

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SISTEMA PARA LA GESTIÓN DE PLAZAS DE ESTACIONAMIENTO PARA EL PARQUEADERO DE LA ESFOT

DESARROLLO DE UN BACKEND

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

SCARLETT STEFANY LUNA CAJAS

DIRECTOR: BYRON GUSTAVO LOARTE CAJAMARCA

DMQ, enero 2025

CERTIFICACIONES

Yo, **SCARLETT STEFANY LUNA CAJAS** 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.

SCARLETT STEFANY LUNA CAJAS

Scarlett.luna@epn.edu.ec

Stefy506lc@outlook.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **SCARLETT STEFANY LUNA CAJAS**, bajo mi supervisión.

Ing. BYRON LOARTE, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

SCARLETT STEFANY LUNA CAJAS

DEDICATORIA

Dedico este trabajo a Dios, porque en todo momento he visto su ayuda, guía y protección, a toda mi familia, principalmente a mi madre, agradezco sus esfuerzos por sacarme adelante, este trabajo es un reconocimiento a su dedicación, también a mis profesores por enseñarme y compartir sus conocimientos, a mis amigos porque su valiosa amistad, además de todo el apoyo que me dieron, a todos ustedes porque fueron el motor que me dio las fuerzas para llegar hasta aquí.

SCARLETT STEFANY LUNA CAJAS

AGRADECIMIENTO

Principalmente agradezco a Dios, porque mediante su guía hoy estoy aquí, por permitirme cada día disfrutar de esta faceta junto a mis seres amados, a mi familia, mi madre sobre todo que me apoyó, se alegró conmigo y luchó conmigo, a mis amigos, siempre me animaron, compartieron conmigo mis avances y el estar pendientes no solo de mi progreso sino también de mi fue fundamental, pero en especial a los que hice en todo este bonito trayecto como David alguien que me apoyo, también con quien aprendí a no rendirme y a su compañía, el apoyo de todos, me fortalecieron para que este sueño se haga realidad.

A mi tutor el Ing. Byron Loarte, por su profesionalismo, su dedicación, su paciencia, sus consejos, sus motivaciones, cada palabra y corrección me ayudaron a que este sueño se plasme más claramente. Y a cada uno de los profesores por su esfuerzo y dedicación presentes en cada enseñanza.

SCARLETT STEFANY LUNA CAJAS

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
RESUMEN	VI
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
2 METODOLOGÍA	7
2.1 Metodología de Desarrollo	7
Roles	8
Artefactos	9
2.2 Diseño de la arquitectura	12
2.3 Herramientas de desarrollo	13
3 RESULTADOS	16
Sprint 0 Configuración del ambiente de desarrollo.	16
Sprint 1 - Codificación de endpoints para el usuario administrador, guardia y externos.	20
Sprint 2 - Codificación de endpoints para el Guardia.	25
Sprint 3 - Codificación de endpoints para el usuario externo.	28
4 CONCLUSIONES	34
5 RECOMENDACIONES	35
6 REFERENCIAS BIBLIOGRÁFICAS	36
ANEXOS	42
ANEXO I	43
ANEXO II	44
ANEXO III	73

RESUMEN

En la Escuela de Formación de Tecnólogos (ESFOT), no solo hay espacios destinados para el estudio y la recreación, sino también espacios de estacionamiento para que el personal administrativo, docentes y estudiantes puedan hacer uso de los mismos cuando lo requieran. Sin embargo, por el uso que se le da a esta área de la institución los controles que se realizan por parte de los guardias son mínimos ya que personas externas y con motivos malintencionados acceden a los espacios disponibles lo que ocasiona congestionamiento, pérdida de tiempo y que los usuarios tengan que buscar disponibilidad en otros parqueaderos de la Escuela Politécnica Nacional incrementando así, el riesgo de incidentes como robos y preocupación constante.

Para abordar los desafíos antes mencionados y ayudar a toda la comunidad de la ESFOT, en este proyecto de Integración Curricular se ha desarrollado un backend el cual permite gestionar de forma eficiente y en tiempo real la ocupación y disponibilidad de cada plaza de estacionamiento que posee el parqueadero, logrando de esta manera, brindar una herramienta tecnológica que permita reducir el tiempo de búsqueda de estacionamientos y sobre todo fortalecer la seguridad en esta área de la ESFOT.

El presente documento se encuentra organizado de la siguiente manera, la primera sección contiene el análisis de la problemática dando como resultado los objetivos, el alcance y el respectivo marco teórico. En la siguiente sección, se describe la metodología Scrum en conjunto con el diseño de interfaces, patrón de arquitectura y las herramientas para el desarrollo del backend. En la tercera sección, se presenta el resultado de cada tarea que se ha especificado en el Sprint Backlog en conjunto con las respectivas pruebas unitarias. Por último, en la última sección se proporcionan las conclusiones y recomendaciones que se han logrado tras finalizar con éxito el desarrollo e implementación del componente backend a producción.

Palabras clave: Backend, Parqueaderos, IoT, Express, Sensores.

ABSTRACT

At the School for the Training of Technologists (ESFOT), there are not only spaces for study and recreation, but also parking spaces for administrative staff, teachers and students to use when required. However, due to the use that is given to this area of the institution, the controls carried out by the guards are minimal, since external people with malicious motives access the available spaces, which causes congestion, loss of time and users have to look for availability in other parking lots of the National Polytechnic School, thus increasing the risk of incidents such as theft and constant concern.

To address the aforementioned challenges and help the entire ESFOT community, this Curricular Integration project has developed a backend to efficiently manage in real time the occupancy and availability of each parking space in the parking lot, thus providing a technological tool to reduce the time spent searching for parking spaces and, above all, to strengthen security in this area of the ESFOT.

This document is organized as follows, the first section contains the analysis of the problem resulting in the objectives, scope and the respective theoretical framework. In the next section, the Scrum methodology is described together with the interface design, architecture pattern and the tools for backend development. In the third section, the result of each task that has been specified in the Sprint Backlog is presented together with the respective unit tests. Finally, the last section provides the conclusions and recommendations that have been achieved after the successful completion of the development and implementation of the backend component to production.

Keywords: Backend, Parking, IoT, Express, Sensors.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La Escuela de Formación de Tecnólogos (ESFOT) como parte de la Escuela Politécnica nacional (EPN) ha sido una institución clave para la formación tecnológica en distintas áreas técnicas, fortaleciendo así el tejido productivo del país a través de una formación integral que responda a las demandas del entorno profesional contemporáneo [1]. Por otra parte, la ESFOT dispone de varias instalaciones de uso múltiple y una de ellas es el parqueadero el cual tiene por objetivo garantizar la seguridad de los vehículos del personal administrativo, docentes, estudiantes y agentes externos que realizan sus labores diarias dentro de la misma. Sin embargo, por el uso que se le da a esta área de la institución los controles que se realizan por parte de los guardias son mínimos ya que personas externas y con motivos malintencionados acceden a los espacios disponibles lo que ocasiona congestionamiento, pérdida de tiempo y que los usuarios tengan que buscar disponibilidad en otras facultades de la EPN incrementando así, el riesgo de incidentes como robos y preocupación constante.

En la actualidad, la implementación de sistemas orientados a la administración de usuarios y la asignación de espacios de estacionamiento se ha convertido en una estrategia que ofrece diversos beneficios como, por ejemplo, se destaca la optimización del uso de las áreas de parqueo, así como la regulación del acceso de los usuarios. Además, estos sistemas facilitan la experiencia de los conductores al minimizar inconvenientes a la hora de aparcar sus vehículos ya que brindan información instantánea sobre los espacios disponibles y su flexibilidad de adaptarse a distintas edificaciones estructurales en los que se implemente [2].

En base a los beneficios que otorga los sistemas de aparcamiento y la problemática que presenta actualmente el estacionamiento de la ESFOT, se ha desarrollado un backend que permita gestionar la ocupación y disponibilidad de cada plaza de estacionamiento que posee el parqueadero, logrando de esta manera brindar una herramienta tecnológica que permita reducir el tiempo de búsqueda de estacionamientos y sobre todo fortalecer la seguridad en esta área de la ESFOT. Además, este componente backend utiliza una solución basada en IoT para el procesamiento de la información en tiempo real, una serie de roles y una base de datos no relacional para un mejor control de la misma.

1.1 Objetivo general

Desarrollar un backend para la gestión de plazas de estacionamiento y del parqueadero de la ESFOT.

1.2 Objetivos específicos

- Recopilar los requisitos para el desarrollo del backend y módulos asociados.
- Establecer un modelo de datos (NoSQL) para la gestión de la información.
- Codificar cada uno de los endpoints asociados a módulos y roles.
- Ejecutar pruebas para corroborar el funcionamiento adecuado de todo el backend.
- Llevar al componente backend en un entorno de producción para su uso.

1.3 Alcance

El alcance en todo proyecto es de suma importancia pues ayuda a definir claramente los objetivos, recursos, tiempos de entrega y otros aspectos que permitan completar el proyecto de manera exitosa. Además, el definir de manera clara el alcance contribuye a una gestión eficiente de cualquier imprevisto que pueda surgir a lo largo de todo el proyecto [3].

Con respecto a este componente backend, el alcance se centra en la gestión de entrada, salida y estancia de vehículos dentro del parqueadero de la ESFOT utilizando para ello una serie de endpoints (públicos y privados), una solución basada en la implementación de Arduino UNO, sensores ultrasónicos y una serie de roles para la confidencialidad y el procesamiento de la información en tiempo real. Además, para desarrollar este componente se hace uso de la metodología Scrum, la cual permite trabajar en equipo y realizar el seguimiento de resultados en base a iteraciones, por otra parte, se hace uso también de un diseño arquitectónico Modelo – Vista – Controlador (MVC) que permita estructurar y organizar todo el proyecto. En cuanto a la etapa de codificación se desarrolla una serie de endpoints por cada módulo acompañado de una serie de pruebas para corroborar el buen

funcionamiento de todo el backend y una etapa final para el despliegue a un entorno de producción. A continuación, se presentan los roles y sus acciones a realizar.

Roles que se han establecido para el proyecto:

- Administrador.
- Guardia.
- Usuario externo.

Endpoints para que el administrador pueda realizar.

- Inicio de sesión.
- Recuperación de contraseña.
- Gestionar guardias.
- Gestionar usuarios externos (personal docente y administrativos).
- Gestionar plaza de estacionamientos.

Endpoints para que el guardia pueda realizar.

- Inicio de sesión.
- Recuperación de contraseña.
- Gestionar perfil.
- Visualizar status de plaza de estacionamientos.
- Gestionar usuarios externos (estudiantes e invitados).

Endpoints para que el usuario externo pueda realizar.

- Inicio de sesión.
- Recuperación de contraseña.
- Gestionar perfil.
- Visualizar status de plaza de estacionamientos.

1.4 Marco Teórico

Backend

En la actualidad conocido como “lado del servidor” es la lógica interna de un sistema software que opera en segundo plano, por lo que no es visible para el usuario final

pero que es esencial para el buen funcionamiento. Además, el backend se centra principalmente en gestionar la lógica empresarial, la manipulación de los datos, la seguridad, optimizaciones y otras tareas que son importantes para el buen funcionamiento de un sistema software completo [4].

Node.js

Es un entorno de ejecución basado en código abierto y multiplataforma que permite la ejecución de código JavaScript. Además, este entorno permite desarrollar y testear aplicaciones que se ejecuten en segundo plano y otras donde se tenga interacción con aplicaciones frontend usando un mismo lenguaje lo que ahorra tiempo y carga de trabajo [5].

JavaScript

Su característica principal es que es un lenguaje de programación ampliamente utilizado para la codificación de aplicaciones de lado del cliente y servidor. Además, por su vasta versatilidad que posee y la gama de funcionalidades ha sido adoptado en diversas librerías y Frameworks modernos [6].

MongoDB Compass

Esta herramienta gráfica facilita la interacción con bases de datos MongoDB, es intuitiva al momento de explorarla y al gestionar los datos almacenados, permitiendo que los usuarios puedan crear, editar o eliminar datos sin necesidad de usar líneas de comandos, esto también permite que los usuarios añadan información de manera sencilla [7] [8].

API RESTful

Es una interfaz que se apoya en los principios de la arquitectura REST, además, permite que la comunicación entre sistemas sea mediante el protocolo HTTP por lo que es ampliamente utilizada en todo tipo de aplicaciones software [9].

JSON Web Token (JWT)

Es un método compacto que permite compartir información de forma segura, además permite que esta sea verificada y confiable, principalmente es usado en la autenticación, asegurando que solo los usuarios con el token adecuado puedan

ingresar y acceder a los recursos protegidos o confidenciales, por otra parte, permite un intercambio de información sustentada en la integridad del contenido y comprobando la identidad del remitente [10].

SWAGGER

Es una herramienta muy útil pues permite diseñar, documentar y consumir APIs RESTful mucho más eficiente. Además, permite a los programadores comprender el código, visualizar los endpoints y las funcionalidades de los mismos por lo que ahorra tiempo y recursos [11].

Thunder Client

Es una extensión para Visual Studio Code la cual actúa como un cliente REST, por lo que en la actualidad se ha convertido en una herramienta muy beneficiosa pues permite realizar pruebas de endpoints dentro de un entorno de desarrollo de una forma fácil, así como realizar peticiones GET, POST, PUT, DELETE y otras características más [12].

HTTP

Es un protocolo de comunicación en la World Wide Web, esto permite que los clientes como los navegadores web puedan realizar solicitudes a servidores y obtener recursos de una forma fácil e integra. Además, su funcionamiento es fácil ya que trabaja bajo la arquitectura cliente-servidor [13].

Endpoint

Se refiere al punto final de una URL y el cual tiene por objetivo acceder a un recurso de un sistema software. Además, estos puntos finales trabajan con operaciones bien definidas como la creación, lectura, actualización o eliminación [14].

REST

Es un estilo de arquitectura que se enfoca principalmente en el diseño y la creación de APIs de tipo RESTful. Además, posee muchas ventajas con respecto a la escalabilidad, flexibilidad, reutilización y mantenibilidad por lo que es ampliamente utilizado en la actualidad y sobre todo para el desarrollo web moderno [15].

JSON

Es un formato de texto ligero que se usa comúnmente en proyectos de desarrollo de software para el intercambio de información. Además, su sintaxis es simple y legible por lo que es 100% compatible con diferentes lenguajes de programación, Frameworks y librerías [16].

Render

Es una plataforma que otorga una gama amplia de servicios y el despliegue de aplicaciones backend que han sido desarrolladas bajo Node.js. Además, su eficiencia lo hace una opción llamativa para los desarrolladores ya que permite maximizar la productividad y el despliegue continuo [17].

2 METODOLOGÍA

El estudio de casos al ser un método de investigación implica realizar un análisis detallado y exhaustivo de uno o varios temas el cual tiene como por objetivo comprender y explicar un fenómeno ya se de forma cualitativa o cuantitativa por lo que es ampliamente utilizado en varias ramas científicas. Además, para ejecutar un estudio de casos se necesita recopilar datos mediante diferentes técnicas y posterior a ello realizar un análisis de los mismos para desarrollar un plan de solución [18].

Fundamentado en lo anterior el presente Trabajo de Integración Curricular hace uso de un estudio de casos ya que el mismo ha permitido de forma clara y directa comprender a fondo la problemática y con ello direccionar de manera correcta el plan de trabajo, para plantear de esta manera una solución viable en base a una recolección de requerimientos y objetivos específicos claros que permitan cumplir el objetivo general del proyecto.

2.1 Metodología de Desarrollo

Al hablar una metodología específicamente en el campo del desarrollo de software se refiere a un conjunto de buenas prácticas dirigidas a organizar, crear, elaborar y llevar a cabo programas informáticos de un manera eficiente, controlada y alineada con los requisitos del negocio. Además, al utilizar una metodología se tiene varios beneficios, por ejemplo, en la etapa de planificación se establece un cronograma para alcanzar los objetivos que se han planteado, por otra parte, se designa roles y responsabilidades lo que permite que todo el equipo trabaje de forma organizada y se logren las metas según el cronograma, pero sobre todo sin dejar de lado la calidad en lo que se realiza [19].

Una metodología ágil principalmente se fundamenta en el trabajo en equipos autoorganizados para el desarrollo de sistemas software. Además, su fin es realizar proyectos más rápidos, flexibles y sobre todo teniendo en cuenta la adaptación para lo cual el proyecto se lo divide en partes pequeñas denominadas iteraciones (Sprints), las mismas que se completan en un plazo aproximado de una a cuatro semanas. Sin duda, esta división permite realizar ajustes según los requisitos del

cliente sin que se afecte el proyecto total y sobre todo que exista una comunicación y retroalimentación constante con todo el equipo de trabajo [20].

Dentro de las metodologías ágiles se encuentra la metodología Scrum, la cual prioriza entregas parciales de software funcional y sobre todo flexibilidad al momento de adaptarse hacia las necesidades del negocio. Además, sus características más relevantes es la división del proyecto en Sprints (las cuales no deben durar más de un mes), los artefactos que son los requisitos que se encuentran estandarizados para una mejor comprensión, un conjunto de roles para un trabajo organizado y otras características más [21]. Teniendo en cuenta lo que otorga la metodología Scrum, en este Trabajo de Integración Curricular se hace uso de la misma para el desarrollo del componente backend ya que la adaptabilidad, flexibilidad y organización son pilares claves para cumplir el objetivo general del presente proyecto. A continuación, se detalla paso a paso como se ha implementado dicha metodología.

Roles

Como parte de las buenas prácticas de Scrum es la definición de roles que tienen que asumir con total responsabilidad sus actividades para poder lograr un producto de calidad [21]. Por esta razón, es importante comprender cada uno de los roles que son parte de esta metodología pues de ellos depende el poner en práctica el proceso ágil, a continuación, se presenta los roles que se han definido para el componente backend.

Product Owner

Su principal tarea es transmitir el propósito del producto a través de las necesidades del negocio. Además, ayuda a priorizar el Product Backlog para que las tareas de mayor importancia para el negocio sean las primeras en ser codificadas [21]. Por lo dicho anteriormente, la **Tabla 2.1** contiene información sobre los roles y su organización.

Scrum Master

Es el puente de comunicación entre el dueño del producto y el equipo de desarrollo. Además, su tarea principal radica en eliminar todos los obstáculos que se pueden

generar durante el proyecto, además de implementar buenas prácticas y organizar reuniones para revisar los avances de los Sprints [21]. Por lo dicho anteriormente, la **Tabla 2. 1** contiene información sobre la organización de los roles.

Development Team

Es el equipo de desarrollo en sí, quienes desarrollan y llevan a cabo las actividades que se han registrado en el Sprint Backlog. Además, su misión es llevar a cabo una organización que se encuentre enfocada en el cumplimiento de los Sprints y el funcionamiento progresivo del sistema software [21]. Por lo dicho anteriormente, la **Tabla 2. 1** contiene información sobre la organización de los roles.

Tabla 2. 1 Designación de roles.

ROL	INTEGRANTES
Product Owner	Ing. Byron Loarte, MSc.
Scrum Master	Ing. Byron Loarte, MSc.
Development Team	Srta. Scarlett Luna

Artefactos

Son documentos guías que de forma estandarizada permiten gestionar el trabajo del equipo, así como registrar las acciones y tareas a realizarse durante todo el proyecto. Además, con estos artefactos se puede verificar el progreso y tomar acciones de manera inmediata en el caso de que se requiera hacer modificaciones [22]. Aplicado al componente backend, a continuación, se detalla los artefactos que se han utilizado.

Recopilación de requerimientos

Es fundamental ya que este artefacto permite definir de manera corta y directa el propósito del proyecto registrando elementos como características, restricciones, requerimientos (funcionales y no funcionales) y los beneficiarios para el negocio [23]. Para presentar una porción de los requisitos del proyecto backend se lo puede ver en la **Tabla 2.2** y los requisitos adicionales se detallan en el **ANEXO II**.

Tabla 2. 2 Levantamiento de requerimientos para el componente backend.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
BACKEND	RR - 003	Para el administrador se requiere crear endpoints para: <ul style="list-style-type: none"> • Gestionar usuarios.
	RR - 006	Para el usuario guardia se requiere crear endpoints para: <ul style="list-style-type: none"> • Visualizar plazas de estacionamiento.
	RR - 007	Para el usuario se requiere crear endpoints para: <ul style="list-style-type: none"> • Actualizar campo “teléfono” en su perfil.

Historias de Usuario

Son una especie de tarjetas con una descripción breve y sencilla que detalla la funcionalidad que el software realiza y lo que el usuario requiere. Además, estas tarjetas poseen el siguiente formato tienen un perfil, la necesidad donde se describe lo que se desea resolver y el propósito que es lo que se desea alcanzar, entre otros campos más [24]. Para presentar una Historia de Usuario del proyecto backend se lo puede ver en la **Tabla 2.3** y las demás tablas adicionales se detallan en el **Anexo II**.

Tabla 2. 3 Historia de usuario para el usuario – Guardia.

HISTORIA DE USUARIO	
Identificador: HU-005	Usuario: Guardia
Nombre historia: Gestionar perfil	
Prioridad en negocio: Media	Riesgo en desarrollo: Media

Iteración asignada: 1
Responsable (s): Scarlett Luna
Descripción: El backend crea un endpoint para que el guardia pueda: <ul style="list-style-type: none"> • Visualizar su perfil. • Actualizar su información de perfil. • Actualizar su contraseña.
Observación: Para que el guardia pueda gestionar su perfil debe ingresar a con sus credenciales correctas.

Product Backlog

Es una lista a forma de resumen que mejora su comprensión y asegura que se enfoquen en las tareas son relevantemente importantes para el proyecto. Además, esta tabla contiene las tareas que realiza cada miembro del equipo, la iteración a la que pertenece, la prioridad para el negocio, entre otros campos más [25]. Para presentar una porción del Product Backlog del proyecto backend se lo puede ver en la **Tabla 2.4** y los requisitos adicionales se detallan en el **ANEXO II**.

Tabla 2. 4 Componente backend – Product Backlog.

PRODUCT BACKLOG				
ID-HU	DESCRIPCIÓN DE LA HISTORIA	ITERACION ASIGNADA	ESTADO	PRIORIDAD
HU-002	Gestionar guardias.	2	Finalizado	Alta
HU-003	Gestionar usuarios externos (personal administrativo y docente).	3	Finalizado	Media

Sprint Backlog

Es una tabla con mucho más detalle del Product Backlog, la cual se organiza en base a iteraciones. Además, esta tabla se divide en cuatro partes que son el objetivo del Sprint, los elementos que son las Historias de usuario, las tareas que posee cada iteración y por último el tiempo de ejecución [26]. Para presentar una porción del Sprint Backlog del proyecto backend se lo puede ver en la **Tabla 2.5** y las iteraciones restantes se detallan en el **ANEXO II**.

Tabla 2. 5 Sprint Backlog – Componente backend

SPRINT BACKLOG						
ID-SB	TAREA	MÓDULO	ID-HU	DESCRIPCIÓN DE LA HISTORIA	DETALLE DE LA TAREA	TIEMPO ESTIMADO (HORAS)
SB-000	Configuración del ambiente de desarrollo	<ul style="list-style-type: none">Instalación y actualización de las herramientas de desarrolloConfiguración de los modelos de datos.				10 H

2.2 Diseño de la arquitectura

La arquitectura es la estructura y organización de componentes que permiten que un software funcione adecuadamente ya sea en lado del servidor o cliente ya que brinda una guía sobre como engrana cada parte. Además, en una arquitectura es esencial garantizar que sea escalable, que posea un alto rendimiento, una gestión de datos eficiente y sobre todo una interacción fluida entre todos los componentes. De esta manera, al tener un buen diseño de la arquitectura se tiene un desarrollo exitoso de la misma [27]. Aplicado al componente backend, a continuación, se detalla la arquitectura que se ha utilizado.

Arquitectura Modelo Vista Controlador (MVC)

Es un patrón de arquitectura se usa para estructurar sistemas software en tres partes, el modelo que maneja el almacenamiento de los datos, como segundo aspecto la vista que muestra al usuario toda la información mediante una interfaz gráfica y el controlador que es el puente entre la capa de vista y la capa del modelo. De esta manera, la ventaja de esta arquitectura es facilitar el mantenimiento y su escalabilidad [28]. En la **Figura 2. 1** se observa la representación de la arquitectura que se ha aplicado al componente backend.

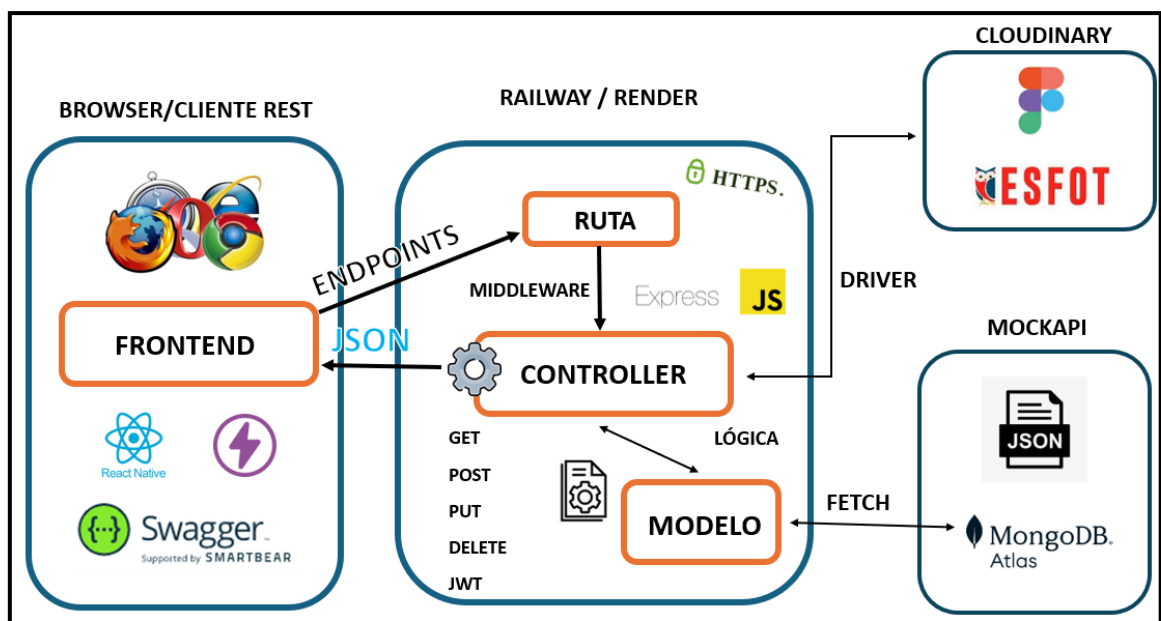


Figura 2. 1 MVC – Componente backend.

2.3 Herramientas de desarrollo

Son fundamentales para desarrollar proyectos de software ya que las mismas aportan beneficios durante la etapa de desarrollo. Además, estas herramientas optimizan la creación y gestión de paquetes, automatizan tareas, reducen el trabajo, minimizan errores, facilitan el trabajo en equipo y sobre todo ofrecen una visión clara del progreso del proyecto [29]. En la **Tabla 2.6** se listan las herramientas que se han elegido para el proyecto backend.

Tabla 2.6 Herramientas – Componente backend.

HERRAMIENTA	JUSTIFICACIÓN
GitHub	Es una plataforma que permite trabajar colaborativamente entre desarrolladores, lo que facilita revisar en tiempo real los avances del proyecto backend mediante repositorios y ramas [30].
Visual Studio Code	Es un editor ligero que soporta múltiples lenguajes de programación y que tiene como principal ventaja la depuración del código del proyecto backend [31].
Thunder Client	Es una extensión de visual studio que además de ser muy útil para realizar pruebas de API REST brinda una interfaz intuitiva y la opción de crear colecciones para organizar las pruebas que se realizan en el proyecto backend [32].
Express	Es un Framework que simplifica la creación de aplicaciones web, además, su característica más importante es el sistema de enrutamiento para las solicitudes HTTP siendo una gran ventaja al momento de realizar los endpoints del proyecto backend [33].
MongoDB	Es una base de datos NoSQL que permite manejar grandes volúmenes de información por lo que es flexible y escalable para el proyecto backend [34].
Render	Permite el despliegue de aplicaciones backend desarrolladas en Node.js lo que lo convierte en una opción llamativa para los desarrolladores ya que permite maximizar su productividad [17].

Librerías

Son colecciones de código que facilitan y agilizan la creación de proyectos de software. Además, entre sus principales características es la reutilización de código ya que esto evita la necesidad de escribir código desde cero, reducir tiempos y esfuerzo garantizando de esta manera un alto nivel de calidad y consistencia en el código [35]. En la **Tabla 2.7** se listan las librerías que se han elegido para el proyecto backend.

Tabla 2.7 Librerías – Componente backend.

LIBRERÍA	DESCRIPCIÓN
Bcryptjs	Está diseñada para almacenar las contraseñas mediante un cifrado [36].
Cors	Controla como se permiten las solicitudes de recursos entre diferentes dominios [37].
Nodemailer	Permite enviar correos electrónicos de manera sencilla mediante diversos métodos de transporte como SMTP [38].
JSON Web Token	Permite la transmisión segura de información [39].
Socket	Permite la transmisión de datos en tiempo real mediante una red de comunicación [40].

3 RESULTADOS

En esta sección, se detallan los logros que se han obtenido a lo largo del desarrollo del proyecto dirigido específicamente al componente backend. Además, estos logros muestran cada una de las actividades, resultados, pruebas y el despliegue de cada endpoint que posee cada usuario. Adicional a ello, cabe resaltar que cada actividad cuenta con unas tareas clave las cuales son esenciales para avanzar con la panificación y la consecución del objetivo general. A continuación, se describen los resultados de cada Sprint y sus tareas asociadas.

Sprint 0 Configuración del ambiente de desarrollo.

Para comenzar con el componente backend es importante configurar el ambiente de desarrollo y otras actividades con el objetivo de evitar contratiempos al momento de codificar cada uno de los módulos respectivos. En ese sentido, las tareas que abarca este Sprint son las siguientes.

- Delimitación de los requisitos a cumplir.
- Estructura general de las carpetas del proyecto.
- Diseño de las colecciones de datos para MongoDB.
- Asignación de roles para cada usuario.

Delimitación de los requisitos a cumplir

Codificación de endpoints para iniciar sesión y confirmación del correo para recuperar contraseña

El backend posee endpoints específicos y fundamentales para que cualquier usuario pueda ingresar al backend de forma segura y la posibilidad de recuperar la contraseña mediante un correo electrónico. Además, es importante señalar que cada usuario posee un rol específico y conforme a ello se relaciona las actividades que puede o no ejecutar dentro de los módulos del backend.

Codificación de endpoints para la gestión de usuarios guardia

El backend posee endpoints específicos y fundamentales para que el administrador pueda gestionar (Crear, Listar, Actualizar, Eliminar y Cambiar de estado) a los usuarios de tipo “guardia”. Además, el administrador tiene la posibilidad de realizar

todas las acciones por medio de verbos HTTP para una fácil administración de la información.

Codificación de endpoints para la gestión de usuarios externos

El backend posee endpoints específicos y fundamentales para que el administrador pueda gestionar (Crear, Listar y Eliminar) a los usuarios externos que se compone de personal docente y administrativo. Además, el administrador tiene la posibilidad de realizar todas las acciones por medio de verbos HTTP para una fácil administración de la información. Mientras que, la gestión de usuarios externos (estudiantes) lo realiza el usuario con rol guardia.

Codificación de endpoints para la gestión de plazas de estacionamiento

El backend posee endpoints específicos y fundamentales para que el administrador pueda gestionar información sobre las plazas de estacionamiento del parqueadero principal de la ESFOT. Además, el administrador tiene la posibilidad de realizar todas las acciones por medio de verbos HTTP para una fácil administración de la información.

Codificación de endpoints para visualizar perfil, gestión de usuarios externos y visualización de plazas de estacionamiento

El backend posee endpoints específicos y fundamentales para que el usuario guardia solo pueda visualizar sus datos personales como parte del primer módulo, un segundo módulo para que pueda gestionar la información de usuarios externos y un tercer módulo para visualizar y reservar el parqueadero o la plaza de estacionamiento para discapacitados. Además, el guardia tiene la posibilidad de realizar todas las acciones por medio de verbos HTTP para una fácil administración de la información.

Codificación de endpoints para la gestión del perfil y visualización de plazas de estacionamiento

El backend posee endpoints específicos y fundamentales para que el usuario externo (personal administrativo, docente, estudiantes e invitados) pueda gestionar la información de su perfil (visualizar su perfil y actualizar su número de contacto) y visualizar las plazas de estacionamiento disponible. Además, los usuarios externos

tienen la posibilidad de realizar todas las acciones por medio de verbos HTTP para una fácil administración de la información.

Estructura general de las carpetas del proyecto

Para la creación de todos los endpoints del componente backend, así como módulos asociados se ha utilizado Visual Studio Code en conjunto con el lenguaje de programación JavaScript, el patrón MVC y el Framework Express lo que hacen una combinación perfecta para el despliegue del servidor web y la gestión de cada una de las solicitudes por parte de los endpoints. A continuación, se observa en la **Figura 3.1** la estructura del proyecto.

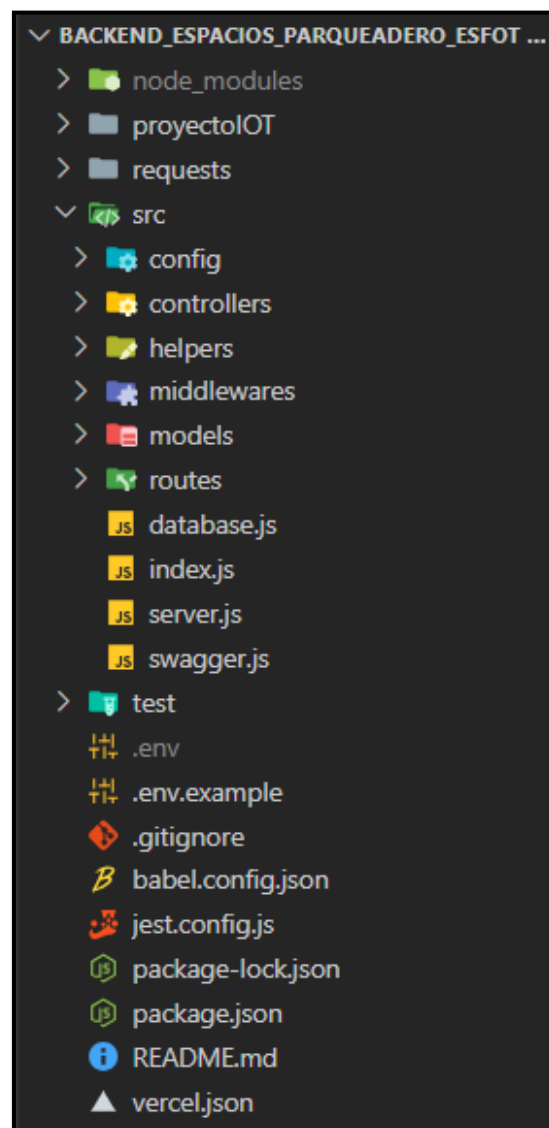


Figura 3.1 Estructura del proyecto.

Diseño de las colecciones de datos para MongoDB

Para la gestión de cada una de las colecciones y documentos para el backend “PoliParking” se ha utilizado MongoDB, el cual es un gestor de Base de datos no relacional que es de mucha utilidad al permitir trabajar con datos en tiempo real e ideal para manejar datos no estructurados permitiendo de esta manera una escalabilidad segura y flexible ante posibles cambios y nuevas implementaciones. En la **Figura 3.2** se observa el diseño de colecciones y para un detalle más completo de los documentos se puede ver en el **ANEXO II**.

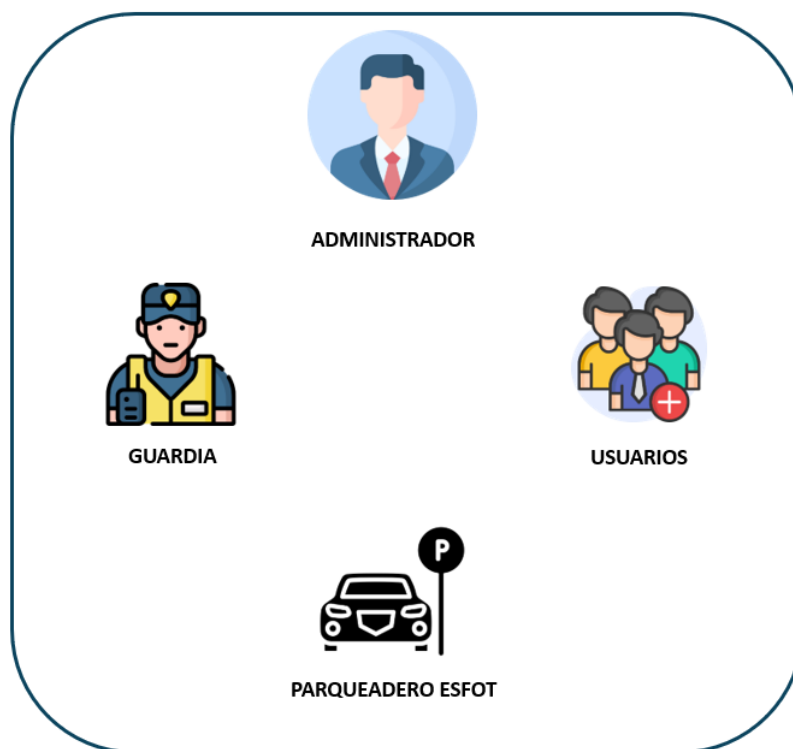


Figura 3.2 Diseño de las colecciones.

Asignación de roles para cada usuario

Para el componente backend se han establecido tres roles cada uno con permisos y funciones específicas, lo cual permite un control adecuado y protección a datos sensibles. Por lo antes dicho, las funciones de cada rol se presentan en la **Figura 3.3**.

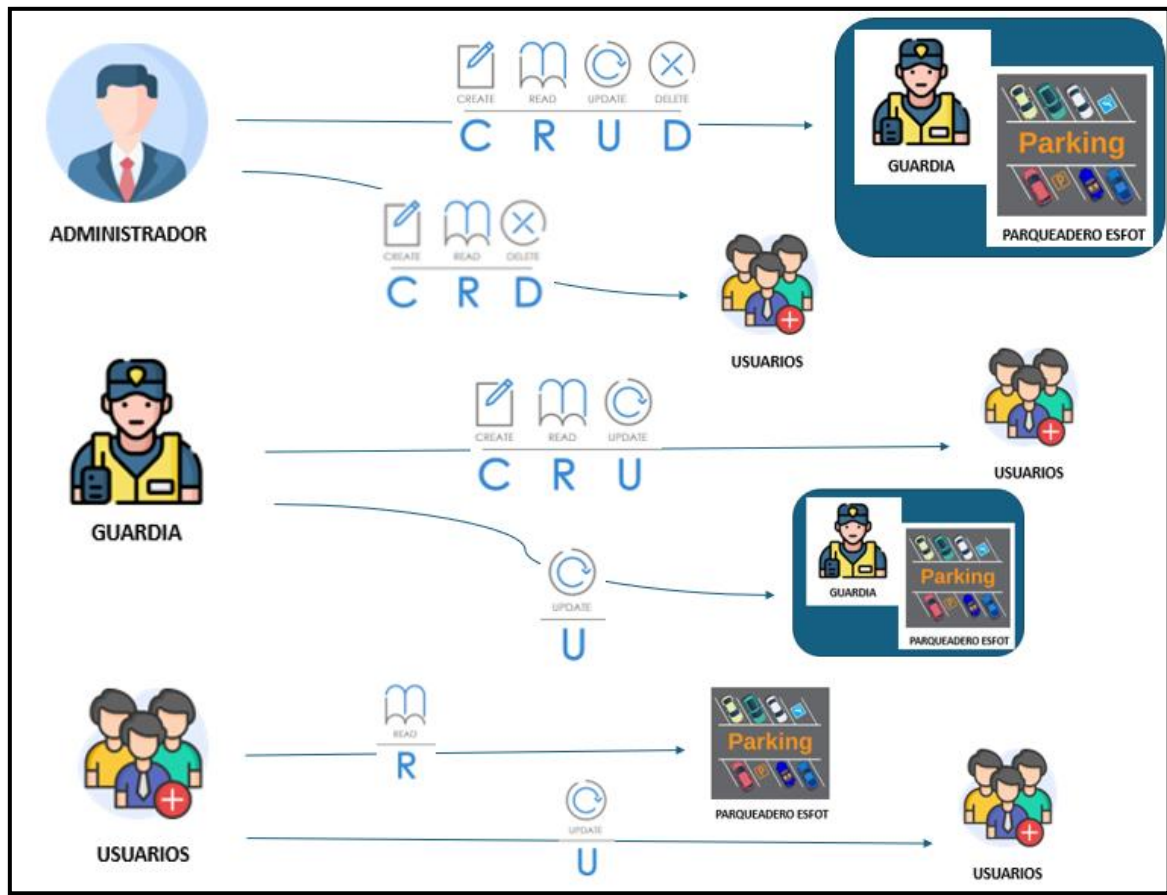


Figura 3.3 Funciones para cada rol.

Sprint 1 - Codificación de endpoints para el usuario administrador, guardia y externos.

Este Sprint se enfoca principalmente en tareas que comparten los tres usuarios como el inicio de sesión, recuperar contraseña y visualizar las plazas de estacionamiento disponible. En ese sentido, las tareas que abarca este Sprint son las siguientes.

- Codificación de endpoints para un correcto inicio de sesión según el rol de cada usuario.
- Codificación de endpoints para la recuperación de la clave mediante el correo electrónico según el rol de cada usuario.
- Codificación de endpoints para la gestión de usuarios externos (estudiantes e invitados).

- ## Codificación de endpoints para el correcto inicio de sesión según el rol de cada usuario

<h2>Responses</h2>	
<h3>Curl</h3> <pre>curl -X 'POST' \ 'https://backend-espacios-esfot.onrender.com/api/administrador/login' \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -d '{ "email": "sluna6360@gmail.com", "password": "Admin.1234" }'</pre>	
<h3>Request URL</h3> <p>https://backend-espacios-esfot.onrender.com/api/administrador/login</p>	
<h3>Server response</h3>	
Code	Details
200	<h4>Response body</h4> <pre>{ "id": "6779816f313e8b16daac2f75", "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImYyMzk4MTZmZWZlZTThMTZkYWVjLWMyY3NSIiwiaWF0IjEwNjYvbmCI6ImFkbWluaXN0cmFkb3IiLCJpcyYXQiOiJE3MzY0ODU3OTQsImV4cCI6MTGTczNju3MjE5NH0.joANRozFKVjlwmXC53N7272yEEFXQ7aMMq6dOUwi6k", "nombre": "Scarlett", "apellido": "Luna", "telefono": 998675191, "email": "sluna6360@email.com"</pre>

```
PASS test/administrador_controller.test.js (6.508 s)
  Pruebas Unitarias - Administrador y Gestión de Usuarios
    ✓ Login administrador exitoso (22 ms)
```

21

Codificación de endpoints para la recuperación de la clave mediante el correo electrónico según el rol de cada usuario

El backend otorga varios endpoints y rutas específicas para que el usuario administrador, guardia y usuarios externos puedan realizar la recuperación de su contraseña mediante el ingreso de un correo válido y posterior a ello generar una nueva contraseña con el enlace de recuperación que se envía al correo electrónico tal y como se lo observa en la **Figura 3.6** y en **Figura 3.7** se presenta el resultado de la prueba unitaria y en el **ANEXO III** para visualizar toda la funcionalidad completa.



Figura 3.6 Recuperación contraseña.

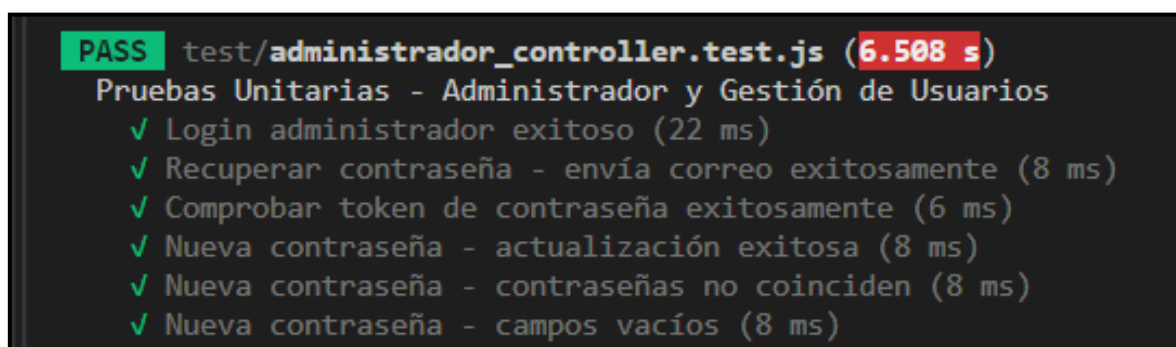


Figura 3.7 Prueba unitaria - recuperación contraseña.

Codificación de endpoints para la gestión de usuarios externos (estudiantes e invitados)

El backend otorga varios endpoints y rutas específicas para que el usuario administrador y guardia puedan gestionar a los usuarios externos tal y como se lo observa en la **Figura 3.8** y en **Figura 3.9** se presenta el resultado de la prueba unitaria y en el **ANEXO III** para visualizar toda la funcionalidad completa.

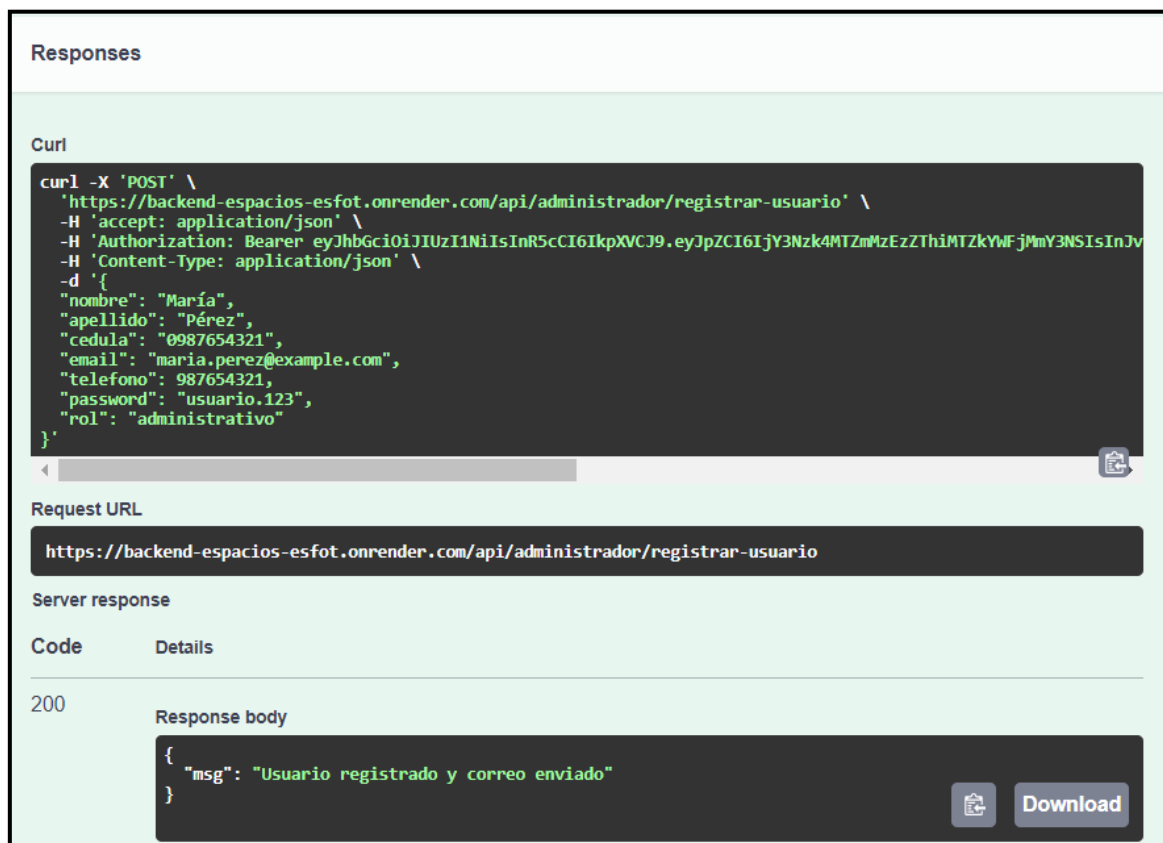


Figura 3.8 Registro de usuario externo.

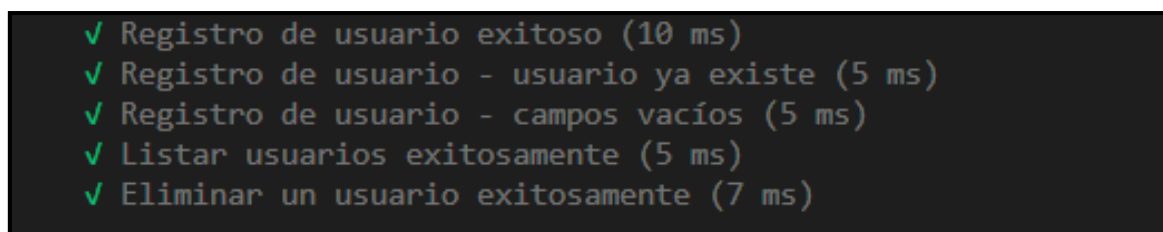
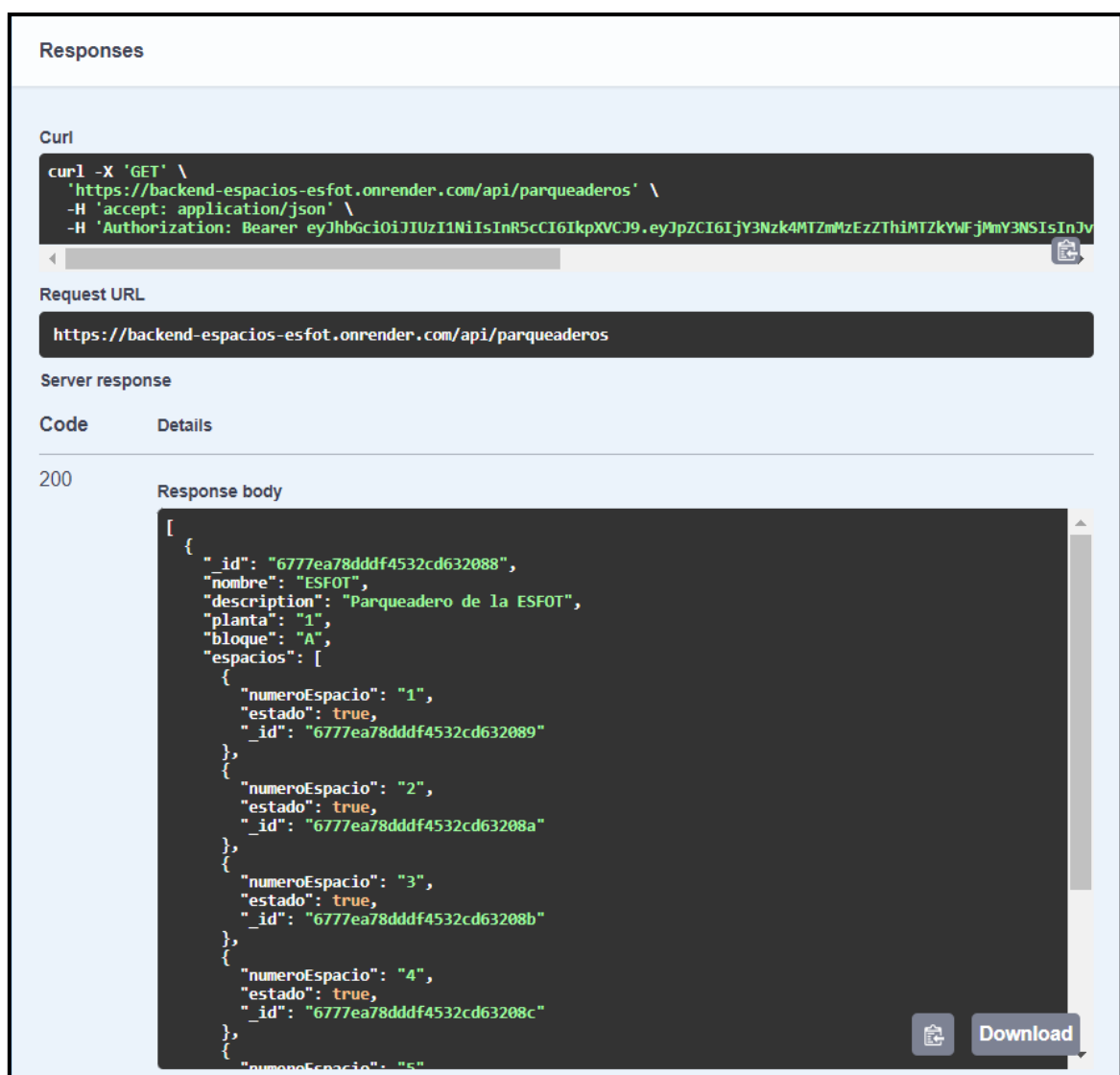


Figura 3.9 Prueba unitaria – Registro usuario externo.

Codificación de endpoints para la visualización de status de plazas de estacionamiento disponible

El backend otorga varios endpoints y rutas específicas para que el usuario administrador, guardia y usuarios externos puedan visualizar las plazas de estacionamiento disponible de una forma gráfica para una mejor presentación de la información tal y como se lo observa en la **Figura 3.10** y en **Figura 3.11** se presenta el resultado de la prueba unitaria y en el **ANEXO III** para visualizar toda la funcionalidad completa.



The screenshot displays a REST client interface with the following sections:

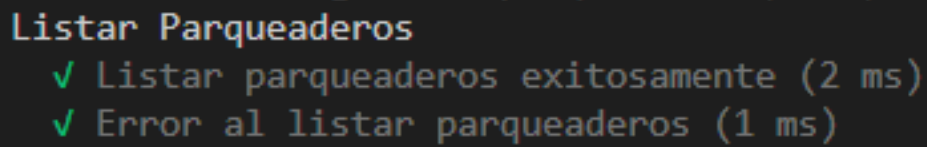
- Responses:** The main header of the client.
- Curl:** A text box containing the curl command:

```
curl -X 'GET' \
  'https://backend-espacios-esfot.onrender.com/api/parqueaderos' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3Nzk4MTZmMzEzZThiMTZkYmFjMmY3NSIsInJv
```
- Request URL:** A text box showing the URL: `https://backend-espacios-esfot.onrender.com/api/parqueaderos`.
- Server response:** A section indicating a successful response with a status code of 200.
- Response body:** A large text box displaying the JSON response data:

```
[
  {
    "_id": "6777ea78dddf4532cd632088",
    "nombre": "ESFOT",
    "description": "Parqueadero de la ESFOT",
    "planta": "1",
    "bloque": "A",
    "espacios": [
      {
        "numeroEspacio": "1",
        "estado": true,
        "_id": "6777ea78dddf4532cd632089"
      },
      {
        "numeroEspacio": "2",
        "estado": true,
        "_id": "6777ea78dddf4532cd63208a"
      },
      {
        "numeroEspacio": "3",
        "estado": true,
        "_id": "6777ea78dddf4532cd63208b"
      },
      {
        "numeroEspacio": "4",
        "estado": true,
        "_id": "6777ea78dddf4532cd63208c"
      },
      {
        "numeroEspacio": "5"
      }
    ]
  }
]
```

At the bottom right of the response body, there is a 'Download' button.

Figura 3.10 Listado de plazas de estacionamiento.



```
Listar Parqueaderos
✓ Listar parqueaderos exitosamente (2 ms)
✓ Error al listar parqueaderos (1 ms)
```

Figura 3.11 Prueba unitaria - listado de plazas de estacionamiento

Sprint 2 - Codificación de endpoints para el Guardia.

Este Sprint se enfoca principalmente en las tareas que puede ejecutar el usuario guardia una vez que haya iniciado sesión exitosamente. En ese sentido, las tareas que abarca este Sprint son las siguientes.

- Codificación de endpoints para visualizar status de plaza de estacionamientos.
- Codificación de endpoints para gestionar usuarios externos (estudiantes e invitados).

Codificación de endpoints para la visualización de status de plazas de estacionamiento

El backend otorga varios endpoints y rutas específicas para que el usuario administrador, guardia y usuarios externos puedan visualizar las plazas de estacionamiento disponible de una forma gráfica y comprensible tal y como se lo observa en la **Figura 3.12** y en **Figura 3.13** se presenta el resultado de la prueba unitaria y en el **ANEXO III** para visualizar toda la funcionalidad completa.

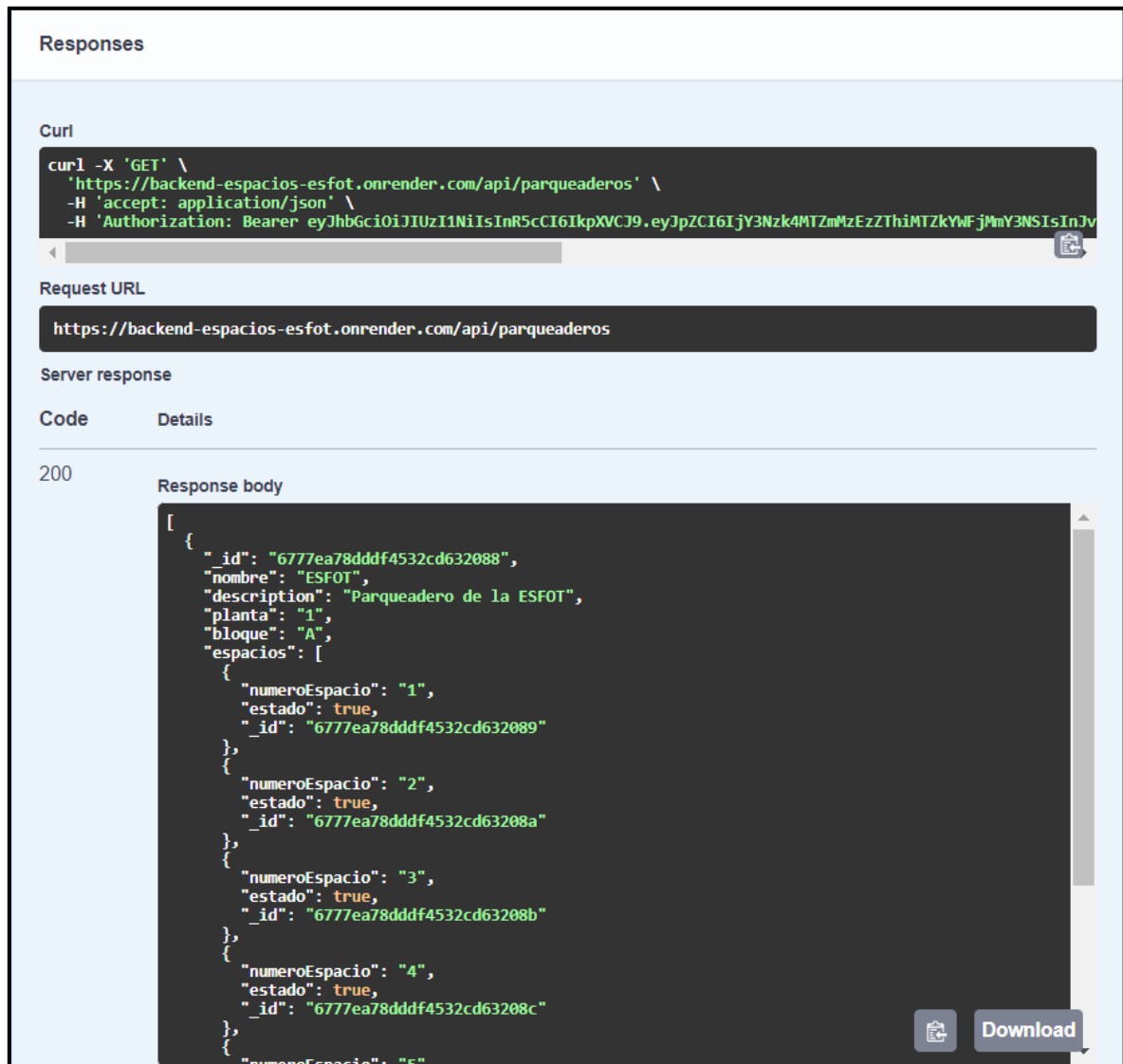


Figura 3.12 Listado de plazas de estacionamiento.

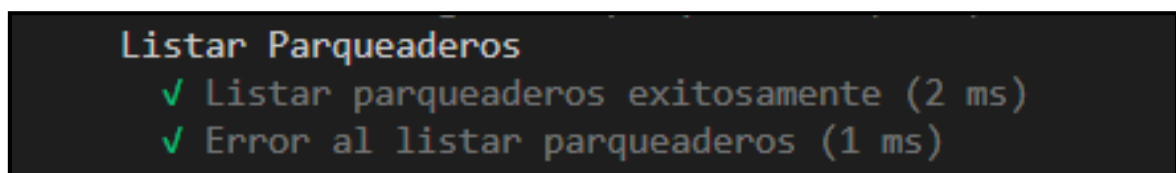


Figura 3.13 Prueba unitaria - listado de plazas de estacionamiento.

Codificación de endpoints para la gestión de usuarios externos (estudiantes e invitados)

El backend otorga varios endpoints y rutas específicas para que el usuario administrador y guardia puedan gestionar a los usuarios externos tal y como se lo

observa en la **Figura 3.14** y en **Figura 3.15** se presenta el resultado de la prueba unitaria y en el **ANEXO III** para visualizar toda la funcionalidad completa.

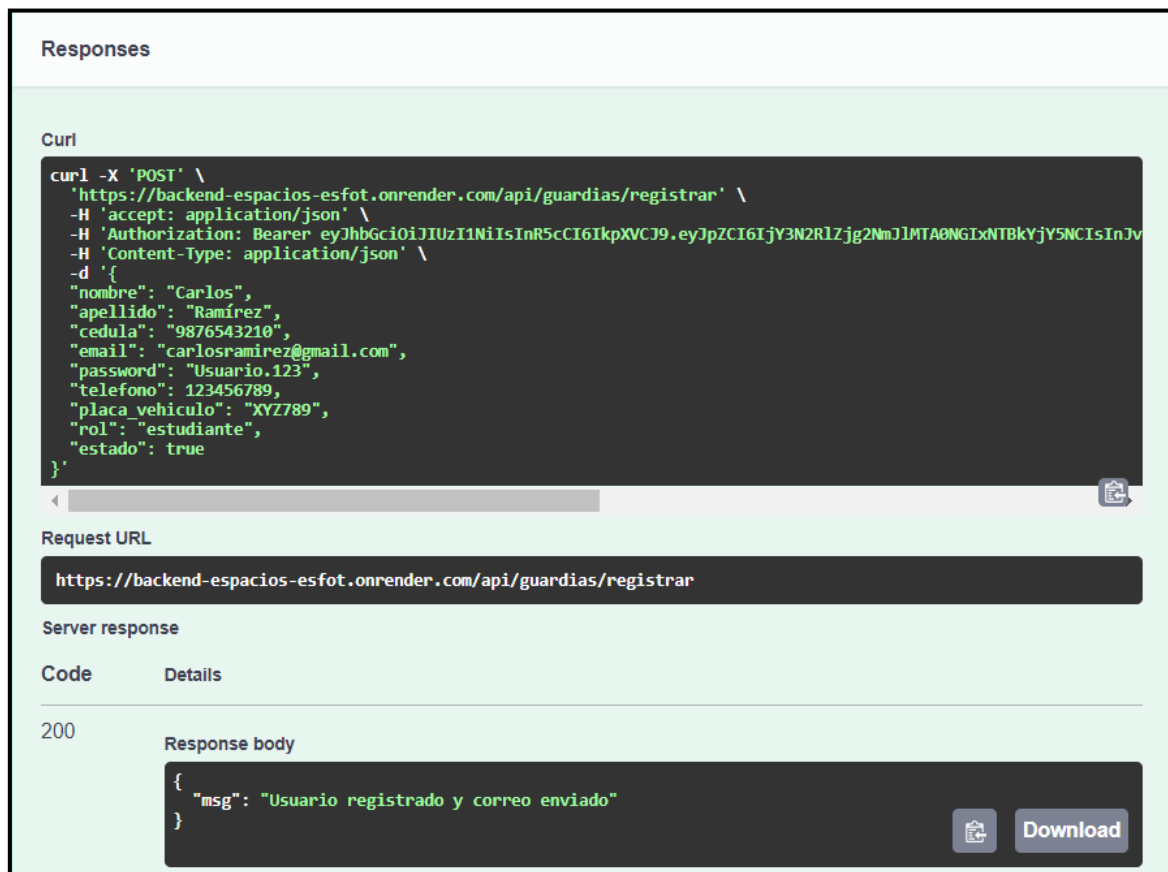


Figura 3.14 Registro de usuario externo.

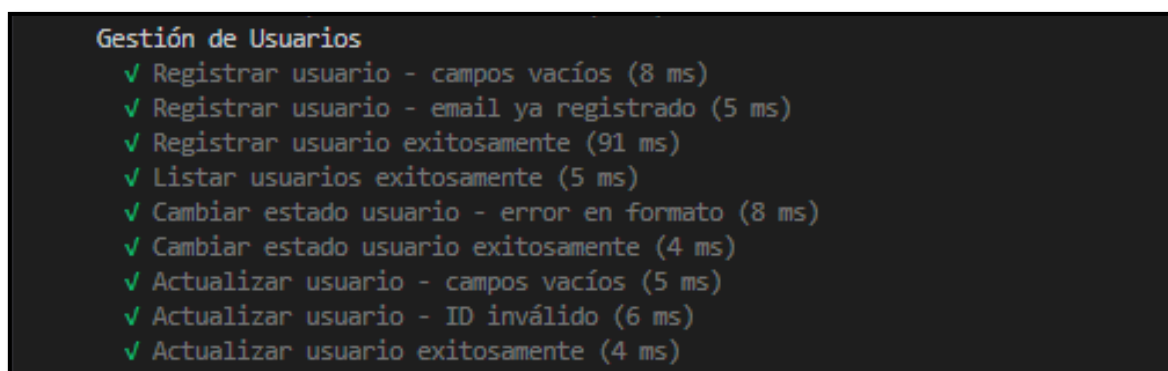


Figura 3.15 Prueba unitaria - Registro de usuario externo.

Sprint 3 - Codificación de endpoints para el usuario externo.

Este Sprint se enfoca principalmente en la tarea que puede ejecutar el usuario externo (personal docente, administrativo, estudiantes e invitados) una vez que haya iniciado sesión exitosamente. En ese sentido, la tarea que abarca este Sprint es la siguiente.

Codificación de endpoints para la gestión de perfil

El backend otorga varios endpoints y rutas específicas para que únicamente el usuario externo pueda gestionar su perfil actualizando su contraseña y número de teléfono, de igual forma este usuario puede visualizar el parqueadero de la ESFOT, como se lo observa en la **Figura 3.16** y en **Figura 3.17** se presenta el resultado de la prueba unitaria y en el **ANEXO III** para visualizar toda la funcionalidad completa.

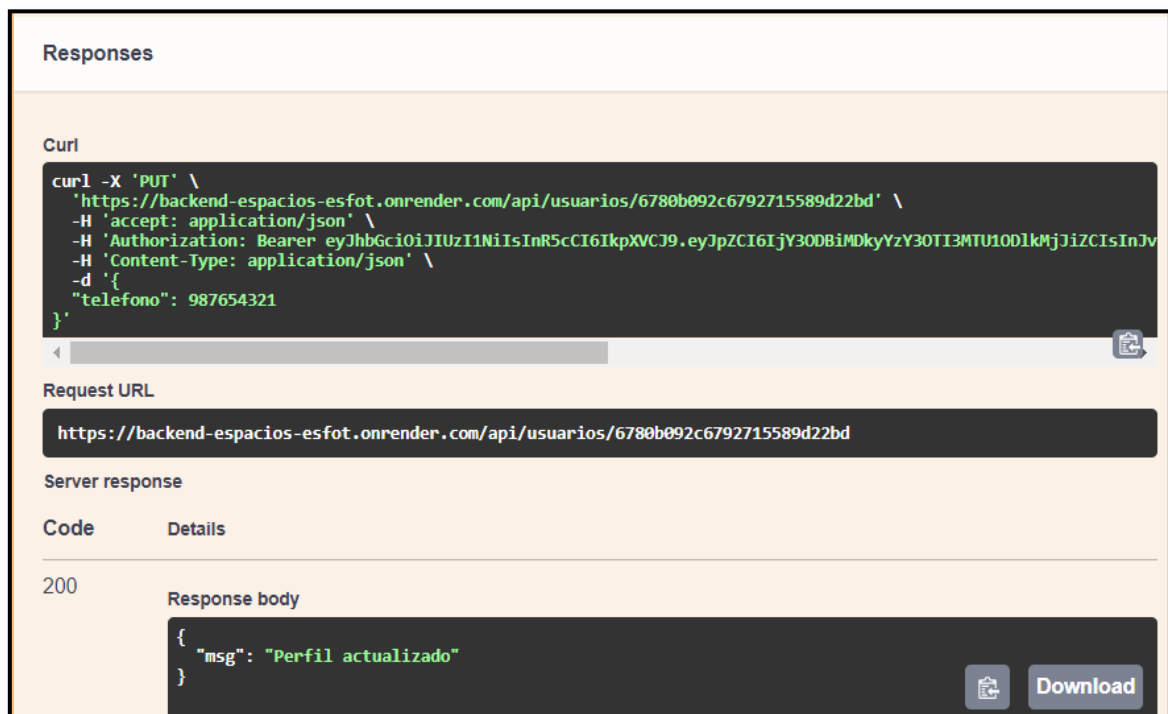


Figura 3.16 Actualización de teléfono.

```
✓ Perfil usuario exitoso (19 ms)
✓ Actualizar contraseña exitosamente (7 ms)
✓ Actualizar perfil exitosamente (11 ms)

Test Suites: 1 passed, 1 total
Tests:       10 passed, 10 total
Snapshots:   0 total
Time:        5.151 s, estimated 6 s
Ran all test suites matching /usuario_controller.test.js/i.
```

Figura 3.17 Prueba unitaria – Acciones usuarios externos.

Sprint 4 – Pruebas.

La implementación de este Sprint abarca todo el conjunto de pruebas que se han realizado en cada uno de los endpoints que se han creado para cada uno de los usuarios respectivamente. En ese sentido, las tareas que abarca este Sprint son las siguientes.

- Pruebas unitarias.
- Pruebas de estrés.
- Pruebas de aceptación.

Pruebas unitarias

Este tipo de prueba se encarga de verificar cómo se comporta un sistema software, pero en unidades más pequeñas de código analizando componentes individuales para que de esta manera sea mucho más fácil verificar errores en las primeras etapas del desarrollo [41]. En ese sentido, esta prueba ha sido fundamental para corroborar el buen funcionamiento de todos los endpoints, así como resolver de forma temprana cualquier eventualidad y que sea útil para realizar mantenimientos a nivel de código. El detalle de la prueba se encuentra en la **Figura 3.18**, el resultado en la **Figura 3.19** y para ver el detalle de cada una de las pruebas restantes se puede dirigir al **ANEXO II**.

```

test("Registrar un guardia exitosamente", async () => {
  req.body = {
    email: "guardia@test.com",
    password: "password123",
    nombre: "Guardia",
  };

  const mockGuardia = {
    ...req.body,
    encrypPassword: jest.fn().mockResolvedValue("hashed-password"),
    save: jest.fn().mockResolvedValue(true),
  };

  guardias.findOne.mockResolvedValue(null);
  guardias.prototype.encrypPassword = mockGuardia.encrypPassword;
  guardias.prototype.save = mockGuardia.save;

  await registroGuardias(req, res);
});

```

Figura 3. 18 Código de prueba unitaria - creación del usuario guardia.

```

PS C:\Scarlett\quinto semestre\tesis\Backend_espacios_parqueadero_esfot - copia> npm run test

> examen2_parqueamiento@1.0.0 test
> cross-env NODE_ENV=test jest --runInBand --detectOpenHandles --verbose

PASS test/administrador_controller.test.js (6.508 s)
  Pruebas Unitarias - Administrador y Gestión de Usuarios
    ✓ Login administrador exitoso (22 ms)
    ✓ Recuperar contraseña - envía correo exitosamente (8 ms)
    ✓ Comprobar token de contraseña exitosamente (6 ms)
    ✓ Nueva contraseña - actualización exitosa (8 ms)
    ✓ Nueva contraseña - contraseñas no coinciden (8 ms)
    ✓ Nueva contraseña - campos vacíos (8 ms)
    ✓ Registrar un guardia exitosamente (10 ms)
    ✓ Listar guardias exitosamente (7 ms)
    ✓ Actualizar perfil del guardia exitosamente (9 ms)
    ✓ Cambiar estado del guardia exitosamente (10 ms)
    ✓ Eliminar un guardia exitosamente (11 ms)
    ✓ Registro de usuario exitoso (10 ms)
    ✓ Registro de usuario - usuario ya existe (5 ms)
    ✓ Registro de usuario - campos vacíos (5 ms)
    ✓ Listar usuarios exitosamente (5 ms)
    ✓ Eliminar un usuario exitosamente (7 ms)

Test Suites: 1 passed, 1 total
Tests:       16 passed, 16 total
Snapshots:   0 total
Time:        6.804 s
Ran all test suites.

```

Figura 3.19 Resultado de prueba unitaria - creación del usuario guardia.

Pruebas de estrés

Este tipo de prueba analiza como un sistema software puede gestionar solicitudes más de lo habitual con el objetivo de detectar posibles cuellos de botella o puntos críticos donde se comienza a tener ciertos fallos. Además, la ventaja de esta prueba es que permite monitorear o distribuir de mejor manera los recursos para un mejor rendimiento [42]. En ese sentido, esta prueba ha sido fundamental para corroborar el comportamiento del backend frente a una gran cantidad de solicitudes entrantes y de existir algún problema poder realizar las correcciones pertinentes. En ese sentido, el detalle de la prueba se encuentra en la **Figura 3.20**, el resultado en la **Figura 3.21** y para ver el detalle de cada una de las pruebas restantes se puede dirigir al **ANEXO II**.



Figura 3.21 Resultado - prueba carga.

Mediante la aplicación de esta prueba de carga, se ha logrado monitorear de manera más efectiva los recursos clave que el backend ocupa, garantizando de esta manera un buen funcionamiento y que cualquier usuario no tenga problemas al momento de realizar sus peticiones.

Pruebas de aceptación

Este tipo de prueba se encarga de que todos los requisitos (funcionales o no funcionales) se centren en las expectativas del cliente para su posterior aprobación. Además, en este tipo de prueba se analiza que el sistema software no solo funcione técnicamente, sino que complementariamente sea útil y satisfactorio para el negocio [43]. En ese sentido, esta prueba ha sido fundamental para lograr la satisfacción del cliente y su aprobación final. De esta manera, el detalle de la prueba se encuentra en la **Tabla 3.1** y el detalle de cada una de las demás pruebas residen respectivamente en el **ANEXO II**.

Tabla 3. 1 Prueba aceptación - Gestionar plazas de estacionamiento.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-004	Identificador de historia de Usuario: HU004
Nombre: Gestionar plazas de estacionamiento.	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el administrador pueda:</p> <ul style="list-style-type: none"> • Crear el parqueadero con la información respectiva. • Listar el parqueadero con sus plazas de estacionamiento. • Actualizar la información del parqueadero. • Eliminar el parqueadero. 	
<p>Instrucciones de funcionamiento:</p> <p>Para actualizar y listar las plazas de estacionamiento se necesita de:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales de administrador correctamente. • Dirigirse al endpoint correspondiente (Parqueadero). 	

<ul style="list-style-type: none"> • Seleccionar la actividad que desee realizar.
Resultado deseado: El backend permite listar y actualizar la información del parqueadero de la ESFOT.
Evaluación: Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.

Mediante la aplicación de esta prueba de aceptación, se ha logrado obtener una satisfacción total por parte del cliente ya que todos los resultados presentan resultados favorables.

Sprint 5 – Deploy.

Este último Sprint se centra en una sola tarea que es realizar el despliegue del backend mediante una plataforma en la nube y en ese sentido, el despliegue de un software permite realizar despliegues rápidos sin necesidad de configuraciones avanzadas y la posibilidad de brindar una escalabilidad sin esfuerzo [17]. Por lo dicho anteriormente, el despliegue del backend se lo ha realizado en Render y para acceder a sus funcionalidades se debe ingresar en la siguiente URL.

<https://backend-espacios-esfot.onrender.com>

Así mismo para acceder a la documentación del backend, se encuentra en la siguiente URL.

<https://backend-espacios-esfot.onrender.com/api-docs>

4 CONCLUSIONES

En este apartado del documento de tesis se listan cada una de las conclusiones una vez que se ha terminado con éxito todo el proyecto de Integración Curricular.

- Se ha desarrollado una herramienta tecnológica completamente funcional, el cual permite gestionar en tiempo real la ocupación y disponibilidad de cada plaza de estacionamiento que posee el parqueadero de la ESFOT y que a futuro sea implementado en los demás espacios que posee el campus de la EPN.
- Parte fundamental de este proyecto ha sido un correcto levantamiento de requerimientos ya que los mismos han permitido que sean analizados meticulosamente y permitir así plasmar objetivos y funcionalidades claras. Por otro lado, también ha permitido llevar un cronograma de las actividades y cumplir de esta manera con los tiempos que se han establecido.
- El definir un modelo de datos NoSQL ha sido relevante para el desarrollo del backend ya que el mismo ha permitido optimizar el rendimiento de las consultas y que la gestión de los datos sea de forma flexible. Además, si a futuro se requiere implementar funcionalidades extras el modelo de datos se puede adaptar sin ningún inconveniente ya que es un modelo escalable.
- El codificar una serie de endpoints públicos y privados permite que exista una buena comunicación e intercambio de información con aplicaciones del lado del cliente, móviles o servicios externos. Además, esta codificación de endpoints asegura seguridad y control ya que los permisos son diferentes para cada usuario o rol.
- La ejecución de un conjunto de pruebas ha permitido verificar toda la funcionalidad del backend, asegurando de esta manera que se entregue un producto de calidad. Además, gracias a las pruebas se ha permitido identificar a tiempo posibles errores y corregirlos antes del despliegue a producción.
- El desplegar el backend a un ambiente de producción finiquita con éxito todo el proceso de planificación, desarrollo y pruebas permitiendo que sea accesible para todo el personal de la ESFOT.

5 RECOMENDACIONES

En este apartado del documento de tesis se listan cada una de las recomendaciones una vez que se ha terminado con éxito todo el proyecto de Integración Curricular y que las mismas sirvan para futuros trabajos.

- Es recomendable que antes de proceder con la etapa de codificación se defina una adecuada base de datos (SQL o NoSQL) para que la gestión de la información sea adecuada, las consultas resulten lo menos complejas posibles y sobre todo se garantice escalabilidad.
- Es recomendable que en la etapa de codificación se realicen pruebas unitarias, ya que estas pruebas permiten comprobar cada funcionalidad del backend y reducir con ello el riesgo de presentar nuevos errores cuando se ejecute otras pruebas.
- Es recomendable realizar un análisis de la plataforma de despliegue ya que las mismas al poseer una capa gratuita puede ocasionar que las respuestas a las peticiones sean lentas y por ende no brinde una buena experiencia de usuario.
- Es recomendable que todo backend disponga de una documentación de sus APIs ya que a más de ser una buena práctica de programación mejora la comprensión de su funcionamiento interno y de los datos que se deben enviar al momento de hacer una petición.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] E. P. Nacional, «Síntesis histórica,» Escuela Politécnica Nacional, [En línea]. Available: <https://www.epn.edu.ec/institucion/sintesis-historica/>. [Último acceso: 03 10 2024].
- [2] BookingNinjas, «Sistema de Gestión de Aparcamientos: Mejora la Eficiencia y la Experiencia del Usuario,» BookingNinjas, 05 04 2024. [En línea]. Available: <https://www.bookingninjas.com/es/blog/todo-lo-que-tenes-que-saber-sobre-el-sistema-de-gesti%C3%B3n-de-aparcamientos>. [Último acceso: 03 10 2024].
- [3] A. Stsepanets, «¿Cuál es y cómo hacer el alcance de un proyecto?,» Gantt Chart GanttPRO Blog, 26 06 2023. [En línea]. Available: <https://blog.ganttpro.com/es/alcance-del-proyecto/#que-es-alcance-de-un-proyecto>. [Último acceso: 04 10 2024].
- [4] IONOS, «¿Qué es el backend?,» IONOS Digital Guide, 06 07 2024. [En línea]. Available: <https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/>. [Último acceso: 04 10 2024].
- [5] freeCodeCamp, «What Exactly is Node.js? Explained for Beginners,» freeCodeCamp.org, 05 12 2022. [En línea]. Available: <https://www.freecodecamp.org/news/what-is-node-js/>. [Último acceso: 04 10 2024].
- [6] G. B, «¿Qué es JavaScript? Introducción básica a JS para principiantes,» Tutoriales Hostinger, 08 03 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-javascript-introduccion-basica/>. [Último acceso: 04 10 2024].
- [7] «¿Qué Es MongoDB?,» MongoDB, [En línea]. Available: <https://www.mongodb.com/es/company/what-is-mongodb>. [Último acceso: 04 10 2024].
- [8] IONOS, «MongoDB Compass: instalación y primeros pasos,» IONOS Digital Guide, 15 07 2024. [En línea]. Available: <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/mongodb-compass/>. [Último acceso: 04 10 2024].
- [9] R. Hat, «¿Qué es una API de REST?,» Red Hat - We make open source technologies for the enterprise, 31 07 2023. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Último acceso: 04 10 2024].

- [10] «Introducción a los tokens web JSON,» Auth0, [En línea]. Available: <https://auth0.com/es/learn/json-web-tokens>. [Último acceso: 04 10 2024].
- [11] R. S. Christoph Nienaber, «ASP.NET Core web API documentation with Swagger / OpenAPI,» Microsoft Learn: Build skills that open doors in your career, 26 08 2024. [En línea]. Available: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?tabs=visual-studio&view=aspnetcore-8.0>. [Último acceso: 04 10 2024].
- [12] «Introducción a los tokens web JSON,» Auth0, [En línea]. Available: <https://auth0.com/es/learn/json-web-tokens>. [Último acceso: 04 10 2024].
- [13] Raffino, «Etecé,» Enciclopedia concepto, 19 11 2023. [En línea]. Available: <https://concepto.de/http/>. [Último acceso: 11 10 2024].
- [14] M. Buenning, «Gestión de endpoints: explicación y ejemplos,» ninjaone, 04 09 2024. [En línea]. Available: <https://www.ninjaone.com/es/blog/que-es-la-gestion-de-endpoints/>. [Último acceso: 12 10 2024].
- [15] J. Rosa, «Qué es REST: Conoce su potencia,» OpenWebinars, 17 05 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>. [Último acceso: 12 10 2024].
- [16] D. A, «¿Qué es JSON?,» Hostinger, 10 06 2023. [En línea]. Available: <https://www.hostinger.com.ar/tutoriales/que-es-json>. [Último acceso: 12 10 2024].
- [17] «¿Qué es Render.com?,» faztweb, [En línea]. Available: <https://fazitweb.com/contenido/nodejs-deploy-en-render-com>. [Último acceso: 12 10 2024].
- [18] I. Rovira, «Estudio de caso,» Psicología y mente, 06 06 2024. [En línea]. Available: <https://psicologiymente.com/psicologia/estudio-de-caso>. [Último acceso: 11 10 2024].
- [19] T. e. innovación, «Top 5 metodologías de desarrollo de software,» UCMA, 5 10 2024. [En línea]. Available: <https://www.universitatcarlemany.com/actualidad/blog/metodologias-de-desarrollo-de-software/>. [Último acceso: 11 10 2024].
- [20] M. Lujan, «Metodologías Ágiles: Definición, Manifiesto, Principios, SCRUM, Kanban,» Innovar o Morir, 11 10 2024. [En línea]. Available: <https://innovaromorir.com/metodologias-agiles-definicion-manifiesto-principios-scrum-kanban/>. [Último acceso: 12 10 2024].
- [21] J. Martins, «Scrum: conceptos clave y cómo se aplica en la gestión de proyectos,» asana, 15 02 2024. [En línea]. Available: <https://asana.com/es/resources/what-is-scrum>. [Último acceso: 11 10 2024].

- [22] F. Gangemi, «Artefacto,» EAC, 31 05 2023. [En línea]. Available: <https://facundogangemi.com/articulos/scrum/artefactos-scrum/>. [Último acceso: 11 10 2024].
- [23] «Guía para la documentación de proyectos de software,» Universidad Nacional del Sur, 2017. [En línea]. Available: https://cs.uns.edu.ar/~ldm/mypage/data/oc/info/guia_para_la_documentacion_de_proyectos_de_software.pdf. [Último acceso: 12 10 2024].
- [24] T. Asana, «Historias de usuario: 3 ejemplos para generar valor para el usuario,» asana, 21 01 2024. [En línea]. Available: <https://asana.com/es/resources/user-stories>. [Último acceso: 12 10 2024].
- [25] A. Raeburn, «Qué es product backlog y guía para hacer uno con ejemplo,» asana, 06 02 2024. [En línea]. Available: <https://asana.com/es/resources/product-backlog>. [Último acceso: 12 10 2024].
- [26] T. Asana, «Backlog: qué es el trabajo pendiente del sprint y ejemplos,» asana, 06 02 2024. [En línea]. Available: <https://asana.com/es/resources/sprint-backlog>. [Último acceso: 12 10 2024].
- [27] RedHat, «¿Qué es una arquitectura de aplicaciones?,» RedHat, 15 03 2023. [En línea]. Available: <https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture>. [Último acceso: 12 10 2024].
- [28] R. Hernandez, «El patrón modelo-vista-controlador: Arquitectura y frameworks explicados,» freeCodeCamp, 28 06 2021. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>. [Último acceso: 12 10 2024].
- [29] E. e. Indeed, «10 herramientas de desarrollo de software preferidas por los desarrolladores,» indeed, 28 06 2024. [En línea]. Available: <https://es.indeed.com/orientacion-laboral/desarrollo-profesional/herramientas-desarrollo-software>. [Último acceso: 12 10 2024].
- [30] Y. Fernández, «Qué es Github y qué es lo que le ofrece a los desarrolladores,» Xataka Basics, 30 10 2019. [En línea]. Available: <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>. [Último acceso: 12 10 2024].
- [31] Microsoft, «Visual Studio,» Microsoft , 13 08 2024. [En línea]. Available: <https://visualstudio.microsoft.com/es/#vscode-section>. [Último acceso: 12 10 2024].
- [32] «Thunder Client,» Thunder Client, 28 09 2024. [En línea]. Available: <https://docs.thunderclient.com/>. [Último acceso: 12 10 2024].

- [33] «¿Qué es Express.js? Todo lo que Debes Saber,» Kinsta, 20 05 2022. [En línea]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-express/>. [Último acceso: 12 10 2024].
- [34] A. Gillis, «MongoDB,» TechTardet, 03 2023. [En línea]. Available: <https://www.techtarget.com/searchdatamanagement/definition/MongoDB>. [Último acceso: 12 10 2024].
- [35] E. Imagina, «¿Qué es una librería y cuál es su función?,» Imagina, 10 10 2024. [En línea]. Available: <https://imaginaformacion.com/tutoriales/que-es-una-libreria>. [Último acceso: 12 10 2024].
- [36] D. Arias, «Hashing in Action: Understanding bcrypt,» Auth0 by OKta, 25 02 2021. [En línea]. Available: <https://auth0.com/blog/hashing-in-action-understanding-bcrypt/>. [Último acceso: 12 10 2024].
- [37] J. Nicolás, «CORS. Qué es, cómo funciona, para qué sirve y cómo solucionarlo,» JuanNi, 15 11 2022. [En línea]. Available: <https://www.juannicolas.eu/cors-que-es-y-como-funciona/>. [Último acceso: 12 10 2024].
- [38] Tomer, «Cómo usar Nodemailer para enviar correos electrónicos desde tu servidor Node.js,» freeCodeCamp, 16 08 2023. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-usar-nodemailer-para-enviar-correos-electronicos-desde-tu-servidor-node-js/>. [Último acceso: 12 10 2024].
- [39] N. seguins, «¿Qué es JSON Web Token?,» alura LATAM, 28 04 2023. [En línea]. Available: <https://www.aluracursos.com/blog/que-es-json-web-token>. [Último acceso: 12 10 2024].
- [40] S. Orllvskyl, «A Complete Guide to Socket Programming in Python,» datacamp, 18 08 2023. [En línea]. Available: <https://www.datacamp.com/tutorial/a-complete-guide-to-socket-programming-in-python>. [Último acceso: 12 10 2024].
- [41] M. Venema, «¿Qué es la prueba unitaria?,» nimble Humanize Work, 26 03 2024. [En línea]. Available: <https://www.nimblework.com/es/agile/pruebas-unitarias/>. [Último acceso: 12 10 2024].
- [42] Tamushi, «Pruebas de estrés de software: ¿qué son y para qué sirven?,» TestingiT, 08 07 2022. [En línea]. Available: <https://www.testingit.com.mx/blog/pruebas-de-estres-de-software>. [Último acceso: 12 10 2024].
- [43] Tamushi, «Pruebas de aceptación de software, ¿Cuándo y por qué son necesarias?,» TestingiT, 23 08 2022. [En línea]. Available:

<https://www.testingit.com.mx/blog/pruebas-aceptacion-software>. [Último acceso: 12 10 2024].

- [44] D. Walsh, «Qué es GitHub y para qué sirve,» 2023. [En línea]. Available: <https://blog.hubspot.es/website/que-es-github#:~:text=Es%20una%20plataforma%20online%20de,proyec>. [Último acceso: 3 Mayo 2024].
- [45] F. Flores, «Qué es Visual Studio Code y qué ventajas ofrece,» 2022. [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. [Último acceso: 3 Mayo 2024].
- [46] F. Pérez, «THUNDER CLIENT LA ALTERNATIVA A POSTMAN EN VSCODE - LATIRUS,» 2020. [En línea]. Available: <https://www.latirus.com/blog/2021/04/23/thunder-client-la-alternativa-a-postman-en-vscode/#:~:text=A%20principios%20de%20Abril%20de,simular%20llamada s%20a%20las%20API..> [Último acceso: 3 Mayo 2024].
- [47] MDN, «Introducción a Express/Node,» 2023. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction. [Último acceso: 3 Mayo 2024].
- [48] Aplyca, «Vercel: Desarrollar, Previsualizar, Enviar,» 2023. [En línea]. Available: <https://www.aplyca.com/blog/blog-que-es-vercel-desarrollar-previsualizar-enviar>. [Último acceso: 3 Mayo 2024].
- [49] DataScientest, «MongoDB : todo sobre la base de datos NoSQL orientada a documentos,» 2024. [En línea]. Available: <https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos#:~:text=ingenier%C3%ADa%20de%20datos.-,MongoDB%20es%20una%20base>. [Último acceso: 3 Mayo 2024].
- [50] A. Ken, «Vercel: ¿Qué es y para qué sirve?,» gluo, 17 09 2024. [En línea]. Available: <https://www.gluo.mx/blog/vercel-que-es-y-para-que-sirve>. [Último acceso: 12 10 2024].
- [51] GeeksforGeeks, «Geeksforgeeks,» 31 marzo 2023. [En línea]. Available: <https://www.geeksforgeeks.org/password-encryption-in-node-js-using-bcryptjs-module/>. [Último acceso: 19 noviembre 2023].
- [52] IONOS, «IONOS Digital Guide,» 12 diciembre 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/cross-origin-resource-sharing/>. [Último acceso: 19 noviembre 2023].
- [53] Eduardo Arias, «Eduardo Arias,» 1 febrero 2023. [En línea]. Available: <https://eduardo-arias.com/dotenv-variables-de-entorno-node->

js/#:~:text=Dotenv%20permite%20asignar%20variables%20a,carpetas%20e
n%20el%20repositorio%20remoto.. [Último acceso: 19 noviembre 2023].

- [54] E. E. P. Gomez, «freeCodeCamp.org.,» 16 agosto 2023. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-usar-nodemailer-para-enviar-correos-electronicos-desde-tu-servidor-node-js/>. [Último acceso: 19 noviembre 2023].
- [55] A. Torres, «freeCodeCamp.org.,» 15 marzo 2023. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/introduccion-a-mongoose-para-mongodb/>. [Último acceso: 19 noviembre 2023].
- [56] L. M. L. Magaña, «freeCodeCamp.org.,» 14 abril 2023. [En línea]. Available: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>. [Último acceso: 19 noviembre 2023].

ANEXOS

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

- **ANEXO I.** Resultado del programa anti-plagio Turnitin.
- **ANEXO II.** Levantamiento de requerimientos y pruebas.
- **ANEXO III.** Manual de usuario.
- **ANEXO IV.** Credenciales de acceso.

ANEXO I

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

F_AA_236

CERTIFICADO DE ORIGINALIDAD TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 21 de enero de 2025

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado DESARROLLO DE UN BACKEND asociado al DESARROLLO DE SISTEMA PARA LA GESTIÓN DE PLAZAS DE ESTACIONAMIENTO PARA EL PARQUEADERO DE LA ESFOT elaborado por la estudiante SCARLETT STEFANY LUNA CAJAS 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 10%.

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,



Firmado digitalmente por:
BYRON GUSTAVO
LOARTE CAJAMARCA

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

ANEXO II

Requerimientos

En la **Tabla 1** se encuentra la recopilación de requerimiento que se han establecido previamente para el desarrollo.

Tabla 1 Recopilación de requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ITEM
Backend	RR-001	Para el usuario administrador, guardia y usuario se requiere crear endpoints para: <ul style="list-style-type: none">• Iniciar y cerrar sesión.• Recuperar contraseña.• Visualización del parqueadero de la ESFOT y su información.
	RR-002	Para el usuario administrador se requiere crear endpoints para: <ul style="list-style-type: none">• Gestionar guardias.
	RR-004	Para el usuario administrador se requiere crear endpoints para: <ul style="list-style-type: none">• Gestionar el parqueadero de la ESFOT.
	RR-005	Para el usuario guardia se requiere crear endpoints para: <ul style="list-style-type: none">• Gestionar perfil
	RR-008	Para el usuario externo se requiere crear endpoints para: <ul style="list-style-type: none">• Gestionar perfil.

Historias de Usuario

Las Historias de Usuario se encuentran detalladas a continuación, las mismas que han sido creadas una vez que se ha finalizado con la fase de levantamiento de requerimientos. De esta manera, se han realizado 8 Historias de Usuario, las cuales se detallan a continuación, desde la **Tabla 2** hasta la **Tabla 9**.

Tabla 2 Historia de usuario - Iniciar sesión.

HISTORIA DE USUARIO	
Identificador: HU-001	Usuario: Administrador, Guardia y Usuario
Nombre historia: Iniciar sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Responsable (s): Scarlett Luna	
Descripción: El backend crea un módulo funcional de inicio de sesión para: <ul style="list-style-type: none">• Ingresar al sistema según el rol del usuario.• Proteger rutas según el rol de usuario.• Reestablecer contraseña.• Visualización del parqueadero de la ESFOT y su información.	
Observación: Para que los usuarios puedan ingresar al sistema deben ingresar con las credenciales correctas, o se realiza el restablecimiento de la contraseña.	

Tabla 3 Historia de Usuario – Gestionar guardias.

HISTORIA DE USUARIO	
Identificador: HU-002	Usuario: Administrador
Nombre historia: Gestionar guardias	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Responsable (s): Scarlett Luna	
Descripción: El backend crea varios endpoints para que el administrador pueda: <ul style="list-style-type: none"> • Registrar guardia. • Actualizar la información o permitir/restringir el acceso del usuario guardia. • Listar a los guardias registrados. • Eliminar guardias. 	
Observación: Para que el administrador pueda gestionar guardias debe ingresar a con las credenciales correctas.	

Tabla 4 Historia de Usuario - Gestionar usuarios externos.

HISTORIA DE USUARIO	
Identificador: HU-003	Usuario: Administrador
Nombre historia: Gestionar usuarios externos (personal docente y personal administrativo).	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 3	
Responsable (s): Scarlett Luna	
Descripción: El backend crea varios endpoints para que el administrador pueda:	

<ul style="list-style-type: none"> • Listar a usuario externos (personal administrativo y docente, estudiantes e invitados). • Registrar usuarios externos (personal administrativo y docente). • Eliminar usuarios externos independientemente del rol que estos tengan.
Observación: Para que el administrador pueda gestionar guardias debe ingresar a con las credenciales correctas.

Tabla 5 Historia de Usuario - Gestionar plazas de estacionamiento y parqueadero de la ESFOT.

HISTORIA DE USUARIO	
Identificador: HU-004	Usuario: Administrador
Nombre historia: Gestionar plazas de estacionamiento y parqueadero de la ESFOT.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 4	
Responsable (s): Scarlett Luna	
Descripción: El backend crea varios módulos para que el administrador pueda: <ul style="list-style-type: none"> • Crear el parqueadero con la información respectiva. • Listar el parqueadero con sus plazas de estacionamiento. • Actualizar la información del parqueadero. • Eliminar el parqueadero. 	
Observación: Para que el administrador pueda gestionar plazas de estacionamiento debe ingresar a con las credenciales correctas.	

Tabla 6 Historia de Usuario – Módulo gestión perfil guardia.

HISTORIA DE USUARIO	
Identificador: HU-005	Usuario: Guardia
Nombre historia: Gestionar perfil.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Responsable (s): Scarlett Luna	
Descripción: El backend crea un endpoint para que el guardia pueda: <ul style="list-style-type: none"> • Visualizar su perfil. • Actualizar su información de perfil. • Actualizar su contraseña. 	
Observación: Para que el guardia pueda gestionar su perfil debe ingresar a con sus credenciales correctas.	

Tabla 7 Historia de Usuario – Gestionar usuario externo.

HISTORIA DE USUARIO	
Identificador: HU-006	Usuario: Guardia
Nombre historia: Gestionar usuario externo (estudiantes e invitados).	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 6	
Responsable (s): Scarlett Luna	
Descripción: El backend crea un endpoint para que el guardia pueda: <ul style="list-style-type: none"> • Crear usuarios externos (estudiantes e invitados). • Listar los usuarios registrados. 	

<ul style="list-style-type: none"> • Actualizar la información o permitir/restringir el acceso del usuario externo. • Actualizar la información de los usuarios externos.
Observación: Para que el guardia pueda gestionar su perfil debe ingresar a con sus credenciales correctas.

Tabla 8 Historia de Usuario – Gestionar reserva del parqueadero.

HISTORIA DE USUARIO	
Identificador: HU-007	Usuario: Guardia
Nombre historia: Gestionar reserva del parqueadero de la ESFOT.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 7	
Responsable (s): Scarlett Luna	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el guardia pueda:</p> <ul style="list-style-type: none"> • Visualizar el parqueadero de la ESFOT • Reservar el parqueadero completo. • Reservar el espacio para personas discapacitadas del parqueadero de la ESFOT. 	
Observación: Para que el usuario pueda visualizar plazas de estacionamiento debe ingresar a con sus credenciales correctas.	

Tabla 9 Historia de usuario – Gestionar perfil.

HISTORIA DE USUARIO	
Identificador: HU-008	Usuario: Usuario externo
Nombre historia: Gestionar perfil.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 8	
Responsable (s): Scarlett Luna	
Descripción: El backend crea un endpoint para que el usuario pueda: <ul style="list-style-type: none"> • Visualizar la información de su perfil. • Actualizar su información de perfil (teléfono). • Actualizar contraseña. • Visualizar el parqueadero de la ESFOT. 	
Observación: Para que el guardia pueda gestionar su perfil debe ingresar a con sus credenciales correctas.	

Product Backlog

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

Tabla 10 Product Backlog.

PRODUCT BACKLOG				
ID-HU	DESCRIPCIÓN DE LA HISTORIA	ITERACION ASIGNADA	ESTADO	PRIORIDAD
HU-001	Iniciar sesión	1	Finalizado	Alta
HU-004	Gestionar plazas de estacionamiento y parqueadero de la ESFOT	2	Finalizado	Media
HU-005	Gestión perfil guardia	2	Finalizado	Media
HU-006	Gestionar usuarios externos (estudiantes e invitados)	2	Finalizado	Media
HU-007	Gestionar reserva del parqueadero de la ESFOT	3	Finalizado	Alta
HU-008	Gestionar perfil usuario externo.	3	Finalizado	Media

Sprint Backlog

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

Tabla 11 Sprint Backlog

ELABORACION DEL SPRINT BACKLOG						
ID-SB	NOMBRE	MÓDULO	HU	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB-000	Configuración del ambiente de desarrollo	<ul style="list-style-type: none">• Instalación y actualización de las herramientas de desarrollo• Configuración de los modelos de datos				10 H
SB-001	Diseño y codificación de endpoints para el usuario administrador, guardia y usuario externo.	Módulo de Iniciar sesión	HU001	Gestionar cuenta	<ul style="list-style-type: none">• Diseño e implementación de endpoints para que los usuarios: administrador, guardia y usuario externo puedan iniciar sesión y recuperar contraseña.• Verificación de la integridad de los datos requeridos.	20H
		Módulo de Recuperar Contraseña				

		Modulo ver parqueadero			<ul style="list-style-type: none"> • Protección de las rutas según el usuario. • Los tres usuarios podrán Visualizar el parqueadero. 	
		Módulo Administrador	HU002	Gestionar guardias	<ul style="list-style-type: none"> • Modelamiento e implementación de endpoints para que el administrador realice acciones tales como: • Registro de información del guardia. • Actualizar la información del guardia en específico. • Consulta de los guardias registrados. • Eliminar al usuario guardia. • Validación de ingreso de información, además de verificación del token del usuario Administrador para que este acceda a los endpoints. 	

SB-002	Diseño y codificación de endpoints para el usuario administrador.	Módulo Administrador	HU003	Gestionar usuarios externos	<ul style="list-style-type: none"> • Modelamiento e implementación de endpoints para que el administrador realice acciones tales como: • Registro de información del usuario externo con el rol de personal administrativo y docente. • Consulta de los usuarios externos registrados. • Eliminar al usuario usuario. • Validación de ingreso de información, además de verificación del token del usuario Administrador para que este acceda a los endpoints. 	50H
--------	---	----------------------	-------	-----------------------------	---	-----

		Módulo Administrador	HU004	Gestionar parqueadero	<ul style="list-style-type: none"> • Modelamiento e implementación de endpoints para que el administrador realice acciones tales como: • Registro de información del parqueadero. • Actualizar la información del parqueadero. • Consulta de la información del parqueadero. • Eliminar el parqueadero. • Validación de ingreso de información, además de verificación del token del usuario Administrador para que este acceda a los endpoints. 	
					<ul style="list-style-type: none"> • Modelamiento e implementación de endpoints para que el guardia realice acciones tales como: 	

SB-003	Diseño y codificación de endpoints para el usuario guardia.	Módulo guardia datos personales	HU005	Gestionar datos personales	<ul style="list-style-type: none"> • Consulta de sus datos personales y actualice su contraseña dentro del sistema. • Validación de ingreso de información, además de verificación del token del usuario guardia. 	50H
		Módulo Guardia	HU006	Gestionar al usuario externo	<ul style="list-style-type: none"> • Modelamiento e implementación de endpoints para que el guardia realice acciones tales como: • Registro de información del usuario externo con el rol de personal estudiante e invitado. • Consulta de los usuarios externos registrados. • Validación de ingreso de información y validación del token por parte del guardia al consumir los endpoints. 	

		Módulo Guardia	HU007	Gestión reserva del parqueadero	<ul style="list-style-type: none"> • Modelamiento e implementación de endpoints para que el guardia realice acciones tales como: • Reserva del parqueadero completo. • Reservar el espacio de parqueo para discapacitados. • Validación de ingreso de información y validación del token por parte del guardia al consumir los endpoints 	
--	--	-------------------	-------	---------------------------------------	--	--

	Diseño y codificación de endpoints para el usuario externo.	Módulo usuario externo datos personales	HU008	Gestionar datos personales	<ul style="list-style-type: none"> Modelamiento e implementación de endpoints para que el usuario externo realice acciones tales como: Consulta de sus datos personales y actualice su contraseña dentro del sistema. Visualizar el parqueadero de la ESFOT. Validación de ingreso de información, además de verificación del token del usuario guardia. 	30
SB-004	Pruebas del Backend	<ul style="list-style-type: none"> Pruebas unitarias. Pruebas de estrés. Pruebas de aceptación. 			20H	
SB-005	Despliegue del Backend	<ul style="list-style-type: none"> Despliegue del Backend en Render. 			20H	
Documentación		<ul style="list-style-type: none"> Trabajo de Integración Curricular. Anexos. 			40H	
TOTAL					240H	

Elaboración de las colecciones de la Base de datos no relacional

A continuación, se presentan las colecciones de la base de datos NoSQL de la Figura 1 a Figura 4.

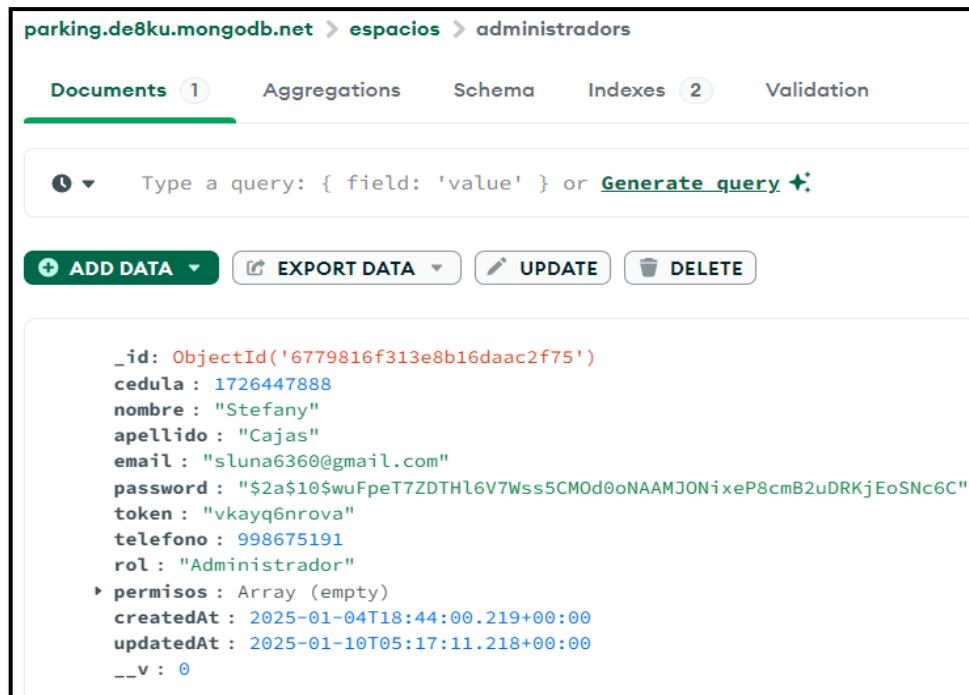


Figura 1 colección administrador.

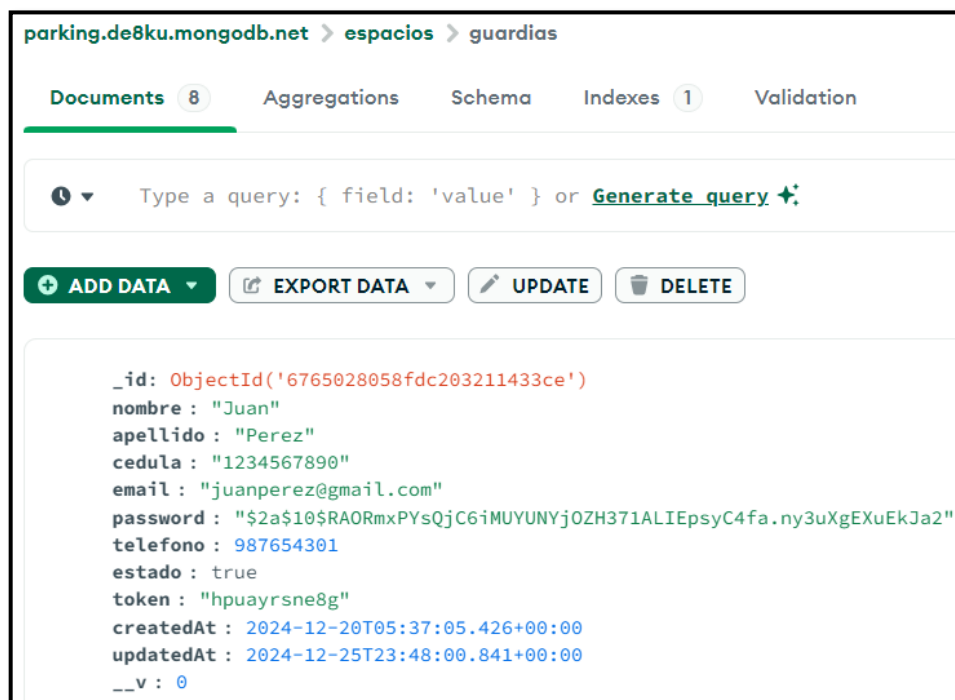


Figura 2 colección guardias.

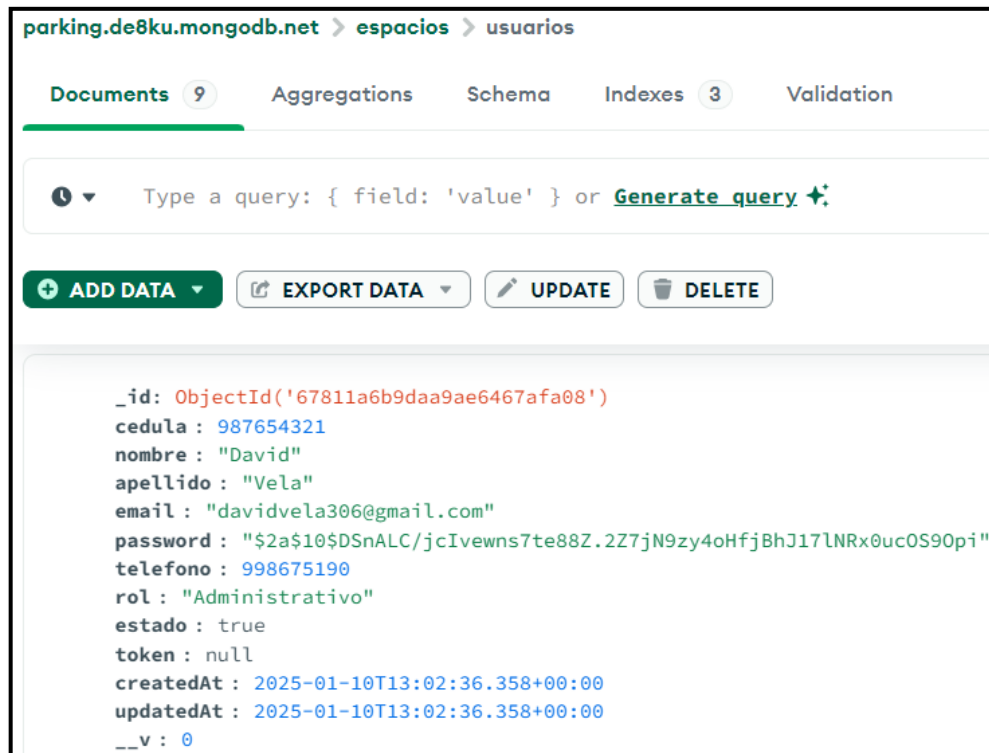


Figura 3 colección usuarios.

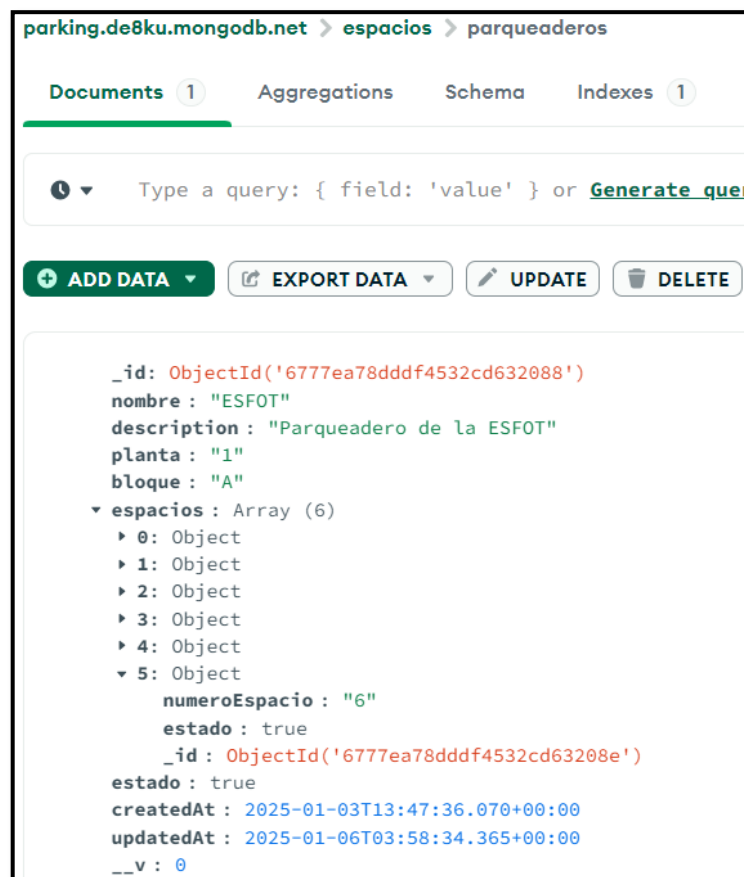


Figura 4 colección parqueadero.

Pruebas

Esta etapa verifica el correcto funcionamiento del backend.

Pruebas unitarias

Estas pruebas se llevan a cabo con el fin de verificar segmentos del código, como se muestran en las **Figura 5** a **Figura 6**.

```
PASS test/guardia_controller.test.js (7.132 s)
Pruebas Unitarias - Gestión de Guardias y Usuarios
Login y Autenticación
  ✓ Login guardia - campos vacíos (23 ms)
  ✓ Login guardia - email no registrado (10 ms)
  ✓ Login guardia - cuenta inactiva (13 ms)
  ✓ Login guardia - contraseña incorrecta (10 ms)
  ✓ Login guardia exitoso (11 ms)
Recuperación de Contraseña
  ✓ Recuperar contraseña - campos vacíos (8 ms)
  ✓ Recuperar contraseña - guardia no encontrado (7 ms)
  ✓ Recuperar contraseña - envío exitoso (7 ms)
Gestión de Token y Contraseña
  ✓ Comprobar token - token no proporcionado (18 ms)
  ✓ Comprobar token - token inválido (7 ms)
  ✓ Comprobar token válido (5 ms)
  ✓ Nueva contraseña - campos vacíos (7 ms)
  ✓ Nueva contraseña - contraseñas no coinciden (4 ms)
  ✓ Nueva contraseña exitosa (8 ms)
Gestión de Perfil
  ✓ Obtener perfil (5 ms)
  ✓ Actualizar contraseña - campos vacíos (8 ms)
  ✓ Actualizar contraseña - contraseña actual incorrecta (5 ms)
  ✓ Actualizar contraseña exitosamente (5 ms)
  ✓ Actualizar perfil - ID inválido (6 ms)
  ✓ Actualizar perfil exitosamente (7 ms)
Gestión de Usuarios
  ✓ Registrar usuario - campos vacíos (8 ms)
  ✓ Registrar usuario - email ya registrado (5 ms)
  ✓ Registrar usuario exitosamente (91 ms)
  ✓ Listar usuarios exitosamente (5 ms)
  ✓ Cambiar estado usuario - error en formato (8 ms)
  ✓ Cambiar estado usuario exitosamente (4 ms)
  ✓ Actualizar usuario - campos vacíos (5 ms)
  ✓ Actualizar usuario - ID inválido (6 ms)
  ✓ Actualizar usuario exitosamente (4 ms)
Gestión de Parqueaderos
  ✓ Cambiar estado parqueadero - estado inválido (4 ms)
  ✓ Cambiar estado parqueadero exitosamente (8 ms)
  ✓ Cambiar estado parqueadero - error del servidor (6 ms)
  ✓ Cambiar estado espacio - estado inválido (4 ms)
  ✓ Cambiar estado espacio - parqueadero no encontrado (3 ms)
  ✓ Cambiar estado espacio - espacio no encontrado (3 ms)
  ✓ Cambiar estado espacio exitosamente (4 ms)
  ✓ Cambiar estado espacio - error del servidor (3 ms)

Test Suites: 1 passed, 1 total
Tests: 37 passed, 37 total
Snapshots: 0 total
Time: 7.591 s
Ran all test suites matching /guardia_controller.test.js/i.
PS C:\Scarlett\quinto semestre\tesis\Backend_espacios_parqueadero_esfot - copia>
```

Figura 5 Prueba unitaria – Guardia.

```

> parqueadero_esfot@1.0.0 test
> cross-env NODE_ENV=test jest --runInBand --detectOpenHandles --verbose parqueadero_controller.test.js

PASS test/parqueadero_controller.test.js
Pruebas Unitarias - Controlador de Parqueaderos
  Registro de Parqueadero
    ✓ Registrar parqueadero exitosamente (5 ms)
    ✓ Error al registrar parqueadero (2 ms)
  Listar Parqueaderos
    ✓ Listar parqueaderos exitosamente (2 ms)
    ✓ Error al listar parqueaderos (1 ms)
  Actualizar Parqueadero
    ✓ Actualizar parqueadero exitosamente (2 ms)
    ✓ Error al actualizar - campos vacíos (1 ms)
    ✓ Error al actualizar - ID inválido (4 ms)
  Eliminar Parqueadero
    ✓ Eliminar parqueadero exitosamente (1 ms)
    ✓ Error al eliminar - ID inválido (2 ms)
    ✓ Error al eliminar - Parqueadero no encontrado (2 ms)

Test Suites: 1 passed, 1 total
Tests:       10 passed, 10 total
Snapshots:   0 total
Time:        1.369 s, estimated 2 s
Ran all test suites matching /parqueadero_controller.test.js/i.

```

Figura 6 Prueba unitaria – Parqueadero.

Pruebas de estrés

A continuación se presentan las restantes pruebas, las cuales van desde la **Figura 7** a la **Figura 12**.

The screenshot shows a web browser interface for configuring an HTTP request. The 'Name' field is set to 'VerParqueadero'. The 'Comments' field is empty. The 'Basic' tab is selected, showing the 'Web Server' section with 'Protocol [http:]' set to 'http', 'Server Name or IP' set to 'backend-espacios-esfotonrender.com', and 'Port Number' set to '80'. The 'HTTP Request' section shows the 'Method' set to 'GET' and the 'Path' set to '/api/parqueaderos'. The 'Content encoding' is set to 'utf-8'. The 'Parameters' tab is selected, showing a table for 'Send Parameters With the Request'.

Name:	Value	URL Encode?	Content-Type	Include Equals?

Figura 7 Petición visualización parqueadero.

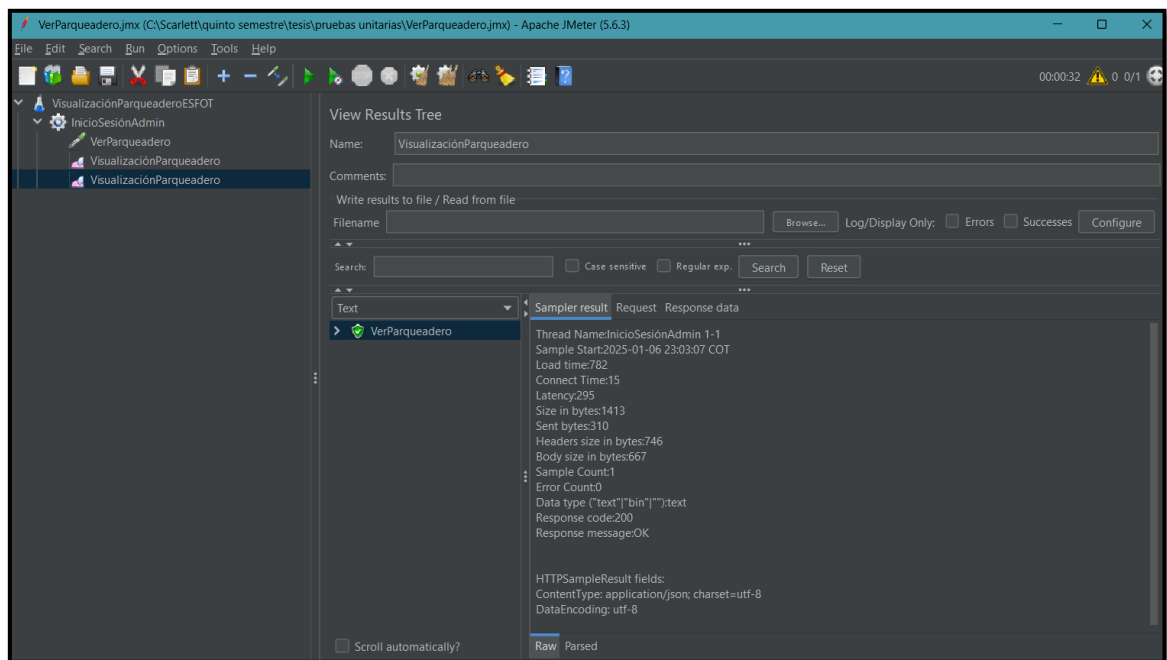


Figura 8 Resultado prueba de estrés.

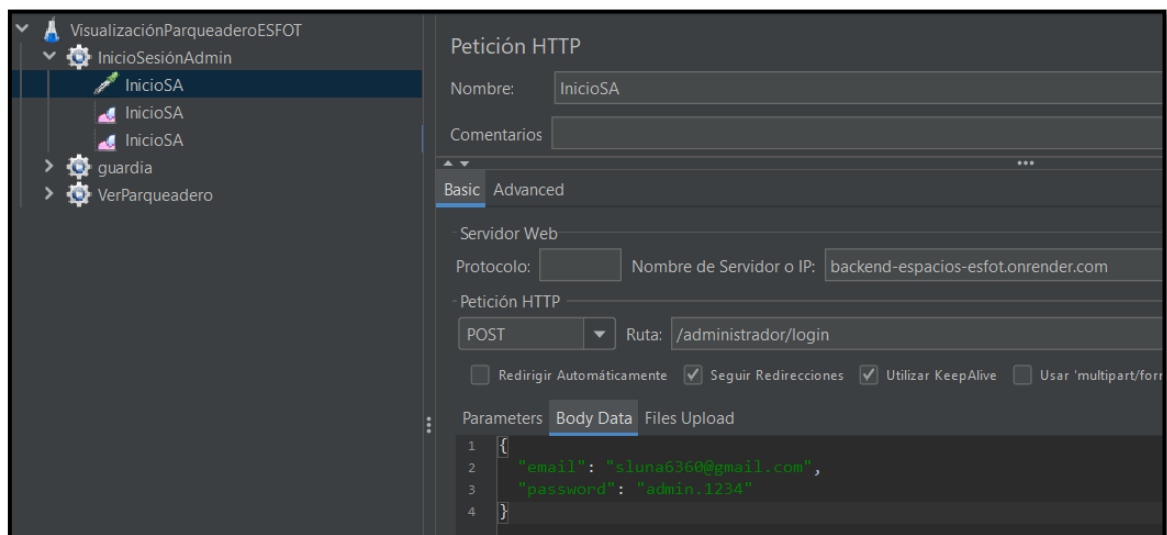


Figura 9 Petición Inició Sesión Administrador.

Ver Resultados en Árbol

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Muestra #	Tiempo de c...	Nombre del ...	Etiqueta	Tiempo de M...	Estado	Bytes	Sent Bytes	Latency	Connect Tim...
1	18:47:56.813	InicioSesión...	InicioSA	270		266	285	266	64

Figura 10 Resultado prueba de estrés Inició Sesión Administrador.

- VisualizaciónParqueaderoESFOT
 - InicioSesiónAdmin
 - guardia
 - Registro Usuario externo
 - Registro Usuario externo
 - Registro Usuario externo
 - VerParqueadero

Petición HTTP

Nombre:

Comentarios

Basic Advanced

Servidor Web

Protocolo: Nombre de Servidor o IP:

Petición HTTP

Ruta:

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multip

Parameters Body Data Files Upload

```

1  {
2    "nombre": "Shanon",
3    "apellido": "Guevara",
4    "cedula": "1717849976",
5    "email": "ejemplo@epn.edu.ec",
6    "telefono": 999669977,
7    "password": "usuario.123",
8    "placa_vehiculo": "PBF-267",
9    "rol": "Invitado",
10   "estado": true
11  }

```

Figura 11 Petición Registro de Usuario externo.

Ver Resultados en Árbol

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Muestra #	Tiempo de c...	Nombre del ...	Etiqueta	Tiempo de M...	Estado	Bytes	Sent Bytes	Latency	Connect Tím...
1	19:03:25.228	guardia 2-1	Registro Usu...	330		265	461	327	49

Figura 12 Resultado de la prueba de estrés Registro Usuario externo.

Pruebas de aceptación

En el siguiente apartado se muestran las pruebas de aceptación restantes que van desde la **Tabla 12** a **Tabla 19**.

Tabla 12 Prueba de aceptación de acciones globales para los usuarios.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-001	Identificador de historia de Usuario: HU001
Nombre: Gestionar el inicio de sesión	
Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que los tres usuarios (administrador, guardia y usuarios externos) puedan: <ul style="list-style-type: none"> • Ingresar al sistema según el rol que tengan. • Reestablecer su contraseña. • Proteger sus rutas según el rol. • Visualizar el parqueadero y la información de este. 	
Instrucciones de funcionamiento: Para actualizar y listar las plazas de estacionamiento se necesita de: <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales de usuario correctamente. • En caso de restablecer contraseña ingresar el correo. • Revisar su correo y restablecer la contraseña. • Al ingresar al sistema podrán visualizar el parqueadero en el endpoint 	

correspondiente (parquero).
Resultado deseado: El backend permite visualizar, actualizar la información y la clave del perfil guardia.
Evaluación: Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.

Tabla 13 Prueba aceptación de la gestión del guardia desde el perfil del administrador.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-002	Identificador de historia de Usuario: HU002
Nombre: Gestionar al usuario guardia desde el perfil del administrador	
Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el administrador pueda: <ul style="list-style-type: none"> • Crear al usuario guardia ingresando la información correctamente. • Desplegar la información correspondiente de todos los guardias registrados. • Actualizar la información o permitir/restringir el acceso del usuario guardia. • Eliminar al guardia. 	
Instrucciones de funcionamiento: Para actualizar y listar las plazas de estacionamiento se necesita de: <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales del administrador correctamente. • Dirigirse al endpoint correspondiente (guardia). • Seleccionar la actividad que desee realizar. 	
Resultado deseado: El backend permite crear, listar, actualizar o eliminar al usuario guardia desde el perfil del administrador.	

Evaluación:

Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.

Tabla 14 Prueba aceptación de la gestión del usuario externo desde el perfil del administrador.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-003	Identificador de historia de Usuario: HU003
Nombre: Gestionar al usuario externo desde el perfil del administrador	
Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el administrador pueda: <ul style="list-style-type: none"> • Crear al usuario externo (solamente con el rol docente, personal administrativo) ingresando la información correctamente. • Desplegar la información correspondiente de todos los usuarios externos registrados. • Actualizar la información o permitir/restringir el acceso del usuario guardia. • Eliminar al usuario externo. 	
Instrucciones de funcionamiento: Para actualizar y listar las plazas de estacionamiento se necesita de: <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales del administrador correctamente. • Dirigirse al endpoint correspondiente (usuarios). • Seleccionar la actividad que desee realizar. 	
Resultado deseado: El backend permite crear, listar, actualizar o eliminar al usuario externo desde el perfil del administrador.	
Evaluación: Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.	

Tabla 15 Prueba aceptación de la gestión el parqueadero desde el perfil del administrador.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-004	Identificador de historia de Usuario: HU004
Nombre: Gestionar plazas de estacionamiento	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el administrador pueda:</p> <ul style="list-style-type: none"> • Crear el parqueadero con la información respectiva. • Listar el parqueadero con sus plazas de estacionamiento. • Actualizar la información del parqueadero. • Eliminar el parqueadero. 	
<p>Instrucciones de funcionamiento:</p> <p>Para actualizar y listar las plazas de estacionamiento se necesita de:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales de administrador correctamente. • Dirigirse al endpoint correspondiente (Parqueadero). • Seleccionar la actividad que desee realizar. 	
<p>Resultado deseado:</p> <p>El backend permite listar y actualizar la información del parqueadero de la ESFOT.</p>	
<p>Evaluación:</p> <p>Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.</p>	

Tabla 16 Prueba aceptación del módulo guardia perfil.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-005	Identificador de historia de Usuario: HU005
Nombre: Gestionar de perfil usuario guardia	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el guardia pueda:</p> <ul style="list-style-type: none"> • Desplegar la información correspondiente a sus datos personales. • Actualizar la información de contacto (Nº telefónico). • Actualizar su contraseña. 	
<p>Instrucciones de funcionamiento:</p> <p>Para actualizar y listar las plazas de estacionamiento se necesita de:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales de guardia correctamente. • Dirigirse al endpoint correspondiente (perfil). • Seleccionar la actividad que desee realizar. 	
<p>Resultado deseado:</p> <p>El backend permite visualizar, actualizar la información y la clave del perfil guardia.</p>	
<p>Evaluación:</p> <p>Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.</p>	

Tabla 17 Prueba aceptación de gestión del usuario externo desde el perfil del guardia.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-006	Identificador de historia de Usuario: HU006
Nombre: Gestionar al usuario externo desde el perfil del guardia	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el guardia pueda:</p> <ul style="list-style-type: none"> • Crear al usuario externo (solamente con el rol estudiante e invitado) ingresando la información correctamente. • Desplegar la información correspondiente de todos los usuarios externos registrados. • Actualizar la información o permitir/restringir el acceso del usuario externo. • Eliminar al guardia. 	
<p>Instrucciones de funcionamiento:</p> <p>Para actualizar y listar las plazas de estacionamiento se necesita de:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales del guardia correctamente. • Dirigirse al endpoint correspondiente (usuarios). • Seleccionar la actividad que desee realizar. 	
<p>Resultado deseado:</p> <p>El backend permite crear, listar, actualizar o eliminar al usuario externo desde el perfil del administrador.</p>	
<p>Evaluación:</p> <p>Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.</p>	

Tabla 18 Prueba aceptación reservación del parqueadero de la ESFOT

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-007	Identificador de historia de Usuario: HU007
Nombre: Gestionar la reservación del parqueadero	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el guardia pueda:</p> <ul style="list-style-type: none"> • Reservar el parqueadero completo. • Reservar el espacio para personas discapacitadas del parqueadero de la ESFOT. 	
<p>Instrucciones de funcionamiento:</p> <p>Para actualizar y listar las plazas de estacionamiento se necesita de:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales de guardia correctamente. • Dirigirse al endpoint correspondiente (parqueadero). • Seleccionar la actividad que desee realizar. 	
<p>Resultado deseado:</p> <p>El backend permite visualizar, actualizar la información y la clave del perfil guardia.</p>	
<p>Evaluación:</p> <p>Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.</p>	

Tabla 19 Prueba aceptación del módulo usuario externo perfil.

PRUEBA DE ACEPTACIÓN “POLI-PARKING”	
Designación (ID): PA-008	Identificador de historia de Usuario: HU008
Nombre: Gestionar de perfil usuario guardia	
<p>Descripción: El componente backend entrega varios endpoints que puede ser utilizados para que el usuario externo pueda:</p> <ul style="list-style-type: none"> • Desplegar la información correspondiente a sus datos personales. • Actualizar la información de contacto (Nº telefónico). • Actualizar su contraseña. 	
<p>Instrucciones de funcionamiento:</p> <p>Para actualizar y listar las plazas de estacionamiento se necesita de:</p> <ul style="list-style-type: none"> • Acceder a la URL del componente backend en un navegador. • Iniciar sesión en el endpoint de login con las credenciales del usuario externo correctamente. • Dirigirse al endpoint correspondiente (perfil). • Visualizar el parqueadero de la ESFOT. • Seleccionar la actividad que desee realizar. 	
<p>Resultado deseado:</p> <p>El backend permite visualizar, actualizar la información y la clave del perfil usuario externo y visualizar el parqueadero de la ESFOT.</p>	
<p>Evaluación:</p> <p>Se ha mostrado una total satisfacción con el resultado que se ha obtenido, lo cual indica que las expectativas han sido alcanzadas de manera exitosa.</p>	

ANEXO III

Seguidamente, para visualizar el funcionamiento del componente backend se debe ingresar al siguiente enlace.

https://youtu.be/7OYq_7z56JU?si=reviQKTV_f7tTfsG

En este enlace se explica de forma clara y sencilla la funcionalidad del componente backend y cada uno de los perfiles o usuarios que se manejan.