

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SISTEMA *WEB* Y APLICACIÓN MÓVIL PARA COMANDAS EN LA CAFETERÍA EPN

DESARROLLO DE UN *FRONTEND*

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

MATEO NICOLAS CUEVA RAMOS

DIRECTOR: ING. YADIRA FRANCO

DMQ, marzo 2023

CERTIFICACIONES

Yo, Mateo Nicolas Cueva Ramos 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.



MATEO NICOLAS CUEVA RAMOS

mateo.cueva@epn.edu.ec

nicolasmcr99@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Mateo Nicolas Cueva Ramos, bajo mi supervisión.



Ing. Yadira Franco R Mg.

DIRECTOR

yadira.franco@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

MATEO NICOLAS CUEVA RAMOS

DEDICATORIA

Con una gran alegría, dedico esta tesis a mis padres Víctor Cueva y Narcisa Ramos, quienes con su ayuda, motivación y esfuerzo me han ayudado a culminar una más de mis metas.

A mi padrino Luis Paucar quien fue una persona muy importante para mí y que ahora ya no está a mi lado, pero mientras estuvo me brindó toda su confianza y amistad y darme motivación, consejos para ser un gran ser humano. A mis hermanos por su apoyo y confidencialidad en momentos felices o tristes.

Finalmente dedico este trabajo a cada uno de mis amigos que conocí gracias a esta nueva etapa de mi vida y también a los que conozco de hace muchos años atrás, por brindarme su amistad y apoyo.

MATEO NICOLAS CUEVA RAMOS

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
ÍNDICE DE CONTENIDO.....	IV
RESUMEN	VI
ABSTRACT	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	4
2 METODOLOGÍA	7
2.1 Metodología de Desarrollo	7
Roles.....	7
Artefactos	9
2.2 Diseño de interfaces	11
Herramienta utilizada para el diseño	12
2.3 Diseño de la arquitectura	12
Patrón arquitectónico	12
2.4 Herramientas de desarrollo.....	13
Librerías.....	14
3 RESULTADOS.....	15
3.1 Sprint 0. Configuración del ambiente de desarrollo.....	15
Definición de los requerimientos para el sistema <i>web</i>	15
Estructura del proyecto sistema <i>web</i>	19
Roles de usuario.....	19
3.2 Sprint 1. Resultado de las interfaces implementadas para el usuario con perfil administrador	20
Consumir varios <i>endpoints</i> para visualizar página informativa de la Cafetería.....	20
Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña.	21
Consumir varios <i>endpoints</i> para generar usuarios cocineros.....	22
Consumir varios <i>endpoints</i> para agregar platos del menú del almuerzo y snacks.	22
Consumir varios <i>endpoints</i> para generar el menú de los almuerzos	23

Consumir varios <i>endpoints</i> para visualizar el pedido y la factura pagada.....	24
Consumir varios <i>endpoints</i> para asignar el pedido al cocinero	25
Consumir varios <i>endpoints</i> para visualizar reporte de pedidos completados.....	26
3.3 Sprint 2. Resultado de las interfaces implementadas para el usuario con perfil cliente.....	26
Consumir varios <i>endpoints</i> para registrarse como cliente.	27
Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña.	27
Consumir varios <i>endpoints</i> para modificar perfil de usuario.....	28
Consumir varios <i>endpoints</i> para gestionar pedido.	29
Consumir varios <i>endpoints</i> para cargar el archivo de confirmación de pago.....	29
Consumir varios <i>endpoints</i> para visualizar el estado del pedido.....	30
3.4 Sprint 3. Resultado de las interfaces implementadas para el usuario con perfil cocinero	30
Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña.	31
Consumir varios <i>endpoints</i> para modificar perfil de usuario.....	31
Consumir varios <i>endpoints</i> para visualizar los pedidos asignados.....	32
Consumir varios en <i>endpoints</i> para cambiar el estado de los pedidos	33
3.5 Sprint 4. Pruebas en el sistema <i>web</i>.....	33
Ejecución de las pruebas unitarias resultados.....	33
Ejecución de pruebas de compatibilidad y resultados	35
Ejecución de pruebas de aceptación y resultados	35
3.6 Sprint 5. Despliegue del sistema <i>web</i>.....	36
Despliegue del sistema <i>web</i> en Vercel	36
4 Conclusiones	38
5 Recomendaciones	39
6 REFERENCIAS BIBLIOGRÁFICAS.....	40
7 ANEXOS	43
ANEXO I.....	44
ANEXO II.....	45
ANEXO III.....	86
ANEXO IV	87

RESUMEN

La Cafetería – Comedor de la Escuela Politécnica Nacional no contaba con un sistema de comandas para la gestión de los almuerzos, ya que al presente para ordenar un almuerzo la comunidad politécnica tenía que acudir de manera tradicional, como el realizar largas filas hasta llegar a caja y así su pedido sea tomado por parte del personal del comedor, y posterior a ello pasar a la sección de despacho, y recibir la sopa, el segundo y la bebida respectiva para su almuerzo. Lo que ocasionaba que se generen una alta demanda de personas en las horas de almuerzo, depreciación en la eficacia atención por parte del comedor, y malestar por parte de docentes, personal administrativo y estudiantes que en ocasiones preferían ir a restaurantes cercanos.

Con el fin de brindar una solución a los múltiples problemas que aquejan al comedor estudiantil y con el objetivo de ofrecer un progreso en la gestión de los almuerzos el actual Trabajo de Integración Curricular ha desarrollado un *frontend* denominado 'Comedor EPN', el cual se encuentra orientado a brindar un servicio de comandas, el encargado del comedor estudiantil puede gestionar el menú diario, los pedidos generados por parte de la comunidad politécnica, además de generar colaboradores del comedor. Logrando de esta forma que los pedidos de almuerzos solicitados por docentes, personal administrativo y estudiantes sean atendidos de forma adecuada y eficiente, y a la vez reducir la alta concentración de personas en la cafetería.

El presente trabajo de integración curricular está estructurado de la siguiente manera: en el primer capítulo se detallan el problema a resolver, objetivos, alcance del sistema *web* seguido del marco teórico. En el segundo capítulo se enfoca a la ejecución de la metodología *Scrum*, prototipos, y herramientas utilizadas para el desarrollo del sistema *web*, además, del diseño arquitectónico. En el tercer capítulo se presenta las actividades de interacción y resultado de cada *Sprint*. Finalmente, en el cuarto capítulo se describe las conclusiones y recomendaciones que se han obtenido según se avanzó con el desarrollo del sistema *web*.

PALABRAS CLAVE: *Frontend*, sistema *web*, *Scrum*, comandas, cafetería, pedidos en línea, *Sprint*, gestión.

ABSTRACT

The Cafeteria - Dining Room of the National Polytechnic School did not have a system of orders for the management of lunches, since at present to order a lunch the polytechnic community had to go in the traditional way, such as standing in long lines to get to the cashier and so their order is taken by the dining room staff, and after that go to the dispatch section, and receive the soup, the second and the respective drink for lunch. This caused a high demand of people during lunch hours, depreciation in the efficiency of the dining room, and discomfort on the part of teachers, administrative staff and students who sometimes preferred to go to nearby restaurants.

In order to provide a solution to the multiple problems that afflict the student dining room and in order to offer progress in the management of lunches, the current work of Curricular Integration has developed a frontend called 'EPN Dining Room', which is oriented to provide a service of orders, the manager of the student dining room can manage the daily menu, orders generated by the polytechnic community, in addition to generating dining room collaborators. In this way, the lunch orders requested by teachers, administrative staff and students are attended in an adequate and efficient way, and at the same time reduce the high concentration of people in the cafeteria.

This curricular integration work is structured as follows: the first chapter details the problem to be solved, objectives, scope of the web system followed by the theoretical framework. The second chapter focuses on the execution of the Scrum methodology, prototypes, and tools used for the development of the web system, as well as the architectural design. The third chapter presents the interaction activities and results of each Sprint. Finally, the fourth chapter describes the conclusions and recommendations that have been obtained as the development of the web system progressed.

KEY WORDS: *Frontend, web system, Scrum, orders, cafeteria, online ordering, Sprint, management.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Hoy en día existen sistemas que ayudan en el control de una comanda, se busca substituir el proceso manual para atender a clientes por una comanda digital.

Existen en la red de internet algunos sistemas y aplicaciones de ayuda en funciones de: manejo de pedidos, manejo de usuarios, manejo de cuentas, pagos y el análisis de estadísticas. Estas funciones mencionadas, las ofrecen algunos sistemas *webs* y aplicaciones que fueron tomadas de una revisión no sistémica realizada por el equipo del proyecto de Integración Curricular; el sistema *FrontRest* [1], permite al mesero detallar alguna petición especial es decir si desea algún cambio en su pedido como que este con poca sal o el termino de cocción de la carne, además que se asemeja a las aplicaciones *Take it Easy* y *Comandas* [2], ya que dichas aplicaciones presentan como principal característica que el usuario gestione el pedido, y que su interfaz sea intuitiva. También está el sistema *Simplygest* [1], en el cual se enfoca al cliente debido a que puede gestionar su pedido desde casa e incluso reservar una mesa.

En el comedor de la Escuela Politécnica Nacional (EPN) durante el transcurso del día se ofrece variedad de desayunos, almuerzos, meriendas y platos especiales, bocadillos de sal o dulce y snacks, y está orientado a los estudiantes de la EPN [3]. La EPN actualmente tiene un promedio de 1100 docentes y administrativos y 6250 estudiantes [4]. Se puede afirmar entonces que la cafetería es el sitio principal al que acude la comunidad politécnica para poder alimentarse. Pese a esto, se generan varios problemas con el servicio que brindan, siendo evidente problema la aglomeración en las horas pico entre las 12:00 y las 14:00 horas. El personal del comedor no logra darse abasto y no se brinda una satisfactoria experiencia a los clientes [3]. Debido a esto, los clientes tienen que buscar nuevas alternativas para su alimentación diaria, con los consiguientes problemas por riesgos de ingesta de alimentos no apropiados, desperdicio de tiempo, distancia de los comedores privados y el aumento económico que estas opciones significan.

Otro problema que tienen los usuarios es el desconocimiento del menú y sus precios que puede variar cada día de la semana, lo que ocasiona pérdida de tiempo y dinero al momento de alimentarse.

Es así como en el presente proyecto de Integración Curricular plantea el desarrollado un frontend del sistema *web* orientado a desplegar información de los menús y la comanda para la cafetería/comedor de la EPN; esto permite apoyar a la solución de los problemas antes mencionados; el sistema *web* brinda a los usuarios una interfaz sencilla y fácil de navegar, con la gestión del menú y control de la comanda, facilitando el acceso a este

servicio por medio de dispositivos con acceso a internet, donde toda la gestión de la información se lo realizará desde un sistema *web*, el cual estará dividido en dos componentes *backend* y *frontend* para una mejor organización y escalabilidad.

1.1 Objetivo general

Desarrollar un sistema *web* y aplicación móvil para comandas en la cafetería EPN.

1.2 Objetivos específicos

- Establecer los requerimientos para el *frontend*
- Diseñar la interfaz del usuario en base a los requerimientos obtenidos.
- Compilar los módulos del *frontend* en base a los requerimientos obtenidos.
- Realizar pruebas de funcionamiento del *frontend* consumiendo la API del *backend*.

1.3 Alcance

En el presente trabajo de integración curricular se propone el desarrollo de un sistema *web* para el manejo de comandas del comedor de la EPN, donde el sistema cuenta con varios submódulos para cada perfil de esta manera lograr facilitar la gestión de pedidos, la elaboración de estos y la interacción con la comunidad de la politécnica de manera intuitiva y fácil, debido a que el desarrollo prioriza, el consumo correcto de datos, relación de interfaces y sobre todo la experiencia de usuario.

El desarrollo del *frontend* tiene como prioridad brindar un correcto funcionamiento y adaptabilidad mediante el consumo de varios *endpoints* generados por el componente *backend* para así presentar la información en tiempo real y de forma compacta en cada una de las interfaces en distintos navegadores *web*. Mediante el uso de varias herramientas para desarrollar un excelente código y metodología.

A continuación, se describirá de manera general las funcionales del *frontend*, el cual contará con los siguientes perfiles:

Perfiles dentro del *frontend*

- Administrador
- Cliente
- Cocinero

Perfil Administrador

- Presentar una página informativa
- Iniciar sesión, cerrar sesión y recuperar contraseña mediante el consumo de *endpoints*
- Modificar perfil de usuario mediante el consumo de *endpoints*
- Gestionar el menú de los almuerzos mediante el consumo de *endpoints*
- Generar usuarios cocineros mediante el consumo de *endpoints*
- Visualizar el pedido y la factura pagada mediante el consumo de *endpoints*
- Asignar el pedido al cocinero mediante el consumo de *endpoints*
- Visualizar reporte de pedidos completados mediante el consumo de *endpoints*.

Perfil Cliente

- Registrarse como cliente mediante el consumo de *endpoints*.
- Iniciar sesión, cerrar sesión y recuperar contraseña mediante el consumo de *endpoints*
- Modificar perfil de usuario mediante el consumo de *endpoints*.
- Gestionar pedido mediante el consumo de *endpoints*.
- Cargar el archivo de confirmación de pago mediante el consumo de *endpoints*.
- Visualizar el estado del pedido mediante el consumo de *endpoints*.

Perfil Cocinero

- Iniciar sesión, cerrar sesión y recuperar contraseña mediante el consumo de *endpoints*.
- Modificar perfil de usuario mediante el consumo de *endpoints*.
- Visualizar los pedidos asignados por el usuario administrador mediante el consumo de *endpoints*.
- Cambiar estado del pedido mediante el consumo de *endpoints*.

1.4 Marco Teórico

El sistema *web* o páginas *web* es uno de los más manipulados por sus poderosas funcionalidades. Además, que para poder hacer uso de ellas se lo realiza desde cualquier navegador *web*, y bueno los sistemas *web* trabajan con bases de datos de tal forma que se puede procesar y hacer visible información a los usuarios.

Ahora se aborda la fase de estudio y análisis de las tecnologías empleadas en el desarrollo del *frontend*. Para que el resultado del desarrollo se muestre amigable, accesible y automatizado.

En el desarrollo del *frontend* se hace uso de las metodologías, para así optimizar los procesos y el producto, con métodos que guían a los involucrados tanto en la planificación como en el desarrollo; una metodología define que, como y cuando hacer determinada actividad durante todo el proceso de desarrollo y mantenimiento de la página *web* [5]. Por ende, ayuda a definir una estrategia global para llevar a cabo el desarrollo del proyecto de Integración Curricular, con elementos como: fases, las cuales son tareas para realizar en cada etapa; productos y entregables; y finalmente, criterios de evaluación, que permiten cumplir los objetivos planteados del proyecto.

Entre los últimos años, las metodologías ágiles han tomado fuerza en el desarrollo de *software*, como lo indica un estudio reciente de *Project Manager Institute* (PMI), es así como para el desarrollo de este proyecto se trabaja con una metodología ágil [6]. Debido a que permite al equipo brindar respuestas rápidas a los cambios constantes que se generan en el desarrollo del proyecto, las metodologías ágiles tienen enfoques flexibles y usan el trabajo autoorganizado del equipo garantizando mejoras.

En cuanto a las plataformas digitales estas se encuentran divididas en *frontend* y *backend*. El *frontend* como bien su nombre lo dice es la parte visible de una plataforma digital, en otras palabras, es lo que puede observar el usuario final. Además, de ser el encargado de determinar la jerarquía de la información que se puede ver y su distribución. El producto final es lo que la mayoría conoce como: una página *web*. En la parte del *frontend* se trabaja con lenguajes destinados a la experiencia del usuario como: HTML, CSS, *JavaScript* y *TypeScript*.

HTML es el lenguaje de marcado de hipertexto, que nos ayuda a definir la estructura de la página *web*, o mejor dicho es el código que comúnmente se maneja para poder realizar la estructuración y despliegue de la página *web* y sus contenidos [7]. Y CSS es un lenguaje

de hojas de estilo, lo que significa, nos autoriza ejecutar estilos de forma selectiva al contenido HTML, para así darle una mejor estructura y estilo [7].

Para el funcionamiento del componente *frontend* se trabaja con *TypeScript* el cual es un lenguaje de programación que se encuentra a un nivel más alto que *JavaScript* (JS) o conocido también como un superconjunto de *JavaScript*, debido a que permite escribir un código con menos errores, más sencillo, coherente o en otras palabras se desarrolla un código más limpio y sólido [8].

Los *Frameworks* en palabras sencillas es una organización previa ya que es parecido a una plantilla, un esquema conceptual y también un *Framework* nos ayuda a desarrollar un proyecto en un tiempo no tan largo y con un código más nítido [9], luego están las librerías que aumentan la capacidad de crear cualquier tipo de interfaz de usuario como React, Angular y Bootstrap entre otros [7].

El *Framework* que se usa para el desarrollo del proyecto de Integración Curricular es Vue.js, debido a que nos permite dividir en bloques el sistema *web* y cada uno de los bloques con su funcionalidad independientemente de la otra, y así comunicarse de forma sencilla [10]. También, ayuda para el diseño de vistas reactivas para cada estado, lo que significa que cuando se realiza cambios en los datos del sistema, los desarrolladores no tengan que realizar de manera manual, sino que cambian automáticamente en la parte visual de la página [10].

Una API nos aporta flexibilidad, es decir simplifican el diseño, la administración y el uso, también ofrecen innovación, que es genial al momento de diseñar [11].

El consumo e integración de datos por parte del *backend* se realiza por una serie de *endpoints* establecidos y configurados. Estos datos lo proveen una API, para ser consumidos por parte del *frontend* y ser presentados al cliente [12]. Además, permite trabajar y gestionar con documentos, videos, imágenes y toda la información requerida para presentar en cada una de las interfaces

El *Testing* es sumamente importante en el desarrollo de un sistema o página *web* ya que así podemos confirmar con certeza la calidad del proyecto de *software*. El *testing* es un proceso para confirmar y validar el desarrollo del proyecto. Para ello se debe realizar la ejecución de componentes y herramientas automatizadas para así testear la funcionalidad antes de su lanzamiento [13]. Puesto que tiene el objetivo de efectuar todos los requisitos esperados para la entrega del proyecto final.

El despliegue es de gran importancia ya que la aplicación *web* pasa de estar alojada en un servidor localmente a uno externo dando la posibilidad que los usuarios tengan mayor acceso siempre y cuando tengan una conexión estable a internet.

2 METODOLOGÍA

Los estudios de casos son una metodología que se usa en la gran parte de las disciplinas, para generar avances y además esta metodología suele juntarse con métodos cualitativos, de forma efectiva, pero admite un tratamiento de información tanto cualitativa como cuantitativa. Bien para el caso de estudio el primer paso es la recopilación de datos. Se debe tener en cuenta que no se trata de una consulta de laboratorio sino de algo del mundo real. Es así como el estudio de casos permite dar respuesta a manifestaciones por medio de varias investigaciones [14].

Por lo antes mencionado, el presente proyecto de Integración Curricular hace uso del estudio de casos, ya que se realiza investigaciones sobre la problemática del aglomeramiento que existe en la cafetería de la EPN por parte de la comunidad politécnica. Lo cual, permite realizar el desarrollo de un *frontend* para crear un pedido, modificar el perfil, gestionar el estado de los pedidos y subir los datos referentes a su pago, ayudando a disminuir la alta demanda de personas en las horas pico de almuerzo.

2.1 Metodología de Desarrollo

En el desarrollo de *software*, las técnicas y métodos organizativos se usan para diseñar soluciones. Tienen el objetivo de organizar el desarrollo del proyecto para que permita una gran escalabilidad de éxito. Por eso es por lo que dependiendo del tipo de metodología dividen o reparten las tareas y responsabilidades teniendo en cuenta el cumplimiento del objetivo ya antes mencionado [15].

Las metodologías ágiles se basan en la metodología incremental, es decir que luego de un cierto ciclo se van cargando nuevos cambios o funcionalidades al sistema *web*. Además, se itera sobre bloques de tiempos cortos y fijos para obtener un resultado completo cada iteración [15]. Por eso se escogió para el desarrollo del proyecto de Integración Curricular la metodología *Scrum*, ya que busca darle fuerza a la velocidad, a la capacidad de adaptación y a los cambios inesperados de los requisitos para la aplicación. Permitiendo el desarrollo del proyecto, basándose en entregas parciales y regulares [15]. A continuación, se explica la manera en que se realizó la implementación de la metodología en el desarrollo del *frontend*

Roles

Scrum está basado en roles, donde cada participante tiene diferentes responsabilidades y se centran en construir un *software* de calidad, los roles que se usan son un *Product Owner*, *Scrum master* y el *Development*, por eso, para el desarrollo de cada una de las

actividades que existen para el *frontend*, cada uno de los integrantes del equipo tienen asignados un rol, y a continuación, se presentan los roles que participan en el desarrollo del *frontend*.

Product Owner

Es el encargado de garantizar una buena comunicación sobre el producto, define los criterios y se asegura que se cumplan. Por efecto, la **TABLA I** presenta la persona encargada, quien proporciona la información necesaria que ayuda a identificar los requerimientos y herramientas para el desarrollo de la página *web*.

Scrum master

Esta entidad es responsable de asegurar que ha comprendido el método, ya que es líder y ayudante, esencialmente de cuyo equipo denominado Equipo *Scrum*, con el objetivo de hacerles comprender a las personas externas del equipo como influye su rol dentro del equipo, de igual manera prevaler su interacción con el equipo y verificar si éstas son beneficiarias o no para dicho equipo *scrum* [16].

El *Scrum master*, tiene como principal característica, la responsabilidad de proporcionar que ha manejado con entendimiento y aprobación el método *Scrum*. Por efecto, la **TABLA I** presenta la persona encargada de este rol, ya que cuenta con los conocimientos necesarios para asegurar un buen ambiente de trabajo.

Development Team

Confirmado por un equipo previamente establecido por personas especializadas y comprometidas, que laboran con el fin de entregar un incremento de producto "Finalizado", cuyo equipo es el encargado de organizar y dirigir sus actividades laborales.

Development Team tiene una importante característica la auto organización, la cual permite que de forma liberal no opten o dependan de alguna autoridad, por ende, sean más bien un equipo de forma independiente en la toma de decisiones [16]. Por efecto, la **TABLA I** presenta la persona encargada.

TABLA I: Designación de Roles en el *frontend*

ROLES	NOMBRES
<i>Product Owner</i>	Ing. Yadira Franco
<i>Scrum master</i>	Ing. Yadira Franco
<i>Development Team</i>	Mateo Cueva

Artefactos

En *Scrum* los artefactos es la información que el *Development Team* maneja para detallar el proyecto de *software*, con las acciones y tareas que se debe realizar y desempeñar durante el desarrollo del proyecto, ya que están pensados en garantizar la transparencia de información [17]. Además, ofrece herramientas que admite un mejor rendimiento en cada *Sprint*. Posteriormente, se presenta los artefactos utilizados en el desarrollo del *frontend*.

Recopilación de Requerimientos

La recopilación de requerimientos en base a lo establecido por la metodología *Scrum*, es la base y el punto de partida con mayor importancia para el desarrollo del sistema, ya que ayuda en la captura y esquematización de las funcionalidades con las que esta un sistema, además de designa las operaciones que se ejecutan en desarrollo del proyecto [18]. Por efecto, la **TABLA II** indica la plantilla que se ha usado para la recopilación de requerimientos y la organización adecuada de cada uno para el desarrollo del proyecto.

TABLA II: Requerimientos recopilados

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
Sistema Web	RR001	Los usuarios con perfil cliente, administrador y cocinero visualizan una página informativa de la Cafetería.

Historias de Usuario

Las historias de usuario contribuyen a describir con más detalle los requerimientos que capturados anteriormente, todo esto desde la perspectiva del usuario final [19]. Esto ayuda al desarrollador a tener una mejor comprensión de lo que quieren los usuarios, y así cumplir con la funcionalidad solicitada. Además, permiten identificar aspectos como: tipo de usuario, riesgo, prioridad. Ante lo mencionado en la **TABLA III** se presenta las Historias de Usuario que contienen las funcionalidades establecidas en el desarrollo del *frontend*.

TABLA III: Historia de usuario 001 – Visualizar página informativa de la Cafetería

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Cocinero, cliente, administrador
Nombre Historia: Visualizar página informativa de la Cafetería	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: El usuario cocinero, cliente, administrador podrá visualizar la información general de la Cafetería e interactuar con los siguientes módulos: <ul style="list-style-type: none"> • Iniciar Sesión • Registrarse 	
Observación: El usuario cocinero, cliente, administrador de la Cafetería pueden consumir varios <i>endpoints</i> del <i>backend</i> dependiendo del inicio de sesión.	

Product Backlog

Dentro del desarrollo del proyecto consiste en la creación de un listado de todas las tareas por realizar para el desarrollo del *frontend* con el fin de que sean visible para cada uno de los integrantes del equipo. En esta lista podemos tomar datos como el número de la tarea a realizar el estado de esta y su prioridad, así conseguimos organizar por prioridades los requerimientos y también de hacer un seguimiento de cada una [20]. Ante lo mencionado la **TABLA IV** muestra la elaboración del *Product Backlog* de acuerdo con los requerimientos solicitados

TABLA IV: *Product Backlog*

ID – HU	HISTORIA DE USUARIO	ITERACION	ESTADO	PRIORIDAD
HU001	Presentar página informativa de la Cafetería	1	Finalizado	Media

Sprint Backlog

Se trata de todas las tareas e iteraciones que se debe realizar en el desarrollo del proyecto las mismas se encuentran organizadas en una **TABLA** de acuerdo con la prioridad. Se realizan entregas parciales permitiendo ir testeando el avance del proyecto, esto quiere decir que el principal objetivo de este es contar con una transparencia dentro del desarrollo [21]. Ante lo mencionado la **TABLA V** indica la elaboración del *Sprint Backlog* en el cual se ha establecido 5 *Sprint* los cuales son: Configuración del ambiente, Consumir *endpoints* para el perfil administrador, Consumir *endpoints* para el perfil cliente, Consumir *endpoints* para el perfil cocinero, Pruebas en el *frontend* y Despliegue del *frontend*.

TABLA V: *Sprint Backlog*

ELABORACIÓN DEL SPRINT BACKLOG					
ID – SB	NOMBRE	MÓDULO	ID - HU	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del ambiente de desarrollo.	-----	-----	<ul style="list-style-type: none">• Definir los requerimientos no funcionales y funcionales del <i>frontend</i>.• Configurar las herramientas del <i>frontend</i>.• Configurar el ambiente de desarrollo del <i>frontend</i>.	20 horas

2.2 Diseño de interfaces

Se hace referencia a la estructura o prototipo diseñado para el sistema *web*, el cual se encuentra basado según los requerimientos obtenidos y establecidos con el *Product Owner*. Además, que las interfaces nos permiten interactuar con el prototipo del proyecto como si este fuera el resultado final. Las interfaces se desarrollan por medio de componentes como texto, imágenes, controles, botones, casillas, alcanzando a ser agradable para el cliente y prevenir posibles errores [22].

Herramienta utilizada para el diseño

Mediante la herramienta Figma que admite diseñar prototipos de interfaces, se forman los diferentes prototipos de cada uno de los módulos definidos en el *frontend* [23], los cuales toman mayor importancia en la fase de desarrollo.

Para el prototipo del sistema *web* se ha desarrollado prototipos y módulos según los requerimientos establecidos, para así tener una base para codificar la funcionalidad y lógica del negocio. Por efecto, en la **Fig. 1** se indica una de las vistas del prototipo diseñado y ejecutado.



Fig. 1: Prototipo de información general del sistema *web*

2.3 Diseño de la arquitectura

La integración de un patrón de arquitectura permite codificar un *software* que permita agregar mayor funcionalidad, sea escalable a futuro y que garantice una alta compatibilidad con nuevas herramientas y librerías [24]. Por consiguiente, en la siguiente sección se detalla la arquitectura que se ha implementado para el desarrollo del *frontend*.

Patrón arquitectónico

El patrón arquitectónico MVC también conocido como Modelo, Vista, Controlador es un patrón de diseño de *software* utilizado para construir interfaces de usuario, datos y lógica

de control. Se enfoca en dividir la lógica de negocios y visualización. Esta división proporciona una mejor organización y control durante el desarrollo del proyecto [25].

- **Modelo**
Especifica los datos que va a contener y va a interactuar la aplicación es decir se realiza toda la parte lógica.
- **Vista**
Define como se muestran los datos en la aplicación.
- **Controlador**
Es la lógica que actualiza el contenido entre el modelo y vista dando respuesta a las interacciones de los usuarios con la aplicación.

De tal manera, en la **Fig. 2** se presenta la implementación del patrón arquitectónico que se ha utilizado en el transcurso del desarrollo del *frontend*, lo que admite una mejor combinación con todas las herramientas y librerías que se manejan para la fase de codificación, pruebas y despliegue.

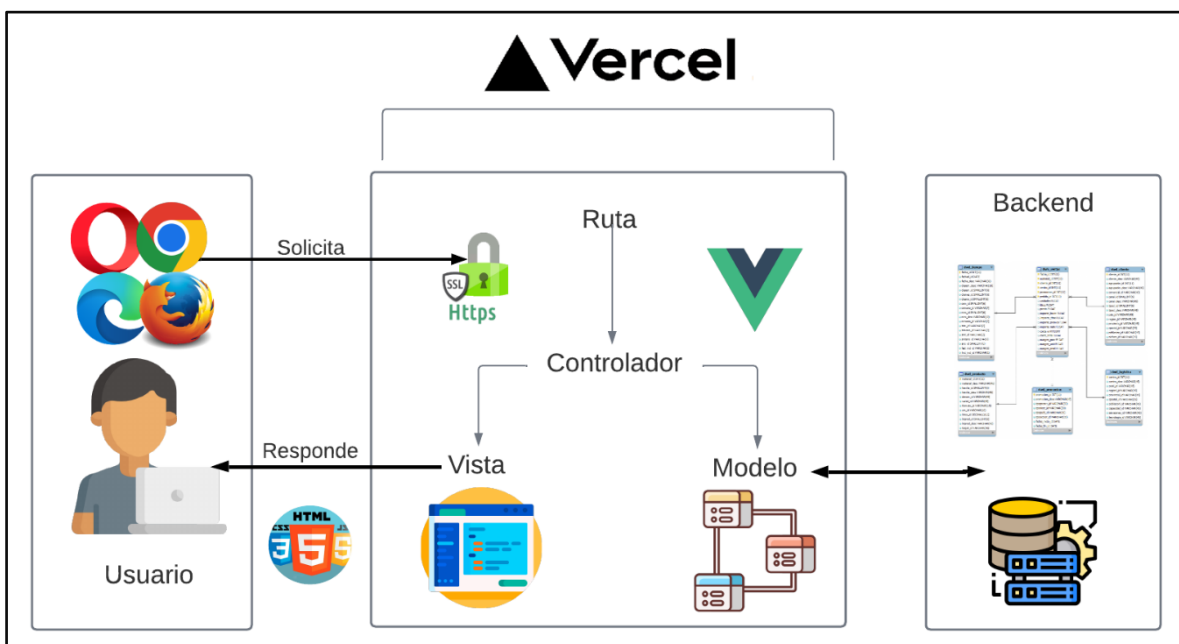


Fig. 2: Patrón arquitectónico – Sistema Web

2.4 Herramientas de desarrollo

El objetivo principal de las herramientas de desarrollo es facilitar, optimizar y tener un mejor desempeño en el desarrollo del proyecto. En base a los requerimientos establecidos como lo muestra la siguiente **TABLA VI**.

TABLA VI: Herramientas para el desarrollo del Sistema *Web*

Herramienta	Justificación
Visual Studio Code	Con Visual Studio Code, se puede escribir código de manera más ágil, además de que nos permite realizar depuración del código y encontrar errores [26].
Vue.js	Con el <i>Framework</i> Vue.js, se logra actualizar y renderizar de manera fácil y ágil los componentes correctos cuando cambian los datos [27].
<i>TypeScript</i>	Con <i>TypeScript</i> , se logra brindar interactividad y eficacia al sistema <i>web</i> [28].
Vercel	Con Vercel el equipo de desarrolladores puede desplegar, y gestionar el sistema <i>web</i> [29].

Librerías

La **TABLA VII**, presenta la descripción de las librerías que se han utilizado para el desarrollo del actual sistema *web*.

TABLA VII: Librerías para el desarrollo del Sistema *Web*

LIBRERÍA	Justificación
Vuetify	Ayuda en el diseño de interfaces de usuario, formas, efectos e interacciones [30].
Axios	Ayuda a realizar peticiones o llamadas hacia el cometido que se encuentra en una ruta HTTP [31].

3 RESULTADOS

Ahora, en el siguiente capítulo se presentará a más detalle cada uno de los procesos realizados en cada una de las interfaces del *frontend*, juntamente con las respectivas pruebas y el despliegue de la página *web* a un entorno de producción. Además, cada uno de los resultados obtenido se presentan mediante una serie de interacciones como se tiene planificado.

3.1 Sprint 0. Configuración del ambiente de desarrollo

Las tareas que implican el cumplimiento del Sprint 0 son las siguientes:

- Definición de los requerimientos para el sistema *web*.
- Estructura del sistema *web*.
- Roles de usuario para el sistema *web*.

Definición de los requerimientos para el sistema *web*

Consumir varios *endpoints* para visualizar página informativa de la Cafetería

En el sistema *web* por medio de varios procedimientos los usuarios con el perfil de administrador, cliente y cocinero tienen que consumir diversos *endpoints* para visualizar página informativa de la Cafetería, esto quiere decir que, los usuarios tienen la posibilidad de observar una página principal que muestra un menú de navegación con varias secciones como: Iniciar Sesión, y Registrarse. Cabe recalcar que para visualizar la página informativa no se necesita iniciar sesión cualquier usuario puede observar e interactuar con su contenido.

Consumir varios *endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña

En el sistema *web* por medio de varios procedimientos el usuario con perfil administrador, cliente y cocinero tienen que consumir diversos *endpoints* para que puedan ingresar y salir de su modulo asignado e incluso recuperar su contraseña, las credenciales (correo electrónico y contraseña) para el usuario administrador ya se encuentran creadas, mientras que para el cocinero las credenciales de acceso le otorga el administrador y por último el usuario cliente obtiene las credenciales según los datos que haya ingresado en el formulario de registro.

Consumir varios *endpoints* para modificar perfil

En el sistema *web* por medio de varios procedimientos los usuarios con perfil cocinero y cliente tienen que consumir diversos *endpoints* para modificar su perfil, esto quiere decir que los cocineros, y los clientes pueden visualizar su perfil general, en el cual pueden actualizar su información personal o realizar algún cambio.

Consumir varios *endpoints* para generar usuarios cocineros

En el sistema *web* por medio de varios procedimientos el usuario con perfil cocinero tiene que consumir diversos *endpoints* para ingresar a su módulo asignado con las credenciales que le entrega el usuario administrador, ya que el administrador es el que tiene el control de gestionar a los usuarios de perfil cocinero de manera que él puede crear, eliminar y visualizar la información de un cocinero.

Consumir varios *endpoints* para visualizar el pedido y la factura pagada

En el sistema *web* por medio de varios procedimientos el usuario con perfil administrador tiene que consumir diversos *endpoints* para visualizar el pedido y la factura cancelada, esto quiere decir que, el administrador tiene que visualizar el detalle del pedido realizado, como: que sopa pidió, que cantidad de arroz y que proteína, etc. Además, verificar en el panel que dicho pedido ya se encuentre pagado por el cliente.

Consumir varios *endpoints* para agregar platos del menú del almuerzo y snacks

En el sistema *web* por medio de varios procedimientos el usuario con perfil administrador tiene que consumir diversos *endpoints* para agregar platos del menú del almuerzo y snacks, esto quiere decir que, el administrador tiene que ir llenando 2 tipos de formularios el primero para agregar un snacks que puede ser una gaseosa, golosinas, pasteles con los campos de nombre del snack, precio, número de stock y en el segundo formulario agregar el plato del menú que son sopa, segundo, bebida con los campos de nombre del plato, descripción, precio, tipo.

Consumir varios *endpoints* para generar el menú de los almuerzos

En el sistema *web* por medio de varios procedimientos el usuario con perfil administrador tiene que consumir diversos *endpoints* para gestionar el menú de los almuerzos, esto quiere decir que el administrador es quien agrega, elimina, actualiza y observa el formulario del pedido, en cual describe los platos de comida que se encuentran disponibles en el día.

Consumir varios *endpoints* para asignar el pedido al cocinero

En el sistema *web* por medio de varios procedimientos el usuario con perfil administrador tiene que consumir diversos *endpoints* para asignar el pedido al cocinero, esto quiere decir que, el administrador visualiza los pedidos y toma uno de los pedidos ya validados para seguidamente asignarle a algún cocinero que se encuentre libre.

Consumir varios *endpoints* para visualizar reporte de pedidos completados

En el sistema *web* por medio de un proceso el usuario con perfil administrador tiene que consumir diversos *endpoints* para visualizar los pedidos que han sido completados satisfactoriamente, en la vista se observa el nombre del cliente que realizó el pedido más el detalle del pedido con su respectivo comprobante de pago.

Consumir varios *endpoints* para registrarse

En el sistema *web* por medio de varios procedimientos el usuario con perfil cliente tiene que consumir diversos *endpoints* para registrarse, esto quiere decir que, el cliente al no poseer una cuenta de acceso al sistema tiene que dirigirse al módulo de registro en el cual está un formulario en el que el cliente debe proporcionar información real y seguidamente las credenciales ingresadas son las que usará para el inicio de sesión. Para el caso del usuario con perfil cocinero el registro es diferente, ya que quien realiza el registro del usuario es el administrador desde el formulario con información proporcionada por el cocinero y con la diferencia que las credenciales de acceso al sistema son enviadas al correo entregado por el usuario.

Consumir varios *endpoints* para generar pedidos

En el sistema *web* por medio de varios procedimientos el usuario con perfil cliente tiene que consumir diversos *endpoints* para generar pedidos, esto quiere decir que el cliente accede a su módulo y en él se dirige a crear un pedido el cual es tipo formulario que se tiene que ir armando según él los platos disponibles, los gustos, y así es la manera de generar los pedidos.

Consumir varios *endpoints* para visualizar el estado del pedido

En el sistema *web* por medio de varios procedimientos el usuario con perfil cliente tiene que consumir diversos *endpoints* para visualizar el estado del pedido, esto quiere decir que el cliente selecciona otro panel en el cual está información general del pedido generado, además, de su estado los cuales son: validando pedido, pedido en proceso y pedido finalizado. Cabe mencionar que el cliente no puede editar su pedido una vez confirmado.

Consumir varios *endpoints* para visualizar los pedidos asignados

En el sistema *web* por medio de varios procedimientos el usuario con perfil cocinero tiene que consumir diversos *endpoints* para visualizar los pedidos asignados, esto quiere decir que en el panel principal el cocinero visualiza de manera más detalla la orden, como el número de pedido y el contenido del pedido a realizar.

Consumir varios *endpoints* para cambiar el estado de los pedidos

En el sistema *web* por medio de varios procedimientos el usuario con perfil cocinero tiene que consumir diversos *endpoints* para cambiar el estado de los pedidos, esto quiere decir que el cocinero puede cambiar el estado del pedido entre: pedido en proceso y pedido finalizado, mismo que se indica en el panel del cliente, pero, con la diferencia que el cocinero puede modificar el estado del pedido. Finalmente, en la **Fig. 3**, indica las acciones que cada uno de los usuarios con perfil administrador, cliente y cocinero realizan dentro del sistema *web* para un mejor entendimiento.

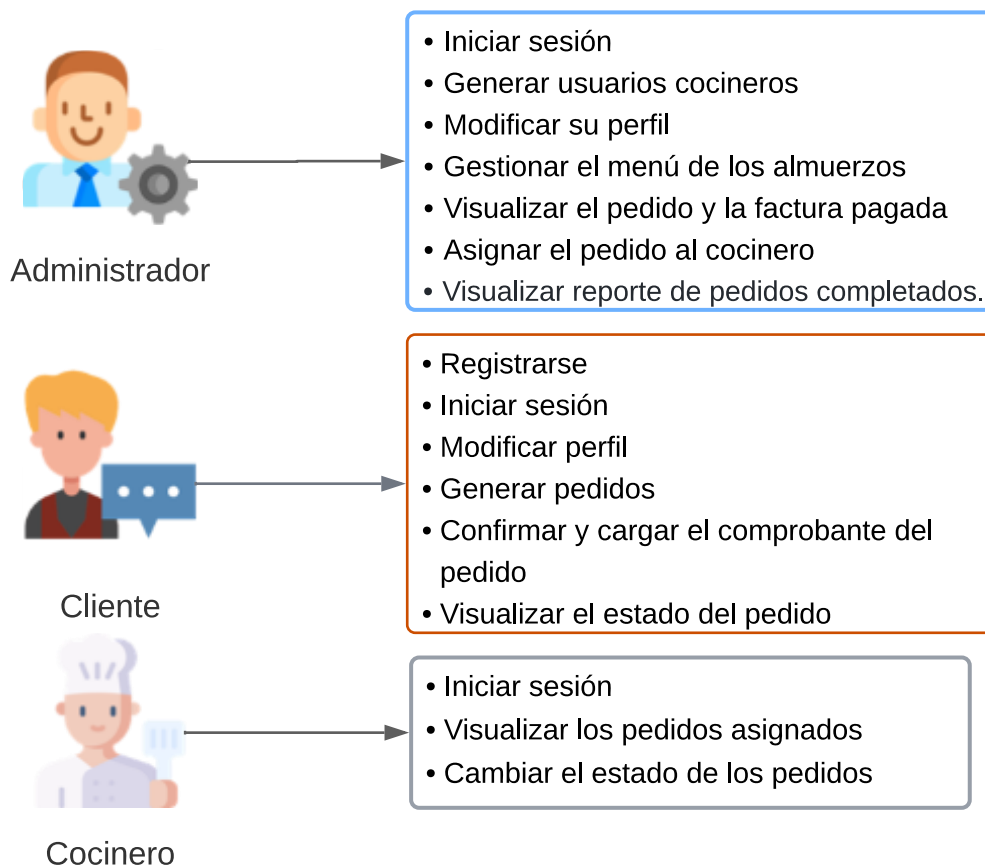


Fig. 3: Usuarios y funcionalidades dentro del sistema

Estructura del proyecto sistema web

La herramienta Visual Studio Code se utiliza de entorno de desarrollo del sistema web, mediante el panel de inicio se crea la estructura, organización, archivos y directorios para el progreso del sistema web basándose en el patrón arquitectónico previamente elegido, es así que en la **Fig. 4** se indica la organización del proyecto.

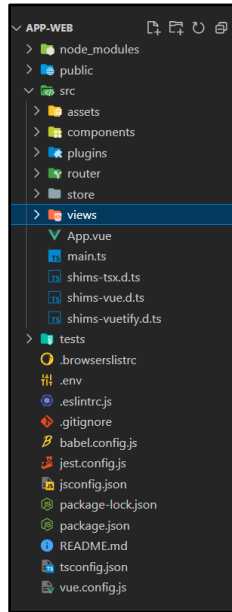


Fig. 4: Estructura del Sistema Web

Roles de usuario

La **Fig. 5** presenta los módulos a los que tienen acceso cada uno de los usuarios: administrador, cliente, y cocinero indicando si es necesario realizar un previo registro.

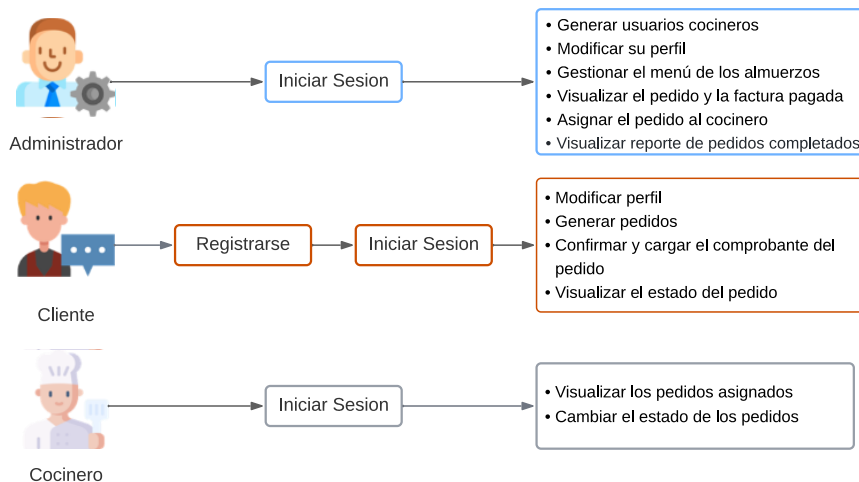


Fig. 5: Usuarios y roles correspondientes

3.2 Sprint 1. Resultado de las interfaces implementadas para el usuario con perfil administrador

Las tareas que implican el cumplimiento del Sprint 1 son las siguientes:

- Consumir varios *endpoints* para visualizar página informativa de la Cafetería.
- Consumir varios *endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña.
- Consumir varios *endpoints* para generar usuarios cocineros.
- Consumir varios *endpoints* para agregar platos del menú del almuerzo y snacks.
- Consumir varios *endpoints* para generar el menú de los almuerzos
- Consumir varios *endpoints* para visualizar el pedido y la factura pagada.
- Consumir varios *endpoints* para asignar el pedido al cocinero.
- Consumir varios *endpoints* para visualizar reporte de pedidos completados.

Consumir varios *endpoints* para visualizar página informativa de la Cafetería

En la página principal del sistema *web* presenta información relevante acerca de la cafetería EPN, entre esa información esta anuncios relevantes, promociones, menú diario y menú de la semana. La página informativa se divide en secciones como: Iniciar Sesión y Registrarse. Además, para observar la página principal no se necesita estar dentro del sistema o registrado, ya que es publica y todo tipo de usuario puede visualizar la página principal. En la **Fig. 6** se presenta la página informativa luego de haber concluido la fase de desarrollo y de pruebas.



Fig. 6: Página principal del sistema *web*

Consumir varios *endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña

Dentro del sistema *web* para iniciar sesión, cerrar sesión y recuperar contraseña se lo realiza mediante un formulario en el caso del inicio de sesión se debe llenar los campos de correo electrónico y contraseña con las credenciales registradas en el formulario de registró en el cual se ingresa datos reales del usuario este es el caso del usuario administrador. Para recuperar la contraseña se lo realiza mediante él envió de un correo al email proporcionado y asi poder restablecer la contraseña del usuario. En la **Fig. 7** y **Fig.8** se presenta las interfaces anteriormente mencionadas, luego de haber concluido la fase de desarrollo y de pruebas.

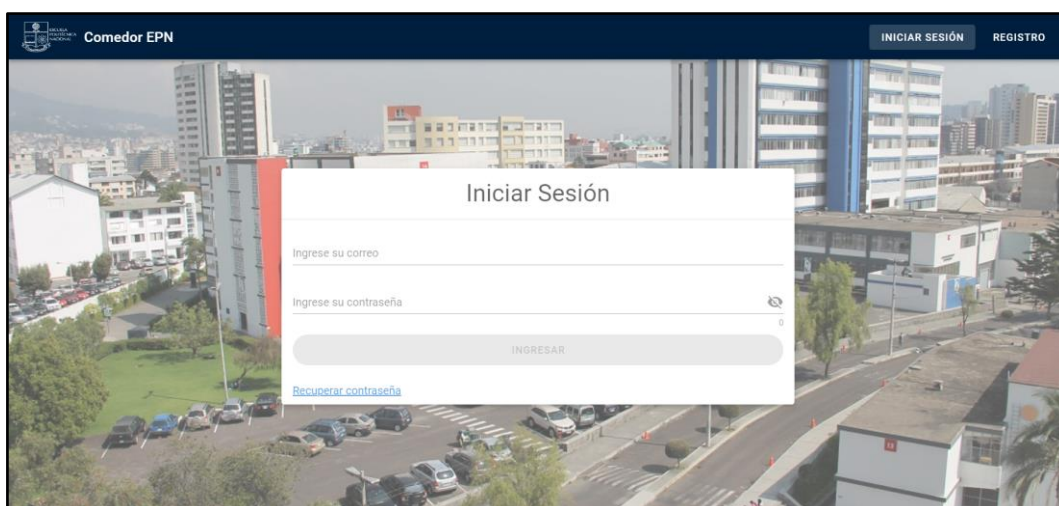


Fig. 7: Página de inicio de sesión del sistema *web*

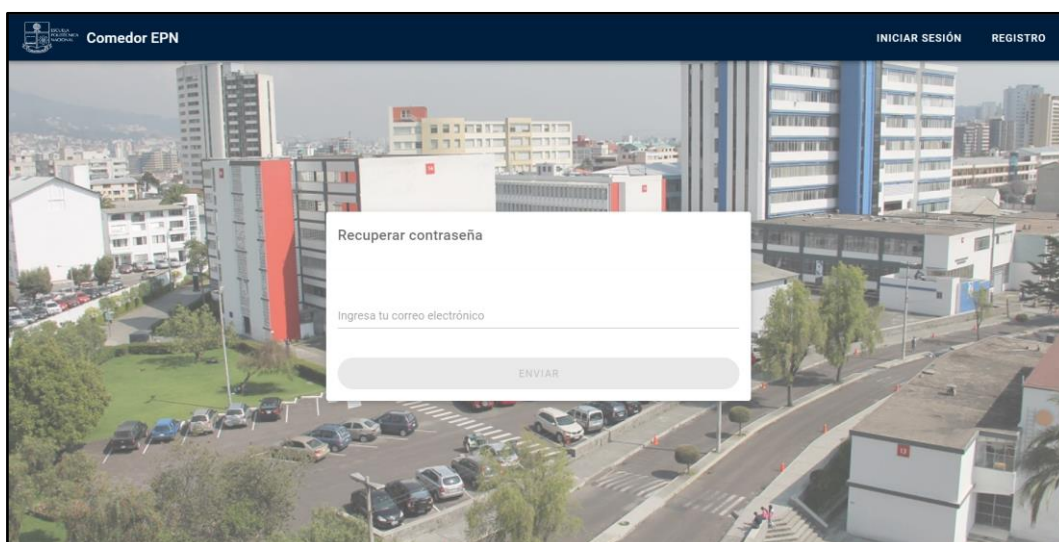


Fig. 8: Página de recuperación de contraseña en el sistema *web*

Consumir varios *endpoints* para generar usuarios cocineros

Para generar nuevos usuarios cocineros solamente el usuario con perfil administrador, cuenta con el acceso a dicho modulo y puede crear un nuevo usuario cocinero, esto quiere decir, para que exista un nuevo usuario con perfil cocinero se necesita tener los datos personales del usuario, para posterior a eso llenar el formulario y crear un nuevo cocinero. En la **Fig. 9** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

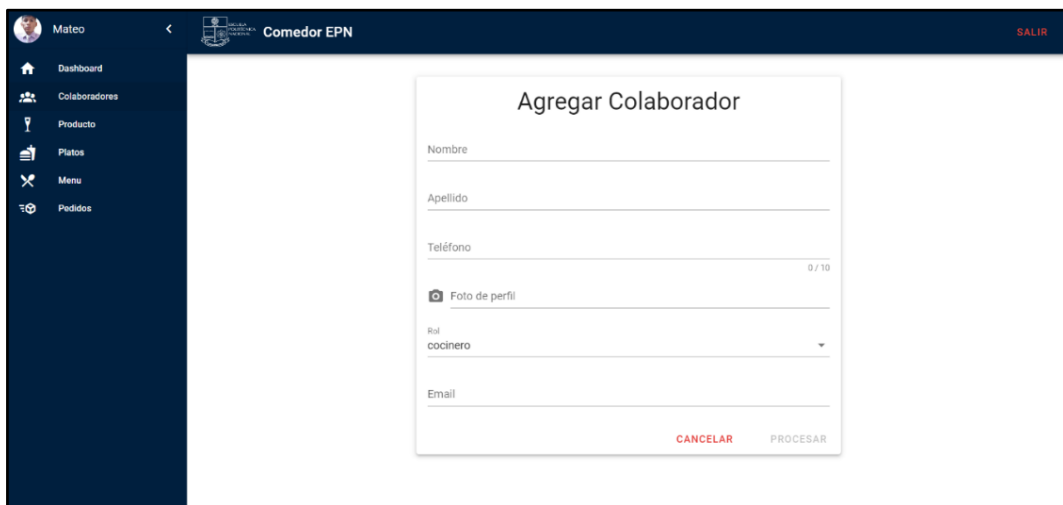
The image shows a web application interface for 'Comedor EPN'. On the left is a dark blue sidebar with a user profile 'Mateo' and a navigation menu with icons and labels: 'Dashboard', 'Colaboradores', 'Producto', 'Platos', 'Menu', and 'Pedidos'. The main content area is white and features a form titled 'Agregar Colaborador'. The form has several input fields: 'Nombre', 'Apellido', 'Teléfono' (with a '0 / 10' character count), 'Foto de perfil' (with a camera icon), 'Email', and a dropdown menu for 'Rol' currently set to 'cocinero'. At the bottom right of the form are two buttons: 'CANCELAR' in red and 'PROCESAR' in grey.

Fig. 9: Página para generar un nuevo colaborador

Consumir varios *endpoints* para agregar platos del menú del almuerzo y snacks

Para generar el menú del día, antes se genera los productos y platos que tiene disponible la cafetería entre los productos están: papas, galletas, chifles, bebidas entre otros, para agregar un producto se llena el formulario que solicita el nombre, el precio y el stock del producto a agregar, y los platos que se preparó para el día dicho formulario cuenta con el nombre del plato, una pequeña descripción, el precio, el tipo de plato (sopa, segundo, bebida), y además, agregar una foto del plato, para agregar los productos y platos se llena un formulario destinado para cada uno, y solamente el usuario con perfil administrador tiene acceso a este módulo, también se pueden eliminar, editar. En las **Fig. 10** y **Fig. 11** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

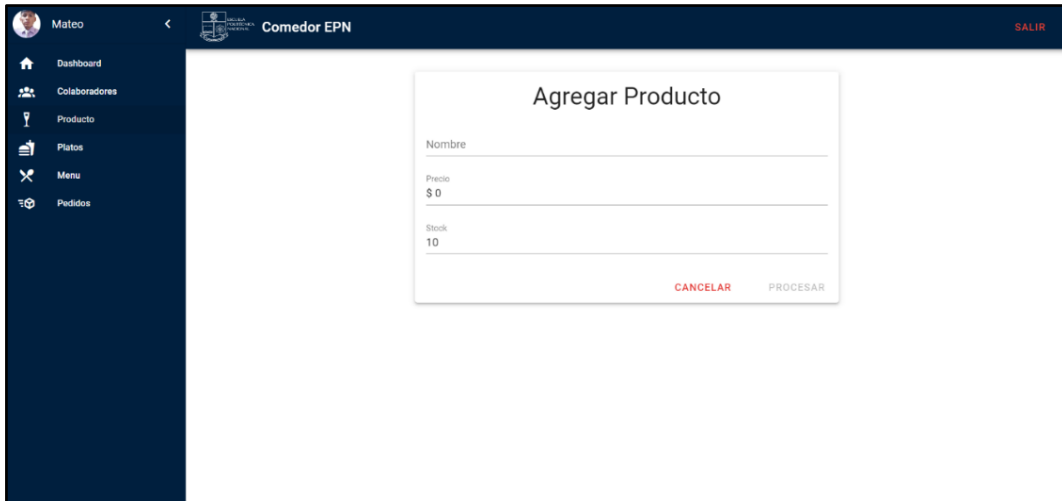


Fig. 10: Página para agregar un nuevo producto de la Cafetería

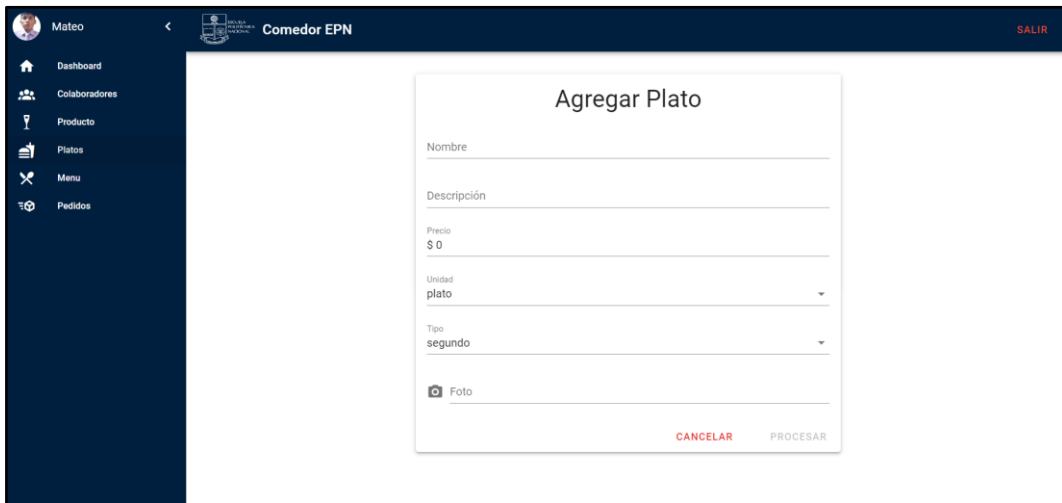


Fig. 11: Página para agregar un nuevo plato del almuerzo

Consumir varios *endpoints* para generar el menú de los almuerzos

En la gestión del menú diario de los almuerzos el usuario administrador tiene la posibilidad de generar, modificar, leer y eliminar el formulario del almuerzo que se prepara día a día, dicho formulario consta del título del menú, la fecha y la opción de seleccionar todos los platos disponibles del día. De tal manera que, en la vista generar pedido del cliente se visualice tanto platos para el almuerzo del día, como productos disponibles de la cafetería. En la **Fig. 12** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

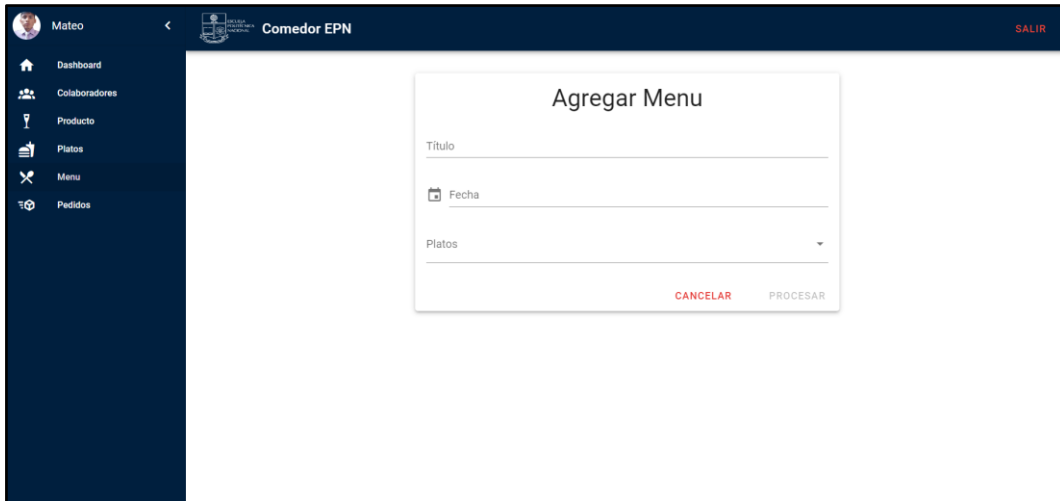


Fig. 12: Página para crear nuevo menú de almuerzo

Consumir varios *endpoints* para visualizar el pedido y la factura pagada

En el módulo de pedidos, solamente el usuario con perfil administrador tiene acceso a realizar las acciones de visualizar el detalle del pedido, asignar el pedido a un usuario cocinero, y rechazar el pedido en el caso que no se efectuó un pago correcto. Para visualizar el pedido el usuario con perfil administrador hace clic en el botón con el icono de un ojo y así puede observar el detalle del pedido, y al momento de hacer clic en el botón con el icono de una hoja de color azul nos redirecciona a observar la foto del comprobante. En las **Fig. 13** y **Fig. 14** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

Cliente	Fecha de creación ↓	Comprobante	Estado	Total	Acción
Pedro Gutierrez	15/1/2023, 12:17:27		pendiente	3.9	
Pepito Lopez	15/1/2023, 11:53:34		completado	2.7	
Juan Lopez	15/1/2023, 11:52:47		verificado	4.7	
Pepito Lopez	12/1/2023, 22:46:59		denegado	2.75	
Juan Lopez	12/1/2023, 18:22:56		completado	3.25	
Pepito Lopez	12/1/2023, 17:07:28		verificado	2	
Pepito Lopez	7/1/2023, 22:07:29		verificado	3.5	
Juan Lopez	6/1/2023, 09:38:18		denegado	3	
Juan Lopez	6/1/2023, 09:27:59		completado	2.2	
Pepito Lopez	6/1/2023, 09:25:58		completado	2.5	

Fig. 13: Página para visualizar los pedidos generados por los clientes

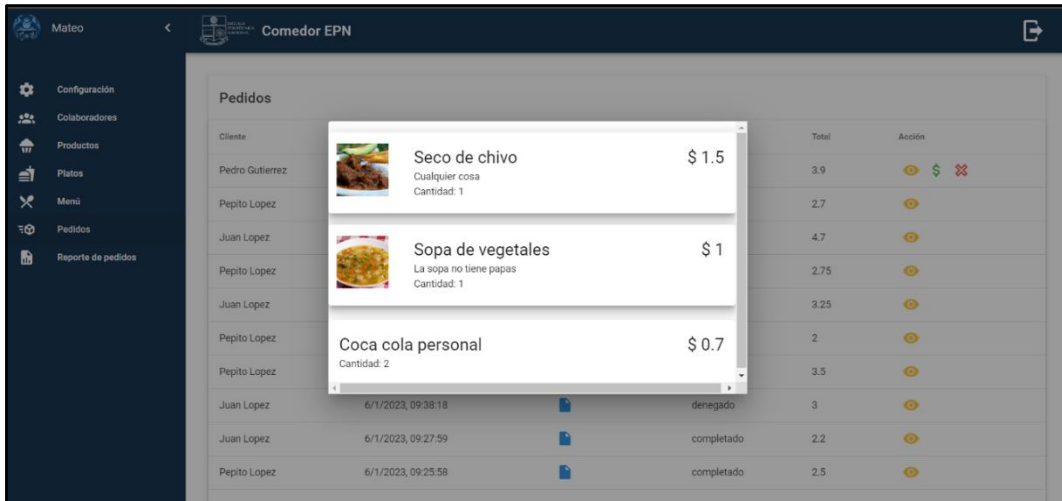


Fig. 14: Página para visualizar el detalle del pedido generado por el cliente

Consumir varios *endpoints* para asignar el pedido al cocinero

Para la asignación de un nuevo pedido generado, el administrador primero valida el pedido este proceso se lo explico mejor en el anterior requerimiento desarrollado. Una vez se valide el pedido el administrador verifica a los usuarios y verifica que cocinero se encuentra con menos pedidos a cargo para asignarle el pedido y este empiece a prepararlo. En la **Fig. 15** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

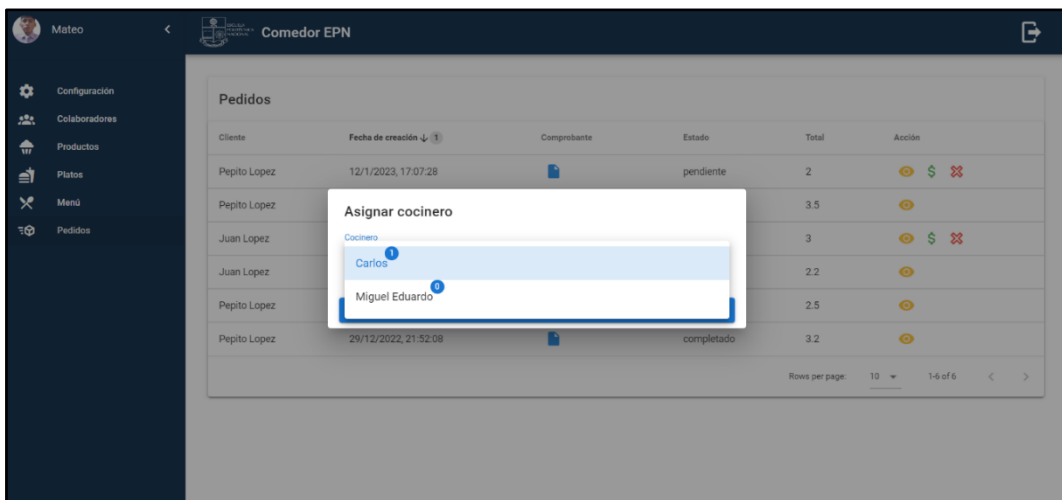
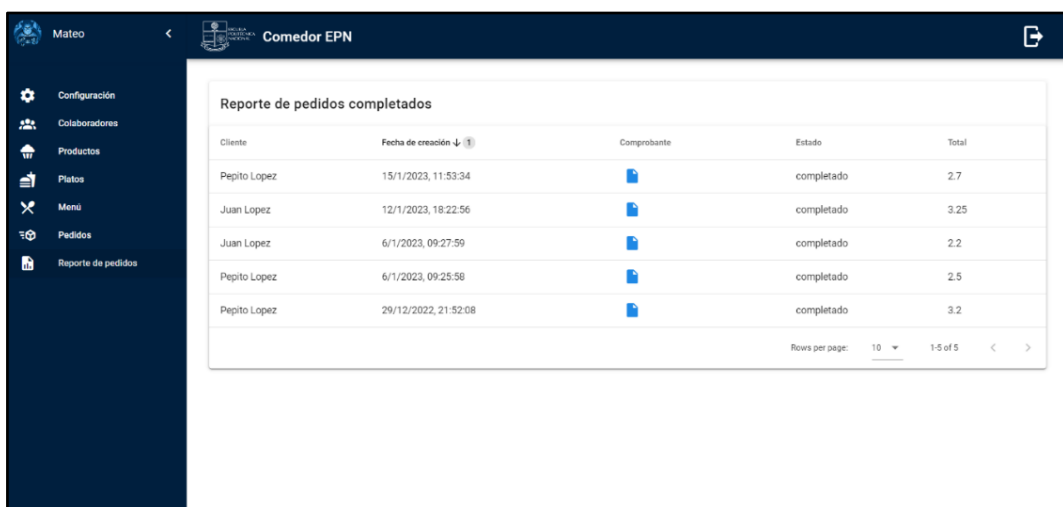







Fig. 15: Página para asignar un pedido a un cocinero

Consumir varios *endpoints* para visualizar reporte de pedidos completados

El módulo reporte de pedidos esta solo disponible para el usuario con perfil administrador en el cual se ha implementado una tabla en la que se presenta una vista de los pedidos completados la cual esta descrita por el nombre del cliente, la fecha y hora de creación del pedido, un botón que redirecciona a la visualización del comprobante de pago, el estado de cada uno de los pedidos el cual es completado y el valor de dicho pedido. En la **Fig. 16** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.



Cliente	Fecha de creación ↓	Comprobante	Estado	Total
Pepito Lopez	15/1/2023, 11:53:34		completado	2.7
Juan Lopez	12/1/2023, 18:22:56		completado	3.25
Juan Lopez	6/1/2023, 09:27:59		completado	2.2
Pepito Lopez	6/1/2023, 09:25:58		completado	2.5
Pepito Lopez	29/12/2022, 21:52:08		completado	3.2

Rows per page: 10 1-5 of 5 < >

Fig. 16: Página para visualizar el reporte de los pedidos completados

3.3 Sprint 2. Resultado de las interfaces implementadas para el usuario con perfil cliente

Las tareas que implican el cumplimiento del Sprint 2 son las siguientes:

- Consumir varios *endpoints* para registrarse como cliente.
- Consumir varios *endpoints* para Iniciar sesión, cerrar sesión y recuperar contraseña
- Consumir varios *endpoints* para modificar perfil de usuario
- Consumir varios *endpoints* para gestionar pedido
- Consumir varios *endpoints* para cargar el archivo de confirmación de pago
- Consumir varios *endpoints* para visualizar el estado del pedido

Consumir varios *endpoints* para registrarse como cliente

Para que el usuario con perfil cliente tenga acceso al sistema *web* lo primero que tiene que realizar es registrarse mediante el ingreso de los datos pedidos en el formulario de registro, y cuyos datos solicitados son: nombre, número de teléfono, foto de perfil, correo electrónico y contraseña. En la **Fig. 17** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

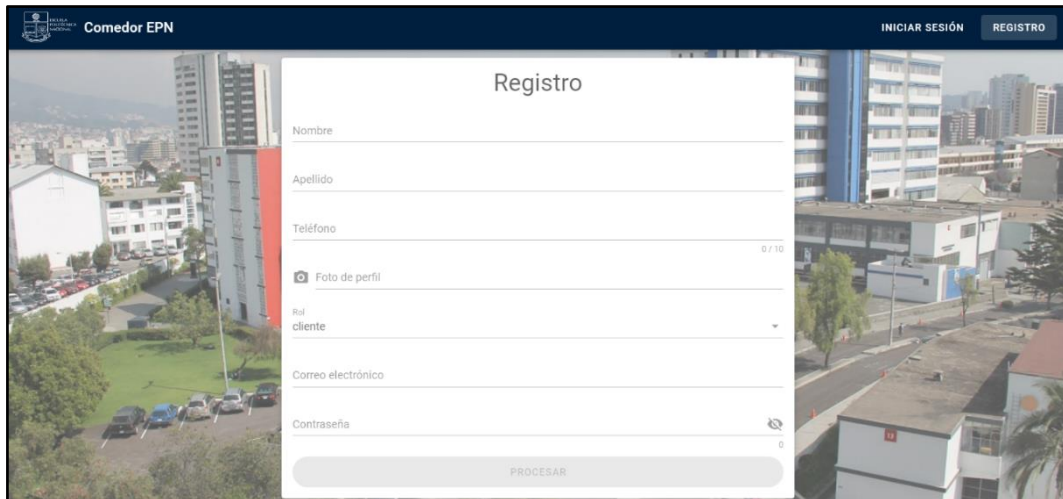


Fig. 17: Página para registro del cliente

Consumir varios *endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña

Dentro del sistema *web* para iniciar sesión, cerrar sesión y recuperar contraseña se lo realiza mediante un formulario en el caso del inicio de sesión se debe llenar los campos de correo electrónico y contraseña con las credenciales registradas en el formulario de registró en el cual se ingresa datos reales del usuario este es el caso del usuario cliente, ya que para el administrador los datos se encuentran establecidos y mientras que para el usuario cocinero las credenciales genera el administrador, tanto correo electrónico como contraseña. En la **Fig. 18** se presenta las interfaces anteriormente mencionadas, luego de haber concluido la fase de desarrollo y de pruebas.

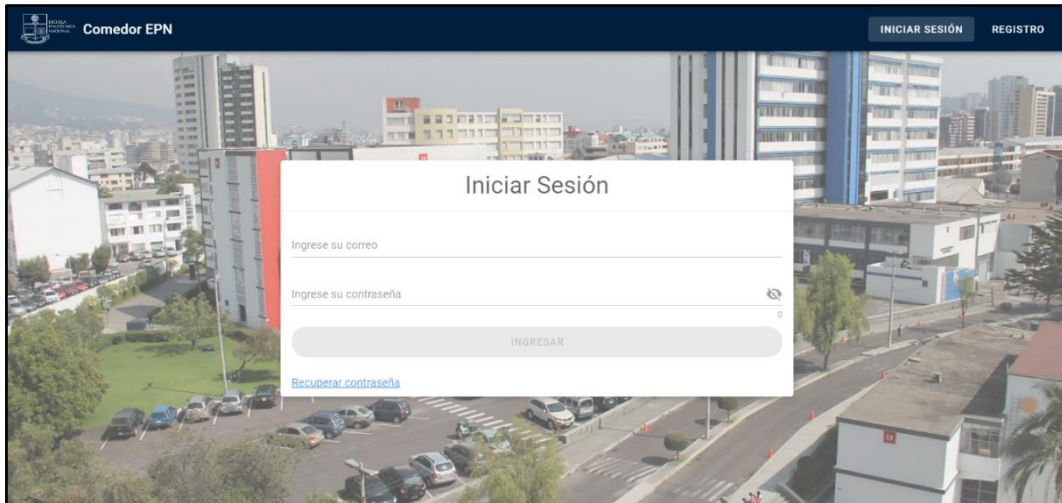


Fig. 18: Página de inicio de sesión en el sistema *web*

Consumir varios *endpoints* para modificar perfil de usuario

Dentro del sistema *web* el usuario con perfil cliente observará su foto de perfil y nombre en el cual al hacer clic en el avatar lo llevará a la vista de su perfil en la cual tiene la posibilidad de modificar algunos de sus datos personales, entre la información que puede cambiar es el nombre, foto de perfil y el número de teléfono y para confirmar los cambios debe guardarlos antes de salir del módulo perfil. En la **Fig. 19** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

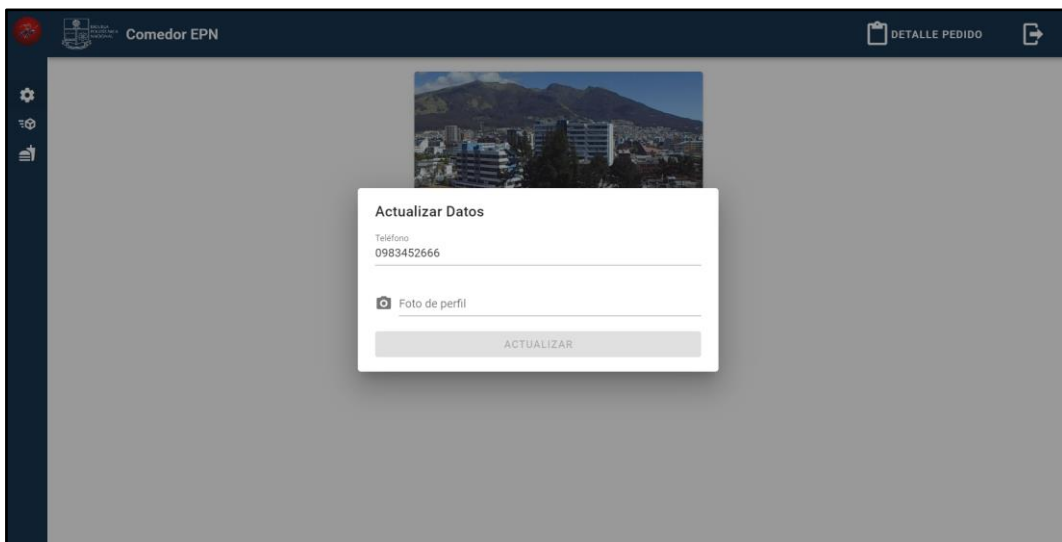


Fig. 19: Página para modificar los datos del perfil de usuario

Consumir varios *endpoints* para gestionar pedido

Dentro del sistema se maneja la vista en la que el usuario gestiona el pedido. El usuario cliente puede ir agregando a su pedido que nomas va a consumir, en dicha vista se presenta los platos y puede elegir cuantos platos o productos desea, o en el caso que se agoten, también se muestra que se terminó el producto, y los más importante el valor de cada uno de los platos, bebidas y snacks que conforman el menú del día. En la **Fig. 20** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

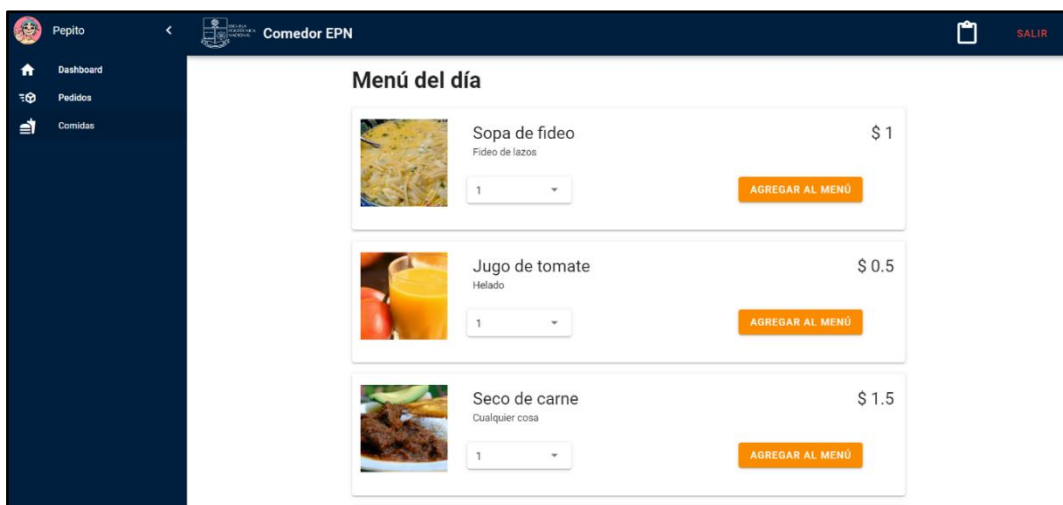


Fig. 20: Página para generar pedido de almuerzo

Consumir varios *endpoints* para cargar el archivo de confirmación de pago

Para realizar el pago del pedido, el usuario con perfil cliente aplasta el botón con el icono de 'tablita' que se encuentra en la barra de navegación del pedido, a continuación, se le redirecciona a una vista en la cual se le detalla el pedido realizado hasta al momento y para confirmar el pedido primero se debe cargar el comprobante sino el botón 'pagar' no se activa. En la **Fig. 21** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

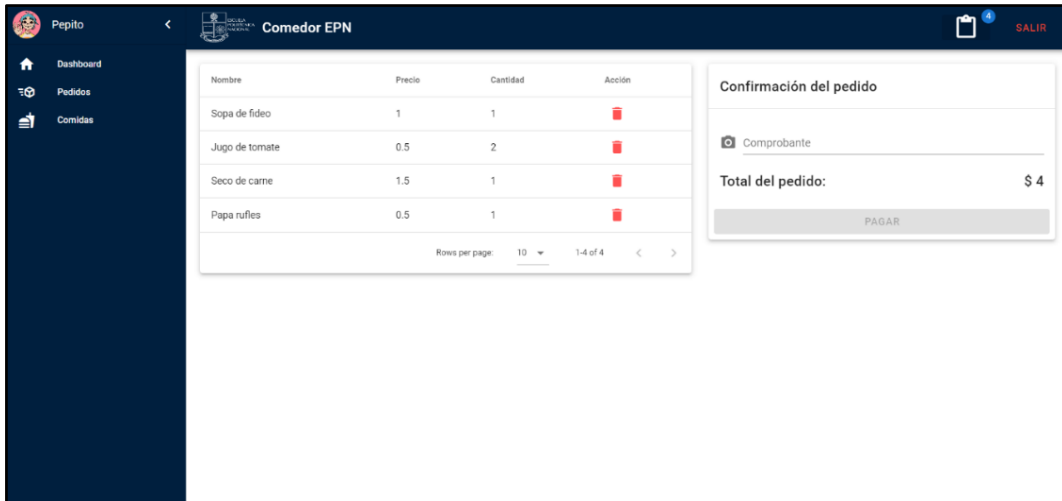


Fig. 21: Página para confirmar y cargar el comprobante de pago del pedido

Consumir varios *endpoints* para visualizar el estado del pedido

Además, en otra vista se visualiza el estado en el que está su pedido, que pueden estar en el estado de validación, proceso, y completado. En la **Fig. 22** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

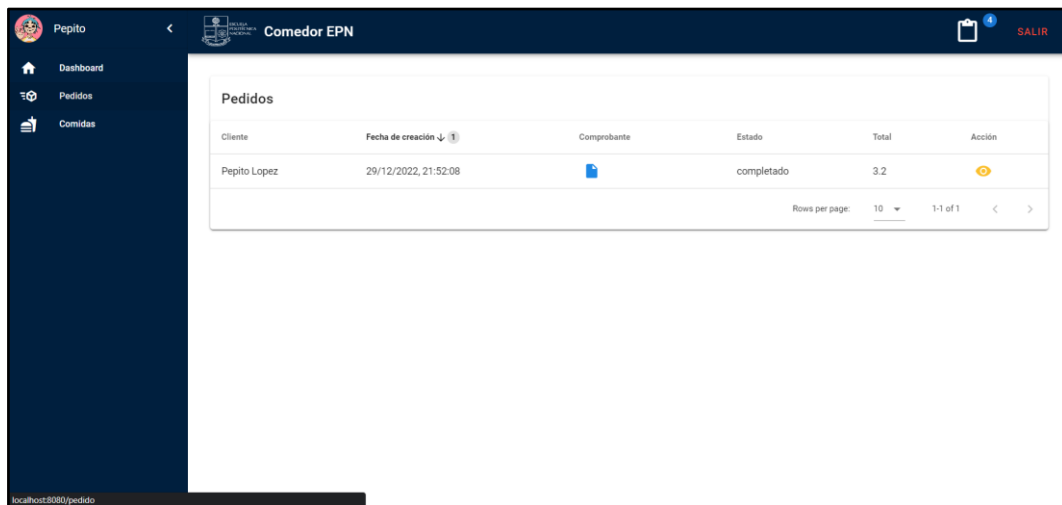


Fig. 22: Página para visualizar pedidos generados y su estado de proceso

3.4 Sprint 3. Resultado de las interfaces implementadas para el usuario con perfil cocinero

Las tareas que implican el cumplimiento del Sprint 3 son las siguientes:

- Consumir varios *endpoints* para Iniciar sesión, cerrar sesión y recuperar contraseña
- Consumir varios *endpoints* para modificar perfil de usuario

- Consumir varios *endpoints* para visualizar pedido asignado
- Consumir varios *endpoints* para cambiar el estado del pedido

Consumir varios *endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña

Dentro del sistema *web* para iniciar sesión, cerrar sesión y recuperar contraseña se lo realiza mediante un formulario en el caso del inicio de sesión se debe llenar los campos de correo electrónico y contraseña con las credenciales registradas en el formulario de registró en el cual se ingresa datos reales del usuario este es el caso del usuario administrador. Para recuperar la contraseña se lo realiza mediante él envió de un correo al email proporcionado y asi poder restablecer la contraseña del usuario. En la **Fig. 23** se presenta las interfaces anteriormente mencionadas, luego de haber concluido la fase de desarrollo y de pruebas.

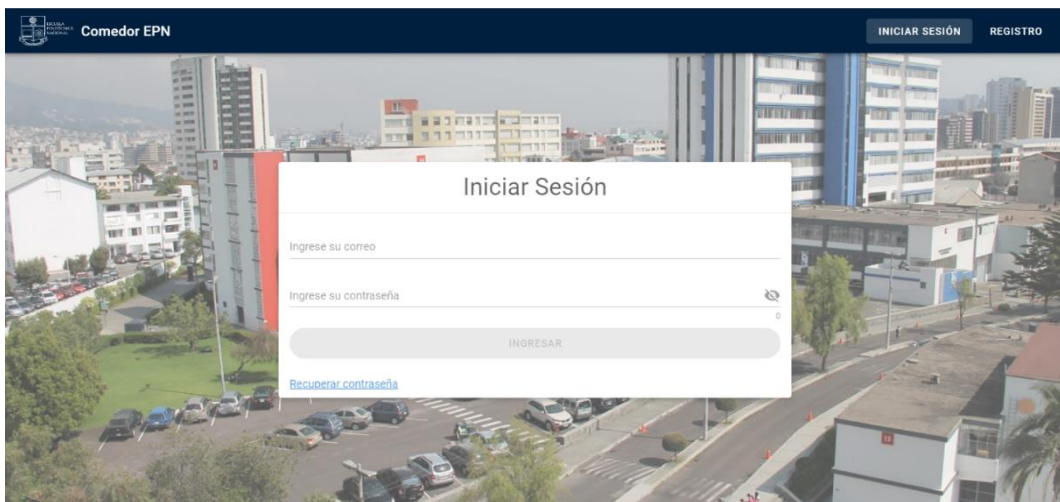


Fig. 23: Página de inicio de sesión en el sistema *web*

Consumir varios *endpoints* para modificar perfil de usuario

Dentro del sistema *web* el usuario con perfil cliente observara su foto de perfil y nombre en el cual al hacer clic en el avatar lo llevara a la vista de su perfil en la cual tiene la posibilidad de modificar algunos de sus datos personales, entre la información que puede cambiar es el nombre, foto de perfil y el número de teléfono y para confirmar los cambios debo guardarlos antes de salir del módulo perfil. En la **Fig. 24** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

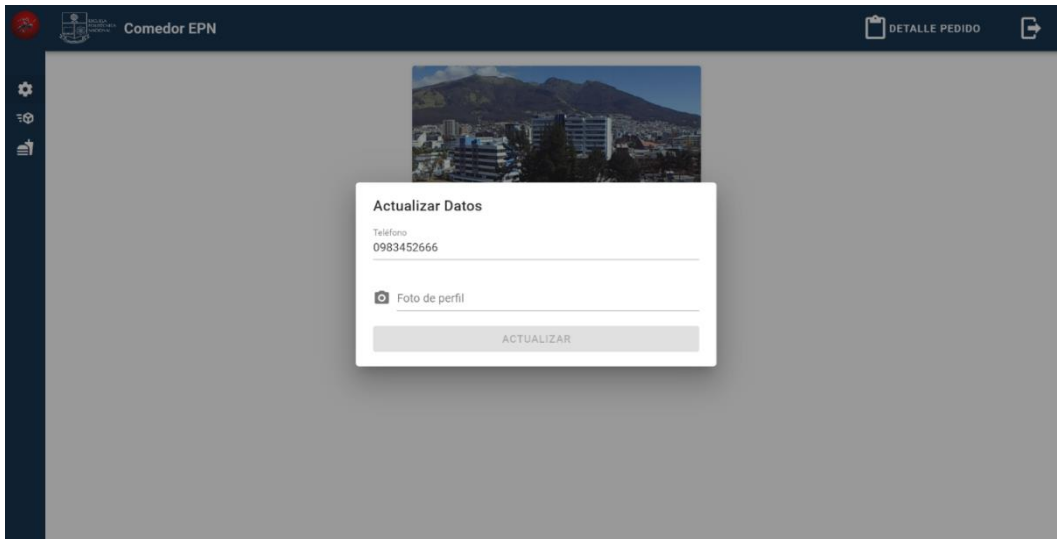


Fig. 24: Página para modificar el perfil de usuario

Consumir varios *endpoints* para visualizar los pedidos asignados

Para el módulo de pedidos se ha implementado una **TABLA** que permite visualizar los pedidos asignados al usuario con perfil cocinero con los siguientes datos: nombre del cliente, comprobante de pago, estado del pedido, su valor, y al final la sección de acción en el cual se implementó 2 botones el primer botón con icono de ojo es para visualizar el detalle del pedido que el usuario con perfil cocinero debe preparar. En la **Fig. 25** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.

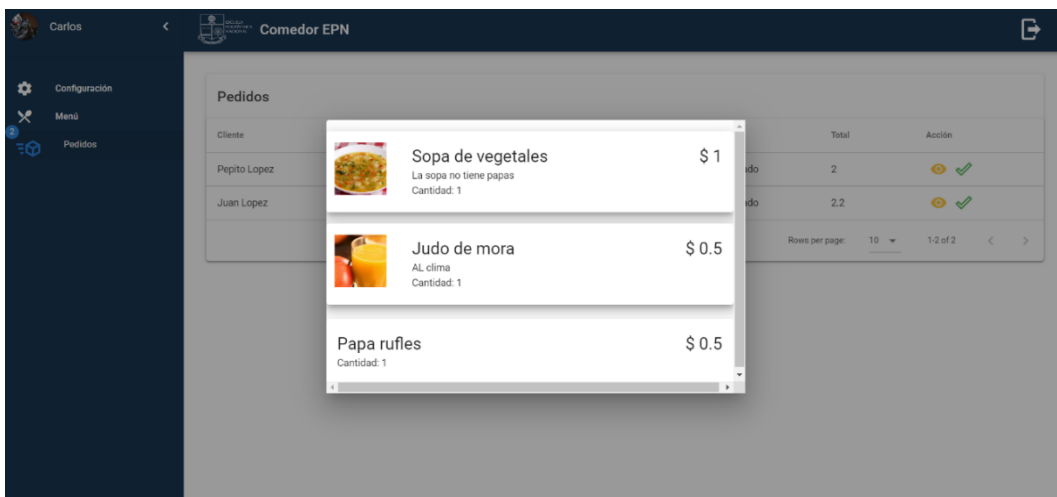
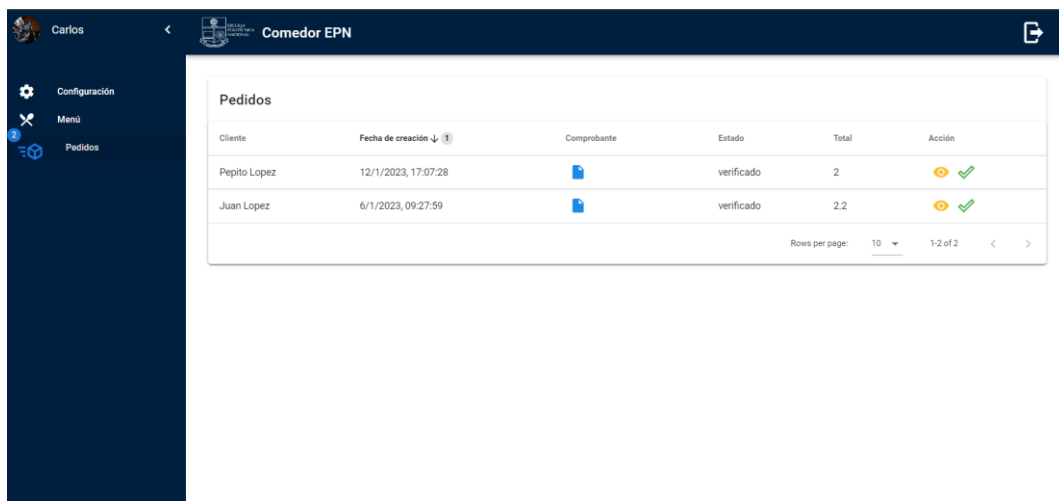


Fig. 25: Página para visualizar los pedidos asignados

Consumir varios en *endpoints* para cambiar el estado de los pedidos

Para el módulo de pedidos se ha implementado en la **TABLA** de descripción de pedido 2 botones los cuales el usuario cocinero al darle clic al segundo botón con icono de visto el pedido asignado cambia de estado el cual es completado y así dar por terminado la preparación del pedido, además que cuando damos clic en el botón dicho pedido asignado al cocinero desaparece y deja a la **TABLA** limpia sin pedidos por realizar. En la **Fig. 26** se presentan las interfaces anteriormente descritas, luego de haber concluido la fase de desarrollo y de pruebas.









Cliente	Fecha de creación ↓	Comprobante	Estado	Total	Acción
Pepito Lopez	12/1/2023, 17:07:28		verificado	2	 
Juan Lopez	6/1/2023, 09:27:59		verificado	2.2	 

Fig. 26: Página para cambiar el estado de los pedidos

3.5 Sprint 4. Pruebas en el sistema *web*

Terminando la fase de codificación de cada uno de los componentes visuales por parte del sistema *web*, se presentan las actividades para el Sprint 4 que son las siguientes:

- Ejecución de las pruebas unitarias y resultados.
- Ejecución de las pruebas de compatibilidad y resultados.
- Ejecución de las pruebas de aceptación y resultados.

Ejecución de las pruebas unitarias resultados

Las pruebas unitarias tienen como fin verificar y comprobar que cada uno de los componentes del *frontend* se encuentren correctamente funcionales. Y para realizar cada una de las pruebas unitarias se utiliza el mismo *Framework* de desarrollo, ya que este cuenta con un entorno para realizar pruebas con la librería Test-utils la cual permite la

renderización y depuración del código fuente y comprobación de cada uno de los componentes.

A continuación, en la **Fig. 27** se presenta la implementación y ejecución de la prueba unitaria para el componente Iniciar Sesión, junto a sus resultados en la **Fig. 28**. El detalle completo de las demás pruebas y resultados se visualizan en el **ANEXO II** del presente documento.

```
tests > unit > login.spec.ts > describe("LoginView.vue"
1  import { mount } from "@vue/test-utils"
2  import CamposLogin from "@/views/Login/
3
4  describe("LoginView.vue", () => {
5    it("Test unit of Login", () => {
6      const wrapper = mount(CamposLogin, {
7        propsData: {
8          usuario: {
9            email: 'mat@gmail',
10           password: '87654321'
11         }
12       }
13     });
14     expect(wrapper.html()).toContain('v
15   });
16 });
```

Fig. 27: Fragmento de código para iniciar sesión en el sistema web

```
PASS tests/unit/login.spec.ts
  LoginView.vue
    ✓ Test unit of Login (144 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        5.984 s, estimated 6 s
Ran all test suites.
```

Fig. 28: Resultado de la prueba

Una vez completada la etapa de pruebas unitarias y verificando los resultados que se han conseguido, se determina que los módulos evaluados del sistema *web* funcionan sin presentar problemas, a nivel de código y validaciones respectivamente.

Ejecución de pruebas de compatibilidad y resultados

Las pruebas de compatibilidad ayudan a verificar si una aplicación *web* es compatible y puede ejecutarse en varios navegadores. Por ello, en la **TABLA VIII** se presenta los navegadores que se han utilizado para comprobar si el contenido es adaptable a cualquier dispositivo. La descripción tanto de las ejecuciones como los resultados del resto de pruebas de compatibilidad se encuentran en el **ANEXO II** del actual documento.

TABLA VIII: Navegadores en los que se realizó las pruebas de compatibilidad

NOMBRE	VERSIÓN	OBSERVACIÓN
<i>Google Chrome</i>	109.0.5414.75	Completamente funcional
<i>Mozilla Firefox</i>	108.0.2	Completamente funcional
<i>Microsoft Edge</i>	109.0.1518.55	Completamente funcional

Ejecución de pruebas de aceptación y resultados

Las pruebas de aceptación son una de las etapas finales del proceso de pruebas de *software*, y es la parte más importante, ya que en esta se garantiza que el sistema *web* es intuitivo hacia el usuario final, para de tal forma navegue sin inconvenientes, satisfaga las necesidades, y cumpla con las exceptivas esperadas. En base a los requerimientos inicialmente solicitados para el desarrollo del sistema, en la **TABLA IX** se muestra el detalle de la primera prueba de aceptación juntamente con su resultado, la descripción de las demás pruebas se encuentra en el **ANEXO II** del actual documento junto con sus resultados.

TABLA IX: Prueba de Aceptación N°1 - Consumir varios *endpoints* para visualizar pagina informativa de la Cafeteria

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0001	Identificador historia de Usuario: HU001
Nombre: Consumir varios <i>endpoints</i> para visualizar la página informativa	
Descripción: El usuario cocinero, cliente, administrador podrá visualizar la información general de la Cafetería e interactuar con los siguientes módulos:	

<ul style="list-style-type: none"> • Iniciar Sesión • Registrarse
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Dirigirse a la URL del sistema <i>web</i> en el navegador. 2. Seleccionar el módulo “Iniciar Sesión”, en el panel superior para visualizar el formulario de acceso al sistema digitando sus respectivas credenciales. 3. Seleccionar el módulo “Registrarse”, en el panel superior para visualizar el formulario de registro en el cual coloca datos reales para su registro como usuario cliente.
<p>Resultado deseado:</p> <p>El sistema <i>web</i> permite visualizar información general de la Cafetería</p>
<p>Evaluación de la prueba:</p> <p>Se verificar el resultado previsto. Conformidad del cliente 100%.</p>

3.6 Sprint 5. Despliegue del sistema *web*

Terminando la fase de codificación de cada uno de los componentes visuales por parte del sistema *web*, se presentan las actividades para el Sprint 5 que son las siguientes:

Despliegue del sistema *web* en Vercel

En esta sección se describe el proceso que se ha llevado a cabo para el despliegue a producción del sistema *web* utilizando la plataforma Vercel. El detalle de esta sección se encuentra en el **ANEXO II** del actual documento. La etapa inicial de este proceso se observa en la **Fig. 29** correspondiente a la creación de un nuevo proyecto en Vercel para alojar el sistema *web*, por otro lado, la **Fig. 30** presenta el despliegue apropiado del sistema *web* bajo un dominio proporcionado por la plataforma.

<https://sistema-web-cafeteria-eqn.vercel.app/>

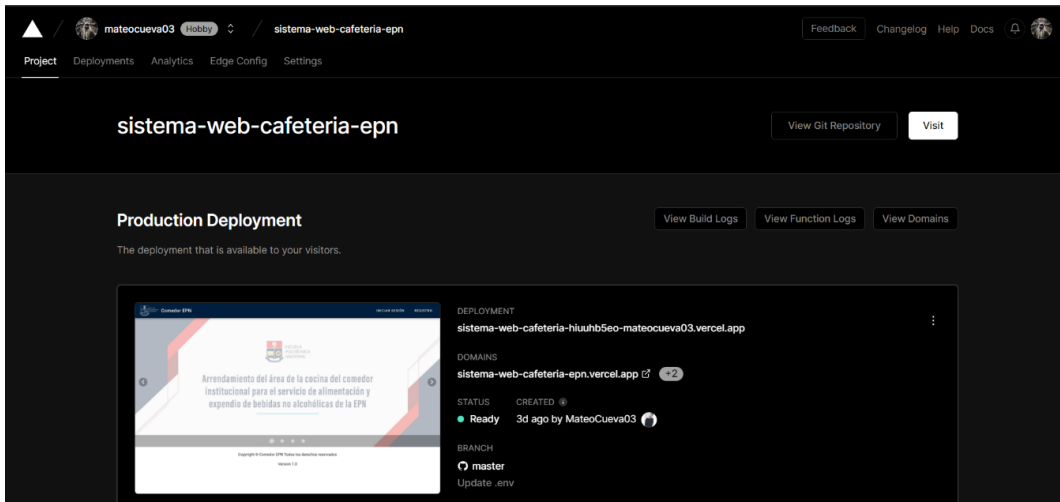


Fig. 29: Creacion de nuevo proyecto en Vercel



Fig. 30: Despliegue del *frontend* en Vercel

4 CONCLUSIONES

Ahora se muestran las conclusiones que se obtuvieron en el del desarrollo del trabajo de integración curricular.

- El *frontend* del sistema *web* cumple con el alcance y los objetivos anteriormente establecidos para el desarrollo del proyecto, logrando así reducir la aglomeración que se genera en las horas de almuerzo con la ayuda del *frontend* del sistema *web* desarrollado.
- Establecer los requerimientos en la parte inicial del proyecto aportan una gran ayuda para el desarrollo del *frontend*, de tal manera que se tiene una investigación y posterior a ello seleccionar las herramientas adecuadas para el desarrollo de cada uno de los requerimientos.
- La metodología ágil *Scrum* que se ha utilizado en la fase de desarrollo del presente trabajo de integración curricular aporta a que el sistema pueda ser entregado y probado en las diferentes etapas de entrega de avances que se han planificado.
- Durante el período de codificación la utilización de las herramientas y bibliotecas para el desarrollo del *frontend* del sistema *web*, tienen gran importancia porque así se ha obtenido un buen resultado en cuanto al funcionamiento de este.
- Finalizada cada una de las etapas de codificación y pruebas del sistema *web* se pone en marcha el despliegue a producción mediante la herramienta Vercel, para de esta manera el público en general tenga acceso al *frontend*.

5 RECOMENDACIONES

Ahora se muestran las recomendaciones que se obtuvieron en el del desarrollo del trabajo de integración curricular.

- Se recomienda que el *frontend* del sistema *web* que se ha desarrollado en el presente trabajo de integración curricular sea robustecido con nuevas funcionalidades, en caso de que lo solicite el personal administrativo de la Cafetería.
- Se recomienda capacitar a todos los posibles usuarios que pueden acceder a la aplicación por el lado del cliente, para que así puedan generar peticiones de manera correcta.
- Como es conocimiento de todos los sistemas de *software* son los más vulnerables, por tal motivo se recomienda implementar medidas de seguridad para proteger los datos que manejan dentro del sistema *web*.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] SoftwarePara, «SoftwarePara,» enero 2022. [En línea]. Available: <https://softwarepara.net/restaurantes/>. [Último acceso: 11 noviembre 2022].
- [2] Take it Easy, «Take it Easy,» 23 agosto 2022. [En línea]. Available: https://play.google.com/store/apps/details?id=com.takeiteasy&hl=es_EC&gl=US. [Último acceso: 11 noviembre 2022].
- [3] G. Sánchez, «Análisis simulación y propuesta de mejora del proceso de servicio de alimentos de la cafetería EPN,» Escuela Politécnica Nacional, Quito, 2015.
- [4] Escuela Politécnica Nacional, «CONVOCATORIA PARA EL SERVICIO DE ARRENDAMIENTO DE LOS LOCALES COMEDOR INSTITUCIONAL Y CAFETERÍA,» 2015. [En línea]. Available: <https://www.epn.edu.ec/convocatoria-para-el-servicio-de-arrendamiento-de-los-locales-comedor-institucional-y-cafeteria/>. [Último acceso: 25 06 2022].
- [5] F. Coelho, «Sgificados.com,» 17 mayo 2019. [En línea]. Available: <https://www.significados.com/metodologia/>. [Último acceso: 12 noviembre 2022].
- [6] Red Hat, «Red Hat,» 19 julio 2022. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology..> [Último acceso: 12 noviembre 2022].
- [7] Geoinnova, «Geoinnova,» 26 noviembre 2020. [En línea]. Available: <https://geoinnova.org/blog-territorio/html-css-y-javascript-lenguajes-para-el-desarrollo-de-paginas-web/>. [Último acceso: 12 noviembre 2022].
- [8] J. M. Alarcon, «Campusmvp,» 20 febrero 2020. [En línea]. Available: <https://www.campusmvp.es/recursos/post/typescript-contra-javascript-cual-deberias-utilizar.aspx>. [Último acceso: 16 noviembre 2022].
- [9] G. Munte, «rockcontent,» 08 enero 2020. [En línea]. Available: <https://rockcontent.com/es/blog/framework/>. [Último acceso: 16 noviembre 2022].
- [1] F. García, «ARSYS,» 27 mayo 2020. [En línea]. Available: <https://www.arsys.es/blog/vuejs>. [Último acceso: 29 noviembre 2022].
- [1] Red Hat, «Red Hat,» 02 junio 2022. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>. [Último acceso: 16 noviembre 2022].
- [1] D. Bernal, «profile,» 04 octubre 2021. [En línea]. Available: https://profile.es/blog/apis-front-end/#Petición_a_una_API_directamente_sin_Front_End. [Último acceso: 16 noviembre 2022].

- [1] M. Martinez, «profile,» 07 septiembre 2021. [En línea]. Available:
[3] <https://profile.es/blog/que-es-el-testing-de-software/>. [Último acceso: 16 noviembre 2022].
- [1] L. Codina, «Codina,» 08 septiembre 2021. [En línea]. Available:
[4] <https://www.lluiscodina.com/estudios-de-caso/>. [Último acceso: 17 noviembre 2022].
- [1] Santander Universidades, «Santander,» 21 diciembre 2020. [En línea]. Available:
[5] <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 17 noviembre 2022].
- [1] L. Goncalves, «ADAPT,» 27 junio 2022. [En línea]. Available:
[6] <https://adaptmethodology.com/es/que-es-la-metodologia-scrum/>. [Último acceso: 17 noviembre 2022].
- [1] C. Ramo, «CristinaRamos,» 20 febrero 2017. [En línea]. Available:
[7] <https://cristinamosvega.com/z-los-artefactos-scrum/>. [Último acceso: 17 noviembre 2022].
- [1] T. Asana, «ASANA,» 14 junio 2022. [En línea]. Available:
[8] <https://asana.com/es/resources/requirements-gathering>. [Último acceso: 17 noviembre 2022].
- [1] T. Asana, «ASANA,» 21 enero 2022. [En línea]. Available:
[9] <https://asana.com/es/resources/user-stories>. [Último acceso: 17 noviembre 2022].
- [2] D. Molina, «IEBS,» 09 diciembre 2021. [En línea]. Available:
[0] <https://www.iebschool.com/blog/que-es-un-product-backlog-y-como-hacer-uno-guia-scrum-agile-scrum/>. [Último acceso: 17 noviembre 2022].
- [2] R. Ollarves, «interfell,» 2022. [En línea]. Available:
[1] <https://blog.interfell.com/artefactos-de-la-metodolog%C3%ADa-scrum>. [Último acceso: 16 noviembre 2022].
- [2] ESIC, «ESIC,» noviembre 2021. [En línea]. Available:
[2] <https://www.esic.edu/rethink/tecnologia/que-es-el-diseno-de-interfaces-web>. [Último acceso: 17 noviembre 2022].
- [2] H. Rodriguez, «crehana,» 30 enero 2022. [En línea]. Available:
[3] <https://www.crehana.com/blog/estilo-vida/que-es-figma/>. [Último acceso: 15 noviembre 2022].
- [2] H. Cervantes, «SG,» 2022. [En línea]. Available: <https://sg.com.mx/revista/29/diseo-la-arquitectura>. [Último acceso: 17 noviembre 2022].
- [2] MDN Plus, «developer,» 10 noviembre 2022. [En línea]. Available:
[5] <https://developer.mozilla.org/es/docs/Glossary/MVC>. [Último acceso: 15 noviembre 2022].
- [2] F. Flores, «OpenWebinars,» 22 julio 2022. [En línea]. Available:
[6] <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas->

7 ANEXOS

A continuación, se muestra la división de los Anexos utilizados para el desarrollo del sistema *web*.

ANEXO I. Turnitin porcentaje máximo 12%.

ANEXO II. Manual técnico

ANEXO III. Manual de usuario

ANEXO IV. Manual de instalación

ANEXO I

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



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

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 1 de marzo de 2023

De mi consideración:

Yo, Franco Rocha Yadira Guissela, en calidad de Director del Trabajo de Integración Curricular titulado **DESARROLLO DE UN FRONTEND asociado DESARROLLO DE SISTEMA WEB Y APLICACIÓN MÓVIL PARA COMANDAS EN LA CAFETERÍA EPN** elaborado por el estudiante **Cueva Ramos Mateo Nicolás** de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito en las secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 4%.

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,



Franco Rocha Yadira Guissela
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos

ANEXO II

Recopilación de requerimientos

La **TABLA X** exhibe los requerimientos que se recopilieron al inicio del proyecto en donde nos indica lo que solicita el *Product Owner*

TABLA X: Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTO		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
FRONTEND	RR002	El usuario con perfil administrador, cliente y cocinero necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Iniciar Sesión • Cerrar sesión • Recuperar contraseña
	RR003	El usuario con perfil administrador y cliente necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros
	RR004	El usuario con perfil administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar el pedido y la factura cancelada
	RR005	El usuario con perfil administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Asignar el pedido al cocinero
	RR006	Consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Agregar platos del menú del almuerzo y snacks
	RR007	El usuario con perfil administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Generar el menú de los almuerzos
	RR008	El usuario con perfil cliente y cocinero necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Modificar su perfil
	RR009	El usuario con perfil cliente necesita consumir varios <i>endpoints</i> para:

		<ul style="list-style-type: none"> • Generar pedidos
	RR010	El usuario con perfil cliente necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Cargar el comprobante de pago del pedido
	RR011	El usuario con perfil cliente necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar el estado del pedido
	RR012	El usuario con perfil cocinero necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Visualizar los pedidos asignados
	RR013	El usuario con perfil cocinero necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> • Cambiar el estado de los pedidos

Historias de Usuario

Una vez que se completa la recolección de los requerimientos, se debe realizar las Historias de Usuario para el *frontend*. Ahora se presenta todas las historias de usuario que se crearon a partir de cada uno de los requerimientos que fueron recolectados anteriormente, dichas historias empiezan desde la **TABLA XI** a la **TABLA XXII**.

TABLA XI: Historia de usuario 02 – Consumir varios *endpoints* para Iniciar sesión, cerrar sesión y recuperar contraseña

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: Administrador, cliente y cocinero
Nombre Historia: Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: El usuario cocinero, cliente, administrador tiene que consumir varios <i>endpoints</i> para poder: <ul style="list-style-type: none"> • Iniciar sesión 	

- Cerrar sesión
- Recuperar contraseña

Para la recuperación de contraseña el usuario debe dirigirse al módulo de olvide mi contraseña, en el cual debe escribir su correo electrónico registrado para así recibir un código que tendrá que digitar luego de proporcionar el correo, de tal manera que pueda cambiar su contraseña.

Observación:

Cuando el usuario cocinero, cliente, administrador ingresen usuario y contraseña de manera incorrecta se presenta un mensaje de error, y si desean restablecer su contraseña se lo realiza enviando un correo al email registrado.

TABLA XII: Historia de usuario 03 – Consumir varios *endpoints* para registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Administrador, cliente
Nombre Historia: Consumir varios <i>endpoints</i> para registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: Los usuarios administrador y cliente deben consumir varios <i>endpoints</i> para registrarse, además el usuario administrador es el encargado en generar un nuevo cocinero y desde su panel también otro usuario administrador.	
Observación: El cocinero puede acceder a su módulo con el correo proporcionado para el registro y con la contraseña que llegara a su correo, mientras que el administrador y el cliente acceden con los datos ingresados en el formulario de registro. Los datos proporcionados deben ser reales.	

TABLA XIII: Historia de usuario 04 – Consumir varios *endpoints* para para visualizar el pedido y la factura cancelada

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para visualizar el pedido y la factura pagada.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para visualizar a detalle el pedido generado y el comprobante de haber realizado el pago de este.	
Observación: Se debe verificar que el pago del pedido este correctamente realizado, de no ser asi el pedido será anulado.	

TABLA XIV: Historia de usuario 05 – Consumir varios *endpoints* para asignar el pedido al cocinero

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para asignar el pedido al cocinero	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para asignar el pedido a un cocinero que se encuentre libre o con un menor número de pedidos asignados.	
Observación: Se tiene que verificar que el cocinero este activo y desocupado al momento de asignarle el pedido.	

TABLA XV: Historia de usuario 06 – Consumir varios *endpoints* para agregar platos del menú del almuerzo y snacks

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para agregar platos del menú del almuerzo y snacks.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para crear el plato que se tiene disponible en el menú del día sea sopa, segundo o bebida, también tiene la posibilidad de agregar snacks.	
Observación: Para agregar tanto un plato como un snack se lo hace desde una vista diferente ya que son productos diferentes.	

TABLA XVI: Historia de usuario 07 – Consumir varios *endpoints* para generar el menú de los almuerzos

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para generar el menú de los almuerzos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Mateo Cueva	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para llenar el formulario del menú diario y seleccionar los platos disponibles para el menú, de tal manera que se cree el menú del día.	
Observación: El usuario con perfil administrador para generar el menú de almuerzo tiene que visualizar los platos disponibles, caso contrario no se puede crear un nuevo menú.	

TABLA XVII: Historia de usuario 08 – Consumir varios *endpoints* para modificar perfil

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: Administrador, cliente, cocinero
Nombre Historia: Consumir varios <i>endpoints</i> para modificar perfil.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Mateo Cueva	
<p>Descripción:</p> <p>El usuario administrador, cliente y cocinero, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para visualizar y actualizar perfil cuando el administrador, cliente y cocinero lo requiera. La lectura y actualización del perfil se realiza mediante un formulario donde se ingresa la siguiente información:</p> <ul style="list-style-type: none"> • Nuevo número de teléfono • Nueva foto de perfil • Nueva contraseña 	
<p>Observación:</p> <p>El usuario cliente puede consumir los <i>endpoints</i> para visualizar y actualizar el perfil a través de su ingreso al sistema <i>web</i> donde puede gestionar la actividad ya mencionada.</p>	

TABLA XVIII: Historia de usuario 09 – Consumir varios *endpoints* para generar pedidos

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: Cliente
Nombre Historia: Consumir varios <i>endpoints</i> para generar pedidos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Mateo Cueva	
<p>Descripción:</p> <p>El cliente, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para crear pedidos cuando el cliente lo requiera. La creación de un pedido se realiza por un formulario, se selecciona la siguiente información.</p> <ul style="list-style-type: none"> • Sopa 	

<ul style="list-style-type: none"> • Segundo • Bebida • Snacks
<p>Observación: El usuario cliente puede consumir el endpoint para crear un pedido a través de su ingreso al sistema <i>web</i> donde puede gestionar la actividad ya mencionada.</p>

TABLA XIX: Historia de usuario 10 – Consumir varios *endpoints* para confirmar y cargar el comprobante de pago del pedido

HISTORIA DE USUARIO	
Identificador (ID): HU010	Usuario: Cliente
Nombre Historia: Consumir varios <i>endpoints</i> para confirmar y cargar el comprobante de pago del pedido.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Mateo Cueva	
<p>Descripción: El cliente, en el sistema <i>web</i> puede consumir un <i>endpoints</i> para realizar la confirmación y el pago del pedido a través de un sistema en el que cargara el comprobante de pago.</p>	
<p>Observación: Si se carga un documento que no garantice el pago del pedido, dicho pedido no será aprobado para su realización.</p>	

TABLA XX: Historia de usuario 11 – Consumir varios *endpoints* para visualizar el estado del pedido

HISTORIA DE USUARIO	
Identificador (ID): HU011	Usuario: Cliente
Nombre Historia: Consumir varios <i>endpoints</i> visualizar el estado del pedido.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Mateo Cueva	

<p>Descripción:</p> <p>El cliente, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para visualizar el estado en el que se encuentra su pedido, lo que se presenta en su vista es si el pedido está en observación, en preparación o finalizado.</p>
<p>Observación:</p> <p>El usuario cliente puede consumir los <i>endpoints</i> para listar un pedido a través de su ingreso al sistema <i>web</i> donde puede gestionar la actividad ya mencionada.</p>

TABLA XXI: Historia de usuario 12 – Consumir varios *endpoints* para visualizar los pedidos asignados

HISTORIA DE USUARIO	
Identificador (ID): HU012	Usuario: Cocinero
Nombre Historia: Consumir varios <i>endpoints</i> para visualizar los pedidos asignados.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 3	
Responsable: Mateo Cueva	
<p>Descripción:</p> <p>El usuario cocinero, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para visualizar en su vista principal el pedido que se le ha sido asignado.</p>	
<p>Observación:</p> <p>Ninguna.</p>	

TABLA XXII: Historia de usuario 13 – Consumir varios *endpoints* para cambiar el estado de los pedidos

HISTORIA DE USUARIO	
Identificador (ID): HU013	Usuario: Cocinero
Nombre Historia: Consumir varios <i>endpoints</i> para cambiar el estado de los pedidos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 3	
Responsable: Mateo Cueva	

<p>Descripción:</p> <p>El usuario cocinero tiene que consumir varios <i>endpoints</i> para cambiar el estado del pedido, dichos estados que puede cambiar el usuario son: preparando, y pedido finalizado.</p>
<p>Observación:</p> <p>El usuario no tiene que tocar algún aparato para realizar el cambio de estado del pedido.</p>

Product Backlog

La **TABLA XXIII** exhibe la prioridad que se le asigna a cada uno de los requisitos o tareas que se implementa en el *frontend*. Los requerimientos se clasifican según las necesidades solicitadas por el dueño y también por la dificultad de su desarrollo.

TABLA XXIII: *Product Backlog.*

ELABORACIÓN DEL PRODUCT BACKLOG				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU002	Iniciar sesión, cerrar sesión y recuperar contraseña	1	Finalizado	Media
HU003	Registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros	1	Finalizado	Alta
HU004	Visualizar el pedido y la factura pagada	1	Finalizado	Alta
HU005	Asignar el pedido al cocinero	1	Finalizado	Media
HU006	Agregar platos del menú del almuerzo y snacks	1	Finalizado	Alta
HU007	Generar el menú de los almuerzos	1	Finalizado	Alta
HU008	Modificar su perfil	2	Finalizado	Media

HU009	Generar pedidos	2	Finalizado	Media
HU010	Confirmar y cargar el comprobante del pedido	2	Finalizado	Media
HU011	Visualizar el estado del pedido	2	Finalizado	Media
HU012	Visualizar los pedidos asignados	3	Finalizado	Media
HU013	Cambiar el estado de los pedidos	3	Finalizado	Alto

Sprint Backlog

La **TABLA XXIV** indica los cinco Sprints en los que se ha desarrollado el sistema *web*, listando las actividades y el tiempo determinado Para cumplir con los entregables que se han establecido con el dueño del producto.

TABLA XXIV: Sprint Backlog.

SPRINT BACKLOG						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB001	Bosquejo e implementación de interfaz del administrador y módulos a su cargo	Modulo información general	HU001	Consumir varios <i>endpoints</i> para visualizar página informativa	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para la vista de la página informativa. • Establecer la información que se va a visualizar en las diferentes secciones informativas. • Pruebas respectivas. 	
		Módulo de autenticación y recuperación de contraseña	HU002	Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña.	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para el inicio de sesión, cierre de sesión y cambio de contraseña. • Implementación de formularios para el inicio de sesión y el registro del usuario cliente 	

				<ul style="list-style-type: none"> Validación de los datos requeridos. Redireccionamiento hacia los módulos asignados. Pruebas respectivas. 		
		Módulo generación de perfiles, para administradores y cocineros	HU003	Consumir varios <i>endpoints</i> para registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros	<ul style="list-style-type: none"> Diseño y codificación del prototipo para crear perfiles de los cocineros. Obtención de las credenciales requeridas. Pruebas respectivas. 	
		Módulo visualización del pedido generado y su respectivo pago	HU004	Consumir varios <i>endpoints</i> para visualizar el pedido y la factura cancelada	<ul style="list-style-type: none"> Diseño y codificación del prototipo para visualizar el pedido y la factura cancelada. Validar los datos recibidos. Pruebas respectivas. 	
		Módulo asignación del pedido para el	HU005	Consumir varios <i>endpoints</i> para asignar el pedido al cocinero	<ul style="list-style-type: none"> Diseño y codificación del prototipo para asignar el pedido al cocinero. Pruebas respectivas. 	

		usuario cocinero				
		Módulo agregar platos disponibles del menú diario y snacks	HU006	Consumir varios <i>endpoints</i> para agregar platos del menú del almuerzo y snacks	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para gestionar el menú de los almuerzos • Pruebas respectivas. 	
		Módulo generar el menú diario de la Cafetería	HU007	Consumir varios <i>endpoints</i> para generar el menú de los almuerzos	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para gestionar el menú de los almuerzos • Implementación de formularios para agregar menú diario. • Pruebas respectivas. 	
SB002	Bosquejo e implementación de interfaz del cliente y módulos a su cargo	Modulo información general	HU001	Consumir varios <i>endpoints</i> para visualizar página informativa	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para la vista de la página informativa. • Establecer la información que se va a visualizar en las diferentes secciones informativas. • Pruebas respectivas. 	
		Módulo de autenticación y recuperación de contraseña	HU002	Consumir varios <i>endpoints</i> para iniciar sesión, cerrar	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para el inicio de sesión, cierre de sesión y cambio de contraseña. 	

				sesión y recuperar contraseña.	<ul style="list-style-type: none"> • Implementación de formularios para el inicio de sesión y el registro del usuario cliente • Validación de los datos requeridos. • Redireccionamiento hacia los módulos asignados. • Pruebas respectivas. 	
		Módulo actualizar perfil	HU008	Consumir varios <i>endpoints</i> para modificar perfil	<ul style="list-style-type: none"> • Diseño y codificación los prototipos para visualizar y editar el perfil de usuario. • Validación de los datos requeridos. • Pruebas respectivas. 	
		Módulo de registro, para los perfiles de usuario clientes	HU003	Consumir varios <i>endpoints</i> para registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros	<ul style="list-style-type: none"> • Diseño y codificación los prototipos para el usuario con perfil cliente pueda registrarse. • Validación de los datos requeridos. • Pruebas respectivas. 	
		Módulo crear un nuevo pedido	HU009	Consumir varios <i>endpoints</i> para generar pedidos.	<ul style="list-style-type: none"> • Diseño y codificación los prototipos para crear un nuevo pedido, por medio de un formulario de datos. • Validación de los datos requeridos. • Pruebas respectivas. 	

			HU010	Consumir varios <i>endpoints</i> para confirmar y cargar el comprobante de pago del pedido	•	
		Módulo – Comentarios y/o sugerencias	HU011	Consumir varios <i>endpoints</i> visualizar el estado del pedido.	<ul style="list-style-type: none"> • Consumir <i>endpoints</i> para visualizar los comentarios, sugerencias y calificación de los servicios. • Diseño de los prototipos. • Codificar los prototipos. • Pruebas respectivas. 	
SB003	Bosquejo e implementación de interfaz del cocinero y módulos a su cargo	Modulo información general	HU001	Consumir varios <i>endpoints</i> para visualizar página informativa	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para la vista de la página informativa. • Establecer la información que se va a visualizar en las diferentes secciones informativas. • Pruebas respectivas. 	

		Módulo de autenticación y recuperación de contraseña	HU002	Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña.	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para el inicio de sesión, cierre de sesión y cambio de contraseña. • Implementación de formularios para el inicio de sesión y el registro del usuario cliente • Validación de los datos requeridos. • Redireccionamiento hacia los módulos asignados. • Pruebas respectivas. 	
			HU012	Consumir varios <i>endpoints</i> para visualizar los pedidos asignados.	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para visualizar los pedidos asignados por parte del administrador. • Pruebas respectivas. 	
			HU013	Consumir varios <i>endpoints</i> para cambiar el estado de los pedidos.	<ul style="list-style-type: none"> • Diseño y codificación del prototipo para cambiar el estado de los pedidos. • Pruebas respectivas. 	

SB004	Pruebas en el <i>frontend</i>	<ul style="list-style-type: none"> • Pruebas unitarias. • Pruebas de compatibilidad. • Pruebas de aceptación. 	
SB005	Despliegue del <i>frontend</i>	<ul style="list-style-type: none"> • Despliegue del <i>frontend</i> en <i>Vercel</i>. 	
Documentación		<ul style="list-style-type: none"> • Documento del Trabajo de Integración Curricular. • Anexos. 	
TOTAL			240H

Diseño de interfaces

A continuación, se presentan los prototipos de cada uno de los módulos del sistema *web*, en los cuales se observa el diseño de las interfaces para cada rol de usuario: cliente, administrador y cocinero. Desde la Fig. 31 hasta la Fig. 45 presentan las interfaces del sistema *web*.



Fig. 31: Diseño de Interfaz - Pagina informativa

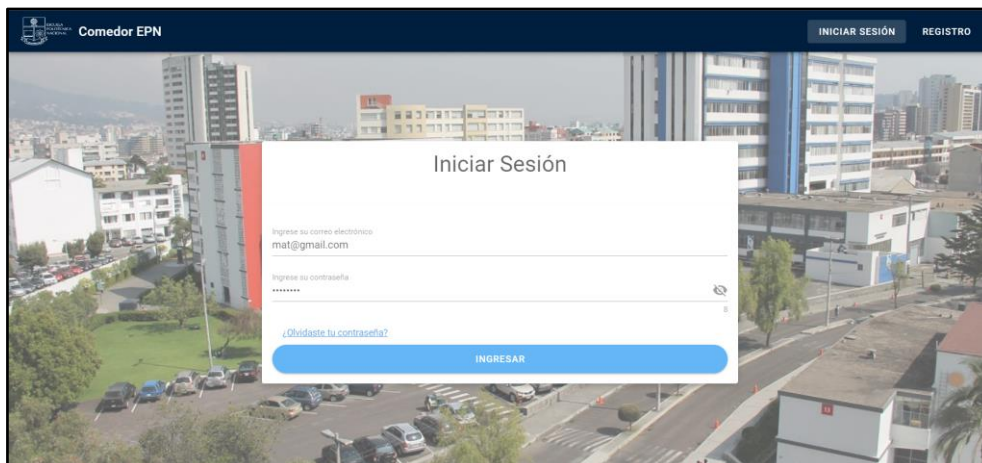


Fig. 32: Diseño de Interfaz - Inicio Sesión

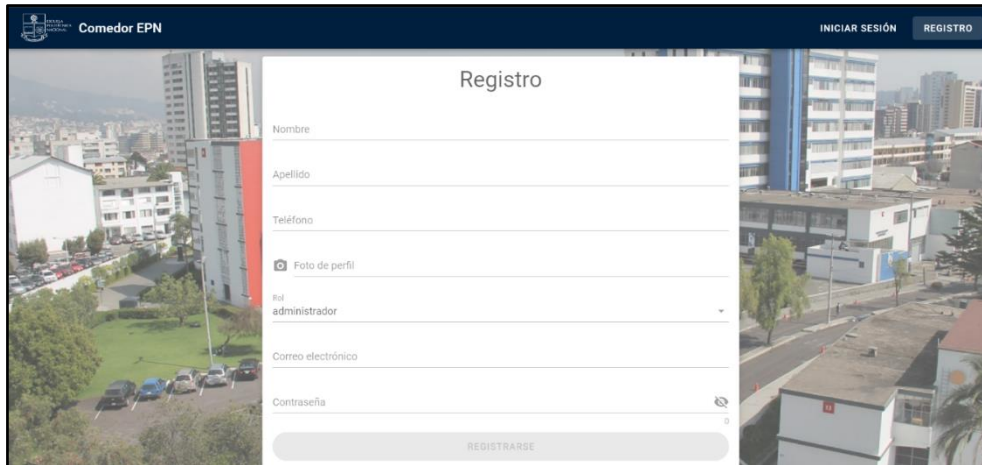


Fig. 33: Diseño de Interfaz - Registro

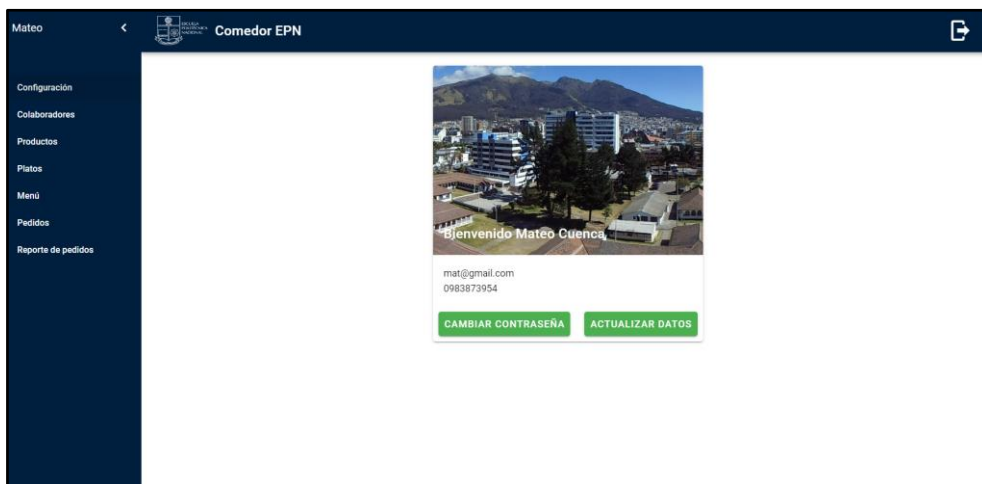


Fig. 34: Diseño de Interfaz - Página de Inicio

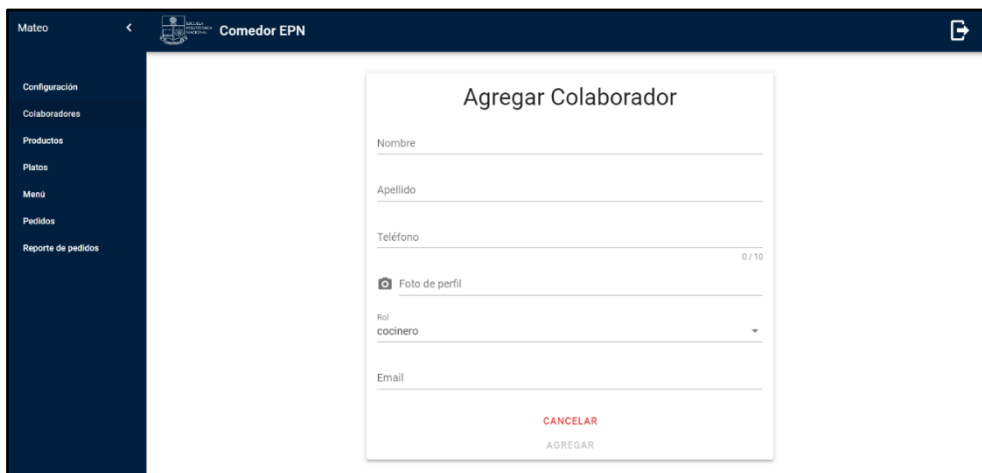


Fig. 35: Diseño de Interfaz - Crear administrador/cocinero

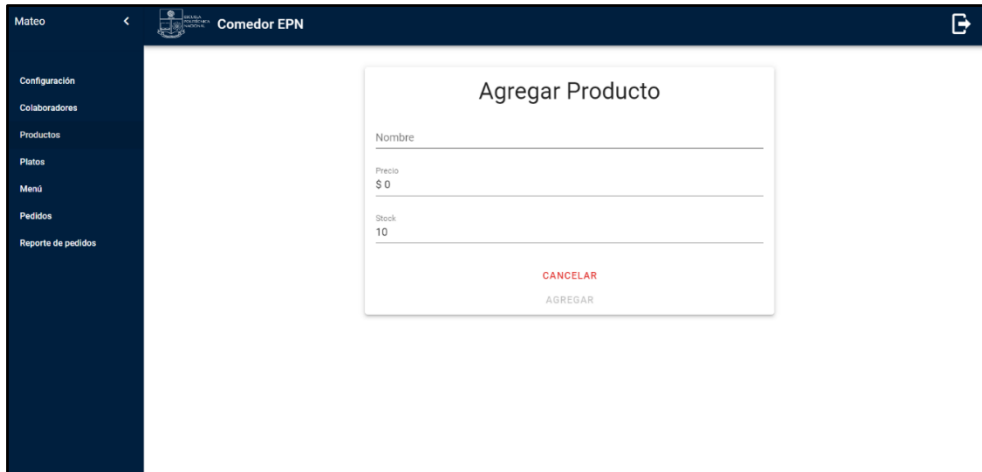


Fig. 36: Diseño de Interfaz - Crear producto

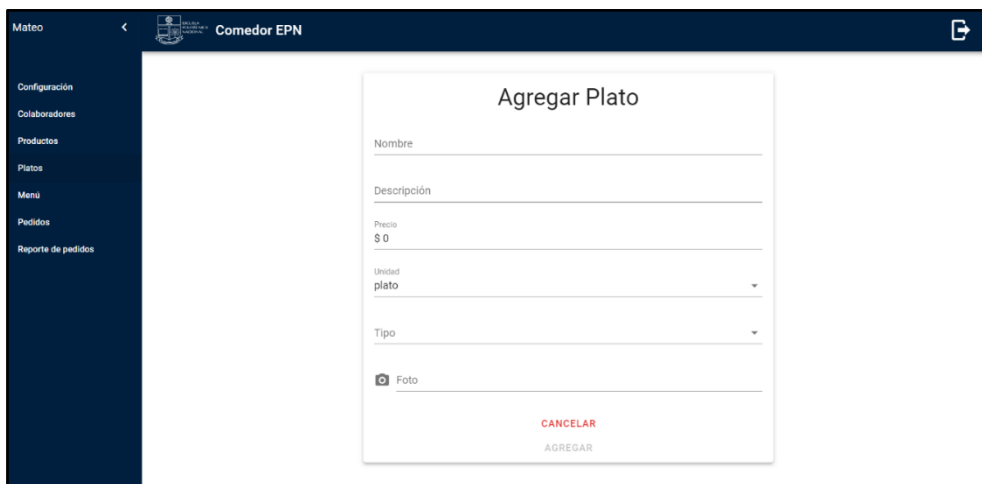


Fig. 37: Diseño de Interfaz - Crear plato

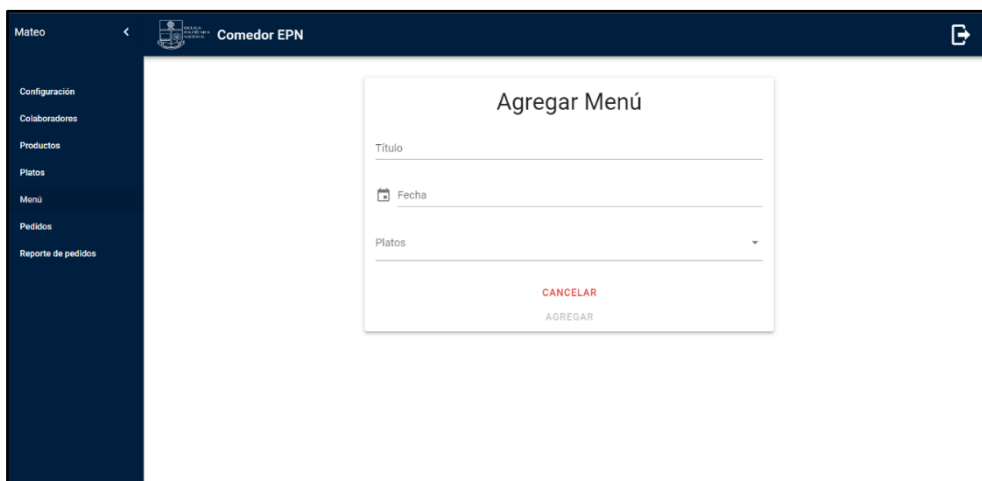


Fig. 38: Diseño de Interfaz – Agregar menú

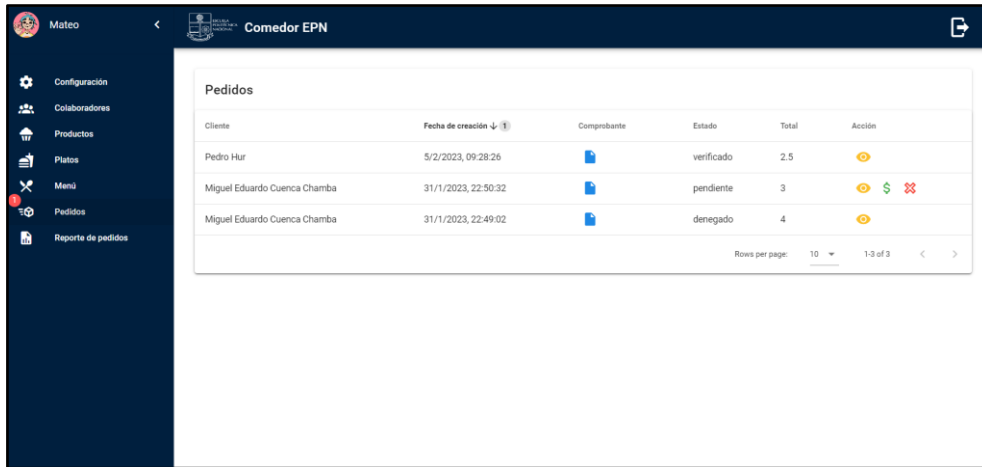


Fig. 39: Diseño de Interfaz - Pedidos generados por los clientes

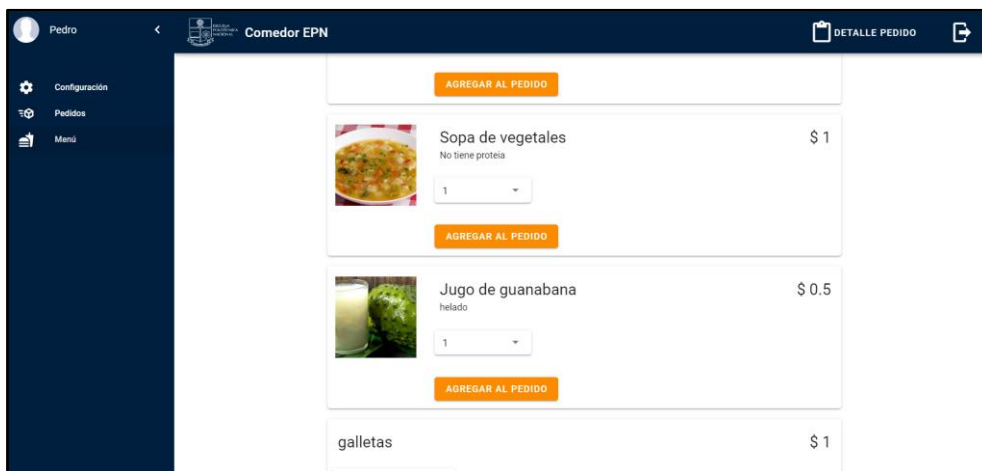


Fig. 40: Diseño de Interfaz – Crear pedido

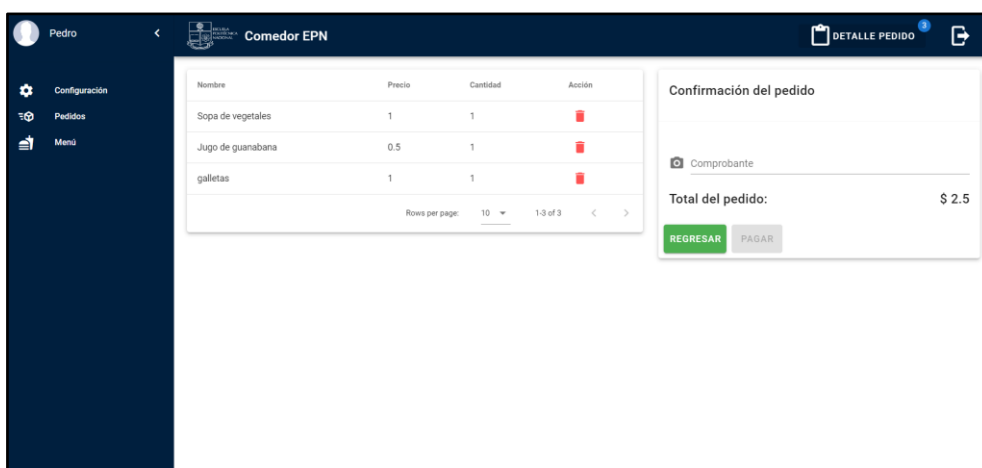


Fig. 41: Diseño de interfaz - Confirmar pago del pedido

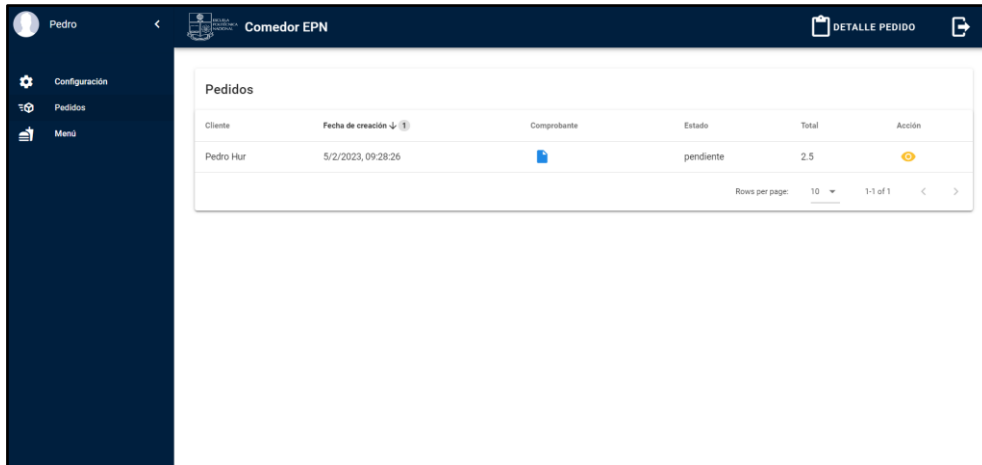


Fig. 42: Diseño de Interfaz - Pedidos solicitados

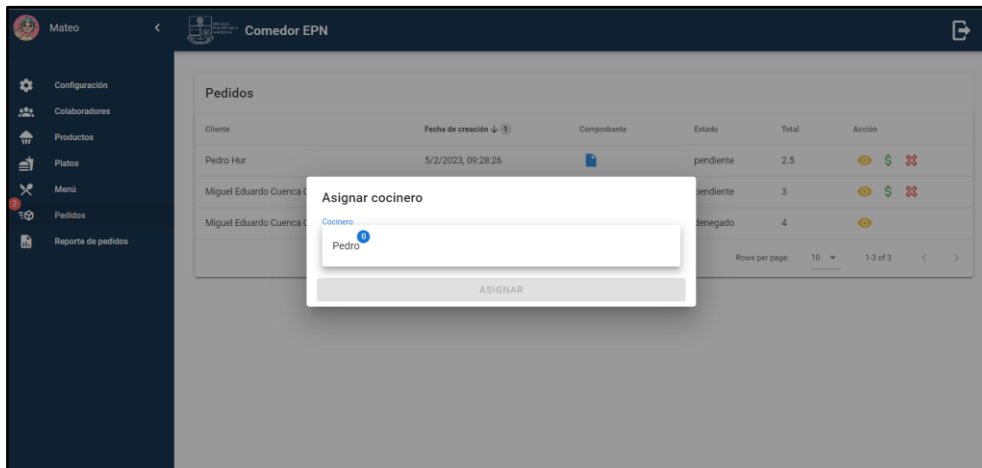


Fig. 43: Diseño de Interfaz - Asignar pedido a un cocinero

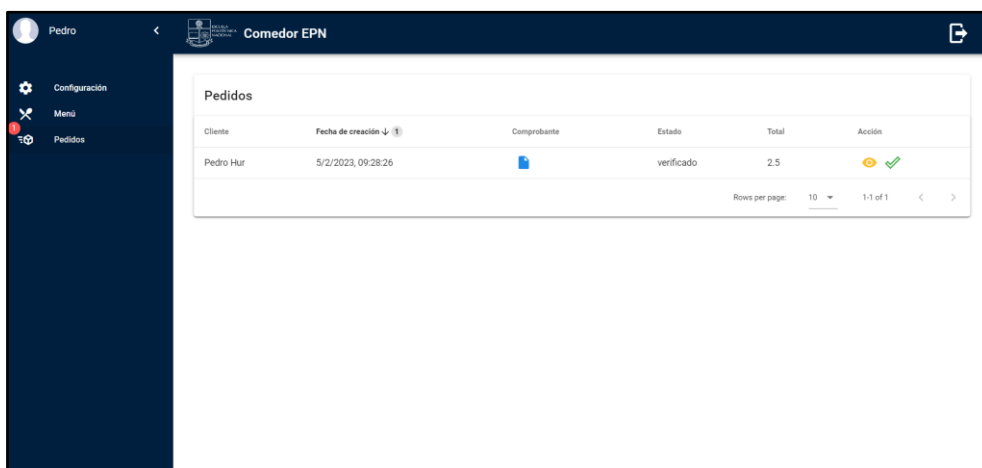


Fig. 44: Diseño de Interfaz - Pedidos asignados

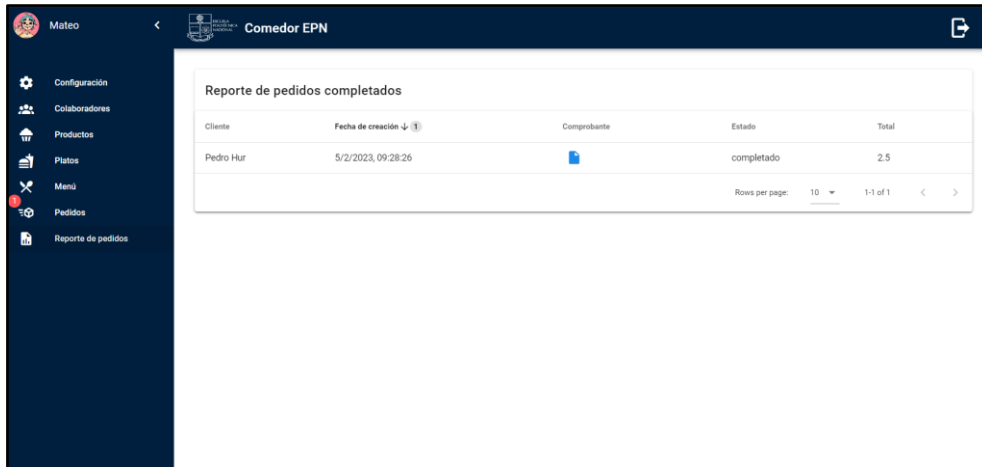


Fig. 45: Diseño de Interfaz - Reporte de pedidos completados

Pruebas

A continuación, se presentan los prototipos de cada uno de los módulos del sistema *web*, en los cuales

Pruebas unitarias

A continuación, se muestra la construcción de un test se crea una clase con funciones para cada uno de los módulos del sistema desarrollado, Los test van desde la **Fig. 46** hasta la **Fig. 50**, en cada una de las Figuras se presenta el proceso antes mencionado, junto con los resultados de cada una de las pruebas unitaria.

```

1  import { mount } from "@vue/test-uti
2  import CardMenu from '@views/Menu/c
3
4  describe("MenuView.vue", () => {
5    it("Test unit del modulo generar c
6      const wrapper = mount(CardMenu,{
7        propsData:{
8          menu:{
9            titulo:'MenúLunes',
10           fecha:'2023-30-01',
11           componentes:[]
12         },
13       }
14     });
15
16     expect(wrapper.text()).toContain
17   });

```

Fig. 46: Prueba unitaria #2 del módulo crear menú

```
PASS tests/unit/menu.spec.ts
  MenuView.vue
    ✓ Test unit del modulo generar card menu (87 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.942 s, estimated 3 s
Ran all test suites matching /unit/i.
```

Fig. 47: Resultado de la prueba unitaria #2

```
tests > unit > comida.spec.ts > ...
1  import { mount } from "@vue/test-utils";
2  import TituloMenu from '@views/Comidas/
3
4  describe("MenuView.vue", () => {
5    it("Test unit del modulo Menú", () =>
6      const wrapper = mount(TituloMenu,{
7        propsData:{
8          titulo: 'Menú Lunes',
9        }
10     });
11
12     expect(wrapper.text()).toContain('Me
13   });
14 });
15
```

Fig. 48: Prueba unitaria #3 del módulo generar pedido

```
> app-web@0.1.0 test
> vue-cli-service test:unit unit

PASS tests/unit/comida.spec.ts
  MenuView.vue
    ✓ Test unit del modulo Menú (23 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.953 s, estimated 3 s
Ran all test suites matching /unit/i.
```

Fig. 49: Resultado de la prueba unitaria #3

```

1 import { mount } from "@vue/test-utils";
2 import PedidoDetalles from '@/views/Pedido/compo
3
4 describe("PedidoVue.vue", () => {
5   it("Test unit modulo detalles del pedido", ()
6     const wrapper = mount(PedidoDetalles,{
7       propsData:{
8         componentes:[],
9         productos:[{
10          _producto:{
11            precio: 0.5,
12            nombre:'Doritos'
13          },
14          cantidad: 10
15        }]
16      }
17    });
18
19     expect(wrapper.html()).toContain('v-card pro
20   });
21 });

```

Fig. 50: Prueba unitaria #4 del módulo detalle del pedido

```

PASS tests/unit/pedido.spec.ts
  PedidoVue.vue
    ✓ Test unit modulo detalles del pedido (135 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        3.303 s
Ran all test suites matching /unit/i.

```

Fig. 51: Resultado de la prueba unitaria #4

Pruebas de compatibilidad

Para esta sección se verifica la correcta funcionalidad del sistema *web* en distintos navegadores con el fin de detectar posibles fallas y si existen cambios en la presentación del sistema. A continuación, desde la **Fig. 52** hasta la **Fig. 63** se presentan los resultados obtenidos en los navegadores *web*.



Fig. 52: Página principal con información general vista desde el navegador Mozilla Firefox

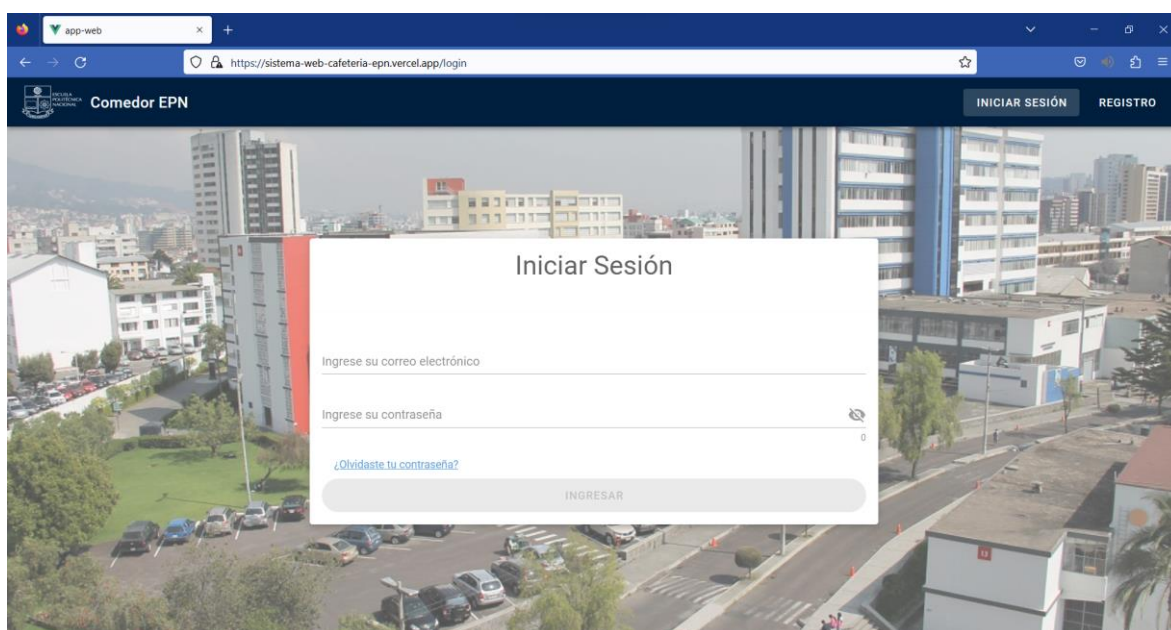


Fig. 53: Página de inicio de sesión vista desde el navegador Mozilla Firefox

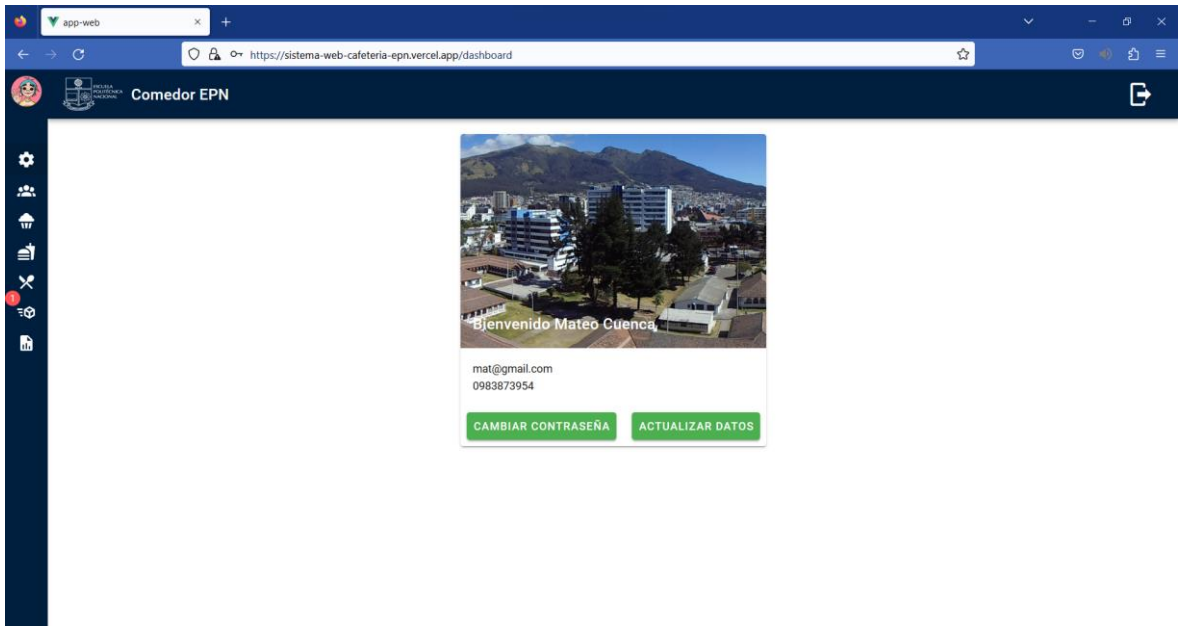


Fig. 54: Página inicial vista desde el navegador Mozilla Firefox

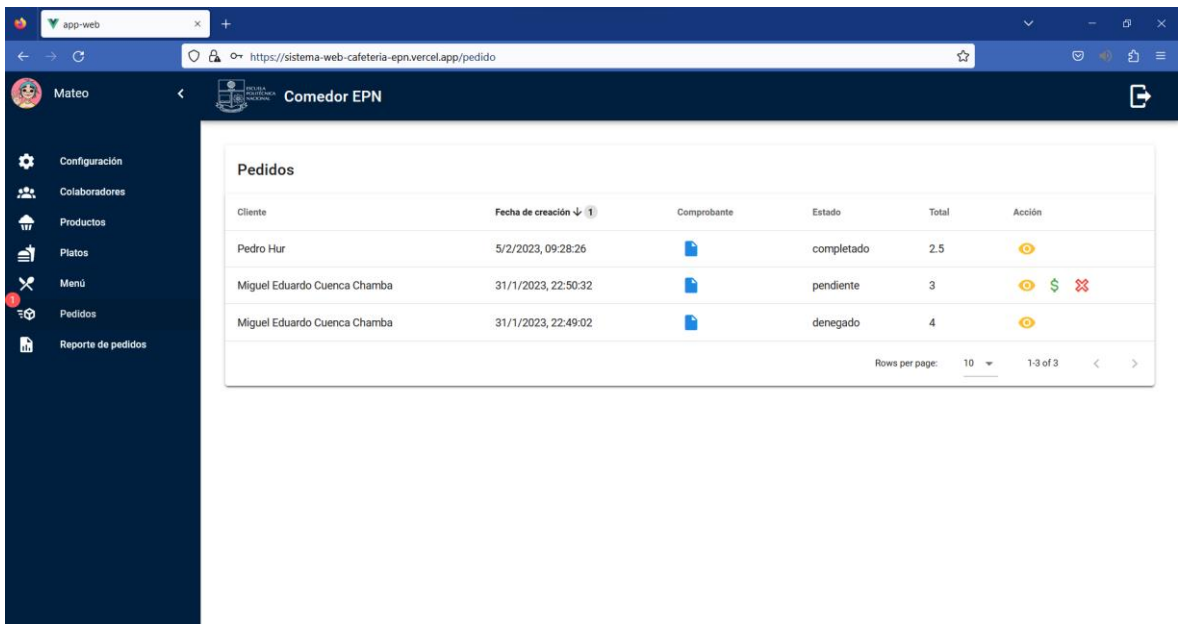


Fig. 55: Módulo de pedidos generados visto desde el navegador Mozilla Firefox



Fig. 56: Página principal con información general vista desde el navegador Microsoft Edge

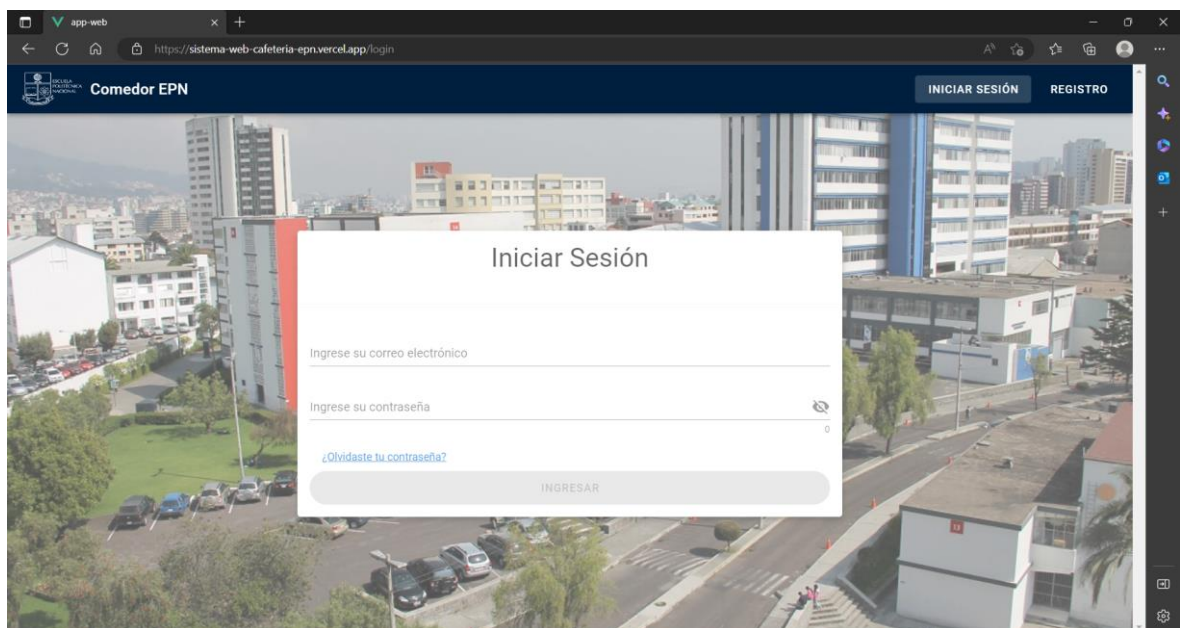


Fig. 57: Página de inicio de sesión vista desde el navegador Microsoft Edge

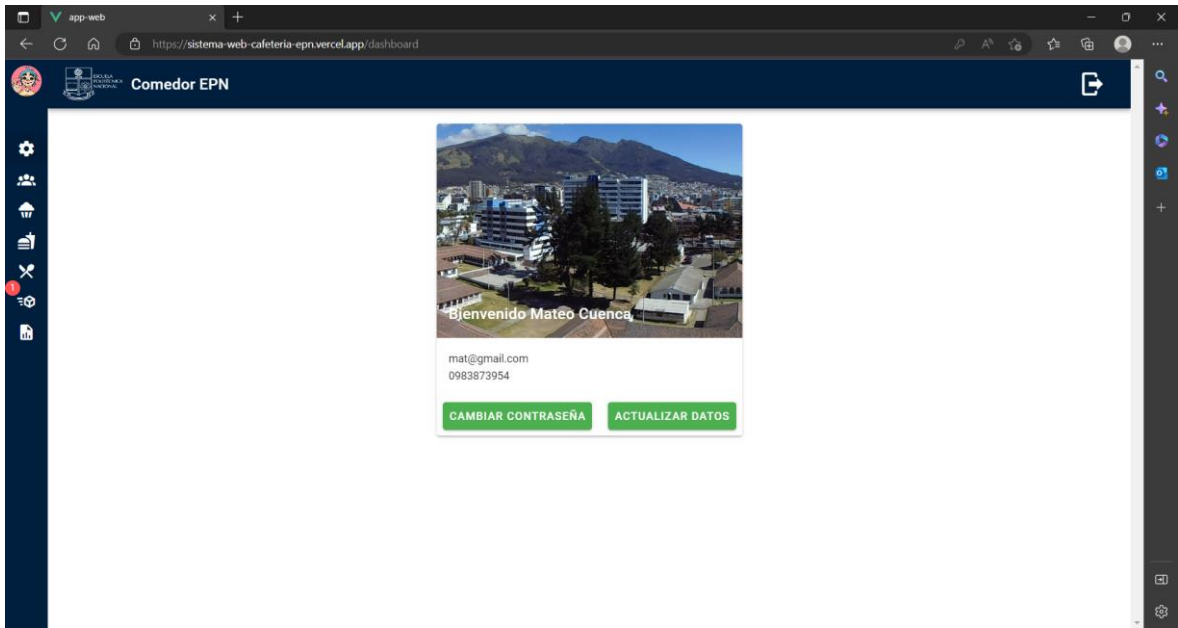


Fig. 58: Página inicial vista desde el navegador Microsoft Edge

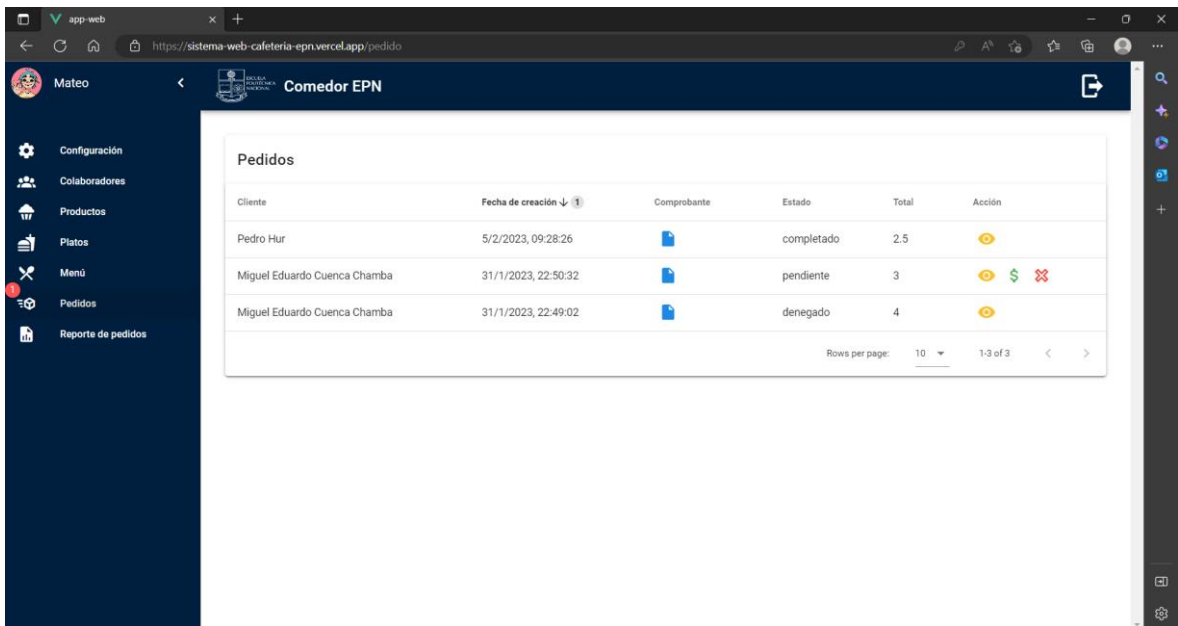


Fig. 59: Módulo de pedidos generados visto desde el navegador Mozilla Firefox

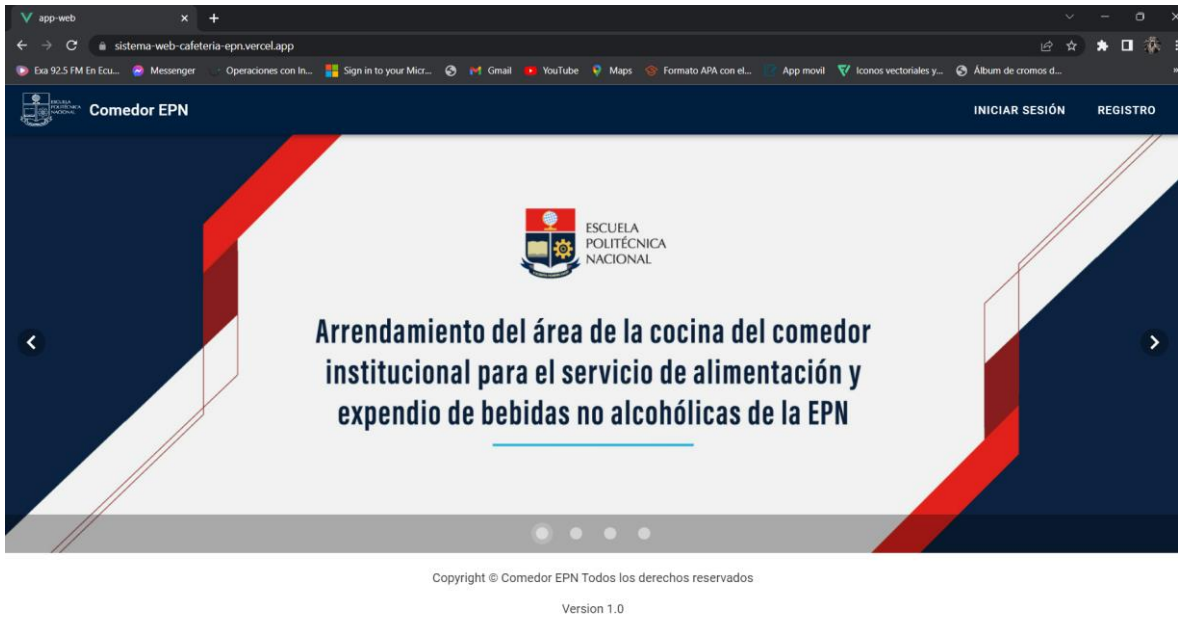


Fig. 60: Página principal con información general vista desde el navegador Google Chrome

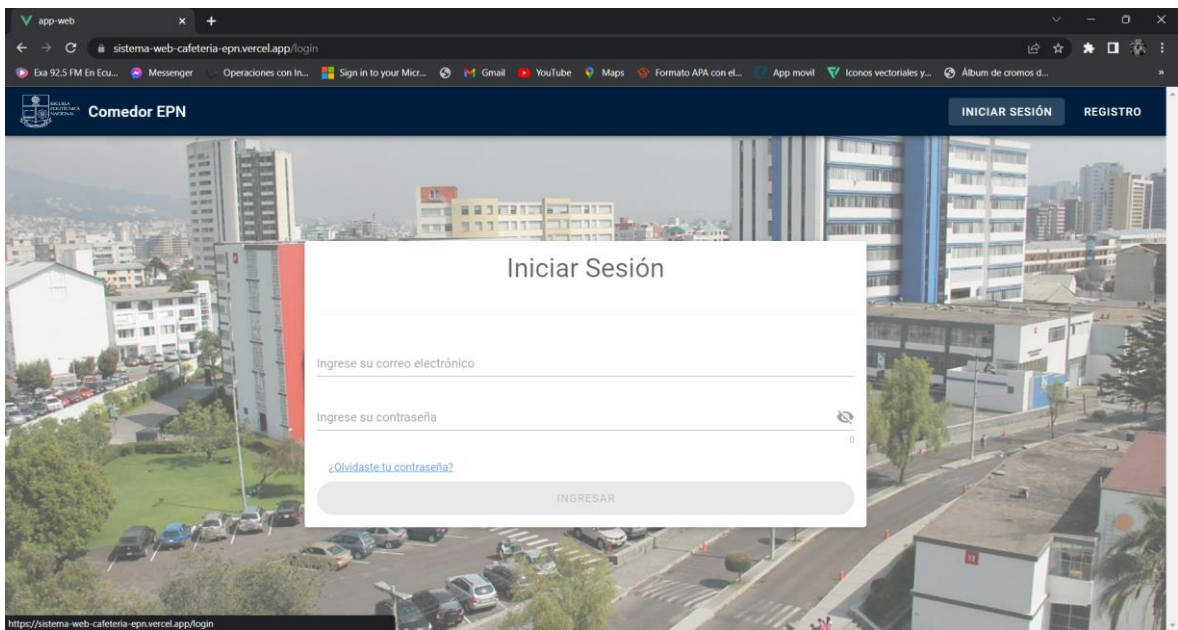


Fig. 61: Página de inicio de sesión vista desde el navegador Google Chrome

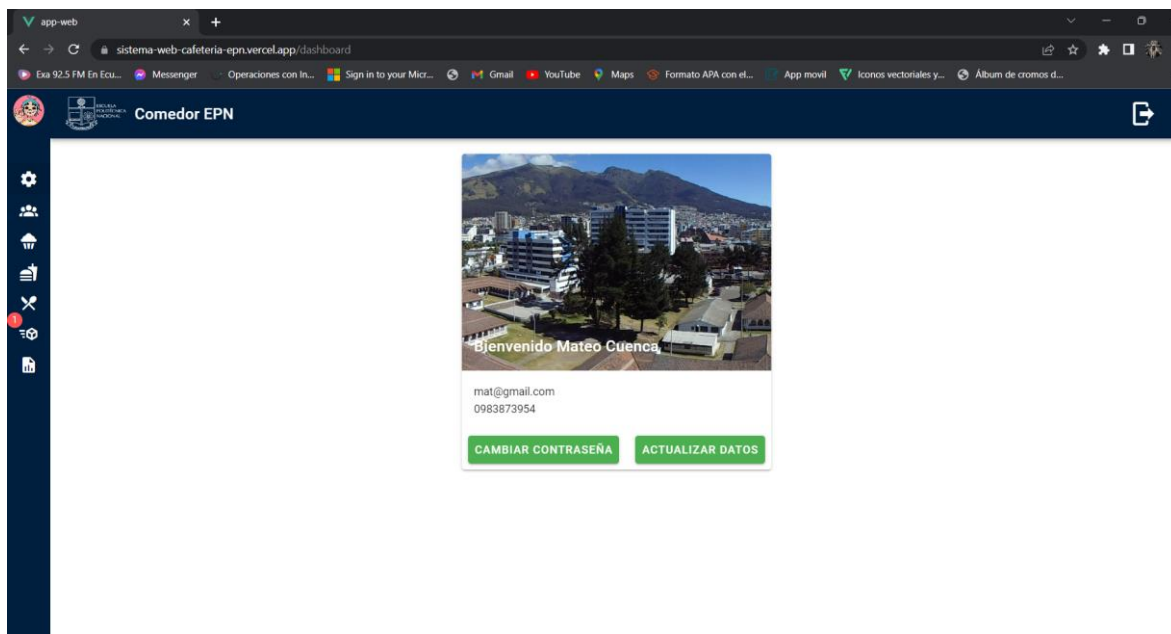


Fig. 62: Página inicial vista desde el navegador Google Chrome

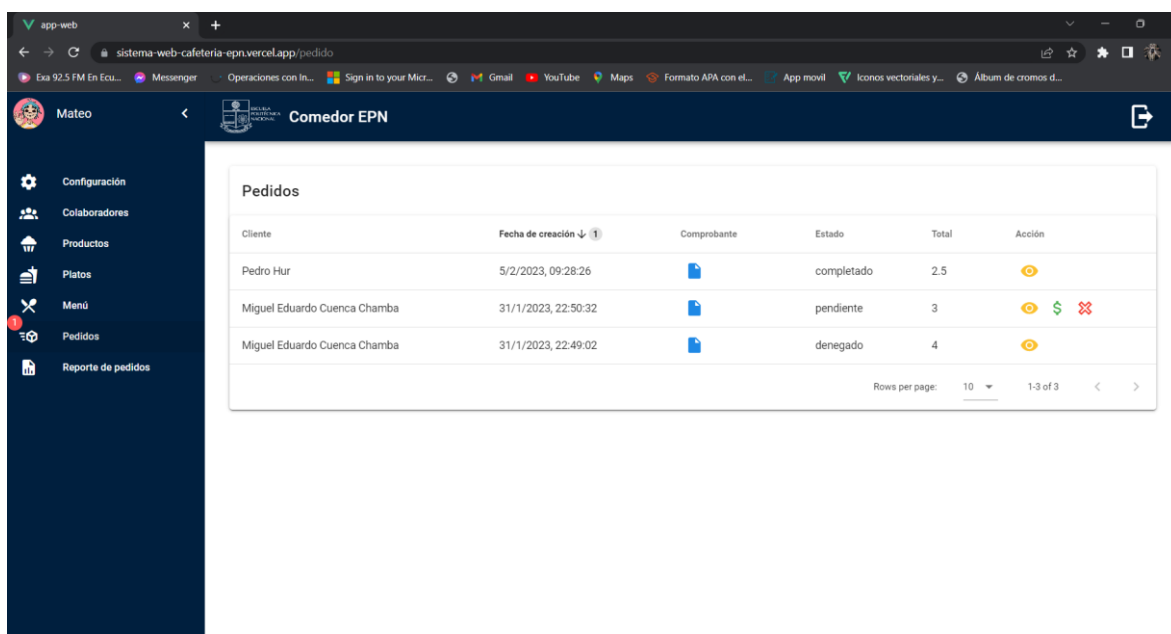


Fig. 63: Módulo de pedidos generados visto desde el navegador Google Chrome

Pruebas de aceptación

A continuación, desde la **TABLA XXV** hasta la **TABLA XXXVI**, se presentan las 12 Pruebas de aceptación restantes. Cabe mencionar que cada prueba expone los procedimientos especificados para cada rol de usuario y su adecuada interacción, ejecución, verificación y aprobación de estas.

TABLA XXV: Prueba de Aceptación N°2 - Consumir varios *endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0002	Identificador historia de Usuario: HU002
Nombre: Consumir varios <i>endpoints</i> para iniciar sesión, cerrar sesión y recuperar contraseña	
<p>Descripción: El usuario cocinero, cliente, administrador tiene que consumir varios <i>endpoints</i> para poder:</p> <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión • Recuperar contraseña <p>Para la recuperación de contraseña el usuario debe dirigirse al módulo de olvide mi contraseña, en el cual debe escribir su correo electrónico registrado para así recibir un código que tendrá que digitar luego de proporcionar el correo, de tal manera que pueda cambiar su contraseña.</p>	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. En la barra de navegación del sistema <i>web</i> se observan las opciones Iniciar Sesión, Registro. 2. Seleccionar “Iniciar Sesión” 3. En el formulario digitar los datos solicitados correo y contraseña, las credenciales son las ingresados en el registro, luego aplastar “Ingresar” y así accederá al sistema. 4. Para “Cerrar Sesión”, simplemente dar clic en el botón que se encuentra en la barra de navegación, el cual tiene como referencia una flecha apuntando hacia la derecha. 5. En el caso de olvidar la contraseña, seleccionar “Olvidaste tu contraseña” y en la vista escribir el correo de la cuenta, dar clic en enviar. 6. Se envía un código al correo electrónico el cual se debe escribir en la siguiente vista, para que permita cambiar la contraseña. 	
<p>Resultado deseado:</p> <p>El sistema <i>web</i> permite iniciar sesión, recuperar contraseña y cerrar sesión.</p>	
<p>Evaluación de la prueba:</p> <p>Se verificar el resultado previsto.</p> <p>Conformidad del cliente 100%.</p>	

TABLA XXVI: Prueba de Aceptación N°3 - Consumir varios *endpoints* para registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0003	Identificador historia de Usuario: HU003
Nombre: Consumir varios <i>endpoints</i> para registrarse como usuario administrador y cliente, y registrar usuarios administradores y cocineros	
Descripción: Los usuarios administrador y cliente deben consumir varios <i>endpoints</i> para registrarse, además el usuario administrador es el encargado en generar un nuevo cocinero y desde su panel también otro usuario administrador.	
Pasos de ejecución:	
Administrador	
<ol style="list-style-type: none"> 1. En la barra de navegación del sistema <i>web</i> se observan las opciones Iniciar Sesión, Registro. 2. Seleccionar “Registro” 3. Digitar en cada campo los datos solicitados, los datos deben ser reales. 4. Una vez registrado, se redirecciona al módulo de “Iniciar Sesión” 	
Cliente	
<ol style="list-style-type: none"> 1. En la barra de navegación del sistema <i>web</i> se observan las opciones Iniciar Sesión, Registro. 2. Seleccionar “Registro” 3. Digitar en cada campo los datos solicitados, los datos deben ser reales. 4. Una vez registrado, se redirecciona al módulo de “Iniciar Sesión” 	
Cocinero	
El registro del colaborador cocinero es diferente, ya que el administrador es quien lo registra.	
<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como administrador. 2. En la barra de navegación lateral, seleccionar el módulo “Colaboradores”, se visualiza una TABLA con todos los colaboradores. 3. Aplastar en el botón con icono de ‘+’, posterior a eso se redirecciona a una vista en la que se presenta un formulario para realizar el registro del nuevo colaborador. 4. Llenar cada uno de los campos con datos reales para registrar al usuario cocinero. 	
Resultado deseado:	
El sistema <i>web</i> permite registrarse	

Evaluación de la prueba:

Se verificar el resultado previsto.

Conformidad del cliente 100%.

TABLA XXVII: Prueba de Aceptación N°4 - Consumir varios *endpoints* para visualizar el pedido y la factura pagada

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0004	Identificador historia de Usuario: HU004
Nombre: Consumir varios <i>endpoints</i> para visualizar el pedido y la factura pagada	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para visualizar a detalle el pedido generado y el comprobante de haber realizado el pago de este	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como administrador. 2. En la barra de navegación lateral, seleccionar el módulo “Pedidos”, se visualiza una TABLA con todos los pedidos generados por los clientes. 3. En la TABLA se muestra información sobre el pedido, en la columna acción. 4. Hacer clic en el botón con icono de ‘ojo’, para así visualizar el detalle del pedido. 5. En la misma TABLA en la columna de comprobante. 6. Hacer clic en el botón con icono de ‘hoja’, para poder visualizar el comprobante del pago del pedido. 	
Resultado deseado:	
El sistema <i>web</i> permite visualizar el pedido y la factura pagada.	
Evaluación de la prueba:	
Se verificar el resultado previsto.	
Conformidad del cliente 100%.	

TABLA XXVIII: Prueba de Aceptación N°5 - Consumir varios *endpoints* para asignar el pedido al cocinero

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0005	Identificador historia de Usuario: HU005
Nombre: Consumir varios <i>endpoints</i> para asignar el pedido al cocinero	

<p>Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para asignar el pedido a un cocinero que se encuentre libre o con un menor número de pedidos asignados.</p>
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como administrador. 2. En la barra de navegación lateral, seleccionar el módulo “Pedidos”, se visualiza una TABLA con todos los pedidos generados por los clientes. 3. En la TABLA se muestra información sobre el pedido, en la columna acción. 4. Hacer clic en el botón con icono de ‘dólar’, para así asignar el pedido a un cocinero. 5. Se muestra una vista con los cocineros y el número de pedidos asignados a cada uno. 6. Seleccionar el cocinero al que se desee asignar el pedido, luego dar clic en ‘Asignar’.
<p>Resultado deseado:</p> <p>El sistema <i>web</i> permite asignar un pedido a un cocinero.</p>
<p>Evaluación de la prueba:</p> <p>Se verificar el resultado previsto.</p> <p>Conformidad del cliente 100%.</p>

TABLA XXIX: Prueba de Aceptación N°6 - Consumir varios *endpoints* para agregar platos del menú del almuerzo y snacks

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0006	Identificador historia de Usuario: HU006
Nombre: Consumir varios <i>endpoints</i> para agregar platos del menú del almuerzo y snacks.	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para crear el plato que se tiene disponible en el menú del día sea sopa, segundo o bebida, también tiene la posibilidad de agregar snacks.	
Pasos de ejecución:	
Agregar Productos	
<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como administrador. 2. En la barra de navegación lateral, seleccionar el módulo “Productos”, se visualiza una TABLA con todos los productos disponibles de la Cafetería. 	

3. Hacer clic en el botón con icono de '+', para agregar un nuevo producto.
4. Se muestra un formulario se debe llenar cada uno de los campos, luego dar clic en el botón 'Agregar'.
5. En la **TABLA** se visualizará el producto recién creado.

Agregar Platos

1. Acceder al sistema *web* como administrador.
2. En la barra de navegación lateral, seleccionar el módulo "Platos", se visualiza una **TABLA** con todos los platos disponibles para el almuerzo.
3. Hacer clic en el botón con icono de '+', para agregar un nuevo plato.
4. Se muestra un formulario se debe llenar cada uno de los campos, luego dar clic en el botón 'Agregar'.
5. En la **TABLA** se visualizará el plato recién creado.

Resultado deseado:

El sistema *web* permite agregar platos del menú del almuerzo y snacks de la Cafetería.

Evaluación de la prueba:

Se verificar el resultado previsto.

Conformidad del cliente 100%.

TABLA XXX: Prueba de Aceptación N°7 - Consumir varios *endpoints* para gestionar el menú de los almuerzos

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0007	Identificador historia de Usuario: HU007
Nombre: Consumir varios <i>endpoints</i> para generar el menú de los almuerzos	
Descripción: El usuario administrador tiene que consumir varios <i>endpoints</i> para llenar el formulario del menú diario y seleccionar los platos disponibles para el menú, de tal manera que se cree el menú del día.	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como administrador. 2. En la barra de navegación lateral, seleccionar el módulo "Menú". 3. Hacer clic en el botón con icono de '+', para agregar el menú del día. 4. Se muestra un formulario se debe llenar cada uno de los campos, luego dar clic en el botón 'Agregar'. 5. El menú creado se mostrará en una TABLA. 	
Resultado deseado:	

El sistema <i>web</i> permite agregar el menú de almuerzo.
Evaluación de la prueba: Se verificar el resultado previsto. Conformidad del cliente 100%.

TABLA XXXI: Prueba de Aceptación N°8 - Consumir varios *endpoints* para modificar perfil

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0008	Identificador historia de Usuario: HU008
Nombre: Consumir varios <i>endpoints</i> para modificar perfil.	
Descripción: El usuario administrador, cliente y cocinero, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para visualizar y actualizar perfil cuando el administrador, cliente y cocinero lo requiera. La lectura y actualización del perfil se realiza mediante un formulario donde se ingresa la siguiente información: <ul style="list-style-type: none"> • Nuevo número de teléfono • Nueva foto de perfil • Nueva contraseña 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como administrador. 2. En la barra de navegación lateral, seleccionar el módulo “Configuración”. 3. Hacer clic en el botón ‘Actualizar datos’, para modificar los datos del usuario. 4. Se muestra un formulario para actualizar los datos según se necesite. 5. Hacer clic en el botón ‘Cambiar contraseña’, para actualizar la contraseña de la cuenta del usuario. 6. Para efectuar los cambios dar clic en el botón ‘Actualizar’ 	
Resultado deseado:	
El sistema <i>web</i> permite modificar el perfil de usuario.	
Evaluación de la prueba: Se verificar el resultado previsto. Conformidad del cliente 100%.	

TABLA XXXII: Prueba de Aceptación N°9 - Consumir varios *endpoints* para generar pedidos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0009	Identificador historia de Usuario: HU009
Nombre: Consumir varios <i>endpoints</i> para generar pedidos.	
<p>Descripción: El cliente, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para crear pedidos cuando el cliente lo requiera. La creación de un pedido se realiza por un formulario, se selecciona la siguiente información.</p> <ul style="list-style-type: none"> • Sopa • Segundo • Bebida • Snacks 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como cliente. 2. En la barra de navegación lateral, seleccionar el módulo “Menú”. 3. Seleccionar el plato que se desea y su cantidad. 4. En la barra de navegación se indica un botón ‘Detalle pedido’, al hacer clic se redireccionará a la vista del pedido con más detalle. 	
<p>Resultado deseado:</p> <p>El sistema <i>web</i> permite generar un pedido.</p>	
<p>Evaluación de la prueba:</p> <p>Se verificar el resultado previsto. Conformidad del cliente 100%.</p>	

TABLA XXXIII: Prueba de Aceptación N°10 - Consumir varios *endpoints* para confirmar y cargar el comprobante de pago del pedido.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0010	Identificador historia de Usuario: HU010
Nombre: Consumir varios <i>endpoints</i> para confirmar y cargar el comprobante de pago del pedido.	
<p>Descripción: El cliente, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para realizar la confirmación y el pago del pedido a través de un sistema en el que cargara el comprobante de pago.</p>	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como cliente. 2. En la barra de navegación, seleccionar el módulo “Detalle pedido”. 3. Se visualiza el detalle del pedido y el valor a pagar. 4. En el módulo de confirmación del pedido, cargar el comprobante de pago, y hacer clic en el botón de ‘Pagar’
<p>Resultado deseado: El sistema <i>web</i> permite confirmar y cargar el comprobante de pago del pedido.</p>
<p>Evaluación de la prueba: Se verificar el resultado previsto. Conformidad del cliente 100%.</p>

TABLA XXXIV: Prueba de Aceptación N°11 - Consumir varios *endpoints* visualizar el estado del pedido.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0011	Identificador historia de Usuario: HU011
Nombre: Consumir varios <i>endpoints</i> visualizar el estado del pedido.	
Descripción: El cliente, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para visualizar el estado en el que se encuentra su pedido, lo que se presenta en su vista es si el pedido está en pendiente, o verificado que quiere decir que el pedido se encuentra en preparación o completado.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como cliente. 2. En la barra de navegación lateral, seleccionar el módulo “Pedidos”. 3. Se visualiza el detalle del pedido y el estado en el que esta. 4. Al momento que el pedido pase a completado el cliente puede pasar a retirar su pedido. 	
Resultado deseado: El sistema <i>web</i> permite visualizar el estado del pedido	
Evaluación de la prueba: Se verificar el resultado previsto. Conformidad del cliente 100%.	

TABLA XXXV: Prueba de Aceptación N°12 - Consumir varios *endpoints* para visualizar los pedidos asignados.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0012	Identificador historia de Usuario: HU012
Nombre: Consumir varios <i>endpoints</i> para visualizar los pedidos asignados.	
Descripción: El usuario cocinero, en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para visualizar en el módulo pedidos todos los pedidos que el administrador le ha asignado.	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como cocinero. 2. En la barra de navegación lateral, seleccionar el módulo "Pedidos". 3. Se visualiza una TABLA con los pedidos asignados a preparar, en la columna acción. 4. Hacer clic en el botón con icono de 'ojo', para visualizar el detalle de cada pedido. 	
Resultado deseado:	
El sistema <i>web</i> permite visualizar los pedidos asignados.	
Evaluación de la prueba:	
Se verificar el resultado previsto.	
Conformidad del cliente 100%.	

TABLA XXXVI: Prueba de Aceptación N°13 - Consumir varios *endpoints* para cambiar el estado de los pedidos

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0013	Identificador historia de Usuario: HU013
Nombre: Consumir varios <i>endpoints</i> para cambiar el estado de los pedidos	
Descripción: El usuario cocinero tiene que consumir varios <i>endpoints</i> para cambiar el estado del pedido, dicho estado que puede cambiar el usuario cocinero es: pedido completado.	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder al sistema <i>web</i> como cocinero. 2. En la barra de navegación lateral, seleccionar el módulo "Pedidos". 3. Se visualiza una TABLA con los pedidos asignados a preparar, en la columna acción. 	

4. Hacer clic en el botón con icono de 'visto', para dar por completado el pedido.
5. El pedido desaparecerá de la bandeja.

Resultado deseado:

El sistema *web* permite finalizar y cambiar el estado del pedido.

Evaluación de la prueba:

Se verificar el resultado previsto.

Conformidad del cliente 100%.

ANEXO III

El siguiente enlace, permite observar el Manual de Usuario, donde se detalla la información con respecto a las funcionalidades del sistema *web*, así como la participación de los perfiles en la interacción.

<https://www.youtube.com/watch?v=V8aZeE5wNbw>

ANEXO IV

A continuación, se especifica las credenciales de acceso para el sistema *web*, así como el enlace al repositorio en GitHub en donde se encuentra el código fuente y en el apartado de README los pasos para realizar la instalación de forma local.

Credenciales de acceso para el sistema *web*

Para acceder al sistema *web* en producción, ingresar a la siguiente URL:

<https://sistema-web-cafeteria-eqn.vercel.app/>

Credenciales para el perfil administrador:

- Correo electrónico: admin@gmail.com
- Contraseña: 12345678

Repositorio del código fuente del sistema *web*

El código fuente de todo el proyecto, se encuentra alojado en el repositorio GitHub, el cual se puede acceder a través de la siguiente URL:

<https://github.com/MateoCueva03/sistemaWebCafeteria-EPN.git>