

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

SISTEMA DE GESTIÓN PARA REPARACIONES ELECTRÓNICAS PARA EL TALLER “ELECTRÓNICA ZURITA”

DESARROLLO DE UN FRONTEND

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

ALAN STEVEN PÉREZ ALMEIDA

DIRECTOR: FERNANDA MALDONADO SOLIZ

DMQ, agosto 2024

CERTIFICACIONES

Yo, Alan Steven Pérez Almeida 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.

ALAN STEVEN PÉREZ ALMEIDA

alan.perez@epn.edu.ec

alanstvn02@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado, bajo mi supervisión.

IVONNE FERNANDA MALDONADO SOLIZ

DIRECTOR

ivonne.maldonadof@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.

ALAN STEVEN PÉREZ ALMEIDA

AGRADECIMIENTO

Primero que nada, agradezco a mis padres y a mi hermano por estar siempre a mi lado con su constante apoyo. Sin su preocupación y sus mensajes de aliento, no estaría donde estoy ahora.

Otro especial agradecimiento a mi profesor Patricio Reinoso y a mi profesor Alex Manzano, quienes me han dejado grandes enseñanzas mientras estudiaba en el colegio, ayudándome a mejorar cada día.

Quiero expresar otro agradecimiento a mis amigos Paul Quishpe y Kevin Sandoval. Aunque no siempre logramos estar juntos físicamente, su apoyo y compañía a lo largo de los años han sido fundamentales para que cada uno de nosotros continúe en su carrera. Gracias por su amistad incondicional.

También quiero agradecer a la ingeniera Ivonne Maldonado, al ingeniero Byron Loarte y al ingeniero Juan Pablo Zaldumbide, quienes nos han llenado de conocimientos durante toda la carrera, brindándonos guía para ser mejores profesionales. Cada uno de sus consejos ha sido de gran ayuda para el desarrollo de este trabajo de titulación.

Finalmente agradezco a todas las autoridades y servidores de la Escuela Politécnica Nacional por darme la oportunidad de realizar mis estudios.

Muchas gracias.

Alan Steven Perez Almeida

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
AGRADECIMIENTO	III
ÍNDICE DE CONTENIDO	IV
RESUMEN.....	VI
ABSTRACT	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
2 METODOLOGÍA	6
2.1 Metodología de Desarrollo	6
Roles	6
Artefactos	7
2.2 Diseño de interfaces (mockups).....	9
Figma	9
2.3 Diseño de la arquitectura	10
Patrón arquitectónico.....	10
2.4 Herramientas de desarrollo	11
Librerías	12
3 RESULTADOS.....	14
3.1 Sprint 0. Configuración del ambiente de desarrollo	14
Requerimientos del componente frontend.....	14
Estructura del proyecto	15
Funcionalidades del rol Técnico.....	16
3.2 Sprint 1. Módulos técnico y cliente.....	17
Inicio y cierre de sesión	17
Recuperación de contraseña	18
Barra de navegación lateral	18
Gestión de perfil del técnico.....	19
Gestión de clientes	20
Visualización de clientes registrados	21
3.3 Sprint 2. Gestión de ordenes.....	22

Gestión de órdenes	22
Visualizar ordenes registradas.....	23
Cambio de estado de la orden.....	24
3.4 Sprint 3. Módulos proformas	25
Generación de proformas	25
Visualización de proformas.....	26
Gestión de técnicos	27
3.5 Sprint 4. Pruebas del componente frontend	28
Pruebas de compatibilidad.....	28
Pruebas de rendimiento.....	32
Prueba de responsividad	34
3.6 Sprint 5. Despliegue.....	39
4 CONCLUSIONES	42
5 RECOMENDACIONES	43
6 Referencias BIBLIOGRAFICAS	44
7 ANEXOS.....	49
ANEXO I.....	50
ANEXO II.....	51
ANEXO III.....	76
ANEXO IV.....	77

RESUMEN

Para un negocio que se centra en la reparación de equipos de cómputo, los cuales son esenciales en la actualidad, es necesario contar con una herramienta que permita gestionar de manera eficiente el trabajo por lo que este proyecto se centra en ayudar a la gestión de atención de un centro de reparación de equipos “Electrónica Zurita” a través del componente frontend realizado con React.

El frontend del sistema se estructura para cumplir con los requerimientos identificados tras un análisis de las necesidades del negocio. Contando con la capacidad de registrar técnicos que tienen acceso a funciones de inicio de sesión, gestión (registrar, eliminar y modificar) de clientes, además de poder generar proformas para las revisiones de los equipos de cómputo, ingresando los detalles del computador, los detalles de la reparación, fechas de ingreso y salida ligadas al cliente. Para el desarrollo se ha hecho uso de la metodología ágil Scrum cumpliendo con entregas continuas, además de que este proyecto se respalda por un backend alojado en Render.

El presente documento detalla el proceso que se ha llevado a cabo para el desarrollo del componente frontend del Sistema de Gestión para Reparaciones Electrónicas para el Taller “Electrónica Zurita, estructurado en cuatro secciones. En la primera sección se describe el planteamiento del problema, objetivos, alcance del proyecto y el marco teórico. La segunda sección aborda la metodología, herramientas y diseños implementados en el proyecto. En la tercera sección se detalla la ejecución de cada sprint del proyecto, presentando todos los resultados en cada etapa del desarrollo. Finalmente, en las últimas secciones 4 y 5 se presentan las conclusiones y recomendaciones que surgieron durante el desarrollo del presente proyecto.

Palabras clave: Gestión de equipos, React, Desarrollo de Software, Metodología Ágil

ABSTRACT

For a business focused on computer equipment repair, which is essential nowadays, it is necessary to have a tool that allows efficient work management. Therefore, this project focuses on helping the management of the service center "Electrónica Zurita" through the frontend component developed with React.

The system's frontend is structured to meet the requirements identified after an analysis of the business needs. It includes the capability to register technicians who have access to functions and login, manage (register, delete, and modify) clients, and generate quotes for equipment inspections by entering the computer details, repair details, and the check-in and check-out dates linked to the client. The development was carried out using the agile Scrum methodology, ensuring continuous deliveries, and the project is supported by a backend hosted on Render.

This document details the process carried out for the development of the frontend component of the Electronic Repair Management System for the "Electrónica Zurita" workshop, structured into four sections. The first section describes the problem statement, objectives, project scope, and theoretical framework. The second section addresses the methodology, tools, and designs implemented in the project. The third section details the execution of each project sprint, presenting all the results at each development stage. Finally, the last sections, 4 and 5, present the conclusions and recommendations that emerged during the project's development.

Keywords: Equipment Management, React, Software Development, Agile Methodology

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En la era actual, es fundamental contar con herramientas que simplifiquen las actividades laborales cotidianas. El crecimiento de las tecnologías y la innovación de negocios va aumentando a un ritmo considerable y, la necesidad de modernizar la interacción entre actividades y clientes se ha vuelto una necesidad obligatoria para entregar un servicio óptimo y transparente. Promover sistemas que permitan organizar y mejorar la calidad de trabajo para los técnicos electrónicos es crucial para el progreso del negocio. Los usuarios y los equipos para reparar son pieza clave para el avance constante de negocios dedicados a las reparaciones electrónicas. Unificar estos elementos mediante herramientas amigables y eficientes, elimina las brechas entre la interacción del trabajo, el técnico, el cliente y las finalidades de cada labor ejecutada; además de ofrecer un mecanismo de promoción para este tipo de negocios. Por esta razón, la creación de un componente web que facilite la ejecución de tareas antes realizadas en persona es esencial. En este contexto, el objetivo principal de este componente frontend es proporcionar una herramienta con una interfaz intuitiva que permita a los técnicos de reparación simplificar y optimizar las actividades de “Electrónica Zurita” [1].

La arquitectura del componente se basa en el consumo de una base de datos flexible y sólida que almacena información detallada sobre cada reparación [2]. Esta base de datos permite una búsqueda eficiente y personalizable de las reparaciones, teniendo en cuenta diferentes criterios como el estado del equipo, el tipo de reparación realizada o la fecha de ingreso del dispositivo.

Dada la diversidad de dispositivos informáticos disponibles en el mercado se vio necesario establecer un sistema de registro que permita ingresar datos específicos de cada equipo. Entre estos datos se incluyen el color, modelo, componentes y número de serie del dispositivo [3]. La implementación de esta lista busca mitigar posibles confusiones entre equipos de distintos clientes.

Este componente se esfuerza por satisfacer la demanda de acceso a información relevante sobre cada equipo en reparación. Busca proporcionar detalles útiles sobre los dispositivos y las reparaciones realizadas, permitiendo a los técnicos tener una comprensión completa de cada caso y generar una proforma que se envía al cliente con estos datos importantes.

1.1 Objetivo general

Desarrollar el componente frontend del sistema de gestión para reparaciones de equipos electrónicos para “Electrónica Zurita”.

1.2 Objetivos específicos

1. Determinar los requerimientos para la generación del sistema de gestión de reparaciones de equipos electrónicos.
2. Diseñar los prototipos en base a los requerimientos.
3. Codificar los prototipos en base a los requerimientos.
4. Consumir los diferentes endpoints del backend para las diferentes vistas del frontend.
5. Verificar el funcionamiento del frontend.
6. Desplegar el frontend.

1.3 Alcance

El alcance del componente frontend del sistema de gestión de reparaciones de equipos electrónicos para "Electrónica Zurita" incluye la interacción con APIs RESTful para mostrar, editar y guardar información sobre los equipos electrónicos gestionados [4]. Este frontend se conecta a un backend que cuenta con una base de datos para el correcto almacenamiento de los datos.

Para este desarrollo, se han planteado 7 módulos necesarios que abarcan todas las funcionalidades requeridas, y que se han desarrollado de la siguiente manera [5]:

1. **Registro e Inicio de sesión:** El componente frontend consume los endpoints para realizar un registro de los técnicos con sus datos básicos (nombre, apellido, teléfono, cédula, teléfono), incluyendo su número de RUC, su correo electrónico y su respectiva contraseña que les permita iniciar sesión o cerrar sesión.
2. **Registro de clientes:** El componente frontend consume los endpoints que permiten el registro de los clientes. Esto incluye datos básicos de la persona (nombre, cédula, teléfono), incluyendo dirección, correo electrónico y si el cliente es recurrente o no.
3. **Listado, actualización y eliminación de clientes:** El componente frontend consume los endpoints que permiten listar los clientes registrados, presentándolos en una tabla organizada por clientes recurrentes y no recurrentes. Los técnicos pueden actualizar o eliminar los datos de los clientes desde esta interfaz.
4. **Registro de órdenes:** El componente frontend consume los endpoints que permiten para registrar órdenes de trabajo, vinculadas a un cliente. El registro de la

orden cuenta con los datos como el nombre del equipo, número de serie, marca, motivo de ingreso, fecha de ingreso, tipo de servicio (mantenimiento, reparación, revisión) y el estado inicial ("Pendiente").

5. **Listado, actualización y finalización de órdenes:** El componente frontend consume los endpoints que permiten listar las órdenes registradas, presentadas en una tabla organizada por tipo de servicio (Mantenimiento, Reparación, Revisión) y el estado Finalizado. Los técnicos tienen la capacidad de actualizar el estado de las órdenes y generar proformas para las reparaciones.
6. **Generación de proformas:** El componente frontend consume los endpoints necesarios que permiten registrar una proforma ligada a los datos de la orden, en la proforma se agrega el costo estimado de la reparación calculado en base a las piezas y precios ingresados por el técnico.
7. **Perfil del cliente:** El componente frontend consume los endpoints que permiten mostrar y actualizar los datos del técnico, incluyendo la capacidad de actualizar su contraseña

El alcance del presente proyecto se limita al desarrollo del componente frontend del sistema de gestión para reparaciones electrónicas para “Electrónica Zurita con un único rol, este rol es el de “Técnico”. Este rol es el encargado de generar todos los datos, estos datos son un CRUD de clientes y equipos a reparar.

1.4 Marco Teórico

El desarrollo del componente frontend del sistema de gestión para reparaciones electrónicas para “Electrónica Zurita” utiliza tecnologías como React, HTML, JavaScript, Tailwind CSS, Axios y Vite, lo que requiere una comprensión profunda de los principios y prácticas fundamentales en el desarrollo web moderno [6]. Estos conceptos proporcionan la base necesaria para crear una interfaz de usuario intuitiva y eficiente que satisfaga las necesidades de los usuarios finales.

Desarrollo Web

El desarrollo web implica el uso de herramientas y tecnologías que permiten crear aplicaciones web rápidas, receptivas y fáciles de mantener. Esto incluye el uso de framework como React, que facilita la creación de interfaces de usuario dinámicas y escalables.

En la actualidad en donde la necesidad de contar con equipos de cómputo para realizar casi cualquier acción la cual antes era siempre en persona, ha conseguido que haya la necesidad de personas que reparen estos dispositivos, por lo que ha surgido la necesidad de que locales comerciales encuentren una manera de poder ampliar su negocio y así mismo utilizar este beneficio de la tecnología para facilitar su trabajo [7].

Por lo que una aplicación web es una respuesta a esta necesidad de tener una plataforma que permita a los técnicos poder realizar sus actividades de manera más directa y facilitando la comunicación con sus clientes.

Tecnologías Utilizadas:

React: Un framework de JavaScript ampliamente utilizado para construir interfaces de usuario interactivas y reutilizables. React utiliza un enfoque basado en componentes que permite dividir la interfaz de usuario en piezas independientes y reutilizables [8].

HTML: Lenguaje de marcado utilizado para estructurar el contenido de una página web, esta consiste en permitir generar un texto presentado de manera organizada que puede ser combinado con más interacciones relacionadas a elementos multimedia, hipervínculos que luego pueden ser editados con estilos gracias a otras tecnologías [9].

Tailwind CSS: Un marco de diseño de CSS que facilita la creación de interfaces de usuario personalizadas y estilizadas. Tailwind CSS utiliza clases predefinidas para aplicar estilos a los elementos HTML, lo que permite una mayor flexibilidad y control sobre el diseño, esto permitiendo generar diseños más atractivos de manera sencilla [10].

JavaScript: Lenguaje de programación utilizado para agregar interactividad y funcionalidad a las páginas web, este es utilizado para darle funcionalidad al maquetado de información, permitiendo que los diseños puedan contener más funcionalidades y que pueda ser interactuado por diferentes personas y no solo de un lado con el maquetado [11].

Axios: Una biblioteca de JavaScript utilizada para realizar solicitudes HTTP desde el navegador o desde Node.js. Axios simplifica la comunicación con servidores remotos y facilita el manejo de respuestas y errores [12].

Vite: Un entorno de desarrollo rápido y moderno para aplicaciones web basadas en JavaScript. Vite proporciona una experiencia de desarrollo en tiempo real y un tiempo de compilación rápido, lo que permite una iteración más rápida durante el desarrollo [13].

Google Maps API: Google Maps API es una API (interfaz de programación de aplicaciones) desarrollada por Google que permite integrar el servicio de mapas Google en

diferentes aplicaciones y plataformas. Esta API puede ser personalizada para darle diferentes funcionalidades y acciones dentro de la aplicación integrada [14].

Principios de Diseño de Interfaz de Usuario

El diseño de una interfaz de usuario efectiva implica la aplicación de principios de diseño centrados en el usuario. Esto incluye la creación de una experiencia de usuario intuitiva y fácil de usar, con una navegación clara y coherente [15].

Metodología Ágil de Desarrollo de Software

El desarrollo del componente frontend se lleva a cabo siguiendo los principios de la metodología ágil, que enfatiza la entrega incremental, la colaboración del equipo y la capacidad de respuesta a los cambios en los requisitos del proyecto. Esto permite una iteración rápida y una adaptación continua a medida que avanza el desarrollo [16].

Finalmente, cabe mencionar que el comprende todos estos conceptos y tecnologías detalladas anteriormente, permiten el desarrollo de un componente frontend efectivo que cumpla con los requisitos de gestión de atención de un centro de reparación de equipos informáticos como "Electrónica Zurita".

2 METODOLOGÍA

El presente proyecto, que desarrolla el frontend para el sistema del centro de reparaciones "Electrónica Zurita", se basa en un enfoque que permite comprender y describir las funcionalidades necesarias para que los técnicos puedan gestionar eficazmente las reparaciones de equipos informáticos. Este enfoque ayuda a identificar todas las funcionalidades esenciales para optimizar la productividad, facilitar el trabajo de los técnicos y mejorar la eficiencia general de su trabajo.

Para alcanzar este objetivo, se ha llevado a cabo una investigación de las necesidades mediante entrevistas y la observación de sus actividades diarias. Esta investigación ha permitido recopilar información sobre las funcionalidades que se consideran totalmente necesarias y comprender el cómo los técnicos esperan interactuar con el sistema. Gracias a los datos obtenidos, se ha generado el conocimiento necesario para traducir estas necesidades en la creación de un sistema que satisfaga las necesidades del negocio.

Para la ejecución de este proyecto se emplea como guía una metodología ágil, misma que permite gestionar proyectos de software de manera flexible y eficiente. Esta metodología permite estructurar ciclos de desarrollo conocidos como "sprints", para la entrega de los componentes funcionales. Además, se realizan reuniones periódicas con los miembros del equipo para identificar y resolver problemas en el desarrollo del proyecto [17].

2.1 Metodología de Desarrollo

Para el desarrollo del componente frontend, se ha utilizado la metodología ágil Scrum, reconocida por su capacidad de facilitar un desarrollo rápido y efectivo a través de iteraciones regulares que permiten una adaptación a cualquier cambio que se necesiten a lo largo del desarrollo. Scrum se caracteriza por su enfoque en la colaboración entre los miembros del equipo y la entrega progresiva de las funcionalidades del sistema, esto permitiendo que el proyecto se ajuste a las necesidades cambiantes de los usuarios y así alineándose con las expectativas de los mismos [18].

Roles

La metodología Scrum asigna roles específicos a cada miembro del equipo de desarrollo, garantizando que cada miembro está involucrado con funciones y responsabilidades hasta la conclusión del proyecto.

Product Owner

Es la persona representante del cliente o el usuario final, responsable de proporcionar los cambios o necesidades para garantizar que el producto cumpla con las expectativas [19].

Scrum Master

Es la persona líder responsable de guiar a los miembros del equipo para cumplir tanto los lineamientos de la metodología, así como de ayudar en la gestión de toda actividad pendiente del proyecto [20].

Development Team

Son las personas responsables de diseñar y desarrollar todo lo necesario para cumplir con las actividades planificadas en los sprints. Cada persona involucrada en este rol tiene la responsabilidad específica de asegurar que el proyecto a desarrollar sea un éxito [21].

A continuación, se presenta la **Tabla 2.1** con el equipo Scrum de este proyecto.

Tabla 2.1 Roles y persona responsable

Roles	Integrantes
Product Owner	Sr. Braulio Morales
Scrum Master	Ing. Ivonne Maldonado
Development Team	Sr. Alan Pérez

Artefactos

En la metodología Scrum, los artefactos son elementos que hacen referencia a toda la información clave necesaria para mejorar el desarrollo del proyecto. Esta información es fundamental para mejorar la colaboración dentro de todo el equipo, asegurando un mejor rendimiento en términos de tiempos y avances del proyecto, y garantizando el cumplimiento de los objetivos establecidos [22].

Recopilación de Requerimientos

La recopilación de los requerimientos es el proceso en el cual se delimita el desarrollo del proyecto de inicio a fin. En este proceso de recopilación se lleva a cabo en la primera etapa del proyecto, ya que esta permite comprender que necesidades se buscan solucionar [23]. El detalle de la recopilación de requerimientos puede observarse en el **Recopilación de Requerimientos**.

Historias de Usuario

Las historias de usuario son herramientas que ayudan a obtener una visión desde la perspectiva del usuario final. Estas historias de usuario suelen ser poco técnicas y proporcionan una descripción de lo que el usuario final desea [24]. A continuación, se presenta la **Tabla 2.2** que muestra la estructura de las historias de usuario utilizadas en este proyecto, el resto de historias de usuario se encuentran detalladas en el **Historias de Usuario**.

Tabla 2.2 Historia de usuario 003 – Consumir endpoints para la gestión de clientes.

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la gestión de clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir los <i>endpoints</i> para realizar la gestión de clientes, para lo cual necesita un formulario con las siguientes entradas: <ul style="list-style-type: none">• Nombre• Cédula• Número de teléfono o número de celular• Correo electrónico• Frecuencia del cliente• Dirección de domicilio	
Observación: El formulario del registro cuenta con verificaciones en cada campo, el nombre del usuario debe contar con máximo 30 caracteres y mínimo 3, el número de cedula y el número de teléfono tengan un número mínimo (5) y máximo de dígitos (13), el correo electrónico se valida que tenga una estructura de correo electrónico (ejemplo @gmail.com), la frecuencia del cliente debe tener una opción obligatoria (si es o no frecuente) y la dirección que se obtiene por medio de Google Maps debe ser seleccionada al dar clic.	

Product Backlog

El Product Backlog es una herramienta que permite planificar y gestionar todas las historias de usuario dentro del producto, proporcionando una visión panorámica de lo que se espera en el producto final. Aquí se incluye cualquier trabajo necesario para beneficiar al producto final. La gestión del Product Backlog es una de las responsabilidades del Product Owner, quien tiene la autoridad de priorizar y ejecutar los elementos en cada etapa del desarrollo [25]. Este artefacto puede observarse de manera detallada en el **Product Backlog**.

Sprint Backlog

El Sprint Backlog es un artefacto esencial en Scrum que ofrece al equipo de desarrollo una visión detallada de las tareas específicas que deben completarse durante cada iteración del sprint. Este backlog se crea al inicio de cada sprint y se actualiza constantemente a medida que el equipo avanza en sus actividades. Permite al equipo visualizar claramente las tareas asignadas, establecer objetivos claros para el sprint y gestionar de manera efectiva el trabajo diario [26]. Este artefacto puede observarse de manera detallada en el **Sprint Backlog**.

2.2 Diseño de interfaces (mockups)

En el desarrollo de productos, el proceso de creación de interfaces o mockups es un proceso clave, ya que estos son representaciones visuales que se utilizan como punto de partida para la creación de las interfaces que deben cumplir con los requerimientos establecidos por el Product Owner. Estas primeras versiones del diseño de interfaces permiten que el usuario interactúe con el producto, brindando una experiencia que es bastante cercana al producto final. Asimismo, permiten realizar pruebas de usabilidad tempranas, identificando posibles mejoras antes de avanzar en el desarrollo completo [27].

Figma

Figma es una herramienta muy útil para el diseño de prototipos en el desarrollo de aplicaciones frontend, ya que permite crear prototipos de cada interfaz de manera colaborativa y en tiempo real. En el presente proyecto, se ha utilizado esta herramienta para diseñar el prototipo de cada interfaz con el objetivo de facilitar la comprensión del producto final por parte del usuario. Esto no solo busca mejorar de la calidad en el producto final, si no también agilizar el proceso de desarrollo [28]. A continuación, se presenta la **Figura 2.1** que muestra un ejemplo del diseño realizado, el resto de interfaces se pueden observar en el **Diseño de Interfaces**.



Figura 2.1 Página principal

2.3 Diseño de la arquitectura

Los patrones arquitectónicos adoptados para el desarrollo de software facilitan la implementación de nuevas funcionalidades, asegurando que el sistema sea escalable a largo plazo y garantiza que va a ser compatible con nuevas herramientas y bibliotecas si el software lo requiere [29].

Patrón arquitectónico

Modelo-Vista-controlador (MVC) es un patrón arquitectónico que separa la lógica del sistema en tres capas. Este patrón es ampliamente utilizado porque esta división permite una mejor organización del desarrollo, facilitando el control más efectivo tanto antes como durante el desarrollo, logrando que el sistema sea escalable y mantenible, donde múltiples desarrolladores puedan trabajar en el mismo código con una comprensión clara de la estructura del proyecto [30].

- **Modelo:** gestionado por hooks de estado y las funciones que gestionan la lógica de negocios y el estado.
- **Vista:** Componentes de React que renderizan la interfaz de usuario (UI) y los componentes funcionales o de clase que utilizan el estado y las props para mostrar datos.
- **Controlador:** Funciones que realizan llamadas a los endpoints (fetch, axios) y la lógica que actualiza el estado basado en las respuestas de las API.

A continuación, se presenta la **Figura 2.2**, que muestra la implementación del patrón MVC en el desarrollo del componente frontend con el objetivo de facilitar la integración del sistema con más librerías y herramientas a futuro si se lo requiere.

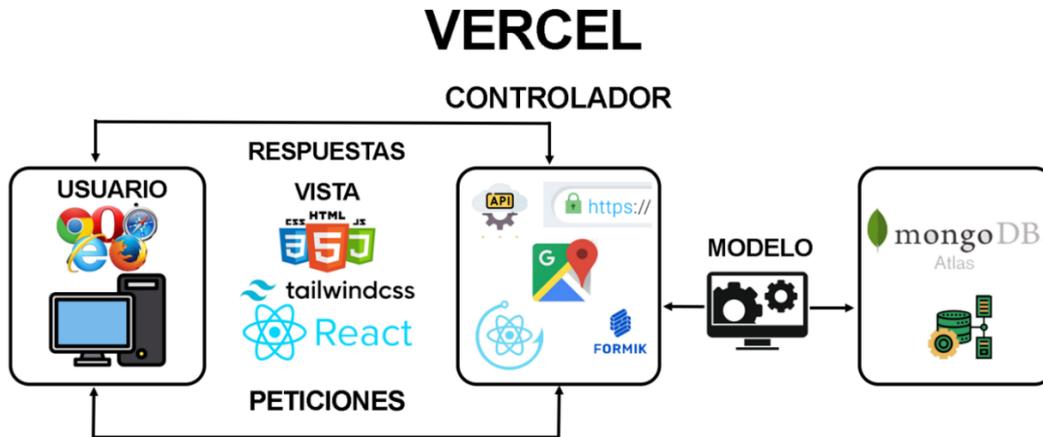


Figura 2.2 Patrón MVC para el componente frontend

2.4 Herramientas de desarrollo

Basándose en los requerimientos y la arquitectura MVC, la **Tabla 2.3** muestra las herramientas utilizadas en el desarrollo de este componente frontend para el centro de reparaciones "Electrónica Zurita".

Tabla 2.3 Herramientas para el componente frontend

Herramienta	Justificación
Visual Studio Code	Es un editor de código fuente altamente valorado por su capacidad para escribir, depurar y ejecutar el código, en este caso del frontend. Se ha elegido este IDE por su versatilidad, ofreciendo soporte para múltiples lenguajes y frameworks, así también por permitir instalar extensiones que amplían la funcionalidad y facilidad de desarrollo [31].
Axios	Es un cliente HTTP que se basa en promesas que ofrece compatibilidad con Node.js y con navegadores web. Esta versatilidad permite utilizar el mismo código en diferentes entornos, facilitando el uso y la integración en el sistema. Axios se utiliza para realizar solicitudes HTTP de manera eficaz a APIs, lo que facilita la gestión de datos y la interacción con servicios externos de manera efectiva [12].

React-DOM	Es una biblioteca de React que proporciona métodos específicos para interactuar con el DOM (Document Object Model). Estos métodos están diseñados para gestionar de manera eficiente los elementos del DOM de una página web. React-dom desempeña un papel importante al permitir la renderización de componentes de React en el sistema, permitiendo la creación y manipulación dinámica de la interfaz de usuario [32].
Tailwind CSS	Es una herramienta que permite aplicar estilos de manera más sencilla directamente en el código HTML del sistema. Este facilita la creación de estilos de forma eficiente y rápida, permitiendo personalizar rápidamente la apariencia visual de del sistema evitando la escritura manual de estilos CSS [33].
Edge	Es un navegador web ligero que viene instalado por defecto en los sistemas operativos Windows, este permite visualizar y revisar los cambios y resultados del componente frontend. Se elige por su rapidez, su versatilidad al venir por default en Windows, su compatibilidad por estar basada en Chromium y sus funcionalidades adicionales [34].
Google Maps API	La API de Google maps es una herramienta que permite integrar Google maps al sistema, generando nuevas experiencias dependiendo el uso que le dé el desarrollador, en este caso sirve para saber la dirección de domicilio del cliente [14].

Librerías

La **Tabla 2.4** presenta las principales bibliotecas o librerías utilizadas en el desarrollo del componente frontend.

Tabla 2.4 Librerías utilizadas en el desarrollo del componente frontend

Librería	Justificación
React Hook Form	Es una biblioteca diseñada para facilitar la validación y gestión del estado de formularios en el sistema. Esta herramienta cuenta con una API robusta, completa y

	fácil de usar, permitiendo a los desarrolladores crear y manejar formularios y otras acciones relacionadas de manera eficiente y sin complicaciones [35]
React-modal	Es una biblioteca que ofrece un componente accesible diseñado para mostrar diálogos modales. Estos diálogos modales son elementos que se superponen en el sistema y que temporalmente ocultan el contenido principal. Para regresar al contenido principal se debe cerrar el modal interactuando con él. Esta biblioteca ha sido utilizada en el sistema para mostrar detalles de proformas y actualizar ciertos datos [36].
React Icons	Es una biblioteca de iconos diseñada para React que proporciona una amplia variedad de iconos listos para usar. Estos iconos son implementados fácilmente mediante componentes de React, lo que facilita su integración y personalización en diferentes partes de la interfaz de usuario [37].
Formik	Es una biblioteca que simplifica la validación y creación de formularios interactivos en aplicaciones React. Esta herramienta se utilizó para validar los datos ingresados en los formularios y limitar la cantidad de caracteres en cada input, asegurando que solo los datos válidos sean enviados al backend [38].
Yup	Es una librería complementaria que se integra con Formik para facilitar la validación de datos en formularios. Cuando Formik detecta una validación incorrecta en el formulario, Yup puede generar mensajes específicos basados en las reglas de validación definidas previamente [39].

3 RESULTADOS

En la siguiente sección se presentan los resultados obtenidos durante la ejecución de cada Sprint, mostrando las interfaces y módulos desarrollados en todas las iteraciones que han sido planificadas con anterioridad.

3.1 Sprint 0. Configuración del ambiente de desarrollo

En el sprint 0 se han obtenido los siguientes resultados:

- Requerimientos del componente frontend
- Estructura del proyecto
- Funcionalidades del rol Técnico

Requerimientos del componente frontend

Consumo de endpoints para inicio de sesión y recuperación de contraseña

El único rol técnico tiene acceso a la interfaz del inicio de sesión. En esta se muestra un formulario que permite iniciar sesión y así mismo una opción para poder redirigirse a una página que permite recuperar la contraseña. Para ello el usuario ya debe estar registrado en el sistema para lo cual inicialmente se cuenta con un técnico inicial que puede registrar a otros técnicos.

Barra de navegación lateral

El usuario técnico que haya iniciado sesión correctamente, dispone de una barra de navegación lateral situada al lado izquierdo. Esta barra muestra todas las opciones necesarias para que el técnico pueda gestionar eficientemente las acciones que este requiera mediante botones y gracias a que tiene un estilo sencillo se puede agregar más funcionalidades a futuro si el técnico lo necesita. Además, la barra de navegación incluye un botón tipo hamburguesa que permite su ocultamiento, facilitando así la navegación entre las diferentes pantallas. Con el propósito de mejorar la accesibilidad del usuario, ésta barra de navegación es estática, de manera que el técnico no tenga que desplazarse verticalmente para acceder al resto de opciones disponibles.

Consumo de endpoints para la gestión de clientes

El usuario técnico consume los endpoints que le permiten gestionar los datos de los clientes, lo que le da acceso a registrar nuevos clientes, visualizar los clientes registrados

mediante una tabla que filtra en datos importantes para el técnico (si es recurrente o no), actualizar los datos de los clientes registrados y eliminar a los clientes.

Consumo de endpoints para la gestión de ordenes

El usuario técnico consume los endpoints que le permiten gestionar las órdenes del centro de reparaciones, teniendo la capacidad de registrar nuevas órdenes que deben estar ligadas a un cliente, también le permite visualizar las ordenes en una tabla que tiene un filtro en base a lo que requiera el técnico (si la orden tiene tipo de servicio mantenimiento, reparación, revisión y si la orden está finalizada) y actualizar la orden al estado finalizado.

Consumo de endpoints para la gestión de proformas

El usuario técnico consume los endpoints que le permiten gestionar las proformas, estas son generadas a partir de los datos de las ordenes que sean de tipo de servicio reparación, lo que le permite al técnico la capacidad de registrar la proforma, una por orden, visualizar la proforma generada para la orden y actualizar la proforma.

Consumo de endpoints para la gestión de técnicos

Inicialmente se cuenta con un usuario técnico inicial (quemado en el backend), éste es el único que puede consumir los endpoints para poder gestionar a otros técnicos, esto le permite registrar nuevos técnicos, listar los técnicos registrados y borrar a los técnicos registrados.

Consumo de endpoints para perfil del técnico

El usuario técnico consume los endpoints que le permiten visualizar sus datos, también le permite actualizar sus datos a excepción de su número de RUC y también le permite actualizar la contraseña de su cuenta de técnico. Esta gestión cuenta con una pantalla simple ya que no es un módulo que va a contar con mucha interacción.

Estructura del proyecto

Para lograr cumplir con la implementación de cada requerimiento, se ha utilizado el IDE Visual Studio Code y para la creación del componente se ha utilizado Create React App, estas herramientas han permitido crear del proyecto en React de manera más rápida, así mismo se debe instalar Tailwind CSS generando los archivos necesarios para que este pueda ser utilizado en todas interfaces de manera sencilla y el archivo de configuración para Vercel [40], que se ha utilizado para el despliegue del componente web.

A continuación, la **Figura 3.1** muestra la estructura del proyecto, donde se puede observar las carpetas utilizadas para la generación de todo el componente frontend.

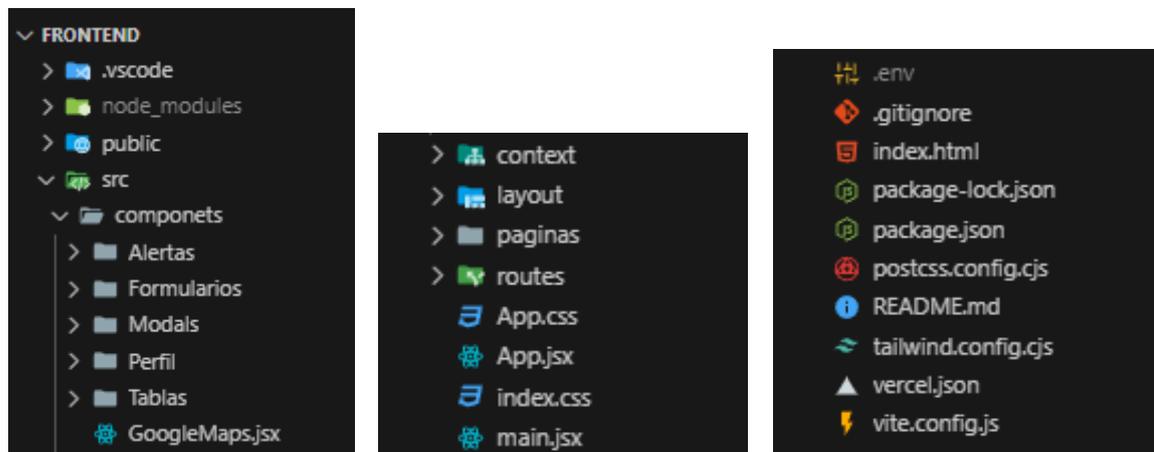


Figura 3.1 Estructura del proyecto

Funcionalidades del rol Técnico

Solo los usuarios técnicos que se encuentren registrados tienen acceso a las acciones de del componente, por lo que la **Figura 3.2** muestra los módulos que puede acceder el usuario técnico.

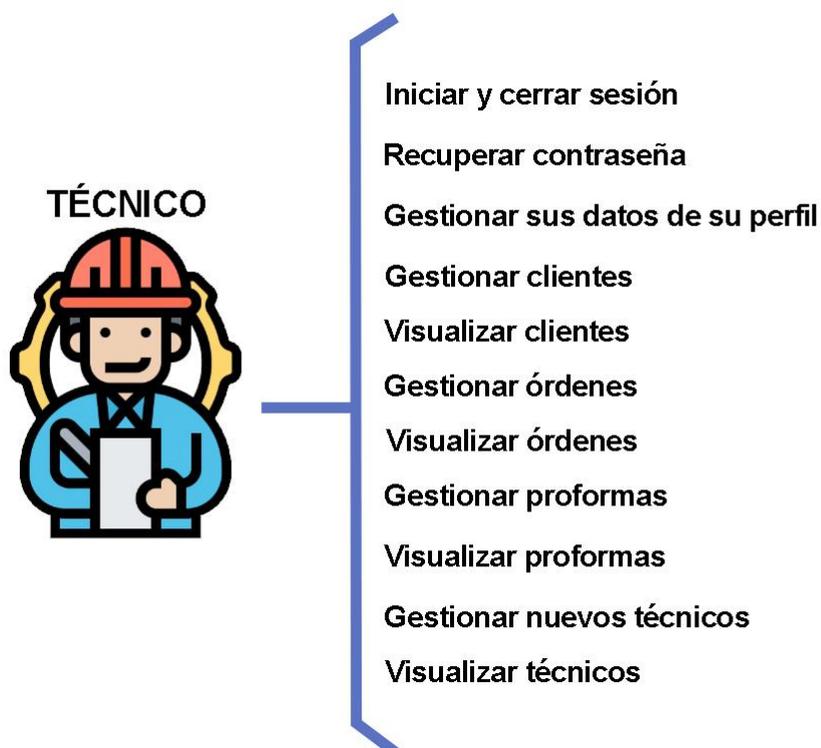


Figura 3.2 Funcionalidades del rol técnico

3.2 Sprint 1. Módulos técnico y cliente

En el sprint 1 se han obtenido los siguientes resultados:

- Inicio y cierre de sesión
- Recuperación de contraseña
- Barra de navegación lateral
- Gestión de perfil del técnico
- Gestión de clientes
- Visualización de clientes registrados

Inicio y cierre de sesión

Se ha implementado un formulario simple que le permite al usuario realizar la acción del inicio de sesión, este formulario le permite ingresar las credenciales de correo electrónico y contraseña, estos datos son enviados y verificados gracias al consumo del endpoint del backend, así también cuenta con un tiempo de espera de 10 minutos que empiezan a ser contado cuando el usuario no utilice el componente, si este no la usa, automáticamente se cierra sesión. Una vez el usuario haya ingresado correctamente, tiene la opción de cerrar sesión. En la **Figura 3.3** se muestra el resultado de la pantalla de inicio de sesión.



Figura 3.3 Inicio de sesión

Recuperación de contraseña

Esta funcionalidad permite restablecer la contraseña, para ello se ha implementado un formulario simple en el cual es necesario ingresar el correo electrónico del usuario, este dato es enviado al endpoint del backend para hacer su validación y si este es correctamente validado, se redirecciona a una página que le permita ingresar una nueva contraseña. La **Figura 3.4** presenta la pantalla generada para la recuperación de la contraseña.



Figura 3.4 Recuperación de contraseña

Barra de navegación lateral

El componente frontend ha logrado implementar con éxito una barra de navegación sencilla que muestra los botones de acción necesarios para el usuario técnico. Esta barra de navegación mejora la accesibilidad y facilita el uso sin necesidad de desplazarse verticalmente al navegar entre las diferentes pantallas, contando con un botón tipo hamburguesa que al ser presionado activa o desactiva la barra, lo cual también mostrará los iconos de cada una de las acciones que necesita el usuario técnico como se muestra en la **Figura 3.5**.

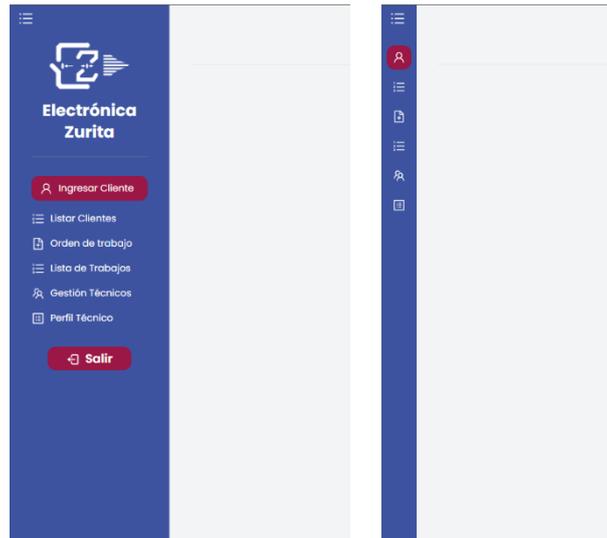


Figura 3.5 Barra de navegacion

Gestión de perfil del técnico

El usuario técnico puede gestionar su perfil, lo que le permite visualizar los datos actuales de su cuenta, así mismo le permite actualizar los datos de la cuenta y su contraseña para inicio de sesión, todo esto consumiendo correctamente los endpoints del backend. A continuación, se muestra el resultado de la implementación de esta funcionalidad en la Figura 3.6.

Datos del Técnico

Nombre técnico: Alan Brito
Correo electrónico: alan.stvn.alm@gmail.com
RUC: 123456745678 Teléfono: 774562567

Actualizar Datos Técnico

Nombre: Apellido:
Teléfono:
Correo electrónico:

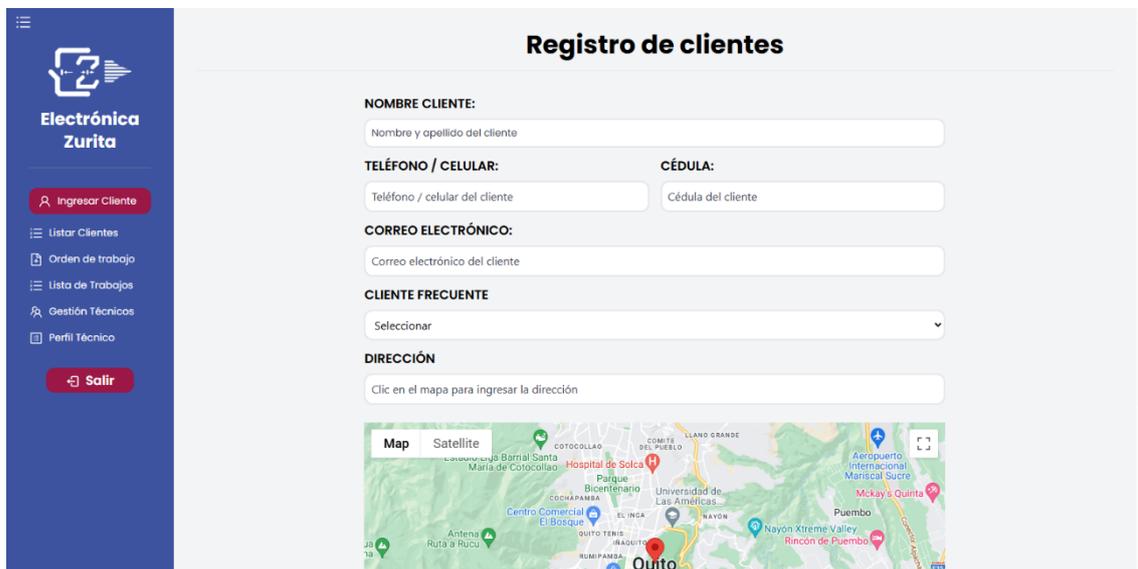
Actualizar Contraseña

Contraseña actual:
Nueva contraseña:

Figura 3.6 Perfil del técnico

Gestión de clientes

A continuación, se presenta las pantallas generadas para la gestión de clientes, estas consumen correctamente los endpoints del backend para poder hacer un registro de los datos de los clientes, como se muestra en la **Figura 3.7**, en esta se muestra un formulario con los datos más importantes que necesita el técnico, como los datos personales, la frecuencia del cliente a las instalaciones y su dirección de domicilio. El registro y la actualización de datos del cliente cuentan con la implementación de Google Maps, permitiendo que el usuario seleccione una ubicación al dar clic en el mapa como se indica en la **Figura 3.8**, siendo esta la dirección que se envíe en el formulario.



Registro de clientes

NOMBRE CLIENTE:
Nombre y apellido del cliente

TELÉFONO / CELULAR: **CÉDULA:**
Teléfono / celular del cliente Cédula del cliente

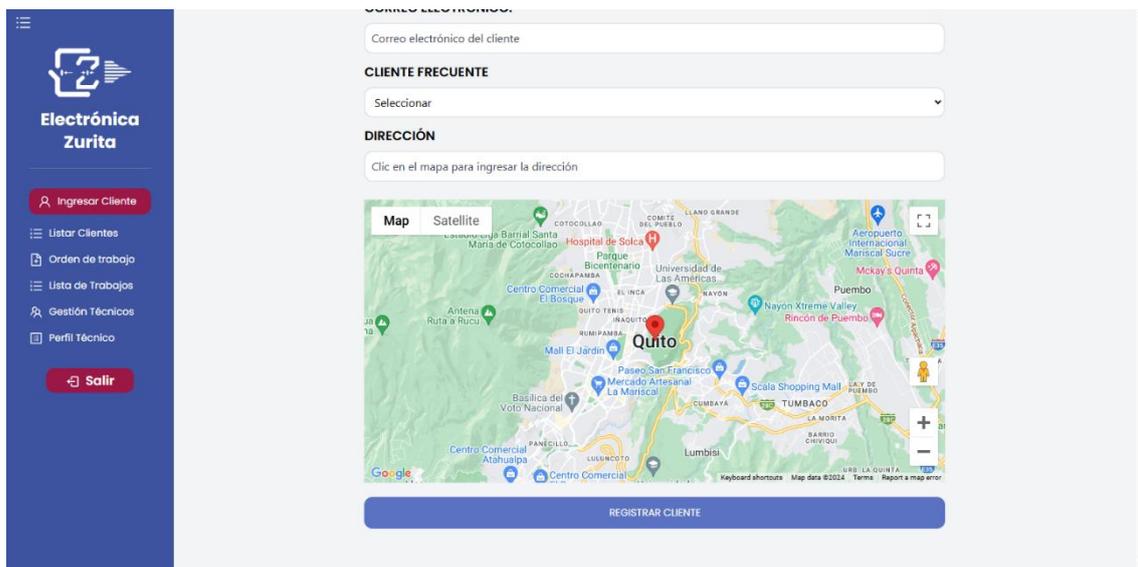
CORREO ELECTRÓNICO:
Correo electrónico del cliente

CLIENTE FRECUENTE
Seleccionar

DIRECCIÓN
Clic en el mapa para ingresar la dirección

Map Satellite

Figura 3.7 Pantalla registro de cliente



CORREO ELECTRÓNICO:
Correo electrónico del cliente

CLIENTE FRECUENTE
Seleccionar

DIRECCIÓN
Clic en el mapa para ingresar la dirección

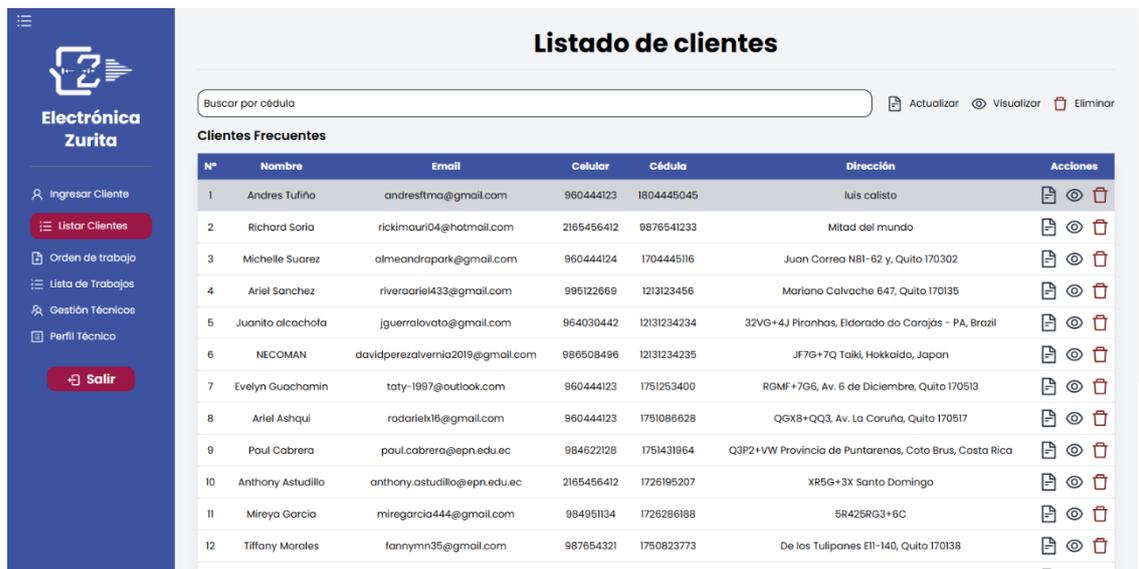
Map Satellite

REGISTRAR CLIENTE

Figura 3.8 Implementación Google Maps

Visualización de clientes registrados

Los clientes registrados se muestran en una pantalla que cuenta con dos tablas (clientes en recurrentes y clientes no recurrentes), además esta pantalla cuenta con una barra que sirve de filtro permitiendo buscar a un cliente por su número de cédula. Para cada cliente se tienen íconos de acciones (actualizar, visualizar o eliminar) que ayudan al técnico a entender cada funcionalidad, la **Figura 3.9** y la **Figura 3.10** muestran con éxito la visualización de las tablas y sus respectivas acciones, esto debido al consumo adecuado de los endpoints del backend.



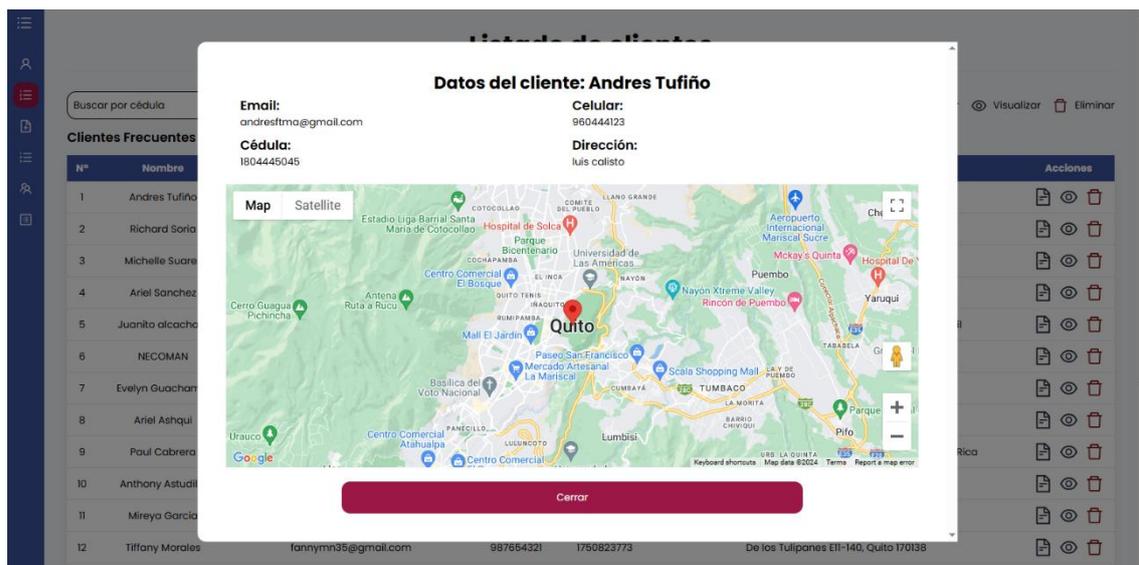
Listado de clientes

Buscar por cédula Actualizar Visualizar Eliminar

Cientes Frecuentes

N°	Nombre	Email	Celular	Cédula	Dirección	Acciones		
1	Andres Tufiño	andresfma@gmail.com	960444123	1804445045	luis calisto			
2	Richard Soria	rickimauri04@hotmail.com	2165456412	9876541233	Mitad del mundo			
3	Michelle Suarez	almeandrapark@gmail.com	960444124	1704445116	Juan Correa N81-62 y, Quito 170302			
4	Ariel Sanchez	riveraarie433@gmail.com	995122669	1213123456	Mariano Calvache 647, Quito 170135			
5	Juanito alcachofa	jguerralovato@gmail.com	964030442	12131234234	32VG+4J Piranhas, Eldorado do Carajás - PA, Brazil			
6	NECOMAN	davidperezalvernia2019@gmail.com	986508496	12131234235	JF7G+7Q Talki, Hokkaido, Japan			
7	Evelyn Guachamin	taty-1997@outlook.com	960444123	1751253400	RGMF+7G6, Av. 6 de Diciembre, Quito 170513			
8	Ariel Ashqui	rodarielx16@gmail.com	960444123	1751086628	QGX8+QQ3, Av. La Coruña, Quito 170517			
9	Paul Cabrera	paul.cabrera@epn.edu.ec	984622128	1751431964	Q3P2+VW Provincia de Puntarenas, Coto Brus, Costa Rica			
10	Anthony Astudillo	anthony.astudillo@epn.edu.ec	2165456412	1726195207	XR5G+3X Santo Domingo			
11	Mireya Garcia	miregarcia444@gmail.com	984951134	1726286188	5R4Z5RQ3+6C			
12	Tiffany Morales	fannymn35@gmail.com	987654321	1750823773	De los Tulipanes E11-140, Quito 170138			

Figura 3.9 Visualización de clientes registrados



Datos del cliente: Andres Tufiño

Email: andresfma@gmail.com **Celular:** 960444123

Cédula: 1804445045 **Dirección:** luis calisto

Map **Satellite**

Cerrar

Figura 3.10 Visualización de datos del cliente

3.3 Sprint 2. Gestión de ordenes

En el sprint 2 se han obtenido los siguientes resultados:

- Gestión de ordenes
- Visualizar ordenes registradas
- Cambio de estado de la orden

Gestión de órdenes

El usuario técnico tiene acceso a la gestión de las órdenes, para ello en esta pantalla se ha incluido 2 formularios, el primero consume un endpoint del backend que permite hacer una búsqueda de un cliente para comprobar que este existe, al finalizar correctamente esa validación se muestra los datos del cliente al que se le va a generar una orden, para ello se necesita llenar el segundo formulario el cual cuenta con un apartado importante que es el tipo de servicio (mantenimiento, revisión o reparación) de la orden, el resultado de esta pantalla se muestra en la **Figura 3.11** y la **Figura 3.12**.

The screenshot shows a web application interface for 'Electrónica Zurita'. The main content area is titled 'Orden de trabajo' and contains a form with the following fields:

- NÚMERO DE CÉDULA CLIENTE:** Input field for 'Número de cédula del cliente' with a 'BUSCAR CLIENTE' button below it.
- Cliente:** Input field for 'Nombre del cliente'.
- Télefono:** Input field for 'Teléfono'.
- Correo electrónico:** Input field for 'Correo electrónico'.
- Equipo:** Input field for 'Tipo de equipo'.
- Modelo:** Input field for 'Modelo del equipo'.
- MARCA:** Input field for 'Marca del equipo'.
- NÚMERO DE SERIE:** Input field for 'Número de serie del equipo'.
- COLOR DEL EQUIPO:** Input field for 'Color del equipo'.
- FECHA DE INGRESO DEL EQUIPO:** Input field for 'Fecha de ingreso del equipo' with a date picker icon.

The left sidebar contains the following navigation items:

- Ingresar Cliente
- Listar Clientes
- Orden de trabajo (highlighted in red)
- Lista de Trabajos
- Gestión Técnicos
- Perfil Técnico
- Salir

Figura 3.11 Registro de orden parte superior

Modelo:
Modelo del equipo

MARCA:
Marca del equipo

NÚMERO DE SERIE:
Número de serie del equipo

COLOR DEL EQUIPO:
Color del equipo

FECHA DE INGRESO DEL EQUIPO:
mm/dd/yyyy

RAZÓN DE INGRESO:
Razones de ingreso del equipo

SERVICIO:
Mantenimiento

REGISTRAR ORDEN

Figura 3.12 Registro de orden parte inferior

Visualizar ordenes registradas

En la **Figura 3.13** y **Figura 3.14** se muestran los resultados del consumo de los endpoints del backend para poder visualizar las ordenes registradas, estas son mostradas en diferentes tablas que tienen por filtros el tipo de servicio (mantenimiento, revisión o reparación) y una última tabla que muestra aquellos servicios que se encuentran en estado finalizado, además de incluir una columna de “fecha de salida” de la orden que se genera gracias a los botones de acción, permitiendo actualizar los estados de las ordenes, visualizar la orden y finalizar la orden.

Listado de Trabajos

Buscar por cédula

Actualizar Estado | Crear Proforma | Visualizar | Finalizar

Mantenimiento

Nº Orden	Cliente	Cédula	Equipo	Fecha Ingreso	Estado	Acciones
0001	Tiffany Morales	1750823773	Consola	7/3/2024	Pendiente	[Documento] [Ojo] [Check]

Reparación

Nº Orden	Cliente	Cédula	Equipo	Fecha Ingreso	Estado	Acciones
0002	Tiffany Morales	1750823773	Consola	7/3/2024	En proceso	[Documento] [Ojo] [Check]

Figura 3.13 Visualización tablas ordenes



Figura 3.14 Visualización datos de orden

Cambio de estado de la orden

El usuario técnico puede hacer una actualización del estado de las órdenes generadas, para ello se cuenta con un botón en la columna de acciones, que por medio del consumo exitoso los endpoints del backend, el técnico puede hacer una actualización de estado de las órdenes, pasando de un estado “Pendiente” a un estado “En proceso” y así mismo pasar a un estado “Finalizado”. El resultado de la pantalla para el cambio de estado se muestra en la **Figura 3.15**.



Figura 3.15 Cambio de estado de orden

3.4 Sprint 3. Módulos proformas

En el sprint 3 se han obtenido los siguientes resultados:

- Generación de proformas
- Visualización de proformas
- Gestión de técnicos

Generación de proformas

El usuario técnico tiene acceso a la generación de proformas, siendo inicialmente redirigido desde el botón de actualizar únicamente de las ordenes con servicio “Reparación”, pudiendo revisar los datos de la orden a la que se liga y contando con un formulario en el cual se puede agregar piezas que tienen su respectivo precio, estas no son limitadas y cada una cuenta con un botón para poder eliminar por si el técnico lo requiere, la **Figura 3.16** muestra el resultado de esta funcionalidad. Además, para confirmar que los datos son correctos se muestra una pantalla con toda la información, tanto de la orden como de las piezas agregadas por el técnico (**Figura 3.17**). Todo esto se logra gracias al consumo exitoso de los endpoints del backend.

Registro de proforma

Datos de la orden 0002

Cliente: Tiffany Morales **Cédula:** 1750823773

Equipo: Consola **Marca:** Xbox

Modelo: One X **Serie:** XBN008975

Fecha de Ingreso: 7/3/2024

Razón: Limpieza completa preventiva y correctiva

Piezas para la reparación

Pieza N°1 **Precio**

Nombre de la pieza \$

Agregar Pieza **Eliminar todas las piezas**

Total: \$0.00

Registrar Proforma

Figura 3.16 Generación de proforma

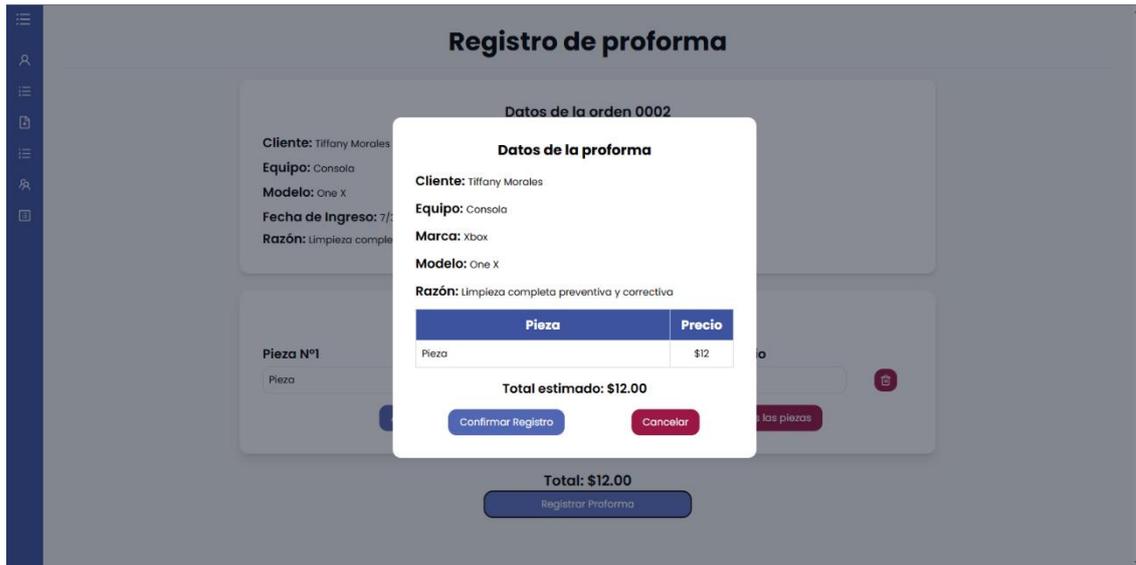


Figura 3.17 Proforma generada

Visualización de proformas

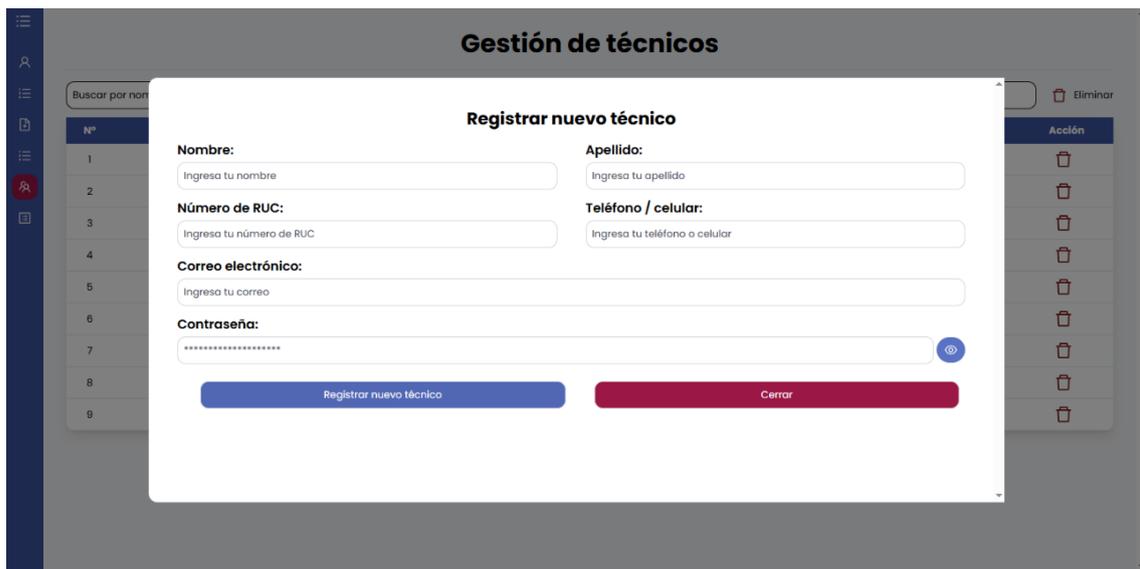
El técnico puede visualizar la proforma consumiendo el endpoint respectivo del backend, para ello en la tabla de ordenes se ha cambiado la actualización de estado de orden a registrar proforma, mientras que visualizar orden pasa a ser visualizar la proforma generada, la cual muestra un dato importante como es el estado. El resultado de esta funcionalidad se muestra en la **Figura 3.18**.



Figura 3.18 Visualización de proforma

Gestión de técnicos

El primer técnico que se encuentra definido inicialmente en la base de datos del backend cuenta con la acción de registro de un nuevo técnico (**Figura 3.19**), esto por medio de un botón ubicado en la parte inferior de la pantalla de la tabla de los clientes registrados, una vez que se ha registrado un nuevo técnico este tiene que hacer una confirmación de su cuenta, la cual se realiza al ingresar al enlace proporcionado en un correo generado por el componente backend, logrando así que el registro sea exitoso (**Figura 3.20**).



The image shows a web application interface for 'Gestión de técnicos'. A modal window titled 'Registrar nuevo técnico' is open over a table of technicians. The modal contains the following fields and buttons:

- Nombre:** Ingesta tu nombre
- Apellido:** Ingesta tu apellido
- Número de RUC:** Ingesta tu número de RUC
- Teléfono / celular:** Ingesta tu teléfono o celular
- Correo electrónico:** Ingesta tu correo
- Contraseña:** A password field with a strength indicator icon.
- Buttons:** 'Registrar nuevo técnico' (blue) and 'Cerrar' (red).

The background table has columns for 'N°', 'Nombre', 'Apellido', 'Teléfono / celular', and 'Acción'. The 'Acción' column contains an 'Eliminar' button.

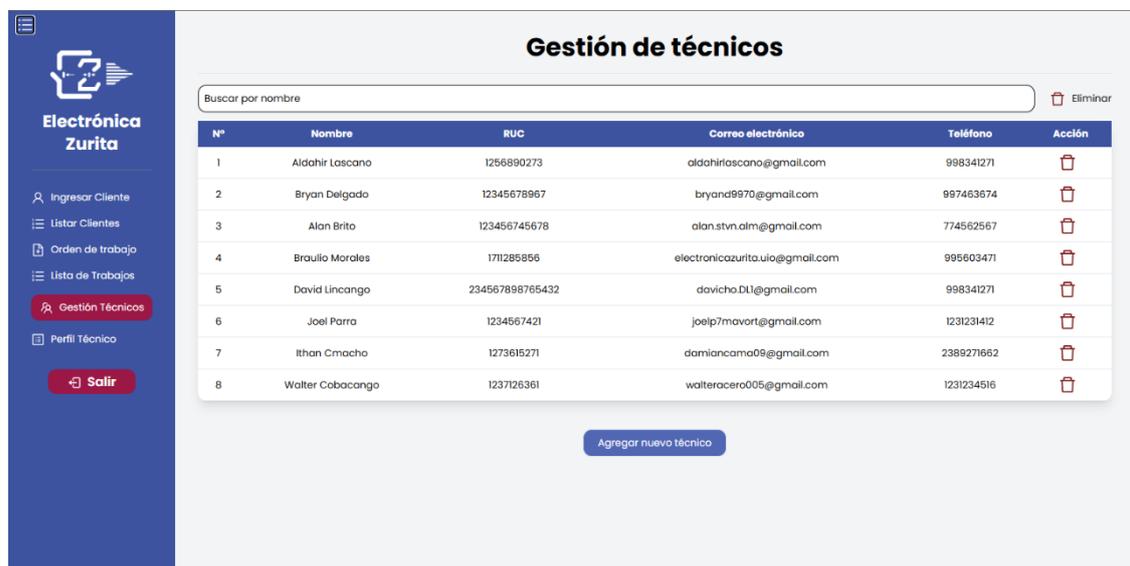
Figura 3.19 Registro de nuevo técnico



Figura 3.20 Confirmación de cuenta técnico

Así también, el primer técnico definido inicialmente en la base de datos del backend tiene la capacidad de ver los técnicos que estén registrados, estos son mostrados en una tabla

(Figura 3.21), contando también con la capacidad de eliminar a los técnicos gracias al botón que se encuentra en la columna de acciones.



N°	Nombre	RUC	Correo electrónico	Teléfono	Acción
1	Aldahir Lascano	1256890273	aldahirascano@gmail.com	998341271	
2	Bryan Delgado	12345678967	bryand9970@gmail.com	997463674	
3	Alan Brito	123456745678	alan.stvn.alm@gmail.com	774562567	
4	Braulio Morales	171285856	electronicazurita.uio@gmail.com	995603471	
5	David Lincango	234567898765432	davicho.DU@gmail.com	998341271	
6	Joel Parra	1234567421	joelp7mavort@gmail.com	1231231412	
7	Ithan Cmacho	1273615271	damiancam09@gmail.com	2389271662	
8	Walter Cobacango	1237126361	waltercero005@gmail.com	1231234516	

Figura 3.21 Visualización de técnicos

3.5 Sprint 4. Pruebas del componente frontend

En el sprint 4 se han obtenido los siguientes resultados:

- Pruebas de compatibilidad
- Pruebas de rendimiento
- Pruebas de responsividad

Pruebas de compatibilidad

Las pruebas de compatibilidad son parte de las estrategias de aseguramiento de la calidad puesto que permiten comprobar si su software funciona correctamente en distintas plataformas de manera que se pueda asegurar que la experiencia es similar y se cumple con las expectativas del equipo o navegador. Para realizar las pruebas de compatibilidad se ha utilizado la herramienta BrowserStack [41], gracias a la versión gratis de ésta herramienta se puede emular los diferentes sistemas operativos así como sus navegadores como lo muestra la Figura 3.22, de tal forma que se garantice que el componente web sea accesible y que la experiencia del usuario sea la misma en cualquier entorno [42]. En el

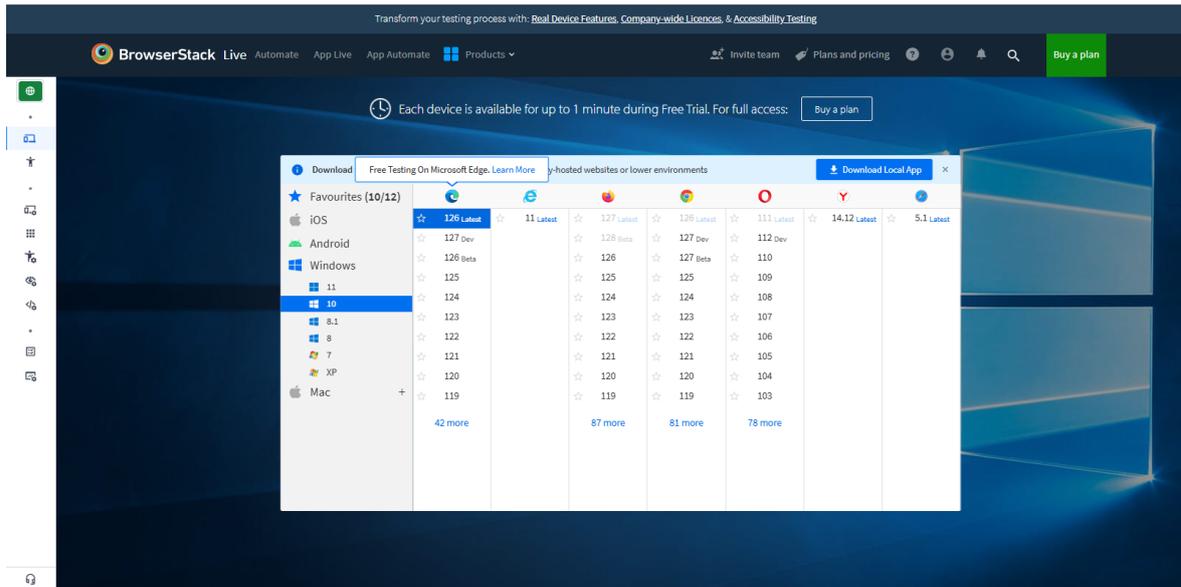


Figura 3.22 Herramienta Browserstack

Una vez que se ha realizado la prueba, a continuación, la **Tabla 3.1** muestra los resultados que se han obtenido, pudiendo visualizar que la compatibilidad ha sido 100% funcional.

Tabla 3.1 Resultados de las pruebas de compatibilidad

SISTEMA OPERATIVO	NAVEGADOR	VERSIÓN DEL NAVEGADOR	COMPATIBILIDAD
Windows 11	Google Chrome	126.0.6478.114	100% Funcional
Windows 11	Mozilla Firefox	127.0	100% Funcional
Windows 10	Opera GX	107.0.5045.606	100% Funcional
Windows 10	Microsoft Edge	126.0.6478.114	100% Funcional
Windows 8.1	Microsoft Edge	126.0.6478.114	100% Funcional
Mac Sonoma	Safari	17.1	100% Funcional

Cabe mencionar que la herramienta BrowserStack va dando resultados en tiempo real de ahí que se haya obtenido el 100% funcional, a continuación, desde la **Figura 3.23** hasta la **Figura 3.28** se muestran algunos resultados de la prueba de compatibilidad que se han realizado, las pruebas restantes pueden observarse en el **Pruebas de Compatibilidad**.



Figura 3.23 Pantalla inicio de sesión Google Chrome Windows 11

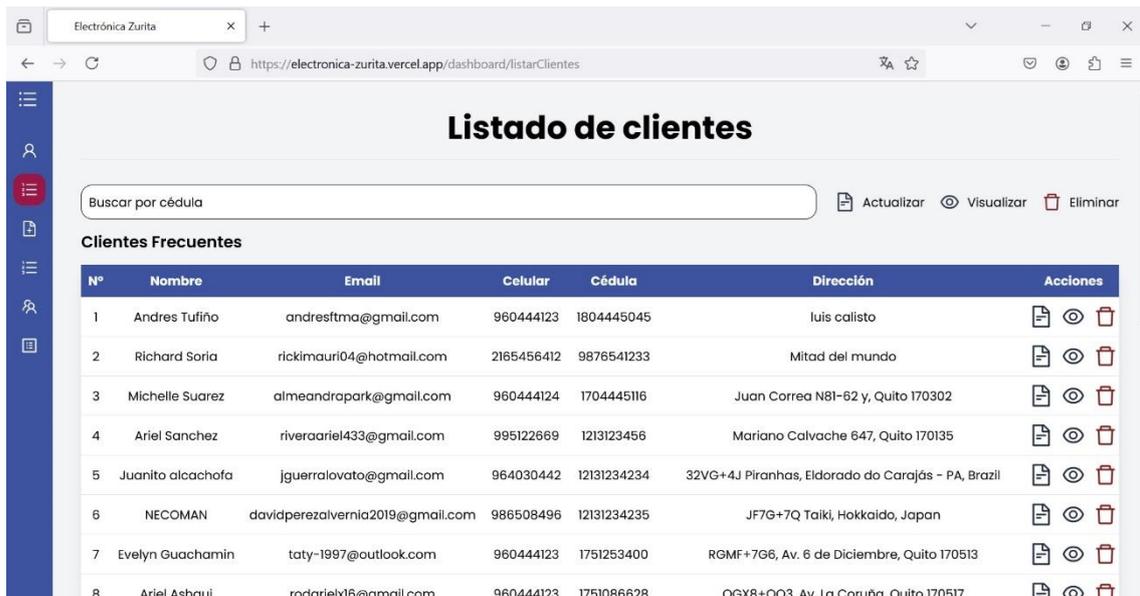


Figura 3.24 Pantalla listado de clientes Mozilla Firefox Windows 11

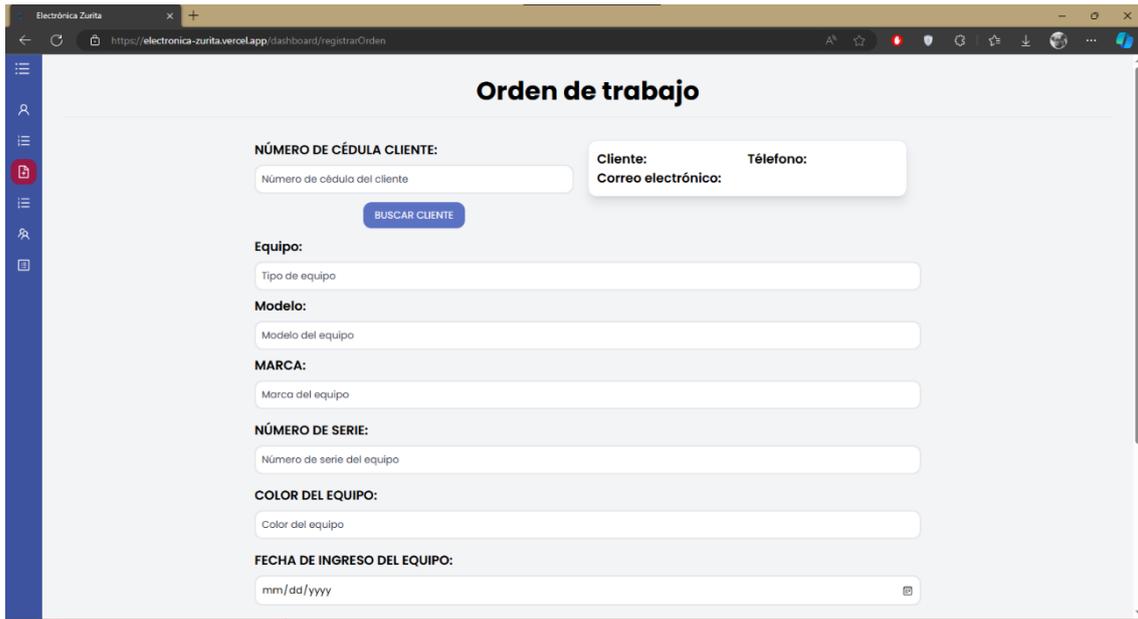


Figura 3.25 Pantalla orden de trabajo Microsoft Edge Windows 10

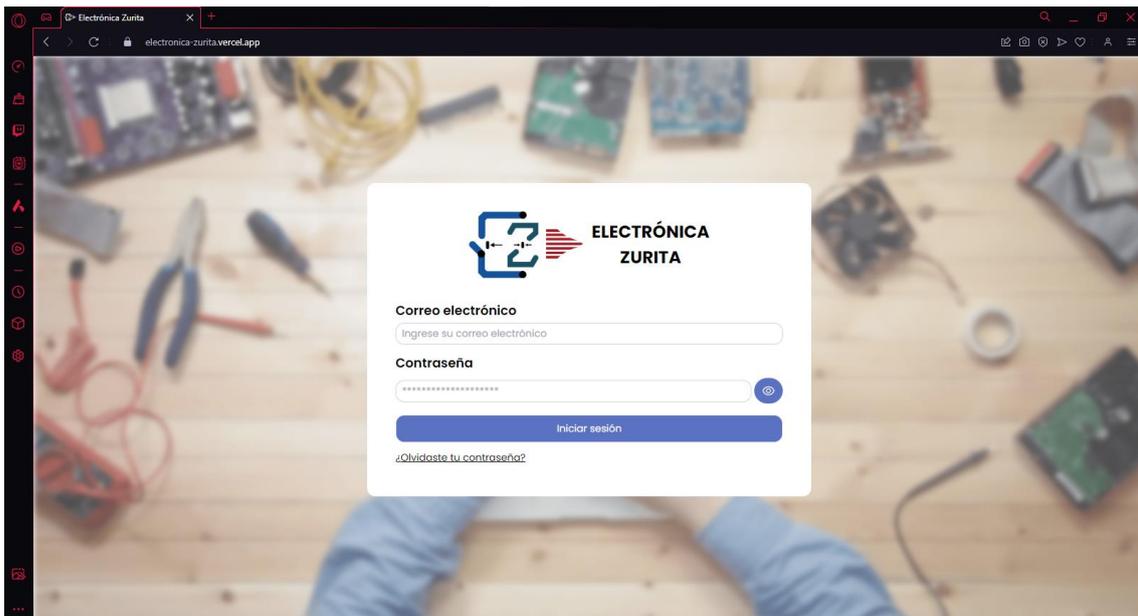


Figura 3.26 Pantalla inicio de sesión Opera GX Windows 10

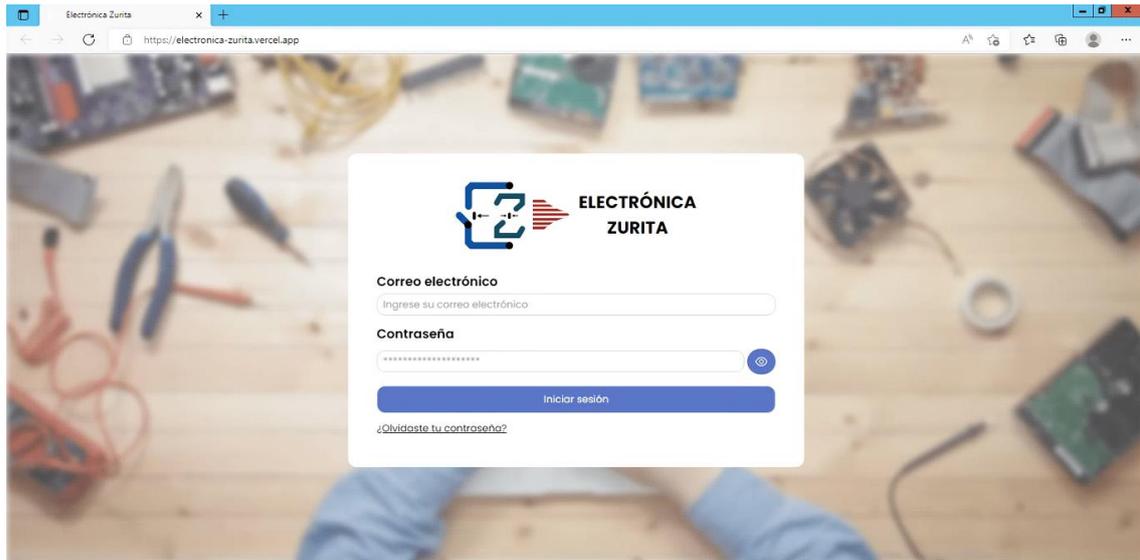


Figura 3.27 Pantalla inicio de sesión Microsoft Edge Windows 8.1

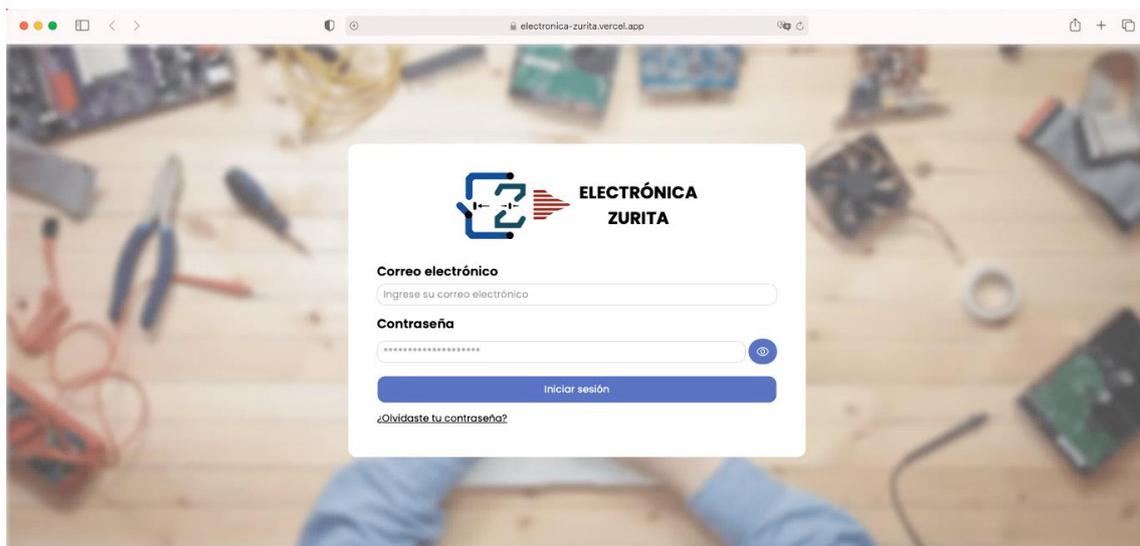


Figura 3.28 Pantalla inicio de sesión Safari Mac Sonoma

Pruebas de rendimiento

Para realizar las pruebas de rendimiento se ha utilizado la herramienta Google Lighthouse, Google Lighthouse es una extensión de Google que permite hacer un análisis de un sitio en específico con la ayuda de PageSpeed Insights al hacer un análisis del rendimiento del componente [43], gracias a esta herramienta se puede hacer un análisis del rendimiento del sistema, la velocidad de este y saber si el componente frontend tiene buenas prácticas para hacerle frente a ataques XSS dando además recomendaciones básicas sobre optimización en buscadores [44]. La **Figura 3.29** muestra el resultado de las pruebas proporcionadas por PageSpeed Insights, el resultado de la prueba de rendimiento más los detalles sobre esta se muestran en la **Figura 3.30**, mientras que los diagnósticos que

obtenido de la misma herramienta sobre la prueba de rendimiento se muestran en la **Figura 3.31**.

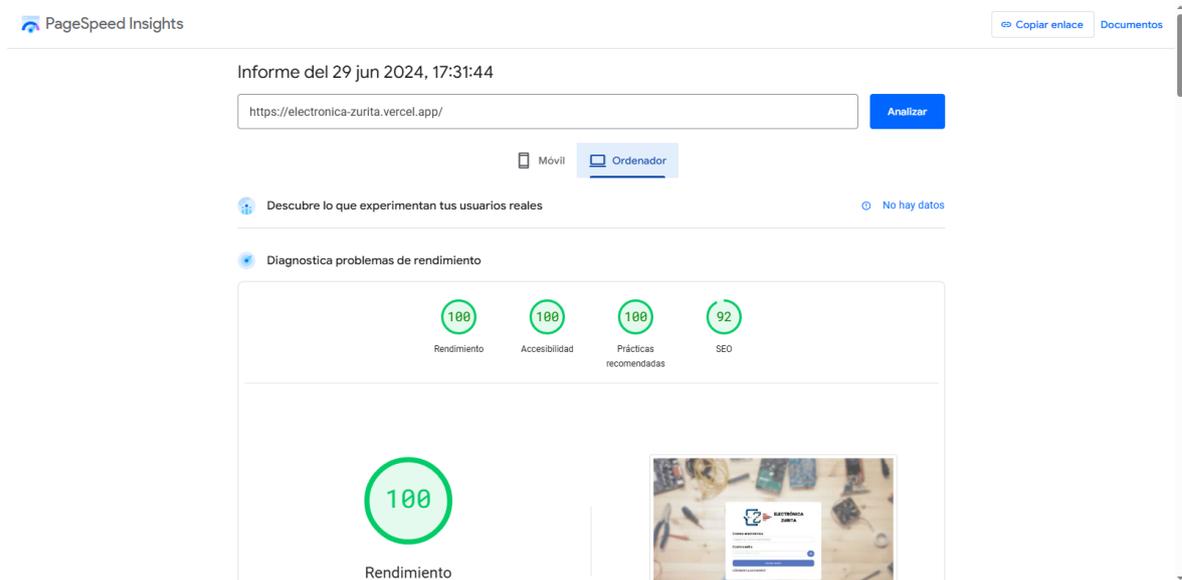


Figura 3.29 Resultado pruebas de PageSpeed Insights

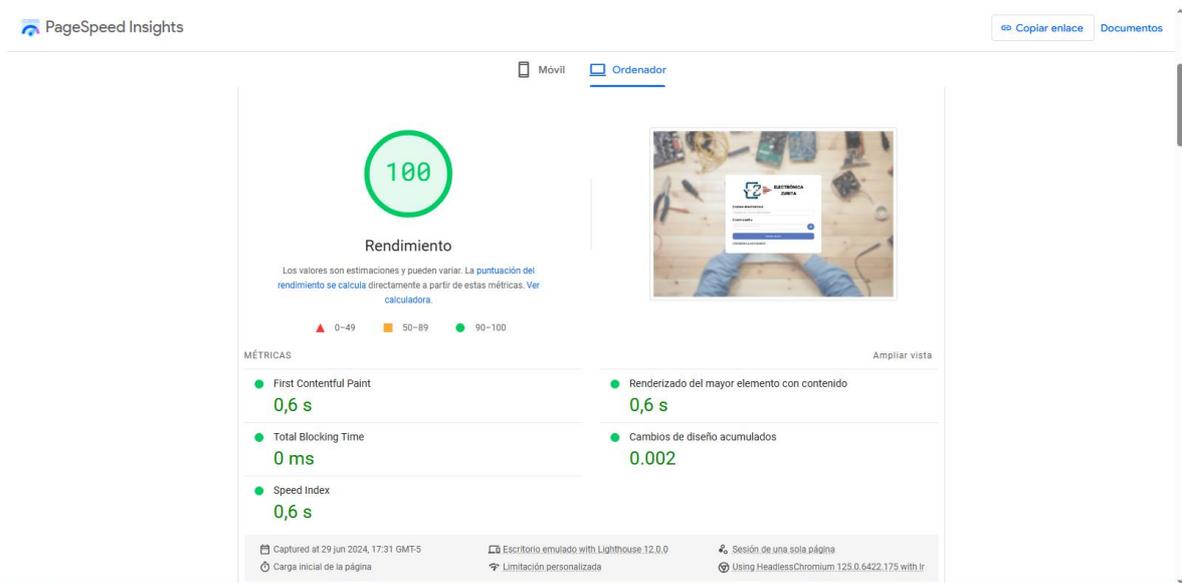


Figura 3.30 Resultado pruebas de rendimiento

Móvil Ordenador

DIAGNÓSTICOS

- Precargar la imagen de renderizado del mayor elemento con contenido
- Los elementos de imagen no tienen `width` y `height` explícitos
- Publica imágenes con formatos de próxima generación – Ahorro potencial de 1221 KiB
- Usa un tamaño adecuado para las imágenes – Ahorro potencial de 19 KiB
- Codifica las imágenes de forma eficaz – Ahorro potencial de 1029 KiB
- Reduce el contenido JavaScript que no se use – Ahorro potencial de 99 KiB
- Evitar cambios de diseño importantes – 2 cambios de diseño detectados
- El tiempo de respuesta inicial del servidor fue breve – El documento raíz ha tardado 40 ms
- Evita cargas útiles de red de gran tamaño – Tamaño total: 1549 KiB
- Evita un tamaño excesivo de DOM – 27 elementos
- Evita encadenar solicitudes críticas – Se han encontrado 2 cadenas
- Tiempo de ejecución de JavaScript – 0,0 s
- Minimiza el trabajo del hilo principal – 0,1 s
- Reducir el uso de código de terceros – El código de un tercero ha bloqueado el hilo principal durante 0 ms
- Renderizado del mayor elemento con contenido – 640 ms

Figura 3.31 Diagnóstico prueba de rendimiento

Prueba de responsividad

Para realizar las pruebas de responsividad del componente frontend, se ha utilizado la herramienta Screenfly. Esta herramienta permite probar el componente en diferentes resoluciones de una amplia variedad de dispositivos, como computadoras, dispositivos móviles, televisores y tabletas [45]. El dispositivo, la resolución utilizada para hacer las pruebas, así como el resultado de la prueba, se han detallado en la **Tabla 3.2** y de la **Figura 3.32** a la **Figura 3.40** se muestran los resultados de las pruebas de responsividad.

Tabla 3.2 Dispositivos utilizados para la prueba de responsividad

DISPOSITIVO	RESOLUCION	Resultado
Computador	1024 x 600	100% Responsive
Computador	1600 x 900	100% Responsive
Computador	1920 x 1080	100% Responsive
Tableta	800 x 480	100% Responsive
Tableta	1280 x 800	100% Responsive
Móvil	414 x 736	100% Responsive
Móvil	720 x 1280	100% Responsive
Televisor	2048 x 1080	100% Responsive
Televisor	3840 x 2160	100% Responsive

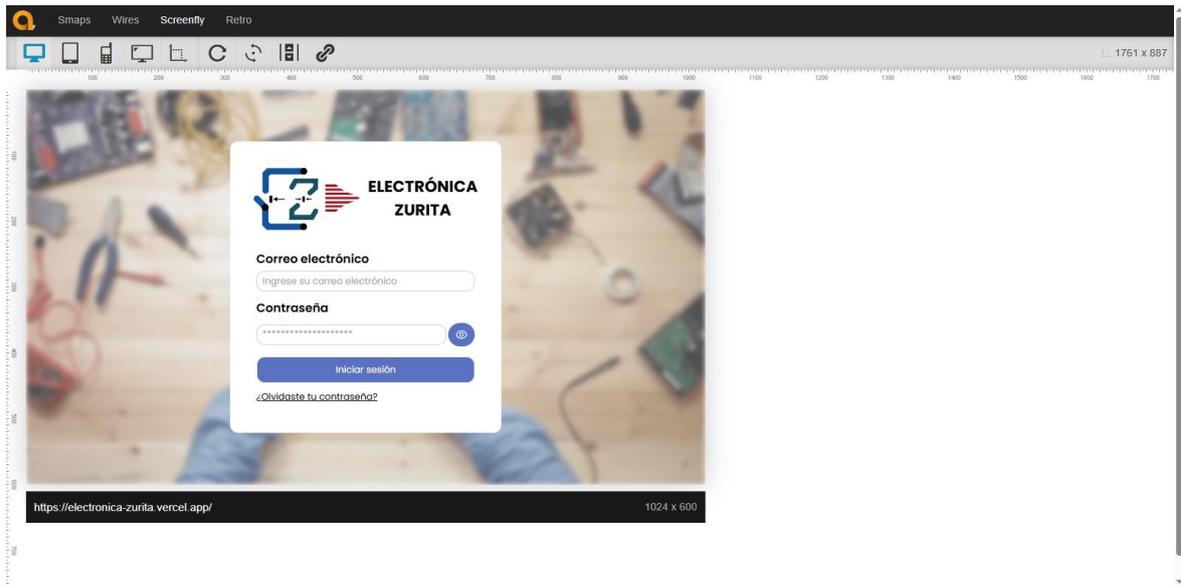


Figura 3.32 Prueba de responsividad computador 1024 x 600

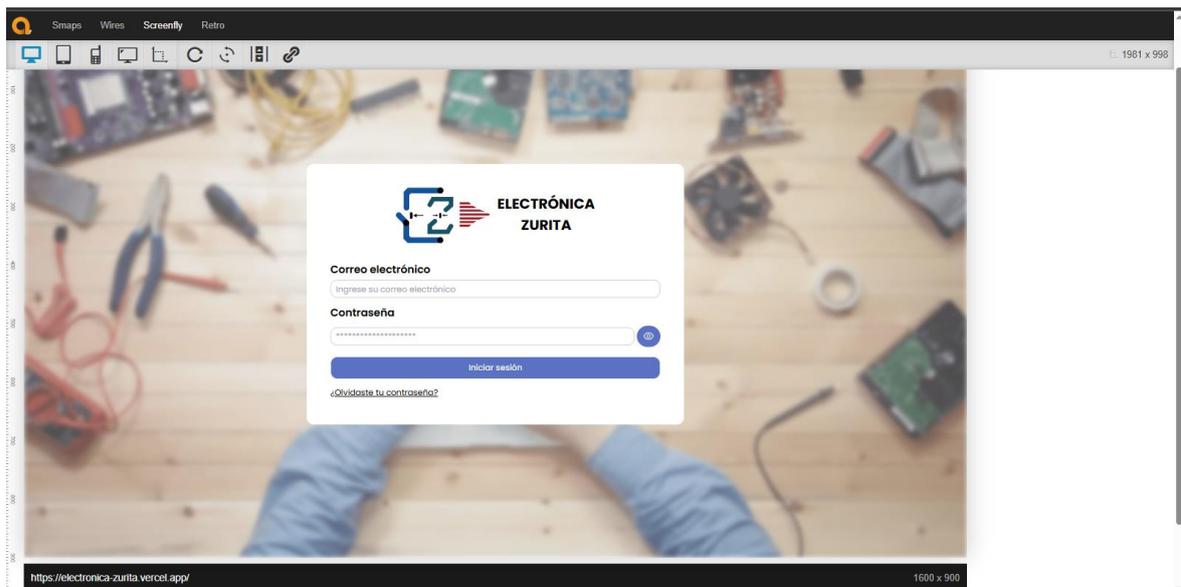


Figura 3.33 Prueba de responsividad computador 1600 x 900

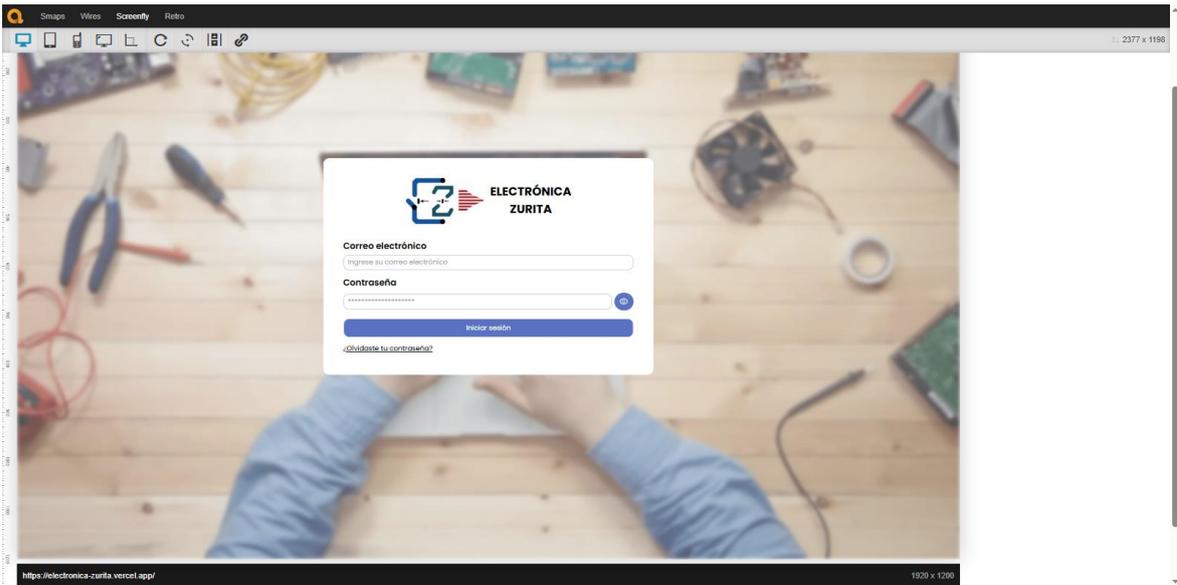


Figura 3.34 Prueba de responsividad computador 1920 x 1080

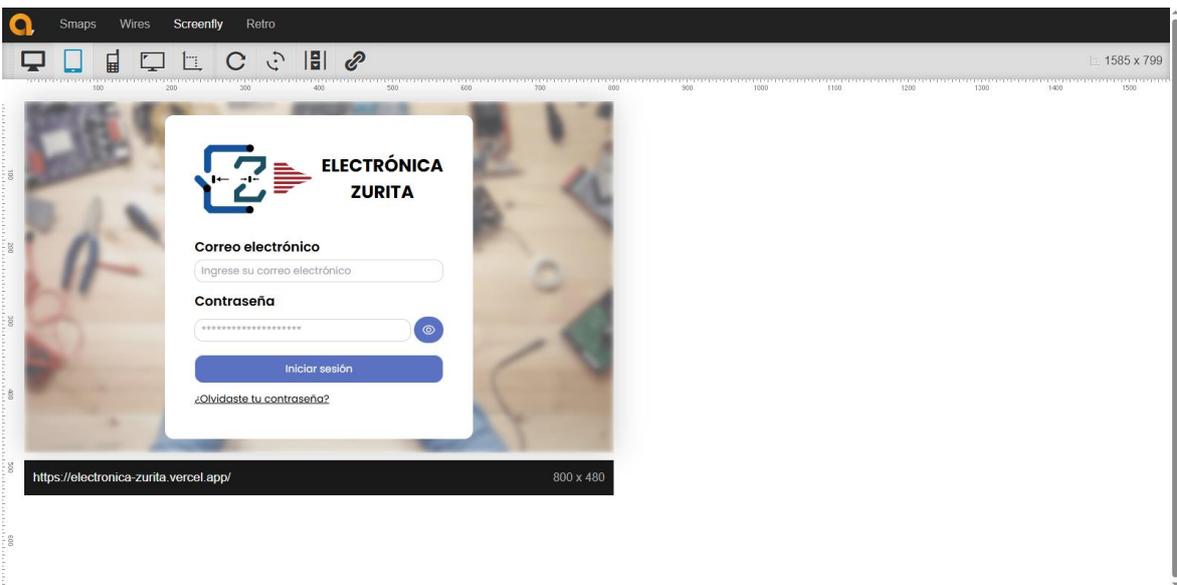


Figura 3.35 Prueba de responsividad tableta 800 x 480

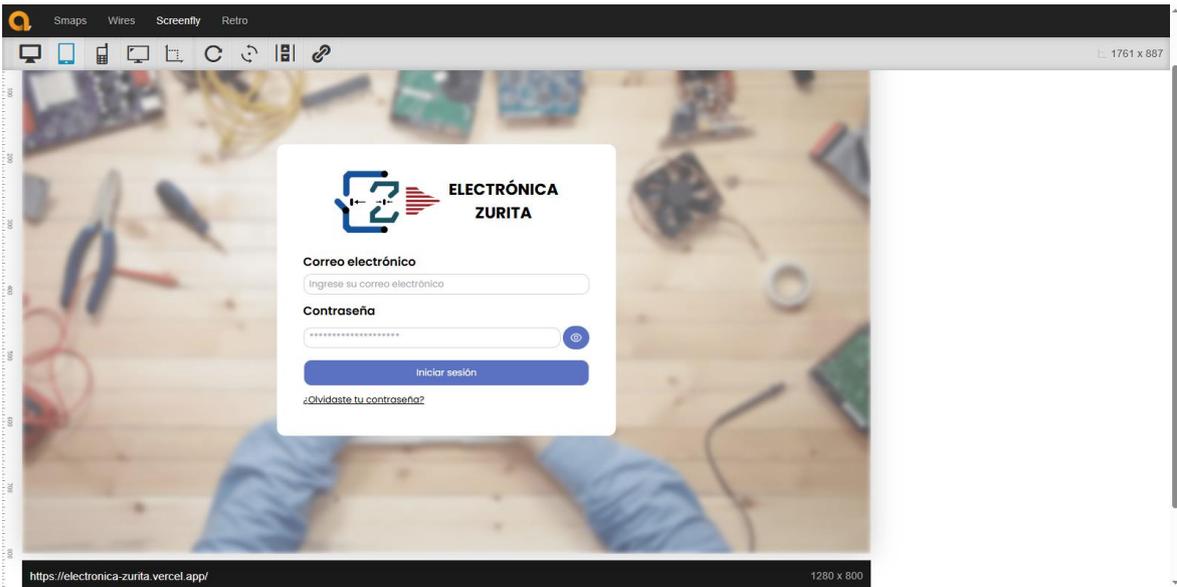


Figura 3.36 Prueba de responsividad tableta 1280 x 800

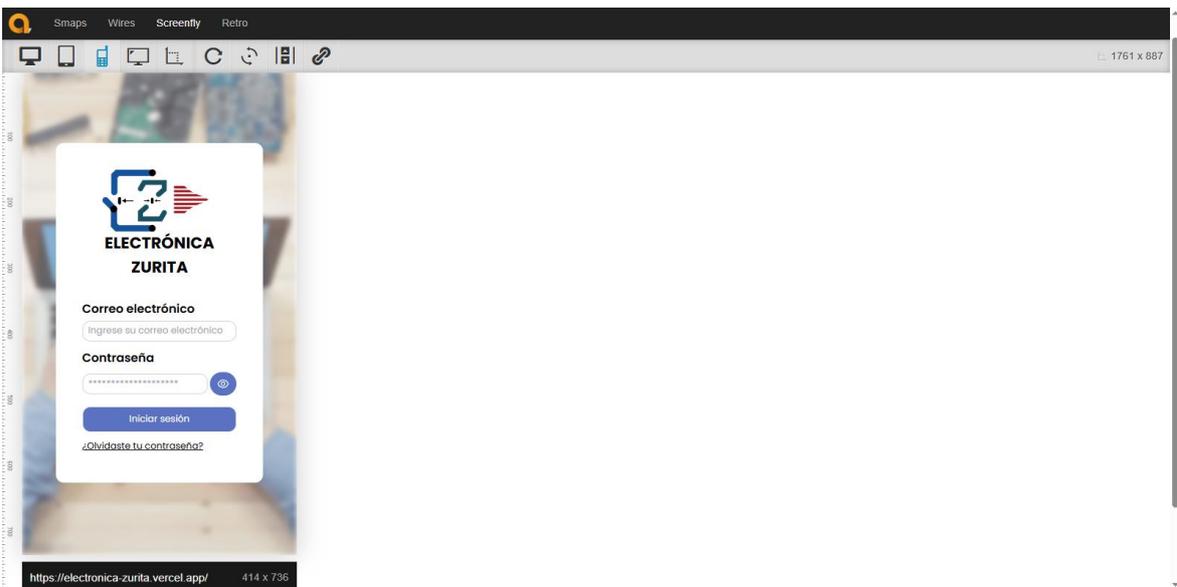


Figura 3.37 Prueba de responsividad móvil 414 x 736

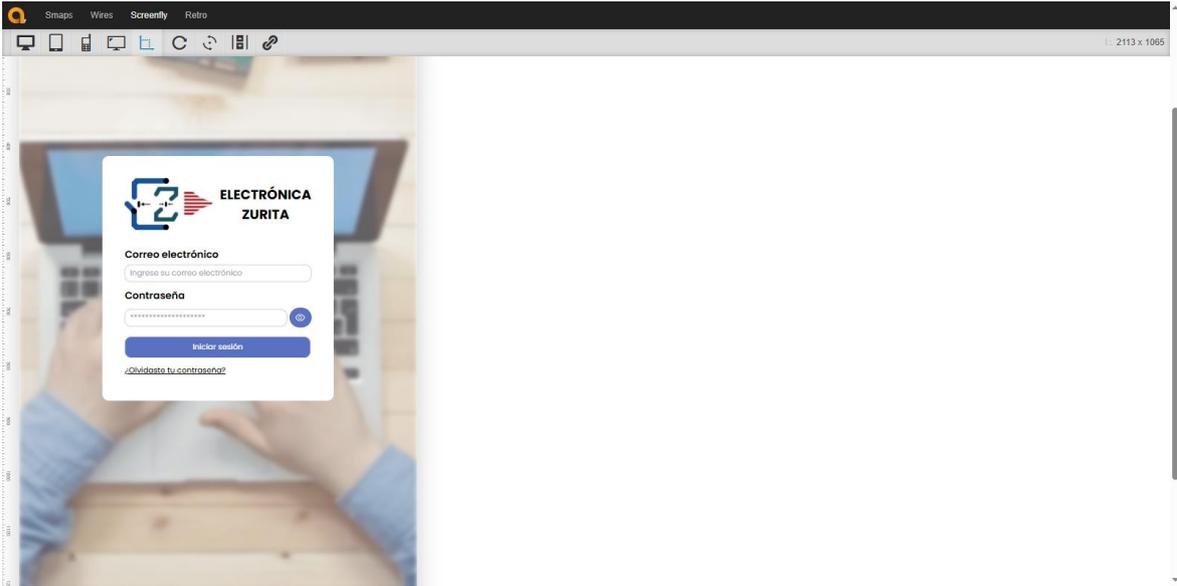


Figura 3.38 Prueba de responsividad móvil 720 x 1280

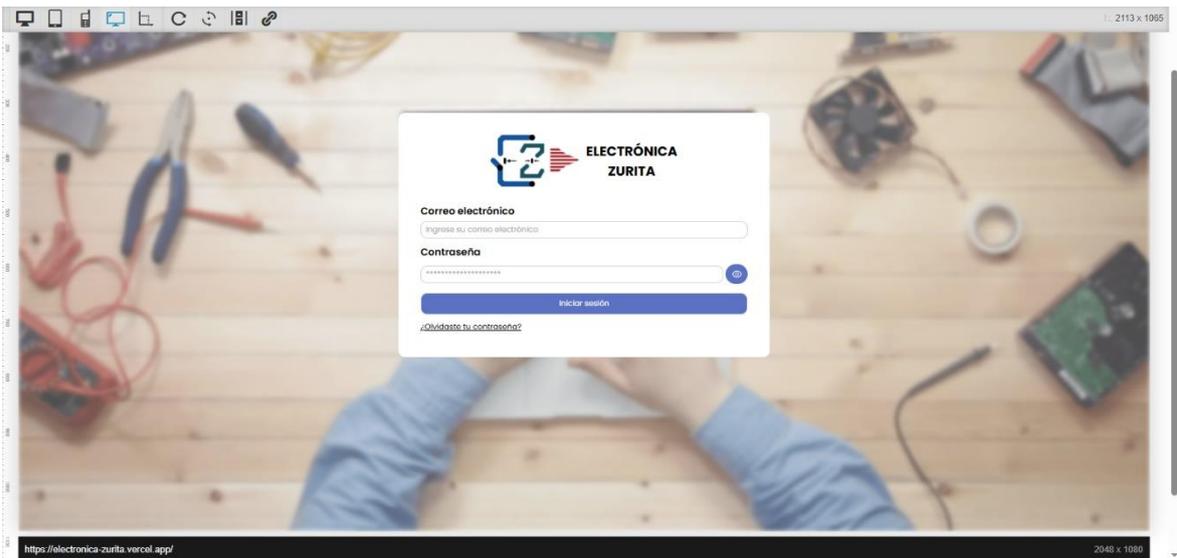


Figura 3.39 Prueba de responsividad televisor 2048 x 1080

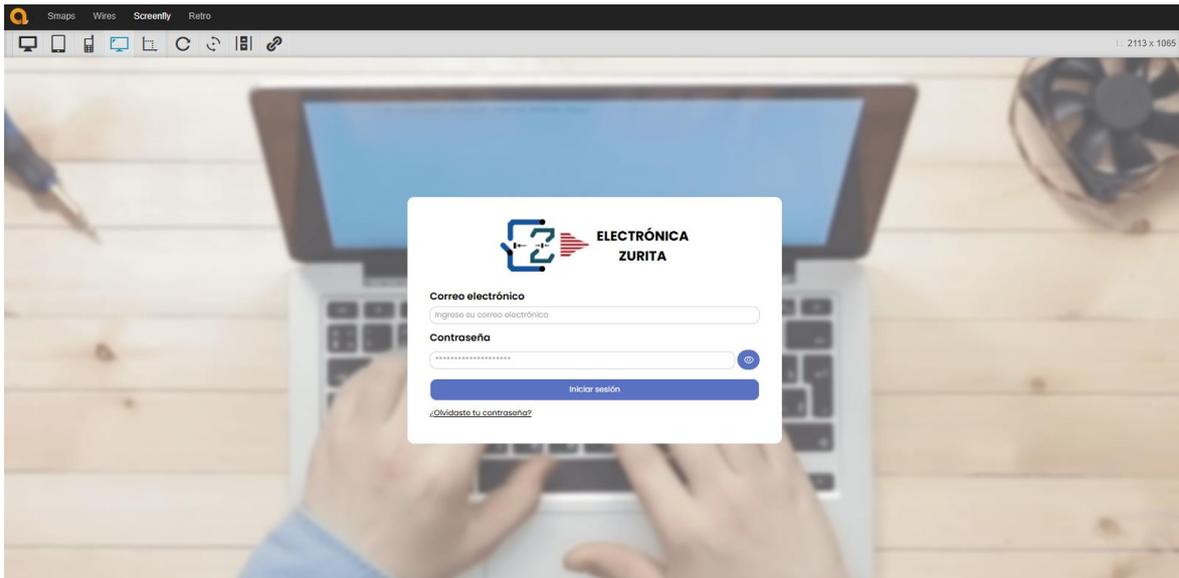
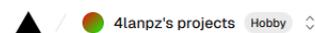


Figura 3.40 Prueba de responsividad televisor 3840 x 2160

3.6 Sprint 5. Despliegue

Finalizado la etapa de desarrollo y codificación del componente frontend, se prosigue a la ejecución del despliegue. El despliegue del presente proyecto se desarrolla con ayuda de la herramienta Vercel, la cual es una herramienta fácil de configurar, que permite obtener un dominio gratis para el proyecto para que éste pueda estar disponible para todos los usuarios [46].

Los pasos importantes para el despliegue del componente frontend como la selección del repositorio para el despliegue se muestran de la **Figura 3.41** hasta la **Figura 3.44**, mientras que el despliegue final del proyecto se presenta en la **Figura 3.45**.



Let's build something new.

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

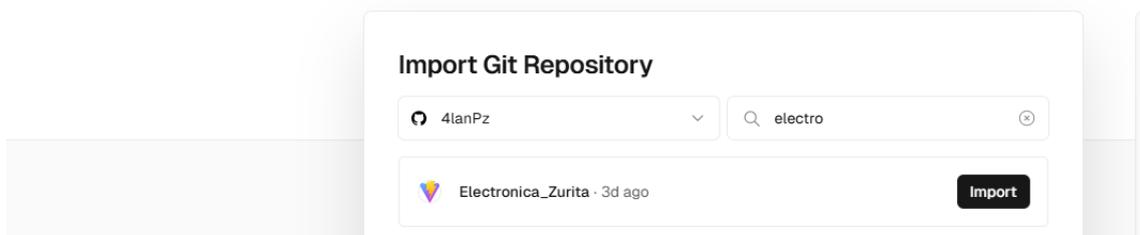


Figura 3.41 Selección repositorio Vercel

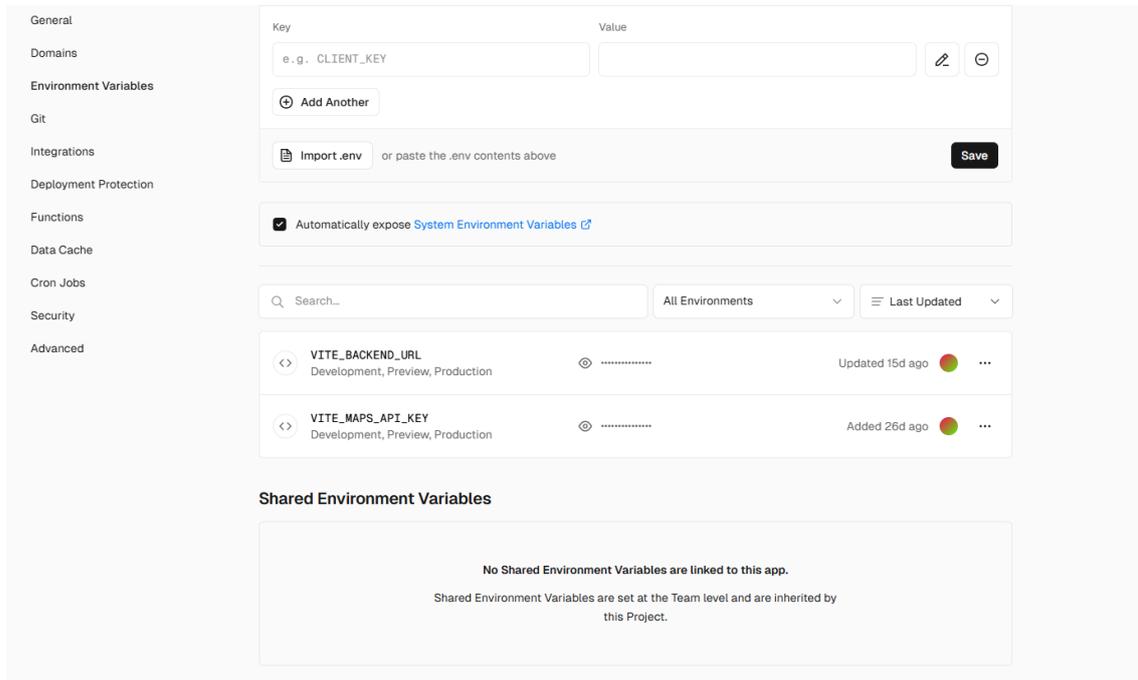


Figura 3.42 Configuración variables de entorno

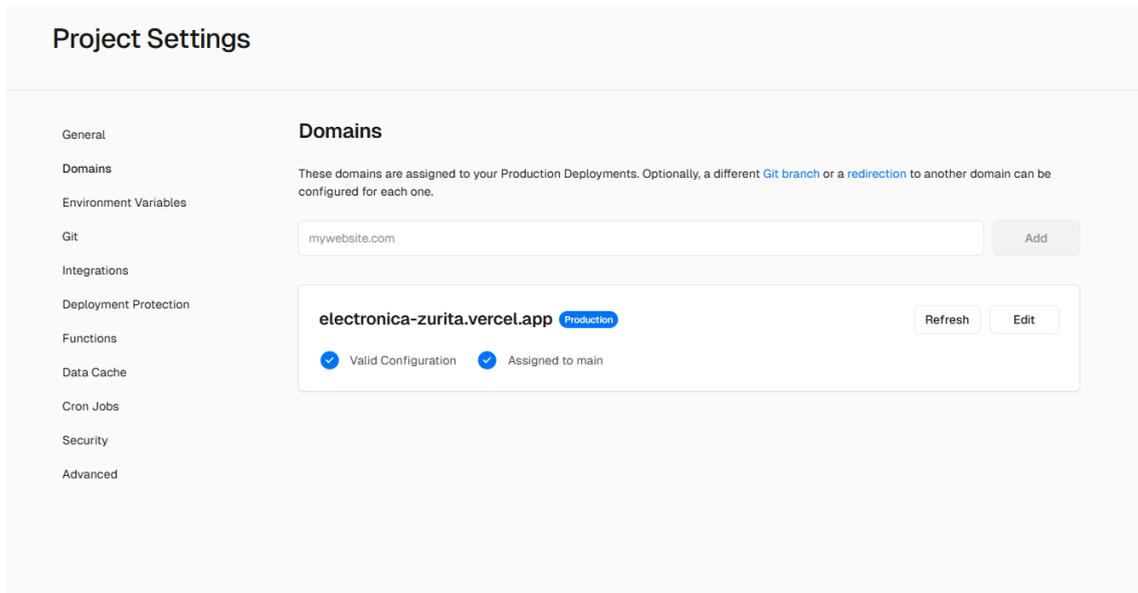


Figura 3.43 Configuración dominio personalizado

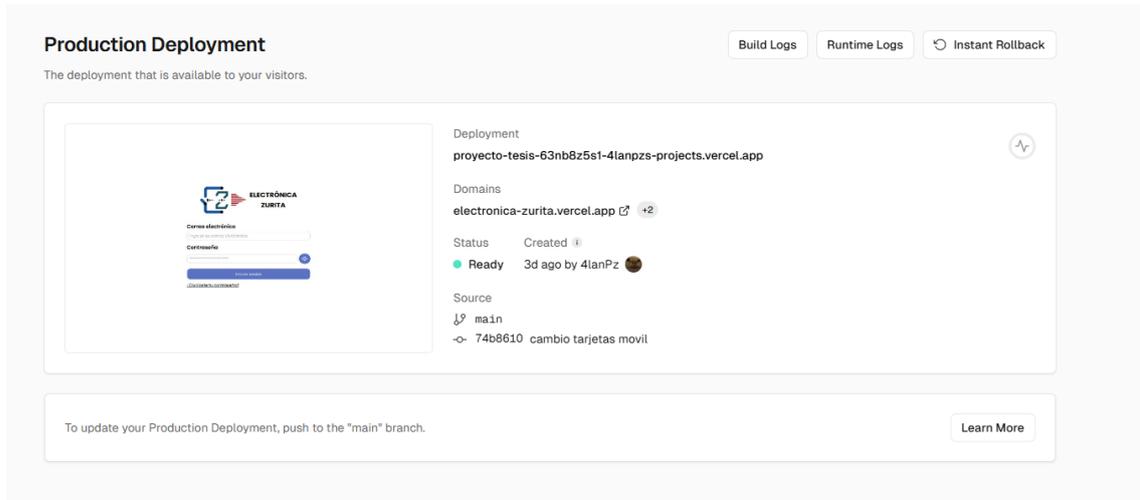


Figura 3.44 Despliegue del componente en Vercel

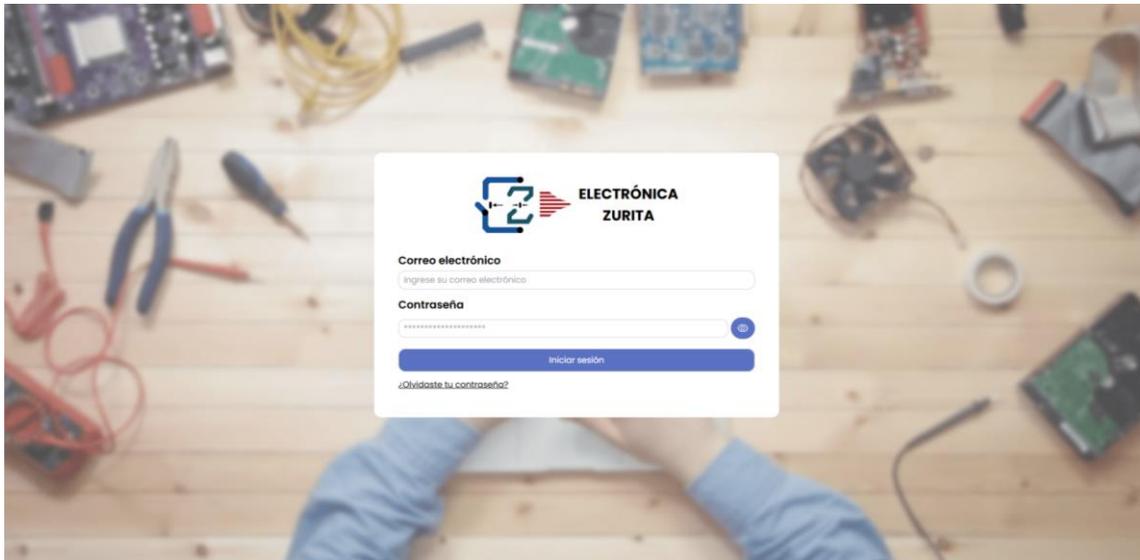


Figura 3.45 Despliegue componente frontend

4 CONCLUSIONES

En la presente sección se presentan las conclusiones obtenidas durante todo el proceso de desarrollo del componente frontend:

- El proyecto ha logrado cumplir exitosamente con cada uno de los sprints definidos gracias a la implementación de la metodología Scrum. Esto a pesar de que hubo desafíos y retrasos durante el proceso de desarrollo, la metodología ha demostrado ser efectiva para gestionar y superar obstáculos, permitiendo así la entrega del proyecto sin ningún tipo de contratiempo.
- El proyecto del componente frontend para el centro de reparaciones "Electrónica Zurita" se ha desarrollado satisfactoriamente gracias a la recopilación de requerimientos de usuario. Este proceso ha sido fundamental para definir claramente el alcance del componente y asegurar que cumple con las expectativas y necesidades del usuario final.
- Las pruebas realizadas en el componente frontend han sido fundamentales para garantizar el óptimo funcionamiento ya que, con resultados favorables obtenidos, se comprueba que el componente cumple con los estándares de rendimiento y usabilidad esperados por el usuario.
- La implementación de diversas herramientas como librerías, bibliotecas y el framework React, ha simplificado notablemente el proceso de desarrollo del componente frontend, ya que, gracias a estas se asegura de que cualquier ajuste necesario pueda ser realizado con agilidad y sin comprometer la calidad del resultado final del componente.
- La elección de la herramienta para poder realizar el despliegue puede ser una tarea muy importante, ya que, si la herramienta que se utiliza no es lo suficientemente capaz de poder manejar varios usuarios a la vez, el proyecto puede llegar a tener malas aceptaciones y no llegar al resultado que espera el usuario final.

5 RECOMENDACIONES

En la siguiente sección se detallan las recomendaciones que se han generado en el tiempo de desarrollo del componente frontend.

- Se recomienda realizar un monitoreo continuo de las funciones del componente para recopilar información de retroalimentación valiosa que dar una guía para mejorar y optimizar las funciones según lo requiera el usuario a futuro.
- Se aconseja al usuario final que realice pruebas de uso para poder familiarizarse con los diferentes mensajes que pueden aparecer durante la interacción con el componente frontend. Esto incluye mensajes de error y confirmación, para evitar confusiones durante el uso en entornos laborales.
- A pesar de que el componente está construido con herramientas y librerías modernas, es recomendable mantenerse actualizado sobre las tendencias y actualizaciones tecnológicas. Esto garantiza que las herramientas utilizadas permanezcan compatibles y eficientes, evitando posibles problemas de librerías o herramientas defectuosas u obsoletas.

6 REFERENCIAS BIBLIOGRAFICAS

- [1] Imaginamos, «¿Cómo la tecnología está transformando el mundo de los negocios?,» LinkedIn, 27 Abril 2023. [En línea]. Available: <https://es.linkedin.com/pulse/c%C3%B3mo-la-tecnolog%C3%ADa-est%C3%A1-transformando-el-mundo-de-los-negocios>. [Último acceso: 23 Mayo 2024].
- [2] MongoDB, «¿Qué es MongoDB?,» MongoDB, Inc., [En línea]. Available: <https://www.mongodb.com/es/company/what-is-mongodb>. [Último acceso: 23 Mayo 2024].
- [3] RedeWeb, «What are Electronic Devices?,» Spanish Electronics, 25 Octubre 2023. [En línea]. Available: <https://www.redeweb.com/en/actualidad/que-son-los-aparatos-electronicos/>. [Último acceso: 23 Mayo 2024].
- [4] Amazon, «¿Qué es una API RESTful?,» Amazon, [En línea]. Available: <https://aws.amazon.com/es/what-is/restful-api/>. [Último acceso: 23 Mayo 2024].
- [5] V. Trafaniuc, «¿Sabes qué es Google APIs? Descúbrelo ahora mismo,» Maplink, 10 Enero 2022. [En línea]. Available: <https://maplink.global/blog/es/que-son-google-apis/>. [Último acceso: 23 Mayo 2024].
- [6] Apuntes de programador, «Desarrollo Web Moderno: Tendencias y Herramientas,» Apuntes De, [En línea]. Available: <https://apuntes.de/nodejs-desarrollo-web/desarrollo-web-moderno/#gsc.tab=0>. [Último acceso: 24 Mayo 2024].
- [7] B. Hamidian Fernández, «¿Por qué los sistemas de información son esenciales?,» *ANUARIO*, vol. 38, pp. 161-183, 2015.
- [8] Deyimar, «Qué es React: definición, características y funcionamiento,» Hostinger Tutoriales, 29 Junio 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-react>. [Último acceso: 23 Mayo 2024].
- [9] Developer Mozilla, «Mozilla Developer,» [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics. [Último acceso: 24 Mayo 2024].
- [10] J. Sierra y S. De Roy, «¿Qué es Tailwind CSS? Guía para principiantes,» FreecodeCamp, 26 Enero 2023. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/que-es-tailwind-css-guia-para-principiantes/>. [Último acceso: 24 Mayo 2024].
- [11] Amazon Web Services, «¿Qué es JavaScript (JS)?,» Amazon Web Services, Inc., [En línea]. Available: <https://aws.amazon.com/es/what-is/javascript/>. [Último acceso: 25 Mayo 2024].
- [12] J. Guaña y R. Barger, «Cómo usar Axios con React: La guía definitiva (2021),» FreecodeCamp, 14 Julio 2022. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-usar-axios-con-react/>. [Último acceso: 25 Mayo 2024].

- [13] Vite, «Introducción,» Vide Dev, [En línea]. Available: <https://es.vitejs.dev/guide/>. [Último acceso: 26 Mayo 2024].
- [14] Google Maps Platform, «Maps JavaScript API,» Google Developer, [En línea]. Available: <https://developers.google.com/maps/documentation/javascript>. [Último acceso: 26 Mayo 2024].
- [15] Kschool, «Los 10 principios de usabilidad para diseño de interfaces de usuario,» Kschool, 15 Abril 2024. [En línea]. Available: <https://kschool.com/blog/usabilidad-ux/los-10-principios-de-usabilidad-para-diseno-de-interfaces-de-usuario/>. [Último acceso: 26 Mayo 2024].
- [16] S. Garrido Sotomayor, «Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa,» IEBS, 14 Noviembre 2023. [En línea]. Available: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>. [Último acceso: 28 Mayo 2024].
- [17] M. Rehkopf, «¿Qué son los sprints en la gestión de proyectos?,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/sprints#:~:text=gesti%C3%B3n%20de%20proyectos%3F-Un%20sprint%20es%20un%20per%C3%ADodo%20breve%20de%20tiempo%20fijo%20en,con%20menos%20quebraderos%20de%20cabeza..> [Último acceso: 28 Mayo 2024].
- [18] Atlassian, «Qué es scrum y cómo empezar,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/scrum#:~:text=Scrum%20es%20un%20marco%20de,de%20valores%2C%20principios%20y%20pr%C3%A1cticas..> [Último acceso: 28 Mayo 2024].
- [19] Plain Concepts, «Product Owner: ¿Qué es y cuáles son sus funciones en Scrum?,» Plain Concepts, 22 Febrero 2022. [En línea]. Available: <https://www.plainconcepts.com/es/product-owner/#:~:text=En%20Scrum%2C%20el%20Product%20Owner,y%20se%20tiene%20que%20hacer..> [Último acceso: 28 Mayo 2024].
- [20] A. Raeburn, «¿Qué es un Scrum Master y cuál es su función?,» Asana, Inc., 2 Febrero 2024. [En línea]. Available: <https://asana.com/es/resources/scrum-master>. [Último acceso: 28 Mayo 2024].
- [21] Miguel, «Scrum Framework Operations: roles y responsabilidades,» We Are Marketing, 19 Octubre 2022. [En línea]. Available: <https://www.wearemarketing.com/es/blog/roles-de-un-equipo-scrum.html#:~:text=El%20Scrum%20Master%20lidera%20al,trabajo%20es%20desbloquear%20y%20facilitar..> [Último acceso: 2 Junio 2024].
- [22] J. Roche, «Artefactos Scrum: las 3 herramientas clave de gestión,» Deloitte Touche Tohmatsu Limited, [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>. [Último acceso: 2 Junio 2024].

- [23] Asana Team, «Guía de 6 pasos para la recopilación de requisitos para asegurar el éxito de tu proyecto,» Asana Team, 13 Noviembre 2022. [En línea]. Available: <https://asana.com/es/resources/requirements-gathering>. [Último acceso: 2 Junio 2024].
- [24] M. Rehkopf, «Historias de usuario con ejemplos y plantilla,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 2 Junio 2024].
- [25] «Scrum: ¿Qué es el Product Backlog?,» Programación y más, [En línea]. Available: <https://programacionymas.com/blog/scrum-product-backlog>. [Último acceso: 3 Junio 2024].
- [26] Atlassian, «Qué es un backlog de sprint, cuáles son sus ventajas y cómo crear uno,» 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: 03 Junio 2024].
- [27] «Qué es un mockup,» Miro, [En línea]. Available: <https://miro.com/es/mockup/que-es-mockup/>. [Último acceso: 3 Junio 2024].
- [28] «¿Qué es Figma y para qué sirve?,» Centro de Estudios de Innovación Diseño y Marketing, [En línea]. Available: <https://cei.es/que-es-figma/>. [Último acceso: 3 Junio 2024].
- [29] D.R, «Diseño arquitectónico,» [En línea]. Available: https://www.sites.upiicsa.ipn.mx/estudiantes/academia_de_informatica/ingeniera_de_diseño/docs/u2/s3/DISEÑO%20ARQUITECTÓNICO.pdf. [Último acceso: 4 Junio 2024].
- [30] S. Fernández, «¿Cómo utilizar el patrón MVC (Model View Controller) en 2023?,» Sebastian Fernandez, 23 Noviembre 2022. [En línea]. Available: <https://www.sebastianfdz.com/articles/que-es-model-view-controller>. [Último acceso: 4 Junio 2024].
- [31] F. Flores, «Qué es Visual Studio Code y qué ventajas ofrece,» OpenWebinars.net, [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. [Último acceso: 4 Febrero 2024].
- [32] React Dev, «Componentes de React DOM,» Meta Platforms, Inc, [En línea]. Available: <https://es.react.dev/reference/react-dom/components>. [Último acceso: 4 Junio 2024].
- [33] Tailwind CSS, «Rapidly build modern websites without ever leaving your HTML.,» Tailwind CSS, [En línea]. Available: <https://tailwindcss.com/>. [Último acceso: 18 Junio 2024].
- [34] J. Arque, «Ya no querrás otro: 7 motivos por los que deberías utilizar Microsoft Edge ahora que usa como base Chromium,» Businnes Insider, 23 Mayo 2021. [En línea].

Available: <https://www.businessinsider.es/7-motivos-utilizar-microsoft-edge-version-chromium-868999>. [Último acceso: 10 Junio 2024].

- [35] BEEKAI, «React Hook Form Performant, flexible and extensible forms with easy-to-use validation.,» BEEKAI, [En línea]. Available: <https://react-hook-form.com/get-started>. [Último acceso: 10 Junio 2024].
- [36] «react-modal,» NPM, 18 Octubre 2022. [En línea]. Available: <https://www.npmjs.com/package/react-modal>. [Último acceso: 10 Junio 2024].
- [37] «react-icons,» NPM, 7 Mayo 2024. [En línea]. Available: <https://www.npmjs.com/package/react-icons>. [Último acceso: 10 Junio 2024].
- [38] C. Pazmiño, «Formik y Yup: valida formularios en React sin complicarte,» Listopro Community, 3 Noviembre 2023. [En línea]. Available: <https://community.listopro.com/formik-y-yup-valida-formularios-en-react-sin-complicarte/#:~:text=Es%20una%20librer%C3%ADa%20que%20simplifica,f%C3%A1cilmente%20el%20estado%20del%20formulario..> [Último acceso: 18 Junio 2024].
- [39] J. Peralta, «Manejar Formularios con Formik y Yup en React,» Medium, 24 Marzo 2020. [En línea]. Available: <https://jperaza.medium.com/validar-formularios-con-formik-y-yup-en-react-703b5630cd35>. [Último acceso: 18 Junio 2024].
- [40] DESARROLLO WEB, TECNOLOGÍA DE EXPERIENCIA, «Vercel: Desarrollar, Previsualizar, Enviar,» Aplyca, 26 Enero 2023. [En línea]. Available: <https://www.aplyca.com/blog/blog-que-es-vercel-desarrollar-previsualizar-enviar>. [Último acceso: 20 Junio 2024].
- [41] BrowserStack, «App & Browser Testing Made Easy,» BrowserStack, [En línea]. Available: <https://www.browserstack.com/>. [Último acceso: 23 Junio 2024].
- [42] M. K. C. a. E. I. M. A. Mascheroni, «Automatización de pruebas de compatibilidad web en un entorno de desarrollo continuo de software,» de *17º Simposio Argentino de Ingeniería en Software (ASSE 2016)*, Buenos Aires, Argentina, 2016.
- [43] Google for Developers, «Lighthouse,» Google for Developers, [En línea]. Available: <https://developer.chrome.com/docs/lighthouse/overview>. [Último acceso: 26 Junio 2024].
- [44] Google for Developers, «Ensure CSP is effective against XSS attacks,» Google for Developers, [En línea]. Available: https://developer.chrome.com/docs/lighthouse/best-practices/csp-xss/?utm_source=lighthouse&utm_medium=lr. [Último acceso: 26 Junio 2024].
- [45] Screenfly, «Screenfly,» Screenfly, [En línea]. Available: así como el resultado de la prueba . [Último acceso: 2024 Julio 29].

7 ANEXOS

A continuación, se listan todos los anexos que respaldan el presente proyecto de titulación.

- ANEXO I. Turnitin
- ANEXO I. Manual técnico
- ANEXO II. Manual de usuario
- ANEXO III. Manual de instalación

ANEXO I

F_AA_236

CERTIFICADO DE ORIGINALIDAD TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 18 de julio de 2024

De mi consideración:

Yo, IVONNE FERNANDA MALDONADO SOLIZ, en calidad de Director del Trabajo de Integración Curricular titulado DESARROLLO DE UN FRONTEND asociado al SISTEMA DE GESTIÓN PARA REPARACIONES ELECTRÓNICAS PARA EL TALLER "ELECTRÓNICA ZURITA" elaborado por el estudiante ALAN STEVEN PÉREZ ALMEIDA de la carrera en TECNOLOGÍA SUPERIOR 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 11%.

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,



Ivonne Maldonado
Docente Ocasional a Tiempo Completo
ESFOT

ANEXO II

Recopilación de Requerimientos

En la **Tabla 1** se presenta la recopilación de requerimientos obtenidos del Product Owner, con los cuales se desarrolla el componente frontend.

Tabla 1 Recopilación de requerimientos

REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
Frontend	RR001	Como usuario técnico necesito consumir endpoints para poder: <ul style="list-style-type: none"> • Iniciar y cerrar sesión. • Recuperar contraseña.
	RR002	Como usuario técnico necesito consumir <i>endpoints</i> para: <ul style="list-style-type: none"> • Gestionar datos del perfil de técnico. • Cambiar de contraseña.
	RR003	Como usuario técnico necesito consumir endpoints para: <ul style="list-style-type: none"> • Gestionar clientes (agregar, editar, eliminar, mostrar). • En el registro del cliente debe contar con un apartado para la dirección de domicilio utilizando Google Maps. • En el registro del cliente debe con un apartado para seleccionar si el cliente es frecuente o no al establecimiento.
	RR004	Como usuario técnico necesito consumir endpoints para: <ul style="list-style-type: none"> • Visualizar los clientes registrados en diferentes tablas que estén con datos filtrados por clientes frecuentes y no frecuentes
	RR005	Como usuario técnico contar con una barra de búsqueda que permita: <ul style="list-style-type: none"> • Visualizar los clientes filtrados por número de cedula ya sean frecuentes o no frecuentes.
	RR006	Como usuario técnico necesito contar con una barra de navegación lateral que me permita: <ul style="list-style-type: none"> • Cambiar entre pantallas para realizar las diferentes acciones. • Dar uso a la barra lateral sin la necesidad de navegar lateralmente. • Ser escalable por si es necesario agregar más funciones en el futuro.
	RR007	Como usuario técnico necesito consumir endpoints para: <ul style="list-style-type: none"> • Gestionar órdenes (agregar, visualizar, finalizar). • En el registro de la orden debe contar con un apartado para seleccionar que tipo de servicio es la orden (mantenimiento, reparación, revisión).
	RR008	Como usuario técnico necesito consumir endpoints para: <ul style="list-style-type: none"> • Visualizar las ordenes registradas separadas por tablas filtradas por tipo de servicio. • Contar con un botón que permita finalizar la orden.

	<ul style="list-style-type: none"> • Visualizar las ordenes registradas con el estado finalizado en una tabla separada independientemente del servicio.
RR009	<p>Como usuario técnico necesito consumir <i>endpoints</i> para:</p> <ul style="list-style-type: none"> • Gestionar proformas (agregar y finalizar) únicamente de ordenes con el servicio “Reparación”. • En el registro de las proformas debe contar con un apartado que permita ingresar cuantas piezas sean necesarias.
RR010	<p>Como usuario técnico necesito consumir endpoints para:</p> <ul style="list-style-type: none"> • Visualizar la proforma generada y su estado mediante un botón en la tabla de reparaciones.
RR011	<p>Como usuario técnico inicial necesito consumir endpoints para:</p> <ul style="list-style-type: none"> • Gestionar nuevos técnicos (agregar y eliminar)
RR012	<p>Como usuario técnico inicial necesito consumir endpoints para:</p> <ul style="list-style-type: none"> • Visualizar los técnicos registrados a través de una tabla

Historias de Usuario

Finalizada la recopilación de requerimientos, se pasa a desarrollar cada una de las historias de usuario para el componente, para ello se presenta las 9 historias de usuario que se han creado a partir de los requerimientos del Product Owner, estas son presentadas desde la **Tabla 2** hasta la **Tabla 12**

Tabla 2 HU001 Iniciar y cerrar sesión

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Técnico
Nombre de historia: Consumir varios <i>endpoints</i> para iniciar y cerrar sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: Alan Pérez	
<p>Descripción: El usuario técnico necesita consumir los <i>endpoints</i> necesarios para iniciar y cerrar sesión. La página principal cuenta con el formulario para poder iniciar sesión con lo que nadie más que los técnicos podrán iniciar sesión. Si el inicio de sesión es exitoso, dentro del componente web contará con un botón “Salir” que le permite cerrar sesión.</p> <p>El usuario que inicie sesión debe contar con un contador de tiempo de espera para que se cierre sesión si el usuario no utiliza la página web en 10 minutos con posibilidad de cambio de tiempo.</p>	

Observación: Usuarios técnicos son los únicos capaces de iniciar sesión y consumir los endpoints del backend.
--

Tabla 3 HU002 Recuperación de contraseña

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: Técnico
Nombre de historia: Consumir el <i>endpoint</i> para recuperar contraseña	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir el <i>endpoints</i> para poder recuperar su contraseña si el usuario técnico la ha olvidado, para ello se emplea un formulario con el único campo: <ul style="list-style-type: none"> • Correo electrónico 	
Observación: Usuarios técnicos deben contar con una cuenta confirmada para poder hacer uso de la recuperación de contraseña. El correo electrónico tiene una validación para que el campo tenga un formato de correo electrónico (ejemplo @gmail.com).	

Tabla 4 HU004 Visualización de clientes

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para visualización de clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir el <i>endpoint</i> para la visualización de los clientes registrados, para ello se necesita que haya 2 tablas que separen los clientes por un filtro "Frecuencia".	
Observación: La tabla con los datos de los clientes debe contar con opciones para poder actualizar los datos del cliente, poder visualizar los datos y poder eliminar al cliente. Debe existir una barra de búsqueda que permita filtrar los clientes de las tablas número de cedula.	

Tabla 5 HU005 Barra de navegación lateral

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Técnico
Nombre de historia: Barra lateral de navegación	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Iteración asignada: 2	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita contar con una barra lateral que contenga las acciones necesarias que ayuden a navegar entre páginas, esta debe ser estática y debe poder ocultarse para poder tener mejor visión de las diferentes pantallas.	
Observación: La barra de navegación es estática con lo cual el usuario técnico no tiene el problema de tener que navegar verticalmente para poder cambiar de pantalla, la barra lateral cuenta con un botón hamburguesa que le permite activar o desactivar, así también si el usuario técnico da clic en cualquier parte que no sea la barra lateral esta se cierra automáticamente.	

Tabla 6 HU006 Gestión de perfil técnico

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la gestión de perfil técnico	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir los <i>endpoints</i> para realizar la gestión de los datos de su perfil, por lo que cuenta con los formularios con los siguientes campos: <ul style="list-style-type: none"> • Nombre • Apellido • Número de teléfono • Correo electrónico El segundo correo electrónico cuenta con los siguientes campos <ul style="list-style-type: none"> • Contraseña actual 	

- Nueva contraseña

Observación: Los formularios para el cambio de datos del técnico cuentan con verificaciones, el nombre y apellido deben tener mínimo 3 caracteres y máximo 20 caracteres, el número de teléfono debe contar con mínimo 5 dígitos y máximo 10, el correo electrónico debe tener la estructura correcta (ejemplo @gmail.com), mientras que el formulario de actualización de contraseña debe contar con una validación que no permita enviar el formulario si la contraseña actual y la nueva son las mismas, así mismo la contraseña debe tener mínimo 6 caracteres.

Tabla 7 HU007 Gestión de órdenes

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la gestión de órdenes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Responsable: Alan Pérez	
<p>Descripción: El usuario técnico necesita consumir los <i>endpoints</i> para realizar la gestión de las órdenes, para esto se cuenta con 2 formularios que tienen los siguientes campos:</p> <ul style="list-style-type: none"> • Cédula del cliente <p>El segundo formulario de la orden cuenta con los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre del equipo electrónico • Modelo del equipo electrónico • Marca del equipo electrónico • Número de serie del equipo electrónico • Fecha de ingreso del equipo electrónico • La razón de ingreso del equipo electrónico • El servicio para el equipo electrónico 	
<p>Observación: Los formularios para las órdenes son 2 ya que el primero valida que el cliente exista y si existe este se liga a la orden, el segundo formulario cuenta con verificaciones para que no puedan estar vacíos y estos deben tener mínimo 3 caracteres con un máximo de 30, en la razón debe haber mínimo 10 caracteres, la fecha de ingreso valida que no sea un día anterior al actual y el servicio es por defecto mantenimiento, pero el técnico puede cambiar si es necesario.</p>	

Tabla 8 HU008 Visualización de ordenes

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la visualización de ordenes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir el <i>endpoint</i> para visualizar las ordenes, estas deben estar separadas por tablas teniendo como filtro el tipo de servicio del equipo electrónico, así mismo debe haber una tabla que se muestre cuando una orden tiene el estado finalizado independientemente del servicio de la orden.	
Observación: Las tablas se separan por los filtros del servicio y estado, cada orden cuenta con opciones que puede utilizar el técnico, estas son, visualizar la orden, actualizar el estado de la orden a “En proceso” y cambiar el estado de la orden a “Finalizado”. Debe existir una barra de búsqueda para poder filtrar todas las órdenes de un cliente por medio de su número de cedula.	

Tabla 9 HU009 Gestión de proformas

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la gestión de proformas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir los <i>endpoints</i> para la gestión de proformas, las ordenes con servicio “Reparación” son las únicas que podrán tener una proforma, la generación de la proforma cuenta con 2 partes, en la primera va a mostrar los datos de la orden a la cual se va a generar la proforma, la segunda parte cuenta con un formulario con los siguientes campos: <ul style="list-style-type: none"> • Nombre de la pieza • Precio de la pieza 	
Observación: El formulario de la proforma debe contar con 2 acciones que permitan al técnico, agregar una nueva pieza sin límite alguno y eliminar una pieza o eliminar todas	

las piezas, estas piezas tienen una validación, esta no puede estar vacía y tampoco puede tener un precio igual o menor a 0.

Tabla 10 HU010 Visualización de proformas

HISTORIA DE USUARIO	
Identificador (ID): HU010	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la visualización de proformas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir el <i>endpoint</i> para la visualización de las proformas, en la cual debe mostrar los datos de las piezas más el total estimado calculado con las piezas de la proforma.	
Observación: Para poder visualizar la proforma se ha cambiado la visualización de orden en las ordenes de servicio "Reparación" a visualizar proforma, por lo cual se comprueba si hay una proforma.	

Tabla 11 HU011 Gestión de nuevos técnicos

HISTORIA DE USUARIO	
Identificador (ID): HU0011	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la gestión de nuevos técnicos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir los <i>endpoints</i> para gestión de nuevos técnicos, para ello se ha utilizado un formulario con los siguientes campos: <ul style="list-style-type: none"> • Nombre del técnico • Apellido del técnico • Número de RUC • Número de teléfono o número de celular • Correo electrónico • Contraseña 	
Observación: Inicialmente se contará con un técnico quemado en el backend, por lo	

que el backend verifica si el técnico quemado es el que intenta realizar la gestión de los nuevos técnicos, el formulario cuenta con verificación de datos como nombre y apellido deben tener mínimo 3 caracteres y máximo 20, el número de RUC debe tener mínimo 10 dígitos y máximo 13 dígitos, el teléfono debe tener mínimo 6 dígitos y máximo 10 dígitos, el correo debe contar con una estructura de correo (ejemplo @gmail.com) y la contraseña debe tener mínimo 6 caracteres.

Tabla 12 HU012 Visualización de técnicos

HISTORIA DE USUARIO	
Identificador (ID): HU0012	Usuario: Técnico
Nombre de historia: Consumir <i>endpoints</i> para la visualización de técnicos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Responsable: Alan Pérez	
Descripción: El usuario técnico necesita consumir los <i>endpoints</i> para visualizar los técnicos que se encuentran registrados. Así mismo debe consumir el endpoint que permita eliminar a un técnico.	
Observación: La tabla debe mostrar todos los técnicos que se encuentren registrados, ya sean los quemados en el backend o los registrados por el técnico. Debe existir una barra de búsqueda que permita filtrar la tabla para buscar a un técnico en específico por el nombre.	

Product Backlog

La **Tabla 13** muestra la prioridad que se ha dado a cada requisito para el componente ya que esta tabla ayuda con la guía en la gestión y tener un seguimiento de los requisitos del componente, estos se han clasificados por la complejidad del desarrollo y el tiempo que se ha tomado.

Tabla 13 Product Backlog

ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Inicio de sesión y cierre de sesión	1	Finalizado	Alta
HU002	Recuperación de contraseña	1	Finalizado	Media
HU003	Gestión de clientes	2	Finalizado	Alta
HU004	Visualización de Clientes	1	Finalizado	Alta

HU005	Barra de navegación lateral	2	Finalizado	Baja
HU006	Gestión de perfil técnico	1	Finalizado	Baja
HU007	Gestión de órdenes	2	Finalizado	Alta
HU008	Visualización de órdenes	1	Finalizado	Alta
HU009	Gestión de proformas	3	Finalizado	Alta
HU010	Visualización de proformas	1	Finalizado	Alta
HU011	Gestión de nuevos técnicos	1	Finalizado	Media
HU012	Visualización de técnicos	1	Finalizado	Media

Sprint Backlog

Las actividades programadas en los sprints para el proyecto se detallan en la siguiente **Tabla 14**, las actividades se han dividido en 5 partes, en estas se detalla la información de cada actividad realizada y el tiempo necesario que se ha tomado para cumplir con los entregables definidos con anterioridad con el Product Owner.

Tabla 14 Sprint Backlog

SPRINT BACKLOG						
ID-SB	NOMBRE	MODULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Establecimiento de la configuración del entorno de desarrollo	-----	-----	-----	<ul style="list-style-type: none"> • Determinar los requerimientos para el <i>frontend</i> del sistema web. • Configurar todas las herramientas que se utiliza en el desarrollo del <i>frontend</i>. • Configurar el entorno de desarrollo del <i>frontend</i>. 	10H
SB001	Consumir <i>endpoints</i> para el usuario técnico	Módulo - Autenticación	HU001	Iniciar sesión y cerrar sesión	<ul style="list-style-type: none"> • Consumir el <i>endpoint</i> para realizar el inicio de sesión y permitir el cierre de sesión. • Crear los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos 	65H
		Módulo - Recuperación de contraseña	HU002	Recuperar Contraseña	<ul style="list-style-type: none"> • Consumir el <i>endpoint</i> que permite la recuperación de la contraseña enviando un correo electrónico. • Crear los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos 	
		Módulo - Gestión de clientes	HU003	Gestión de clientes	<ul style="list-style-type: none"> • Consumir los <i>endpoints</i> para crear y actualizar los datos del cliente. • Crear los prototipos de la interfaz. 	

				<ul style="list-style-type: none"> • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos. 	
		Módulo – Visualizar clientes	HU004	Visualización de clientes <ul style="list-style-type: none"> • Consumir los <i>endpoint</i> para gestionar los clientes mostrándolos en una tabla y permitiendo eliminarlos. • Crear de los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Creación de barra de búsqueda para filtrar clientes por número de cedula. • Realizar las pruebas para la comprobación del funcionamiento. 	
		Módulo – Barra de navegación lateral	HU005	Barra de navegación lateral <ul style="list-style-type: none"> • Crear los prototipos de la barra de navegación. • Codificar los prototipos de la interfaz. • Añadir las diferentes opciones que necesita el técnico. • Verificación de funcionalidad y pruebas correspondientes en cada pantalla. 	
		Módulo – Gestión de perfil técnico	HU006	Gestión de perfil técnico <ul style="list-style-type: none"> • Consumir los <i>endpoints</i> para actualizar los datos y contraseña del técnico. • Crear los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Validación de los campos requeridos • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos. 	
SB002	Consumir <i>endpoints</i> para el usuario técnico	Modulo – Gestión de órdenes	HU007	Gestión de órdenes <ul style="list-style-type: none"> • Consumir los <i>endpoints</i> para crear órdenes de trabajo para un cliente. • Crear los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos. 	45H

		Modulo - Visualizar lista de trabajos	HU008	Visualizar órdenes	<ul style="list-style-type: none"> • Consumir los <i>endpoints</i> para visualizar las órdenes de trabajo mostrando los datos en diferentes tablas que también permiten hacer una actualización de datos. • Crear de los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Creación de barra de búsqueda para filtrar órdenes por número de cedula de cliente. • Realizar las pruebas para la comprobación del funcionamiento. 	
SB003	Consumir <i>endpoints</i> para el usuario técnico	Modulo – Gestión de proformas	HU09	Gestión de proformas	<ul style="list-style-type: none"> • Consumir los <i>endpoints</i> para crear proformas en base a una orden de trabajo que tenga tipo de servicio “Reparación”. • Crear los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos. 	45H
		Modulo - Visualizar proformas	HU010	Visualizar proformas	<ul style="list-style-type: none"> • Consumir los <i>endpoints</i> para visualizar las proformas generadas mediante la misma tabla de órdenes cambiando la opción ver orden a ver proforma. • Crear de los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento. 	
		Modulo – Gestión de nuevos técnicos	HU011	Gestión de nuevos técnicos	<ul style="list-style-type: none"> • Consumir el <i>endpoint</i> para crear agregar un nuevo técnico. • Crear los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Realizar las pruebas para la comprobación del funcionamiento y validación de los campos. 	

		Modulo – Visualización de técnicos	HU012	Visualización de técnicos	<ul style="list-style-type: none"> • Consumir los <i>endpoint</i> para visualizar los técnicos mostrándolos en una tabla y permitiendo eliminarlos. • Crear de los prototipos de la interfaz. • Codificar los prototipos de la interfaz. • Creación de barra de búsqueda para buscar un técnico por nombre. • Realizar las pruebas para la comprobación del funcionamiento. 	
SB004	Pruebas en el <i>frontend</i>	<ul style="list-style-type: none"> • Pruebas de compatibilidad • Pruebas de rendimiento • Pruebas de responsividad 				15H
SB005	Despliegue del <i>frontend</i>	<ul style="list-style-type: none"> • Despliegue del <i>frontend</i> 				10H
	Documentación	<ul style="list-style-type: none"> • Documentación del Trabajo de integración Curricular • Anexos 				50H
TOTAL						240H

Diseño de Interfaces

Las interfaces que se han diseñado para los diferentes módulos del componente front end se presentan desde la **Figura 1** hasta la **Figura 19**, estas muestran todas las pantallas utilizadas por el usuario técnico.



Figura 1 Pantalla: Inicio de sesión

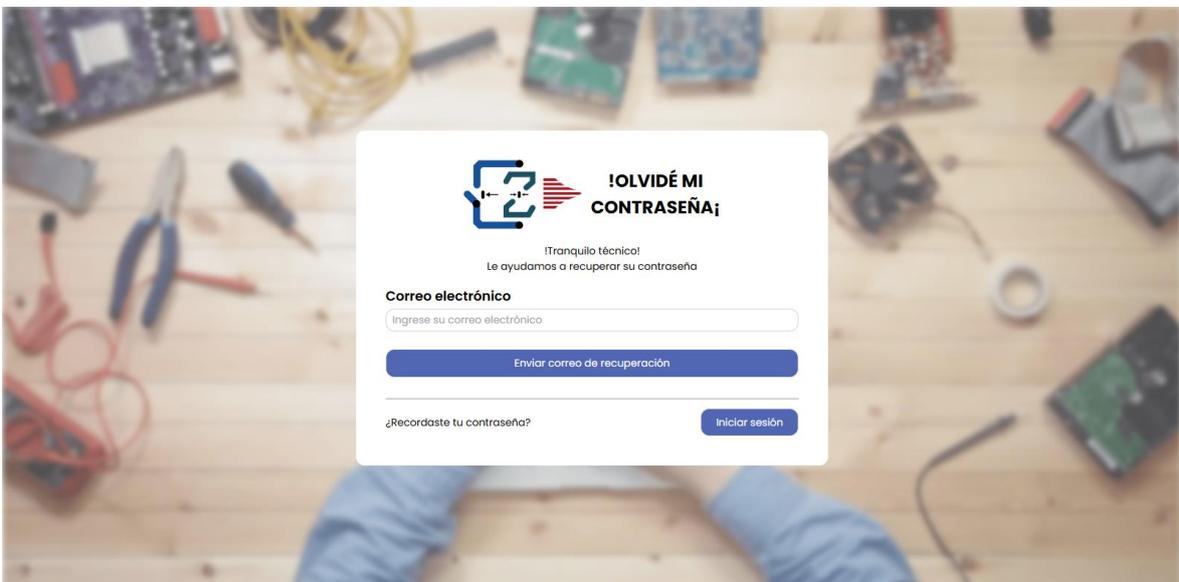


Figura 2 Pantalla: Recuperación de contraseña

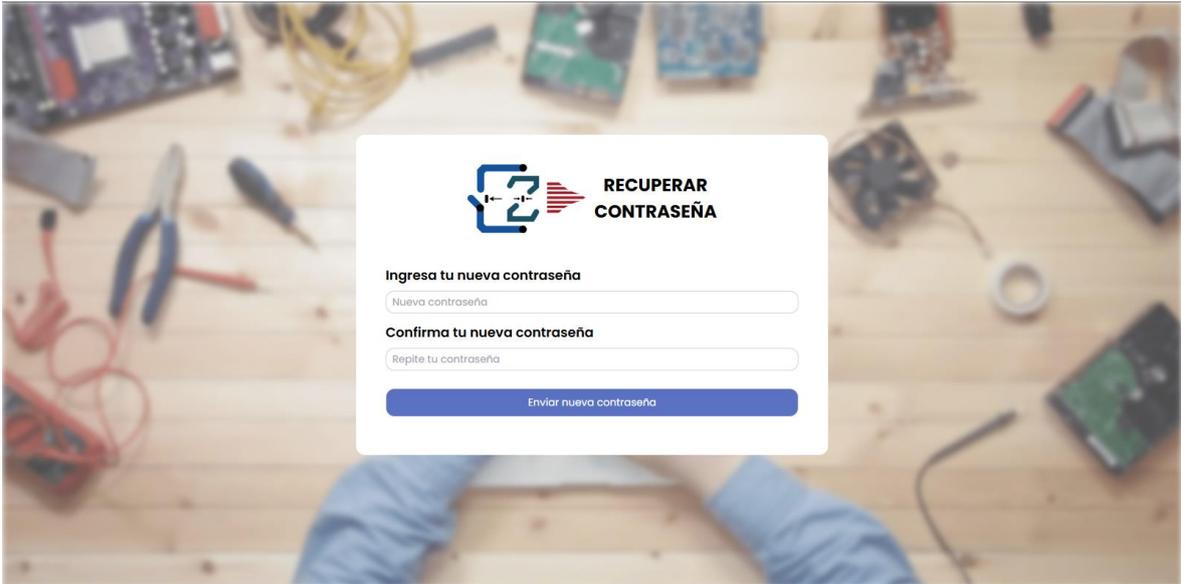


Figura 3 Pantalla: Ingreso de nueva contraseña

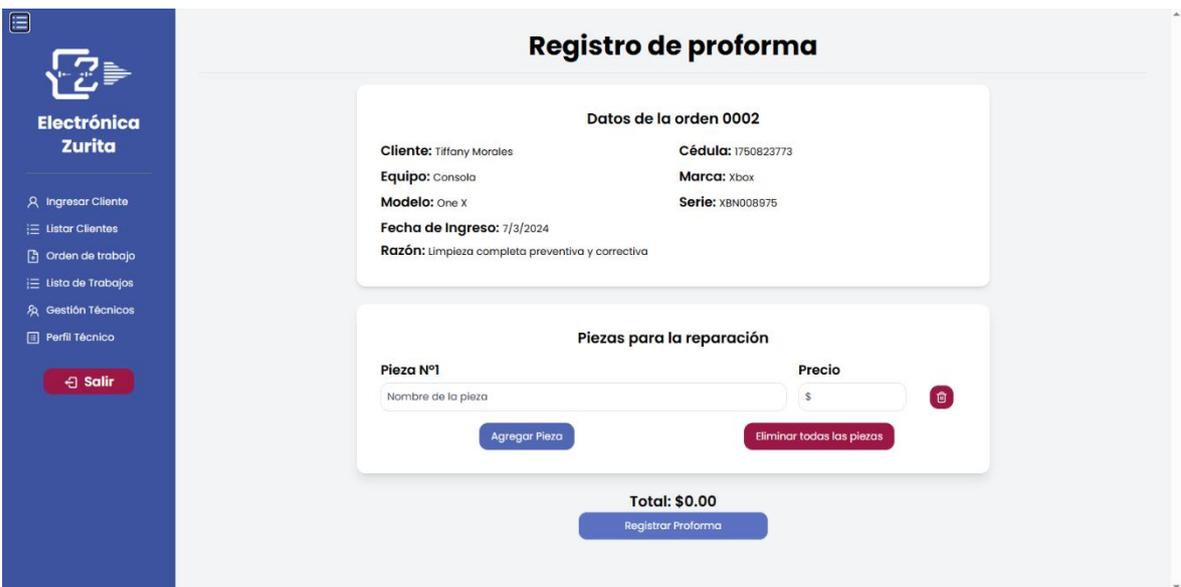


Figura 4 Pantalla: Registro de proforma

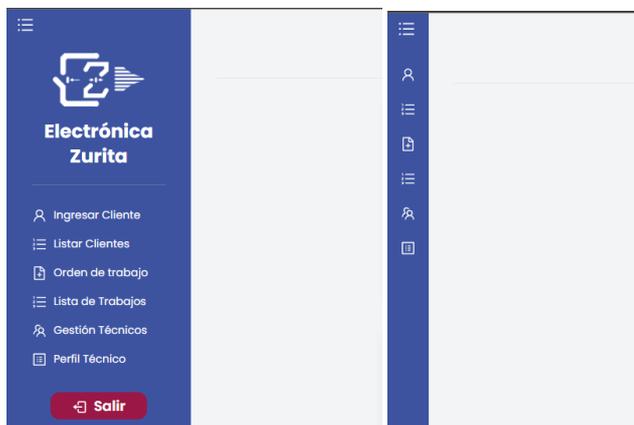


Figura 5 Barra de navegación activa y desactivada

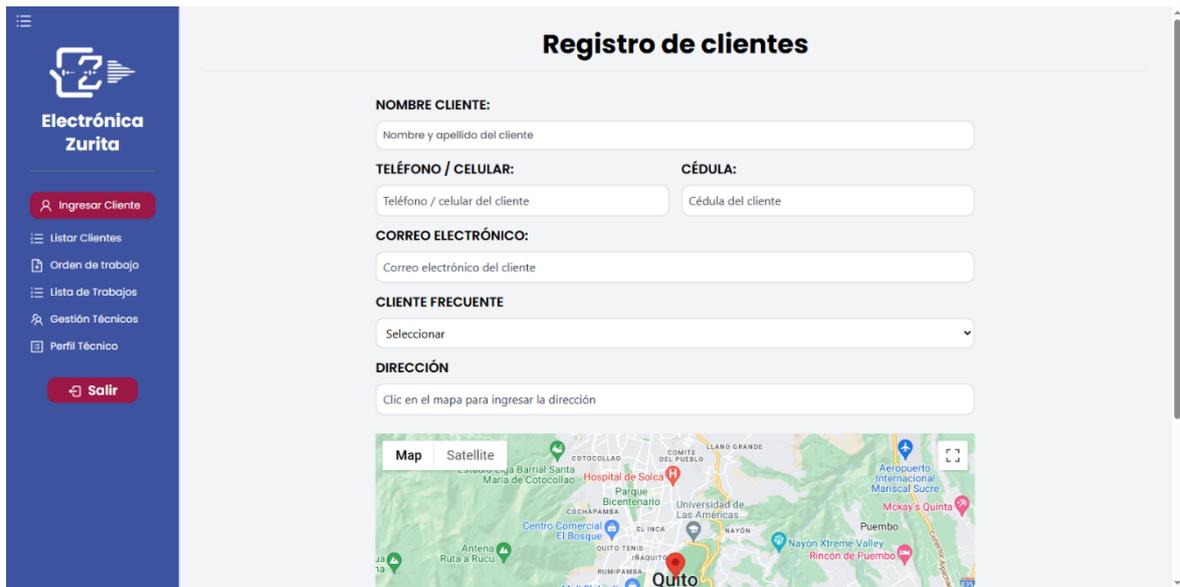


Figura 6 Pantalla: Registro de Clientes parte superior

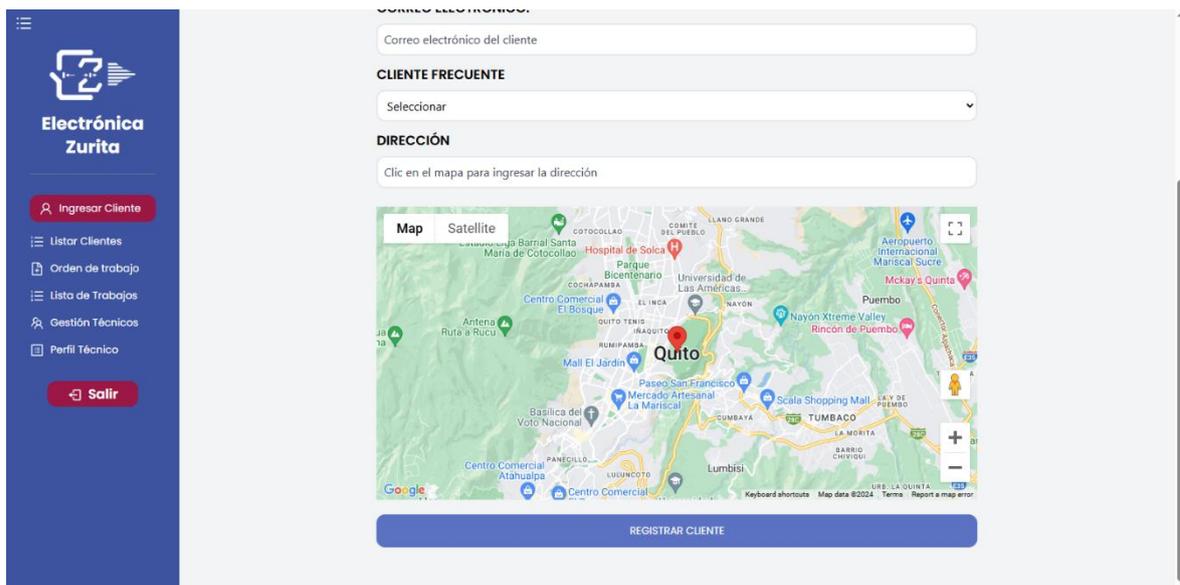


Figura 7 Pantalla: Registro de clientes parte inferior e implementación de Google Maps

Listado de clientes

Buscar por cédula Actualizar Visualizar Eliminar

Cientes Frecuentes

N°	Nombre	Email	Celular	Cédula	Dirección	Acciones
1	Andres Tufiño	andrestma@gmail.com	960444123	1804445045	Luis callisto	
2	Richard Soria	rickimauri04@hotmail.com	2165456412	9876541233	Mitad del mundo	
3	Michelle Suarez	almeandrapark@gmail.com	960444124	1704445116	Juan Correa N81- 62 y, Quito 170302	
4	Ariel Sanchez	riveraariel433@gmail.com	995122669	1213123456	Mariano Calvache 647, Quito 170135	
5	Juanito alcachofa	jguerralovato@gmail.com	964030442	12131234234	32VG+4J Piranhas, Eldorado do Carajás - PA, Brazil	
6	NECOMAN	davidperezalvernia2019@gmail.com	986508496	12131234235	JF7G+7Q Taiiki, Hokkaido, Japan	
7	Evelyn Guachamin	taty-1997@outlook.com	960444123	1751253400	RGMF+7G6, Av. 6 de Diciembre, Quito 170513	
8	Ariel Ashqui	rodarielk@gmail.com	960444123	1751086628	QGXB+QQ3, Av. La Coruña, Quito 170517	
9	Paul Cabrera	paul.cabrera@epn.edu.ec	984622128	1751431964	Q3P2+VW Provincia de Puntarenas, Coto Brus, Costa Rica	
10	Anthony Astudillo	anthony.astudillo@epn.edu.ec	2165456412	1726195207	XR5G+3X Santo Domingo	
11	Mireya Garcia	miregarcia444@gmail.com	984951134	1726286188	5R425RG3+6C	
12	Tiffany Morales	fannymn35@gmail.com	987654321	1750823773	De los Tulipanes E11-140, Quito 170138	

Figura 8 Pantalla: Lista de clientes

Orden de trabajo

NÚMERO DE CÉDULA CLIENTE: Número de cédula del cliente

Cliente: **Télefono:**

Correo electrónico:

Equipo: Tipo de equipo

Modelo: Modelo del equipo

MARCA: Marca del equipo

NÚMERO DE SERIE: Número de serie del equipo

COLOR DEL EQUIPO: Color del equipo

FECHA DE INGRESO DEL EQUIPO: mm/dd/yyyy

Figura 9 Pantalla: Registro orden de trabajo parte superior

Modelo:
Modelo del equipo

MARCA:
Marca del equipo

NÚMERO DE SERIE:
Número de serie del equipo

COLOR DEL EQUIPO:
Color del equipo

FECHA DE INGRESO DEL EQUIPO:
mm/aa/yyyy

RAZÓN DE INGRESO:
Razones de ingreso del equipo

SERVICIO:
Mantenimiento

REGISTRAR ORDEN

Figura 10 Pantalla: Registro orden de trabajo parte inferior

Listado de Trabajos

Buscar por cédula Actualizar Estado Crear Proforma Visualizar Finalizar

Mantenimiento

Nº Orden	Cliente	Cédula	Equipo	Fecha Ingreso	Estado	Acciones
0001	Tiffany Morales	1750823773	Consola	7/3/2024	Pendiente	

Reparación

Nº Orden	Cliente	Cédula	Equipo	Fecha Ingreso	Estado	Acciones
0002	Tiffany Morales	1750823773	Consola	7/3/2024	En proceso	

Figura 11 Pantalla: Lista de órdenes de trabajo

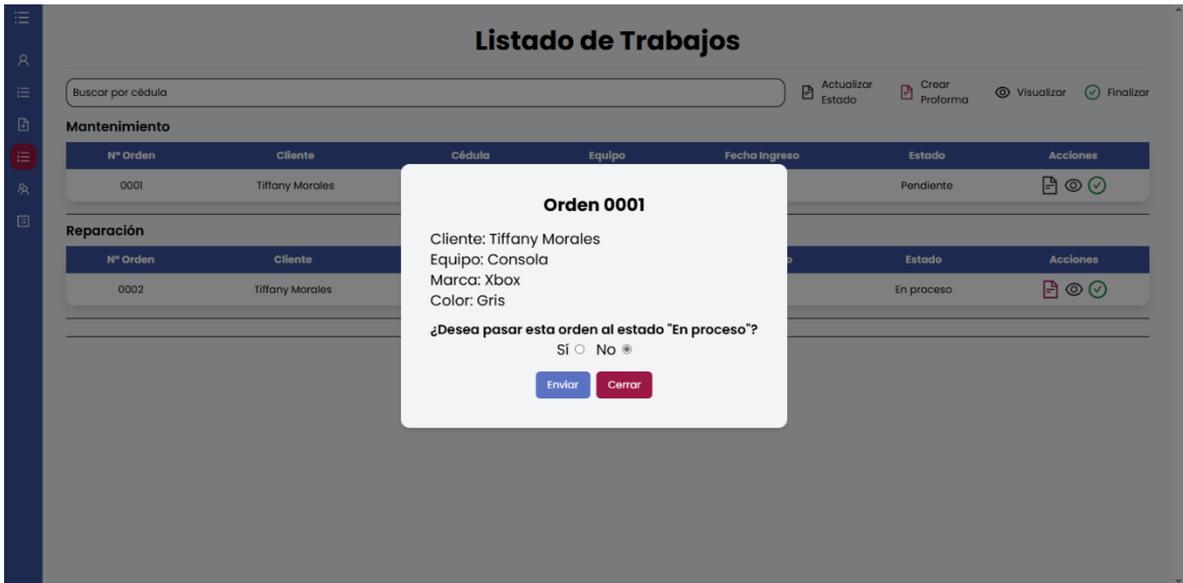


Figura 12 Modal: Actualización de estado de orden a "En proceso"



Figura 13 Modal: Visualización de orden

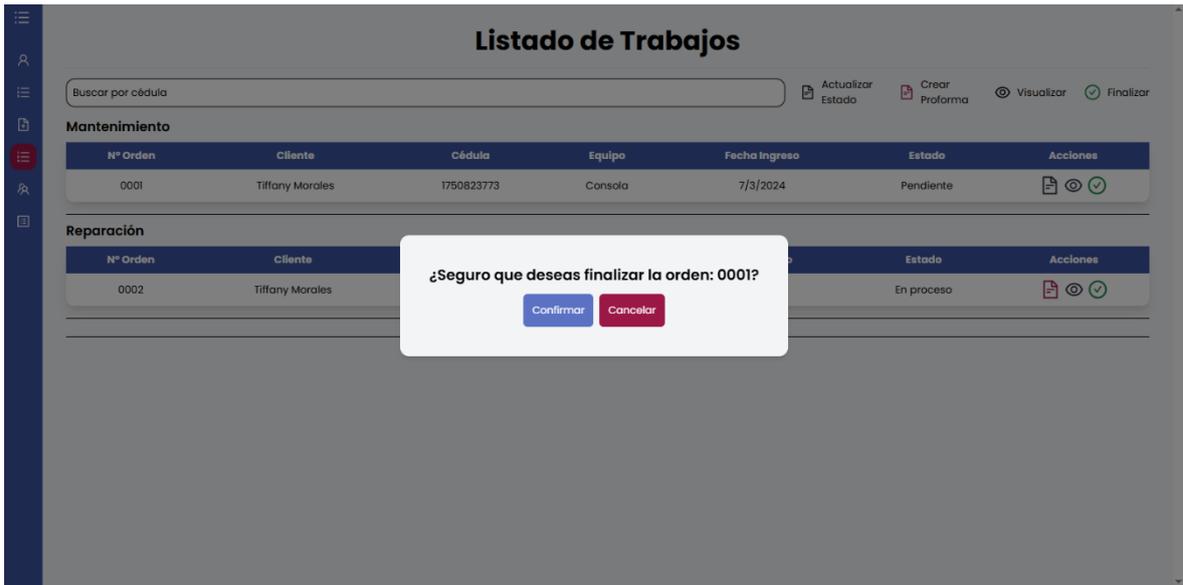


Figura 14 Modal: Actualización de estado de orden a “Finalizado”

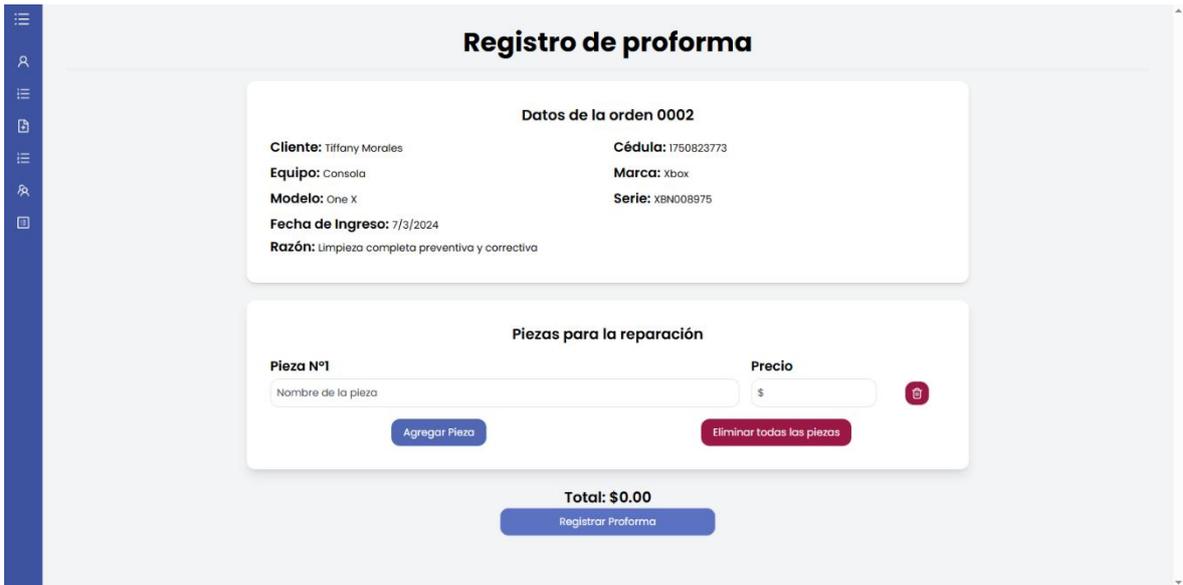


Figura 15 Formulario de proforma

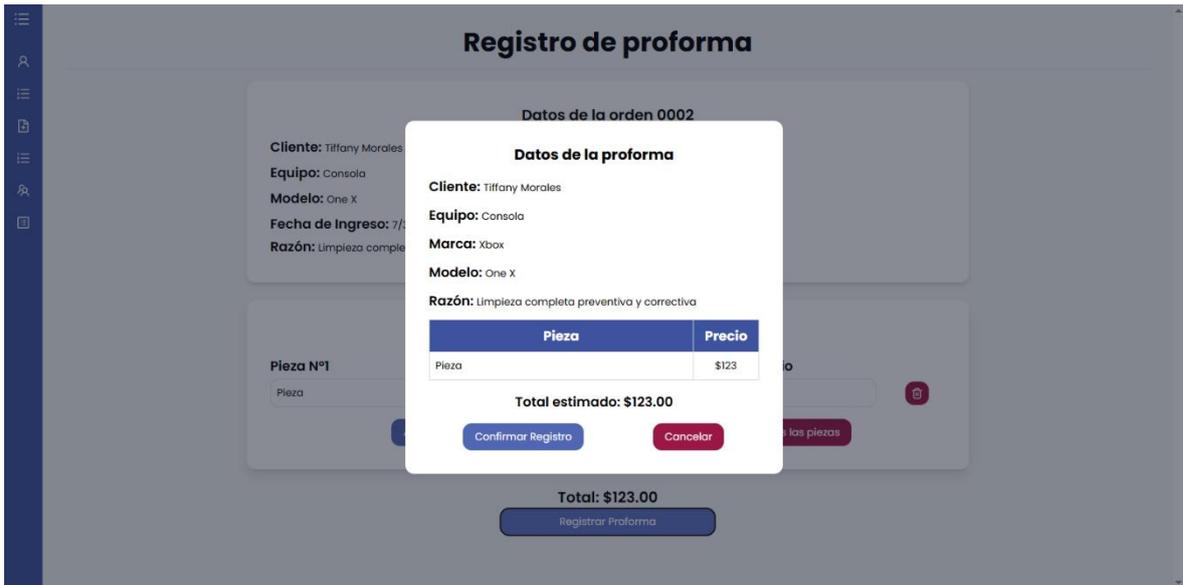


Figura 16 Visualización proforma para registro



Figura 17 Pantalla: Listado de técnicos

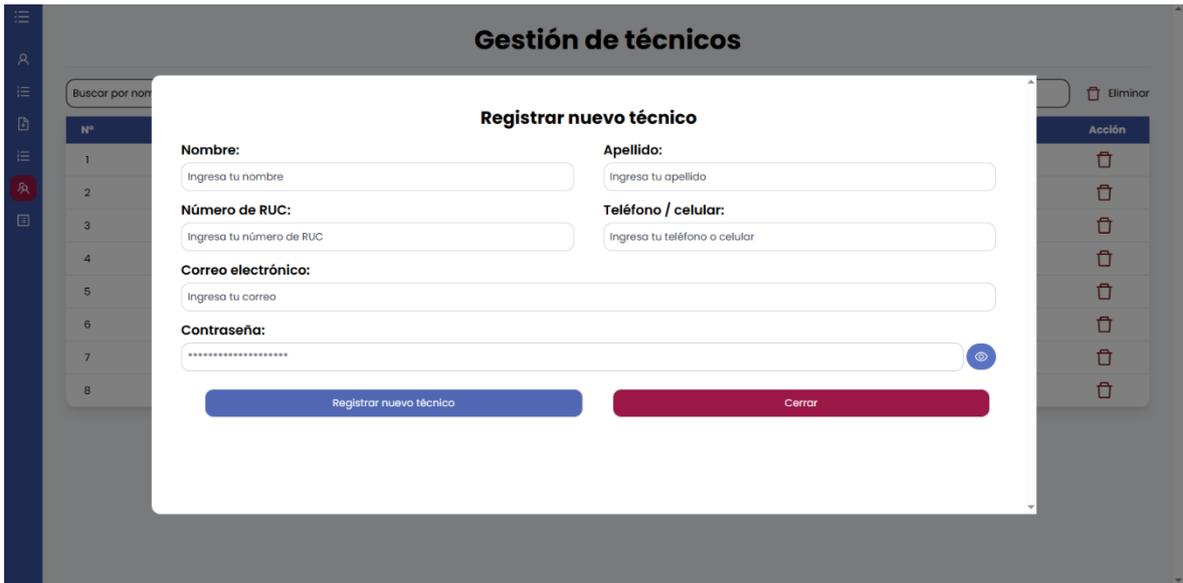


Figura 18 Modal: Registro de nuevo técnico

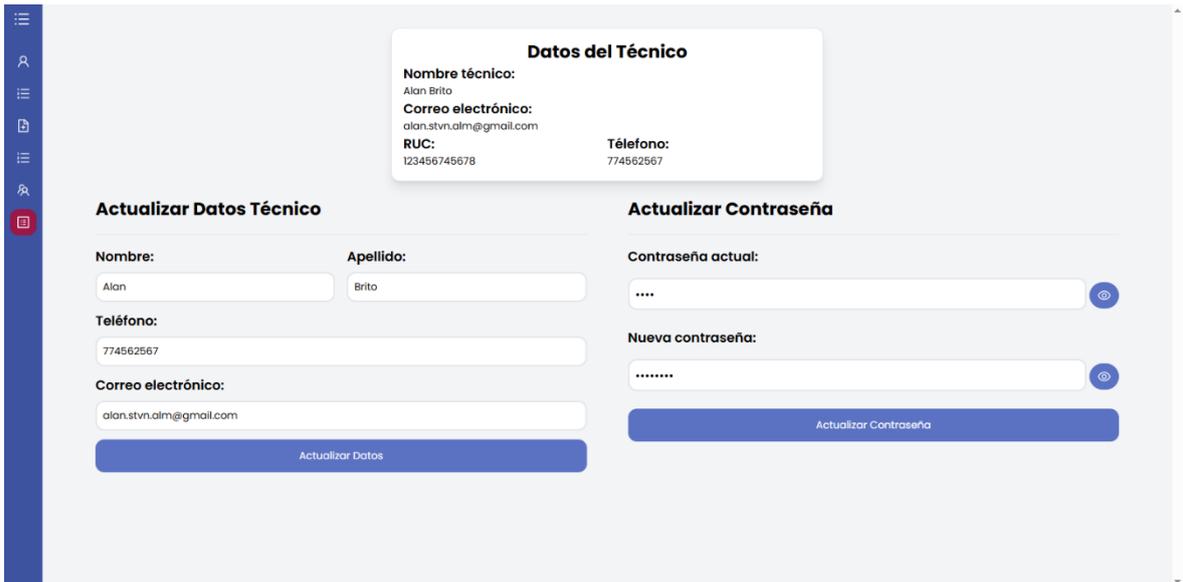


Figura 19 Pantalla: Gestión de datos perfil técnico

Pruebas de Compatibilidad

La **Figura 20** hasta la **Figura 24** muestran algunas de las pruebas de compatibilidad que se han llevado a cabo para comprobar el correcto funcionamiento del componente frontend.

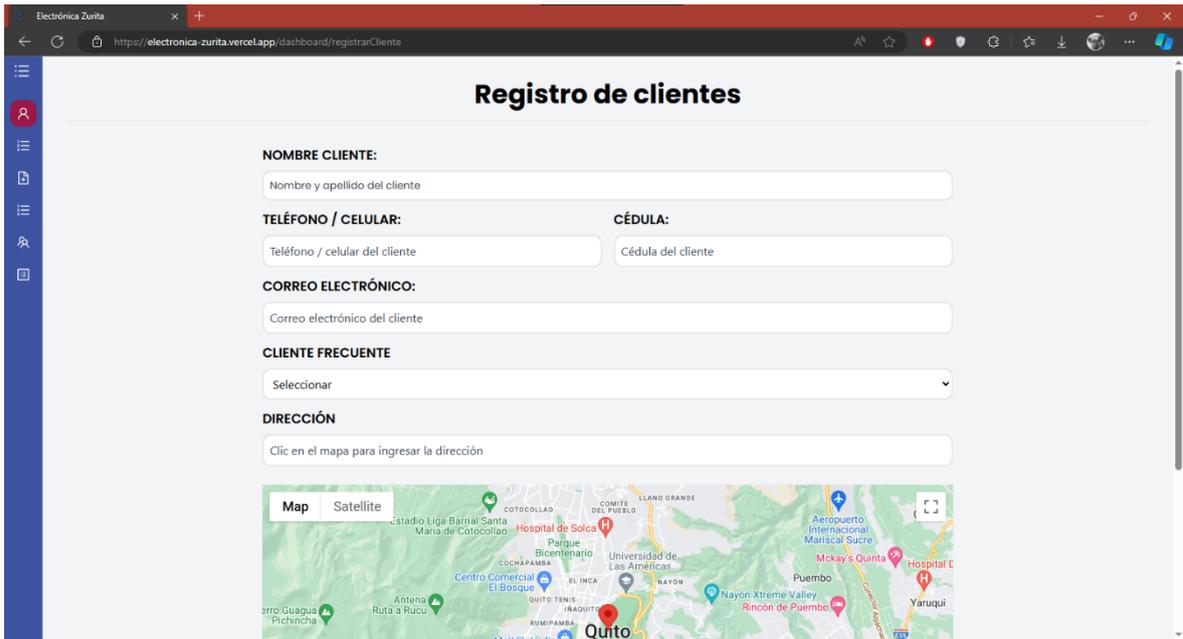


Figura 20 Pantalla Registro de cliente Microsoft Edge Windows 10

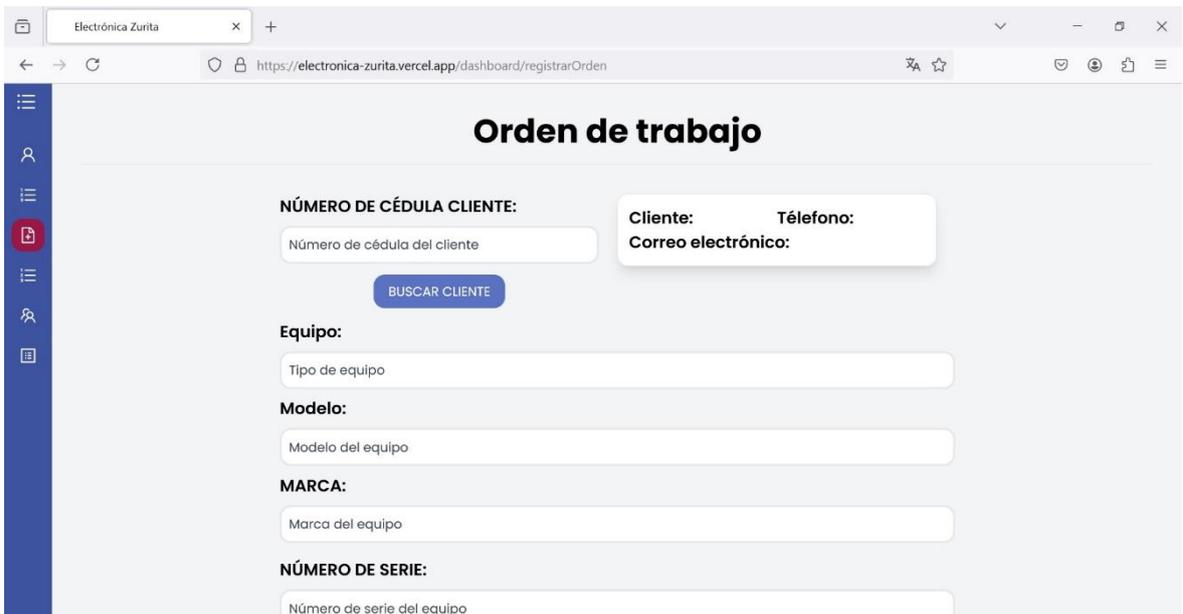


Figura 21 Pantalla Registro de orden de trabajo Mozilla Firefox Windows 11

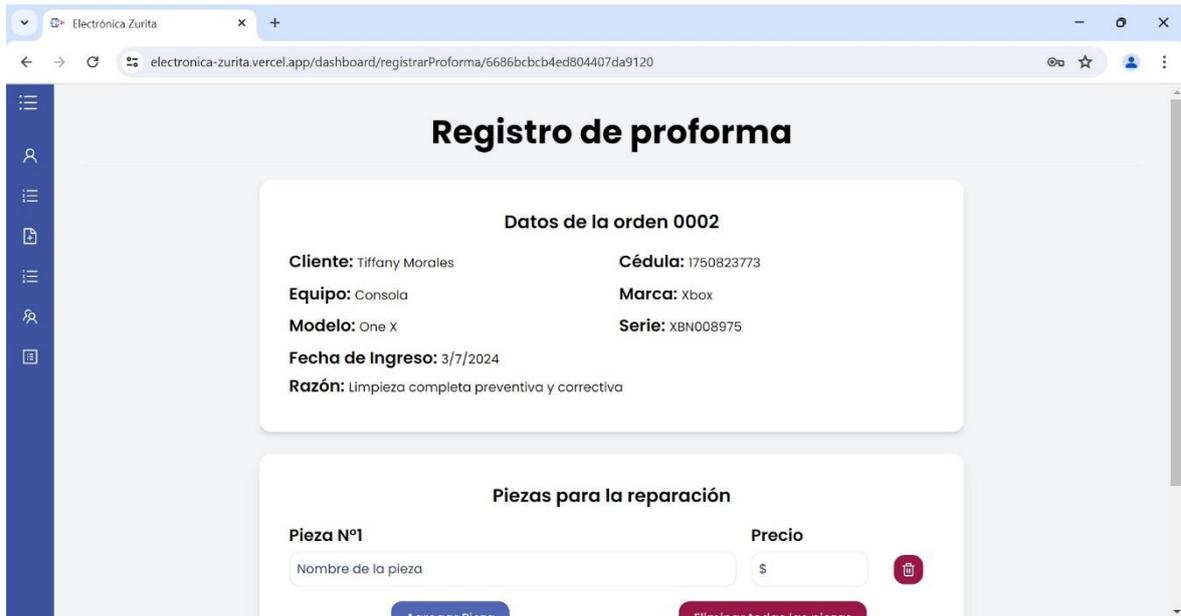


Figura 22 Pantalla Registro de proforma Google Chrome Windows 11

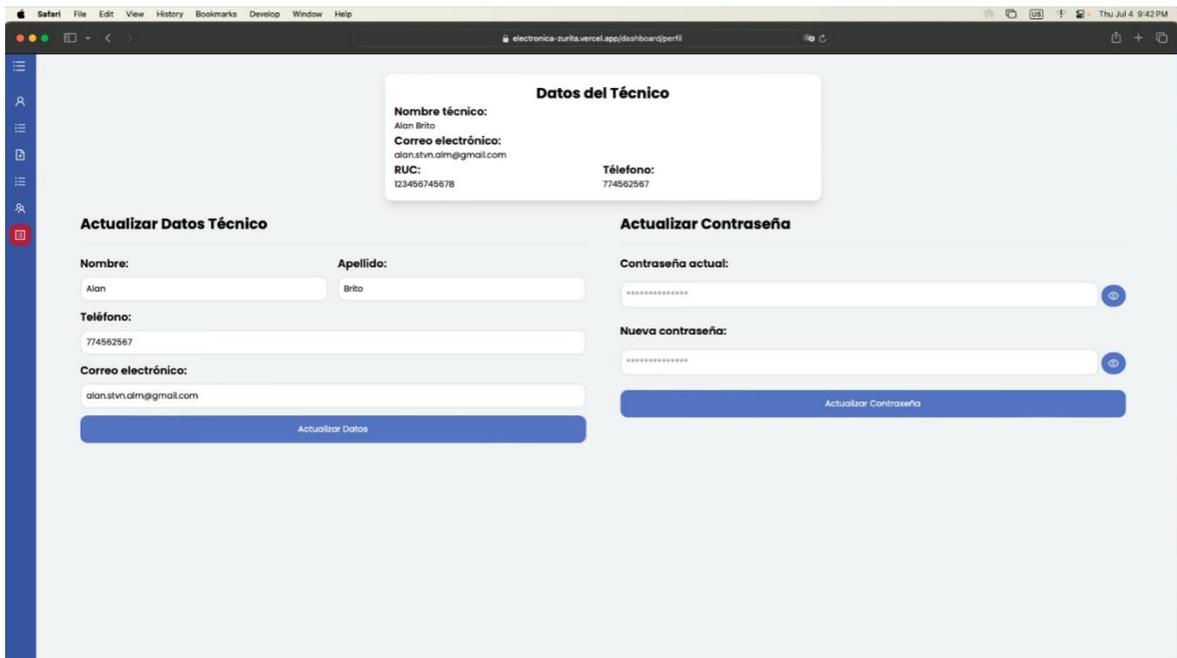


Figura 23 Pantalla perfil de técnico Safari Mac Sonoma

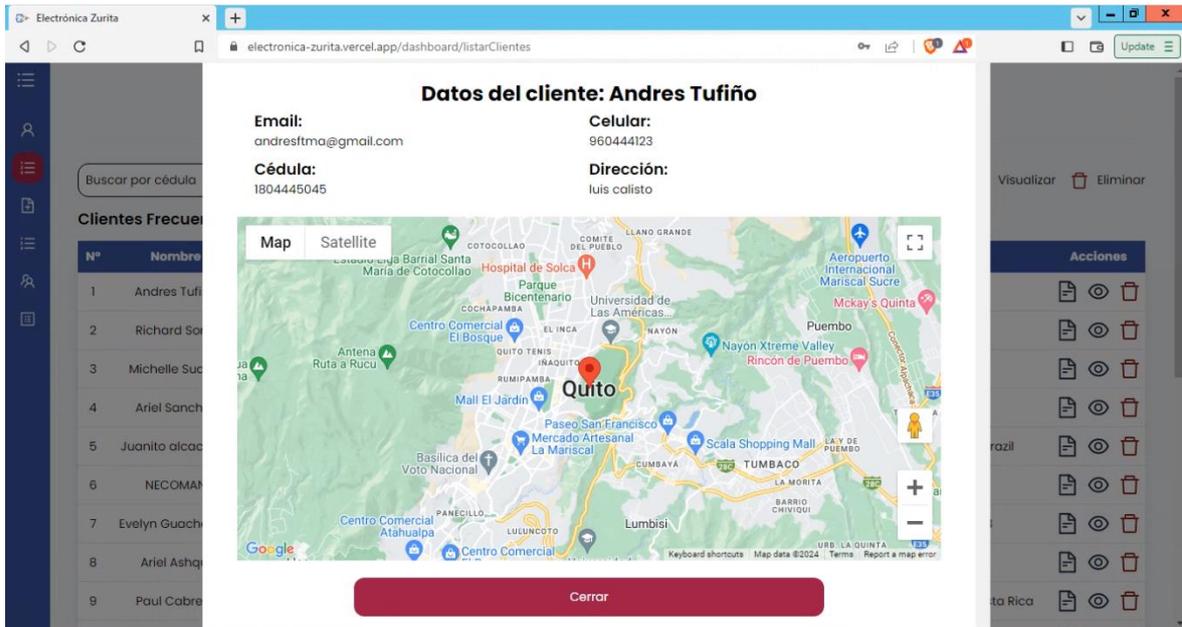


Figura 24 Modal datos del cliente Brave Windows 8.1

ANEXO III

El manual de usuario del componente frontend se puede visualizar al ingresar en el siguiente enlace:

<https://youtu.be/ZxiNBwn6CJU>

ANEXO IV

En este apartado se proporcionan las credenciales para el acceso al componente web, así mismo se proporciona el enlace al repositorio en el cual se alberga todo el código desarrollado para el presente proyecto que cuenta con un README que detalla los pasos para la instalación del componente web.

Credenciales de acceso para usuario técnico al sistema web

- electronicazurita.uio@gmail.com
- Zurita@2024.

Enlace al repositorio del componente frontend

El código del presente proyecto ha sido alojado en un repositorio de GitHub, el cual se puede acceder a través del siguiente enlace:

https://github.com/4lanPz/Electronica_Zurita