



MANUAL TÉCNICO DEL SISTEMA WEB

AUTORES:

MARILYN DIANA ULLOA MARCIAL

JERMIN SHADIN VASQUEZ TORRES

Riobamba – Ecuador

2025

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	3
ÍNDICE DE FIGURAS	4
1. REQUERIMIENTOS TÉCNICOS	5
1.1 REQUERIMIENTOS MÍNIMOS DE HARDWARE.....	5
1.1.1 Computadoras de escritorio o portátiles	5
1.1.2 Conexión a Internet.....	5
1.2 REQUERIMIENTOS MÍNIMOS DE SOFTWARE	5
1.2.1 Sistema Operativo.....	5
1.2.2 Navegador web.....	5
2. ASPECTOS TÉCNICOS	5
2.1 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	6
2.1.1 JavaScript	6
2.1.2 TypeScript	6
2.1.3 Angular.....	6
2.1.4 Node.js.....	6
2.1.5 PostgreSQL.....	7
2.1.6 PgAdmin 4.....	7
3. Planificación	8
3.1 Roles del proyecto.....	8
3.2 Roles del sistema.....	8
3.3 Product Backlog.....	9
3.4 Sprint Backlog.....	12
3.5 Buenas prácticas y tecnologías aplicadas.....	15
4. DIAGRAMAS DE MODELAMIENTO	17
4.1 DIAGRAMA DE CASOS DE USO	17
4.2 DIAGRAMA DE SECUENCIA.....	20
4.3 DIAGRAMA DE COMPONENTES.....	21

4.4	DIAGRAMA DE DESPLIEGUE	21
4.5	DICCIONARIO DE DATOS	22
5.	ROLES DEL SISTEMA.....	39
	BIBLIOGRAFÍA.....	40

ÍNDICE DE TABLAS

Tabla 3-1: Roles del proyecto.....	8
Tabla 3-2: Roles del sistema.....	8
Tabla 3-3: Sistema T-shirt realizado.....	10
Tabla 3-4: Product Backlog.....	10
Tabla 3-5: Sprint Backlog.....	12

ÍNDICE DE FIGURAS

Ilustración 4-1: Diagrama de casos de uso del módulo seguridad	18
Ilustración 4-2: Diagrama de casos de uso del módulo test vocacional.....	19
Ilustración 4-3: Diagrama de casos de uso del módulo simulador.....	20
Ilustración 4-4: Diagrama de secuencias del sistema	21
Ilustración 4-5: Diagrama de componentes del sistema	21
Ilustración 4-6: Diagrama de despliegue del sistema	22

REQUERIMIENTOS TÉCNICOS

1.1 REQUERIMIENTOS MÍNIMOS DE HARDWARE

El correcto manejo del sistema web implementado para el restaurante “La Hueca del Sabor” requiere de condiciones mínimas tanto a nivel de hardware como software. A continuación, se detallan las recomendaciones y especificaciones para garantizar una experiencia grata, fluida y estable durante la operación del sistema en entornos reales de producción.

1.1.1 *Computadoras de escritorio o portátiles:*

- **Procesador:** Intel Core i3 (8ª generación) o equivalente AMD Ryzen 3.
- **Memoria RAM:** 4 GB mínimo (recomendado 8 GB para entornos de administración).
- **Almacenamiento:** Al menos 500 MB de espacio disponible para ejecución local de archivos temporales, caché del navegador y registros.
- **Pantalla:** Resolución mínima de 1366x768 píxeles.

1.1.2 *Conexión a Internet:*

- **Velocidad mínima:** 5 Mbps de descarga y 1 Mbps de subida.
- **Recomendación:** Conexión estable por cable o red Wi-Fi de banda 5 GHz para evitar latencias al operar desde el fronted.

1.2 REQUERIMIENTOS MÍNIMOS DE SOFTWARE

1.2.1 *Sistema Operativo:*

- Para usuarios del sistema
 - Computadoras: Windows 10 o superior
 - macOS 10.14 (Mojave) o superior
- Para el servidor (Contabo VPS):
 - Windows Server 2019 Standard Edition (implementado actualmente en el entorno de producción).

1.2.2 *Navegador web:*

Para poder acceder al sistema desde el cliente (frontend), se recomienda el uso de un navegador actualizado:

- Google Chrome (versión 100 o superior).
- Mozilla Firefox (versión 100 o superior).
- Microsoft Edge (versión 100 o superior).
- Safari (versión 13 o superior, para dispositivos Apple).

2. ASPECTOS TÉCNICOS

El desarrollo del sistema web para La Hueca del Sabor que se fundamenta en una arquitectura moderna basada en tecnologías libres, escalables que son ampliamente aceptadas en la industria del desarrollo del software. Dichas herramientas han sido seleccionadas estratégicamente para garantizar un buen rendimiento, la seguridad y la mantenibilidad del sistema durante el tiempo. En esta sección se describen las tecnologías empleadas durante el proceso de construcción de la solución.

2.1 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

2.1.1 React.js

React es una biblioteca de JavaScript de código abierto que fue desarrollada por Meta (antes llamada Facebook) utilizada para construir interfaces de usuario interactivas. Su modelo de componentes permite reutilizar bloques de código de manera eficiente y sencilla, facilitando el mantenimiento y la escalabilidad del sistema frontend. Dando un enfoque declarativo y la virtual DOM optimizan el rendimiento, lo que resulta útil para sistemas con actualización frecuente de vistas, como la toma de pedidos en tiempo real (Wieruch, 2022)

2.1.2 Node.js

Node.js es un entorno de ejecución para JavaScript que permite construir aplicaciones desde el lado del servidor con una gran capacidad de manejo de concurrencia. Ya que está basado en el motor V8 de Google, brinda una alta velocidad y eficiencia. En este proyecto, Node.js acompañado con Express.js serán útiles para la implementación lógica del backend, gestionar rutas HTTP, conectarse con la base de datos y sobre todo asegurar los servicios mediante autenticación y autorización (Tilkov & Vinoski, 2010)

2.1.3 PostgreSQL

Es un sistema de gestión de bases de datos relacional orientado a objetos. Soporta extensiones avanzadas como funciones personalizadas, tipos de datos definidos por el usuario y transacciones ACID. Lo que le dio a ser escogido como motor principal para el sistema debido a su fuerte robustez, seguridad y amplia compatibilidad con herramientas actuales de desarrollo (PostgreSQL Global Development Group, 2024)

2.1.4 PgAdmin 4

PgAdmin 4 es una herramienta gráfica de administrador para PostgreSQL. Permite gestionar bases de datos de manera visual, exportar datos, realizar consultas SQL y administrar esquemas. En este proyecto, se empleó tanto en ambientes de desarrollo como de producción para la supervisión de tablas, relaciones y el control de acceso a datos (PgAdmin Development Team, 2024).

2.1.5 Express.js

Express es un framework web minimalista para Node.js que facilita la construcción de APIs RESTful. Su estructura ligera y modular lo convierte en una opción eficiente para proyectos de microservicios. En este sistema, fue clave para estructurar los diferentes endpoints de acceso y las validaciones del backend (Homes, 2019)

2.1.6 Docker (entorno de despliegue en VPS Contabo)

Para dar una garantía de portabilidad y consistencia entre los entornos de desarrollo y producción, se utilizó Docker. Los contenedores permiten

Empaquetar la aplicación con todas sus dependencias, facilitando el despliegue en la nube. Este enfoque minimiza los problemas por diferencias de configuración y facilita el escalado futuro del sistema (Merkel, 2024).

3. PLANIFICACIÓN

La planificación del desarrollo del sistema web para el restaurante La Hueca del Sabor se llevó a cabo bajo el enfoque ágil de la metodología Scrum, lo que permitió organizar el trabajo en entregas incrementales, fomentar una colaboración constante con los usuarios finales y realizar ajustes oportunos según los requerimientos emergentes. Esta metodología encaja perfectamente a las necesidades del restaurante, que requiere un sistema escalable, funcional en un corto periodo de tiempo.

3.1 Roles del proyecto

En la **Tabla 3-9** se describe a los stakeholders involucrados en el proyecto.

Durante el desarrollo del sistema se establecieron los siguientes roles clave, distribuidos entre los integrantes del equipo y los actores académicos responsables del proceso de titulación:

Tabla 3-1: Roles del proyecto

Persona Responsable	Rol
Lic. Joselyn Ulloa	Product Owner
Srita. Marilyn Ulloa	Scrum Master
Srita. Marilyn Ulloa	Development Team
Sr. Jermin Vasquez	Development Team

Realizado por: Ulloa & Vasquez, 2025.

Este esquema de trabajo promueve la organización eficiente de tareas y asegura la supervisión académica conforme a los lineamientos institucionales

3.2 Roles del sistema

El sistema web implementado contempla la existencia de usuario, cada uno contempla la existencia de múltiples perfiles de usuario, cada uno con un conjunto específico de funcionalidades de acuerdo con su rol dentro de un flujo operativo del restaurante La Hueca del Sabor. Esta segmentación garantiza un control adecuado de los permisos, resguardo de la información y eficiencia en las operaciones.

A continuación, se detalla cada uno de los roles y las operaciones que se puede ejecutar en el sistema:

Tabla 3-2: Detalles de roles funcionales del sistema

Rol	Operaciones
Mesero	<ul style="list-style-type: none">• Iniciar sesión con credenciales asignadas.• Visualizar menú del día (filtrado por disponibilidad).• Registrar nuevos pedidos por mesa.• Modificar o eliminar pedidos en estado “pendiente” (antes de enviarlos).• Enviar pedido a cocina.• Visualizar estado actualizado de cada pedido (En preparación, Listo, Entregado).• Cancelar pedidos en coordinación con caja (con justificación).
Cocina	<ul style="list-style-type: none">• Acceder a la lista de pedidos pendientes por orden de llegada.• Marcar pedidos como “En preparación”.• Marcar pedidos como “Listo para entregar”.• Visualizar historial del día por hora y estado del pedido.• Filtrar pedidos por categoría (bebidas, platos fuertes, extras, etc.).• Registrar observaciones internas para el área de caja (opcional).
Cajero	<ul style="list-style-type: none">• Acceder al panel de pedidos listos para cobrar.• Visualizar detalle de cada pedido por mesa y total a pagar.• Ingresar valor recibido y calcular cambio.• Generar comprobante en formato PDF.• Registrar observaciones del cliente (por ejemplo, “factura con RUC”).• Marcar pedido como “Pagado” y enviar notificación a mesero.• Anular pedidos con causa justificada (opcional, con contraseña de administrador).
Administrador/Gerente	<ul style="list-style-type: none">• Visualizar historial del día por hora y estado del pedido.• Crear, editar y eliminar productos del menú.• Cambiar disponibilidad diaria (activo/inactivo).• Crear, editar o eliminar cuentas de usuarios (meseros, cocina, caja).• Generar reportes por fecha, producto o rol (PDF o Excel).• Visualizar estadísticas de ventas por día, semana y mes (dashboard).• Monitorear estado general de pedidos en tiempo real.• Habilitar o deshabilitar opciones de pago (efectivo, transferencia).• Exportar bitácora de operaciones realizadas por cada usuario.

Realizado por: Ulloa & Vasquez, 2025.

3.3 Product Backlog

El Product Backlog representa una lista ordenada y priorizada de todas las funcionalidades, mejoras, tareas necesarias para el desarrollo, y correcciones para el restaurante “La Hueca del Sabor”. Su elaboración se fundamenta con los requisitos funcionales que se obtuvieron tras entrevistas realizadas con el personal administrativo, observación de los procesos internos y

necesidades frecuentes detectadas en la operación diaria del negocio.

La evaluación de cada historia de usuario o historia técnica fue empleada según su prioridad, complejidad y esfuerzo estimado. Para esto se utilizó el sistema de estimación tipo T-shirt sizing, en el que se asignan tallas que representan esfuerzo de desarrollo:

- XS: Actividad pequeña y complejidad baja.
- S: Actividad pequeña y complejidad media/alta.
- M: Actividad y complejidad medias.
- L: Actividad grande y complejidad baja/media.
- XL: Actividad grande y complejidad alta.

A continuación se detallan las historias técnicas (HT) y de usuario (HU) que componen la pila inicial del producto

Tabla 3-3: Product Backlog.

Historias Técnicas			
Id	Nombre de historia	Prioridad	Estimación T-shirt
HT01	Implementar control de acceso por roles	Alta	M
HT02	Crear base de datos relacional	Alta	L
HT03	Validación de stock antes de confirmar pedido	Alta	M
HT04	Soporte para manejo offline de pedidos	Media	M
HT05	Dashboard con librerías gráficas (Chart.js)	Alta	M
HT06	Exportación de reportes a PDF y Excel	Alta	M
Total estimado			6 tareas (1L, 5M)
Historias de Usuario			
Id	Nombre de historia	Prioridad	Estimación T-shirt
HU01	Iniciar sesión en el sistema	Alta	S
HU02	Visualizar menú de productos disponibles	Alta	S
HU03	Tomar pedido del cliente	Alta	M
HU04	Editar pedido antes de confirmarlo	Media	S
HU05	Enviar pedido a cocina y caja	Alta	S
HU06	Visualizar pedidos en cocina	Alta	S
HU07	Ver pedidos en la caja para cobrar	Alta	S
HU08	Generar comprobante de pago PFD	Alta	M
HU09	Actualizar disponibilidad diaria de productos	Alta	S
HU10	Visualizar dashboard gerencial	Alta	M
HU11	Generar reportes personalizados por filtros	Alta	M
HU12	Exportar reportes a PDF o Excel	Alta	M
Total estimado			12 tareas (7 S, 5 M)

Realizado por: Ulloa & Vasquez, 2025.

3.4 Sprint Backlog

Se representa a detalle la planificación y calendarización de los sprints que componen el desarrollo del sistema web para el restaurante “La Hueca del Sabor”. Cada sprint corresponde a una iteración de trabajo de una semana, en la que se implementan funcionalidades específicas, y tomadas del Product Backlog.

En el que se han planificado un total de 10 sprints, distribuidos desde el 24 de marzo de 2015 y junio de 2025. Cada sprint tiene una duración estimada de una semana, con objetivos definidos, entregables concretos y una carga horaria de promedio de 10 a 20 horas por iteración.

A continuación, se detalla el desglose de cada sprint, con las tareas asignadas, su duración estimada y el entregable funcional esperando.

Tabla 3-4: Sprint Backlog

ID	Tarea	Puntos Estimados	Fecha inicio	Fecha fin	Puntos Totales
SPRINT 1 (1 SEMANA)					
HT02	HT02: Crear base de datos relacional HT01: Control de acceso por roles	16	24/03/2025	28/03/2025	16
SPRINT 2 (1 SEMANA)					
HU01	HU01: Iniciar sesión HU02: Visualizar menú de productos	12	31/03/2025	04/04/2025	28
SPRINT 3 (1 SEMANA)					
HU9	HU03: Tomar pedido del cliente	10	07/04/2025	11/04/2025	10
SPRINT 4 (1 SEMANA)					
HU04	HU04: Editar pedido antes de confirmarlo HT03: Validación de stock	12	14/04/2025	18/04/2025	22
SPRINT 5 (1 SEMANA)					
HU30	HU05: Enviar pedido a cocina y caja	10	21/04/2025	25/04/2025	10
SPRINT 6 (1 SEMANA)					
HU06	HU06: Visualizar pedidos en cocina HU07: Ver pedidos en caja	12	28/04/2025	02/05/2025	22
SPRINT 7 (1 SEMANA)					
HU50	HU08: Generar comprobante en PDF HT06: Exportar reportes PDF/Excel	14	05/05/2025	09/05/2025	14
SPRINT 8 (1 SEMANA)					
HU46	HU09: Actualizar disponibilidad de platos HT04: Soporte offline	14	12/05/2025	16/05/2025	14

SPRINT 9 (1 SEMANA)					
HT4	HU10: Dashboard gerencial HU11: Reportes personalizados HU12: Exportar reportes HT05: Dashboard con gráficas	20	19/05/2025	23/05/2025	20
SPRINT 10 (1 SEMANA)					
	Pruebas finales, correcciones, documentación y despliegue en servidor	20	26/05/2025	30/05/2025	20

Realizado por: Ulloa & Vasquez, 2025.

3.5 Buenas prácticas y tecnologías aplicadas

Para el desarrollo del sistema se requerirá del uso de varias tecnologías que nos ayudarán a lo largo de los procesos, estas tecnologías pueden variar según las necesidades o las disposiciones impuestas por la ESPOCH, a continuación, se mencionan las tecnologías que se utilizarán para el desarrollo del sistema:

- Angular: Para el desarrollo Front-end se implementará mediante el framework angular, el cual es utilizado como estándar para el desarrollo web de plataformas institucionales de la ESPOCH.
- Node.js: Para el desarrollo Back-end se implementará mediante el Node.js, el cual es utilizado como estándar para el desarrollo de la parte del servidor para plataformas institucionales de la ESPOCH.
- PostgreSQL: El gestor de base de datos será PostgreSQL, este es el gestor designado para las bases de datos dentro de la ESPOCH, por lo cual mediante el uso de VPNs vamos a acceder a sus servidores y elaborar las bases de datos requeridas ahí.
- Postman: El software Postman es utilizado para la validación de las rutas creadas en el Back-end del proyecto, testeo y posteriores correcciones para su implementación en el Front-end del proyecto.
- Crypto-js: La implementación de Crypto-js fue fundamental para implementar medidas de seguridad en nuestro sistema. Esta nos permite encriptar la información sensible de nuestro sistema. Nuestro sistema cuenta con un apartado del Administrador el cual requiere de credenciales (correo institucional) para su autenticación. En este proceso se recibe información sensible del usuario autenticado, por lo cual, se implementó encriptación en el Back-end con el algoritmo AES (Advanced Encryption Standard), se definió una clave de encriptación para agilizar el proceso de encriptación y desencriptación, esta es almacenada de forma local para evitar sus accesos. El proceso de encriptación y desencriptación no deja posibilidad de acceso desde el Front-end toda la información obtenida es enviada al Back-end para su encriptación y de esa manera almacena en el Front-end, para cuando se requiera de esta información es enviada al Back-end para su desencriptación y su uso.
- Adm-zip: Se utilizó la librería de Adm-zip para satisfacer las necesidades de análisis de archivos .zip. Esta nos permite extraer archivos comprimidos para obtener su contenido para un análisis. En este caso los reactivos para el sistema se encuentran comprimidos, por lo cual, es necesario primero su extracción y análisis (ordenamiento, implementación de etiquetas y reagrupación) para su posterior almacenamiento en la base de datos.
- Axios: Esta librería fue fundamental durante el desarrollo para la implementación de APIs de un servidor tercero. Esto fue requerido para la obtención de las IPs mediante una API de uso libre, para el registro de procesos en auditoría. Además, fue requerido para la implementación del servicio de DTIC para la validación de cédulas y obtención de información.

4. DIAGRAMAS DE MODELAMIENTO

4.1 DIAGRAMA DE CASOS DE USO

En el diagrama de casos de uso se puntualiza cada rol que desempeñan los actores relacionados con el sistema web de gestión para “La Hueca del Sabor”. En este caso, los actores principales son: Mesero, Cocina, Caja y Gerencia, quienes relacionan directamente con las funcionalidades que son clave del sistema, como la toma y gestión de pedidos, la preparación de estos, el proceso de facturación y el análisis de métricas del negocio. Cada actor cuenta con un conjunto específico de tareas, las cuales fueron moldeadas para representar de una manera clara y estructurada su participación dentro del flujo de trabajo del restaurante.

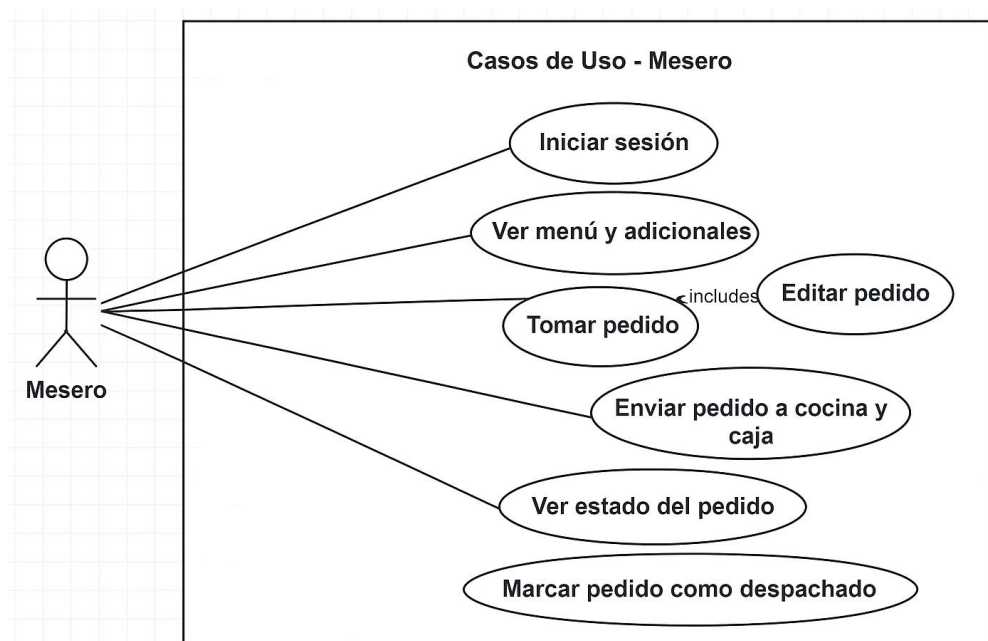


Ilustración 4-1: Diagrama de casos de uso del módulo Mesero

Realizado por: Ulloa & Vasquez, 2025.

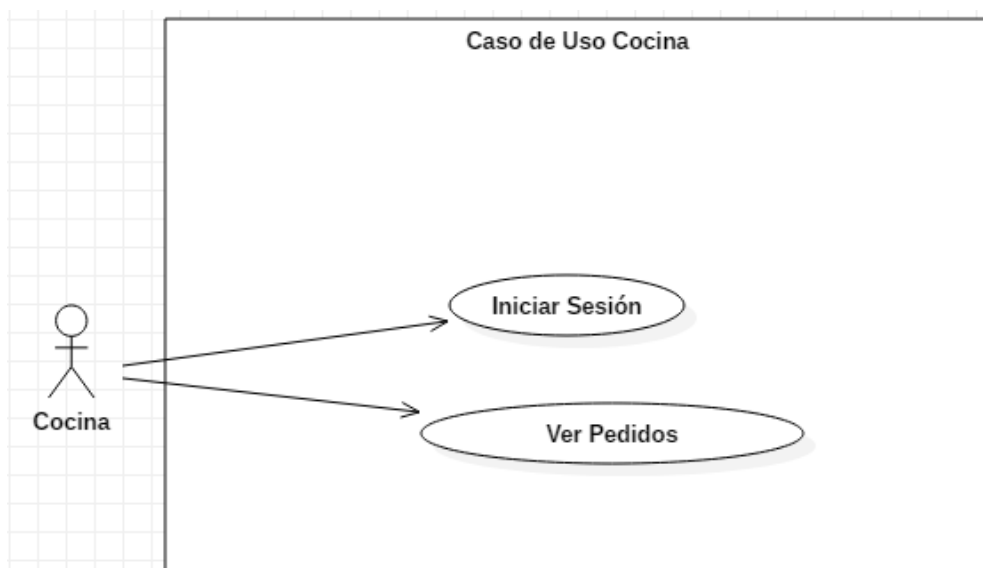


Ilustración 4-2: Diagrama de casos de uso del módulo Cocina

Realizado por: Ulloa & Vasquez, 2025.

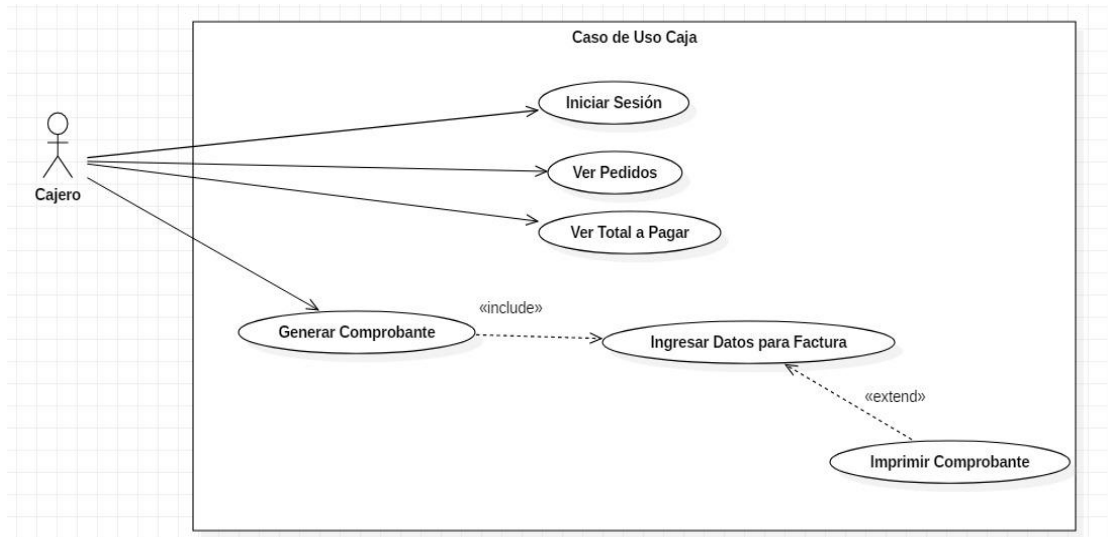


Ilustración 4-3: Diagrama de casos de uso del módulo Caja

Realizado por: Ulloa & Vasquez, 2025.

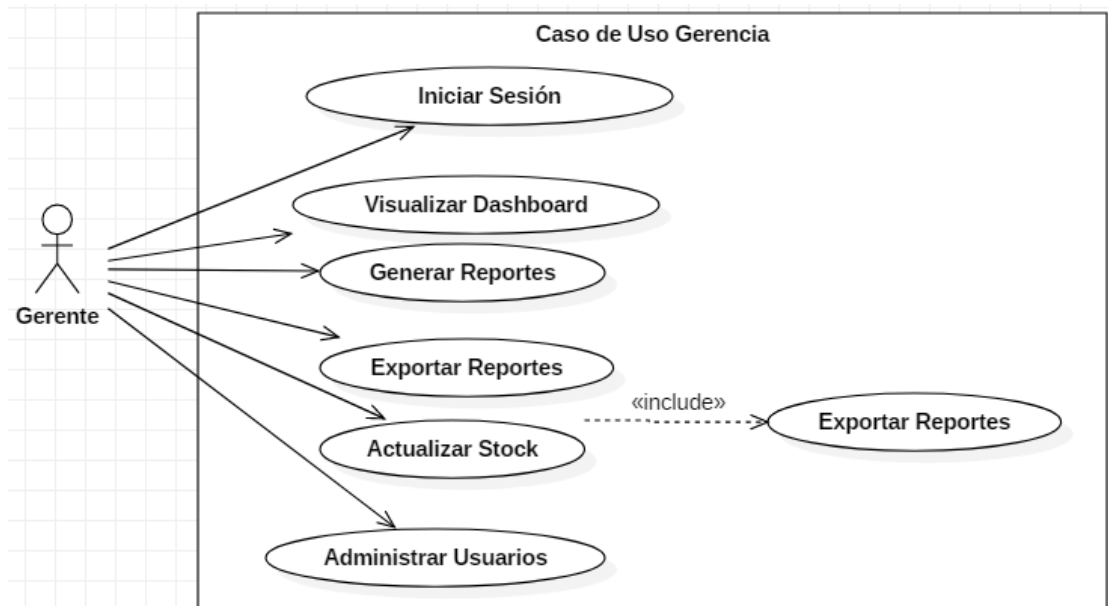


Ilustración 4-4: Diagrama de casos de uso del módulo Gerencia

Realizado por: Ulloa & Vasquez, 2025.

4.2 DIAGRAMA DE SECUENCIA

Como vemos en la **Ilustración 3-3**, que muestra cómo los usuarios interactúan con el sistema y cómo los componentes del backend procesan las solicitudes, el sistema sigue una secuencia de pasos: desde la interfaz de usuario se envía una solicitud a los servicios web, esta llama a la lógica de negocio en procesos, y finalmente se accede a la base de datos a través modelo.

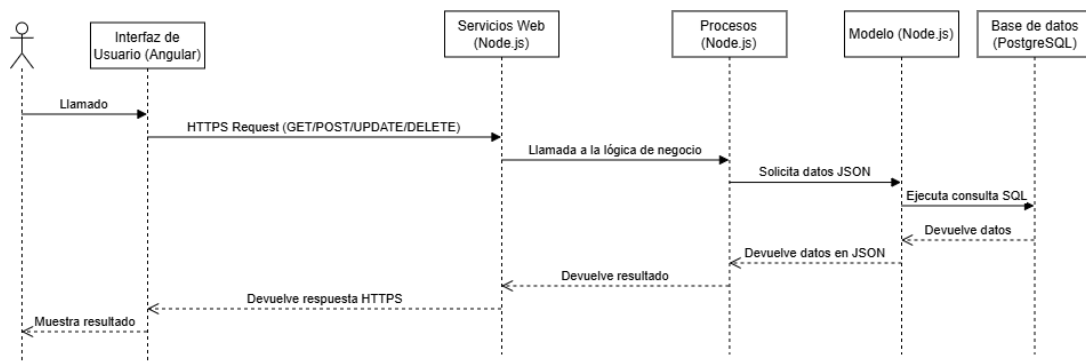


Ilustración 4-5: Diagrama de secuencias del sistema

Realizado por: Ulloa & Vasquez, 2025.

4.3 DIAGRAMA DE COMPONENTES

Como vemos en la **Ilustración 3-4** para representar cómo se estructuran las carpetas y módulos del proyecto. El frontend se comunica con el backend, que a su vez está dividido en componentes como servicios web, procesos y modelo el cual extrae los datos de la base de datos. Esta vista es útil ya que muestra cómo se organiza el código y cómo los componentes dependen entre sí.

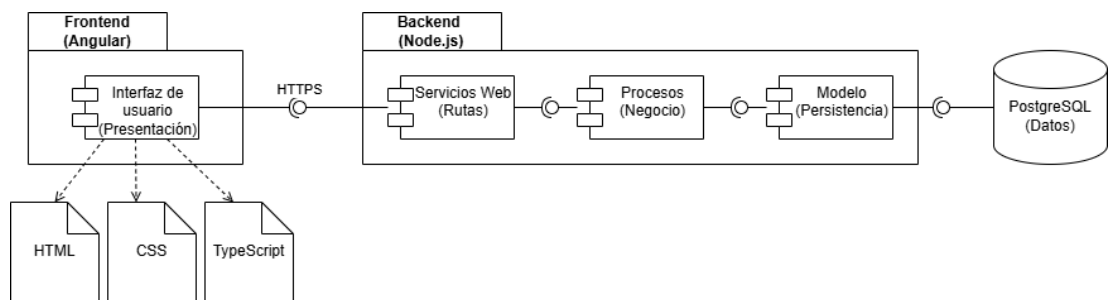


Ilustración 4-5: Diagrama de componentes del sistema

Realizado por: Ulloa & Vasquez, 2025.

4.4 DIAGRAMA DE DESPLIEGUE

como vemos en la **Ilustración 3-5**, que muestra cómo el cliente interactúa con el frontend el cual a su vez interactúa con el backend que se ejecutan en un servidor web, mientras que la base de datos se encuentra en un servidor separado. Este diagrama es importante para entender cómo se distribuyen los recursos físicos y cómo se comunican los componentes en un entorno de producción.

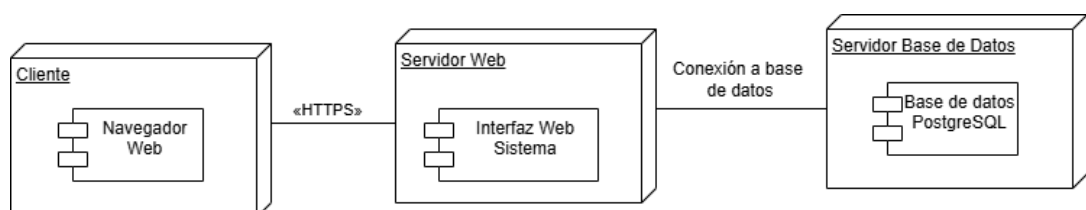


Ilustración 4-6: Diagrama de despliegue del sistema

Realizado por: Ulloa & Vasquez, 2025.

4.5 DISEÑO DE LA BASE DE DATOS

Para el diseño de nuestra base de datos se trabajó en la creación de distintos esquemas, que nos permiten organizar las tablas de acuerdo con su propósito y función en la base de datos, en la **Ilustración 4-7** se muestran los esquemas utilizados.



Ilustración 4-7: Esquemas de la DB.

Realizado por: Ulloa & Vasquez, 2025.

Nuestra base de datos cuenta con un total de 55 tablas organizadas por esquemas, representados en la **Tabla 4-1**.

Tabla 4-1: Tablas por esquema DB.

Esquema	Nº de tablas
Auditoria	2
Central	5
Public	-
Seguridad	5
Simulador	23
TestVocacional	20
TOTAL	55

Realizado por: Ulloa & Vasquez, 2025.

Para el esquema de seguridad se implementaron las tablas de la **Ilustración 4-8**, para la gestión de roles, perfiles y menús.

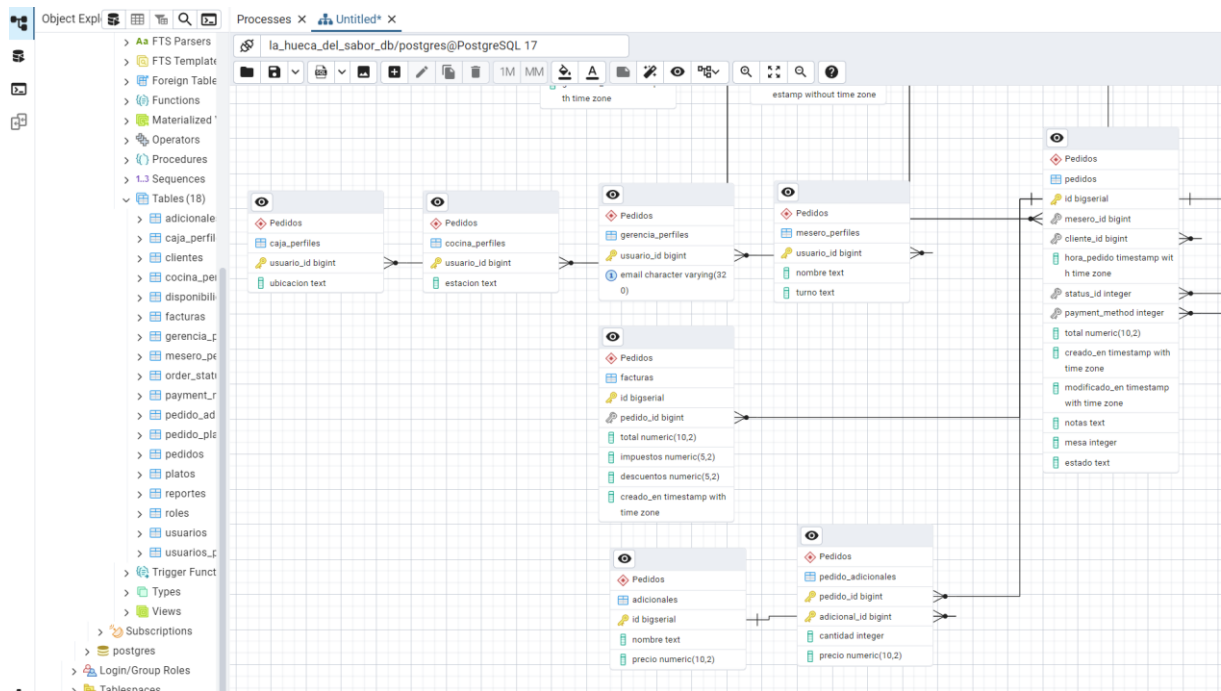


Ilustración 4-8: Base de Datos.

Realizado por: Ulloa & Vasquez, 2025.

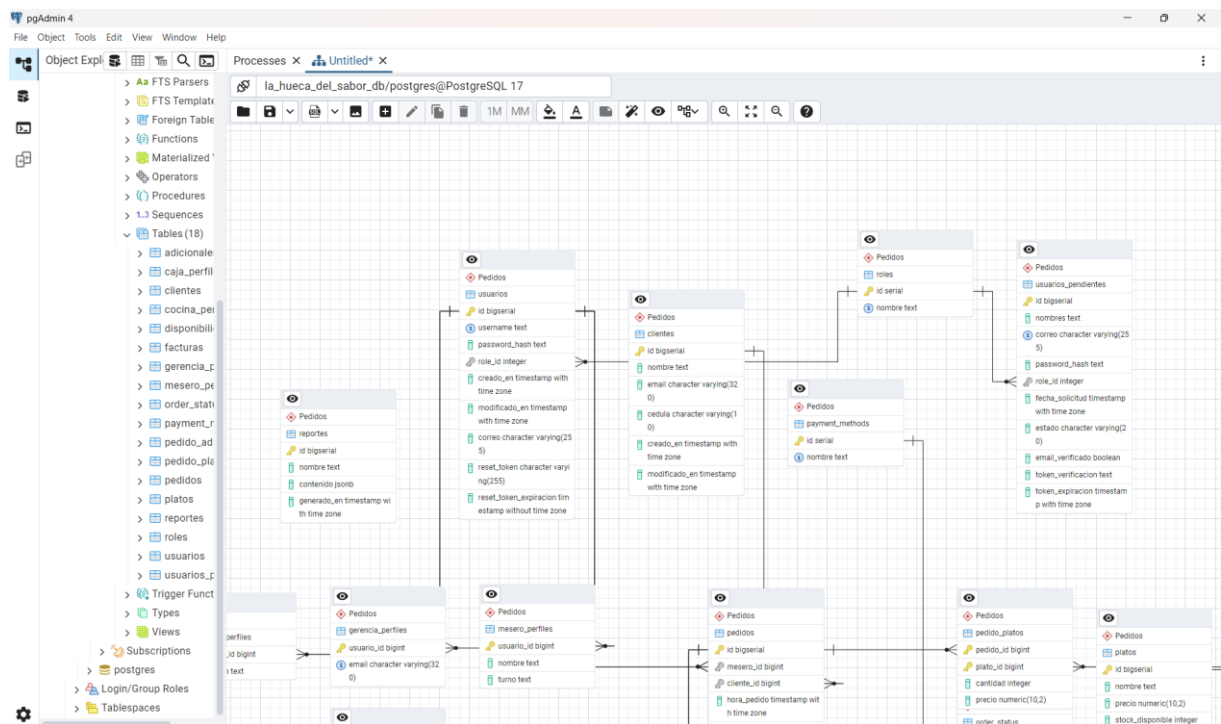


Ilustración 4-9: Base de Datos.

Realizado por: Ulloa & Vasquez, 2025.

3. Tabla: clientes

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
nombre	text	Sí		
email	character varying(320)	Sí		
cedula	character varying(10)	Sí		
creado_en	timestamp with time zone	Sí	now()	
modificado_en	timestamp with time zone	Sí		

4. Tabla: pedidos

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
mesero_id	bigint	Sí		FK
cliente_id	bigint	Sí		FK
hora_pedido	timestamp with time zone	Sí		
status_id	integer	Sí		FK
payment_method	integer	Sí		FK
total	numeric(10,2)	Sí		
creado_en	timestamp with time zone	Sí	now()	
modificado_en	timestamp with time zone	Sí		
notas	text	Sí		
mesa	integer	Sí		
estado	text	Sí		

5. Tabla: facturas

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
pedido_id	bigint	Sí		FK
total	numeric(10,2)	Sí		
impuestos	numeric(5,2)	Sí		
descuentos	numeric(5,2)	Sí		
creado_en	timestamp with time zone	Sí	now()	

6. Tabla: platos

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
pedido_id	bigint	No		PK, FK
plato_id	bigint	No		PK, FK
cantidad	integer	Sí		
precio	numeric(10,2)	Sí		

7. Tabla: pedido_platos

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
pedido_id	bigint	No		PK, FK
plato_id	bigint	No		PK, FK
cantidad	integer	Sí		
precio	numeric(10,2)	Sí		

8. Tabla: adicionales

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
nombre	text	Sí		
precio	numeric(10,2)	Sí		

9. Tabla: pedido_adicionales

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
pedido_id	bigint	No		PK, FK
adicional_id	bigint	No		PK, FK
cantidad	integer	Sí		
precio	numeric(10,2)	Sí		

10. Tabla: order_status

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	serial	No		PK
code	text	Sí		
label	text	Sí		

11. Tabla: payment_methods

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	serial	No		PK
nombre	text	Sí		

12. Tabla: reportes

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
nombre	text	Sí		
contenido	jsonb	Sí		
generado_en	timestamp with time zone	Sí	now()	

13. Tabla: disponibilidad de datos

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
plato_id	bigint	Sí		FK
fecha	date	Sí		
stock	integer	Sí		
actualizado_en	timestamp with time zone	Sí		

14. Tabla: usuarios_pendientes

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
id	bigserial	No		PK
nombres	text	Sí		
correo	character varying(255)	Sí		
password_hash	text	Sí		
role_id	integer	Sí		FK
fecha_solicitud	timestamp with time zone	Sí	now()	
estado	character varying(20)	Sí		
email_verificado	boolean	Sí		
token_verificacion	text	Sí		
token_expiracion	timestamp with time zone	Sí		

15. Tabla: caja_perfiles

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
usuario_id	bigint	No		PK, FK
ubicacion	text	Sí		

16. Tabla: cocina_perfiles

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
usuario_id	bigint	No		PK, FK
estacion	text	Sí		

17. Tabla: gerencia_perfiles

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
usuario_id	bigint	No		PK, FK
email	character varying(320)	Sí		

18. Tabla: simulador.tb_cuestionario

Nombre	Tipo de Dato	Nulo	Predeterminado	Clave
usuario_id	bigint	No		PK, FK
nombre	text	Sí		
turno	text	Sí		

5. ROLES DEL SISTEMA

El sistema está compuesto inicialmente con 4 roles principales: Mesero, Cocina, Cajero y administrador/Gerente. En la siguiente tabla se especifica que operaciones puede realizar cada rol:

Rol	Operaciones
-----	-------------

Mesero	<ul style="list-style-type: none"> • Iniciar sesión con credenciales asignadas. • Visualizar menú del día (filtrado por disponibilidad). • Registrar nuevos pedidos por mesa. • Modificar o eliminar pedidos en estado “pendiente” (antes de enviarlos). • Enviar pedido a cocina. • Visualizar estado actualizado de cada pedido (En preparación, Listo, Entregado). • Cancelar pedidos en coordinación con caja (con justificación). • Visualizar historial de sus pedidos.
Cocina	<p>Operaciones:</p> <ul style="list-style-type: none"> • Acceder a la lista de pedidos pendientes por orden de llegada. • Marcar pedidos como “En preparación”. • Marcar pedidos como “Listo para entregar”. • Visualizar historial del día por hora y estado del pedido. • Filtrar pedidos por categoría (bebidas, platos fuertes, extras, etc.). • Registrar observaciones internas para el área de caja (opcional).
Cajero	<p>Operaciones:</p> <ul style="list-style-type: none"> • Acceder al panel de pedidos listos para cobrar. • Visualizar detalle de cada pedido por mesa y total a pagar. • Ingresar valor recibido y calcular cambio. • Generar comprobante de pago en formato PDF. • Registrar observaciones del cliente (ej. “factura con RUC”). • Marcar pedido como “Pagado” y enviar notificación a mesero. • Anular pedidos con causa justificada (opcional, requiere contraseña de administrador). • Descargar historial de transacciones.

	<ul style="list-style-type: none"> •Filtros por fecha, cliente, método de pago o estado del pedido.
Administrador / Gerente	<p>Operaciones:</p> <ul style="list-style-type: none"> • Visualizar historial del día por hora y estado del pedido. • Crear, editar y eliminar productos del menú. • Cambiar disponibilidad diaria de platos y adicionales. • Crear, editar o eliminar cuentas de usuarios (meseros, cocina, caja). • Generar reportes por fecha, producto o rol (en PDF o Excel). • Visualizar estadísticas de ventas por día, semana y mes (dashboard gerencial). • Monitorear estado general de pedidos en tiempo real. • Habilitar o deshabilitar métodos de pago (efectivo, transferencia). • Exportar bitácora de operaciones realizadas por cada usuario. • Asignación de roles y perfiles a usuarios registrados. • Confirmar usuarios pendientes con rol y credenciales asignadas.

BIBLIOGRAFÍA

- PgAdmin Development Team. (2024). *PgAdmin 4 - Documentation*. Recuperado de <https://www.pgadmin.org/docs/pgadmin4/latest/index.html>
- Holmes, A. (2019). *Getting MEAN with Mongo, Express, Angular, and Node* (2nd ed.). Manning Publications.
- Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, (239), 2.
- PgAdmin Development Team. (2024). *PgAdmin 4 - Documentation*. <https://www.pgadmin.org/docs/pgadmin4/latest/index.html>
- PostgreSQL Global Development Group. (2024). *PostgreSQL: The world's most advanced open source relational database*. <https://www.postgresql.org>
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80–83.
- Wieruch, R. (2022). *The Road to React* (Updated Edition). <https://www.roadtoreact.com/>

