ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN SISTEMA E-COMMERCE PARA EL MINIMARKET “MIKA Y VALE”

COMPONENTE BACKEND

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

MIGUEL ANGEL PAREDES REASCOS

miguel.paredes@epn.edu.ec

DIRECTOR: YADIRA GUISSELA FRANCO ROCHA

yadira.franco@epn.edu.ec

DMQ, AGOSTO 2024

CERTIFICACIONES

Yo, **Miguel Angel Paredes Reascos** 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.

MIGUEL ANGEL PAREDES REASCOS

miguel.paredes@epn.edu.ec

os.miguelparedes@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **Miguel Angel Paredes Reascos**, bajo mi supervisión.

Yadira Guissela Franco Rocha

DIRECTOR

yadira.franco@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.

MIGUEL ANGEL PAREDES REASCOS

DEDICATORIA

Quiero dedicar el desarrollo de este trabajo a todas las personas que me han acompañado a lo largo de este recorrido. En primer lugar, a mis padres, Roberto Paredes y Sandra Reascos, quienes me dieron la vida y han guiado cada paso que he dado. Agradezco profundamente a mis hermanas, Angélica Paredes y Carmen Paredes, por su apoyo incondicional en todo momento. También a mi cuñado, Daniel Cevallos, cuyas enseñanzas han sido clave para mi constante mejora.

Mis queridos sobrinos, Johan Cevallos y Aaron Cevallos, han llenado mi vida de momentos únicos y llenos de alegría. Y por supuesto, a mis abuelos, Mercedes Cruz, Fermín Reascos, Wilson Paredes, Susana Carrera, a mis tíos, Geovany Reascos, Helena Paredes, Liz Reascos, y mis primos, Cristiana Alarcón, Valeria Vaca, Sebastián Vaca, quienes siempre han estado atentos y apoyándome en mis estudios y en cada uno de mis proyectos.

Además, quiero dedicar un agradecimiento especial a mis padrinos Fernando Vaca y Sofia Paredes, con quienes siempre he podido contar y me han demostrado su apoyo incondicional en mis estudios. A mi tío abuelo Fausto Clavijo y a su esposa Doris, quienes me recibieron en su casa cuando necesitaba un cambio de aire. Esto fue fundamental para mí, ya que me animó y me permitió continuar con mis estudios con las energías renovadas.

A Karen Tipán, quien me enseñó la importancia de la responsabilidad, dedicación, organización y de nunca rendirme, a no conformarme con lo mínimo y siempre me acompañó en momentos buenos y en los malos siempre me dio palabras de ánimo y aliento, me enseño cuánto valgo y el potencial que tengo por explotar.

A los amigos y compañeros que hice a lo largo de la carrera, con quienes he compartido momentos de alegría y risas, sin ellos sé que no hubiese sido lo mismo el haber llegado a escribir esta dedicatoria.

Su constante aliento y apoyo han sido pilares fundamentales en mi camino, y por eso dedico este logro a cada uno de ustedes, quienes han dejado una marca imborrable en mi vida.

MIGUEL ANGEL PAREDES REASCOS

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por cada una de las personas que puso en mi camino, por ser quien me fue guiando, dando sabiduría y me ayudo a completar este logro.

Quiero agradecer a mis padres Roberto Paredes y Sandra Reascos quienes siempre me han guiado y cuando me sentía mal me dieron palabras de ánimo para levantarme al día siguiente, por todo el apoyo emocional incondicional. Ellos son un pilar fundamental para mi vida ya que sé que puedo contar con su apoyo en cualquier momento, puedo confiar en ellos, han sido mi motivación y fuente perseverancia.

También quisiera agradecer a mis hermanas Carmen Paredes y Angélica Paredes, así como a mi cuñado Daniel Cevallos, por todos los momentos de alegría que me han brindado y por todas las ocasiones en las que eligieron ayudarme antes de seguir con sus propias actividades.

Agradezco mis abuelos Mercedes Cruz, Fermín Reascos, Wilson Paredes, Susana Carrera, a mis tíos, Geovany Reascos, Helena Paredes, Liz Reascos, con quienes siempre he podido contar con sus oraciones y bendiciones para poder seguir adelante y nunca desviarme del camino.

Mis agradecimientos también van para mis padrinos, Fernando Vaca y Sofia Paredes por todo el apoyo que me han brindado a lo largo de mi vida estudiantil, gracias por facilitarme muchas cosas y permitirme seguir progresando hasta completar este logro.

Quiero reconocer a mi tío abuelo Fausto Clavijo y su esposa Doris, por abrirme las puertas de su casa cuando más lo necesitaba, aconsejarme cuando me encontraba perdido, las comidas, los viajes y anécdotas que tuvimos mientras estuve en su hogar.

Agradezco a Karen Tipán por cada uno de los momentos en los que me acompaño, por cada vez que me dio palabras de ánimo, por todo el apoyo que recibí de su parte en momentos complicados, todos los momentos de risas, gracias por demostrar muchas veces lo que yo significo para ti.

A mis amigos de la universidad Cristian Simba, Washington Villares, Melany Barrera, Jordan Flores, Dennis Cataña, Brandon Sandoval, Gilmar Morales, Alejandro Moreira, Jared Valenzuela, Madeline Pumacuro, por todos los momentos que pasamos juntos risas, jugar cartas, contar chistes, hablar incoherencias.

Tambien quisiera realizar un agradecimiento a los profesores que me dieron clases, se preocuparon en mi aprendizaje, se convirtieron en maestros en los que se que puedo acudir si necesito un consejo.

Finalmente a Elizan, Aileen, Wanner, Victoria y Spidey por todas las tardes/noches que me acompañaron a la distancia mientras realiazaba mis deberes, adelantaba mis proyectos. Me permitieron vivir momentos únicos e inigualables.

MIGUEL ANGEL PAREDES REASCOS

ÍNDICE DE CONTENIDO

RESUMEN

El minimarket "Mika y Vale" se enfrenta al desafío de destacarse en un mercado local competitivo mediante la implementación de un E-Commerce. Este sistema permitirá a los clientes navegar y seleccionar productos con facilidad, ya sea en línea o visitando físicamente el local. Con el objetivo de facilitar la experiencia de compra, el E-Commerce se centra en un Backend desarrollado con una API REST. Esta infraestructura tecnológica utiliza Node.js como entorno de ejecución y MongoDB Atlas como base de datos, asegurando la gestión eficiente de categorías, productos y pedidos.

La metodología ágil SCRUM guia el desarrollo del proyecto, permitiendo entregas iterativas y rápidas adaptaciones a medida que evoluciona el mercado y las necesidades del negocio. La seguridad de la información es una prioridad, asegurando transacciones seguras y protección de datos para generar confianza en los clientes.

Además de la plataforma en línea, "Mika y Vale" mantiene su infraestructura física para clientes que prefieren una experiencia tradicional de compra. Los usuarios pueden interactuar sin interrupciones, con la opción de recibir soporte a través de WhatsApp y correo electrónico.

El objetivo final es mejorar la accesibilidad y comodidad para los clientes actuales, al mismo tiempo que se facilita la expansión del minimarket hacia nuevos mercados, sin requerir grandes inversiones en infraestructura física. Esto posiciona a "Mika y Vale" como un competidor sólido en el sector minorista local, adaptándose ágilmente a las demandas del mercado digital actual.

PALABRAS CLAVE: E-Commerce, Backend, API REST, Node.js, MongoDB Atlas, Scrum, Seguridad de la información.

ABSTRACT

The minimarket "Mika y Vale" is facing the challenge of standing out in a competitive local market by implementing an E-Commerce system. This system will allow customers to browse and select products easily, either online or by visiting the physical store. Aiming to enhance the shopping experience, the E-Commerce system focuses on a Backend developed with a REST API. This technological infrastructure uses Node.js as the runtime environment and MongoDB Atlas as the database, ensuring efficient management of categories, products, and orders.

The SCRUM agile methodology guides the project's development, enabling iterative deliveries and quick adaptations as the market and business needs evolve. Information security is a priority, ensuring secure transactions and data protection to build customer trust.

In addition to the online platform, "Mika y Vale" maintains its physical infrastructure for customers who prefer a traditional shopping experience. Users can interact seamlessly, with the option to receive support via WhatsApp and email.

The ultimate goal is to improve accessibility and convenience for current customers while facilitating the minimarket's expansion into new markets without requiring significant investments in physical infrastructure. This positions "Mika y Vale" as a strong competitor in the local retail sector, adapting agilely to the demands of the current digital market.

KEYWORDS: E-Commerce, Backend, REST API, Node.js, MongoDB Atlas, Scrum, Information Security.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El minimarket “Mika y Vale” se enfrenta al problema de destacarse en un mercado local competitivo y diversificado. Debido a esto se vio en la necesidad de realizar un E-Commerce, para enseñar sus productos disponibles a sus clientes, facilitando el poder escoger los productos que necesitan y realizar sus pedidos [1]. A través de este modelo de negocio permitirá que el minimarket pueda ampliarse llegando a otros mercados, puesto que no tiene la necesidad de tener una gran infraestructura física y de mantenimiento de esta [2].

Para aquellos clientes que prefieren lo tradicional o no se encuentran muy familiarizados con las compras en línea, “Mika y Vale” cuenta con una infraestructura física, que permite que estos clientes puedan visitar el local y realizar sus compras, el administrador puede crear una o varias cuentas de cajeros, mismas que servirá para registrar los pedidos de los clientes que acudieron al local, para los clientes que prefieren realizar sus pedidos en línea tendrán la opción de poder ir a retirar sus productos o recibirlos en la puerta de su hogar por una pequeña comisión. Al poder realizar sus compras en línea permite a los usuarios el poder realizar sus compras sin interrupciones y si tienen alguna duda puede hablar con el equipo de asistencia por medio de la aplicación de mensajería WhatsApp o enviando un correo [4].

El enfoque del proyecto se centra en desarrollar un Backend que permite la creación del E-Commerce a través de una API REST. Implementando funcionalidades clave como la gestión de las categorías, gestión de los productos, el procesamiento de pedidos y ventas, la administración de usuarios. También se encarga de garantizar y proteger la información de los clientes. Permite al usuario, administrador y cajero visualizar el pedido realizado, además se envía un correo que confirma que se ha realizado el pedido con éxito. También permite a los cajeros realizar ventas a dentro del local con la finalidad de ofrecer un producto que se encuentran en stock. Esto contribuirá a mantener la lealtad de los clientes y a promover la confianza en la plataforma.

Además, el Backend proporciona una base sólida para futuras mejoras y expansiones del sistema de E-Commerce. A medida que "Mika y Vale" crezca y evolucione, el Backend se adapta y escala para satisfacer las necesidades cambiantes del negocio.

1.1 Objetivo general

Desarrollar el Backend del sistema E-Commerce para minimarket “Mika y Vale”.

1.2 Objetivos específicos

1. OE1 Levantar los requerimientos para Backend.
2. OE2 Diseñar la arquitectura y el modelo de datos en función de los requerimientos.
3. OE3 Desarrollar los endpoints necesarios para la API REST.
4. OE4 Realizar pruebas.
5. OE5 Desplegar el Backend.

1.3 Alcance

El sistema permite al minimarket “Mika y Vale” tener tres perfiles los mismos que son el del administrador, cajero y del cliente. El administrador puede gestionar la creación de categorías de productos, creación de productos y la autentificación de este, con la finalidad de que solo el administrador pueda crear o modificar las categorías y productos. El cajero puede realizar los pedidos de los clientes que decidan ir al local a realizar sus compras.

El cliente puede registrarse desde la web, para poder acceder a su cuenta debe de confirmar su cuenta por medio de un correo, en caso de olvidar su contraseña el cliente tiene la opción de poder reestablecer su contraseña, una vez iniciada su sesión puede ir agregando a su carro de compras los productos que necesite, además puede colocar en favoritos uno o varios productos, el mismo tiene el registro o la administración tras la cual realizar el pedido puede tener una revisión en su envío, hora de llegada e información detallada de sus productos.

Se usan herramientas de desarrollo que permiten la gestión de las versiones del proyecto, agilización de la creación del código de este, proporcionando las funcionalidades que se plantearon en los requerimientos. Se usa la metodología SCRUM como guía en el proyecto, permitiendo desplegar el sistema web a producción.

El proyecto cuenta con tres perfiles, cada uno tiene permisos para realizar ciertas acciones en los siguientes módulos:

Perfil: **Administrador**

El administrador tiene acceso a los siguientes endpoints:

- Endpoints para inicio y cierre de sesión.
- Endpoints para gestionar las categorías de productos.
- Endpoints para gestionar los productos.
- Endpoints para gestionar perfiles de cajeros.
- Endpoints para reportes: visualización, búsqueda de información, filtrado y estado de Pedidos.
- Endpoints para cambiar el estado de un pedido.

Perfil: **Cajero**

- Endpoints para inicio y cierre de sesión.
- Endpoints para visualizar y buscar categorías de productos.
- Endpoints para visualizar y buscar productos.
- Endpoints para agregar productos al carrito de ventas.
- Endpoints para visualizar el carrito de ventas.
- Endpoints para realizar ventas.
- Endpoints para reportes: visualización, búsqueda de información, filtrado y estado de pedidos.
- Endpoints para cambiar el estado de un pedido.

Perfil: **Cliente**

- Endpoints para inicio y cierre de sesión, confirmación de cuenta y reestablecer la contraseña.
- Endpoints para visualizar y buscar categorías de productos.
- Endpoints para visualizar y buscar productos.
- Endpoints para agregar productos al carrito de compras.
- Endpoints para visualizar el carrito de compras.

- Endpoints para realizar el pedido, una confirmación de la elaboración del pedido por correo y visualización del pedido.
- Endpoints para visualizar y buscar los pedidos realizados por el propio cliente.
- Endpoints para colocar productos en su apartado de favoritos.
- Endpoinst para realizar búsquedas en sus productos favoritos.

1.4 Marco Teórico

Desarrollo de software

El desarrollo de software son todos los procesos involucrados a la hora de crear un programa de software, lo que lo convierte en una herramienta fundamental en la mayoría de las empresas, puesto que la gran mayoría ya dependen de la tecnología para poder funcionar. Los desarrolladores son los encargados de elaborar soluciones digitales según las necesidades que tenga la cada empresa [4].

Desarrollo web

En el caso de este proyecto se enfoca en el desarrollo web. El desarrollo web consta en elaborar un sitio web, mismo que puede ser elaborado a través de diversos lenguajes de programación, con la finalidad de que sea funcional en internet, cada sitio web tiene una URL única, permitiendo de esta manera ser distinguido de otros sitios web. Un sitio web consta de dos partes el Backend y el Frontend. El Backend es el contacto directo con el servidor, se encuentra en segundo plano realizando varias funciones, se encarga de toda la lógica para que funcione un sitio web. El Frontend es la parte visible e interactiva para el usuario, y está asociado con la experiencia y la interfaz del usuario [5].

E-Commerce

El comercio electrónico se refiere a la compra y venta de bienes y servicios a través de Internet. Mismos que se han vuelto cada vez más populares en los últimos años debido a su conveniencia y accesibilidad. Los sistemas de E-Commerce permiten la realización de transacciones comerciales en línea, brindando a los clientes la posibilidad de comprar productos desde cualquier ubicación y en cualquier momento [2].

API

La Interfaz de Programación de Aplicaciones o más conocida por sus siglas en ingles API es un conjunto de reglas que definen como diferentes programas o aplicaciones de ordenador se conecten y comuniquen entre sí. Una API REST es una API, pero se ajusta a la Transferencia de Estado Representacional o por sus siglas REST, lo que la hace más flexible y ligera de integrar aplicaciones. Facilita la transferencia bidimensional de datos en formatos JSON o XML. Además, por medio de sus métodos y que son verbos HTTP podemos saber que acción estamos realizando Crear, Leer, Actualizar y Borrar o por sus siglas en ingles CRUD [9].

JavaScript

Para la implementación del Backend del E-Commerce, usaremos el lenguaje de programación JavaScript. JavaScript es un lenguaje de programación que permite crear elementos que mejoran la interacción de los usuarios como pueden ser formularios rellenables, gráficos animados, menús desplegables, etc. JavaScript cuenta con multitud de frameworks y librerías que permiten simplificar y agilizar proyectos complejos [6]. Usaremos el entorno de ejecución Node.js, este entorno de ejecución gratuito nos permite ejecutar código JavaScript fuera del navegador, es de código abierto lo que permite que desarrolladores puedan crear servidores, aplicaciones web, etc. [7]. Node.js también cuenta con frameworks para la elaboración de este proyecto se usa Express.

Express

Express es un framework muy popular para la elaboración de API REST o servidores, permitiendo elaborar una o muchas páginas, proporciona herramientas, peticiones y respuestas HTTP, enrutamiento y middleware, que permiten desarrollar y desplegar los servidores. Cuenta con una herramienta de interfaz de línea de comandos Node Package Manager (NPM), que permite obtener paquetes de desarrollo para los proyectos [8].

MongoDB

Para el almacenamiento de la información se utiliza una base de datos, una base de datos es un conjunto información organizada que se recopila y almacenan. Pueden almacenar números, letras, booleanos, palabras, imágenes, etc. [11]. Existen muchas bases de datos que permiten guardar y visualizar la información la nube, se utilizará MongoDB Atlas como base de datos. MongoDB Atlas es una plataforma de datos multicloud, que usa un modelo de datos de documentos tipo JSON, de esta manera se pueden realizar consultas, transferir datos, actualizar y analizar datos en tiempo real [10].

Pruebas de Backend

Son pruebas del lado del servidor que permiten verificar el correcto funcionamiento de la base de datos, de esta manera se comprueba que la aplicación web o el programa se encuentran sin desperfectos, como corrupción o perdida de datos. Este tipo de pruebas se las debe de realizar correctamente, ya que una mala implementación puede ocasionar muchas complicaciones entre las cuales están: corrupción de datos, perdidas de datos, etc. [35].

2 METODOLOGÍA

Una metodología son técnicas y/o métodos de carácter científico que se aplican durante un proceso de investigación para encontrar una solución, rige la forma en la que vamos a aplicar procedimientos para la investigación [12]. Para el desarrollo del sistema utilizara el método de estudio de casos que consiste en aplicar conocimientos para la resolución o búsqueda de soluciones de problemas, se emplean conocimientos previos para encontrar dicha solución [13]. La solución planteada es el desarrollo del Backend destinado a gestionar las categorías de productos, los productos y los pedidos en línea. Permitiendo a los usuarios realizar sus pedidos de una manera más sencilla sin interrupciones, así se genera el aumento de las ventas de “Mika y Vale” por medio de herramientas tecnológicas.

2.1 Metodología de Desarrollo

Es el conjunto de técnicas y métodos que favorecen en el desarrollo de una solución de software informático, ayudan a minimizar y a anticiparse a los errores, permiten ahorrar tiempo y gestionar de mejor manera los recursos disponibles [14].

Uno de los objetivos que la mayoría de las metodologías tienen es el promover la organización de equipos de trabajo para poder desarrollar de una manera más eficiente y efectiva las funciones de un programa. Por ello es importante elegir una metodología que facilite el desarrollo de un producto, ya que sin una metrología clara se producirán inconvenientes que dificultan su elaboración y en el peor de los casos ocasiona que se tenga un mal producto. Existen dos tipos de metodologías de desarrollo de software las tradicionales y las agiles [15].

La metodología tradicional tiene la característica de definir estrictamente cuales son los requisitos del proyecto, los ciclos de desarrollo no son flexibles con la finalidad de que no se puedan realizar cambios; además la tradicional es lineal lo que significa que sus etapas deben de ser una tras otra, no se puede iniciar la siguiente etapa si no se ha culminado la

anterior y tampoco se puede volver a una etapa anterior. La metodología ágil permite que, en el transcurso del desarrollo del proyecto, tenga la facilidad de irse adaptando a las necesidades que van surgiendo en el camino, por este motivo es una de las más utilizadas hoy en día [15].

Una vez explicadas las metodologías, se ha decidido para el desarrollo del presente proyecto el uso de la metodología ágil SCRUM, la cual permite una mejor colaboración, flexibilidad y entrega rápida de productos gracias a su marco ágil [16]. En esta metodología usa Sprints, mismos que se deben de resolver en un tiempo determinado que no debe de ser mayor a un mes, el tiempo que debe de durar cada Sprint debe de ser de una a cuatro semanas. El equipo SCRUM está conformado por Product Owner, Scrum Master y Equipo de Desarrollo [17].

Roles

Los roles dentro de un equipo de trabajo permiten garantizar el éxito de un proyecto, ya que cada rol cuenta con sus debidas responsabilidades, por lo que facilita la toma de decisiones, de esta manera cada miembro del equipo de trabajo tiene un papel crucial, para designar los papeles que cumplen cada integrante del grupo vamos a indicar los roles que son usados en este proyecto y los detallamos a continuación:

Product Owner

Es la persona responsable de identificar cuáles son las funcionalidades prioritarias para el desarrollo durante de los Sprints [17], se encarga de que aumentar el valor y de minimizar el riesgo de aceptación al momento de lanzar el producto, hace que la voz del cliente sea escuchada por el equipo de desarrollo [18].

Scrum Master

Es la persona responsable de asegurar que el equipo se mantenga enfocado en el desarrollo del proyecto, a la vez que elimina impedimentos que puedan afectar al equipo [17], facilita la comunicación entre el equipo de desarrollo, permitiendo de esta manera asegurar que el producto final sea exitoso [19].

Development Team

Suele estar formado entre 3 y 9 profesionales, mismos que están encargados del desarrollo del producto autoorganizándose y autogestionándose con la finalidad de entregar una versión funcional y utilizable del producto al final de cada Sprint, todos los integrantes comparten un mismo rol, dentro de la propia organización pueden asignarse roles

independientes, pero esto ya es responsabilidad y organización de cada equipo de trabajo [20]. En la **Tabla 2.1** se puede ver la designación de los roles.

Tabla 2.1: Designación de roles de equipo de desarrollo

ROL	INTEGRANTE
Product Owner	Sandra Jeanneth Jacho Chicaiza
Scrum Master	Ing. Yadira Guissela Franco Rocha
Development Team	Miguel Angel Paredes Reascos

Artefactos

Son todos los medios que garantizan la transparencia, registro de información, productividad y calidad de los proyectos que utilicen metodología SCRUM [21], cada artefacto permite que el equipo planifique y priorice las actividades que se deben de seguir para los siguientes Sprints, ya que permiten ver cómo va el progreso del proyecto [22].

Recopilación de Requerimientos

Un requerimiento es un acuerdo o especificaciones que se ponen en un proyecto, servicio o resultado, con la finalidad de satisfacer las necesidades que pueda tener el cliente, en otras palabras, representan el resultado de los acuerdos, guiando el su desarrollo e implementación [23]. En la **Tabla 2.1** se observa el formato utilizado en la recopilación de requerimientos del presente proyecto. En el **ANEXO II** se encuentra la **Tabla 1** donde se visualiza la lista completa.

Tabla 2.2: Recopilación de Requerimientos

RECOPILACION DE REQUERIMIENTOS		
TIPO DE SISTEMA	**ID – RR**	**ENUNCIADO DEL ITEM**
BACKEND	**RR-001**	Como usuario **cliente** se necesita generar varios endpoints para: • Registrarse • Confirmar la cuenta • Reestablecer contraseña

Historias de Usuario

Son pequeñas descripciones de los requerimientos que tiene el cliente. En estas historias se las debe de realizar describiendo el rol, su funcionalidad específica y lo que se espera como resultado. También se las puede ver como formalidades relacionadas con las soluciones a los problemas de la organización [24]. El formato utilizado para las historias de usuario se observa en la **Tabla 2.3**. En el **ANEXO II** a partir de la **Tabla 2** hasta la **Tabla 7** se pueden observar todas las historias de usuario.

Tabla 2.3: Historia de Usuario Formato

<table>
<tr><td colspan="2">HISTORIA DE USUARIO</td></tr>
<tr><td>Identificador: RR-001</td><td>Usuarios: Cliente</td></tr>
<tr><td colspan="2">Nombre Historia: Cuenta del cliente</td></tr>
<tr><td>Prioridad en el negocio: Alta</td><td>Riesgo en el desarrollo: Medio</td></tr>
<tr><td colspan="2">Iteración asignada: 2</td></tr>
<tr><td colspan="2">Responsable: Miguel Paredes</td></tr>
<tr><td colspan="2">Descripción: Como usuario cliente, quiero poder registrarme en el sistema para poder realizar pedidos, restablecer mi contraseña en caso de olvidarla.
Registrarse.
Confirmar la cuenta.
Reestablecer contraseña.</td></tr>
<tr><td colspan="2">CA:
Registro de Cliente:
Endpoint para permitir a un usuario registrarse en la plataforma.
Captura de información básica del cliente: nombre, correo electrónico, contraseña y teléfono.
Validación de datos y creación de la cuenta en el sistema.
Envío de un correo de confirmación de registro.
Se creará una nueva cuenta si el correo y/o teléfono no se encuentren previamente registrados.
Verificar que el usuario haya confirmado su cuenta.
Confirmar la cuenta:
Endpoint que permita verificar que la cuenta ha sido confirmada.</td></tr>
</table>

○ Se envía un correo con el que se verifica la identidad del cliente. • **Restablecer Contraseña:** ○ Endpoint para permitir a un usuario cliente restablecer su contraseña olvidada. ○ Envío de un enlace de restablecimiento al correo electrónico registrado. ○ Validación de la identidad del usuario mediante el enlace de restablecimiento. ○ Confirmación de que la contraseña ha sido actualizada exitosamente.
Observación: Únicamente el cliente es el que puede registrarse y reestablecer su contraseña en caso de que se le olvide.

Product Backlog

Es la lista de funciones ordenada por las prioridades necesarias para cumplir los objetivos del proyecto, aquí se detallan las tareas que se deben de ir realizando, todo el equipo de desarrollo puede verlas, de esta manera se puede tener una vista panorámica de lo que se tiene que hacer [25]. El formato utilizado para el Product Backlog se puede observar en la **Tabla 2.4**. En el **ANEXO II** se encuentra la **Tabla 8** donde se puede observar la lista completa.

Tabla 2.4: Product Backlog

ELABORACIÓN DEL PRODUCT BACKLOG				
ID-HU	**HISTORIA DE USUARIO**	**ITERACIÓN**	**ESTADO**	**PRIORIDAD**
HU-001	Cuenta del cliente	2	Finalizado	Media

Sprint Backlog

Son todas las tareas que se deben de finalizar en cada Sprint, estas tareas se obtienen del Product Backlog en cada sesión de proyección de desarrollo de cada Sprint, de esa manera se controla de mejor manera el alcance que tiene porque se indica al equipo de desarrollo que es lo que hará y que no [26]. El formato utilizado para el Sprint Backlog se encuentra en la **Tabla 2.5**. En el **ANEXO II** se encuentra la **Tabla 9** donde se puede observar la lista completa.

Tabla 2.5: Sprint Backlog

ELABORACIÓN DE SPRINT BACKLOG						
ID-SB	**NOMBRE**	**MODULO**	**ID-HU**	**HISTORIA DE USUARIO**	**TAREA**	**TIEMPO ESTIMADO**
SB-000	Configuraciones para el desarrollo del proyecto.	N/A	N/A	N/A	• Diseño de base de datos • Recolección de requerimientos funcionales y no funcionales • Diseño de la arquitectura del módulo Backend • Estructura del proyecto • Instalación de dependencias necesarias	20H

2.2 Diseño de la arquitectura

Es una estructura y diseño de alto nivel para definir la interacción entre sí de los diferentes componentes, su organización y la manera en la que completan los requisitos del sistema, permite la toma de decisiones en el desarrollo ya que da una vista global del proyecto [27].

Patrón arquitectónico

Son soluciones popularizadas por lo que permiten ser reutilizadas para dar soluciones a los diseños de arquitectura, se enfocan en la distribución de responsabilidades, la comunicación entre sí de los componentes, gestión de la información y estructura del sistema [28]. Para el desarrollo de este proyecto se usa el patrón arquitectónico Modelo Vista Controlador (MVC), este modelo se divide en tres partes: el modelo es el encargado de los datos y la lógica, la vista de la representación de la interfaz de usuario y el controlador de las interacciones entre los dos anteriores (modelo y vista) [28]. En la **Ilustración 2.1** se visualiza el patrón de arquitectura MVC y las herramientas utilizadas en el Backend.

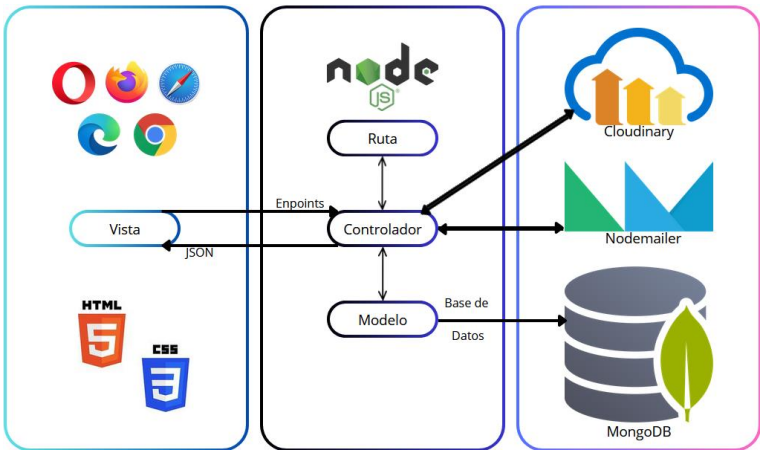


Ilustración 2.1: Patrón Arquitectónico del Backend

2.3 Herramientas de desarrollo

Son herramientas que permiten agilizar el proceso de desarrollo de proyectos, ya que reducen el estrés y reducen los tiempos de elaboración en cada fase, de esta manera se puede presentar mejores propuestas al cliente y por ende un producto final [29]. Algunas de las herramientas para el desarrollo del presente proyecto se encuentran en **Tabla 2.6**, fueron seleccionadas debido a mi amplia experiencia con ellas. Además, algunas de las bibliotecas utilizadas se mencionan en la **Tabla 2.7**. En **ANEXO II** se observan todas herramientas, las librerías utilizadas en la **Tabla 10** y **Tabla 11** respectivamente.

Tabla 2.6: Herramientas para el desarrollo del Backend.

Herramienta	Justificación
Visual Studio Code	Editor de código disponible para Windows, Linux y macOS, cuenta con Git y extensiones que facilitan el uso y ejecución de lenguajes de programación [30].
Node.js	Entorno que permite ejecutar código JavaScript fuera del navegador, permite que desarrolladores puedan aplicaciones web e integrar API REST. [7]

Tabla 2.7: Librerías

Nombre	Versión	Descripción
Bcrypt	5.1.1	Librería que usa un algoritmo lento para el cifrado seguro y verificación de contraseñas [35].
Cloudinary	2.0.3	Api multimedia que permite gestionar, transformar imágenes o videos para sitios web o aplicaciones móviles [36].

3 RESULTADOS

Se observa el progreso realizado de la elaboración de las funcionalidades del Backend. Se detallan de manera clara los resultados de cada uno de los Sprints, indicando el proceso a seguir para su desarrollo. En **ANEXO II** se encuentran las **Evidencias de los resultados** que evidencias el desarrollo de cada uno de los Sprints.

3.1 Sprint 0. Configuración del ambiente de desarrollo

La planificación para la elaboración de este Sprint consiste en las preparaciones para el desarrollo del proyecto, diseño de la base de datos, recolección de requerimientos, diseño de la arquitectura y la instalación de dependencias que ayudan a agilizar la elaboración del proyecto. Por lo tanto, las tareas a cumplir en este Sprint son las siguientes:

- Recolección de requerimientos funcionales y no funcionales.
- Diseño de base de datos.
- Diseño de la arquitectura del módulo Backend.
- Estructura del proyecto.
- Instalación de dependencias necesarias.

Recolección de requerimientos

Para la elaboración de un proyecto es necesario saber cuáles son las necesidades que tiene el cliente, de esta manera se puede presentar un producto que cumpla con todas las funcionalidades en base a las necesidades indicadas para solventar los problemas del cliente, esto nos facilita la elaboración de las funcionalidades obtenidas de las historias de

usuarios, así podremos crear los endpoints necesarios para realizar el despliegue y que el Frontend los pueda consumir.

Diseño de base de datos

Para el manejo de la información, se hace uso de una base de datos. Como ya se ha indicado, se emplea MongoDB Atlas como base de datos en la nube. Este modelo nos permite hacer adaptaciones a medida que se vaya desarrollando el proyecto, lo que nos permitirá tener una mejor escalabilidad y solventar nuevas necesidades que pueda tener el cliente con el pasar del tiempo. En la **Ilustración 3.1** se puede observar el modelo lógico del diseño que se utiliza en el desarrollo. En **Tabla 12** se encuentra el modelo físico del diseño de la base de datos.

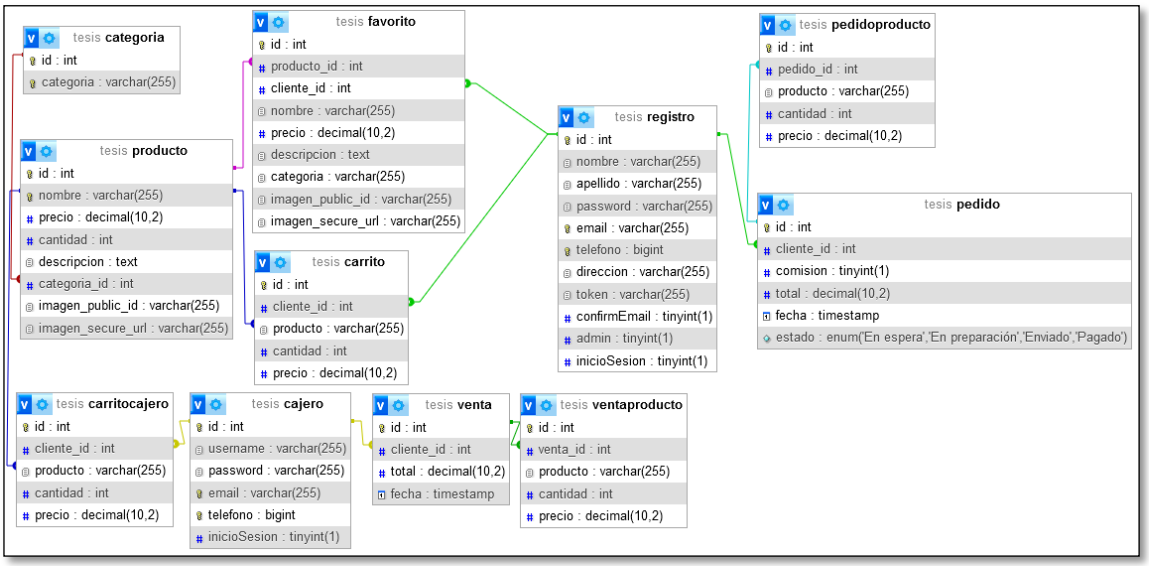


Ilustración 3.1: Modelo Lógico

Diseño de la arquitectura del módulo Backend

Para el diseño se utilizará uso de la arquitectura MVC que se puede visualizar en **Ilustración 2.1**, esta arquitectura nos permite dividir el código en componentes separados para facilitar su administración y en caso ser necesarios modificaciones en los mismos sin que estos cambios los afecten entre sí, como ya se mencionó anteriormente se divide en tres partes:

- Modelo (M): Gestiona la información que será utilizada juntamente con los métodos para poder ser almacenada y/o extraída de la base de datos.
- Vista (V): Elaboración de enpoints para el uso y consumo del Frontend.
- Controlador (C): Permite las interacciones del usuario con la información.

Estructura del proyecto

Al aplicar el patrón arquitectónico mencionado previamente, se observa en **Ilustración 3.1** la estructura implementada para la elaboración del presente proyecto. Se utilizan diversas carpetas que permiten organizar de mejor manera los archivos que se utilizan para la elaboración de las funcionalidades. Dentro de estas carpetas se tiene una carpeta que contiene las librerías para el funcionamiento del programa y otra carpeta que tiene todos los archivos y subcarpetas para el desarrollo del proyecto. Además, en esta última carpeta se hallan los archivos relacionados con la conexión a la base de datos, gestión del servidor y el levantamiento de este. Las subcarpetas contienen los archivos necesarios para ejecutar las diferentes funcionalidades del proyecto.

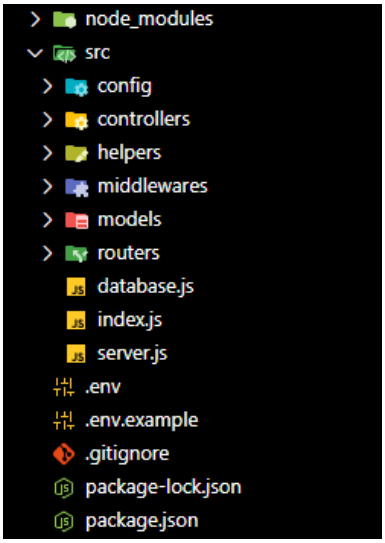


Ilustración 3.2: Estructura del Proyecto

Instalación de dependencias necesarias

Para comenzar con el desarrollo del proyecto, es necesario buscar las dependencias necesarias que permiten generar todas las funcionalidades del proyecto, en la **Tabla 3.1** se observan algunas dependencias utilizadas. En el **ANEXO II** se encuentra **Librerías** con todas las dependencias y versiones de estas utilizadas. Una vez instaladas y realizadas todas las configuraciones necesarias se puede continuar con la elaboración de los siguientes Sprints.

Tabla 3.1: Dependencias

Nombre	Versión	Descripción
Bcrypt	5.1.1	Librería que usa un algoritmo lento para el cifrado seguro y verificación de contraseñas [35].
Cloudinary	2.0.3	API multimedia que permite gestionar, transformar imágenes o videos para sitios web o aplicaciones móviles [36].

3.2 Sprint 1. Diseño e implementación de funcionalidades para el personal

La planificación de las tareas para la elaboración de este Sprint son las siguientes:

- Endpoints para CRUD perfiles de cajeros.
- Endpoints para CRUD categorías.
- Endpoints para CRUD productos.

CRUD perfiles de cajeros

En el desarrollo de esta tarea se crean varios endpoints que nos permiten crear el perfil de los cajeros, actualizar su información, eliminar, mostrar y buscar. Para el desarrollo de la creación de la preparación del perfil se utilizó el método POST, que permite enviar información para posteriormente almacenarla en la Base de Datos. La **Ilustración 3.3** permite observar cuales son los datos enviados para la creacion del perfil y el resultado final del endpoint.

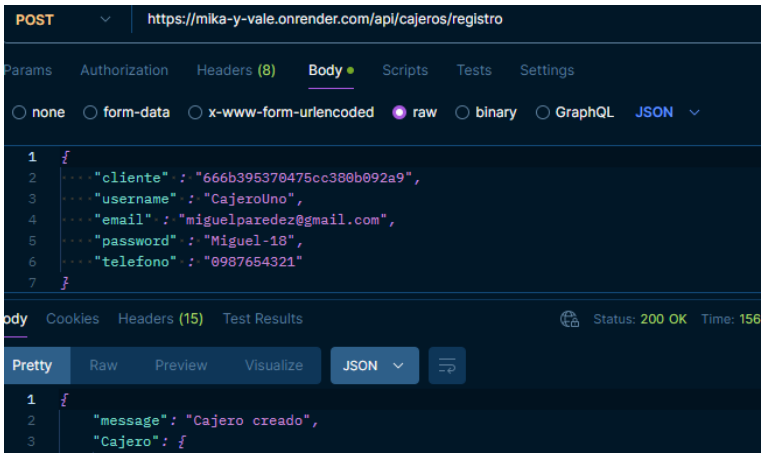


Ilustración 3.3: Endpoint para crear perfil cajero

Para el endpoint para la actualización de un perfil de cajero se utiliza un método PUT, que permite enviar la información con la finalidad de realizar reajustes en la base de datos. La **Ilustración 3.4** permite observar cuales son los datos que serán modificados del perfil de cajero y el resultado final del endpoint.

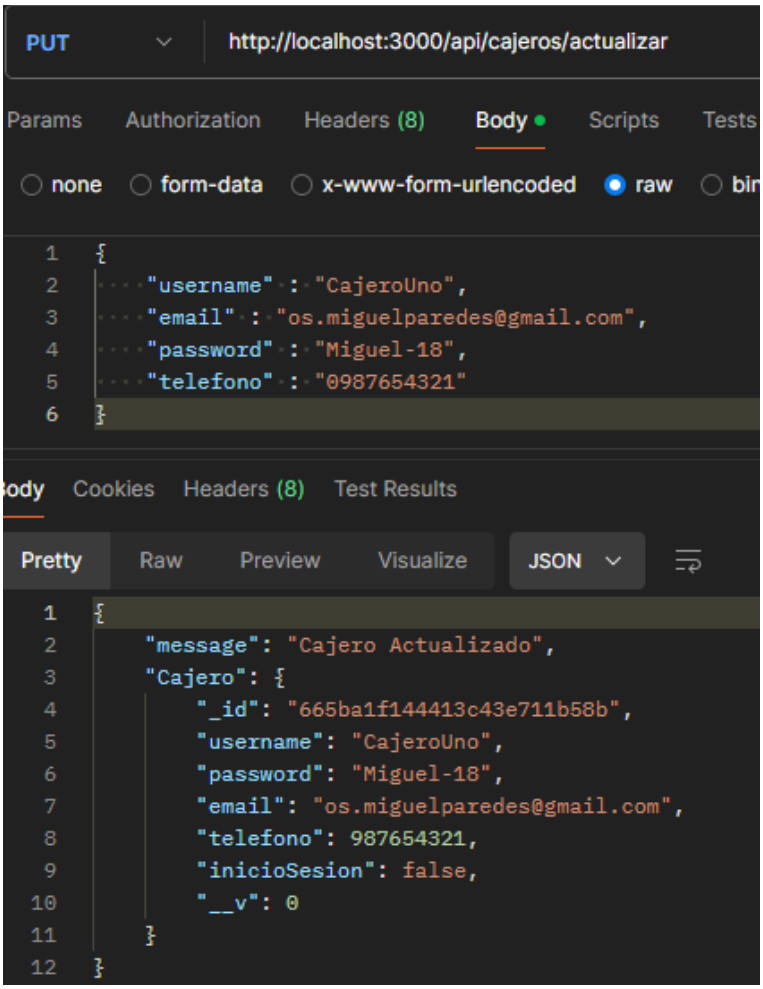


Ilustración 3.4: Endpoint para actualizar datos de un perfil de un cajero

Para el endpoint de eliminar un perfil de cajero se utiliza el método DELETE, que permite eliminar la información de la base de datos. La **Ilustración 3.5** permite observar el dato que es necesario para eliminar y el resultado final del endpoint.

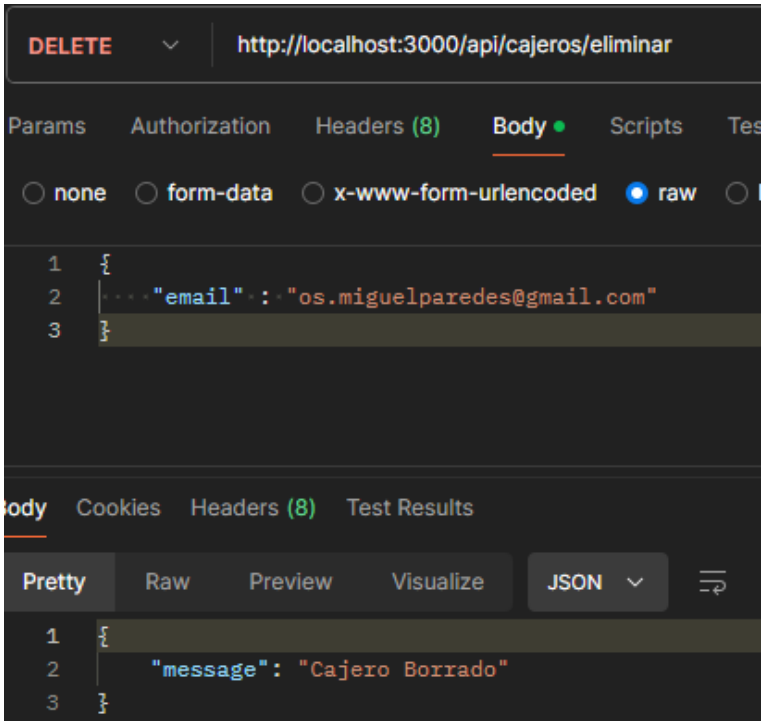


Ilustración 3.5: Endpoint para eliminar el perfil de un cajero

Para el endpoint de mostrar se utiliza el método GET, que permite observar la información. La **Ilustración 3.6** permite observar el resultado del endpoint.

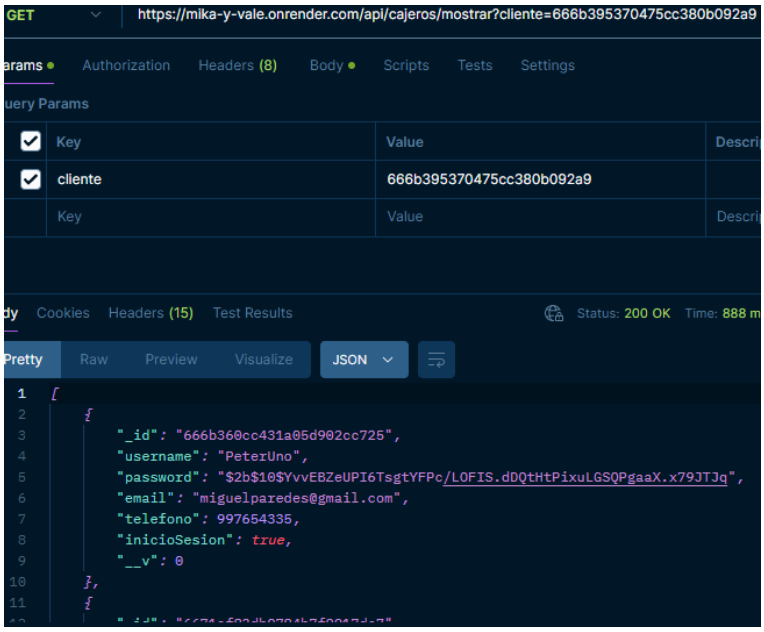


Ilustración 3.6: Endpoint para ver todos los perfiles de cajero

Para el endpoint de buscar se utiliza el método GET, que permite observar la información. La **Ilustración 3.7** permite observar el resultado del endpoint.

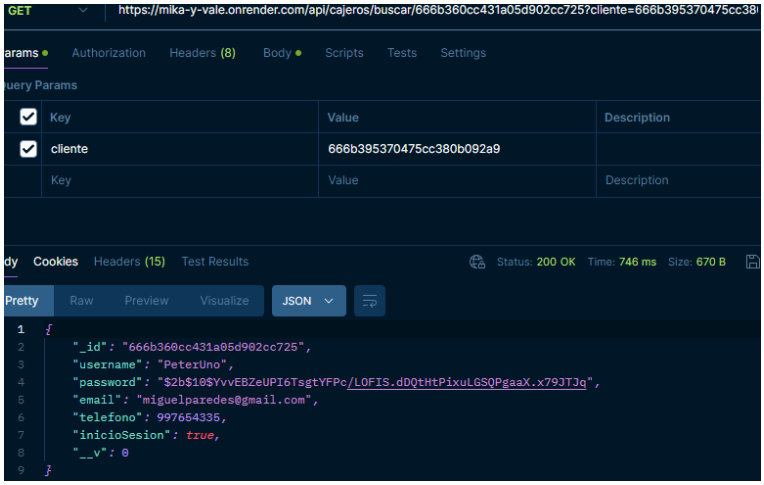


Ilustración 3.7: Endpoint para buscar el perfil de un cajero

CRUD categorías

En el desarrollo de esta tarea, se crean varios endpoints que permiten gestionar categorías de productos. La **Ilustración 1** muestra los datos enviados a través del método POST para la creación de una categoría. Además, la **Ilustración 2** e **Ilustración 3** proporcionan información sobre los métodos, los campos necesarios y el resultado de actualizar y eliminar categorías respectivamente.

CRUD productos

En la elaboración de esta tarea, se crean varios endpoints que permiten gestionar los productos. La **Ilustración 4** muestra los datos enviados a través del método POST para la creación de un producto. Además, la **Ilustración 5** e **Ilustración 6** proporcionan información sobre los métodos, los campos necesarios y el resultado de actualizar y eliminar producto respectivamente.

3.3 Sprint 2. Diseño e implementación de funcionalidades gestión de cuentas

La planificación de las tareas para la elaboración de este Sprint son las siguientes:

- Endpoints para registro de usuarios.
- Endpoints para inicio y cierre de sesión.
- Endpoints para gestionar los productos favoritos.

Registro de Usuarios

En el desarrollo de esta tarea se crean endpoints que permiten registrarse, confirmar la cuenta y reestablecer la contraseña. Para la creación de la cuenta se utiliza el método POST el cual permite enviar y validar la información del usuario. La **Ilustración 3.8** indica la información necesaria para crearse una cuenta e indica un mensaje de que revise su correo para poder confirmar la cuenta.

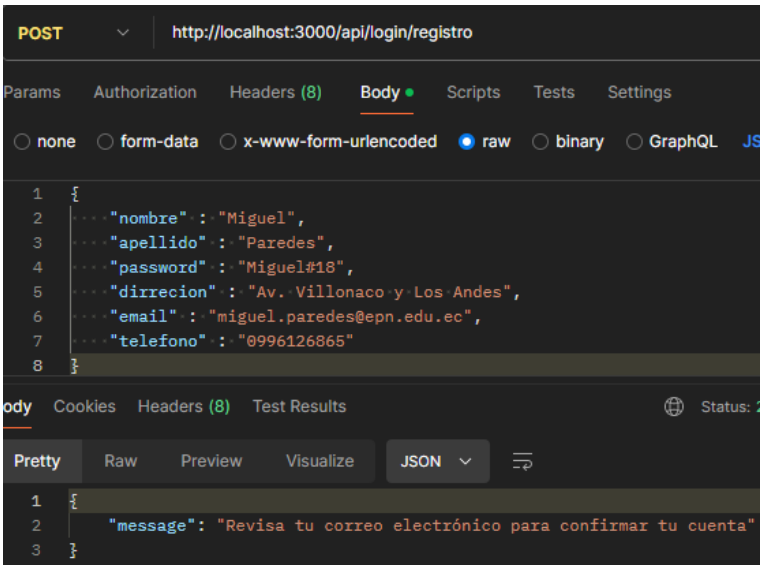


Ilustración 3.8: Endpoint para registro de un usuario

Para la confirmación de la cuenta en la se observa cómo es que al cliente le llega un correo con la finalidad de verificar la cuenta, al momento de dar clic en el botón, se le redirecciona a una página que le confirma la verificación de la cuenta. El mensaje del correo se muestra en la **Ilustración 7**, el mensaje que confirma la verificación de la cuenta se observa en la **Ilustración 3.9**.

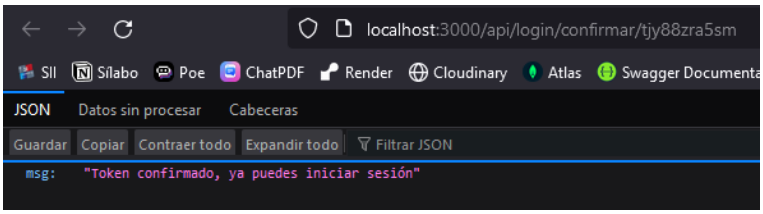


Ilustración 3.9: Endpoint para verificar la cuenta

Para reestablecer la contraseña el usuario debe de proporcionar el correo al cual le llega un mensaje para restituirla, así como se muestra en la **Ilustración 3.10**, al usuario del cliente le llega un correo como se muestra en la **Ilustración 8**, en el mensaje del correo hay un botón que al momento de dar clic en confirma el cambio de contraseña, así como

se observa en la **Ilustración 3.11** y le redirecciona a una página que le permite cambiar su contraseña. En la **Ilustración 3.12** se observa los datos que se necesitan para reestablecer la contraseña.

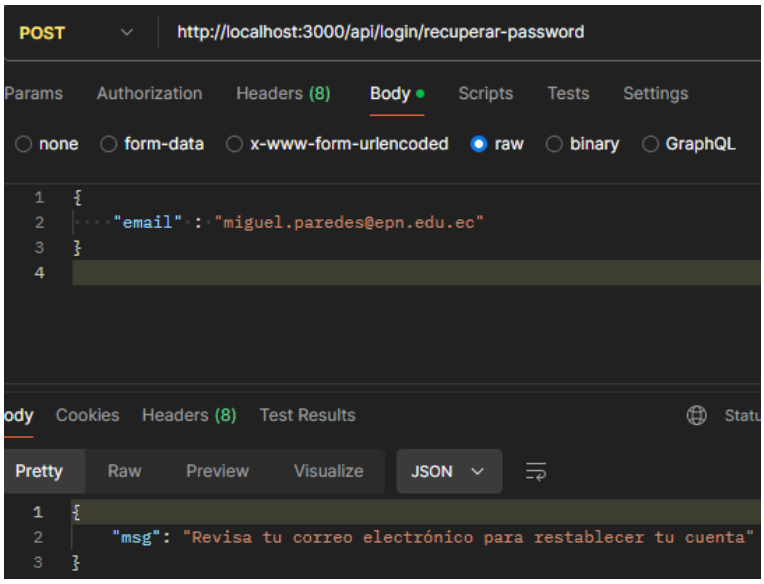


Ilustración 3.10: Endpoint para solicitar reestablecer la contraseña

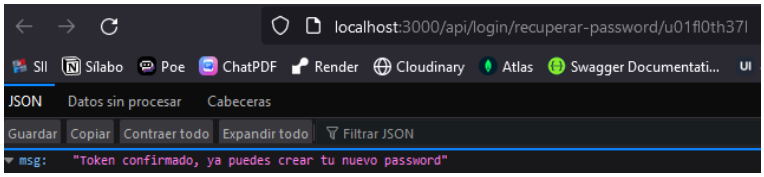


Ilustración 3.11: Endpoint para confirmar el cambio de contraseña

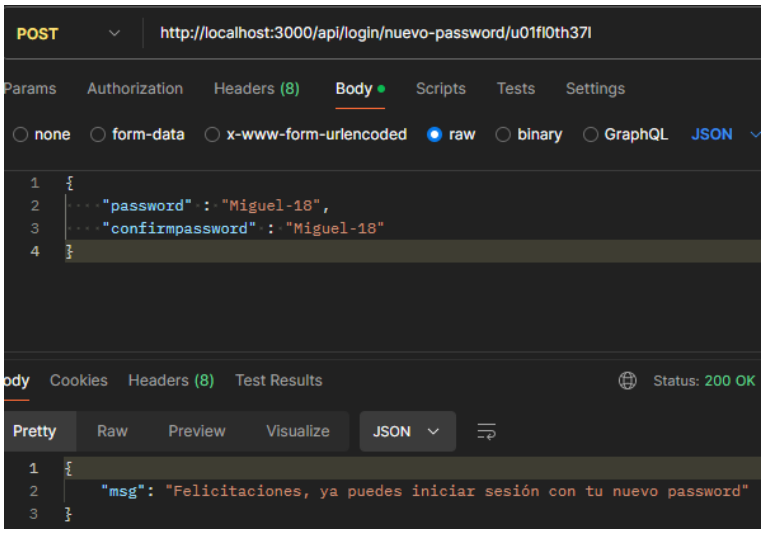


Ilustración 3.12: Endpoint para cambiar la contraseña

Inicio y cierre de sesión

En la elaboración de esta tarea, se crean dos endpoints que nos permiten iniciar y cerrar la sesión. La **Ilustración 9** muestra los datos enviados a través del método POST para iniciar sesión. Además, la **Ilustración 10** indica como se cierra la sesión.

CRUD productos favoritos

En la elaboración de esta tarea, se crean varios endpoints que permiten agregar, eliminar, mostrar, buscar productos favoritos y buscar productos favoritos por categoría. La **Ilustración 11** muestra los datos enviados a través del método POST para agregar un producto a favoritos. Además, **Ilustración 12**, **Ilustración 13**, **Ilustración 14** e **Ilustración 15**, proporcionan información sobre los métodos, los campos necesarios y el resultado de eliminar, mostrar, buscar productos favoritos y buscar productos favoritos por categoría respectivamente.

3.4 Sprint 3. Diseño e implementación de funcionalidades gestión de pedidos y ventas

La planificación de las tareas para la elaboración de este Sprint son las siguientes:

- Enpoints para visualizar y buscar productos.
- Endpoints para gestión del carrito.
- Endpoints para gestionar pedidos.
- Endpoints para gestionar ventas.
- Enpoints para gestionar el estado de los pedidos.
- Enpoints para reporte de pedidos y ventas.

Visualizar y buscar productos

En el desarrollo de esta tarea se crean endpoints que permiten ver y buscar categorías, mostrar productos de una categoría, ver y buscar productos. Para ver las categorías se usa el método GET, en el cual muestra las categorías existentes, en la **Ilustración 3.13** se observan todas las categorías.

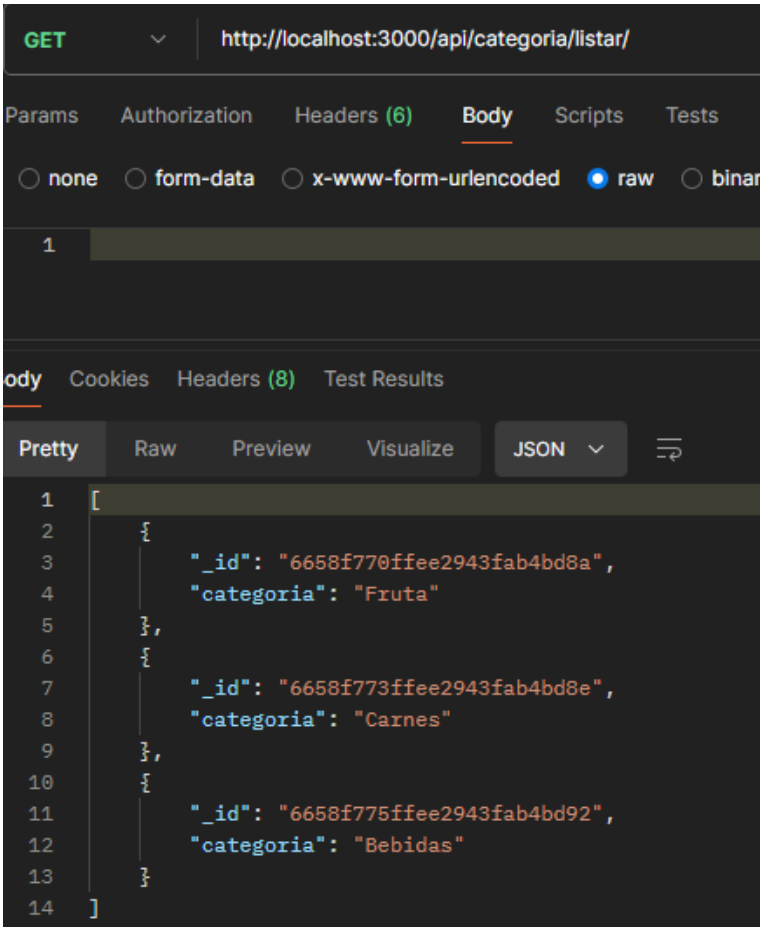


Ilustración 3.13: Endpoint para ver todas las categorías

Para realizar una búsqueda de categorías se usa el método GET, en el cual muestra la categoría que se está buscando. En la **Ilustración 3.14** se observa cómo se realiza una búsqueda de una categoría en específico.

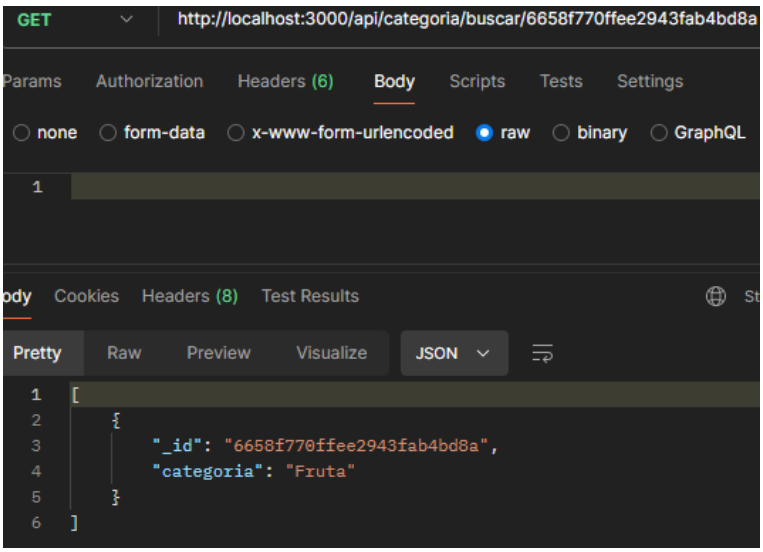


Ilustración 3.14: Endpoint para buscar una categoría

Para ver todos los productos se usa el método GET, el cual nos muestra todos los productos con los que cuenta el minimarket. En la **Ilustración 3.15** se observan todos los productos.

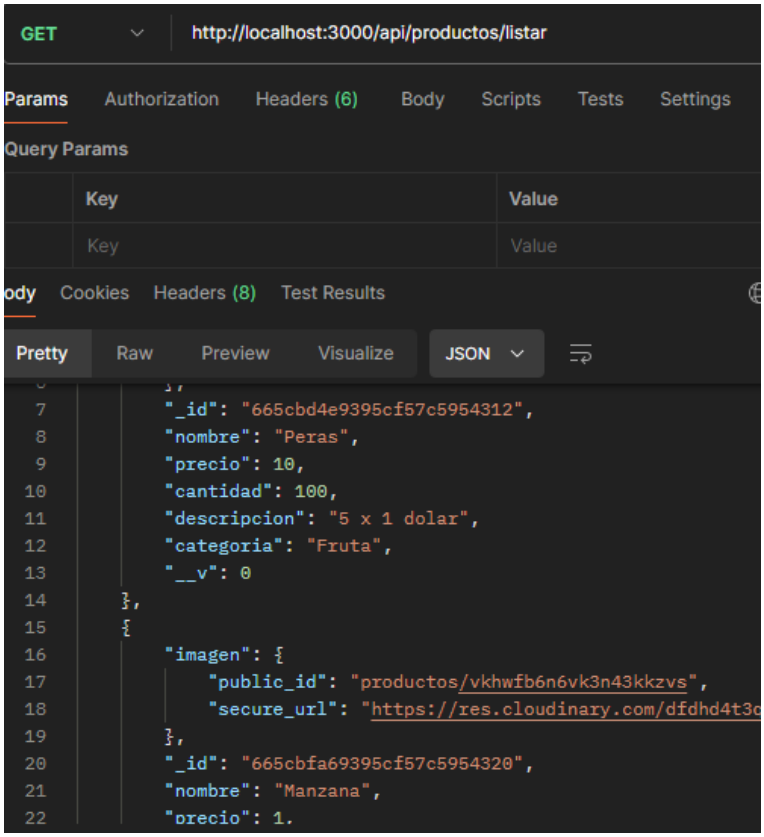


Ilustración 3.15: Endpoint para ver todos los productos

Para buscar un producto se usa el método GET, el cual mostrara el producto buscado. En la **Ilustración 3.16** se observa cómo se realiza la búsqueda de un producto en específico.

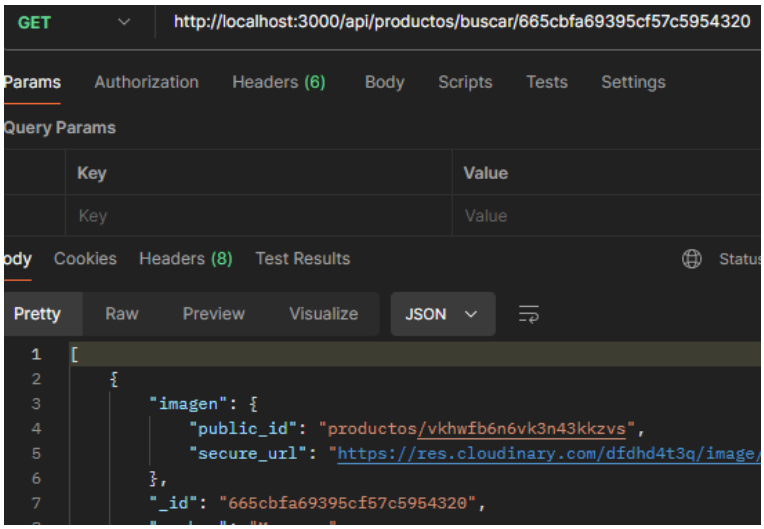


Ilustración 3.16: Endpoint para busca un producto

Para visualizar todos los productos de una mismo tipo se usa el método GET, que va a enseñar todos los productos que contiene esa categoría. En **Ilustración 3.17** la se observan todos los productos de una misma categoría.

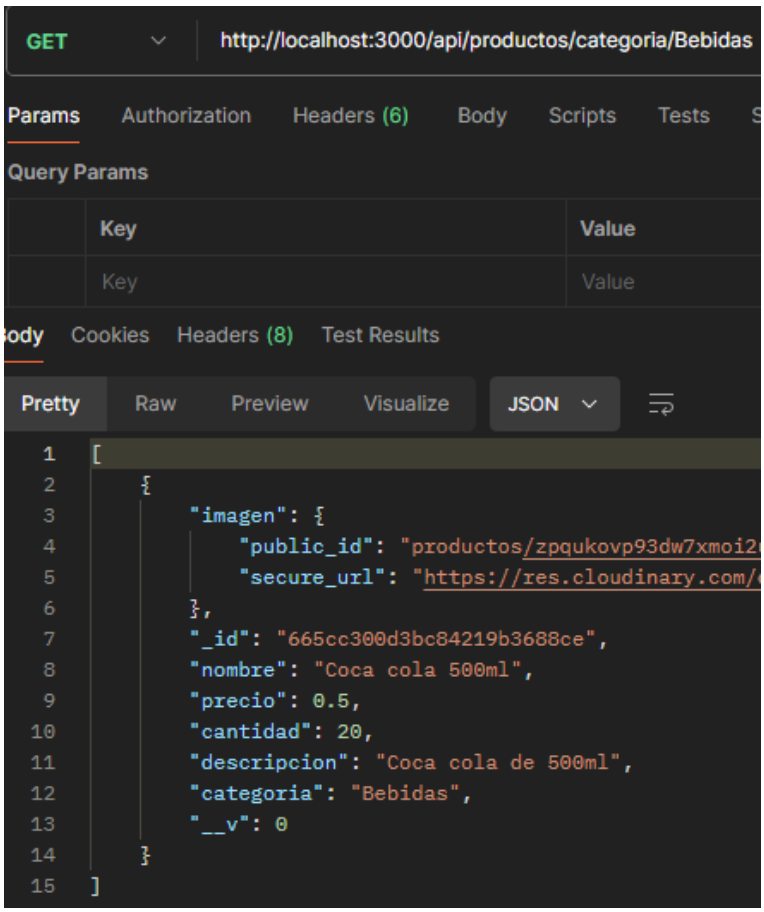


Ilustración 3.17: Endpoint para ver todos los productos de una categoría

CRUD carrito

En el desarrollo de esta tarea se crean endpoints que permiten agregar, mostrar, borrar, actualizar y eliminar todos los productos del carrito. Para agregar productos al carrito se usa el método POST el cual permite almacenar producto en él carrito. La **Ilustración 16** indica la información necesaria para agregar un producto al carrito. Además, la **Ilustración 17**, **Ilustración 18**, **Ilustración 19** e **Ilustración 20,** proporcionan información sobre los métodos, los campos necesarios y el resultado de mostrar, borrar, actualizar y eliminar todos los productos del carrito respectivamente.

Gestión de pedidos

En el desarrollo de esta tarea se crean endpoints que permiten realizar el pedido con los productos del carrito, mostrar, buscar pedidos realizados, y ver el pedido confirmado por correo. Para guardar el pedido se usa el método POST el cual permite realizar el pedido del cliente e indicar si el cliente quiere retirar o que le dejen por una comisión su pedido. La **Ilustración 21** indica la información necesaria para realizar el pedido. Además, la **Ilustración 22** e **Ilustración 23**, proporcionan información sobre los métodos, los campos necesarios y el resultado de mostrar y buscar pedidos realizados respectivamente. En la **Ilustración 24** se observa el correo que le llega al cliente confirmando que se ha realizado el pedido, asimismo en el correo se encuentra un botón que permite observar el pedido realizado, así como se ve en la **Ilustración 25.**

Gestión de ventas

En el desarrollo de esta tarea se crean endpoints que permiten realizar la venta con los productos del carrito, mostrar y buscar ventas realizadas. Para guardar la venta usamos el método POST el cual permite realizar la venta al cajero. La **Ilustración 26** indica la información necesaria para realizar la venta. Además, la **Ilustración 27** e **Ilustración 28**, proporcionan información sobre los métodos, los campos necesarios y el resultado de mostrar y buscar ventas realizados respectivamente.

Gestionar el estado de los pedidos

En el desarrollo de esta tarea se crean endpoints que permiten observar todos los pedidos, ver todos los pedidos que tengan el mismo estado, ver un pedido en específico y de cambiar el estado de los pedidos. Para ver todos los pedidos se usan el método GET el cual permite visualizar los pedidos realizados de todos los clientes. La **Ilustración 29** muestra el resultado del endpoint. Además, la **Ilustración 31**, **Ilustración 32** e **Ilustración 33** proporcionan información sobre los métodos, los campos necesarios y el resultado de ver todos los pedidos con el mismo estado, ver un pedido en específico, de cambiar los estados de los pedidos respectivamente.

Reporte de pedidos y ventas

En el desarrollo de esta tarea se crean endpoints que permiten ver todos los pedidos, ver todas las ventas. Para observar todos los pedidos se usa el método GET que va a indicar todos los pedidos realizados. La **Ilustración 34** muestra el resultado del endpoint. Además, la **Ilustración 35**, proporciona información sobre los métodos, los campos necesarios y el

resultado de ver todas las ventas, filtrar para encontrar un pedido y filtrar para encontrar una venta respectivamente.

3.5 Sprint 4. Pruebas y despliegue

La planificación de las tareas para la elaboración de este Sprint son las siguientes:

- Pruebas:
    - Pruebas unitarias.
    - Pruebas de carga.
    - Pruebas de estrés.
- Depliegue
    - Despliegue del proyecto.

Pruebas unitarias

Las pruebas unitarias tienen el objetivo de verificar que porciones de código funcionen correctamente de manera individual. Para llevar a cabo estas pruebas, se separa y encapsula el código para poder realizar pruebas individuales, lo que permite encontrar errores y garantizar que el componente funcione como se espera [44].

Se realizan pruebas específicas para múltiples controladores del sistema, con un total de 3 pruebas ejecutadas para cada archivo de controladores. Para realizar las pruebas se utiliza el framework Jest el cual nos permite verificar el correcto comportamiento de los controladores. En **Ilustración 3.18** se observa que las pruebas para los controladores de inicio de sesión del cajero, buscar un cajero en específico y mostrar todos los cajeros resultaron satisfactorias.

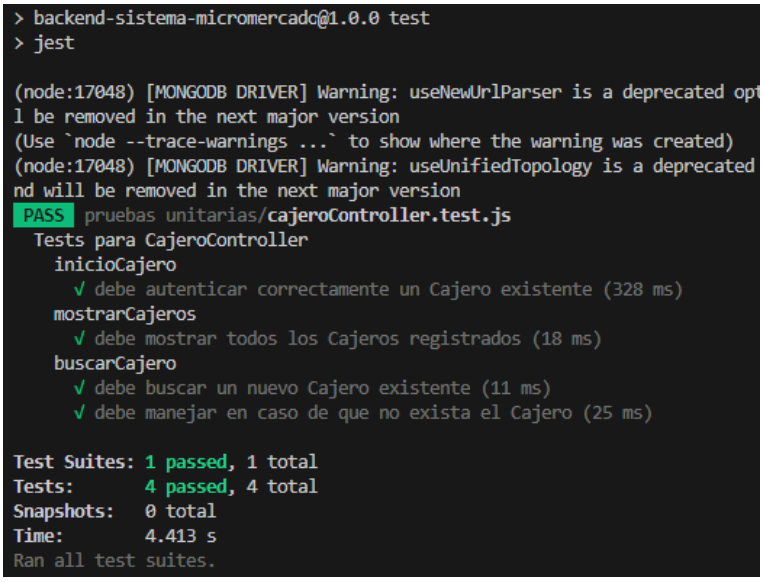


Ilustración 3.18: Resultado prueba unitaria Cajeros

En **Ilustración 36** se observa que las pruebas para los controladores de mostrar las categorías, registrar categorías y actualizar categorías resultaron satisfactorias. En **Ilustración 37** se observa que las pruebas para los controladores de registrar un producto en favoritos, mostrar todos los favoritos y mostrar todos los productos favoritos de una categoría resultaron satisfactorias.

En **Ilustración 38** se observa que las pruebas para los controladores de inicio de sesión del cliente, registrar un nuevo usuario y reestablecer la contraseña resultaron satisfactorias. En **Ilustración 39** se observa que las pruebas para los controladores de mostrar todos los pedidos, buscar un pedido en específico y listar todos los productos del pedido que se está realizando resultaron satisfactorias.

En **Ilustración 40** se observa que las pruebas para los controladores de mostrar todos los productos, buscar un producto en específico y mostrar todos los productos de una categoría resultaron satisfactorias. En **Ilustración 41** se observa que las pruebas para los controladores de ver todos los pedidos realizados por los clientes, ver todas las ventas realizadas por los cajeros y mostrar el historial de ventas del cajero resultaron satisfactorias.

Para el desarrollo de estas pruebas, se recomienda enfocarse en los controladores de métodos GET, ya que son los que suelen ser utilizados por varios usuarios simultáneamente.

Pruebas de carga

Las pruebas de carga son parte de las pruebas que se realizan para ver cómo se comportan las aplicaciones, páginas web y sistemas cuando muchas personas los están usando al mismo tiempo. Estas pruebas simulan lo que sucede cuando muchos usuarios acceden al sistema al mismo tiempo. Ayudan a ver si el sistema puede manejar bien esa situación y a identificar problemas como cuánto puede soportar antes de volverse lento o fallar [45].

Para el desarrollo de estas pruebas se utilizara Locust, mismo que permite simular que el comportamiento de usuarios que van a utilizar nuestro sistema. Se llevó a cabo una prueba con 50 usuarios como máximo, que resultó satisfactoria según se observa en la **Ilustración 3.19**.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/api/categoria/buscar/666654e9a8c13ebcbbb5b63f	144	0	140	160	850	156.35	135	1003	60	2.1	0
GET	/api/categoria/listar	150	0	140	180	240	147.57	134	240	292	1.8	0
GET	/api/favoritos/buscar/6664c272219a547611166ad2?cliente=66665012a8c13ebcbbb5b4ec	145	0	290	330	400	288.8	265	447	36	2	0
GET	/api/favoritos/categoria/Bebidas?cliente=66665012a8c13ebcbbb5b4ec	159	0	280	340	420	291	267	988	51	2	0
GET	/api/favoritos/listar?cliente=66665012a8c13ebcbbb5b4ec	158	0	280	310	360	286.25	265	406	34	1.9	0
GET	/api/productos/buscar/665cbfa69395cf57c5954320	149	0	140	160	200	143.94	3	214	35.76	2.1	0
GET	/api/productos/categoria/Bebidas	148	0	140	180	900	157.73	133	1124	48	1.7	0
GET	/api/productos/listar	176	0	140	180	210	147.1	133	266	2123	2.7	0
	Aggregated	1229	0	150	300	360	202.36	3	1124	372.03	16.3	0

Ilustración 3.19: Resultado pruebas de carga con 50 usuarios

Para estas pruebas, se recomienda utilizar métodos GET, ya que otros métodos como POST, DELETE y UPDATE incluyen validaciones que impiden realizar múltiples peticiones con el mismo usuario. Además, al realizar estas pruebas, es importante considerar el público al que está dirigido el proyecto. Dado que se trata de una tienda de barrio, se realizan las pruebas de carga con un máximo de 50 usuarios simultáneos

Pruebas de estres

Las pruebas de estrés o resistencia se utilizan durante la fase de pruebas de un sistema. Su objetivo es evaluar los límites del sistema y anticipar escenarios de riesgo bajo cargas extremas. Estas pruebas consisten en someter una aplicación a condiciones de carga que superan su capacidad operativa esperada, para observar cómo responde. El propósito es identificar puntos de fallo y determinar los límites del sistema. Durante las pruebas de estrés, se simulan varios escenarios que llevan al software más allá de sus límites normales de funcionamiento, evaluando el tiempo de respuesta, el uso de memoria, el rendimiento y la estabilidad general [46].

Para el desarrollo de estas pruebas se utiliza Locust, mismo que nos configurar la cantidad de usuarios que ira aumentando en por segundo en un tiempo determinado, de esta manera se busca averiguar cuál es la capacidad máxima de respuestas que el Backend puede proporcionar.

Se llevaron a cabo pruebas con 20,000 usuarios para evaluar el rendimiento del sistema bajo una carga significativamente mayor. En la **Ilustración 3.20** se observa que las peticiones totales se mantuvieron mínimas antes de experimentar un aumento exponencial de los fallos. En la **Ilustración 42** se puede ver que los tiempos de respuesta presentaron varios picos, indicando fluctuaciones, mientras que el tiempo de respuesta mediano muestra menos variación.

Ilustración 3.20: Resultado solicitudes totales por segundo con 20 mil usuarios

Finalmente, la **Ilustración 43** se observa el incremento progresivo de los usuarios. Para estas pruebas de estrés, se recomienda realizarlas con una alta cantidad de usuarios, e ir incrementando la cantidad de usuarios con la finalidad de saber el momento exacto en el que el Backend empezara a tener más errores en las respuestas.

Despliegue del proyecto

Para desplegar esta API, se usa la plataforma de Render, la cual nos permite conectar el repositorio directamente. De esta manera, cualquier cambio realizado en el repositorio se implementa automáticamente en Render. El enlace para acceder a la API es el siguiente:

https://mika-y-vale.onrender.com

4 CONCLUSIONES

En esta sección se da a conocer las respectivas conclusiones obtenidas a lo largo del desarrollo del componente Backend:

- Los requerimientos recolectados en el componente Backend fueron documentados y cumplidos dentro de los plazos establecidos. Esto fue posible gracias al Product Owner, quien proporcionó todos los requerimientos necesarios. Esto facilitó el proceso de desarrollo asegurando que el Frontend recibiera todos los endpoints necesarios para su funcionamiento.
- El diseño de la arquitectura del sistema y del modelo de datos se realizó conforme a los requerimientos establecidos, garantizando así una estructura que facilita la escalabilidad y la comprensión del proyecto.
- La metodología SCRUM facilitó la elaboración de los endpoints necesarios para que sean consumidas por el Frontend. Además, dividió el trabajo en sprints para poder implementar todas las funcionalidades de manera secuencial y ordenada, lo que permitió manejar eficientemente los cambios solicitados por el Product Owner.
- El realizar pruebas luego de realizar la API REST nos permite revisar las interacciones que tiene el sistema con los usuarios posibles puntos débiles o áreas de mejora, las pruebas unitarias nos permitieron ver cómo interactúa el sistema con los usuarios, las pruebas de carga permitieron simular peticiones de varios usuarios al mismo tiempo y las pruebas de estrés nos permitieron ver el límite de nuestro proyecto.
- Desplegar el Backend en un entorno remoto nos permite acceder a él desde cualquier parte del mundo, facilita la escalabilidad y posibilita la implementación de las actualizaciones necesarias para cumplir con todas las necesidades requeridas

5 RECOMENDACIONES

En este capítulo se presentan las recomendaciones identificadas a lo largo del desarrollo del componente Backend:

- Para asegurar una integración efectiva entre el Backend y el Frontend, es crucial entender las necesidades específicas del cliente. Esto permitirá que el Backend proporcione los endpoints necesarios para el consumo adecuado por parte del Frontend. Es recomendable especificar claramente los campos que el Frontend debe incluir al realizar solicitudes, garantizando así que se ejecuten correctamente y evitando posibles bloqueos temporales por CORS.
- Después de desarrollar los controladores, es crucial revisar que estén funcionando correctamente para garantizar su consumo posterior. Esto no solo previene problemas para el Frontend, sino que también permite identificar los campos necesarios que este debe enviar para un funcionamiento óptimo.
- Es fundamental utilizar herramientas que permitan simular el envío de datos, como imágenes e información. Esto asegura el correcto funcionamiento de los controladores encargados de realizar modificaciones en la base de datos. Estas prácticas ayudan a evitar posibles fallos durante la implementación y garantizan una integración fluida del sistema.
- Es fundamental mantener una comunicación continua con el equipo de Frontend para entender y anticipar sus necesidades. Esto ayuda a prevenir cambios de último momento en los modelos o controladores.
- Antes de proceder con la conexión de los despliegues de Backend y Frontend, es recomendable realizar pruebas locales para verificar cómo interactúan ambos componentes. Esto garantiza una integración fluida y correcta entre ambos sistemas.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] Altitude Digital Marketing. “7 ventajas y 7 desventajas de un comercio electrónico (Ecommerce)”. Altitude Digital Marketing. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://altitude.ec/ecommerce/comercio-digital-ventajas/

[2] E. Bello. “¿Qué es ecommerce y cómo crear tu propio comercio electrónico?” Thinking for Innovation. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.iebschool.com/blog/comercio-online-ecommerce/

[3] Diana. “Ventajas del comercio electrónico: todo a saber y a evitar”. Tutoriales Hostinger. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.hostinger.es/tutoriales/comercio-electronico-ventajas?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=9069516&gad_source=1&gclid=EAIaIQobChMI3fei4fb2hQMV259aBR3PbACfEAAYASAAEgLgs_D_BwE

[4] E. Bello. “Desarrollo de software: Todo lo que tienes que saber”. Thinking for Innovation. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.iebschool.com/blog/desarrollo-de-software-tecnologia/

[5] M. Coppola. “Desarrollo web: qué es, etapas y principales lenguajes”. Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://blog.hubspot.es/website/que-es-desarrollo-web#que-es

[6] Gustavo. “¿Qué es JavaScript? Introducción básica a JS para principiantes”. Tutoriales Hostinger. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.hostinger.es/tutoriales/que-es-javascript-introduccion-basica/#¿Que_es_JavaScript

[7] NodeJs. “Node.js — Run JavaScript Everywhere”. Google Translate. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://nodejs-org.translate.goog/en?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sc&_x_tr_hist=true

[8] Kinsta. “¿Qué es Express.js? Todo lo que Debes Saber”. Kinsta®. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://kinsta.com/es/base-de-conocimiento/que-es-express/

[9] IBM. “¿Qué es una API REST? | IBM”. IBM in Deutschland, Österreich und der Schweiz. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.ibm.com/es-es/topics/rest-apis

[10] MongoDB. “MongoDB atlas | plataforma de datos multicloud para desarrolladores | mongodb”. MongoDB. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.mongodb.com/es/atlas

[11] Microsoft. “Conceptos básicos sobre bases de datos - Soporte técnico de Microsoft”. Microsoft Support. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://support.microsoft.com/es-es/topic/conceptos-básicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204

[12] F. Coelho. “Metodología: qué es, concepto y definición”. Enciclopedia Significados. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.significados.com/metodologia/

[13] A. Estrada y K. Alfaro. “El método de casos como alternativa pedagógica para la enseñanza de la bibliotecología y las ciencias de la información”. Scielo. Accedido el 5 de mayo de 2024. [En línea]. Disponible: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-358X2015000100009

[14] Universitat Carlemany. “¿Cuáles son las mejores metodologías de desarrollo de software?” UCMA. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://www.universitatcarlemany.com/actualidad/blog/metodologias-de-desarrollo-de-software/

[15] Santander Universidades. “Metodologías de desarrollo de software: ¿qué son?” Santander | Open Academy. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html

[16] Redacción APD. “¿Qué es la metodología Scrum y cómo aplicarla? | APD”. apd. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://www.apd.es/metodologia-scrum-que-es/

[17] donetonic. “Qué es un sprint en la metodología Scrum y qué beneficios aporta”. DoneTonic. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://donetonic.com/es/que-es-un-sprint-en-la-metodologia-scrum/

[18] Imagina Group. “Product owner en scrum: Rol y funciones”. ✓ Formación para empresas | Cursos bonificados FUNDAE. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://imaginaformacion.com/tutoriales/que-es-product-owner-scrum

[19] C. Staff. “¿Qué es un Scrum Master (y cómo llegar a serlo)? | Coursera”. Coursera. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://www.coursera.org/mx/articles/what-is-a-scrum-master

[20] Deloitte. “Scrum: Roles y responsabilidades | Deloitte España”. Deloitte Spain. Accedido el 12 de mayo de 2024. [En línea]. Disponible: https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html

[21] VIEWNEXT. “Artefactos Scrum ¿Qué son y para qué sirven? - Viewnext”. Viewnext. Accedido el 16 de mayo de 2024. [En línea]. Disponible: https://www.viewnext.com/artefactos-scrum/

[22] Miro. “Artefactos scrum: ¿qué son? ¿para qué sirven? Mira ejemplos”. https://miro.com/. Accedido el 16 de mayo de 2024. [En línea]. Disponible: https://miro.com/es/agile/que-son-artefactos-scrum/

[23] I. Zabala. “La recopilación de los requisitos de un proyecto - Enredando Proyectos”. Enredando Proyectos. Accedido el 16 de mayo de 2024. [En línea]. Disponible: https://enredandoproyectos.com/recopilar-los-requisitos-de-un-proyecto/

[24] E. Ledesma. “SCRUM: Cómo escribir historias de usuarios sin morir en el intento - Proyectum”. Proyectum. Accedido el 17 de mayo de 2024. [En línea]. Disponible: https://proyectum.com/sistema/blog/scrum-como-escribir-historias-de-usuarios-sin-morir-en-el-intento/

[25] A. Raeburn. “This specialized to-do list keeps developers focused [2024]”. Asana. Accedido el 17 de mayo de 2024. [En línea]. Disponible: https://asana.com/es/resources/product-backlog

[26] Team Asana. “Backlog: Qué es el trabajo pendiente del sprint y ejemplos [2024] • Asana”. Asana. Accedido el 18 de mayo de 2024. [En línea]. Disponible: https://asana.com/es/resources/sprint-backlog

[27] A. Ken. “Arquitectura de software: ¿Qué es y qué tipos hay?” Gluo. Accedido el 19 de mayo de 2024. [En línea]. Disponible: https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay

[28] Yapiko. “Patrones de arquitectura de software | Yapiko”. Yapiko. Accedido el 19 de mayo de 2024. [En línea]. Disponible: https://yapiko.com/es/blog/patrones-arquitectura-software/#¿Que_es_un_patron_de_arquitectura_de_software

[29] OK HOSTING. “Herramientas de desarrollo de software | software de desarrollo”. OK HOSTING. Accedido el 19 de mayo de 2024. [En línea]. Disponible: https://okhosting.com/blog/herramientas-de-desarrollo-de-software/

[30] F. Flores. “Qué es visual studio code y qué ventajas ofrece”. OpenWebinars.net. Accedido el 19 de mayo de 2024. [En línea]. Disponible: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

[31] J. Gómez. “Qué es git y para qué sirve | Tokio”. Tokio School. Accedido el 19 de mayo de 2024. [En línea]. Disponible: https://www.tokioschool.com/noticias/que-es-git/

[32] Juan. “¿Qué es postman? Características y ventajas - assembler institute”. Assembler Institute. Accedido el 20 de mayo de 2024. [En línea]. Disponible: https://assemblerinstitute.com/blog/que-es-postman/

[33] Render. “The render API | render docs”. The Render API | Render Docs. Accedido el 20 de mayo de 2024. [En línea]. Disponible: https://docs.render.com/api

[34] T. Hamilton. “Tutorial de pruebas de backend (ejemplos)”. Guru99. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.guru99.com/es/what-is-backend-testing.html

[35] José Luis. “¿Qué es la biblioteca bcrypt en node? - geek nómada”. Geek Nómada. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://geeknomada.blog/nodejs/que-es-la-biblioteca-bcrypt-en-node/

[36] NPM. “Cloudinary”. npm. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.npmjs.com/package/cloudinary

[37] NPM. “Cors”. npm. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.npmjs.com/package/cors

[38] NPM. “Dotenv”. npm. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.npmjs.com/package/dotenv

[39] A. Casero. “¿Qué es req.file en Node.js? | KeepCoding Bootcamps”. KeepCoding Bootcamps. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://keepcoding.io/blog/que-es-req-file-en-node-js/

[40] M. Rolfo. “Validación de entradas de usuario en su aplicación Express.js con express-validator”. Codigoencasa.com. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://codigoencasa.com/validacion-de-entradas-de-usuario-en-su-aplicacion-express-js-con-express-validator/

[41] NPM. “Fs-extra”. npm. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.npmjs.com/package/fs-extra

[42] A. Torres. “Introducción a Mongoose para MongoDB”. freeCodeCamp.org. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.freecodecamp.org/espanol/news/introduccion-a-mongoose-para-mongodb/

[43] E. E. P. Gomez. “Cómo usar nodemailer para enviar correos electrónicos desde tu servidor node.js”. freeCodeCamp.org. Accedido el 24 de mayo de 2024. [En línea]. Disponible: https://www.freecodecamp.org/espanol/news/como-usar-nodemailer-para-enviar-correos-electronicos-desde-tu-servidor-node-js/

[44] Yeeply. “▷ ¿Qué son las pruebas unitarias y cómo llevar una a cabo?” Yeeply. Accedido el 27 de junio de 2024. [En línea]. Disponible: https://www.yeeply.com/blog/tendencias-habilidades/que-son-pruebas-unitarias/

[45] OpenText. “¿Qué son las pruebas de carga? ¿Cómo funciona? OpenText”. OpenText. Accedido el 28 de junio de 2024. [En línea]. Disponible: https://www.opentext.com/es-es/que-es/load-testing

[46] Testing IT. “Pruebas de estrés de software: ¿qué son y para qué sirven?” Empresa de pruebas de software | Testing IT. Accedido el 29 de junio de 2024. [En línea]. Disponible: https://www.testingit.com.mx/blog/pruebas-de-estres-de-software

7 ANEXOS

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

- **ANEXO I.** Certificado de Originalidad.
- **ANEXO II.** Manual técnico
- **ANEXO III.** Manual de usuario
- **ANEXO IV.** Manual de instalación

ANEXO I

F_AA_236

CERTIFICADO DE ORIGINALIDAD

TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 26 de julio de 2024

De mi consideración:

Yo, **YADIRA GUISSELA FRANCO ROCHA**, en calidad de Director del Trabajo de Integración Curricular titulado **BACKEND** asociado al **DESARROLLO DE UN SISTEMA E-COMMERCE PARA EL MINIMARKET "MIKA Y VALE",** elaborado por el estudiante **MIGUEL ANGEL PAREDES REASCOS,** de la carrera en **DESARROLLO DE SOFTWARE**, certifico que he empleado la herramienta antiplagio "**TURNITIN**" para la revisión de originalidad del documento escrito producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del **8%.**

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,

ANEXO II

Recopilación de requerimientos, historias de usuario, producto backlog, Sprint backlog y las evidencias realizadas de los resultados se encuentran a continuación.

Recopilación de requerimientos

Tabla 1: Recopilación de Requerimientos

RECOPILACION DE REQUERIMIENTOS		
TIPO DE SISTEMA	**ID - RR**	**ENUNCIADO DEL ITEM**
BACKEND	**RR-002**	Como usuario **administrador** se necesita generar varios endpoints para: • Gestionar perfiles de cajeros. • Visualizar cajeros creados. • Buscar el perfil de un cajero. • Gestionar categorías. • Gestionar productos.
	RR-003	Como usuario **administrador, cajeros** y **cliente** se necesita generar varios endpoints para: • Iniciar Sesión. • Cerrar Sesión.
	RR-004	Como usuario **administrador**, **cliente, cajero y visitante** se necesita generar endpoints para: • Visualizar las categorías de productos. • Buscar una categoría • Visualizar productos. • Buscar un producto. • Buscar productos por su categoría.
	RR-005	Como usuario **cajero y cliente** se necesitan varios endpoints para: • Agregar productos al carrito. • Visualizar el carrito. • Realizar pedidos o ventas. • Ver los pedidos o ventas realizadas. • Buscar pedidos o ventas realizadas. • Ver el estado de pedidos

	RR-006	Como usuario **administrador y cajero** se necesitan varios endpoints para: • Visualizar todos los pedidos y ventas de todos los clientes y cajeros. • Buscar un pedido o venta realizado por cualquier cliente o cajero. • Visualizar el estado de los pedidos. • Buscar pedidos según su estado. • Cambiar el estado de un Pedido.
	RR-007	Como usuario **cliente** se necesita generar endpoint para: • Agregar o eliminar un producto en la lista de productos favoritos. • Visualizar mis productos favoritos. • Buscar un producto de mi lista de favoritos. • Buscar los productos por categoría de mi lista de favoritos.

Historias de usuario

Tabla 2: Historias de Usuario 2

<table>
<tr><td colspan="2">HISTORIA DE USUARIO</td></tr>
<tr><td>Identificador: RR-002</td><td>Usuarios: Administrador</td></tr>
<tr><td colspan="2">Nombre Historia: CRUD de cajeros, categorías y productos.</td></tr>
<tr><td>Prioridad en el negocio: Alta</td><td>Riesgo en el desarrollo: Medio</td></tr>
<tr><td colspan="2">Iteración asignada: 2</td></tr>
<tr><td colspan="2">Responsable: Miguel Paredes</td></tr>
<tr><td colspan="2">Descripción: Como usuario administrador, necesito gestionar perfiles de cajeros, visualizar los cajeros creados, gestionar categorías y productos, ver todos los pedidos de los clientes y buscar un pedido específico de un cliente.
Gestionar perfiles de cajeros
Visualizar cajeros creados
Gestionar categorías
Gestionar productos</td></tr>
<tr><td colspan="2">CA:
Gestionar perfiles de cajeros:
Endpoint para gestionar los perfiles de cajeros.
Ingresar información básica del cajero: nombre, contraseña, correo y celular.
Validación de datos y creación de cuenta en el sistema.
Se crearán nuevos perfiles de cajeros si el correo y/o teléfono no se encuentran previamente registrados.
Visualizar cajeros creados:
Endpoint para visualizar todos los perfiles de cajeros.
Buscar el perfil de un cajero:
Endpoint para buscar el perfil de un cajero.
Gestionar categorías:
Endpoint para gestionar las categorías.
No se puede crear dos veces la misma categoría.
Gestionar productos:
Endpoint para gestionar los productos.
No se puede crear dos veces el mismo producto.
Se debe de verificar la existencia de la categoría antes de crear el nuevo producto.
Se debe de colocar la imagen del producto.
Se ingresan los siguientes campos para la creación del producto: nombre, precio, cantidad descripción, categoría e imagen.</td></tr>
<tr><td colspan="2">Observación: Solo el administrador tendrá acceso a estos endpoints.</td></tr>
</table>

Tabla 3: Historias de Usuario 3

<table>
<tr><td colspan="2">HISTORIA DE USUARIO</td></tr>
<tr><td>Identificador: RR-003</td><td>Usuarios: Usuario administrador, cajeros y cliente.</td></tr>
<tr><td colspan="2">Nombre Historia: Inicio y cierre sesiones.</td></tr>
<tr><td>Prioridad en el negocio: Alta</td><td>Riesgo en el desarrollo: Bajo</td></tr>
<tr><td colspan="2">Iteración asignada: 1</td></tr>
<tr><td colspan="2">Responsable: Miguel Paredes</td></tr>
<tr><td colspan="2">Descripción: Como usuario administrador, cajero o cliente, quiero poder iniciar sesión, además de una autentificación y autorización en el sistema para tener acceso a las debidas funcionalidades que tiene cada rol por los permisos que tengo, finalmente el cerrar sesión cuando haya terminado de utilizar el sistema.
Iniciar Sesión.
Cerrar Sesión.</td></tr>
<tr><td colspan="2">CA:
Iniciar Sesión:
Endpoints para iniciar sesión.
Ingresar las credenciales de email y contraseña para iniciar sesión.
Cerrar Sesión:
Endpoint para poder cerrar sesión.</td></tr>
<tr><td colspan="2">Observación: Al momento de iniciar sesión se autentificará y autorizaran los permisos respectivos a cada usuario para que puedan realizar sus actividades, además, en caso de que el usuario no realice ninguna acción en su cuenta se cerrara la sesión.</td></tr>
</table>

Tabla 4: Historias de Usuario 4

<table>
<tr><td colspan="2">HISTORIA DE USUARIO</td></tr>
<tr><td>Identificador: RR-004</td><td>Usuarios: Usuario administrador, cliente, cajero y visitante.</td></tr>
<tr><td colspan="2">Nombre Historia: Visualización de categorías y productos.</td></tr>
<tr><td>Prioridad en el negocio: Alta</td><td>Riesgo en el desarrollo: Bajo</td></tr>
<tr><td colspan="2">Iteración asignada: 1</td></tr>
<tr><td colspan="2">Responsable: Miguel Paredes</td></tr>
<tr><td colspan="2">Descripción: Como usuario administrador, cajero, cliente o público, quiero poder ver, buscar las categorías y productos sin la necesidad de iniciar sesión.
Visualizar las categorías de productos.
Buscar una categoría.
Visualizar productos.
Buscar un producto.
Buscar productos por su categoría.</td></tr>
<tr><td colspan="2">CA:
Visualizar las categorías de productos:
Endpoints para visualizar las categorías de los productos.
Cualquier persona puede ver las categorías.
Buscar una categoría:
Endpoints para buscar una categoría.
Cualquier persona puede buscar una categoría.
Visualizar productos:
Endpoints para ver productos.
Cualquier persona puede ver productos.
Buscar un producto:
Endpoints para buscar un producto.
Cualquier persona puede buscar un producto.
Buscar productos por su categoría:
Endpoints para buscar un producto.
Cualquier persona puede buscar productos según su categoría.</td></tr>
<tr><td colspan="2">Observación: Cualquier persona que visite el sitio web puede ver y buscar categorías y/o productos</td></tr>
</table>

Tabla 5: Historias de Usuario 5

HISTORIA DE USUARIO	
Identificador: RR-005	**Usuarios:** Usuario cliente y cajero.
Nombre Historia: Gestión de Pedidos y Ventas.	
Prioridad en el negocio: Alta	**Riesgo en el desarrollo:** Alta
Iteración asignada: 3	
Responsable: Miguel Paredes	
Descripción: Como usuario cajero o cliente, quiero poder agregar productos a mi carrito, visualizar mi carrito, realizar un pedido en base a mi carrito, ver los pedidos que he realizado y buscar un pedido en específico. • Agregar productos al carrito. • Visualizar el carrito. • Realizar pedidos o ventas. • Ver los pedidos o ventas realizadas. • Buscar pedidos o ventas realizadas. • Ver el estado de pedidos.	
CA: • **Agregar productos al carrito:** o Validación de inicio de sesión del usuario. o Endpoints para gestionar los productos del carrito. o El usuario selecciona uno a uno los productos y la cantidad que necesita de cada uno. o Cada usuario tiene su propio carrito de compras. • **Visualizar el carrito:** o Endpoints para visualizar el carrito de compras. o El usuario podrá ver los productos que se encuentran en el carrito. o Antes de realizar el pedido el usuario observara el carrito de compras, con el valor total a pagar. • **Realizar pedidos:** o Endpoints para realizar pedidos. o Cuando el usuario realizar su pedido se envía un correo confirmación para indicar que el pedido se realizó con éxito.	

• **Ver los pedidos realizados:**
  o Endpoints para visualizar los pedidos.
  o El usuario podrá revisar el historial de los pedidos realizados.
• **Buscar pedidos realizados:**
  o Endpoints para buscar un pedido.
  o El usuario podrá buscar un pedido en específico.
• **Ver el estado de pedidos:**
  o Endpoints para visualizar el estado de los pedidos.

Observación: Cuando el cliente ha pagado su pedido podrá realizar un nuevo.

Tabla 6: Historias de Usuario 6

HISTORIA DE USUARIO	
Identificador: RR-006	**Usuarios:** Usuario administrador y cajero.
Nombre Historia: Reportes de pedidos de clientes o cajeros.	
Prioridad en el negocio: Alta	**Riesgo en el desarrollo:** Media
Iteración asignada: 2	
Responsable: Miguel Paredes	
Descripción: Como usuario administrador o cajero, quiero visualizar cualquier pedido realizado, buscar el pedido de un cliente en específico y cambiar los estados de los pedidos. • Visualizar todos los pedidos y ventas de todos los clientes y cajeros. • Buscar un pedido realizado por cualquier cliente y/o cajero. • Visualizar el estado de los pedidos. • Buscar pedidos según su estado. • Cambiar el estado de un Pedido.	

CA:
• **Visualizar todos los pedidos y/o ventas de todos los clientes y cajeros:**
  o Validación de inicio de sesión del usuario.
  o Endpoints para visualizar todos los pedidos y ventas.
  o El personal del minimarket podrá visualizar todos los pedidos realizados.
• **Buscar un pedido o venta realizado por cualquier cliente o cajero:**	

○ Endpoints para buscar un pedido de un cliente. ○ El personal del minimarket buscar el pedido de un cliente en específico. • **Visualizar el estado de los pedidos**: ○ Enpoints para ver los estados de todos los productos ○ El personal del minimarket podrá ver los estados de todos los pedidos de los clientes. • **Buscar pedidos según su estado:** ○ Endpoints para poder ver todos los pedidos con un mismo estado. ○ El personal del minimarket podrá buscar los pedidos según el estado en el que estos se encuentren. • **Cambiar el estado de un Pedido:** ○ Endpoint para cambiar el estado de un pedido. ○ Se cambia el estado del pedido a “En preparación”. ○ Cuando ya se va a enviar el pedido se cambia el estado a “Enviado”. ○ Cuando el cliente ya ha pagado se cambia el estado a “Pagado”.
Observación: Tanto el administrador como los perfiles de cajeros creados por el administrador son los únicos roles que podrán ver todos los pedidos y ventas que hayan sido realizados por clientes y por otros cajeros respectivamente.

Tabla 7: Historias de Usuario 7

HISTORIA DE USUARIO	
Identificador: RR-007	**Usuarios:** Usuario cliente.
Nombre Historia: Productos Favoritos.	
Prioridad en el negocio: Alta	**Riesgo en el desarrollo:** Medio
Iteración asignada: 2	
Responsable: Miguel Paredes	
Descripción: Como usuario cliente quiero agregar productos a una lista la cual será de favoritos que me permitirá facilitar la búsqueda de productos. • Agregar o eliminar un producto en la lista de productos favoritos. • Visualizar mis productos favoritos. • Buscar un producto de mi lista de favoritos. • Buscar los productos por categoría de mi lista de favoritos.	

CA:

- **Agregar o eliminar un producto en la lista de productos favoritos:**
  - Endpoint para permitir a un usuario cliente agregar un producto a su lista de favoritos.
  - Validación de inicio de sesión del usuario.
  - Almacenamiento del producto en una lista de favoritos del usuario.
  - Confirmación de la operación exitosa.
- **Visualizar mis productos favoritos:**
  - Endpoint para poder ver los productos que el cliente tiene en su lista de favoritos.
- **Buscar un producto de mi lista de favoritos:**
  - Enpoint para poder buscar un producto de la lista de favoritos.
- **Buscar los productos por categoría de mi lista de favoritos:**
  - Enpoint para poder ver todos los productos de una categoría de la lista de favoritos.

Observación: Solo el cliente que haya iniciado sesión podrá colocar productos en su lista de favoritos, podrá ver únicamente su lista de productos favoritos y realizar pedidos.

Product Backlog

Tabla 8: Product Backlog

ELABORACIÓN DEL PRODUCT BACKLOG				
ID-HU	**HISTORIA DE USUARIO**	**ITERACIÓN**	**ESTADO**	**PRIORIDAD**
HU-001	Cuenta del cliente	2	Finalizado	Media
HU-002	CRUD de cajeros, categorías y productos	2	Finalizado	Media
HU-003	Inicio y cierre de sesiones	1	Finalizado	Baja
HU-004	Visualización de categorías y productos	1	Finalizado	Baja
HU-005	Gestión de Pedidos y Ventas	3	Finalizado	Alta
HU-006	Reportes de pedidos de clientes o cajeros	2	Finalizado	Media
HU-007	Productos Favoritos	2	Finalizado	Media

Sprint Backlog

Tabla 9: Sprint Backlog

ELABORACIÓN DE SPRINT BACKLOG						
ID-SB	**NOMBRE**	**MODULO**	**ID-HU**	**HISTORIA DE USUARIO**	**TAREA**	**TIEMPO ESTIMADO**
SB-000	Configuraciones para el desarrollo del proyecto.	N/A	N/A	N/A	• Diseño de base de datos • Recolección de requerimientos funcionales y no funcionales • Diseño de la arquitectura del módulo backend • Estructura del proyecto • Instalación de dependencias necesarias	20H
SB-001	Diseño e implementación de endpoints para las funcionalidades para el personal	Administrador	RR-002	CRUD de cajeros, categorías y productos	• Diseñar endpoints para: o Gestionar perfiles de cajeros o Visualizar cajeros creados o Buscar el perfil de un cajero o Gestionar categorías o Gestionar productos	40H
SB-002	Diseño e implementación de endpoints para las funcionalidades para gestión de cuentas	Clientes	RR-001	Cuenta del cliente	• Diseño de endpoints que permitan: o Registrarse. o Confirmar la cuenta. o Reestablecer contraseña.	30H
		Sesiones	RR-003	Inicio y cierre sesiones	• Diseño de endpoints que permitan: o Iniciar Sesión. o Cerrar Sesión.	20H
		Lista de Favoritos	RR-007	Productos Favoritos	• Diseño de enpoints que permitan: o Agregar o eliminar un producto en la lista de productos favoritos. o Visualizar mis productos favoritos. o Buscar un producto de mi lista de favoritos. o Buscar los productos por categoría de mi lista de favoritos.	10H
SB-003	Diseño e implementación de endpoints para las	Presentación de Productos	RR-004	Visualización de categorías y productos	• Diseño de endpoints que permitan:	25H

	funcionalidades para gestión de pedidos y ventas				○ Visualizar las categorías de productos. ○ Buscar una categoría. ○ Visualizar productos. ○ Buscar un producto. ○ Buscar productos por su categoría.	
		Cajero y Cliente	RR-005	Gestión de Pedidos y Ventas	• Diseño de endpoints que permitan: ○ Agregar productos al carrito. ○ Visualizar el carrito. ○ Realizar pedidos o ventas. ○ Ver los pedidos o ventas realizadas. ○ Buscar pedidos o ventas realizadas. ○ Ver el estado de pedidos.	35H
		Administrador y cajero	RR-006	Reportes de pedidos de clientes o cajeros	• Diseñar endpoints para: ○ Visualizar todos los pedidos y ventas de todos los clientes y cajeros. ○ Buscar un pedido o venta realizado por cualquier cliente o cajero. ○ Visualizar el estado de los pedidos. ○ Buscar pedidos según su estado. ○ Cambiar el estado de un Pedido.	20H
SB-004	Pruebas y Despliegue	N/A	N/A	N/A	• Pruebas y despliegue del proyecto.	20H

Herramientas de desarrollo

Tabla 10: Herramientas para el desarrollo

Herramienta	Justificación
Visual Studio Code	Editor de código disponible para Windows, Linux y macOS, cuenta con Git y extensiones que facilitan el uso y ejecución de lenguajes de programación [30].
Node.js	Entorno que permite ejecutar código JavaScript fuera del navegador, permite que desarrolladores puedan aplicaciones web e integrar API REST. [7]
Git	Sistema de control de versiones que permite el trabajo colaborativo por medio de ramas en proyectos [31].
MongoDB	Plataforma de datos multicloud, que usa modelo de datos de documentos tipo JSON, de esta manera se almacenan los datos y se realizan consultas, transfieren, actualizan y analizan en tiempo real [10].
Postman	Permite hacer pruebas para revisar el correcto funcionamiento de proyectos de desarrollos web, además facilita la creación y uso de APIs [32].
Render	Posibilita el desplegué de sitios estáticos, servicios web, servicios privados, etc., es una plataforma que nos permitirá alojar en la nube y desplegar el Backend [33].

Librerías

Tabla 11: Librerías API

Nombre	Versión	Descripción
Bcrypt	5.1.1	Librería que usa un algoritmo lento para el cifrado seguro y verificación de contraseñas [35].
Cloudinary	2.0.3	Api multimedia que permite gestionar, transformar imágenes o videos para sitios web o aplicaciones móviles [36].
Cors	2.8.5	Paquete que proporciona un middleware Connect/Express [37].
Dotenv	16.4.5	Modulo que permite carga las variables de entorno desde archivos .env en aplicaciones Node.js [38].
Express	4.19.2	Framework popular para la elaboración de API REST o servidores proporciona herramientas, peticiones y respuestas HTTP [8].
Express-fileupload	1.5.0	Extensión de Express que permite simplificar la carga de archivos, permite recibir y almacenar los datos del cliente en el lado del servidor [39].
Express-validator	7.0.1	Extensión de Express que nos ayuda con las validaciones [40].
Fs-extra	11.2.0	Agrega archivos que no se encuentran incluidos en el módulo Fs [41].
Mongoose	8.2.3	Biblioteca de modelado de datos que permite conectar con MongoDB y Node.js [42].
Nodemailer	6.9.13	Modulo que permite enviar correos electrónicos por medio de aplicaciones de Node.js [43].

Evidencias de los resultados

Sprint 0

Tabla 12: Configuración del ambiente de desarrollo

Diseño de la Base de Datos
-- Creación de la tabla 'Categoria' CREATE TABLE Categoria (id INT AUTO_INCREMENT PRIMARY KEY, categoria VARCHAR(255) NOT NULL UNIQUE); -- Creación de la tabla 'Producto' CREATE TABLE Producto (id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(255) NOT NULL UNIQUE, precio DECIMAL(10, 2) NOT NULL, cantidad INT NOT NULL, descripcion TEXT, categoria_id INT, imagen_public_id VARCHAR(255), imagen_secure_url VARCHAR(255), FOREIGN KEY (categoria_id) REFERENCES Categoria(id)); -- Creación de la tabla 'Cajero' CREATE TABLE Cajero (id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL UNIQUE, telefono BIGINT NOT NULL UNIQUE, inicioSesion BOOLEAN DEFAULT FALSE); -- Creación de la tabla 'CarritoCajero' CREATE TABLE CarritoCajero (id INT AUTO_INCREMENT PRIMARY KEY, cliente_id INT NOT NULL, producto VARCHAR(255) NOT NULL,

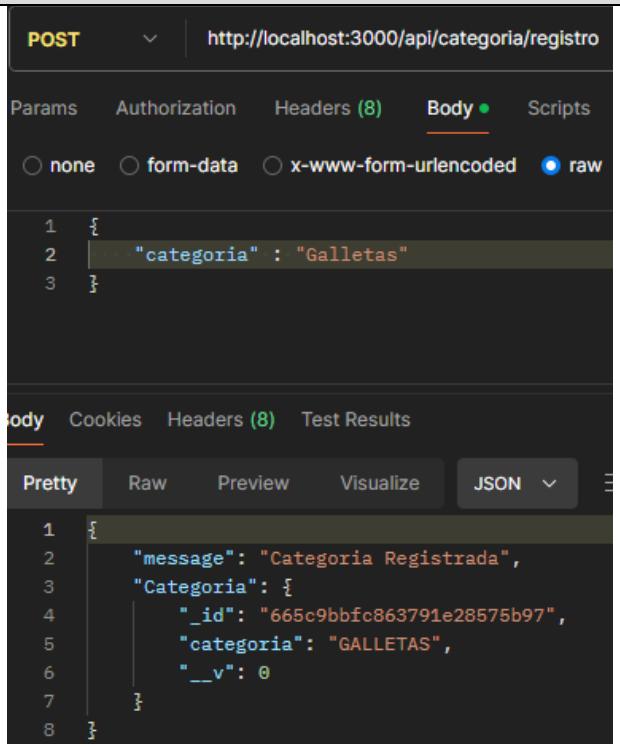
```
    cantidad INT NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (cliente_id) REFERENCES Cajero(id),
    FOREIGN KEY (producto) REFERENCES Producto(nombre)
);
-- Creación de la tabla 'Venta'
CREATE TABLE Venta (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT NOT NULL,
    total DECIMAL(10, 2) NOT NULL,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
    FOREIGN KEY (cliente_id) REFERENCES Cajero(id)
);
-- Creación de la tabla 'VentaProducto'
CREATE TABLE VentaProducto (
    id INT AUTO_INCREMENT PRIMARY KEY,
    venta_id INT NOT NULL,
    producto VARCHAR(255) NOT NULL,
    cantidad INT NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (venta_id) REFERENCES Venta(id)
);
-- Creación de la tabla 'Registro'
CREATE TABLE Registro (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    apellido VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    telefono BIGINT NOT NULL UNIQUE,
    direccion VARCHAR(255) NOT NULL,
    token VARCHAR(255) DEFAULT NULL,
    confirmEmail BOOLEAN DEFAULT FALSE,
    admin BOOLEAN DEFAULT FALSE,
    inicioSesion BOOLEAN DEFAULT FALSE
);
```

```
-- Creación de la tabla 'Carrito'
CREATE TABLE Carrito (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT NOT NULL,
    producto VARCHAR(255) NOT NULL,
    cantidad INT NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (cliente_id) REFERENCES Registro(id),
    FOREIGN KEY (producto) REFERENCES Producto(nombre)
);
-- Creación de la tabla 'Pedido'
CREATE TABLE Pedido (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT NOT NULL,
    comision BOOLEAN DEFAULT FALSE NOT NULL,
    total DECIMAL(10, 2) NOT NULL,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
    estado ENUM('En espera', 'En preparación', 'Enviado', 'Pagado') DEFAULT 'En
espera' NOT NULL,
    FOREIGN KEY (cliente_id) REFERENCES Registro(id)
);
-- Creación de la tabla 'PedidoProducto'
CREATE TABLE PedidoProducto (
    id INT AUTO_INCREMENT PRIMARY KEY,
    pedido_id INT NOT NULL,
    producto VARCHAR(255) NOT NULL,
    cantidad INT NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (pedido_id) REFERENCES Pedido(id)
);
-- Creación de la tabla 'Favorito'
CREATE TABLE Favorito (
    id INT AUTO_INCREMENT PRIMARY KEY,
    producto_id INT NOT NULL,
    cliente_id INT NOT NULL,
    nombre VARCHAR(255) NOT NULL,
```

```
    precio DECIMAL(10, 2) NOT NULL,
    descripcion TEXT,
    categoria VARCHAR(255),
    imagen_public_id VARCHAR(255),
    imagen_secure_url VARCHAR(255),
    FOREIGN KEY (producto_id) REFERENCES Producto(id),
    FOREIGN KEY (cliente_id) REFERENCES Registro(id)
);
```

Sprint 1

Tabla 13: Evidencia CRUD categorías

Gestión de categorías
 **Ilustración 1**: Endpoint para crear categorías

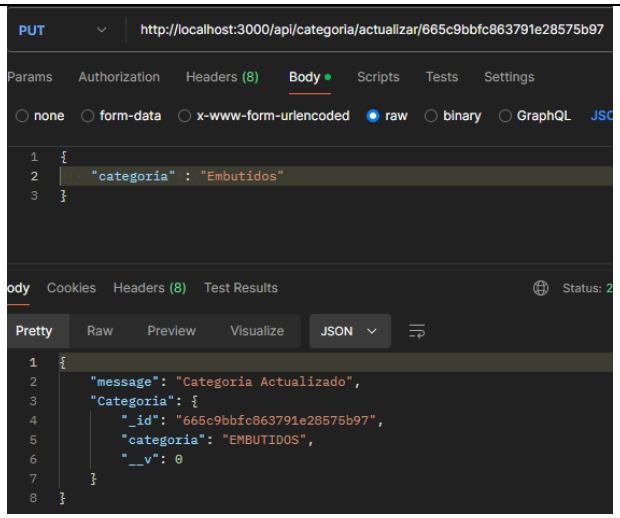


Ilustración 2: Endpoint para actualizar categorías

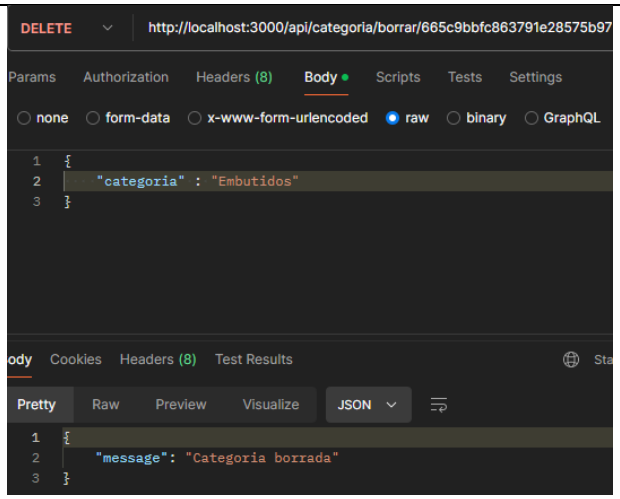
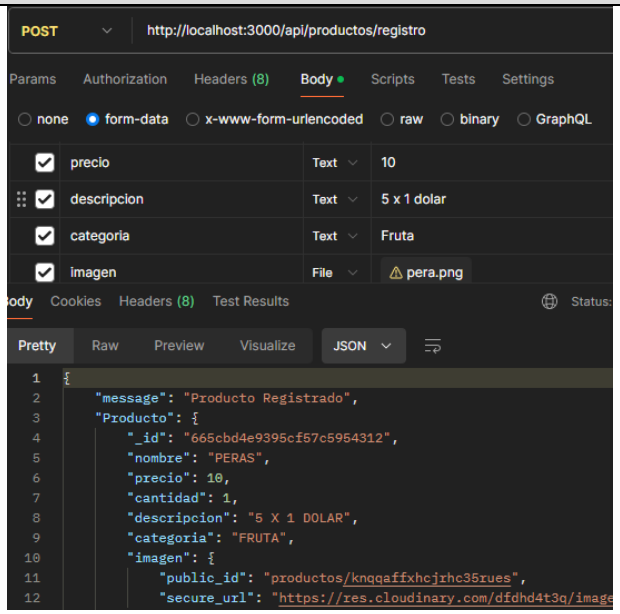
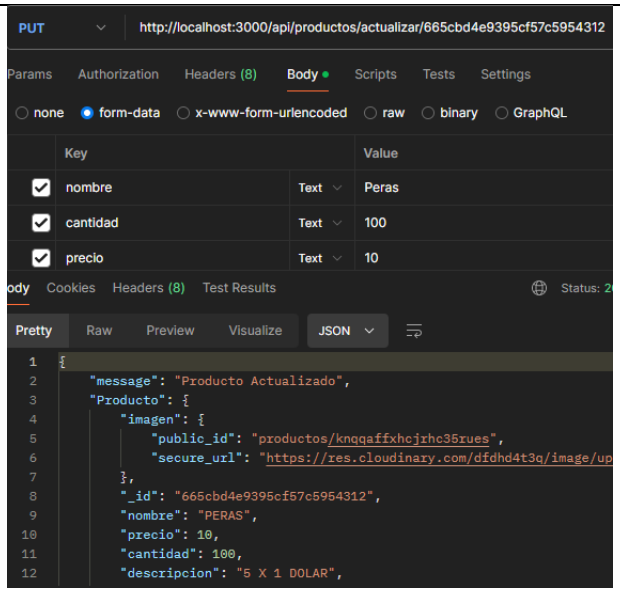


Ilustración 3: Endpoint para eliminar categorías

Tabla 14: Evidencia CRUD productos

Gestión de productos
 **Ilustración 4:** Endpoint para crear productos
 **Ilustración 5:** Endpoint para actualizar productos

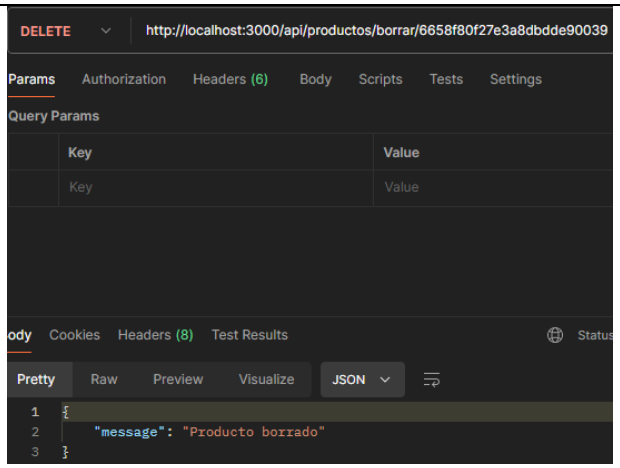


Ilustración 6: Endpoint para eliminar productos

Sprint 2

Tabla 15: Evidencia registro de usuarios

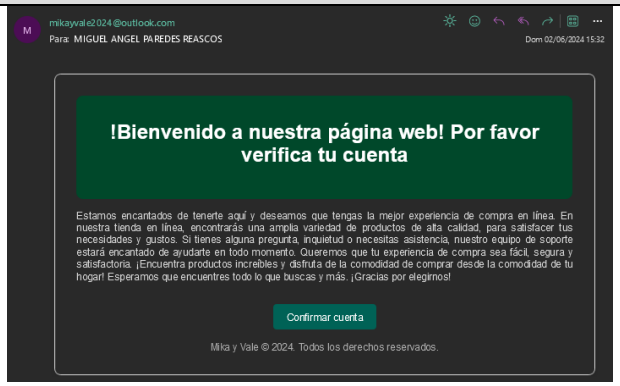
Gestión registro de Usuarios


Ilustración 7: Correo de verificación de cuenta

Ilustración 8: Correo para reestablecer la contraseña

Tabla 16: Evidencia inicio y cierre de sesión

Gestión inicio y cierre de sesión

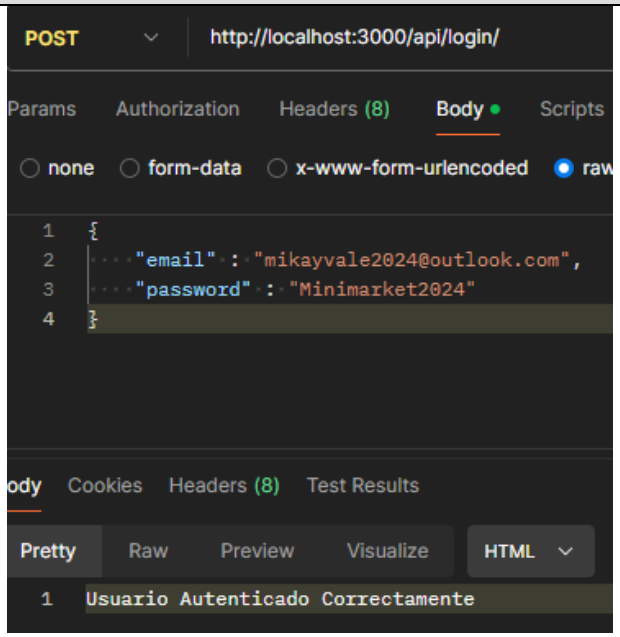



Ilustración 9: Endpoint para inicio de sesión

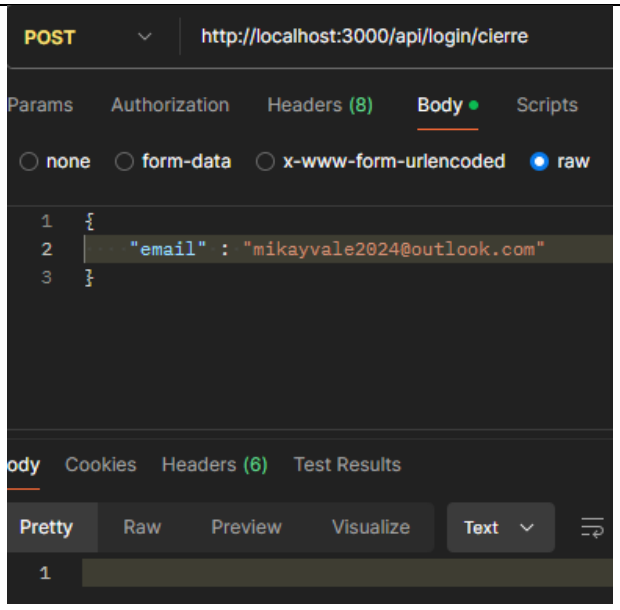



Ilustración 10: Enpoint para cierre de sesión

Tabla 17: CRUD Productos Favoritos

Gestión Productos Favoritos

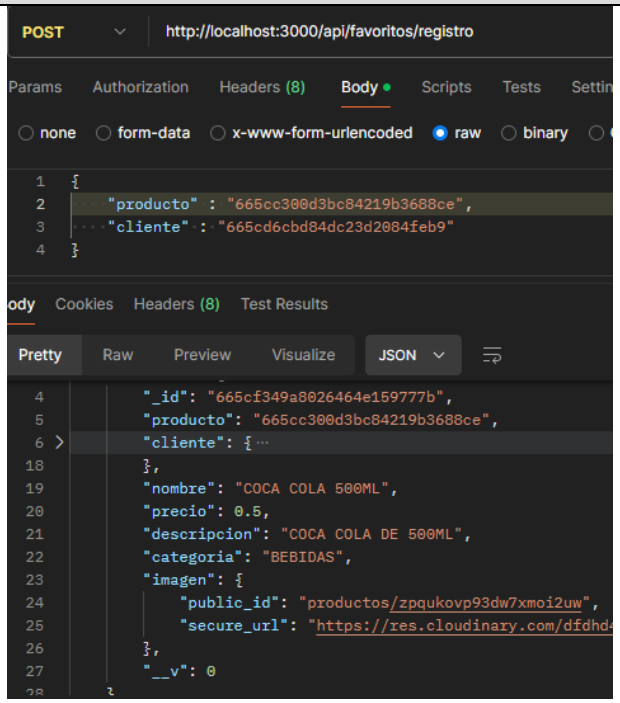



Ilustración 11: Endpoint para agregar un producto a favoritos

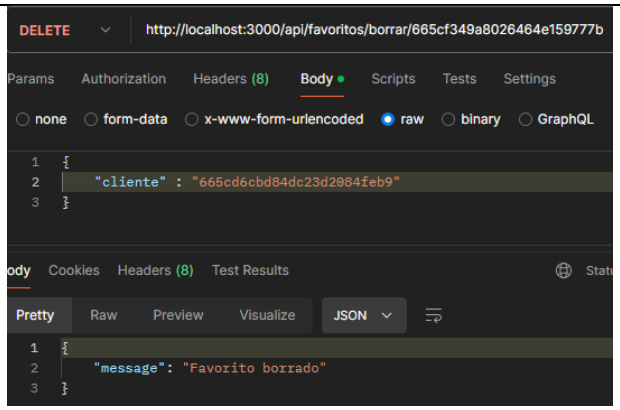



Ilustración 12: Enpoint para eliminar un producto de la lista de favoritos

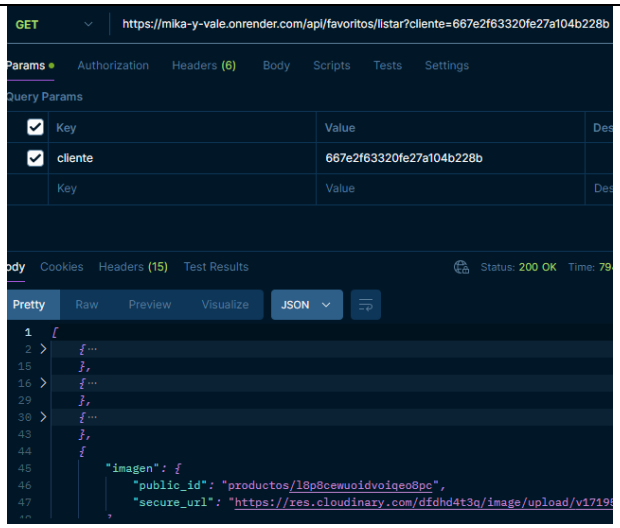


Ilustración 13: Enpoint para mostrar todos los productos de la lista de favoritos

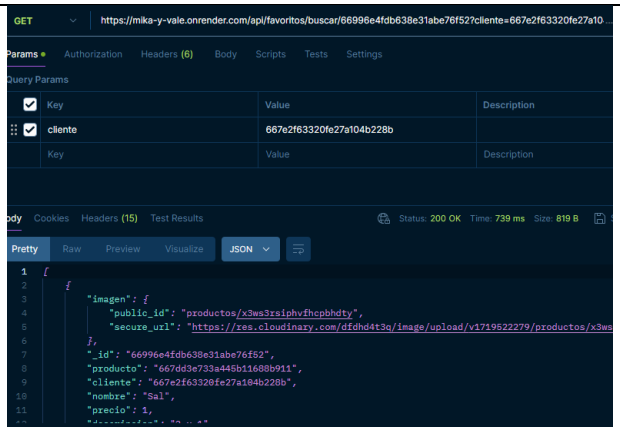


Ilustración 14: Enpoint para buscar un producto de la lista de favoritos

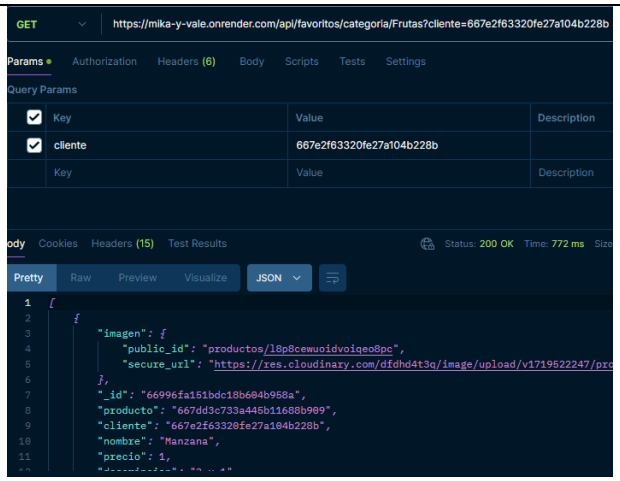
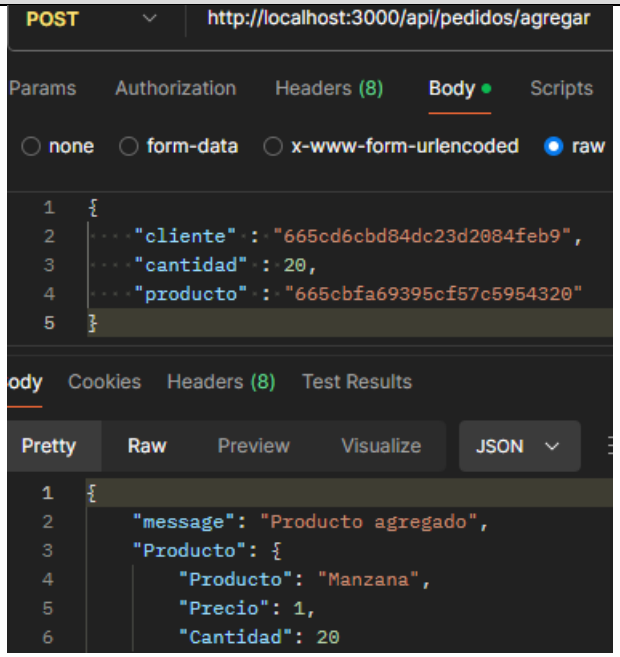
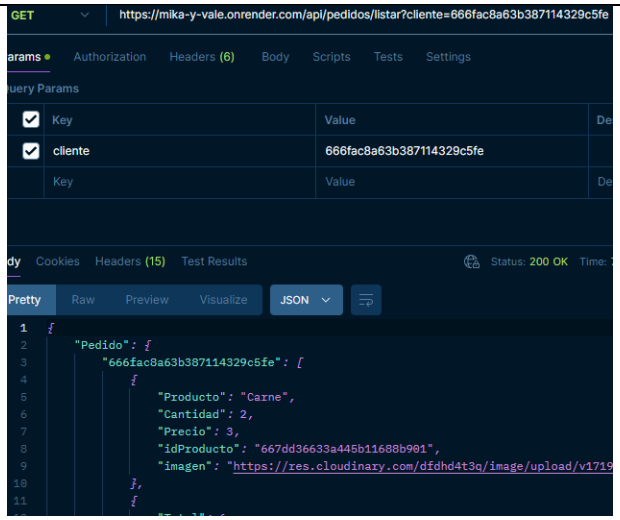


Ilustración 15: Enpoint para buscar productos favoritos por categoría

Sprint 3

Tabla 18: Evidencia CRUD carrito

Gestión del carrito
 **Ilustración 16:** Endpoint para agregar productos al carrito
 **Ilustración 17:** Endpoint para ver los productos del carrito

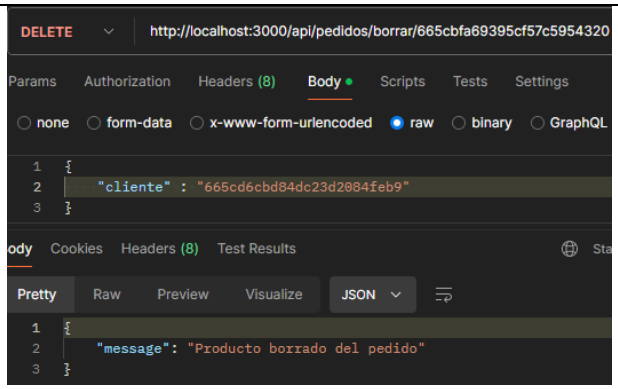


Ilustración 18: Endpoint para eliminar un producto del carrito

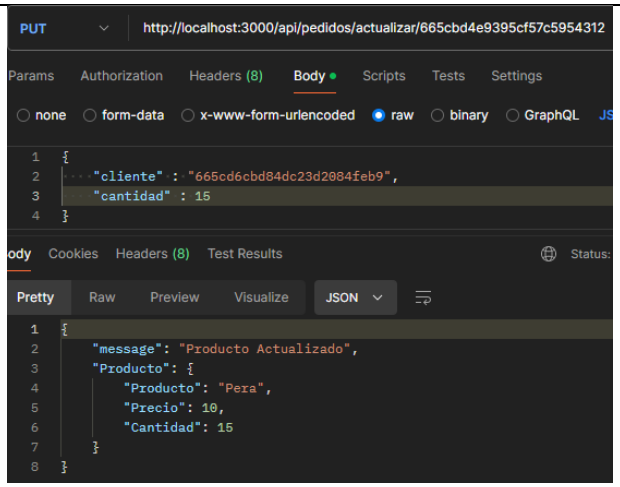


Ilustración 19: Endpoint para actualizar un producto

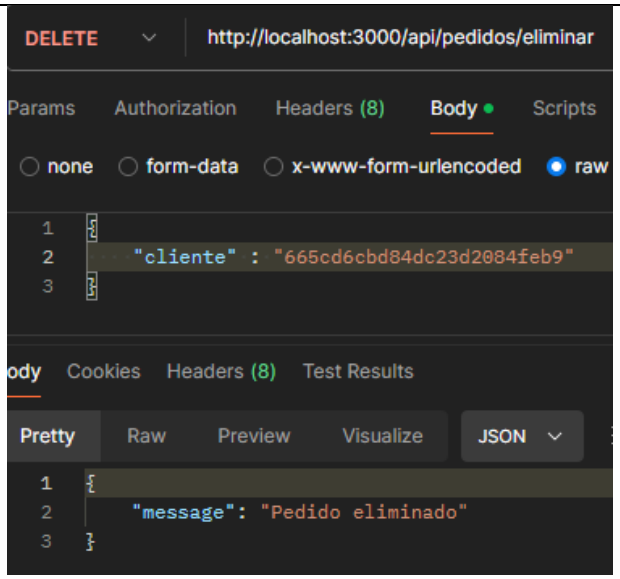


Ilustración 20: Endpoint para eliminar todos los productos del carrito

Tabla 19: Evidencia Pedidos

Gestión Pedidos

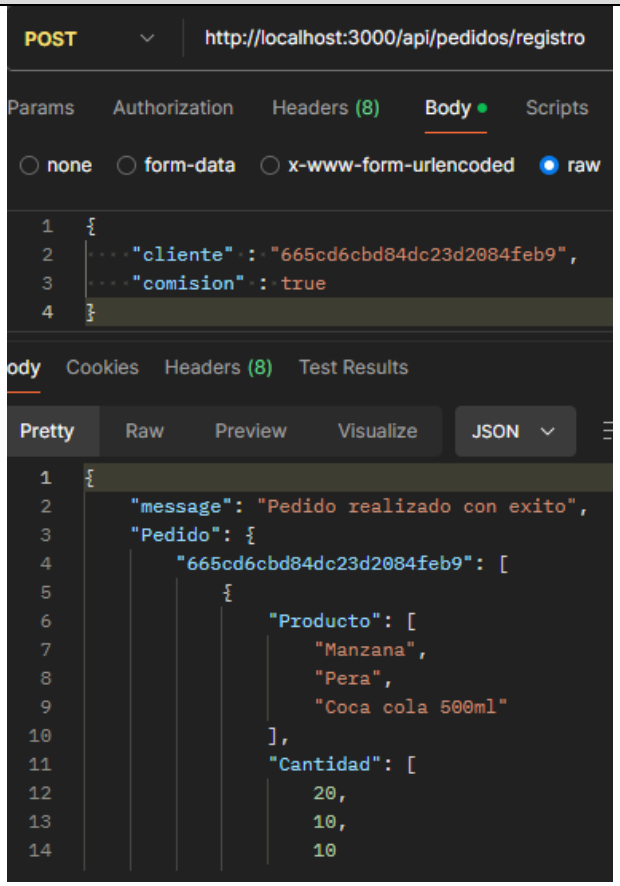



Ilustración 21: Endpoint para realizar el pedido

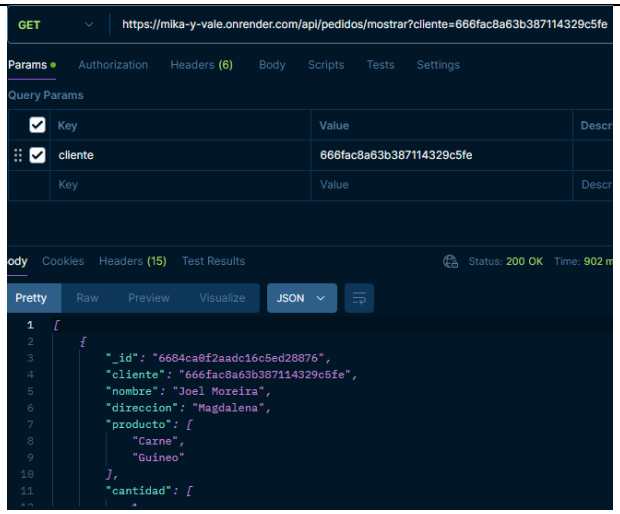



Ilustración 22: Endpoint para ver todos los pedidos realizados

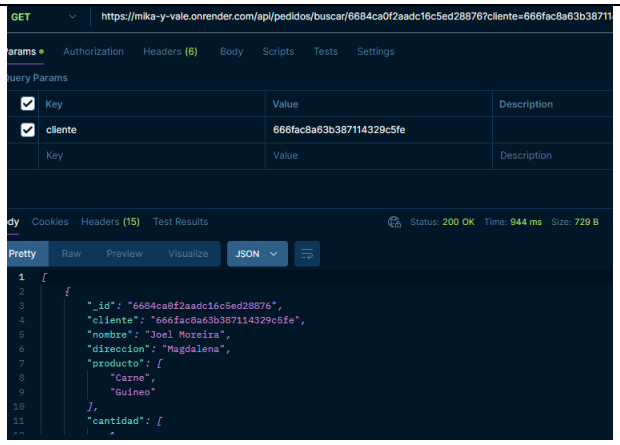


Ilustración 23: Endpoint para buscar un pedido

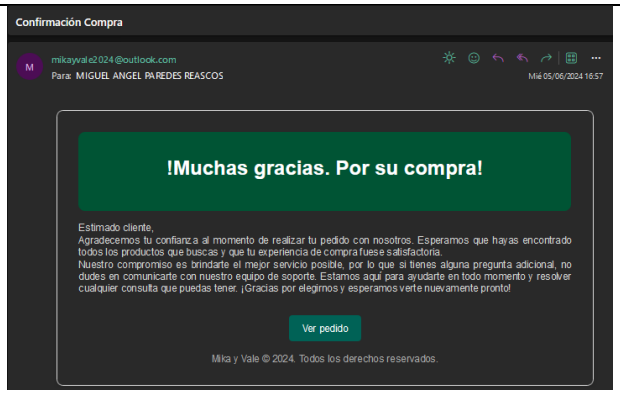


Ilustración 24: Correo de la confirmación del pedido

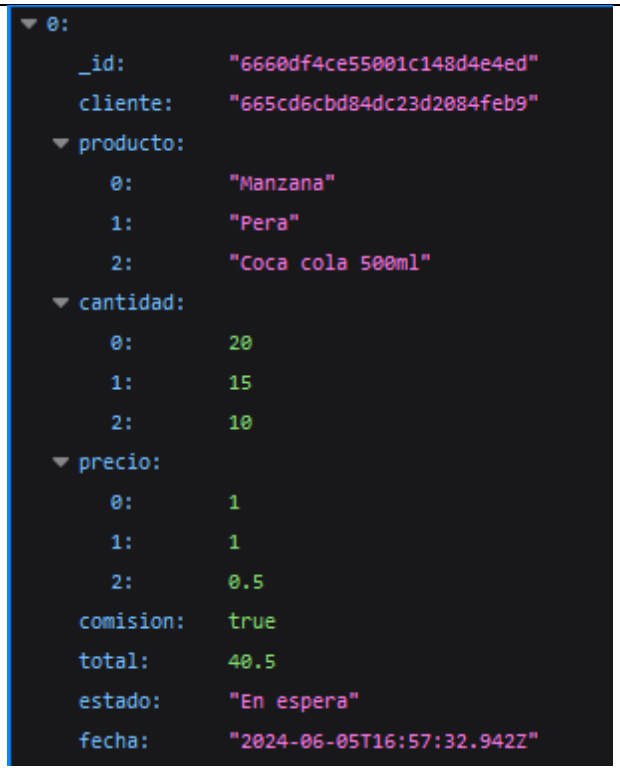
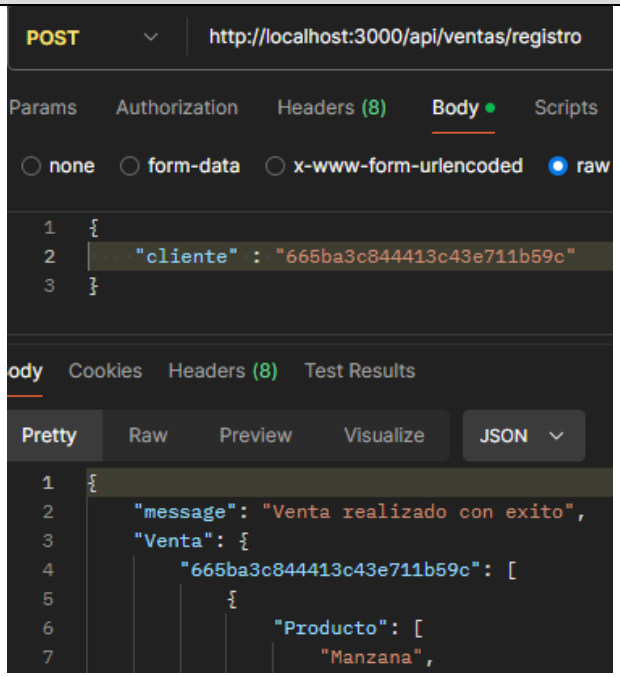
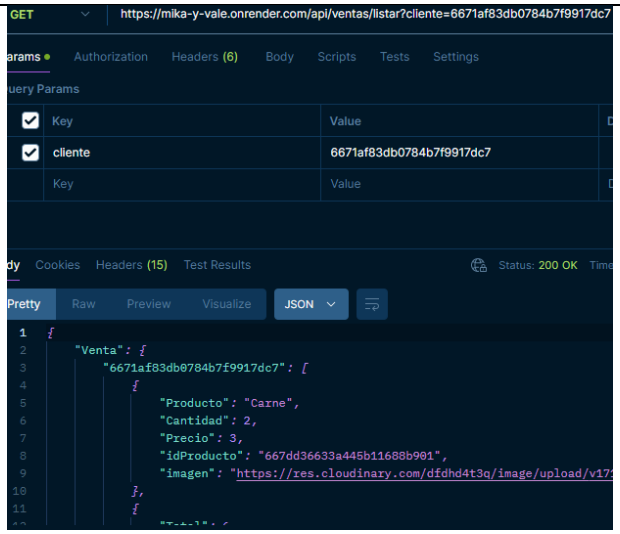


Ilustración 25: Endpoint para ver el pedido realizado

Tabla 20: Evidencia Ventas

Gestión Ventas
 **Ilustración 26:** Endpoint para realizar la venta
 **Ilustración 27:** Endpoint para ver todas las ventas realizadas

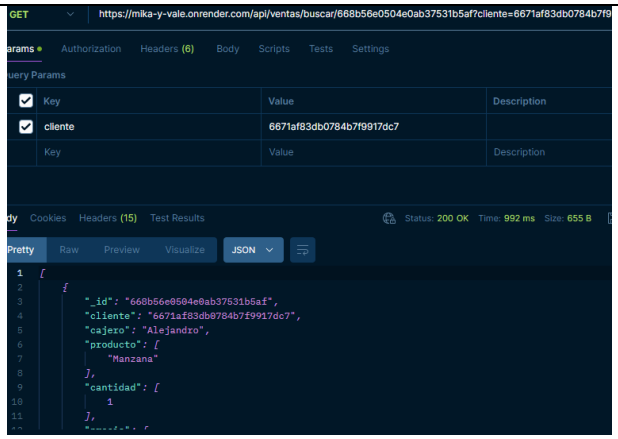


Ilustración 28: Endpoint para buscar una venta

Tabla 21: Evidencia estado de pedidos

Gestión estado de pedidos

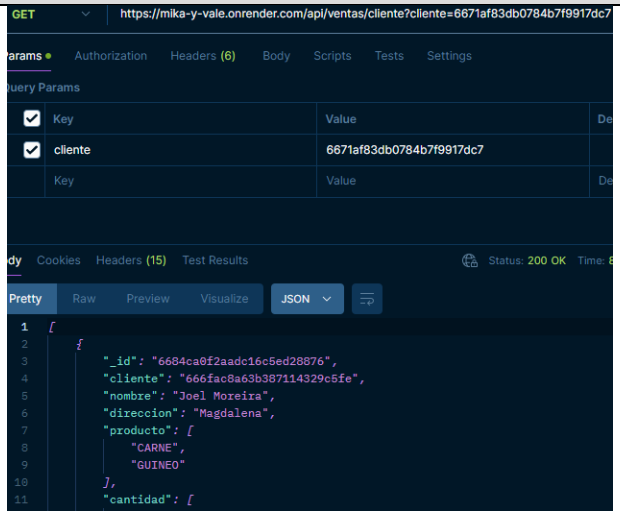



Ilustración 29: Endpoint para ver todos los pedidos

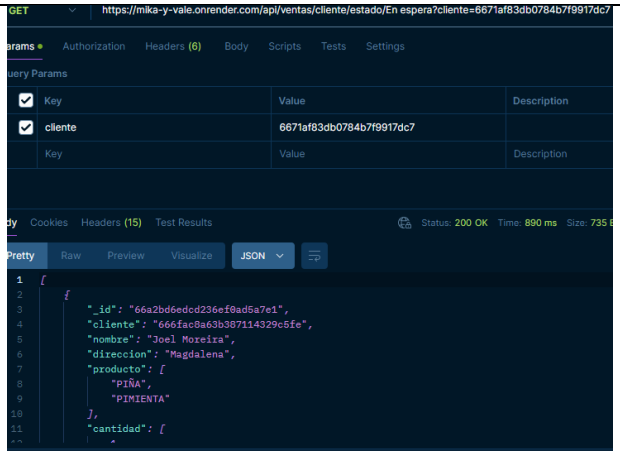



Ilustración 30: Endpoint para ver todos los pedidos con el mismo estado

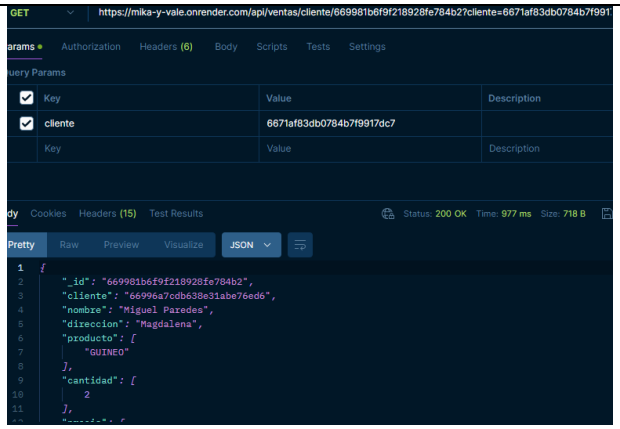


Ilustración 31: Endpoint para ver un pedido en específico

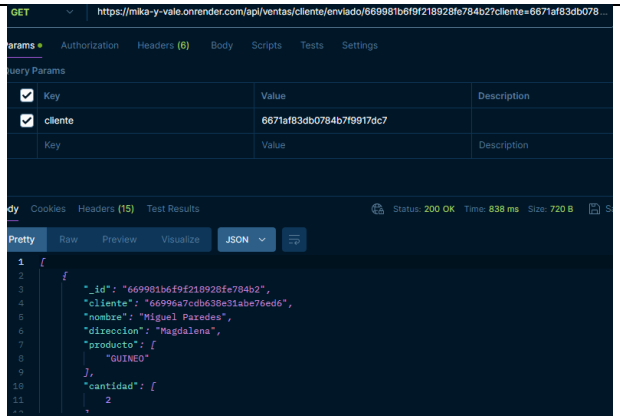


Ilustración 32: Endpoint para cambiar el estado del pedido a Enviado

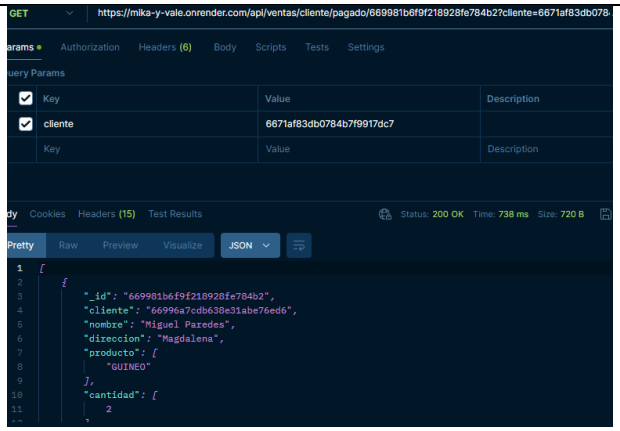


Ilustración 33: Endpoint para cambiar el estado del pedido a Pagado

Tabla 22: Evidencia reporte de pedidos y ventas

Gestión estado de pedidos

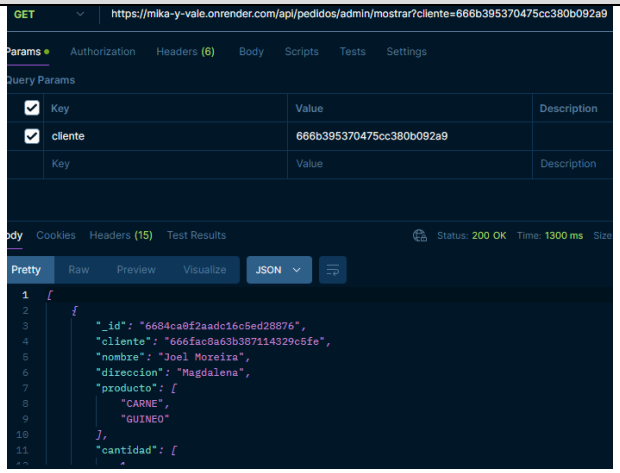



Ilustración 34: Endpoint para ver todos los pedidos

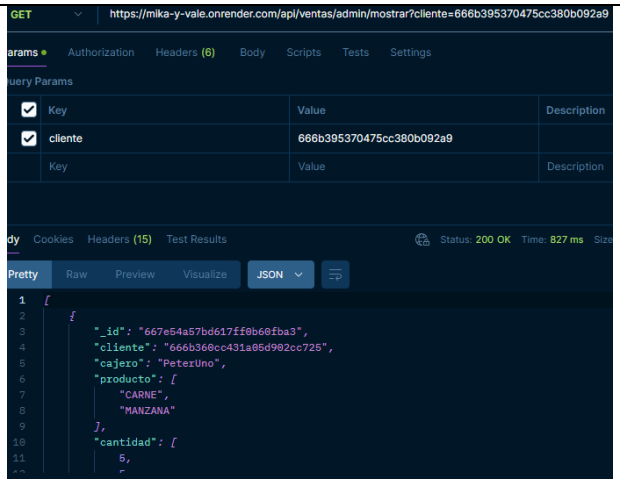



Ilustración 35: Endpoint para ver todas las ventas

Sprint 4

Tabla 23: Evidencia Pruebas Unitarias

Pruebas Categorías

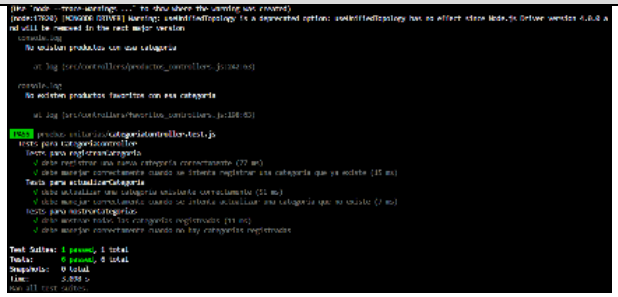



Ilustración 36: Resultado prueba unitaria Categorías

Pruebas Favoritos

Ilustración 37: Resultado prueba unitaria Favoritos

Pruebas Registro

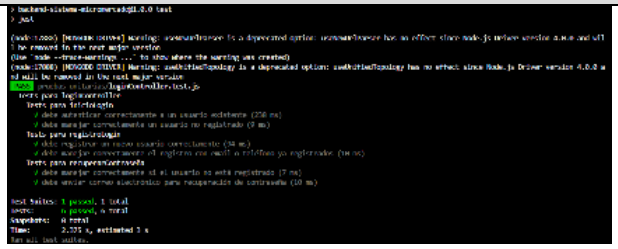


Ilustración 38: Resultado prueba unitaria Registro

Pruebas Pedidos

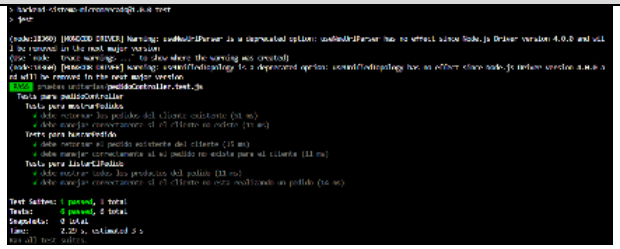


Ilustración 39: Resultado prueba unitaria Pedidos

Pruebas Productos

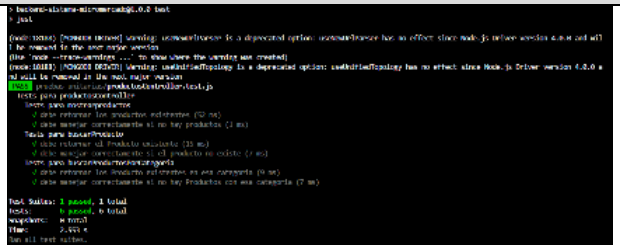


Ilustración 40: Resultado prueba unitaria Productos

Pruebas Ventas

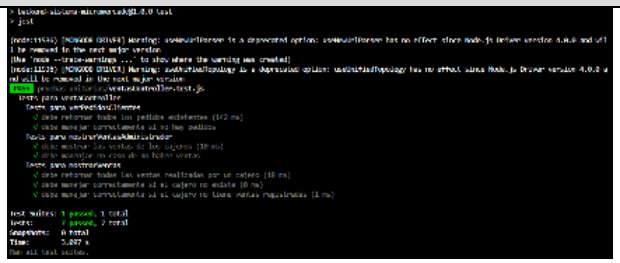
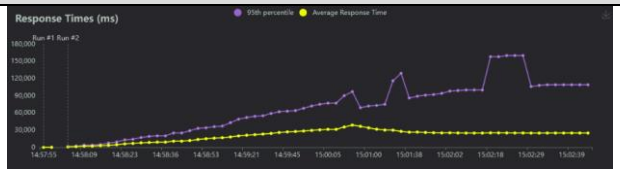
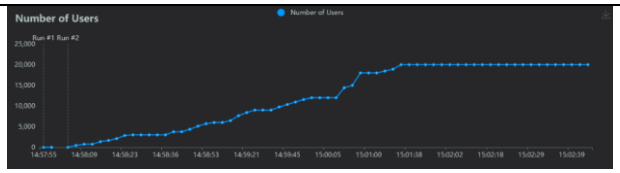


Ilustración 41: Resultado prueba unitaria Ventas

Tabla 24: Evidencia Pruebas de Estrés

Pruebas de Estrés
 **Ilustración 42:** Tiempos de respuesta con 20 mil usuarios
 **Ilustración 43:** Número de Usuarios con 20 mil usuarios

ANEXO III

A continuación, se adjunta la URL de un vídeo explicativo sobre el manejo de las funcionalidades que se proporciona en este componente, desde el usuario Cliente hasta el usuario Administrador de manera que sea sencilla la accesibilidad a la página web:

https://youtu.be/q__iGjz24fE

ANEXO IV

El manual de instalación de la API y su funcionamiento se encuentra detallado en el README del repositorio de Github, donde se encuentra alojado el proyecto.

https://github.com/Miguel-Paredes/backend-Servicio-Web

El enlace del despliegue de la API se adjunta a continuación. Para su utilización se deberá agregar al final los Endpoints de esta manera: */api/endpoint/*

https://mika-y-vale.onrender.com

Tabla 0.1: Credenciales

ACCESOS	
USUARIOS	**CREDENCIALES**
Administrador	mikayvale2024@outlook.com Minimarket2024
Cliente	miguelparedes2023@outlook.com Tesis-24
Cajero	alejo321@gmail.com Cajero-24

Para acceder a los repositorios de imágenes y la base de datos en producción, contactar con el desarrollador para dar los permisos correspondientes.