**Carrera Profesional de Ingeniería de Sistemas Computacionales**

**INFORME**

**MEDICIÓN DE COMPETENCIAS**

**Competencias:**

**Resolución de Problemas**

**Diseño y Desarrollo de Soluciones**

**Uso de Herramientas Modernas**

**Versión 1.3**

**Grupo x**

**Integrantes del proyecto**

| **Número** | **Apellidos y Nombres** |
| --- | --- |
| 1 | CHAVEZ TINTAYA DAVID |
| 2 | Albert Jualian Castro Rivas |
| 3 | Calmette Miyazawa - Franco André |
| 4 | Juan José Vigo Choque |
| 5 | Caspa Huerta Jeremy Angel Zackary |
| 6 |  |

**Agosto del 2024**

**Competencia General: Resolución de Problemas**

Identifica y comprende una situación problemática pudiendo hacer uso de medios físicos y digitales y a partir de ello, diseña ideas o alternativas de solución elaborando un plan para implementarlo y evaluar su impacto. Las ideas o alternativas que proponen son innovadoras y se transforman en oportunidades o acciones concretas que agregan valor.

**C1: Analiza la arquitectura del software para proponer un plano de sistema, analizando la situación organizacional de la empresa para identificar los requisitos.**

* 1. **1.2. HALLAZGOS**
  2. Se ha observado que muchas soluciones actuales de reservas en restaurantes son obsoletas o están mal implementadas, lo que lleva a la insatisfacción del cliente. A nivel global, diversas aplicaciones han sido desarrolladas con éxito, destacando la mejora en la eficiencia operativa y la satisfacción del cliente. En el ámbito local, sin embargo, los sistemas de reservas están menos desarrollados, lo que representa una oportunidad de innovación.
  3. **1.3. PROPUESTAS**
  4. Listar las diferentes propuestas que tuvieron antes de elegir el proyecto que fue seleccionado.
  5. ***Propuesta 01: Red social para comensales y restaurantes.***
  6. Ventajas: Interacción social y marketing.
  7. Desventajas: Enfoque demasiado amplio y complicado.
  8. **Propuesta 02: Sistema web de reservas para la compra de entradas de espectáculos.**
  9. Ventajas: Similar en estructura.
  10. Desventajas: No alineado con el sector restaurantero.
  11. ***Propuesta 03: aplicación para adopción de mascotas***
  12. Ventajas: Socialmente impactante.
  13. Desventajas: No enfocado en el sector gastronómico.
  14. Finalmente, se eligió desarrollar una aplicación centrada exclusivamente en reservas para restaurantes, basada en las necesidades específicas del sector.
  15. **1.4. IMPACTOS**
  16. **Impacto social:** Mejora la experiencia del cliente al permitir reservas más ágiles y cómodas, reduciendo tiempos de espera y garantizando una mejor planificación para los usuarios.
  17. **Impacto cultural:** Promueve el uso de tecnología en un sector tradicional como la restauración, contribuyendo a la digitalización y modernización del servicio en la cultura gastronómica local.
  18. **Impacto político:** Aunque menos directo, puede apoyar regulaciones locales de ocupación en restaurantes, al facilitar un control más preciso de la cantidad de personas en el local, ayudando a cumplir con normativas de aforo.
  19. **Impacto ambiental:** Al gestionar mejor las reservas, se evita el desperdicio de recursos como energía y alimentos, optimizando la cantidad de mesas ocupadas y reduciendo el uso innecesario de insumos.
  20. **Impacto ético:** Garantiza la privacidad y seguridad de los datos personales de los usuarios, un aspecto crítico en aplicaciones tecnológicas.
  21. **Impacto económico:** Aumenta la eficiencia operativa de los restaurantes, lo que puede traducirse en mayores ingresos por una mejor gestión de las reservas y la reducción de sobrecargas.

**C2: Escoge el plano ideal para el sistema que se va a implementar y utiliza herramientas digitales. Las cita adecuadamente.**

Proporciona la interfaz con la que interactúan los usuarios. Se desarrolló utilizando:

* HTML: El lenguaje de marcado de hipertexto que se utiliza para crear la estructura del contenido
* CSS: Las hojas de estilo en cascada que se utilizan para definir cómo se muestra el contenido
* JavaScript: El lenguaje que se utiliza para definir el comportamiento de la página

**Herramientas de Desarrollo y Gestión:**

* Visual Studio Code: Entorno de desarrollo integrado (IDE) utilizado para la implementación del código.
* Git/GitHub: Control de versiones para el seguimiento de cambios y colaboración en equipo.

**C3: Elabora arquitecturas de software e implementa las alternativas de solución, justificando la elección.**

En el desarrollo del sistema de reservas para el restaurante "Fusión y Sazón", se analizaron diferentes arquitecturas y alternativas de solución. Se seleccionó una arquitectura en capas debido a las siguientes ventajas:

1. Organización y Separación de Responsabilidades**:** Esta arquitectura permite dividir el sistema en capas independientes, como interfaz de usuario, lógica de negocio y acceso a datos, facilitando el mantenimiento y la escalabilidad del sistema.
2. Escalabilidad y Mantenimiento: La separación clara entre componentes simplifica la integración de nuevas funcionalidades en el futuro, asegurando la escalabilidad del sistema.
3. Flexibilidad: La arquitectura en capas soporta cambios en una capa específica sin afectar a las demás, lo que agiliza la implementación de mejoras.

**Frontend:** html,css,java script lo que garantiza una interfaz atractiva, reusable y fácil de mantener.

**Backend:** Spring ,java , github.

**Base de Datos:** SQL Server para almacenar y procesar datos de manera eficiente.

**C4: Propone soluciones alineadas al ciclo de vida de desarrollo del software, utilizando herramientas CASE.**

Herramientas CASE:

* Visual Paradigm**:**
  + Se empleó para la creación de diagramas UML, como casos de uso, diagramas de clases, secuencias y despliegue. Esto facilitó una representación clara de los componentes y su interacción.
* SQL Server Management Studio (SSMS):
  + Utilizado para el modelado de la base de datos relacional, desarrollando los diagramas conceptual, lógico y físico.

**C5: Desarrolla el plan de acción para la implementación, utilizando el diagrama de Gantt y EDT.**

Fase 1: Recolección de Requisitos (2 semanas)

* Tarea 1.1: Identificar necesidades de los interesados.
* Tarea 1.2: Documentar requisitos funcionales y no funcionales.
* Tarea 1.3: Validar los requisitos con los interesados.

Fase 2: Diseño del Sistema (3 semanas)

* Tarea 2.1: Modelar casos de uso, clases y secuencias (UML).
* Tarea 2.2: Diseñar la base de datos (modelo conceptual, lógico y físico).
* Tarea 2.3: Prototipar la interfaz de usuario en Figma.

Fase 3: Desarrollo del Sistema (3 semanas)

* Tarea 3.1: Implementar el frontend con React y Tailwind CSS.
* Tarea 3.2: Desarrollar el backend con Spring MVC y API RESTful.
* Tarea 3.3: Integrar la base de datos relacional en SQL Server.
* Tarea 3.4: Configurar contenedores con Docker.

Fase 4: Pruebas y Validación (2 semanas)

* Tarea 4.1: Realizar pruebas unitarias, de integración y rendimiento.
* Tarea 4.2: Ejecutar pruebas de usabilidad con usuarios finales.

Fase 5: Despliegue y Entrega (1 semana)

* Tarea 5.1: Configurar y desplegar el sistema
* Tarea 5.2: Capacitar a los administradores del sistema.

**C6:** **Desarrolla las actividades validadas en el ciclo de vida de desarrollo y el diagrama de Gantt, utilizando el lenguaje UML.**

#### Fase 1: Recolección de Requisitos

* Actividades:
  + Identificar necesidades del cliente y stakeholders.
  + Documentar requisitos funcionales y no funcionales.
  + Validar con los interesados mediante entrevistas y reuniones.
* Diagrama UML: Caso de uso para el proceso de recolección de requisitos.

#### Fase 2: Diseño del Sistema

* Actividades:
  + Diseñar el modelo conceptual, lógico y físico de la base de datos.
  + Crear diagramas de clases, secuencias y actividades.
  + Prototipar interfaces de usuario.
* Diagramas UML:
  + Diagrama de clases: Representa las entidades del sistema (Usuario, Reserva, Menú).
  + Diagrama de secuencia: Describe el flujo de interacción entre el cliente, el sistema y la base de datos para realizar una reserva.

#### Fase 3: Desarrollo del Sistema

* Actividades:
  + Implementación del frontend (React, Tailwind CSS).
  + Desarrollo del backend (Spring MVC) y las APIs RESTful.
  + Integración de la base de datos (SQL Server).
* Diagramas UML:
  + Diagrama de componentes: Muestra los módulos del sistema, incluyendo frontend, backend y base de datos.
  + Diagrama de despliegue: Representa cómo los componentes se ejecutan en el entorno de producción.

#### Fase 4: Pruebas y Validación

* Actividades:
  + Realizar pruebas unitarias, de integración y rendimiento.
  + Validar usabilidad con usuarios finales.
* Diagrama UML: Diagrama de actividades para el flujo de validación de una reserva.

#### Fase 5: Despliegue y Entrega

* Actividades:
  + Configurar el entorno de producción en Azure.
  + Capacitar al personal administrativo para usar el sistema.
* Diagrama UML: Diagrama de despliegue que representa los nodos del sistema (servidor, base de datos y dispositivos de usuario).

**C7: Realiza las actividades utilizando herramientas de arquitectura de software (IBM RSA u otro equivalente).**

**CAPÍTULO III: HERRAMIENTAS DE INGENIERÍA**

* Comparación de metodologías o marco de trabajo y estándares de ingeniería, selección de lo que se utilizará (justificar)

1. Cascada (Waterfall) - No pasa a otra fase sin acabar la previa, es útil para proyectos con requisitos muy claros y bien establecidos desde el inicio, sin posibilidades de cambios frecuentes. No es recomendable para otro tipo de proyectos.
2. Incremental - Divide el proyecto en módulos o fases de manera independiente y luego integrarlas. Es recomendable para proyectos cuyos requisitos evolucionan a lo largo del tiempo. Esto quiere decir proyectos largos y complejos. Además permite flexibilidad y adaptación alta ante cambios, esto lo hace aplicable a desarrollos ágiles y de validación continua.
3. Reutilización - Como en la incremental, pero con un enfoque de componentes, esto lo vuelve ideal para sistemas grandes y modulares con requisitos de mantenimiento a largo plazo y con un enfoque en la calidad.
4. Prototipado - Sirve para buscar soluciones específicas de software y apoyar el diseño de interfaces de usuario. Ayuda a que se reduzcan el número de propuestas de cambios de requerimientos posterior a la fecha de entrega. Su aplicación ideal surge ante proyectos que requieren validación temprana de ideas haciendo énfasis en la parte UX / UI, o proyectos que dependen de una alta interacción con los usuarios finales o stakeholders.
5. Espiral - También conocido como modelo basado en el riesgo, se divide en ciclos con 4 fases: Planificación, determinación de objetivos, análisis de riesgo y desarrollo y pruebas. El caso de uso ideal es un proyecto en el que se priorice el análisis y la gestión de riesgo con requisitos cambiantes, necesidad de retroalimentación constante y sobre todo recursos limitados o sistemas críticos.
6. Híbrido - El híbrido es como si recogieras todas las ideas de las otras metodologías, combina las ágiles con waterfall. Su caso de uso ideal es para proyectos que requieren combinar la estructura y planificación detallada de waterfall con la flexibilidad y adaptación ágil.

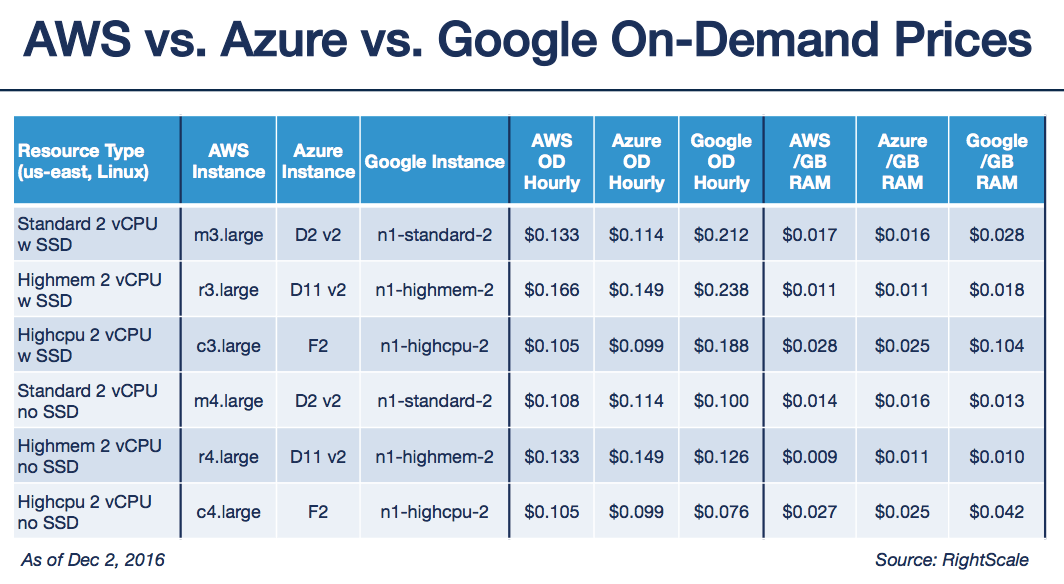
Tras analizar las diversas Metodologías de Trabajo disponibles para llevar a cabo el proyecto y conversar, por unanimidad se decidió que usaríamos un modelo ágil de desarrollo dado a que la estructura de waterfall es muy incómoda de trabajar y no permite realizar cambios fácilmente durante el desarrollo. La metodología híbrida tampoco fue una opción porque no requerimos de la planificación de waterfall. Posteriormente debatimos sobre cuál sería el enfoque ágil a usar. Y, en este punto, volvimos a descartar metodologías; descartamos espiral ya que no tenemos riesgos significativos o sistemas críticos y nuestros recursos son suficientes; descartamos prototipado no tenemos ni usuarios finales ni stakeholders con los que interactuar constantemente y aún así el UX / UI no es el foco del proyecto; finalmente descartamos el modelo incremental porque el incremental se enfoca en un proyecto a largo plazo, mientras que el de reutilización realmente permite trabajar con un proyecto de alta calidad, escalable y que facilita el mantenimiento durante el tiempo. Pero no solo eso, sino que el modelo de reutilización tiene la ventaja de que ayuda con la estandarización y la consistencia a través de componentes siendo una buena oportunidad de aprender otra metodología de trabajo y comprometernos no sólo con las tecnologías sino también con el marco de trabajo.

* Selección de las herramientas hardware y software, fundamentado.
* Puede presentarse en un cuadro o con viñetas.
* **Indicar las limitaciones y ventajas de utilizar dichas herramientas.**
* Hardware:
  + Servidor Web
    - CPU - Capaz de manejar solicitudes web.
    - Conexión a internet - De alta velocidad para garantizar una conexión estable.
    - Memoria RAM - Gestionar tráfico y procesos concurrentes.
    - Almacenamiento - Disco Duro o SSD para almacenar el contenido.
  + Router o Módem - Proporcionar la conexión a Internet desde el ISP.
  + Cliente - Computadoras, laptops, celulares y/o tablets para acceder a la web.
* Software:
  + Diseño y Prototipo
    - Figma - Diseño y Prototipado de Interfaz de Usuario de manera simple.
  + Frontend
    - React - Librería JS con Interacción fácil y simple altamente componetizable.
    - TypeScript - Tipado al archivo lógico haciéndolo fácil de mantener y reduciendo posibles errores futuros.
    - CSS - Estilizar de manera personalizada.
    - Tailwind - Estilos genéricos altamente reutilizables.
  + Backend
    - Spring MVC(Java) - Crear la API Restful de forma simple.
  + Base de Datos
    - SQL Server (SQL) - Sistema de Gestión de Base de Datos Relacionales, facilita los procesos ETL para almacenamiento de datos.
  + Control de Versiones
    - Git - Controlar los cambios que se realizan a través del tiempo por los distintos colaboradores del proyecto.
    - Github - Almacenar, compartir y trabajar junto con otros usuarios mediante repositorios.
  + Otros
    - Docker - Facilita la creación y la ejecución de arquitecturas, estandarizado y altamente escalable.
    - Postman - Herramienta para probar las APIs y también sirve como una forma de documentarlas mediante una interfaz sencilla y fácil de usar.
    - Github Actions - permite ejecutar flujos de trabajo cuando otros eventos suceden en tu repositorio.
    - Azure

Github Actions tiene un límite de uso gratuito basado en:



Azure for Students tiene un límite de uso gratuito basado en: un crédito de 100 USD para usarlo en 12 meses. Mientras tengas el crédito, obtendrás también servicios gratuitos.



**CAPITULO IV: GENERACIÓN DE SOLUCIONES**

**4.1.** n una tabla de doble entrada las restricciones vinculadas a las variables asociadas al proyecto.

| **Aspectos** | **Restricción realista** | **Argumentación de cómo el proyecto incluye la restricción** |
| --- | --- | --- |
| Salud pública y seguridad | Cumplimiento de aforo y distanciamiento en el restaurante | | El sistema debe incluir la gestión del aforo según las normativas | | --- |  |  | | --- | |
| Tecnológica | | Disponibilidad de infraestructura para hosting | | --- |  |  | | --- | | | La infraestructura del servidor debe soportar el tráfico previsto | | --- |  |  | | --- | |
| Ética | | Protección de datos personales de los clientes | | --- |  |  | | --- | | | El sistema debe cumplir con leyes de privacidad (GDPR, etc.) | | --- |  |  | | --- | |
| Legal | | Cumplimiento de normativas de comercio electrónico | | --- |  |  | | --- | | | Cumplimiento de normativas de comercio electrónico | | --- |  |  | | --- | |
| Cultural | | Adaptación a las prácticas locales de gestión de reservas | | --- |  |  | | --- | | | El sistema debe ser flexible para adaptarse a las costumbres y prácticas culturales del mercado local. | | --- |  |  | | --- | |
| Ambiental | | Optimización en la gestión de recursos del restaurante | | --- |  |  | | --- | | | Reducir desperdicios mediante una mejor planificación de mesas y recursos. | | --- |  |  | | --- | |
| Social | | Inclusión de accesibilidad para personas con discapacidad | | --- |  |  | | --- | | | El sistema debe ser accesible y cumplir con estándares de accesibilidad para personas con discapacidad. | | --- |  |  | | --- | |
| Económica | | Presupuesto limitado para el desarrollo y mantenimiento | | --- |  |  | | --- | | | El proyecto debe ajustarse al presupuesto asignado sin comprometer la calidad. | | --- |  |  | | --- | |
| Otras (estético, comercial, etc) | | Diseño atractivo y funcional | | --- |  |  | | --- | | La interfaz debe ser atractiva y fácil de usar para maximizar y la experiencia del usuario |

**C8: Valida la solución propuesta utilizando métricas de la metodología de desarrollo de software.**

**4.5. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA**

La solución propuesta será validada mediante las siguientes métricas:

* **Métricas de Usabilidad**: Pruebas con usuarios reales para evaluar la facilidad de uso de la aplicación.
* **Métricas de Productividad**: Tiempos de carga y procesamiento de reservas.
* **Métricas de Seguridad**: Cumplimiento con estándares de protección de datos personales.
* **Métricas de Rendimiento**: Número de reservas gestionadas por minuto sin errores.

**4.6. CRONOGRAMA DEL PROYECTO Y EDT (ESTRUCTURA DE DESGLOSE DE TRABAJO)**

* El cronograma del proyecto debe mostrarse con el diagrama de gantt.
* Utilice herramientas de gestión de proyectos para ir mostrando el avance de su solución (bitrix24, Excel, MS Project, gantt Project, Trello, etc)

El proyecto será gestionado utilizando un diagrama de Gantt que incluye las siguientes fases:

* **Fase 1**: Recolección de Requisitos (2 semanas)
* **Fase 2**: Diseño del Sistema (3 semanas)
* **Fase 3**: Desarrollo de la Aplicación (6 semanas)
* **Fase 4**: Pruebas y Validación (2 semanas)
* **Fase 5**: Despliegue y Entrega (1 semana)

La EDT incluirá tareas específicas como el desarrollo del frontend, backend, pruebas unitarias y de integración.

**C9: Evalúa los esquemas de solución y los coteja con el objetivo del proyecto realizado.**

**ANÁLISIS DE RESTRICCIONES REALISTAS VS ALTERNATIVAS DE SOLUCIÓN**

Analice en una tabla de doble entrada cada una las alternativas en base a las restricciones realistas (económicas, tecnológicas, de seguridad, ético, político, entre otras)

| Restricción | | **Solución 1 (Software Genérico)** | | --- |  |  | | --- | | | **Solución 2 (Desarrollo Interno)** | | --- |  |  | | --- | | | **Solución 3 (SaaS)** | | --- |  |  | | --- | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Salud pública | | --- |  |  | | --- | | | Baja adaptabilidad | | --- |  |  | | --- | | | Alta adaptabilidad | | --- |  |  | | --- | | | Media adaptabilidad | | --- |  |  | | --- | |
| | Tecnológica | | --- |  |  | | --- | | | Baja escalabilidad | | --- |  |  | | --- | | | Alta escalabilidad | | --- |  |  | | --- | | | Media escalabilidad | | --- |  |  | | --- | |
| | Ética | | --- |  |  | | --- | | | Seguridad limitada | | --- |  |  | | --- | | | Alta seguridad | | --- |  |  | | --- | | | Moderada seguridad | | --- |  |  | | --- | |
| Legal | | Cumplimiento básico | | --- |  |  | | --- | | | Cumplimiento total | | --- |  |  | | --- | | | Cumplimiento parcial | | --- |  |  | | --- | |
| | Cultural | | --- |  |  | | --- | | | No personalizable | | --- |  |  | | --- | | | Altamente personalizable | | --- |  |  | | --- | | | Moderadamente flexible | | --- |  |  | | --- | |
| | Ambiental | | --- |  |  | | --- | | | Baja optimización | | --- |  |  | | --- | | | Alta optimización | | --- |  |  | | --- | | | Media optimización | | --- |  |  | | --- | |
| Social | | Poca accesibilidad | | --- |  |  | | --- | | | Alta accesibilidad | | --- |  |  | | --- | | | Media accesibilidad | | --- |  |  | | --- | |
| | Económica | | --- |  |  | | --- | | | Bajo costo inicial | | --- |  |  | | --- | | | Alto costo de desarrollo | | --- |  |  | | --- | | | Costo moderado | | --- |  |  | | --- | |
| | Estético/Comercial | | --- |  |  | | --- | | | Diseño limitado | | --- |  |  | | --- | | | Diseño completamente personalizable | | --- |  |  | | --- | | Diseño predeterminado |

**4.4. ANÁLISIS DE ARQUITECTURAS DE SOFTWARE**

Se analizaron varias arquitecturas para el desarrollo de la aplicación, incluyendo:

* **Monolítica**: Sencilla de implementar, pero difícil de escalar.
* **Microservicios**: Altamente escalable y flexible, pero requiere mayor esfuerzo en la gestión.
* **Arquitectura en capas**: Organizada y fácil de mantener, facilita la separación de responsabilidades.

**Conclusión**: Se seleccionó una **arquitectura en capas**, ya que permite una organización clara del sistema, con una buena separación entre la interfaz, la lógica de negocio y el acceso a los datos, garantizando escalabilidad y facilidad de mantenimiento.

**4.5. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA**

La solución propuesta será validada mediante las siguientes métricas:

* **Métricas de Usabilidad**: Pruebas con usuarios reales para evaluar la facilidad de uso de la aplicación.
* **Métricas de Productividad**: Tiempos de carga y procesamiento de reservas.
* **Métricas de Seguridad**: Cumplimiento con estándares de protección de datos personales.
* **Métricas de Rendimiento**: Número de reservas gestionadas por minuto sin errores.

**4.6. CRONOGRAMA DEL PROYECTO Y EDT (ESTRUCTURA DE DESGLOSE DE TRABAJO)**

* El cronograma del proyecto debe mostrarse con el diagrama de gantt.
* Utilice herramientas de gestión de proyectos para ir mostrando el avance de su solución (bitrix24, Excel, MS Project, gantt Project, Trello, etc)

El proyecto será gestionado utilizando un diagrama de Gantt que incluye las siguientes fases:

* **Fase 1**: Recolección de Requisitos (2 semanas)
* **Fase 2**: Diseño del Sistema (3 semanas)
* **Fase 3**: Desarrollo de la Aplicación (6 semanas)
* **Fase 4**: Pruebas y Validación (2 semanas)
* **Fase 5**: Despliegue y Entrega (1 semana)

La EDT incluirá tareas específicas como el desarrollo del frontend, backend, pruebas unitarias y de integración.

**C10: Utiliza software, herramientas de gestión de proyectos, Excel u otra herramienta equivalente para poder obtener visualmente los resultados del plan de solución.**

**4.6. CRONOGRAMA DEL PROYECTO Y EDT (ESTRUCTURA DE DESGLOSE DE TRABAJO)**

* El cronograma del proyecto debe mostrarse con el diagrama de gantt.
* Utilice herramientas de gestión de proyectos para ir mostrando el avance de su solución (bitrix24, Excel, MS Project, gantt Project, Trello, etc)

El proyecto será gestionado utilizando un diagrama de Gantt que incluye las siguientes fases:

* **Fase 1**: Recolección de Requisitos (2 semanas)
* **Fase 2**: Diseño del Sistema (3 semanas)
* **Fase 3**: Desarrollo de la Aplicación (6 semanas)
* **Fase 4**: Pruebas y Validación (2 semanas)
* **Fase 5**: Despliegue y Entrega (1 semana)

La EDT incluirá tareas específicas como el desarrollo del frontend, backend, pruebas unitarias y de integración.

**Atributo del graduado: Diseño y Desarrollo de Soluciones**

Diseña soluciones creativas para problemas complejos de ingeniería y diseña sistemas, componentes o procesos para satisfacer necesidades identificadas dentro de restricciones realistas, según se requiera, de salud y seguridad pública, el costo del ciclo de vida, el cero carbono neto, de recursos, culturales, sociales, económicos y ambientales.

**C1: Documenta de manera justificada en una tabla de doble entrada las restricciones vinculadas a las variables asociadas al proyecto.**

| **Aspectos** | **Restricción realista** | **Argumentación de cómo el proyecto incluye la restricción** |
| --- | --- | --- |
| Salud pública y seguridad | Cumplimiento de aforo y distanciamiento en el restaurante | | El sistema debe incluir la gestión del aforo según las normativas | | --- |  |  | | --- | |
| Tecnológica | | Disponibilidad de infraestructura para hosting | | --- |  |  | | --- | | | La infraestructura del servidor debe soportar el tráfico previsto | | --- |  |  | | --- | |
| Ética | | Protección de datos personales de los clientes | | --- |  |  | | --- | | | El sistema debe cumplir con leyes de privacidad (GDPR, etc.) | | --- |  |  | | --- | |
| Legal | | Cumplimiento de normativas de comercio electrónico | | --- |  |  | | --- | | | Cumplimiento de normativas de comercio electrónico | | --- |  |  | | --- | |
| Cultural | | Adaptación a las prácticas locales de gestión de reservas | | --- |  |  | | --- | | | El sistema debe ser flexible para adaptarse a las costumbres y prácticas culturales del mercado local. | | --- |  |  | | --- | |
| Ambiental | | Optimización en la gestión de recursos del restaurante | | --- |  |  | | --- | | | Reducir desperdicios mediante una mejor planificación de mesas y recursos. | | --- |  |  | | --- | |
| Social | | Inclusión de accesibilidad para personas con discapacidad | | --- |  |  | | --- | | | El sistema debe ser accesible y cumplir con estándares de accesibilidad para personas con discapacidad. | | --- |  |  | | --- | |
| Económica | | Presupuesto limitado para el desarrollo y mantenimiento | | --- |  |  | | --- | | | El proyecto debe ajustarse al presupuesto asignado sin comprometer la calidad. | | --- |  |  | | --- | |
| Otras (estético, comercial, etc) | | Diseño atractivo y funcional | | --- |  |  | | --- | | La interfaz debe ser atractiva y fácil de usar para maximizar y la experiencia del usuario |

**C2: En una tabla de doble entrada plantea alternativas de diseño a problemas complejos de ingeniería teniendo en cuenta restricciones realistas realistas, según se requiera, de salud y seguridad pública, el costo del ciclo de vida, el cero carbono neto, de recursos, culturales, sociales, económicos y ambientales.**

**4.3. ANÁLISIS DE RESTRICCIONES REALISTAS VS ALTERNATIVAS DE SOLUCIÓN**

Analice en una tabla de doble entrada cada una las alternativas en base a las restricciones realistas (económicas, tecnológicas, de seguridad, ético, político, entre otras)

| Restricción | | **Solución 1 (Software Genérico)** | | --- |  |  | | --- | | | **Solución 2 (Desarrollo Interno)** | | --- |  |  | | --- | | | **Solución 3 (SaaS)** | | --- |  |  | | --- | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Salud pública | | --- |  |  | | --- | | | Baja adaptabilidad | | --- |  |  | | --- | | | Alta adaptabilidad | | --- |  |  | | --- | | | Media adaptabilidad | | --- |  |  | | --- | |
| | Tecnológica | | --- |  |  | | --- | | | Baja escalabilidad | | --- |  |  | | --- | | | Alta escalabilidad | | --- |  |  | | --- | | | Media escalabilidad | | --- |  |  | | --- | |
| | Ética | | --- |  |  | | --- | | | Seguridad limitada | | --- |  |  | | --- | | | Alta seguridad | | --- |  |  | | --- | | | Moderada seguridad | | --- |  |  | | --- | |
| Legal | | Cumplimiento básico | | --- |  |  | | --- | | | Cumplimiento total | | --- |  |  | | --- | | | Cumplimiento parcial | | --- |  |  | | --- | |
| | Cultural | | --- |  |  | | --- | | | No personalizable | | --- |  |  | | --- | | | Altamente personalizable | | --- |  |  | | --- | | | Moderadamente flexible | | --- |  |  | | --- | |
| | Ambiental | | --- |  |  | | --- | | | Baja optimización | | --- |  |  | | --- | | | Alta optimización | | --- |  |  | | --- | | | Media optimización | | --- |  |  | | --- | |
| Social | | Poca accesibilidad | | --- |  |  | | --- | | | Alta accesibilidad | | --- |  |  | | --- | | | Media accesibilidad | | --- |  |  | | --- | |
| | Económica | | --- |  |  | | --- | | | Bajo costo inicial | | --- |  |  | | --- | | | Alto costo de desarrollo | | --- |  |  | | --- | | | Costo moderado | | --- |  |  | | --- | |
| | Estético/Comercial | | --- |  |  | | --- | | | Diseño limitado | | --- |  |  | | --- | | | Diseño completamente personalizable | | --- |  |  | | --- | | Diseño predeterminado |

**C3: Documenta el diseño y desarrollo de soluciones a problemas complejos de ingeniería haciendo uso correcto de herramientas y componentes, respondiendo a restricciones realistas vinculantes.**

**VISIÓN DE LA SOLUCIÓN**

El sistema será una aplicación web moderna que facilite la gestión de reservas en restaurantes. Permitirá a los usuarios reservar mesas, consultar menús y pagar en línea de forma segura. Asimismo, proporcionará a los administradores herramientas para gestionar reservas y menús, mejorando la eficiencia operativa de los restaurantes. La solución será fácil de usar, accesible y escalable, adaptándose a las necesidades del mercado local.

**5.3.1. Características del sistema**

| **ID** | **Descripción** | **Prioridad** | **Objetivo de negocio asociado** |
| --- | --- | --- | --- |
| CAR-01 | El sistema debe permitir la reserva de mesas a través de un navegador web. | Alta | ON-1 |
| CAR-02 | El sistema debe permitir la visualización de disponibilidad de mesas en tiempo real. | Alta | ON-2 |
| CAR-03 | El sistema debe permitir la cancelación o modificación de reservas. | Media | ON-3 |
| CAR-04 | El sistema debe permitir la cancelación o modificación de reservas. | Media | ON-4 |

**5.4. CONTEXTO DEL SISTEMA**

**5.4.1 Interesados**

| **Nombre** | **Descripción** | **Responsabilidades** |
| --- | --- | --- |
| Jeremy | Scrum Master | - Facilitar las reuniones del equipo.  - Eliminar impedimentos.  - Asegurar la comunicación efectiva entre el equipo y los interesados. |
| Albert | Desarrollador Frontend | - Desarrollo y diseño de la interfaz de usuario.  - Coordinación de pruebas de usabilidad. |
| Franco | Desarrollador Backend | - Implementación de la lógica del servidor.  - Integración con la base de datos. |
| David | tester | - Realización de pruebas y validación del sistema.  - Documentación de resultados de pruebas. |
| Juanjo | Product Owner | - Aprobar la visión y el alcance del sistema.  - Proveer requisitos del negocio. |

**5.4.2. Entorno de operación**

El sistema será usado desde navegadores web y deberá soportar los siguientes:

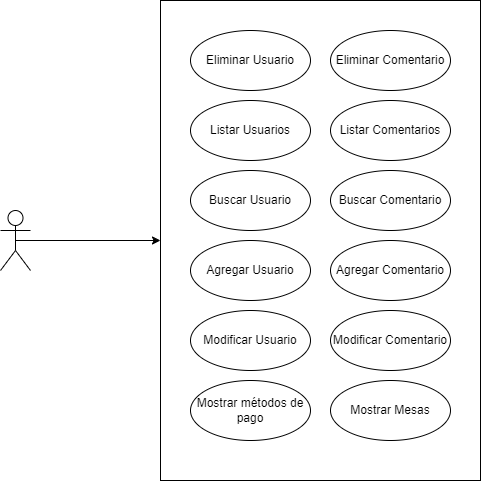
* Internet Explorer 10+
* Firefox 10+
* Google Chrome 17+

Se debe usar el servidor de base de datos legado con Oracle 21c.  
La integración con sistemas externos, incluyendo el sistema de autorización de pagos, se lleva a cabo mediante servicios web.

**5.5. REQUERIMIENTOS DE ARQUITECTURA**

**5.5.1. Modelo de Casos de uso del sistema**

* Diagrama general de casos de uso del sistema



* Especificación de casos de Uso del sistema

#### **1. Caso de Uso: Eliminar Usuario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador eliminar un usuario del sistema.
* **Precondiciones**: El usuario debe estar registrado.
* **Flujo Principal**:
  1. El administrador selecciona un usuario de la lista.
  2. El sistema solicita confirmación.
  3. El administrador confirma la eliminación.
  4. El sistema elimina al usuario.

#### **2. Caso de Uso: Listar Usuarios**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador ver todos los usuarios registrados.
* **Precondiciones**: El administrador debe estar autenticado.
* **Flujo Principal**:
  1. El administrador solicita la lista de usuarios.
  2. El sistema muestra la lista.

#### **3. Caso de Uso: Buscar Usuario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador buscar un usuario específico en el sistema.
* **Precondiciones**: El administrador debe estar autenticado.
* **Flujo Principal**:
  1. El administrador ingresa el criterio de búsqueda.
  2. El sistema muestra los resultados de la búsqueda.

#### **4. Caso de Uso: Agregar Usuario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador agregar un nuevo usuario al sistema.
* **Precondiciones**: El administrador debe estar autenticado.
* **Flujo Principal**:
  1. El administrador ingresa la información del nuevo usuario.
  2. El sistema valida los datos.
  3. El sistema agrega el nuevo usuario.

#### **5. Caso de Uso: Modificar Usuario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador actualizar la información de un usuario existente.
* **Precondiciones**: El administrador debe estar autenticado y el usuario debe existir.
* **Flujo Principal**:
  1. El administrador selecciona un usuario.
  2. El administrador modifica la información.
  3. El sistema valida los datos.
  4. El sistema actualiza la información del usuario.

#### **6. Caso de Uso: Eliminar Comentario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador eliminar un comentario del sistema.
* **Precondiciones**: El comentario debe existir.
* **Flujo Principal**:
  1. El administrador selecciona un comentario.
  2. El sistema solicita confirmación.
  3. El administrador confirma la eliminación.
  4. El sistema elimina el comentario.

#### **7. Caso de Uso: Listar Comentarios**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador ver todos los comentarios.
* **Precondiciones**: El administrador debe estar autenticado.
* **Flujo Principal**:
  1. El administrador solicita la lista de comentarios.
  2. El sistema muestra la lista.

#### **8. Caso de Uso: Buscar Comentario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador buscar un comentario específico.
* **Precondiciones**: El administrador debe estar autenticado.
* **Flujo Principal**:
  1. El administrador ingresa el criterio de búsqueda.
  2. El sistema muestra los resultados.

#### **9. Caso de Uso: Agregar Comentario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador agregar un nuevo comentario.
* **Precondiciones**: El administrador debe estar autenticado.
* **Flujo Principal**:
  1. El administrador ingresa el nuevo comentario.
  2. El sistema valida el comentario.
  3. El sistema agrega el nuevo comentario.

#### **10. Caso de Uso: Modificar Comentario**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador actualizar un comentario existente.
* **Precondiciones**: El comentario debe existir.
* **Flujo Principal**:
  1. El administrador selecciona un comentario.
  2. El administrador modifica el comentario.
  3. El sistema valida el comentario.
  4. El sistema actualiza el comentario.

#### **11. Caso de Uso: Mostrar Métodos de Pago**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al administrador ver los métodos de pago disponibles.
* **Precondiciones**: No hay, se puede consultar libremente.
* **Flujo Principal**:
  1. El usuario solicita ver los métodos de pago.
  2. El sistema muestra los métodos de pago disponibles.

#### **12. Caso de Uso: Mostrar Mesas**

* **Actor**: Administrador de sistema
* **Descripción**: Permite al usuario ver la disponibilidad de mesas.
* **Precondiciones**: Ninguna.
* **Flujo Principal**:

1. El usuario solicita ver la disponibilidad de mesas.
2. El sistema muestra las mesas disponibles.

* Priorización de casos de uso

**Alta Prioridad**:

* Eliminar Usuario
* Agregar Usuario
* Listar Usuarios

**Media Prioridad**:

* Modificar Usuario
* Buscar Usuario
* Eliminar Comentario
* Agregar Comentario

**Baja Prioridad**:

* Listar Comentarios
* Mostrar Métodos de Pago
* Mostrar Mesas

**5.5.2. Atributos de calidad**

**Usabilidad:** Interfaz intuitiva y fácil de usar para los administradores y usuarios.

**Rendimiento:** Respuestas del sistema en menos de 2 segundos.

**Seguridad:** Protección de datos y acceso restringido sólo a administradores autorizados.

**Mantenibilidad:** Código estructurado y documentado para facilitar futuras modificaciones.

**5.5.3. Restricciones asociadas al sistema**

#### **1 Restricciones del Cliente**

* El sistema debe ser accesible desde dispositivos móviles y de escritorio.
* Debe cumplir con regulaciones de protección de datos.

#### **2 Restricciones Organizacionales**

* El sistema debe integrarse con el sistema de gestión de pagos existente.
* La implementación debe completarse en un plazo de 10 meses.
* Debe desarrollarse utilizando tecnologías aprobadas por la organización.

**Atributo del graduado: Uso de Herramientas**

Crea, selecciona, aplica, y reconoce las limitaciones de las técnicas, recursos y herramientas modernas apropiadas de ingeniería y tecnologías de la información, incluyendo la predicción y el modelado, en problemas complejos de ingeniería

**C1: Elabora un informe donde selecciona herramientas y/o técnicas apropiadas para el desarrollo de la solución a un problema complejo de ingeniería, reconociendo al limitaciones y ventajas.**

## **Actores**

* Admin
* Usuario
* Cliente
* Sistema

#### **1. Administrar Usuarios**

**Actor(es) Involucrado(s)**: Admin

Descripción: El administrador puede gestionar los usuarios del sistema, incluyendo las operaciones de agregar, modificar, eliminar, buscar y listar usuarios.

Precondiciones: El admin debe estar autenticado.

Postcondiciones: Los usuarios serán gestionados de acuerdo con la operación realizada (crear, modificar, eliminar).

Extensiones: Eliminar, Listar, Buscar, Agregar, Modificar.

#### **2. Administrar Menú**

**Actor(es) Involucrado(s)**: Admin

**Descripción**: El administrador tiene la capacidad de gestionar los ítems del menú del restaurante. Incluye agregar nuevos platillos, modificarlos, eliminarlos, buscar ítems específicos y listarlos.

**Precondiciones**: El admin debe estar autenticado.

**Postcondiciones**: El menú se actualizará con los cambios realizados.

**Extensiones**: Eliminar, Listar, Buscar, Agregar, Modificar.

#### **3. Administrar Comentarios**

**Actor(es) Involucrado(s)**: Admin

**Descripción**: El admin puede gestionar los comentarios de los clientes en la plataforma, pudiendo ver, agregar, modificar o eliminar comentarios.

**Precondiciones**: El admin debe estar autenticado.

**Postcondiciones**: Los comentarios serán gestionados según la operación realizada.

**Extensiones**: Agregar, Eliminar, Modificar, Ver.

#### **4. Consultar Menú**

**Actor(es) Involucrado(s)**: Cliente

**Descripción**: Los clientes pueden consultar el menú disponible en el restaurante.

**Precondiciones**: El cliente debe estar registrado o autenticado.

**Postcondiciones**: El cliente podrá visualizar el menú actualizado.

**Inclusiones**: Mostrar Menú.

#### **5. Comentar**

**Actor(es) Involucrado(s)**: Cliente

**Descripción**: El cliente puede agregar comentarios sobre su experiencia o sobre el menú. También puede modificar o eliminar comentarios previos.

**Precondiciones**: El cliente debe estar autenticado y haber realizado una consulta o reserva previa.

**Postcondiciones**: Los comentarios del cliente serán gestionados en el sistema.

**Extensiones**: Agregar, Eliminar, Modificar, Ver.

#### **6. Administrar Reservas**

**Actor(es) Involucrado(s)**: Cliente

**Descripción**: El cliente puede realizar, modificar o cancelar reservas en el restaurante.

**Precondiciones**: El cliente debe estar autenticado.

**Postcondiciones**: La reserva será actualizada o eliminada según la acción del cliente.

**Extensiones**: Actualizar Reserva, Reservar Mesa.

#### **7. Pagar**

**Actor(es) Involucrado(s)**: Cliente

**Descripción**: El cliente puede realizar pagos a través de diferentes métodos como Yape, Plin o Transferencia.

**Precondiciones**: El cliente debe haber realizado una reserva y seleccionado el método de pago.

**Postcondiciones**: El pago será registrado y la reserva quedará confirmada.

**Inclusiones**: Mostrar Métodos de Pago.

#### **8. Reservar Mesa**

**Actor(es) Involucrado(s)**: Cliente

**Descripción**: El cliente puede seleccionar una mesa disponible y reservarla para una fecha y hora específicas.

**Precondiciones**: El cliente debe estar autenticado y consultar la disponibilidad de mesas.

**Postcondiciones**: La mesa será reservada y el cliente recibirá una confirmación.

**Inclusiones**: Ver Mesas.

#### **9. Ver Mesas**

**Actor(es) Involucrado(s)**: Cliente

**Descripción**: El cliente puede consultar la disponibilidad de mesas antes de realizar una reserva.

**Precondiciones**: El cliente debe estar autenticado.

**Postcondiciones**: La información sobre la disponibilidad de mesas será mostrada.

**Inclusiones**: Mostrar Mesas.

**5.1.3. Objetivos del Negocio**

| **ID** | **Descripción del objetivo de negocio** |
| --- | --- |
| **ON-1** | Incrementar en un 50% las reservas de mesas mediante la implementación de un sistema de gestión de reservas en línea que permita a los clientes reservar, consultar el menú, y realizar pagos desde la web y dispositivos móviles. |
| **ON-2** | Proporcionar herramientas que integren métodos de pago digitales (como Yape, Plin, transferencias) y permitan a los clientes visualizar las mesas disponibles en tiempo real en los próximos 6 meses. |
| **ON-3** | Permitir a los administradores del sistema gestionar de manera eficiente el menú, los comentarios de los clientes y los usuarios registrados, con el objetivo de mejorar continuamente la experiencia del cliente a lo largo de 12 meses. |
| **ON-4** | Lanzar el sistema de reservas, gestión de menú y administración de usuarios antes del 30 de junio de 2025, asegurando que cumpla con todos los requisitos de accesibilidad y usabilidad. |

**5.2. REQUISITOS DEL SISTEMA**

**Requisitos Funcionales**:

El sistema debe permitir a los usuarios realizar reservas en línea para restaurantes.

Debe mostrar horarios y mesas disponibles en tiempo real.

Los administradores podrán gestionar menús, reservas y comentarios de los clientes.

El sistema permitirá a los clientes cancelar o modificar reservas.

**Requisitos No Funcionales**:

El sistema debe ser accesible desde cualquier navegador web (Firefox, Chrome, etc.).

Debe garantizar altos niveles de seguridad, cumpliendo con normativas de protección de datos (GDPR).

La plataforma debe ser escalable para soportar un incremento en el número de usuarios.

Debe ser fácil de usar y accesible, cumpliendo con los estándares de accesibilidad para personas con discapacidad.

**C2: Elabora un informe donde aplica correctamente al menos dos herramientas y/o técnicas modernas apropiadas, sin requerir soporte del docente, que le permite analizar el problema, diseñar solución y crear la solución correcta.**

**2.3. ANÁLISIS DEL PROBLEMA**

**2.3.1. IDENTIFICACIÓN Y FORMULACIÓN DEL PROBLEMA**

El restaurante enfrenta problemas en la gestión de reservas y en la consulta de eventos por parte de los clientes. Actualmente, el proceso de reservas se realiza manualmente o mediante llamadas telefónicas, lo que genera ineficiencias y errores, como la sobreasignación de mesas o la falta de confirmaciones oportunas. Además, la visibilidad de eventos es limitada, dificultando la promoción y la interacción con los clientes. La falta de una plataforma digital afecta tanto la experiencia del usuario como la operatividad del restaurante, lo que ha motivado la necesidad de crear un sistema web que permita gestionar estos procesos de forma automatizada y más eficiente.

**2.3.2. DEFINICIÓN DE OBJETIVOS**

**Objetivo general**

Diseñar un sistema web para el restaurante “Fusión y sazón ” utilizando las directivas establecidas en las mejores prácticas de desarrollo web.

**Objetivos específicos**

* Realizar el modelado del sistema desarrollando el modelo de caso de uso del negocio y el modelo de análisis del negocio.
* Realizar la toma de requerimientos para identificar las necesidades del restaurante.
* Realizar el modelo de casos de uso del sistema y el modelo de análisis del sistema

**2.3.3. ALCANCE DE LA SOLUCIÓN**

La propuesta abarca la creación de un sistema web que permita a los usuarios realizar reservas, previsualizar la disponibilidad de las mesas, consulta de menú y consultar eventos. El sistema contará con un proceso de login tanto para clientes como para administrador, a su vez proporcionará una interfaz amigable para los usuarios.

**C3: Crea soluciones a problemas complejos de ingeniería utilizando herramientas y / o técnicas para el análisis, modelado o diseño, predicción y/o construcción de producto documentando correctamente las evidencias.**

**2.1.1 Aplicación de sistema web para la gestión de pedidos de restaurantes**

Según la investigación realizada por Rodríguez. et al (2022), se evaluó la implementación de un sistema web para la gestión de pedidos del Restaurante Náutico ubicado en Trujillo, Perú. Se incluyó la observación directa y la recolección de datos a través de fichas de observación, con una muestra de 30 procesos de gestión de pedidos. Se plantearon varias hipótesis que indican que el uso del sistema web reduciría el tiempo de registro, búsqueda de información y emisión de reportes de pedidos, además de aumentar la satisfacción del cliente. Los resultados mostraron una mejora significativa en estos indicadores, evidenciando que la implementación del sistema web no solo optimizó los tiempos de gestión, sino que también elevó el nivel de satisfacción del cliente en comparación con el sistema anterior. Este estudio se enmarca dentro de investigaciones previas que han explorado el uso de tecnologías en la industria gastronómica, destacando la importancia de la digitalización en la mejora de procesos operativos y la experiencia del cliente en restaurantes.

**2.1.2 Sistema informático web para automatizar el control de ventas del Restaurante Espuma del Mar**

En la tesis realizada por Vera Chiquito, G. V. (2023), el proyecto estuvo centrado en la implementación de un sistema web para automatizar el control de ventas en el restaurante Espuma del Mar, ubicado en Puerto López, Manabí, Ecuador. Se utilizó un enfoque cualitativo de tipo descriptivo, con métodos como análisis documental, observación directa y entrevistas para identificar problemas en el control de ventas. Los resultados mostraron que el 100% de los clientes aceptó la importancia del sistema, mejorando el registro de ventas y reduciendo tiempos.

**2.1.3 Desarrollo de un sistema web y aplicación móvil para la gestión de reservas y control de hospedaje**

En el trabajo de investigación realizado por Freire, C & Naveda, J (2019),tuvo como objetivo el desarrollo de una aplicación web y móvil para el hotel Al Safi El Paraíso, ubicado en la ciudad de Ambato, Ecuador, con la finalidad de mejorara la gestión de procesos de reservas, control de hospedaje, entre otras facilidades administrativas. Se implementó un enfoque de investigación principalmente cualitativo. Al finalizar con el proyecto concluye que se cumplió exitosamente con los objetivos, contando con un sistema funcional y capaz de agilizar los procesos del hotel.

**2.1.4 Plataforma web para la gestión de reservas en restaurantes**

En el trabajo de investigación llevado a cabo por Manazo Jimenez, R(2022), se realizó un una plataforma web para la gestión de reservas en restaurantes.El enfoque del proyecto fue práctico y orientado al desarrollo, centrado en la creación de una plataforma funcional. La información se evaluó y recolectó a través de investigación de sistemas similares, análisis de requisitos, y pruebas de funcionalidad. El proyecto obtuvo como resultados el desarrollo de la plataforma, cumpliendo con los objetivos establecidos, permitiendo así mismo mejorar la experiencia de los consumidores.

**2.1.5 Desarrollo e implementación de una aplicación Web para la Gestión de Boletería**

En el estudio realizado por Falconi Tamayo (2020), se desarrolló una aplicación web para optimizar la gestión del área de boletería y generación de reportes del Teatro-Restaurant de la empresa Vilaró, ubicada en Guayaquil, Colombia. Para la realización del proyecto de desarrollo se utilizó la metodología en cascada. El proyecto se fundamentó en un marco teórico y tecnológico, y se definieron requerimientos específicos, junto con casos de uso y modelado de datos para guía. Como resultado se logró mejorar la eficiencia en la gestión de boletería, eliminando errores en ventas y descuadres de caja.

**2.2. BASES TEÓRICAS**

Para la planificación del proyecto, se han aplicado diversas herramientas y conceptos clave para asegurar el correcto desarrollo del software; entre estos se incluyen la evaluación de requerimientos de negocio, el análisis de requisitos funcionales y no funcionales, así como la elaboración de diagramas de casos de uso y diagramas de clases, que ayudan a modelar el sistema de manera eficiente. Además, se utiliza SQL como gestor de base de datos, Visual Studio Code como entorno de desarrollo integrado (IDE) para la implementación en HTML y la integración de APIs diversas para ampliar las funcionalidades del sistema.

**2.3. ANÁLISIS DEL PROBLEMA**

**2.3.1. IDENTIFICACIÓN Y FORMULACIÓN DEL PROBLEMA**

El restaurante enfrenta problemas en la gestión de reservas y en la consulta de eventos por parte de los clientes. Actualmente, el proceso de reservas se realiza manualmente o mediante llamadas telefónicas, lo que genera ineficiencias y errores, como la sobreasignación de mesas o la falta de confirmaciones oportunas. Además, la visibilidad de eventos es limitada, dificultando la promoción y la interacción con los clientes. La falta de una plataforma digital afecta tanto la experiencia del usuario como la operatividad del restaurante, lo que ha motivado la necesidad de crear un sistema web que permita gestionar estos procesos de forma automatizada y más eficiente.

**2.3.2. DEFINICIÓN DE OBJETIVOS**

**Objetivo general**

Diseñar un sistema web para el restaurante “Fusión y sazón ” utilizando las directivas establecidas en las mejores prácticas de desarrollo web.

**Objetivos específicos**

* Realizar el modelado del sistema desarrollando el modelo de caso de uso del negocio y el modelo de análisis del negocio.
* Realizar la toma de requerimientos para identificar las necesidades del restaurante.
* Realizar el modelo de casos de uso del sistema y el modelo de análisis del sistema

**2.3.3. ALCANCE DE LA SOLUCIÓN**

La propuesta abarca la creación de un sistema web que permita a los usuarios realizar reservas, previsualizar la disponibilidad de las mesas, consulta de menú y consultar eventos. El sistema contará con un proceso de login tanto para clientes como para administrador, a su vez proporcionará una interfaz amigable para los usuarios.

**C4: Informe con los resultados obtenidos al hacer uso herramientas y/o técnicas seleccionadas, justificando la elección en base a los antecedentes seleccionados y recomendaciones brindadas.**

**2.1.1 Aplicación de sistema web para la gestión de pedidos de restaurantes**

Según la investigación realizada por Rodríguez. et al (2022), se evaluó la implementación de un sistema web para la gestión de pedidos del Restaurante Náutico ubicado en Trujillo, Perú. Se incluyó la observación directa y la recolección de datos a través de fichas de observación, con una muestra de 30 procesos de gestión de pedidos. Se plantearon varias hipótesis que indican que el uso del sistema web reduciría el tiempo de registro, búsqueda de información y emisión de reportes de pedidos, además de aumentar la satisfacción del cliente. Los resultados mostraron una mejora significativa en estos indicadores, evidenciando que la implementación del sistema web no solo optimizó los tiempos de gestión, sino que también elevó el nivel de satisfacción del cliente en comparación con el sistema anterior. Este estudio se enmarca dentro de investigaciones previas que han explorado el uso de tecnologías en la industria gastronómica, destacando la importancia de la digitalización en la mejora de procesos operativos y la experiencia del cliente en restaurantes.

**2.1.2 Sistema informático web para automatizar el control de ventas del Restaurante Espuma del Mar**

En la tesis realizada por Vera Chiquito, G. V. (2023), el proyecto estuvo centrado en la implementación de un sistema web para automatizar el control de ventas en el restaurante Espuma del Mar, ubicado en Puerto López, Manabí, Ecuador. Se utilizó un enfoque cualitativo de tipo descriptivo, con métodos como análisis documental, observación directa y entrevistas para identificar problemas en el control de ventas. Los resultados mostraron que el 100% de los clientes aceptó la importancia del sistema, mejorando el registro de ventas y reduciendo tiempos.

**2.1.3 Desarrollo de un sistema web y aplicación móvil para la gestión de reservas y control de hospedaje**

En el trabajo de investigación realizado por Freire, C & Naveda, J (2019),tuvo como objetivo el desarrollo de una aplicación web y móvil para el hotel Al Safi El Paraíso, ubicado en la ciudad de Ambato, Ecuador, con la finalidad de mejorara la gestión de procesos de reservas, control de hospedaje, entre otras facilidades administrativas. Se implementó un enfoque de investigación principalmente cualitativo. Al finalizar con el proyecto concluye que se cumplió exitosamente con los objetivos, contando con un sistema funcional y capaz de agilizar los procesos del hotel.

**2.1.4 Plataforma web para la gestión de reservas en restaurantes**

En el trabajo de investigación llevado a cabo por Manazo Jimenez, R(2022), se realizó un una plataforma web para la gestión de reservas en restaurantes.El enfoque del proyecto fue práctico y orientado al desarrollo, centrado en la creación de una plataforma funcional. La información se evaluó y recolectó a través de investigación de sistemas similares, análisis de requisitos, y pruebas de funcionalidad. El proyecto obtuvo como resultados el desarrollo de la plataforma, cumpliendo con los objetivos establecidos, permitiendo así mismo mejorar la experiencia de los consumidores.

**2.1.5 Desarrollo e implementación de una aplicación Web para la Gestión de Boletería**

En el estudio realizado por Falconi Tamayo (2020), se desarrolló una aplicación web para optimizar la gestión del área de boletería y generación de reportes del Teatro-Restaurant de la empresa Vilaró, ubicada en Guayaquil, Colombia. Para la realización del proyecto de desarrollo se utilizó la metodología en cascada. El proyecto se fundamentó en un marco teórico y tecnológico, y se definieron requerimientos específicos, junto con casos de uso y modelado de datos para guía. Como resultado se logró mejorar la eficiencia en la gestión de boletería, eliminando errores en ventas y descuadres de caja.

**2.2. BASES TEÓRICAS**

Para la planificación del proyecto, se han aplicado diversas herramientas y conceptos clave para asegurar el correcto desarrollo del software; entre estos se incluyen la evaluación de requerimientos de negocio, el análisis de requisitos funcionales y no funcionales, así como la elaboración de diagramas de casos de uso y diagramas de clases, que ayudan a modelar el sistema de manera eficiente. Además, se utiliza SQL como gestor de base de datos, Visual Studio Code como entorno de desarrollo integrado (IDE) para la implementación en HTML y la integración de APIs diversas para ampliar las funcionalidades del sistema.

**2.3. ANÁLISIS DEL PROBLEMA**

**2.3.1. IDENTIFICACIÓN Y FORMULACIÓN DEL PROBLEMA**

El restaurante enfrenta problemas en la gestión de reservas y en la consulta de eventos por parte de los clientes. Actualmente, el proceso de reservas se realiza manualmente o mediante llamadas telefónicas, lo que genera ineficiencias y errores, como la sobreasignación de mesas o la falta de confirmaciones oportunas. Además, la visibilidad de eventos es limitada, dificultando la promoción y la interacción con los clientes. La falta de una plataforma digital afecta tanto la experiencia del usuario como la operatividad del restaurante, lo que ha motivado la necesidad de crear un sistema web que permita gestionar estos procesos de forma automatizada y más eficiente.

**2.3.2. DEFINICIÓN DE OBJETIVOS**

**Objetivo general**

Diseñar un sistema web para el restaurante “Fusión y sazón ” utilizando las directivas establecidas en las mejores prácticas de desarrollo web.

**Objetivos específicos**

* Realizar el modelado del sistema desarrollando el modelo de caso de uso del negocio y el modelo de análisis del negocio.
* Realizar la toma de requerimientos para identificar las necesidades del restaurante.
* Realizar el modelo de casos de uso del sistema y el modelo de análisis del sistema

**2.3.3. ALCANCE DE LA SOLUCIÓN**

La propuesta abarca la creación de un sistema web que permita a los usuarios realizar reservas, previsualizar la disponibilidad de las mesas, consulta de menú y consultar eventos. El sistema contará con un proceso de login tanto para clientes como para administrador, a su vez proporcionará una interfaz amigable para los usuarios.