

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 FRONTEND

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

DAVID ALEJANDRO VELA TOTOY

DIRECTOR: BYRON GUSTAVO LOARTE CAJAMARCA

DMQ, enero 2025

CERTIFICACIONES

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

David Alejandro Vela Totoy

david.vela@epn.edu.ec

davidvela306edu@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **DAVID ALEJANDRO VELA TOTOY**, bajo mi supervisión.

Ing. BYRON LOARTE MSc

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

DAVID ALEJANDRO VELA TOTOY

DEDICATORIA

A Dios, por ser mi luz y esperanza en todo momento.

A mi padre, Gustavo V., por su amor incondicional, su disciplina, su espíritu de superación y por enseñarme a nunca rendirme.

A mi madre, Maribel T., por su amor incondicional, por su bondad y su plena sonrisa.

A mi abuelo, Octavio T., por su carácter y coraje en los momentos difíciles, por ser guía en mi camino y nunca defraudarme.

A mi abuela Fanny O., por su amor eterno, su cuidado intensivo y por ser guía constante en el sendero de mi vida.

A mis hermanos Celeste V., Karen V. y Carlos V., por ser mi apoyo incondicional en cada paso del camino, mi motor en los momentos más difíciles y por siempre creer en mí, incluso cuando yo dudaba. Gracias por su amor, confianza y por recordarme constantemente el valor de la familia en mi vida.

A mis tíos Mónica T. y Marlon T., por su eterno amor, fraternidad y afecto, por ser mis compañeros incondicionales en mi etapa más temprana y por ser mi guía en los momentos actuales.

A Mateo C. por su generosidad y bondad, y por estar siempre dispuesto a ayudarme en los momentos más desafiantes. Su empatía y amabilidad han sido un verdadero regalo en mi vida.

A Ricardo T., Carlos T. y Marlene E., por personificar y dar verdadero sentido a las palabras amparo y nobleza. Gracias por acogerme en su hogar, hacerme sentir parte de ustedes y ser mi apoyo incondicional durante esta travesía.

A la familia Espín, por tenderme una mano en los momentos en que más lo necesitaba y por acogerme como uno más de ustedes, aún sin ser parte de su familia.

A Scarlett L., por enseñarme el sentido de humanidad, por ser mi apoyo y fiel compañera en esta travesía, y por brindarme su amor y cariño incondicional a lo largo de todo este proceso.

A mis amigos, en especial a Steven M., Anthony G., Anderson Y., Yulitza A., Damaris P., Ana P. y Ana P., quienes han sido y siguen siendo un regalo de Dios en mi vida, acompañándome y brindándome su apoyo incondicional.

AGRADECIMIENTO

Agradezco eternamente a la Escuela Politécnica Nacional, por brindarme en sus aulas la oportunidad de formarme como un profesional de excelencia, por fomentar en mí una mentalidad crítica y creativa, y por proporcionarme las herramientas necesarias para contribuir de manera significativa a la sociedad. Esta institución ha sido un pilar fundamental en mi desarrollo, y me siento profundamente orgulloso de ser parte de su legado académico, que me ha permitido crecer no solo como estudiante, sino como ser humano.

Mi más sincero agradecimiento a mis tutores, quienes, con su destreza, compromiso y dedicación, compartieron su conocimiento de manera precisa, valiosa y enriquecedora. Cada uno de ellos dejó una huella indeleble en mi formación profesional y personal, enseñándome no solo los fundamentos de mi disciplina, sino también el valor de la ética, el esfuerzo y la perseverancia. En especial, a mi tutor de tesis, Byron L., quien me guió con sabiduría, paciencia y una visión clara a lo largo de todo este proceso. Su apoyo incondicional y su orientación me permitieron transformar mis ideas en logros concretos. Sin duda, fue un pilar esencial para llevar a cabo este proyecto.

A mis compañeros de universidad, quienes, con su compañerismo, colaboración y apoyo mutuo, contribuyeron de manera significativa a mi crecimiento académico y personal. Gracias por compartir sus experiencias y conocimientos, por su disposición a ayudarme y por ser parte esencial de esta etapa de mi vida.

Por último, quiero agradecer a todos aquellos que estuvieron presentes en este proceso, ya sea brindándome su apoyo, compartiendo sus conocimientos o motivándome a seguir adelante, incluso en los momentos más difíciles. Cada uno de ustedes ha sido fundamental en la consecución de este logro, y mi gratitud hacia ustedes es infinita.

ÍNDICE DE CONTENIDO

| | |
|---|------|
| CERTIFICACIONES..... | I |
| DECLARACIÓN DE AUTORÍA..... | II |
| DEDICATORIA..... | III |
| AGRADECIMIENTO..... | IV |
| ÍNDICE DE CONTENIDO..... | V |
| RESUMEN | VII |
| ABSTRACT | VIII |
| 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO | 1 |
| 1.1 Objetivo general..... | 2 |
| 1.2 Objetivos específicos | 2 |
| 1.3 Alcance | 2 |
| 1.4 Marco Teórico | 3 |
| 2 METODOLOGÍA | 6 |
| 2.1 Metodología de Desarrollo | 6 |
| Roles | 7 |
| Artefactos | 8 |
| 2.2 Diseño de interfaces | 11 |
| Herramienta utilizada para el diseño | 11 |
| 2.3 Diseño de la arquitectura | 12 |
| Arquitectura Modelo Vista Controlador (MVC) | 12 |
| 2.4 Herramientas de desarrollo..... | 13 |
| Librerías..... | 14 |
| 3 RESULTADOS..... | 16 |
| Sprint 0 – Configuración del ambiente de desarrollo. | 16 |
| Sprint 1 Rol Administrador y creación de interfaces..... | 19 |
| Sprint 2 Rol Guardia y creación de interfaces. | 26 |
| Sprint 3 Rol usuario externo y creación de interfaces. | 30 |
| Sprint 4 Pruebas para el componente frontend. | 32 |
| Sprint 5 Despliegue del frontend. | 37 |
| 4 CONCLUSIONES | 39 |
| 5 RECOMENDACIONES..... | 40 |
| 6 REFERENCIAS BIBLIOGRÁFICAS..... | 41 |

| | |
|----------------|----|
| 7 ANEXOS..... | 45 |
| ANEXO I..... | 46 |
| ANEXO II..... | 47 |
| ANEXO III..... | 80 |
| ANEXO IV | 81 |

RESUMEN

La Escuela de Formación de Tecnólogos (ESFOT) posee diversas áreas y como parte de sus instalaciones incluye un parqueadero el cual tiene por objetivo garantizar la seguridad e integridad de los vehículos de docentes, personal administrativo y estudiantes que realizan sus labores diarias dentro de la misma. Sin embargo, para que este parqueadero funcione de manera adecuada, es necesario validar el ingreso únicamente a miembros autorizados, así como gestionar eficientemente la disponibilidad de los espacios disponibles ya que, sin el apoyo de herramientas tecnológicas, este proceso resulta ineficaz y costoso, incrementando así el riesgo de incidentes como robos y la incapacidad de monitorear en tiempo real el flujo de vehículos y los espacios destinados a personas con discapacidad.

Para abordar los desafíos antes citados, en este Trabajo de Integración Curricular se ha desarrollado un frontend que permita de forma gráfica y en tiempo real gestionar la ocupación y disponibilidad de cada plaza de estacionamiento que posee el parqueadero de la ESFOT. De esta manera, toda la comunidad politécnica y demás usuarios autorizados pueden acceder a funciones específicas según su rol facilitando la reserva de espacios y el monitoreo del flujo vehicular por medio de una solución basada en sensores.

Este documento se ha estructurado de la siguiente manera, en el primer capítulo se detalla la problemática, objetivos, alcance y el respectivo marco teórico que sustenta el desarrollo del componente frontend. Además, en el segundo capítulo, se expone el desarrollo del componente basado en la metodología Scrum, así como el diseño de interfaces, patrón de arquitectura, herramientas y librerías. Por otra parte, en el tercer capítulo, se detalla el resultado de cada una de las tareas que se han definido por cada Sprint, así como el resultado de las pruebas unitarias para validar la calidad del componente. Por último, en el cuarto y quinto capítulo, se definen las respectivas conclusiones y recomendaciones que se han logrado una vez que el componente frontend se ha puesto en producción de forma exitosa para su uso y difusión.

Palabras clave: Frontend, Scrum, React, Sensores ultrasónicos, React.

ABSTRACT

The School for the Training of Technologists (ESFOT) has several areas and as part of its facilities includes a parking lot which aims to ensure the safety and integrity of the vehicles of teachers, administrative staff and students who perform their daily work within it. However, for this parking lot to function properly, it is necessary to validate the entry of only authorized members, as well as to efficiently manage the availability of available spaces. Without the support of technological tools, this process is inefficient and costly, thus increasing the risk of incidents such as theft and the inability to monitor in real time the flow of vehicles and spaces for people with disabilities.

To address the aforementioned challenges, in this Curricular Integration Work we have developed a frontend that allows graphically and in real time to manage the occupancy and availability of each parking space that has the parking lot of the ESFOT. In this way, the entire polytechnic community and other authorized users can access specific functions according to their role, facilitating the reservation of spaces and the monitoring of vehicular flow through a sensor-based solution.

This document has been structured as follows: the first chapter details the problems, objectives, scope and the respective theoretical framework that supports the development of the frontend component. In addition, in the second chapter, the development of the component based on the Scrum methodology, as well as the design of interfaces, architecture pattern, tools and libraries is exposed. On the other hand, in the third chapter, the result of each of the tasks that have been defined for each Sprint is detailed, as well as the result of the unit tests to validate the quality of the component. Finally, in the fourth and fifth chapters, the respective conclusions and recommendations are defined once the frontend component has been successfully put into production for its use and dissemination.

Keywords: Frontend, Scrum, React, Ultrasonic sensors, React.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

A lo largo de los años la Escuela de Formación de Tecnólogos (ESFOT) ha desempeñado un papel fundamental en la Educación Superior Tecnológica del Ecuador, formando una considerable cantidad de profesionales en diversas carreras técnicas y que en la actualidad incluya innovadores programas educación dual que combina la formación teórica con la práctica en entornos laborales reales [1]. Además, como parte de sus instalaciones incluye un parqueadero el cual tiene por objetivo garantizar la seguridad e integridad de los vehículos de docentes, personal administrativo y estudiantes que realizan sus labores diarias dentro de la misma. Sin embargo, para que este parqueadero funcione de manera adecuada, es necesario validar el ingreso únicamente a miembros autorizados, así como gestionar eficientemente la disponibilidad de los espacios disponibles ya que, sin el apoyo de herramientas tecnológicas, este proceso resulta ineficaz y costoso, incrementando así el riesgo de incidentes como robos y suplantaciones de identidad, así como la incapacidad de monitorear en tiempo real el flujo de vehículos. Además, la ausencia de un sistema automatizado genera largas filas en las horas pico, provocando congestión y pérdida de tiempo.

En la actualidad, la adopción de tecnologías emergentes y sistemas basados en sensores han permitido que algunas empresas puedan optimizar la gestión de espacios de estacionamiento mediante la automatización de procesos como el control de acceso, la identificación de vehículos y la detección de disponibilidad en tiempo real. Además, la integración de soluciones basadas en sistemas de sensores facilita la recopilación de datos que pueden ser analizados para implementar mejoras continuas, reduciendo costos y sobre todo ofrecer una experiencia más fluida y segura para los usuarios [2].

Para abordar la problemática antes citada, en este Trabajo de Integración Curricular se ha desarrollado un frontend que permita de forma gráfica y en tiempo real gestionar la ocupación y disponibilidad de cada plaza de estacionamiento que posee el parqueadero de la ESFOT. De esta manera, toda la comunidad politécnica y demás usuarios autorizados pueden acceder a funciones específicas según su rol facilitando la reserva de espacios y el monitoreo del flujo vehicular por medio de una solución basada en sensores. Por último, esta solución tecnológica mejora la experiencia del usuario al reducir el tiempo de búsqueda de estacionamientos creando un entorno más ordenado y accesible para toda la comunidad universitaria por medio de nuevas tecnologías emergentes.

1.1 Objetivo general

Desarrollar un frontend para la gestión de plazas de estacionamiento para el parqueadero en la ESFOT.

1.2 Objetivos específicos

1. Identificar los requisitos que sean necesarios para el desarrollo y usuarios finales.
2. Diseñar interfaces de usuario tomando en cuenta requisitos y experiencia de usuario.
3. Codificar el frontend tomando en cuenta el consumo de endpoints y autenticación basada en roles.
4. Realizar pruebas para corroborar el correcto funcionamiento antes de su publicación.
5. Desplegar el frontend a un entorno de producción para su difusión y uso.

1.3 Alcance

El alcance en un proyecto permite delimitar con exactitud los objetivos, tiempos de entrega, el producto final que se pretenden alcanzar, entre otros aspectos. Además, al establecer el alcance de manera clara, se garantiza que las metas se logren sin incurrir en demoras ni mucho menos en un exceso de carga laboral para todo el equipo de trabajo [3].

Por lo antes citado, este proyecto se focaliza en la gestión del flujo de entrada, salida y estancia de vehículos dentro del parqueadero de la ESFOT utilizando para ello una interfaz amigable, una solución basada en el uso de sensores y una serie de roles. Además, para desarrollar este componente y sus funcionalidades se parte inicialmente con la toma de necesidades y la utilización de la metodología Scrum, la cual permite realizar un trabajo en equipo y el seguimiento de resultados y tareas a través de iteraciones. Posterior a la etapa de codificación, se realiza los prototipos, la selección de un diseño arquitectónico Modelo – Vista – Controlador (MVC), una estructura modular basada en componentes de React y una serie de pruebas indispensables con el objetivo de obtener un correcto funcionamiento para la etapa de despliegue a producción. A continuación, se presentan los roles que forman parte de este componente.

Roles para el proyecto.

- Administrador, Guardia y Usuario.

Endpoints de consumo para que el administrador pueda.

- Iniciar sesión.
- Gestionar guardias.
- Gestionar usuarios.
- Gestionar plaza de estacionamientos.

Endpoints de consumo para que el guardia pueda.

- Iniciar sesión.
- Gestionar perfil.
- Gestionar status de plaza de estacionamientos.

Endpoints de consumo para que el usuario pueda.

- Iniciar sesión.
- Gestionar perfil.
- Visualizar status de plaza de estacionamientos.

1.4 Marco Teórico

Frontend

Se refiere al área del desarrollo web que se enfoca en el aspecto visual e interactivo de un sitio, es decir, el diseño del mismo. Además, esta área abarca desde la organización estructural hasta los elementos estéticos como colores, fondos y dimensiones, así como las animaciones y efectos visuales. Sin duda el frontend viene a ser la parte con la cual los usuarios interactúan directamente y que se ejecuta en el navegador, comúnmente conocido como aplicación de lado del cliente [4].

Interfaz y experiencia de usuario (UI/UX)

La interfaz de usuario (UI) hace referencia a los componentes visuales tales como pantallas, botones, interruptores, íconos y otros elementos gráficos con los que el usuario interactúa al

utilizar un sitio web, una aplicación o cualquier dispositivo electrónico. Por su parte, la experiencia de usuario (UX) abarca la totalidad de la interacción que el usuario tiene con un producto, desde su usabilidad hasta las sensaciones generadas durante su uso. Aunque la interfaz de usuario puede influir en la experiencia del usuario, ambas son distintas, al igual que las funciones que desempeñan los diseñadores y desarrolladores [5].

HTTP-HTTPS

El Protocolo de Transferencia de Hipertexto (HTTP) es un conjunto de reglas que regula la comunicación entre cliente y servidor, facilitando el compartir información a través de la red. En contraste, el Protocolo Seguro de Transferencia de Hipertexto (HTTPS) asegura que todos los datos transmitidos sean cifrados, proporcionando de esta manera una capa adicional de seguridad en las comunicaciones [6].

Navegador web

Permite acceder a cualquier recurso en Internet, recuperando información de diferentes sitios y presentando su resultado en un dispositivo electrónico, ya sea de escritorio o móvil. Además, la transmisión de esta información se realiza por medio del protocolo HTTP garantizando que los usuarios puedan visualizar el contenido sin importar el navegador que utilicen o su ubicación [7].

Endpoint

Es una URL o punto final individual que se usa para acceder a un recurso producido por una aplicación web o desde una API RESTful [8].

JavaScript

Es un lenguaje de programación interpretado, ligero, y orientado a objetos. Aunque es ampliamente conocido principalmente por su aplicación en el desarrollo web, también es utilizado en otros entornos que no sea el navegador web [9].

React

Es una biblioteca de JavaScript que optimiza el desarrollo de interfaces de usuario interactivas. Además, su enfoque declarativo permite definir interfaces basadas en estados y que la actualización sea automática mediante un eficiente sistema de renderizado lo que mejora el rendimiento y escalabilidad de las aplicaciones [10].

JSX

React utiliza JSX qué es la combinación de JavaScript y XML, una sintaxis similar a HTML lo que ayuda construir componentes de manera simplificada y de fácil legibilidad [10].

Redux

Es una biblioteca para gestionar el estado global de la aplicación y sus componentes asociados. Además, esta biblioteca es independiente ya que se puede utilizar con cualquier capa o marco de interfaz de usuario, incluyendo Ember, Angular, React, vanilla JS y Vue [11].

Hooks

Son funciones especiales que permiten utilizar características de React, como el estado y el ciclo de vida, en componentes funcionales [12].

Componentes React

Es uno de los conceptos esenciales de React ya que constituyen los cimientos para la construcción de interfaces de usuario. Además, se puede crear componentes propios o si se lo requiere utilizar los ya prefabricados por la comunidad [13].

2 METODOLOGÍA

Según Robert Stake una metodología de estudio de casos se enfoca en el análisis detallado de un caso único con el fin de comprender su complejidad y singularidad dentro de contextos específicos, siendo especialmente útil para el estudio de fenómenos sociales y educativos. Además, identifica tres tipos de estudios de caso el intrínseco, cuyo propósito es entender un caso por su importancia particular; el instrumental, que utiliza un caso para investigar cuestiones más generales; y el colectivo, que examina varios casos simultáneamente para obtener una perspectiva más amplia sobre un problema común. De esta manera, esta metodología permite a los investigadores profundizar en situaciones concretas y complejas empleando para ello diversas fuentes de datos y técnicas como entrevistas y observaciones lo que facilita la identificación de elementos tanto comunes como distintivos del fenómeno en estudio [14].

Por lo dicho anteriormente, el presente Trabajo de Integración Curricular utiliza un estudio de casos para poder entender a fondo la problemática utilizando para ello varias fuentes de información y posterior a ello, plantear una solución viable que cumpla con los parámetros requeridos y sobre todo que sea medible y alcanzable.

2.1 Metodología de Desarrollo

Se refiere a un conjunto de herramientas, procedimientos, técnicas y otros elementos que asisten a desarrolladores en la planificación, codificación, pruebas e implementación de sistemas informáticos. Además, una metodología se estructura por lo general en fases, las cuales pueden subdividirse en sub-fases, guiando de esta manera a los programadores en la elección de los métodos más adecuados para cada etapa del proyecto, así como en su planificación, gestión, control y evaluación [15].

Por otra parte, las metodologías ágiles permiten desarrollar sistemas informáticos desde un enfoque ágil con el objetivo de ajustarse rápidamente a cualquier modificación y responder eficazmente con el menor impacto posible. Además, en un contexto global de rápida evolución, las empresas requieren sistemas software de forma ágil para garantizar de esta manera que sus productos y servicios se mantengan competitivos y de alta calidad; y es por esta razón que las metodologías ágiles se vuelven tan importantes en la actualidad ya que las mismas se enfocan en colaborar de manera eficiente durante todo el proceso, entregar continuamente valor al cliente, trabajar en ciclos cortos de desarrollo, retroalimentación continua y otros aspectos relevantes [16].

Scrum es una metodología relevantemente útil para proyectos donde los requisitos son cambiantes o no se están completamente definidos desde el principio. Además, al ser un enfoque ágil permite trabajar colaborativamente con el dueño del producto lo que garantiza que al final se entregue un producto de calidad. Por otra parte, esta metodología provee un conjunto de directrices, artefactos y roles que ayudan a todo el equipo de trabajo a enfocarse en cada entregable y mejorar continuamente mientras se desarrolla un producto de manera incremental [17]. En este sentido, para el desarrollo del componente frontend se ha utilizado la metodología ágil Scrum la cual permite trabajar con equipos multidisciplinarios y en ciclos iterativos enfocados en el cliente permitiendo de esta manera, cumplir con cada una de las fases y el objetivo general del proyecto.

Roles

Un rol se refiere a la función que desempeña una persona dentro de una organización o empresa con el objetivo de que actúe siguiendo ciertas pautas y expectativas. Además, los roles definen la posición que tiene una persona dentro de un equipo al momento de realizar un trabajo [18]. Específicamente hablando en el contexto del desarrollo de software un rol se refiere a las diferentes funciones y responsabilidades que debe desempeñar una persona al momento de trabajar dentro de un equipo y es por esta razón, que para el componente frontend se han definido los siguientes roles.

Product Owner

Actúa como representante de las partes interesadas externas y de clarificar toda lógica del negocio o empresa. Además, cabe destacar que este rol se encuentra enfocado en el negocio y no en el equipo de desarrollo [19]. Por lo citado anteriormente, la **Tabla 2.1** muestra a la persona responsable de llevar a cabo esta acción y de definir la prioridad de las funcionalidades que se van a implementar.

Scrum Master

Es un facilitador encargado de guiar el proceso de desarrollo dentro del marco de trabajo Scrum. Además, su principal responsabilidad es asegurar que todo el equipo aplique correctamente los valores, prácticas y reglas ágiles. Entre sus funciones principales se encuentra la organización de reuniones diarias de seguimiento, la eliminación de obstáculos que puedan afectar al equipo y el apoyo en la planificación de las iteraciones (Sprints) [19]. Por lo citado anteriormente, la **Tabla 2.1** muestra a la persona responsable de llevar a cabo esta acción y de que todo el equipo trabaje de manera coordinada.

Development Team

Es un grupo multifuncional y autoorganizado con la responsabilidad de entregar un producto incremental y que sea potencialmente desplegable para su revisión y aprobación. Además, este equipo suele estar compuesto por un grupo reducido pero comprometido con las tareas asignadas [19]. Por lo citado anteriormente, la **Tabla 2.1** muestra a la persona responsable de llevar a cabo esta acción y de entregar progresivamente un producto funcional al dueño del producto.

Tabla 2.1 Designación de Roles.

| ROL | INTEGRANTES |
|------------------|-------------------------|
| Product Owner | Ing. Byron Loarte, MSc. |
| Scrum Master | Ing. Byron Loarte, MSc. |
| Development Team | Sr. David Vela. |

Artefactos

Se refiere a un conjunto de documentos que se desarrollan y utilizan como elementos esenciales para el registro y estandarización de información. Además, su objetivo es proporcionar transparencia sobre el trabajo que se ha planificado, se encuentra realizando y se ha realizado, permitiendo de esta manera, que todo el equipo Scrum pueda planificar y priorizar las tareas para los próximos Sprints [20]. Por lo citado anteriormente, los artefactos que se han utilizado en el presente proyecto son cuatro los cuales han permitido plasmar toda la lógica del negocio de una forma clara y precisa.

Recopilación de Requerimientos

Permite llevar el registro de requisitos funcionales y no funcionales del producto final, el cual es un requisito necesario antes de comenzar un proyecto ya que la carencia o un levantamiento mal definido puede llevar al fracaso del proyecto [21]. Con base en lo mencionado anteriormente la **Tabla 2.2** presenta un pequeño fragmento de la Recopilación de Requerimientos y si se requiere ver la tabla completa la misma se aloja de forma completa en el **ANEXO II**.

Tabla 2.2 Recopilación de requerimientos.

| RECOPIACIÓN DE REQUERIMIENTOS | | |
|-------------------------------|---------|---|
| TIPO DE SISTEMA | ID - RR | ENUNCIADO DEL ITEM |
| Frontend | RR-001 | <p>Para el usuario administrador, guardia y usuario externo se requiere una interfaz para:</p> <ul style="list-style-type: none"> Visualizar página de bienvenida. |

Historias de Usuario

Permite llevar el registro de las funcionalidades que debe incluir un sistema de software y de determinar cuál es el valor que proporciona al cliente al momento de ser implementada. Además, las Historias de Usuario pueden tener un formato estándar que permita registrar toda la información de una forma sencilla y que sea de fácil comprensión para todo el equipo [22]. Con base en lo mencionado anteriormente la **Tabla 2.3** presenta un pequeño ejemplar de una Historia de Usuario y si se requiere ver las demás tablas completas las mismas se alojan de forma completa en el **ANEXO II**.

Tabla 2.3 Historias de Usuario.

| HISTORIA DE USUARIO | |
|--|--|
| Identificador: HU-001 | Usuario: Administrador, Guardia y Usuario |
| Nombre historia: Visualizar página de bienvenida | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Baja |
| Iteración asignada: 1 | |
| Responsable (s): Vela David | |
| <p>Descripción: El frontend crea varias páginas para que los usuarios, guardias y usuarios externos puedan:</p> <ul style="list-style-type: none"> Visualizar información sobre el negocio, la solución que otorga, los servicios que ofrece y un apartado de contactos. | |

Product Backlog

Permite llevar el registro de información sobre las Historias de Usuario y de la iteración a las cuales pertenecen, así como la prioridad que tiene en el negocio y para el desarrollo siendo relativamente útil para organizar el trabajo a realizar. Además, para muchos equipos, este artefacto es el principal elemento de gestión del proyecto y el más importante fuera del código [23]. Con base en lo mencionado anteriormente la **Tabla 2.4** presenta un pequeño fragmento del Product Backlog y si se requiere ver la tabla completa la misma se aloja de forma completa en el **ANEXO II**.

Tabla 2.4 Product Backlog.

| PRODUCT BACKLOG | | | |
|---------------------------------|--------------------|------------|-----------|
| DESCRIPCIÓN DE LA HISTORIA | ITERACION ASIGNADA | ESTADO | PRIORIDAD |
| Visualizar página de bienvenida | 1 | Finalizado | Media |
| Iniciar sesión | 2 | Finalizado | Alta |

Sprint Backlog

Permite llevar el registro de información sobre las tareas y actividades específicas que el equipo de un proyecto debe ejecutar durante una iteración (Sprint). Además, este artefacto se construye a partir de los elementos del Product Backlog y el cual debe gestionarse adecuadamente ya que de lo contrario puede existir el riesgo de trabajar en elementos que no sean relevantes para los clientes o la hoja de ruta del proyecto [24]. Con base en lo mencionado anteriormente la **Tabla 2.5** presenta un pequeño fragmento del Sprint Backlog y si se requiere ver la tabla completa la misma se aloja de forma completa en el **ANEXO II**.

Tabla 2.5 Sprint Backlog.

| SPRINT BACKLOG | | | | | | |
|----------------|---|-------------------------|-------|----------------------------|--|----------------|
| ID-SB | TAREA | MÓDULO | ID-HU | DESCRIPCION DE LA HISTORIA | DETALLE DE LA TAREA | TIEMPO (HORAS) |
| SB-002 | Implementación del sistema de autenticación | Módulo de autenticación | HU-02 | Iniciar sesión | <ul style="list-style-type: none"> • Diseño de la interfaz. • Consumir endpoint. • Codificar manejo de errores. • Implementar un sistema de enrutamiento basado en roles. • Verificar la funcionalidad. | 30 H |

2.2 Diseño de interfaces

Una Interfaz de usuario es un espacio de interacción que posee elementos estéticos visuales. Además, una interfaz facilita la comunicación entre el sistema informático y el usuario mediante la representación gráfica de objetos, iconos, elementos multimedia y otros elementos [25]. Como parte del proceso de desarrollo de software las interfaces permiten presentar los resultados de manera visual para el dueño del producto y por tal motivo se presenta a continuación la herramienta que se ha utilizado para la creación de interfaces.

Herramienta utilizada para el diseño

Para el diseño de todas las interfaces se ha utilizado la herramienta Figma, la cual es una plataforma colaborativa en la nube que permite crear interfaces de usuario (UI), crear prototipos interactivos y colaborar en proyectos de diseño en tiempo real lo que facilita la eficiencia al recibir retroalimentación desde entornos colaborativos. A continuación, se presenta un ejemplar

de un prototipo en la **Figura 2.1** y para ver los restantes prototipos, los mismos se alojan de forma completa en el **ANEXO II**.

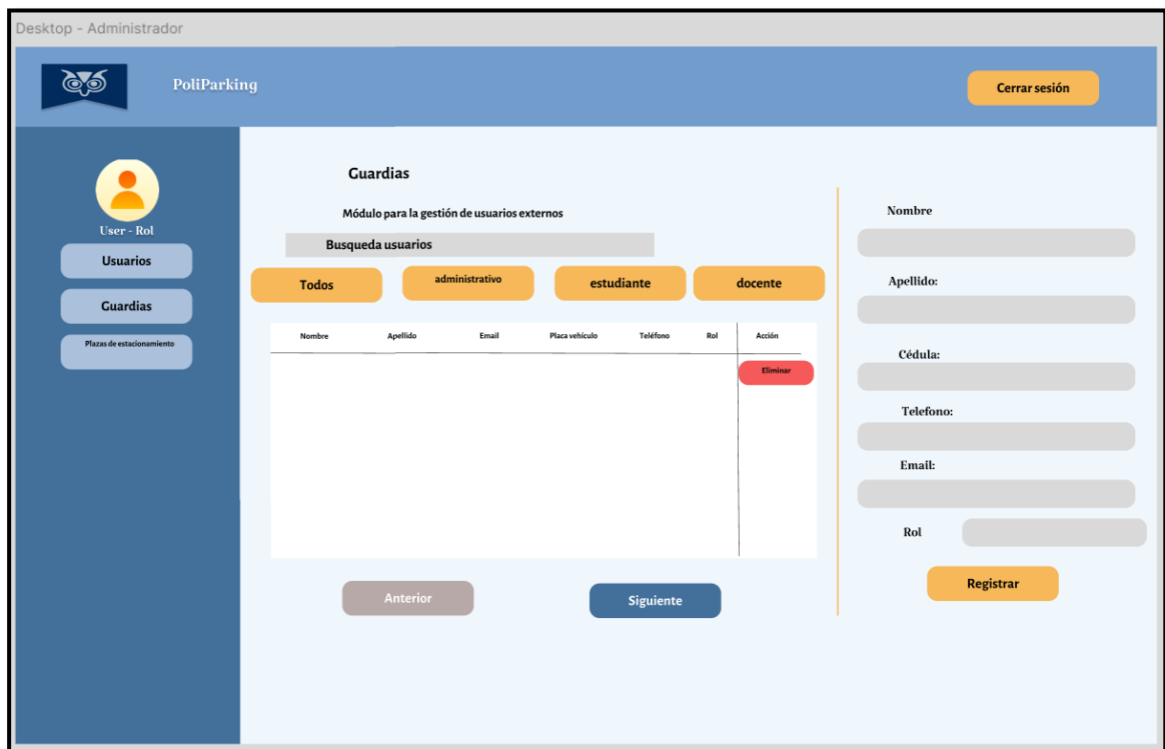


Figura 2.1 Interfaz de la gestión de guardias.

2.3 Diseño de la arquitectura

Una arquitectura de software en el desarrollo frontend se refiere a la estructura y organización de los diferentes componentes y módulos que lo conforman. Además, la elección de una arquitectura depende de factores como la complejidad, los requisitos, la escalabilidad, el mantenimiento, rendimiento, así como las tecnologías y herramientas que se utilizan en el desarrollo frontend.

Arquitectura Modelo Vista Controlador (MVC)

El patrón de arquitectura (MVC) propuesto por Trygve Reenskaug en 1979 se estructura en tres componentes esenciales el Modelo que encapsula los datos y las reglas relacionadas con su gestión; la Vista que exhibe toda la información del negocio a través de interfaces de usuario y el Controlador que interpreta las acciones entrantes del usuario y ejecuta las operaciones de la lógica del negocio sobre los otros componentes. Por este motivo, este patrón fomenta un desarrollo más ordenado y escalable a futuro si se lo requiere [26]. En la **Figura 2.2** se puede apreciar una representación gráfica de la arquitectura que se ha utilizado para el frontend.

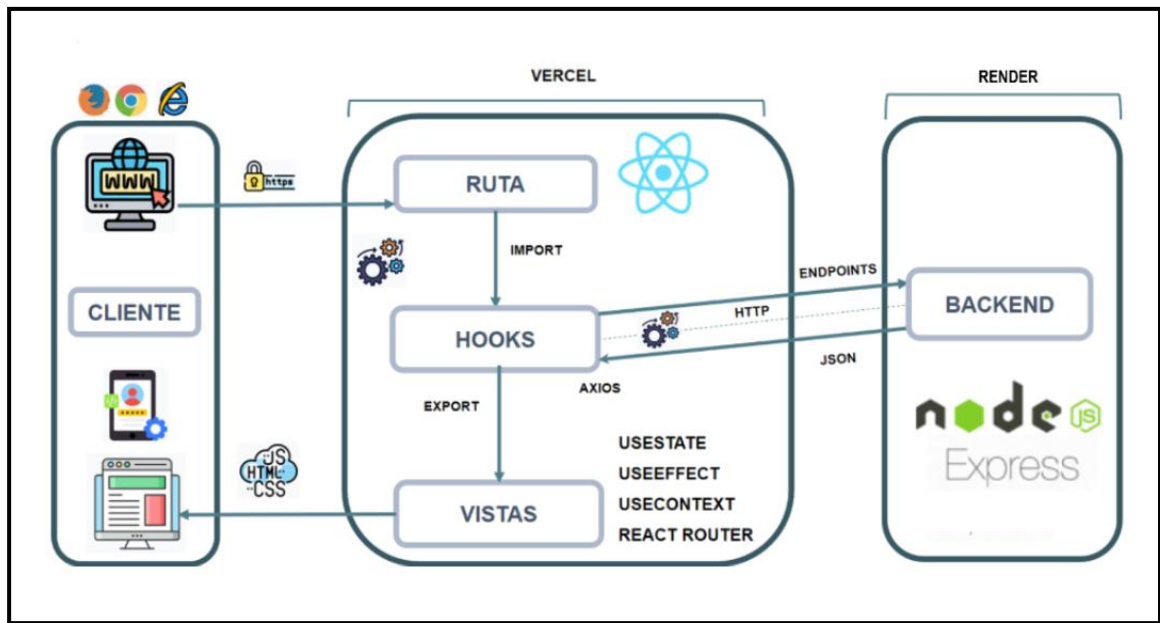


Figura 2.2 Diseño de la arquitectura frontend.

2.4 Herramientas de desarrollo

Una herramienta es un conjunto de utilidades que los desarrolladores utilizan para la creación, depuración, prueba y mantenimiento de sistemas informáticos. Además, estas herramientas están diseñadas estratégicamente para mejorar la eficiencia del desarrollador y centrarse en optimizar los flujos de trabajo y tareas más complejas [27]. En la **Tabla 2.6** se presenta las herramientas que se han usado para el actual proyecto frontend.

Tabla 2.6 Herramientas para la creación del componente frontend.

| HERRAMIENTA | JUSTIFICACIÓN |
|---------------------------|---|
| Git | Es un sistema que permite la gestión de versiones de código de una forma fácil y colaborativa siendo útil para llevar el registro de los cambios y avances del frontend [28]. |
| GitHub | GitHub es una plataforma que permite almacenar, compartir y colaborar en tiempo real con el código del frontend [29] |
| Visual Studio Code | Es un editor de código eficiente y ligero compatible con varios Sistemas Operativos y cuenta con un extenso ecosistema de |

| | |
|-----------------|--|
| | extensiones que permite desarrollar el código frontend de una forma fácil [30] |
| Chromium | Es un navegador de código abierto que busca ofrecer una experiencia web más segura y rápida al momento de ir probando cada una de las interfaces que se han desarrollado [31]. |

Librerías

Una biblioteca o librería en programación es un cúmulo de funciones y rutinas predefinidas que resuelven tareas comunes lo que permite que los desarrolladores puedan avanzar en sus proyectos sin necesidad de escribir código desde cero, ahorrando así tiempo y esfuerzo [32]. A continuación, en la **Tabla 2.7** se presenta las librerías que se han usado para el actual proyecto frontend.

Tabla 2.7 Librerías para la creación del componente frontend.

| LIBRERÍA | DESCRIPCIÓN |
|-------------------|--|
| jwt-decode | JWT-decode es una función o biblioteca que se emplea para decodificar tokens JWT (JSON Web Tokens) sin llevar a cabo la verificación de su firma. Esta herramienta permite extraer y visualizar los datos contenidos en el token, como el encabezado y la carga útil, facilitando el acceso a dicha información sin requerir su validación. [33] |
| React JS | Es una biblioteca que se ha utilizado para el desarrollo de interfaces de usuario del frontend [34]. |
| Socket io | Socket.IO es una biblioteca de JavaScript que simplifica el uso de WebSockets al ofrecer una capa de abstracción, ideal para desarrollar aplicaciones en tiempo real como chats y juegos. [35] |

| | |
|--------------|--|
| Axios | Es una biblioteca de cliente HTTP que ha permitido realizar solicitudes a endpoints, externos o servidores backend [36]. |
|--------------|--|

3 RESULTADOS

En esta sección, se resaltan cada uno de los logros que se han alcanzado durante el desarrollo del componente frontend centrándose específicamente en el resultado de la ejecución minuciosa de cada una de las fases que abarcan la planificación, ejecución, pruebas y el despliegue a producción. Además, cada fase del proceso implica la realización de tareas fundamentales cuyo cumplimiento es crucial para avanzar en las iteraciones de la metodología garantizando de esta manera un correcto cumplimiento del objetivo general. A continuación, se exponen los resultados que se han alcanzado en cada Sprint ofreciendo de esta manera una visión detallada sobre los procedimientos y tareas que se han realizado para una mejor comprensión.

Sprint 0 – Configuración del ambiente de desarrollo.

El inicio del proyecto se centra específicamente en preparar el ambiente de codificación, así como la especificación de requisitos y las responsabilidades de los roles de usuario que deben poseer para garantizar de esta manera un componente frontend de calidad y que se adapte a los requisitos que se han establecido. Por esta razón, este Sprint incluye las siguientes actividades.

- Especificación de requisitos para todo el componente frontend.
- Configuración y estructura del componente frontend.
- Funciones y responsabilidades de los roles de usuario para el componente frontend.

Especificación de requisitos para todo el componente frontend

Visualizar página de inicio

La primera interfaz que se puede visualizar dentro del componente es la página de inicio, la misma que le da una bienvenida al usuario y que posee un botón de inicio de sesión. Además, en esta primera página al ser informativa no es requerido que el usuario se encuentre con una sesión iniciada o registrado previamente.

Consumir endpoints de inicio de sesión

Para la interfaz de inicio de sesión, es necesario que el usuario se encuentre previamente registrado para que posterior a ello el usuario puede iniciar sesión mediante el formulario respectivo y mostrar los mensajes informativos si fuera el caso. Además, para que el frontend pueda realizar toda esta funcionalidad es vital que se realice el consumo del endpoint respectivo del backend.

Consumir endpoints para gestión de perfiles

Tras iniciar sesión, los usuarios externos (personal docente, administrativos, estudiantes e invitados) y guardias tienen acceso a la gestión de su perfil, la cual es una interfaz que permite visualizar y editar la información personal del usuario, así como cambiar su contraseña. Además, para que el frontend pueda realizar toda esta funcionalidad es vital que se realice el consumo del endpoint respectivo del backend asegurando que la información se actualice de manera segura y eficiente en la base de datos. Por último, para el caso del usuario externo no es permitido editar la placa vehicular.

Consumir endpoints para listar y gestionar parqueaderos

Esta interfaz ofrece a los usuarios la capacidad de visualizar y administrar los parqueaderos, según su rol. Además, los administradores pueden consumir los endpoints que les permiten listar todos los parqueaderos y modificar la información existente. Por otro lado, los guardias y usuarios externos pueden consumir los endpoints para ver el listado de parqueaderos y consultar la disponibilidad de las plazas de estacionamiento. Por último, no es permitido eliminar ni agregar nuevos parqueaderos.

Consumir endpoints para actualizar la contraseña

Esta interfaz ofrece a los usuarios la capacidad de consumir endpoints que permita la actualización de su contraseña. Además, esta funcionalidad solo es permitida para el usuario guardia y el usuario externo.

Consumir endpoints para gestionar usuarios externos

Esta interfaz ofrece a los usuarios la capacidad de gestionar usuarios externos, para que de esta manera el usuario administrador pueda únicamente gestionar personal docente y administrativos mientras que, el usuario guardia pueda únicamente gestionar estudiantes e invitados.

Configuración y estructura del componente frontend

Para el desarrollo del componente frontend en su totalidad y de cada una de las interfaces de usuario se utiliza principalmente Visual Studio Code (VS Code) en conjunto con React.js y Tailwind CSS. Facilitando de esta manera, un entorno de trabajo eficiente y productivo a través de la organización de archivos, la integración de plugins y la escritura de código de forma legible. En la **Figura 3.1** se detalla la estructura de todo el componente frontend para una mejor comprensión.

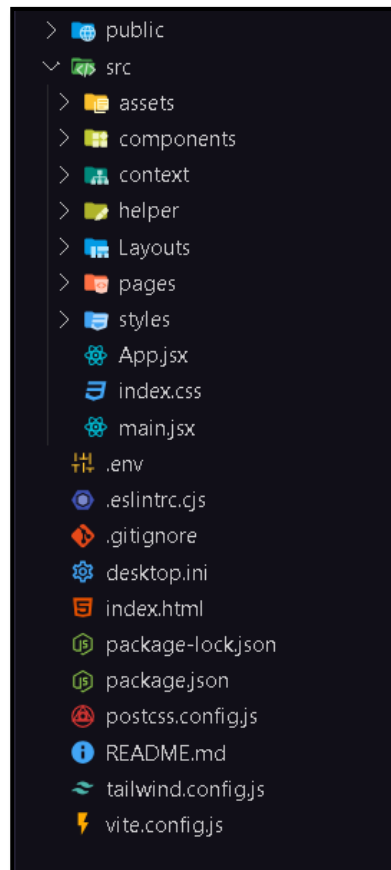


Figura 3.1 Estructura de carpetas y archivos para el componente frontend.

Funciones y responsabilidades de los roles de usuario para el componente frontend

Para todo el componente frontend se han establecido tres roles específicos con el propósito de asignar funciones concretas a cada uno de ellos, lo que permite un control más eficiente sobre las interfaces de usuario y su gestión. Muestra de ello en la **Figura 3.2** se detalla la asignación de roles dentro del componente.

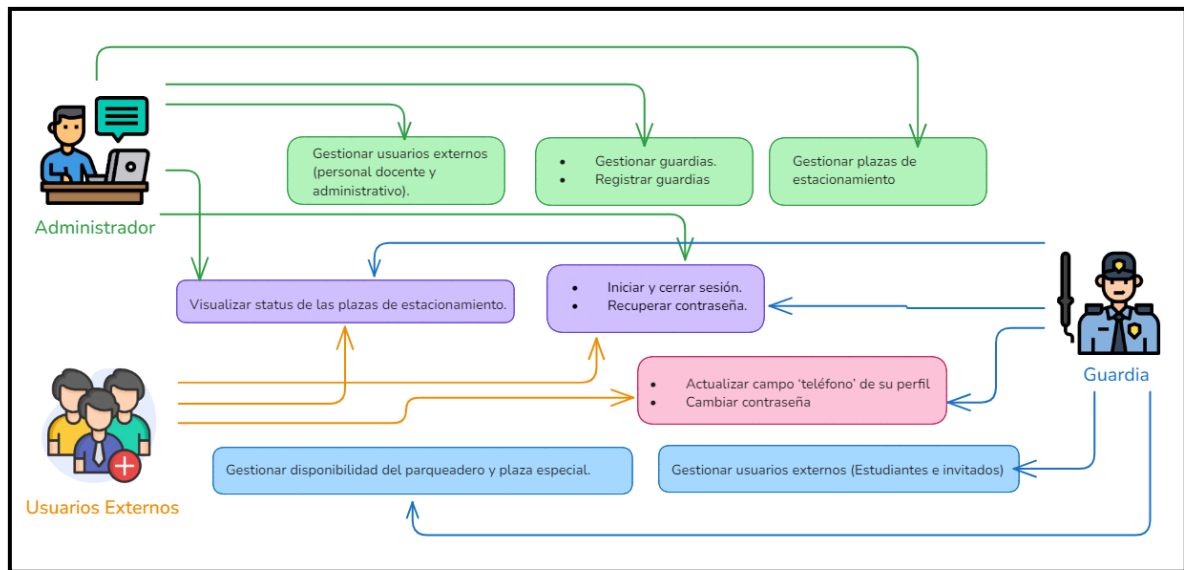


Figura 3.2 Acciones por roles.

Sprint 1 Rol Administrador y creación de interfaces.

Este Sprint se enfoca en describir cada una de las tareas que se han codificado para el usuario administrador en conjunto con el resultado de la prueba unitaria. Por lo anterior dicho, las tareas que abarca este Sprint son las siguientes.

- Diseño de la landing page de inicio.
- Diseño del dashboard general.
- Consumo de endpoints para inicio de sesión.
- Consumo de endpoints para cambiar contraseña.
- Consumo de endpoints para gestionar guardias.
- Consumo de endpoints para gestionar usuarios externos (personal docente y administrativos).
- Consumo de endpoints para gestionar plazas de estacionamiento.

Diseño de la landing page de inicio

Se ha diseñado una landing page inicial donde todos los usuarios pueden visualizar información detallada sobre el negocio, su propuesta de valor, servicios que ofrece, métodos de contacto disponibles y otros aspectos. Además, esta página cuenta con un menú superior que incluye el logotipo del proyecto y un botón principal para iniciar sesión tal y como se muestra en la **Figura**

3.3, mientras que en el **ANEXO III** se ofrece una explicación a detalle de todas las demás secciones y funcionalidades.



Figura 3.3 Landing page de inicio.

Diseño del dashboard general

Se ha diseñado un layout específico tipo dashboard para todos los usuarios, el cual facilita la navegación eficiente entre los distintos módulos disponibles. Además, este menú incluye opciones que permiten al usuario acceder rápidamente a las funcionalidades correspondientes a su rol, así como un acceso directo para realizar el cierre de sesión tal y como se muestra en la **Figura 3.4**, acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.5** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.



Figura 3.4 Dashboard y layout general.

```
✓ src/__tests__/layouts/SideBar.test.jsx (3)
✓ Layouts con AuthProvider y Router (3)
  ✓ AdminLayout se renderiza correctamente cuando el usuario es administrador
  ✓ GuardiaLayout se renderiza correctamente cuando el usuario es guardia
  ✓ UsuarioLayout se renderiza correctamente cuando el usuario es un usuario externo

Test Files 1 passed (1)
Tests 3 passed (3)
Start at 12:10:32
Duration 378ms

PASS Waiting for file changes...
```

Figura 3.5 Prueba unitaria para el dashboard y layout general.

Consumo de endpoints para inicio de sesión

El administrador por medio de una interfaz puede realizar un inicio de sesión en el backend proporcionando información como dirección de correo, clave de acceso, selección del rol y tras pasar exitosamente las validaciones correspondientes, el usuario obtiene acceso a las funcionalidades administrativas que han sido especialmente asignadas a su cargo tal y como se muestra en la **Figura 3.6** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.7**, mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad. Por último, cabe destacar que esta funcionalidad es la misma para el usuario guardia y usuarios externos (personal docente, administrativos, estudiantes e invitados).

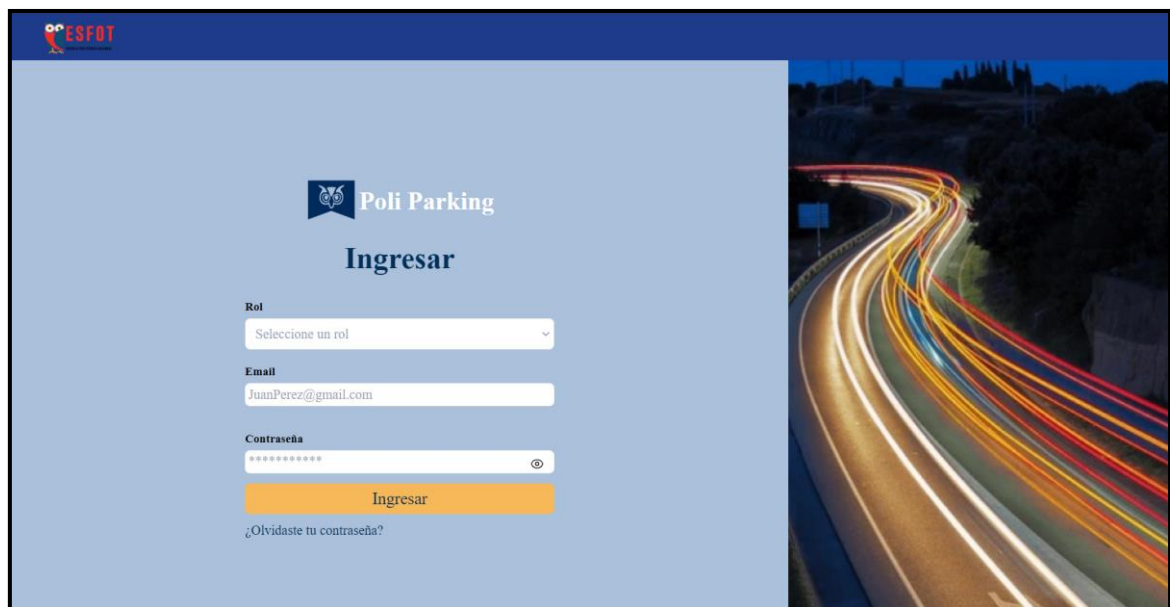


Figura 3.6 Formulario de inicio de sesión.

```
✓ src/__tests__/pages/Login.test.jsx (2) 1162ms
✓ Login Component (2) 1162ms
  ✓ maneja el envío exitoso del formulario 617ms
  ✓ maneja errores en el envío del formulario 544ms

Test Files 1 passed (1)
Tests 2 passed (2)
Start at 12:12:20
Duration 1.81s

PASS Waiting for file changes...
```

Figura 3.7 Prueba unitaria del formulario de inicio de sesión.

Consumo de endpoints para cambiar contraseña

El usuario por medio de una interfaz posee la funcionalidad de cambiar su contraseña cuando lo requiera promoviendo de esta manera buenas prácticas de seguridad sobre la protección de su información personal y módulos que han sido asignados tal y como se muestra en la **Figura 3.8** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.9** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad. Por último, cabe destacar que esta funcionalidad es la misma para el usuario guardia y usuarios externos (personal docente, administrativos, estudiantes e invitados).

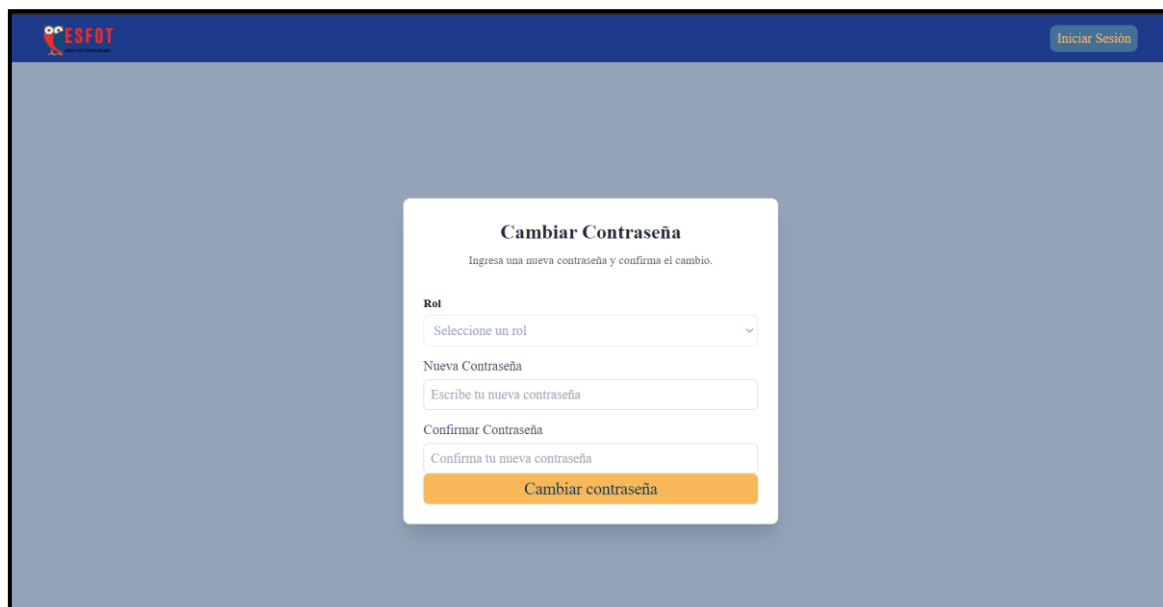


Figura 3.8 Formulario para cambiar contraseña.

```
✓ src/__tests__/pages/ChangePassword.test.jsx (3) 1803ms
✓ ChangePassword Component (3) 1802ms
  ✓ valida que las contraseñas coincidan 470ms
  ✓ maneja correctamente el cambio de contraseña para un usuario 664ms
  ✓ maneja correctamente los errores del servicio 667ms

Test Files 1 passed (1)
Tests 3 passed (3)
Start at 12:14:33
Duration 2.15s

PASS Waiting for file changes...
```

Figura 3.9 Prueba unitaria para el formulario cambiar contraseña.

Consumo de endpoint para gestionar guardias

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad para la gestión de usuarios con perfil guardia. Además, cabe destacar que la gestión implica registrar, eliminar y listar guardias en estado activo permitiendo de esta manera tomar decisiones informadas y administrar eficientemente el personal de seguridad tal y como se muestra en la **Figura 3.10** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.11** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.

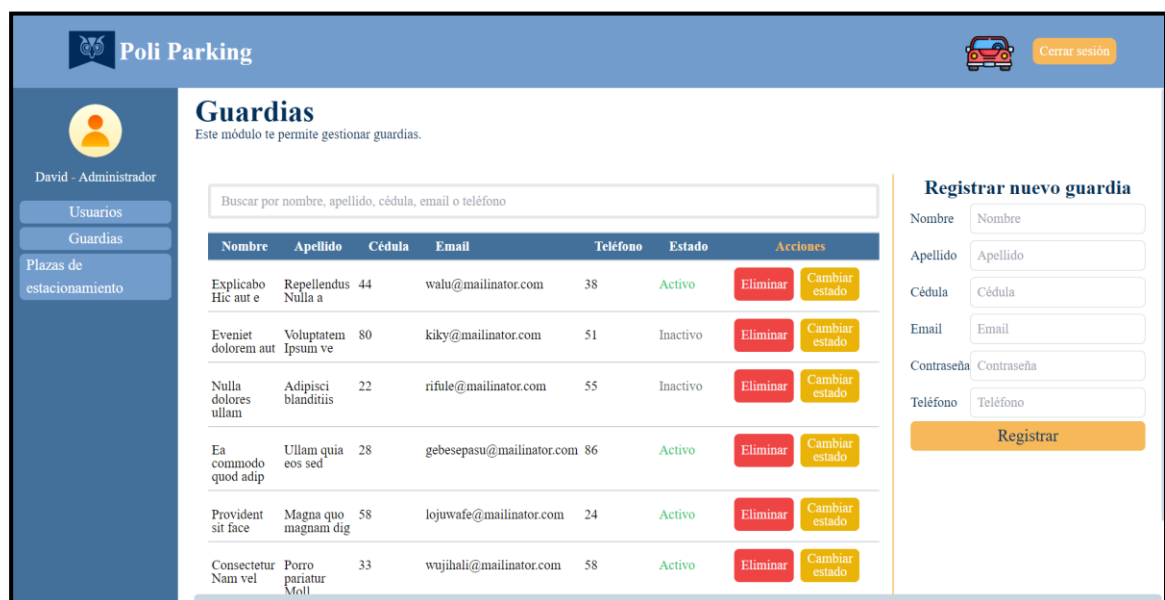


Figura 3.10 Módulo guardias para el administrador.


```
✓ src/__tests__/components/Perfil/RegistroGuardia.test.jsx (5)
✓ RegistroGuardia - Administrador (5)
  ✓ se renderiza correctamente el formulario
  ✓ muestra error si algún campo está vacío
  ✓ llama a adminService.createGuardia con los valores correctos al registrar
  ✓ muestra mensaje de éxito después de crear un guardia correctamente
  ✓ muestra mensaje de error si falla la creación del guardia

Test Files  1 passed (1)
Tests       5 passed (5)
Start at    08:47:41
Duration    781ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 3.11 Pruebas unitarias para el módulo guardias del administrador.

Consumo de endpoints para gestionar usuarios externos (personal docente y administrativos)

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad para la gestión de usuarios externos (personal docente y administrativos) lo que le permite llevar un control y supervisión eficaz de las cuentas que se han registrado en el backend. Además, cabe destacar que la gestión implica registrar, eliminar, actualizar y listar usuarios externos permitiendo de esta manera tomar decisiones informadas y aplicar todos los controles de seguridad necesarios tal y como se muestra en la **Figura 3.12** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.13** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.

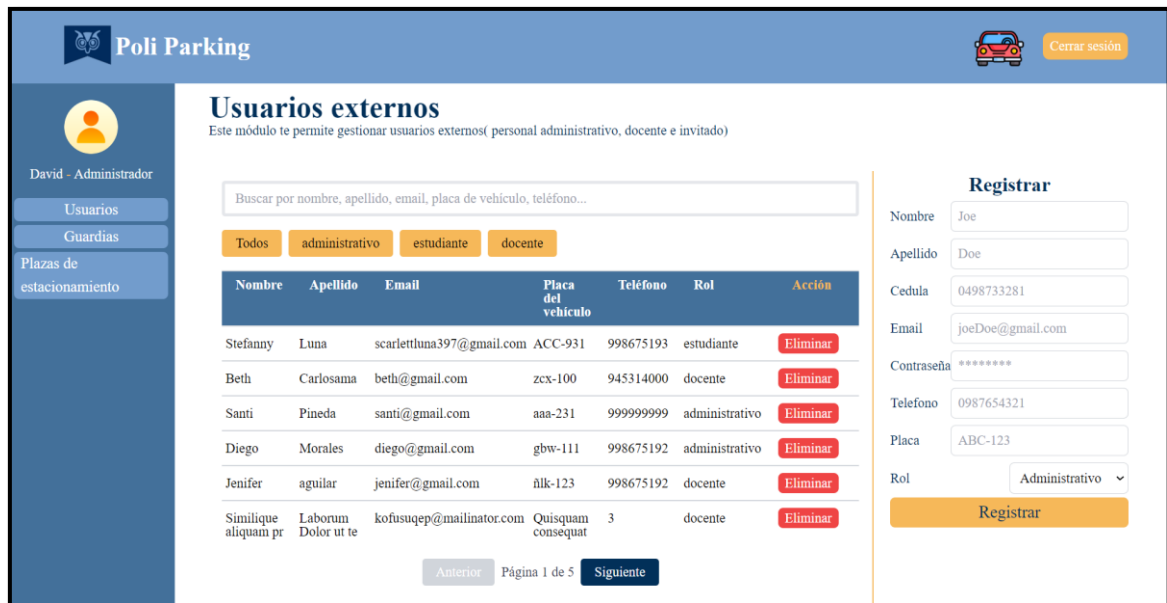


Figura 3.12 Módulo usuarios externos del administrador.

```

✓ src/__tests__/components/RegisterAdminUser.test.jsx (2)
  ✓ RegisterAdminUser (2)
    ✓ se renderiza correctamente
    ✓ se envia los datos correctamente

Test Files 1 passed (1)
Tests      2 passed (2)
Start at   09:02:17
Duration   912ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 3.13 Pruebas unitarias del módulo usuarios externos del administrador.

Consumo de endpoints para gestionar plazas de estacionamiento

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad para la gestión de plazas de estacionamiento. Además, cabe destacar que la gestión implica registrar, eliminar y listar plazas de estacionamiento lo que le permite al administrador tener una visión global y en tiempo real de la información para una gestión integral de los espacios disponibles tal y como se muestra en la **Figura 3.14** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.15** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.

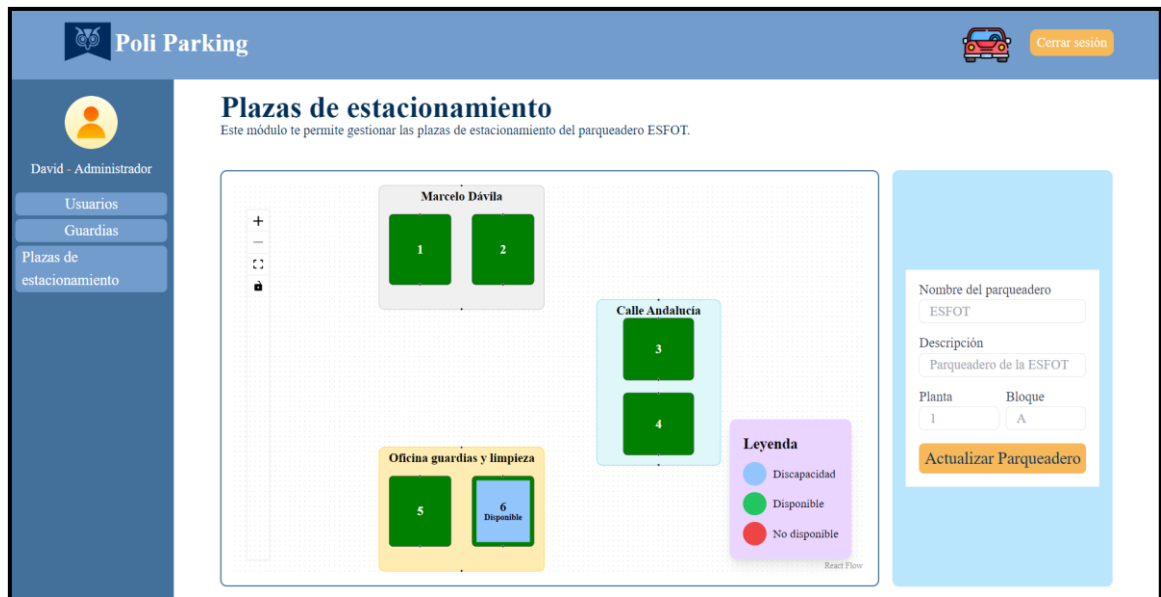


Figura 3.14 Módulo plazas de estacionamiento del administrador.

```

✓ src/__tests__/pages/Administrador/ParqueaderosAdminPage.test.jsx (3)
✓ ParqueaderosAdminPage (3)
  ✓ muestra mensaje cuando no hay parqueaderos
  ✓ renderiza el formulario de actualización con todos sus campos
  ✓ muestra mensaje de éxito después de actualizar

Test Files  1 passed (1)
Tests       3 passed (3)
Start at    12:18:00
Duration    712ms

PASS Waiting for file changes...

```

Figura 3.15 Pruebas unitarias del módulo de plazas de estacionamiento del administrador.

Sprint 2 Rol Guardia y creación de interfaces.

Este Sprint se enfoca en describir cada una de las tareas que se han codificado para el usuario guardia en conjunto con el resultado de la prueba unitaria. Por lo anterior dicho, las tareas que abarca este Sprint son las siguientes.

- Consumo de endpoints para gestionar perfil.
- Consumo de endpoints para visualizar el status de la plaza de estacionamiento.
- Consumo de endpoints para gestionar usuarios externos (estudiantes e invitados).

Consumo de endpoints para gestionar perfil

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad para la gestión de su información personal. Además, cabe destacar que la gestión implica visualizar y editar su perfil, de esta manera la visualización permite al guardia revisar detalladamente su información personal, credenciales y otros datos relevantes, asegurando transparencia y exactitud en el registro, mientras que la función de edición otorga al guardia la autonomía de actualizar ciertos campos de su perfil tal y como se muestra en la **Figura 3.16** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.17** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.

Poli Parking

David - Guardia

Perfil

Usuarios

Plazas de estacionamiento

Perfil

Este módulo te permite gestionar tu perfil personal.

David Vela

111111111111

guardia@guardia.com

998675192

Telefono
0999999999

Actualizar teléfono

Contraseña actual

Nueva contraseña

Actualizar contraseña

Cerrar sesión

Figura 3.16 Módulo perfil del guardia.

```
✓ src/__tests__/components/FormularioG.test.jsx (3)
  ✓ FormularioG (3)
    ✓ llama a fetchPut con los valores correctos
    ✓ muestra alerta de éxito después de actualizar
    ✓ muestra alerta de error si ocurre un problema

Test Files  1 passed (1)
Tests       3 passed (3)
Start at    13:11:54
Duration    361ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 3.17 Pruebas unitarias del módulo perfil del guardia.

Consumo de endpoints para visualizar el status de la plaza de estacionamiento

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad de visualizar en tiempo real la disponibilidad de las plazas de estacionamiento, facilitando de esta manera la gestión del flujo de vehículos y la optimización del uso de espacios disponibles. Además, esta interfaz permite una rápida toma de decisiones para dirigir a los usuarios externos a los espacios libres y mantener el flujo eficiente de vehículos entrantes y salientes tal y como se muestra en la **Figura 3.18** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.19** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad. Por último, cabe destacar que esta funcionalidad es la misma para el usuario externo (personal docente, administrativos, estudiantes e invitados).



Figura 3.18 Módulo plazas de estacionamiento del guardia.

```

✓ src/__tests__/pages/Guardias/ParqueaderosGuardiasPage.test.jsx (3)
✓ ParqueaderosGuardiasPage (3)
  ✓ puede cambiar el estado general del parqueadero
  ✓ puede cambiar el estado del espacio para discapacitados
  ✓ muestra mensaje cuando no hay parqueaderos

Test Files 1 passed (1)
Tests      3 passed (3)
Start at   16:47:38
Duration   561ms

PASS Waiting for file changes...

```

Figura 3.19 Pruebas unitarias del módulo plazas de estacionamiento del guardia.

Consumo de endpoints para gestionar usuarios externos (estudiantes e invitados)

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad para la gestión de usuarios externos (estudiantes e invitados) lo que le permite llevar un control y supervisión eficaz de las cuentas que se han registrado en el backend. Además, cabe destacar que la gestión implica registrar, eliminar, actualizar y listar usuarios externos permitiendo de esta manera tomar decisiones informadas y aplicar todos los controles de seguridad necesarios tal y como se muestra en la **Figura 3.20** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.21** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.

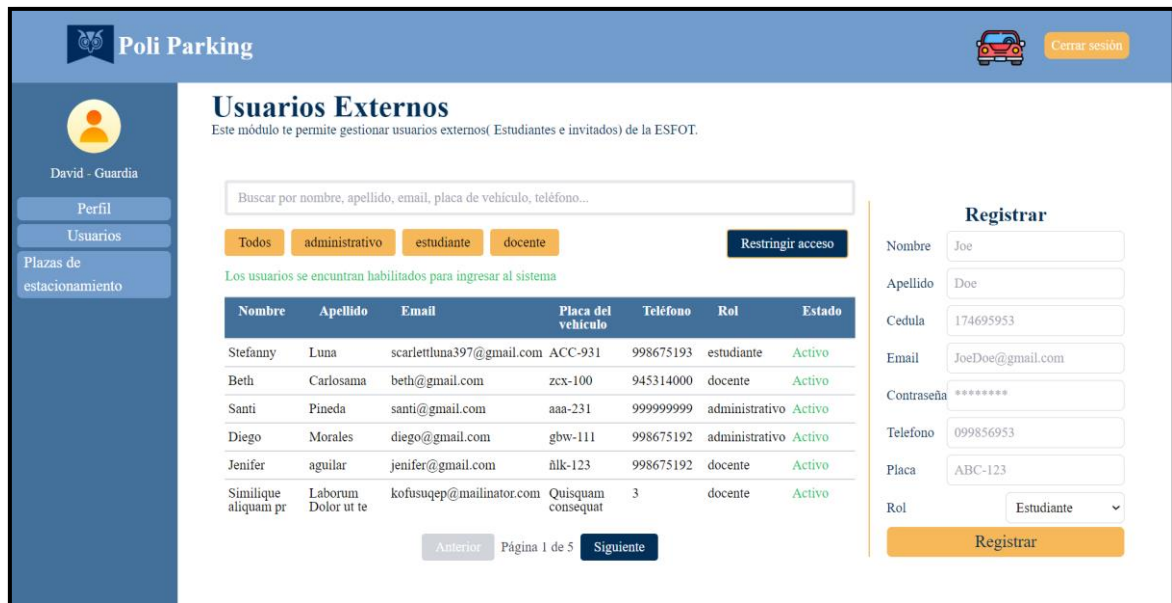


Figura 3.20 Módulo usuarios externos del guardia.

```

✓ src/__tests__/pages/Guardias/UsuariosGuardiasPage.test.jsx (4)
✓ UsuariosGuardiasPage (4)
  ✓ se renderiza correctamente
  ✓ maneja la búsqueda y filtrado correctamente
  ✓ maneja la paginación correctamente
  ✓ maneja la filtración por rol correctamente

Test Files 1 passed (1)
Tests 4 passed (4)
Start at 12:39:02
Duration 778ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 3.21 Pruebas unitarias del módulo usuarios externos del guardia.

Sprint 3 Rol usuario externo y creación de interfaces.

Este Sprint se enfoca en describir la única tarea que se ha codificado para el usuario externo (personal docente, administrativos, estudiantes e invitados) en conjunto con el resultado de la prueba unitaria. En ese sentido, la tarea que abarca este Sprint es el consumo de endpoints para para gestionar su perfil personal ya que las demás tareas se han mencionado previamente en el Sprint 1 y 2 al ser tareas que se comparten con los demás usuarios.

Consumo de endpoints para gestionar el perfil personal

El usuario por medio de una interfaz se encuentra equipado con la funcionalidad para la gestión de su información personal. Además, cabe destacar que la gestión implica por una parte la visualización de información personal y detalle de su vehículo y, por otra parte, la posibilidad de editar sus datos en el caso de que lo requieran, sin embargo, una restricción importante es que no puede modificar la placa vehicular lo cual constituye una medida de seguridad para preservar la funcionalidad completa del frontend tal y como se muestra en la **Figura 3.22** y acto seguido se ha realizado una prueba unitaria para verificar la funcionalidad del código cuyo resultado se detalla en la **Figura 3.23** mientras que en el **ANEXO III** se ofrece una explicación a detalle de toda la funcionalidad.

Figura 3.22 Módulo perfil del usuario externo.

```
✓ src/__tests__/components/FormularioU.test.jsx (3)
✓ FormularioU (3)
  ✓ muestra error si el campo está vacío
  ✓ llama a fetchPut con los valores correctos
  ✓ muestra alerta de éxito después de actualizar

Test Files 1 passed (1)
Tests      3 passed (3)
Start at   16:49:21
Duration   326ms

PASS Waiting for file changes...
```

Figura 3.23 Pruebas unitarias del módulo perfil del usuario externo.

Sprint 4 Pruebas para el componente frontend.

Este Sprint describe todas las tareas que están relacionadas con las pruebas del frontend, las cuales abarcan.

- Unitarias.
- Compatibilidad.
- Rendimiento.
- Aceptación.

Pruebas unitarias

Este tipo de pruebas en el campo del software se centran específicamente en evaluar componentes individuales o porciones de código, con el objetivo de asegurar que cada uno funcione correctamente de manera aislada y que luego no exista inconvenientes cuando se integren con los demás componentes. De esta manera, se logra identificar errores en etapas tempranas del desarrollo lo que es crucial para mantener la calidad del código [37]. En la **Figura 3.24** se ilustra un ejemplar de la prueba y en la **Figura 3.25** el resultado que la herramienta ha otorgado, mientras que en el **ANEXO II** se presentan los resultados restantes.

| Buscar por nombre, apellido, email, placa de vehículo, teléfono... | | | | | | |
|--|---------------------|---------------------------|--------------------|-----------|----------------|----------|
| Todos | administrativo | estudiante | docente | | | |
| Nombre | Apellido | Email | Placa del vehículo | Teléfono | Rol | Acción |
| Stefanny | Luna | scarlettluna397@gmail.com | ACC-931 | 998675193 | estudiante | Eliminar |
| Beth | Carlosama | beth@gmail.com | zcx-100 | 945314000 | docente | Eliminar |
| Santi | Pineda | santi@gmail.com | aaa-231 | 999999999 | administrativo | Eliminar |
| Diego | Morales | diego@gmail.com | gbw-111 | 998675192 | administrativo | Eliminar |
| Jenifer | aguilar | jenifer@gmail.com | ñlk-123 | 998675192 | docente | Eliminar |
| Similique aliquam pr | Laborum Dolor ut te | kofusuqep@mailinator.com | Quisquam consequat | 3 | docente | Eliminar |
| Anterior | | Página 1 de 5 | | Siguiente | | |

Figura 3.24 Componente DataTable usado para mostrar información.

```

✓ src/___tests___/components/common/DataTable.test.jsx (6)
✓ DataTable component (6)
  ✓ Renderizar "No hemos encontrado coincidencias" cuando los datos están vacíos
  ✓ muestra una tabla con datos, columnas y acciones válidos
  ✓ muestra una tabla con datos y columnas, pero sin acciones
  ✓ representa los valores booleanos de los datos como "Activo" o "Inactivo"
  ✓ muestra los valores de cadena de los datos tal cual
  ✓ llama al controlador onClick cuando se hace clic en el botón de acción

Test Files 1 passed (1)
Tests 6 passed (6)
Start at 09:32:12
Duration 312ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 3.25 Prueba unitaria del componente DataTable.

Los resultados de esta prueba han permitido corroborar que todos los componentes individuales del frontend funcionen adecuadamente como se esperaba y que al momento de integrar con los demás no presente fallos a nivel de código y funcionalidad.

Pruebas de compatibilidad

Este tipo de pruebas en el campo del software tienen como objetivo verificar que el software funcione adecuadamente en distintos navegadores de los dispositivos electrónicos. Además, estas pruebas son esenciales para asegurar una experiencia de usuario consistente y accesible para todos [38]. Por lo antes expuesto, en la **Tabla 3.1** Prueba de compatibilidad realizada en diferentes exploradores web se incluye los principales navegadores para ejecutar la prueba y en la **Figura 3.26** se ilustra el resultado que se ha obtenido y en el **ANEXO II** se ofrece un resumen completo de los resultados que se han obtenido por cada navegador web.

Tabla 3.1 Prueba de compatibilidad realizada en diferentes exploradores web

| EXPLORADOR WEB | EDICIÓN | NOTA |
|----------------|----------------|--------------------------------|
| Brave | v1.73.104 | Funcionando sin inconvenientes |
| Chromium | 134.0.6952.0 | Funcionando sin inconvenientes |
| Microsoft Edge | 131.0.2903.146 | Funcionando sin inconvenientes |

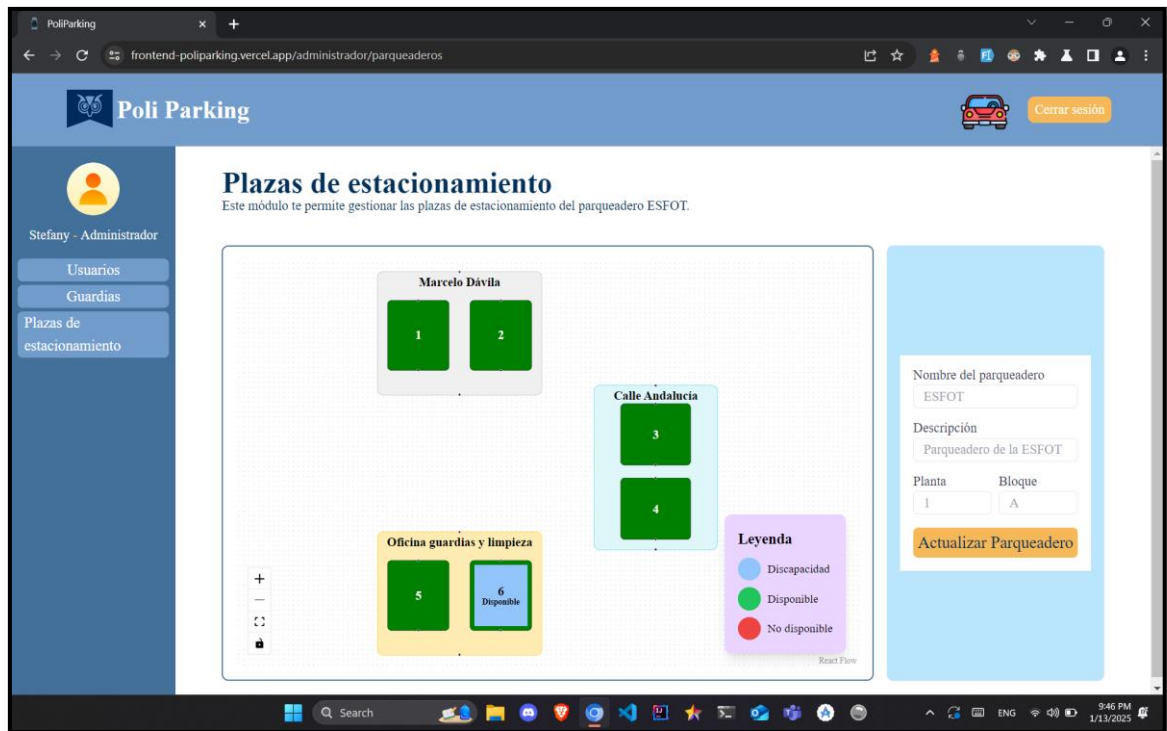


Figura 3.26 Prueba de compatibilidad en el navegador Chromium.

Los resultados de esta prueba han permitido demostrar que todas las interfaces son completamente funcionales en múltiples navegadores web ya sea a nivel de información o contenido multimedia, lo que garantiza una buena experiencia de usuario.

Pruebas de rendimiento

Este tipo de pruebas en el campo del software permiten evaluar cómo responde un sistema software bajo diversas condiciones de carga, analizando factores tales como tiempo de carga, respuesta, detalle descriptivo, etc. Además, estas pruebas simulan escenarios de carga intensa para garantizar que el software pueda manejar un gran número de usuarios simultáneamente sin comprometer el rendimiento [39]. Por lo antes expuesto, en la **Figura 3.27** se presenta una imagen que ilustra los resultados de las métricas de rendimiento, mientras que en el **ANEXO II** se detalla un análisis completo de las restantes pruebas.

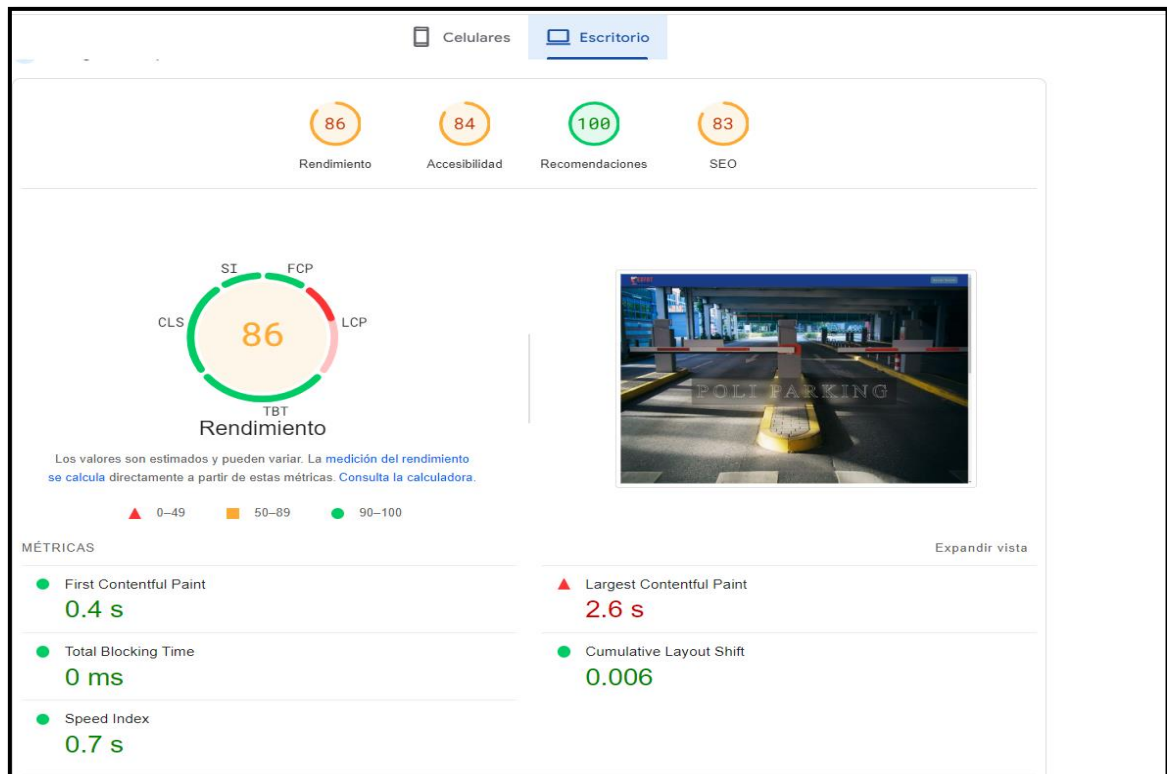


Figura 3.27 Prueba de rendimiento.

Esta prueba ha permitido confirmar que el frontend mantiene un rendimiento óptimo y una estabilidad consistente, incluso bajo altas cargas de trabajo lo que garantiza que el frontend puede manejar sin problema alguno varias peticiones de manera concurrente.

Pruebas de aceptación

Este tipo de pruebas en el campo del software permiten determinar si el software cumple con los requisitos que se han establecido por el cliente o usuario final. Además, este tipo de prueba es clave para validar que todas las funcionalidades se encuentren implementadas correctamente y satisfacen las expectativas iniciales y del negocio [40]. Por lo antes expuesto, en la **Tabla 3.2** Prueba de aceptación para la historia de usuario HU-002 se incluye el formato de prueba de aceptación, pasos y criterios que han sido determinados para su aceptación. Mientras que en el **ANEXO II** se lista las demás pruebas basadas en el mismo formato.

Tabla 3.2 Prueba de aceptación para la historia de usuario HU-002.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|--|------------------------------------|
| Identificador de la prueba: PA-002 | historia de Usuario: HU-002 |
| Nombre de la prueba: Iniciar sesión | |
| <p>Descripción: El frontend creará un módulo funcional de login para:</p> <ul style="list-style-type: none"> • Ingresar al sistema según el rol del usuario. • Persistir datos en el navegador • Proteger rutas según el rol de usuario. • Recuperar contraseña. | |
| <p>Fases de realización:</p> <p>Para iniciar sesión todos los usuarios:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • En lado derecho superior dar clic en Iniciar sesión. • Llenar los campos del rol, correo electrónico y contraseña. • Si alguno de estos campos falta o elige otro rol tendrá un mensaje de error • Dar clic en “Iniciar Sesión”. • Ingresar a su dashboard <p>Para salir de la sesión los usuarios:</p> <ul style="list-style-type: none"> • Haber iniciado sesión en el componente frontend. • En el lado derecho superior dar clic en “Cerrar sesión” <p>Para recuperar contraseña:</p> <ul style="list-style-type: none"> • Ingresar a la URL del frontend en un navegador. • En lado derecho superior dar clic en “Iniciar Sesión”. • Dar clic en “¿Olvidaste tu contraseña?”. • Llenar el formulario con tu correo electrónico. | |

| |
|---|
| <ul style="list-style-type: none"> • Confirmar el correo enviado a tu correo. • Llenar el formulario respectivo para el cambio de contraseña. • Dar clic en “Cambiar contraseña” |
| <p>Resultado deseado:</p> <p>El componente de frontend simplifica las acciones de inicio y cierre de sesión, además de permitir a los usuarios recuperar su contraseña.</p> |
| <p>Análisis de la prueba:</p> <p>El resultado cumple con las expectativas y el cliente expresa su plena satisfacción al 100%.</p> |

Estas pruebas han permitido determinar que todo el componente frontend cumple a cabalidad con todos los requisitos funcionales y no funcionales, garantizando de esta manera que sea de gran utilidad para el negocio, usuarios finales y para su lanzamiento a producción.

Sprint 5 Despliegue del frontend.

Este último Sprint se destaca principalmente por el despliegue a producción del frontend una vez que se han validado y aprobado cada una de las pruebas. En ese sentido, el frontend ha sido desplegado en la plataforma Vercel, el cual es una plataforma diseñada para equipos multifuncionales que ofrece la posibilidad de construir y desplegar sistemas software optimizando de esta manera tiempo y recursos [41]. Así, la **Figura 3.28** presenta el repositorio del proyecto en GitHub mientras que la **Figura 3.29** presenta el despliegue en la plataforma que se ha detallado previamente.

Por último, para acceder al frontend en producción se lo puede acceder mediante el siguiente enlace.

<https://frontend-poliparking.vercel.app/>

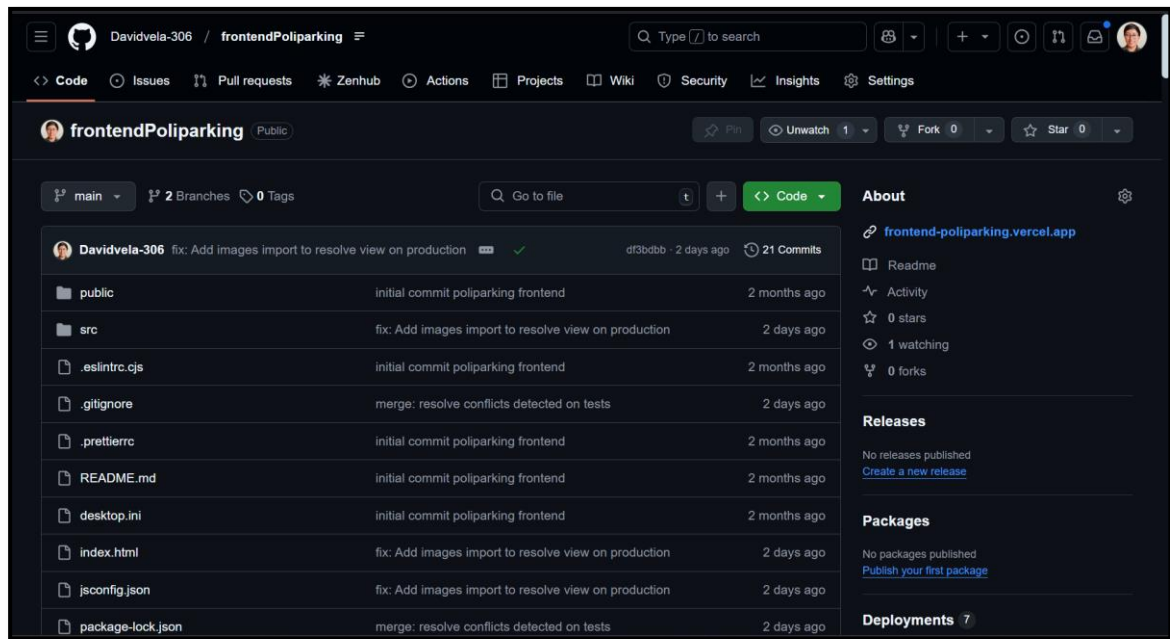


Figura 3.28 Repositorio de GitHub.

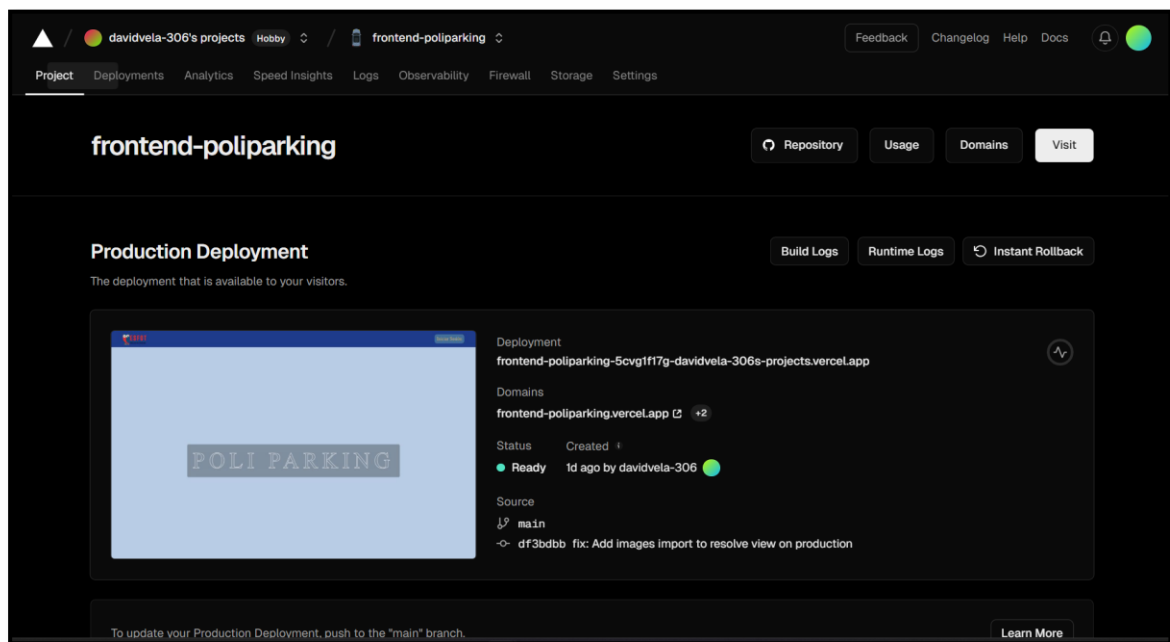


Figura 3.29 Despliegue en Vercel.

4 CONCLUSIONES

A continuación, se presenta cada una de las conclusiones que se han obtenido una vez que se ha finalización por completo el proceso de planificación, codificación, pruebas y despliegue del frontend.

- Para lograr un desarrollo con éxito, es esencial tener una comprensión profunda de la lógica del negocio, así como de las necesidades y expectativas de los usuarios. En este contexto, gracias a una buena recopilación de requisitos se ha logrado implantar un frontend que no solo cumple con los objetivos que se han establecido, sino que también ofrece una escalabilidad de requisitos funcionales y no funcionales a futuro.
- Una planificación cuidadosa al momento de diseñar interfaces de usuario se garantiza que las mismas aporten con funcionales mínimas, sean accesibles, intuitivas y atractivas para los usuarios finales. Es así que, gracias a una buena planificación se ha logrado codificar un conjunto de interfaces que no solo cubran la experiencia del usuario, si no también requisitos técnicos y aporte a la lógica del negocio.
- En la etapa de codificación al considerar aspectos tales como herramientas librerías, patrón de arquitectura, autenticación basada en roles, protección de rutas, prueba unitaria de componentes, consumo de endpoints (públicos y privados), entre otros, se ha logrado con éxito desarrollar un frontend que no solo garantiza la seguridad y funcionalidad, sino que permite una escalabilidad gracias a una arquitectura modular y basada en componentes.
- Realizar pruebas exhaustivas antes de su lanzamiento a producción es un aspecto crítico en el desarrollo de sistemas software. Es así que, gracias a un conjunto de pruebas se ha logrado de forma temprana identificar y corregir errores potenciales errores que afecten la experiencia de los usuarios finales, la funcionalidad de módulos asociados, así como la operatividad y calidad del frontend.
- Gracias al despliegue del componente frontend a un entorno de producción, el mismo no solo posibilita su uso y difusión dentro del parqueadero principal de la ESFOT, sino también que el mismo, pueda ser de utilidad en todo el campus de la Escuela Politécnica Nacional.

5 RECOMENDACIONES

A continuación, se presenta cada una de las recomendaciones, las cuales pueden servir de guía para futuros proyectos similares.

- Es recomendable usar diversas técnicas como observaciones, entrevistas, encuestas y talleres para la obtención de requisitos, ya que una buena recolección de requisitos no solo reduce el riesgo de malentendidos, sino que también establece una base sólida para el desarrollo de funcionalidades.
- Es recomendable realizar un plan de actualización de paquetes, ya que los mismos con el pasar del tiempo suelen recibir por parte de la comunidad parches de seguridad y otros aspectos que permiten la integridad y el robo de información.
- Es recomendable realizar una buena elección sobre la arquitectura de software, ya que la misma debe brindar compatibilidad y sobre todo escalabilidad a futuro. Por ende, es aconsejable optar por arquitecturas basada en componentes ya que las mismas no solo mejoran el mantenimiento, sino que permiten una rápida adaptación a los cambios tecnológicos o requisitos del usuario.
- Es recomendable seleccionar herramientas y librerías que no solo permitan optimizar el flujo de trabajo, sino que el equipo de desarrollo se encuentre familiarizado con las mismas, ya que esto aumenta la eficiencia y reduce la curva de aprendizaje.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Escuela de Formación de Tecnólogos, «¿Quiénes Somos?,» [En línea]. Available: <https://esfot.epn.edu.ec/index.php/quienes-somos>. [Último acceso: 05 10 2024].
- [2] Conure, «Seven Ways IoT Technology Is Helping to Create Smart Campuses - Conure,» 01 08 2022. [En línea]. Available: <https://www.conurets.com/seven-ways-iot-technology-is-helping-to-create-smart-campuses/>. [Último acceso: 04 10 2024].
- [3] asana, «La guía rápida para definir el alcance de tu proyecto en 8 pasos,» 17 02 2024. [En línea]. Available: <https://asana.com/es/resources/project-scope>. [Último acceso: 04 10 2024].
- [4] I. J. B. García, «Backend y Frontend, ¿Qué es y cómo funcionan en la programación?,» servnet, 30 03 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>. [Último acceso: 04 10 2024].
- [5] Coursera Staff, «Diseño UI vs. UX: ¿Cuál es la diferencia?,» coursera, 29 11 2023. [En línea]. Available: <https://www.coursera.org/mx/articles/ui-vs-ux-design>. [Último acceso: 04 10 2024].
- [6] Amazon Web Services, Inc., «¿Cuál es la diferencia entre HTTP y HTTPS?,» 2023. [En línea]. Available: <https://aws.amazon.com/es/compare/the-difference-between-https-and-http/#:~:text=Los%20mensajes%20HTTP%20son%20de,los%20datos%20de%20forma%20cifrada>. [Último acceso: 04 10 2024].
- [7] «¿Qué es un navegador web?,» firefox, [En línea]. Available: <https://www.mozilla.org/es-ES/firefox/browsers/what-is-a-browser/>. [Último acceso: 04 10 2024].
- [8] «Conceptos básicos de puntos finales de API para líderes empresariales,» intuit mailchimp, [En línea]. Available: <https://mailchimp.com/es/resources/what-is-an-api-endpoint/#:~:text=El%20punto%20final%20es%20la,otras%20empresas%2C%20aplicaciones%20y%20sistemas>. [Último acceso: 05 10 2024].
- [9] MDN contributors, «Acerca de JavaScript,» MDN, 08 08 2023. [En línea]. Available: <https://developer.mozilla.org/es/docs/conflicting/Web/JavaScript>. [Último acceso: 04 10 2024].
- [10] S. L. Mora, «digital55,» 28 11 2023. [En línea]. Available: <https://digital55.com/blog/desarrollo-react-ventajas-contratar-programador-react/>. [Último acceso: 04 10 2024].
- [11] React Redux, «Why Use React Redux?,» 19 06 2021. [En línea]. Available: <https://react-redux.js.org/introduction/why-use-react-redux>. [Último acceso: 05 10 2024].
- [12] React v18.3.1 Documentation, «Hooks integrados en React,» 2024. [En línea]. Available: <https://es.react.dev/reference/react/hooks>. [Último acceso: 05 10 2024].

- [13] R. v. Documentation, «Tu primer componente,» 2024. [En línea]. Available: <https://es.react.dev/learn/your-first-component>. [Último acceso: 05 10 2024].
- [14] R. STAKE, de *Investigación con estudio de casos*, Madrid, Ediciones Morata, S. L, 1995, pp. 11-15.
- [15] O. Tinoco Gómez, P. P. Rosales López y J. Salas Bacalla, «Criterios de selección de metodologías de desarrollo de software,» *Revista de la Facultad de Ingeniería Industrial - Universidad Nacional Mayor de San Marcos*, vol. 13, pp. 70-74, 2010.
- [16] C. I. RIVAS, V. P. CORONA, J. F. GUTIÉRREZ y L. HERNÁNDEZ, «Metodologías actuales de desarrollo de software,» *Revista de Tecnología e Innovación*, vol. 10, nº 11, pp. 980-986, 2015.
- [17] J. Martins, «asana,» Asana, Inc., 15 02 2024. [En línea]. Available: <https://asana.com/es/resources/what-is-scrum>. [Último acceso: 12 10 2024].
- [18] «Definición de 'Rol',» Hrider, [En línea]. Available: <https://www.hrider.net/es/diccionario/rol.html>. [Último acceso: 12 10 2024].
- [19] S. M. y. C. D. A. Mundra, «Practical Scrum-Scrum Team: Way to Produce Successful and Quality Softwar,» de *Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software. Proceedings of the 2013 13th International Conference on Computational Science and Its Applications, ICCSA 2013. 119-123. 10.1109/ICCSA.2013.25.*, India, 2013.
- [20] Miro, «Artefactos Scrum: ¿qué son? ¿Para qué sirven?,» Miro, [En línea]. Available: <https://miro.com/es/agile/que-son-artefactos-scrum/>. [Último acceso: 12 10 2024].
- [21] J. D. Y. W. y. M. A. A. H. Saeeda, «A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project,» *Journal of Software: Evolution and process*, p. 2, 2019.
- [22] F. & B. S. Dalpiaz, «Agile Requirements Engineering with User Stories,» de *2018 IEEE 26th International Requirements Engineering Conference (RE)*, Utrecht, 2018.
- [23] T. Sedano, P. Ralph y C. Péraire, *The Product Backlog*, 2019.
- [24] «Qué es un backlog de sprint [y cómo crear uno] | Atlassian,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/sprint-backlog#:~:text=Un%20backlog%20de%20sprint%20mantiene,tareas%20que%20hay%20que%20hacer..> [Último acceso: 12 10 2024].
- [25] S. Morejón, «Principios del proceso de Diseño de Interfaz de Usuario,» *Revista Cubana de transformación digital*, vol. 1, nº 3, pp. 143-145, 2021.
- [26] T. Dey, «A Comparative Analysis on Modeling and Implementing,» de *International Conference on Web Services Computing (ICWSC) 2011*, 2011.
- [27] H. Fuentes, «Isil,» 27 10 2023. [En línea]. Available: <https://isil.pe/blog/tecnologia/mejores-herramientas-desarrollo->

- software/#¿Que_es_una_herramienta_de_Desarrollo_de_Software. [Último acceso: 12 10 2024].
- [28] S. Chacon y B. Straub, «Pro git, everything you need to know about git,» de *Pro Git*, apress, 20204, pp. 14-15.
- [29] «About GitHub and Git,» GitHub, 2024. [En línea]. Available: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>. [Último acceso: 13 10 2024].
- [30] G. C. Cuadrado, «Visual Studio Code: Editor de código para desarrolladores,» openwebinars, 2022 07 22. [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. [Último acceso: 19 10 2024].
- [31] The Chromium Projects, 2024. [En línea]. Available: <https://www.chromium.org/chromium-projects/>. [Último acceso: 13 10 2024].
- [32] Anónimo, «Librería,» Educaopen, [En línea]. Available: <https://www.educaopen.com/digital-lab/metaterminos/l/libreria>. [Último acceso: 13 10 2024].
- [33] Auth0, 2024. [En línea]. Available: <https://auth0.com/docs>. [Último acceso: 13 10 2024].
- [34] C. Deshpande, «The Best Guide to Know What Is React,» Simpli learn , 25 07 2024. [En línea]. Available: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs#:~:text=ReactJS's%20primary%20goal%20is%20to,is%20the%20JavaScript%20virtual%20DOM..> [Último acceso: 13 10 2024].
- [35] «Aplicaciones en Tiempo Real con Socket.IO,» Investigate to innovate, 2017. [En línea]. Available: <https://www.iti.es/oferta-formativa/aplicaciones-tiempo-real-socket-io/>. [Último acceso: 13 10 2024].
- [36] Reed y J. C. Guña, «Cómo usar Axios con React: La guía definitiva (2021),» 14 07 2022. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-usar-axios-con-react/>. [Último acceso: 13 10 2024].
- [37] Tamushi, «¿Qué son las pruebas unitarias de software?,» TestingIT, 11 07 2022. [En línea]. Available: <https://www.testingit.com.mx/blog/pruebas-unitarias-de-software>. [Último acceso: 19 10 2024].
- [38] A. Z. Chernyak, «Pruebas de compatibilidad: qué son, tipos, proceso, características, herramientas y mucho más.,» ZapTest, [En línea]. Available: <https://www.zaptest.com/es/pruebas-de-compatibilidad-que-son-tipos-proceso-caracteristicas-herramientas-y-mucho-mas>. [Último acceso: 19 10 2024].
- [39] Anónimo, «7 Tipos de pruebas de rendimiento,» stackscale, 17 08 2023. [En línea]. Available: <https://www.stackscale.com/es/blog/tipos-de-pruebas-de-rendimiento/>. [Último acceso: 19 10 2024].

- [40] 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: 19 10 2024].
- [41] anónimo, «Vercel: Desarrollar, Previsualizar, Enviar,» Aplyca, 26 01 2023. [En línea]. Available: <https://www.aplyca.com/blog/blog-que-es-vercel-desarrollar-previsualizar-enviar>. [Último acceso: 11 01 2025].
- [42] G. C. Cuadrado, «Visual Studio Code: Editor de código para desarrolladores,» OpenWebinars, 22 07 2022. [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. [Último acceso: 14 10 2024].
- [43] A. Richaud, «Antonio Richaud,» 24 Enero 2024. [En línea]. Available: <https://antonio-richaud.com/blog/archivo/publicaciones/10-netlify.html>.

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha utilizado para el desarrollo del frontend, 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 FRONTEND asociado al DESARROLLO DE SISTEMA PARA LA GESTIÓN DE PLAZAS DE ESTACIONAMIENTO PARA EL PARQUEADERO DE LA ESFOT elaborado por el estudiante DAVID ALEJANDRO VELA TOTOY de la carrera en DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta antiplagio "TURNITIN" para la revisión de originalidad del documento escrito producto del Trabajo de Integración Curricular indicado.

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

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

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

Atentamente,



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

ANEXO II

Recopilación de requerimientos:

En la **Tabla 1 Recopilación de requerimientos** se visualiza la recopilación de requerimientos que han sido definidos al inicio del proyecto.

Tabla 1 Recopilación de requerimientos.

| RECOPIACIÓN DE REQUERIMIENTOS | | |
|-------------------------------|---------|--|
| TIPO DE SISTEMA | ID - RR | ENUNCIADO DEL ITEM |
| Frontend | RR-002 | Para el usuario administrador, guardia y usuario se requiere consumir endpoints para: <ul style="list-style-type: none">• Iniciar y cerrar sesión.• Recuperar contraseña.• Visualización del parqueadero de la ESFOT y su información. |
| | RR-003 | Para el usuario administrador se requiere consumir endpoints para: <ul style="list-style-type: none">• Gestionar guardias. |
| | RR-004 | Para el usuario administrador se requiere consumir endpoints para: <ul style="list-style-type: none">• Gestionar usuarios externos (personal docente y administrativo). |
| | RR-005 | Para el usuario administrador se requiere consumir endpoints para: <ul style="list-style-type: none">• Gestionar el parqueadero de la ESFOT. |
| | RR-006 | Para el usuario guardia requiere consumir endpoints para: <ul style="list-style-type: none">• Gestionar su perfil. |
| | RR-007 | Para el usuario guardia se requiere consumir endpoints para: <ul style="list-style-type: none">• Gestionar usuarios externos (estudiantes e invitados). |

| | | |
|--|---------------|--|
| | RR-008 | Para el usuario guardia se requiere consumir endpoints para: <ul style="list-style-type: none"> Gestionar la reserva del parqueadero y espacio de parqueo para personas discapacitadas de la ESFOT. |
| | RR-009 | Para el usuario externo se requiere consumir endpoints para: <ul style="list-style-type: none"> Gestionar su perfil, el cual solo podrá editar el campo “teléfono” en su perfil y contraseña. |
| | RR-010 | Para el usuario externo se requiere consumir endpoints para: <ul style="list-style-type: none"> Visualizar el parqueadero de la ESFOT. |

Historias de Usuario

Concluida con la etapa de recopilación de requerimientos a continuación se presentan las Historias de Usuario para el frontend las cuales van desde la **Tabla 2** Visualizar página de hasta la **Tabla 11**.

Tabla 2 Visualizar página de bienvenida.

| HISTORIA DE USUARIO | |
|--|--|
| Identificador: HU-001 | Usuario: Administrador, Guardia y Usuario |
| Nombre historia: Visualizar página de bienvenida | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Baja |
| Iteración asignada: 1 | |
| Responsable (s): Vela David | |
| Descripción: El frontend crea varias páginas para que los usuarios, guardias y usuarios puedan: Visualizar información sobre el negocio, qué solución tiene, qué servicios ofrece y cómo contactarlos. | |
| Observación: Los usuarios pueden ver la página al acceder a la url. | |

Tabla 3 Iniciar sesión.

| HISTORIA DE USUARIO | |
|--|--|
| Identificador: HU-002 | Usuario: Administrador, Guardia y Usuario |
| Nombre historia: Iniciar sesión | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Iteración asignada: 2 | |
| Responsable (s): Vela David | |
| Descripción: El frontend crea un módulo funcional de login para: <ul style="list-style-type: none"> • Ingresar al sistema según el rol del usuario. • Persistir datos en el navegador • Proteger rutas según el rol de usuario. • Recuperar contraseña. | |
| Observación: Para que los usuarios puedan ingresar al sistema deben ingresar con las credenciales correctas, en caso contrario notificarlo. | |

Tabla 4 Gestionar guardias.

| HISTORIA DE USUARIO | |
|--|------------------------------------|
| Identificador: HU-003 | Usuario: Administrador |
| Nombre historia: Gestionar guardias | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 4 | |
| Responsable (s): Vela David | |
| Descripción: El Frontend crea varios endpoints para que el administrador pueda: <ul style="list-style-type: none"> • Registrar guardia • Listar guardias • Cambiar el estado para el guardia. • Eliminar guardias | |

Observación: Para que el administrador pueda gestionar guardias debe ingresar a con las credenciales correctas.

Tabla 5 Gestionar usuarios.

| HISTORIA DE USUARIO | |
|---|------------------------------------|
| Identificador: HU-004 | Usuario: Administrador |
| Nombre historia: Gestionar usuarios | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 5 | |
| Responsable (s): Vela David | |
| Descripción: El frontend consume varios endpoints para que el administrador pueda: <ul style="list-style-type: none">• Registrar usuarios externos (Administrativo y docente).• Listar usuarios externos.• Eliminar usuarios externos. | |
| Observación: Para que el administrador pueda gestionar usuarios debe ingresar a con las credenciales correctas. | |

Tabla 6 Gestionar plazas de estacionamiento y parqueadero de la ESFOT.

| HISTORIA DE USUARIO | |
|---|------------------------------------|
| Identificador: HU-005 | Usuario: Administrador |
| Nombre historia: Gestionar plazas de estacionamiento y parqueadero de la ESFOT | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 3 | |
| Responsable (s): Vela David | |

| |
|---|
| <p>Descripción: El frontend consume varios endpoints para que el administrador pueda:</p> <ul style="list-style-type: none"> • Listar el parqueadero con sus plazas de estacionamiento. • Actualizar la información del parqueadero. |
| <p>Observación: Para que el administrador pueda gestionar plazas de estacionamiento debe ingresar a con las credenciales correctas.</p> |

Tabla 7 Gestionar perfil.

| HISTORIA DE USUARIO | |
|---|------------------------------------|
| Identificador: HU-006 | Usuario: Guardia |
| Nombre historia: Gestionar perfil | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 7 | |
| Responsable (s): Vela David | |
| <p>Descripción: El frontend consume un endpoint para que el guardia pueda:</p> <ul style="list-style-type: none"> • Detallar información de perfil. • Actualizar su dato de contactó (teléfono). • Actualizar contraseña. | |
| <p>Observación: Para que el guardia pueda gestionar su perfil debe ingresar a con sus credenciales correctas.</p> | |

Tabla 8 Gestionar usuarios externos (Estudiantes e invitados).

| HISTORIA DE USUARIO | |
|--|------------------------------------|
| Identificador: HU-007 | Usuario: Guardia |
| Nombre historia: Gestionar usuarios externos (Estudiantes e invitados) | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 8 | |
| Responsable (s): Vela David | |
| Descripción: El frontend consume un endpoint para que el guardia pueda: <ul style="list-style-type: none"> • Registrar usuarios externos (estudiantes e invitados). • Listar usuarios externos. • Editar información de usuarios externos • Restringir o permitir acceso a usuarios externos. | |
| Observación: Para que el guardia pueda gestionar plazas de estacionamiento debe ingresar a con las credenciales correctas. | |

Tabla 9 Gestionar plazas de estacionamiento.

| HISTORIA DE USUARIO | |
|---|------------------------------------|
| Identificador: HU-008 | Usuario: Guardia |
| Nombre historia: Gestionar plazas de estacionamiento | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 8 | |
| Responsable (s): Vela David | |
| Descripción: El frontend consume un endpoint para que el guardia pueda: <ul style="list-style-type: none"> • Visualizar el parqueadero de la ESFOT. | |

| |
|--|
| <ul style="list-style-type: none"> • Reservar parqueadero completo. • Reservar plaza de estacionamiento para persona con discapacidad. |
| Observación: Para que el guardia pueda gestionar plazas de estacionamiento debe ingresar a con las credenciales correctas. |

Tabla 10 Gestionar perfil.

| HISTORIA DE USUARIO | |
|--|------------------------------------|
| Identificador: HU-009 | Usuario: Usuario externo |
| Nombre historia: Gestionar perfil | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |
| Iteración asignada: 9 | |
| Responsable (s): Vela David | |
| Descripción: El frontend consume un endpoint para que el usuario pueda: <ul style="list-style-type: none"> • Listar su información de perfil. • Actualizar su dato de contacto de su perfil (teléfono). • Actualizar contraseña. • Visualizar el parqueadero de la ESFOT. | |
| Observación: Para que el usuario pueda gestionar su perfil debe ingresar a con sus credenciales correctas. | |

Tabla 11 Visualizar disponibilidad de plazas de estacionamiento.

| HISTORIA DE USUARIO | |
|--|------------------------------------|
| Identificador: HU-010 | Usuario: Usuario externo |
| Nombre historia: Visualizar disponibilidad de plazas de estacionamiento | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Media |

| |
|---|
| Iteración asignada: 9 |
| Responsable (s): Vela David |
| <p>Descripción El frontend consume un endpoint para que el guardia pueda:</p> <ul style="list-style-type: none"> Visualizar el parqueadero de la ESFOT. <p>El frontend muestra una leyenda para que la persona con discapacidad pueda reservar la plaza de estacionamiento preferencial</p> |
| Observación: Para que el usuario pueda gestionar su perfil debe ingresar a con sus credenciales correctas. |

Product Backlog

A continuación, en la **Tabla 12** se presenta la prioridad de cada uno de los módulos que han sido implementados en el componente.

Tabla 12 Product Backlog.

| PRODUCT BACKLOG | | | | |
|-----------------|---|--------------------|------------|-----------|
| ID-HU | DESCRIPCIÓN DE LA HISTORIA | ITERACION ASIGNADA | ESTADO | PRIORIDAD |
| HU-003 | Gestionar plazas de estacionamiento (Administrador) | 3 | Finalizado | Alta |
| HU-004 | Gestionar guardias (Administrador) | 4 | Finalizado | Media |
| HU-005 | Gestionar usuarios (Administrador) | 5 | Finalizado | Media |
| HU-006 | Gestionar perfil (Guardia) | 6 | Finalizado | Media |
| HU-007 | Gestionar reserva de plaza de estacionamiento y del parqueadero (Guardia) | 7 | Finalizado | Alta |

| | | | | |
|---------------|---|----|------------|-------|
| HU-008 | Gestionar usuario (Guardia) | 8 | Finalizado | Media |
| HU-009 | Gestionar perfil (Usuario externo) | 9 | Finalizado | Media |
| HU-010 | Visualizar disponibilidad de plazas de estacionamiento (Usuario externo) | 10 | Finalizado | Alta |

Sprint Backlog

A continuación, en la **Tabla 13** se presenta la elaboración del Sprint Backlog para el componente frontend.

Tabla 13 Sprint Backlog.

| ELABORACION DEL SPRINT BACKLOG | | | | | | |
|--------------------------------|--|--------|-----|----------------------|--|-----------------|
| ID-SB | NOMBRE | MÓDULO | HU | HISTORIAS DE USUARIO | TAREAS | TIEMPO ESTIMADO |
| SB-000 | Configuración del ambiente de desarrollo | N/A | N/A | N/A | <ul style="list-style-type: none">• Recopilación y definición de requerimientos• Instalación de herramientas necesarias• Configuración del repositorio de código• Diseño y creación de los prototipos• Diseño de arquitectura para la aplicación web en base a los requerimientos. | 40H |

| | | | | | | |
|--------|--|--------------------------------------|--------|-------------------------------------|--|-----|
| SB-001 | Implementación de la página de entrada | Módulo de generación de landing page | HU-001 | Visualizar páginas de entrada | <ul style="list-style-type: none"> Diseñar el layout del menú Diseñar la landing page de inicio. | 10H |
| SB-003 | Consumo de endpoints para el usuario Administrador en el sistema web | Módulo de parqueaderos | HU-003 | Gestionar plazas de estacionamiento | <ul style="list-style-type: none"> Diseñar la interfaz para parqueaderos Implementación del CRUD de parqueaderos | 40H |
| | | Módulo de guardias | HU-004 | Gestionar guardias | <ul style="list-style-type: none"> Diseñar la interfaz para guardias Implementación del CRUD de guardias | |
| | | Módulo de usuarios | HU-005 | Gestionar Usuarios | <ul style="list-style-type: none"> Diseñar la interfaz para usuarios Implementación del CRD de usuarios | |
| | Consumo de endpoints para el usuario Guardia | Módulo de perfil | HU-006 | Gestionar perfil | <ul style="list-style-type: none"> Diseñar la interfaz para perfil Implementación del RU de guardia | |

| | | | | | | |
|--------|--|-------------------------------------|--------|---|--|-----|
| SB-004 | | Módulo de parqueaderos | HU-007 | Gestionar reserva de plaza de estacionamiento y del parqueadero | <ul style="list-style-type: none"> Diseñar la interfaz de parqueaderos Implementación del CRUD de parqueaderos | 20H |
| | | Modulo usuario externo | HU-008 | Gestionar usuario externo | <ul style="list-style-type: none"> Diseñar la interfaz de parqueaderos Implementación del CRU de parqueaderos. | |
| SB-005 | Consumo de endpoints para el usuario 'Usuario externo' | Módulo de perfil | HU-009 | Gestionar perfil | <ul style="list-style-type: none"> Diseñar la interfaz para perfil Implementación del CRUD de perfil | 10H |
| | | Módulo de plazas de estacionamiento | HU-010 | Visualizar disponibilidad de plazas de estacionamiento | <ul style="list-style-type: none"> Consumir endpoint para listar el parqueadero y plazas de estacionamiento. | 10H |

| | | | |
|---------------|------------|--|-------------|
| SB-006 | Pruebas | <ul style="list-style-type: none"> • Pruebas unitarias. • Pruebas de compatibilidad • Prueba de rendimiento. • Pruebas de aceptación | 20H |
| SB-007 | Despliegue | <ul style="list-style-type: none"> • Despliegue en Vercel. | 20H |
| Documentación | | <ul style="list-style-type: none"> • Trabajo de Integración Curricular • Anexos | 40H |
| TOTAL | | | 240H |

Diseño de interfaces

A continuación, se presentan los diseños restantes que van desde la **Figura 1** Interfaz de inicio de sesión hasta la **Figura 3** Interfaz de landing page.

Figura 1 muestra la interfaz de inicio de sesión de PoliParking. El diseño está dividido en dos secciones principales. A la izquierda, sobre un fondo azul claro, se encuentra el logo de PoliParking (un búho) y el título "Iniciar sesión". Debajo de esto, hay campos de entrada para "Email:" (con el ejemplo "usuario@correo.com") y "Contraseña:" (con caracteres ocultos por asteriscos). Un botón naranja "Ingresar" está ubicado al final de esta sección. A la derecha, hay una imagen de fondo que muestra una carretera curvada con luces de coches desenfocadas, creando un efecto de movimiento.

Figura 1 Interfaz de inicio de sesión.

Figura 2 muestra la interfaz del módulo "Usuarios externos" de PoliParking. La interfaz tiene una barra superior azul con el logo de PoliParking y un botón "Cerrar sesión". A la izquierda, hay un menú lateral con un icono de usuario y las opciones "User - Rol", "Usuarios", "Guardias" y "Plazas de estacionamiento". El contenido principal está dividido en tres secciones. La sección superior izquierda, titulada "Usuarios externos", contiene un subtítulo "Módulo para la gestión de usuarios externos" y un campo de "Busqueda usuarios". Debajo de esto, hay cuatro botones de filtro: "Todos", "administrativo", "estudiante" y "docente". La sección central es una tabla con las siguientes columnas: "Nombre", "Apellido", "Email", "Placa vehículo", "Teléfono", "Rol" y "Acción". La columna "Acción" contiene un botón rojo "Eliminar". La sección inferior izquierda tiene dos botones: "Anterior" y "Siguiente". La sección derecha, titulada "Nombre", contiene campos de entrada para "Apellido:", "Cédula:", "Telefono:", "Email:" y "Rol", seguidos de un botón naranja "Registrar".

Figura 2 Interfaz del módulo usuarios externos.

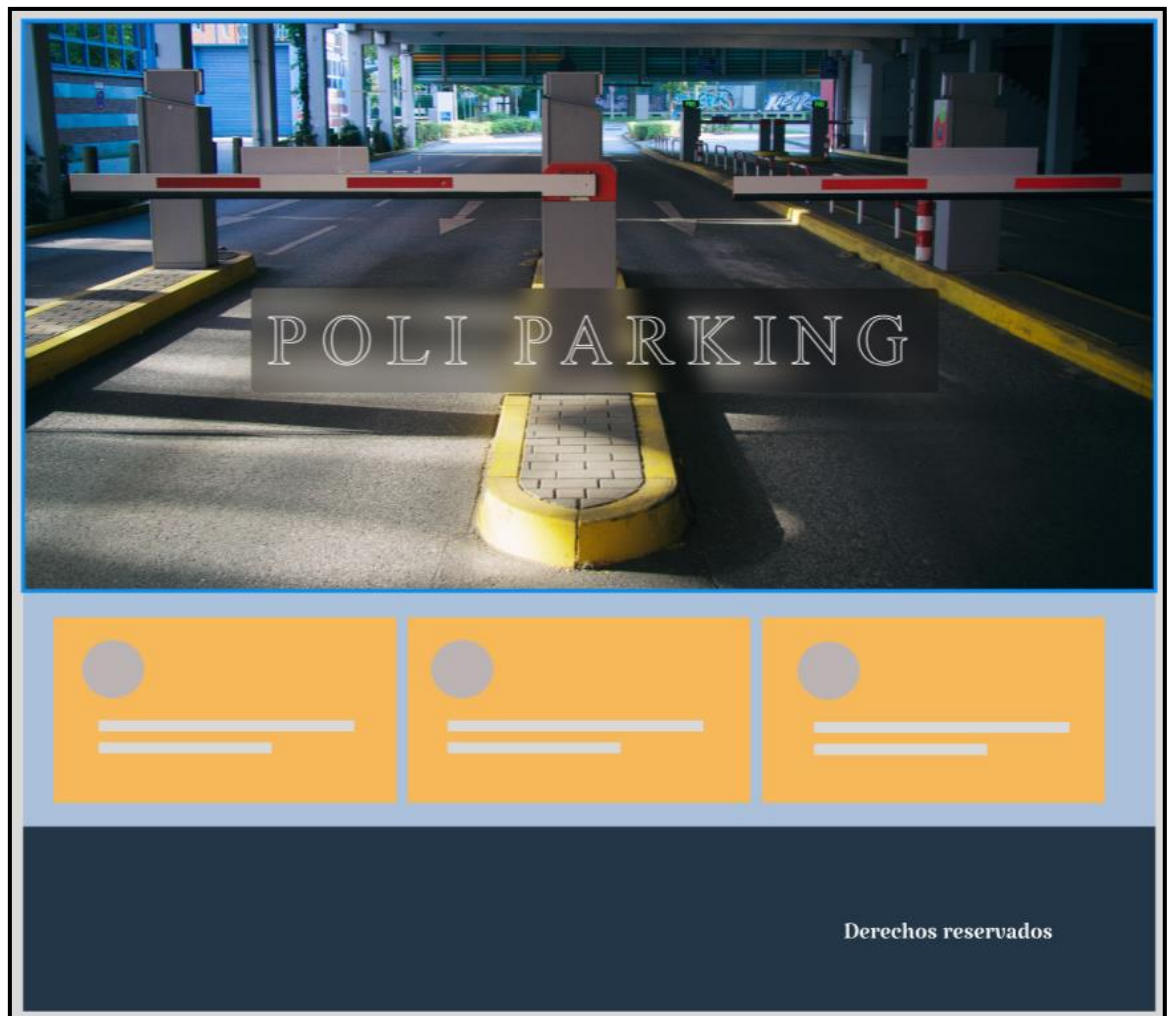


Figura 3 Interfaz de landing page.

Pruebas

Al concluir la etapa de desarrollo, se procede a realizar pruebas que verifican la calidad del frontend.

Pruebas unitarias

Las pruebas restantes se detallan desde la **Figura 4** hasta la **Figura 9**.

```
113 test("muestra mensaje cuando no hay parqueaderos", async () => {
114   parkingService.getParking.mockResolvedValueOnce([]);
115
116   const authContextValue = createMockAuthContext(mockUsers.admin);
117   const { getByText } = renderWithProviders(
118     <ParqueaderosAdminPage />,
119     authContextValue,
120   );
121
122   await act(async () => {
123     await parkingService.getParking();
124   });
125
126   expect(
127     getByText("No hay parqueaderos cargados en el sistema"),
128   ).toBeInTheDocument();
129 });
130
131 test("renderiza el formulario de actualización con todos sus campos", async () => {
132   const authContextValue = createMockAuthContext(mockUsers.admin);
133   const { getByPlaceholderText, getByText } = renderWithProviders(
134     <ParqueaderosAdminPage />,
135     authContextValue,
136   );
137
138   await act(async () => {
139     await parkingService.getParking();
140   });
141
142   expect(getByPlaceholderText("ESFOT")).toBeInTheDocument();
143   expect(getByPlaceholderText("Parqueadero de la ESFOT")).toBeInTheDocument();
144   expect(getByPlaceholderText("1")).toBeInTheDocument();
145   expect(getByPlaceholderText("A")).toBeInTheDocument();
146   expect(getByText("Actualizar Parqueadero")).toBeInTheDocument();
147 });
148
```

Figura 4 Fragmento de código de la prueba unitaria en el componente **ParqueaderosAdminPage**.

| | | |
|---|-------|---------|
| ✓ src/___tests___/pages/Administrador/ParqueaderosAdminPa... 2487ms | 62 | 0 |
| ✓ ParqueaderosAdminPage 2484ms | Pass | Fail |
| ✓ renderiza correctamente el título y la descripción 1145ms | 62 | Total |
| ✓ muestra mensaje de carga mientras obtiene los datos 304ms | | |
| ✓ muestra mensaje cuando no hay parqueaderos 34ms | | |
| ✓ renderiza el formulario de actualización con todos sus cam... 286ms | | |
| ✓ muestra mensaje de éxito después de actualizar 224ms | | |
| ✓ maneja errores durante la actualización 293ms | | |
| ✓ muestra el estado del parqueadero correctamente 163ms | | |
| ✓ muestra mensaje de parqueadero reservado cuando estado ... 30ms | | |
| | Files | 12 |
| | Pass | 12 |
| | Time | 278.58s |

Figura 5 Resultado de las pruebas unitarias del componente ParqueaderosAdminPage.

```
test("maneja errores al cambiar el estado del parqueadero", async () => {
  const authContextValue = createMockAuthContext(mockUsers.guardia);
  guardiaService.changeStatus.mockRejectedValueOnce(
    new Error("Error de red"),
  );

  const { getByText } = renderWithProviders(
    <ParqueaderosGuardiasPage />,
    authContextValue,
  );

  await act(async () => {
    await parkingService.getParking();
  });

  await act(async () => {
    fireEvent.click(getByText("Reservar parqueadero"));
  });

  expect(guardiaService.changeStatus).toHaveBeenCalled();
});
```

Figura 6 Fragmento de código de la prueba unitaria en el componente ParqueaderosGuardiasPage.

| | | | |
|--|--------|--------|---------|
| ✓ src/___tests___/pages/Guardias/ParqueaderosGuardiasPage.... | 2434ms | Total | |
| ✓ ParqueaderosGuardiasPage | 2431ms | | |
| ✓ renderiza correctamente el parqueadero y sus botones | 824ms | Files | 12 |
| ✓ puede cambiar el estado general del parqueadero | 588ms | ✓ Pass | 12 |
| ✓ puede cambiar el estado del espacio para discapacitados | 305ms | ⌚ Time | 278.58s |
| ✓ muestra mensaje de carga mientras obtiene los datos | 203ms | | |
| ✓ muestra mensaje cuando no hay parqueaderos | 14ms | | |
| ✓ maneja errores al cambiar el estado del parqueadero | 281ms | | |
| ✓ maneja errores al cambiar el estado del espacio para discap... | 211ms | | |

Figura 7 Pruebas unitarias en el componente ParqueaderosGuardiasPage

```
it('valida la longitud mínima de la contraseña', async () => {
  renderWithProviders(<ChangePassword />);

  const passwordInput = screen.getByPlaceholderText('Escribe tu nueva contraseña');
  await userEvent.type(passwordInput, '123');

  const submitButton = screen.getByText('Cambiar contraseña');
  fireEvent.click(submitButton);

  await waitFor(() => {
    expect(screen.getByText('La contraseña debe tener al menos 8 caracteres.')).toBeInTheDocument();
  });
});
```

Figura 8 Fragmento de código de la prueba unitaria en el componente ChangePassword.

| | | | |
|---|--------|----------|--------|
| ✓ src/___tests___/pages/ChangePassword.test.jsx | 4425ms | | |
| ✓ ChangePassword Component | 4422ms | | |
| ✓ renderiza correctamente el componente | 270ms | | |
| ✓ muestra errores de validación cuando se envía el formulari... | 147ms | | |
| ✓ valida que las contraseñas coincidan | 1088ms | | |
| ✓ valida la longitud mínima de la contraseña | 557ms | | |
| ✓ maneja correctamente el cambio de contraseña p... | 1355ms | | |
| ✓ maneja correctamente los errores del servicio | 1000ms | | |
| | | 62 Pass | 0 Fail |
| | | 62 Total | |

Figura 9 Pruebas unitarias para cambiar la contraseña en el componente ChangePassword.

Pruebas de rendimiento

Las pruebas restantes se muestran desde la **Figura 10** hasta la **Figura 15**.

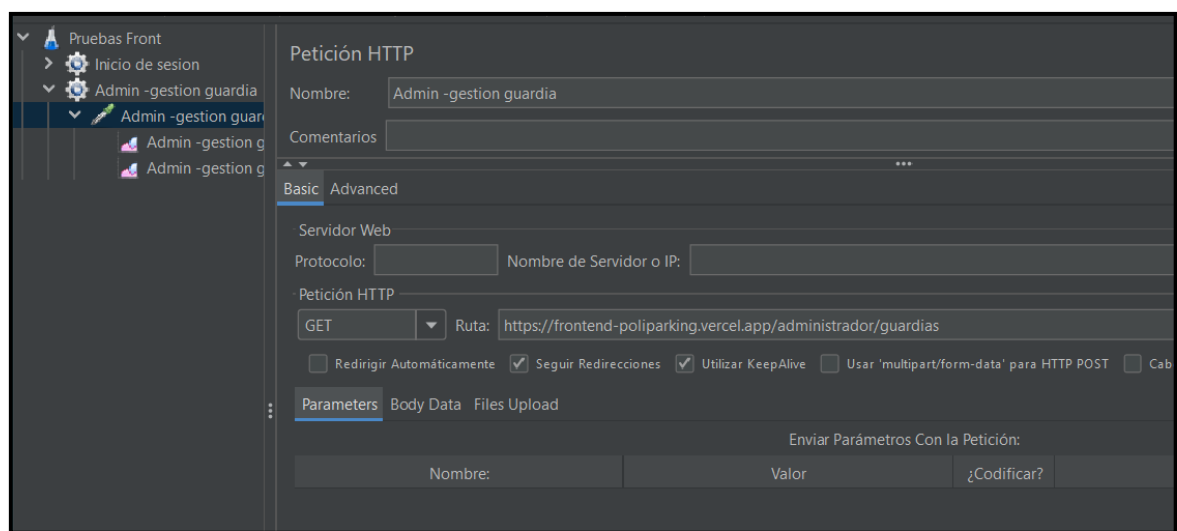


Figura 10 Prueba de rendimiento para visualizar el módulo guardia.

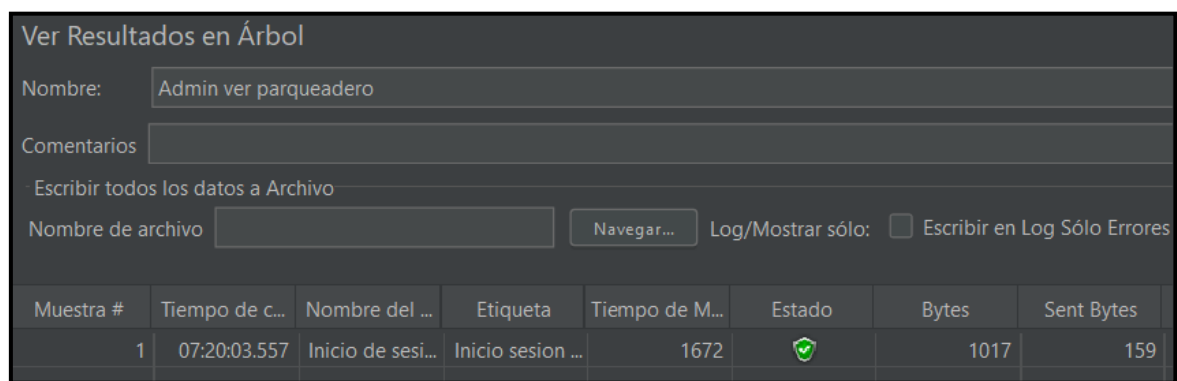


Figura 11 Resultado de prueba de rendimiento para visualizar el módulo de guardias.

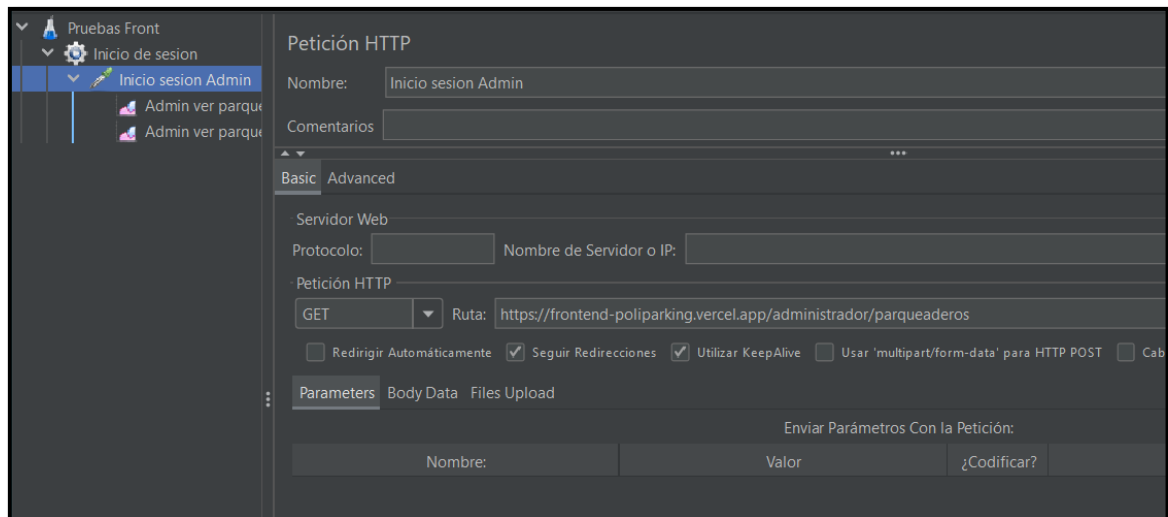


Figura 12 Prueba de rendimiento para visualizar el módulo plaza de estacionamiento.

Ver Resultados en Árbol

Nombre: guardia

Comentarios

Escribir todos los datos a Archivo

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

| Muestra # | Tiempo de com... | Nombre del hilo | Etiqueta | Tiempo de Mue... | Estado | Bytes | Sent Bytes | Latency | Connect Time(...) |
|-----------|------------------|-----------------|----------------|------------------|--------|-------|------------|---------|-------------------|
| 1 | 07:34:38.042 | guardia 3-1 | guardia - user | 560 | | 1017 | 441 | 556 | 436 |

Figura 13 Resultado de prueba de rendimiento para visualizar el módulo plaza de estacionamiento.



Figura 14 Prueba de rendimiento para visualizar el módulo usuarios externos.

Ver Resultados en Árbol

Nombre: Admin -gestion guardia

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores


| Muestra # | Tiempo de c... | Nombre del ... | Etiqueta | Tiempo de M... | Estado | Bytes | Sent Bytes |
|-----------|----------------|-----------------|----------------|----------------|---|-------|------------|
| 1 | 07:26:47.976 | Admin -gesti... | Admin -gest... | 524 |  | 1017 | 155 |
| | | | | | | | |

Figura 15 Resultado para la prueba de rendimiento para visualizar el módulo usuario externos.

Pruebas de compatibilidad

Las pruebas restantes van desde la **Figura 16** hasta la **Figura 18**.

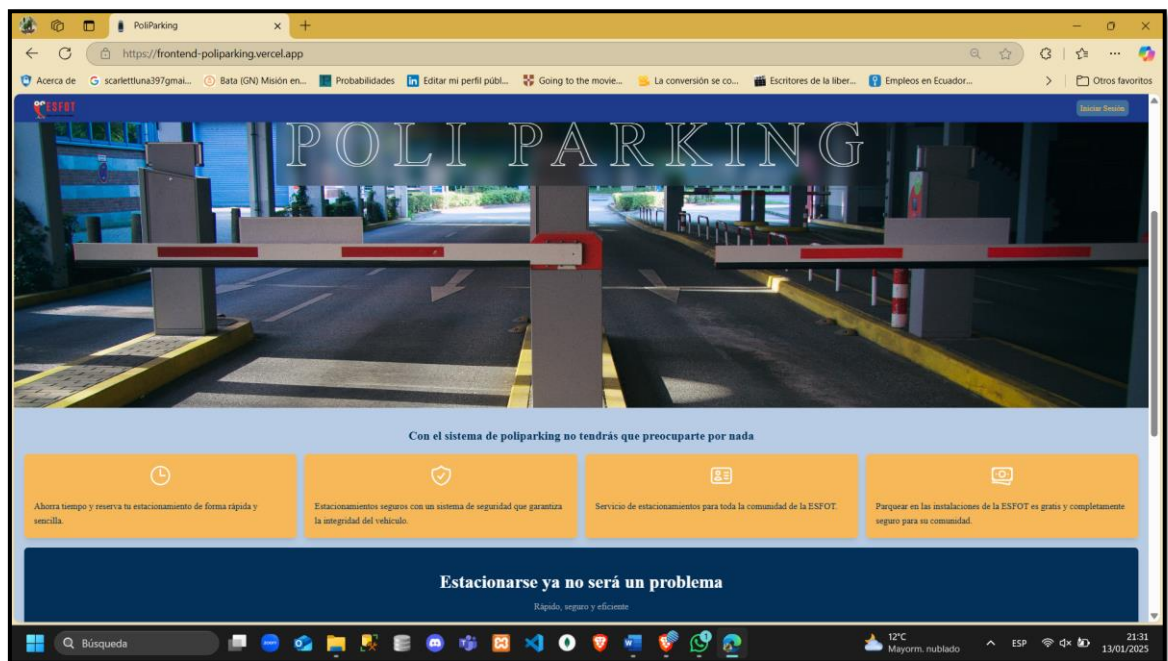


Figura 16 Página de landing page en el navegador Microsoft Edge.

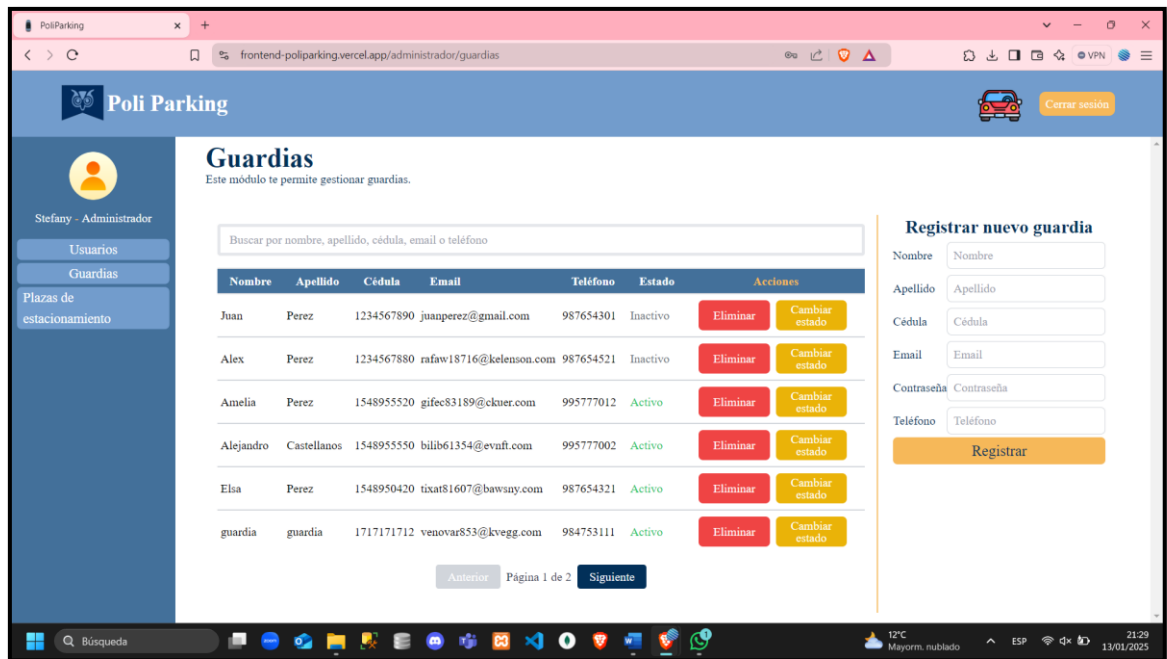


Figura 17 Página del módulo guardias en el navegador Brave.

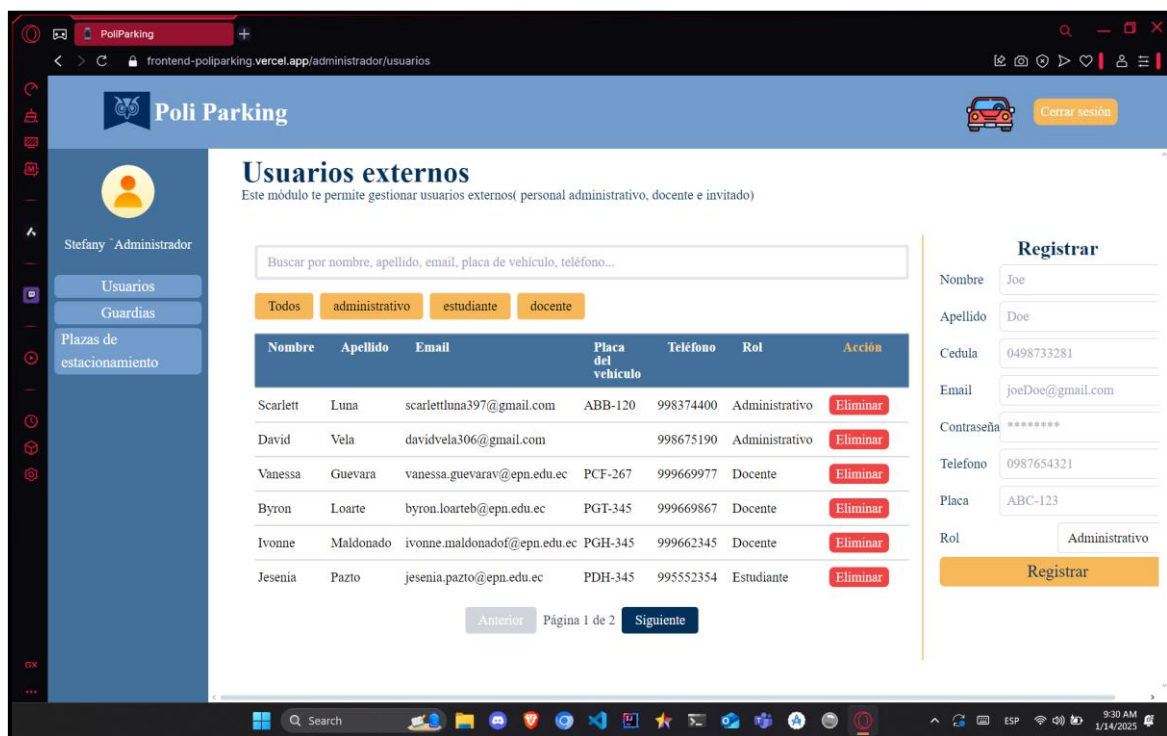


Figura 18 Página del módulo usuarios externos en el navegador Opera GX.

Pruebas de aceptación

A continuación, se presentan las pruebas de aceptación restantes que abarcan desde la **Tabla 14** hasta la **Tabla 23**

Tabla 14 Prueba de aceptación para visualizar página de entrada

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|--|------------------------------------|
| Identificador de la prueba: PA-001 | historia de Usuario: HU-001 |
| Nombre de la prueba: Visualizar página de entrada | |
| Descripción: El frontend crea una landing page para que los usuarios, guardias y usuarios puedan: Visualizar información sobre el negocio, qué solución tiene, qué servicios ofrece y cómo contactarlos. | |
| Fases de realización: Para acceder al frontend todos los usuarios deben: <ul style="list-style-type: none">• Ingresar a la ruta correspondiente.• Desplazarse por la página. | |
| Resultado deseado: El componente de frontend hace una presentación eficaz de la problemática resuelta y provee la propuesta de valor del sistema. | |
| Análisis de la prueba: El resultado cumple con las expectativas y el cliente expresa su plena satisfacción al 100%. | |

Tabla 15 Prueba de aceptación para Iniciar sesión.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|--|------------------------------------|
| Identificador de la prueba: PA-002 | historia de Usuario: HU-002 |
| Nombre de la prueba: Iniciar sesión | |

Descripción: El frontend crea un módulo funcional de login para:

- Ingresar al frontend según el rol del usuario.
- Persistir datos en el navegador
- Proteger rutas según el rol de usuario.
- Recuperar contraseña.

Fases de realización:

Para iniciar sesión todos los usuarios:

- Ingresar a la URL del frontend en un navegador.
- En lado derecho superior dar clic en Iniciar sesión.
- Llenar los campos del rol, correo electrónico y contraseña.
- Si alguno de estos campos falta o elige otro rol tendrá un mensaje de error
- Dar clic en “Iniciar Sesión”.
- Ingresar a su dashboard

Para salir de la sesión los usuarios:

- Haber iniciado sesión en el componente frontend.
- En el lado derecho superior dar clic en “Cerrar sesión”

Para recuperar contraseña:

- Ingresar a la URL del frontend en un navegador.
- En lado derecho superior dar clic en “Iniciar Sesión”.
- Dar clic en “¿Olvidaste tu contraseña?”.
- Llenar el formulario con tu correo electrónico.
- Confirmar el correo enviado a tu correo.
- Llenar el formulario respectivo para el cambio de contraseña.
- Dar clic en “Cambiar contraseña”

| |
|--|
| <p>Resultado deseado:</p> <p>El componente de frontend simplifica las acciones de inicio y cierre de sesión, además de permitir a los usuarios recuperar su contraseña.</p> |
| <p>Análisis de la prueba:</p> <p>El resultado cumple con las expectativas y el cliente expresa su plena satisfacción al 100%.</p> |

Tabla 16 Prueba de aceptación para gestionar guardias por parte del administrador.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|---|---|
| Designación (ID): PA-003 | Identificador de historia de Usuario: HU-003 |
| Nombre historia: Gestionar guardias | |
| <p>Descripción: El Frontend consume varios endpoints para que el administrador pueda:</p> <ul style="list-style-type: none"> • Registrar guardia • Listar guardias • Cambiar el estado para el guardia. • Eliminar guardias | |
| <p>Instrucciones de funcionamiento:</p> <p>Para listar los guardias el administrador debe:</p> <ul style="list-style-type: none"> • Dar clic en el menú lateral izquierdo en la opción Guardias • Visualizar la lista de guardias. <p>Para Registrar un guardia el administrador debe:</p> <ul style="list-style-type: none"> • Llenar el formulario de registro de guardia correctamente. • Dar clic en el botón Registrar. • Visualizar la alerta de registro. <p>Para cambiar el estado para el guardia el administrador debe:</p> <ul style="list-style-type: none"> • Dar clic en el botón Cambiar estado del respectivo guardia. | |

| |
|--|
| <p>Para eliminar el guardia el administrador debe:</p> <ul style="list-style-type: none"> • Dar clic en el botón Eliminar del respectivo guardia. |
| <p>Resultado deseado:</p> <p>El frontend permite gestionar el módulo Guardias.</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 de aceptación para gestionar usuarios externos por parte del administrador

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|---|---|
| Designación (ID): PA-004 | Identificador de historia de Usuario: HU-004 |
| Nombre historia: Gestionar Usuarios externos | |
| <p>Descripción: El frontend consume varios endpoints para que el administrador pueda:</p> <ul style="list-style-type: none"> • Listar usuarios externos. • Registrar usuarios externos (Administrativo y docente). • Eliminar usuarios externos. | |
| <p>Instrucciones de funcionamiento:</p> <p>Para listar los usuarios externos el administrador debe:</p> <ul style="list-style-type: none"> • Dar clic en el menú lateral izquierdo en la opción Usuarios • Visualizar la lista de usuarios externos. <p>Para Registrar un usuario externo (Administrativo o docente) el administrador debe:</p> <ul style="list-style-type: none"> • Llenar el formulario de registro de guardia correctamente. • Dar clic en el botón Registrar. • Visualizar la alerta de registro. | |

| |
|--|
| <p>Para eliminar el usuario externo el administrador debe:</p> <ul style="list-style-type: none"> • Dar clic en el botón Eliminar del respectivo usuario externo. |
| <p>Resultado deseado:</p> <p>El frontend permite gestionar el módulo Usuarios.</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 de aceptación para gestionar plazas de estacionamiento y parqueadero de la ESFOT por parte del administrador.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|---|---|
| Designación (ID): PA-005 | Identificador de historia de Usuario: HU-005 |
| Nombre historia: Gestionar plazas de estacionamiento y parqueadero de la ESFOT | |
| <p>Descripción: El frontend consume varios endpoints para que el administrador pueda:</p> <ul style="list-style-type: none"> • Listar el parqueadero con sus plazas de estacionamiento. • Actualizar la información del parqueadero. | |
| <p>Instrucciones de funcionamiento:</p> <p>Para listar el parqueadero con sus plazas de estacionamiento el administrador debe:</p> <ul style="list-style-type: none"> • Dar clic en el menú lateral izquierdo en la opción Plazas de estacionamiento. • Visualizar el parqueadero con la disponibilidad de las plazas de estacionamiento. <p>Para actualizar la información del parqueadero el administrador debe:</p> <ul style="list-style-type: none"> • Llenar el formulario de actualización de parqueadero correctamente. • Dar clic en el botón Actualizar parqueadero. • Visualizar la alerta de actualización. | |
| <p>Resultado deseado:</p> <p>El frontend permite gestionar el módulo del parqueadero.</p> | |

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 19 Prueba de aceptación para gestionar el perfil por parte del guardia

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|--|---|
| Designación (ID): PA-006 | Identificador de historia de Usuario: HU-006 |
| Nombre historia: Gestionar perfil | |
| Descripción: El frontend consume varios endpoints para que el guardia pueda: <ul style="list-style-type: none">• Detallar información de perfil.• Actualizar su dato de contactó (teléfono).• Actualizar contraseña. | |
| Instrucciones de funcionamiento: <p>Para listar la información de perfil el guardia debe:</p> <ul style="list-style-type: none">• Dar clic en el menú lateral izquierdo en la opción Perfil.• Visualizar su información personal. <p>Para actualizar su dato de contacto el guardia debe:</p> <ul style="list-style-type: none">• Llenar el formulario de actualización de teléfono correctamente.• Dar clic en el botón Actualizar teléfono.• Visualizar la alerta de actualización. <p>Para Actualizar contraseña el guardia debe:</p> <ul style="list-style-type: none">• Llenar el formulario de actualización de contraseña correctamente.• Dar clic en el botón Actualizar contraseña.• Visualizar la alerta de actualización. | |
| Resultado deseado: <p>El frontend permite gestionar el módulo Perfil.</p> | |

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 20 Gestionar usuarios externos por parte del guardia.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|---|---|
| Designación (ID): PA-007 | Identificador de historia de Usuario: HU-007 |
| Nombre historia: Gestionar usuarios externos (Estudiantes e invitados) | |
| Descripción: El frontend consume varios endpoints para que el guardia pueda: <ul style="list-style-type: none">• Listar usuarios externos.• Registrar usuarios externos (estudiantes e invitados).• Editar información de usuarios externos• Restringir o permitir acceso a usuarios externos. | |
| Instrucciones de funcionamiento: <p>Para listar los usuarios externos el guardia debe:</p> <ul style="list-style-type: none">• Dar clic en el menú lateral izquierdo en la opción Usuarios.• Visualizar todos los usuarios externos. <p>Para registrar un usuario externo el guardia debe:</p> <ul style="list-style-type: none">• Llenar el formulario de registro correctamente.• Dar clic en el botón Registrar.• Visualizar la alerta de registro. <p>Para Editar información de un usuario externo el guardia debe:</p> <ul style="list-style-type: none">• Dar clic en la acción Editar respectivo al usuario deseado• Llenar el formulario correctamente.• Dar clic en el botón Actualizar datos• Visualizar la alerta de actualización | |

| |
|--|
| <p>Para Restringir o permitir acceso a usuarios externos el guardia debe:</p> <ul style="list-style-type: none"> • Dar clic en el botón llamado Restringir acceso o Habilitar acceso dependiendo del caso. • Verificar el mensaje de alerta. |
| <p>Resultado deseado:</p> <p>El frontend permite gestionar el módulo Usuarios.</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 21 Prueba de aceptación para gestionar plazas de estacionamiento por parte del guardia.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|---|---|
| Designación (ID): PA-008 | Identificador de historia de Usuario: HU-008 |
| Nombre prueba: Gestionar plazas de estacionamiento | |
| <p>Descripción: El frontend consume un endpoint para que el guardia pueda:</p> <ul style="list-style-type: none"> • Visualizar el parqueadero de la ESFOT. • Reservar parqueadero completo. • Reservar plaza de estacionamiento para persona con discapacidad. | |
| <p>Instrucciones de funcionamiento:</p> <p>Para listar el parqueadero de la ESFOT el guardia debe:</p> <ul style="list-style-type: none"> • Dar clic en el menú lateral izquierdo en la opción Plazas de estacionamiento. • Visualizar el parqueadero con sus respectivas plazas de estacionamiento. <p>Para reservar el parqueadero por completo el guardia debe:</p> <ul style="list-style-type: none"> • Dar clic en el botón llamado Reservar parqueadero. • Visualizar el mensaje Parqueadero reservado | |

| |
|--|
| <p>Para reservar la plaza de estacionamiento para persona con discapacidad el guardia debe:</p> <ul style="list-style-type: none"> • Dar clic en el botón Discapacidad: Reservar plaza. • Verificar que la plaza se haya puesto de color rojo. |
| <p>Resultado deseado:</p> <p>El frontend permite gestionar el módulo del parqueadero junto con las acciones de reserva.</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 22 Prueba de aceptación para gestionar perfil por parte del usuario externo.

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|---|--|
| Designación (ID): PA-009 | Identificador de historia de Usuario: HU009 |
| Nombre historia: Gestionar perfil | |
| <p>Descripción: El frontend consumirá un endpoint para que el usuario pueda:</p> <ul style="list-style-type: none"> • Listar su información de perfil. • Actualizar su dato de contacto de su perfil (teléfono). • Actualizar contraseña. | |
| <p>Instrucciones de funcionamiento:</p> <p>Para listar la información de perfil el usuario externo debe:</p> <ul style="list-style-type: none"> • Dar clic en el menú lateral izquierdo en la opción Perfil. • Visualizar su información personal. <p>Para actualizar su dato de contacto el usuario externo debe:</p> <ul style="list-style-type: none"> • Llenar el formulario de actualización de teléfono correctamente. • Dar clic en el botón Actualizar teléfono. | |

| |
|---|
| <ul style="list-style-type: none"> • Visualizar la alerta de actualización. <p>Para Actualizar contraseña el usuario externo debe:</p> <ul style="list-style-type: none"> • Llenar el formulario de actualización de contraseña correctamente. • Dar clic en el botón Actualizar contraseña. • Visualizar la alerta de actualización. |
| <p>Resultado deseado:</p> <p>El frontend permite gestionar el módulo del parqueadero.</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 23 Prueba de aceptación para gestionar perfil por parte del usuario externo

| PRUEBA DE ACEPTACIÓN “POLI-PARKING” | |
|--|--|
| Designación (ID): PA-010 | Identificador de historia de Usuario: HU010 |
| Nombre historia: Visualizar disponibilidad de plazas de estacionamiento | |
| <p>Descripción:</p> <p>El frontend consume un endpoint para que el guardia pueda:</p> <ul style="list-style-type: none"> • Visualizar el parqueadero de la ESFOT. <p>El frontend muestra una leyenda para que la persona con discapacidad pueda reservar la plaza de estacionamiento preferencial</p> | |
| <p>Instrucciones de funcionamiento:</p> <p>Para listar el parqueadero de la ESFOT el usuario externo debe:</p> <ul style="list-style-type: none"> • Dar clic en el menú lateral izquierdo en la opción Plazas de estacionamiento. • Visualizar el parqueadero con sus respectivas plazas de estacionamiento. <p>Para reservar la plaza de estacionamiento preferencial el usuario externo debe:</p> | |

- Llamar al número telefónico ubicado el lado superior derecho del módulo Plazas de estacionamiento.

Resultado deseado:

El frontend permite gestionar el módulo Plazas de estacionamiento.

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.

ANEXO III

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

<https://www.youtube.com/watch?v=yyvhCNp7jI0>

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

ANEXO IV

A continuación, se encuentran las credenciales para acceder al frontend, así como un enlace al repositorio de GitHub con el código completo y más información en la sección README.

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

<https://frontend-poliparking.vercel.app/>

Credenciales para el perfil de administrador:

- Usuario del administrador: sluna6360@gmail.com
- Contraseña: admin.1234

Credenciales para el perfil guardia:

- Usuario del guardia: tixat81607@bawsny.com
- Contraseña: Guardia.456

Credenciales para el perfil usuario externo:

- Usuario del usuario externo: usuario@usuario.com
- Contraseña: usuario.123

Repositorio de GitHub

El proyecto se encuentra en el enlace siguiente:

<https://github.com/Davidvela-306/frontendPoliparking>