermana, Karina Basantes, cuya presencia ha añadido alegría y vitalidad a mi vida. Gracias por tu amistad, comprensión y por siempre estar a mi lado.

A mi pequeña sobrina, Rafaela Abigail Gavela, cuya inocencia y alegría han iluminado mis días. Tu presencia ha sido mi recordatorio constante de la importancia de la familia.

A mi profesor Byron Loarte, a quien agradezco por su guía experta y por saber dirigirnos con firmeza cuando fue necesario. Su dedicación a nuestra formación ha sido fundamental en este camino académico.

Esta tesis está dedicada a ustedes, quienes han sido mi sostén y motivación. Su influencia se refleja en cada palabra escrita y en cada logro alcanzado.

DAVID SEBASTIAN BASANTES LUCERO

AGRADECIMIENTO

Quiero expresar mi sincero agradecimiento a mi familia, amigos y maestro por su invaluable apoyo durante la elaboración de esta tesis, gracias por su amor y estímulo constante. A mis amigos, por su compañía y aliento inquebrantable. A mis maestros, por su guía experta y dedicación. Su apoyo fue fundamental en cada paso de este camino académico. Este logro es también suyo.

DAVID SEBASTIAN BASANTES LUCERO

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
Para el rol administrador se producen los siguientes endpoints:	3
Para el rol de cliente se producen los siguientes endpoints:	4
1.4 Marco Teórico.....	4
2 METODOLOGÍA.....	8
2.1 Metodología de Desarrollo.....	8
Roles	9
Product Owner.....	9
Scrum Master	9
Development Team	9
Artefactos	10
Recopilación de Requerimientos	10
Historias de Usuario	11
2.2 Diseño de la arquitectura	13
Patrón arquitectónico.....	13
2.3 Herramientas de desarrollo.....	14
Librerías	15
3 RESULTADOS	17
3.1 Sprint 0. Preparación del entorno de desarrollo.	17
3.2 Sprint 1. Resultados que se han obtenido en la creación de endpoints para el rol de administrador.	22
3.3 Sprint 2. Resultados que se han obtenido en la implementación de endpoints destinados al perfil del cliente.	31
3.4 Sprint 3. Evaluaciones para el componente backend.....	35

3.5 Sprint 4 Despliegue a un entorno de producción para el componente backend.
39

4	CONCLUSIONES	40
5	RECOMENDACIONES	41
6	REFERENCIAS BIBLIOGRÁFICAS	42
7	ANEXOS.....	48
	ANEXO I	49
	ANEXO II	50
	ANEXO III	83
	ANEXO IV.....	84

RESUMEN

El emprendimiento “Saturnina”, emerge en el año 2020 con un gran impacto y acogida en redes sociales, distinguiéndose por la venta y elaboración de prendas de vestir personalizadas. Sin embargo, el proceso de gestión de pedidos se lo realiza de forma manual y evidentemente esto conlleva una serie de problemas tales como pérdida de información, errores humanos al registrar solicitudes, demoras en la recuperación de datos de solicitudes anteriores, redundancia de información y una comunicación ineficiente con el equipo durante el proceso de personalización de prendas. Lo que afecta al negocio en el mercado y en su capacidad para atraer a nuevos clientes.

Para abordar la problemática anteriormente mencionada y como parte del marco de este Trabajo de Integración Curricular, se ha desarrollado y desplegado un backend dedicado a la gestión de venta de productos personalizados. Otorgando de esta manera una respuesta tecnológica a las necesidades de Saturnina y de sus potenciales clientes. De esta manera, este componente permite que los clientes puedan explorar y adquirir productos a través de un catálogo virtual y un carrito de compras, y que el cliente usuario administrador tenga un mayor control del inventario y una gestión adecuada de los pedidos, fortaleciendo de esta manera la competitividad y el posicionamiento en el mercado.

La estructura de este documento se organiza de la siguiente manera en primer lugar, se identifica la problemática asociada al negocio, el objetivo general, objetivos específicos, alcance y el marco teórico correspondiente. A continuación, se detalla la aplicación de la metodología Scrum, la arquitectura de software, librerías, herramientas y su interacción con el Framework FastAPI, así como con la Base de datos SurrealDB. Posteriormente, se ofrece los resultados que se han obtenido en cada iteración (Sprint) en el desarrollo de cada módulo del backend. Finalmente, se presentan las conclusiones y recomendaciones derivadas del proceso de planificación, desarrollo y despliegue a producción.

PALABRAS CLAVE: Backend, Python, FastAPI, SurrealDB, E-commerce.

ABSTRACT

The enterprise “Saturnina” emerges in 2020 with a significant impact and reception in social networks, which is distinguished by the sale and manufacture of customized clothing. However, the order management process is done manually and obviously this leads to a few problems such as loss of information, human error in recording requests, delays in retrieving data from previous requests, redundancy of information and inefficient communication with the team during the process of customizing garments. This affects the business in the marketplace and its ability to attract new customers.

To address the issues and as part of the framework of this Curricular Integration Work, a backend dedicated to the management of customized product sales has been developed and deployed. Thus, providing a technological response to the needs of Saturnina and its potential customers. In this way, this component allows customers to explore and purchase products through a virtual catalog and a shopping cart, and the client administrator user to have greater control of inventory and proper management of orders, thus strengthening competitiveness and market positioning.

The structure of this document is organized as follows: first, it identifies the problems associated with the business, the general objective, specific objectives, scope, and the corresponding theoretical framework. Next, it details the application of the Scrum methodology, the software architecture, libraries, tools, and its interaction with the FastAPI Framework, as well as with the SurrealDB Database. Subsequently, the results obtained in each iteration (Sprint) in the development of each backend module are presented. Finally, conclusions and recommendations derived from the planning process, development and deployment to production are presented.

KEYWORDS: Backend, Python, FastAPI, SurrealDB, E-commerce

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Saturnina es un negocio que surgió durante la pandemia y comenzó su trayectoria con la introducción de una gorra de flores bordada a mano como su primer producto. A medida que el tiempo avanzó, Saturnina amplió su inventario incorporando más productos, como camisetas y jeans personalizados. La característica distintiva de Saturnina radica en su enfoque en la personalización manual, combinando técnicas de estampado para plasmar las ideas de los clientes en sus prendas. En la actualidad, la tienda ofrece la capacidad de personalizar una amplia variedad de prendas de vestir, desde gorras hasta calzado. La gestión de cada pedido se realiza de forma manual ya sea utilizando las redes sociales o aprovechando la presencia de los clientes en el establecimiento [1].

La gestión manual de solicitudes ha ocasionado diversas problemáticas, como la pérdida de información, errores humanos al registrar solicitudes, demoras en la recuperación de datos de solicitudes anteriores, redundancia de información y una comunicación ineficiente con el equipo durante el proceso de personalización de prendas. Adicionalmente, la carencia de un catálogo digital ha tenido un impacto desfavorable en la reputación de Saturnina y en su capacidad para atraer a nuevos clientes [2].

Los avances tecnológicos han simplificado la vida diaria tanto para los individuos como para los diversos emprendimientos que van surgiendo, proporcionando una serie de ventajas y una comunicación rápida con los clientes. En este sentido, la exigencia de disponer de un sistema de venta en línea se percibe cada vez más, permitiendo a los negocios mantenerse competitivos [3]. Además, las empresas con presencia en el comercio electrónico reconocen la relevancia de las percepciones de los usuarios, siendo conscientes de que un comentario negativo podría perjudicar significativamente la percepción pública de la compañía. La satisfacción y retención de los clientes son elementos cruciales para lograr una satisfacción total del cliente, y para alcanzar este objetivo se implementan diversas estrategias, como el enfoque en la calidad, promociones, opciones de pago y servicios de entrega a domicilio, entre otros. La adopción de un comercio electrónico contribuye a cumplir con estos objetivos, aprovechando las nuevas tecnologías disponibles y futuras para mantener la

escalabilidad, permanecer competitivos en el mercado y la atracción de nuevos clientes [4].

Por lo antes mencionado, en este proyecto de Integración Curricular involucra el desarrollo y puesta a producción de un backend para facilitar la administración de pedidos y mejorar la comunicación con los clientes. Además, la utilización de herramientas digitales permite superar los desafíos asociados con la gestión manual, como la pérdida de información, errores humanos y demoras en la recuperación de datos. Además, el backend proporciona una serie de endpoints para simplificar la correspondencia y el intercambio de datos con aplicaciones del lado del cliente y móviles, así mismo brinda una sólida capa de seguridad por medio de roles para un acceso seguro a la información. Logrando de esta manera, que el negocio tenga una buena reputación y un adecuado posicionamiento dentro del mercado con tecnologías de vanguardia.

1.1 Objetivo general

Desarrollar el backend del e-commerce para la tienda SATURNINA.

1.2 Objetivos específicos

1. Definir cada uno de los requisitos y roles esenciales para el desarrollo del componente.
2. Elaborar la base de datos de múltiples modelos en consonancia con la recopilación de necesidades.
3. Programar cada uno de los endpoints del backend en función de los roles determinados.
4. Realizar pruebas para verificar la funcionalidad de cada uno de los endpoints.
5. Desplegar el backend a un ambiente de producción para su consumo.

1.3 Alcance

Debido a la pandemia y la adopción de diversos dispositivos tecnológicos, se ha observado un significativo aumento en las ventas en línea, lo cual resulta beneficioso para cualquier empresa o negocio. Asimismo, las plataformas virtuales posibilitan que los negocios ofrezcan servicios adicionales de manera más accesible y alcancen a un

público objetivo mucho más extenso. Mediante el desarrollo de un backend que es parte de un e-commerce, Saturnina tiene la capacidad de expandirse y fortalecer su posición en el mercado, atrayendo a nuevos clientes y proporcionando experiencias de compra memorables para su clientela actual [5].

En el campo de la programación el termino backend, se ha vuelto un componente necesario para el desarrollo de aplicaciones fullstack. Comúnmente, se suele llamar aplicaciones del lado del servidor y es la parte oculta que el usuario final no visualiza ya que por detrás se encuentra inmerso toda la lógica para el procesamiento de la información. Además, existen algunos lenguajes de programación que se emplean para desarrollar estas aplicaciones tales como: Java, PHP, Python, JavaScript, entre otros [6].

Por otra parte, el presente proyecto propone la codificación, pruebas y despliegue de un backend tipo e-commerce para el negocio Saturnina, permitiendo a los usuarios realizar un pedido de personalización de ropa de manera segura desde cualquier dispositivo con acceso a Internet. Además, el cliente puede revisar el catálogo de productos, el estado de su pedido, el historial de envíos, entre otros. Mientras que el administrador puede administrar todos los pedidos, productos e información inherente al negocio. Utilizando para ello, una metodología ágil de desarrollo de software, un modelo de arquitectura, una base de datos multimodelo, instrumentos y librerías que permitan codificar, realizar pruebas y la puesta a producción del backend para su consumo. Por último, existen dos roles de usuario cada uno con diferentes permisos que se describen subsiguientemente.

Roles que se establecen:

- Administrador y Cliente.

Para el rol administrador se producen los siguientes endpoints:

- Acceder y salir del backend, así como cambiar su clave de acceso.
- Visualizar y actualizar su perfil de usuario.
- Administrar categorías.
- Administrar productos.
- Administrar pedidos.

- Administrar comentario del producto.
- Observar estado pedidos.
- Administrar comentarios y/o sugerencias.

Para el rol de cliente se producen los siguientes endpoints:

- Registrarse.
- Acceder y salir del backend, así como cambiar su clave de acceso.
- Visualizar y actualizar su perfil de usuario
- Observar categorías.
- Observar productos.
- Administrar carrito de compras.
- Observar pedidos.
- Calificar el producto.
- Enviar comentarios y/o sugerencias.

1.4 Marco Teórico

Python

Es un lenguaje de programación versátil que se encuentra en aplicaciones de diversas arquitecturas, se trata de un lenguaje de uso general que es compatible con diversas plataformas. Esto implica que tiene la capacidad de ejecutarse en distintos sistemas operativos sin requerir la compilación del código. Por esta razón, se emplea en una variedad de aplicaciones, desde el desarrollo de software, análisis de datos, aprendizaje automático hasta la automatización de tareas. Por otro lado, posee un enfoque orientado a objetos, lo que facilita la configuración eficiente del código, al proporcionar una amplia gama de funciones y bibliotecas, así como una gran flexibilidad en la creación de programas complejos. La claridad y legibilidad de su sintaxis contribuyen significativamente a potenciar la comprensión y la gestión del código, así como su comunidad activa y la abundancia de Frameworks, como Django y Flask para el desarrollo web, contribuyen su estatus como un lenguaje poderoso y versátil en el mundo de la programación [7].

FastAPI

Es un Framework conocido por su velocidad en el desarrollo web con Python, se especializa en la eficiente creación de APIs. La combinación de su velocidad, sintaxis simple y enfoque en el rendimiento coloca a FastAPI en la categoría de los Frameworks más poderosos y listos para ser utilizados en el desarrollo y despliegue de APIs. Además, su capacidad para ofrecer soluciones eficaces y rápidas lo convierten en una elección sólida para proyectos que buscan maximizar la eficiencia y la agilidad en el desarrollo de APIs en Python. Esta característica lo posiciona como una herramienta potente y eficiente para aquellos que buscan desplegar APIs de tipo RESTful de manera rápida y efectiva [8].

JSON

Es un protocolo accesible que organiza la información de manera estructurada y se emplea principalmente para la transmisión de información entre servidores y clientes. Es sencillo de comprender para las personas y simultáneamente, las máquinas tienen la capacidad de entenderlo e intercambiar información de manera sencilla. [9].

JSON Web Token (JWT)

Mecanismo para transmitir información segura entre clientes y servidores en formato de objetos JSON, especialmente útil en aplicaciones web donde se requiere autenticar usuarios y compartir información de una forma segura. Su diseño simple y capacidad para transmitir información de manera compacta lo hacen ampliamente adoptado en el desarrollo de aplicaciones modernas [10].

API RESTful

Es un estilo de arquitectura que utilizan los sistemas para intercambiar información incluyendo una serie de reglas, protocolos y una serie de guías para administrar la comunicación en una red como lo es Internet, posibilitando de esta manera que los sistemas tengan una comunicación de manera escalable utilizando verbos HTTP como POST, PUT, DELETE, GET, PATCH y HEAD lo que facilita realizar operaciones en los recursos de manera efectiva [11].

SurrealDB

Es una base de datos para cualquier tipo de aplicación, el cual ofrece una implementación en la nube, así como de manera local. Además, se integra perfectamente con lenguajes enfocados al backend como lo es Python. La flexibilidad de la implementación de SurrealDB abarca desde configuraciones de un solo nodo hasta clústeres distribuidos. Esta flexibilidad permite una autenticación más personalizada a niveles de tablas, registros y campos, proporcionando de esta manera una solución muy robusta para diversos requisitos de autenticación y autorización [12].

FL0.com

Es un proveedor de servicios en la nube cuyo objetivo es facilitar la creación y despliegue de aplicaciones web de diversa índole. Su público objetivo está orientado a desarrolladores individuales o equipos de trabajo que no requieren tediosas configuraciones de infraestructura. Además, ofrece servicios auxiliares que suelen necesitar las aplicaciones, como bases de datos, almacenamiento en memoria, tareas en segundo plano, entre otras [13].

Swagger

Es una herramienta que se utiliza para la documentación de APIs y su ventaja radica en la consulta de todos los endpoints desde una interfaz amigable. Sin duda esto facilita la tarea de los desarrolladores para que comprendan cómo interactuar con las APIs sin necesidad de analizar el código fuente [14].

Pip

Es el sistema de gestión de paquetes que facilita la instalación y administración de bibliotecas y paquetes de Python. Siendo útil para instalar módulos y bibliotecas de terceros que no vienen preinstalados con Python y que el desarrollo sea mucho más rápido [15].

Pruebas de software

Son procesos necesarios que permite verificar el funcionamiento y la calidad de un producto software. Además, permite al desarrollador identificar los errores que se pueden encontrar dentro del código, el propósito principal al realizar una prueba de software es asegurar que la aplicación cumpla con todos los estándares y ofrecer al cliente un producto final de calidad [16].

Web Services

Facilitan la comunicación a través de internet permitiendo que dos máquinas puedan intercambiar información entre sí, sin importar las distinciones entre los lenguajes de programación en los que se hayan desarrollado o la plataforma en la que se estén ejecutando [17].

2 METODOLOGÍA

El enfoque del estudio de casos es una forma que se emplea para detallar un fenómeno o situación particular [18]. Su desarrollo se enfoca en la exploración detallada de un fenómeno, en lugar de centrarse en el análisis estadístico de datos preexistentes. Este método facilita la comprensión, explicación, reconocimiento de patrones y la obtención de enseñanzas prácticas. Además, el análisis de situaciones se utiliza en múltiples campos como tecnología, psicología, educación, sociología y gestión, con el fin de explorar fenómenos complicados y fijar un fundamento resistente para la toma de decisiones junto con la creación de teorías [19].

Teniendo en cuenta lo especificado previamente, este proyecto opta por emplear el análisis de casos, ya que permite una indagación más detallada sobre los desafíos que los negocios afrontan en el comercio con la venta de sus productos, así como su posición en el mercado. En consecuencia, se ha desarrollado un backend para el negocio Saturnina, con la meta de fortalecer las ventas del establecimiento, simplificando el acceso de los usuarios por medio de la utilización de tecnologías avanzadas en el desarrollo de software.

2.1 Metodología de Desarrollo

Simbolizan la comprensión de estrategias y métodos organizacionales orientados a abordar retos a través de un plan estructurado y metódico que comprende fases sucesivas, instrumentos, recursos, roles y obligaciones, junto con estándares estrictos para valorar con exactitud el progreso de los resultados que se están obteniendo. Su meta fundamental consiste en mitigar el riesgo de fracaso y alcanzar un nivel de desarrollo de software excepcional en términos de calidad, mediante una estructura organizativa que aumenta la eficacia y asegura la integración de mejores prácticas con el fin de lograr exitosamente los objetivos predefinidos [20].

Las metodologías ágiles buscan ofrecer rápidamente fragmentos de software funcional con el propósito de elevar la gratificación del cliente. Se implementa mediante ciclos breves, denominados iteraciones o Sprints, con el propósito de adaptarse rápidamente a los cambios y recibir retroalimentación en las fases iniciales [21]. La metodología ágil se adecua a la documentación del software y se adapta a las modificaciones que puedan presentarse en varias etapas del ciclo de vida del programa, en lugar de oponerse a ellos [22].

Por lo antes mencionado el desarrollo del backend utiliza la metodología ágil Scrum, ya que facilita la construcción de un software de alta calidad al centrarse en los problemas de los usuarios y priorizando los módulos de mayor valor para el negocio. Además, el impulso de un trabajo en equipo y sobre todo manteniendo una constante retroalimentación sobre el avance de las funcionalidades y posibles cambios [23]. Por último, esta metodología permite realizar reuniones constantes con el cliente para valorar el progreso del proyecto y la implementación minuciosa de cada etapa a lo largo del desarrollo del proyecto. A continuación, se presenta los roles y artefactos de la metodología en el desarrollo del componente backend.

Roles

Scrum establece un trabajo basado en roles, cada uno de los roles son importantes para que se pueda producir un software de alta calidad [24]. En el siguiente apartado, se describen cada uno de estos roles.

Product Owner

Es un individuo que se encarga de aumentar la eficacia y maximizar el valor del producto. Este rol toma decisiones sobre el trabajo a realizar y también se encarga de la comunicación con los stakeholders [25]. De esta manera, se detalla la designación de la persona en la **Tabla 2.1**, con su función respectiva en el desarrollo del proyecto del backend.

Scrum Master

Es un individuo que lidera al equipo con el objetivo de que todos los integrantes comprendan a fondo sobre la metodología Scrum y que a la vez puedan aplicar todas las reglas y procesos de manera óptima. Además, colabora estrechamente con el Product Owner en aspectos comerciales con el fin de optimizar el rendimiento de la inversión (ROI) [25]. De esta manera, se detalla la designación de la persona en la **Tabla 2.1**, con su función respectiva en el desarrollo del proyecto del backend.

Development Team

Es un grupo de trabajo compuesto por profesionales con las habilidades técnicas necesarias para contribuir al avance del proyecto. Su labor se lleva a cabo mediante Sprints, que son acciones específicas para realizarlos en períodos de tiempo corto, generalmente de una a cuatro semanas. Durante cada Sprint, el equipo construye y

entrega un incremento o mejora del producto [25]. De esta manera, se detalla la designación de la persona en el ejemplo de la **Tabla 2.1**, con su función respectiva en el desarrollo del proyecto del backend.

Tabla 2.1 Roles dentro del backend.

ASIGNACIÓN	INTEGRANTE
Product Owner	Sra. Shirley Rivadeneria.
Scrum Master	Ing. Byron Loarte, MSc.
Development Team	David Sebastián Basantes Lucero.

Artefactos

Son documentos creados y empleados como contenedores de información dentro del marco de trabajo Scrum, los cuales proporcionan una claridad sobre el trabajo, el avance y los objetivos del proyecto [26]. Para el desarrollo del proyecto del backend, se han incorporado los siguientes elementos.

Recopilación de Requerimientos

Este artefacto implica un proceso de investigación, comprensión y documentación de los requisitos precisos que un proyecto necesita desde su inicio hasta su conclusión [27]. La recolección de requisitos se presenta en la **Tabla 2.2** del proyecto y la lista integral se halla en el **ANEXO II**.

Tabla 2.2 Ejemplo de requerimientos del Product Owner.

EJEMPLO DE TOMA DE REQUERIMIENTOS		
EJEMPLO SISTEMA	EJEMPLO IDENTIFICADOR	EJEMPLO ENUNCIADO
Backend	RR-006	En calidad de administrador, es necesario crear diversos endpoints para: <ul style="list-style-type: none"> • Administrar pedidos.

Historias de Usuario

Es una explicación detallada sobre las características de un sistema software, observada desde la óptica del usuario final. Su objetivo es transmitir de forma nítida y simple cómo esa característica del software agrega valor al cliente y al producto [28]. La **Tabla 2.3** incluye un ejemplar para el proyecto y en el **ANEXO II** está disponible el listado completo.

Tabla 2.3 Formato de ejemplo de las historias de los usuarios.

EJEMPLO HISTORIA DE USUARIO	
Identificador: HU-006	Rol: Administrador
Ejemplo de la Historia: Administrar productos	
Urgencia: Alta	Riesgo desarrollo: Media
Asignación de iteración: 1	
Desarrollador asignado: David Basantes	
Detalle: Como rol administrador necesita generar varios endpoints para: <ul style="list-style-type: none"> • Administrar productos. 	
Observación: El acceso al endpoint mencionado se encuentra reservado exclusivamente para el usuario administrador y para hacerlo es necesario iniciar sesión.	

Product Backlog

Es un inventario estructurado que sintetiza todas las actividades a llevar a cabo durante la elaboración de un producto al aplicar la metodología Scrum. Se distingue por su adaptabilidad, dado que puede ser ajustado según progresa el proyecto. Cada componente del Backlog posee una explicación minuciosa de las labores a ejecutar, las cuales se llevan a cabo de acuerdo con las prioridades establecidas [29]. De esta manera, se puede visualizar la tabla que se ha definido para el Product Backlog en la **Tabla 2.4** del proyecto y en el **ANEXO II** está disponible el listado completo.

Tabla 2.4 Estructura de ejemplo del Product Backlog.

CONSTRUCCIÓN DEL PRODUCT BACKLOG				
IDENTIFICADOR	EJEMPLO HISTORIA DE USUARIO	ITERACIÓN ASIGNADA	ESTATUS	IMPORTANCIA
HU-005	Administrar categorías	2	Finalizado	Alta
HU-006	Administrar productos	2	Finalizado	Alta

Sprint Backlog

Después de la confección del Product Backlog, se realiza un listado más pormenorizado de las labores que deben ejecutarse, ordenadas y categorizadas a través de Sprints o Ciclos [30]. Cada una de estas actividades está sujeta a un plazo específico para su conclusión. La **Tabla 2.5** contiene el formato para el Sprint Backlog de este proyecto y en él **ANEXO II** se encuentra disponible el listado completo de las tareas.

Tabla 2.5 Ejemplo del Sprint Backlog.

SPRINT BACKLOG						
IDENTIFICADOR	NOMBRE	MÓDULO	ID-HU	HU	TAREA	ESTIMACIÓN
SB001	Diseño e implementación de endpoints para el usuario administrador.	Administración de categorías	HU004	Administrar categorías	<ul style="list-style-type: none"> Elaboración y ejecución de endpoints para generar, observar, suprimir y renovar las categorías. 	10H
					<ul style="list-style-type: none"> Revisión en SurrealDB. 	

					<ul style="list-style-type: none"> • Verificación de los datos 	
		Admiración de productos	HU005	Administrar productos	<ul style="list-style-type: none"> • Elaboración y ejecución de endpoints para generar, observar, suprimir y renovar los productos. • Revisión en SurrealDB. • Verificación de los datos 	

2.2 Diseño de la arquitectura

En el contexto de la producción de software, la palabra “arquitectura” alude a la organización y disposición esencial de un sistema o aplicación. Funciona como cimiento sobre el que se edifica la totalidad del software, determinando cómo los distintos elementos se vinculan y colaboran entre sí, así como la disposición de los datos. Adicionalmente, el objetivo primordial de la arquitectura es garantizar la excelencia, eficiencia, facilidad de mantenimiento y capacidad de ampliación del software. Esto implica la toma de decisiones fundamentales en cuanto a los patrones de diseño, la selección de tecnologías, la administración de información y la repartición de tareas entre los distintos segmentos [31]. A continuación, se ofrece una explicación acerca de la arquitectura que se ha empleado en backend.

Patrón arquitectónico

MVC es un esquema arquitectónico de estructuración de software que reparte las responsabilidades entre tres componentes fundamentales el Modelo, la Vista y el Controlador. Siguiendo esta división, el modelo se ocupa de administrar la información, la vista se encarga de la exhibición de la información y el controlador maneja la lógica

del negocio. Esta división fomenta la generación de componentes autónomos y reutilizables, además de estimular la cooperación entre programadores, perfeccionando así la conservación del sistema software. Adicionalmente, el patrón MVC posibilita efectuar modificaciones en la vista sin alterar la lógica subyacente, otorgando de esta manera adaptabilidad y potencial de expansión en el proceso de elaboración de software [32]. A continuación, se presenta el diseño arquitectónico que se ha utilizado en el backend tal y como se ilustra en la **Figura 2.1** incluyendo la incorporación de librerías externas y varias herramientas de desarrollo vinculadas.

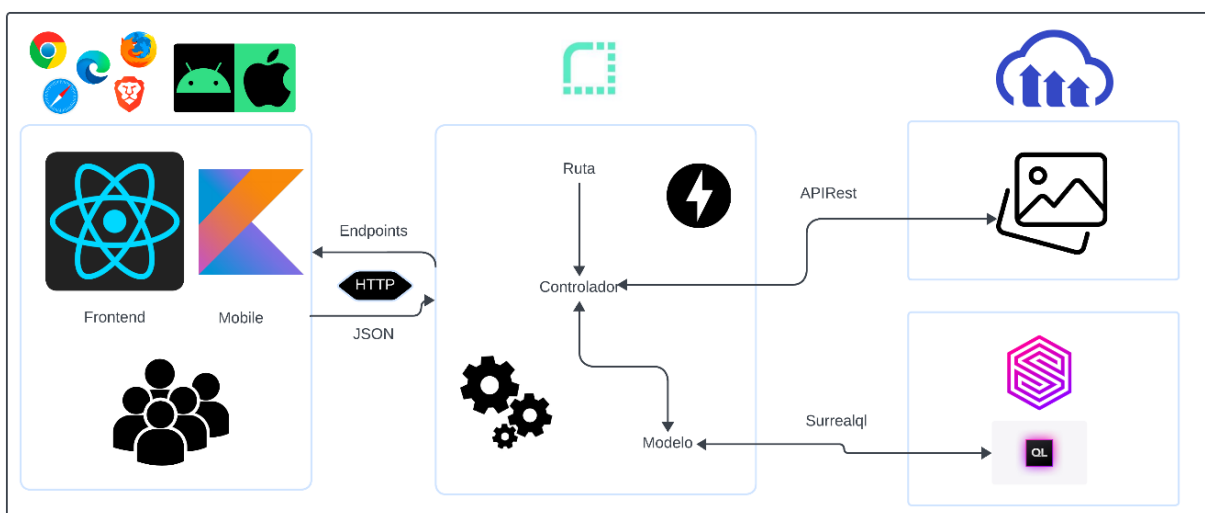


Figura 2.1 Patrón de Arquitectura – backend.

2.3 Herramientas de desarrollo

Las herramientas promueven la simplificación y aceleración del proceso de codificación. Estos instrumentos proporcionan funcionalidades sofisticadas y entornos de usuario comprensibles que permiten a los programadores enfocarse en la estrategia empresarial y disminuir el tiempo invertido en tareas monótonas. La selección de los instrumentos más adecuados potencia la eficacia y la aceleración en la creación de código en un lapso más corto [31]. A continuación, se exhibe el conjunto de herramientas que se han elegido tal y como se presenta en la **Tabla 2.6**, así como su aporte al desarrollo del proyecto backend.

Tabla 2.6 Herramientas para el desarrollo del backend.

HERRAMIENTA	JUSTIFICACIÓN
VS Code	El editor de código que se ha empleado en la creación del proyecto backend [33].
GitHub y Git	Facilita la gestión de versiones del código del proyecto backend [34].
FLO	Ofrece servicios de hospedaje confiables y que pueden expandirse de manera eficiente, posibilitando así la implementación eficaz del backend y asegurando su disponibilidad y rendimiento [13].
FastApi	Facilita la creación de APIs y la calidad del código en el backend [8].
SurrealDB	Ofrece una base de datos multimodal fiable y efectiva, diseñada para guardar y administrar datos de manera segura, así como la intención con el backend [12].
Uvicorn	Ofrece un servidor de desarrollo ágil y efectivo, diseñado para poner en marcha la API elaborada con FastAPI [35].

Librerías

Se trata de archivos desarrollados en diversos lenguajes de programación que proporcionan funcionalidades adicionales con el propósito de facilitar y optimizar los procesos dentro del código fuente. Considerando esto, se detalla en la **Tabla 2.7** las librerías o bibliotecas que se han empleado para mejorar la implementación de cada uno de los endpoints en el desarrollo del backend.

Tabla 2.7 Librerías para la integración del backend.

LIBRERÍA	DESCRIPCIÓN
python-dotenv	Facilita la importación de las variables de entorno de los archivos .env, garantizando la carga segura de datos delicados [37].
Bcrypt	Garantiza la comprobación de contraseñas cifradas [38].
PyJWT	Ofrece operaciones criptográficas seguras para el manejo de tokens [39].
python-multipart	Contribuye a simplificar la gestión de datos durante la carga de archivos [40].
pydantic	Se utiliza para definir y comprobar modelos de datos de manera simple [41].

3 RESULTADOS

En esta parte del documento se enumeran todos los resultados que se han logrado a partir de la codificación de cada uno de los endpoints correspondientes al backend. Además, se presenta los resultados que se han obtenido en cada prueba, junto con su correspondiente implementación en el entorno de producción.

3.1 Sprint 0. Preparación del entorno de desarrollo.

El conjunto de actividades que se han planificado para este Sprint son los siguientes.

- Establecimiento de los requisitos y limitaciones.
- Estructuración del proyecto backend.
- Diseño y desarrollo de la Base de datos multimodelo.
- Definición y asignación de roles de usuario.

Establecimiento de los requerimientos y limitaciones

Registro de usuarios

Se han establecido un grupo de endpoints especializados para realizar las acciones de crear perfil y verificar el correo. Además, este mecanismo garantiza una integridad de los datos, así como su validación previa a su almacenamiento en la base de datos asociada.

Acceder, salir y restablecer contraseña en el backend

Se han elaborado un grupo de endpoints especializados para el acceso, salida y restauración de contraseña, lo cual brinda una capa de seguridad robusta tanto para los administradores como para los clientes. Además, estos endpoints aseguran la autenticación segura y la gestión eficiente de sesiones de usuario.

Modificar perfil

Se ha establecido un endpoint que se dedica a la modificación de datos de perfil, permitiendo al administrador y clientes actualizar su información de manera rápida y segura. Además, este mecanismo garantiza la flexibilidad y actualización constante de los perfiles de usuario cuando lo requieran.

Crear categorías

Para el usuario administrador, se cuenta con un endpoint que simplifica las operaciones de creación, listado, actualización y eliminación de información, lo cual otorga un control completo sobre las categorías. Además, este mecanismo asegura una gestión eficaz y estructurada de las categorías almacenadas en la base de datos.

Gestionar productos

Este módulo está diseñado para las operaciones CRUD sobre los productos que el administrador puede gestionar. Además, esto asegura una gestión eficiente y actualizada, permitiendo una adaptación dinámica a las demandas del mercado.

Actualizar pedidos

Con el propósito de una administración eficiente, se ha implementado un endpoint que permite al administrador actualizar el estado de los pedidos. Simultáneamente, brinda a los clientes una visión en tiempo real de dichos cambios, mejorando la transparencia en el proceso de compra.

Crear calificaciones

El endpoint específico para la gestión de calificaciones le permite al administrador revisar y responder sugerencias, recomendaciones o felicitaciones de los clientes. Además, ofrece la capacidad de eliminar comentarios no pertinentes, manteniendo un espacio de retroalimentación constructiva.

Visualizar categorías

Mediante un endpoint dedicado, los clientes pueden explorar y visualizar todas las categorías previamente creadas por el administrador, lo que proporciona una experiencia de usuario intuitiva y personalizada.

Visualizar productos

El endpoint de visualización de productos permite a los clientes explorar de manera integral todos los productos que han sido creados por el administrador lo que facilita la toma de decisiones informada y por otra parte mejora la experiencia de compra.

Crear carrito de compra

Un endpoint específico para la gestión del carrito de compra brinda a los clientes la capacidad de organizar y revisar los productos deseados antes de finalizar la adquisición, de esta manera se optimiza el proceso de compra y aumenta la satisfacción del usuario.

Visualizar pedido

El endpoint de visualización de pedido proporciona a los clientes información detallada sobre el estado de su compra y los cambios que se han registrado por el administrador. Además, garantiza transparencia y seguimiento en cada etapa del proceso de pedido.

Calificación de productos y experiencia

El endpoint para la calificación de productos permite a los clientes expresar sus opiniones de manera respetuosa, contribuyendo a una retroalimentación valiosa. Además, esto fortalece la relación cliente-plataforma-administrador y fomenta la mejora continua de los productos que se han ofrecido.

Por último, la **Figura 3.1** exhibe un diagrama que representa las diferentes capacidades que se han asignado a los usuarios clasificados en roles de administrador y cliente. Además, el propósito subyacente de esta representación es potenciar el entendimiento y la sabiduría sobre las capacidades disponibles de los usuarios en dichos roles.

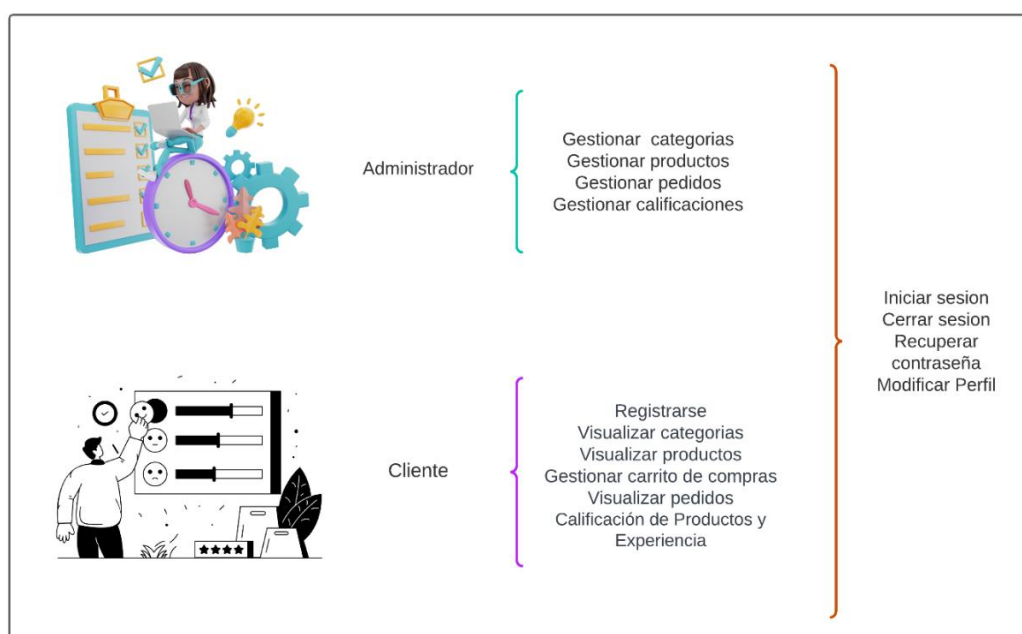


Figura 3.1 Perfiles de usuario y capacidades dentro del backend.

Estructuración del proyecto backend

La implementación de todo el componente backend se ha ejecutado a través de Visual Studio Code, el cual es un entorno de desarrollo esencial para la configuración de módulos, archivos de configuración y directorios esenciales en la creación del backend y módulos asociados. Además, la estructura detallada del proyecto backend se muestra en la **Figura 3.2**, la cual ofrece una perspectiva integral de la distribución del proyecto.

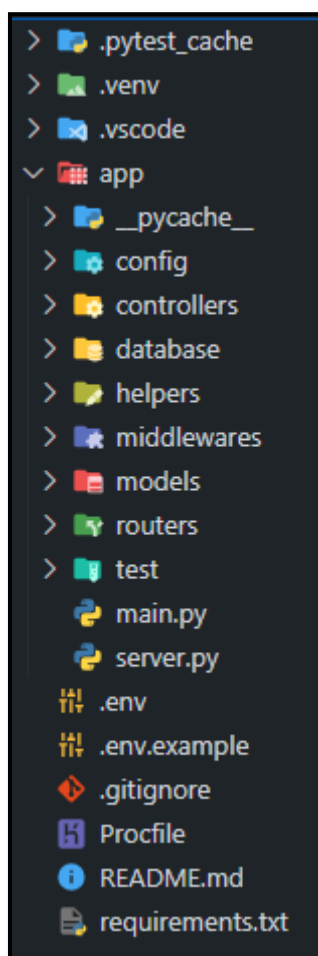


Figura 3.2 Estructuración del proyecto backend.

Diseño y desarrollo de la Base de datos multimodelo

Se ha incorporado SurrealDB, el cual es una base de datos con una orientación relacional, garantizando la integridad de la información. Además, SurrealDB no solo es compatible con datos de tipo relacional, sino también con datos no relacionales y de grafos, lo que posibilita un almacenamiento de datos eficiente. Esta aproximación facilita la administración y la obtención de información, como se representa en la **Figura**

3.3 el diseño de cada una de las tablas y el diseño completo se puede observar en el **ANEXO II**.

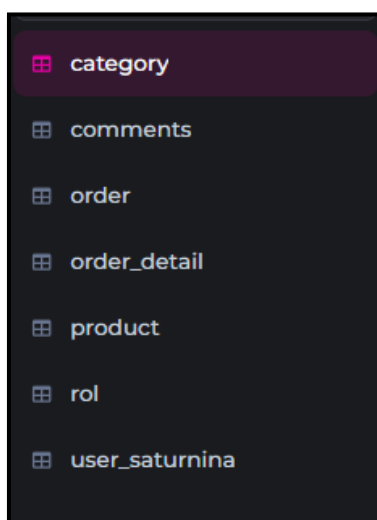


Figura 3.3 Estructura de la base de datos.

Definición y asignación de roles de usuario

Se han definido específicamente 2 roles de usuario con el propósito de asignar responsabilidades y autorizaciones particulares, con ello se garantiza un adecuado control de acceso y una eficaz administración en los distintos módulos del backend. Por último, la **Figura 3.4** se presenta la gráfica de estos roles de usuario.

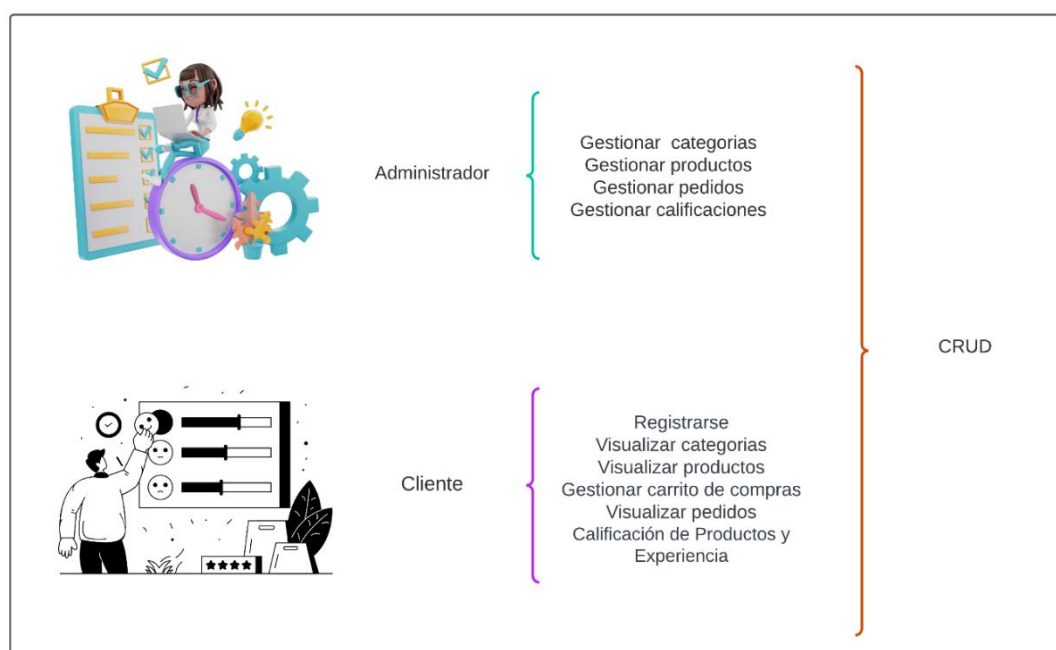


Figura 3.4 Roles que se han asignado en el backend.


```

def test_admin_login():
    payload = {
        "email": "sebastian2405lucero@hotmail.com",
        "password": "@asdaw@qweDb"
    }
    response = client.post("/api/v1/login", json=payload)
    if response.status_code == 202:
        assert "detail" in response.json()
        assert "token" in response.json()['detail']
        assert "nombre" in response.json()['detail']
        assert "apellido" in response.json()['detail']
        assert "rol" in response.json()['detail']
        assert "email" in response.json()['detail']
        assert "id" in response.json()['detail']
    elif response.status_code == 409:
        assert "Necesitas activar tu cuenta revisa tu correo para confirmar" in response.json()['detail']['msg']

app\test\test_user.py::test_login ✓

```

Figura 3.6 Test unitario de inicio de sesión.

Crear endpoints para observar y cambiar el perfil del usuario

Con la finalidad de examinar y cambiar el perfil del usuario, se han configurado varios endpoints con sus rutas asociadas, habilitando a los usuarios con roles de administrador y cliente realizar modificaciones completas en los datos personales de su perfil. Entre estos endpoints, se ha integrado uno de tipo POST, de acceso privado, orientado a la actualización de datos. Adicionalmente, se agrega otro endpoint de clase GET con un acceso privado que simplifica la obtención integral de la información del usuario que ha sido autenticado previamente. La implementación de esta acción se detalla en la **Figura 3.7** y el detalle del test unitario relacionado en la **Figura 3.8**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.

The screenshot shows a web browser interface for testing a REST API endpoint. The 'Request URL' is `http://localhost:8000/api/v1/user/user_saturinaX3Aduarv161uh97q49gus2r`. The 'Server response' shows a status code of 202 (Undocumented) and a response body of `{ "detail": { "msg": "Datos actualizados correctamente" } }`. The 'Request' section shows a curl command with headers for accept, authorization, and content-type, and a JSON body with user details like nombre, apellido, telefono, and email.

Figura 3.7 Endpoint para cambiar el perfil del usuario.


```
def test_user_update():
    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}"}
    id_user = 'user_saturnina:duarv161uh97q49gus2r '
    payload = {
        "nombre": "Sebastian",
        "apellido": "Lucero",
        "telefono": "0990095963",
        "email": "sebastian2405lucero@hotmail.com"
    }
    response = client.put(f'/api/v1/user/{id_user}', headers=header, json=payload)
    assert response.status_code == 202
    assert "Datos actualizados correctamente" in response.json()["detail"]["msg"]

app\test\test_user.py::test_user_update ✓
```

Figura 3.8 Resultado del test para cambiar el perfil de usuario.

Crear endpoints para administrar categorías

Con el propósito de administrar eficientemente las clasificaciones de cada artículo, se ha incorporado varios endpoints junto con sus rutas, el acceso a estos endpoints se habilita para el rol con función de administrador, el cual puede crear clasificaciones de forma completa. En este contexto, se han incorporado dos endpoints de acceso privado uno de tipo PUT destinado a la actualización de los datos de una categoría preexistente y otro diseñado para obtener datos de una categoría específica a través de su identificador. De igual manera, se ha integrado un endpoint de acceso privado de tipo POST con el propósito de registrar una categoría adicional y se ha activado un endpoint público GET para recuperar el listado exhaustivo de todas las categorías almacenadas, según se representa en la **Figura 3.9** y el detalle del test unitario relacionado en la **Figura 3.10**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.

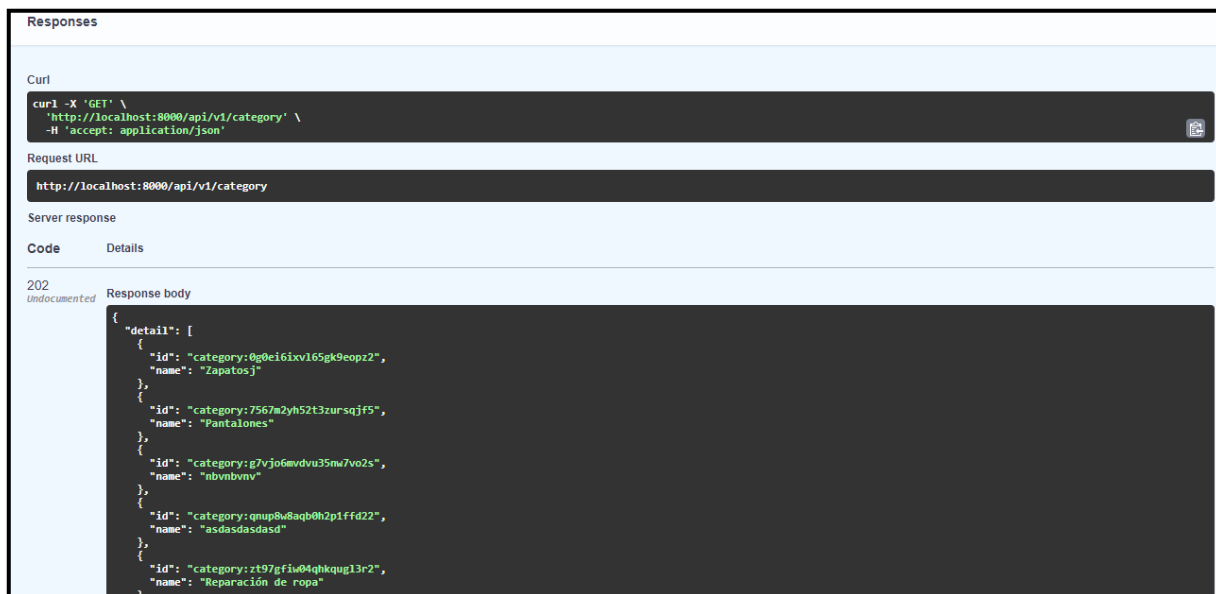


Figura 3.9 Obtener todas las categorías.



Figura 3.10 Resultado del test unitario.

Crear endpoints para administrar productos

Con la finalidad de asegurar una administración completa de los productos del negocio, se han establecido varios endpoints con sus rutas correspondientes, los endpoints habilitan al rol de administrador llevar a cabo la gestión de productos. En esta situación, se ha incorporado un endpoint reservado de clase POST para inscribir un artículo, junto con un endpoint privado de tipo PUT destinado a actualizar la información del producto. También, se han agregado dos endpoint de acceso público de tipo GET que simplifica la obtención del registro de un producto mediante su identificador, tal como se ilustra en la **Figura 3.11** y el detalle del test unitario relacionado en la **Figura 3.12**. Además,

es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.

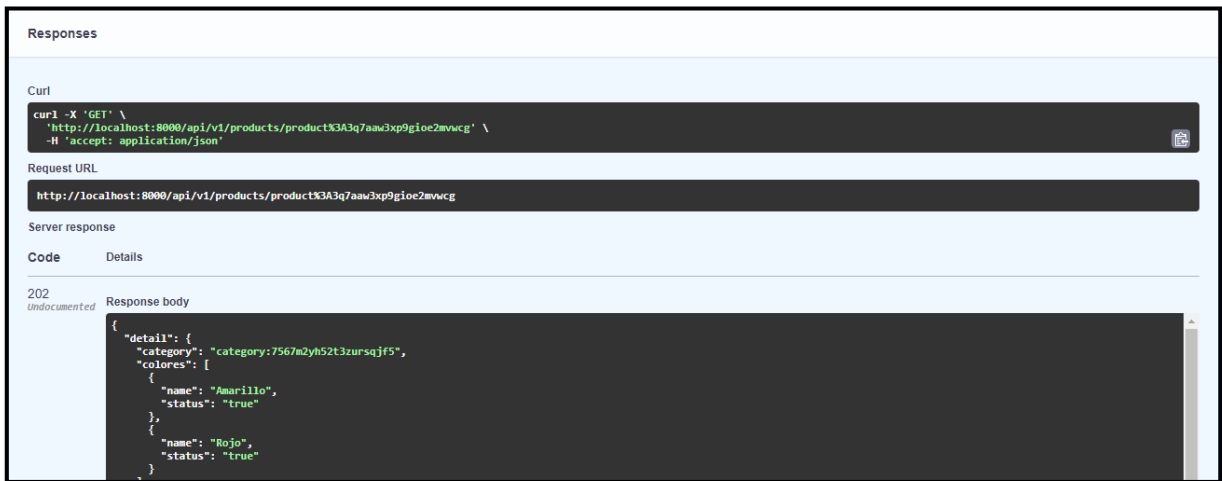


Figura 3.11 Adquirir producto con identificador.

```

def test_products():

    response = client.get('/api/v1/products')

    if response.status_code == 404:
        assert "No existe ningún producto" in response.json()['detail']['msg']

    if response.status_code == 202:
        assert "id" in response.json()['detail'][0]

app\test\test_admin.py::test_products ✓

```

Figura 3.12 Prueba unitaria de obtención de producto mediante identificador.

Crear endpoints para administrar pedidos

Con el objetivo de administrar los pedidos de manera eficiente, se han incorporado endpoints específicos. Con el objetivo de alcanzar esto, se ha establecido un endpoint restringido de tipo POST para ingresar nuevos pedidos. Además, se han introducido dos endpoints de acceso restringido de tipo GET, uno diseñado para obtener detalles específicos de una solicitud específica a través de su identificador. Además, se ha incorporado otro endpoint para obtener el listado exhaustivo de todos los pedidos

almacenados. De igual manera, se ha añadido un endpoint de acceso restringido de tipo PUT con el propósito de modificar el estado de un pedido ya existente, el cual es accedido por su identificador, tal como se ilustra en la **Figura 3.13** y el detalle del test unitario relacionado en la **Figura 3.14**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.

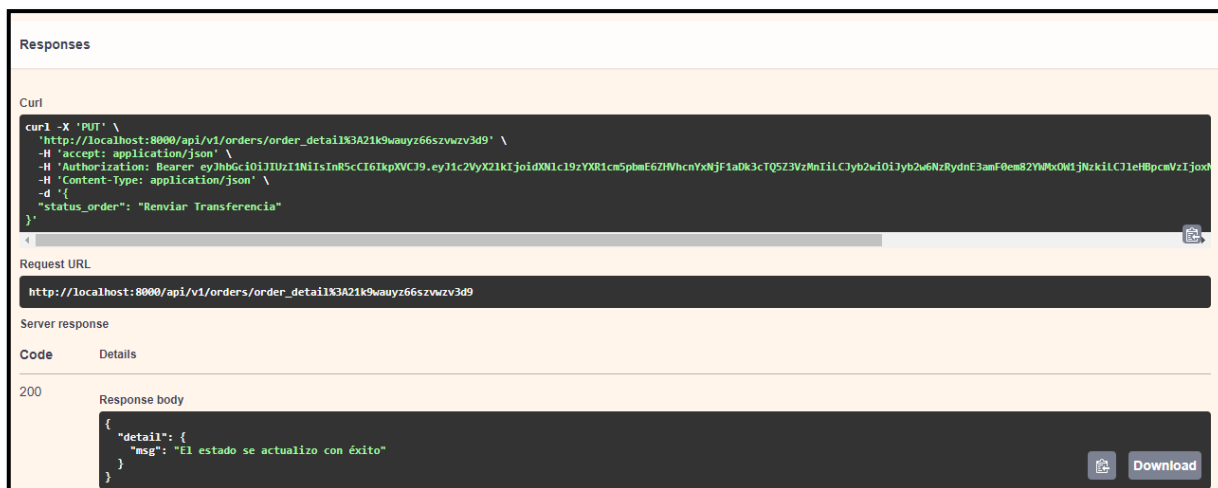


Figura 3.13 Cambiar un pedido.



Figura 3.14 Resultado de la prueba unitaria – cambiar pedido.

Crear endpoints para observar estado de pedidos

Con el objetivo de administrar de manera efectiva las órdenes, se ha implementado un endpoint privado de tipo POST para el registro de nuevas solicitudes. Además, se han desarrollado dos endpoints de tipo GET de acceso privado, uno orientado a conseguir datos pormenorizados de una orden particular a través de su ID, y otro para recopilar el listado completo de todas las solicitudes registradas. De igual manera, se ha

que permite la eliminación de un comentario o sugerencia particular. De la misma forma, se ha establecido un endpoint de clase GET con el propósito de recuperar todos los comentarios y sugerencias, tal como se ilustra en la **Figura 3.17** y el detalle del test unitario relacionado en la **Figura 3.18**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.

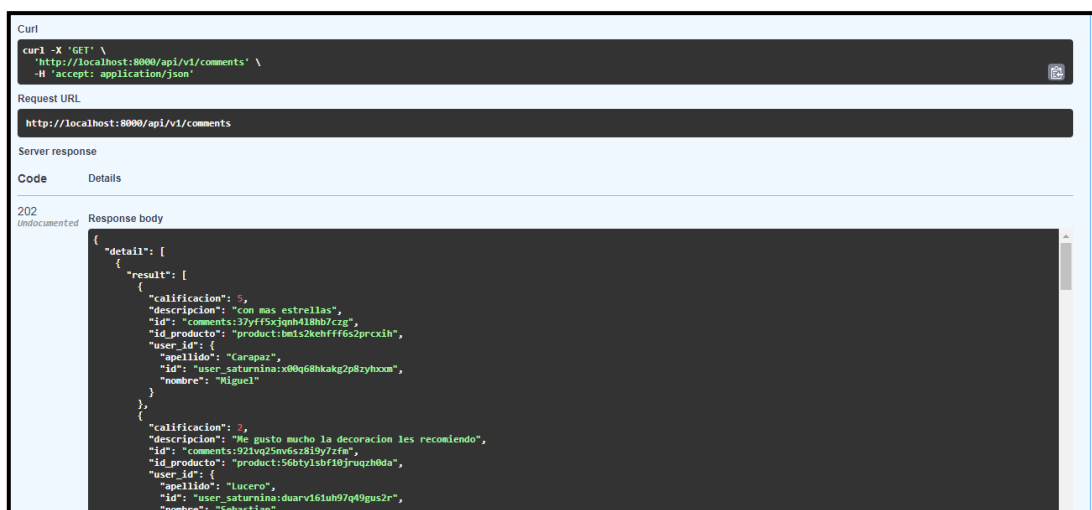


Figura 3.17 Listado de todos los comentarios creados.



Figura 3.18 Prueba unitaria para obtener todos los comentarios.

Crear endpoints para observar detalle de un pedido

Con el objetivo de permitir la visualización minuciosa de las órdenes, se han establecido diversos endpoints con sus rutas asociadas, otorgando a los usuarios con roles de administrador y cliente la habilidad de obtener la información exhaustiva sobre órdenes particulares. Estos endpoints privados de tipo GET facilitan el acceso a la información

pormenorizada de una orden, abarcando aspectos como el usuario, artículos pedidos, fecha de emisión y la situación presente de la orden. Asimismo, se ha creado un nuevo endpoint destinado a la visualización completa de todos los pedidos, como se ilustra en la **Figura 3.19** y el detalle del test unitario relacionado en la **Figura 3.20**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.

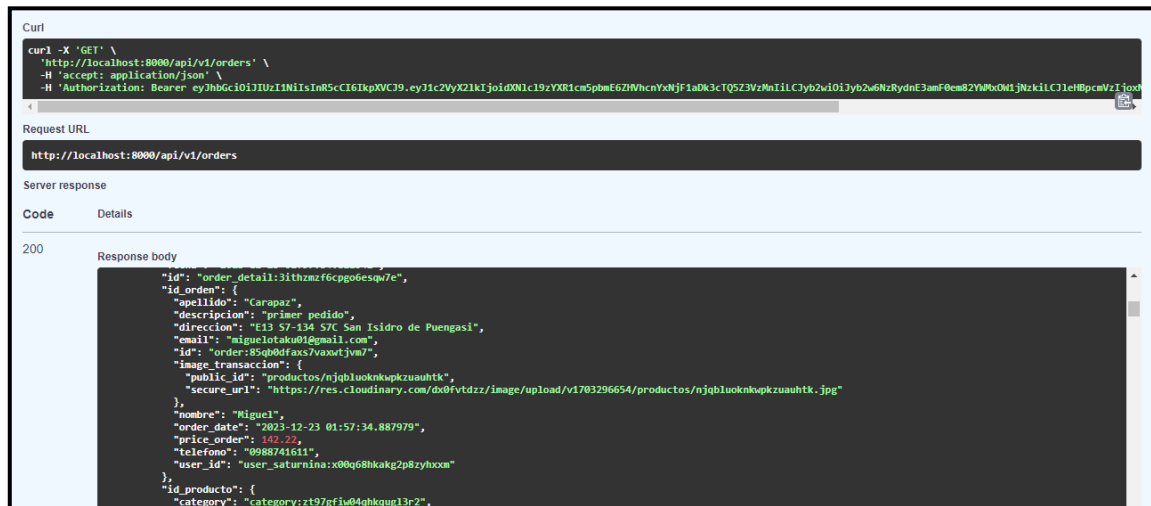


Figura 3.19 Visualización de la condición de un pedido.

```

def test_order():

    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}"}

    response = client.get('/api/v1/orders', headers=header)

    if response.status_code == 404:
        assert "No tienes ningún pedido" in response.json()['detail']['msg']

    elif response.status_code == 200:
        assert "id" in response.json()['detail'][0]['result'][0]

app\test\test_admin.py::test_order ✓

```

Figura 3.20 Prueba unitaria para observar el detalle de un pedido.

3.3 Sprint 2. Resultados que se han obtenido en la implementación de endpoints destinados al perfil del cliente.

En el Sprint actual, se han definido las siguientes tareas.

- Crear endpoints para registrarse.
- Crear endpoints para administrar pedidos.
- Crear endpoints para visualizar historial de pedidos.
- Crear endpoints para enviar observaciones o sugerencias.

Crear endpoints para registrarse

Con la intención de facilitar el procedimiento de registro, se han integrado diferentes endpoints con sus correspondientes rutas, brindando a los usuarios con rol cliente la capacidad de establecer una cuenta en el backend, mediante un endpoint accesible de manera pública de tipo POST, los usuarios tienen la posibilidad de suministrar los datos necesarios para finalizar el proceso de registro, incluyendo información como el nombre de usuario, dirección de email, clave de acceso, entre otros, tal y como se ilustra en la **Figura 3.21** y el detalle del test unitario relacionado en la **Figura 3.22**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.



Figura 3.21 Registro de un usuario.


```

def test_register():
    payload = {
        "nombre": "David",
        "apellido": "Basantes",
        "email": "sebastian2405lucero@gmail.com",
        "telefono": "090095964",
        "password": "@asdaw@qweDb"
    }

    response = client.post('/api/v1/register', json=payload)
    if response.status_code == 201:
        assert "Revisa tu correo para activar tu cuenta" in response.json()["detail"]["msg"]

    elif response.status_code == 409:
        assert 'Este email ya se encuentra en uso' in response.json()["detail"]["msg"]

app\test\test_user.py::test_register ✓

```

Figura 3.22 Prueba unitaria para registrarse.

Crear endpoints para administrar pedidos

Con la intención de administrar los pedidos, se han incorporado varios endpoints con sus rutas correspondientes, otorgando al usuario con rol cliente la habilidad de gestionar de manera eficaz todas sus órdenes. Estos endpoints, categorizados como GET, POST y PUT con restricciones de acceso, proporcionan capacidades esenciales para observar, generar, y modificar pedidos, los cuales han sido diseñados para proporcionar una experiencia sencilla en la gestión de pedidos, tal y como se ilustra en la **Figura 3.23**. Asimismo, al efectuar una solicitud de compra, el usuario tiene la opción de ejecutar la transferencia, la cual conlleva la petición de cargar una imagen del comprobante de pago. Por último, se presenta el resultado de la prueba unitaria en la **Figura 3.24**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.



```

Curl
curl -X 'POST' \
  "http://localhost:8000/api/v1/order" \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6ImlzbnVudC9y31c2VyX21k1joid0M1c19zYXR1cm5pbmE6ZHVhcnVxMjF1aDk3cTQ5Z3VzbnI1ClJyb2wiO1Jyb2w6NzRydmE3amF0eS82Y0h0b0w1JnZk1lClleIHRpcmlzIj09' \
  -H 'Content-Type: multipart/form-data' \
  -F 'data={"user_id": "user_satumimaadur-v161ah97q49gus2r", "price_order": 12.5, "products": [{"id_producto": "product:bm1s2kehff6s2prcxih", "cantidad": 1, "talla": "Talla x", "color": "F"}, {"id_producto": "product:bm1s2kehff6s2prcxih", "cantidad": 1, "talla": "Talla x", "color": "F"}], "transfer_image=@agenda.png;type=image/png}'

```

Request URL
http://localhost:8000/api/v1/order

Server response

Code	Details
201	Response body

```

{"detail": {"msg": "Pedido realizado"}}

```

Figura 3.23 Crear pedido.

```

def test_create_order():
    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}",
             "Content-Type": "multipart/form-data"}
    transfer_image_file = open("C:/Users/sebas/Downloads/icon-foreground.png", "rb")

    payload = FormData(
        data={"user_id": "user_saturnina:duarv161uh97q49gus2r", "price_order": 12.5, "products": [{"
        }
    )
    files={'transfer_image': ("icon-foreground.png", transfer_image_file)}
    response = client.post("/api/v1/order", headers=header, data=payload, files=files)
    assert response.status_code == 201
    assert "Pedido realizado" in response.json()["detail"]["msg"]

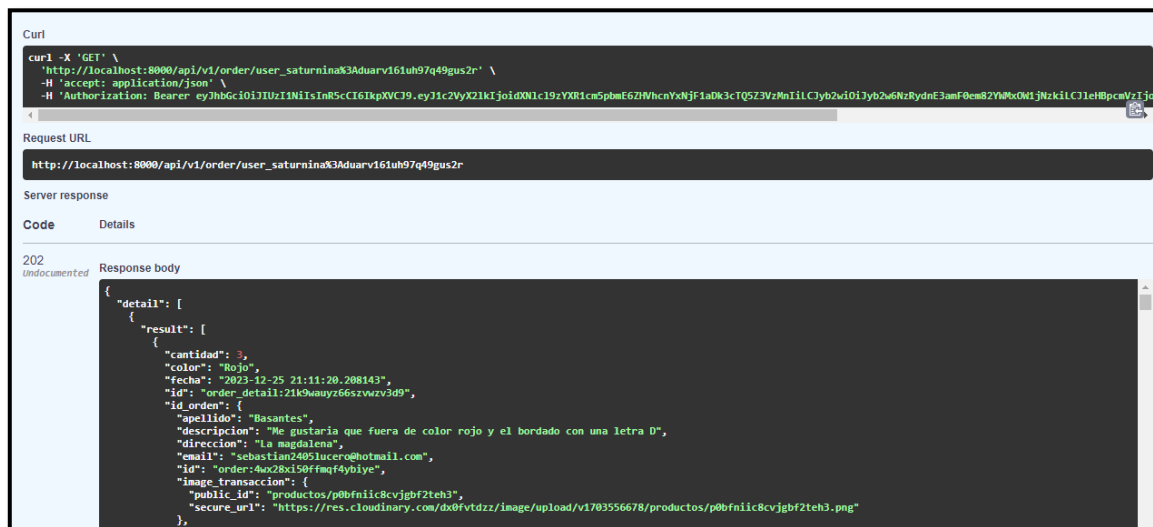
```

app\test\test_user.py::test_create_order ✓

Figura 3.24 Prueba unitaria para crear un pedido.

Crear endpoints para visualizar historial de pedidos

Con la finalidad de revisar el historial de pedidos, se han incorporado endpoints con sus rutas asociadas, brindando al usuario con rol cliente la capacidad de examinar el listado íntegro de las solicitudes de pedidos. Por medio de un endpoint privado de tipo GET, cuenta con la capacidad de acceder a un resumen integral de todas las órdenes anteriores, que engloba datos relevantes como el código de la orden, los artículos pedidos, la fecha de emisión y la situación actual de cada orden. Esto se muestra de manera detallada en la **Figura 3.25** y el detalle del test unitario relacionado en la **Figura 3.26**. Además, es crucial destacar que todo el procedimiento se describe meticulosamente en el **ANEXO III**.



```

Curl
curl -X 'GET' \
  'http://localhost:8000/api/v1/order/user_saturnina3Aduarv161uh97q49gus2r' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IWR5LmludWUiLCJ1eWciOiJ0aWQxN1c1YXR1cm5pbmE6ZWhhcnVhcnJf1aDk3cTQ5Z3VzMaIiLCJyb2wiOiJyb2w6NzRydmlE3amF0em82YW90WjJnZk1lCl1hH0pcaVz1jd'

Request URL
http://localhost:8000/api/v1/order/user_saturnina3Aduarv161uh97q49gus2r

Server response
Code    Details
202
Undocumented
Response body
{
  "detail": [
    {
      "result": [
        {
          "cantidad": 3,
          "color": "Rojo",
          "fecha": "2023-12-25 21:11:20.208143",
          "id": "order_detail:21k9wauryz66szvzv3d9",
          "id_order": {
            "apellido": "Basantes",
            "descripcion": "Me gustaria que fuera de color rojo y el bordado con una letra D",
            "direccion": "La Magdalena",
            "email": "sebastian2409lucero@hotmail.com",
            "id": "order:4ox28x159ffmqf4ybiye",
            "image_transaccion": {
              "public_id": "productos/p0bfniic8cvjgbf2teh3",
              "secure_url": "https://res.cloudinary.com/dx0fvtdzz/image/upload/v1703556678/productos/p0bfniic8cvjgbf2teh3.png"
            }
          }
        }
      ]
    }
  ]
}

```

Figura 3.25 Historial de pedidos.


```

def test_comments_create():
    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}"}
    payload = {

        "descripcion": "Me gusto mucho la decoracion les recomiendo",
        "user_id": "user_saturnina:duarv161uh97q49gus2r",
        "id_producto": "product:3q7aaw3xp9gioe2mvweg",
        "calificacion": 4

    }
    response = client.post("/api/v1/comments",json=payload,headers=header)
    if response.status_code == 422:
        assert "No puedes realizar mas comentarios de este producto" in response.json()['detail']['msg']
    elif response.status_code == 201:
        assert "Tu comentario se ha creado" in response.json()['detail']['msg']

app\test\test_user.py::test_comments_user ✓

```

Figura 3.28 Prueba unitaria para la creación de comentarios y/o sugerencias.

3.4 Sprint 3. Evaluaciones para el componente backend.

En el Sprint actual, se han definido las siguientes tareas.

- Ejecutar y detallar los resultados de los test unitarios.
- Ejecutar y detallar los resultados de los test de compatibilidad.
- Ejecutar y detallar los resultados de los test de rendimiento.
- Ejecutar y detallar los resultados de los test de aceptación.

Ejecutar y detallar los resultados de los test unitarios

Estos test unitarios se enfocan en evaluar componentes individuales del código, examinando tanto su funcionalidad aislada como su interacción con otros elementos del sistema software. Estas pruebas son cruciales para asegurar la solidez del código y permiten identificar fallos y problemas antes de ser implementados en un ambiente de producción [37]. Por tal motivo, se ha optado por emplear la biblioteca pytest para efectuar pruebas unitarias en Python, debido a su enfoque directo y eficiente, de esta manera la **Figura 3.29** ilustra el fragmento de código diseñado para recuperar un usuario mediante un identificador único y resultado de ejecutar dicha prueba. Además, para una visión completa el **ANEXO II** contiene un desglose de todas las pruebas adicionales junto con sus resultados específicos.

```

def test_user_detail_id():
    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}"}
    id_user = 'user_saturnina:q55gk5b4dqs1ugntekc5 '
    response = client.get(f'/api/v1/user/{id_user}', headers=header)

    assert response.status_code == 202
    assert "nombre" in response.json()['detail']
    assert "apellido" in response.json()['detail']
    assert "telefono" in response.json()['detail']
    assert "email" in response.json()['detail']
    assert "id" in response.json()['detail']

app\test\test_user.py::test_user_detail_id ✓

```

Figura 3.29 Código y resultado de la prueba para obtener usuario.

Los hallazgos que se han obtenido a través de las pruebas unitarias corroboran el funcionamiento correcto de cada uno de los módulos, evidenciando que se han llevado a cabo con éxito todas las validaciones necesarias.

Ejecutar y detallar los resultados de los test de compatibilidad

Estos test buscan confirmar el correcto funcionamiento del software con distintas plataformas. La meta es garantizar que el sistema ser probado opere de forma uniforme, previniendo inconsistencias y eventuales obstáculos técnicos que puedan presentarse al interactuar con varios componentes del ambiente tecnológico [38]. Por tal motivo, se ha hecho uso de los clientes HTTP Swagger, Thunder Client y Postman, los cuales han permitido realizar estas pruebas correspondientes, tal como se evidencia en la **Tabla 3.1** el cual lista los clientes HTTP que se han utilizado y los resultados que se han obtenidos en cada prueba se los puede ver en el **ANEXO II**.

Tabla 3.1 Clientes REST - Test de compatibilidad.

NOMBRE	VERSIÓN
Swagger	3.0
Thunder Client	2.16.0
Postman	10.16.0

Los resultados en los diversos clientes REST han demostrado su adecuada operatividad tanto en la rapidez de respuesta, así como en la correcta exhibición de los datos.

Ejecutar y detallar los resultados de los test de rendimiento

Estos test se diseñan para medir la eficiencia y la resistencia del sistema software ante condiciones de uso intensivo a través de una cantidad considerable de peticiones o volúmenes de datos para evaluar su comportamiento, rapidez de reacción y firmeza. El objetivo es comprobar que el sistema pueda gestionar picos de alta demanda sin inconvenientes, identificar posibles puntos críticos, descubrir errores y mejorar la escalabilidad [39]. Para este propósito, se ha seleccionado la herramienta locust el cual ofrece un método accesible para llevar a cabo esta prueba. Además, el resultado y el análisis correspondiente se ilustra en la **Figura 3.30** y los detalles adicionales de los test restantes se encuentran en el **ANEXO II**.

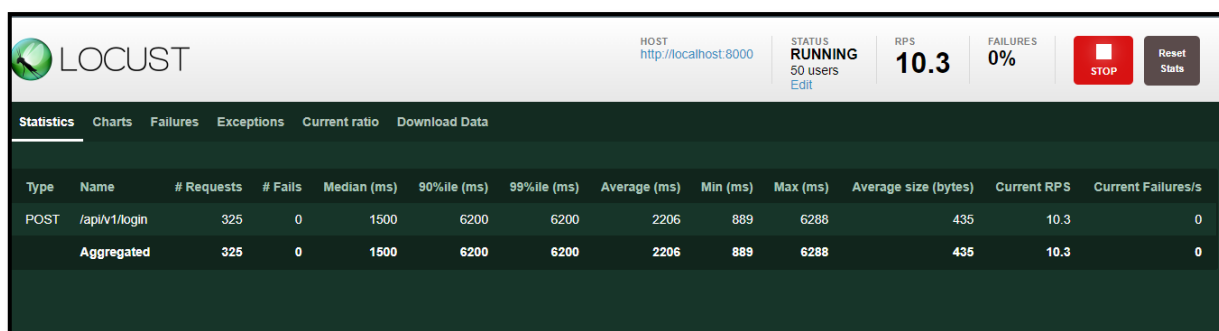


Figura 3.30 Test de rendimiento al componente login.

Se ha realizado una evaluación para emular la actividad de 50 usuarios, con una frecuencia de acceso de 5 usuarios cada segundo. Los datos que se han recabado de la prueba de inicio de sesión revelan que las respuestas por parte del backend han cumplido con todas las expectativas.

Ejecutar y detallar los resultados de los test de aceptación

Estos test aspiran a verificar que el sistema informático cumpla con las condiciones y anticipaciones fijadas por el dueño del producto final. El objetivo es asegurar que la ejecución del proyecto sea conforme a los requisitos que se han definido al comienzo del proyecto [40]. Como consecuencia, se exhibe el análisis de una prueba de

aceptación en la **Tabla 3.2** junto con su resultado, mientras que los pormenores de las demás pruebas se encuentran accesibles en el **ANEXO II**.

Tabla 3.2 Ejemplo de prueba de aceptación - Administración categorías.

EJEMPLO DE PRUEBA DE ACEPTACIÓN	
Identificador: PA-005	Ejemplo de Historia de Usuario: HU005
Nombre: Administrar categorías	
<p>Descripción: En el backend, tanto el cliente como el administrador deben crear varios endpoints para:</p> <ul style="list-style-type: none"> • Crear • Alterar • Borrar • Observar la lista de categorías 	
<p>Pasos por seguir:</p> <p>Para poder administrar las categorías:</p> <ul style="list-style-type: none"> • Acceder al backend con un usuario con el rol de administrador. • Ir al endpoint de visualizar las categorías y dar ejecutar. • Ir al endpoint de crear categorías llenar los datos del formulario y dar en ejecutar. • Ir al endpoint de actualizar categorías poner el id de la categoría a actualizar y dar en ejecutar. • Ir al endpoint de eliminar categorías poner el id de la categoría y dar en ejecutar no se eliminará si un producto está ligado a la categoría. 	
<p>Resultado esperado:</p> <p>Como administrador necesita realizar un CRUD completo sobre todas las categorías, junto con las validaciones establecidas</p>	
<p>Evaluación de la prueba:</p> <p>El administrador ha logrado administrar las categorías y por ello la prueba es validada junto con la aprobación del Product Owner.</p>	

Después de efectuar la evaluación descrita, se ha confirmado que cada uno de los módulos satisface las condiciones establecidas al inicio del proyecto y, adicionalmente, se ha recibido la aprobación completa del Product Owner.

3.5 Sprint 4 Despliegue a un entorno de producción para el componente backend.

A continuación, se detallan la única actividad asignada al Sprint, la cual, una vez finalizada la etapa de programación, pruebas y aceptación de la dueña de la tienda, se realiza la implementación a un entorno de producción utilizando la plataforma FLO.com, la cual es una plataforma especializada en el alojamiento web, ofreciendo diversos servicios destinados a facilitar la creación, implementación y administración de aplicaciones del lado del servidor [13].

La URL de acceso al componente backend es la siguiente:

<https://api-saturnina.1.us-1.fl0.io/api/v1/docs>

Finalmente, la propietaria de la tienda de personalización "Saturnina" ha generado un documento oficial que atestigua el cumplimiento cabal de los requerimientos y especificaciones del componente backend que se han definido al comienzo del proyecto. Este documento se encuentra detallado en el **ANEXO II**.

4 CONCLUSIONES

En este apartado, se exponen las conclusiones derivadas de todo el proceso de desarrollo del componente.

- Gracias al registro de la recopilación de requisitos como parte de la metodología ágil que se ha realizado en la fase inicial, garantiza una comprensión precisa de las actividades a llevar a cabo y previene la incorporación de funciones no solicitadas que puedan influir en el incumplimiento de las metas.
- Gracias a una adecuada planificación en la estructuración de la Base de Datos ha constituido un factor determinante para garantizar una adecuada alineación entre los datos y los requisitos en el contexto del desarrollo del backend. Este enfoque ha facilitado la administración eficaz de la información y fomenta la integridad de los datos con otras aplicaciones que pueden ser aplicaciones del lado del cliente o móviles.
- La organización efectiva del backend se ha logrado a través de la aplicación del diseño arquitectónico Modelo-Vista-Controlador (MVC). Este patrón de software ha permitido simplificar el código al separar las diversas características en segmentos de código y establecer una disposición de archivos bien definida. Lo que posibilita que el backend pueda escalar para abordar funciones más complejas.
- La codificación de los endpoints y componentes del lado del servidor en base a roles ha resultado esencial para asegurar de esta manera la seguridad y el control de todo el componente, facilitando de esta manera, una gestión más eficaz de los recursos y datos otorgados por medio rutas bien definidas en Swagger.
- Gracias a las pruebas se ha verificado toda la funcionalidad de los endpoints, así como la detección de posibles errores y que los mismos sean corregidos antes de que el backend sea desplegado a producción.
- Implementar el backend en un entorno de nube, como FLO, ofrece múltiples ventajas, incluyendo escalabilidad, flexibilidad y disponibilidad. Al hacer uso de las características avanzadas que proporciona la nube, se simplifica la administración y supervisión del backend, resultando en una solución con capacidad de escalamiento.

5 RECOMENDACIONES

En este apartado, se exponen las recomendaciones derivadas de todo el proceso de desarrollo del componente.

- Se aconseja que se utilice las bibliotecas pydantic y pydantic_core en las versiones 1.10.13 y 2.10.1, respectivamente, lo que asegura una correcta manipulación de datos multipart en el backend. Estas versiones específicas han demostrado ser compatibles y estables, evitando posibles errores al utilizar los modelos de pydantic. Además, al utilizar estas versiones recomendadas, se facilita la ejecución automática de métodos de clases, lo cual es esencial para garantizar un funcionamiento fluido y preciso del sistema software al procesar datos multipart.
- Se aconseja invertir tiempo y recursos en la fase de elaboración de la planificación del modelo de la Base de Datos, el cual debe incluir una comprensión exhaustiva y un análisis detallado de todos los requisitos que se han recopilado, con el objetivo de identificar de manera precisa las entidades, atributos y relaciones relevantes.
- Se recomienda enfocarse en mejorar la codificación del procesamiento de imágenes, con el objetivo de mitigar posibles riesgos de ciberseguridad y garantizar un desarrollo seguro y eficiente. Al optimizar los métodos de codificación, se fortalece la resistencia del sistema software contra posibles vulnerabilidades y amenazas, contribuyendo así a la integridad y seguridad de las imágenes utilizadas en el componente backend. Este enfoque proactivo en la codificación no solo eleva el nivel de seguridad, sino que también asegura un rendimiento óptimo al trabajar con recursos visuales.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Rivadeneira, «Saturnina,» Instagram, 15 Septiembre 2020. [En línea]. Available: <https://www.instagram.com/saturnina.uio/>. [Último acceso: 4 Octubre 2023].
- [2] Tecnipesa, «¿Qué es el picking? Características, ventajas y desventajas,» Tecnipesa, 23 Marzo 2023. [En línea]. Available: <https://www.tecnipesa.com/blog/245-picking-caracteristicas-ventajas-desventajas#:~:text=Picking%20Manual&text=Desventajas%3A%20La%20flexibilidad%20de%20este,dimensiones%20o%20con%20muchas%20referencias..> [Último acceso: 16 Noviembre 2023].
- [3] A. Damen, «E-commerce internacional. Su importancia, tendencias y futuro,» Monei Blog, 13 Julio 2022. [En línea]. Available: <https://monei.com/es/blog/cross-border-ecommerce/#por-que-es-importante-el-e-commerce-internacional>. [Último acceso: 13 Octubre 2023].
- [4] Pragma, «Qué es el e-commerce y cómo funciona,» Pragma, 28 Septiembre 2023. [En línea]. Available: <https://www.pragma.co/es/blog/que-es-el-ecommerce-y-como-funciona>. [Último acceso: 13 Octubre 2023].
- [5] Limac, «La importancia del E-commerce para las empresas,» Limac, 30 Octubre 2020. [En línea]. Available: <https://limac.com.mx/la-importancia-del-e-commerce-para-las-empresas/>. [Último acceso: 16 Octubre 2023].
- [6] F. Machuca, «Backend: ¿qué es y para qué sirve este tipo de programación?,» Crehana, 19 Mayo 2022. [En línea]. Available: <https://www.crehana.com/blog/transformacion-digital/que-es-el-backend-y-como-usarlo/>. [Último acceso: 17 Octubre 2023].
- [7] R. W, «Qué es Python: conoce uno de los lenguajes de programación más populares,» Hostinger, 26 Junio 2023. [En línea]. Available: https://www.hostinger.es/tutoriales/que-es-python#Desarrollo_web. [Último acceso: 9 Noviembre 2023].

- [8] P. Vidal, «¡Acelera tus aplicaciones web con FastAPI! Descubre el poder de este moderno y potente framework para Python,» El Código Perfecto, 31 Julio 2023. [En línea]. Available: <https://elcodigoperfecto.blog/2023/07/31/accelera-tus-aplicaciones-web-con-fastapi-descubre-el-poder-de-este-moderno-y-potente-framework-para-python/>. [Último acceso: 9 Noviembre 2023].
- [9] D. A, «¿Qué es JSON?,» Hostinger, 10 Enero 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-json>. [Último acceso: 9 Noviembre 2023].
- [10] L. López Magaña, «Qué es Json Web Token y cómo funciona,» OpenWebinars, 1 Enero 2020. [En línea]. Available: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>. [Último acceso: 9 Noviembre 2023].
- [11] I. de Souza, «API rest: conoce la importancia de ese recurso para el desempeño de una página web,» Rock Content, 17 Marzo 2020. [En línea]. Available: <https://rockcontent.com/es/blog/api-rest/>. [Último acceso: 9 Noviembre 2023].
- [12] «Surreal DB, The ultimate cloud database.,» Medium, 29 Septiembre 2022. [En línea]. Available: <https://medium.com/featurepreneur/surreal-db-the-ultimate-cloud-database-78f0c2dfde1d>. [Último acceso: 9 Noviembre 2023].
- [13] «Hosting apps on Render.com,» Medium, Julio 2022. [En línea]. Available: <https://medium.com/@kanezi.com/hosting-apps-on-render-com-15eddbc18490>. [Último acceso: 9 Noviembre 2023].
- [14] «Swagger: más comodidad en el desarrollo de API,» Ionos, 17 Febrero 2021. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-swagger/>. [Último acceso: 11 Noviembre 2023].
- [15] M. Escalante, «Qué es pip,» ABC Xperts, 21 Agosto 2023. [En línea]. Available: <https://abcxperts.com/que-es-pip/>. [Último acceso: 11 Noviembre 2023].
- [16] Certus, «Descubre qué son las pruebas de software y su importancia,» Certus, 22 Enero 2021. [En línea]. Available: <https://www.certus.edu.pe/blog/pruebas-de-software-importancia/#%C2%BFQue-es-una-prueba-de-software>. [Último acceso: 11 Noviembre 2023].

- [17] Ionos, «Web services: servicios de máquina a máquina,» Ionos, 15 Abril 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/web-services/>. [Último acceso: 11 Noviembre 2023].
- [18] L. Codina, «Estudios de caso: características, tipología y bibliografía comentada,» Lluís Codina, 19 Junio 2023. [En línea]. Available: <https://www.lluiscodina.com/estudios-de-caso/>. [Último acceso: 16 Noviembre 2023].
- [19] I. R. Salvador, «Estudio de caso: características, objetivos y metodología,» Psicología y Mente, 8 Marzo 2018. [En línea]. Available: <https://psicologiaymente.com/psicologia/estudio-de-caso>. [Último acceso: 16 Noviembre 2023].
- [20] Santander Universidades, «Metodologías de desarrollo de software: ¿qué son?,» Santander Universidades, 21 Diciembre 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 16 Noviembre 2023].
- [21] Atlassian, «¿Qué es la metodología ágil?,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile>. [Último acceso: 17 Noviembre 2023].
- [22] Red Hat, «¿Qué es la metodología ágil?,» Red Hat, 19 Julio 2022. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology>. [Último acceso: 17 Noviembre 2023].
- [23] C. Salazar, «Ventajas de usar Scrum en tu proyecto,» Platzi, 2015. [En línea]. Available: <https://platzi.com/blog/ventajas-scrum/#:~:text=Scrum%20permite%20a%20los%20equipos%20de%20desarrollo%20priorizar%20los%20m%C3%B3dulos,las%20cambiantes%20necesidades%20del%20proyecto..> [Último acceso: 17 Noviembre 2023].
- [24] M. Á. De Rios, «Scrum: qué es y cómo funciona este marco de trabajo,» Global Growth Agents, 9 Mayo 2022. [En línea]. Available: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>. [Último acceso: 17 Noviembre 2023].

- [25] S. Monroy, «¿Cuáles son los roles de la metodología Scrum?,» APD, 14 Diciembre 2021. [En línea]. Available: <https://www.apd.es/roles-metodologia-scrum/>. [Último acceso: 17 Noviembre 2023].
- [26] Miro, «Una guía completa sobre los artefactos Scrum,» Miro, [En línea]. Available: <https://miro.com/es/agile/que-son-artefactos-scrum/>. [Último acceso: 18 Noviembre 2023].
- [27] «Qué es la recopilación de requisitos: definición y herramientas | Guía completa,» Visure, [En línea]. Available: <https://visuresolutions.com/es/blog/requirements-gathering/>. [Último acceso: 18 Noviembre 2023].
- [28] M. Rehkopf, «Historias de usuario con ejemplos y plantilla,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 18 Noviembre 2023].
- [29] A. Raeburn, «Qué es product backlog y guía para hacer uno con ejemplo,» Asana, 25 Agosto 2022. [En línea]. Available: <https://asana.com/es/resources/product-backlog>. [Último acceso: 19 Noviembre 2023].
- [30] Asana, «¿Qué es el trabajo pendiente del sprint o sprint backlog?,» Asana, 17 Agosto 2022. [En línea]. Available: <https://asana.com/es/resources/sprint-backlog>. [Último acceso: 18 Noviembre 2023].
- [31] P. Huet, «Arquitectura de software: Qué es y qué tipos existen,» OpenWebinars, 24 Agosto 2022. [En línea]. Available: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/#:~:text=Qu%C3%A9%20es%20la%20arquitectura%20de%20software,-Todo%20el%20mundo&text=La%20arquitectura%2C%20referida%20al%20software,paso%20previo%20a%20cualquier%20implementa>. [Último acceso: 18 Noviembre 2023].
- [32] Desarrollo web, «Te explicamos de manera general el patrón de arquitectura del software MVC (Model - View - Controller o Modelo - Vista - Controlador). Cómo se separan las distintas capas atendiendo a sus responsabilidades.,» Desarrollo web, 20 Septiembre 2023. [En línea]. Available:

- <https://desarrolloweb.com/articulos/que-es-mvc.html>. [Último acceso: 18 Noviembre 2023].
- [33] Microsoft, «Visual Studio Code,» Microsoft, [En línea]. Available: <https://code.visualstudio.com/>. [Último acceso: 19 Noviembre 2023].
- [34] Mozilla, «Git y GitHub,» Mozilla, 2 Agosto 2023. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/GitHub. [Último acceso: 19 Noviembre 2023].
- [35] F. Soto, «Curso de FastAPI: Fundamentos,» Hackmd, 2021. [En línea]. Available: <https://hackmd.io/@KET6cIXdTAWGqjNlepGr9w/rkUN1wIF#:~:text=Uvicorn%20%F0%9F%A6%84%3A%20Una%20librer%C3%ADa%20que,computadora%20se%20convierta%20en%20servidor..> [Último acceso: 19 Noviembre 2023].
- [36] SurrealDB, «The ultimate multi-model database Develop easier. Build faster. Scale quicker.,» SurrealDB, [En línea]. Available: <https://surrealdb.com/>. [Último acceso: 19 Noviembre 2023].
- [37] Keep Coding, «¿Qué son las pruebas unitarias de software?,» Keep Coding, 3 Mayo 2023. [En línea]. Available: <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>. [Último acceso: 9 Diciembre 2023].
- [38] Energy5, «La importancia de las pruebas de compatibilidad en la confiabilidad del sistema,» Energy5, 5 Diciembre 2023. [En línea]. Available: <https://energy5.com/es/la-importancia-de-las-pruebas-de-compatibilidad-en-la-confiabilidad-del-sistema>. [Último acceso: 9 Diciembre 2023].
- [39] T. «Pruebas de estrés de software: ¿qué son y para qué sirven?,» Testing it, 8 Julio 2022. [En línea]. Available: <https://www.testingit.com.mx/blog/pruebas-de-estres-de-software#:~:text=Las%20pruebas%20de%20estr%C3%A9s%20de%20software%20%E2%80%94tambi%C3%A9n%20conocidas%20como%20pruebas,de%20riesgo%20ante%20cargas%20extremas..> [Último acceso: 9 Diciembre 2023].

- [40] T. «Pruebas de aceptación de software, ¿Cuándo y por qué son necesarias?,» Testing IT, 23 Agosto 2022. [En línea]. Available: <https://www.testingit.com.mx/blog/pruebas-aceptacion-software>. [Último acceso: 9 Diciembre 2023].
- [41] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.

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.** Manual de Usuario.
- **ANEXO III.** Manual de Instalación.
- **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 anti plagio Turnitin.



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 17 de febrero de 2024

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 UN SISTEMA WEB Y APP DE E-COMMERCE PARA LA TIENDA SATURNINA elaborado por el estudiante DAVID SEBASTIÁN BASANTES LUCERO de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 02%.

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

La **Tabla I** exhibe los requisitos que se han recolectado al inicio del proyecto, donde se reflejan las solicitudes que han sido solicitados por el Product Owner.

Tabla I Recopilación de requerimientos.

RECOPIACION DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ITEM
Backend	RR-001	Como usuario cliente necesita generar varios endpoints para: <ul style="list-style-type: none"> • Registrarse
	RR-002	Como usuario administrador y cliente necesitan generar varios endpoints para: <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión • Recuperar contraseña
	RR-003	Como usuario administrador y cliente necesitan generar varios endpoints para: <ul style="list-style-type: none"> • Modificar perfil
	RR-004	Como usuario administrador necesita generar varios endpoints para: <ul style="list-style-type: none"> • Administrar categorías.
	RR-005	Como usuario administrador necesita generar varios endpoints para: <ul style="list-style-type: none"> • Administrar productos.
	RR-007	Como usuario administrador necesita generar varios endpoints para: <ul style="list-style-type: none"> • Administrar comentarios.
	RR-008	Como usuario cliente necesita generar varios endpoints para: <ul style="list-style-type: none"> • Visualizar categorías

	RR-09	Como usuario cliente necesita generar varios endpoints para: <ul style="list-style-type: none"> • Visualizar productos.
	RR-10	Como usuario cliente necesita generar varios endpoints para: <ul style="list-style-type: none"> • Administrar carrito de compras.
	RR-011	Como usuario cliente necesita generar varios endpoints para: <ul style="list-style-type: none"> • Visualizar pedidos.
	RR-012	Como usuario cliente necesita generar varios endpoints para: <ul style="list-style-type: none"> • Calificar el producto • Calificar el servicio

Historias de Usuario

Después de finalizar la Recopilación de requerimientos, se procede a implementar las Historias de usuario del backend. Desde la **Tabla II** hasta la **Tabla XIII** se puede observar un total de 12 Historias de usuario.

Tabla II Historia de usuario 1 – Registrarse.

HISTORIA DE USUARIO	
Identificador: HU-001	Usuario: Cliente
Nombre Historia: Registrarse	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Media
Iteración asignada: 3	
Responsable: David Basantes	
Descripción: El usuario cliente necesita generar varios endpoints para que se pueda registrar.	
Observación: El usuario cliente necesita registrarse para acceder a los endpoints asignados. El usuario administrador solo recibe credenciales creadas por el equipo de TI para acceder a los endpoints correspondientes.	

Tabla III Historia de usuario 2 - Iniciar sesión, cerrar sesión, recuperar contraseña.

HISTORIA DE USUARIO	
Identificador: HU-002	Usuario: Administrador y cliente.
Nombre Historia: Iniciar sesión, cerrar sesión, recuperar contraseña	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Media
Iteración asignada: 2	
Responsable: David Basantes	
<p>Descripción: El perfil administrador y cliente necesitan generar varios endpoints para:</p> <ul style="list-style-type: none"> • Iniciar Sesión • Cerrar Sesión • Modificar contraseña 	
<p>Observación: Los usuarios con perfil administrador y usuario necesitan estar registrados para acceder a los endpoints mencionados. Para cambiar la contraseña se envía un correo al e-mail del usuario respectivo.</p>	

Tabla IV Historia de usuario 3 - Modificar perfil.

HISTORIA DE USUARIO	
Identificador: HU-003	Usuario: Administrador y cliente.
Nombre Historia: Modificar perfil	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Media
Iteración asignada: 1	
Responsable: David Basantes	
<p>Descripción: Se necesitan generar varios endpoints para que el perfil administrador y cliente puedan:</p> <ul style="list-style-type: none"> • Modificar perfil 	
<p>Observación: Los usuarios con perfil administrador y usuario necesitan acceder a los endpoints para editar su perfil, los campos que pueden modificar son, nombre, apellido, correo.</p>	

Tabla V Historia de usuario 4 – Administrar categorías.

HISTORIA DE USUARIO	
Identificador: HU-004	Usuario: Administrador
Nombre Historia: Administrar categorías	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Media
Iteración asignada: 2	
Responsable: David Basantes	
Descripción: El Backend permite generar varios endpoints para que el perfil administrador pueda: <ul style="list-style-type: none"> • Administrar categorías 	
Observación: El usuario administrador es el único que puede acceder a los endpoints antes mencionados, para poder hacerlo se requiere inicio de sesión.	

Tabla VI Historia de usuario 5 – Administrar productos.

HISTORIA DE USUARIO	
Identificador: HU-005	Usuario: Administrador
Nombre Historia: Administrar productos	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Media
Iteración asignada: 2	
Responsable: David Basantes	
Descripción: El Backend permite generar varios endpoints para que el perfil administrador pueda: <ul style="list-style-type: none"> • Administrar productos 	
Observación: El usuario administrador es el único que puede acceder a los endpoints antes mencionados, para poder hacerlo se requiere inicio de sesión.	

Tabla VII Historia de usuario 6 - Administrar pedidos.

HISTORIA DE USUARIO	
Identificador: HU-006	Usuario: Administrador, empleado, cliente
Nombre Historia: Administrar pedidos	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Media
Iteración asignada: 3	
Responsable: David Basantes	

<p>Descripción: El Backend permite generar varios endpoints para que el perfil administrador pueda:</p> <ul style="list-style-type: none"> • Administrar pedidos
<p>Observación: El usuario con perfil administrador puedes visualizar y actualizar el estado del pedido.</p>

Tabla VIII Historia de usuario 7 - Administrar comentarios.

HISTORIA DE USUARIO	
Identificador: HU-007	Usuario: Administrador
Nombre Historia: Administrar comentarios	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Media
Iteración asignada: 2	
Responsable: David Basantes	
<p>Descripción: El Backend permite generar varios endpoints para que el perfil administrador pueda:</p> <ul style="list-style-type: none"> • Visualizar los comentarios y/o sugerencias • Atender comentarios y/o sugerencias 	
<p>Observación: El usuario administrador es el único perfil que puede acceder a los endpoints mencionados. Para poder ver y administrar los comentarios y/o sugerencias tiene que iniciar sesión.</p>	

Tabla IX Historia de usuario 8 – Visualizar categorías.

HISTORIA DE USUARIO	
Identificador: HU-008	Usuario: Cliente
Nombre Historia: Visualizar categorías	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Media
Iteración asignada: 1	
Responsable: David Basantes	
<p>Descripción: El Backend permite generar varios endpoints para que el perfil empleado pueda:</p> <ul style="list-style-type: none"> • Visualizar categorías 	
<p>Observación: El usuario cliente necesita visualizar las categorías creadas por el administrador</p>	

Tabla X Historia de usuario 9 – Visualizar productos.

HISTORIA DE USUARIO	
Identificador: HU-009	Usuario: Cliente
Nombre Historia: Visualizar productos	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Alta
Iteración asignada: 1	
Responsable: David Basantes	
Descripción: El Backend permite generar varios endpoints para que el perfil cliente pueda: <ul style="list-style-type: none"> • Visualizar productos 	
Observación: El usuario con perfil cliente puede acceder al endpoint para poder ver todos los productos con los que cuenta a disposición.	

Tabla XI Historia de usuario 10 – Administrar carrito de compras.

HISTORIA DE USUARIO	
Identificador: HU-010	Usuario: Cliente
Nombre Historia: Administrar carrito de compras	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Iteración asignada: 4	
Responsable: David Basantes	
Descripción: El Backend permite generar varios endpoints para que el perfil cliente pueda: <ul style="list-style-type: none"> • Administrar carrito de compras 	
Observación: El usuario cliente es el único perfil que puede acceder a los endpoints mencionados. Para poder administrar los pedidos tiene que iniciar sesión.	

Tabla XII Historia de usuario 11 – Visualizar pedidos.

HISTORIA DE USUARIO	
Identificador: HU-011	Usuario: Cliente
Nombre Historia: Visualizar pedidos	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Iteración asignada: 1	
Responsable: David Basantes	

<p>Descripción: El Backend permite generar varios endpoints para que el perfil cliente pueda:</p> <ul style="list-style-type: none"> • Visualizar pedidos
<p>Observación: El usuario cliente necesita visualizar sus pedidos y ver el estado en el que se encuentra su pedido.</p>

Tabla XIII Historia de usuario 12 – Calificar el producto.

HISTORIA DE USUARIO	
Identificador: HU-012	Usuario: Cliente
Nombre Historia: Calificar el producto.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Iteración asignada: 2	
Responsable: David Basantes	
<p>Descripción: El Backend permite generar varios endpoints para que el perfil cliente pueda:</p> <ul style="list-style-type: none"> • Calificar el producto 	
<p>Observación: El usuario cliente es el único perfil que puede crear comentarios acerca de si le gusto producto.</p>	

Product Backlog

Se proporciona la descripción integral de todo el Product Backlog y se muestra la priorización asignada a cada Historia de Usuario, considerando su relevancia para el propietario del producto y su complejidad en el proceso de desarrollo como se presenta en la **Tabla XIV**.

Tabla XIV Product Backlog.

ELABORACION DEL PRODUCT BACKLOG				
HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PROIRIDAD
HU-001	Registrarse	3	Terminado	Media
HU-002	Iniciar sesión, cerrar sesión y recuperar contraseña	2	Terminado	Media
HU-003	Modificar perfil	1	Terminado	Media

HU-004	Administrar categorías	2	Terminado	Alta
HU-007	Administrar comentarios	2	Terminado	Media
HU-008	Visualizar categorías	1	Terminado	Media
HU-009	Visualizar productos	1	Terminado	Media
HU-0010	Administrar carrito de compras	4	Terminado	Alta
HU-011	Visualizar pedidos	1	Terminado	Alta
HU-011	Calificar el producto	2	Terminado	Alta

Sprint Backlog

En la **Tabla XV** se detallan los Sprints que se han desarrollado por el backend, incluyendo una descripción de las actividades que se han realizado y el tiempo que se ha establecido para su cumplimiento, según lo establecido por el dueño del producto

Tabla XV Sprint Backlog.

ELABORACION 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 	10 H
		Módulo de inicio de sesión	HU002	Iniciar sesión, cerrar sesión y modificar contraseña	<ul style="list-style-type: none"> Diseño e implementación de endpoints para que el usuario administrador pueda iniciar sesión, cerrar sesión y recuperar contraseña. Validación de los datos requeridos. Registro en la base de datos. 	

SB-001	Diseño e implementación de endpoints para el usuario administrador.	Módulo de perfil de usuario	HU003	Modificar perfil de usuario	<ul style="list-style-type: none">• Diseño e implementación de endpoints para visualizar y modificar el perfil.• Validación de los datos requeridos.• Consulta a la base de datos	120 H
--------	---------------------------------------------------------------------	-----------------------------	-------	-----------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------

		Módulo de gestión de pedidos	HU-006	Administrar pedidos	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para crear, visualizar, eliminar y actualizar pedidos. • Validación de los datos requeridos. 	
		Módulo de gestión de calificaciones	HU-007	Administrar comentarios y/o sugerencias	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para Administrar comentarios y/o sugerencias. • Validación de los datos requeridos. • Registro en la base de datos. 	
SB-002	Diseño e implementación de endpoints para el usuario cliente.	Módulo de inicio de sesión	HU002	Iniciar sesión, cerrar sesión, registrarse y modificar contraseña	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para que el usuario empleado pueda iniciar sesión, cerrar sesión y recuperar contraseña. • Validación de los datos requeridos. • Registro en la base de datos. 	40 H

		Módulo de perfil de usuario	HU003	Modificar perfil de usuario	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para visualizar y modificar el perfil. • Validación de los datos requeridos. • Consulta a la base de datos 	
		Módulo de visualizar categorías	HU07	Visualizar categorías	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para visualizar las categorías. 	
		Módulo de visualizar productos	HU08	Visualizar productos	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para visualizar los productos. 	
		Módulo de Administrar carrito de compras	HU09	Administrar las compras que se vayan a realizar	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para poder realizar compras. 	
		Módulo de estado del pedido	HU011	Visualizar estado del pedido	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para visualizar el estado del pedido. 	

		Módulo visualizar calificación del producto	HU012	Enviar comentarios y/o sugerencias	<ul style="list-style-type: none"> • Diseño e implementación de endpoints para crear comentarios y/o sugerencias para el perfil cliente. • Validación de los datos requeridos. • Consulta a la base de datos 	
SB-004	Pruebas del Backend	<ul style="list-style-type: none"> • Pruebas unitarias • Pruebas de compatibilidad • Pruebas de rendimiento • Pruebas de aceptación 				20H
SB-005	Despliegue del Backend	<ul style="list-style-type: none"> • Despliegue del Backend en FL0.com 				20H
Documentación		<ul style="list-style-type: none"> • Trabajo de Integración curricular • Anexos 				30H
TOTAL						240H

Elaboración de la base de datos relacional

En esta sección se describen las 8 tablas de la Base de datos, la cual ha sido diseñada para administrar toda la información del backend. La **Figura I** proporciona una visión detallada de cada tabla y sus relaciones, asegurando una organización adecuada de los datos.

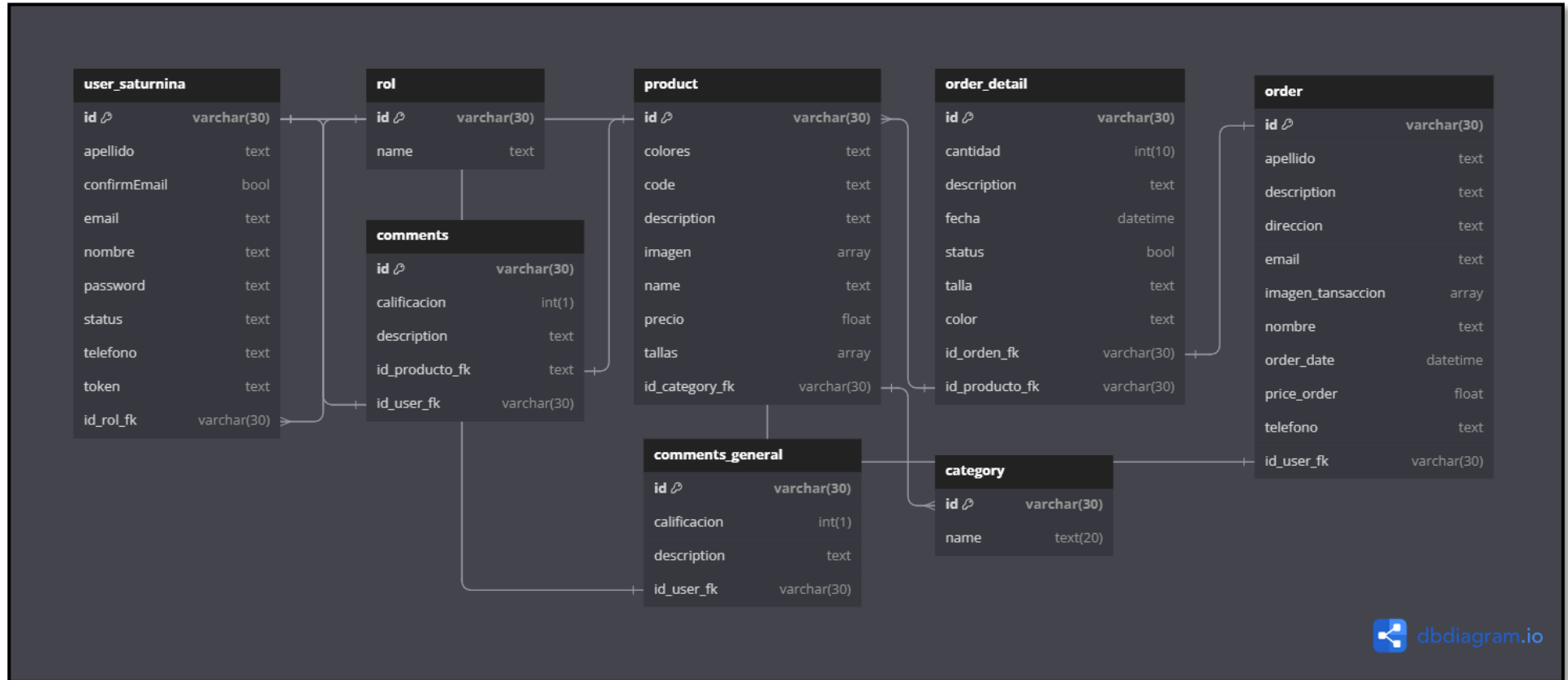


Figura I Estructura de la base de datos.

Pruebas

Después de completar la etapa de desarrollo, se realizan pruebas correspondientes con el objetivo de asegurar que el rendimiento del backend cumpla con los criterios definidos tanto en la recopilación de requisitos como en las expectativas del dueño del producto.

Pruebas Unitarias

En el marco de esta investigación, se han llevado a cabo pruebas unitarias en el backend con el objetivo de evaluar y validar diversas funcionalidades. Cada funcionalidad fue examinada mediante bloques de código específicos. El primer bloque, ilustrado en la **Figura II**, se enfoca en la eliminación de productos junto con los resultados de esta evaluación. Por otro lado, el bloque de código que aparece en la **Figura III** está diseñado para actualizar una categoría específica a través de su identificador junto con los resultados de su prueba. Estas pruebas unitarias son cruciales para asegurar la calidad y solidez del backend, ya que permiten identificar posibles errores o inconsistencias en el código que se ha implementado.

```
def test_delete_product():
    id_product = 'product:yibkcmgrmzhvs0vyjvzk'
    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}"}

    response = client.delete(f"/api/v1/products/{id_product}", headers=header)

    if response.status_code == 404:
        assert "Este producto no existe" in response.json()['detail']['msg']
    elif response.status_code == 403:
        assert "necesitas primero poner en 'Cancelado' todas los pedidos que contengan este producto " in response.json()['detail']['msg']
    elif response.status_code == 202:
        assert "Tu producto se ha eliminado" in response.json()['detail']['msg']
    elif response.status_code == 201:
        assert "Tu comentario se ha creado" in response.json()['detail']['msg']

app(test)\test_admin.py::test_delete_product ✓
```

Figura II Código eliminación producto y resultado de la prueba.

```

def test_update_category():
    id_category = 'category:m090ohysn991cam0dgjy '
    token = get_auth_token()
    header = {"Authorization": f"Bearer {token}"}
    payload = {
        "name": "Reparación de ropa"
    }

    response = client.put(f'/api/v1/category/{id_category}', headers=header, json=payload)

    if response.status_code == 400:
        assert "No puede ser igual al nombre que ya posee" in response.json()['detail']['msg']

    elif response.status_code == 400:
        assert "Este nombre de categoría ya existe en otra categoría" in response.json()['detail']['msg']

    elif response.status_code == 422:
        assert "No existe esta categoría" in response.json()['detail']['msg']

    elif response.status_code == 200:
        assert "Categoría actualizada" in response.json()['detail']['msg']

app\test\test_admin.py::test_update_category ✓

```

Figura III Código actualización de producto y resultado de la prueba.

Pruebas de compatibilidad

Se han utilizado diversos clientes HTTP, específicamente Swagger, ThunderClient y Postman, para llevar a cabo las pruebas de compatibilidad. Estos clientes se emplearon con el propósito de verificar la funcionalidad adecuada de la API en distintos entornos y confirmar la coherencia de la información entre ellos.

Para observar las pruebas que se han realizado con Swagger, se puede observar la **Figura IV**, para el resultado de la prueba inicio de sesión; en la **Figura V**, se presentan los resultados de la prueba para crear usuarios; en la **Figura VI**, se expone el resultado de la prueba para obtener un producto mediante ID; y por último en la **Figura VII**, se exhibe el resultado para eliminar una categoría mediante ID.

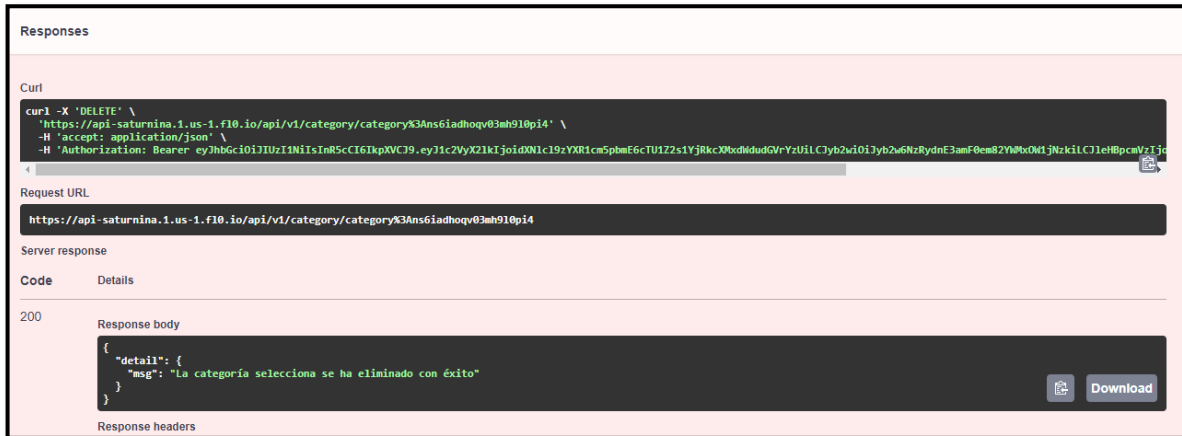


Figura VII Resultado de eliminación de categoría por id con Swagger.

Para revisar las pruebas realizadas con ThunderClient, se puede observar en la **Figura VIII**, que muestra el resultado de la prueba de inicio de sesión; en la **Figura IX**, se presentan los resultados de la prueba para crear usuarios; en la **Figura X**, se expone el resultado de la prueba para obtener un producto mediante ID; y finalmente, en la **Figura XI**, se exhibe el resultado para eliminar una categoría mediante ID.

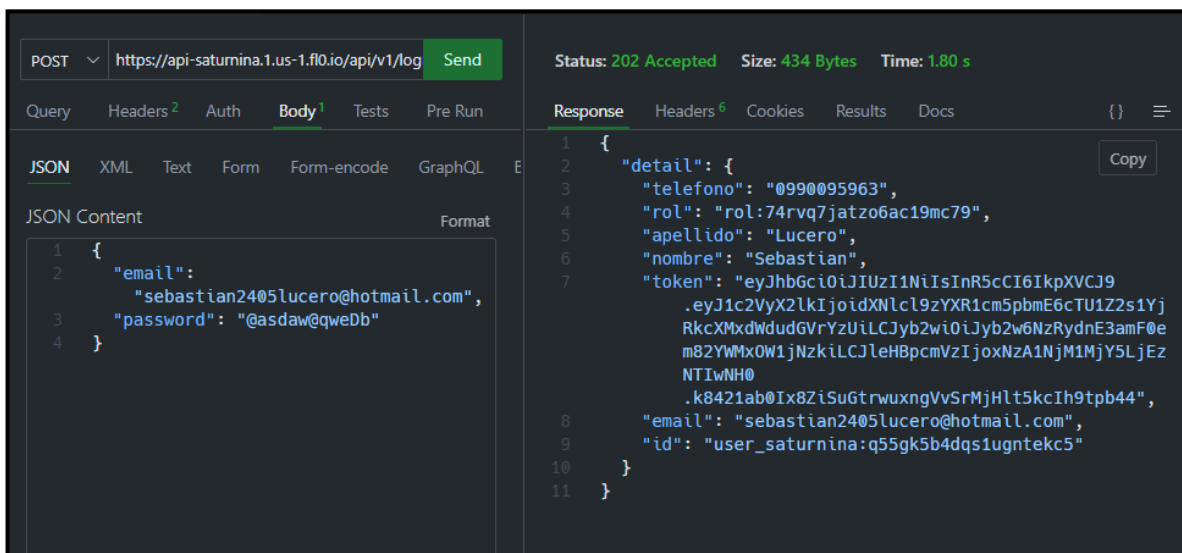


Figura VIII Resultado del login con ThunderClient

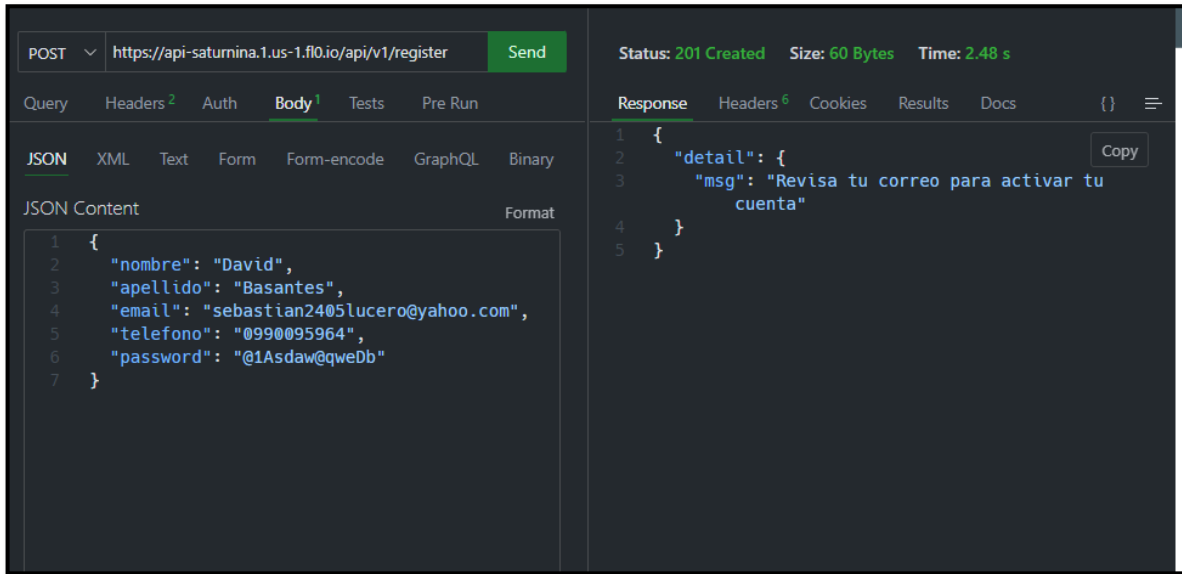


Figura IX Resultado del registro con ThunderClient

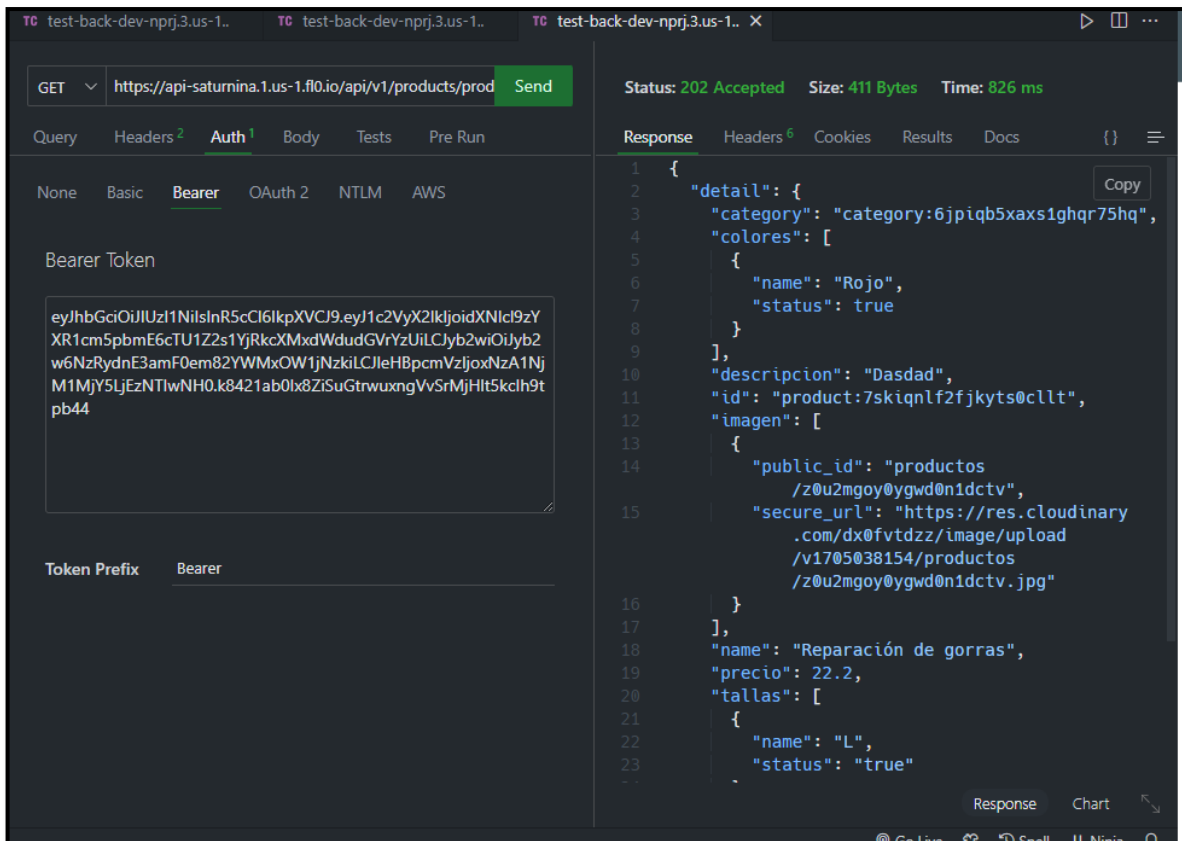


Figura X Resultado de obtener un producto con su id con ThunderClient

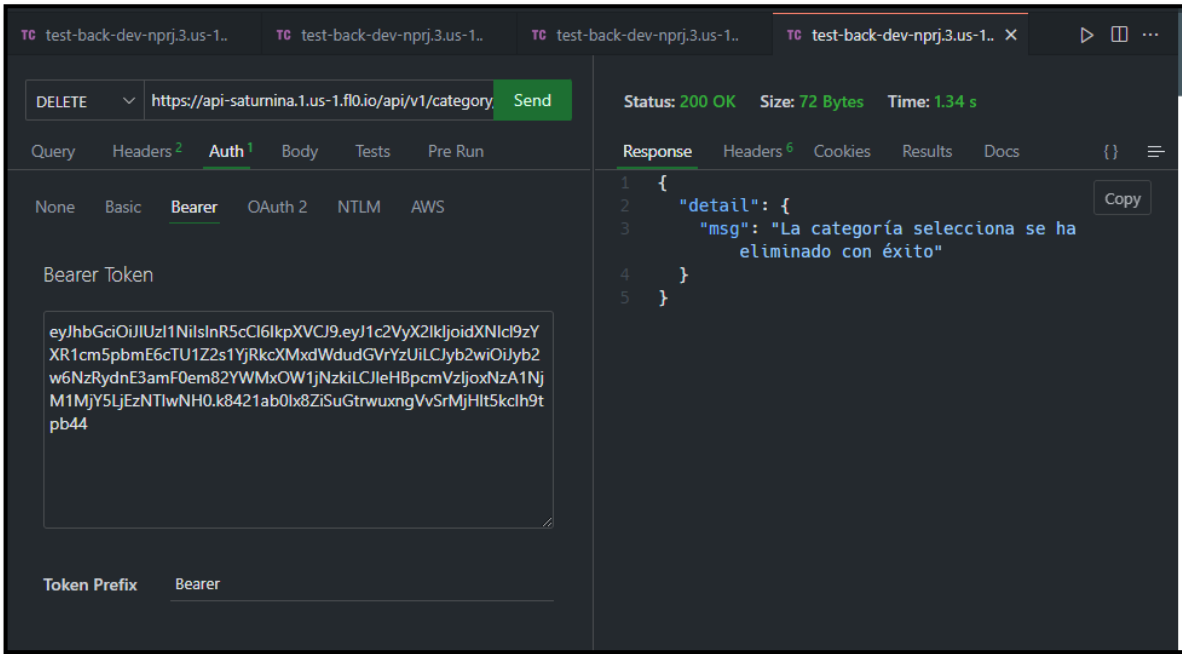


Figura XI Eliminar categoría por id con ThunderClient.

Para visualizar las pruebas realizadas con Postman, se puede referenciar la **Figura XI**, la cual muestra el resultado de la prueba de inicio de sesión; en la **Figura XII**, se presentan los resultados de la prueba para crear usuarios; en la **Figura XIV**, se expone el resultado de la prueba para obtener un producto mediante ID; y finalmente, en la **Figura XV**, se exhibe el resultado para eliminar una categoría mediante ID.

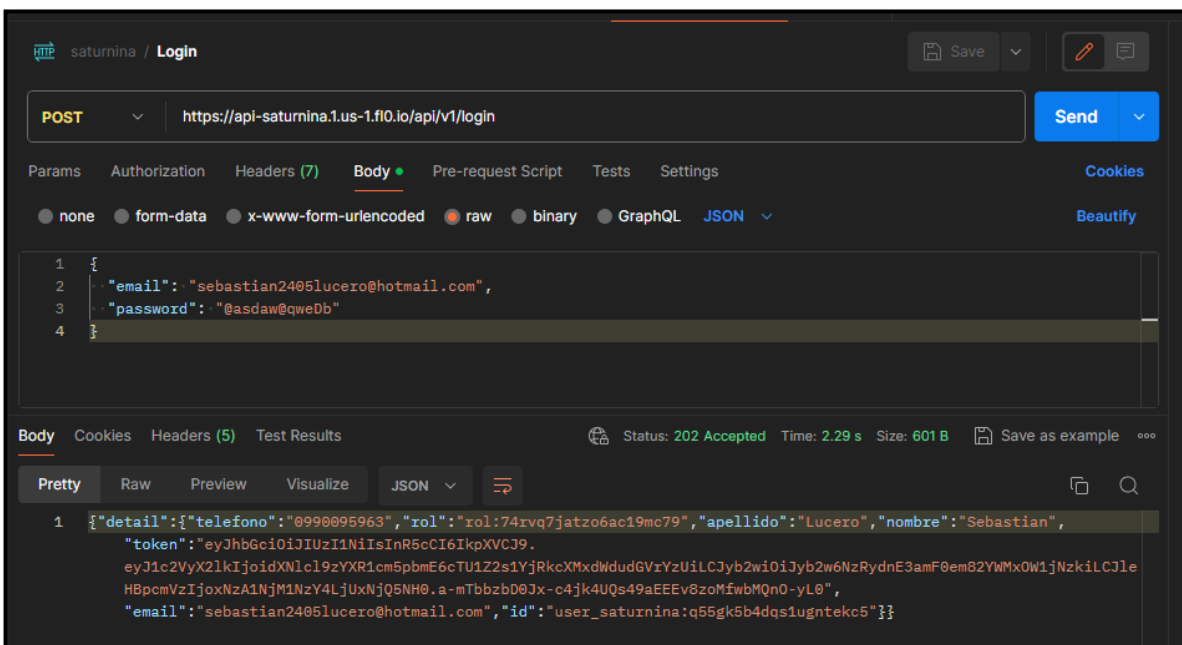


Figura XII Resultado del login con Postman.

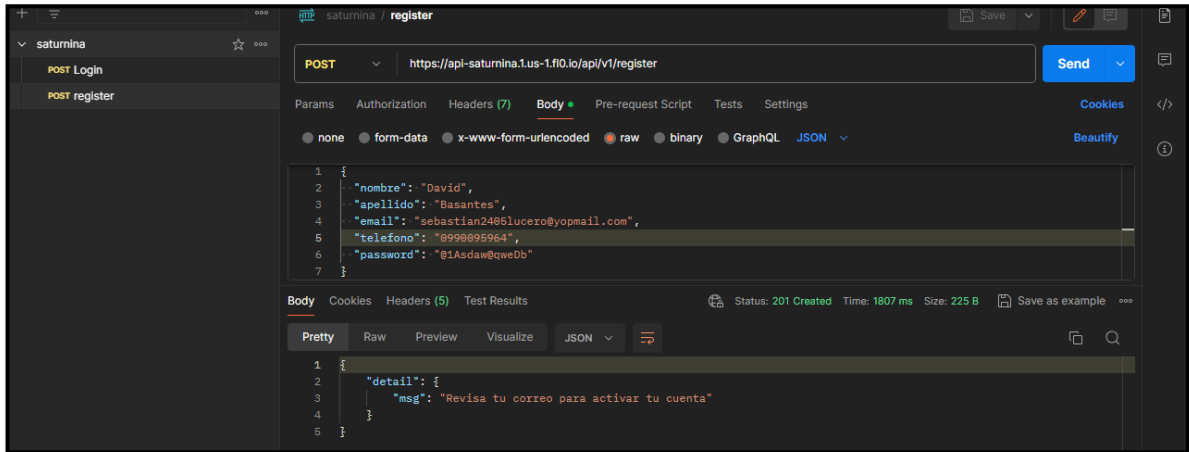


Figura XIII Resultado de registrarse con Postman.

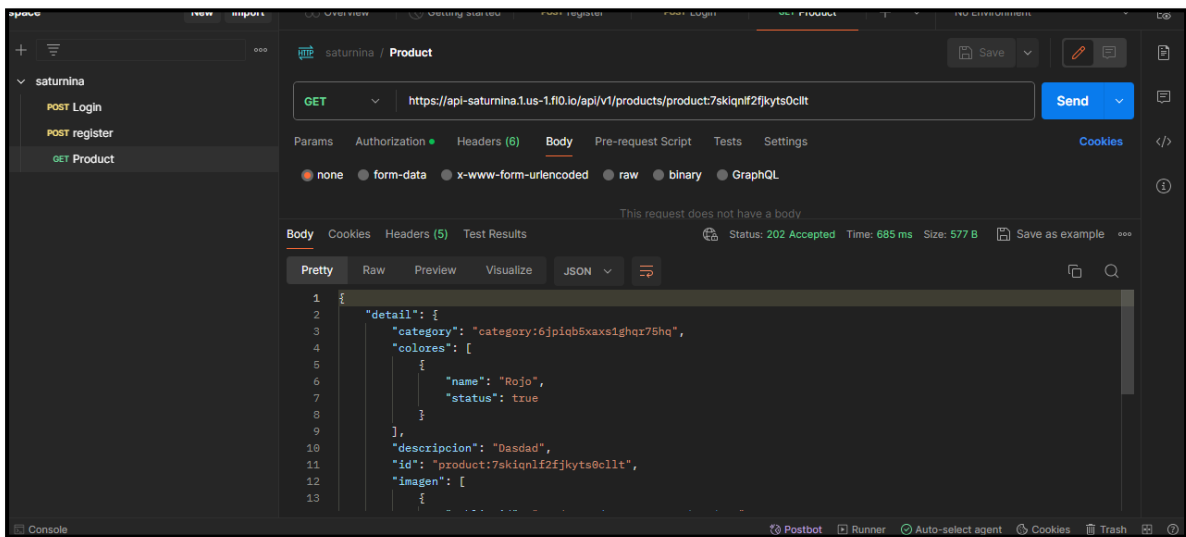


Figura XIV Resultado de obtener un producto por id con Postman.

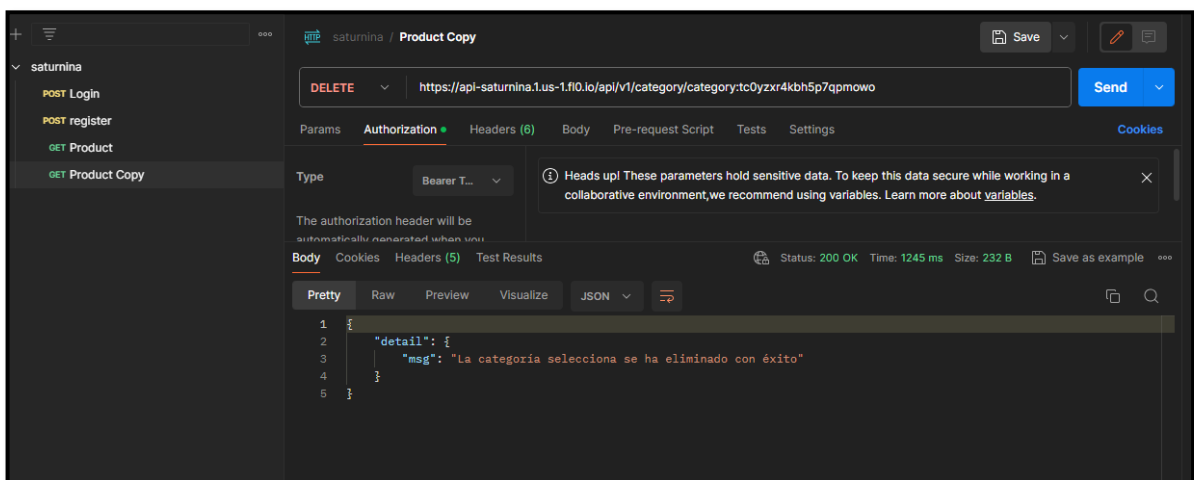
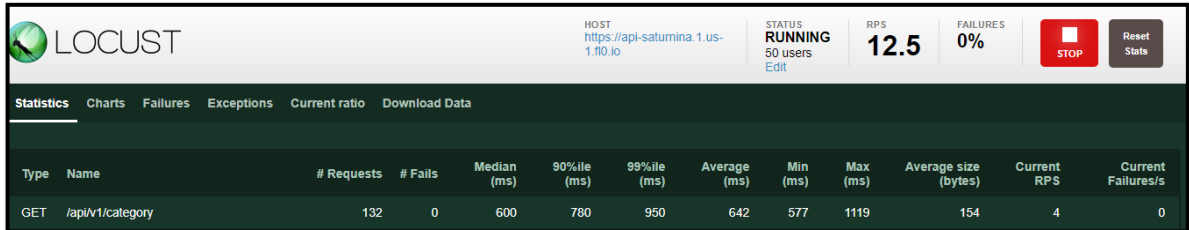


Figura XV Resultado de eliminar una categoría por id con Postman.

Pruebas de rendimiento

En esta sección se exponen tres pruebas de estrés que ilustran cómo se comportan los endpoints frente a un flujo específico de solicitudes en un intervalo de tiempo determinado. En la **Figura XVI** se visualiza la obtención de las categorías, en **Figura XVII** se valida la obtención de productos y en la **Figura XVIII** se observa la obtención de productos por id.



LOCUST HOST: https://api-saturnina.1.us-1.f0.io STATUS: RUNNING 50 users Edit RPS: 12.5 FAILURES: 0% STOP Reset Stats

Statistics Charts Failures Exceptions Current ratio Download Data

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/api/v1/category	132	0	600	780	950	642	577	1119	154	4	0

Figura XVI Obtener categorías con Locust.

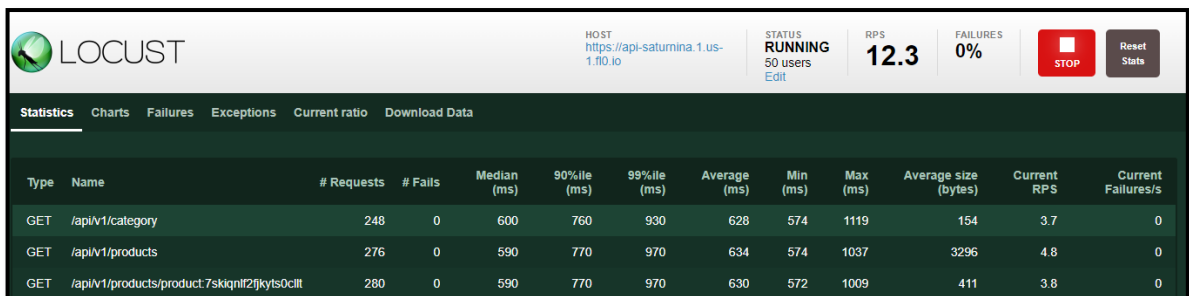


LOCUST HOST: https://api-saturnina.1.us-1.f0.io STATUS: RUNNING 50 users Edit RPS: 12.2 FAILURES: 0% STOP Reset Stats

Statistics Charts Failures Exceptions Current ratio Download Data

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/api/v1/category	190	0	600	760	950	635	575	1119	154	3.5	0
GET	/api/v1/products	200	0	600	780	930	640	574	1037	3296	4.1	0

Figura XVII Obtener productos con Locust.



LOCUST HOST: https://api-saturnina.1.us-1.f0.io STATUS: RUNNING 50 users Edit RPS: 12.3 FAILURES: 0% STOP Reset Stats

Statistics Charts Failures Exceptions Current ratio Download Data

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/api/v1/category	248	0	600	760	930	628	574	1119	154	3.7	0
GET	/api/v1/products	276	0	590	770	970	634	574	1037	3296	4.8	0
GET	/api/v1/products/product:7skiqnl2fjkyts0c1l	280	0	590	770	970	630	572	1009	411	3.8	0

Figura XVIII Obtener producto por id con Locust.

Pruebas de aceptación

A continuación, se detallan las pruebas de aceptación desde la **Tabla XVI** hasta la **Tabla XXVII**. Estas pruebas revisten una importancia crucial para describir y validar el proceso vinculado a las diversas tareas asignadas. Es imperativo que estas pruebas se ejecuten correctamente para asegurar el desarrollo adecuado de cada tarea y verificar su desempeño correcto.

Tabla XVI Prueba de aceptación 1 – Registrarse.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-001	Historia de Usuario: HU001
Nombre: Registrarse	
Descripción: El usuario cliente en el Backend deben generar varios endpoints para: <ul style="list-style-type: none"> • Registrarse 	
Pasos de ejecución: Para el registro del usuario cliente: <ul style="list-style-type: none"> • Ingresar a la URL del componente Backend en un navegador. • Ir al endpoint crear usuario. • Llenar todos los campos solicitados del formulario. • Dar clic en Execute. 	
Resultado deseado: Como cliente se necesita tener un método de tipo POST para poder registrar un nuevo usuario, junto con las validaciones establecidas	
Evaluación de la prueba: El cliente puede registrarse y por ello la prueba es validada junto con la aprobación del Product Owner.	

Tabla XVII Prueba de aceptación 2 - Inicio de sesión, cierre de sesión y modificación de contraseña.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-002	Historia de Usuario: HU002
Nombre: Iniciar sesión, cerrar sesión y modificar contraseña	
<p>Descripción: El usuario cliente y administrador en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Iniciar sesión. • Cerrar sesión. • Modificar contraseña. 	
<p>Pasos de ejecución:</p> <p>Para el inicio de sesión todos los usuarios:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Ir al endpoint login. • Colocar el email y contraseña del usuario. • Copiar el token y dar en authorize y dar en authorize • Dar clic en Execute. <p>Para cerrar sesión todos los usuarios:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente <i>Backend</i>. • Ir al apartado de authorize y dar en logout. <p>Para modificar la contraseña todos los usuarios:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente backend en un navegador. • Ir al endpoint olvidar contraseña. • Colocar el email y dar en execute. • Dirigirse al email del usuario y dar clic en el enlace enviado. • Llenar todos los campos solicitados del formulario. • Dar clic en Confirmar. • Dirigirse nuevamente al Login e Iniciar sesión con la nueva contraseña. 	
<p>Resultado deseado:</p> <p>Como cliente y administrador necesitan tener endpoints que les permita iniciar sesión cerrar sesión y modificar contraseña, junto con las validaciones establecidas</p>	

Evaluación de la prueba:

El cliente y administrador logra iniciar sesión, cerrar sesión y modificar contraseña, por ello la prueba es validada junto con la aprobación del Product Owner.

Tabla XVIII Prueba de aceptación 3 – Visualización y modificación de perfil.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-003	Historia de Usuario: HU003
Nombre: Modificar perfil	
<p>Descripción: El usuario cliente y administrador en el Backend deben generar varios endpoints para poder editar su perfil en base a los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre • Apellido • Teléfono • Email 	
<p>Pasos de ejecución:</p> <p>Para los usuarios cliente y administrador:</p> <ul style="list-style-type: none"> • Ingresar sesión en el backend en un navegador. • Ir al endpoint actualizar perfil. • Colocar el user id • Llenar todos los campos solicitados del formulario. • Dar clic en Execute. 	
<p>Resultado deseado:</p> <p>Como cliente y administrador necesitan poder editar sus datos y poder visualizarlos, junto con las validaciones establecidas</p>	
<p>Evaluación de la prueba:</p> <p>El cliente y administrador pueden modificar su perfil y por ello la prueba es validada junto con la aprobación del Product Owner.</p>	

Tabla XIX Prueba de aceptación 4 – Administrar categorías.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-004	Historia de Usuario: HU004
Nombre: Gestión de categorías	
<p>Descripción: El usuario administrador en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Crear categorías • Visualizar categorías • Actualizar categorías • Eliminar categorías 	
<p>Pasos de ejecución:</p> <p>Para el usuario administrador:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Iniciar sesión como usuario con el rol de administrador • Ir al endpoint de crear categorías y dar execute. • Ir al endpoint de visualizar categorías y dar execute. • Ir al endpoint de actualizar categorías y dar execute. • Ir al endpoint de eliminar categorías y dar execute. 	
<p>Resultado deseado:</p> <p>Como administrador necesita tener un CRUD para poder crear, visualizar, actualizar y eliminar una categoría, junto con las validaciones establecidas</p>	
<p>Evaluación de la prueba:</p> <p>El administrador logra administrar las categorías y por ello la prueba es validada junto con la aprobación del Product Owner.</p>	

Tabla XX Prueba de aceptación 5 – Administrar Productos.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-005	Historia de Usuario: HU005
Nombre: Administrar productos	

<p>Descripción: El usuario administrador en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Crear productos • Visualizar productos • Actualizar productos • Eliminar productos
<p>Pasos de ejecución:</p> <p>Para el usuario administrador:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente Backend en un navegador. • Iniciar sesión como usuario con el rol de administrador • Ir al endpoint de crear productos y dar execute. • Ir al endpoint de visualizar productos y dar execute. • Ir al endpoint de actualizar productos y dar execute. • Ir al endpoint de eliminar productos y dar execute.
<p>Resultado deseado:</p> <p>Como administrador necesita tener un CRUD para poder crear, visualizar, actualizar y eliminar un producto, junto con las validaciones establecidas.</p>
<p>Evaluación de la prueba:</p> <p>El administrador logra administrar los productos y por ello la prueba es validada junto con la aprobación del Product Owner.</p>

Tabla XXI Prueba de aceptación 6 – Administrar pedidos.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-006	Historia de Usuario: HU006
Nombre: Administrar Pedidos	
<p>Descripción: El usuario administrador en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar pedidos • Actualizar el pedido 	
<p>Pasos de ejecución:</p> <p>Para Administrar pedidos:</p>	

<ul style="list-style-type: none"> • Ingresar a la URL del componente Backend en un navegador. • Iniciar sesión como administrador. • Ir al endpoint de visualizar pedidos y dar execute. • Ir al endpoint de actualizar el pedido y dar execute
<p>Resultado deseado:</p> <p>Como administrador necesita tener endpoints que permitan visualizar todos los pedidos que se han realizado y poder actualizar los pedidos con descripciones para que los clientes pueda visualizarlo, junto con las validaciones establecidas</p>
<p>Evaluación de la prueba:</p> <p>El administrador logra administrar los pedidos y por ello la prueba es validada junto con la aprobación del Product Owner.</p>

Tabla XXII Prueba de aceptación 7 – Administrar Comentarios.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-007	Historia de Usuario: HU007
Nombre: Administrar Comentarios	
<p>Descripción: El usuario administrador y cliente en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar comentarios y/o sugerencias • Crear comentarios y/o sugerencias • Eliminar comentarios y/o sugerencias 	
<p>Pasos de ejecución:</p> <p>Para el registro del usuario cliente:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Iniciar sesión como administrador. • Ir al endpoint de visualizar comentarios generales o comentarios y dar execute. • Ir al endpoint de eliminar comentarios generales o comentarios y dar execute. • Iniciar sesión como usuario • Ir al endpoint de visualizar comentarios generales y dar execute • Ir al endpoint de crear comentarios generales, llenar los campos del formulario y dar execute 	

<ul style="list-style-type: none"> • Ir al endpoint de actualizar comentarios generales, llenar los campos del formulario y dar execute
<p>Resultado deseado:</p> <p>Como cliente necesita tener gestión sobre los comentarios que se realizan y ver si se tiene un aporte con un criterio aceptable, junto con las validaciones establecidas</p>
<p>Evaluación de la prueba:</p> <p>El administrador logra administrar los comentarios y por ello la prueba es validada junto con la aprobación del Product Owner.</p>

Tabla XXIII Prueba de aceptación 8 – Visualizar categorías.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-008	Historia de Usuario: HU008
Nombre: Visualizar categorías	
<p>Descripción: El usuario cliente y administrador en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar las categorías 	
<p>Pasos de ejecución:</p> <p>Para el registro del usuario cliente:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Ir al endpoint de login e iniciar sesión. • Ir al endpoint de Listar Categorías. • Dar clic en Execute. 	
<p>Resultado deseado:</p> <p>Como cliente y administrador necesitan visualizar las categorías creadas, junto con las validaciones establecidas</p>	
<p>Evaluación de la prueba:</p> <p>El cliente y administrador logran observar de manera correcta todas las categorías y por ello la prueba es validada junto con la aprobación del Product Owner.</p>	

Tabla XXIV Prueba de aceptación 9 – Visualiza productos.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-009	Historia de Usuario: HU009
Nombre: Visualizar productos	
Descripción: El usuario cliente y administrador en el Backend deben generar varios endpoints para: <ul style="list-style-type: none"> • Visualizar los productos 	
Pasos de ejecución: Para el registro del usuario cliente: <ul style="list-style-type: none"> • Ingresar a la URL del componente Backend en un navegador. • Ir al endpoint de login e iniciar sesión. • Ir al endpoint de Obtener productos. • Dar clic en Execute. 	
Resultado deseado: Como cliente y administrador necesitan poder visualizar cada uno de los productos creados, junto con las validaciones establecidas	
Evaluación de la prueba: El cliente y administrador logran ver los productos y por ello la prueba es validada junto con la aprobación del Product Owner.	

Tabla XXV Prueba de aceptación 10 – Administrar carrito de compras.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-001	Historia de Usuario: HU001
Nombre: Registrarse	
Descripción: El usuario cliente en el Backend deben generar varios endpoints para: <ul style="list-style-type: none"> • Visualizar los productos y cantidades seleccionadas • Visualizar el precio por productos y el valor total a pagar. 	
Pasos de ejecución: Para el registro del usuario cliente:	

<ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Iniciar sesión como usuario. • Ir al endpoint de crear pedido • Llenar todos los campos solicitados del formulario. • Dar clic en Execute.
<p>Resultado deseado:</p> <p>Como cliente necesita poder general una orden con los productos que seleccione, junto con las validaciones establecidas</p>
<p>Evaluación de la prueba:</p> <p>El cliente crea de manera eficiente una orden y por ello la prueba es validada junto con la aprobación del Product Owner.</p>

Tabla XXVI Prueba de aceptación 11 – Visualizar pedidos.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-011	Historia de Usuario: HU011
Nombre: Visualizar pedidos	
<p>Descripción: El usuario cliente en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Visualizar su pedido • Actualizar pedido 	
<p>Pasos de ejecución:</p> <p>Para el registro del usuario cliente:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Iniciar sesión como usuario. • Ir al endpoint de Buscar pedido • Llenar todos los campos solicitados del formulario. • Dar clic en Execute. • Ir al endpoint de actualizar orden • Llenar todos los campos solicitados del formulario • Dar clic en Execute 	

<p>Resultado deseado:</p> <p>Como cliente necesita visualizar sus pedidos y actualizarlos en caso de ser necesario, junto con las validaciones establecidas</p>
<p>Evaluación de la prueba:</p> <p>El cliente logra visualizar todos sus pedidos y actualizarlos y por ello la prueba es validada junto con la aprobación del Product Owner.</p>

Tabla XXVII Prueba de aceptación 12 - Calificar producto.

PRUEBA DE ACEPTACIÓN	
Identificador: PA-001	Historia de Usuario: HU001
Nombre: Registrarse	
<p>Descripción: El usuario cliente en el Backend deben generar varios endpoints para:</p> <ul style="list-style-type: none"> • Calificar el producto comprado • Actualizar su comentario 	
<p>Pasos de ejecución:</p> <p>Para el registro del usuario cliente:</p> <ul style="list-style-type: none"> • Ingresar a la URL del componente <i>Backend</i> en un navegador. • Iniciar sesión como usuario. • Ir al endpoint de crear comentario producto • Llenar todos los campos solicitados del formulario. • Dar clic en Execute. • Ir al endpoint de actualizar comentario producto • Llenar todos los campos solicitados del formulario • Dar clic en Execute 	
<p>Resultado deseado:</p> <p>Como cliente necesita poder actualizar y visualizar los comentarios de los productos que ha adquirido y que esté Finalizado la entrega, junto con las validaciones establecidas</p>	
<p>Evaluación de la prueba:</p> <p>El cliente logra crear y actualizar los comentarios de los productos y por ello la prueba es validada junto con la aprobación del Product Owner.</p>	

Certificado que confirma el cumplimiento de todos los requisitos y funcionalidades

SATURNINA

Quito, 26 de enero de 2024

CERTIFICADO

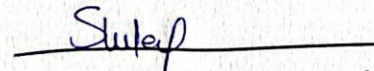
A quien interese

Yo, **Shirley Stefania Rivadeneira Armas** cédula de ciudadanía **1718299603** como dueña de "Saturnina", certifico:

Que el Sr, **David Sebastian Basantes Lucero** con cédula de ciudadanía **1750974550**, estudiante de la Carrera de Tecnología Superior en Desarrollo de Software, procedió a desarrollar e implementar un sistema web para el negocio mencionado anteriormente, mismo que cumple con todos los requerimientos y funcionalidades definidas en las reuniones mantenidas.

Es todo cuanto puedo certificar en honor a la verdad, el interesado puede hacer uso de este como bien requiera.

Atentamente,


Shirley Stefania Rivadeneira Armas

ANEXO III

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

https://www.youtube.com/watch?v=nBuVLAy6JE8&ab_channel=ESFOT_EPN

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 presenta las credenciales de acceso del backend, además del repositorio de GitHub, donde se encuentra todo el código y los pasos a seguir para su instalación en el apartado del README.

Credenciales de acceso:

Para acceder al backend en su versión de producción, se usa la URL:

<https://api-saturnina.1.us-1.fl0.io/api/v1/docs>

Credenciales del perfil administrador:

- Correo del administrador: sebastian2405lucero@hotmail.com
- Contraseña: @a1sdaw@qweDb

Credenciales para el perfil cliente:

- Correo de cliente: sebastian2405lucero@gmail.com
- Contraseña: @a1sdaw@qweDb

Repositorio del backend

El proyecto está alojado en un repositorio de GitHub, el cual se puede acceder a través de la siguiente URL:

<https://github.com/DAVIDS2405/Saturnina/tree/main>