



UNIVERSIDAD DE CIENFUEGOS
INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA

Universidad

de Cienfuegos

“Carlos Rafael Rodríguez”

Facultad de Ingeniería

**Sistema de gestión del proceso de reservas del
comedor y la planificación del menú en la
Universidad de Cienfuegos.**

**Trabajo de diploma para optar por el título de
Ingeniería en Informática**

Autores:

Roxana Mesa García

Samuel Orlando Abrantes Capote

Tutores:

Dr. Eduardo Concepción

MsC. Oscar Luis Muñoz

Cienfuegos, Cuba

Curso: 2023

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad Cienfuegos los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

(Autor)

(Autor)

(Tutor)

(Cotutor)

El verdadero progreso es el que pone la tecnología al alcance de todos.

Henry Ford

Agradecimientos

Roxana Mesa García

No podemos pasar por alto a las personas que de una forma u otra contribuyeron a la realización de un sueño:

- ✓ *Un agradecimiento especial a mis padres por ir de la mano conmigo, por ayudarme en todos los momentos difíciles y por no permitir que me rindiera, gracias por ser mis guías.*
- ✓ *Un agradecimiento eterno a mi hijo Roxley que fue mi motor impulsor y a mi compañero de vida que me ha apoyado en todo siempre.*
- ✓ *A mi toda mi familia, abuelos, primos y tíos que han confiado en mi infinitamente.*
- ✓ *A mis amigos que son como familia por el apoyo y la confianza.*
- ✓ *A mis tutores por su valiosa asesoría, por el apoyo y la ayuda en este proceso.*
- ✓ *A mis profesores que fueron mi guía en la carrera, y en especial a al profe Richard Darián por estar siempre al pendiente de nosotros desde que comenzó hasta el último día.*
- ✓ *A todos mis compañeros de clases por el apoyo incondicional durante la carrera.*

A todos

¡Muchas Gracias!

Samuel Orlando Abrahantes Capote

- ✓ *En primer lugar, agradecer a mis padres por ser los promotores de mis sueños, gracias a ellos por cada día confiar y creer en mis expectativas, gracias a mi madre por siempre anhelar lo mejor para mi vida y a mi padre por cada consejo y cada una de sus palabras que me guiaron durante esta trayectoria.*
- ✓ *Quiero agradecer a mi hermana que aún a la distancia ha sido como una madre y un motivo de superación en mi vida.*
- ✓ *Gracias a mis abuelas que a lo largo de mi existencia me han aportado la sabiduría suficiente para llegar a ser lo que hoy soy y han sido como un faro en todo momento. Gracias a mis abuelos que me miran desde la mayor de las alturas y sé que hoy donde quiera que estén festejan orgullosos de mí.*
- ✓ *También quiero agradecer a mi esposa por permanecer junto a mí durante esta etapa de mi vida con tanto cariño y comprensión.*
- ✓ *Gracias a mis tíos y primos que estuvieron presentes en todo momento y al pendiente de mí, gracias por su confianza.*
- ✓ *Gracias a mi compañera de tesis con la cuál fue un placer infinito realizar este proyecto y poder contar con su apoyo y cooperación. Gracias a todos mis compañeros de clases.*
- ✓ *Gracias a mis tutores y profesores que fueron la base de mis conocimientos sin los cuales esta tesis no hubiera sido posible.*
- ✓ *Gracias a la vida por este triunfo, por permitirme cerrar una puerta de mi vida que marcará el comienzo de una nueva etapa. Gracias verdaderamente a todas las personas que me apoyaron y creyeron en la realización de esta tesis. Y finalmente gracias a Dios.*



Dedicatoria

A mis padres y a mi hijo que son mi mayor inspiración.

A mi abuelo Carmelo.

Roxana Mesa García.

*A mis padres y mi hermana que son las personas más
importantes en mi vida.*

Samuel Orlando Abrahantes Capote.

Resumen

El Comedor de la Universidad de Cienfuegos realiza de forma manual todo el proceso de reservación de alimentos tanto para los estudiantes becarios como los trabajadores autorizados a recibir este servicio. De igual forma se realiza la planificación de recursos necesarios para elaborar los alimentos, así como la recaudación diaria. Es por ello que surge la necesidad de elaborar un sistema informático que realice la gestión de las reservas, la gestión de los menús diarios, la planificación de recursos y finalmente el importe a recaudar.

Entre los principales beneficios que reporta, se tiene que las reservas del servicio se podrán hacer on-line, conociendo con anterioridad las ofertas disponibles. Para la administración del comedor también resultó muy importante ya que le permite realizar la planificación de recursos logrando un proceso más eficiente.

En el trabajo se muestra el estudio y análisis de sistemas similares pero que por sus características no pueden ser aplicados en la Universidad. Se muestra la selección de las herramientas, tecnologías y procesos de desarrollo, acordando utilizar como lenguaje de programación PHP, HTML, CSS, JavaScript, como gestor de base de datos MySQL y como metodología de desarrollo AUP en el escenario 4.

Como resultado fundamental se logró el desarrollo de la aplicación Web que cuenta con los módulos para gestionar usuarios, reservas, la confección de menú, planificación, ofertas, la generación de pedidos y finalizando con la recaudación. Se garantiza que cada usuario solo tenga acceso a la información acorde al cargo mediante la asignación de roles y permisos en cada funcionalidad.

Palabras clave: Menú, Planificación, Reservas.

Abstract

The dining room of the University of Cienfuegos manually carries out the entire food reservation process for both scholarship students and workers authorized to receive this service. In the same way, the planning of resources necessary to prepare the food is carried out, as well as daily collection. This is why the need arises to develop a computer system that manages reservations, manages daily menus, resource planning and finally the amount to be collected.

Among the main benefits it provides, it is that reservations for the service can be made online, knowing in advance the available offers. For the administration of the dining room, it was also very important since it allows it to carry out resource planning, achieving a more efficient process.

The work shows the study and analysis of similar systems but that due to their characteristics cannot be applied at the University. The selection of tools, technologies and development processes is shown, agreeing to use PHP, HTML, CSS, JavaScript as the programming language, MySQL as the database manager and AUP as the development methodology in scenario 4.

As a fundamental result, the development of the Web application was achieved, which has modules to manage users, reservations, menu preparation, planning, offers, generating orders and ending with collection. It is guaranteed that each user only has access to the information according to the position by assigning roles and permissions in each functionality.

Keywords: Menu, Planning, Reservations.

Índice

Introducción	2
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
Introducción:	7
1.1 Conceptos asociados al dominio del problema:	7
1.2 Análisis de los sistemas informáticos referidos al campo de acción:.....	7
1.2.1 Sistemas Internacionales:.....	8
1.2.2 Sistemas Nacionales:	8
1.3 Metodología de desarrollo de software.....	9
1.4 Arquitectura de software	12
1.4.1 Estilos arquitectónicos	14
1.5 Tecnologías y herramientas para el desarrollo de la solución	14
1.6 Conclusiones parciales	17
Capítulo 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN:.....	18
2.1 Análisis del proceso de gestión de reservas y planificación.....	18
2.2 Requisitos de software	21
2.2.1 Requisitos Funcionales.....	22
2.2.2 Requisitos no funcionales	26
2.3 Historias de usuarios	27
2.4 Descripción de la arquitectura	30
2.5 Diagrama de clases diseño	31
2.6 Modelo de datos	32
2.7 Estándar de codificación empleado	33
2.8 Conclusiones parciales	33
CAPÍTULO 3: VALORACIÓN DE LA VIABILIDAD DE LA PROPUESTA DE SOLUCIÓN	34
3.1 Técnicas de validación de requisitos	34
3.2 Métrica aplicada a los requisitos.....	34

3.3 Pruebas de software.....	35
3.3.1 Pruebas de Caja Blanca	37
3.3.2 Pruebas de Caja Negra	41
3.4 Validación de la solución propuesta.....	44
3.5 Conclusiones parciales	47
Conclusiones generales	48
Recomendaciones.	49
Referencias	50



Índice de Tablas

<i>Tabla 1. Descripción de los métodos teóricos y empíricos.</i>	5
<i>Tabla 2. Comparación entre las metodologías AUP y RUP.</i>	12
<i>Tabla 3. Descripción de los requisitos funcionales del sistema.</i>	26
<i>Tabla 4. Descripción de los requisitos no funcionales del sistema.</i>	27
<i>Tabla 5. Historia de usuario del requisito funcional "Agregar oferta".</i>	29
<i>Tabla 6. Métodos de pruebas aplicada a la solución.</i>	37
<i>Tabla 7. Categoría de la complejidad ciclomática.</i>	40
<i>Tabla 8. Caso de prueba del camino básico No. 1</i>	40
<i>Tabla 9. Caso de prueba del camino básico No. 2</i>	41
<i>Tabla 10. Prueba de caja negra al requisito funcional Registrar usuario.</i>	43
<i>Tabla 11. Prueba de caja negra al requisito funcional Iniciar sesión.</i>	44

Índice de Figuras

<i>Figura 1. Descripción de la propuesta de solución.</i>	<i>20</i>
<i>Figura 2. Arquitectura Modelo – Vista - Controlador.</i>	<i>30</i>
<i>Figura 3. Aplicación de la arquitectura a la solución propuesta.</i>	<i>31</i>
<i>Figura 4. Diagrama de clases del diseño.</i>	<i>32</i>
<i>Figura 5. Modelo de datos de la propuesta de solución.</i>	<i>32</i>
<i>Figura 6. Funcionalidad login (acceso al sistema).</i>	<i>38</i>
<i>Figura 7. Grafo de flujo del código de la función login (acceso al sistema).</i>	<i>39</i>
<i>Figura 8. Interfaz del estudiante en el sistema.</i>	<i>45</i>
<i>Figura 9. Interfaz del administrador en el sistema.</i>	<i>45</i>
<i>Figura 10. Interfaz de los productos existentes en el sistema.</i>	<i>46</i>
<i>Figura 11. Interfaz de los productos disponibles en el sistema.</i>	<i>46</i>

Introducción

Uno de los rasgos notables en el siglo XXI fue el desarrollo acelerado de las Tecnologías de Información y Comunicaciones (TIC), las cuales han tenido un impacto significativo en el mundo. Las TIC son herramientas que aportan a la sociedad del conocimiento y brindan al desarrollo nuevas formas de organizarse, comunicarse, educar, enseñar y aprender; con ello la transformación y evolución de la sociedad.

En Cuba el acceso a las TIC ha sido limitado en comparación con otros países de la región. En la década de 1970 se identifica en Cuba la necesidad de la participación eficaz y consecuente de las Instituciones de la Educación Superior (IES) en el desarrollo de las tecnologías, lo que se materializa con la creación de carreras afines a la Computación y la Electrónica. Desde entonces el número y diversidad de carreras con respecto a esta rama ha tenido un aumento significativo con la misión de formar profesionales comprometidos y altamente calificados en el área de las TIC. [1]

La Universidad de Cienfuegos, ha estado trabajando en los últimos años para incorporar las TIC en su plan de estudios y en la gestión universitaria en general. La Universidad ha desarrollado programas de capacitación para el personal docente y administrativo en el uso de las TIC, y ha establecido una plataforma virtual de aprendizaje que permite a los estudiantes acceder a materiales educativos y participar en actividades en línea. En cuanto a la investigación, la Universidad ha establecido un centro de investigación en tecnologías de la información y comunicación, que se encarga de investigar y desarrollar soluciones tecnológicas innovadoras para problemas específicos.

La Universidad de Cienfuegos “Carlos Rafael Rodríguez” es una institución que implementa las normativas y políticas del Ministerio de Educación Superior en el ámbito territorial. Se conforma por una estructura organizativa, ágil y eficaz; sustentada en la ciencia, la tecnología, la innovación, la calidad, la sostenibilidad y la racionalidad económica con una concepción humanista y comprometida con el desarrollo de la sociedad.

Actualmente cuenta con siete facultades, subdivididas en departamentos docentes y centros de estudios, para la organización de los procesos de formación de profesionales, postgrados, ciencia, innovación y extensión, con un alto compromiso con su encargo social y vínculo con la comunidad.

Posee una Residencia Estudiantil para estudiantes de los municipios u otras provincias del país. Es por ello que ofrece servicios de alimentación para sus estudiantes becados, así como profesores y otros trabajadores. Aunque se ofrecen menús diferenciados, el comedor es común pero dividido en dos secciones independientes.

Los profesores pueden ver las ofertas del menú el día antes, cuando van a comprar su tique diario; en ese instante deciden lo que desean reservar. Es una inconformidad el hecho de que deben hacer una cola para almorzar y otra para comprar los tiques, aspecto que les hace perder mucho tiempo. Además, en ocasiones no se dispone del efectivo para recibir devoluciones sobre el importe pagado, ocasionándoles pérdidas monetarias, que, aunque parecen insignificantes, al sumar una semana o mes, produce una afectación a su economía personal.

Para el caso de los estudiantes, su almuerzo es gratuito y el menú es más dirigido, lo conocen en el momento en que efectúan la alimentación. Ellos no realizan su reserva, se realiza auxiliándose de la encargada de la residencia estudiantil que informa de manera diaria al administrador del comedor, la asistencia de los becarios.

Debido a la situación que vive el país, y por ende la Universidad, la planificación del menú es diaria. Lo realiza el administrador apoyándose en las normas de alimentación que rigen en la universidad, en el parte de la contadora que informa la cantidad de tiques vendidos y la responsable de la residencia estudiantil con la cantidad de estudiantes presentes y la información recibida del almacenero con la disponibilidad de productos en el almacén del Comedor. Con estas informaciones, de forma manual, conforma la propuesta del pedido del día siguiente.

Esta información es entregada de forma manuscrita al departamento de Aseguramiento Técnico Material (ATM) para que actualice sus inventarios y entregue al almacenero el pedido del menú a elaborar con sus correspondientes cantidades.

Teniendo en cuenta estas condiciones se define como **problema a resolver**: La necesidad de gestionar la información referente al servicio de alimentación y planificación del menú en el comedor de la Universidad de Cienfuegos de forma rápida, confiable y disponible a todos los interesados.

Identificándose como el **objeto de estudio**: el proceso de servicio de alimentación y planificación del menú en el comedor de la Universidad y como **campo de acción**: la gestión de la información referente al servicio de alimentación y planificación en el comedor de la Universidad de Cienfuegos.

Dar la posibilidad de publicar el menú del día con anterioridad, controlar las reservas diarias, elaborar la planificación de las cantidades reales a utilizar, resulta vital en la toma de decisiones del personal implicado en esta actividad.

Es importante y necesario, que exista un flujo donde la información sea confiable, íntegra, segura y esté disponible en todo momento. El uso de las Tecnologías de la Información (TIC) garantizaría una posible solución a los problemas planteados.

Se define como **idea a defender** del presente trabajo: La elaboración de una aplicación para gestionar, planificar y controlar la información referente al servicio de alimentación y planificación en el comedor de la Universidad de Cienfuegos, permitirá agilizar el proceso de reservas de alimentación, planificación del menú y control de las recaudaciones diarias, garantizando un manejo seguro de los datos.

Para dar solución al problema planteado anteriormente, se define el siguiente **objetivo general**: Elaborar una aplicación web para la gestión de la información referente al servicio de alimentación y planificación del menú en el comedor de la Universidad de Cienfuegos.

Planteado el objetivo general anteriormente se despliegan los siguientes **objetivos específicos**:

- ✓ Analizar los procesos que se llevan a cabo en la gestión del servicio de alimentación y planificación en el comedor de la Universidad de Cienfuegos.
- ✓ Identificar las funcionalidades que debe implementarse para cada área implicada.
- ✓ Diseñar la aplicación “Sistema de Servicio de Alimentación”
- ✓ Implementar la aplicación en ambiente web que facilite la gestión de los procesos del comedor.
- ✓ Validar el sistema propuesto.

Para alcanzar estos objetivos se precisa la realización de diferentes tareas dentro de la investigación como son:

- ✓ Entrevista con los trabajadores del comedor, de contabilidad y las personas que reciben el servicio de alimentación para identificar las particularidades del proceso y definir responsabilidades.
- ✓ Análisis de las particularidades del servicio de alimentación en el comedor. Caracterización del proceso de reserva, publicación y configuración del menú y la gestión de los pagos por el servicio.
- ✓ Estudio de aplicaciones desarrolladas en la Universidad u otras entidades relacionadas con el problema a resolver.
- ✓ Diseño de una arquitectura que responda a las características del trabajo a implementar.
- ✓ Selección de las herramientas, metodologías, lenguajes y tecnologías para la implementación del sitio Web.
- ✓ Diseño de la interfaz gráfica de la aplicación.
- ✓ Implementación del diseño propuesto para la aplicación y los requerimientos funcionales y no funcionales.
- ✓ Diseño de casos de pruebas funcionales.
- ✓ Documentación de la información generada durante las diferentes fases de la implementación del proyecto.

Con la implementación de este sistema que se propone se alcanza el siguiente **aporte práctico**: una nueva herramienta que permite gestionar los servicios de alimentación y planificación en el comedor de la Universidad de Cienfuegos, se logrará que tanto estudiantes como trabajadores conozcan con anterioridad el menú diario, que puedan realizar ellos mismos sus reservas. Se facilitará el proceso de planificación del menú y la recaudación de las reservas. Al informatizarse el proceso se reducen los tiempos de la planificación y se controlan los inventarios de recursos. La aplicación web a desarrollar garantiza que toda la información en él gestionada sea rápida, confiable, segura y esté disponible a todos los clientes del sistema.

El **método científico de la investigación** es una metodología para obtener nuevos conocimientos; un método de investigación debe basarse en lo empírico y la medición, y estar sujeto a los principios específicos de las pruebas de razonamiento. A continuación, se muestran mediante una tabla los métodos utilizados que son teórico y empírico para el desarrollo de esta investigación:

Nivel del método	Clasificación	Importancia para la investigación
Teóricos	Analítico - Sintético	Este método se utiliza para analizar y comprender los procesos de reservación del servicio de alimentos y planificación del menú en la Universidad de Cienfuegos para así establecer conclusiones sobre cómo informatizar dichos procesos.
	Histórico – Lógico	Fue necesario analizar el progreso del problema, la existencia de metodología y sistemas informáticos similares al que se desea elaborar.
Empíricos	Observación	Posibilitó identificar las carencias más significativas de este proceso y quiénes intervienen.
	Entrevista	Permitió obtener información, evaluar las características del proceso y llegar a la conclusión de la necesidad del sistema.

Tabla 1. Descripción de los métodos teóricos y empíricos.

El presente documento de investigación está estructurando en 3 capítulos tal y como se describe a continuación:

Capítulo 1 Fundamentos teórico-metodológicos: Se expone los elementos teóricos referente al proceso de reservas y planificación del menú en el comedor de la Universidad de Cienfuegos, que se usa de apoyo a la investigación del problema. En este capítulo se analizarán los sistemas informáticos ya existentes tanto internacionales como nacionales relacionados con la investigación. Se mencionan la metodología, arquitectura de software, estilos y herramientas a utilizar en el desarrollo de la solución.

Capítulo 2 Descripción de la propuesta de solución: Se describe la propuesta de solución. Se muestra artefactos correspondientes al escenario número 4 de la metodología AUP en las fases de análisis, diseño e implementación.

Capítulo 3 Valoración de la viabilidad de la propuesta de solución: Se efectúa la viabilidad de la propuesta de solución mediante la validación de los requisitos y el diseño propuesto, asimismo de la realización de pruebas funcionales, unitarias y de aprobación al sistema informático.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción:

En este capítulo se exponen los elementos teóricos y conceptuales referente al proceso de gestión de servicios de alimentación en comedores tanto universitarios como de empresas, aspecto que sirve de base a la investigación del problema planteado. Entre los temas que se abordan, se explican los conceptos relacionados con el objeto de estudio, se realiza una descripción de sistemas de gestión de alimentación existentes en el entorno, así como el flujo y el análisis crítico de los procesos o software que sirven de antecedentes y que son similares a la temática plateada. Se realiza además una explicación de las metodologías de desarrollo de software, de las herramientas y tecnologías potenciales a utilizar en el desarrollo de la propuesta de solución.

1.1 Conceptos asociados al dominio del problema:

Para una mejor comprensión de la investigación se precisa la conceptualización de algunas terminologías utilizadas en el dominio del problema:

Reservación del servicio de alimentación: es un proceso donde el comensal mediante la reservación de un evento asegura la prestación del servicio de alimentación en los respectivos complejos comedores. Se hace manualmente o por medio de sistemas de reservación.

Planificación: se entiende básicamente como un proceso que implicará la observación de una serie de pasos que se establecerán a priori y para los cuales, quienes realizan la planificación, deberán utilizar una serie de herramientas y expresiones (BALLOU, 1991).

Sistemas de reservación online son un medio para acceder a los servicios de una empresa o institución, con un tiempo determinado de antelación a que este sea brindado. Las reservaciones reducen el riesgo de no recibir el servicio por un cliente y disminuye el tiempo de espera para tener acceso al mismo. Estas brindan la posibilidad de conocer de antemano la demanda y se pueda llevar a cabo una preparación previa para dar respuesta a sus necesidades.

1.2 Análisis de los sistemas informáticos referidos al campo de acción:

El uso de las TIC en el mundo se ha desarrollado relevantemente en los últimos años, ejemplo de ello es el desempeño y uso de sistemas informáticos para la gestión del proceso de reservas y planificación de menú como los que se especifica a continuación:

1.2.1 Sistemas Internacionales:

La planificación y el control de los servicios que se prestan en los comedores al que acceden grandes cantidades de comensales con características o condiciones que los diferencia entre sí, se hace cada vez más necesaria en las entidades con estas particularidades. En el mundo ya existen empresas que se dedican al desarrollo de aplicaciones que automatizan y optimizan estos procesos, pero casi todas son aplicaciones por las cuales es necesario pagar por su adquisición, licencia de utilización, mantenimientos, etc. Muchas de estas aplicaciones se implementan para elaborar menús saludables, variados y óptimos para la salud humana, pero no incorporan la planificación de recursos, control de reservas, pagos, etc.

Algunos ejemplos son:

- ✓ **Iara Comedores** está diseñado especialmente para empresas que gestionan comedores escolares, tanto en el ámbito público como privado. Está adaptado a las especificaciones de las distintas comunidades autónomas españolas. Iara Comedores gestiona con solvencia todos los procesos, ayudando a optimizar los servicios, tanto a nivel funcional como económico. Desarrollado con el conocimiento recogido desde hace 15 años y gracias al talento de profesionales altamente cualificados, Iara Comedores se ha convertido en uno de los Software que gestiona más centros escolares en España. [2]
- ✓ **SGI – MOD.COMEDOR:** Es un sistema desarrollado para gestionar la operación del servicio de comedor y el suministro de alimentos especiales (dietas) para servicios hospitalarios. [3]
- ✓ **SATN2007**, de Cadre Distribución S.L que está enfocado fundamentalmente al área de la salud. Este software cuenta con una comprensiva base de información y conocimiento de los alimentos, proporciona análisis completos de dietas, recetas y menús. [4]

Estas aplicaciones informáticas no pueden ser utilizadas en nuestra universidad debido a los costos que implica su explotación y configuración, no cubren además todos los procesos de planificación y control del servicio de reservas, gestión de comensales y recursos y el pago o gratuidades por los servicios recibidos.

1.2.2 Sistemas Nacionales:

En Cuba algunas instituciones se han proyectado hacia el desarrollo de software que contribuya a lograr una buena planificación y control de los servicios de comedores. Algunos ejemplos son:

- ✓ En el Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), por ejemplo, se utiliza el Sistema de Control de Comedores, SISCAMED. Este usa un sistema de tarjetas magnéticas, en el cual los usuarios registrados deben presentarla antes de ser atendidos. Además, permite obtener la cantidad de comensales registrados que reciben cada servicio de manera diaria o mensualmente, en los comedores incluyendo los comensales registrados que reciben doble ración en el mismo evento (desayuno, almuerzo, comida) del día. [4]

Este sistema no pudo ser usado en nuestra universidad debido a que: existen grandes diferencias entre ambas universidades, ejemplo, el número de comensales es muy elevado. El mecanismo de utilización de las tarjetas magnéticas ya no hace viable su uso.

- ✓ La Empresa de Telecomunicaciones en el Territorio también cuenta con un sistema informático: Reporte de Alimentación desarrollado por sus especialistas pero que imposibilita su uso ya que maneja varios comedores al mismo tiempo y además el precio de los menús es referido a escalas fijas según la categorización de los trabajadores.

Luego de realizar el estudio de varios sistemas similares se puede afirmar que los mismos no dan una solución directa al problema de la investigación pues no integran las funcionalidades necesarias para adaptarse a las necesidades específicas presentadas.

1.3 Metodología de desarrollo de software

Las metodologías de desarrollo de software son enfoques sistemáticos y estructurados que se utilizan para planificar, diseñar, construir, probar y entregar software de alta calidad. Son un conjunto de prácticas, técnicas y procesos que se utilizan para guiar el desarrollo de software y mejorar la eficiencia, la calidad y la productividad de equipo de desarrollo.

Hay muchas metodologías de desarrollo de software diferentes, cada una con sus propias fortalezas y debilidades.

Las metodologías tradicionales se rigen bajo una estructura secuencial inalterable basada en etapas. Dichas etapas son: el análisis de requerimientos, diseño, programación, prueba y mantenimiento. Se les da gran importancia a los requerimientos del proyecto. Dado que se precisa de un amplio estudio para su definición, y una vez establecidos, no pueden alterarse de ninguna manera.

Un ejemplo de dichas metodologías es Proceso Unificado de Desarrollo (Rational Unified Process, RUP). Este es uno de los procesos más generales que existe, está enfocado a cualquier tipo de proyecto así no sea de software, se basa en la documentación generada en cada una de sus fases. El Proceso Unificado de Desarrollo Software (RUP) se caracteriza por estar dirigido por casos de uso,

centrado en la arquitectura y ser iterativo e incremental, es un marco de desarrollo compuesto de cuatro partes que se mencionan a continuación:

- ✓ Inicio
- ✓ Elaboración
- ✓ Construcción
- ✓ Transición

Cada una de estas fases está dividida en una serie de iteraciones que ofrecen como resultado un incremento del producto desarrollado, que añade o mejora las funcionalidades del sistema en desarrollo. En estas iteraciones se realizan las actividades definidas en el ciclo de vida clásico. Aunque todas ellas suelen incluir trabajo en la mayoría de estas actividades, el grado de esfuerzo dentro de cada una varía a lo largo del proyecto y una actividad se puede realizar completamente en más de una iteración en dependencia de la complejidad del proyecto. RUP tiene definido cuatro elementos: los roles, que responden al ¿Quién?, los artefactos, que responden al ¿Cómo?, las actividades que responden al ¿Qué? y los flujos de trabajo, que responden al ¿Cuándo?

RUP tiene definidos los siguientes flujos de trabajo:

- ✓ Modelado del negocio
- ✓ Requerimientos.
- ✓ Análisis y Diseño.
- ✓ Implementación.
- ✓ Pruebas.
- ✓ Despliegue.
- ✓ Gestión del proyecto.
- ✓ Configuración y control de cambios.
- ✓ Entorno.

RUP utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como lenguaje de notación. [5]

Por otra parte, las metodologías ágiles se definen por su versatilidad debido a lo sencillo que les resulta el adaptarse a los cambios. Sin alterar las características originales del proyecto en el proceso. Lográndose un control mucho más dinámico del mismo, amén de una mejor respuesta a las necesidades del cliente.

Teniendo en cuenta las características de equipo, la participación del cliente en la solución y el carácter cambiante de los requisitos, se decide utilizar una metodología ágil, de ellas, se opta por utilizar Proceso Unificado Ágil (AUP).

Proceso Unificado Ágil o Agile Unified Process (AUP).

Es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP. [6]

En la siguiente tabla se refleja una breve comparación de ventajas y desventajas de ambas metodologías, la cual fue de mucha importancia para la elección de la misma en la presente investigación:

	AUP (Metodología ágil)	RUP (Metodología pesada)
V E N	Permite al equipo dividir el proyecto en etapas y así centrarse en cada una de forma individual. Esto permite trabajar más rápido.	Es una forma disciplinada de asignar tareas y responsabilidades, pues los roles están muy bien definidos, de esta manera es bastante útil para definir roles en los proyectos.
	Permiten adaptar el proyecto a medida que avanza. Así, ante cualquier cambio que surja, es muy sencillo volver a organizar el equipo en relación con los nuevos objetivos.	Este proceso es de los más utilizados para el desarrollo del software por la mayoría de las empresas, pues su enfoque es bastante óptimo y tiende a ser una metodología viable para la mayoría de estas.

T A J A S	La estrecha interacción entre el equipo y el cliente garantiza un progreso constante, la mejora de la calidad, la capacidad de realizar cambios de manera efectiva y la opción de presentar nuevas ideas sobre el producto que se está desarrollando.	Ofrece a cada usuario, un filtrado personalizado de la definición del proceso publicado, acorde con su rol dentro del proyecto.
D E S V E N T A J A S	Al inicio del proyecto, es difícil determinar con precisión la cantidad de tiempo y dinero que se necesitará para completarlo, debido a los requisitos en constante cambio.	Por el grado de complejidad puede ser no muy adecuado. Debido a que es un proceso bastante grande y complejo es muy común que no sea el adecuado para cualquier proyecto pequeño
	El equipo necesita tener una base sólida y habilidades.	Método pesado. En muchos aspectos tiende a ser muy pesado, pues como se explica la complejidad es alta.
	La falta de atención a la documentación puede dificultar que los nuevos miembros del equipo accedan a la misma. [7]	Al ser una metodología bastante cara y con bastantes requerimientos en cuanto a roles (personales), a veces los costos son muy elevados, dando como resultado una imposibilidad por costear el proyecto. [8]

Tabla 2. Comparación entre las metodologías AUP y RUP.

Teniendo en cuenta la clasificación de las metodologías expuestas con anterioridad, las características del sistema y el corto tiempo para su implementación, se decide utilizar como guía para el proceso de desarrollo la metodología ágil AUP en el escenario No.4. Además, el cliente siempre va a interactuar con el equipo de desarrollo para perfeccionar los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

1.4 Arquitectura de software

La arquitectura, referida al software, es un concepto que surge ya en los años 60 y se refiere a una planificación basada en modelos, patrones y abstracciones teóricas, a la hora de realizar una pieza de software de cierta complejidad y como paso previo a cualquier implementación. De esta forma se dispone de una guía teórica detallada que nos permite entender cómo van a encajar cada una de las

piezas de nuestro producto o servicio. Por tanto, en arquitectura llamamos patrón a cualquier solución general y reutilizable para problemas recurrentes en ingeniería del software en un contexto dado, son similares a los patrones usados en la programación, pero orientados específicamente a la estructura a un nivel superior y más genérico. [9]

Una arquitectura de software se selecciona y diseña con base en objetivos (requisitos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como el mantenimiento, la auditoría, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real. [10]

En ocasiones, y cuando tenemos estos hechos bien planteados y razonados, elegir un patrón de arquitectura también puede ser una cuestión de familiaridad, comodidad o simple preferencia, por eso es aconsejable probarlos, para intentar también familiarizarse con ellos y con el diferente flujo de trabajo que proponen. Ejemplos de estas arquitecturas más universales y menos conocidas cabe destacar:

- ✓ Cliente-servidor: es muy usado sobre todo en el diseño de webs y servicios online, y se basa en el concepto de la existencia de un servidor (que proporciona el servicio) y una serie de clientes, que piden al servidor y reciben una respuesta del mismo.
- ✓ Patrón de capas: se subdivide la estructura del programa en un número de capas que representan una subtask, cada una perteneciendo a un nivel de abstracción diferente. Cada capa está diseñada para proporcionar un servicio a la siguiente capa de mayor nivel. Generalmente se utilizan las capas de aplicación, dominio y persistencia.
- ✓ Patrón master – slave: consiste en dos grupos, el primero es llamado el maestro (master) y el otro el grupo de esclavos (slaves). Los esclavos realizan la tarea propuesta por el maestro, computan los resultados y los envían de nuevo a este, quien los presenta, almacena o procesa. Esto se realiza así para tener una parte que autoriza y dirige los cálculos necesarios y otras partes que lo procesan de manera agnóstica a estas decisiones. [9]

Otras arquitecturas menos conocidas son:

- ✓ En pizarra.
- ✓ Entre pares.
- ✓ En pipeline.
- ✓ Modelo Vista Controlador.

1.4.1 Estilos arquitectónicos

Los estilos y patrones arquitectónicos son una forma clara de plasmar la solución de un problema mediante el uso de arquitectura de software, se usa también para comunicar un diseño arquitectónico a las etapas posteriores del desarrollo de software.

En un estilo de la arquitectura de software se localizan tres tipos de elementos: de procesamiento (que transforman los datos), de datos (información a procesar) y de conexión (como llamadas a procedimientos almacenados, mensajes, etc.) además se enfatiza sobre las restricciones de estos elementos y sus relaciones posibles. La principal característica de un estilo arquitectónico es la comunicación de diseños. [11]

Se pueden encontrar diversos tipos de estilos arquitectónicos entre los que se pueden citar:

- ✓ Arquitectura centrada en datos: Se enfoca en el almacenamiento de datos al que tienen acceso otros componentes.
- ✓ Arquitectura de flujo de datos: Se utiliza cuando hay transformaciones en secuencia sobre ciertos datos.
- ✓ Arquitecturas de llamada y retorno: Se basa en una comunicación de llamadas y retornos entre módulos.
- ✓ Arquitecturas orientadas a objetos: Los componentes de un sistema son datos y operaciones encapsulados y se comunican a través del paso de mensajes.
- ✓ Arquitecturas en capas: Se tienen varias capas, cada capa se comunica con las capas inferiores o superiores a través del paso de información y servicios. [12]

1.5 Tecnologías y herramientas para el desarrollo de la solución

Para un desarrollo web se utilizan programas de computación para asegurar que el servidor, la aplicación y la base de datos se desempeñen de manera conjunta y eficiente. También son necesarias una serie de tecnologías y herramientas para dar estructura, forma e interactividad a los elementos que conformarán el aspecto visual del sitio. El diseño y desarrollo de sitios web y aplicaciones online conlleva la elección y el conocimiento de las tecnologías que se decidan emplear, con el objetivo de favorecer su implementación y garantizar un mantenimiento del sitio o proyecto. [13] Las herramientas de desarrollo web permiten a los programadores crear software y depurar o probar su código antes de publicar sus proyectos en Internet. Estas herramientas no ayudan a crear una aplicación o un sitio web.

Sin embargo, permiten a los desarrolladores verificar dos veces la seguridad y la apariencia de la aplicación o el sitio, asegurándose de que funcione correctamente y se vea increíble. [14]

✓ **PHP (versión 8.2):**

En inglés Hypertext Preprocessor, es un lenguaje de programación de uso libre y gratuito que funciona como vínculo entre el sitio web, previamente desarrollado con otro lenguaje, y un determinado servidor de datos. Con él, es posible recopilar datos de formularios, modificar bases de datos, administrar archivos dentro de servidores, entre otras. Es la tecnología backend con la que se han implementado más servidores en Internet. PHP se utiliza generalmente para diseñar páginas web dinámicas, es decir, una página cuyo contenido no sea siempre el mismo. Muchas de los sitios web más populares en el mundo utilizan PHP para impulsarlas. [13]

✓ **HTML5:**

HyperText Markup Language por sus siglas en inglés, es una de las tecnologías indispensables para cualquier desarrollador y es el componente más básico de la web. Consiste en un conjunto de códigos cortos, que se clasifican como archivos de textos en las etiquetas, lo que le da estructura y significado al sitio web. Igualmente proporciona información de párrafos, imágenes, tablas, entre otras, para describir el contenido de una página web. HTML es muy básico y necesita de otras tecnologías para dotar de estética y mayor funcionalidad al sitio. [13]

✓ **CSS:**

Cascading Style Sheets en inglés, es un lenguaje de diseño web creado para trabajar en conjunto con lenguajes como HTML. CSS sirve para estructurar, de forma visual, el contenido generado por las etiquetas del HTML a través de estilos, colores, tipografías, backgrounds y efectos sencillos. Todo esto con el fin de lograr una interfaz de usuario agradable y fácil de entender. Es una de las tecnologías web más utilizadas por los desarrolladores al momento de crear sus proyectos. [13]

✓ **JavaScript:**

Este conocido lenguaje de programación es uno de las tecnologías web más utilizadas ya que provee de mayor interactividad y dotan de dinamismo a los sitios a través de animaciones y elementos web. también permite desarrollar aplicaciones tan elaboradas como las redes sociales. Normalmente se utiliza como complemento de HTML y CSS. De esto, podemos decir

que HTML es la estructura y CSS es el adorno de una página web, mientras que JavaScript es el dinamismo, la interacción. [13]

✓ **Bootstrap 4:**

Bootstrap es un framework de desarrollo front-end, gratuito y de código abierto desarrollado específicamente para ayudar a crear sitios y aplicaciones web. Como herramienta de desarrollo web, Bootstrap agiliza el proceso de diseño de sitios estelares con capacidad de respuesta para dispositivos móviles al proporcionar una selección de diseños de plantillas. Bootstrap usa su código único para hacer que todas las imágenes respondan según el tamaño de la pantalla del dispositivo. [14]

✓ **Visual Studio Code:**

Es uno de los favoritos de todos los tiempos para muchos desarrolladores, principalmente porque esta herramienta es muy flexible y puede proporcionar a los desarrolladores las herramientas para un ciclo simplificado de compilación y depuración de código. Este editor de código puede manejar labores que van desde la depuración hasta la ejecución de tareas e incluso el control de versiones. Todos los archivos con Visual Studio Code utilizan una estructura jerárquica clara. Cada archivo de código se encuentra en carpetas específicas relacionadas con un proyecto o tema asociado, lo que facilita y agiliza la búsqueda de cualquier cosa en cualquier momento. [14]

✓ **XAMPP:**

Es una herramienta de desarrollo que te permite probar tu desarrollo web basado en PHP en tu propio ordenador sin necesidad de tener acceso a internet. XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web. [15]

✓ **MySQL (versión3.3):**

Es un sistema de administración de bases de datos relacionales. Es un software de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo. MySQL es uno de los sistemas más popularizados para almacenar y

administrar datos. Con administrar nos referimos a las acciones CRUD: Create (crear), Read (leer), Update (actualizar) y Delete (borrar). [16]

✓ **Visual Paradigm:**

Es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. [17]

- ✓ **Axure RP 10:** Es un software para crear prototipos y especificaciones para sitios web y aplicaciones. Ofrece colocación de arrastrar y soltar.

1.6 Conclusiones parciales

En este capítulo se estudiaron algunos de los sistemas existentes relacionados con la gestión de la información referente al servicio de alimentación y se definieron los principales conceptos asociados al objeto de estudio. Se analizaron los lenguajes, tecnologías, herramientas y metodologías necesarias para el desarrollo del sistema, concluyendo utilizar AUP como metodología de desarrollo del software. En síntesis, el primer capítulo sienta las bases teóricas y metodológicas necesarias para el desarrollo del proyecto propuesto.

Capítulo 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

Introducción.

En el presente capítulo se plantea y se describe la propuesta de solución y los mecanismos generados por el escenario número 4 en la metodología Agile Iniciad Procesos (AUP) en las etapas de análisis, diseño e implementación.

2.1 Análisis del proceso de gestión de reservas y planificación.

El proceso de reservas y planificación del menú del comedor de la Universidad de Cienfuegos actualmente tiene una serie de actividades que son el resultado de una larga jornada de trabajo consultando documentación. Dicho proceso tiene varios subprocesos que se describen a continuación:

Reservas:

- ✓ Ventas de tiques: Los trabajadores deben presentarse en el comedor antes de la 2:00pm para comprar el tique. Tienen derecho a ver las ofertas del menú y a escoger los productos. El pago se efectúa en dicho momento con la contadora.
- ✓ Contadora: La contadora es la encargada de la venta de tique, y responsable de cada mañana depositar el efectivo adquirido el día anterior en contabilidad.
- ✓ Conteo de estudiantes becados: La responsable de la residencia estudiantil tiene la tarea de contar las camas tendidas (es una forma de saber la cantidad de estudiantes becados presente en la escuela).

Pre-Planificación:

En horas de la mañana el administrador del comedor realiza la planificación del día siguiente con la información que le brinda:

- ✓ Almacenero: Es el encargado de aprobar o no el menú según la disponibilidad de los productos que quedan en el almacén.
- ✓ Contadora: Es la encargada de llevar la cuenta los tiques vendidos el día anterior para saber la cantidad de trabajadores que van a almorzar.
- ✓ Responsable de la residencia estudiantil: Es la encargada de llevar la información de la cantidad de estudiantes becados presentes en la escuela.

El administrador del comedor consulta con el almacenero asegurando que todos los ingredientes estén disponibles en el almacén. Si la cantidad de ingredientes necesarias no está en su totalidad, se requiere realizar un ajuste al menú adicionando o sustituyendo ingredientes. Conociendo la cantidad de comensales el administrador confecciona la planificación de los eventos (desayuno, almuerzo y cena) de forma manuscrita.

Planificación:

Luego de conocer la cantidad de comensales y la disponibilidad del almacén, que obtiene del proceso de preplanificación, el administrador del comedor elabora la planificación para el día siguiente. Para ello busca en las normas establecidas, la ración que se necesita de cada producto por comensal, así como el listado de ingredientes con sus respectivas normas imprescindibles para la confección de cada producto.

Luego de tener este desglose, se consolidan estas cantidades, se llevan a las unidades de medidas adecuadas, se multiplican por el total de comensales y se obtienen el total de las cantidades de cada producto y sus ingredientes.

Post-Planificación:

Este listado de pedidos es llevado al Departamento de ATM (Aseguramiento Técnico Material), donde ocurre el proceso de actualización de inventarios y se elabora el autorizo de predespacho o vales de salida de productos que es enviado al almacén para despachar cada uno de ellos y ser entregados al jefe de Cocina.

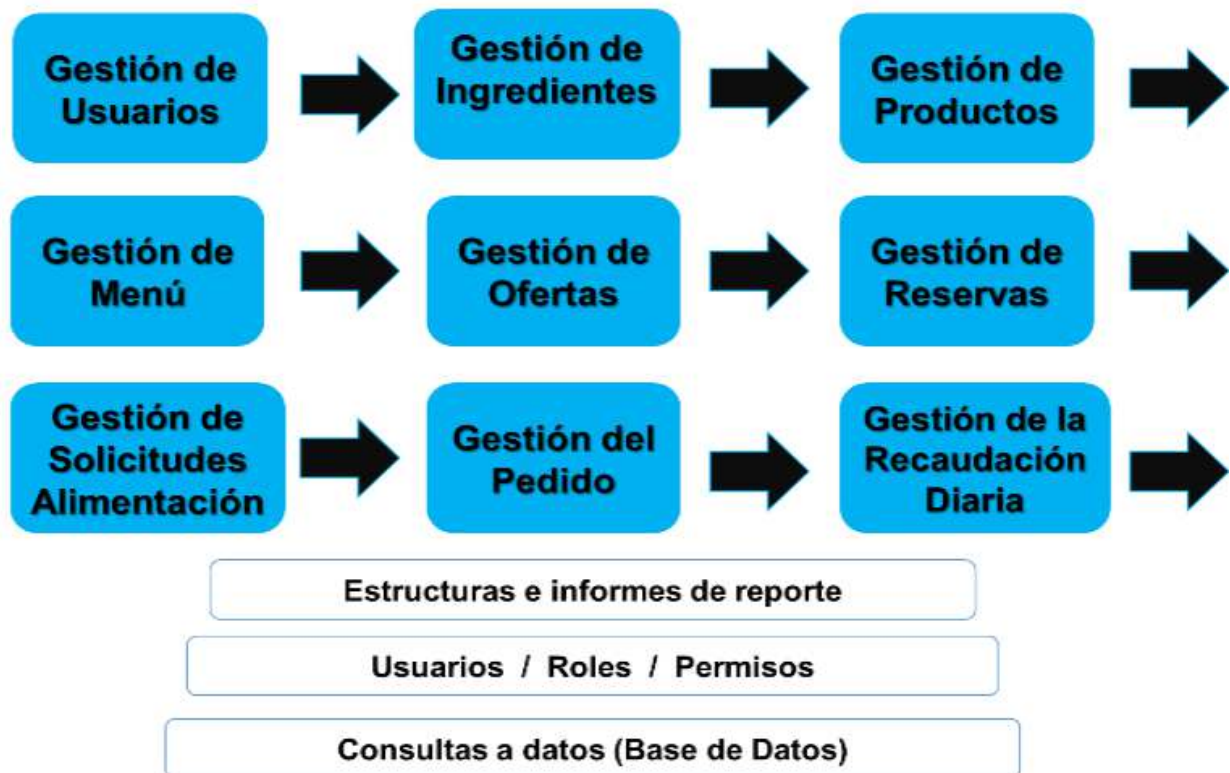


Figura 1. Descripción de la propuesta de solución.

Características y beneficios del sistema propuesto:

- ✓ **Gestión de Usuarios:** Permitirá registrar los usuarios que tendrán acceso a la aplicación, definiendo sus roles y permisos.
Beneficios: Se tendrá un control del personal que puede acceder al sistema y realizar las reservaciones, así como de los trabajadores que por sus funciones necesiten información del sistema.
- ✓ **Gestión de Ingredientes:** Permitirá gestionar la disponibilidad de ingredientes para la confección de productos que se ofertarán en el menú.
Beneficios: Se tiene un control de que ingredientes se necesitan con su respectiva norma, para elaborar cada producto del menú.
- ✓ **Gestión de Productos:** Permitirá gestionar las disponibilidades de los productos a ofertar en el menú.
Beneficios: Se conocerá en tiempo real, los productos en existencia para conformar el menú diario. Además, los comensales conocerán la ración y el precio que deben abonar para aquellos que así lo requieran.

- ✓ **Gestión del Menú:** Permitirá en dependencia de las disponibilidades de productos, elaborar el catálogo de menús a ofertar en la entidad.
Beneficios: Las posibles opciones de menú quedarán registradas en el catálogo del sistema y estarán disponibles para las opciones que lo necesiten.
- ✓ **Gestión de Ofertas:** Permitirá la confección de ofertas a partir del catálogo de menús, especificando tipo de evento, fecha y tipo de comensal al que estará destinada.
Beneficios: Las ofertas diarias o de la semana estarán visible y disponibles para todos los comensales.
- ✓ **Gestión de Reservas:** Permitirá que tanto los estudiantes como los trabajadores internos o externos, puedan realizar sus reservaciones en tiempo real, teniendo en cuenta los horarios establecidos.
Beneficios: No tendrán que desplazarse hacia el local del comedor a realizar sus reservas, se elimina la tarea de la responsable de la residencia estudiantil de contar los estudiantes presentes y de la contadora la venta de tiques el día antes. Siempre se podrá conocer las reservaciones realizadas.
- ✓ **Gestión de Solicitudes de Alimentos:** Permitirá consolidar las reservas realizadas por los comensales.
Beneficios: Conocer de forma más rápida el total de comensales, por días y cada tipo de eventos.
- ✓ **Gestión de Pedidos:** Permitirá conocer el consolidado de productos e ingredientes necesarios para elaborar las raciones de los diferentes menús.
Beneficios: Tener el control de las cantidades de productos e ingredientes necesarios para la elaboración de los alimentos.
- ✓ **Gestión de la Recaudación:** Permitirá consolidar los importes diarios por fechas y eventos.
Beneficios: Se conocerá en tiempo real cuánto debe aportar los menús elaborados al cierre del día.

El sistema además de los usuarios, controla los roles y permisos que tendrán dentro del mismo. Contará con los reportes e informes acorde al rol de cada usuario y la información estará disponible según los niveles permitidos.

2.2 Requisitos de software

Los requisitos de un proyecto de software son las funciones, características y restricciones que debe cumplir el producto final. En otras palabras, los requisitos definen qué debe hacer el software, cómo debe verse y las condiciones que deben cumplirse para que se considere exitoso. [18]

2.2.1 Requisitos Funcionales

Un requisito funcional es una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como características que el usuario detecta. [19] En la presente investigación, en interacción con el cliente se obtuvieron en total 33 requisitos funcionales, los cuales se muestran en la siguiente tabla:

No.	Nombre	Descripción	Prioridad	Complejidad
RF #1	Registrar usuario	La aplicación permite que el usuario con el rol de administrador registre a todos los usuarios con el usuario, contraseña y rol correspondiente.	Alta	Media
RF #2	Mostrar usuarios	El sistema permite que el usuario con rol de administrador visualice todos los usuarios registrados en el sistema.	Alta	Media
RF #3	Modificar usuario	El sistema permite que el usuario con rol de administrador pueda modificar cualquier usuario que haya sido registrado.	Media	Alta
RF #4	Eliminar usuario	El sistema permite que el usuario con rol de administrador elimine cualquier usuario existente en el sistema.	Baja	Baja
RF #5	Iniciar sesión	Los usuarios introducen sus datos, donde el sistema los comprueba, si son correctos accede al sistema, si no, lanza un mensaje de error.	Alta	Alta
RF #6	Cerrar sesión	Una vez se acceda al sistema, el usuario puede cerrar la sesión y salir de la aplicación. En caso de interactuar en ella, deberá ir al inicio del sistema para cerrar la sesión.	Baja	Baja

RF #7	Seleccionar fecha a reservar	El sistema permite que los usuarios puedan seleccionar la fecha que deseen reservar o por otra parte revisar las ofertas del menú.	Media	Media
RF #8	Seleccionar evento a reservar	El sistema permite que los usuarios con rol de estudiante y trabajador interno puedan seleccionar el evento (desayuno, almuerzo y comida) que deseen reservar o por otra parte revisar las ofertas del menú.	Media	Media
RF #9	Mostrar menú del día	El sistema permite que los usuarios visualicen las ofertas del menú del día y evento deseado.	Media	Media
RF #10	Seleccionar productos a consumir	La aplicación web permite que los usuarios con rol de trabajador interno y externo, elijan los productos que deseen de los que se ofertan.	Baja	Baja
RF #11	Reservar servicio	La aplicación web permite que los usuarios puedan reservar el servicio de alimentación para el día y evento seleccionado, siempre y cuando sea antes del horario límite para hacerla.	Media	Media
RF #12	Mostrar reservaciones realizadas	El sistema permite que los usuarios visualicen todas las reservaciones que haya hecho anteriormente.	Alta	Media
RF #13	Cancelar reservación	El sistema permite que los usuarios puedan cancelar reservaciones que haya hecho anteriormente.	Baja	Baja
RF #14	Agregar ingrediente	El sistema permite que el usuario con rol de administrador pueda	Media	Media

		agregar y guardar los ingredientes que están disponibles en el almacén con su nombre.		
RF #15	Mostrar ingredientes	El sistema permite que el usuario con rol de administrador pueda revisar los ingredientes existentes.	Media	Alta
RF #16	Modificar ingrediente	El sistema permite que el usuario con rol de administrador una vez guardado el ingrediente, pueda modificarlo, tanto en el nombre como la unidad de medida.	Media	Media
RF #17	Eliminar ingrediente	El sistema permite que el usuario con rol de administrador pueda eliminar un ingrediente ya existente.	Baja	Baja
RF #18	Agregar producto	El sistema permite que el usuario con rol de administrador pueda agregar y guardar los productos que están disponibles en el almacén con su nombre, precio, unidad de medida y seleccionar los ingredientes para su confección.	Media	Alta
RF #19	Mostrar productos	El sistema permite que el usuario con rol de administrador pueda revisar los productos ya existentes.	Media	Media
RF #20	Mostrar detalles del producto	El sistema permite que el usuario con rol de administrador pueda revisar los detalles del producto una vez guardado en el sistema.	Media	Media
RF #21	Modificar producto	El sistema permite que el usuario con rol de administrador una vez guardado el producto, pueda modificarlo, tanto en el nombre, precio o unidad de medida.	Media	Alta

RF #22	Eliminar producto	El sistema permite que el usuario con rol de administrador pueda eliminar un producto ya existente.	Baja	Baja
RF #23	Agregar menú	El sistema permite que el usuario con rol de administrador confeccione el menú de los usuarios seleccionando los productos.	Media	Alta
RF #24	Mostrar menús	El sistema permite que el usuario con rol de administrador visualice todos los menús confeccionados.	Alta	Media
RF #25	Modificar menú	El sistema permite que el usuario con rol de administrador pueda modificar un menú ya existente.	Media	Alta
RF #26	Eliminar menú	El sistema permite que el usuario con rol de administrador pueda eliminar un menú ya existente.	Baja	Baja
RF #27	Agregar oferta	El sistema permite que el usuario con rol de administrador pueda confeccionar las ofertas que se brindarán a los usuarios, seleccionando el menú, fecha, evento y tipo de comensal.	Alta	Alta
RF #28	Mostrar ofertas	El sistema permite que el usuario con rol de administrador pueda visualizar todas las ofertas existentes.	Media	Media
RF #29	Modificar oferta	El sistema permite que el usuario con rol de administrador una vez guardada la oferta, pueda modificarla, tanto el menú, fecha, evento como tipo de comensal.	Media	Alta

RF #30	Eliminar oferta	El sistema permite que el usuario con rol de administrador pueda eliminar una oferta ya existente.	Baja	Baja
RF #31	Mostrar total de reservaciones realizadas por los usuarios	El sistema permite que el usuario con rol de administrador pueda revisar las reservaciones que hayan hecho todos los usuarios.	Alta	Alta
RF #32	Mostrar pedido	El sistema permite que el usuario con rol de administrador pueda revisar el pedido que se hará al almacén donde puede visualizar la fecha, evento, productos y el total de raciones a pedir.	Alta	Alta
RF #33	Mostrar el importe a recaudar	El sistema permite que el usuario con rol de administrador visualice el importe total a recaudar por evento y día de las reservaciones que hayan hecho los usuarios con rol de trabajador interno y externo.	Alta	Alta

Tabla 3. Descripción de los requisitos funcionales del sistema.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) son las restricciones o los requisitos impuestos al sistema. Especifican el atributo de calidad del software. Los requisitos no funcionales se ocupan de problemas como la mantenibilidad, el rendimiento, la portabilidad, la seguridad, la confiabilidad y muchos más. [20] Se definieron 6 RNF que se describen a continuación:

No.	Descripción	Tipo de requisito
RNF #1	El sistema debe tener una interfaz sencilla, agradable y amigable. Debe tener claridad y buena organización de la información permitiendo la interpretación correcta e inequívoca de la misma.	Requisito de interfaz o apariencia externa.

RNF #2	El sistema puede ser usado por personas con o sin experiencia en el trabajo con sistemas web por lo que el diseño debe ser ameno y de fácil uso. Garantizar una secuencia lógica para la navegación del sitio.	Requisito de usabilidad.
RNF #3	La aplicación puede ser utilizados por varios usuarios, aunque solo pueden ver la información brindada; nunca modificarlo o eliminarlo. Para la protección de mismo solo puede ser modificada la información por el administrador del comedor.	Requisito de seguridad.
RNF #4	Para garantizar la protección de la información deben establecerse políticas de seguridad tales como, niveles de acceso, políticas de contraseñas y resguardo de la información.	Requisito de seguridad.
RNF #5	Se requiere un navegador web instalado en las computadoras clientes.	Requisito de software.
RNF #6	El sistema requiere de una computadora que haga la función, esta debe cumplir con las siguientes características: • Memoria RAM con 1GB o más • Capacidad de Disco Duro de 20 GB o más	Requisito de hardware.

Tabla 4. Descripción de los requisitos no funcionales del sistema.

2.3 Historias de usuarios

Una **historia de usuario** es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos. Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes. [21] Se declaran en total 33 historias de usuarios, donde a continuación se muestra la historia de usuario más relevante de esta investigación “Agregar oferta”:

Número: 27	Requisito: Agregar oferta
Programador: Roxana Mesa García	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 15 horas
Riesgo en desarrollo: Medio	Rol: Administrador
<u>Descripción:</u> El sistema permite que el usuario con rol de administrador pueda confeccionar y agregar una nueva oferta. Para ello debe primeramente ingresar a la sección “Gestión de Oferta”. <u>Botones:</u> Gestión de Oferta: Opción que permite ingresar a la sección para gestionar las ofertas. Agregar: Opción que permite agregar una nueva oferta. Guardar: Permite guardar la nueva oferta y lo agrega a la lista de ofertas. Cancelar: Permite regresar a la interfaz principal, sin tener en cuenta los cambios realizados. <u>Campos:</u> Menú: Campo de selección donde se el menú que se brindará a los usuarios. Es obligado. Fecha: Campo de selección que permite seleccionar la fecha. Es obligado. Evento: Campo de selección donde se escoge el evento (desayuno, almuerzo o comida) para dicho menú. Es obligado. Comensal: Campo de selección donde se escoge el tipo de comensal para la confección de dicha oferta. Es obligado. Observaciones: ✓ Los campos que son obligatorios deben ser llenados, sino el sistema emite un mensaje de error.	
Prototipo de interfaz gráfica de usuario:	


CERRAR SESIÓN



RESERVACIONES

GESTION DE OFERTAS

GESTION DE MENU

GESTION DE PRODUCTOS

GESTION DE INGREDIENTES

PEDIDO

IMPORTE A RECAUDAR

GESTION DE USUARIOS

←
OFERTAS





MENÚ :

FECHA :

EVENTO :

COMENSAL :

Guardar

Cancelar

Tabla 5. Historia de usuario del requisito funcional "Agregar oferta".

2.4 Descripción de la arquitectura

La arquitectura de software se refiere a una planificación basada en modelos, patrones y abstracciones teóricas, a la hora de realizar una pieza de software de cierta complejidad y como paso previo a cualquier implementación. De esta forma se dispone de una guía teórica detallada que nos permite entender cómo van a encajar cada una de las piezas de nuestro producto o servicio. [9]

Patrón de arquitectura Modelo Vista Controlador

Para la propuesta de solución se eligió el patrón Modelo-Vista-Controlador (MVC), el cual especifica la forma de estructurar el código de la aplicación dividiendo los datos de la misma, la interfaz de usuario, y la lógica de control en tres elementos distintos:

Modelo: Es responsable de acceder a la capa de almacenamiento de datos. Es el sistema gestión de base de datos. Producto.php

Vista: Es responsable de recibir datos del modelo mediante el controlador y lo muestra en una página web donde permite al usuario interactuar con ella. En este componente están index.php

Controlador: Maneja la entrada del usuario. Es quien representa la lógica del negocio y está conformado por las clases controladoras productoController.php y librerías. Solicita los datos al modelo y se los entrega a la vista para que esta se lo muestre al usuario.

Se muestra a continuación en la siguiente figura como interactúa estos tres componentes en la arquitectura definida:

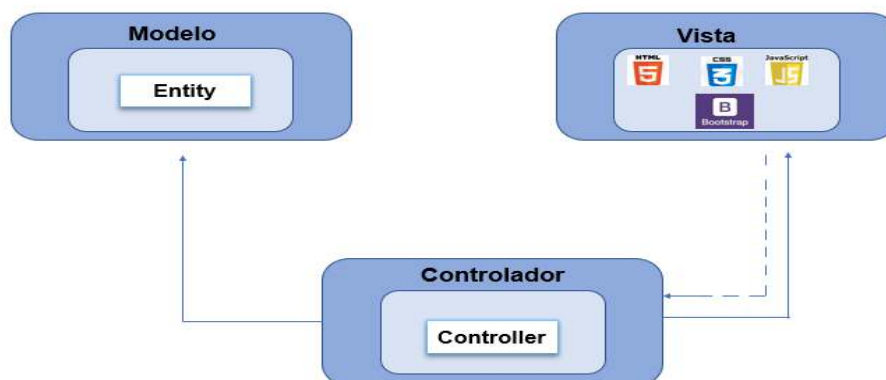


Figura 2. Arquitectura Modelo – Vista - Controlador.

A continuación, se muestra la aplicación de la arquitectura a la propuesta de solución:

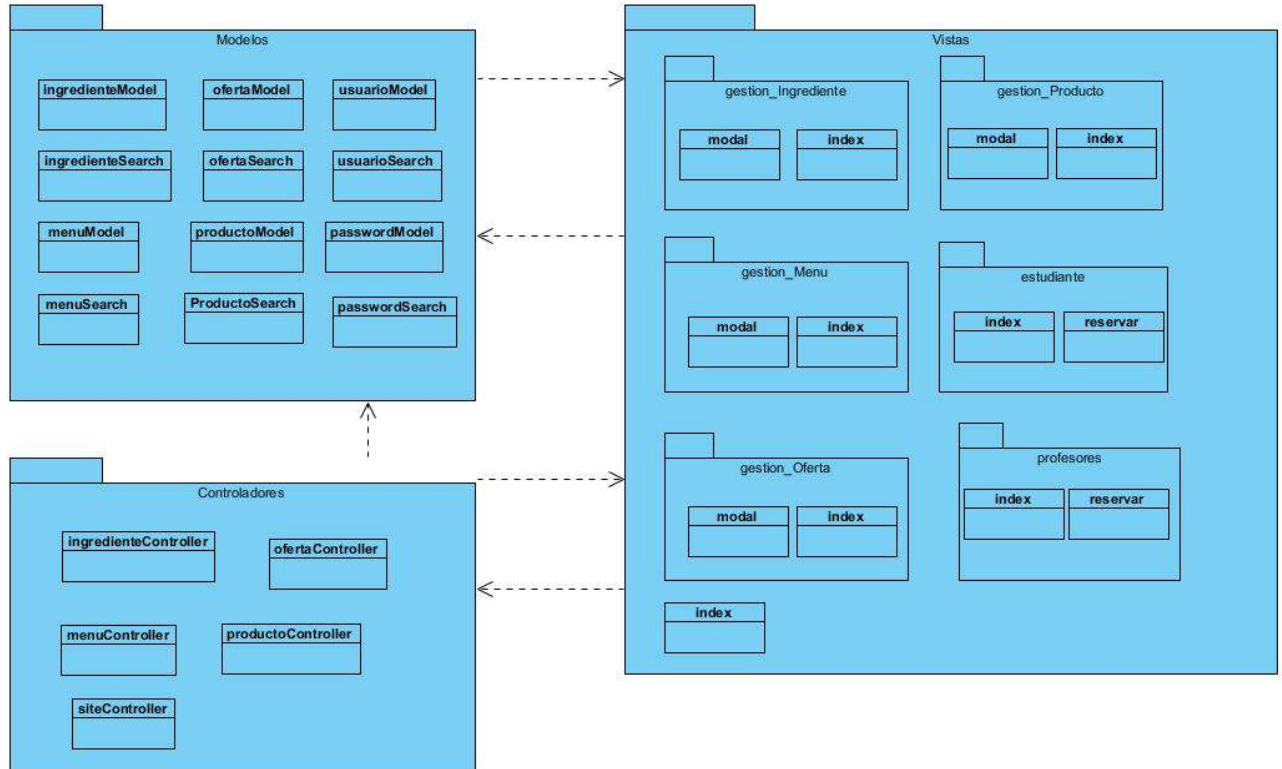


Figura 3. Aplicación de la arquitectura a la solución propuesta.

2.5 Diagrama de clases diseño

El propósito de un diagrama de clases es describir las clases que conforman el modelo de un determinado sistema. Dado el carácter de refinamiento iterativo que caracteriza un desarrollo orientado a objeto, el diagrama de clases va a ser creado y refinado durante las fases de análisis y diseño, estando presente como guía en la implementación del sistema.

Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos. [22]

A continuación, se muestra el diagrama de clases de la propuesta de solución representado a través de una relación de asociación:

Figura 4. Diagrama de clases del diseño.

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten. La mayoría de los modelos de datos se pueden representar por medio de un diagrama de base de datos acompañantes. [23]

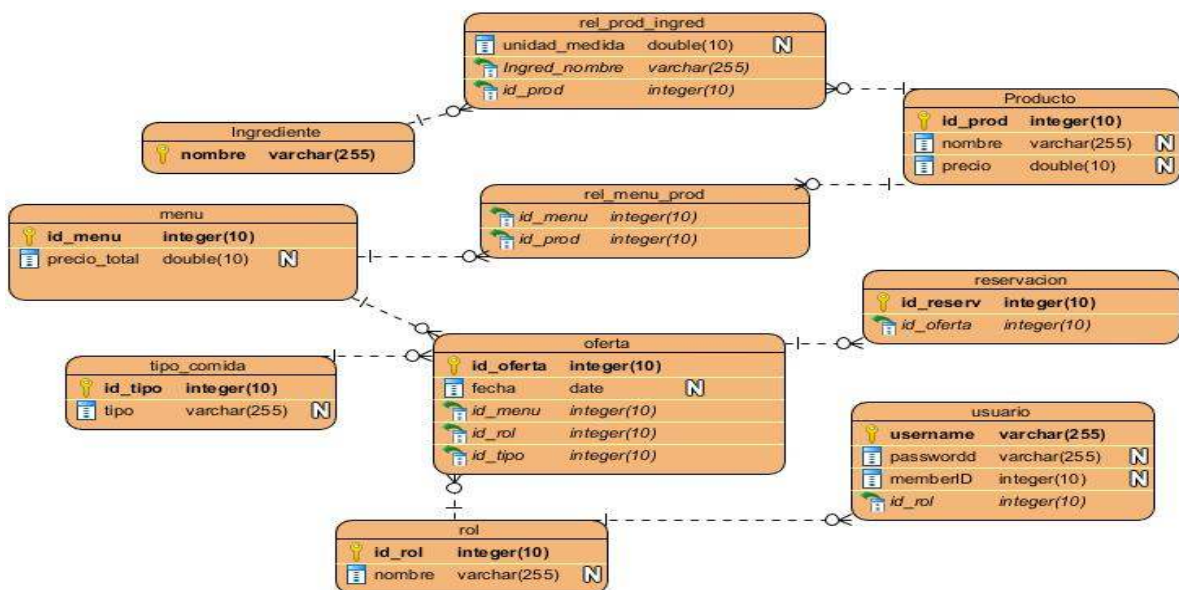


Figura 5. Modelo de datos de la propuesta de solución.

2.7 Estándar de codificación empleado

Un estándar de programación es una forma de "normalizar" la programación de forma tal que al trabajar en un proyecto cualquiera las personas involucradas en el mismo tengan acceso y comprendan el código. En otras palabras, define la escritura y organización del código fuente de un programa. Se decide aplicar el estándar de codificación **camelCase** o conocido también como **letra de caja camello**, es el nombre informal que se da en inglés a un estilo de escritura que se aplica a frases o palabras compuestas. Éste facilitará la lectura, entendimiento y mantenimiento del código. A continuación, se describe a grandes rasgos las convenciones de nomenclatura:

General:

- ✓ Se excluye el uso de las tildes y la letra ñ, la que será sustituida por nn.
- ✓ Se utilizarán nombres que sean claros, precisos y libres de ambigüedades. Ejemplo: "idProducto" y no solamente "id".
- ✓ El nombre de las variables y funciones comenzarán con letra minúscula y si este está compuesto por varias palabras, todas las palabras internas que lo componen comienzan con mayúscula. Ejemplo: "productoController".
- ✓ El contenido se identificará con tb para las tablas y nm para los nomencladores, nunca utilizando espacios en blanco.
- ✓ Los nombres de las clases serán descriptivos y simples. Usar palabras completas, evitar acrónimos y abreviaturas.
- ✓ Las variables temporales serán claras y concretas.
- ✓ Las variables booleanas sugerirán respuestas o contenidos de tipo Sí/No, por ejemplo: "esProducto".
- ✓ No se utilizarán nombres de variables, clases o funciones que puedan ser ambiguos.

2.8 Conclusiones parciales

La aplicación web diseñada agiliza y facilita la gestión del proceso de reservas y planificación del menú en el comedor de la Universidad de Cienfuegos. La arquitectura permite la ampliación de las funcionalidades del sistema. Se describieron en interacción entre el cliente y los desarrolladores los requisitos funcionales y no funcionales del sistema. Se pusieron en práctica las aplicaciones de los patrones de diseño que aseguran obtener un esquema del mismo de fácil entendimiento y mantenimiento.

CAPÍTULO 3: VALORACIÓN DE LA VIABILIDAD DE LA PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se reconoce la factibilidad de la propuesta de solución a través de la validación de los requisitos y el diseño propuesto, además de la realización de las pruebas funcionales, unitarias y de aceptación al sistema informático propuesto.

3.1 Técnicas de validación de requisitos

Con el propósito de ratificar que los requisitos de software obtenidos definen el sistema que el cliente desea se realizó un proceso de validación de los mismos, en el cual se aplicaron las técnicas siguientes: [10]

- ✓ Revisiones de los requisitos: Se hicieron revisiones de todos los requisitos por parte de los desarrolladores del sistema y por el cliente. Las revisiones originaron no conformidades de tipo técnicas, las cuales fueron corregidas a tiempo. Con el cliente se efectuaron dos revisiones donde en la primera, se adicionaron varios detalles a los requisitos funcionales para mejorar el sistema y en el segundo encuentro, se añadieron también particularidades, los cuales fueron aprobados finalmente.
- ✓ Diseño de prototipos web: A través de los prototipos se le pudo mostrar al cliente un modelo ejecutable de la aplicación que permitió tener una visión previa de cómo sería este y a través de la interacción con el cliente se confirmó si cumplían sus peticiones, los cuales fueron aceptados por los mismos.
- ✓ Generación de casos de pruebas: Fueron establecidos casos de pruebas para cada historia de usuario como parte del desarrollo de validación de los requisitos funcionales de la aplicación web.

3.2 Métrica aplicada a los requisitos

Con la finalidad de medir la calidad de la especificación de los requisitos se aplicó una de las métricas Calidad de la especificación (CE), que son la declaración exacta de las necesidades particulares que se deben cumplir, o las características esenciales que un cliente necesita y que un proveedor debe entregar. Para lograr cuán comprensibles y necesarios son los requisitos, primero que nada, se calcula el total de los requisitos de la especificación como se describe a continuación:

Tre: Total de requisitos de especificación.

Tf: Total de requisitos funcionales.

Tnf: Total de requisitos no funcionales.

Se obtiene como resultado de la sustitución de valores:

$$\mathbf{Tre = Tf + Tnf}$$

$$\mathbf{Tre = 33 + 6}$$

$$\mathbf{Tre = 39}$$

Para establecer por último la Especificidad de los Requisitos (ER) o carencia de ambigüedades en los mismos se determina mediante la siguiente operación:

Nui: Cantidad de requisitos que coincide el nivel de entendimiento con el cliente.

$$\mathbf{ER = Nui / Tre}$$

$$\mathbf{ER = 38 / 39}$$

$$\mathbf{ER = 0.97}$$

En este caso solo se produjo una contradicción en la interpretación de los requisitos. Mientras más cerca esté el resultado de ER de 1, menor será la ambigüedad.

Se obtuvo un final satisfactorio demostrando que el grado de ambigüedad de los requisitos es muy bajo (3%) ya que el 97% es entendible. Fueron cambiados y convalidados los requisitos ambiguos con el objetivo de asegurar una mejor interpretación.

3.3 Pruebas de software

Las pruebas de software son el mejor método para confirmar la funcionalidad del producto que se está desarrollando. Deben aplicarse en todas las fases del desarrollo, desde el inicio del proyecto hasta el final, de manera que puedan descubrirse fallos aislados en un primer momento o detectar un comportamiento erróneo al final. Estas pruebas sirven, por ejemplo, para descubrir fallos en el diseño y la arquitectura del software, en la seguridad o en la escalabilidad. [24]

Lo que se lleva a cabo durante estas pruebas es una evaluación minuciosa para verificar la funcionalidad de un determinado producto, es decir, comprobar que funciona tal y como debería hacerlo. Esto aporta varios beneficios que incluyen la solución de errores en una fase temprana, con lo que el esfuerzo para buscar una solución es menor. También se reducen gastos y hay un aumento del rendimiento. [24]

Niveles de pruebas:

Los niveles de prueba se definen como grupos de actividades de prueba que se organizan y se gestionan en conjunto y son realizadas según el nivel de desarrollo en que se encuentre el producto [25]. Están asociados al "qué" se prueba, pero considerando el sistema como un conjunto de partes; en esa medida cada nivel de pruebas se enfoca en un nivel de composición de partes del sistema. Existen 4 niveles de pruebas: **unidad, integración, sistema, aceptación** [26] como se describen a continuación:

- ✓ Pruebas de componentes o unitarias: Estas pruebas se enfocan en los componentes, unidades o módulos, es decir, los elementos más pequeños del software. Algunos objetivos de estas pruebas son verificar que los comportamientos funcionales y no funcionales del componente son los diseñados y especificados.
- ✓ Pruebas de integración: Estas pruebas se centran en la interacción entre distintos componentes o sistemas. Uno de sus objetivos principales es encontrar defectos, que bien pueden estar en las interfaces o en los componentes o sistemas.
- ✓ Pruebas de sistema: Tienen como objetivo la reducción de riesgos, verificar que los requerimientos funcionales y no funcionales del sistema sean cumplidos, validar que el sistema funciona como se espera, generar confianza en la calidad del sistema, prevenir y encontrar defectos.
- ✓ Pruebas de aceptación: Este nivel de prueba al igual que el de sistemas, se enfoca en el comportamiento de todo el sistema o producto. Algunos objetivos son: validar que el sistema se encuentra completo y que funcionará de acuerdo con lo esperado, establecer confianza en la calidad del sistema, verificar que los comportamientos funcionales y no funcionales son los especificados y proveer información que ayude a decidir si el sistema está óptimo para salir a producción. [25]

3.3.1 Pruebas de Caja Blanca

Una estrategia de pruebas en un conjunto de definiciones de alto nivel que determina como se llevara a cabo los procesos de pruebas de software. Para la presente investigación se eligieron los métodos de pruebas de caja blanca (o estructurales) y caja negra (o funcionales). A continuación, se describe en detalles mediante una tabla los tipos de pruebas, métodos y técnicas empleadas:

Tipo de prueba	Método de prueba	Técnica de prueba
Unitarias	Caja blanca	Camino básico
Funcionalidad	Caja negra	Partición de equivalencia

Tabla 6. Métodos de pruebas aplicada a la solución.

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- ✓ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ✓ Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- ✓ Ejecuten todos los bucles en sus límites operacionales.
- ✓ Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplica al software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. [27]

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se

construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

- ✓ A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
- ✓ Se calcula la complejidad ciclomática del grafo.
- ✓ Se determina un conjunto básico de caminos independientes.
- ✓ Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. [27]

A continuación, se muestra el proceso de prueba que se le realizó a la funcionalidad login de la clase modelo "username", por ser la función de mayor relevancia del sistema desarrollado. Primero que nada, se analiza el código y luego se enumeran las instrucciones correspondientes de esa funcionalidad:

```
public function login($username,$password){
    if (!$this->isValidUsername($username)) return false; (1)
    if (strlen($password) < 3) return false; (2)

    $row = $this->get_user_hash($username); (3)
    if(!$row == null){ (4)
        try {
            if($this->password_verify($password,$row['passwordd']) == 1){ (5)
                |
                $_SESSION['loggedin'] = true; (6)
                $_SESSION['username'] = $row['username']; (7)
                $_SESSION['memberID'] = $row['memberID']; (8)
                return true;

            }} catch(PDOException $e) {

                echo '<p class="bg-danger">'. $e->getMessage(). '</p>'; (9)
            }
        }
    }
} (10)
```

Figura 6. Funcionalidad login (acceso al sistema).

Seguidamente de este proceso se debe hacer un grafo de flujo correspondiente al código de la función como se muestra en la figura:

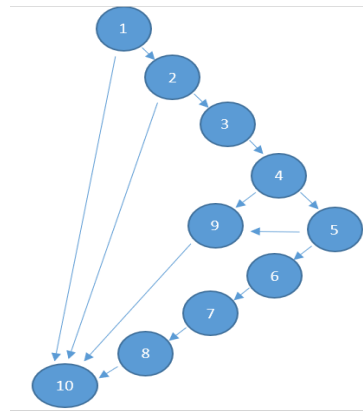


Figura 7. Grafo de flujo del código de la función login (acceso al sistema).

Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, que no es más que la métrica de calidad de software de mayor aceptación, principalmente por su independencia del lenguaje de programación. El objetivo principal de esta métrica no es contar la cantidad de ciclos y estructuras de control, sino contar la cantidad de caminos diferentes que existen en un fragmento de código. [28] La complejidad ciclomática puede ser calculada de 3 formas:

1. $V(G) = a - n + 2$, siendo a el número de arcos o aristas del grafo y n el número de nodos.
2. $V(G) = r$, siendo r el número de regiones cerradas del grafo.
3. $V(G) = c + 1$, siendo c el número de nodos de condición.

Realizando los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

- | | | |
|-----------------------|---------------|-------------------|
| 1. $V(G) = a - n + 2$ | 2. $V(G) = r$ | 3. $V(G) = c + 1$ |
| $V(G) = 13 - 10 + 2$ | $V(G) = 5$ | $V(G) = 4 + 1$ |
| $V(G) = 5$ | | $V(G) = 5$ |

Por lo que el conjunto de caminos básico sería:

Camino básico 1: 1-2-3-4-5-6-7-8-10

Camino básico 2: 1-2-3-4-5-9-10

Camino básico 3: 1-2-3-4-9-10

Camino básico 4: 1-2-10

Camino básico 5: 1-10

La guía de revisión de código, ofrece la categoría de complejidad que se muestra en la siguiente tabla: [29]

Puntuación	Categoría del código
De 0 a 10	Código estable, complejidad aceptable.
De 11 a 15	Riesgo medio, más complejidad.
De 16 a 20	Código de alto riesgo, demasiadas decisiones para una unidad de código.
Por encima de 50	Código no testeable.

Tabla 7. Categoría de la complejidad ciclomática.

Según la categorización mostrada anteriormente, el resultado obtenido de la complejidad ciclomática calculado del código login, se clasifica dicho código como estable, de riesgo bajo y complejidad aceptable.

Luego de conocer todos los caminos básicos se definen los casos de pruebas para cada uno de los ellos. A continuación, se muestra el resultado de las pruebas aplicadas a los caminos básicos 1 y 2:

Descripción: Se verificará que el usuario pueda iniciar sesión		
Condición de ejecución:	Entrada:	Resultados esperados:
Se validará que el nombre del usuario sea correcto y esté registrado en el sistema y la contraseña tenga más de tres caracteres.	(campos llenados correctamente).	Que el usuario pueda iniciar sesión a la aplicación Sistema de Servicio de Alimentación.
Resultado Obtenido: Satisfactorio		

Tabla 8. Caso de prueba del camino básico No. 1

Descripción: Se verificará que el usuario no pueda iniciar sesión		
Condición de ejecución:	Entrada:	Resultados esperados:
Se validará que el nombre del usuario sea correcto y esté registrado en el sistema y la contraseña tenga más de tres caracteres.	(campos llenados incorrectamente).	El usuario no podrá iniciar sesión a la aplicación Sistema de Servicio de Alimentación.
Resultado Obtenido: Satisfactorio		

Tabla 9. Caso de prueba del camino básico No. 2

Calcular la complejidad ciclomática permitió obtener el número de caminos independientes dentro del fragmento de código por los cuales se puede resolver una petición (determina cuan complejo es un fragmento de código). Sirvió como insumo para determinar el número de pruebas necesarias para asegurar que se ejecuta cada camino al menos una vez.

3.3.2 Pruebas de Caja Negra

Las pruebas de caja negra se centran principalmente en los requisitos funcionales y en las especificaciones del sistema. Están diseñadas para buscar errores, por medio de la interacción con la interfaz del sistema o software a probar, aquí se tienen en cuenta las entradas y salidas del software sin preocuparnos por tener un conocimiento previo de la estructura interna del software y de cómo el código fue construido. [30]

Las pruebas de caja negra son un enfoque complementario de las pruebas de caja blanca, en las cuales se buscan hallar distintos errores a los encontrados en las pruebas de caja blanca. También ayudan a identificar cualquier ambigüedad o contradicción con las especificaciones funcionales, donde separa la perspectiva del usuario final a la del desarrollador. [30]

Para la implementación de las pruebas de caja negra generalmente se utilizan diferentes técnicas como las que se describen a continuación:

- ✓ **Pruebas de casos de uso:** Se centran en diseñar casos de prueba basados en la descripción del comportamiento externo del sistema según lo definido en los casos de uso y verificar si el software se comporta de acuerdo con esas interacciones y objetivos.

- ✓ **Análisis de valores límites:** Esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- ✓ **Partición de equivalencia:** Esta técnica es un enfoque importante en el diseño de pruebas de software. Este método se utiliza para identificar conjuntos de datos de entrada que deben ser probados de manera similar ya que se espera que el comportamiento del sistema sea equivalente para estos conjuntos de datos.

Con el objetivo de comprobar las funcionalidades del sistema se utilizó la técnica de partición de equivalencia el cual la idea detrás de esta técnica es identificar clases de datos que sean "equivalentes" y asegurarse de que probamos al menos un caso de cada clase.

En el caso del requisito funcional **Registrar usuario** se aplica la técnica de partición de equivalencia de la siguiente manera:

✓ **Usuario:**

Datos Válidos:

- El nombre el usuario no puede tener más de 17 caracteres.
- No se puede repetir dos nombres de usuario iguales.
- Solo se permite caracteres alfanuméricos.
- Es un campo obligatorio.

✓ **Contraseña:**

Datos Válidos:

- La contraseña debe contener más de tres caracteres.
- Al repetir la contraseña debe coincidir con la primera.
- Es un campo obligatorio

A continuación, a través de la siguiente tabla se le hacen varias pruebas con juegos de datos válidos e inválidos al requisito funcional Registra Usuario donde se conoce la respuesta del sistema con dichos datos, el cual el resultado es satisfactorio.

Pruebas con juegos de datos	Respuesta del sistema
Nombre de usuario repetido.	Realizar Registro El nombre de usuario proporcionado ya está en uso. carlo Estudiante Contraseña Confirmar Contraseña Registrar Cancelar
Nombre de usuario correcto y contraseña con dos caracteres.	Realizar Registro La contraseña es demasiado corta. Roxana Estudiante Registrar Cancelar
Nombre de usuario correcto y las contraseñas no coinciden	Realizar Registro Las contraseñas no coinciden. Roxana Estudiante Registrar Cancelar

Tabla 10. Prueba de caja negra al requisito funcional Registrar usuario.

Para el caso del requisito funcional **Iniciar sesión** se aplica la técnica de partición de equivalencia de la siguiente manera:

✓ **Usuario:**

- El usuario debe estar registrado previamente.
- El nombre el usuario no puede tener más de 17 caracteres.
- No se puede repetir dos nombres de usuario iguales.
- Solo se permite caracteres alfanuméricos.
- Es un campo obligatorio.

✓ **Contraseña:**

- La contraseña debe contener más de tres caracteres.
- Es un campo obligatorio

Pruebas con juegos de datos	Respuesta del sistema
El nombre del usuario contiene un caracter extraño.	Por favor Inicie Sesión Nombre de usuario o contraseña incorrectos. juan! Iniciar Sesión
El nombre de usuario es válido y la contraseña contiene solo dos caracteres.	Por favor Inicie Sesión Nombre de usuario o contraseña incorrectos. Carlos Andres .. Iniciar Sesión
El nombre del usuario es válido y el campo de la contraseña está vacío.	Por favor Inicie Sesión Nombre de usuario o contraseña incorrectos. admin Contraseña Iniciar Sesión

Tabla 11. Prueba de caja negra al requisito funcional Iniciar sesión.

3.4 Validación de la solución propuesta

La validación de la solución propuesta es un proceso crucial dentro del ciclo de desarrollo de software en el que se verifica si la solución diseñada para un problema específico cumple con los requisitos del cliente y si resuelve eficazmente la necesidad planteada. Básicamente, es un paso para asegurarse de que lo que se ha construido es lo que el cliente esperaba y necesita.

Como estrategia para la validación de la aplicación Sistema de Servicio de Alimentación se establecieron un conjunto de escenarios poniendo a prueba el sistema.

✓ Escenario de prueba #1:

Un estudiante de la Universidad desea entrar en la aplicación para visualizar las ofertas del menú y desea cambiarlo.

El estudiante podrá visualizar el menú, pero no podrá cambiar las ofertas del menú ya que en el sistema se establecieron roles y permisos, que el estudiante no podrá acceder a la modificación del menú. El sistema permite que cuando un estudiante o el administrador inicien sesión en el sistema, sean interfaces distintas, como se muestra a continuación.

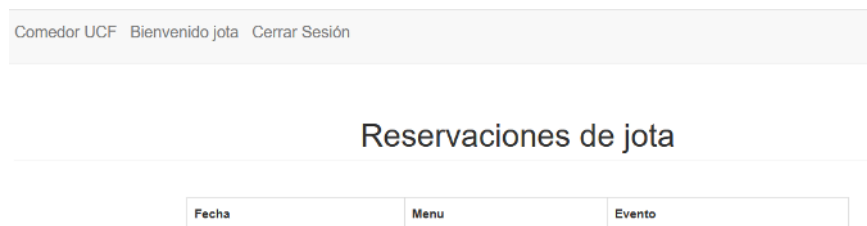


Figura 8. Interfaz del estudiante en el sistema.



Figura 9. Interfaz del administrador en el sistema.

✓ Escenario de prueba #2:

El administrador desea confeccionar un menú con un producto no existente en el sistema.

El administrador sólo puede confeccionar el menú con los productos que existan en el sistema. En caso de que el producto no se encuentre registrado, el administrador puede agregarlo al sistema para añadirlo luego en el menú.

A continuación, se muestra en la figura los productos existentes actualmente en el sistema:

Comedor UCF

Ofertas Menús Productos Ingredientes Registrar Usuario Usuarios Reservaciones Bienvenido admin Cerrar Sesión

Gestión de Productos

+ Nuevo Producto

Producto	Ingredientes	Precio	
Arroz	aceite,	3	 
Frijoles	ajo,	5	 
Pollo	aceite,	20	 
chicharo	aceite,	20	 

Figura 10. Interfaz de los productos existentes en el sistema.

Seguidamente se muestra en la siguiente figura, solo los productos disponibles en el sistema para la confección del menú:

Comedor UCF

Ofertas Menús Productos In Bienvenido admin Cerrar Sesión

Producto #1

Arroz

Arroz
chicharo
Frijoles
Pollo

Figura 11. Interfaz de los productos disponibles en el sistema.

3.5 Conclusiones parciales

La aplicación de técnicas de validación de requisitos fue fundamental en el proceso de desarrollo del software ya que contribuyó significativamente a la calidad y éxito del proyecto. Permitió verificar y confirmar que los requisitos obtenidos estaban en correspondencia con las necesidades y expectativas del cliente. Además, el uso de la métrica CE proporcionó una medida cuantitativa de la calidad de la especificación de estos, proveyendo las ambigüedades existentes para su corrección. Las pruebas de caja blanca sirvieron para evaluar la estructura interna del código fuente para garantizar su corrección y eficiencia. Las pruebas de caja negra permitieron evaluar el comportamiento del sistema desde el punto de vista de usuario, sin requerir un conocimiento detallado de su estructura interna.

Conclusiones generales

El presente trabajo de diploma ha representado un esfuerzo significativo para comprender, analizar y resolver los desafíos asociados a la gestión del proceso de reservas del comedor y la planificación del menú en la Universidad de Cienfuegos. A lo largo de esta investigación, se ha abordado un riguroso proceso metodológico para el desarrollo de una solución informática que atienda eficientemente estas necesidades.

- ✓ Se analizó todo el proceso que se lleva a cabo en la gestión del servicio de alimentación y planificación en el comedor de la Universidad de Cienfuegos.
- ✓ Se identificaron todas las funcionalidades por cada área implicadas que fueron luego implementadas en la aplicación.
- ✓ Se estudiaron sistemas informáticos semejantes que ayudaron en el diseño de la aplicación. Se utilizó la metodología ágil AUP que proporcionó flexibilidad y dinamismo al desarrollo realizado.
- ✓ La elección de tecnologías y herramientas para el desarrollo de la solución fue fundamentada en un exhaustivo análisis de los requerimientos y una evaluación meticulosa de alternativas disponibles. Se utilizó como lenguaje de programación PHP, HTML, CSS, y JavaScript. Como Gestor de Base de Datos, MySQL.
- ✓ Se identificaron 33 requisitos funcionales y 6 no funcionales, los cuales derivaron en 33 historias de usuario.
- ✓ Se implementó en ambiente Web, la aplicación “Sistema de Servicio de Alimentación” para el proceso de gestión de reservas del comedor y la planificación del menú en la Universidad de Cienfuegos.
- ✓ Los requisitos fueron validados mediante la realización de pruebas de validación de requisitos, pruebas de caja blanca y caja negra, lo cual garantizó la calidad y correcto funcionamiento del software desarrollado, su eficacia y capacidad de cumplir con los requisitos establecidos.

Recomendaciones.

Para dar continuidad a la presente investigación se recomienda:

- ✓ Desplegar la aplicación en producción para comenzar su uso por parte de estudiantes, trabajadores y personal del comedor.
- ✓ Integrar el sistema de autenticación al Directorio Activo de la Universidad de Cienfuegos.
- ✓ Continuar ampliando el sistema para añadir nuevas funcionalidades como por ejemplo integrar la gestión de pagos de los servicios.

- [1] V. PrietoDíaz et al., «Impacto de las tecnologías de la información y las comunicaciones en la educación y nuevos paradigmas del enfoque educativo», *Educ. Médica Super.*, vol. 25, n.º 1, pp. 95-102, mar. 2011.
- [2] load.es, «Software lara Comedores». Accedido: 28 de noviembre de 2023. [En línea]. Disponible en: <https://iaracatering.es/comedores/>
- [3] «Sistema de Gestión de Comedores», Grupo Integral VC. Accedido: 28 de noviembre de 2023. [En línea]. Disponible en: <https://gpovc.com/sistema-de-gestion-de-comedores-y-comensales/>
- [4] «TD_1744_08.pdf». Accedido: 28 de noviembre de 2023. [En línea]. Disponible en: https://repositorio.uci.cu/jspui/bitstream/ident/TD_1744_08/1/TD_1744_08.pdf
- [5] «Proceso Unificado de Rational», *Wikipedia, la enciclopedia libre*. 31 de agosto de 2023. Accedido: 3 de octubre de 2023. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Proceso_Unificado_de_Rational&oldid=153442814
- [6] «AUP Ingeniería de Software». Accedido: 5 de octubre de 2023. [En línea]. Disponible en: https://ingenieriadesoftware.mex.tl/63758_aup.html
- [7] «Ventajas y desventajas de las metodologías Agile (ágiles) – EBF – El blog de los mejores máster en A Coruña». Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <https://ebf.com.es/blog/ventajas-y-desventajas-de-las-metodologias-agiles-y-su-aplicacion-en-el-trabajo/>
- [8] «Ventajas y desventajas del modelo RUP», Ventajas y desventajas del modelo RUP. Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <https://unidad2ingesoftware.blogspot.com/2019/10/ventajas-y-desventajas-del-modelo-rup.html>
- [9] «Arquitectura de software: Qué es y qué tipos existen», OpenWebinars.net. Accedido: 28 de septiembre de 2023. [En línea]. Disponible en: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- [10] «IS__Libro_Sommerville_9.pdf». Accedido: 30 de septiembre de 2023. [En línea]. Disponible en: http://artemisa.unicauca.edu.co/~cardila/IS__Libro_Sommerville_9.pdf
- [11] «DDRS_U2_Contenido.pdf». Accedido: 3 de octubre de 2023. [En línea]. Disponible en: https://dmd.unadmexico.mx/contenidos/DCEIT/BLOQUE2/DS/04/DDRS/U2/descargables/DDRS_U2_Contentido.pdf
- [12] «Modelo de Arquitectura de Software».
- [13] «Las mejores tecnologías para el desarrollo web - Conócelas», Comunicare - Agencia de Marketing Online. Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <https://www.comunicare.es/mejores-tecnologias-para-el-desarrollo-de-tu-sitio-web/>
- [14] N. Meneses, «Las 15 mejores herramientas de Desarrollo Web (Guía 2023)», Coding Dojo Latam. Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <https://www.codingdojo.la/2023/01/05/las-15-mejores-herramientas-de-desarrollo-web-guia-2023/>
- [15] «¿QUE ES XAMPP Y COMO PUEDO USARLO? – Nettix Perú». Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo/>
- [16] P. Londoño, «Qué es MySQL, para qué sirve y características principales». Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <https://blog.hubspot.es/website/que-es-mysql>
- [17] «Visual Paradigm - EcuRed». Accedido: 13 de octubre de 2023. [En línea]. Disponible en: https://www.ecured.cu/Visual_Paradigm
- [18] «requisitos de software - Buscar con Google». Accedido: 12 de noviembre de 2023. [En línea]. Disponible en: https://www.google.com/search?q=requisitos+de+software&sca_esv=8663678267382092&sxsrf=AM9HkkKQjuuKBrn3W04GyJ2KMTCxoOQ%3A1699807092410&source=hp&ei=dP9QZfnmFLeLwbkPocOM0AQ&iflsi g=A06bgOgAAAAAZVEnhEPjQuWgwLCyYYCZBGUzGfIDAW5&ved=0ahUKewj58LPv8r6CAxw3RTABHaEhAO oQ4dUDCAk&uact=5&oq=requisitos+de+software&gs_lp=Egdnd3Mtd2l6IHZyZXFlaXNpdG9zIGRIIHNVznR3 YXJIMgUQAABiABDIIEAAAYgAQYBRAAGIAEMgUQAABiABDIIEAAAYgAQYBRAAGIAEMgUQAABiABDIIEAAAYgAQYBR

- AAGIAEMgUQABiABEihWIAAWPNKcAB4AJABAjgB5iSgAfN8qgEJNi0yLjEuMS4zuAEDyAEA-AEBwgiLEc4YgAQYsQMYgwHCAGsQABiABBixAxiDAciCBBAAGAPCAhEQLhiDARjHARixAxjRAXiABMICCxAAGIoFGLEDGIMB&sclient=gws-wiz#ip=1
- [19] «Qué son los Requisitos Funcionales: Ejemplos, Definición, Guía Completa», Visure Solutions. Accedido: 28 de septiembre de 2023. [En línea]. Disponible en: <https://visuresolutions.com/es/blog/requerimientos-funcionales/>
- [20] «requisitos no funcionales - Buscar con Google». Accedido: 28 de septiembre de 2023. [En línea]. Disponible en: https://www.google.com/search?q=requisitos+no+funcionales&sca_esv=569265910&source=hp&ei=cTIWZfbZGsySwbkPk9G26AQ&iflsig=AO6bgOgAAAAZRZAgc_Npu9A5wvvhL3vt9vAB0yLoATpR&oq=requisitos+no+funcio&gs_lp=Egdnd3Mtd2l6lhRyZXF1aXNpdG9zIG5vIGZ1bmNpbyoCCAAyBxAAGIoFGEMyBxAAGIoFGEMyBRAAGIAEMgUQABiABDIFEAAyGAQyBRAAGIAEMgUQABiABDIFEAAyGAQyBRAAGIAEMgUQABiABEiaPIAAWNANcAB4AJABAjgBnwagAYBLqgEJMyOzLjcuOC4xuAEDyAEA-AEBwgiLEAAyGAQYsQMYgwHCAGsQLhiABBixAxiDAciCERAUgIMBGMcBGLEDGNEDGIAEwgIIEAAyGAQYsQPCAgOQABiKBRixAxiDARhD&sclient=gws-wiz
- [21] «Historias de usuario», *Wikipedia, la enciclopedia libre*. 13 de junio de 2023. Accedido: 3 de octubre de 2023. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Historias_de_usuario&oldid=151816830
- [22] «Tutorial de diagrama de clases UML | Lucidchart». Accedido: 30 de septiembre de 2023. [En línea]. Disponible en: <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>
- [23] «Qué es un modelo de base de datos», Lucidchart. Accedido: 30 de septiembre de 2023. [En línea]. Disponible en: <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos>
- [24] «Pruebas de software: Tipos e importancia», UNIR. Accedido: 25 de noviembre de 2023. [En línea]. Disponible en: <https://www.unir.net/ingenieria/revista/pruebas-software/>
- [25] C. Gomez, «Niveles de prueba de software», Diario de QA. Accedido: 10 de octubre de 2023. [En línea]. Disponible en: <https://www.diariodeqa.com/post/niveles-de-prueba>
- [26] «niveles de prueba - Buscar con Google». Accedido: 10 de octubre de 2023. [En línea]. Disponible en: [https://www.google.com/search?q=niveles+de+prueba&sca_esv=572205757&source=hp&ei=eUslZcPAG7DOkPIPg7OGmAg&iflsig=AO6bgOgAAAAZSVZifv4R6bfSzQUxFBr_k6BbQz-c6LN&ved=0ahUKEwiD4ZeXxeuBAXUwJOQIHYZAYMQ4dUDCAk&uact=5&oq=niveles+de+prueba&gs_lp=Egdnd3Mtd2l6lhFuaXZlbGVzIGRIIHBydWViYTIIEAAyGAQyBxAAGIoFGEMyBRAAGIAEMgUQABiABDIFEAAyGAQyBRAAGIAESJYLUABY9wdwAHgAkAEAmAGFBaABxw6qAQswLjMuMS4wLjEuMbgBA8gBAPgBAQ&sclient=gws-wiz](https://www.google.com/search?q=niveles+de+prueba&sca_esv=572205757&source=hp&ei=eUslZcPAG7DOkPIPg7OGmAg&iflsig=AO6bgOgAAAAZSVZifv4R6bfSzQUxFBr_k6BbQz-c6LN&ved=0ahUKEwiD4ZeXxeuBAXUwJOQIHYZAYMQ4dUDCAk&uact=5&oq=niveles+de+prueba&gs_lp=Egdnd3Mtd2l6lhFuaXZlbGVzIGRIIHBydWViYTIIEAAyGAQyBxAAGIoFGEMyBRAAGIAEMgUQABiABDIFEAAyGAQyBRAAGIAEMgUQABiABDIFEAAyGAQyBRAAGIAESJYLUABY9wdwAHgAkAEAmAGFBaABxw6qAQswLjMuMS4wLjEuMbgBA8gBAPgBAQ&sclient=gws-wiz)
- [27] «Pruebas de caja blanca - EcuRed». Accedido: 23 de noviembre de 2023. [En línea]. Disponible en: https://www.ecured.cu/Pruebas_de_caja_blanca
- [28] «Complejidad ciclomática», JohnDev's site. Accedido: 25 de noviembre de 2023. [En línea]. Disponible en: <https://www.johndev.co/posts/ciclotomic-complexity/>
- [29] Auditor, «Complejidad ciclomática y calidad del código», Auditoría de código. Accedido: 5 de diciembre de 2023. [En línea]. Disponible en: <https://auditoriadecodigo.com/complejidad-ciclotomatica-mide-la-calidad-y-seguridad-de-tu-codigo/>
- [30] «Pruebas de caja negra». Accedido: 25 de noviembre de 2023. [En línea]. Disponible en: <https://es.linkedin.com/pulse/pruebas-de-caja-negra-sqa-s-a>