

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN SISTEMA CRUD WEB PARA EL ÁREA DE VENTAS DEL RESTAURANTE “EL SABOR CAJABAMBINO” UTILIZANDO ASP.NET CORE MVC**

Asignatura:

**PROGRAMACIÓN APLICADA II**

**Integrantes:**

Limay Rodriguez, Adriana Anthonela

Ramos Guerra, Aihnoa Jolie

Perez Briceño, Darick André

Valdiviezo Zavaleta, Jesús Arturo

**Docente:**

Ing. Mendoza Santos Carlos Eduardo

Cajamarca – Perú

Agosto, 2025

Datos Informativos

Versiones

|  |  |
| --- | --- |
| **Versión** | **Fecha** |
| 1.0 | 04.08.2025 |

REPOSITORIO: <https://github.com/Arturo969/5-Ciclo-Programacion-Aplicada-Restaurante-.git>

ÍNDICE DE CONTENIDOS

[CAPÍTULO 1. PLANEACION DE LA SOLUCION DE BI 8](#_Toc205148145)

[1.1. Introducción 8](#_Toc205148146)

[1.2. Justificación 9](#_Toc205148147)

[1.3. Descripción del sistema de gestión web 9](#_Toc205148148)

[1.4. Objetivos 10](#_Toc205148149)

[1.4.1. Objetivo General 10](#_Toc205148150)

[1.4.2. Objetivos Específicos 10](#_Toc205148151)

[1.5. Descripción de los Stakeholders 11](#_Toc205148152)

[CAPÍTULO 2. MODELADO DEL NEGOCIO 12](#_Toc205148153)

[2.1. Descripción de la empresa u organización 12](#_Toc205148154)

[2.2. Organigrama de la empresa 13](#_Toc205148155)

[2.3. Recursos 14](#_Toc205148156)

[2.3.1. Personal 14](#_Toc205148157)

[2.3.2. Hardware y Software 14](#_Toc205148158)

[2.3.3. Otros 14](#_Toc205148159)

[2.4. Alcance 14](#_Toc205148160)

[2.4.1. Dentro del alcance 14](#_Toc205148161)

[2.4.2. Fuera del alcance 15](#_Toc205148162)

[2.5. Tareas 15](#_Toc205148163)

[CAPÍTULO 3. ANALISIS DE REQUERIMIENTOS 16](#_Toc205148164)

[3.1. El Negocio 16](#_Toc205148165)

[3.1.1. Misión 16](#_Toc205148166)

[3.1.2. Visión 16](#_Toc205148167)

[3.1.3. Objetivos organizacionales 16](#_Toc205148168)

[3.1.3.1. Generales 16](#_Toc205148169)

[3.1.3.2. Específicos 16](#_Toc205148170)

[3.2. Descripción de los procesos del negocio 17](#_Toc205148171)

[3.3. Problemas del negocio 17](#_Toc205148172)

[3.4. Selección de los entrevistados 17](#_Toc205148173)

[3.5. Entrevista 17](#_Toc205148174)

[3.5.1. Definición de requerimientos del negocio 18](#_Toc205148175)

[3.6. Resumen de los requerimientos obtenidos en la entrevista 20](#_Toc205148176)

[3.7. Base de Datos Relacional 21](#_Toc205148177)

[Esquema GENERAL 21](#_Toc205148178)

[ Categoria 21](#_Toc205148179)

[ Mesa 21](#_Toc205148180)

[ Producto 21](#_Toc205148181)

[ ProductoIngrediente 22](#_Toc205148182)

[Esquema CLIENTE 22](#_Toc205148183)

[Esquema INVENTARIO 22](#_Toc205148184)

[Esquema PERSONAL 23](#_Toc205148185)

[Esquema TRANSACCION 23](#_Toc205148186)

[3.8. Diagrama Relacional 24](#_Toc205148187)

[CAPÍTULO 4. MARCO TEÓRICO 25](#_Toc205148188)

[4.1. Sistemas de Gestión Web 25](#_Toc205148189)

[4.2. Bases de Datos Relacionales (RDBMS): 25](#_Toc205148190)

[4.2.1. Microsoft SQL Server: 25](#_Toc205148191)

[4.3. Operaciones CRUD 25](#_Toc205148192)

[4.4. Arquitectura de Software 25](#_Toc205148193)

[4.4.1. Patrón Modelo-Vista-Controlador (MVC): 25](#_Toc205148194)

[4.5. ASP.NET Core MVC 25](#_Toc205148195)

[CAPÍTULO 5. ARQUITECTURA DEL SISTEMA DE GESTIÓN WEB SIMPLIFICADO 26](#_Toc205148196)

[5.1. ASP.Net Core MVC: 26](#_Toc205148197)

[5.1.1. Controladores (Controllers): 26](#_Toc205148198)

[5.1.2. Datos (data) 26](#_Toc205148199)

[5.1.3. Vistas (Views): 26](#_Toc205148200)

[5.1.4. Modelos de Vista (View Models): 26](#_Toc205148201)

[5.1.5. wwwroot/css 26](#_Toc205148202)

[5.1.6. Dependencias 26](#_Toc205148203)

[5.1.6.1. Paquetes usados 26](#_Toc205148204)

[CAPÍTULO 6. DESARROLLO DEL SISTEMA DE GESTIÓN WEB 27](#_Toc205148205)

[6.1. Entorno y Tecnologías de Desarrollo 27](#_Toc205148206)

[6.2. Implementación de la Arquitectura MVC 27](#_Toc205148207)

[6.3. Implementación de las Funcionalidades CRUD 28](#_Toc205148208)

[CAPÍTULO 7. PRUEBA DEL SISTEMA DE GESTIÓN WEB 29](#_Toc205148209)

[7.1. Objetivo de las Pruebas 29](#_Toc205148210)

[7.2. Metodología de Pruebas 29](#_Toc205148211)

[7.3. Casos de Prueba y Resultados 29](#_Toc205148212)

[7.4. Conclusiones de las Pruebas 30](#_Toc205148213)

[7.4.1. Resumen 30](#_Toc205148214)

[7.4.2. Problemas Encontrados (y Solucionados) 30](#_Toc205148215)

[7.4.3. Conclusión 30](#_Toc205148216)

[CONCLUSIONES 31](#_Toc205148217)

[RECOMENDACIONES 32](#_Toc205148218)

[REFERENCIAS 33](#_Toc205148219)

[ANEXOS 34](#_Toc205148220)

ÍNDICE DE TABLAS

**No se encuentran elementos de tabla de ilustraciones.**

ÍNDICE DE FIGURAS

[**Figura 1** Diagrama base de datos 24](#_Toc205147613)

RESUMEN

Este informe detalla el desarrollo y la implementación de un sistema de gestión web simplificado para el área de ventas del restaurante "El Sabor Cajabambino", una solución construida con ASP.NET Core MVC y SQL Server. El proyecto se enfoca en la implementación de operaciones CRUD (Crear, Leer, Actualizar, Borrar) para optimizar la gestión de pedidos, inventario, mesas y clientes, resolviendo los problemas derivados de los procesos manuales actuales y sentando las bases para una gestión de datos más eficiente y una mejor toma de decisiones operativas.

# PLANEACION DE LA SOLUCION DE BI

## Introducción

Los objetivos principales del proyecto incluyen diseñar y desarrollar un sistema CRUD web que optimice los procesos del área de ventas del restaurante mediante ASP.NET Core MVC, así como implementar un modelo de base de datos relacional que sostenga los procesos de venta, inventario, pedidos y clientes, velando por la integridad y consistencia de los datos. En este contexto, la arquitectura Modelo-Vista-Controlador (MVC) emerge como un patrón de diseño fundamental para el desarrollo de aplicaciones robustas y escalables, ya que separa la lógica de la aplicación, la interfaz de usuario y el manejo de datos, facilitando así su mantenimiento y desarrollo.

Por lo tanto, el desarrollo de sistemas de información ha evolucionado, adoptando frameworks que simplifican el proceso y promueven buenas prácticas de programación. En particular, ASP.NET Core MVC es un framework de código abierto, multiplataforma y de alto rendimiento que se ha consolidado como una opción sólida para la creación de aplicaciones web dinámicas. De este modo, la sinergia de este framework con sistemas de gestión de bases de datos relacionales como SQL Server permite una gestión de datos eficiente y segura, crucial para la integridad de la información empresarial.

Este informe, por consiguiente, detalla el desarrollo de un sistema de gestión web simplificado para el área de ventas del restaurante "El Sabor Cajabambino". Específicamente, el proyecto se centra en la implementación de las operaciones básicas de datos conocidas como CRUD (Crear, Leer, Actualizar, Borrar) para gestionar de manera integral pedidos, inventario, mesas y clientes. A lo largo del documento, se describirá la planificación, diseño y desarrollo de esta solución, demostrando cómo la aplicación de un enfoque estructurado y las tecnologías adecuadas pueden resolver problemas operacionales reales.

Finalmente, la estructura de este informe se presenta de la siguiente manera: el CAPÍTULO 1. PLANEACIÓN DEL SISTEMA DE GESTIÓN WEB establece los fundamentos del proyecto, mientras que el CAPÍTULO 2. MODELADO DEL NEGOCIO describe la empresa y su contexto. Además, el CAPÍTULO 3. ANÁLISIS DE REQUERIMIENTOS aborda el levantamiento de información y el diseño de la base de datos. Por último, los CAPÍTULOS 4, 5, 6 y 7 se centran en el marco teórico, la arquitectura, el desarrollo y las pruebas del sistema, respectivamente.

## Justificación

La implementación de un sistema CRUD web para el área de ventas del restaurante "El Sabor Cajabambino" es crucial para optimizar los procesos de registro y control operacional. Actualmente, la gestión de ventas, pedidos y el inventario se realiza de manera manual o a través de herramientas poco integradas, lo que genera inconsistencias en los datos, lentitud en el servicio y dificultades para un control efectivo.

La ausencia de una plataforma centralizada y robusta dificulta la gestión de pedidos y reservas, el seguimiento del inventario en tiempo real y la correcta administración de la información de los clientes.

El desarrollo de esta solución, utilizando tecnologías como ASP.NET Core MVC y SQL Server, permitirá digitalizar y centralizar estos procesos. El sistema transformará la gestión manual en un flujo de trabajo ágil y preciso, lo que no solo mejorará la eficiencia operativa y la satisfacción del cliente, sino que también proporcionará una base sólida para el crecimiento del negocio.

## Descripción del sistema

El sistema en cuestión es una plataforma diseñada para optimizar los procesos del área de ventas del restaurante "El Sabor Cajabambino". Desarrollado bajo la arquitectura de software Modelo-Vista-Controlador (MVC), el sistema utiliza el framework ASP.NET Core MVC como su eje tecnológico.

Asimismo, para la persistencia y gestión de los datos, se ha implementado una base de datos relacional en SQL Server. Esta base de datos almacena de manera estructurada toda la información crítica del negocio, como pedidos, inventario de cocina, estado de las mesas y datos de los clientes.

Además, el núcleo funcional del sistema se basa en la implementación de operaciones CRUD (Crear, Leer, Actualizar, Borrar) en cada uno de los módulos principales. Esto permite al personal del restaurante realizar las siguientes tareas de manera eficiente:

* Gestión de pedidos y reservas: Administrar y registrar nuevas órdenes y reservas, así como consultar y actualizar su estado.
* Control de inventario: Mantener un registro detallado del stock de ingredientes, facilitando el control y la reposición.
* Monitoreo de mesas: Visualizar en tiempo real la ocupación y disponibilidad de las mesas para una asignación ágil.
* Administración de clientes: Mantener una base de datos de clientes para mejorar la atención y personalización del servicio.
* En conjunto, la sinergia entre ASP.NET Core MVC y SQL Server, junto con la implementación de estas funcionalidades CRUD, provee una solución robusta y escalable que busca modernizar y mejorar la operación diaria del restaurante.

## Objetivos

### Objetivo General

Diseñar y desarrollar un sistema CRUD web para optimizar los procesos del área de ventas en el restaurante "El Sabor Cajabambino", utilizando el framework ASP.NET Core MVC para la gestión eficiente de datos e interacción con los usuarios.

### Objetivos Específicos

* Diseñar e implementar un modelo de base de datos relacional que sustente los procesos de venta, inventario, pedidos y clientes, garantizando la integridad y consistencia de los datos.
* Implementar un módulo de gestión de pedidos y reservas que permita al personal del restaurante crear, consultar, modificar y eliminar registros, mejorando la organización de las ventas y la atención al cliente.
* Crear una interfaz para la gestión del inventario de cocina, que facilite el registro de ingresos y salidas de productos, permitiendo un control actualizado del stock disponible.
* Desarrollar un panel de visualización del estado de las mesas, que permita al personal monitorear en tiempo real la disponibilidad y ocupación, agilizando la asignación de mesas a los clientes.
* Desarrollar un módulo de administración de clientes que permita registrar, consultar y actualizar la información de los usuarios para una mejor personalización del servicio.

## Descripción de los Stakeholders

**Gerente General**

El dueño de "El Sabor Cajabambino" es el Sr. José Carlos Álvarez. Empresario Experto: Con más de 15 años en el rubro de restaurantes, lo que le ha proporcionado un conocimiento profundo del mercado y las preferencias de los comensales en Cajamarca.

**Estudiantes UNC**

Los estudiantes de Ingeniería de Sistemas del 5to ciclo de la Universidad Nacional de Cajamarca que participan en el curso de Programación Aplicada representan un grupo de stakeholders clave en este proyecto. Poseen una sólida base en fundamentos de programación, bases de datos y análisis de sistemas, lo que les permite comprender la arquitectura y los procesos necesarios para la implementación de una solución para la gestión de datos.

# MODELADO DEL NEGOCIO

## Descripción de la empresa u organización

El restaurante el sabor cajabambino es una empresa dedicada a transmitir el sabor típico de Cajamarca, que se esfuerza por ofrecer una experiencia culinaria que celebre la rica tradición gastronómica de la región, utilizando ingredientes frescos y recetas transmitidas de generación en generación y su guía fundamental es una visión clara y una misión enfocada en nuestro compromiso con la sociedad cajamarquina, buscando no solo deleitar paladares, sino también contribuir al desarrollo cultural y económico local.

**Razón social**

MEDINA LEZAMA, MARIA BERANIZ

**Ubicación**

Av. La Paz #720, Cajamarca, Cajamarca, Perú.

**Rubro Económico**

Elaboración y venta de alimentos preparados.

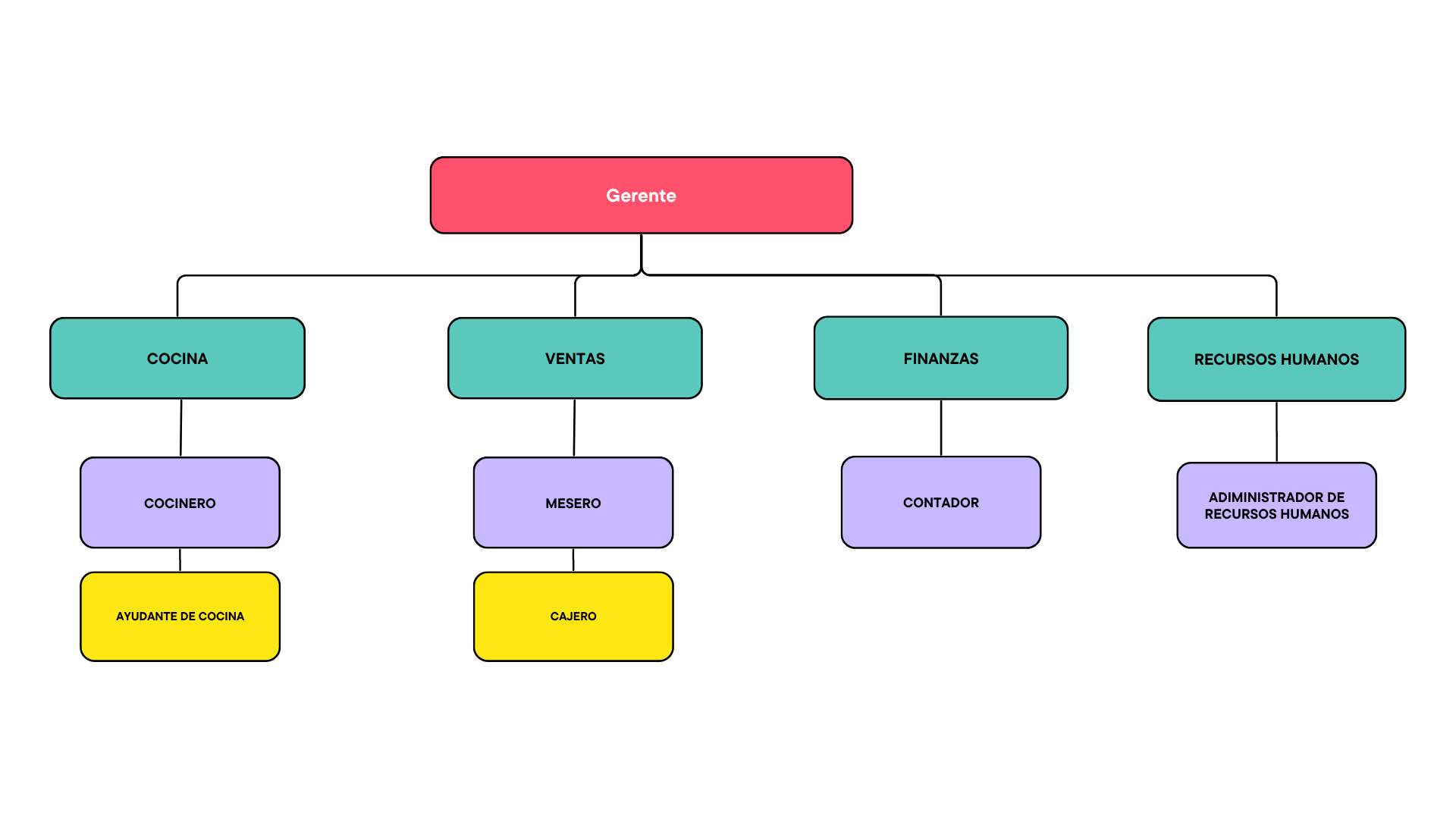
**Clientes**

Público diverso que incluye residentes locales, turistas nacionales e internacionales, familias, parejas, grupos de amigos y empresas. Todos ellos comparten el interés por disfrutar de la auténtica tradición culinaria cajamarquina en un ambiente acogedor que se enriquece con la experiencia de música en vivo, ideal para comidas cotidianas, celebraciones especiales y eventos corporativos.

**Competidores**

Otros restaurantes de comida típica cajamarquina en la ciudad u otros que ofrecen platos tradicionales peruanos en general.

## Organigrama de la empresa



## Recursos

### Personal

Alumnos de la Escuela de Ingeniería de Sistemas. (4 alumnos)

### Hardware y Software

* Laptops
* PC
* Visual Studio Community 2022
* SQL Server
* ASP.net Core MVC
* Gemini
* Bootstrap
* CSS
* HTML
* C#
* Páginas Razor
* Figma

### Otros

* Google Forms

## Alcance

El alcance de esta propuesta se limita principalmente al área de ventas y, de manera complementaria, al área de inventario del restaurante "El Sabor Cajabambino". Si bien otras áreas del restaurante pueden generar datos importantes, el enfoque primordial de esta aplicación recaerá en las transacciones de ventas y la gestión de inventario, dado que son los pilares fundamentales para el análisis de rentabilidad y la optimización de recursos en el negocio.

### Dentro del alcance

* Plantear la administración del proyecto.
* Analizar los requerimientos.
* Diseñar la base de datos relacional.
* Los mockups con Figma.
* Desarrollar un sistema simplificado como propuesta

### Fuera del alcance

* El proyecto no incluirá la exposición o visualización de información que pueda perjudicar o no se alinee con los objetivos estratégicos y la confidencialidad de la empresa. Asimismo, no se realizará el despliegue ni mantenimiento del sistema.

## Tareas

Las tareas a realizar se enfocarán en el desarrollo de una aplicación web para el área de ventas del restaurante "El Sabor Cajabambino". El trabajo principal consistirá en la implementación de funcionalidades CRUD (Crear, Leer, Actualizar, Borrar) para una gestión eficiente de los procesos clave del negocio. Las tareas se dividirán en los siguientes módulos funcionales:

* **Gestión de Ventas y Pedidos:** Se desarrollará un módulo para registrar y administrar los pedidos realizados en el restaurante, así como para gestionar las reservas de mesas. Esto permitirá un seguimiento detallado de las transacciones y agilizará el servicio al cliente.
* **Control de Inventario:** Se implementará una funcionalidad para gestionar el inventario de cocina, lo que incluye el registro de ingresos y salidas de productos. El objetivo es mantener un control preciso del stock y evitar desabastecimientos.
* **Administración de Mesas:** Se creará un módulo para visualizar y gestionar el estado de las mesas del restaurante (disponible, ocupada, reservada, mantenimiento). Esto facilitará la asignación de mesas y optimizará la atención en el local.
* **Gestión de Clientes:** Se desarrollará un apartado para almacenar y administrar la información de los clientes, lo que permitirá un mejor servicio y la posibilidad de futuras estrategias de fidelización.

# ANALISIS DE REQUERIMIENTOS

## El Negocio

### Misión

En Sabor Cajabambino, nuestra misión es celebrar la riqueza de la cocina de Cajabamba y Cajamarca, compartiendo con cada visitante una experiencia gastronómica auténtica y acogedora. Ofrecemos un viaje de sabores que abarca desde los platos típicos transmitidos por generaciones hasta opciones a la carta cuidadosamente elaboradas, todo en un ambiente que honra nuestra cultura, conecta a las personas y evoca la calidez de nuestra tierra.

### Visión

Visualizamos a Sabor Cajabambino como un legado de sabor y hospitalidad donde cada experiencia gastronómica, arraigada en nuestra herencia culinaria de Cajabamba y Cajamarca, trasciende el presente y perdura en el tiempo. Buscamos expandir nuestro alcance sin perder la esencia que nos define: el sabor auténtico, la calidez de nuestra hospitalidad y el orgullo inquebrantable por nuestra identidad cultural.

### Objetivos organizacionales

#### Generales

Crear experiencias gastronómicas memorables que celebren los sabores únicos de Cajabamba y Cajamarca, construyendo una base de clientes leales y asegurando el crecimiento sostenido de Sabor Cajabambino.

#### Específicos

* + - Fomentar un sentido de pertenencia fuerte en el equipo de trabajo.
    - Construir una comunidad de comensales leales.
    - Realizar alianzas estratégicas con proveedores que ofrezcan la mejor calidad de insumos.

## Descripción de los procesos del negocio

El conjunto de procesos a modelar comprenderá la operación integral del restaurante Sabor Cajabambino:

**El proceso de atención al cliente:** Este proceso empieza desde el momento en que el cliente llega al restaurante, incluyendo la recepción, la asignación de mesas (con o sin reserva), la presentación del menú. la toma de pedidos, el servicio de bebidas y alimentos, la atención a consultas y solicitudes, la presentación de la cuenta, el cobro y la despedida.

**El proceso de gestión de pedidos para llevar/a domicilio:** Este proceso empieza desde la recepción del pedido, su registro, la comunicación a la cocina, el empaquetado adecuado, la gestión del pago y la entrega al cliente, se enfocará en la precisión, la rapidez y la calidad de la entrega.

**El proceso de gestión de inventario de la cocina y el almacén:** Control de stock, el registro de las salidas para la preparación de los platos y la generación de pedidos de reposición.

## Problemas del negocio

La insuficiente digitalización de procesos operativos como las ventas y la gestión del inventario limitan la agilidad y la eficiencia del negocio, lo que se traduce en una capacidad restringida para optimizar la experiencia del cliente, escalar las operaciones de manera efectiva ante la diversificación de la oferta gastronómica y la futura implementación de canales digitales, y mantener una ventaja competitiva en el mercado.

## Selección de los entrevistados

En este caso se ha seleccionado a José Carlos Álvarez el dueño del restaurante “El sabor Cajabambino” ya que es el único con disponibilidad para realizar las entrevistas y el recojo de datos necesarios.

## Entrevista

El proceso de entrevista es fundamental para comprender a fondo las necesidades y desafíos del restaurante "El Sabor Cajabambino" en relación con sus datos. Se realizaron preguntas estructuradas y abiertas con el gerente del negocio, el Sr. José Carlos Álvarez (Dueño/Gerente General). El objetivo es indagar sobre sus tareas diarias, su perspectiva sobre el desempeño en las áreas de interés y los retos que enfrenta en la gestión y análisis de la información. Se prestó especial atención a la toman decisiones actualmente y qué tipo de información o herramientas desearía tener en el futuro para mejorar este proceso. Esta interacción directa permitió obtener una visión cualitativa de las expectativas y las áreas de mejora desde la perspectiva de quien administra el negocio.

### Definición de requerimientos del negocio

**Entrevista para obtener información que sirva de apoyo a la toma de decisiones en el área de recursos humanos de la empresa EL SABOR CAJABAMBINO**

***Estimado colaborador:*** *El propósito de esta encuesta es comprender mejor los desafíos y necesidades relacionadas con la* ***gestión de ventas e inventario*** *en "El Sabor Cajabambino".*

*La información recopilada será fundamental para el diseño y el desarrollo de un sistema de gestión web para el área de ventas que permita optimizar estos procesos, mejorar la toma de decisiones y, en última instancia, beneficiar el crecimiento y la eficiencia de nuestro restaurante.*

**Lugar de la entrevista:** Restaurant el Sabor Cajabambino

**INSTRUCCIONES:** Por favor, lea cuidadosamente cada pregunta y marque con una (X) la respuesta que mejor se ajuste a su realidad, o escriba su respuesta donde sea necesario. ¡Gracias por su valioso tiempo!

**1. Actualmente, ¿cómo registra y controla las ventas diarias en el restaurante?**

1. *Manualmente (cuadernos, hojas de cálculo básicas)*
2. A través de un sistema de punto de venta (POS)
3. Opción 4
4. *Otros: Sistema de registro de ventas alquilado a terceros*

**2. ¿Qué tan fácil le resulta obtener información sobre las ventas de días, semanas o meses anteriores?**

1. Muy fácil
2. Fácil
3. *Regular*
4. Difícil
5. Muy difícil

**3. ¿Con qué frecuencia analiza los datos de ventas para identificar productos más vendidos o patrones de consumo?**

1. Diariamente
2. Semanalmente
3. *Mensualmente*
4. Nunca
5. *Otros: No es tanto un análisis, reviso los datos sobre todo para llevar la contabilidad y no tener problemas con SUNAT.*

**4. ¿Considera que la información actual sobre ventas es suficiente para tomar decisiones sobre promociones o cambios en el menú?** **¿Por qué?**

1. Sí
2. *No*

La información de las ventas sirve para llevar la contabilidad. No la uso regularmente para tomar decisiones

**5. ¿Cómo realiza actualmente el control de inventario de materias primas e insumos en el restaurante?**

1. *Manualmente (fichas, conteo físico)*
2. A través de un sistema de inventario o un módulo de POS
3. Otros:

**6. ¿Qué tan fácil es saber en cualquier momento la cantidad exacta de un ingrediente o producto disponible?**

1. Muy fácil
2. Fácil
3. *Regular*
4. Difícil
5. Muy difícil

**7. ¿Ha experimentado problemas de desabastecimiento o exceso de inventario de algún producto o ingrediente?**

1. Sí, frecuentemente
2. *Sí, ocasionalmente*
3. No, rara vez
4. Nunca

**Si su respuesta es sí, ¿cuáles han sido las consecuencias (ej. pérdidas de ventas, productos vencidos)?**

El encargado de abastecer el inventario y gestionarlo es el cocinero, por lo que no sé cómo maneja todo ese proceso. Pero ha existido situaciones donde no se ha logrado un abastecimiento adecuado.

**8. ¿Qué dificultades encuentra en la gestión de inventario para el restaurante (ej. seguimiento de ingredientes, control de stock, pérdidas)?**

No puedo llevar un seguimiento adecuado del inventario. El sistema que alquilo tiene un apartado de inventario, pero nunca lo he usado.

**9. ¿Qué piensa de la automatización de los procesos de registro de ventas e inventario en "El Sabor Cajabambino" a través de un sistema de gestión web? ¿Cómo cree que le ayudaría en su trabajo?**

No tengo mucho conocimiento sobre los detalles del sistema, pero según lo que me han explicado creo que sería una herramienta muy útil para llevar una gestión mucho más práctica y eficiente de las ventas y el inventario.

**10. ¿Hay alguna otra información o comentario que considere relevante para el diseño de este sistema"?**

A para organizar mejor a los clientes y tener un registro más claro de las actividades en el momento, quizá una forma de visualizar el estado de las mesas.

**11. Indique el rango en el que se encuentra su rendimiento laboral.**

1. Del 75% - 100%
2. *Del 50% - 75%*
3. Del 25% - 50%
4. Del 0% - 25%

**12.** **Qué piensas de la gestión de los datos de ventas e inventario en la mejora del desempeño, ¿Cómo cree Ud. que le ayudaría?**

Creo que sería una herramienta interesante de implementar en mi negocio. Definitivamente, sería útil.

**Gracias por tu tiempo.**

## Resumen de los requerimientos obtenidos en la entrevista

Los requerimientos recopilados según las entrevistas, encuestas y evaluación de reportes fueron:

### Requerimientos Funcionales

* **Gestión de Pedidos y Reservas de Mesas**: Es fundamental contar con un sistema que facilite la toma de pedidos, la asignación de mesas y la gestión de reservas. Esto mejorará la organización del servicio y agilizará la atención a los clientes, reduciendo los tiempos de espera. Mejora en el acceso a la información de ventas: Urge optimizar la facilidad de acceso y consulta de la información de ventas diarias, semanales y mensuales del sistema alquilado, ya que el acceso a datos históricos es "Regular".
  + Permitir el registro de pedidos por parte del
  + Registrar y gestionar reservas de clientes
  + Visualizar el historial de pedidos
* **Control del Inventario de Ingredientes**: Se necesita un módulo para llevar un registro preciso del inventario de cocina. El objetivo es poder registrar entradas y salidas de productos de manera sencilla, lo que permitirá tener un control del stock en tiempo real y evitar problemas de desabastecimiento o mermas innecesarias. Organización y evaluación del desempeño de meseros: Se necesita organizar mejor a los meseros y tener un registro más claro de su actividad para evaluar su desempeño.
  + Modificar los registros de inventario
  + Mostrar el stock actual
  + Generar alertas por bajo inventario
* **Administración del Estado de las Mesas**: Se requiere una interfaz visual y práctica que permita al personal conocer el estado actual de cada mesa (disponible, ocupada, reservada). Esto ayudará a una mejor planificación del servicio y una asignación más eficiente de los espacios.
  + Visualizar el estado actual de cada mesa (Disponible, ocupada, reservada, mantenimiento)
  + Permitir actualizar el estado de cada mesa
* **Organización y registro de los datos de los Colaboradores**
  + Listar los colaboradores
  + Editar los datos de los colaboradores
* **Registro y Administración de Clientes**: Existe la necesidad de crear un registro de clientes que capture información básica como nombres y datos de contacto. Esto servirá como una base para mejorar la personalización del servicio y para futuras campañas de fidelización.
  + Listar y gestionar los registros de los clientes.
  + Buscar clientes

### Requerimientos no funcionales

* **Usabilidad**
  + Interfaz amigable
  + Diseño responsivo
* **Seguridad**
  + Autenticación con roles diferenciados (admin, mesero, cocina)
  + Protección de datos

## Base de Datos Relacional

La base de datos Fuente será una base de datos relacional SQL desarrollada en SQL Server, la cual cuenta con diferentes esquemas y tablas que contienen información de cinco meses de pedidos.

### Esquema GENERAL

Este esquema contiene información general y de catálogo del restaurante, incluyendo productos y la disposición de las mesas. Tablas:

#### Categoria

* + Id\_Categoria
  + Nombre
  + Descripcion

#### Mesa

* + Id\_Mesa
  + Capacidad
  + Ubicación
  + Estado

#### Producto

* + Id\_Producto
  + Nombre
  + Precio
  + Descripcion
  + Foto
  + Id\_Categoria
  + EsPreparado

#### ProductoIngrediente

* + Id\_Producto
  + Id\_Item
  + Cantidad

### Esquema CLIENTE

Este esquema gestiona la información de los clientes y sus reservas. Tablas:

* **Cliente**
  + Id\_Cliente
  + Nombres
  + ApellidoPaterno
  + ApellidoMaterno
  + DNI
  + Telefono
  + CorreoElectronico
  + Direccion
  + FechaDeNacimiento
* **Reserva**
  + Id\_Reserva
  + Fecha
  + Hora
  + Id\_Mesa
  + Id\_Cliente
  + NumeroPersonas
  + Estado
  + Comentarios

### Esquema INVENTARIO

Este esquema se encarga de la gestión de inventario de materias primas e insumos. Tablas:

* **Inventario**
  + Id\_Item
  + ItemNombre
  + Id\_ItemCategoria
  + UnidadMedida
  + Stock
  + CostoPorUnidad
  + FechaDeExpiracion
  + NivelReorden
  + CantidadReorden
  + NecesitaReorden
* **ItemCategoria**
  + Id\_ItemCategoria
  + Categoria
  + Descripcion

### Esquema PERSONAL

Este esquema almacena la información de los empleados del restaurante. Tablas:

* **Empleado**
  + Id\_Empleado
  + NombreCompleto
  + DNI
  + FechaNacimiento
  + Direccion
  + Telefono
  + CorreoElectronico
  + Rol
  + Turno
  + FechaContratacion
  + Salario
  + Estado
  + Usuario
  + Contrasenia

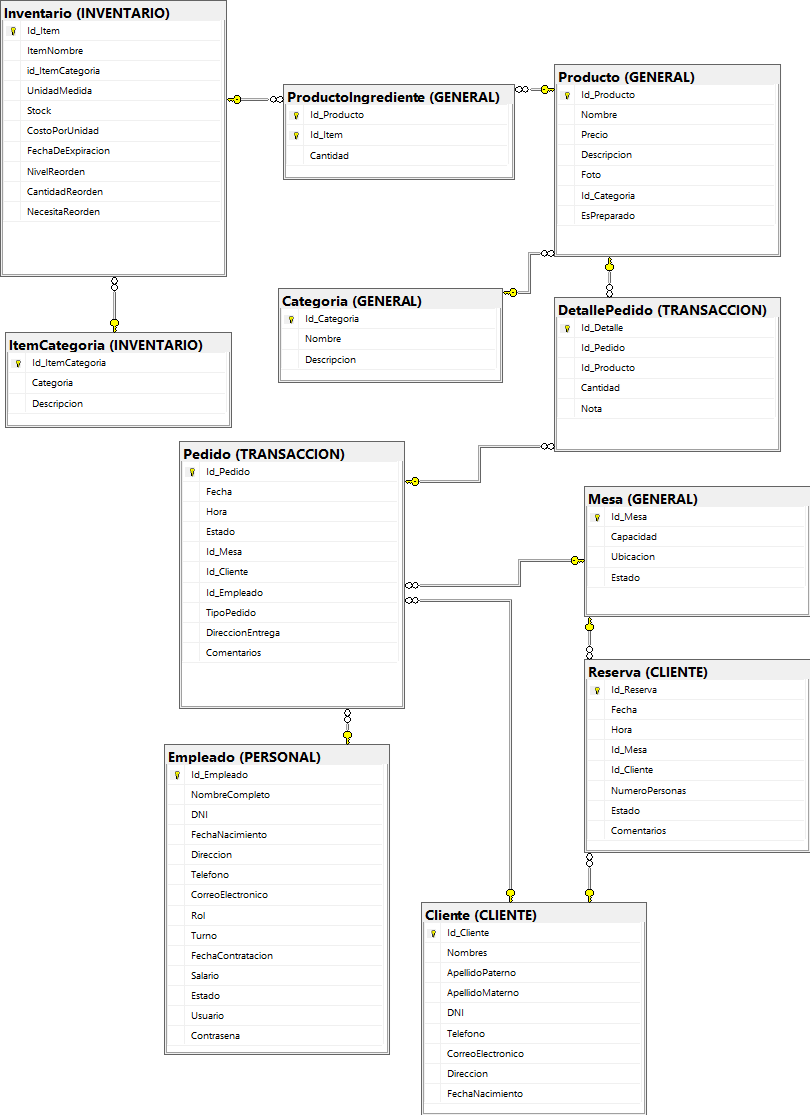
### Esquema TRANSACCION

Este esquema registra todos los detalles de los pedidos y sus transacciones. Tablas:

* **DetallePedido**
  + Id\_Detalle
  + Id\_Pedido
  + Id\_Producto
  + Cantidad
  + PrecioUnitario
  + Nota
* **Pedido**
  + Id\_Pedido
  + Fecha
  + Hora
  + Estado
  + Id\_Mesa
  + Id\_Cliente
  + Id\_Empleado
  + TipoPedido
  + DireccionEntrega
  + Comentarios

## Diagrama Relacional

**Figura 1**  
Diagrama base de datos



# MARCO TEÓRICO

## Bases de Datos Relacionales (RDBMS):

### Base de datos

Una base de datos (BD) es un conjunto estructurado de datos integrados, almacenados en un soporte no volátil (como discos o cintas), con redundancia controlada y diseñados para ser compartidos por múltiples usuarios y aplicaciones. Los datos se organizan de manera independiente de los programas que los utilizan, siguiendo un modelo de datos que captura las interrelaciones y restricciones del mundo real. La BD incluye mecanismos para la actualización, recuperación y seguridad de los datos [1].

### Base de datos relacional

Una base de datos relacional es un tipo de BD basada en el modelo relacional propuesto por Edgar Codd en 1970. En este modelo, los datos se organizan en tablas (relaciones) compuestas por filas (tuplas) y columnas (atributos). Cada tabla tiene una clave primaria que identifica registros únicos, y las relaciones entre tablas se establecen mediante claves foráneas. Este modelo permite operaciones eficientes mediante el lenguaje SQL (Structured Query Language) y garantiza la integridad de los datos mediante restricciones como la integridad referencial [1].

### Microsoft SQL Server:

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) desarrollado por Microsoft. Proporciona un entorno robusto para almacenar, recuperar y administrar datos mediante el lenguaje Transact-SQL (T-SQL), una extensión de SQL estándar con características adicionales como procedimientos almacenados, triggers y manejo avanzado de transacciones. SQL Server incluye herramientas para análisis de datos, inteligencia empresarial (BI) y soporte para aplicaciones críticas en entornos empresariales [2].

### T-SQL

Transact-SQL (T-SQL) es el lenguaje de programación utilizado en SQL Server para interactuar con la base de datos. Extiende el estándar SQL con funcionalidades como [2]:

* Control de flujo: Estructuras IF-ELSE, WHILE.
* Procedimientos almacenados: Bloques de código reutilizables.
* Manejo de errores: TRY-CATCH.
* Funciones personalizadas. T-SQL permite optimizar consultas, automatizar tareas y garantizar la integridad de los datos.

## Operaciones CRUD

### Definición

CRUD es un acrónimo que representa las cuatro operaciones básicas en el manejo de datos persistentes en sistemas de bases de datos y aplicaciones [3]:

* **Create (Crear)**: Insertar nuevos registros en una base de datos.
* **Read (Leer)**: Consultar o recuperar datos existentes.
* **Update (Actualizar)**: Modificar registros existentes.
* **Delete (Eliminar)**: Borrar registros.

## Patrón Modelo-Vista-Controlador (MVC)

El patrón MVC es un modelo de diseño de software que busca organizar la aplicación en tres componentes principales, cada uno con una responsabilidad específica. Esta separación de responsabilidades es fundamental para lograr un diseño limpio y modular, facilitando el desarrollo, la depuración y el mantenimiento del software [3].

* + **Modelo (Model):** Representa los datos y la lógica de negocio de la aplicación. Es el "cerebro" que gestiona la información, las reglas y las operaciones que se pueden realizar con esos datos. Por ejemplo, en una aplicación de comercio electrónico, el Modelo podría manejar la información de los productos, los pedidos de los clientes y las reglas para calcular precios o descuentos. El Modelo es independiente de la interfaz de usuario [3].
  + **Vista (View):** Es la interfaz de usuario de la aplicación. Su única responsabilidad es mostrar la información al usuario, basándose en los datos que recibe del Modelo. La Vista no contiene lógica de negocio ni manipula directamente los datos; simplemente los presenta de una manera visualmente atractiva. Siguiendo el ejemplo del comercio electrónico, la Vista sería la página web que muestra la lista de productos o el carrito de compras [3].
  + **Controlador (Controller):** Actúa como el intermediario entre el Modelo y la Vista. Es el componente que recibe las solicitudes del usuario (por ejemplo, un clic en un botón, la entrada de datos en un formulario), interactúa con el Modelo para realizar las operaciones necesarias (como recuperar datos o guardar información) y, finalmente, selecciona la Vista adecuada para mostrar la respuesta al usuario. El Controlador es el "director de orquesta" que coordina la interacción entre los otros dos componentes [3].

## ASP.NET Core MVC

ASP.NET Core MVC es un marco de trabajo moderno, ligero, de código abierto y altamente probado, diseñado para construir sitios web dinámicos y API web. Está optimizado para funcionar con ASP.NET Core, la plataforma de desarrollo web de Microsoft. Proporciona una forma estructurada y basada en patrones para desarrollar aplicaciones web, ofreciendo un control total sobre el marcado HTML y promoviendo prácticas de desarrollo como el Desarrollo Dirigido por Pruebas (TDD) [3].

# DESARROLLO DEL SISTEMA

## Entorno y Tecnologías de Desarrollo

* **Lenguaje de Programación:** C#.
* **Framework:** ASP.NET Core MVC.
* **Gestor de Base de Datos:** SQL Server.
* **ORM (Mapeador Objeto-Relacional):** **Entity Framework Core**, que es la tecnología que te permite interactuar con la base de datos a través de tus clases de C#.
* **Entorno de Desarrollo Integrado (IDE):** Microsoft Visual Studio Community 2022 Versión 17.14.5

## **Implementación del patrón MVC**

### Modelos (Models):

* *Models/Categoria.cs*

|  |
| --- |
| public partial class Categoria  {  public int IdCategoria { get; set; }  public string Nombre { get; set; } = null!;  public string? Descripcion { get; set; }  public virtual ICollection<Producto> Productos { get; set; } = new List<Producto>();  } |

* *Models/Cliente.cs*

|  |
| --- |
| public partial class Cliente  {  public int IdCliente { get; set; }  public string Nombres { get; set; } = null!;  public string ApellidoPaterno { get; set; } = null!;  public string ApellidoMaterno { get; set; } = null!;  public string Dni { get; set; } = null!;  public string? Telefono { get; set; }  public string? CorreoElectronico { get; set; }  public string? Direccion { get; set; }  public DateOnly? FechaNacimiento { get; set; }  public virtual ICollection<Pedido> Pedidos { get; set; } = new List<Pedido>();  public virtual ICollection<Reserva> Reservas { get; set; } = new List<Reserva>();  } |

* *Models/DetallePedido.cs*

|  |
| --- |
| public partial class DetallePedido  {  public int IdDetalle { get; set; }  public int IdPedido { get; set; }  public int IdProducto { get; set; }  public int Cantidad { get; set; }  public string? Nota { get; set; }  public virtual Pedido IdPedidoNavigation { get; set; } = null!;  public virtual Producto IdProductoNavigation { get; set; } = null!;  } |

* *Models/Empleado.cs*

|  |
| --- |
| public partial class Empleado  {  public int IdEmpleado { get; set; }  public string NombreCompleto { get; set; } = null!;  public string Dni { get; set; } = null!;  public DateOnly? FechaNacimiento { get; set; }  public string? Direccion { get; set; }  public string? Telefono { get; set; }  public string? CorreoElectronico { get; set; }  public string Rol { get; set; } = null!;  public string Turno { get; set; } = null!;  public DateOnly? FechaContratacion { get; set; }  public decimal? Salario { get; set; }  public string? Estado { get; set; }  public string Usuario { get; set; } = null!;  public string Contrasena { get; set; } = null!;  public virtual ICollection<Pedido> Pedidos { get; set; } = new List<Pedido>();  } |

* *Models/Inventario.cs*

|  |
| --- |
| public partial class Inventario  {  public int IdItem { get; set; }  public string ItemNombre { get; set; } = null!;  public int IdItemCategoria { get; set; }  public string? UnidadMedida { get; set; }  public decimal? Stock { get; set; }  public decimal CostoPorUnidad { get; set; }  public DateOnly? FechaDeExpiracion { get; set; }  public decimal? NivelReorden { get; set; }  public decimal? CantidadReorden { get; set; }  public bool? NecesitaReorden { get; set; }  public virtual ItemCategorium? IdItemCategoriaNavigation { get; set; }  public virtual ICollection<ProductoIngrediente> ProductoIngredientes { get; set; } = new List<ProductoIngrediente>();  } |

* *Models/ItemCategorium.cs*

|  |
| --- |
| public partial class ItemCategorium  {  public int IdItemCategoria { get; set; }  public string Categoria { get; set; } = null!;  public string? Descripcion { get; set; }  public virtual ICollection<Inventario> Inventarios { get; set; } = new List<Inventario>();  } |

* *Models/Mesa.cs*

|  |
| --- |
| public partial class Mesa  {  public int IdMesa { get; set; }  public int? Capacidad { get; set; }  public string? Ubicacion { get; set; }  public string? Estado { get; set; }  public virtual ICollection<Pedido> Pedidos { get; set; } = new List<Pedido>();  public virtual ICollection<Reserva> Reservas { get; set; } = new List<Reserva>();  } |

* *Models/Pedido.cs*

|  |
| --- |
| public partial class Pedido  {  public int IdPedido { get; set; }  public DateOnly Fecha { get; set; }  public TimeOnly Hora { get; set; }  public string Estado { get; set; } = null!;  public int? IdMesa { get; set; }  public int? IdCliente { get; set; }  public int? IdEmpleado { get; set; }  public string? TipoPedido { get; set; }  public string? DireccionEntrega { get; set; }  public string? Comentarios { get; set; }  public virtual ICollection<DetallePedido> DetallePedidos { get; set; } = new List<DetallePedido>();  public virtual Cliente? IdClienteNavigation { get; set; }  public virtual Empleado? IdEmpleadoNavigation { get; set; }  public virtual Mesa? IdMesaNavigation { get; set; } } |

* *Models/Producto.cs*

|  |
| --- |
| public partial class Producto  {  public int IdProducto { get; set; }  public string Nombre { get; set; } = null!;  public decimal Precio { get; set; }  public string? Descripcion { get; set; }  public string? Foto { get; set; }  public int IdCategoria { get; set; }  public bool? EsPreparado { get; set; }  public virtual ICollection<DetallePedido> DetallePedidos { get; set; } = new List<DetallePedido>();  public virtual Categoria IdCategoriaNavigation { get; set; } = null!;  public virtual ICollection<ProductoIngrediente> ProductoIngredientes { get; set; } = new List<ProductoIngrediente>();  } |

* *Models/ProductoIngrediente.cs*

|  |
| --- |
| public partial class ProductoIngrediente  {  public int IdProducto { get; set; }  public int IdItem { get; set; }  public decimal Cantidad { get; set; }  public virtual Inventario IdItemNavigation { get; set; } = null!;  public virtual Producto IdProductoNavigation { get; set; } = null!;  } |

* *Models/Reserva.cs*

|  |
| --- |
| public partial class Reserva  {  public int IdReserva { get; set; }  public DateOnly Fecha { get; set; }  public TimeOnly Hora { get; set; }  public int? IdMesa { get; set; }  public int? IdCliente { get; set; }  public int NumeroPersonas { get; set; }  public string? Estado { get; set; }  public string? Comentarios { get; set; }  public virtual Cliente? IdClienteNavigation { get; set; }  public virtual Mesa? IdMesaNavigation { get; set; }  } |

* **Controladores (Controllers):**
* **Vistas (Views):**

## Implementación de las Funcionalidades CRUD

* **Módulo de Gestión de Pedidos:**
* **Módulo de Gestión de Inventario:**
* **Módulo de Gestión de Mesas:**
* **Módulo de Gestión de Clientes:**
* **Desafíos y Soluciones**

# PRUEBA DEL SISTEMA DE GESTIÓN WEB

## Objetivo de las Pruebas

* **Objetivo:** El objetivo principal de esta etapa fue validar que las funcionalidades desarrolladas en el sistema de gestión web operen de acuerdo con los requerimientos establecidos. Se buscó asegurar que todas las operaciones CRUD (Crear, Leer, Actualizar, Borrar) funcionen correctamente en cada uno de los módulos.
* **Enfoque de las Pruebas:** Se utilizó un enfoque de **pruebas funcionales (caja negra)**, lo que significa que se validó el comportamiento de la aplicación desde la perspectiva del usuario, sin inspeccionar el código interno.

## Metodología de Pruebas

* **Pruebas Funcionales:** Se verificó el correcto funcionamiento de cada una de las funcionalidades del sistema (creación de pedidos, actualización de inventario, etc.).
* **Pruebas de Integración:** Se comprobó que los diferentes componentes del sistema (la interfaz de usuario, la lógica de negocio y la base de datos) interactúen entre sí sin errores. Por ejemplo, que los datos guardados desde un formulario se reflejen correctamente en la base de datos.
* **Pruebas de Usabilidad (Opcional):** Se hizo una verificación simple para asegurar que la navegación sea intuitiva y la interfaz sea clara y fácil de usar.

## Casos de Prueba y Resultados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Prueba | Pasos de Prueba | Resultado Esperado | Resultado Obtenido | Estado |
| C1: Crear un nuevo pedido. | 1. Ir a la página de "Pedidos". 2. Llenar el formulario con datos válidos. 3. Hacer clic en "Guardar". | El pedido se registra en la base de datos y se muestra en la lista. | El pedido se guardó y se visualizó correctamente. | APROBADO |
| C2: Visualizar la lista de productos. | 1. Ir a la página de "Inventario". 2. Verificar que todos los productos se muestren. | La tabla de productos muestra todos los registros de la base de datos. | La lista se cargó sin errores y mostró la información correcta. | APROBADO |
| C3: Actualizar el estado de una mesa. | 1. Ir al panel de "Mesas". 2. Cambiar el estado de la mesa 5 de "Libre" a "Ocupada". | El estado de la mesa 5 se actualiza visualmente y en la base de datos. | El estado se actualizó correctamente. | APROBADO |

Exportar a Hojas de cálculo

* **Nota:** Incluye al menos dos o tres casos de prueba por cada módulo para demostrar que verificaste las funcionalidades más importantes (crear, leer, actualizar, borrar).

## Conclusiones de las Pruebas

### Resumen

Describe el total de casos de prueba ejecutados y el porcentaje de éxito.

### Problemas Encontrados (y Solucionados)

Es buena idea mencionar uno o dos errores menores que encontraste durante la verificación y cómo los corregiste. Por ejemplo: "Se encontró un error al intentar eliminar un pedido que no existía, el cual fue corregido implementando una validación previa." Esto demuestra una capacidad de resolución de problemas.

### Conclusión

Afirma que las pruebas realizadas confirman que el sistema cumple con los requerimientos funcionales y está listo para su uso, aunque podrías mencionar que futuras pruebas de carga o seguridad podrían ser beneficiosas.

# CONCLUSIONES

# RECOMENDACIONES

# REFERENCIAS

[1] D. Cuadra, E. Castro y P. Martínez, *Diseño de Bases de Datos*. Ministerio de Educación, Cultura y Deporte, 2003.

[2] Microsoft, "¿Qué es SQL Server?," *Documentación oficial de Microsoft*, 2023. Disponible: <https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver17>.

[3] Microsoft Learn. (2025). *Introducción a ASP.NET Core MVC*. Recuperado de <https://learn.microsoft.com/es-mx/aspnet/core/mvc/overview?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website#filters>

# ANEXOS

ANEXO N° 1. Proceso de atención al client