

**Instituto de Educación Superior Tecnológico Público
"De las Fuerzas Armadas"**



TRABAJO DE APLICACIÓN PROFESIONAL

**IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN EN ORDEN
DE PEDIDOS PARA EL CONTROL DE LA POLLERÍA *EL RICO
POLLO*, EN EL DISTRITO DE LOS OLIVOS, LIMA- 2023.**

**PARA OPTAR EL TÍTULO PROFESIONAL TÉCNICO EN
COMPUTACIÓN E INFORMÁTICA**

PRESENTADO POR:

PAZ VILLEGAS, Diana Marisol

SILVA ALBERCA, Roberto Carlos

LIMA, PERÚ

2024

Este trabajo está dedicado a Dios. A nuestros padres por hacer de nosotros buenas personas por su apoyo incondicional a lo largo de este período de aprendizaje.

AGRADECIMIENTOS

Damos gracias a Dios por estar con nosotros a lo largo de nuestra carrera, guiándonos y brindando fuerzas en nuestros momentos débiles.

A nuestros padres por su apoyo y confianza. Gracias por ayudarnos alcanzar nuestras metas como persona y estudiantes. Este proyecto es un tributo a su legado y a la eterna admiración que sentimos por ustedes.

A nuestros profesores que me han brindado los mejores conocimientos en diferentes áreas de nuestra carrera profesional.

Al “Instituto de Educación Superior Tecnológico Publico las Fuerzas Armadas”, (IESTPFFAA) por abrirnos las puertas y brindar la oportunidad de avanzar en nuestra carrera profesional.

ÍNDICE

CARATULA	1
AGRADECIMIENTO	4
RESUMEN.....	10
INTRODUCCIÓN	11
CAPÍTULO I.....	12
DETERMINACIÓN DEL PROBLEMA	12
Formulación del problema	13
1.1.2 Problema específico	13
1.2. Objetivos	13
1.3 Justificación.....	14
CAPÍTULO II	15
MARCO TEÓRICO	15
2.1. Estado de arte	16
2.2. Bases teóricas	18
2.2.1. Servidor Web.....	18
2.2.2. Base de Datos	18
2.2.3. Software Libre.....	19
2.2.4. MySQL.....	20
2.2.5.HTML.....	20
2.2.6.CSS (Cascading Style Sheets).....	21
2.2.7. Rup	22
2.2.8. Arquitectura Cliente/Servidor	22
2.2.9. JavaScript	23
2.2.10. Php.....	24
2.2.11 Xampp	25
2.2.12. Front-end.....	25
2.2.13. Back-end.....	26

2.2.14. Visual Code	27
CAPÍTULO III	28
DESARROLLO DEL TRABAJO	28
3.1 Finalidad	29
3.2 Propósito	29
3.3 Componentes	29
3.3.1 Levantamiento de la información	31
3.3.2 Análisis de la información	32
3.3.2.1 Análisis del proceso propuesto	32
3.3.2 Diseño del sistema web	32
3.3.3.1 Maqueta de la página principal del sistema web	33
3.3.3.2 Estructura del sistema web	33
3.3.3.3 Descripción General de la Estructura del Sistema web	37
3.3.3.4 Flujoograma de Base de Datos en MySQL	42
3.3.3.5 Representación descriptiva de las tablas de la base de datos	42
3.3.4 Codificación del sistema	45
3.3.5 Pruebas del sistema	75
3.4 Actividades	76
3.5 Limitaciones	77
CAPÍTULO IV	78
RESULTADOS	79
CAPÍTULO V	80
CONCLUSIONES Y RECOMENDACIONES	80
CONCLUSIONES	81
RECOMENDACIONES	82
REFERENCIAS BIBLIOGRÁFICAS	83
APÉNDICES	85

Apéndice B: Estructura presupuestaria	87
Apéndice C: Manual de Usuario	88

Índice de Figuras

Figura 1. Servidor Web	18
Figura 2. Base de Datos.....	19
Figura 3. Software Libre	19
Figura 4. MySQL	20
Figura 5. HTML	21
Figura 6. CSS	21
Figura 7. Fases de la metodología RUP	22
Figura 8. Arquitectura Cliente/Servidor.....	23
Figura 9. JavaScript.....	23
Figura 10. Php	24
Figura 11. Xampp.....	25
Figura 12. Front-end.....	26
Figura 13. Back-end	26
Figura 14. Visual code	27
Figura 15. Etapas del desarrollo de la implementación	31
Figura 16. Cuestionario realizado al propietario.....	32
Figura 17. Control de servicios de la pollería El Rico Pollo.....	32
Figura 18. Control de servicios de la pollería El Rico Pollo.....	33
Figura 19. Menú principal - Opción Panel.....	34
Figura 20. Menú principal.....	34
Figura 21. Menú principal - Opción Platos	35
Figura 22. Menú principal - Opción Salas	35
Figura 23. Menú principal - Opción Ajustes.....	36
Figura 24. Menú principal - Opción Venta	36
Figura 25. Login del sistema web	37
Figura 26. Vista de la ventana principal del usuario - administrador	37
Figura 27. Platos.....	38
Figura 28. Descripción de Salas.....	38
Figura 29. Usuarios	39
Figura 30. Pedidos.....	39
Figura 31. Opción Mesas	40
Figura 32. Opción Ajustes / Salir	40

Figura 33.Inicio	41
Figura 34.Vista de la ventana principal del usuario - Mozo	41
Figura 35. Diagrama de base de datos.....	42
Figura 36. Tabla: platos.....	42
Figura 37.Tabla Pedidos.....	43
Figura 38.Tabla configuración	43
Figura 39. <i>Tabla: Salas</i>	43
Figura 40.Tabla detalle pedidos	44
Figura 41.Temp_pedidos.....	44
Figura 42.Tabla Usuarios	44
Figura 43.Vista del sistema web en el navegador Google Chrome	75
Figura 44.Vista del sistema web en el navegador Microsoft Edge	76

RESUMEN

El objetivo del presente trabajo fue implementar un sistema de apoyo a la gestión de la pollería EL Rico Pollo, En El Distrito De Los Olivos, Lima.

En este negocio se realizan diversos procesos manualmente, como uno de ellos son los pedidos de los clientes, por lo que se optó por optimizar este proceso realizarlo de manera automatizado.

Para ello se usaron herramientas tecnológicas como la metodología Proceso Racional Unificado (RUP), ya que es la parte principal que tiene como objetivo ordenar y estructurar el desarrollo de software.

Las tareas se planificaron utilizando el software Ms Project ya que nos permite planear, gestionar y controlar nuestro proyecto en los plazos establecidos para así evitar retrasos y beneficiar a la productividad.

Se aplicó como herramienta de investigación un cuestionario que consiste en una serie de preguntas con el propósito de obtener información sobre la pollería.

El diseño se dividió en dos fases.

La primera fase fue diseñar la base de datos para el sistema de gestión MySQL, y la otra fase fue realizar el maquetado del sistema web, seguido de la programación o codificación.

Las pruebas finales relacionadas y la implementación se realizaron utilizando el lenguaje PHP.

Palabras claves: Sistema de gestión – Control de ventas - Pollería

INTRODUCCIÓN

Hoy en día, el sector empresarial se esfuerza por mejorar su desempeño comercial con el objetivo de optimizar los servicios que brinda a una amplia gama de usuarios y traducir esto en mayores ganancias. La Pollería El Rico Pollo decidió implementar un sistema automatizado que permitiera un mayor control sobre los servicios prestados, y esto motivó el desarrollo de este estudio.

Este sistema de control gestiona las ventas del día, registra pedidos, etc., y muestra informes proporcionados con información detallada de los pedidos para la toma de decisiones del propietario.

Este estudio consta de los siguientes capítulos:

Capítulo I: Aquí se presenta la situación actual de este establecimiento, se fijan objetivos generales y específicos, y se mencionan los aspectos técnicos y comerciales del estudio. También se menciona la legitimidad.

Capítulo II: Este segundo capítulo se rescata los antecedentes de este rubro en relación a autorización de los procesos comerciales de pollerías. Además, se apoyan conceptos y definiciones que forman un fundamento teórico.

Capítulo III: Este capítulo describe cada etapa del desarrollo del trabajo presentando los lenguajes de programación, codificaciones y otros.

Capítulo IV: A continuación, se presentan los hallazgos obtenidos luego de finalizar los trabajos en el sistema de la pollería.

Capítulo V: El capítulo final proporciona conclusiones después de completar el trabajo de aplicación especializada y también señala las recomendaciones a considerar al ejecutar un sistema web en producción.

CAPÍTULO I
DETERMINACIÓN DEL PROBLEMA

Formulación del problema

Muchas empresas del sector de venta de comida están implementando sistemas para optimizar la automatización. Este negocio, tiene posicionado 2 años de apertura y viene realizando los procesos de manera manualmente en lugar de utilizar un sistema de pedidos, lo que genera insatisfacción en los clientes debido a retrasos en las entregas. Se presenta la preocupación si el pedido se ingresó correctamente o no y si tomará mucho tiempo la entrega. De manera similar, los gerentes de los negocios desconocen cuántas ventas realizan por día. Otro problema es que no se cuenta con un historial sobre el registro de tus pedidos ya que cada pedido está duplicado.

1.1.1 Problema general

PG. ¿Cómo se optimizará el control de pedidos de ventas en la pollería El Rico Pollo, en el distrito de los Olivos, Lima?

1.1.2 Problema específico

PE1. ¿Cómo se obtendrá la información de los pedidos de ventas en la pollería El Rico Pollo, en el distrito de los Olivos, Lima?

PE2. ¿Cómo se evitaría la pérdida de información o manipulación de los pedidos de las ventas en la pollería *El Pollo Rico*, en el distrito de los Olivos, Lima?

PE3. ¿Cómo se obtendría la información resumida de los pedidos de ventas en la pollería *El Rico pollo*, en el distrito de los Olivos, Lima?

PE4. ¿Cómo se controlaría el historial al realizar los pedidos de ventas en la pollería El Rico Pollo, en el distrito de los Olivos, Lima?

1.2. Objetivos

1.2.1. Objetivo general

OG. Implementar un sistema de pedidos de ventas para optimizar la administración y el control de la pollería El Rico Pollo.

1.2.2. Objetivos específicos

OE1. Registrar la información de los pedidos de ventas en un software de base datos para contar con datos exactos y oportunos en forma adecuada.

OE2. Identificar la pérdida de información o manipulación de pedidos en la pollería El Rico Pollo.

OE3. Identificar la información de los pedidos en la pollería El Rico Pollo.

OE4. Controlar el historial de los pedidos de la pollería El Rico Pollo.

1.3 Justificación

En la actualidad la pollería El Rico Pollo, está ubicada en el distrito de los Olivos, este negocio carece de una implementación de tickets conectada a una base de datos para la optimizar el proceso de compra y recepción el pedido.

A nivel técnico, Consta con el aporte evidenciado la propuesta de solución que plantea una respuesta positiva al problema observado en beneficio a un mejor control en reducir los errores en toma de pedido, esto nos permitirá optimizar la confianza de los compradores cuando ingresen a consumir en la pollería.

A nivel empresarial, será fundamental la confianza de los compradores cuando ingresen a consumir a la pollería El Rico Pollo, ya que se optimizará el proceso de control de las ventas en tiempo real, reducir los errores en toma de pedido y el movimiento de la caja de los dueños han logrado un control muy eficiente en la pollería.

CAPÍTULO II
MARCO TEÓRICO

2.1. Estado de arte

2.1.1 Antecedentes internacionales

Sánchez (2018), planteó la siguiente investigación denominada: Sistema de pedidos para la mejora de la administración de la pollería La Carreta 2018, El desarrollo se basa en procesos como generar pedidos, resolver problemas provocados por pedidos retrasados, reducir errores durante la aceptación de pedidos, gestionar en tiempo real las ventas y los movimientos de caja, y lo tengo. La metodología utilizada se denomina Proceso Unificado Racional (RUP) y consta de cuatro fases: Iniciar, Refinar, Construir y Migrar. También se utilizó el Lenguaje Unificado de Modelado (UML) para representar y documentar el proceso. Como resultado del proyecto se concluyó que el sistema de pedidos tiene como finalidad apoyar la administración de la pollería La Carreta.

Barrios (2020). Este proyecto está dirigido a restaurantes de la ciudad de Sahagún Córdoba con el objetivo de mejorar la atención al cliente y minimizar los tiempos de pedido. A través de una plataforma virtual compuesta por una aplicación móvil desarrollada con tecnología Bootstrap, la aplicación accede al dispositivo del cliente y procesa el pedido cuando el cliente llega a la tienda o cuando realiza una reserva y gestionar su pedido en el restaurante desde casa para crear y generar. Existe una aplicación que permite gestionar todo lo relacionado con los productos ofertados y observar los pedidos que algunos usuarios tienen registrados en la base de datos. Las aplicaciones se desarrollan utilizando tecnologías HTML, CSS, JavaScript y JQuery en el lado del cliente y el lenguaje de programación PHP como administrador de base de datos MySQL.

Vásquez (2021), Este trabajo se realizó con el objetivo de implementar un sistema informático para mejorar la atención al cliente en el restaurante Pollería Solymar - Bagua Grande - Utcubamba en el estado Amazonas. Si bien pertenece al sistema de investigación de gestión de TI y es aplicable a aplicaciones según su dirección, es una tecnología con el servicio al cliente como variable dependiente y el sistema informático como variable independiente. Antes de utilizar el sistema se realizó una prueba previa y luego de su implementación se realizó una prueba posterior a una población de 100 clientes de un restaurante utilizando una encuesta como método de investigación y un cuestionario como herramienta.

2.1.2 Antecedentes nacionales

Yoshi (2018), Este proyecto fue creado por un “chifa Paraíso Oriental”. Tomando como base los procesos principales la creación de pedidos y Facturación, identificación de soluciones a los retrasos de pedidos, minimizar errores al recibir pedidos, tener la capacidad de controlar y flujo de caja gracias a la implementación Sistema de pedidos. Esto se lleva a cabo según el método a desarrollar un Proceso Unificado Racional (RUP) correspondiente a sus fases: Iniciación, Desarrollo, Construcción, Transición. El lenguaje de modelado unificado (UML) se utiliza para la presentación y documentación. El resultado del proyecto es la gestión de Chifa Paraíso Oriental para mejorar el control.

Valenzuela (2018). Por su parte realizó un estudio intitulado una aplicación de gestión de pedidos GOCHICKEN, que automatiza el proceso de gestión de pedidos de la pollería “El Gran Pollo Gordo”. El siguiente proyecto se divide en siete capítulos. El capítulo 1 se basa en un análisis de la organización y los desafíos actuales. El capítulo 2 trata sobre los fundamentos teóricos de la empresa y describe los procesos de negocio de Pollería. El Capítulo 3 se basa en el inicio y la planificación del proyecto, y el proyecto se aprueba formalmente con el nombramiento de un director de proyecto. El capítulo 4, se basa en la implementación, seguimiento y control del proyecto. El Capítulo 5, se basa en el cierre del proyecto, cuyo objetivo es la finalización formal de todas las actividades incluidas en el proyecto. El Capítulo 6 se basa en la evaluación del desempeño del proyecto. El capítulo 7 se basa en las conclusiones y recomendaciones del proyecto, cuyos resultados están alineados con los objetivos adoptados y las recomendaciones propuestas.

Cruz (2021). En la era de la tecnología de la información, el comportamiento del consumidor es cada vez más sofisticado. Desde la compra y adquisición de servicios que requieren que las empresas implementen nuevas tecnologías como la búsqueda. Para poder competir en el mercado. Administre estratégicamente los pedidos dentro de su organización en los procesos y problemas que resulten en retrasos en el cumplimiento de las solicitudes del servicio. El propósito de esta investigación es introducir chatos y mejorar la gestión de pedidos. Los resultados de la implementación del cabo de gestión de pedidos nos permiten presentar una comparación previa y posterior a la prueba de la métrica "nivel de cumplimiento de pedidos". Antes de introducir el cabo, el valor del índice era del 52%. Después de implementar y probar la funcionalidad del cabo, este valor fue del 82%. Se dice que la productividad ha mejorado un 30% y el número de pedidos.

2.2. Bases teóricas

2.2.1. Servidor Web

Un servidor web es una plataforma computacional muy potente que resguarda datos para ser consultados por diversos usuarios. Cuenta con un software que entrega la información solicitada por visitantes o miembros de un grupo de trabajo, a través de un dispositivo conectado a la red. Un servidor web algunas pueden ser definido como un hardware y como un software; porque efectivamente posee cualidades de ambos. Es una computadora con una alta potencia que le permite mantenerse conectada a la red las 24 horas del día, los 365 días del año; además de contar con un sistema operativo, de almacenamiento y de procesamiento sumamente eficaces para atender las consultas de los usuarios. (HubSpot ,2022).

Figura 1. *Servidor Web*



<https://rockcontent.com/es/blog/que-es-un-servidor/>

2.2.2. Base de Datos

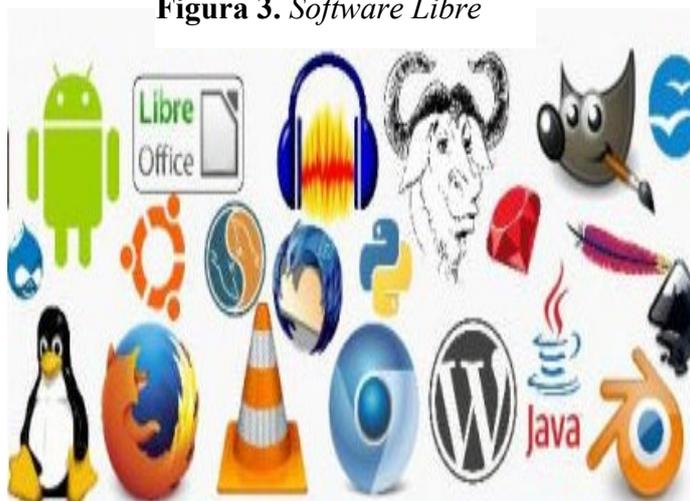
Es esencialmente una colección organizada de información de un programa de computadora puede seleccionar rápidamente los datos deseados. Una base de datos es un sistema de almacenamiento de documentos electrónicos, están organizadas por campos, registros y archivos. Un campo es una pieza de información; Un registro es un conjunto completo de campos; Un archivo es una colección de registros. Por ejemplo, una guía telefónica es similar a un archivo. Contiene una lista de registros, cada registro incluye tres campos: nombre, dirección y número de teléfono. (Armetrics,2022).

Figura 2. Base de Datos

<https://atlantictech.io/base-de-datos-como-se-dividen/>

2.2.3. Software Libre

Dr. Richard Stallman inició el movimiento del software libre en 1983 y comenzó a desarrollar el sistema operativo GNU en 1984. Es software libre: cualquiera es libre de copiarlo, distribuirlo y realizarle más o menos cambios. Este software brinda a cualquier persona la libertad de utilizar el programa para cualquier tipo de trabajo, sin ninguna obligación de compartir el programa con el desarrollador ni con ninguna organización en particular. (Luis Miguel Arteaga Mejía, 2001).

Figura 3. Software Libre

<https://espaciotecnologico.co/migrar-de-windows-a-linux/software-libre/>

2.2.4. MySQL

Fue desarrollado en 1995 por la empresa sueca Michael Widenius. Es el sistema de gestión de bases de datos relacionales más utilizado en la actualidad porque está basado en código abierto. Originalmente desarrollado por MySQL , fue adquirida por Sun Microsystems en 2008 y luego por Oracle Corporation en 2010 ya contaba con su propio motor InnoDB para MySQL. MySQL, es un sistema de gestión de bases de datos con doble licencia. Si bien es de código abierto.(TICPortal,2024).

Figura 4. *MySQL*



<https://openwebinars.net/blog/que-es-mysql/>

2.2.5.HTML

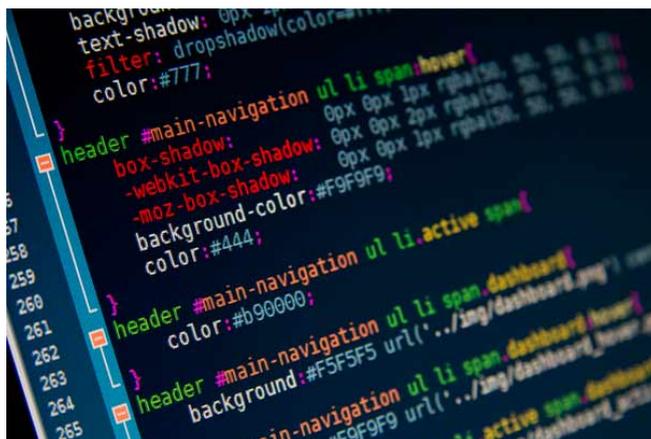
HTML (Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*) es el componente más básico de la Web. Definir el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript). "Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «Word *Wide Web*» (Red Informática Mundial). (mozilla.org. - 1998–2024).

Figura 5. HTML

<https://www.nextu.com/blog/que-es-html-rc22/>

2.2.6.CSS (Cascading Style Sheets)

CSS es una abreviatura de "Hojas de estilo en cascada". Básicamente, es un lenguaje que maneja el diseño y la presentación de sitios web. Funciona con el lenguaje HTML, que maneja el contenido básico del sitio web. Puede haber varias hojas de estilo porque tiene propiedades que se heredan de otras propiedades. Para muchas personas, una sencilla plantilla de blog es suficiente, si desea personalizar la apariencia de su sitio web, deberá implementar CSS. Cuando combinas CSS con un buen CMS, puedes ampliar el alcance de tu contenido. (Gustavo B. Ene 11, 2023).

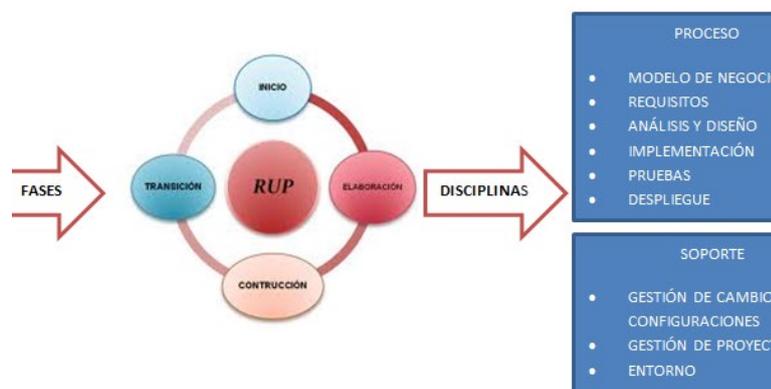
Figura 6. CSS

<https://www.shutterstock.com/image-photo/software-developer-programming-code-abstract-600nw-375299896.jpg>

2.2.7. Rup

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo a necesidades. (Tec Gurus 2022).

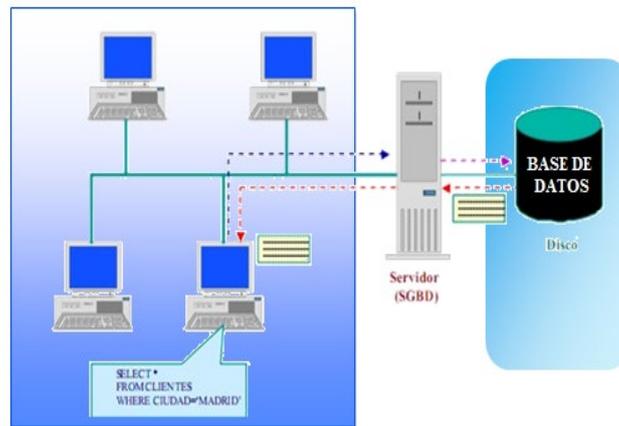
Figura 7. Fases de la metodología RUP



<https://metodogiarupgrupo23.blogspot.com/2014/03/fases-metodologia-rup.html>

2.2.8. Arquitectura Cliente/Servidor

Cliente-Servidor es uno de los estilos arquitectónicos distribuidos más conocidos, el cual está compuesto por dos componentes, el proveedor y el consumidor. El proveedor es un servidor que brinda una serie de servicios o recursos los cuales son consumido por el Cliente. En una arquitectura Cliente-Servidor existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar, en este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado. (Oscar Blancarte-2021).

Figura 8. Arquitectura Cliente/Servidor

<https://blog.infranetworking.com/modelo-cliente-servidor/>

2.2.9. JavaScript

(Desarrollado por Sun Microsystems en 1995). El lenguaje Java sigue una filosofía similar, pero está evolucionando. Java es un lenguaje de programación independiente de C, Pascal y Visual Basic. Esto significa que puede ejecutar programas Java fuera de su navegador.

Aunque JavaScript es un lenguaje completo que se puede utilizar para crear todo tipo de aplicaciones, JavaScript sólo "funciona" dentro de páginas HTML. (Amazon Web Services-2023).

Figura 9. JavaScript

```

15  const LOCALE = globalThis.navigator.language
16
17  const div = document.body.appendChild(document.createElement('div'))
18  const list = div.appendChild(document.createElement('ol'))
19
20
21  const dayNames = new Map()
22
23  for (let i = 0; i < 7; ++i) {
24    const d = Temporal.PlainDate.from({
25      year: Temporal.Now.plainDateISO().year,
26      month: 1,
27      day: i + 1,
28    })
29
30    dayNames.set(d.dayOfWeek, d.toLocaleString(LOCALE, { weekday: 'long' }))
31  }
32
33  for (const num of [...dayNames.keys()].sort((a, b) => a - b)) {
34    list.appendChild(Object.assign(
35      document.createElement('li'),
36      { textContent: dayNames.get(num) },
37    ))
38  }
39

```

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

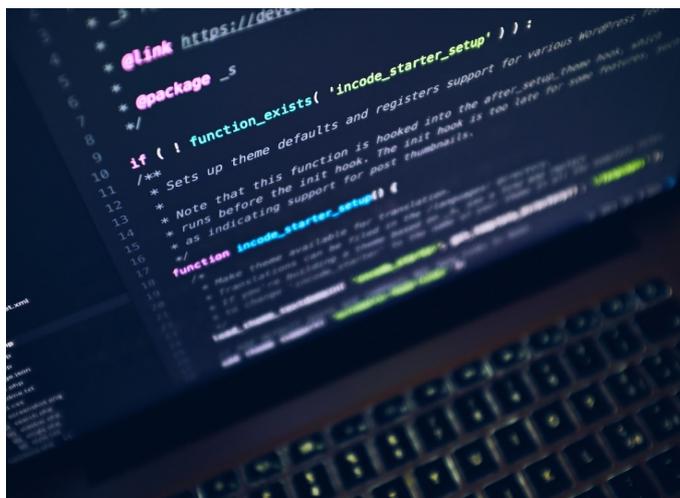
2.2.10. Php

(Creado por RASMUS LERDORF en junio de 1994).

PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. PHP originalmente significaba Personal Home Page (Página personal), pero ahora significa el inicialismo recursivo PHP: Hypertext Preprocessor.

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP (ITD Consulting SAC).

Figura 10. *PHP*

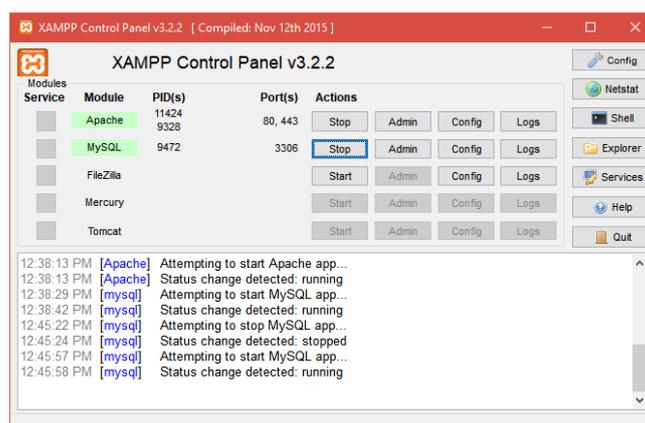


<https://www.epitech-it.es/que-es-php/>

2.2.11 Xampp

(Creado por APACHE FRIENDS ORGANIZATION el 4 de septiembre de 2002) Es un servidor web local multiplataforma que permite crear y probar páginas web y otros elementos de programación. Sin embargo, Xampp incluye varias herramientas que mejoran y facilitan la experiencia del desarrollador. (Tutoriales Dongee -2024).

Figura 11. *Xampp*



<https://www.dongee.com/tutoriales/que-es-xampp/#:~:text=comenzar%20a%20utilizarlo,-%C2%BFQu%C3%A9%20es%20Xampp%20y%20cu%C3%A1l%20es%20su%20utilidad%3F,facilitan%20la%20experiencia%20al%20desarrollador.>

2.2.12. Front-end

El frontend, también conocido como el lado del cliente, es la parte de una página web que los usuarios ven y con la que interactúan directamente. Esto incluye el diseño visual, la estructura de la página, los estilos, el contenido y la interactividad. En pocas palabras, el frontend es lo que convierte una página web en una experiencia atractiva y funcional para los usuarios. El front end developer implica traducir el diseño visual y las especificaciones de una página web en código HTML, CSS y JavaScript. Estos lenguajes de programación permiten crear la estructura de la página, aplicar estilos de diseño y añadir interactividad y funcionalidades a través de scripts (Kiwop -2024).

Figura 12. *Front-end*

<https://www.arimetrics.com/glosario-digital/frontend>

2.2.13. Back-end

El back-end son los datos y la infraestructura que permiten que las aplicaciones funcionen. Almacenen y procesen los datos de las aplicaciones para los usuarios. (Amazon Web Services-2023).

Figura 13. *Back-end*

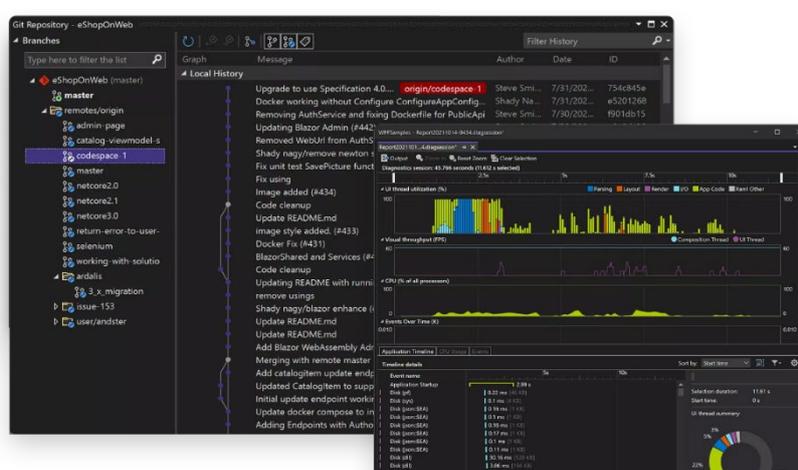
<https://www.arimetrics.com/glosario-digital/backend>

2.2.14. Visual Code

(Desarrollado por Microsoft con primer lanzamiento el 29 de abril de 2015).

Es un editor de código fuente ligero y potente que se ejecuta en el escritorio y está disponible para Windows, Linux y macOS, lo que permite a los programadores crear aplicaciones multiplataforma. Tiene soporte integrado para JavaScript, TypeScript y Node.js, y un amplio ecosistema de extensiones para otros lenguajes y entornos de ejecución como son C++, C#, Java, Python, Go, .NET. (Microsoft- 2024).

Figura 14. *Visual code*



<https://visualstudio.microsoft.com/es/#:~:text=Descargar%20Visual%20Studio-,%20Visual%20Studio%20Code,casi%20cualquier%20lenguaje%20de%20programaci%C3%B3n>

CAPÍTULO III
DESARROLLO DEL TRABAJO

3.1 Finalidad

La finalidad de este trabajo es contribuir con la gestión de pedidos para llevar un mejor control automatizado que permitirá de la forma más rápida, eficiente, controlada tener resultados oportunos.

3.2 Propósito

El trabajo tiene como propósito la implementación de un software con una base de datos para registrar los pedidos de una forma ordenada e identificar la pérdida o manipulación de pedidos en la pollería el Rico Pollo.

3.3 Componentes

La estructura del desarrollo del sistema está implementada en una estructura backend.

Así mismo los componentes complementarios que se utilizaron fueron:

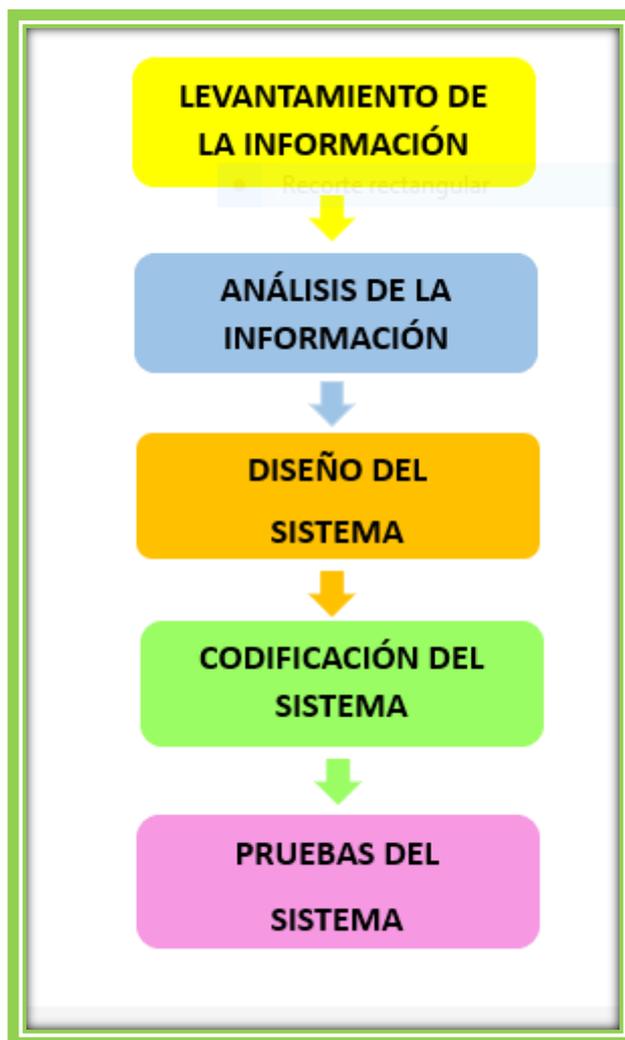
- ✓ **HTML:** Un lenguaje de marcado que define la estructura del contenido. Consiste en un conjunto de elementos que se utilizan para insertar etiquetas esto le permite crear un hipervínculo desde una palabra o imagen a otro sitio web.
- ✓ **CSS:** Las siglas (Cascading Style Sheets) CSS significan las siglas (Cascading Style Sheets) CSS significan, Hojas de estilo en cascada y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc..) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma automática y masiva. Estos se denominan estilos en cascada porque el código se lee, procesa y aplica de arriba a abajo.
- ✓ **PHP:** Es un lenguaje de código abierto especializado para el desarrollo web y puede integrarse en HTML, el lenguaje PHP prioriza las conexiones entre de usuario. Se utiliza para una variedad de aplicaciones y es de código abierto, por lo que los desarrolladores pueden editar su estructura y aplicarla a cualquier proyecto.
- ✓ **Bootstrap:** Es una biblioteca de herramientas de código abierto optimizadas para el diseño de sitios web y aplicaciones.
- ✓ **jQuery:** Esta es una biblioteca de JavaScript que le permite crear fácilmente sitios web dinámicos es desarrollado por John Resig en 2006 y se ha convertido en una de las herramientas más populares en el desarrollo web.

- ✓ **MySQL:** Un sistema de gestión de bases de datos relacionales de código abierto. Al igual que con otras bases de datos relacionales, MySQL almacena los datos en tablas formadas por filas y columnas.

XAMPP: Un servidor web local multiplataforma que le permite crear y probar sitios web y otros elementos de programación, es principalmente del sistema de gestión de bases de datos MySQL.

- ✓ **Visual Code:** Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y la Web. Incluye soporte de depuración, controles Git integrados, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
- ✓ **Microsoft Project:** Este software se usó para establecer las actividades que se necesitaron para la ejecución del sistema en el orden de la pollería.
- ✓ **Rational Rose:** Esta herramienta de diseño orientado a objetos se utilizó principalmente al momento de realizar el análisis de la información recopilada, ya que permitió el modelamiento de los diagramas de casos de uso del sistema de control de los pedidos.
- ✓ **Requerimientos:** El Trabajo de Aplicación Profesional para la pollería El Rico Pollo, abarca las etapas de:
Comenzó conociendo a nuestros clientes y recopilando toda la información sobre sus requisitos.

Figura 15. *Etapas del desarrollo de la implementación*



Nota: En la imagen, representa cada una de las fases del desarrollo.

3.3.1 Levantamiento de la información

En esta primera etapa, se generó un cuestionario y se entrevistó a los propietarios de la pollería para conocer la situación actual e identificar sus necesidades.

Figura 16. Cuestionario realizado al propietario

CUESTIONARIO AL PROPIETARIO DE LA POLLERIA

1. ¿Cómo realiza las ordenes de pedidos en la pollería?
2. ¿Qué tipos de productos venden actualmente?
3. ¿Qué información se obtiene del cliente para la toma de pedidos?
4. ¿Qué documentos utiliza para realizar las órdenes de pedidos?
5. ¿Tiene un sistema de soporte informático?
6. ¿Crees que es necesario contar con un sistema de orden de pedidos?
7. ¿Qué información se desea almacenar en la base de datos?
8. ¿Cuenta con Servicio de Internet?

Luego de realizar la entrevista al propietario, se acordó implementar un sistema de control de servicios web para la pollería EL RICO POLLO.

3.3.2 Análisis de la información

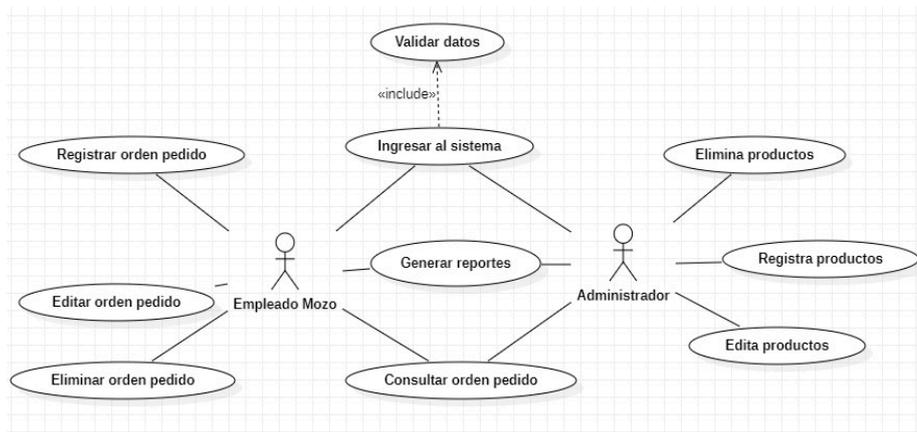
Luego de recopilar la información, se realizó un análisis de requisitos y se modelaron casos de uso de procesos.

3.3.2.1 Análisis del proceso propuesto.

Luego de una comprensión más profunda del proceso actual, se propuso una solución que fue modelada en un diagrama de casos de uso e implementada en el sistema web.

3.3.2 Diseño del sistema web

Figura 17. Control de servicios de la pollería El Rico Pollo



Al diseñar el sistema, se determinó las necesidades de información como contar con reportes como resultados.

Para implementar el diseño se utilizó el lenguaje de marcado de hipertexto siendo que es una herramienta que permite trabajar más confiable, hojas de estilo en cascada, el lenguaje de programación JavaScript y la biblioteca JQuery, estos dos últimos servirán para la programación de códigos.

Por lo tanto, al mismo tiempo que se creaba el diseño, la base de datos se desarrolló utilizando el sistema de gestión de bases de datos MySQL.

3.3.3.1 Maqueta de la página principal del sistema web

Figura 18. Control de servicios de la pollería El Rico Pollo



3.3.3.2 Estructura del sistema web

Se muestra el menú principal con sus respectivas opciones:

Figura 19. Menú principal

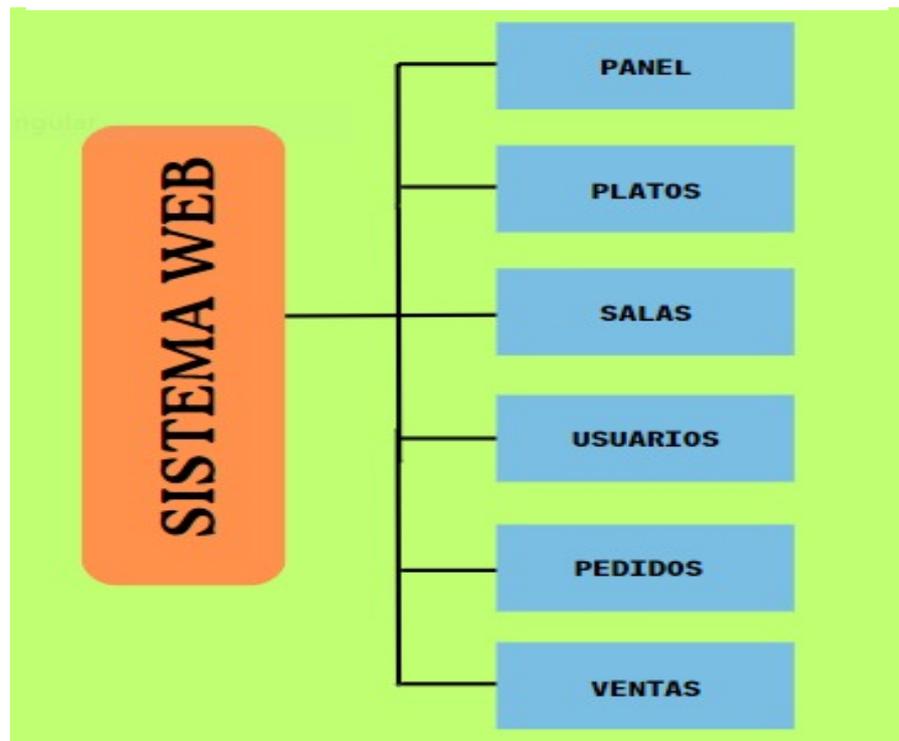


Figura 20. Menú principal - Opción Panel

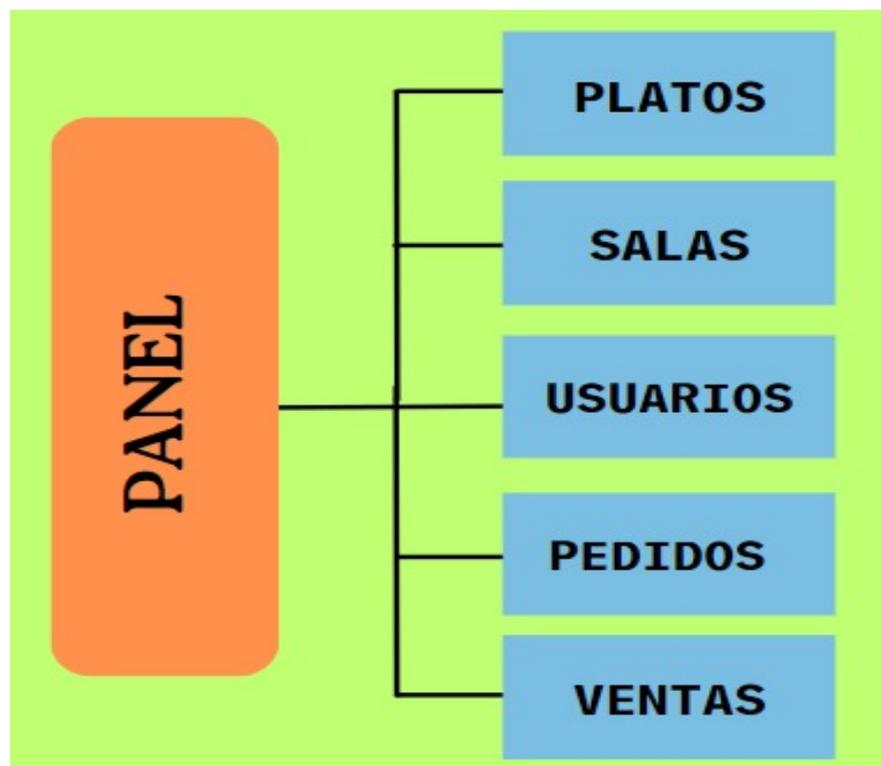


Figura 19. Menú principal - Opción Platos

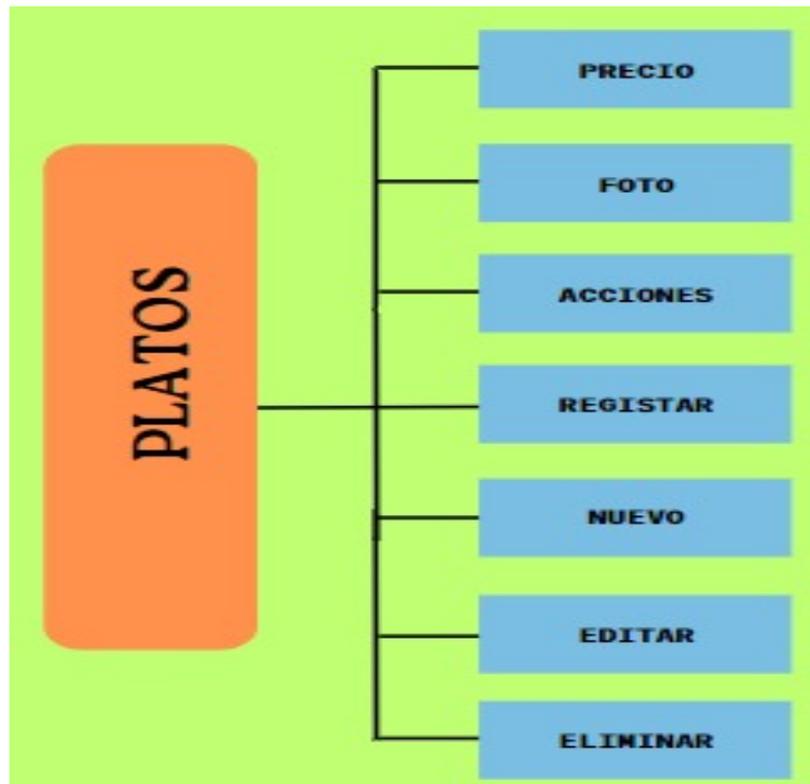


Figura 20. Menú principal - Opción Salas



Figura 21. Menú principal - Opción Ajustes

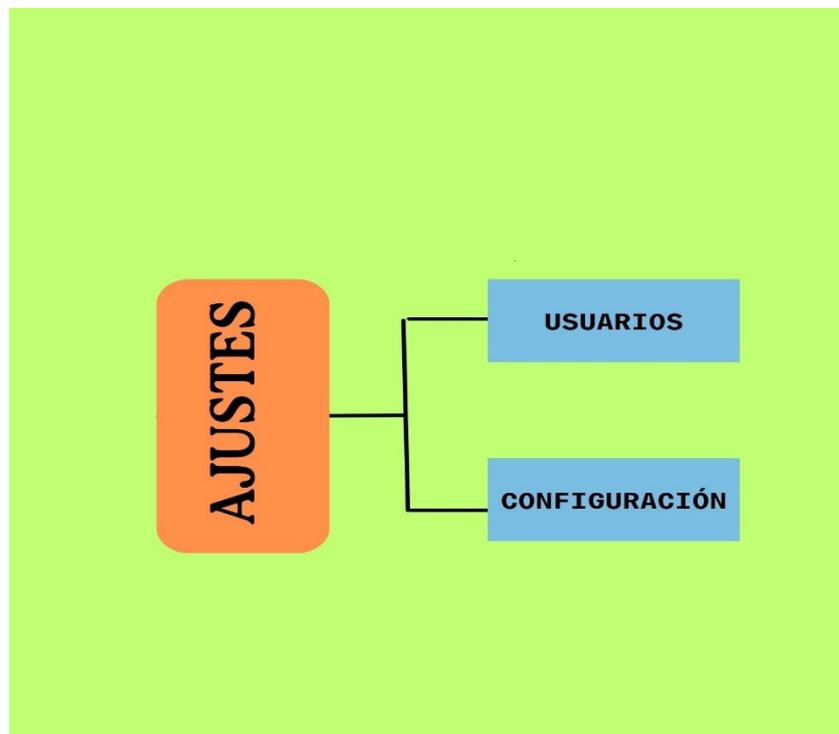
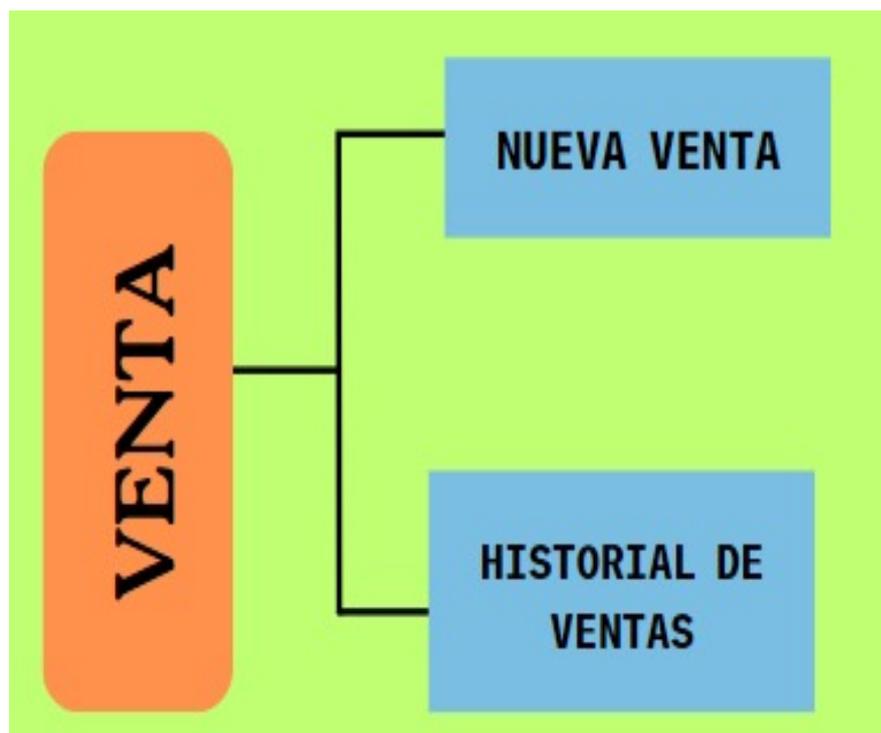


Figura 22. Menú principal - Opción Venta



3.3.3.3 Descripción General de la Estructura del Sistema web

INGRESO AL SISTEMA WEB PARA EL CONTROL DE LA POLLERIA EL RICO POLLO.

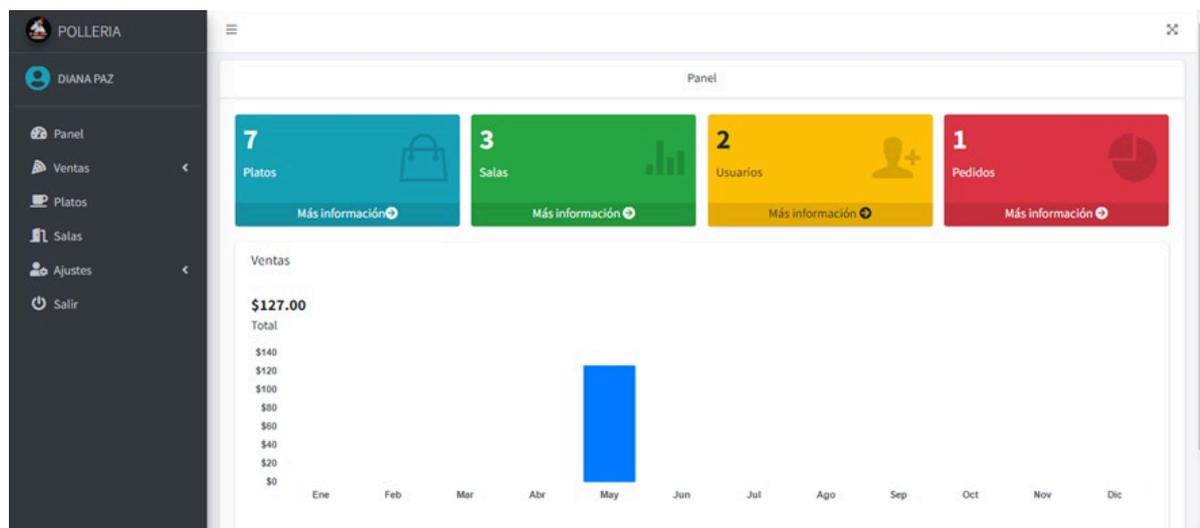
Escribir la siguiente dirección, para ingresar al sistema web:

Figura 23. Login del sistema web

<http://localhost/Sistema%20de%20orden%20polleria/>

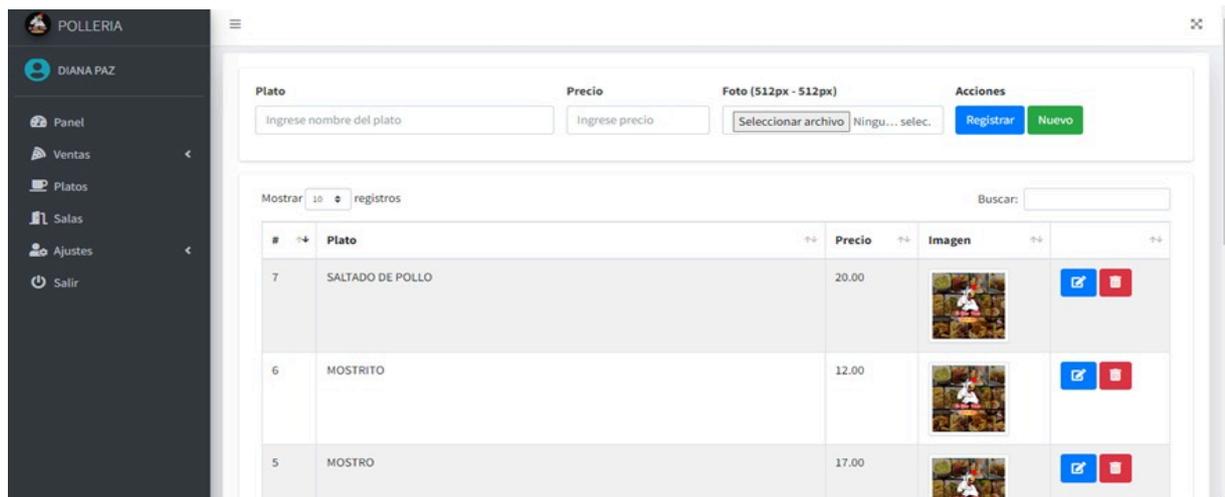
Luego se visualiza la siguiente ventana:

Figura 24. Vista de la ventana principal del usuario - administrador



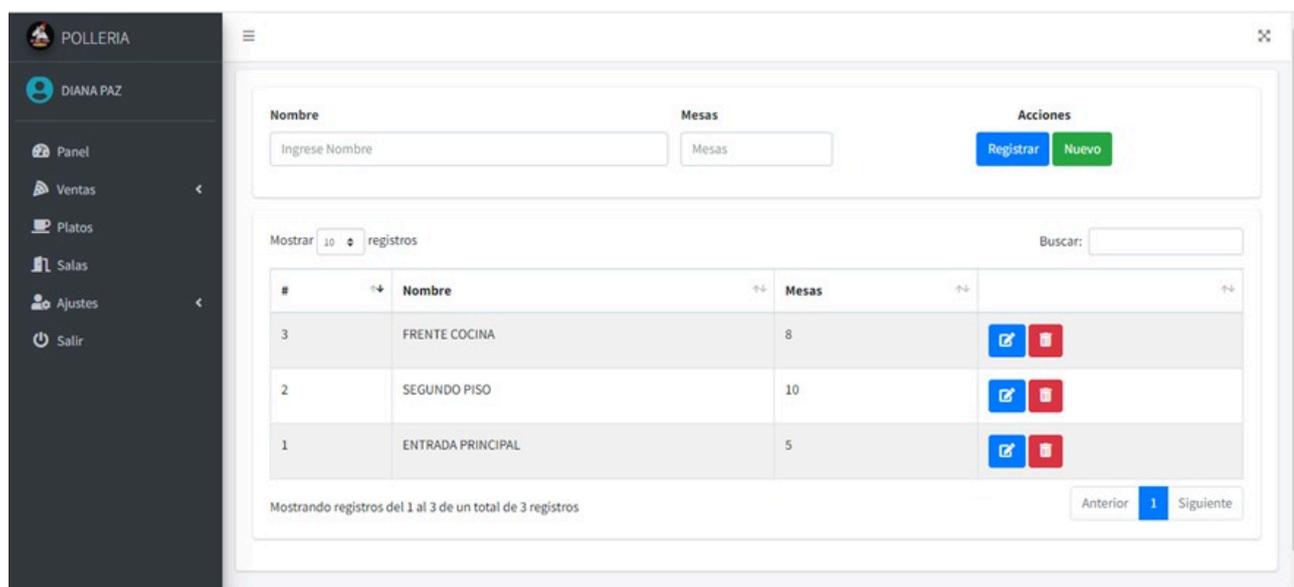
Nota: Opción Panel: Esta opción contiene 4 sub opciones:

Figura 25. Platos



Nota: Esta sub opción permite visualizar los platos a la carta que se le ofrecen al público.

Figura 26. Descripción de Salas



Nota: Esta sub opción permite ver las mesas que contiene cada área.

Figura 27. Usuarios

Nombre: Ingrese Nombre
 Correo: Ingrese correo Electrónico
 Rol: Seleccionar
 Contraseña: Ingrese Contraseña

Registrar Nuevo

Mostrar 10 registros

#	Nombre	Correo	Rol
2	ROBERTO SILVA	mozo@gmail.com	Mozo
1	DIANA PAZ	admin@gmail.com	Administrador

Mostrando registros del 1 al 2 de un total de 2 registros

Derechos de autor © 2023 EL RICO POLLO. Reservados todos los derechos. Version 3.2.0

Nota: Esta sub opción permite modificar el usuario y visualizar sus datos.

Figura 28. Pedidos

Historial pedidos

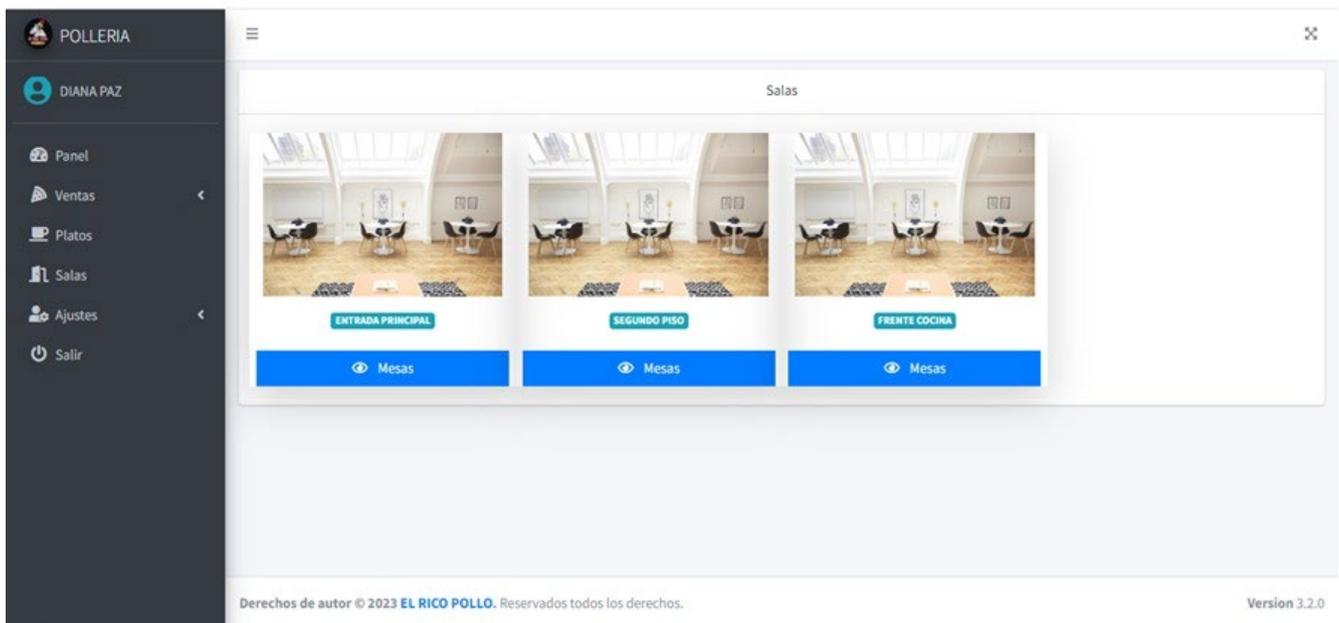
Mostrar 10 registros

#	Sala	Mesa	Fecha	Total	Usuario
4	SEGUNDO PISO	10	2023-05-25 15:03:11	25.00	DIANA PAZ
3	FRENTE COCINA	5	2023-05-25 15:04:17	10.00	DIANA PAZ
2	FRENTE COCINA	3	2023-05-25 15:03:43	33.00	DIANA PAZ
1	ENTRADA PRINCIPAL	1	2023-05-25 15:03:27	59.00	DIANA PAZ

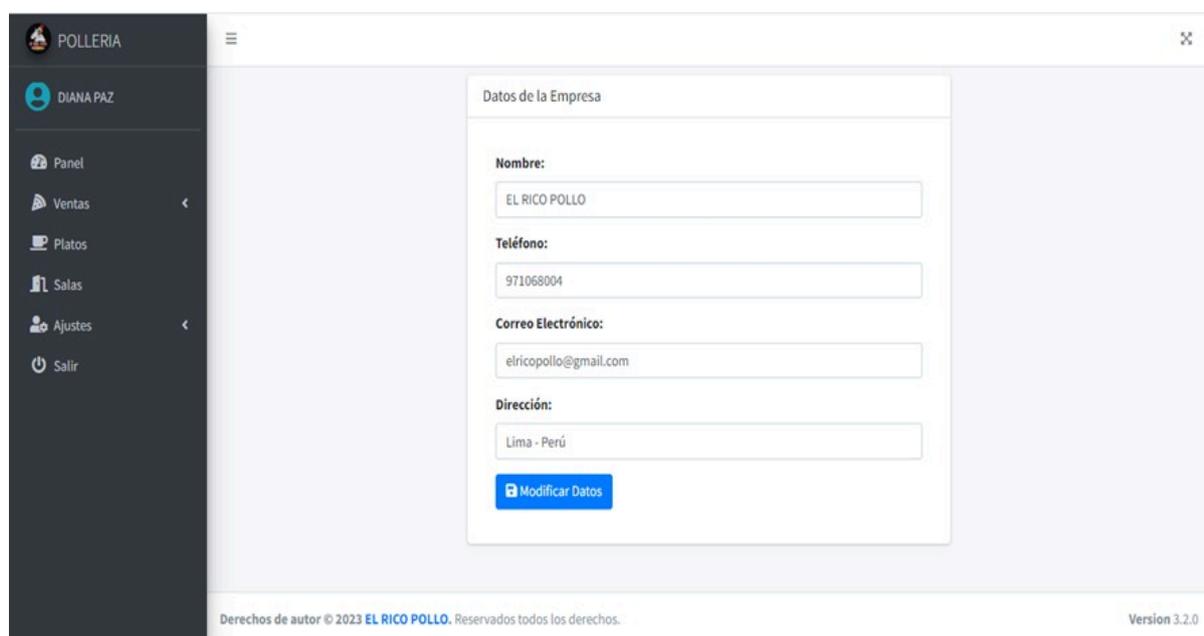
Mostrando registros del 1 al 4 de un total de 4 registros

Derechos de autor © 2023 EL RICO POLLO. Reservados todos los derechos. Version 3.2.0

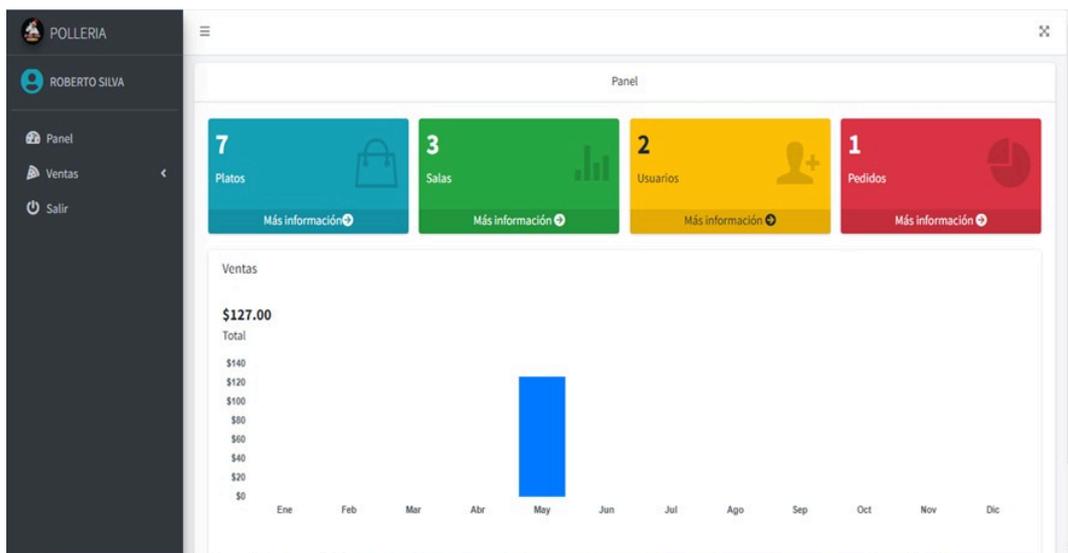
Nota: Esta sub opción permite visualizar los pedidos generados en el día.

Figura 29. Opción Mesas

Nota: Esta opción muestra las salas del establecimiento y mesas disponibles.

Figura 30. Opción Ajustes / Salir

Nota: Esta opción permite modificar datos de la empresa y cerrar sesión.

Figura 31.*Inicio*

Nota: segundo usuario ingresara al sistema desde su cuenta.

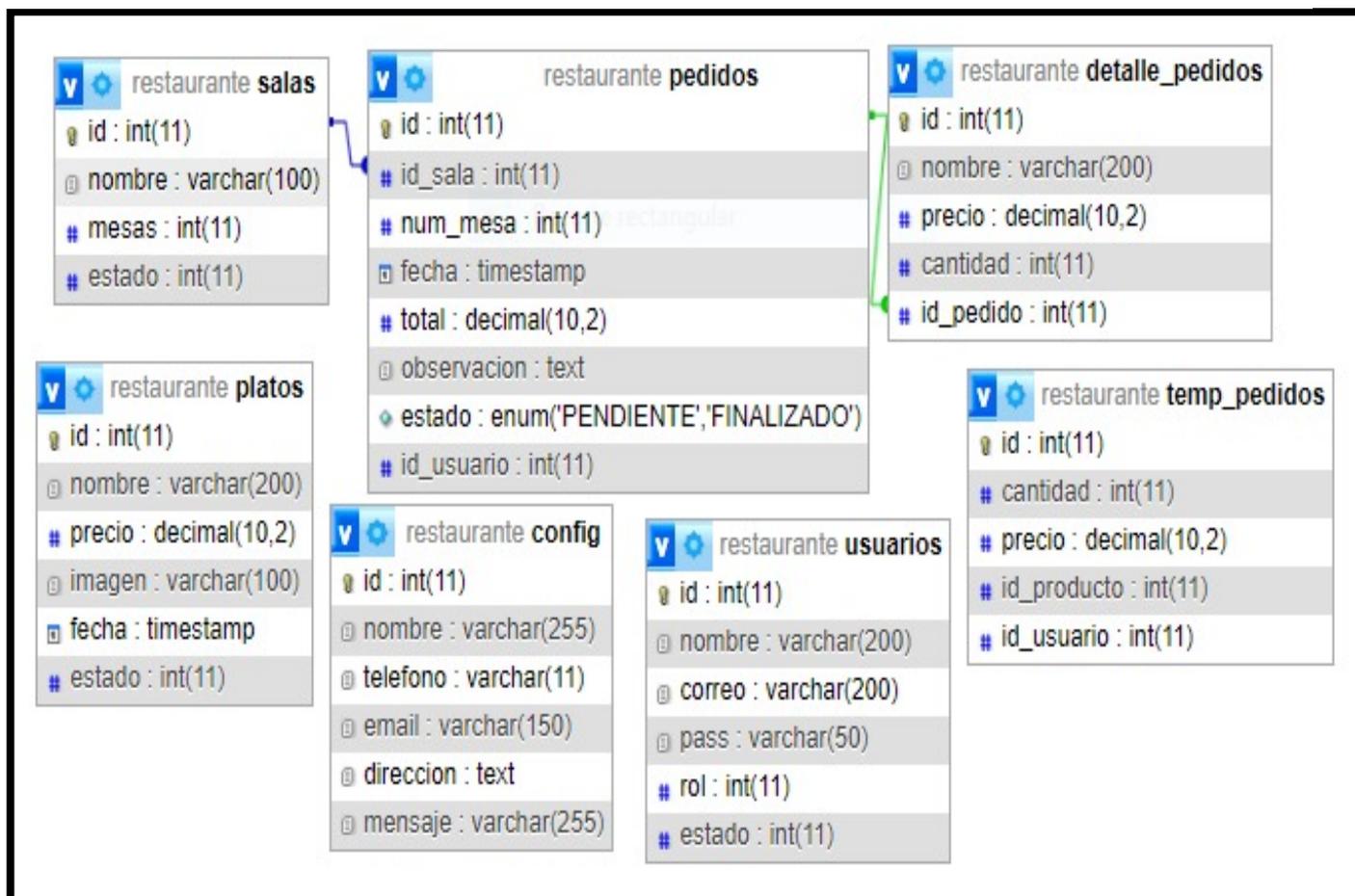
Figura 32.*Vista de la ventana principal del usuario - Mozo*

The login page for 'EL RICO POLLO' has a light gray background. At the top, the text 'EL RICO POLLO' is displayed in a large, dark font. Below it, the subtitle 'Inicia sesión para iniciar tu sesión' is centered. The login form consists of two input fields: the first is for the email address, containing 'mozo@gmail.com', and the second is for the password, represented by a series of dots. To the right of each input field is an icon: an envelope for email and a padlock for password. Below the input fields is a prominent blue button with the text 'INICIAR SESIÓN' in white capital letters.

Nota: Como se puede observar el mozo solo tiene acceso para generar las ventas.

3.3.3.4 Flujo de Base de Datos en MySQL

Figura 33. Diagrama de base de datos



3.3.3.5 Representación descriptiva de las tablas de la base de datos

Figura 34. Tabla: platos

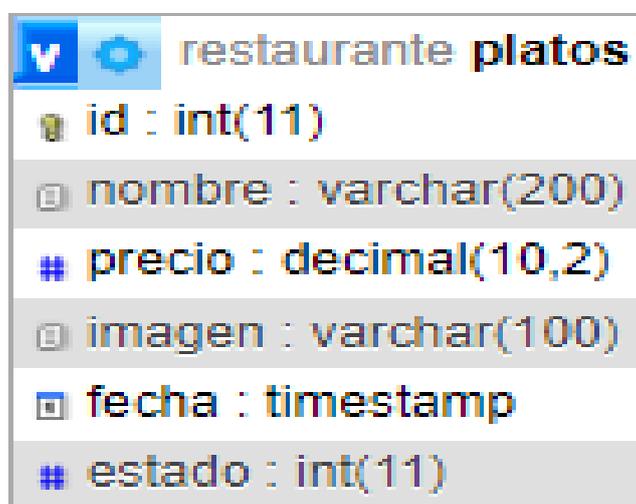


Figura 36.Tabla: Salas

restaurantes salas	
	id : int(11)
	nombre : varchar(100)
	mesas : int(11)
	estado : int(11)

Figura 35.Tabla configuración

restaurantes config	
	id : int(11)
	nombre : varchar(255)
	telefono : varchar(11)
	email : varchar(150)
	direccion : text
	mensaje : varchar(255)

Figura 39.Tabla Pedidos

restaurantes pedidos	
	id : int(11)
	id_sala : int(11)
	num_mesa : int(11)
	fecha : timestamp
	total : decimal(10,2)
	observacion : text
	estado : enum('PENDIENTE','FINALIZADO')
	id_usuario : int(11)

Figura 40. Tabla Usuarios

v restaurante usuarios	
🔑	id : int(11)
📄	nombre : varchar(200)
📄	correo : varchar(200)
📄	pass : varchar(50)
#	rol : int(11)
#	estado : int(11)

Figura 37. Temp_pedidos

v restaurante temp_pedidos	
🔑	id : int(11)
#	cantidad : int(11)
#	precio : decimal(10,2)
#	id_producto : int(11)
#	id_usuario : int(11)

Figura 42. Tabla detalle pedidos

v restaurante detalle_pedidos	
🔑	id : int(11)
📄	nombre : varchar(200)
#	precio : decimal(10,2)
#	cantidad : int(11)
#	id_pedido : int(11)

3.3.4 Codificación del sistema

Creación de la base de datos

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
-- Volcado de datos para la tabla `config`
--

INSERT INTO `config` (`id`, `nombre`, `telefono`, `email`, `direccion`, `mensaje`) VALUES
(1, 'POLLERIA', '971068004', 'elricopollo@gmail.com', 'Lima - Perú', 'Gracias por la compra');

-----

--
-- Estructura de tabla para la tabla `detalle_pedidos`
--

CREATE TABLE `detalle_pedidos` (
  `id` int(11) NOT NULL,
  `nombre` varchar(200) NOT NULL,
  `precio` decimal(10,2) NOT NULL,
  `cantidad` int(11) NOT NULL,
  `id_pedido` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
-- Volcado de datos para la tabla `detalle_pedidos`
--

INSERT INTO `detalle_pedidos` (`id`, `nombre`, `precio`, `cantidad`, `id_pedido`) VALUES
(1, 'MOSTRO', '10.00', 1, 1),
(2, 'MOSTRITO', '25.00', 1, 1),
(3, 'SALTADO DE POLLO', '8.00', 3, 1),
(4, '1/4 POLLO + PAPAS FRITAS + ENSALADA + CREMAS + GASEOSA', '25.00', 1, 2),
(5, 'MOLLEJITAS', '8.00', 1, 2),
(6, '1/2 POLLO + PAPAS FRITAS + CREMAS + GASEOSA + ENSALADA', '10.00', 1, 3),
(7, '1 POLLO ENTERO + PAPAS FRITAS + CREMAS + ENSALADA + GASEOSA', '25.00', 1, 4);

```

```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 25-05-2023 a las 22:11:41
-- Versión del servidor: 10.4.27-MariaDB
-- Versión de PHP: 8.2.0

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `restaurante`
--

-----

--
-- Estructura de tabla para la tabla `config`
--

CREATE TABLE `config` (
  `id` int(11) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  `telefono` varchar(11) NOT NULL,
  `email` varchar(150) NOT NULL,
  `direccion` text NOT NULL,
  `mensaje` varchar(255) NOT NULL
```

```
-- Estructura de tabla para la tabla `pedidos`
```

```
--
```

```
CREATE TABLE `pedidos` (  
  `id` int(11) NOT NULL,  
  `id_sala` int(11) NOT NULL,  
  `num_mesa` int(11) NOT NULL,  
  `fecha` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),  
  `total` decimal(10,2) NOT NULL,  
  `observacion` text DEFAULT NULL,  
  `estado` enum('PENDIENTE','FINALIZADO') NOT NULL DEFAULT 'PENDIENTE',  
  `id_usuario` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

```
-- Volcado de datos para la tabla `pedidos`
```

```
--
```

```
INSERT INTO `pedidos` (`id`, `id_sala`, `num_mesa`, `fecha`, `total`, `observacion`, `estado`, `id_usuario`) VALUES  
(1, 1, 1, '2023-05-25 20:03:27', '59.00', '', 'FINALIZADO', 1),  
(2, 3, 3, '2023-05-25 20:03:43', '33.00', '', 'FINALIZADO', 1),  
(3, 3, 5, '2023-05-25 20:04:17', '10.00', '', 'FINALIZADO', 1),  
(4, 2, 10, '2023-05-25 20:03:11', '25.00', '', 'PENDIENTE', 1);
```

```
-- .....
```

```
-- Estructura de tabla para la tabla `platos`
```

```
CREATE TABLE `platos` (  
  `id` int(11) NOT NULL,  
  `nombre` varchar(200) NOT NULL,  
  `precio` decimal(10,2) NOT NULL,  
  `imagen` varchar(100) DEFAULT NULL,  
  `fecha` timestamp NULL DEFAULT NULL,  
  `estado` int(11) NOT NULL DEFAULT 1  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

```
-- Volcado de datos para la tabla `platos`
--

INSERT INTO `platos` (`id`, `nombre`, `precio`, `imagen`, `fecha`, `estado`) VALUES
(1, 'MOSTRO', '10.00', '', NULL, 1),
(2, 'MOSTRITO', '25.00', '', NULL, 1),
(3, 'SALTADO DE POLLO POLLO', '8.00', '', NULL, 1);

-- .....

--
-- Estructura de tabla para la tabla `salas`
--

CREATE TABLE `salas` (
  `id` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `mesas` int(11) NOT NULL,
  `estado` int(11) NOT NULL DEFAULT 1
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
-- Volcado de datos para la tabla `salas`
--

INSERT INTO `salas` (`id`, `nombre`, `mesas`, `estado`) VALUES
(1, 'ENTRADA PRINCIPAL', 5, 1),
(2, 'SEGUNDO PISO', 10, 1),
(3, 'FRENTE COCINA', 8, 1);

-- .....
```

```
-- Estructura de tabla para la tabla `temp_pedidos`
--

CREATE TABLE `temp_pedidos` (
  `id` int(11) NOT NULL,
  `cantidad` int(11) NOT NULL,
  `precio` decimal(10,2) NOT NULL,
  `id_producto` int(11) NOT NULL,
  `id_usuario` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-----

--
-- Estructura de tabla para la tabla `usuarios`
--

CREATE TABLE `usuarios` (
  `id` int(11) NOT NULL,
  `nombre` varchar(200) NOT NULL,
  `correo` varchar(200) NOT NULL,
  `pass` varchar(50) NOT NULL,
  `rol` int(11) NOT NULL,
  `estado` int(11) NOT NULL DEFAULT 1
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
-- Volcado de datos para la tabla `usuarios`
--

INSERT INTO `usuarios` (`id`, `nombre`, `correo`, `pass`, `rol`, `estado`) VALUES
(1, 'DIANA PAZ', 'admin@gmail.com', '21232f297a57a5a743894a0e4a801fc3', 1, 1);

--
-- Índices para tablas volcadas
--
```

```
-- Indices de la tabla `config`  
--  
ALTER TABLE `config`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- Indices de la tabla `detalle_pedidos`  
--  
ALTER TABLE `detalle_pedidos`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `id_pedido` (`id_pedido`);  
  
--  
-- Indices de la tabla `pedidos`  
--  
ALTER TABLE `pedidos`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `id_sala` (`id_sala`);  
  
--  
-- Indices de la tabla `platos`  
--  
ALTER TABLE `platos`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- Indices de la tabla `salas`  
--  
ALTER TABLE `salas`  
  ADD PRIMARY KEY (`id`);  
  
--
```

```
|  
-- Indices de la tabla `temp_pedidos`  
--  
ALTER TABLE `temp_pedidos`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- Indices de la tabla `usuarios`  
--  
ALTER TABLE `usuarios`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- AUTO_INCREMENT de las tablas volcadas  
--  
  
--  
-- AUTO_INCREMENT de la tabla `config`  
--  
ALTER TABLE `config`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
  
--  
-- AUTO_INCREMENT de la tabla `detalle_pedidos`  
--  
ALTER TABLE `detalle_pedidos`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;  
  
--  
-- AUTO_INCREMENT de la tabla `pedidos`  
--  
ALTER TABLE `pedidos`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;  
  
--
```

```
-- AUTO_INCREMENT de la tabla `platos`
--
ALTER TABLE `platos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT de la tabla `salas`
--
ALTER TABLE `salas`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT de la tabla `temp_pedidos`
--
ALTER TABLE `temp_pedidos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;

--
-- AUTO_INCREMENT de la tabla `usuarios`
--
ALTER TABLE `usuarios`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `detalle_pedidos`
--
ALTER TABLE `detalle_pedidos`
  ADD CONSTRAINT `detalle_pedidos_ibfk_1` FOREIGN KEY (`id_pedido`) REFERENCES `pedidos` (`id`);

--
-- Filtros para la tabla `pedidos`
--
```

HTML_INDEX

```
<?php
```

```
session_start();
```

```
if (!empty($_SESSION['active'])) {
```

```
    header('location: src/');
```

```
} else {
```

```
    if (!empty($_POST)) {
```

```
        $alert = "";
```

```
        if (empty($_POST['correo']) || empty($_POST['pass'])) {
```

```
            $alert = '<div class="alert alert-warning alert-dismissible fade show" role="alert">
```

```
                Ingrese correo y contraseña
```

```
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
```

```
                    <span aria-hidden="true">&times;</span>
```

```
                </button>
```

```
        header('Location: src/dashboard.php');
    } else {
        $alert = '<div class="alert alert-danger alert-dismissible fade
show" role="alert">
            Contraseña incorrecta
            <button type="button" class="close" data-
dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>';
        session_destroy();
    }
}
}
?>
```

```
        </div>';  
    } else {  
        require_once "conexion.php";  
        $user = mysqli_real_escape_string($conexion,  
$_POST['correo']);  
        $pass = md5(mysqli_real_escape_string($conexion,  
$_POST['pass']));  
        $query = mysqli_query($conexion, "SELECT * FROM usuarios  
WHERE correo = '$user' AND pass = '$pass'");  
        mysqli_close($conexion);  
        $resultado = mysqli_num_rows($query);  
        if ($resultado > 0) {  
            $dato = mysqli_fetch_array($query);  
            $_SESSION['active'] = true;  
            $_SESSION['idUser'] = $dato['id'];  
            $_SESSION['nombre'] = $dato['nombre'];  
            $_SESSION['rol'] = $dato['rol'];  
        }  
    }  
}
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-  
scale=1">  
  <title>Login</title>  
  
  <!-- Google Font: Source Sans Pro -->  
  <link rel="stylesheet"  
href="https://fonts.googleapis.com/css?fa-  
mily=Source+Sans+Pro:300,400,400i,700&display=fallback">  
  <!-- Font Awesome -->  
  <link rel="stylesheet" href="assets/plugins/fontawesome-  
free/css/all.min.css">  
  <!-- icode bootstrap -->
```

```
<link rel="stylesheet" href="assets/plugins/icheck-
bootstrap/icheck-bootstrap.min.css">

<!-- Theme style -->

<link rel="stylesheet" href="assets/dist/css/adminlte.min.css">

</head>

<body class="hold-transition login-page">
<div class="login-box">
  <div class="login-logo">

    <a href="#"><b>EL RICO </b>POLLO</a>

  </div>

  <!-- /.login-logo -->

  <div class="card">

    <div class="card-body login-card-body">

      <p class="login-box-msg">Inicia sesión para iniciar tu sesión</p>
```

```
<form action="" method="post" autocomplete="off">
<?php echo (isset($alert)) ? $alert : " ; ?>
<div class="input-group mb-3">
    <input type="email" class="form-control" name="correo" placeholder="CORREO">
    <div class="input-group-append">
        <div class="input-group-text">
            <span class="fas fa-envelope"></span>
        </div>
    </div>
</div>
<div class="input-group mb-3">
    <input type="password" class="form-control" name="pass" placeholder="PASSWORD">
    <div class="input-group-append">
```

```
<div class="input-group-text">
  <span class="fas fa-lock"></span>
</div>
</div>
</div>
<div class="row">
  <!-- /.col -->
  <div class="col-4">
    <button type="submit" class="btn btn-primary btn-
block">INICIAR SESIÓN</button>
  </div>
  <!-- /.col -->
</div>
</form>
</div>
<!-- /.login-card-body -->
```

```
</div>
```

```
</div>
```

```
<!-- /.login-box -->
```

```
<!-- jQuery -->
```

```
<script src="assets/plugins/jquery/jquery.min.js"></script>
```

```
<!-- Bootstrap 4 -->
```

```
<script  
src="assets/plu-  
gins/bootstrap/js/bootstrap.bundle.min.js"></script>
```

```
<!-- AdminLTE App -->
```

```
<script src="assets/dist/js/adminlte.min.js"></script>
```

```
</body>
```

```
</html>
```

PHP- CODIFICACION DEL PANEL

```
<?php
```

```
session_start();
```

```
include_once "includes/header.php";
```

```
include "../conexion.php";
```

```
$query1 = mysqli_query($conexion, "SELECT COUNT(id) AS total  
FROM salas WHERE estado = 1");
```

```
$totalSalas = mysqli_fetch_assoc($query1);
```

```
$query2 = mysqli_query($conexion, "SELECT COUNT(id) AS total  
FROM platos WHERE estado = 1");
```

```
$totalPlatos = mysqli_fetch_assoc($query2);
```

```
$query3 = mysqli_query($conexion, "SELECT COUNT(id) AS total  
FROM usuarios WHERE estado = 1");
```

```
$totalUsuarios = mysqli_fetch_assoc($query3);
```

```
$query4 = mysqli_query($conexion, "SELECT COUNT(id) AS total  
FROM pedidos WHERE estado = 1");
```

```
$totalPedidos = mysqli_fetch_assoc($query4);
```

```
$query5 = mysqli_query($conexion, "SELECT SUM(total) AS total  
FROM pedidos");
```

```
$totalVentas = mysqli_fetch_assoc($query5);
```

```
?>
```

```
<div class="card">
```

```
  <div class="card-header text-center">
```

```
    Panel
```

```
  </div>
```

```
  <div class="card-body">
```

```
    <div class="row">
```

```
      <div class="col-lg-3 col-6">
```

```
        <!-- small box -->
```

```
        <div class="small-box bg-info">
```

```
          <div class="inner">
```

```
            <h3><?php echo $totalPlatos['total']; ?></h3>
```

```

        <p>Platos</p>
    </div>
    <div class="icon">
        <i class="ion ion-bag"></i>
    </div>
    <a href="platos.php" class="small-box-footer">Más información<i class="fas fa-arrow-circle-right"></i></a>
</div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-success">
        <div class="inner">
            <h3><?php echo $totalSalas['total']; ?></h3>

```

```

        <p>Salas</p>
    </div>

    <div class="icon">
        <i class="ion ion-stats-bars"></i>
    </div>

    <a href="salas.php" class="small-box-footer">Más infor-
mación <i class="fas fa-arrow-circle-right"></i></a>

    </div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-warning">
        <div class="inner">
            <h3><?php echo $totalUsuarios['total']; ?></h3>

```

```

        <p>Usuarios</p>
    </div>
    <div class="icon">
        <i class="ion ion-person-add"></i>
    </div>
    <a href="usuarios.php" class="small-box-footer">Más in-
formación <i class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-6">
    <!-- small box -->
    <div class="small-box bg-danger">
        <div class="inner">
            <h3><?php echo $totalPedidos['total']; ?></h3>

```

```
<p>Pedidos</p>
```

```
</div>
```

```
<div class="icon">
```

```
<i class="ion ion-pie-graph"></i>
```

```
</div>
```

```
<a href="lista_ventas.php" class="small-box-footer">Más  
información <i class="fas fa-arrow-circle-right"></i></a>
```

```
</div>
```

```
</div>
```

```
<!-- ./col -->
```

```
</div>
```

```
<div class="row">
```

```
<div class="col-lg-12">
```

```
<div class="card">
```

```
<div class="card-header border-0">
```

```

<div class="d-flex justify-content-between">
  <h3 class="card-title">Ventas</h3>
</div>
</div>
<div class="card-body">
  <div class="d-flex">
    <p class="d-flex flex-column">
      <span class="text-bold text-lg">${<?php echo $totalVentas['total']; ?></span>
      <span>Total</span>
    </p>
  </div>
  <!-- /.d-flex -->

  <div class="position-relative mb-4">
    <canvas id="sales-chart" height="200"></canvas>

```

```
</div>
```

```
<?php include_once "includes/footer.php"; ?>
```

```
<script src="../../assets/js/dashboard.js"></script>
```

PHP- CODIFICACION DE PEDIDOS

```
<?php
session_start();
if ($_SESSION['rol'] == 1 || $_SESSION['rol'] == 3) {
    include_once "includes/header.php";
?>

<div class="card card-primary card-outline">
    <div class="card-header">
        <h3 class="card-title">
            <i class="fas fa-edit"></i>
            Platos
        </h3>
    </div>
    <div class="card-body">
        <div class="row">
            <div class="col-7 col-sm-9">
```

```
<div class="tab-content" id="vert-tabs-right-
tabContent">
```

```
<div class="tab-pane fade show active" id="vert-tabs-
right-home" role="tabpanel" aria-labelledby="vert-tabs-right-home-
tab">
```

```
<input type="hidden" id="id_sala" value="<?php
echo $_GET['id_sala'] ?>">
```

```
<input type="hidden" id="mesa" value="<?php echo
$_GET['mesa'] ?>">
```

```
<div class="row">
```

```
<?php
```

```
include "../conexion.php";
```

```
$query = mysqli_query($conexion, "SELECT *
FROM platos WHERE estado = 1");
```

```
$result = mysqli_num_rows($query);
```

```
if ($result > 0) {
```

```
while ($data = mysqli_fetch_assoc($query)) { ?>
```

```

<div class="col-md-3">
    <div class="col-12">
        
    </div>
    <h6 class="my-3"><?php echo $data['nom-
bre']; ?></h6>

<div class="bg-gray py-2 px-3 mt-4">
    <h2 class="mb-0">
        $<?php echo $data['precio']; ?>
    </h2>
</div>

<div class="mt-4">

```

```

        <a class="btn btn-primary btn-block btn-
flat addDetalle" href="#" data-id="<?php echo $data['id']; ?>">
            <i class="fas fa-cart-plus mr-2"></i>
            Agregar
        </a>
    </div>
</div>
<?php }
} ?>
</div>
</div>
<div class="tab-pane fade" id="pedido" role="tabpa-
nel" aria-labelledby="pedido-tab">
    <div class="row" id="detalle_pedido"></div>
    <hr>
    <div class="form-group">

```

```

        <label for="observacion">Observaciones</label>
        <textarea id="observacion" class="form-control"
rows="3" placeholder="Observaciones"></textarea>
    </div>
    <button class="btn btn-primary" type="button"
id="realizar_pedido">Realizar pedido</button>
</div>
</div>
</div>
<div class="col-5 col-sm-3">
    <div class="nav flex-column nav-tabs nav-tabs-right h-
100" id="vert-tabs-right-tab" role="tablist" aria-
orientation="vertical">
        <a class="nav-link active" id="vert-tabs-right-home-
tab" data-toggle="pill" href="#vert-tabs-right-home" role="tab" aria-
controls="vert-tabs-right-home" aria-selected="true">Platos</a>

```

```
        <a class="nav-link" id="pedido-tab" data-toggle="pill"
href="#pedido" role="tab" aria-controls="pedido" aria-
selected="false">Pedido</a>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- /.card -->
```

```
</div>
```

```
<?php include_once "includes/footer.php";
```

```
} else {
```

```
    header('Location: permisos.php');
```

```
}
```

```
?>
```

3.3.5 Pruebas del sistema

La prueba es el proceso de evaluar y validar el software para determinar si el sistema cumple con los requisitos del usuario.

Las pruebas realizadas son las siguientes:

- **Pruebas funcionales:** Estas pruebas se realizaron con bastante regularidad para garantizar que cada proceso se ejecutara según lo requerido.
- **Pruebas de usabilidad:** Estas pruebas se realizaron para verificar la facilidad de uso del al interactuar con sistemas web.
- **Pruebas de compatibilidad:** se ejecutó para garantizar que el sistema se muestre correctamente en navegadores web como Google Chrome y Microsoft Edge.

Figura 38. Vista del sistema web en el navegador Google Chrome

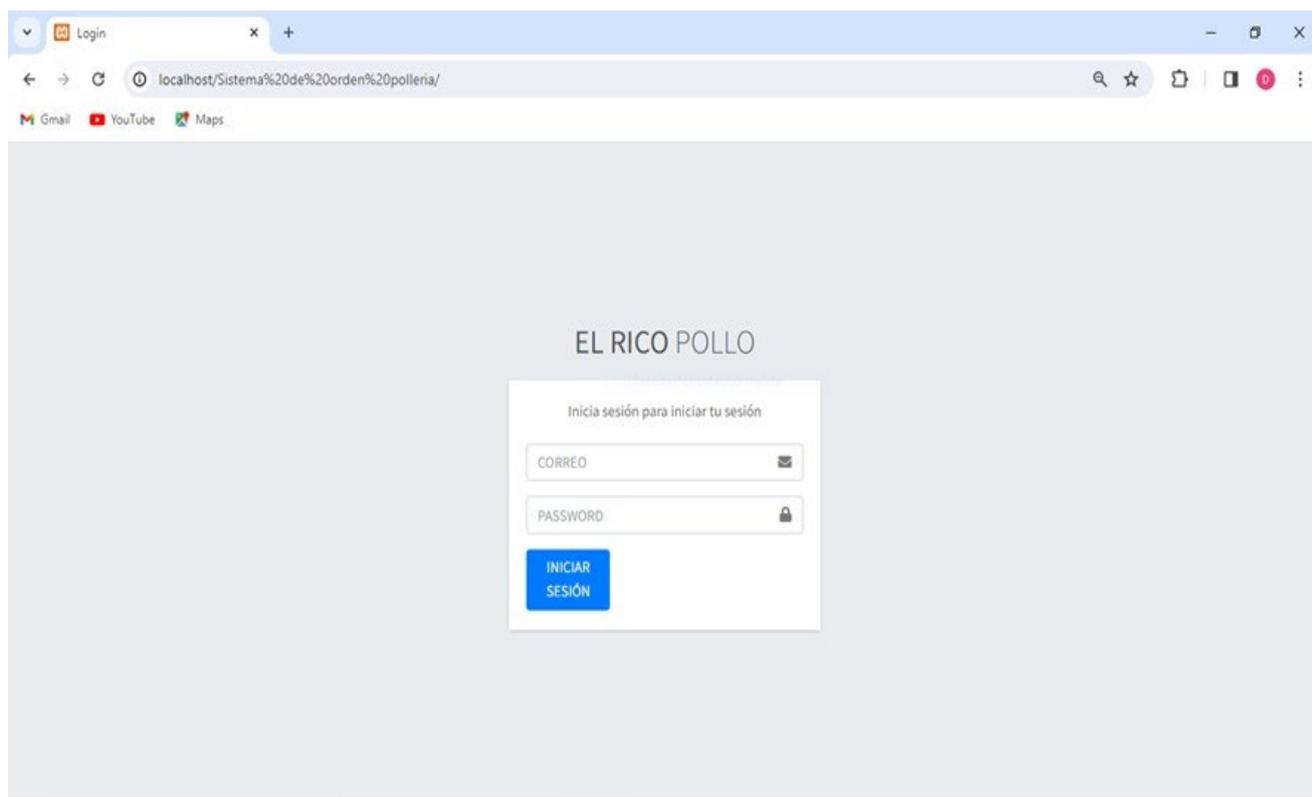
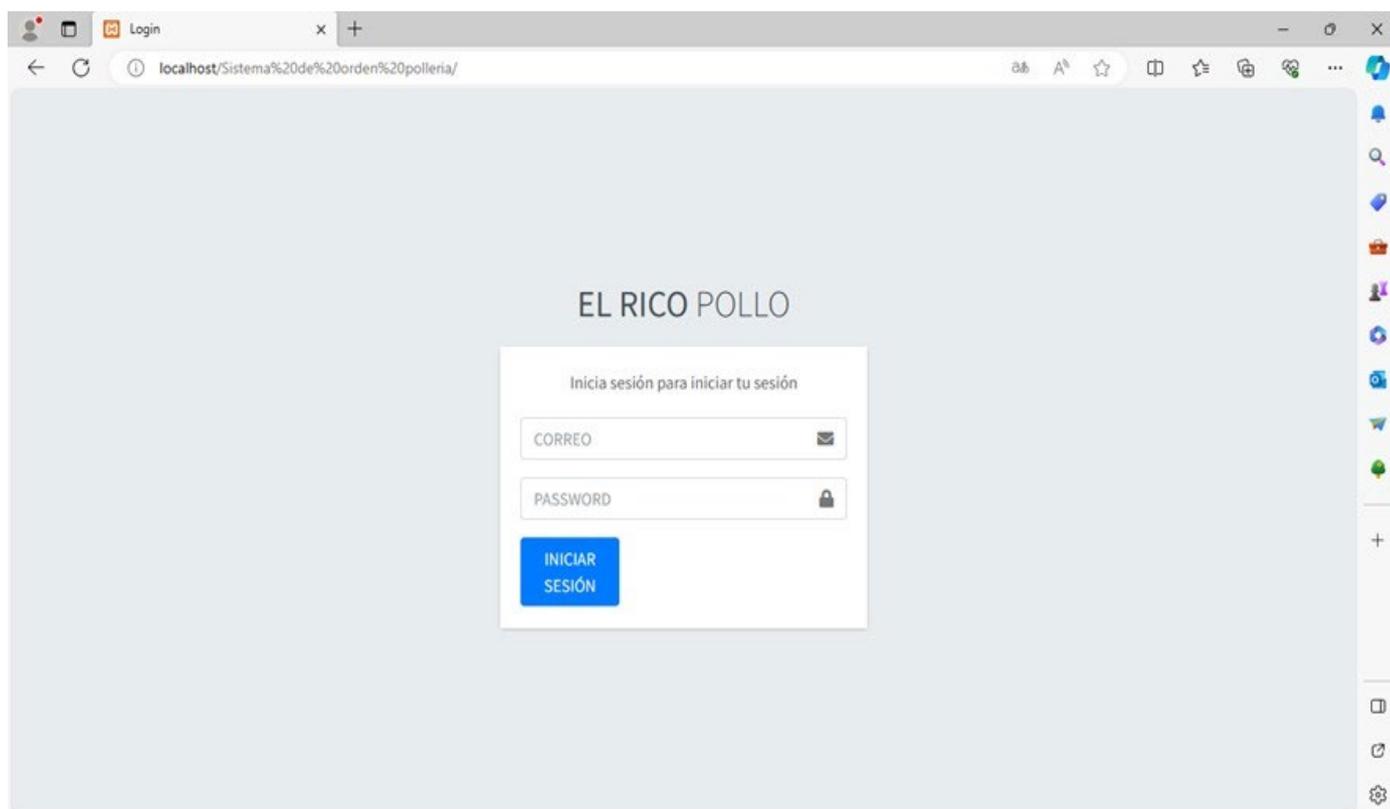


Figura 39. Vista del sistema web en el navegador Microsoft Edge



3.4 Actividades

Este trabajo se desarrolló en las siguientes etapas:

- **Levantamiento de la información:** En esta etapa, se entrevistó a los propietarios de la pollería basándose en un cuestionario previamente complementado y también se comprobaron los documentos utilizados por los propietarios.
- **Análisis de la información:** En este punto, el modelado de procesos se realizó utilizando el software Rational Rose y el lenguaje UML.
- **Diseño del sistema:** En esta etapa se creó el diseño y maquetación del sistema web y también se desarrolló una base de datos que contiene información sobre la pollería.
- **Codificación del sistema:** En este punto la programación de sistemas web se realizó utilizando el lenguaje PHP.
- **Pruebas del sistema:** Durante esta fase, se realizaron pruebas de funcionalidad, usabilidad y compatibilidad para resolver los errores que ocurrieran.
- **Implementación del sistema:** En esta etapa final, el sistema web está cargado en un hosting y listo para producción.

3.5 Limitaciones

- Al momento de subir el sistema web al hosting, hubo inconvenientes en algunas páginas las cuales no se mostraban los resultados correctamente, lo cual nos tomó tiempo en hacer las pruebas correspondientes.
- Dificultad en la entrega de la información manual para ejecutar el diseño con los campos que se requerían para la Base de Datos.

CAPÍTULO IV
RESULTADOS

RESULTADOS

- Se ha optimizado la gestión del servicio de la pollería y en general la atención al cliente, lo cual refleja el desarrollo como negocio y demuestra frente al público el uso de la tecnología.
- La información de la pollería se almacenó en una base de datos vinculada a un sistema web.
- El sistema web implementado cumple con los requisitos establecidos por el propietario de la pollería.
- Los usuarios finales el administrador como el mozo, les resulto optimo al registrar las ventas mientras interactuaban con el sistema web.
- Un factor importante en esta aplicación es que permitió dos grandes aspectos, el primero tiempo/ mano de obra el segundo se evita la perdida de información.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Se implementó un sistema de control de los pedidos de ventas satisfactoriamente, logrando tener el control de este proceso en el negocio que permite tener resultados exactos y oportunos permitiendo que como negocio de comida utilice la tecnología para sus procesos.
- Se registra la información de los pedidos de ventas en un software de Base de datos y se usaron herramientas tecnológicas para este sistema web que fue desarrollado siguiendo el procedimiento de desarrollo de software RUP utilizando el lenguaje UML e instrumentos CASE Rational Rose. Para desarrollar la base de información (datos) se utilizó el sistema de gestión de datos MySQL.
- Con esta implementación ya no se va perder la información porque se contará con un respaldo de seguridad del mismo.
- Con el registro en la Base de Datos, permite identificar los pedidos de ventas por día, por hora por apellidos de los clientes y por tipo de platos.
- Se realizó las pruebas funcionales y de rendimientos al sistema, así como en los diferentes navegadores Internet Explorer y Google Chrome.
- Finalmente, se logró el control del historial de los pedidos de ventas de la pollería El Rico Pollo.

RECOMENDACIONES

- a) Crear copias de seguridad de su base de datos y sistemas web.
- b) Capacitar a los beneficiarios en la operación del sistema web de la pollería para su adecuado manejo.
- c) Todas consultas del personal en relación al uso del sistema tratarlo con el administrador del sistema web.
- d) Cambiar cada tres meses las contraseñas de los usuarios del sistema web.
- e) Al surgir nuevos requisitos, se debe prestar atención a la fase de análisis, ya que es la fase principal del desarrollo de un sistema web.
- f) Para evitar accidentes y daños, se debe colocar los equipos informáticos en un lugar adecuado dentro de la pollería.

REFERENCIAS BIBLIOGRÁFICAS

Arimerics (s.f.) Que es Base de Datos. <https://www.arimerics.com/glosario-digital/base-de-Datos>

Barrios Salgado, C. D. (2020). *Desarrollo de una plataforma para la gestión de pedidos en los restaurantes de Sahagún córdoba* [Tesis de pregrado, título profesional ingeniería de sistemas y telecomunicaciones].Repositorio <https://repositorio.unicordoba.edu.co/server/api/core/bitstreams/671acb49-c707-4f19-9a58-fb212cc10e85/content>

Cappola M. (2022, junio 17). Qué es un servidor web, para qué sirve, cómo funciona y ejemplos. HubSpot. <https://blog.hubspot.es/website/que-es-servidor-web>

Cruz Naupay, R. R. (2021) *Chatbot para la gestión de pedidos en la pollería Nardos Chicken [tesis para obtener el título profesional de Ingeniero de Sistemas, Universidad Cesar Vallejo]* Repositorio <https://repositorio.ucv.edu.pe/handle/20.500.12692/80571>

De Souza (2019, junio 14). ¿Qué es un servidor web y para qué sirve en Internet? Rockcontent. <https://rockcontent.com/es/blog/que-es-un-servidor/>

Epitech Spain (2021, mayo, 19). ¿Qué es PHP en programación y para qué sirve este lenguaje de código abierto? <https://www.epitech-it.es/que-es-php/>

Gustavo B. (2023, enero, 11) ¿Qué es CSS? Hostinger tutoriales.<https://www.hostinger.mx/tutoriales/que-es-css>

Jesús (2022, abril, 25) Conoce qué es XAMPP y por qué deberías usarlo en tus proyectos. Dongee <https://www.dongee.com/tutoriales/que-es-xampp/>

Kiwop (2023, mayo, 23) *Frontend: La parte visual y funcional de una página web* <https://www.kiwop.com/blog/frontend>

Nex U. (s.f). *¿Qué es HTML? ¿Por qué necesitas saberlo?* <https://www.nextu.com/blog/que-es-html-rc22/>

RUP (2014, marzo, 14) *Fases Metodología RUP*.

<https://metodogiarupgrupo23.blogspot.com/2014/03/fases-metodologia-rup.html>

Robledano (2019, setiembre, 19). *Qué es MySQL: Características y ventajas*. OpenWebinars.

<https://openwebinars.net/blog/que-es-mysql/>

Sánchez Serpa, D. G. (2018). *Sistema de pedidos para la mejora de la administración de la pollería La Carreta 2018*. [Tesis de pregrado, título profesional de Ingeniero de Sistemas, Universidad Alas Peruanas]. Repositorio

https://repositorio.uap.edu.pe/bitstream/handle/20.500.12990/4153/Tesis_Sistema_Mejora_Administraci%c3%b3n.pdf?sequence=1&isAllowed=y

Schiaffarino A. (2019, marzo, 12) Modelo cliente servidor. Infranetworking

<https://blog.infranetworking.com/modelo-cliente-servidor/>

Vásquez Espinoza, M. R. (2021). *Implementación de un sistema informático para mejorar la atención de clientes en el restaurant pollería Solymar* [Tesis para optar el título profesional de ingeniera de sistemas y telemática]. Repositorio

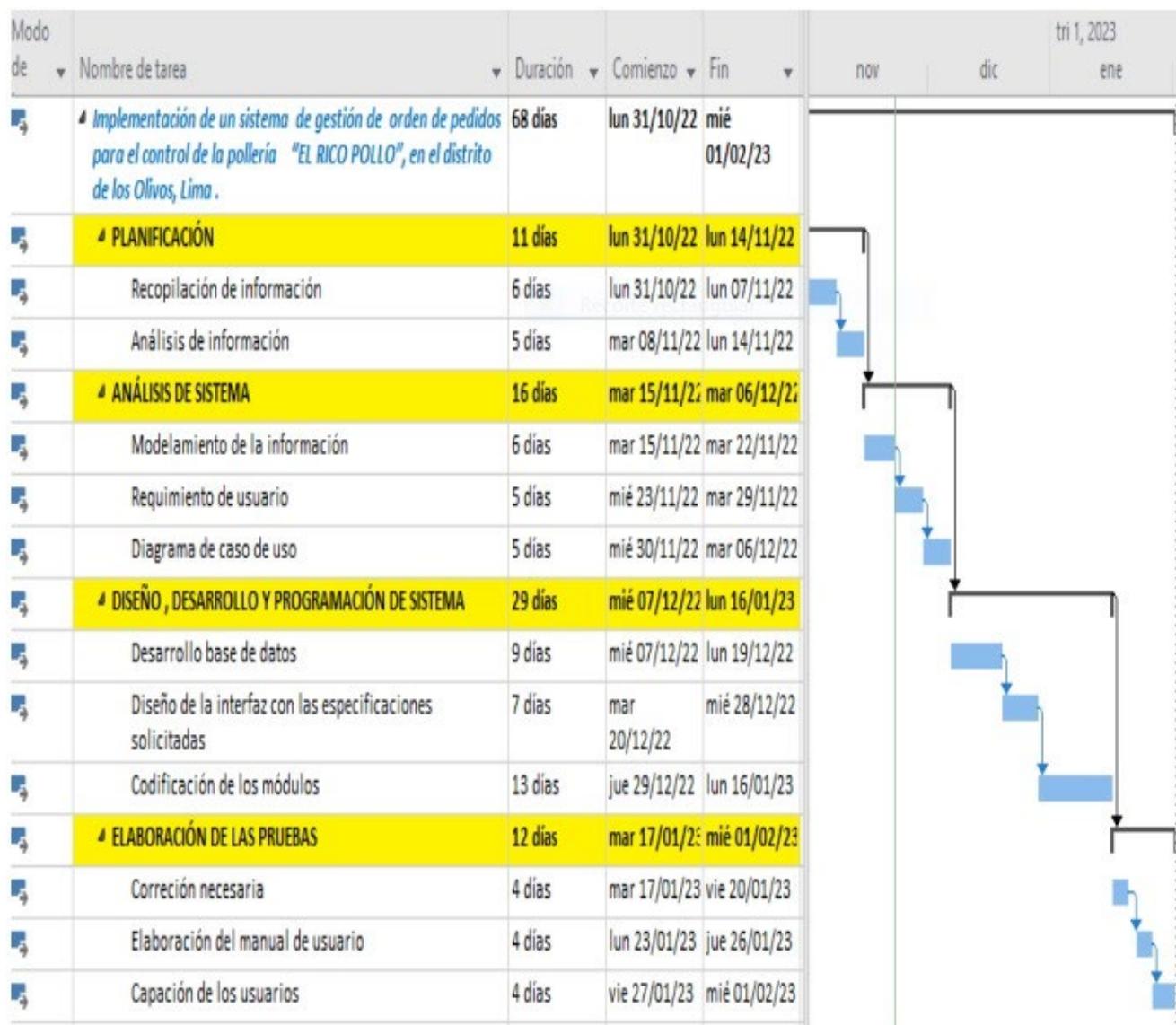
<https://repositorio.upa.edu.pe/handle/20.500.12897/189>

Valenzuela Salazar, G.N. (2018). *Gestión de control de pedidos con el apoyo del aplicativo móvil gochicken para la pollería el Gran Pollo Gordo E.I.R.L. Pisco* [Tesis de pregrado, título profesional de Ingeniero de Sistemas, Universidad Alas Peruanas]. Repositorio <https://repositorio.uap.edu.pe/handle/20.500.12990/7547>

Yshiy Ganmi, J. (2018) *Sistema de pedidos y la administración de la chifa Paraíso Oriental SAC - 2017* [Tesis de pregrado, título profesional de Ingeniero de Sistemas, Universidad Alas Peruanas]. Repositorio <https://repositorio.uap.edu.pe/xmlui/handle/20.500.12990/6522>

APÉNDICES

Apéndice A: Cronograma de Acciones(actividades)



Apéndice B: Estructura presupuestaria

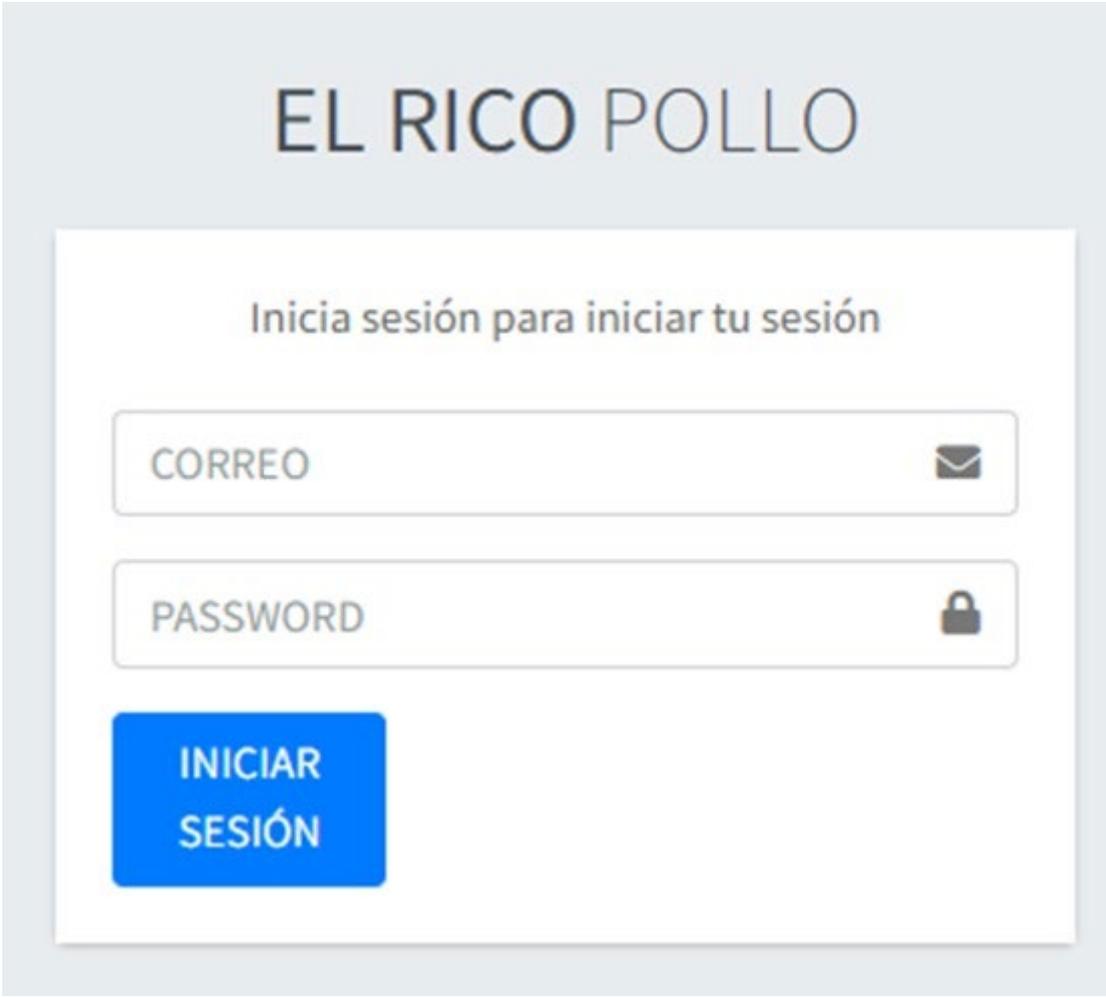
Materiales			
Detalle	Cantidad (Und)	Valor(S/.)	Total
Laptop i5	1	1900	1,900.00
Impresora	1	200	200.00
USBs 1Tb	1	100	100.00
Hojas(paquetes)	2	12	24.00
Tinta	2	30	60.00
subtotal			S/. 2,284.00
Servicios			
Detalle	Cantidad	Valor(S/.)	Total
Internet 15 MB	1	65	65.00
Movilidad Ida	2	50	100.00
Movilidad Vuelta	2	50	100.00
Otros		200	200.00
subtotal			S/. 465.00
Total			S/. 2,749.00

Apéndice C: Manual de Usuario

Ingreso al sistema web para el control de servicios de la pollería

Para ingresar al sistema web, se debe escribir la siguiente URL:

<http://localhost/Sistema%20de%20orden%20polleria/>



The image shows a login interface for a system named "EL RICO POLLO". The interface is centered on a light gray background. At the top, the text "EL RICO POLLO" is displayed in a large, dark blue font. Below this, the instruction "Inicia sesión para iniciar tu sesión" is written in a smaller, dark gray font. There are two input fields: the first is labeled "CORREO" and has an envelope icon on the right; the second is labeled "PASSWORD" and has a padlock icon on the right. Below these fields is a prominent blue button with the text "INICIAR SESIÓN" in white, bold, uppercase letters.

Ingresamos el usuario (admin), la contraseña (admin) cuenta del administrador:

EL RICO POLLO

Inicia sesión para iniciar tu sesión

admin@gmail.com

.....

INICIAR SESIÓN

Nota: Al ingresar el usuario y la contraseña se visualiza la siguiente ventana:

Opción salas: Esta opción permite crear y modificar el ambiente del establecimiento y sus mesas.

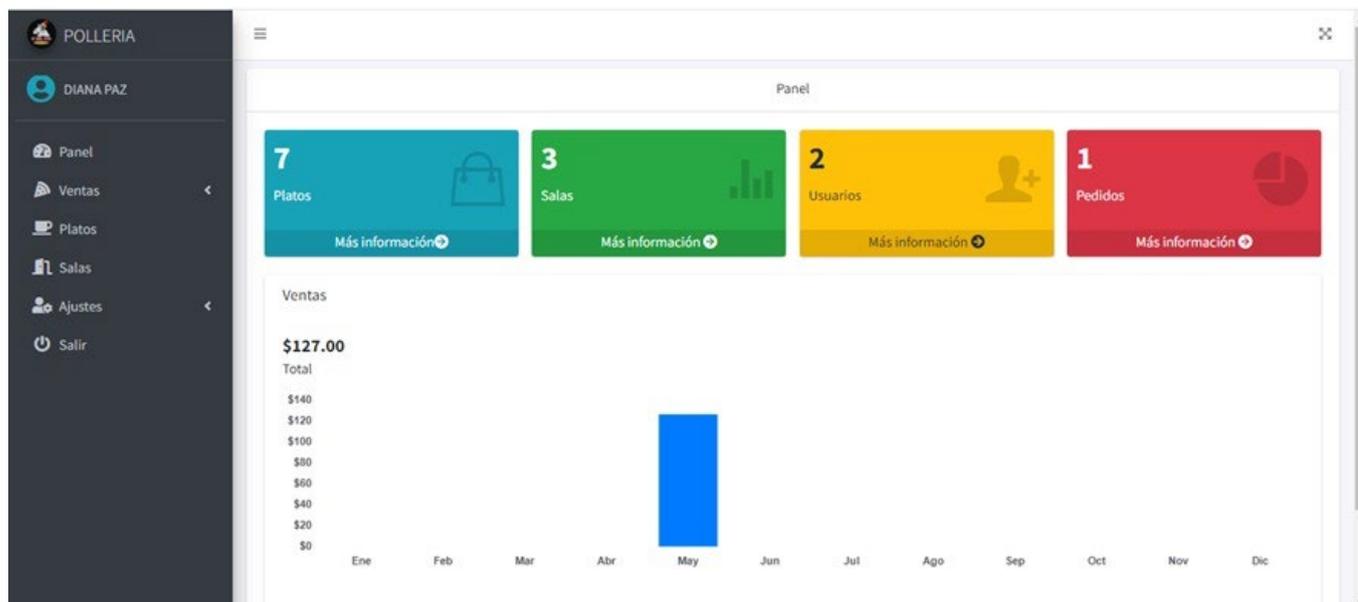
Salas

Mostrar registros Buscar:

#	Nombre	Mesas	
3	FRENTE COCINA	8	
2	SEGUNDO PISO	10	
1	ENTRADA PRINCIPAL	5	

Mostrando registros del 1 al 3 de un total de 3 registros Anterior **1** Siguiente

Iniciaremos creando los ambientes y mesas del establecimiento.



Opción platos: Esta opción contiene permite crear y modificar el plato del cliente.



Al realizar la venta se crea el plato deseado por el cliente.

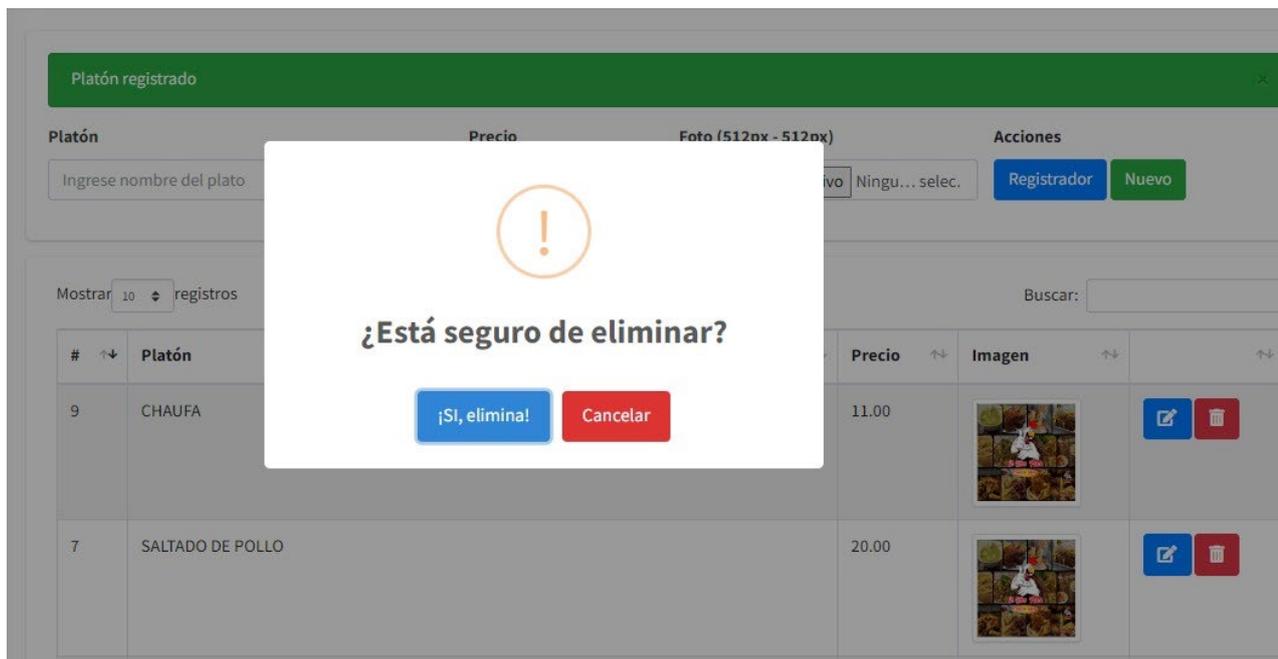
Formulario de gestión de platos:

Platón **Precio** **Foto (512px - 512px)** Ninguno... selec. **Acciones**

Mostrar registros

#	Platón	Precio	Imagen	Acciones
9	CHAUFA	11.00		<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
7	SALTADO DE POLLO	20.00		<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Al modificar el pedido se muestra una ventana de conformidad para poder editar.



Opción Ventas: Esta opción contiene dos sub opciones.



➤ **Nueva Venta:** Esta sub opción permite realizar la venta del día.

Salas



ENTRADA PRINCIPAL

Mesas



SEGUNDO PISO

Mesas



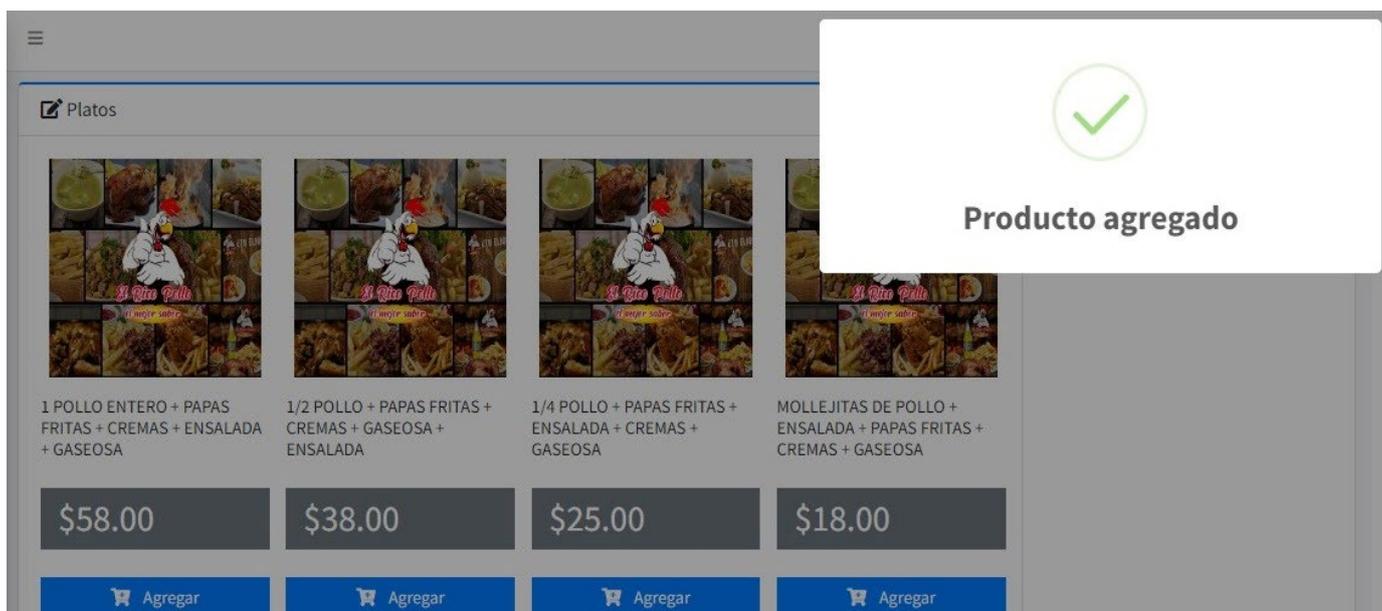
FRENTE COCINA

Mesas

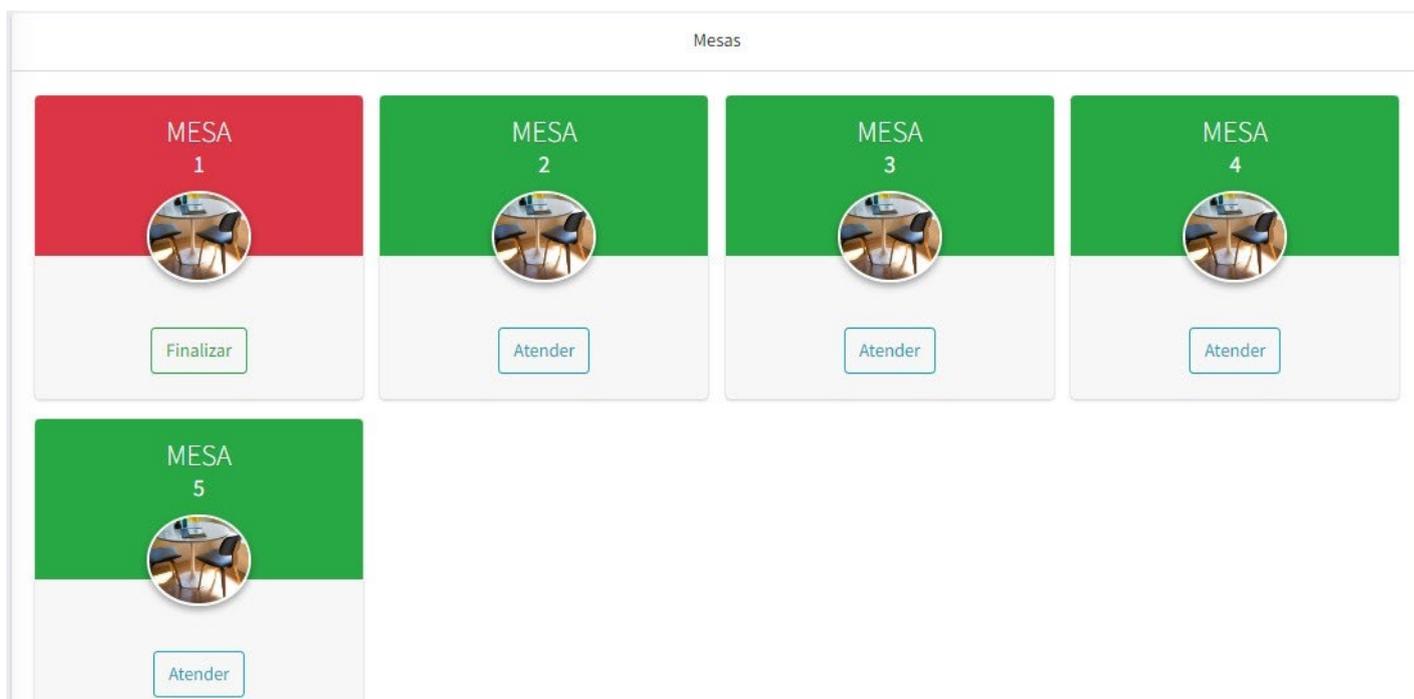
Platos

				Platos
1 POLLO ENTERO + PAPAS FRITAS + CREMAS + ENSALADA + GASEOSA	1/2 POLLO + PAPAS FRITAS + CREMAS + GASEOSA + ENSALADA	1/4 POLLO + PAPAS FRITAS + ENSALADA + CREMAS + GASEOSA	MOLLEJITAS DE POLLO + ENSALADA + PAPAS FRITAS + CREMAS + GASEOSA	Pedido
\$58.00	\$38.00	\$25.00	\$18.00	
Agregar	Agregar	Agregar	Agregar	

Al finalizar de crear los platos a gusto del cliente podemos AGREGAR.



Una vez agregado saldrá una ventana de conformidad.



Se puede observar que la mesa 1 está siendo atendida así mismo en el historial se observa que su pedido se encuentra PENDIENTE.

➤ **Historial de ventas:** Esta sub opción permite mostrar las ventas realizadas en el día.

Historial pedidos

Mostrar registros Buscar:

#	Sala	Mesa	Fecha	Total	Usuario	
4	SEGUNDO PISO	10	2023-05-25 15:03:11	25.00	DIANA PAZ	Pendiente
3	FRENTE COCINA	5	2023-05-25 15:04:17	10.00	DIANA PAZ	Completado
2	FRENTE COCINA	3	2023-05-25 15:03:43	33.00	DIANA PAZ	Completado
1	ENTRADA PRINCIPAL	1	2023-05-25 15:03:27	59.00	DIANA PAZ	Completado

Mostrando registros del 1 al 4 de un total de 4 registros Anterior **1** Siguiente

Fecha: 2023-12-18 06:56:01

Mesa: 1

\$278.00

Platos

Precio
58.00



1 POLLO ENTERO + PAPAS FRITAS + CREMAS +
ENSALADA + GASEOSA

Precio
25.00



1/4 POLLO + PAPAS FRITAS + ENSALADA + CREMAS
+ GASEOSA

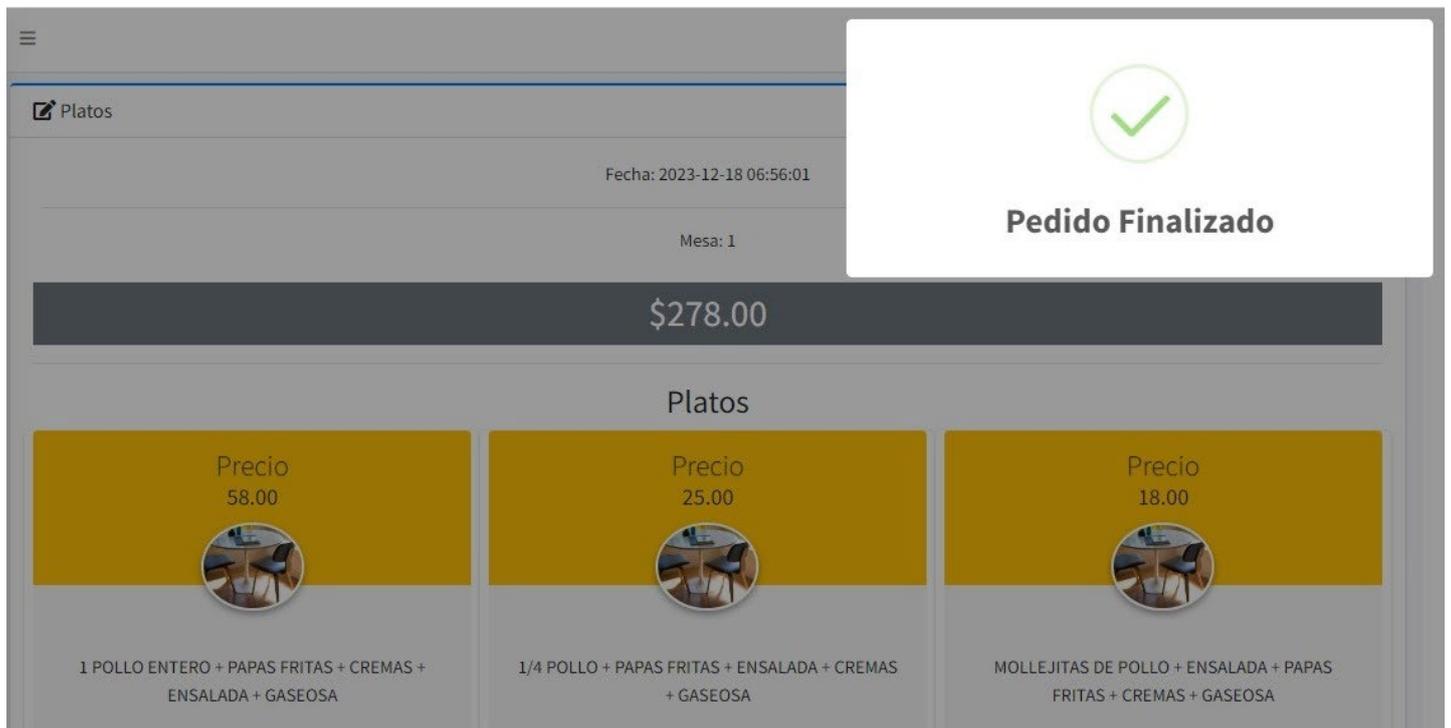
Precio
18.00



MOLLEJITAS DE POLLO + ENSALADA + PAPAS
FRITAS + CREMAS + GASEOSA

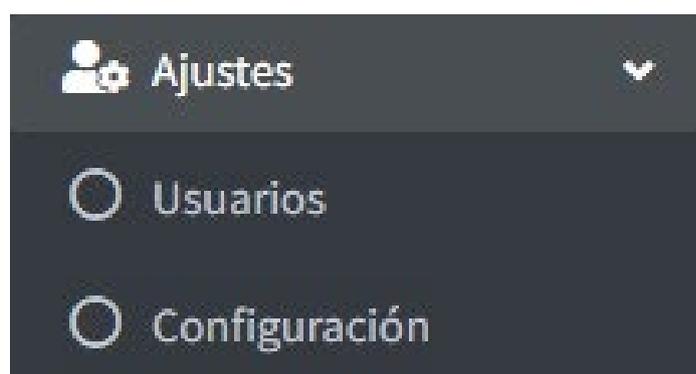
🛒 Finalizar

Finalizaremos indicando su monto a pagar.



Finalmente se ejecutó la compra

Opción Ajustes: Esta opción contiene dos sub opciones.



- **Usuarios:** Esta sub opción permite mostrar y agregar a nuevos usuarios.

Nombre

Correo

Rol

Contraseña

[Registrar](#)

[Nuevo](#)

Mostrar registros

Buscar:

#	Nombre	Correo	Rol	
2	ROBERTO SILVA	mozo@gmail.com	Mozo	<input type="checkbox"/> <input type="checkbox"/>
1	DIANA PAZ	admin@gmail.com	Administrador	<input type="checkbox"/> <input type="checkbox"/>

Mostrando registros del 1 al 2 de un total de 2 registros

[Anterior](#)
1
[Siguiete](#)

Se puede observar los usuarios que tienen el sistema.

- **Configuración:** Esta sub opción modificar algunos datos de la empresa.

Datos de la Empresa

Nombre:

Teléfono:

Correo Electrónico:

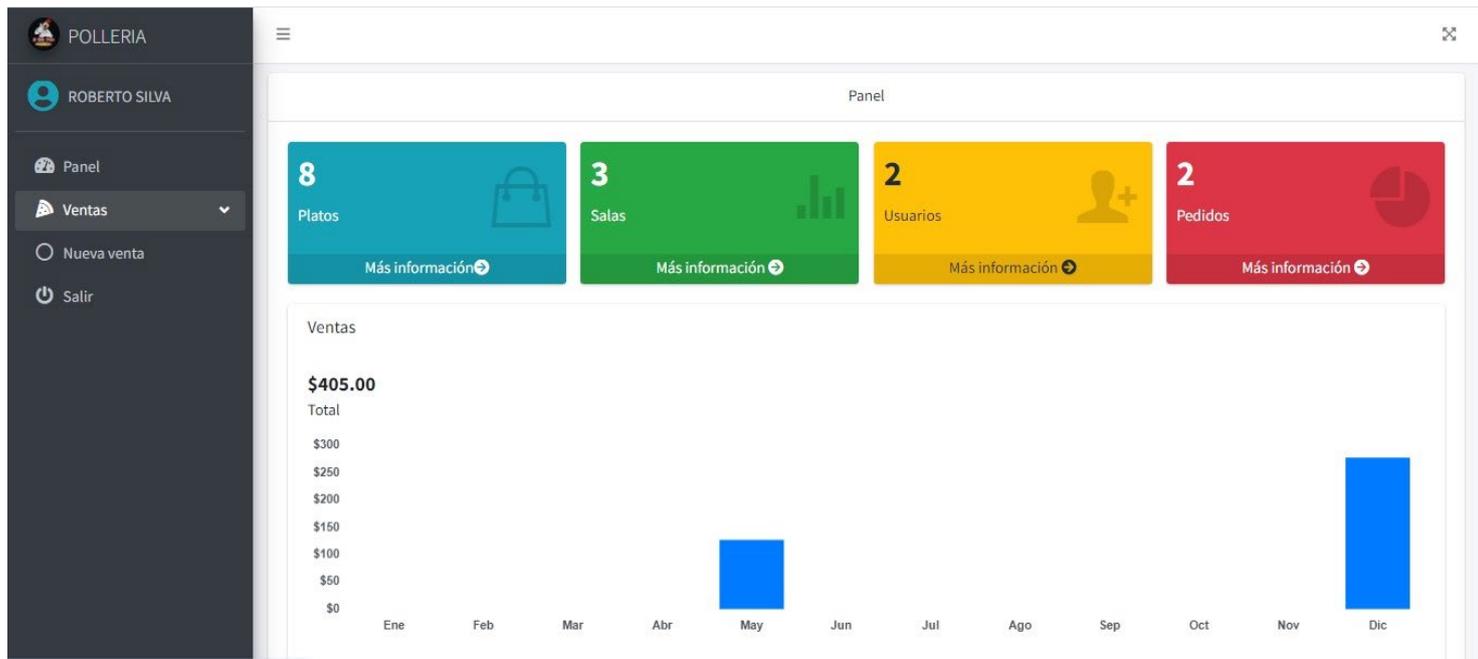
Dirección:

[Modificar Datos](#)

Opción Salir: Esta opción permite cerrar sesión para ingreso de otro usuario.

A login form for "EL RICO POLLO". The form is centered on a light grey background. At the top, the text "EL RICO POLLO" is displayed in a large, dark grey font. Below this, the text "Inicia sesión para iniciar tu sesión" is centered. The form contains two input fields: the first is for the email address, containing "mozo@gmail.com" and a mail icon; the second is for the password, containing four dots and a lock icon. Below the input fields is a blue button with the text "INICIAR SESIÓN" in white.

Una vez finalizada la sesión del administrador iniciaremos con el acceso del MOZO (usuario y contraseña).



Como se puede observar el mozo solo tiene acceso para poder gestionar solo ventas.

