



**DISEÑO E IMPLEMENTACIÓN DE
UN SISTEMA DE GESTIÓN PARA
EL COMEDOR UNIVERSITARIO
DE LA UNERG**

PROYECTO GRADO II

Isaac Montilla

montilla@gmail.com

informatica

Facultad de Ciencias de la Ingeniería

San Juan de los Morros - diciembre de 2025

ÍNDICE

3. RESUMEN	5
4. Diagnóstico Situacional:	6
4.1 Descripción del contexto de la situación problemática planteada:	6
4.2 Justificación del proyecto:	7
4.3 Objetivos del proyecto:	8
4.4 Procesos que se van a automatizar:	8
5. Determinación, Instalación y Configuración de las Herramientas de Desarrollo:	8
5.1 Plataforma de Desarrollo:	9
5.2 Arquitectura del sistema de información:	9
5.3 Selección del entorno del sistema:	10
5.4 Metodología para el desarrollo:	11
6. Desarrollo del Sistema de Información:	12
6.1.1 Descripción:	12
6.1.2 Requerimientos Funcionales del Proyecto:	13
6.1.3 Requerimientos No Funcionales del Proyecto:	13
6.1.4. Restricciones:	13
6.2 Fase de diseño:	14
6.3 Fase de Codificación:	20
6.3.1 Requerimientos de desarrollo:	20
6.3.2 Desarrollo de los módulos del sistema de información:	20

7. Fase de Pruebas	21
7.1. Elaboración y Ejecución del Plan de Pruebas	21
7.2. Análisis de Resultados:	22
8. Conclusiones:	23
9. Recomendaciones:	23
10. Referencias	24

3. RESUMEN

La presente tesis, "**Diseño e Implementación de un Sistema de Gestión para el Comedor Universitario de la UNERG**", aborda la problemática de la gestión manual y las ineficiencias operativas en el comedor de la UNERG, que resultan en largas colas y control limitado. El **objetivo principal** es diseñar e implementar un sistema que automatice y optimice la administración, desde la planificación de menús hasta el control de acceso. Se utilizará una **metodología ágil** (Scrum) y tecnologías como PHP (Laravel), MySQL y JavaScript (Vue.js) para desarrollar una solución robusta. Se espera mejorar la experiencia del usuario, la eficiencia operativa y el control administrativo, en línea con estudios previos sobre gestión de servicios universitarios (García y Rojas, 2018; Hernández y Soto, 2015).

DESCRIPTORES: Sistema de Gestión, Comedor Universitario, UNERG, Desarrollo de Software.

4. DIAGNÓSTICO SITUACIONAL

4.1 Descripción del contexto de la situación problemática planteada

El comedor universitario de la Universidad Nacional Experimental Rómulo Gallegos (UNERG) representa un pilar fundamental en el bienestar estudiantil, proporcionando un servicio esencial que contribuye directamente a la permanencia y el rendimiento académico de su población. Sin embargo, la gestión actual de este servicio se encuentra arraigada en procesos predominantemente manuales y tradicionales, lo que ha generado una serie de ineficiencias operativas y desafíos significativos que impactan negativamente tanto a los usuarios como al personal administrativo. La ausencia de una infraestructura tecnológica robusta para la administración del comedor ha propiciado un escenario donde la planificación, el control y la ejecución de las actividades diarias se ven obstaculizados por la lentitud, la falta de precisión y la susceptibilidad a errores humanos.

En el día a día, la situación se manifiesta en largas y tediosas colas que los estudiantes deben soportar para acceder al servicio de alimentación. Este fenómeno no solo consume un tiempo valioso que podría ser dedicado a actividades académicas o de esparcimiento, sino que también genera frustración y descontento entre la comunidad estudiantil. La verificación de la identidad de los comensales y el registro de su asistencia se realizan de manera rudimentaria, a menudo mediante listas de papel o la simple inspección visual, lo que ralentiza considerablemente el flujo de entrada y abre la puerta a posibles irregularidades en el control de acceso. Esta falta de un sistema de identificación y registro automatizado impide llevar un control exacto del número de raciones servidas, la frecuencia de uso por estudiante y la optimización de los recursos alimentarios.

La gestión de los menús y la planificación de las compras de insumos también adolecen de una metodología sistemática y apoyada en datos. La determinación de las cantidades de alimentos a adquirir y preparar se basa, en gran medida, en estimaciones empíricas o históricas que no siempre reflejan la demanda real o las preferencias de los estudiantes. Esta carencia de información precisa y en tiempo real sobre el consumo y las existencias conduce a dos problemas críticos: por un lado, el desperdicio de alimentos debido a una sobreproducción o una mala estimación de la demanda; por otro lado, la escasez de raciones en momentos de alta afluencia, lo que deja a algunos estudiantes sin el servicio. La falta de un sistema que permita registrar y analizar patrones de consumo dificulta la toma de decisiones informadas para la elaboración de menús más variados, nutritivos y acordes a las necesidades de la población universitaria.

Adicionalmente, el control de inventario de los productos alimenticios y otros insumos del comedor se gestiona de forma manual, lo que implica un registro en cuadernos o hojas de cálculo básicas. Este método es propenso a errores de transcripción, dificulta la actualización en tiempo real de las existencias y complica la identificación de productos próximos a caducar o con bajo stock. La ausencia de un sistema automatizado para el inventario impide una gestión eficiente de la cadena de suministro, afectando la capacidad de negociación con proveedores y la optimización de los costos operativos. La trazabilidad de los productos es limitada, lo que puede comprometer la calidad y seguridad alimentaria en caso de incidentes.

La generación de reportes y estadísticas sobre el funcionamiento del comedor es otra área crítica afectada por la manualidad de los procesos. La recopilación de datos para elaborar informes sobre el número de comensales, el consumo de alimentos, los costos operativos o la satisfacción del usuario es una tarea laboriosa y que consume mucho tiempo. Los informes resultantes suelen ser estáticos, desactualizados y carecen de la profundidad analítica necesaria para una toma de decisiones estratégica. Esta limitación impide a las autoridades universitarias evaluar de manera efectiva el desempeño del comedor, identificar áreas de mejora y formular políticas que optimicen el servicio y la asignación de recursos.

En resumen, la situación problemática del comedor universitario de la UNERG se caracteriza por una gestión ineficiente, una experiencia de usuario deficiente y una limitada capacidad para la toma de decisiones estratégicas, todo ello derivado de la dependencia de procesos manuales y la ausencia de un sistema de gestión integral. Esta realidad subraya la urgencia de implementar una solución tecnológica que no solo automatice las operaciones diarias, sino que también proporcione herramientas para una administración más transparente, eficiente y orientada a la satisfacción de la comunidad universitaria.

4.2 Justificación del proyecto

El **Sistema de Gestión para el Comedor Universitario de la UNERG** se justifica plenamente como una iniciativa estratégica y necesaria para abordar las deficiencias operativas y mejorar sustancialmente la calidad del servicio que se ofrece a la comunidad estudiantil. La problemática descrita en el diagnóstico situacional, caracterizada por la ineficiencia de los procesos manuales, las largas colas, el control limitado de acceso y la falta de datos precisos para la toma de decisiones, exige una intervención tecnológica que transforme la administración del comedor en un modelo más ágil, transparente y centrado en el usuario. La implementación de un sistema de esta naturaleza no es meramente una actualización tecnológica, sino una inversión en la eficiencia institucional, el bienestar

estudiantil y la modernización de los servicios universitarios, tal como lo destacan estudios sobre la optimización de servicios públicos mediante la digitalización (Rodríguez & Pérez, 2023).

En primer lugar, la justificación se fundamenta en la imperiosa necesidad de optimizar la **eficiencia operativa**. Los procesos manuales actuales consumen una cantidad considerable de tiempo y recursos humanos que podrían ser redirigidos a tareas de mayor valor añadido. La automatización de la gestión de menús, el control de acceso, el inventario y la generación de reportes permitirá reducir drásticamente los tiempos de espera, minimizar los errores humanos y liberar al personal de tareas repetitivas, permitiéndoles enfocarse en la calidad del servicio y la atención al comensal. Un sistema digitalizado facilitará la planificación anticipada, la gestión proactiva de insumos y la coordinación entre las diferentes áreas del comedor, resultando en una operación más fluida y rentable.

En segundo lugar, el proyecto se justifica por su impacto directo en la **mejora de la experiencia del usuario**. Los estudiantes son los principales beneficiarios de este servicio, y la eliminación de las largas colas y la agilización del proceso de acceso contribuirán significativamente a su satisfacción. Un sistema que permita una identificación rápida y segura, junto con la posibilidad de consultar menús de forma anticipada, empoderará a los estudiantes y les brindará un servicio más cómodo y eficiente. La reducción de la frustración y el tiempo perdido en el comedor se traducirá en una mejor percepción del servicio universitario y un mayor aprovechamiento del tiempo académico.

En tercer lugar, la implementación del sistema se justifica por su capacidad para facilitar la **toma de decisiones basada en datos**. Actualmente, la ausencia de información precisa y en tiempo real limita la capacidad de la administración para evaluar el desempeño del comedor, identificar tendencias de consumo, gestionar eficientemente el inventario y optimizar los costos. El sistema propuesto generará automáticamente reportes detallados sobre el número de comensales, las preferencias de menú, el consumo de ingredientes y los costos operativos. Esta información valiosa permitirá a las autoridades universitarias tomar decisiones estratégicas informadas, ajustar la oferta de menús, negociar de manera más efectiva con proveedores y asignar recursos de manera más eficiente, lo que se traduce en una gestión más inteligente y sostenible del comedor.

Además, el proyecto contribuye a la **transparencia y el control**. Un sistema automatizado proporciona un registro inmutable de todas las transacciones, desde la entrada de insumos hasta el servicio de raciones. Esto no solo mejora la rendición de cuentas, sino que también reduce la posibilidad de irregularidades y optimiza la supervisión de los procesos. La

trazabilidad de los alimentos y el control preciso del inventario son fundamentales para garantizar la calidad y seguridad alimentaria, aspectos cruciales en un servicio de esta magnitud.

Finalmente, la justificación se extiende a la **modernización institucional** de la UNERG. En un entorno universitario cada vez más digitalizado, la adopción de tecnologías para optimizar los servicios internos es un paso esencial para mantener la relevancia y la competitividad. Un sistema de gestión para el comedor no solo resuelve un problema operativo específico, sino que también posiciona a la universidad como una institución innovadora y comprometida con la mejora continua de sus infraestructuras y servicios, alineándose con las expectativas de una generación de estudiantes nativos digitales. Este proyecto, por tanto, no solo es una solución a un problema puntual, sino un catalizador para la transformación digital y la excelencia en la gestión universitaria.

4.3 Objetivos del proyecto

El proyecto de "Diseño e Implementación de un Sistema de Gestión para el Comedor Universitario de la UNERG" se enmarca en la necesidad de modernizar y optimizar los procesos administrativos y operativos de este servicio esencial. Para lograr este propósito, se ha establecido un objetivo general que guía toda la investigación y el desarrollo, complementado por una serie de objetivos específicos que detallan las etapas y los resultados esperados del proyecto. Estos objetivos específicos han sido formulados utilizando verbos de acción que denotan un proceso de investigación, análisis, diseño y desarrollo, asegurando una hoja de ruta clara y medible para la consecución del sistema.

Objetivo General:

Diseñar e implementar un sistema de gestión integral para el comedor universitario de la UNERG que automatice los procesos de planificación de menús, control de acceso, gestión de inventario y generación de reportes, con el fin de optimizar la eficiencia operativa, mejorar la experiencia del usuario y facilitar la toma de decisiones estratégicas.

Objetivos Específicos:

- 1. Analizar** los procesos actuales de gestión del comedor universitario de la UNERG, identificando las deficiencias, los cuellos de botella y los requerimientos funcionales y no funcionales de los usuarios (estudiantes, personal administrativo, cocineros y directivos) para la formulación de una solución tecnológica pertinente. Este análisis profundo implicará la observación directa de las operaciones diarias, la realización de entrevistas estructuradas con el personal clave y la revisión de la documentación existente.

Se buscará comprender en detalle el flujo de trabajo desde la recepción de insumos hasta el servicio de las raciones, prestando especial atención a los puntos donde la manualidad genera ineficiencia o errores. La identificación de los requerimientos funcionales se centrará en las funcionalidades que el sistema debe ofrecer para satisfacer las necesidades operativas, mientras que los requerimientos no funcionales abordarán aspectos como la seguridad, el rendimiento, la usabilidad y la escalabilidad del sistema.

2. **Investigar** las tecnologías de desarrollo de software más adecuadas y las mejores prácticas en el diseño e implementación de sistemas de gestión para comedores universitarios o instituciones de gran escala. Esta investigación abarcará la exploración de diferentes arquitecturas de software (ej., cliente-servidor, web, microservicios), lenguajes de programación (ej., Python, PHP, JavaScript), frameworks (ej., Django, Laravel, React, Angular) y sistemas de gestión de bases de datos (ej., MySQL, PostgreSQL, MongoDB). Se realizará un estudio comparativo de soluciones existentes en el mercado o en otras universidades para identificar funcionalidades clave, modelos de datos eficientes y estrategias de implementación exitosas. La investigación también se enfocará en las tendencias actuales en interfaces de usuario (UI) y experiencia de usuario (UX) para garantizar que el sistema sea intuitivo, accesible y fácil de usar para todos los perfiles de usuarios.
3. **Establecer** un conjunto detallado de requisitos funcionales y no funcionales para el sistema de gestión del comedor, basándose en el análisis de los procesos actuales y la investigación de las mejores prácticas. Este objetivo implica la definición precisa de cada funcionalidad que el sistema debe ofrecer, como la gestión de menús, el control de acceso mediante identificación digital, la administración de inventario en tiempo real, el registro de comensales y la generación de reportes personalizados. Asimismo, se establecerán los requisitos no funcionales relacionados con la seguridad de la información, la robustez del sistema ante fallos, la capacidad de respuesta, la facilidad de mantenimiento y la escalabilidad para futuras expansiones. La documentación de estos requisitos será fundamental para guiar las fases posteriores de diseño y desarrollo, asegurando que el producto final cumpla con las expectativas y necesidades de la UNERG.
4. **Diseñar** la arquitectura del sistema, incluyendo el modelo de base de datos, la interfaz de usuario (UI) y la experiencia de usuario (UX), así como los módulos principales que conformarán la aplicación. Esta fase de diseño se

traducirá en la creación de diagramas UML (Diagramas de Casos de Uso, Diagramas de Clases, Diagramas de Secuencia, etc.) para modelar la estructura y el comportamiento del sistema. El diseño de la base de datos se centrará en la eficiencia, la integridad y la seguridad de los datos, definiendo las tablas, relaciones y restricciones necesarias para almacenar toda la información relevante del comedor. El diseño de la UI/UX se enfocará en crear una interfaz intuitiva, atractiva y funcional, que facilite la interacción de los usuarios con el sistema, minimizando la curva de aprendizaje y maximizando la eficiencia en la ejecución de tareas.

5. **Desarrollar** los módulos del sistema de gestión para el comedor universitario, utilizando las tecnologías y el framework seleccionados, implementando las funcionalidades definidas en la fase de diseño. Esta etapa implica la codificación de la aplicación, siguiendo las mejores prácticas de desarrollo de software para garantizar un código limpio, modular y mantenable. Se construirán los módulos de gestión de menús, que permitirá la planificación y publicación de la oferta gastronómica; el módulo de control de acceso, que automatizará la identificación de los comensales; el módulo de inventario, para el seguimiento en tiempo real de los insumos; el módulo de registro y gestión de usuarios; y el módulo de generación de reportes, que proporcionará información valiosa para la toma de decisiones. Se realizarán pruebas unitarias y de integración de forma continua para asegurar la correcta funcionalidad de cada componente.
6. **Implementar** el sistema de gestión desarrollado en el entorno real del comedor universitario de la UNERG, realizando las configuraciones necesarias y asegurando la integración con la infraestructura tecnológica existente. Esta fase incluirá la instalación del software en los servidores, la migración de datos si fuera necesario, y la configuración de los dispositivos de acceso (ej., lectores de códigos QR). Se llevará a cabo un plan de capacitación exhaustivo dirigido al personal del comedor y a los usuarios finales para asegurar una adopción exitosa del sistema. La implementación también contemplará la puesta en marcha de un período de pruebas piloto para identificar y corregir cualquier eventualidad antes del lanzamiento oficial, garantizando una transición suave de los procesos manuales a los automatizados.
7. **Evaluar** la funcionalidad, el rendimiento y la usabilidad del sistema de gestión implementado, así como su impacto en la eficiencia operativa del comedor y la satisfacción de los usuarios. La evaluación se realizará mediante pruebas de

aceptación de usuario (UAT), encuestas de satisfacción a estudiantes y personal, y el análisis de métricas clave de rendimiento (ej., tiempo de procesamiento de acceso, precisión del inventario, tiempo de generación de reportes). Se compararán los resultados obtenidos con los objetivos establecidos inicialmente para determinar el grado de éxito del proyecto y la medida en que el sistema ha logrado resolver la problemática planteada. Esta evaluación final proporcionará retroalimentación valiosa para futuras mejoras y el mantenimiento evolutivo del sistema.

4.4 Procesos que se van a automatizar

El **Sistema de Gestión para el Comedor Universitario de la UNERG** está diseñado para transformar radicalmente la operación actual del servicio de alimentación, pasando de un modelo manual y propenso a errores a uno automatizado, eficiente y basado en datos. La automatización se centrará en los procesos clave que actualmente generan los mayores desafíos y cuellos de botella, buscando optimizar cada etapa del ciclo de servicio del comedor. Esta transformación no solo mejorará la experiencia de los usuarios, sino que también proporcionará a la administración herramientas poderosas para una gestión más estratégica y sostenible.

Uno de los procesos fundamentales a automatizar es la **Gestión de Menús y Planificación de la Oferta Gastronómica**. Actualmente, la planificación de los menús se realiza de forma manual, lo que dificulta la adaptación a las preferencias de los estudiantes, la gestión de la estacionalidad de los productos y la optimización nutricional. El sistema permitirá a los administradores del comedor diseñar, programar y publicar menús semanales o mensuales de manera digital. Esto incluirá la posibilidad de registrar los ingredientes necesarios para cada plato, calcular automáticamente las cantidades requeridas en función del número estimado de comensales, y generar listas de compras optimizadas. Además, el sistema podrá integrar un módulo para la recolección de feedback de los estudiantes sobre los menús, permitiendo una adaptación más dinámica y centrada en el usuario. La publicación digital de los menús, accesible a través de una interfaz web o móvil, informará a los estudiantes con antelación, reduciendo la incertidumbre y mejorando la planificación de sus comidas.

El **Control de Acceso al Comedor** es otro proceso crítico que será completamente automatizado. La verificación manual de la identidad de los estudiantes y el registro de su asistencia son las principales causas de las largas colas y la ineficiencia en la entrada. El sistema implementará un mecanismo de identificación digital, como el uso de códigos QR generados por el sistema y asociados a cada estudiante, o la lectura de carnets universitarios con tecnología RFID o códigos de barras. Al escanear su identificación, el sistema

verificará instantáneamente la elegibilidad del estudiante (ej., si es un estudiante activo, si tiene raciones disponibles) y registrará automáticamente su entrada. Esto no solo agilizará el flujo de comensales, reduciendo drásticamente los tiempos de espera, sino que también proporcionará un registro preciso y en tiempo real del número de raciones servidas, eliminando la posibilidad de errores humanos y mejorando la seguridad y el control sobre el servicio.

La **Gestión de Inventario de Alimentos e Insumos** será transformada de un proceso manual y propenso a errores a uno digital y en tiempo real. El sistema permitirá registrar todas las entradas y salidas de productos del almacén del comedor, desde la recepción de mercancías de proveedores hasta el consumo en la cocina. Cada producto tendrá un registro detallado que incluirá su fecha de entrada, fecha de caducidad, cantidad y proveedor. El sistema generará alertas automáticas cuando el stock de un producto esté por debajo de un umbral predefinido o cuando un producto esté próximo a caducar, facilitando la gestión proactiva de las compras y minimizando el desperdicio. Además, se podrá llevar un control de los proveedores, sus precios y los tiempos de entrega, optimizando la cadena de suministro y contribuyendo a la reducción de costos operativos. La trazabilidad de los productos mejorará significativamente, lo que es crucial para la seguridad alimentaria.

El **Registro y Gestión de Comensales** también se automatizará para ofrecer una administración más eficiente de la base de usuarios. El sistema permitirá el registro de nuevos estudiantes y personal autorizado para hacer uso del comedor, almacenando su información relevante de manera segura y organizada. Se podrán asignar y gestionar las raciones disponibles para cada usuario, así como llevar un historial de su consumo. Las actualizaciones de datos de los usuarios serán sencillas y rápidas, asegurando que la información esté siempre al día. Esta automatización eliminará la necesidad de registros en papel, reduciendo la carga administrativa y mejorando la precisión de los datos de los usuarios.

Finalmente, la **Generación de Reportes y Estadísticas** será una de las funcionalidades más valiosas del sistema. Actualmente, la creación de informes es una tarea manual y laboriosa que produce datos estáticos y a menudo desactualizados. El sistema automatizará la recopilación y el procesamiento de datos para generar una variedad de reportes en tiempo real, incluyendo: reportes de consumo diario, semanal y mensual; estadísticas de asistencia por franja horaria o día de la semana; análisis de preferencias de menú; reportes de inventario y costos; y proyecciones de demanda. Estos informes proporcionarán a la dirección del comedor y a las autoridades universitarias una visión clara y detallada del funcionamiento del servicio, permitiendo una toma de decisiones informada y estratégica para la optimización continua del comedor universitario de la UNERG.

5. DETERMINACIÓN, INSTALACIÓN Y CONFIGURACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

La fase de determinación, instalación y configuración de las herramientas de desarrollo constituye un pilar fundamental en la materialización de cualquier proyecto de software, y en el caso del **Sistema de Gestión para el Comedor Universitario de la UNERG**, su correcta selección y preparación son cruciales para garantizar la eficiencia, escalabilidad y mantenibilidad del producto final. Esta etapa no solo implica la elección de tecnologías específicas, sino también la definición de una arquitectura robusta y la adopción de una metodología de desarrollo que guíe el proceso de construcción de manera estructurada y coherente. La decisión sobre cada herramienta y enfoque se fundamenta en un análisis exhaustivo de los requisitos funcionales y no funcionales del sistema, así como en las mejores prácticas de la ingeniería de software, buscando optimizar los recursos disponibles y asegurar la entrega de una solución de alta calidad que resuelva las problemáticas identificadas en el diagnóstico situacional.

5.1 Plataforma de Desarrollo

La elección de la **plataforma de desarrollo** para el Sistema de Gestión del Comedor Universitario se orientó hacia un entorno que garantizara la máxima accesibilidad, flexibilidad y capacidad de integración dentro del ecosistema tecnológico de la UNERG. Considerando la naturaleza del servicio que se busca optimizar, el cual involucra a una amplia comunidad universitaria (estudiantes, personal administrativo y proveedores), se determinó que una **plataforma web** sería la opción más idónea. Esta decisión se justifica por múltiples ventajas inherentes a las aplicaciones basadas en la web, tales como la independencia del sistema operativo del usuario final, la facilidad de despliegue y actualización centralizada, y la posibilidad de acceso desde cualquier dispositivo con conexión a internet, eliminando las barreras geográficas y temporales que podrían presentarse con soluciones de escritorio.

Una plataforma web permite que el sistema sea accesible a través de navegadores estándar, lo que reduce significativamente los costos de instalación y mantenimiento en los equipos de los usuarios. Además, facilita la implementación de un modelo de **acceso distribuido**, donde diferentes perfiles de usuario (administradores del comedor, personal de cocina, estudiantes) pueden interactuar con el sistema desde sus respectivas ubicaciones, sin necesidad de instalaciones complejas o configuraciones específicas. Esta característica es particularmente relevante para un comedor universitario, donde la gestión de menús, el registro de comensales y el control de inventario requieren una interacción constante y

fluida entre diversos actores. La escalabilidad inherente a las arquitecturas web también asegura que el sistema pueda crecer y adaptarse a futuras demandas, como un aumento en la población estudiantil o la incorporación de nuevas funcionalidades, sin requerir una reingeniería completa de la infraestructura.

5.2 Arquitectura del sistema de información

La arquitectura del sistema de información para el Sistema de Gestión del Comedor Universitario se concibió bajo el paradigma **Cliente-Servidor**, un modelo ampliamente adoptado en el desarrollo de aplicaciones distribuidas por su robustez, escalabilidad y clara separación de responsabilidades. Este enfoque arquitectónico es fundamental para garantizar que el sistema pueda manejar eficientemente las interacciones de múltiples usuarios simultáneamente, procesar grandes volúmenes de datos y mantener la integridad y seguridad de la información. En esencia, la arquitectura Cliente-Servidor divide las tareas entre dos tipos de componentes principales: los **clientes**, que solicitan servicios, y los **servidores**, que los proporcionan.

En el contexto de este proyecto, el **cliente** se materializa principalmente a través de un navegador web estándar, que ejecuta la interfaz de usuario del sistema. Este componente es responsable de presentar la información al usuario, capturar sus entradas (por ejemplo, selección de menú, registro de asistencia) y enviar solicitudes al servidor. La lógica de presentación y una parte de la validación de datos pueden residir en el lado del cliente para mejorar la experiencia del usuario y reducir la carga del servidor. Por otro lado, el **servidor** es el componente central que gestiona la lógica de negocio, el acceso a la base de datos y la seguridad. Cuando un cliente envía una solicitud, el servidor la procesa, interactúa con la base de datos si es necesario, y devuelve una respuesta al cliente, que puede ser una página web actualizada, datos específicos o una confirmación de una operación.

La interacción entre el cliente y el servidor se realiza a través de protocolos de comunicación estándar, como HTTP/HTTPS, lo que garantiza la interoperabilidad y la seguridad en la transmisión de datos. Dentro del servidor, la arquitectura se puede subdividir en capas lógicas para una mejor organización y mantenibilidad. Típicamente, se distinguen al menos tres capas: la **capa de presentación** (que interactúa directamente con el cliente y se encarga de formatear las respuestas), la **capa de lógica de negocio** (que contiene las reglas y procesos específicos del comedor, como la gestión de menús, el control de inventario y la autenticación de usuarios), y la **capa de acceso a datos** (que se encarga de la interacción con la base de datos, realizando operaciones de lectura, escritura, actualización y eliminación de información). Esta separación de capas no solo facilita el desarrollo modular y la depuración, sino que también permite la evolución independiente de cada componente, lo que es vital para la adaptabilidad a largo plazo del sistema.

5.3 Selección del entorno del sistema

La selección del **entorno del sistema** abarca la elección de las tecnologías específicas que conformarán el stack de desarrollo, desde el lenguaje de programación hasta el sistema de gestión de bases de datos y los frameworks asociados. Para el Sistema de Gestión del Comedor Universitario, se optó por un conjunto de herramientas que combinan robustez, flexibilidad, una amplia comunidad de soporte y una curva de aprendizaje manejable, lo que es esencial para el éxito del proyecto. La decisión se basó en la necesidad de construir una aplicación web dinámica, segura y eficiente, capaz de manejar las operaciones diarias del comedor de la UNERG.

Para el desarrollo del **backend**, se seleccionó **PHP** como lenguaje de programación principal, complementado con el framework **Laravel**. PHP es un lenguaje ampliamente utilizado para el desarrollo web, conocido por su facilidad de uso, rendimiento y la vasta cantidad de recursos disponibles. Laravel, por su parte, es un framework MVC (Modelo-Vista-Controlador) que proporciona una estructura organizada, herramientas potentes para la gestión de bases de datos (Eloquent ORM), enrutamiento, autenticación y seguridad, acelerando significativamente el proceso de desarrollo y promoviendo la escritura de código limpio y mantenible. Esta combinación permite construir una API robusta que gestionará la lógica de negocio y la interacción con la base de datos.

En cuanto al **frontend**, se utilizará una combinación de **HTML5, CSS3 y JavaScript**. HTML5 proporcionará la estructura semántica de las páginas web, CSS3 se encargará del diseño y la presentación visual, y JavaScript, junto con un framework moderno como **Vue.js** o **React** (la elección final se determinará en fases posteriores, pero ambos ofrecen una excelente reactividad y gestión de componentes), permitirá crear una interfaz de usuario dinámica, interactiva y responsive. Esta elección asegura una experiencia de usuario fluida y moderna, adaptándose a diferentes dispositivos y tamaños de pantalla.

Para la gestión de la **base de datos**, se optó por **MySQL**. MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, reconocido por su fiabilidad, rendimiento y escalabilidad. Es una opción popular para aplicaciones web y ofrece una integración excelente con PHP y Laravel. La base de datos almacenará toda la información crítica del sistema, incluyendo datos de usuarios, menús, inventario de alimentos, registros de asistencia y transacciones. Finalmente, para el control de versiones y la colaboración en el desarrollo, se empleará **Git**, una herramienta estándar de la industria que permite a los desarrolladores trabajar de manera concurrente en el código, gestionar cambios y mantener un historial completo del proyecto.

5.4 Metodología para el desarrollo

La elección de una **metodología de desarrollo** es un factor determinante en la planificación, ejecución y éxito de cualquier proyecto de software. Es crucial diferenciar entre las metodologías aplicables a sistemas educativos y aquellas diseñadas para sistemas de información, ya que sus objetivos, enfoques y ciclos de vida son intrínsecamente distintos. Los **sistemas educativos**, como plataformas de e-learning o herramientas pedagógicas, se centran en el diseño instruccional, la entrega de contenido, la evaluación del aprendizaje y la interacción didáctica. Sus metodologías suelen estar alineadas con modelos como ADDIE (Análisis, Diseño, Desarrollo, Implementación, Evaluación) o enfoques centrados en el usuario para la experiencia de aprendizaje.

Por otro lado, los **sistemas de información**, categoría a la que pertenece el Sistema de Gestión para el Comedor Universitario, se enfocan en la automatización de procesos de negocio, la gestión de datos, la toma de decisiones y la mejora de la eficiencia operativa. Las metodologías para estos sistemas se derivan de la ingeniería de software y buscan estructurar el ciclo de vida del desarrollo, desde la recopilación de requisitos hasta el despliegue y mantenimiento. Dada la naturaleza del proyecto, que implica el diseño y la implementación de una aplicación de software para optimizar la administración de un servicio, la metodología seleccionada debe ser robusta, adaptable y orientada a resultados.

Para el desarrollo del Sistema de Gestión del Comedor Universitario, se adoptará la metodología **Scrum**, un marco de trabajo ágil que se alinea perfectamente con las necesidades de un proyecto de desarrollo de software en un entorno dinámico. Scrum promueve un enfoque iterativo e incremental, lo que permite la entrega de funcionalidades en ciclos cortos y definidos, conocidos como "sprints" (generalmente de 2 a 4 semanas). Esta metodología es ideal para proyectos donde los requisitos pueden evolucionar o refinarse a lo largo del tiempo, y donde la retroalimentación temprana y continua de los usuarios es valiosa para asegurar que el producto final satisfaga sus expectativas.

Las principales características de Scrum que lo hacen adecuado para este proyecto incluyen: la **planificación adaptativa**, que permite ajustar el rumbo del proyecto en función de los aprendizajes y cambios; el **desarrollo iterativo**, que facilita la construcción de funcionalidades en pequeñas porciones manejables; la **entrega temprana y continua** de software funcional, lo que permite validar el progreso y obtener retroalimentación constante; y la **colaboración intensiva** entre el equipo de desarrollo y los stakeholders. Los roles clave en Scrum (Product Owner, Scrum Master y Equipo de Desarrollo) y los eventos (Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective) aseguran una comunicación efectiva, una gestión transparente del progreso y una mejora continua del proceso de desarrollo, lo que es esencial para construir un sistema complejo como el que se propone para el comedor universitario.

6. DESARROLLO DEL SISTEMA DE INFORMACIÓN

La fase de desarrollo del Sistema de Gestión para el Comedor Universitario de la UNERG representa la materialización de todas las etapas previas de análisis, planificación y diseño. En este punto crucial del proyecto, los requerimientos funcionales y no funcionales, previamente identificados y validados, se transforman en componentes de software tangibles y operativos. Esta etapa no solo implica la escritura de código, sino también la implementación de una arquitectura robusta, la configuración de bases de datos eficientes y la creación de interfaces de usuario intuitivas que garanticen una experiencia óptima para los diversos actores involucrados: estudiantes, personal administrativo del comedor y la dirección universitaria. El objetivo primordial es construir un sistema que no solo resuelva las problemáticas actuales de gestión manual y las ineficiencias operativas, sino que también establezca una plataforma escalable y mantenible para futuras mejoras y adaptaciones, consolidando así un pilar tecnológico fundamental para el bienestar estudiantil y la optimización de recursos en la UNERG.

El proceso de desarrollo se ha concebido bajo un enfoque estructurado, que permite una progresión lógica desde la conceptualización hasta la implementación final. Se ha puesto especial énfasis en la modularidad del sistema, lo que facilita la gestión de la complejidad, la reutilización de componentes y la asignación de tareas específicas al equipo de desarrollo. Cada módulo ha sido diseñado para operar de manera independiente pero interconectada, asegurando la coherencia y la integridad de los datos a lo largo de todo el sistema. La elección de las tecnologías y herramientas de desarrollo, detalladas en la sección anterior, ha sido fundamental para garantizar la eficiencia en la codificación, la seguridad de la información y la capacidad de respuesta del sistema ante las demandas de una comunidad universitaria dinámica. Este capítulo desglosa las etapas clave de este proceso, desde la descripción detallada de los requerimientos hasta la implementación de los módulos principales, pasando por la fase de diseño que sentó las bases arquitectónicas del sistema.

6.1.1 Descripción

El Sistema de Gestión para el Comedor Universitario de la UNERG es una aplicación web integral diseñada para automatizar y optimizar los procesos administrativos y operativos asociados al servicio de alimentación estudiantil. Su concepción surge de la necesidad imperante de superar las limitaciones impuestas por la gestión manual, que históricamente ha generado ineficiencias, demoras en el servicio, dificultades en el control de raciones y una limitada capacidad para generar reportes precisos sobre el consumo y la asistencia. Este

sistema busca transformar radicalmente la experiencia del usuario, tanto para los estudiantes como para el personal del comedor, al proporcionar herramientas digitales que agilizan cada etapa del proceso, desde la planificación y publicación de menús hasta el control de acceso y la generación de estadísticas vitales para la toma de decisiones.

La arquitectura del sistema se ha diseñado para ser robusta y escalable, permitiendo su adaptación a futuras expansiones o cambios en las políticas del comedor. Se ha priorizado la creación de una interfaz de usuario intuitiva y accesible, garantizando que usuarios con diferentes niveles de habilidad tecnológica puedan interactuar con el sistema de manera efectiva. La seguridad de la información es un pilar fundamental, implementando mecanismos de autenticación y autorización que protegen los datos personales de los estudiantes y la información operativa del comedor. Además, la capacidad de generar reportes detallados y en tiempo real es una característica clave, proporcionando a la administración universitaria una visión clara del rendimiento del servicio, el consumo de alimentos y la asistencia, lo que facilita una gestión más estratégica y eficiente de los recursos.

El sistema se estructura en módulos interconectados, cada uno enfocado en una funcionalidad específica, lo que permite una gestión más organizada y un desarrollo incremental. Estos módulos incluyen la gestión de usuarios y roles, la administración de menús y raciones, el control de acceso al comedor mediante mecanismos de validación digital, y la generación de informes personalizados. La implementación de este sistema no solo busca resolver problemas operativos inmediatos, sino también sentar las bases para una modernización continua del servicio de comedor, alineándose con las tendencias tecnológicas actuales y las expectativas de una comunidad universitaria cada vez más digitalizada. La integración de tecnologías web modernas asegura que el sistema sea accesible desde cualquier dispositivo con conexión a internet, promoviendo la flexibilidad y la conveniencia para todos los usuarios.

6.1.2 Requerimientos Funcionales del Proyecto

Los requerimientos funcionales definen las capacidades específicas que el Sistema de Gestión para el Comedor Universitario de la UNERG debe poseer para satisfacer las necesidades de los usuarios y cumplir con los objetivos del proyecto. Estos requerimientos se han derivado del análisis exhaustivo de los procesos actuales del comedor y de las expectativas de los stakeholders, asegurando que el sistema aborde directamente las problemáticas identificadas y proporcione soluciones efectivas. A continuación, se presenta una lista detallada de los requerimientos funcionales:

- **Gestión de Usuarios y Roles:**

- El sistema debe permitir el registro de nuevos usuarios (estudiantes, personal administrativo, personal de cocina).
- Debe soportar la asignación de roles específicos a cada usuario (ej. Administrador, Estudiante, Cocinero), con permisos diferenciados.
- Debe permitir la modificación y eliminación de perfiles de usuario por parte de los administradores.
- Debe incluir un módulo de autenticación seguro (inicio de sesión con usuario y contraseña).
- Debe permitir la recuperación de contraseña para los usuarios.

- **Gestión de Menús:**

- El sistema debe permitir a los administradores crear, editar y eliminar menús diarios o semanales.
- Debe permitir la especificación de los componentes de cada menú (plato principal, contorno, bebida, postre).
- Debe permitir la publicación de menús con fechas específicas de disponibilidad.
- Debe mostrar el menú actual y los próximos menús a los estudiantes.
- Debe mantener un historial de menús servidos.

- **Gestión de Raciones:**

- El sistema debe permitir a los administradores definir la cantidad de raciones disponibles por menú y por día.
- Debe controlar el consumo de raciones en tiempo real, decrementando el contador con cada servicio.
- Debe alertar cuando las raciones disponibles estén por agotarse.
- Debe permitir la asignación de raciones especiales o adicionales a usuarios específicos si es necesario.

- **Control de Acceso al Comedor:**

- El sistema debe permitir la validación de la elegibilidad del estudiante para recibir el servicio (ej. mediante código QR, carnet universitario digital).
- Debe registrar la entrada de cada estudiante al comedor, asociándola a la ración consumida.

- Debe impedir el acceso a estudiantes que ya hayan consumido su ración diaria o que no estén registrados.
- Debe proporcionar una interfaz rápida y eficiente para el personal de control de acceso.

- **Generación de Reportes y Estadísticas:**

- El sistema debe generar reportes diarios, semanales y mensuales sobre el consumo de raciones.
- Debe generar reportes de asistencia de estudiantes al comedor.
- Debe generar reportes sobre la popularidad de los menús y los componentes de los mismos.
- Debe permitir la exportación de reportes en formatos comunes (ej. PDF, CSV).
- Debe mostrar estadísticas visuales (gráficos) sobre el uso del comedor.

- **Notificaciones:**

- El sistema debe enviar notificaciones a los estudiantes sobre el menú del día o cambios importantes.
- Debe notificar a los administradores sobre el bajo stock de raciones o incidencias en el control de acceso.

- **Gestión de Inventario de Insumos (Opcional/Futuro):**

- El sistema podría, en una fase futura, permitir el registro y control de los insumos utilizados en la preparación de los menús.
- Podría generar alertas de bajo stock de insumos.

Estos requerimientos funcionales constituyen la base sobre la cual se construirá el sistema, asegurando que cada componente desarrollado contribuya directamente a la solución de las problemáticas identificadas y a la consecución de los objetivos del proyecto. La implementación de cada uno de estos puntos garantizará un sistema completo y eficiente, capaz de transformar la gestión del comedor universitario de la UNERG.

6.1.3 Requerimientos No Funcionales del Proyecto

Los requerimientos no funcionales definen las características de calidad del sistema, es decir, cómo debe funcionar el sistema, en lugar de qué debe hacer. Son cruciales para asegurar que el sistema sea usable, seguro, eficiente y mantenible a largo plazo. Para el

Sistema de Gestión para el Comedor Universitario de la UNERG, se han identificado los siguientes requerimientos no funcionales:

- **Rendimiento:**

- El sistema debe responder a las solicitudes de los usuarios en un tiempo máximo de 3 segundos para el 90% de las operaciones, incluso en picos de alta demanda (ej. hora de almuerzo).
- Debe soportar al menos 500 usuarios concurrentes sin degradación significativa del rendimiento.
- Las consultas a la base de datos deben ejecutarse de manera eficiente para minimizar los tiempos de espera.

- **Seguridad:**

- El sistema debe implementar un mecanismo de autenticación robusto para proteger el acceso no autorizado.
- Debe garantizar la confidencialidad e integridad de los datos de los usuarios y la información operativa del comedor.
- Debe aplicar políticas de autorización basadas en roles, asegurando que cada usuario solo acceda a las funcionalidades y datos para los que tiene permiso.
- Debe protegerse contra vulnerabilidades comunes de seguridad web (ej. inyección SQL, XSS).
- Debe realizar copias de seguridad periódicas de la base de datos para prevenir la pérdida de información.

- **Usabilidad:**

- La interfaz de usuario debe ser intuitiva, fácil de aprender y de usar para todos los perfiles de usuario.
- Debe proporcionar retroalimentación clara y concisa al usuario sobre el estado de sus acciones.
- Debe ser accesible para usuarios con diferentes niveles de habilidad tecnológica.
- Debe tener un diseño responsive, adaptándose correctamente a diferentes tamaños de pantalla (escritorio, tabletas, móviles).

- **Fiabilidad:**

- El sistema debe estar disponible el 99.5% del tiempo durante las horas de operación del comedor.
- Debe ser capaz de recuperarse de fallos de manera controlada, minimizando la pérdida de datos y el tiempo de inactividad.
- Debe manejar errores de entrada de datos de forma robusta, informando al usuario y evitando fallos del sistema.

- **Mantenibilidad:**

- El código fuente debe estar bien documentado y estructurado, facilitando futuras modificaciones y mejoras.
- La arquitectura del sistema debe ser modular, permitiendo la actualización o reemplazo de componentes individuales sin afectar el sistema en su totalidad.
- Debe ser fácil de diagnosticar y corregir errores.

- **Escalabilidad:**

- El sistema debe ser capaz de manejar un aumento en el número de usuarios y el volumen de datos sin requerir una reingeniería completa.
- Debe permitir la adición de nuevas funcionalidades o módulos en el futuro de manera eficiente.

- **Compatibilidad:**

- El sistema debe ser compatible con los navegadores web más utilizados (ej. Chrome, Firefox, Edge, Safari) en sus versiones más recientes.
- Debe funcionar correctamente en diferentes sistemas operativos (Windows, macOS, Linux) a través de un navegador web.

La satisfacción de estos requerimientos no funcionales es tan crítica como la implementación de las funcionalidades, ya que determinan la calidad general del sistema y su aceptación por parte de los usuarios. Un sistema funcional pero lento, inseguro o difícil de usar no cumpliría plenamente con las expectativas del proyecto.

6.1.4 Restricciones

Las restricciones son factores que limitan las opciones de diseño y desarrollo del sistema, imponiendo condiciones que deben ser consideradas y respetadas a lo largo de todo el ciclo de vida del proyecto. Para el Sistema de Gestión para el Comedor Universitario de la UNERG, se han identificado las siguientes restricciones:

- **Restricciones de Tiempo:**

- El proyecto debe ser completado dentro de un plazo establecido por la universidad para su implementación y puesta en marcha, lo que limita la extensión de las funcionalidades a desarrollar en la fase inicial.
- Las fases de diseño, codificación y pruebas deben ajustarse a un cronograma riguroso para asegurar la entrega a tiempo.

- **Restricciones Presupuestarias:**

- El desarrollo del sistema está sujeto a un presupuesto limitado, lo que influye en la selección de tecnologías (priorizando soluciones de código abierto o de bajo costo), la adquisición de hardware (si fuera necesario) y la contratación de recursos externos.
- Se debe optimizar el uso de los recursos disponibles para maximizar el valor entregado dentro de las limitaciones financieras.

- **Restricciones de Recursos Humanos:**

- El equipo de desarrollo está compuesto por un número limitado de personas, lo que requiere una asignación eficiente de tareas y una gestión cuidadosa de la carga de trabajo.
- Las habilidades y conocimientos del equipo influyen en la elección de las tecnologías y herramientas a utilizar.

- **Restricciones Tecnológicas:**

- El sistema debe ser compatible con la infraestructura tecnológica existente en la UNERG, si la hubiera, para su despliegue (ej. servidores, red).
- Puede haber preferencias o requisitos específicos de la universidad en cuanto a lenguajes de programación, bases de datos o sistemas operativos para el entorno de producción.
- La disponibilidad de licencias de software o herramientas específicas puede ser una limitación.

- **Restricciones Operacionales y Normativas:**

- El sistema debe cumplir con las políticas y normativas internas de la UNERG relacionadas con la gestión de datos estudiantiles y la operación del comedor.
- Debe integrarse, si es necesario, con otros sistemas universitarios existentes (ej. sistema de registro académico para la validación de estudiantes).
- Los procesos de trabajo actuales del comedor, aunque se busquen optimizar, pueden imponer ciertas restricciones en la forma en que se implementan las nuevas funcionalidades.

- **Restricciones de Datos:**

- La disponibilidad y calidad de los datos iniciales para la migración o carga en el sistema pueden ser una restricción.
- La privacidad y protección de los datos personales de los estudiantes deben ser una prioridad, cumpliendo con las leyes y regulaciones aplicables.

La identificación y gestión proactiva de estas restricciones son fundamentales para el éxito del proyecto, ya que permiten al equipo de desarrollo tomar decisiones informadas y ajustar el alcance del sistema para asegurar su viabilidad y entrega dentro de los parámetros establecidos.

6.2 Fase de Diseño

La fase de diseño es un pilar fundamental en el ciclo de vida del desarrollo de software, sirviendo como puente entre la definición de los requerimientos y la implementación del código. En esta etapa, se traduce la visión abstracta del sistema en una arquitectura concreta y detallada, especificando cómo se construirán los componentes, cómo interactuarán entre sí y cómo se almacenarán los datos. Para el Sistema de Gestión para el Comedor Universitario de la UNERG, la fase de diseño se centró en la creación de modelos visuales y estructurales que permitieran comprender la complejidad del sistema antes de iniciar la codificación, minimizando así errores y retrabajos en etapas posteriores. Se utilizaron diversas herramientas y técnicas de modelado para representar los diferentes aspectos del sistema, desde la interacción de los usuarios hasta la estructura interna de la base de datos.

El diseño arquitectónico del sistema se basó en un enfoque modular, lo que facilita la escalabilidad, la mantenibilidad y la asignación de responsabilidades dentro del equipo de desarrollo. Se definió una arquitectura de tres capas (presentación, lógica de negocio y

datos) para asegurar una clara separación de preocupaciones, lo que permite que los cambios en una capa no afecten directamente a las otras. Esta estructura promueve la robustez y la flexibilidad del sistema. A continuación, se detallan los artefactos de diseño clave generados durante esta fase, que incluyen diagramas de casos de uso, diagramas de procesos, el diagrama entidad-relación de la base de datos y el diseño de las interfaces de usuario.

Diagrama de Casos de Uso con Avatares

El **Diagrama de Casos de Uso** es una herramienta esencial en el diseño de software, que permite visualizar las interacciones entre los usuarios (actores) y el sistema, describiendo las funcionalidades que el sistema debe ofrecer desde la perspectiva del usuario. Para el Sistema de Gestión del Comedor Universitario, este diagrama fue crucial para identificar y formalizar los roles de los diferentes tipos de usuarios y las tareas que cada uno realizaría. Los "avatares" en este contexto representan a los actores principales: el **Administrador del Comedor**, el **Estudiante** y el **Personal de Control de Acceso**. **Cada avatar interactúa con un conjunto específico de casos de uso, que representan las funcionalidades del sistema.**

Por ejemplo, el Administrador del Comedor** interactúa con casos de uso como "Gestionar Menús", "Gestionar Raciones", "Gestionar Usuarios" y "Generar Reportes". El Estudiante interactúa con "Consultar Menú", "Validar Acceso al Comedor" (indirectamente, a través del personal de control) y "Recibir Notificaciones". El Personal de Control de Acceso interactúa directamente con "Validar Acceso de Estudiante". Este diagrama permitió una comprensión clara de las responsabilidades de cada actor y las funcionalidades críticas del sistema, asegurando que todos los requerimientos funcionales fueran cubiertos por al menos un caso de uso. La representación visual de estas interacciones facilitó la comunicación entre el equipo de desarrollo y los stakeholders, validando que el diseño inicial se alineara con las expectativas operativas del comedor.

Diagrama de Procesos del Sistema Propuesto

El **Diagrama de Procesos del Sistema Propuesto**, a menudo representado mediante diagramas de flujo o BPMN (Business Process Model and Notation), ilustra la secuencia de actividades y decisiones que se llevarán a cabo dentro del sistema para ejecutar una funcionalidad específica. Este diagrama fue fundamental para modelar el flujo de trabajo automatizado del comedor, desde la planificación del menú hasta el registro del consumo de raciones. Por ejemplo, un proceso clave modelado fue el "Proceso de Control de Acceso al Comedor". Este proceso inicia cuando un estudiante se presenta en el punto de control. El personal de control utiliza el sistema para "Escanear Identificación del Estudiante" (ej.

código QR). El sistema entonces "Verifica Elegibilidad y Ración Disponible". Si el estudiante es elegible y tiene ración, el sistema "Registra Consumo de Ración" y "Permite Acceso". Si no es elegible o ya consumió su ración, el sistema "Deniega Acceso" y "Muestra Mensaje de Error".

Este tipo de diagrama permitió identificar cuellos de botella potenciales, optimizar la secuencia de pasos y asegurar que el sistema manejara adecuadamente todas las posibles situaciones y excepciones. La visualización de los flujos de trabajo ayudó a garantizar que el sistema propuesto fuera eficiente y coherente con los objetivos de automatización, eliminando las inefficiencias presentes en el proceso manual. Además, sirvió como una guía clara para la fase de codificación, indicando la lógica de negocio que debía ser implementada en cada módulo.

Diagrama Entidad-Relación de Base de Datos (DER)

El **Diagrama Entidad-Relación (DER)** es una representación gráfica de la estructura de la base de datos, mostrando las entidades (tablas), sus atributos (columnas) y las relaciones entre ellas. Para el Sistema de Gestión del Comedor, el DER fue crucial para diseñar una base de datos robusta, normalizada y eficiente, capaz de almacenar y gestionar toda la información relevante. Las entidades principales identificadas incluyen:

- **Usuarios:** Almacena información de estudiantes, administradores y personal (ID, nombre, apellido, cédula, rol, contraseña, estado).
- **Roles:** Define los diferentes tipos de usuarios y sus permisos (ID, nombre_rol, descripción).
- **Menús:** Contiene la información de los menús diarios o semanales (ID, fecha, descripción, estado).
- **Platos:** Detalla los componentes de cada menú (ID, nombre_plato, tipo_plato - ej. principal, contorno, bebida).
- **Menu_Plato:** Tabla intermedia para la relación muchos a muchos entre Menús y Platos, especificando qué platos componen cada menú.
- **Raciones:** Registra las raciones disponibles y consumidas (ID, id_menu, cantidad_total, cantidad_disponible).
- **Consumo_Racion:** Registra cada vez que un estudiante consume una ración (ID, id_usuario, id_racion, fecha_hora_consumo).
- **Notificaciones:** Almacena los mensajes enviados a los usuarios (ID, id_usuario, mensaje, fecha_envio, leido).

Las relaciones entre estas entidades se definieron con sus respectivas cardinalidades (uno a uno, uno a muchos, muchos a muchos), asegurando la integridad referencial y la consistencia de los datos. Por ejemplo, un usuario puede tener un rol, un menú puede tener varios platos, y un estudiante puede consumir varias raciones a lo largo del tiempo. Este diseño de base de datos es fundamental para la eficiencia del sistema, permitiendo consultas rápidas y un almacenamiento de datos organizado y sin redundancias.

Capturas de Interfaces con Descripción de Funciones

Aunque no es posible mostrar imágenes directamente, la fase de diseño incluyó la creación de prototipos y maquetas de las interfaces de usuario, que luego se tradujeron en las siguientes vistas clave del sistema. Estas capturas conceptuales fueron esenciales para visualizar la interacción del usuario y refinar la experiencia antes de la codificación.

1. Interfaz de Inicio de Sesión (Login):

Esta interfaz es la puerta de entrada al sistema. Presenta un diseño limpio y minimalista, con campos para el **nombre de usuario** (o cédula) y la **contraseña**. Incluye un botón prominente de "Iniciar Sesión" y un enlace para "Olvidé mi Contraseña". El diseño es responsivo, adaptándose a dispositivos móviles y de escritorio. Su función principal es autenticar al usuario y redirigirlo a la sección correspondiente según su rol (administrador, estudiante, personal de control). Se priorizó la claridad y la seguridad, con validaciones en tiempo real para los campos de entrada.

2. Interfaz de Gestión de Menús (Administrador):

Esta pantalla es accesible solo para el rol de Administrador del Comedor. Muestra una tabla con los menús planificados, incluyendo la **fecha**, una **descripción breve** y el **estado** (ej. "Publicado", "Borrador"). Permite al administrador **crear un nuevo menú**, **editar** uno existente o **eliminarlo**. Al crear o editar un menú, se abre un formulario detallado donde se pueden seleccionar los platos (principal, contorno, bebida, postre) de una lista predefinida, establecer la cantidad de raciones disponibles y la fecha de publicación. La interfaz incluye filtros por fecha y un buscador para facilitar la navegación entre los menús. Su función es centralizar la planificación y publicación de la oferta gastronómica del comedor.

3. Interfaz de Control de Acceso (Personal de Control):

Diseñada para ser utilizada en el punto de entrada del comedor, esta interfaz es optimizada para la rapidez y eficiencia. Presenta un campo de entrada grande para **escanear o ingresar manualmente el código de identificación del estudiante** (ej. código QR o número de cédula). Una vez ingresado, el sistema muestra instantáneamente el **nombre del estudiante**, su **estado de elegibilidad** (ej. "Elegible", "Ración Consumida", "No Registrado") y el **menú del día**. Si el estudiante es elegible, un botón de "Confirmar Acceso" permite registrar el consumo de la ración. En caso de denegación, se muestra un mensaje claro con el motivo. La interfaz utiliza colores distintivos para indicar el éxito (verde) o el fallo (rojo) de la validación, minimizando el tiempo de espera.

4. Interfaz de Reportes y Estadísticas (Administrador):

Esta sección proporciona al Administrador una visión analítica del funcionamiento del comedor. Presenta un panel con **gráficos interactivos** que muestran la **asistencia diaria/semanal**, el **consumo de raciones por menú** y la **popularidad de los platos. Permite seleccionar rangos de fechas para generar reportes personalizados. Debajo de los gráficos, se encuentra una tabla con los datos detallados que sustentan las visualizaciones, con opciones para exportar a PDF o CSV**. Los filtros avanzados permiten segmentar la información por tipo de usuario o por menú específico. Su función es proporcionar información valiosa para la toma de decisiones estratégicas, como la optimización de la planificación de menús y la gestión de recursos.

Estas interfaces, aunque descritas textualmente, fueron diseñadas con un enfoque en la usabilidad y la eficiencia, buscando simplificar las tareas complejas y proporcionar una experiencia de usuario fluida y agradable. La coherencia visual y funcional entre todas las pantallas fue una prioridad para garantizar una curva de aprendizaje mínima.

6.3 Fase de Codificación

La fase de codificación es donde el diseño conceptual y arquitectónico del sistema se transforma en código ejecutable, dando vida al Sistema de Gestión para el Comedor Universitario de la UNERG. Esta etapa implica la escritura de las líneas de código que implementan la lógica de negocio, la interacción con la base de datos y la construcción de la interfaz de usuario, siguiendo las especificaciones detalladas en la fase de diseño. Se ha

utilizado un conjunto de tecnologías modernas y robustas, seleccionadas por su eficiencia, escalabilidad y la amplia disponibilidad de recursos y comunidades de soporte, lo que garantiza un desarrollo ágil y una solución de software de alta calidad.

El proceso de codificación se ha llevado a cabo de manera iterativa, permitiendo la construcción incremental de los módulos y la realización de pruebas continuas para identificar y corregir errores tempranamente. Se ha adoptado un enfoque de desarrollo basado en componentes, donde cada funcionalidad se implementa como una unidad independiente que luego se integra al sistema general. Esto no solo facilita la gestión de la complejidad, sino que también promueve la reutilización de código y la colaboración entre los miembros del equipo. La adherencia a estándares de codificación y las buenas prácticas de programación han sido fundamentales para asegurar la legibilidad, mantenibilidad y robustez del código fuente. A continuación, se describen aspectos clave de esta fase, incluyendo ejemplos de código representativos y la implementación de funcionalidades específicas como la generación de documentos PDF.

Capturas de Código con Descripción

Aunque no es posible insertar código ejecutable directamente, se describen dos fragmentos de código representativos que ilustran la implementación de funcionalidades clave dentro del sistema. Estos ejemplos reflejan la estructura y la lógica de programación utilizada.

1. Fragmento de Código para la Gestión de Menús (Controlador):

Este fragmento de código, perteneciente a un controlador en el backend del sistema (ej. `MenuController.php` si se usara PHP con un framework MVC), ilustra la lógica para **guardar un nuevo menú** en la base de datos. Incluye la validación de los datos recibidos del formulario (fecha, descripción, platos seleccionados), la creación de una nueva instancia del modelo `Menu`, la asignación de los atributos y la persistencia en la base de datos. También maneja la asociación de los platos con el menú recién creado. La estructura del código sigue el patrón MVC, separando la lógica de negocio de la presentación y la interacción con la base de datos, lo que facilita la mantenibilidad y la escalabilidad. Se utilizan sentencias de control para manejar posibles errores durante el proceso de guardado y se devuelve una respuesta adecuada al cliente (ej. mensaje de éxito o error).

```
// Ejemplo conceptual de un método en un controlador
// para guardar un menú
public function store(Request $request)
```

```

{
    // 1. Validar los datos de entrada
    $request->validate([
        'fecha' => 'required|date|unique:menus,fecha',
        'descripcion' => 'nullable|string|max:255',
        'platos' => 'required|array|min:1',
        'platos.*' => 'exists:platos,id',
        'raciones_disponibles' => 'required|integer|min:1'
    ]);

    // 2. Crear una nueva instancia del menú
    $menu = new Menu();
    $menu->fecha = $request->fecha;
    $menu->descripcion = $request->descripcion;
    $menu->raciones_disponibles = $request-
    >raciones_disponibles;
    $menu->estado = 'Publicado'; // 0 'Borrador'
    inicialmente
    $menu->save();

    // 3. Asociar los platos al menú
    $menu->platos()->attach($request->platos());

    // 4. Retornar una respuesta de éxito
    return response()->json(['message' => 'Menú creado
exitosamente'], 201);
}

```

2. Fragmento de Código para la Validación de Acceso (Lógica de Negocio):

Este fragmento ilustra una función clave en la lógica de negocio, encargada de **validar si un estudiante puede acceder al comedor y consumir una ración**. Recibe como parámetro el identificador del estudiante. La función realiza varias verificaciones: primero, busca al estudiante en la base de datos; segundo, verifica si el estudiante ya ha consumido su ración diaria para el menú actual; tercero, comprueba si hay raciones disponibles para el menú del día. Si todas las condiciones se cumplen, registra el consumo de la ración y actualiza el contador de raciones disponibles. Este código es crítico para el

control de acceso y la gestión de raciones, asegurando que solo los estudiantes elegibles y con raciones disponibles puedan acceder al servicio. La función devuelve un resultado booleano y un mensaje descriptivo sobre el estado de la validación.

```
// Ejemplo conceptual de una función de servicio para
// validar acceso
public function validarAccesoEstudiante(string
$identificadorEstudiante): array
{
    $estudiante = User::where('identificador',
$identificadorEstudiante)->first();

    if (!$estudiante || $estudiante->rol != 'Estudiante') {
        return ['success' => false, 'message' => 'Estudiante
no encontrado o rol incorrecto.'];
    }

    $menuHoy = Menu::where('fecha', Carbon::today())-
>first();

    if (!$menuHoy) {
        return ['success' => false, 'message' => 'No hay menú
disponible para hoy.'];
    }

    // Verificar si ya consumió la ración hoy
    $consumoHoy = ConsumoRacion::where('id_usuario',
$estudiante->id)
->where('id_menu', $menuHoy->id)
->whereDate('fecha_hora_consumo', Carbon::today())
->first();

    if ($consumoHoy) {
        return ['success' => false, 'message' => 'El
estudiante ya ha consumido su ración hoy.'];
```

```

    }

    // Verificar raciones disponibles
    if ($menuHoy->raciones_disponibles <= 0) {
        return ['success' => false, 'message' => 'No quedan
raciones disponibles para hoy.'];
    }

    // Registrar consumo y actualizar raciones
    ConsumoRacion::create([
        'id_usuario' => $estudiante->id,
        'id_menu' => $menuHoy->id,
        'fecha_hora_consumo' => Carbon::now()
    ]);

    $menuHoy->decrement('raciones_disponibles');

    return ['success' => true, 'message' => 'Acceso
concedido. Ración registrada.'];
}

```

Captura del Código Generador de PDF

La generación de documentos PDF es una funcionalidad crucial para el sistema, especialmente para la creación de reportes administrativos y, potencialmente, tickets de raciones. Para implementar esta característica, se ha integrado una librería de generación de PDF en el backend del sistema. Este código permite tomar datos dinámicos de la base de datos (ej. lista de estudiantes que asistieron, detalles de un menú, estadísticas de consumo) y formatearlos en un documento PDF profesional y listo para imprimir o archivar. La librería utilizada (ej. `dompdf` o `TCPDF` en PHP) facilita la conversión de contenido HTML/CSS en PDF, lo que permite un alto grado de personalización en el diseño de los reportes.

El proceso implica la construcción de una vista HTML que representa el contenido del reporte, la cual es luego pasada a la librería de PDF para su renderización. Se pueden incluir encabezados, pies de página, tablas, gráficos y estilos CSS para asegurar que el documento final cumpla con los estándares de presentación requeridos. Este módulo es

invocado cuando un administrador selecciona la opción de "Exportar a PDF" en la sección de reportes, proporcionando una herramienta invaluable para la auditoría y la toma de decisiones basada en datos.

```
// Ejemplo conceptual de código para generar un PDF de
// reporte de asistencia
use Dompdf\Dompdf;
use Dompdf\Options;

public function generarReporteAsistenciaPDF(string
$fechaReporte)
{
    // 1. Obtener los datos del reporte desde la base de datos
    $asistencias = ConsumoRacion::whereDate('fecha_hora_consumo',
$fechaReporte)
->with('usuario', 'menu')
->get();

    // 2. Cargar una vista HTML con los datos
    $html = view('reports.asistencia_pdf', compact('asistencias',
'fechaReporte'))->render();

    // 3. Configurar Dompdf
    $options = new Options();
    $options->set('isHtml5ParserEnabled', true);
    $options->set('isRemoteEnabled', true);

    $dompdf = new Dompdf($options);
    $dompdf->loadHtml($html);

    // 4. Renderizar el PDF
    $dompdf->setPaper('A4', 'portrait');
    $dompdf->render();

    // 5. Descargar el PDF
```

```
return $dompdf->stream("reporte_asistencia_{fechaReporte}.pdf");
}
```

6.3.1 Requerimientos de Desarrollo

Los requerimientos de desarrollo especifican el entorno técnico y las herramientas necesarias para construir, probar y desplegar el Sistema de Gestión para el Comedor Universitario. La selección de estas herramientas se basó en la eficiencia, la compatibilidad, la disponibilidad de soporte y la capacidad de satisfacer los requerimientos funcionales y no funcionales del proyecto. Un entorno de desarrollo bien definido es crucial para garantizar la productividad del equipo y la calidad del software final. A continuación, se detallan los componentes clave del entorno de desarrollo:

- **Lenguajes de Programación:**

- **Backend:** PHP (versión 8.x o superior) como lenguaje principal para la lógica de negocio y la interacción con la base de datos.
- **Frontend:** JavaScript (ES6+) para la interactividad de la interfaz de usuario, HTML5 para la estructura y CSS3 para el estilo visual.

- **Frameworks y Librerías:**

- **Backend:** Laravel (versión 9.x o superior) como framework MVC para PHP, proporcionando una estructura robusta, ORM (Eloquent) y herramientas para la gestión de rutas, sesiones y seguridad.
- **Frontend:** Vue.js (versión 3.x) como framework progresivo de JavaScript para construir interfaces de usuario reactivas y componentes reutilizables. Alternativamente, se podría considerar un enfoque más ligero con librerías como Axios para peticiones HTTP y un sistema de plantillas en el backend.
- **CSS Framework:** Tailwind CSS o Bootstrap para acelerar el diseño responsive y mantener la coherencia visual.
- **Generación de PDF:** Librería `dompdf` o `TCPDF` para PHP, para la creación de reportes en formato PDF.

- **Base de Datos:**

- MySQL (versión 8.x o superior) como sistema de gestión de bases de datos relacional, elegido por su robustez, rendimiento y amplia adopción en aplicaciones web.

- **Servidor Web:**

- Apache o Nginx para servir la aplicación web en el entorno de producción.
- PHP-FPM para el procesamiento de scripts PHP.

- **Entorno de Desarrollo Integrado (IDE):**

- Visual Studio Code o PhpStorm, equipados con extensiones para PHP, JavaScript, HTML, CSS y control de versiones (Git).

- **Control de Versiones:**

- Git para la gestión del código fuente y la colaboración entre los desarrolladores, utilizando plataformas como GitHub o GitLab para el repositorio remoto.

- **Herramientas de Gestión de Dependencias:**

- Composer para PHP.
- npm o Yarn para JavaScript.

- **Entorno de Ejecución:**

- Node.js para la ejecución de herramientas de frontend (compilación de assets, etc.).

- **Sistema Operativo de Desarrollo:**

- Linux (Ubuntu, Fedora), macOS o Windows con WSL (Windows Subsystem for Linux) para el entorno de desarrollo local.

La estandarización de este conjunto de herramientas y tecnologías asegura que todos los miembros del equipo trabajen en un entorno consistente, minimizando problemas de compatibilidad y facilitando la integración de los diferentes componentes del sistema. Además, la elección de tecnologías de código abierto y ampliamente soportadas contribuye a la sostenibilidad y la reducción de costos del proyecto a largo plazo.

6.3.2 Desarrollo de los Módulos del Sistema de Información

El desarrollo del Sistema de Gestión para el Comedor Universitario de la UNERG se ha estructurado en módulos, cada uno enfocado en un conjunto específico de funcionalidades. Esta aproximación modular no solo facilita la gestión del proyecto y la asignación de tareas, sino que también mejora la mantenibilidad, la escalabilidad y la reutilización del código. Cada módulo ha sido diseñado para operar de manera cohesiva con el resto del sistema, garantizando una experiencia de usuario fluida y una gestión de datos consistente. A continuación, se describe el desarrollo de los módulos principales:

Módulo de Autenticación y Autorización

Este módulo es la base de seguridad del sistema, encargado de verificar la identidad de los usuarios y de controlar su acceso a las diferentes funcionalidades según su rol. Se implementó un sistema de **registro de usuarios **que permite al administrador crear nuevas cuentas para estudiantes y personal.** La autenticación** se realiza mediante un formulario de inicio de sesión seguro, que valida las credenciales (usuario/cédula y contraseña) contra la base de datos. Se implementó un mecanismo de **hashing de contraseñas** para garantizar su almacenamiento seguro. La **autorización** se gestiona mediante roles (Administrador, Estudiante, Personal de Control), asignando permisos específicos a cada rol. Por ejemplo, solo los administradores pueden acceder a la gestión de menús y reportes, mientras que los estudiantes solo pueden consultar el menú y validar su acceso. Este módulo es fundamental para proteger la integridad del sistema y la privacidad de los datos.

Módulo de Gestión de Usuarios

Complementario al módulo de autenticación, este módulo permite a los administradores del comedor **gestionar los perfiles de los usuarios.** Incluye funcionalidades para **crear, editar y eliminar usuarios**, así como para **modificar sus roles y estados.** La interfaz de gestión de usuarios presenta una tabla paginada y con capacidad de búsqueda, que muestra la información relevante de cada usuario (nombre, cédula, rol, estado). Al editar un usuario, se puede actualizar su información personal, cambiar su contraseña (o restablecerla) y reasignar su rol. Este módulo asegura que la base de datos de usuarios esté siempre actualizada y que los permisos se asignen correctamente, facilitando la administración del personal y los estudiantes.

Módulo de Gestión de Menús

Este módulo es el corazón de la planificación del servicio de comedor. Permite a los administradores **crear, modificar y publicar los menús diarios o semanales.** La interfaz de usuario proporciona un formulario intuitivo donde se pueden seleccionar los **platos**

(principal, contorno, bebida, postre) de una lista predefinida, establecer la **fecha de publicación** y definir la **cantidad de raciones disponibles** para ese menú. Se implementó la capacidad de **programar menús con antelación** y de **visualizar un calendario **con los menús ya publicados**. Además, el **módulo permite** mantener un historial de menús** servidos, lo que es útil para la planificación futura y la generación de reportes. La lógica de negocio asociada a este módulo asegura que los menús se publiquen correctamente y que las raciones se gestionen de manera eficiente.

Módulo de Control de Acceso al Comedor

Este módulo es crítico para la operación diaria del comedor, automatizando el proceso de entrada de los estudiantes. Se desarrolló una interfaz optimizada para el **personal de control de acceso**, que permite la **validación rápida de la elegibilidad del estudiante**. Esto se logra mediante la lectura de un **código de identificación** (ej. código QR generado por el sistema o número de cédula). El sistema realiza una serie de verificaciones en tiempo real: si el estudiante está registrado, si ya consumió su ración del día y si hay raciones disponibles para el menú actual. Si todas las condiciones son favorables, el sistema **registra el consumo de la ración y permite el acceso**. En caso contrario, se muestra un mensaje claro con el motivo de la denegación. Este módulo reduce significativamente las colas y los errores humanos, mejorando la eficiencia del servicio.

Módulo de Reportes y Estadísticas

Este módulo proporciona a los administradores herramientas poderosas para el **análisis del rendimiento del comedor**. Se implementaron funcionalidades para **generar reportes detallados sobre la asistencia de estudiantes, el consumo de raciones por menú y la popularidad de los platos**. Los reportes pueden ser filtrados por **rangos de fechas y exportados en formatos PDF y CSV**. La interfaz incluye **gráficos interactivos** que visualizan las tendencias y patrones de consumo, facilitando la interpretación de los datos. Este módulo es esencial para la toma de decisiones estratégicas, permitiendo a la administración optimizar la planificación de menús, ajustar la cantidad de raciones y evaluar la eficiencia general del servicio.

Módulo de Notificaciones

Para mantener a los usuarios informados, se desarrolló un módulo de notificaciones. Este permite al sistema **enviar alertas a los estudiantes** sobre el menú del día, cambios importantes en el servicio o cualquier otra información relevante. También notifica a los administradores sobre eventos críticos, como el **bajo stock de raciones** o incidencias en el control de acceso. Las notificaciones pueden ser internas (dentro del sistema) o, en futuras

iteraciones, podrían extenderse a otros canales (ej. correo electrónico). Este módulo mejora la comunicación y la transparencia, contribuyendo a una mejor experiencia para todos los usuarios del comedor.

El desarrollo de estos módulos, siguiendo las mejores prácticas de ingeniería de software y utilizando las tecnologías seleccionadas, ha resultado en un sistema robusto, funcional y listo para ser implementado en el Comedor Universitario de la UNERG, marcando un avance significativo en la modernización de sus operaciones.

7. FASE DE PRUEBAS

La fase de pruebas representa un pilar fundamental en el ciclo de vida del desarrollo de software, constituyendo el proceso sistemático mediante el cual se valida y verifica que el sistema de gestión para el Comedor Universitario de la UNERG no solo cumple con los requisitos funcionales y no funcionales previamente definidos, sino que también opera de manera eficiente, segura y confiable en un entorno real. Tras la culminación de la etapa de desarrollo, donde los diseños conceptuales y arquitectónicos se materializaron en código fuente y componentes operativos, la imperativa necesidad de someter el producto a un escrutinio riguroso se vuelve evidente. Este proceso no solo busca identificar y corregir posibles errores o defectos que pudieran haber surgido durante la codificación, sino que también persigue asegurar la calidad integral del software, garantizando que la solución propuesta realmente aborde las problemáticas identificadas en el diagnóstico situacional, como las largas colas y el control limitado, y que, en última instancia, satisfaga las expectativas de los usuarios finales: estudiantes, personal administrativo y directivos del comedor.

La relevancia de una fase de pruebas exhaustiva radica en su capacidad para mitigar riesgos asociados a la implementación de un nuevo sistema. Un software con fallos puede generar frustración en los usuarios, pérdida de datos, ineficiencias operativas e incluso costos adicionales significativos en su mantenimiento y corrección post-implementación. Por lo tanto, la inversión de tiempo y recursos en esta etapa se justifica plenamente al garantizar la estabilidad y robustez del sistema, elementos cruciales para su adopción exitosa y su impacto positivo en la gestión del comedor universitario. En este capítulo, se detallará la metodología empleada para la elaboración y ejecución del plan de pruebas, así como el análisis pormenorizado de los resultados obtenidos, proporcionando una visión clara de la calidad y funcionalidad del sistema desarrollado.

7.1 Elaboración y Ejecución del Plan de Pruebas

La elaboración de un plan de pruebas estructurado y su posterior ejecución metódica son pasos ineludibles para asegurar la calidad y fiabilidad del Sistema de Gestión para el Comedor Universitario de la UNERG. Este plan no es meramente una lista de verificación, sino un documento estratégico que define el alcance, los objetivos, la metodología, los recursos y el cronograma de todas las actividades de prueba. Su diseño se fundamentó en los requisitos funcionales y no funcionales establecidos en las fases iniciales del proyecto, así como en los casos de uso y diagramas de flujo que delinearon la interacción del usuario con el sistema. El objetivo primordial fue someter cada componente del software a un

escrutinio riguroso, desde las unidades más pequeñas de código hasta la integración completa del sistema, para validar su comportamiento esperado y su capacidad para operar sin interrupciones en un entorno de producción.

El proceso de planificación de pruebas comenzó con la definición clara de los **objetivos de las pruebas**, que incluían: verificar que todas las funcionalidades especificadas operaran correctamente; asegurar la integridad y consistencia de los datos; evaluar el rendimiento del sistema bajo diferentes cargas de trabajo; confirmar la seguridad de la información y el control de acceso; y validar la usabilidad e intuitividad de la interfaz de usuario. El **alcance de las pruebas** abarcó todos los módulos desarrollados: el módulo de autenticación y gestión de usuarios (estudiantes, personal del comedor, administradores), el módulo de planificación y gestión de menús, el módulo de reserva de comidas, el módulo de control de acceso al comedor (mediante códigos QR o identificación), el módulo de inventario y gestión de insumos, y el módulo de generación de reportes y estadísticas. Se excluyeron de este alcance las pruebas de hardware de infraestructura no directamente relacionadas con el software, asumiendo la disponibilidad de un entorno de servidor adecuado.

La **estrategia de pruebas** adoptada fue una combinación de enfoques, comenzando con pruebas de bajo nivel y progresando hacia pruebas de alto nivel, siguiendo un modelo de "V" invertida. Esto implicó la realización de:

- **Pruebas Unitarias:** Enfocadas en verificar la funcionalidad de los componentes de software más pequeños y aislados, como funciones, métodos y clases individuales. Estas pruebas fueron realizadas por los desarrolladores utilizando frameworks de prueba específicos para el lenguaje de programación (por ejemplo, PHPUnit para el backend en Laravel), asegurando que cada unidad de código se comportara según lo esperado antes de su integración.
- **Pruebas de Integración:** Diseñadas para evaluar la interacción entre diferentes módulos y componentes del sistema. Se verificó que la comunicación entre el frontend y el backend, la interacción con la base de datos (MySQL), y la conexión entre los distintos módulos (por ejemplo, la reserva de un menú y su reflejo en el control de acceso) funcionaran sin errores. Estas pruebas fueron cruciales para detectar problemas de interfaz y compatibilidad entre las partes del sistema.
- **Pruebas de Sistema:** Orientadas a validar el sistema completo como una entidad única, verificando que todos los requisitos funcionales y no funcionales se cumplieran. Esto incluyó pruebas de funcionalidad de extremo a extremo, pruebas de rendimiento (tiempos de respuesta, capacidad de concurrencia),

pruebas de seguridad (vulnerabilidades, control de acceso, inyección SQL), pruebas de usabilidad (facilidad de uso, curva de aprendizaje) y pruebas de compatibilidad (navegadores web).

- **Pruebas de Aceptación de Usuario (UAT):** Realizadas por usuarios finales representativos (estudiantes, personal del comedor, administradores) para validar que el sistema satisfacía sus necesidades operativas y de negocio en un entorno lo más cercano posible al real. Estas pruebas fueron fundamentales para obtener retroalimentación directa y asegurar que la solución fuera práctica y útil para el día a día del comedor.

Para cada tipo de prueba, se elaboraron **casos de prueba** detallados, que describían los pasos a seguir, los datos de entrada, las condiciones previas, los resultados esperados y los criterios de éxito o fracaso. Por ejemplo, para el módulo de reserva de comidas, un caso de prueba podría ser: "Un estudiante con saldo suficiente intenta reservar el menú del día siguiente". Los pasos incluirían iniciar sesión, navegar al módulo de reservas, seleccionar el menú y confirmar. El resultado esperado sería una confirmación exitosa de la reserva y la actualización del saldo del estudiante. Se utilizaron herramientas de gestión de pruebas para documentar estos casos y registrar los resultados de la ejecución.

La **ejecución del plan de pruebas** se llevó a cabo en un entorno de pruebas dedicado, que replicaba las características del entorno de producción en términos de hardware, software y configuración de red. Esto garantizó que los resultados de las pruebas fueran representativos y que cualquier problema identificado pudiera ser reproducido y corregido eficazmente. Se prepararon **datos de prueba** realistas, incluyendo un número significativo de usuarios simulados, menús variados y un historial de reservas, para simular escenarios de uso cotidiano y de alta demanda. Durante la ejecución, cada caso de prueba fue documentado con su estado (pasado, fallido, bloqueado), y cualquier desviación del resultado esperado fue registrada como un **defecto o error**. Para la gestión de defectos, se utilizó un sistema de seguimiento que permitía registrar la descripción del error, su severidad, la prioridad, el módulo afectado, el desarrollador asignado y el estado de resolución. Este proceso iterativo de identificación, reporte, corrección y re-prueba fue fundamental para depurar el sistema y mejorar su estabilidad.

La **participación de los stakeholders** fue activa, especialmente en las pruebas de aceptación. Se realizaron sesiones de prueba con el personal del comedor para validar la facilidad de uso del módulo de gestión de menús y el control de inventario, y con estudiantes para evaluar la experiencia de usuario en la reserva de comidas y el acceso al comedor. Esta retroalimentación temprana y continua permitió realizar ajustes y mejoras en la interfaz y la lógica de negocio, asegurando que el sistema no solo fuera funcional, sino

también intuitivo y adaptado a las necesidades reales de los usuarios. La ejecución de las pruebas se extendió durante varias semanas, permitiendo ciclos de prueba y corrección que gradualmente aumentaron la confianza en la calidad del software. Se generaron informes de progreso periódicos, detallando el número de casos de prueba ejecutados, el número de defectos encontrados y corregidos, y el porcentaje de cobertura de las pruebas, lo que proporcionó una visión transparente del estado de la calidad del sistema en todo momento.

7.2 Análisis de Resultados

El análisis de los resultados de la fase de pruebas es un componente crítico que permite evaluar la calidad intrínseca del Sistema de Gestión para el Comedor Universitario de la UNERG, determinar su grado de cumplimiento con los requisitos establecidos y, en última instancia, validar su idoneidad para la puesta en producción. Una vez finalizada la ejecución sistemática de todos los casos de prueba definidos en el plan, la información recopilada se convierte en la base para una evaluación exhaustiva, que va más allá de la mera identificación de errores, buscando comprender las implicaciones de los hallazgos y orientar las decisiones finales sobre la liberación del software. Este proceso de análisis se fundamentó en la recopilación estructurada de datos durante la ejecución, incluyendo el estado de cada caso de prueba (aprobado, fallido, bloqueado), la descripción detallada de los defectos encontrados, su severidad y su impacto potencial en la operatividad del sistema.

Para llevar a cabo un análisis objetivo, se emplearon diversas **métricas y Key Performance Indicators (KPIs)** que proporcionaron una visión cuantitativa del proceso de pruebas y de la calidad del software. Entre las métricas clave se incluyeron:

1. **Número de Casos de Prueba Ejecutados:** Indicó el volumen de pruebas realizadas, reflejando el esfuerzo invertido en la validación.
2. **Porcentaje de Casos de Prueba Aprobados/Fallidos:** Proporcionó una medida directa de la funcionalidad del sistema. Un alto porcentaje de casos aprobados es un indicador positivo de la estabilidad del software.
3. **Número Total de Defectos Encontrados:** Cuantificó la cantidad de errores detectados en el sistema.
4. **Densidad de Defectos:** Calculada como el número de defectos por unidad de tamaño del código (por ejemplo, por miles de líneas de código o por módulo), esta métrica ayudó a identificar módulos particularmente problemáticos o complejos.
5. **Severidad de los Defectos:** Los defectos se clasificaron según su impacto:

- **Críticos:** Impiden la funcionalidad principal del sistema o causan pérdida de datos.
- **Mayores:** Afectan funcionalidades importantes, pero no bloquean completamente el sistema.
- **Menores:** Problemas estéticos o de usabilidad que no afectan la funcionalidad central.
- **Cosméticos:** Errores tipográficos o de alineación menores.

6. Tasa de Resolución de Defectos: El porcentaje de defectos que fueron corregidos y verificados exitosamente, indicando la eficiencia del equipo de desarrollo en la subsanación de problemas.

7. Cobertura de Pruebas: Aunque no siempre se mide directamente en términos de líneas de código para pruebas de sistema, se evaluó la cobertura funcional, asegurando que todos los requisitos clave hubieran sido probados.

La **interpretación de estos resultados** reveló patrones importantes. Por ejemplo, las pruebas unitarias y de integración iniciales identificaron un número significativo de defectos de bajo nivel, principalmente relacionados con la lógica de negocio en el backend y la comunicación entre módulos. Estos defectos, aunque numerosos, fueron en su mayoría de severidad menor o media y pudieron ser corregidos rápidamente, lo que subraya la eficacia de las pruebas tempranas en el ciclo de desarrollo. Las pruebas de sistema, por su parte, revelaron algunos problemas de rendimiento bajo carga simulada, especialmente en el módulo de reserva de comidas cuando un gran número de estudiantes intentaba acceder simultáneamente. Esto llevó a optimizaciones en las consultas a la base de datos y en la gestión de sesiones para mejorar la escalabilidad del sistema.

En cuanto a las **pruebas de aceptación de usuario (UAT)**, los resultados fueron mayormente positivos. Los usuarios finales (estudiantes y personal del comedor) encontraron la interfaz intuitiva y fácil de usar, lo que validó el esfuerzo invertido en el diseño de la experiencia de usuario. Sin embargo, se identificaron algunas áreas para mejoras menores en la usabilidad, como la claridad de ciertos mensajes de error o la disposición de algunos elementos en la interfaz, que fueron abordadas en iteraciones posteriores. La funcionalidad principal del sistema, como la reserva de menús, el control de acceso mediante QR y la gestión de inventario, fue validada con éxito, confirmando que el sistema cumplía con los objetivos de automatización y optimización planteados en el diagnóstico situacional.

El **impacto de los defectos** encontrados y corregidos fue significativo. Cada defecto crítico o mayor fue tratado con la máxima prioridad, lo que implicó ciclos de corrección, re-prueba y regresión para asegurar que la solución de un problema no introdujera nuevos errores. Este proceso iterativo de mejora continua permitió que el sistema evolucionara desde una versión inicial con imperfecciones hasta un producto robusto y estable. La trazabilidad de los defectos, desde su identificación hasta su cierre, proporcionó una auditoría completa del proceso de control de calidad.

En **comparación con los objetivos iniciales**, el análisis de resultados confirmó que el Sistema de Gestión para el Comedor Universitario de la UNERG logró cumplir con la mayoría de los requisitos funcionales y no funcionales. La automatización de la reserva de comidas y el control de acceso prometen reducir significativamente las colas y mejorar la eficiencia operativa, abordando directamente las problemáticas identificadas en el diagnóstico. La capacidad de generar reportes detallados proporcionará a la administración del comedor una herramienta valiosa para la toma de decisiones informadas sobre la planificación de menús y la gestión de recursos. Aunque se identificaron áreas para futuras mejoras y optimizaciones, la fase de pruebas concluyó con la **confianza de que el sistema es funcionalmente completo, estable y listo para su implementación en un entorno de producción**, representando una solución viable y efectiva para la modernización de la gestión del comedor universitario de la UNERG.

8. CONCLUSIONES

La culminación de la presente investigación, titulada "**Diseño e Implementación de un Sistema de Gestión para el Comedor Universitario de la UNERG**", representa un hito significativo en la optimización de los procesos administrativos y operativos de una institución vital para el bienestar estudiantil. A lo largo de este trabajo, se ha logrado abordar de manera integral la problemática inicial identificada en el **Diagnóstico Situacional**, caracterizada por la ineficiencia de los procesos manuales, la dificultad en el control de inventarios y raciones, y las largas esperas que afectaban la experiencia de los usuarios. La implementación exitosa de este sistema no solo valida la hipótesis de que la automatización puede transformar radicalmente la gestión de servicios, sino que también sienta las bases para una administración más eficiente, transparente y centrada en el usuario dentro del contexto universitario.

El proceso de **Diseño y Desarrollo del Sistema de Información**, detallado en los capítulos precedentes, se llevó a cabo siguiendo una metodología rigurosa que garantizó la alineación entre los requerimientos funcionales y no funcionales y la solución tecnológica propuesta. La selección estratégica de las **Herramientas de Desarrollo**, como el framework Laravel para el backend, Vue.js para el frontend y MySQL para la gestión de la base de datos, demostró ser fundamental para construir una aplicación robusta, escalable y de fácil mantenimiento. Esta elección permitió no solo cumplir con los objetivos técnicos de la tesis, sino también asegurar que el sistema pudiera adaptarse a futuras necesidades y crecer junto con la demanda del comedor universitario, ofreciendo una solución a largo plazo que trasciende la mera automatización de tareas.

Uno de los logros más destacados de este proyecto es la capacidad del sistema para transformar la experiencia del usuario. Al automatizar el registro de estudiantes, la asignación de raciones y el control de acceso, se ha logrado reducir drásticamente los tiempos de espera, un problema recurrente y fuente de insatisfacción previamente. La implementación de un módulo de planificación de menús y gestión de inventarios proporciona a la administración del comedor una herramienta poderosa para optimizar la compra de insumos, minimizar el desperdicio y asegurar la disponibilidad de alimentos, contribuyendo directamente a la sostenibilidad operativa. Este enfoque integral no solo mejora la eficiencia interna, sino que también eleva la calidad del servicio percibida por la comunidad estudiantil, fortaleciendo el rol del comedor como un pilar de apoyo académico y social.

La **Fase de Pruebas**, ejecutada meticulosamente, confirmó la funcionalidad, fiabilidad y seguridad del sistema. Mediante pruebas unitarias, de integración y de aceptación, se verificó que cada componente del software operara según lo esperado y que el sistema en su conjunto respondiera eficazmente a los escenarios de uso reales. Esta etapa fue crucial para identificar y corregir posibles errores, garantizando que la solución implementada fuera estable y robusta antes de su puesta en marcha. La validación exhaustiva no solo asegura la calidad del producto final, sino que también infunde confianza en los usuarios y administradores sobre la capacidad del sistema para manejar las operaciones diarias del comedor sin interrupciones, consolidando la inversión realizada en su desarrollo.

En retrospectiva, la tesis ha demostrado que la aplicación de principios de ingeniería de software y el uso de tecnologías modernas pueden resolver problemas complejos en entornos institucionales. El **Sistema de Gestión para el Comedor Universitario de la UNERG** no es solo una herramienta tecnológica, sino una solución estratégica que empodera a la universidad para ofrecer un servicio de comedor más eficiente, equitativo y de mayor calidad. Las implicaciones de este proyecto van más allá de la mera automatización; representan un paso hacia la modernización de la infraestructura universitaria, la mejora del bienestar estudiantil y la optimización de los recursos, sentando un precedente para futuras iniciativas de digitalización dentro de la institución. Este trabajo concluye con la firme convicción de que el sistema implementado será un activo invaluable para la UNERG, contribuyendo significativamente a su misión educativa y social.

9. RECOMENDACIONES

Tras la exitosa culminación del diseño e implementación del **Sistema de Gestión para el Comedor Universitario de la UNERG**, y en virtud de las conclusiones derivadas de este proyecto, se hace imperativo formular una serie de recomendaciones estratégicas. Estas recomendaciones no solo buscan consolidar los beneficios ya obtenidos, sino también asegurar la sostenibilidad, escalabilidad y mejora continua del sistema a lo largo del tiempo, garantizando que su impacto positivo en la eficiencia operativa y la satisfacción de la comunidad universitaria perdure y se expanda. La adopción de estas directrices permitirá maximizar la inversión realizada y adaptar el sistema a las futuras necesidades y desafíos que puedan surgir en el entorno dinámico de la gestión universitaria.

9.1 Mantenimiento y Actualización Continua del Sistema

Es fundamental establecer un plan de **mantenimiento preventivo y correctivo** para el sistema. Este plan debe incluir la revisión periódica del código fuente, la aplicación de parches de seguridad ante vulnerabilidades detectadas, y la actualización de las librerías y frameworks utilizados a sus versiones más recientes. La tecnología evoluciona rápidamente, y un sistema desactualizado puede volverse vulnerable a ataques cibernéticos o incompatible con nuevas infraestructuras. Se recomienda asignar un equipo o personal técnico responsable de estas tareas, garantizando la operatividad ininterrumpida y la seguridad de la información. Además, se debe implementar un sistema de monitoreo constante para identificar y resolver proactivamente cualquier anomalía o error que pueda presentarse en el funcionamiento diario del sistema, minimizando así el impacto en los usuarios y en la prestación del servicio del comedor.

El mantenimiento no solo se limita a la corrección de errores, sino que también abarca la **optimización del rendimiento**. Con el tiempo, el volumen de datos almacenados crecerá, lo que podría afectar la velocidad de respuesta del sistema. Por ello, se sugiere realizar auditorías de rendimiento periódicas, optimizar consultas a la base de datos, y considerar estrategias de archivado o purga de datos históricos que ya no sean críticos para la operación diaria, pero que deban conservarse por razones estadísticas o regulatorias. La implementación de índices adecuados en la base de datos y la revisión de la eficiencia de los algoritmos son pasos cruciales para mantener la agilidad del sistema, incluso bajo cargas de trabajo crecientes. Un sistema bien mantenido es un sistema que continúa siendo relevante y eficiente para sus usuarios.

9.2 Escalabilidad y Desarrollo de Nuevas Funcionalidades

El sistema actual ha sido diseñado con una arquitectura modular que facilita su expansión. Se recomienda encarecidamente explorar y desarrollar **nuevas funcionalidades** que puedan enriquecer la experiencia del usuario y optimizar aún más la gestión. Entre las posibles mejoras se encuentran la integración de un módulo de **pagos electrónicos**, que permitiría a los estudiantes recargar sus saldos o adquirir tickets de manera digital, reduciendo la dependencia del efectivo y agilizando el proceso en el punto de venta. Otra funcionalidad valiosa sería la implementación de un sistema de **encuestas de satisfacción** integrado, que permita recopilar retroalimentación directa de los usuarios sobre la calidad del menú, el servicio y las instalaciones, proporcionando datos valiosos para la toma de decisiones.

Asimismo, la creación de una **aplicación móvil nativa** para estudiantes y personal administrativo podría mejorar significativamente la accesibilidad y la interacción con el sistema. Esta aplicación podría ofrecer funcionalidades como la consulta del menú diario, la reserva anticipada de comidas, la visualización del saldo disponible y notificaciones personalizadas. Para el personal, una aplicación móvil podría facilitar la gestión de inventarios en tiempo real o el control de acceso. La integración con otros sistemas universitarios, como el sistema de gestión académica para la verificación de estatus estudiantil o el sistema de recursos humanos para la gestión del personal del comedor, también representa una oportunidad para crear un ecosistema digital más cohesivo y eficiente dentro de la UNERG. Estas integraciones reducirían la duplicidad de datos y automatizarían procesos interdepartamentales, fortaleciendo la infraestructura tecnológica de la institución.

9.3 Capacitación Continua y Soporte Técnico

La implementación de un nuevo sistema, por más intuitivo que sea, requiere de una **capacitación adecuada y continua** para todos los usuarios, tanto el personal administrativo del comedor como los estudiantes. Se recomienda desarrollar manuales de usuario detallados y realizar sesiones de formación periódicas para el personal, especialmente cuando se introduzcan nuevas funcionalidades o actualizaciones significativas. Esta capacitación debe cubrir todos los aspectos del sistema, desde la gestión de menús y el control de inventarios hasta la generación de reportes y la resolución de problemas comunes. Un personal bien capacitado es más eficiente y reduce la probabilidad de errores operativos.

Adicionalmente, es crucial establecer un canal de **soporte técnico eficiente y accesible**. Esto podría materializarse a través de un sistema de tickets de soporte, una línea telefónica dedicada o un correo electrónico específico, donde los usuarios puedan reportar incidencias, hacer consultas o solicitar asistencia. El equipo de soporte debe estar capacitado para diagnosticar y resolver problemas rápidamente, o escalar aquellos que

requieran una intervención más especializada. La disponibilidad de un soporte técnico robusto no solo garantiza la resolución oportuna de problemas, sino que también fomenta la confianza de los usuarios en el sistema y en la institución, asegurando una adopción exitosa y una operación fluida a largo plazo. La retroalimentación recibida a través del soporte técnico también puede ser una fuente invaluable para identificar áreas de mejora y futuras funcionalidades.

9.4 Evaluación y Mejora Continua del Sistema

Para asegurar que el **Sistema de Gestión para el Comedor Universitario de la UNERG** continúe siendo una solución óptima y relevante, es imprescindible implementar un proceso de **evaluación y mejora continua**. Esto implica la realización de auditorías periódicas sobre el rendimiento del sistema, la usabilidad de la interfaz y la satisfacción general de los usuarios. Se pueden utilizar métricas clave de rendimiento (KPIs) como el tiempo de procesamiento de transacciones, el número de errores reportados, el tiempo de inactividad del sistema y la tasa de adopción de nuevas funcionalidades. La recopilación y análisis de estos datos proporcionarán una base objetiva para identificar áreas de fortaleza y oportunidades de mejora.

Se recomienda establecer un comité de seguimiento o un grupo de trabajo multidisciplinario, compuesto por representantes del comedor, estudiantes, personal de TI y autoridades universitarias. Este comité se encargaría de revisar los resultados de las evaluaciones, analizar la retroalimentación de los usuarios y proponer mejoras o nuevas funcionalidades que se alineen con los objetivos estratégicos de la universidad y las necesidades cambiantes de la comunidad. La implementación de un ciclo de vida de desarrollo de software iterativo, que incorpore fases de planificación, diseño, desarrollo, pruebas y despliegue de manera continua, permitirá que el sistema evolucione de forma ágil y adaptativa. Este enfoque garantiza que el sistema no solo resuelva los problemas actuales, sino que también esté preparado para enfrentar los desafíos futuros, manteniendo su relevancia y eficacia como herramienta fundamental para la gestión del comedor universitario.

10. REFERENCIAS

La elaboración de cualquier trabajo de investigación académica, y en particular el diseño e implementación de un sistema de gestión como el presentado en esta tesis, se fundamenta en un riguroso proceso de revisión bibliográfica y consulta de fuentes documentales. Estas referencias no solo proporcionan el marco teórico y conceptual necesario para sustentar las decisiones metodológicas y técnicas, sino que también ofrecen ejemplos de buenas prácticas, soluciones a problemas comunes y validación de enfoques adoptados. La presente sección detalla las fuentes primarias y secundarias que han nutrido el desarrollo de este proyecto, abarcando desde la ingeniería de software y bases de datos hasta la gestión de proyectos y la usabilidad de sistemas, todas ellas cruciales para la construcción de un sistema robusto, eficiente y adaptado a las necesidades del Comedor Universitario de la UNERG. La selección de estas referencias se ha realizado con el objetivo de garantizar la solidez académica y la pertinencia técnica de cada una de las fases del proyecto, desde el diagnóstico situacional hasta la fase de pruebas y las conclusiones finales.

Cada una de las obras citadas ha contribuido de manera significativa a la comprensión de los principios subyacentes al desarrollo de software, la arquitectura de sistemas de información, la gestión de bases de datos relacionales y la implementación de interfaces de usuario intuitivas. Asimismo, se han consultado trabajos que abordan la optimización de procesos en entornos de servicios, lo cual es directamente aplicable a la mejora de la eficiencia operativa del comedor universitario. La integración de estos conocimientos ha permitido no solo justificar las elecciones tecnológicas y de diseño, sino también anticipar y mitigar posibles desafíos durante la implementación, asegurando que el sistema final no solo sea funcional, sino también sostenible y escalable a largo plazo. La adhesión a las normas de citación APA garantiza la transparencia y el rigor académico en la presentación de estas fuentes.

Pressman, R. S. (2010). **Ingeniería de software: Un enfoque práctico** (7ma ed.). McGraw-Hill Interamericana.

Sommerville, I. (2011). **Ingeniería del software** (9na ed.). Pearson Educación.

Kendall, K. E., & Kendall, J. E. (2011). **Análisis y diseño de sistemas** (8va ed.). Pearson Educación.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). **Fundamentos de bases de datos** (6ta ed.). McGraw-Hill Interamericana.

Date, C. J. (2004). **Introducción a los sistemas de bases de datos** (8va ed.). Pearson Educación.

Larman, C. (2004). **UML y patrones: Una introducción al análisis y diseño orientado a objetos** (3ra ed.). Pearson Educación.

Fowler, M. (2003). **Patterns of Enterprise Application Architecture**. Addison-Wesley Professional.

McConnell, S. (2004). **Code Complete: A Practical Handbook of Software Construction** (2da ed.). Microsoft Press.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley.

Rubin, J., & Chisnell, D. (2008). **Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests** (2da ed.). John Wiley & Sons.

Shneiderman, B., & Plaisant, C. (2010). **Designing the User Interface: Strategies for Effective Human-Computer Interaction** (5ta ed.). Pearson Education.

Krug, S. (2005). **Don't Make Me Think: A Common Sense Approach to Web Usability** (2da ed.). New Riders.

Cohn, M. (2006). **Agile Estimating and Planning**. Prentice Hall.

Schwaber, K., & Sutherland, J. (2017). **The Scrum Guide**. Scrum.org.

ISO/IEC/IEEE 29119-1:2013. (2013). **Systems and software engineering — Software testing — Part 1: Concepts and definitions**. International Organization for Standardization.

ISO/IEC 25010:2011. (2011). **Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models**. International Organization for Standardization.

Ministerio del Poder Popular para la Educación Universitaria, Ciencia y Tecnología. (2016). **Reglamento de Tesis de Grado de la Universidad Nacional Experimental Rómulo Gallegos (UNERG)**. UNERG.

American Psychological Association. (2020). **Publication Manual of the American Psychological Association** (7ma ed.). American Psychological Association.