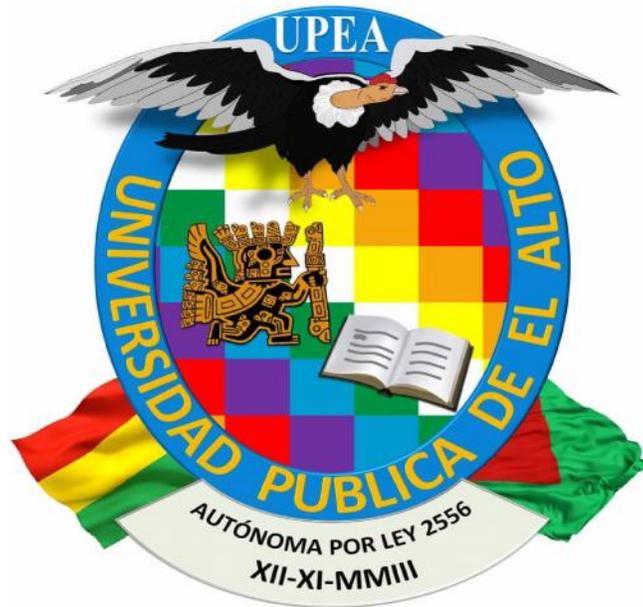


# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

“SISTEMA WEB DE SEGUIMIENTO Y CONTROL ADMINISTRATIVO

CASO: “SERVIALINUT”

Para optar al título de Licenciatura en Ingeniería de Sistemas

**Mención: Gestión y Producción**

**Postulante: Randy Richard Chipana Yujra**

**Tutor Metodológico: Ing. Marisol Arguedas Balladares**

**Tutor Especialista: Lic. María Magdalena Aguilar Guanto**

**Tutor Revisor: Ing. Milton Osvaldo Zurita Benito**

EL ALTO – BOLIVIA

2022

## **DEDICATORIA**

El presente proyecto de grado agradezco a Dios a mis padres, Ricardo Chipana Rojas y Viviana Yujra Mamani y Hermanos con todo mi amor y cariño, por ser el pilar fundamental en todo lo que soy, en mi educación, tanto académica, como de la vida, por su incondicional apoyo.

## **AGRADECIMIENTO**

A Dios por haberme permitido llegar hasta esta etapa de mi vida.

A mi familia por haberme brindado todo el apoyo necesario para llegar hasta estas instancias y poder culminar la carrera.

A mi tutor metodológico Ing. Marisol Arguedas Balladares de igual forma a mi tutor especialista Lic. María Magdalena Aguilar Guanto también a mi tutor revisor Ing. Milton Osvaldo Zurita Benito. Quienes supieron guiarme con sus observaciones y sugerencias para el desarrollo y finalización del presente proyecto de grado.

A los docentes de la Carrera de Ingeniería de Sistemas, desde el primer día de mis clases hasta la culminación de la carrera, por haber compartido sus conocimientos y haber sido parte de nuestra formación dentro de esta prestigiosa universidad.

A mis compañeros y amigos dentro de la carrera por su amistad y apoyo moral durante la vida universitaria.

## RESUMEN

El “SISTEMA WEB DE SEGUIMIENTO Y CONTROL ADMINISTRATIVO CASO: “SERVIALINUT”, se desarrolló con el propósito de mejorar los procesos manuales en la administración de la empresa. Para la emisión de reportes oportunos sobre las ventas realizadas, clientes, ingresos y salidas.

Con la metodología empleada en el presente proyecto que brinda un desarrollo de software orientado a la web. basada en UML. Para el desarrollo del sistema se utilizaron lenguajes como PHP y JavaScript, para la administración de la base de datos se utilizó el entorno MariaDB, utilizando el servidor Apache XAMPP.

Para la construcción del sistema se utilizó Vue framework para el desarrollo del sistema bajo la arquitectura, Bootstrap y JQuery son framework's que se utilizó para el maquetado del sistema. Una vez concluido el Sistema web de seguimiento y control administrativo se procedió a elaborar las pruebas de funcionamiento como también las pruebas de calidad de software y de seguridad basadas en las normas ISO/IEC-9126 respectivamente, con las que se pudo constatar que el sistema responde a los requerimientos institucionales, además se realizó el análisis costo/beneficio donde se establece que el presente proyecto es rentable. Para finalmente concluir que los objetivos planteados fueron alcanzados y que el producto desarrollado cumple con los requerimientos, funcionalidades y normas de calidad que requiere el usuario.

# INDICE GENERAL

## CAPÍTULO I

<b>1. MARCO PRELIMINAR</b> .....	4
<b>1.1. INTRODUCCIÓN</b> .....	4
<b>1.2. ANTECEDENTES DEL PROYECTO</b> .....	5
1.2.1. Antecedentes Institucionales.....	5
1.2.2. Antecedentes De Trabajos Afines .....	6
<b>1.3. PLANTEAMIENTO DEL PROBLEMA</b> .....	7
1.3.1. Problema Principal.....	7
1.3.2. Problemas Secundarios .....	7
<b>1.4. OBJETIVOS</b> .....	8
1.4.1. Objetivo General.....	8
1.4.2. Objetivos Específicos .....	8
<b>1.5. JUSTIFICACIÓN DEL TEMA</b> .....	9
1.5.1. Justificación Técnica .....	9
1.5.2. Justificación Económica .....	9
Beneficios.....	9
1.5.3. Justificación Social.....	9
<b>1.6. METODOLOGÍA</b> .....	10
<b>1.7 HERRAMIENTAS</b> .....	14
<b>1.8 LÍMITES Y ALCANCES</b> .....	17
1.8.1 Limites.....	17
1.8.2 Alcances .....	17
<b>1.9 APORTES</b> .....	18
<b>2. MARCO TEORICO</b> .....	19
<b>2.1 INTRODUCCION</b> .....	19
<b>2.2. MARCO TEÓRICO</b> .....	19
<b>2.3. INGENIERÍA DE SOFTWARE</b> .....	20
2.3.1 Metodologías de desarrollo de software.....	20
2.3.2. Proceso unificado agil (AUP).....	21
2.3.3. JUSTIFICACIÓN DEL USO DEL PROCESO UNIFICADO ÁGIL .....	25
COMO MODELO DE DESARROLLO .....	25
2.3.4. PRINCIPIOS DEL PROCESO UNIFICADO ÁGIL.....	27

<b>2.4. INGENIERÍA WEB</b> .....	28
<b>2.4.1. Basados en UML (UWE)</b> .....	30
<b>2.4.2. Atributos de los sistemas y aplicaciones basados en la web</b> .....	34
<b>2.4.3. Métodos empleados en ingeniería web</b> .....	37
<b>2.4.4. Herramientas y tecnologías</b> .....	37
<b>2.5 METODO DE PRUEBAS DE SOFTWARE</b> .....	40
<b>2.5.1 Pruebas de Caja Blanca</b> .....	41
<b>2.5.2 Pruebas de Software</b> .....	42
<b>2.6 METRICAS DE CALIDAD DE SOFTWARE</b> .....	42
<b>2.6.1 Calidad de Software</b> .....	42
<b>2.6.2 Seguridad del Software y Estándares de Calidad</b> .....	43
<b>2.6.3 Métricas de Calidad</b> .....	43
<b>2.7 METODOS DE ESTIMACION DE COSTO DE SOFTWARE</b> .....	44
<b>2.7.1 Método de Estimación de costo COCOMO</b> .....	44
<b>2.8 ISO 9126</b> .....	47
<b>2.8.1 Factores de Calidad según ISO 9126</b> .....	50
<b>3. MARCO APLICATIVO</b> .....	52
<b>3.1 INTRODUCCION</b> .....	52
<b>3.2 ANALISIS Y DISEÑO DEL SISTEMA</b> .....	52
<b>3.2.1 Análisis de la situación actual</b> .....	52
<b>3.3 FASE INICIAL</b> .....	53
<b>3.3.1 Requerimientos de Usuario</b> .....	54
<b>3.3.2 Requerimientos Tecnológicos</b> .....	54
<b>3.4 FASE DE ELABORACION</b> .....	55
<b>3.4.1 Modelo del negocio</b> .....	56
<b>3.4.2 Modelo de Caso de Uso</b> .....	57
<b>3.4.3 Diagramas Caso de Uso</b> .....	58
<b>3.4.4 Diagrama de Secuencia</b> .....	64
<b>3.4.5 Diagrama de Colaboración</b> .....	66
<b>3.4.6 Diagrama de Estados</b> .....	68
<b>3.4.7 Diagrama de Clases</b> .....	70
<b>3.5 FASE DE CONSTRUCCION</b> .....	71
<b>3.5.1 Diagrama de Componentes</b> .....	71

3.5.2 Arquitectura del Sistema.....	72
3.5.3 Implementación del sistema.....	72
3.6 FASE DE TRANSICIÓN.....	84
3.7 METRICAS DE CALIDAD.....	85
3.7.1 Introducción.....	85
3.8 FIABILIDAD .....	¡Error! Marcador no definido.
3.9 FUNCIONABILIDAD .....	¡Error! Marcador no definido.
3.9.1 Completitud de la implementación funcional .....	87
3.10 MANTENIMIENTO .....	90
3.11 ANALISIS DE COSTOS.....	91
4 CAPITULO IV .....	93
4.1 CONCLUSIONES.....	93
4.2 RECOMENDACIONES.....	93
5 BIBLIOGRAFIA V .....	94
ANEXOS.....	95

## INDICE DE FIGURAS

Figura 2.1: Marco Teórico.....	16
Figura 2.2: Esquema Proceso Unificado Ágil.....	18
Figura 2.3: (Ciclo de Vida AUP) Fuente-Scott Ambler o (Agile Unified Process AUP).....	20
Figura 2.4: Fases del Proceso Unificado Ágil.....	23
Figura 2.5: Esquema de Desarrollo-Ingeniería Web.....	25
Figura 2.6: Ingeniería Web Basada en UML.....	27
Figura 2.7: (IWeb Basado en UML) fuente Jesse James Garret 2000.....	28
Figura 2.8: (Fases de Desarrollo UWE) Fuente-Nora Koch.....	29
Figura 2.9: (Fases del UWE) Fuente-Nora Koch.....	30
Figura 2.10 Grafico de Flujo.....	38
Figura 3.1: Organigrama de la Empresa Servialinut.....	50
Figura 3.2: Modelo de negocio .....	53
Figura 3.3: Modelo de Caso de Uso.....	54
Figura 3.4: Caso de Uso: Realiza Suscripción.....	55
Figura 3.5: Caso de Uso: Registra Ventas.....	57
Figura 3.6: Caso de Uso: Administrador.....	58
Figura 3.7: Caso de Uso: Generación de Reportes.....	60
Figura 3.8: Diagrama de Secuencia: Realiza Ingreso .....	61
Figura 3.9: Diagrama de Secuencia: Registra Ventas.....	62
Figura 3.10: Diagrama de Secuencia: Registra Distribución.....	62
Figura 3.11: Diagrama de Secuencia: Generación de Reportes.....	63
Figura 3.12: Diagrama de Ingreso: Realiza Ingreso Fuente: Elaboración propia.....	63
Figura 3.13: Diagrama de colaboración: Registra Ventas.....	6
Figura 3.14: Diagrama de Colaboración: Registra Distribución.....	64
Figura 3.15: Diagrama de Colaboración: Generación de Reportes.....	65
Figura 3.16: Diagrama de Estados: Realiza Ingreso Fuente: Elaboración propia.....	65
Figura 3.17: Diagrama de Estados: Registra Ventas.....	66
Figura 3.18: Diagrama de Estados: Registra Administrador.....	66

Figura 3.19: Diagrama de Estados: Generación de Reportes.....	67
Figura 3.20: Diagrama de Clases.....	67
Figura 3.21: Diagrama de Componentes.....	68
Figura 3.22: Arquitectura del sistema.....	69
Figura 3.23: Inicio de Sesión.....	70
Figura 3.24: Pantalla principal.....	70
Figura 3.25: Pantalla de Ingreso de Alimentación.....	71
Figura 3.26: Características de la suscripción.....	71
Figura 3.27: Búsqueda de Clientes.....	72
Figura 3.28: Pantalla de Categoría de Dietas.....	72
Figura 3.29: Pantalla Nueva Dieta.....	73
Figura 3.30: Pantalla Editar Dieta.....	73
Figura 3.31: Búsqueda de Categoría.....	74
Figura 3.32: Nuevo Encargado.....	74
Figura 3.33: Pantalla Editar Encargado.....	75
Figura 3.34: Búsqueda de Encargado.....	75
Figura 3.35: Nuevo Ingreso.....	76
Figura 3.36: Pantalla detalle Ingreso.....	76
Figura 3.37: Búsqueda de Encargado.....	77
Figura 3.38: Nuevo Cliente.....	77
Figura 3.39: Pantalla editar Cliente.....	78
Figura 3.40: Búsqueda de Encargado.....	78
Figura 3.41: Nuevo Venta.....	79
Figura 3.42: Pantalla detalle Venta.....	79
Figura 3.43: elegir permisos.....	80
Figura 3.44: Permisos del Administrador.....	80

## **INDICE DE TABLAS**

Tabla 2.1: Detalle de coeficientes de COCOMO.....	41
Tabla 2.2 Ecuaciones del Método COCOMO.....	42
Tabla 2.3: Atributos FAE.....	43

<b>Tabla 3.1: Caso de Uso: Realiza el Ingreso.....</b>	<b>56</b>
<b>Tabla 3.3: Caso de Uso: Registro Administrativo.....</b>	<b>59</b>
<b>Tabla 3.4: Caso de Uso: Generación de Reportes.....</b>	<b>60</b>
<b>Tabla 3.5 Fiabilidad.....</b>	<b>82</b>
<b>Tabla 3.6 Análisis de Fiabilidad.....</b>	<b>83</b>
<b>Tabla 3.7 Fiabilidad a la finalización del proyecto.....</b>	<b>83</b>
<b>Tabla 3.8 Análisis de Fiabilidad a la finalización del proyecto.....</b>	<b>84</b>
<b>Tabla 3.9 Análisis de Funcionalidad.....</b>	<b>85</b>
<b>Tabla 3.9 Mejora de la Funcionalidad a la finalización del proyecto.....</b>	<b>86</b>
<b>Tabla 3.10 Resumen métricas de calidad.....</b>	<b>87</b>
<b>Tabla 4.13 costo del proyecto.....</b>	<b>89</b>

## **CAPÍTULO I**

### **1. MARCO PRELIMINAR**

#### **1.1. INTRODUCCIÓN**

Los Sistemas de Información y las Tecnologías de Información están cambiando la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, tales como la automatización de los procesos operativos, tareas administrativas.

Las Tecnologías de la Información se conceptualizan como la integración y convergencia de la computación, las telecomunicaciones y la técnica para el procesamiento de datos, donde sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.

Los componentes mencionados conforman los protagonistas del desarrollo informático en una sociedad, tanto para su desarrollo como para su aplicación, además se reconoce que las tecnologías de la información constituyen el núcleo central de una transformación multidimensional que experimenta la economía y la sociedad; de aquí lo importante que es el estudio y dominio de las influencias que tal transformación impone al ser humano como ente social, ya que tiende a modificar no sólo sus hábitos y patrones de conducta, sino, incluso, su forma de pensar.

La empresa “SERVIALINUT” está dedicada al servicio de alimentación y nutrición a hospitales.

Así como muchas de las empresas del mismo rubro que en los últimos años fueron apareciendo en la ciudad de La Paz y El Alto y de esta manera aporta al desarrollo del país en el ámbito de nutrición y alimentación.

Tiene como actividad diaria el registro de clientes en cuanto a sus dietas de los hospitales con los que trabaja y la cantidad de registros va en aumento, pero este proceso se realiza manualmente en planillas, existen datos duplicados u omitidos, por

tal motivo es difícil acceder a esta información y se recurren a constantes fallas de atención ocasionando que el cliente empeore su estado clínico llevándolo en algunos casos hasta la muerte consecuentemente procesos legales y por ende pérdidas económicas.

Considerando todo lo mencionado, el propósito de este proyecto es implementar un Sistema Web de Seguimiento y Control de la información de clientes de la empresa "SERVIALINUT" para que brinde información oportuna y actualizada al personal de la empresa, para mantener la salud y bienestar físico del cliente y ofrecer servicios de calidad y finalmente cumplir con la misión de la empresa.

## **1.2. ANTECEDENTES DEL PROYECTO**

### **1.2.1. Antecedentes Institucionales**

La nutrición hospitalaria es fundamental para la recuperación del paciente y nuestra participación es contribuir activamente con el personal médico para coadyuvar así a mejorar la calidad de atención al usuario identificando oportunamente la prevalencia de la desnutrición hospitalaria y prescribir oportunamente el apoyo nutricional indicado para cada usuario.

Es por ello que la Empresa de nutrición "SERVIALINUT" busca proporcionar al usuario una alimentación adecuada de acuerdo a la fisiopatología y de esta manera contribuir a la recuperación de la salud, manteniendo y mejorando su estado nutricional.

#### ➤ Misión

Proporcionar los servicios de nutrición y dietética apegado a las normas establecidas por esta institución, para los niveles de alimentación del usuario y contribuir a la recuperación de la salud manteniendo y mejorando su estado nutricional.

#### ➤ Visión

Ser una unidad eficiente y capacitada que otorgue sus servicios cumpliendo cabalmente con nuestra normatividad, y que garantice una nutrición y dietética adecuada para todos los usuarios en esta institución.

La empresa realiza el servicio al Seguro Social Universitario (SSU) que esta ubicado en la avenida 6 de agosto - La paz

### **1.2.2. Antecedentes De Trabajos Afines**

El Sistema Web de Administración y Seguimiento incorpora nuevos elementos conceptuales y para consolidarlos se busca antecedentes de prácticas con las metodologías utilizadas destacando los siguientes:

- ) Limachi Machaca, K. B. (2018). *Sistema web para el control y seguimiento de ventas de productos artesanales caso: Bolivia Tech Hub* (Doctoral dissertation).
- ) Peralta, T., & Briggith, A. (2018). *Aplicación web para el control y seguimiento del rendimiento académico en estudiantes de secundaria de la Institución Educativa Privada Jesus el Maestro.*
- ) Vega Macías, E. J., & Villón Pesantes, D. E. (2021). *Desarrollo de un prototipo en ambiente web para la gestión de control y seguimiento de los mantenimientos realizados a los vehículos en una lubricadora* (Bachelor's thesis, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales.).
- ) LISSETTI, L. M. S. (2022). *SISTEMA WEB PARA EL CONTROL DE RECEPCIÓN DE PEDIDOS Y SEGUIMIENTO DE RUTAS EN LA INDUSTRIA TRANSFORMADORA DE PLÁSTICO* (Doctoral dissertation, UNIVERSIDAD AGRARIA DEL ECUADOR).
- ) CUEVAS COYO, G. C. (2020). *SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y SEGUIMIENTO ACADÉMICO.*
- ) CONDORI QUISPE, L. F. (2020). *SISTEMA DE INFORMACIÓN WEB DE CONTROL Y SEGUIMIENTO DE ASOCIADOS.*

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

La información de un cliente es muy importante para la empresa, ya que la misma es necesaria para la toma de decisiones, como: clientes registrados en cada hospital, fechas de vencimiento de cada servicio, renovación de servicios, etc. Acceder a esta información es muy dificultosa y morosa, ya que el responsable del área contable tiene que llenar los datos manualmente en planillas físicas, y luego para poder visualizar se tiene que buscar los registros uno por uno. El personal de ventas al momento de realizar una cobranza, renovación o venta de servicios a un cliente no cuentan con información fidedigna. También no se pueden detectar a clientes con servicios vencidos. Por tal motivo existen pérdidas económicas, información duplicada, insatisfacción y desconfianza en la empresa.

En ese contexto se plantea el siguiente problema:

#### **1.3.1. Problema Principal**

¿Cómo se puede optimizar el Seguimiento y Control Administrativo en la empresa “SERVIALINUT”, tal que se cuente con información oportuna y confiable, para mantener la salud y bienestar de los clientes?

#### **1.3.2. Problemas Secundarios**

- La información respecto a los clientes y sus dietas se guardan de forma manual en planillas físicas, esto ocasiona pérdida de tiempo al momento de querer acceder a las mismas.
- No se cuenta con estadísticas respecto a los clientes, por lo que no se puede tomar decisiones de forma oportuna para cumplir con sus objetivos.
- No se tiene el inventario de los ingredientes lo que ocasiona que el personal improvise los ingredientes faltantes que causa daños a los pacientes.
- El personal no tiene acceso a la información de clientes, lo que genera incertidumbre y malestar al cliente por la pérdida de tiempo.

- No se realiza un control a las fechas de vencimiento de los servicios a los clientes, ocasionando que a largo tiempo sean olvidados y no se pueda cobrar o renovar el servicio, lo que genera pérdida económica a la empresa.
- No se tiene un rol designado de atención lo que ocasiona desorden en la atención a los clientes.

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Desarrollar un Sistema Web de Seguimiento y Control administrativo en la empresa "SERVIALINUT", para optimizar la información y que sea oportuna y confiable manteniendo la salud y bienestar de los clientes.

Se representará en el Árbol de Objetivos. Ver Anexo B

### **1.4.2. Objetivos Específicos**

- ✓ Automatizar la información respecto a los clientes, para poder tener información rápida y accesible.
- ✓ Brindar a gerencia general información oportuna y actualizada de sus clientes y servicios que le ayude en la toma de decisiones.
- ✓ Generar un registro de inventario de tipos de menú, para que el encargado del área ahorre tiempo, y no omita y/o duplique los menús.
- ✓ Brindar información sobre los servicios de clientes al personal de ventas que requieran dicha información, que permita a la empresa disminuir los gastos innecesarios y así brindar una mejor atención a los clientes.
- ✓ Facilitar un control a las fechas de vencimiento de los servicios de clientes, para que se puedan cobrar los servicios con anticipación.
- ✓ Generar un rol de atención para no tener pérdidas a largo plazo

## **1.5. JUSTIFICACIÓN DEL TEMA**

### **1.5.1. Justificación Técnica**

Para la implementación de este sistema no será necesario aplicar tecnología de punta, esto puede ser un terminal o un dispositivo, que tenga acceso desde cualquier equipo a una conexión de red internet que la Institución ya la tiene establecida, mediante un Hosting.

Para ello la empresa implementara un hosting y dominio de acuerdo a las necesidades del sistema.

### **1.5.2. Justificación Económica**

#### **Costos**

El software será desarrollado con la ayuda de herramientas de licencia gratuita, también mencionar que en la parte del hardware no requerirá equipos especializados, por lo cual se adaptará, favorablemente a los equipos con los que ya cuenta la Empresa “SERVIALINUT”.

#### **Beneficios**

El software reducirá en gran manera los gastos en los que se incurren por errores, durante las operaciones que se realizan en la Empresa “SERVIALINUT”, gracias a un diseño con tecnología de base de datos, y el sistema que se desarrolle permitirá contar con un mejor control y de esta manera incrementar utilidades, el sistema web realizara procesos de manera automática minimizando costos y tiempo.

### **1.5.3. Justificación Social**

A medida que avanza el tiempo, la cantidad de clientes va en aumento, entonces la empresa debe ser capaz de atender y mejorar sus servicios, para que los mismos se sientan seguros y confíen en la empresa.

Con el sistema, el personal de la empresa cuenta con información rápida oportuna y segura de los clientes, y mejora la atención a los mismos, también es una ayuda para gerencia general y el área contable porque les facilita en el control y seguimiento de los servicios, les ayuda en la toma de decisiones para lograr el objetivo de la empresa y los clientes obtienen una mejor atención oportuna y confiable.

## **1.6. METODOLOGÍA**

Los aspectos de la investigación que tomamos en cuenta son:

### **PROCESO UNIFICADO AGIL (AUP).**

El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) en inglés 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 Agil, Gestión de Cambios Agil, y Refactorización de Base de Datos para mejorar la productividad.

El proceso unificado (Unified Process o UP) es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (Rational Unified Process) de IBM.

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.

**Al igual que en RUP, en AUP se establecen cuatro fases:**

- **Inception(Concepción):** El objetivo de esta fase es obtener una comprensión común clienteequipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Las disciplinas se llevan a cabo de manera sistemática, a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software de trabajo que responda a las necesidades de sus interlocutores. Las disciplinas son:

**1. Modelo.** El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.

**2. Aplicación.** El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.

**3. Prueba.** El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.

**4. Despliegue.** El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.

**5. Gestión de configuración.** El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.

**6. Gestión de proyectos.** El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.

**7. Entorno.** El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc) estén disponibles para el equipo según sea necesario.

### **UML-Based Web Engineering (UWE)**

Es un SI donde una gran cantidad de datos volátiles, altamente estructurados, van a ser consultados, procesados y analizados mediante navegadores. Una de las principales características va a ser su alto grado de interacción con el usuario, y el diseño de su interfaz debe ser claro, simple y debe estar estructurado de tal manera que sea orientativo para cada tipo de usuarios.

**Fuente: Metodologías para el Desarrollo de Aplicaciones Web: UWE, Daniel Mínguez Sanz, 2014**

Se escoge la metodología porque es una herramienta para modelar aplicaciones web y es utilizada en la ingeniería web.

UWE, se basa en el proceso unificado (UML) pero adaptados al desarrollo web.

#### **Fases de UWE:**

Las fases o etapas a utilizar son:

- **Captura, análisis y especificación de requisitos.**
- **Diseño conceptual.**
- **Diseño de navegaciones.**
- **Diseño de presentación.**

El producto final de este paso es, una serie de vistas de interfaz de usuario que se presentan mediante diagramas de interacción UML.

Entre los Diagramas de UML que se realizan están:

- **Diagramas de Secuencia.**
- **Diagramas de Colaboración.**
- **Diagramas de Estado.**
- **Diagramas de Casos de Uso.**
- **Diagramas de Actividad.**

(WIP, por sus siglas en inglés de Work In Progress), asigna límites concretos a cuántos elementos pueden estar en progreso en cada estado del flujo de trabajo. Mide el lead time (tiempo medio para completar un elemento, a veces llamado tiempo de ciclo), optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible.

Para el modelado y análisis del sistema se utilizará UWE que sirve para modelar aplicaciones web, presta una especial atención a la sistematización y personalización (sistemas adaptativos). Provee de perfiles UML, metamodelos, un proceso de desarrollo dirigido por modelos, y herramientas de soporte para el diseño sistemático de aplicaciones web. Utiliza notación basada en UML para aplicaciones web en general y para aplicaciones adaptativas en particular. La metodología consta de seis modelos:

- Modelo de casos de uso para capturar los requisitos del sistema.
- Modelo conceptual para el contenido (modelo del dominio).
- Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.
- Modelo de estructura de presentación, modelo de flujo de presentación.
- Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
- Modelo de adaptación.

Para la implementación de este sistema web se utilizará las siguientes herramientas tecnológicas:

- Plataforma: Windows
- Lenguaje de programación: PHP versión 5.6 o superior
- Sistema de gestor de Base de Datos: MySQL versión 5.6 o superior
- Herramientas de maquetación: HTML5, CSS3, Javascript

## 1.7 HERRAMIENTAS

Las herramientas que se utilizarán para el desarrollo del presente sistema, se describen a continuación; observando las características que presentan cada lenguaje y gestor de base de datos considerando los datos más relevantes.

### ○ PHP

PHP (acronimo de "PHP: Hypertext Preprocessor") es un lenguaje "open source" interpretado de alto nivel, desarrollado en páginas HTML y ejecutado en el servidor.

Es un acrónimo recursivo que significa PHP Hypertext Preprocessor, es un lenguaje de programación interpretado de alto nivel al lado del servidor para internet, muy similar en su sintaxis al lenguaje C. con algunas diferencias, no compila como al igual que C, ya que es un intérprete, por tanto, cada vez que se debe ejecutar un programa, lo interpreta verificando toda su sintaxis.

### ○ MARIADB

Las bases de datos también conocida como bancos de datos no son necesariamente digitales.

Una biblioteca es una base de datos compuesta por documentos y textos impresos almacenados en estanterías e indexados y catalogados para su consulta.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius (fundador de MySQL), la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria que reemplaza con ventajas a MyISAM y otro llamado XtraDB en sustitución de InnoDB. Tiene

una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, APIs y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Este SGBD surge a raíz de la compra de Sun Microsystems compañía que había comprado previamente MySQL AB por parte de Oracle. MariaDB es un fork directo de MySQL que asegura la existencia de una versión de este producto con licencia GPL.

Monty decidió crear esta variante porque estaba convencido de que el único interés de Oracle en MySQL era reducir la competencia que MySQL suponía para el mayor vendedor de bases de datos relacionales del mundo, que es Oracle.

Para el desarrollo del proyecto se eligió Gestor de Base de Datos MariaDB por las características que conlleva este Gestor de Base de Datos como Multiplataforma, seguridad, accesibilidad entre otros.

### ○ jQuery

jQuery, en su núcleo, es una biblioteca de manipulación de Modelo de Objetos del Documento (DOM). El DOM es una representación en estructura de árbol de todos los elementos de una página web. JQuery simplifica la sintaxis para buscar, seleccionar y manipular estos elementos DOM. Por ejemplo, JQuery se puede usar para encontrar un elemento en el documento con una determinada propiedad (por ejemplo, todos los elementos con una etiqueta h1), cambiar uno o más de sus atributos (por ejemplo, color, visibilidad) o hacer que responda a un evento (por ejemplo, un clic del ratón).

jQuery también proporciona un paradigma para el manejo de eventos que va más allá de la selección y manipulación básicas de elementos DOM. La asignación de eventos y la definición de la función de devolución de llamada de eventos se realizan en un solo paso en una única ubicación en el código. JQuery también pretende incorporar otra funcionalidad de JavaScript muy utilizada (por ejemplo, entradas y salidas de fundido cuando se ocultan elementos, animaciones mediante la manipulación de las propiedades de CSS).

Los principios de desarrollo con JQuery son:

La separación de JavaScript y HTML: La biblioteca jQuery proporciona la sintaxis simple para añadir eventos manipuladores a la DOM utilizando JavaScript, en lugar de añadir eventos atributos HTML para llamar a funciones de JavaScript. Por lo tanto, alienta a los desarrolladores a separar completamente el código JavaScript del marcado HTML.

Brevidad y claridad: JQuery promueve la brevedad y la claridad con funciones como funciones "chainable" y nombres de funciones abreviados.

Eliminación de incompatibilidades entre navegadores: Los motores de JavaScript de diferentes navegadores difieren ligeramente, por lo que el código JavaScript que funciona para un navegador puede no funcionar para otro. Al igual que otros kits de herramientas de JavaScript, jQuery maneja todas estas inconsistencias en varios navegadores y proporciona una interfaz consistente que funciona en diferentes navegadores.

Extensibilidad: Los nuevos eventos, elementos y métodos pueden agregarse fácilmente y luego reutilizarse como un complemento.

- **Ajax**

Significa JavaScript asíncrono y XML (Asynchronous JavaScript and XML). Es un conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano. Espera, ¿qué es AJAX de nuevo? Vamos a revisar cada término por separado.

JavaScript es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. XML es otra variante de un lenguaje de marcado como HTML, como lo sugiere su nombre: eXtensible Markup Language. Mientras HTML está diseñado para mostrar datos, XML está diseñado para contener y transportar datos.

Tanto JavaScript como XML funcionan de forma asíncrona en AJAX. Como resultado, cualquier aplicación web que use AJAX puede enviar y recuperar datos del servidor sin la necesidad de volver a cargar toda la página.

- **Bootstrap**

Es un framework originalmente creado por Twitter, que nos permite crear interfaces web. Ofrece estilos ya definidos par en diseño de páginas web de una forma rápida, limpia, intuitiva y sencilla. Su característica principal es que soporta diseños sensibles o también conocidos como diseños responsivos. Esto significa que el diseño gráfico de la página se ajusta dinámicamente. Tomando en cuenta las características del dispositivo usado.

- **HTTP Apache**

Es un software libre servidor ATTP de código abierto para plataformas Unix (BSD, GNU/Linux,etc.) Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio visual y nos sirve como servidor de páginas web.

- **JavaScript (JS)**

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

## **1.8 LÍMITES Y ALCANCES**

### **1.8.1 Limites**

El Sistema Web tiene los siguientes límites:

- El sistema no registra la ingreso y salida del personal de la empresa.
- El Sistema no estará disponible para los clientes, solo tendrán acceso el personal de la empresa.
- El sistema no realizará pedidos en línea.
- El sistema no realiza pagos en línea.
- El sistema no estará enlazado al sistema de impuestos.

### **1.8.2 Alcances**

Basado en las principales necesidades de la empresa, el Sistema cuenta con los siguientes módulos y funcionalidades:

### **1.8.2.1 Módulos**

- ✓ Módulo de administración para la gestión de usuarios, roles, accesos del Sistema.
- ✓ Módulo de gestión de clientes y servicios, para registrar, modificar, actualizar los datos, para verificar el detalle del servicio.
- ✓ Módulo de reportes para informes diarios, semanales mensuales y anuales.
- ✓ Módulo de información en el sitio web (marketing).

### **1.8.2.2 Funcionalidades**

- ✓ Funcionalidad responsiva, para que la interfaz del Sistema sea agradable en los dispositivos móviles de los usuarios.
- ✓ Funcionalidad de seguimiento de clientes, que permita ver la información de estos y hacer un seguimiento a los servicios que tienen.
- ✓ Funcionalidad de vista e impresión, para poder entregar información detallada de los servicios adquiridos a los clientes que requieran.
- ✓ Funcionalidad de control de visitas, permitirá que el vendedor asigne una fecha para poder revisar el reporte del cliente, con el fin de no perjudicar el desarrollo de sus actividades.

## **1.9 APORTES**

Los aportes que ofrecerá el siguiente proyecto será automatizar sus procesos habituales de la administración de la empresa minimizando y optimizando el costo y tiempo del manejo de la información asimismo coadyuve en una mejor comunicación y a un mejor seguimiento de los servicios con los clientes.

## 2 CAPITULO II

### 2. MARCO TEORICO

#### 2.1 INTRODUCCION

En el desarrollo de este capítulo se describe la teoría sobre las metodologías, técnicas y herramientas que se utiliza para el desarrollo del sistema web de seguimiento y control administrativo, de la información de clientes para la empresa “SERVIALINUT”, sin embargo, no contempla la teoría completa, pero contiene los conceptos más importantes para que pueda aplicarse y despejar cualquier duda.

#### 2.2. MARCO TEÓRICO

El marco teórico también llamado marco referencial y a veces con un sentido más restringido marco conceptual, tiene como propósito dotar al proyecto de investigación de un sistema coherente de conceptos y proposiciones que permitan abordar con propiedad las diferentes derivaciones correspondientes al planteamiento del problema. Cuando el investigador procede a la elaboración del marco teórico trata de manera integral al problema de Investigación dentro de un ámbito donde este cobre sentido, incorporando los conocimientos previos referentes al mismo.

**Es decir:**

**Figura 2.1: Marco Teórico**



## **2.3. INGENIERÍA DE SOFTWARE**

Existe una variedad de definiciones para ingeniería de software, entre tantas se mencionan las más destacadas.

- Esa forma de ingeniería que aplica los principios de la informática y las matemáticas para conseguir soluciones rentables a problemas software (SEI, 1990).
- El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales (Bauer, 1969).

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos, más conocidos como softwares.

Para nosotros, este servicio trasciende a la programación, que es la base para crear una aplicación. La ingeniería de software engloba toda la gestión de un proyecto. Desde el análisis previo de la situación, el planteamiento del diseño hasta su implementación, pasando por las pruebas recurrentes para su correcto funcionamiento. Podríamos decir que la ingeniería del software es el continente donde se aloja el contenido, que sería el software en sí.

### **2.3.1 Metodologías de desarrollo de software**

Las metodologías de desarrollo de software en ingeniería de software, son marco de trabajo usados para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Una metodología de desarrollo de software se refiere aun framework (marco de trabajo) que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

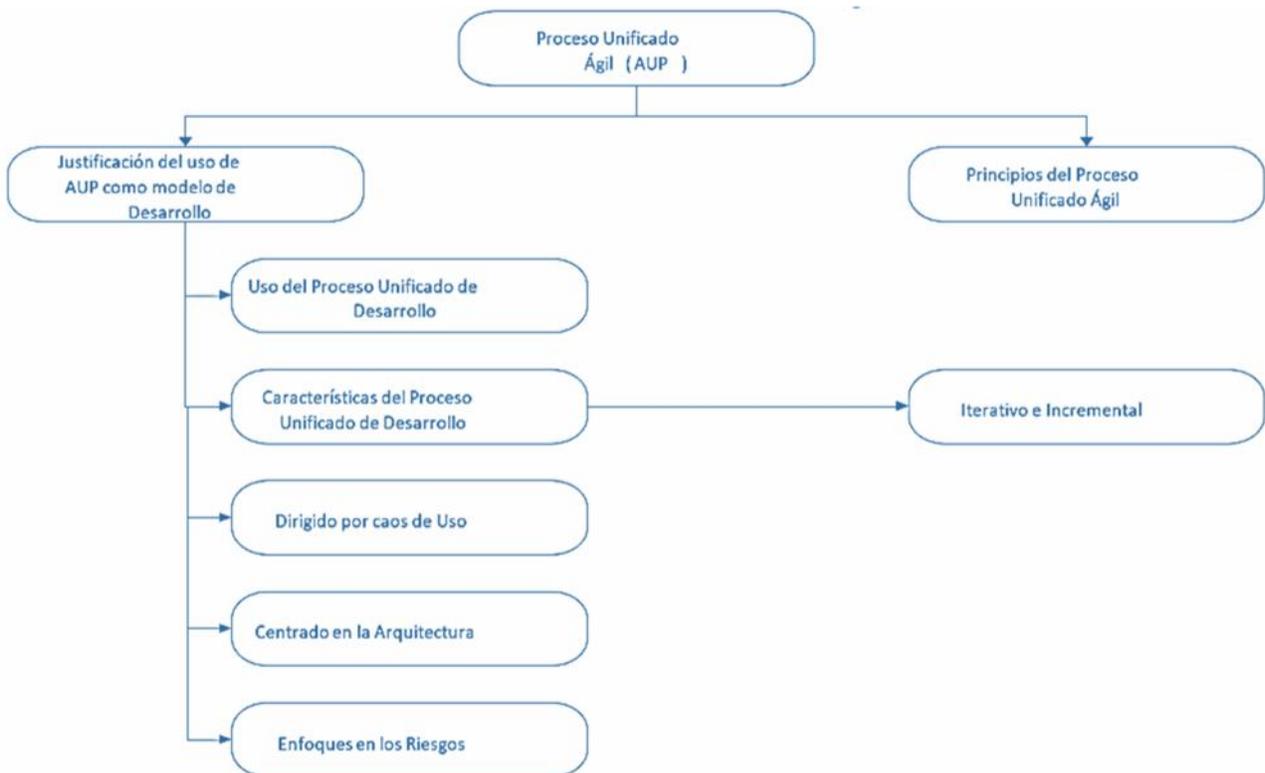
A lo largo del tiempo una gran cantidad de métodos han sido desarrollados diferenciándose por sus fortalezas y debilidades. El framework para metodologías de desarrollo de software consiste:

- Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software.
- Herramientas, modelos y métodos para asistir al proceso de desarrollo de software. Estos framework son a menudo vinculados a algún tipo de organización, que además desarrollan, apoyan el uso y promueven la metodología, además es a menudo documentada en algún tipo de documentación formal.

### 2.3.2. Proceso unificado ágil (AUP)

Para un tener un mejor panorama de la metodología se propone un esquema gráfico, mismo que nos servirá de guía a métrica, que desarrollemos las definiciones puntuales de la metodología AUP. (Proceso Unificado Ágil). El esquema gráfico fue cuidadosamente elaborado, y se espera que sea un aporte o referencia de futuras investigaciones. A continuación, se presenta el esquema gráfico de la metodología AUP con los puntos más importantes seleccionados de acuerdo a los requerimientos de estudio del proyecto de grado propuesto.

**Figura 2.2: Esquema Proceso Unificado Ágil**



Los procesos ágiles de desarrollo de software, conocidos anteriormente como metodologías livianas intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto.

Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración; el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos Ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamientos".

La oficina debe incluir revisores, diseñadores de iteración, escritores de documentación y directores de proyecto, los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. El proceso unificado ágil de Scott Ambler o (Agile Unified Process AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP) este se describe de una manera simple de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen validos en (RUP).

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 riesgos desde etapas iniciales del proyecto.

Donde se demuestre la validez de la arquitectura para los requisitos claves del sistema y que determinan los riesgos técnicos.

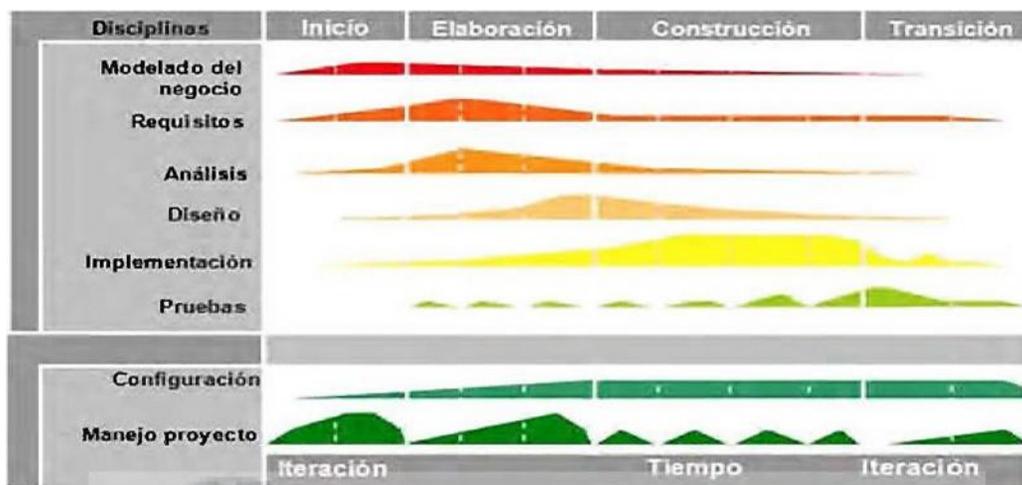
El proceso AUP establece un modelo más simple que el que aparece en RUP por lo que reúne una única disciplina de las disciplinas de modelado de negocio, requisitos, análisis y diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y entorno) coinciden con las restantes de RUP.

### 2.3.2.1. Ciclo de vida del proceso unificado ágil (AUP)

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados.

- Inicio: El objetivo de esta fase es obtener comprensión común cliente equipo de desarrollo y definir una o varias arquitecturas para el mismo.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.
- Transición: El sistema se lleva a los entornos de administración donde se somete a pruebas de validación, aceptación y finalmente se despliega el sistema.

**Figura 2.3: (Ciclo de Vida AUP) Fuente-Scott Ambler o (Agüe Inified Process AUP)**



Las disciplinas se llevan a cabo de manera sistemática, a las definiciones de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar y entregar el software de trabajo que responda a las necesidades de sus interlocutores.

- Modelados: El objetivo de esta es entender el negocio de la organización, el problema del dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.
- Aplicación: El objetivo de esta es transformar su modelo en código ejecutable y realizar un nivel básico de las pruebas.
- Prueba: El objetivo de esta consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona como tal como está establecido, verificando que se cumplan los requisitos.
- Despliegue: El objetivo de esta es la presentación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.
- Gestión de Configuración: El objetivo de esta es la gestión de acceso a herramientas de su proyecto. Esto incluye no solo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
- Gestión de Proyecto: El objetivo de esta es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas el seguimiento de los progresos, etc.), coordinación con el personal los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro de lo propuesto.
- Entono: El objetivo de esta disciplina es apoyar el resto de los esfuerzos para garantizar que el proceso sea el adecuado la orientación (normas y directrices), y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

### **2.3.3. JUSTIFICACIÓN DEL USO DEL PROCESO UNIFICADO ÁGIL COMO MODELO DE DESARROLLO**

Un modelo de ciclo de vida de software es una vista de las actividades que se llevan a cabo durante el desarrollo de éste, e intenta determinar el orden de las etapas involucradas y proporcionar criterios para avanzar de unas a otras.

Por tanto, definir un ciclo de vida permite llevar un mayor control sobre las tareas, evitando que estas se vayan eligiendo y realizando de manera desordenada, según parezca que vayan surgiendo necesidades, que podrían ser puntuales y fácilmente evitables.

#### **2.3.3.1. Uso del proceso unificado de desarrollo**

Debido al carácter relativamente investigador de este proyecto, y la necesidad de modificar los requisitos que surgirán según se fueran evaluando y probando las distintas posibilidades con las que se cuenta para desarrollarlo, un modelo pesado no se ajusta de manera adecuada.

Sin embargo, un modelo puramente ágil necesita de un equipo de desarrollo con experiencia para ser llevado a cabo de manera satisfactoria, por lo que este tampoco es el caso más adecuado para su aplicación. Es por ello que se optó por un modelo que combina características: el Proceso Unificado de Desarrollo propuesto por Rumbaugh, Booch y Jacobson.

#### **2.3.3.2. Características del proceso unificado de desarrollo**

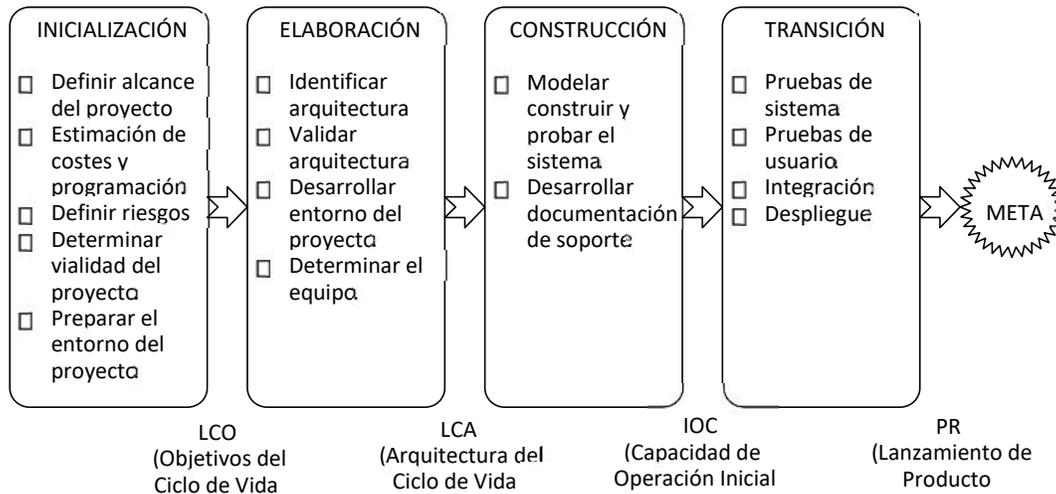
Al igual que con cualquier otro modelo de desarrollo, del Proceso Unificado, también se pueden destacar ciertas características.

) Iterativo e Incremental.

El proceso unificado es un marco de desarrollo compuesto por cuatro fases:

- Inicio
- Elaboración
- Construcción
- Transición

**Figura 2.4: Fases del Proceso Unificado Ágil**



Cada una de ellas es, a su vez, dividida en una serie de iteraciones que ofrecen como resultado un incremento del servicio, que añade o mejora las funcionalidades del sistema en desarrollo. Es decir, un "incremento" se implica necesariamente una ampliación de dicho sistema. Durante cada una de estas iteraciones se realizarán a su vez las actividades definidas en el ciclo de vida clásico: requisitos, análisis, diseño, implementación y las pruebas. Si una iteración cumple sus metas, publicando una nueva versión del servicio que triple ciertos casos de uso, el desarrollo continúa con las siguientes.

Cuando no las cumple, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque. Dirigido por los Casos de Uso.

Un sistema de tipo software se crea para servir a sus usuarios por lo que, para construir un sistema exitoso, se debe conocer qué es lo que se quieren y necesitan. El término "Usuario" no se refiere solamente a los usuarios humanos sino también a otros sistemas, es decir, representa a algo o alguien que interactúa con el sistema a desarrollar. En el proceso unificado, los casos de uso se utilizan para capturar los requisitos funcionales y para definir los objetivos de las iteraciones. En cada una, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño de uso.

➤ Centrado en la Arquitectura.

El concepto de arquitectura de software involucra los aspectos estáticos más significativos del sistema, y actúa como vista de diseño, dando una perspectiva completa y describiendo los elementos más importantes. La arquitectura surge de los propios casos de uso, sin embargo, también está influenciada por muchos otros factores, como la plataforma en la que se ejecutará, el uso de estándares, la existencia de sistemas heredados (aunque éste no sea el caso que nos ocupa) o los requerimientos no funcionales.

Puesto que la arquitectura y los casos de uso están relacionados, por una parte, cuando son realizados, acomodarse en la arquitectura, y ésta debe ser lo bastante flexible para realizar todos los casos de uso en el futuro. De palabras de los propios creadores del Proceso Unificado, es un problema semejante al del "huevo y la gallina", en la realidad, arquitectura y casos de uso deben evolucionar en paralelo.

➤ Enfocado en los Riesgos.

Para disminuir la posibilidad de fallos en las iteraciones o incluso la de cancelación del proyecto se deben llevar a cabo sucesivos análisis de riesgos durante todo el desarrollo.

Por supuesto, los riesgos principales deben ser identificados en una etapa temprana del ciclo de vida, y además, los resultados de cada iteración deben seleccionarse en un orden que asegure que estos son considerados primero.

#### **2.3.4. PRINCIPIOS DEL PROCESO UNIFICADO ÁGIL**

El proceso unificado ágil (AUP), es ágil porque está basada en los siguientes principios:

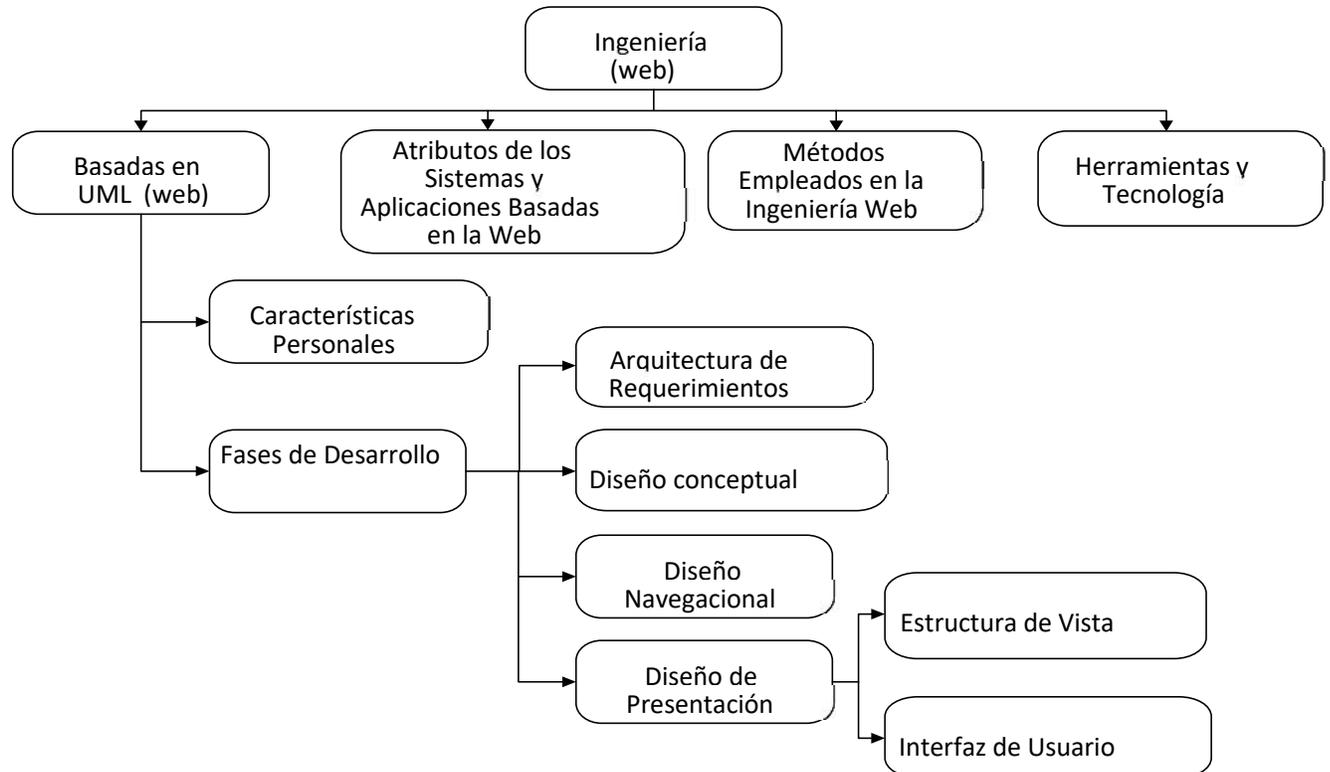
- El personal sabe lo que está haciendo. La gente no va a leer detalladamente el proceso de documentación, pero algunos quieren una orientación de alto nivel y / o formación de vez en cuando. El AUP producto proporciona enlaces a muchos de los detalles, si usted está interesado, pero no obliga a aquellos que no lo deseen.
- Simplicidad. Todo se describe concisamente utilizando un puñado de páginas, no miles de ellos.
- Agilidad. Ágil ARRIBA. El ajuste a los valores y principios de la Alianza Ágil.

- Centrarse en actividades de alto valor. La atención se centra en las actividades que se ve que son esenciales para el desarrollo, no todas las actividades que suceden forman parte del proyecto.
- Herramientas de la independencia. Usted puede usar cualquier conjunto de herramientas que usted desea como el AUP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.
- Adaptación de este producto para satisfacer sus propias necesidades. El AUP producto es de fácil acomodo común a través de cualquier herramienta de edición de HTML. No se necesita comprar una herramienta especial, o tomar un curso, para adaptar el AUP.

## 2.4. INGENIERÍA WEB

Antes de abordar el desarrollo se presenta la estructura gráfica propuesta para un mejor entendimiento y comprensión de la Ingeniería Web.

**Figura 2.5: Esquema de Desarrollo-Ingeniería Web**



La ingeniería Web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operaciones y evolución de aplicaciones de alta calidad en la World Wide Web. La ingeniería Web se debe al crecimiento desenfrenado que está teniendo la Web ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que personas tiendan a realizar todas sus actividades por esta vía. En esta sección del proyecto debido a su impacto e importancia en la Web se seguirán los lineamientos de (Pressman Vol. 6, 2007) destacando los puntos más importantes y que tengan relación con el proyecto de grado.

La World Wide Web y el internet que la alimentan son, posiblemente los desarrollos más importantes en la historia de la computación. Estas tecnologías han llevado a todos (con cientos de millones más que eventualmente seguirán) a la era de la informática; además se han convertido en parte integral de la vida diaria en la primera década del siglo XXI. Para quienes pueden recordar un mundo sin la Web, el crecimiento caótico de la tecnología tiene su origen en la otra era los primeros días del software. Era una época de poca disciplina pero enorme entusiasmo y creatividad. Eran tiempos en que los programadores a menudo ingresaban a sistemas en conjunto, a veces para bien, a veces para mal. La actitud prevaleciente parecía ser: "hazlo rápido y entra en el campo; nosotros lo limpiaremos (y mejor entiende qué lo que realmente se necesita construir) conforme avancemos". Suena familiar

Esto conduce a una pregunta clave: ¿se puede aplicar principios, conceptos y métodos de la ingeniería de software al desarrollo Web? Es posible aprovechar muchos de ellos, pero su aplicación puede requerir un giro un tanto diferente.

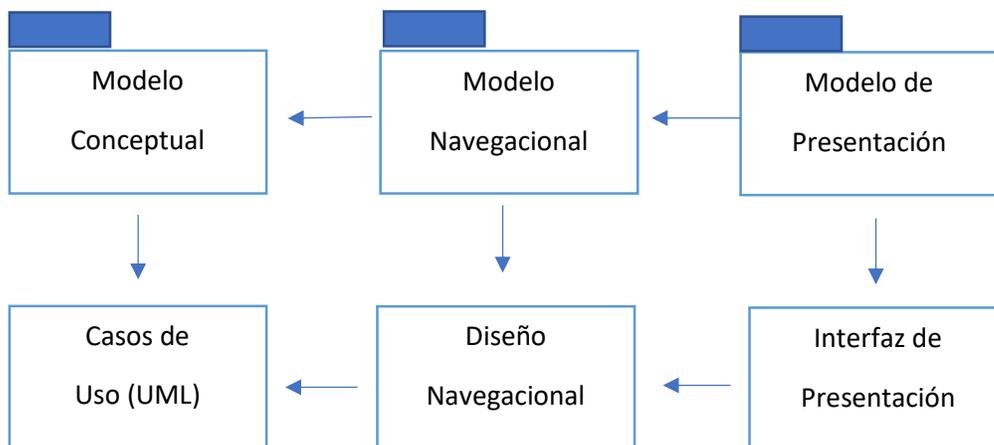
Pero qué ocurre si persiste un enfoque sin disciplina respecto al desarrollo Web, en ausencia de un proceso disciplinado dirigido a desarrollar sistemas basados en la Web, existe una creciente preocupación que se enfrenten serios problemas en su desarrollado, despliegue y mantenimientos exitosos. En esencia, la infraestructura de aplicación que se está creando en la actualidad puede conducir a una "Web enmarañada" conforme se adentra más en este nuevo siglo. Esta frase entraña un cúmulo de aplicaciones basadas en la Web mal desarrolladas y que tienen muy altas probabilidades de fracaso. Peor aún,

conforme los sistemas basados en la Web crecen con mayor complejidad, una falla en uno puede propagar y propagará amplios problemas por medio de muchos. Cuando esto ocurra, la confianza en todo el internet será sacudida. Peor aún, podría conducir a una regulación gubernamental innecesaria y mal concebida, lo que provocaría en daño irreparable a estas tecnologías únicas. Para evitar una Web enmarañada y lograr mayor éxito en el desarrollo y las aplicaciones de sistemas basados en la Web complejos y a gran escala, existe una apremiante necesidad de enfoques disciplinados, nuevos métodos, herramientas con que desarrollar, desplegar, evaluar los sistemas y aplicaciones basadas en la Web. Tales enfoques y técnicas deben considerar las características especiales de los nuevos medios, los ambientes y escenarios operativos, y la multiplicidad de perfiles de usuario que colocan desafíos adicionales al desarrollo de aplicaciones, basadas en la Web.

#### 2.4.1. Basados en UML (UWE)

UWE (UML Web Engineering, en español Ingeniería Web Basada en UML) es Una metodología que permite especificar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones detalla ésta, con una gran cantidad de definiciones, en el proceso de diseño lista qué debe utilizarse. Procede de manera iterativa e incremental, Coincidiendo con UML, incluyendo flujos de trabajo y puntos de control

**Figura 2.6: Ingeniería Web Basada en UML**



UWE se especializa en la especificación de aplicaciones que se adaptan, y por eso hace énfasis especial en las características de personalización, y la definición de los modelos de usuario o en un patrón de características de navegación basados en preferencias, tareas o conocimientos. Otros aspectos de interés de la metodología UWE es la orientación a objetos, usuarios y la definición de un modelo de referencia que da soporte a la metodología y formaliza los' modelos por el grado de restricciones y definiciones que proporciona. UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimientos o tareas de usuario. Los principales aspectos en los que se fundamenta UWE son los siguientes:

) Uso de Una Notación Estándar.

Para todos los modelos (UML: Lenguaje de Modelo Unificado)

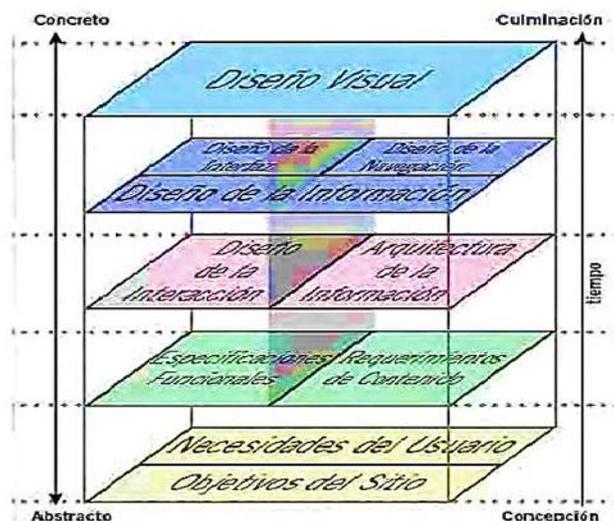
) Definición de Métodos.

Definición de pasos para la construcción de los diferentes modelos.

) Especificación de Restricciones.

Se recomienda el uso de restricciones estrictas.

**Figura 2.7: (IWeb Basado en UML) fuente Jesse James Garret 2000**



### 2.4.1.1. CARACTERÍSTICAS PRINCIPALES

La metodología UWE define vistas especiales, representadas gráficamente por diagramas UML, tales como el modelo de navegación y el modelo de representación.

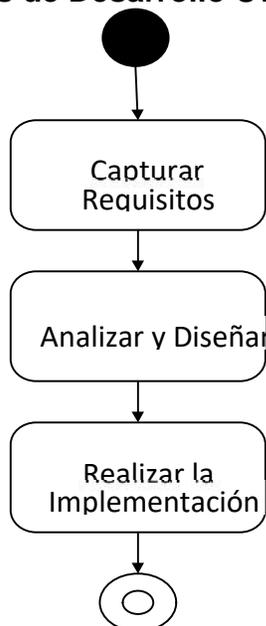
Los diagramas se pueden adaptar como mecanismos de extensión basados en estereotipos que proporciona UML. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada para el dominio específico a la que se conoce como "Perfil UML".

Un perfil de UML consiste en una jerarquía de estereotipos y un conjunto de restricciones. Los estereotipos son utilizados para representar instancias de las clases. Las ventajas de utilizar los perfiles de UML es que casi todas las herramientas CASE de UML los reconocen. Los modelos deben ser fácilmente adaptables al cambio en cualquier etapa del desarrollo.

### 2.4.1.2. FASES DE DESARROLLO (UWE)

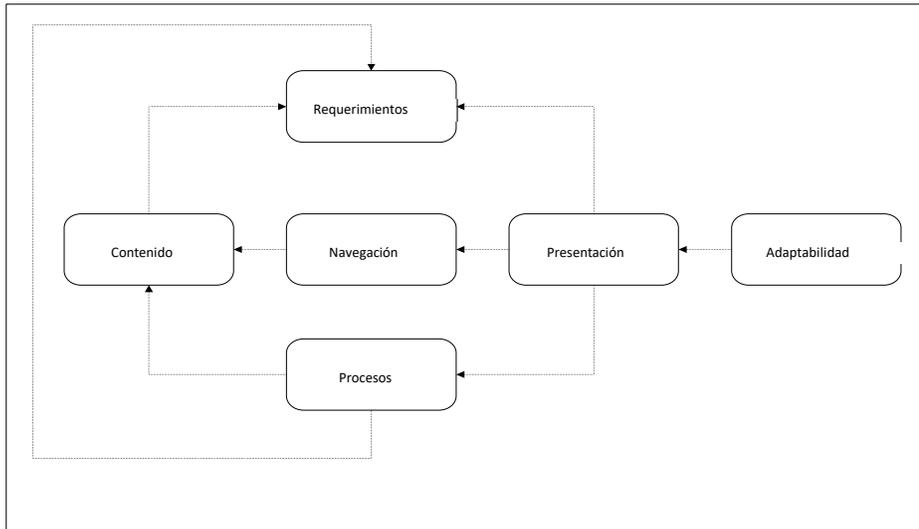
Con respecto al proceso de creación de la aplicación, UWE se vale mediante el uso de metodologías estándares reconocidas como UML principalmente y también del lenguaje de especificaciones de restricciones asociados OCL (Object Constraint Language, en español lenguaje de restricciones para objetos).

**Figura 2.8: (Fases de Desarrollo UWE) Fuente-Nora Koch**



Para recolectar los requerimientos necesarios de las aplicaciones Web, esta metodología propone una ampliación utilizada en el proceso de creación, mismo que se divide en las siguientes cuatro actividades:

**Figura 2.9: (Fases del UWE) Fuente-Nora Koch**



### ) **Análisis de Requisitos.**

Plasma los requerimientos funcionales de la aplicación Web, mediante modelos de casos de uso.

### ) **Diseño Conceptual.**

Se define mediante un modelo de dominio, considerando los requisitos plasmados en los casos de uso, el diagrama de clases representará los conceptos con un gran porcentaje de detalle.

### ) **Diseño Navegacional.**

Comprende la construcción del modelo de navegación en dos pasos.

- **MODELOS DE ESPACIO NAVEGACIONAL:** Su objetivo es especificar qué objetos pueden ser visitados a través de la aplicación.

- **MODELO DE ESTRUCTURA NAVEGACIONAL:** Amplía el modelo con un conjunto de estructuras de acceso necesarias para la navegación como los índices, consultas y visitas guiadas.

) **Diseño de Presentación.**

Permite la especificación lógica de la aplicación Web. Basada sobre este modelo lógico, una representación física puede ser contribuida. Representa las interfaces del usuario por medio de vistas estándares de integración UML. Dentro de este modelo se distinguen dos diferentes vistas:

- **ESTRUCTURA DE VISTA:** muestra la estructura del espacio de presentación.
- **INTERFAZ DE USUARIO (UI POR SUS SIGLAS EN INGLÉS DE USER INTERFACE):**

#### **2.4.2. Atributos de los sistemas y aplicaciones basados en la web**

En los primeros días de la World Wide Web (1990 a 1995) los "sitios Web" consistían en poco más que de un conjunto de archivos de hipertexto ligados, que presentaban información mediante texto y gráficos limitados. Conforme el tiempo pasó, el HTML aumentó al desarrollar herramientas (por ejemplo XML, Java) que permitieron a los ingenieros Web ofrecer capacidades de cálculo junto con información. Nacieron los sistemas y aplicaciones basados en la Web (se les refería de manera colectiva como Web Apps). En la actualidad, las Web Apps han evolucionado en sofisticadas herramientas de computación que no sólo proporcionan función por si misma al usuario final, sino que también se han integrado con bases de datos corporativas y aplicaciones de negocios.

Existe poco debate en cuanto a que las Web Apps son diferentes a las muchas otras categorías de software informático. Powell resume las diferencias principales cuando establece que los sistemas basados en la Web "involucran una mezcla entre publicación impresa y desarrollo de software, entre marketing e informática, entre comunicaciones internas, relaciones externas, arte, además de la tecnología". En la gran mayoría de las Web Apps se encuentran los siguientes atributos.

- **INTENSIDAD DE RED.** Una Web Apps reside en una red y debe satisfacer las necesidades de una variada comunidad de clientes. Una Web Apps puede residir en el internet (y, en consecuencia, permite una comunicación mundial abierta).
- **CONCURRENCIA.** Un gran número de usuarios puede tener acceso a las Web Apps al mismo tiempo. En muchos casos, los patrones de uso entre los usuarios finales variaran enormemente.
- **DESEMPEÑO.** Si un usuario de las Web Apps debe esperar demasiado (para ingresar, procesamiento en el lado del servidor, formateo y despliegue en el lado del cliente) puede decidir irse a cualquier otra parte.
- **GOBERNADA POR LOS DATOS.** La función primordial de muchas Web Apps es usar hipermedia para presentar contenidos de texto, gráfico, audio y video al usuario final. Además, por lo general, las Web Apps se utilizan para tener acceso a información que existe en base de datos que originalmente no será parte integral del ambiente basado en la Web (por ejemplo, comercio electrónico o aplicaciones financieras).
- **EVOLUCIÓN CONTINÚA.** A diferencia del software de aplicación convencional, que evoluciona a lo largo de una serie de planeadas espaciadas cronológicamente, las aplicaciones Web evolucionan de manera continua, no es raro que algunas Web Apps (específicamente, su contenido) se actualicen sobre una agenda minuto a minuto, o que el contenido sea calculado de manera independiente para cada solicitud. Algunos argumentan que la evolución continua de las Web Apps hace que el trabajo realizado sobre ellas sea análogo.
- **SEGURIDAD.** Puesto que las Web Apps están disponibles mediante el acceso a la red, es difícil, si no imposible, limitar la población de usuarios finales que pueden tener acceso a la aplicación. Con la finalidad de proteger el contenido confidencial y ofrecer modos seguros de transmisión de datos, se deben implementar fuertes medidas de seguridad a lo largo de la infraestructura que sustenta una Web Apps y dentro de la aplicación.
- **ESTÉTICA.** Una parte innegable de la apariencia de una Web Apps es su presentación y la disponibilidad de sus elementos. Cuando una aplicación se diseña para comercializar

o vender productos o ideas, la estética puede tener tanto que ver con el éxito como en diseño técnico.

Estos atributos generalmente se aplican a todas las Web Apps, pero con diferentes grados de influencia. ¿Pero qué hay de las Web Apps por ellas mismas? ¿Qué Problema abordan? En trabajo IWeb es usual encontrar las siguientes categorías de aplicaciones.

- **INFORMATIVO:** se proporciona contenido de solo lectura con navegación y enlaces simples.
- **PERSONALIZABLE:** la comunicación entre una comunidad de usuarios ocurre por medio de tableros de anuncios o mensajería instantánea.
- **ENTRADAS DE USUARIO:** la entrada con base en formularios es el principal mecanismo para las necesidades de comunicación.
- **ORIENTADA A TRANSACCIONES:** el usuario hace una solicitud (por ejemplo, realiza un pedido) que ejecuta la Web Apps.
- **ORIENTADA A SERVICIOS:** la aplicación proporciona un servicio al usuario; por ejemplo, lo asesora en la determinación del pago de una hipoteca.
- **ACCESO A UNA BASE DE DATOS:** el usuario consulta una gran base de datos, extrae información.
- **ALMACÉN DE DATOS.** El usuario consulta una colección de grandes bases de datos y extrae información.

Los atributos comentados en esta sección, las categorías de aplicaciones destacadas líneas arriba, representan importantes hechos de vida para los ingenieros Web. El éxito radica en vivir dentro de las restricciones que imponen dichos atributos y aun así producir una Web Apps exitosa.

### **2.4.3. Métodos empleados en ingeniería web**

El panorama de los métodos de Web abarca un conjunto de labores técnicas que permiten al ingeniero Web comprender, características y luego construir una Web Apps de alta calidad.

#### **• MÉTODOS DE COMUNICACIÓN.**

Definen el enfoque con que facilita la comunicación entre ingenieros Web y los demás participantes de la Web Apps (por ejemplo, usuarios finales, clientes de negocios, expertos en problemas de dominio, diseñadores de contenido, líderes de equipo, gestores de proyecto). Las técnicas de comunicación son particularmente importantes durante la recolección de requisitos y siempre que sea elevado un incremento en la Web Apps.

### **2.4.4. Herramientas y tecnologías**

Las herramientas que se utilizarán en el proyecto serán los siguientes:

#### **2.4.4.1. Php**

PHP acrónimo de Hypertext Preprocessor es un lenguaje de programación del lado del servidor diseñado originalmente para la generación de páginas web dinámicas. Es un lenguaje de programación interpretado o de script que permite insertar fragmentos de código dentro de una página HTML y realizar determinadas acciones de forma fácil y eficiente.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en una herramienta ideal al momento de crear una página web dinámica:

- ) Fácil de mantener y mejorar el código desarrollado.
- ) Soporte para una gran cantidad de base de datos como MySQL, PostgreSQL, Oracle, MariaDB, MS SQL Server, entre otras.
- ) Facilidad de integración con varias bibliotecas externas utilizando Composer.
- ) Amplia comunidad de desarrolladores, por ser de código abierto, goza del aporte de gran número de programadores.

) Permite aplicar técnicas de programación orientada a objetos.

#### **2.4.4.2 WordPress**

WordPress es un sistema de gestión de contenidos web (CMS o content management system), que en pocas palabras es un sistema para publicar contenido en la web de forma sencilla.

Es un software de código abierto (se puede tener acceso a todo el código) que además podemos tratar de mejorar dentro de su comunidad.

WordPress llegó para democratizar la web, como otros CMS. Desde el año 2003, es un sistema de gestión de contenidos que hace que la creación de contenido web no dependa sólo de programadores y de personas de alto conocimiento técnico.

#### **WordPress se divide en tres partes:**

**Core:** WordPress en sí, que es absolutamente gratuito y descargable.

**Temas:** que sirven para cambiar la apariencia de la web. Hay un enorme repositorio gratuito, pero también hay recursos de pago fuera del repositorio.

**Plugins:** utilidades que pueden convertir tu web en casi cualquier cosa. Igualmente, que los temas, hay un repositorio gratuito y miles de empresas que venden sus funcionalidades.

#### **2.4.4.3. JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden ser interpretados por los navegadores.

#### **2.4.4.4. JQuery**

Jquery es un framework de JavaScript, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejo de eventos, desarrollar animaciones y agregar interacción con la técnica AJAX en páginas web (Comesaña, 2012).

#### **2.4.4.5. Bootstrap**

Bootstrap es un framework de CSS, de código abierto, en otras palabras, es un conjunto de archivos CSS que se incluye en una página y se puede empezar a maquetar el sitio web en minutos, sin tocar una sola línea de CSS, esto agiliza el desarrollo de aplicaciones web quitándonos de encima toda la carga del diseño y dándonos un diseño elegante y bueno gracias a sus clases ya predefinidas.

#### **2.4.4.6 Gestor de base de datos Mysql**

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

Son muchas las razones para escoger MySQL como solución para la administración de datos (Gilfillan, 2003):

- ) Coste: El coste de MySQL es gratuito para la mayor parte de 10s usos y su servicio de asistencia resulta económico.
- ) Asistencia: Existe una nutrida y activa comunidad MySQL.
- ) Facilidad de uso: MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- ) Velocidad: MySQL es mucho más rápido que la mayor parte de sus rivales.
- ) Funcionalidad: MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplication, funciones SSL e integración con la mayor parte de los entornos de programación.

- ) Portabilidad: MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

#### **2.4.4.7 Ajax**

Es un framework para aplicaciones web que ayuda a desarrollar aplicaciones basadas en AJAX, una tecnología para construir páginas web dinámicas del lado del cliente.

La información es leída desde el servidor o enviada a éste a través de peticiones Javascript. Sin embargo, es requerido algún procesamiento del lado del servidor para manejar peticiones, por ejemplo, para buscar o guardar información.

Esto es alcanzado más fácilmente con el uso de un framework dedicado a procesar peticiones AJAX. En el artículo donde nació el término "Ajax", J. J. Garret describe esta tecnología como "un intermediario... entre el usuario y el servidor" [1].

Este motor AJAX pretende reducir la espera para el usuario cuando una página trata de acceder al servidor. La meta del framework es proveer este motor AJAX y funciones asociadas al servidor y del lado del cliente.

### **2.5 METODO DE PRUEBAS DE SOFTWARE**

El desarrollo de software implica la realización de una serie de actividades predispuestas a incorporar errores.

Hay una de dos maneras de probar cualquier producto construido la primera es si se conoce la función específica para la que se diseñó el producto, se aplican pruebas, que demuestren que cada función es plenamente operacional, mientras se buscan los errores en cada función; la segunda si se conoce el funcionamiento interno del producto, se aplican pruebas para asegurarse de que "todas las piezas encajan"; es decir, que las operaciones internas se realizan de acuerdo a las especificaciones, y que se han probado todos los componentes internos de manera adecuada al primer enfoque de prueba se le denomina prueba de caja negra; al segundo, prueba de caja blanca. (Pressman, 2010, pág. 383)

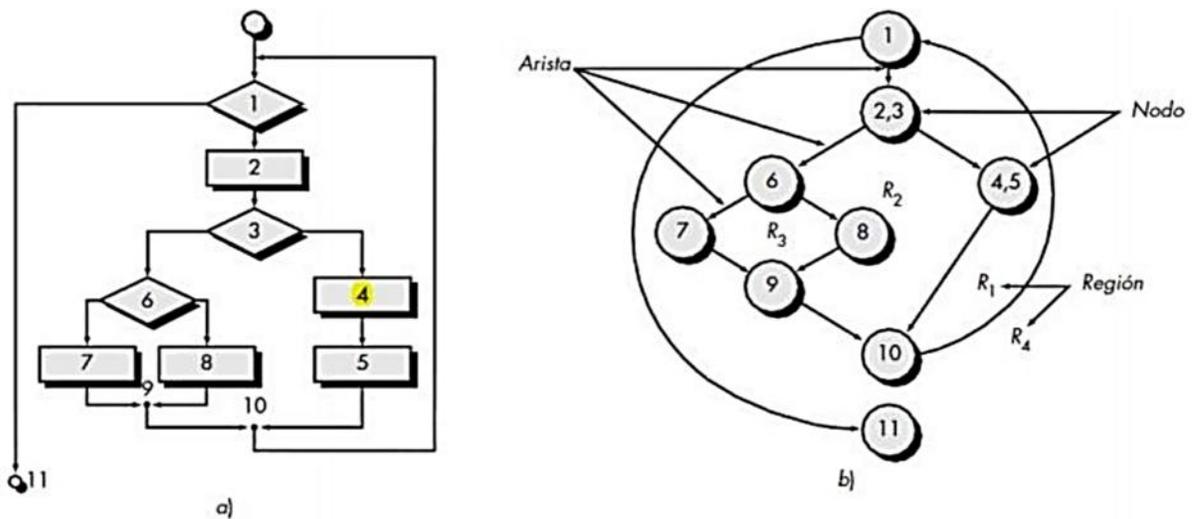
### 2.5.1 Pruebas de Caja Blanca

Este tipo de prueba es un diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de pruebas. Según (Pressman, 2010, pág. 414) la prueba de caja blanca, en ocasiones llamada pruebas de caja de vidrio, es una filosofía de diseño de casos de pruebas que usan las estructuras de control descrita como parte del diseño a nivel de componentes para derivar casos de pruebas.

Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

1. Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
2. Revisen todas las decisiones lógicas en sus lados verdadero y falso.
3. Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras y operativas.
4. Revisen estructuras de datos internas para garantizar su validez.

Figura N.º 2.10 Grafico de Flujo



Fuente: (Pressman S. R., 2010)

Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo

que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones. Ejemplos típicos de ello son las pruebas unitarias. (Marquez, 2018, pág. 8)

## **2.5.2 Pruebas de Software**

Luego de haber desarrollado el sistema, se procede a realizar las pruebas de los procedimientos que se implementaron. Se tratará de encontrar todo posible error durante un proceso antes que se entre en aplicación. (Peño, 2015, pág. 15)

### **2.5.2.1 Pruebas de Caja Blanca**

Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones. Ejemplos típicos de ello son las pruebas unitarias. (Marquez, 2018, pág. 8)

### **2.5.2.2 Pruebas de Caja Negra**

Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

En las pruebas de caja negra, nos enfocamos solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos en los requerimientos de software y especificaciones funcionales. (Torrera, 2017, pág. 2)

## **2.6 METRICAS DE CALIDAD DE SOFTWARE**

### **2.6.1 Calidad de Software**

La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

Es la aptitud de un producto o servicio para satisfacer las necesidades del usuario. Es la cualidad de todos los productos, no solamente de equipos sino también de programas.

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta.

Certificación del software Es consecuencia del proceso de aseguramiento de la calidad, pero nunca es el objetivo final. La calidad de software no se certifica, lo que se certifica, son los procedimientos para construir un software de calidad. Los procedimientos deben ser correctos y estar en función de la normalización (ISO 9000, CMMI, Moprosoft, etc.). (Peña J. M., 2011, pág. 6)

### **2.6.2 Seguridad del Software y Estándares de Calidad**

En aras de garantizar la seguridad del software resulta imprescindible buscar un desarrollo casi perfecto, es por eso que deben tenerse en cuenta factores de calidad.

Uno de ellos no inventar lo que ya está hecho. Los sistemas se basan en patrones similares. Es preciso, entonces, estudiar para no reinventar soluciones a problemas que ya han sido resueltos.

El mecanismo con el que se estructura la información es crucial. En este aspecto la obsesión nunca se queda corta, todo elemento debe estar documentado, espaciado sin fallas y debe ser de rápido entendimiento. (Equipo Editorial, 2019, pág. 81)

### **2.6.3 Métricas de Calidad**

El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del software.

Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad.

El primer objetivo del equipo de proyecto es medir errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad. (Olsina D. L., + 2003, pág. 12)

## 2.7 METODOS DE ESTIMACION DE COSTO DE SOFTWARE

### 2.7.1 Método de Estimación de costo COCOMO

COCOMO permite realizar estimaciones en función del tamaño del software y de un conjunto de factores de coste y de escala. En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo y características propias de proyecto.

Los factores de escala incluyen la parte de escala producida a medida que un proyecto de software incrementa su tamaño. Por un lado, COCOMO define tres modos de desarrollo de software o tipos de proyectos las cuales son las siguientes.

Orgánico. - Proyectos relativamente sencillos, menores de 50 KLDC líneas de código, en los cuales se tiene experiencia de proyectos, se encuentra en entornos estables.

Semi - acoplado. - Proyectos intermedios en complejidad y tamaño, menores a 300 KLDC líneas de código, donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

Empotrado: Proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloba en un entorno de gran innovación técnica, además se trabajan con unos requisitos muy restrictivos y de gran volatilidad. (Pressman R., 2010, pág. 15)

**Tabla 2.1: Detalle de coeficientes de COCOMO**

MODO	Básico				Intermedio			
	a	b	c	d	A	B	C	D
Orgánico	2,4	1,5	2,5	0,8	3,2	1,5	2,5	0,8
Semi-acoplado	3,0	1,2	2,5	0,5	3,0	1,2	2,5	0,5
Empotrado	3,6	1,0	2,5	0,2	2,8	1,0	2,5	0,2

**Fuente: (Pressman, 2010)**

Por otro lado, existen diferentes modelos que define COCOMO y son las siguientes:

- ) Modelo básico. -Se basa exclusivamente en el tamaño expresado en LDC.
- ) Modelo intermedio. -Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- ) Modelo avanzado. -Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Para la realización del COCOMO previamente necesitamos conocer el número de líneas de código, posteriormente para poder realizar los cálculos del método de estimación usamos las siguientes ecuaciones. (Calibrado, 1981, pág. 5)

**Tabla 2.2 Ecuaciones del Método COCOMO**

Variable	Ecuación	Tipo/Unidad
<b>Esfuerzo</b>	$E = a (KLCD)^b * FAE$	Personas/mes
<b>Tiempo</b>	$T = c (E)^d$	Meses
<b>Número de personas</b>	$NP = \frac{E}{T}$	Personas
<b>Costo total</b>	$CT = NP \times T \times \text{sueldo mes}$	\$us.
<b>Costo por líneas de código</b>	$\text{Costo LDC} = \frac{CT}{LDC}$	\$us.

**Fuente: (Calibrado, 1981).**

Por otro lado, también vamos hallar la variable FAE (factor de ajuste del esfuerzo), la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observa en la siguiente tabla:

**Tabla 2.3: Atributos FAE.**

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
<b>Atributos de software</b>						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos de hardware</b>						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
<b>Atributos de personal</b>						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
<b>Atributos del Proyecto</b>						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

**Fuente: (Pressman, 2010)**

## Descripción de Valores

Atributos de software

RELY: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto.

DATA: tamaño de la base de datos en relación con el tamaño del programa.

CPLX: representa la complejidad del producto. Atributos de hardware

TIME: limitaciones en el porcentaje del uso de la CPU.

STOR: limitaciones en el porcentaje del uso de la memoria.

VIRT: volatilidad de la máquina virtual.

TURN: tiempo de respuesta requerido. Atributos del personal

ACAP: calificación de los analistas.

AEXP: experiencia del personal en aplicaciones similares.

PCAP: calificación de los programadores.

VEXP: experiencia del personal en la máquina virtual.

LEXP: experiencia en el lenguaje de programación a usar. Atributos del proyecto

MODP: uso de prácticas modernas de programación.

TOOL: uso de herramientas de desarrollo de software.

SCED: limitaciones en el cumplimiento de la planificación. (2016, pág. 3)

## **2.8 ISO 9126**

La ISO 9126 es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que son medidos por una serie de métricas. Estas métricas miden artefactos obtenidos en etapas tardías del desarrollo de software, aumentando el costo de detección y corrección de errores. Por esta razón, en la literatura ha surgido un mayor interés por la definición de métricas que pretenden evaluar una o varias de las características de calidad definidas en el estándar ISO 9126, en etapas tempranas del desarrollo de software.

El modelo ISO 9126 fue desarrollado por la ISO (organización de estandarización internacional por sus siglas en Ingles ISO) y este es uno de los grandes grupos reconocidos por los estándares aplicados internacionalmente a través de un amplio rango de solicitudes. (2018, pág. 54)

## Funcionabilidad

Conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades. Idoneidad: Se enfoca a evaluar si el SW cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.

- ✓ **Exactitud:** Permite evaluar si el SW presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- ✓ **Interoperabilidad:** Permite evaluar la habilidad del SW de interactuar con otros sistemas previamente especificados.
- ✓ **Seguridad:** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o promediado, a los programas y datos.
- ✓ **Conformidad:** Evalúa si el SW se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares. (Figueroa, 2012, pág. 2)
- ✓ **Confiabilidad:** Conjunto de atributos relacionados con la capacidad de mantener un nivel de presentación bajo condiciones establecidas durante un periodo de tiempo establecido.
- ✓ **Madurez:** Permite medir la frecuencia de falla por errores en el SW.
- ✓ **Recuperación:** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesario para lograrlo.
- ✓ **Tolerancia de fallos:** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del SW o de cometer infracciones de su interfaz específica. (Figueroa, 2012, pág. 3)

## Usabilidad

Conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- ✓ **Comprensión:** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del SW.

- ✓ **Facilidad de Aprender:** Establece atributos del SW relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- ✓ **Operatividad:** Agrupa los conceptos que evalúan la operación y el control del sistema. (Figuroa, 2012, pág. 3)

## **Eficiencia**

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del SW y la cantidad de recursos necesitados bajo condiciones establecidas. Comportamiento en el tiempo: Atributos del SW relativos a los tiempos de respuesta y de procedimiento de los datos.

- ✓ **Comportamiento de recursos:** Atributos de SW relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.
- ✓ **Mantenibilidad:** Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema SW.
- ✓ **Estabilidad:** Capacidad del SW de tener un desempeño normal a pesar de hacerse modificaciones.
- ✓ **Facilidad de análisis:** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- ✓ **Facilidad de cambios:** Capacidad del que tiene el SW para que la modificación pueda ser válida.
- ✓ **Facilidad de pruebas:** Capacidad del que tiene el SW para que la modificación pueda ser válida. (Figuroa, 2012, pág. 4)

## **Portabilidad**

Conjunto de atributos relacionados con la capacidad de un sistema SW para ser transferido desde una plataforma a otra.

- ✓ **Adaptabilidad:** Evalúa la oportunidad para adaptar el SW a diferentes ambientes sin necesidad de aplicarle modificaciones.
- ✓ **Facilidad de instalación:** Es el esfuerzo necesario para instalar el SW en un ambiente determinado.

- ✓ **Cumplimiento:** Permite evaluar si el SW de adhiere a estándares o convenciones relativas a portabilidad.
- ✓ **Capacidad de reemplazo:** Se refiere a la oportunidad y el esfuerzo usando en sustituir el SW por otro producto con funciones similares.

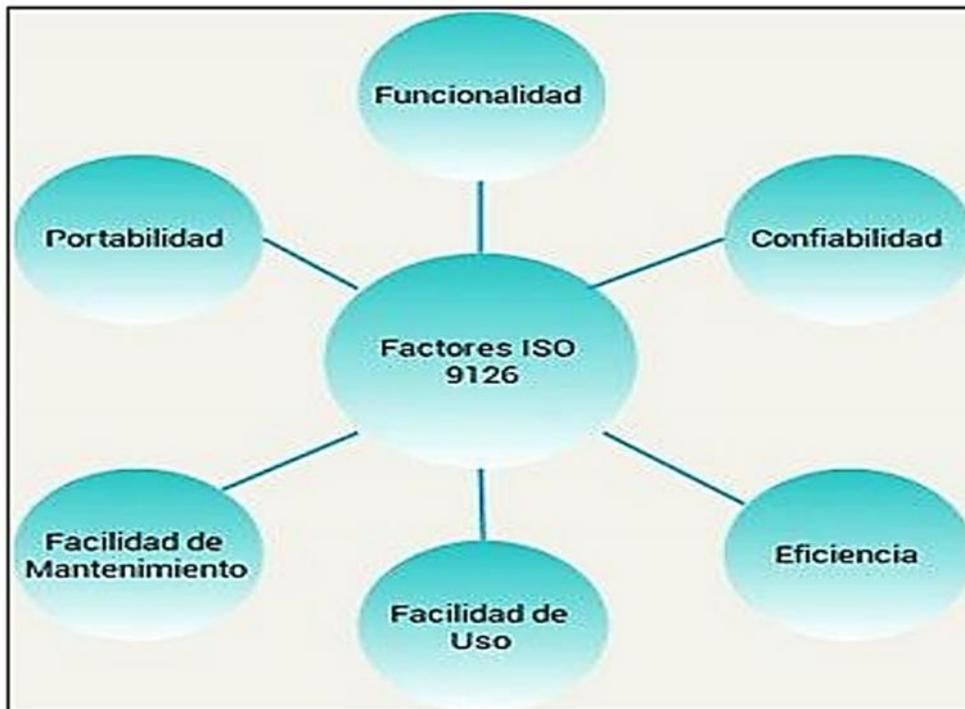
La calidad de estos sistemas y la frecuencia con la que deberían ser usados.

Los problemas incluyen baja representación, poca utilidad y consumo, lo que hace difícil atender las necesidades específicas de cada estudiante. Además, la educación en línea ha sido criticada muchas veces por no ser un sistema educativo que soporte las necesidades del sistema educativo presencial. (Figuroa, 2012, pág. 5)

### 2.8.1 Factores de Calidad según ISO 9126

El estándar ISO – 9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son:

**Figura 2.11: Factores de calidad ISO 9126**



**Fuente:(Ressman, 2015)**

### Lista de factores:

- ✓ **Corrección:** mide el grado en que un programa satisface sus especificaciones y consigue los objetivos del usuario.
- ✓ **Fiabilidad:** mide el grado en que se puede separar que un programa lleve a cabo sus funciones esperadas con la precisión requerida.
- ✓ **Eficiencia:** mide la cantidad de recursos de computadora y de código requerido por un programa para que lleve a cabo las funciones específicas.
- ✓ **Integridad:** es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado
- ✓ **Facilidad de uso:** es el esfuerzo requerido para aprender un programa e interpretar la información de entrada y salida.
- ✓ **Facilidad de Mantenimiento:** es el esfuerzo requerido para localizar y arreglar programas.
- ✓ **Facilidad de Pruebas:** es el esfuerzo requerido para probar un programa.
- ✓ **Flexibilidad:** es el esfuerzo requerido para modificar un sistema operativo.
- ✓ **Portabilidad:** es el esfuerzo requerido para transferir un software de un hardware o un entorno de sistemas a otro.
- ✓ **Reusabilidad:** es el grado en que un programa (o partes de un programa) se puede reutilizar en otro.
- ✓ **Facilidad de interoperación:** es el esfuerzo requerido para asociar un programa a otro. (Ressman, 2015, pág. 8)

## **3 CAPITULO III**

### **3. MARCO APLICATIVO**

#### **3.1 INTRODUCCION**

En este capítulo se presenta el desarrollo del proyecto en el cual necesariamente se tiene que seguir y ejecutar un conjunto de actividades que se encuentran en el proceso de desarrollo que se ha optado por emplear. En el desarrollo del presente proyecto se utiliza la metodología del El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) y la herramienta de modelado UML, con el objetivo de producir software de alta calidad que cumpla con los requerimientos de los usuarios, dentro de una planificación que cubra el ciclo de vida del desarrollo de software.

El Proceso Unificado Agil describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.

#### **3.2 ANALISIS Y DISEÑO DEL SISTEMA**

Primeramente, se realiza el análisis de la situación actual describiendo el entorno de trabajo.

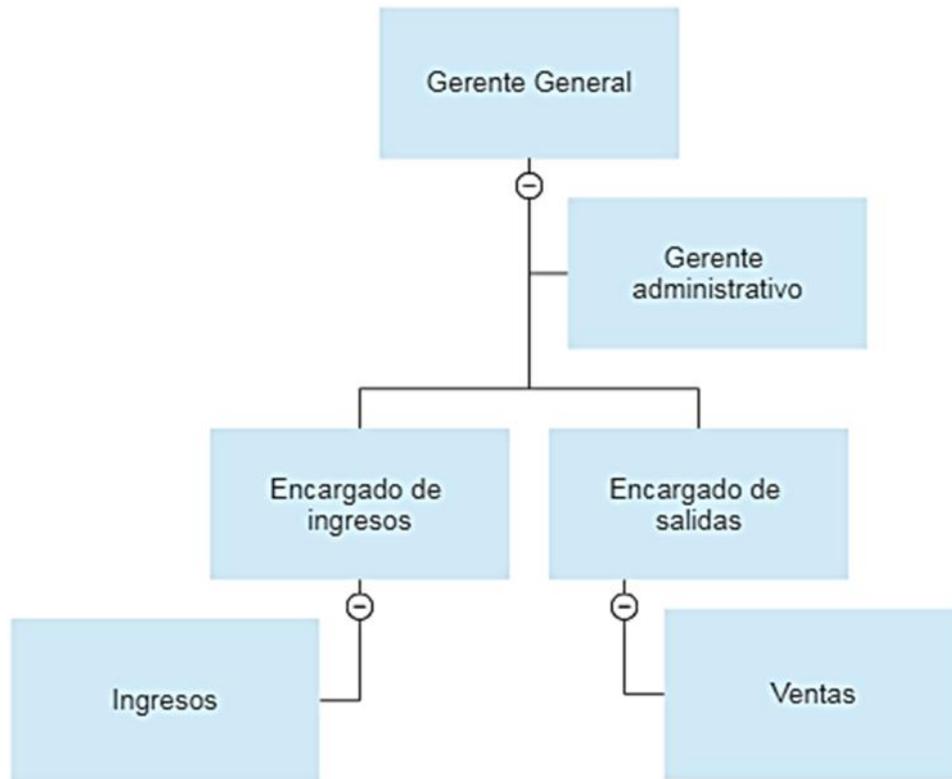
##### **3.2.1 Análisis de la situación actual**

Para lograr el éxito en el desarrollo del sistema, es fundamental la comprensión de los requisitos del usuario necesarios para la construcción del sistema. El análisis de sistemas se centra en la descripción detallada como base para su implementación del sistema propuesto

###### **3.2.1.1 Estructura Orgánica de la Empresa**

La empresa SERVIALINUT, muestra una estructura definida en la administración de ingreso y salida de alimentación, con características propias y diferente a su actividad empresarial.

**Figura 3.1: Organigrama de la Empresa Servialinut**



### **Proceso de Ingresos**

El encargado de ingresos es el que realiza el ingreso de los tipos de alimentación para posteriormente se pueda vender las dietas a los clientes.

### **Proceso de Salida**

El encargado de salidas es el que realiza la salida de los tipos de alimentación a los clientes este se encarga de hacer el registro y proporcionar un reporte al Seguro Social Universitario (SSU) ubicado en la avenida 6 de agosto – La Paz

### **3.3 FASE INICIAL**

En esta fase primeramente se analizó todos los problemas que existían en la empresa SERVIALINUT posteriormente se hizo un estudio de los requerimientos de los usuarios.

### 3.3.1 Requerimientos de Usuario

El paso fundamental dentro del análisis de sistemas es el análisis de requisitos para así comprender las necesidades del usuario, además permite tener una mejor visión del problema y encontrar el camino hacia la solución, que satisfaga esas necesidades.

Dentro de este contexto, se identifica una serie de requisitos que son descritos a continuación:

- ✓ Captura de datos del cliente y registro del servicio ofrecido
- ✓ Registrar las ordenes de ingreso realizadas por los encargados
- ✓ Generar el reporte de los servicios vendidos
- ✓ Controlar la venta de servicio de cada cliente
- ✓ Controlar las pérdidas, devoluciones, la cantidad de ingresos, la cantidad vendida y el importe subtotal que se recibe en el proceso de la venta.
- ✓ Generar reportes de ventas de acuerdo a criterios que maneja la empresa.
- ✓ Generar reportes de los ingresos realizados.

### 3.3.2 Requerimientos Tecnológicos

#### Requerimiento de Software

La herramienta tecnológica a utilizar será el PHP que es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. Entre sus principales características se pueden destacar su potencia, alto rendimiento y su facilidad de aprendizaje. PHP es una eficaz herramienta de desarrollo para los programadores web, ya que proporciona elementos que permiten generar de manera rápida y sencilla sitios web dinámicos.

PHP es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java. Su sintaxis es muy similar a la de estos lenguajes, haciendo muy sencillo su aprendizaje incluso a programadores noveles.

PHP es un lenguaje para la creación de sitios web del que se pueden destacar las siguientes características:

- - Es un potente y robusto lenguaje de programación embebido en documentos HTML.
- - Dispone de librerías de conexión con la gran mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información permanente en el servidor.
- Proporciona soporte a múltiples protocolos de comunicaciones en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.).

La arquitectura utilizada será la de cliente/servidor, ya que es el más aconsejable por el tipo de sistemas que se va desarrollando.

Esta arquitectura es beneficiosa porque se utilizará el motor de base de datos SQL Server, debido a su flexibilidad y las múltiples opciones que tiene.

### **Requerimiento de Hardware**

En cuanto a hardware se requieren mínimamente los siguientes componentes:

- Procesador Core dúo o superior
- 512 MB de memoria RAM o superior.
- Disco Duro de almacenamiento mayor a 128 Gigabyte.
- Monitor de 14 pulgadas o superior.

### **Plataforma**

La plataforma de software es Windows 7 o superior que contienen una gran variedad de características y posibilidades que optimizan el rendimiento y proporcionan una alta disponibilidad de recuperación de fallos y distintos niveles de seguridad.

## **3.4 FASE DE ELABORACION**

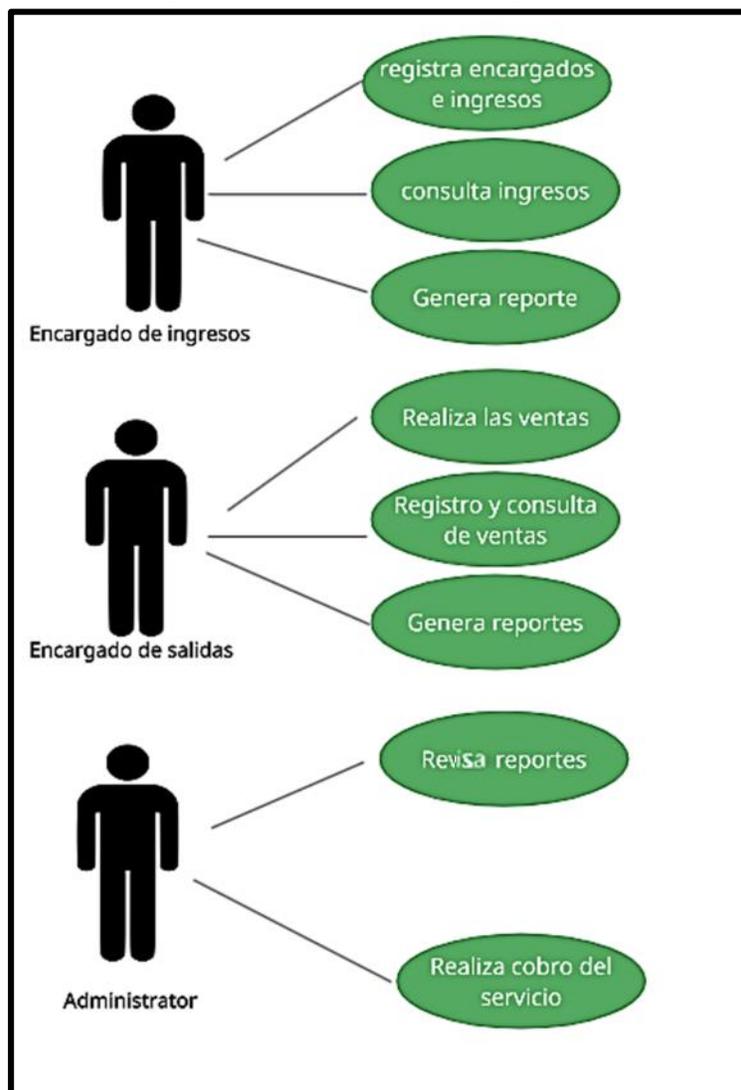
En esta fase se establece el contexto del sistema mediante el modelo del negocio a través de casos de uso logrando definir el alcance del proyecto.

### 3.4.1 Modelo del negocio

El modelo de negocio es una técnica para comprender los procesos de la organización. El modelo del negocio describe los procesos en términos de casos de uso y actores del contexto que se corresponden con los procesos del sistema y los usuarios.

El modelado del negocio detallado a continuación (Ver Figura 3.2) identifica los procesos más importantes del contexto del sistema y describe los procesos exactos relacionados con los actores y casos de uso encontrados.

**Figura 3.2: Modelo de negocio**



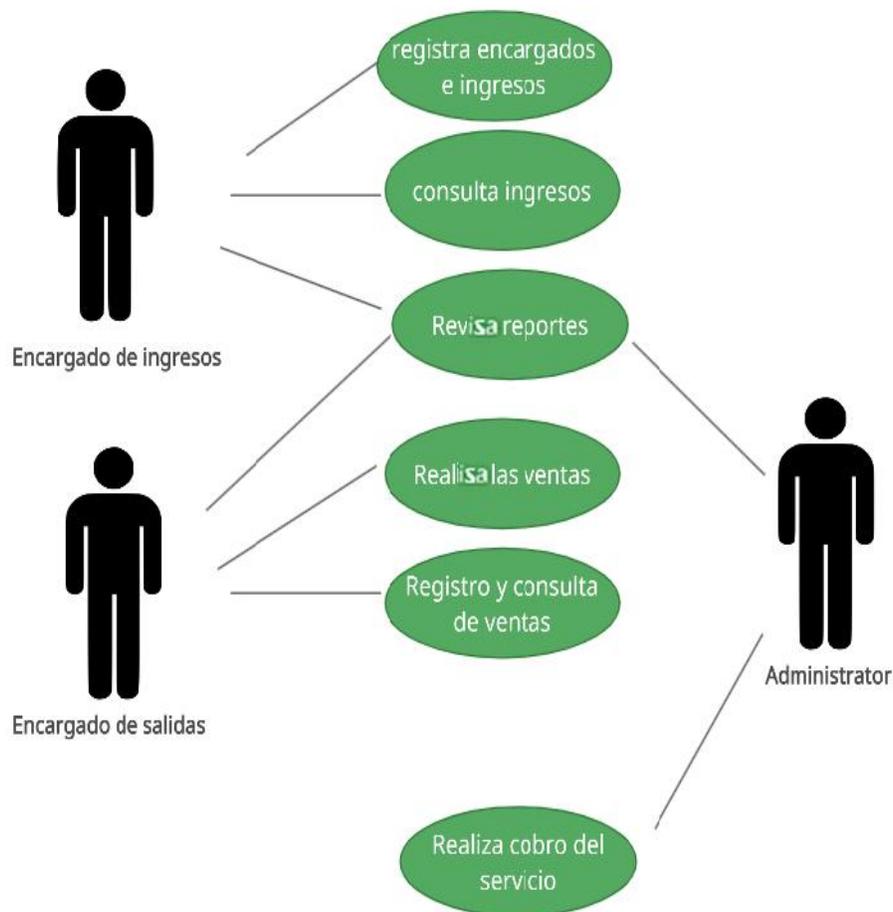
**Fuente: Elaboración propia**

### 3.4.2 Modelo de Caso de Uso

El modelo de Caso de Uso (Figura 3.3) nos permite representar a los diferentes actores y los casos de uso de la empresa. En los siguientes diagramas podemos observar a los principales actores que interactúan con el sistema.

**Figura 3.3: Modelo de Caso de Uso**

#### **Sistema automatizado de administracion y control "SERVIALINUT"**



**Fuente: Elaboración propia**

### 3.4.2.1 Actores

El encargado de ingresos es una persona que realiza las tareas de registro de clientes, coordinación de tipos de alimentación, recojo de la mercancía y elabora informes para entregar a la institución de los cuales se brindó el servicio.

#### Encargado de Ventas

El encargado de ventas realiza el registro de las partes diarias de la venta de servicios de alimentación, posteriormente realiza el informe de los servicios brindados a las diferentes áreas de la institución.

#### Gerencial

El Gerente General es una persona que utiliza el sistema para solicitar reportes de los ingresos provenientes de la venta y la distribución de los servicios de alimentación y realiza los cobros correspondientes.

### 3.4.3 Diagramas Caso de Uso

A continuación, se detalla los diagramas de casos de uso más representativos:

- ❖ Caso de Uso: Realiza el Ingreso

Este caso de uso lo inicia el encargado de ingreso



**Fuente: Elaboración propia**

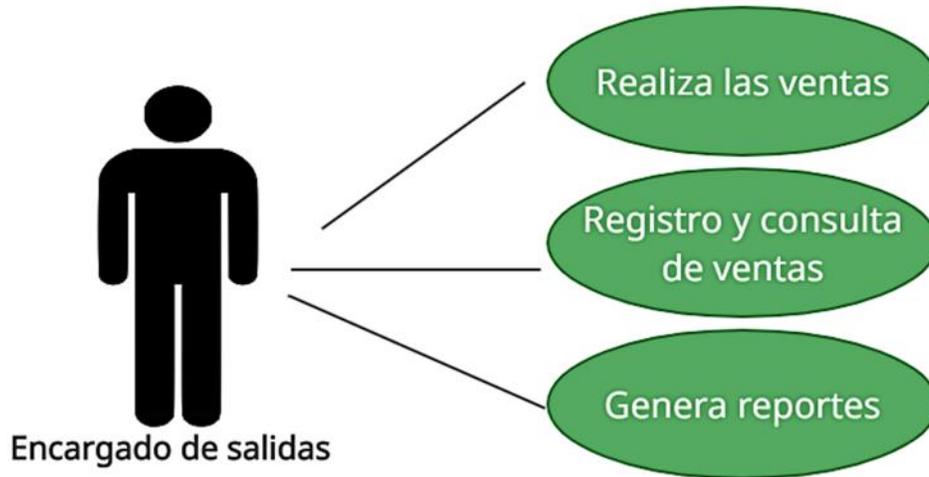
**Tabla 3.1: Caso de Uso: Realiza el Ingreso**

<b>Caso de Uso: Realiza el ingreso</b>	
Actores:	Encargado de ingreso
Propósito:	Realiza el registro, verificación y actualización de datos.
Resumen:	El encargado ingresa al sistema, realiza el registro de datos del cliente, posteriormente realiza el ingreso y finalmente la actualización de los datos ya registrados.
Tipo:	Primario
Referencias	
Referencias Cruzadas	
<b>Curso Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos  a) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. b) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Introduce los datos del cliente	4. Registra los datos introducidos
5. Confirma datos introducidos	6. Guarda datos
7. Se elige la opción detalle del cliente.	8. Mostrar el formulario y actualiza automáticamente los clientes
9. Realizar los cambios respectivos del cliente	10. Siempre que se haga un cambio. este se almacenará en el informe de actualización de clientes.
11. Guardar	12. Guardara la actualización
13. Para Salir de la tabla apretar cancelar	

**Fuente: Elaboración propia**

❖ Caso de Uso Registra ventas

Figura 3.5: Caso de Uso: Registra Ventas



Fuente: Elaboración propia

Tabla 3.2: Caso de Uso: Registra Ventas

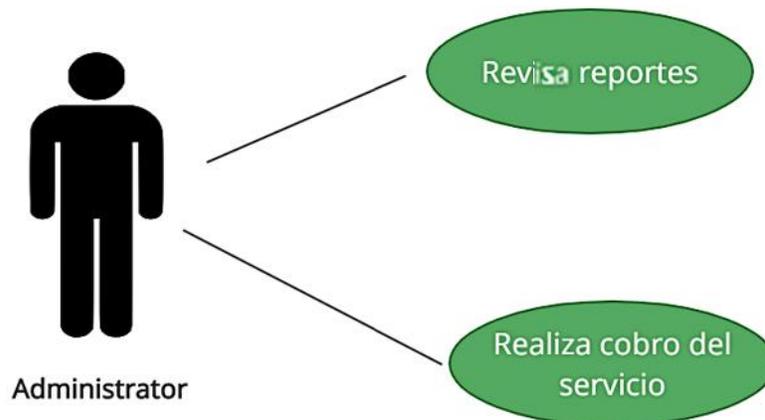
Caso de Uso:	Registra ventas
Actores:	Encargado de ventas
Propósito:	Realizar el registro de las ventas
Resumen:	El encargado de ventas ingresa al sistema, realiza el registro de los datos de las ventas de los medios de prensa, verifica medios de prensa más vendidos y posteriormente realiza el cobro de los servicios
Tipo: Referencias	Primario
Referencias Cruzadas	

Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos  e) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. d) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Elige la opción ventas	4. Nos muestra la interfaz de formulario de ventas para la adición de datos
5. Introduce los datos de la venta	6. Registra los datos introducidos
7. Confirma datos introducidos	8. Guarda datos
9. Se quiere realizar alguna otra operación volver al menú principal	10. saldrá al menú principal
11. Para salir del sistema apretar cancelar.	

Fuente: Elaboración propia

❖ Caso de Uso administrador

Figura 3.6: Caso de Uso: Administrador



Fuente: Elaboración propia

**Tabla 3.3: Caso de Uso: Registro Administrativo**

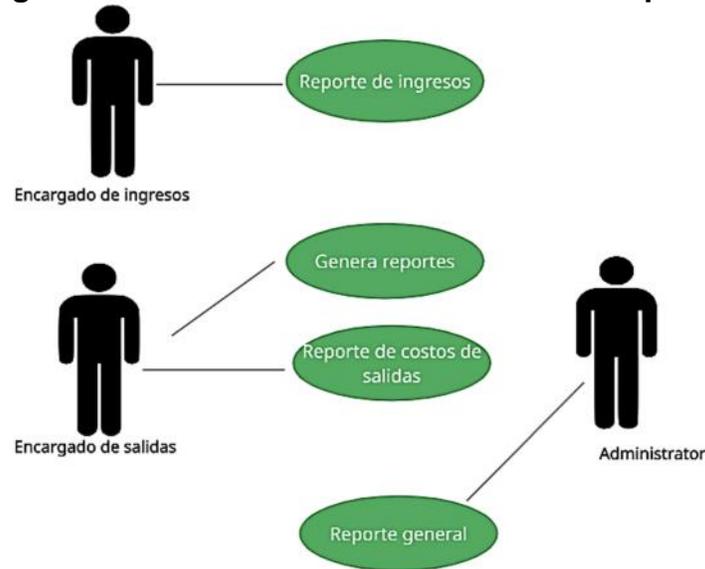
<b>Caso de Uso: Registro Administrativo</b>	
Actores:	Administración
Propósito:	Realizar el registro de las distribuciones
Resumen:	El encargado de administración ingresa al sistema, realiza el registro de la distribución de los tipos de alimentación que se proporcionara en el día, posteriormente realiza el informe de la distribución que se realiza en la empresa.
Tipo:	Primario
Referencias	
Referencias Cruzadas	
<b>Curso Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos
	a) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. b) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Elige la opción Ingreso	4. Nos muestra la interfaz de formulario de ingreso para la adición de datos
5. Introduce los datos de la distribución de ingresos	6. Registra los datos introducidos
7. Confirma datos introducidos	8. Guarda datos
9. Apretar el botón imprimir formulario	10. Mandara el sistema a imprimir el formulario
11. Se quiere realizar alguna otra operación volver al menú principal	12. Saldrá al menú principal
13. Para salir del sistema apretar cancelar.	

**Fuente: Elaboración propia**

❖ Caso de Uso Generación de Reportes

**GENERACION DE REPORTES**

**Figura 3. 7: Caso de Uso: Generación de Reportes**



**Fuente: Elaboración propia**

**Tabla 3.4: Caso de Uso: Generación de Reportes**

<b>Caso de Uso: Generación de Reportes</b>	
Actores:	Encargado de ingresos, Encargado de ventas y administración
Propósito:	Mostrar los distintos reportes para poder realizar el control de la información que se tiene.
Resumen:	El encargado de Ingresos, Encargado de ventas y Administración una vez que se decida qué tipo de información requiere, selecciona un reporte, posteriormente actualiza la información, con lo que se tiene la información, que puede ser impresa o que puede ser visualizada por pantalla
Tipo: Referencias	Primario
Referencias Cruzadas	

Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos
3. Elige la opción reportes	a) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. b) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
5. Elige si se imprime el reporte	4. El sistema visualiza la información a través de una pantalla
6. Para salir del sistema apretar cancelar.	

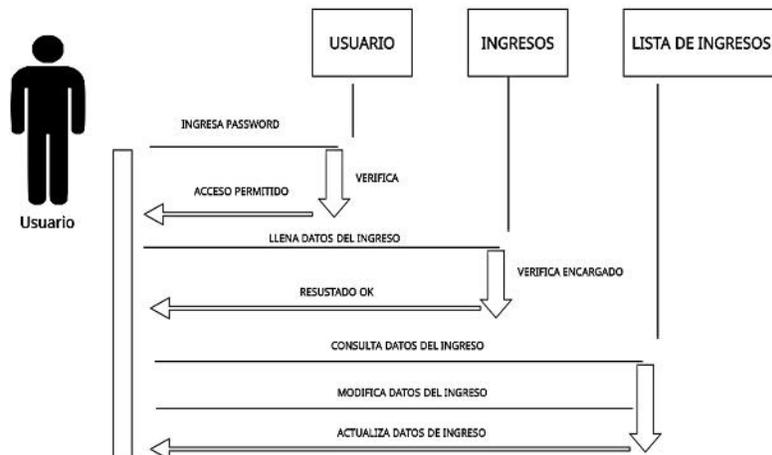
**Fuente: Elaboración propia**

### 3.4.4 Diagrama de Secuencia

Mediante los Diagramas de Secuencia se explica los pasos a seguir en forma secuencial de los diferentes procesos que existen en el sistema actual y que serán desarrollados de acuerdo a los casos de uso esenciales, comenzando de la acción del actor y la respuesta del sistema mediante la acción de las clases externas.

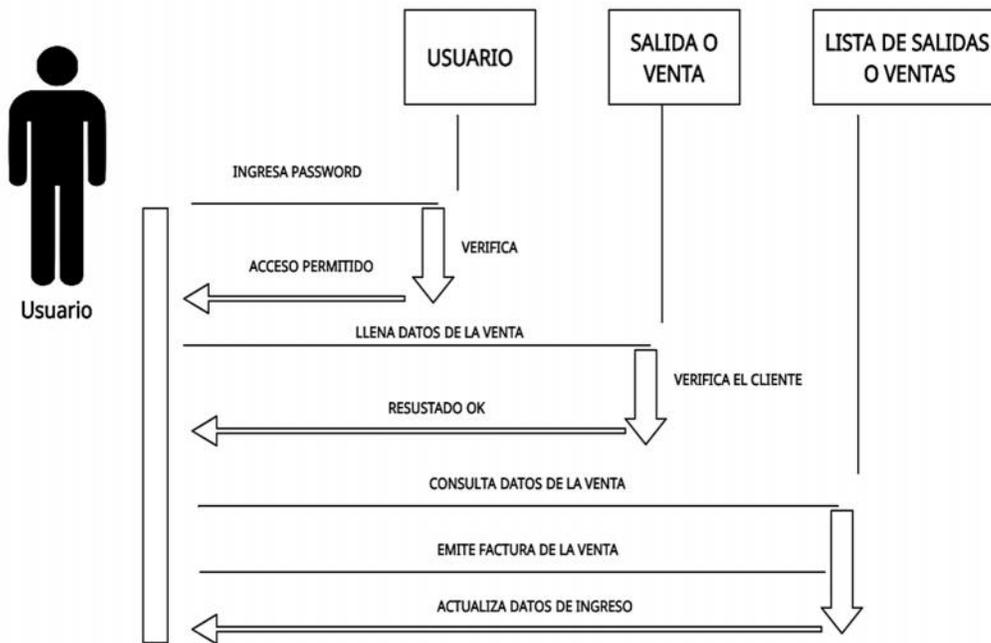
**Realiza el Ingreso.** Los usuarios serán: Encargado del Ingreso que se encuentran en la Figura 3.8

**Figura 3.8: Diagrama de Secuencia: Realiza Ingreso Fuente: Elaboración propia**



**Registro Ventas.** Los usuarios serán: Encargado de Ventas

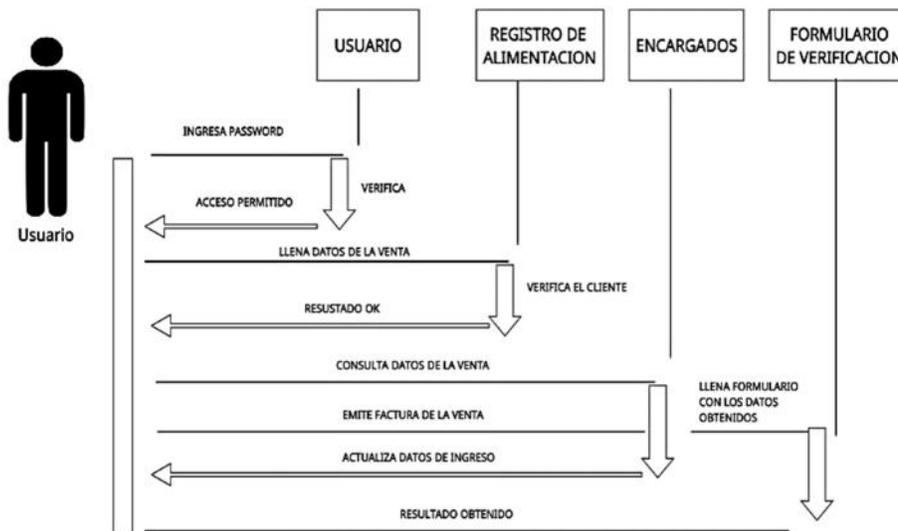
**Figura 3.9: Diagrama de Secuencia: Registra Ventas**



**Fuente: Elaboración propia**

**Registra Distribución.** Los usuarios serán: Encargado de Administracion.

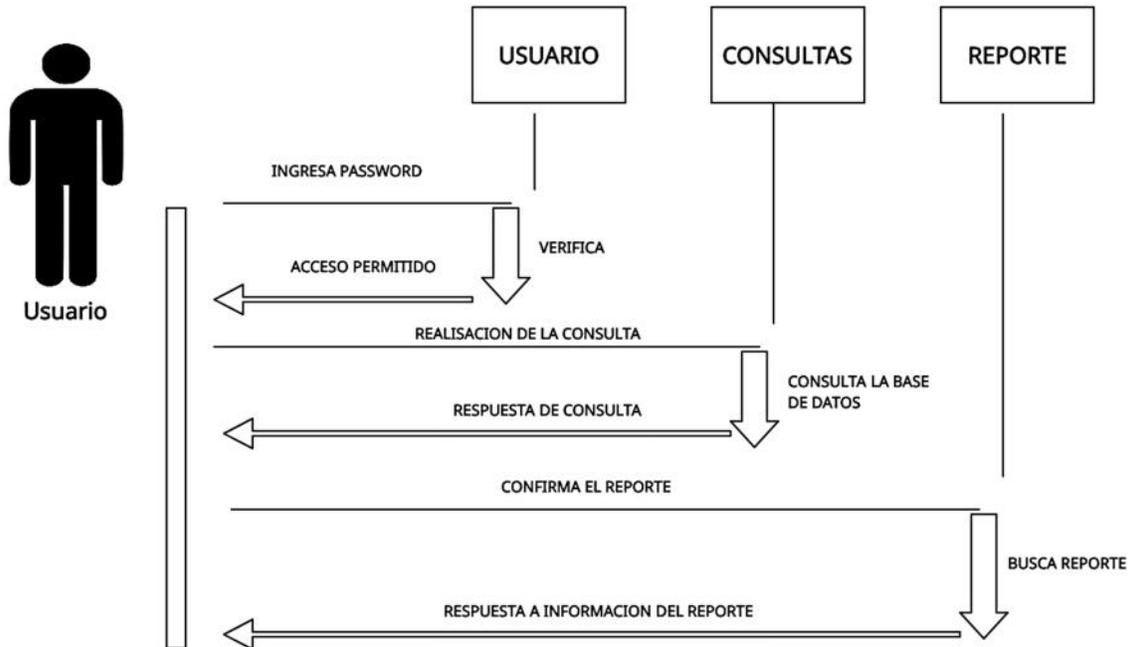
**Figura 3.10: Diagrama de Secuencia: Registra Distribución**



**Fuente: Elaboración propia**

**Generación de Reportes.** Los usuarios serán: Encargado de Circulación, Encargado de Ventas y Gerencial.

**Figura 3.11: Diagrama de Secuencia: Generación de Reportes**



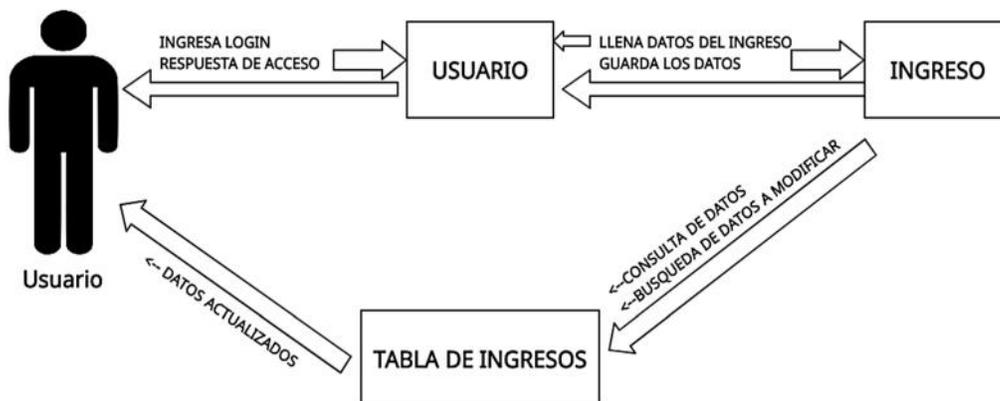
**Fuente: Elaboración propia**

### 3.4.5 Diagrama de Colaboración

Estos diagramas están elaborados de acuerdo a los diagramas de secuencia y a los mensajes encontrados entre las clases del sistema y se dan de acuerdo a los casos de uso generales que existen.

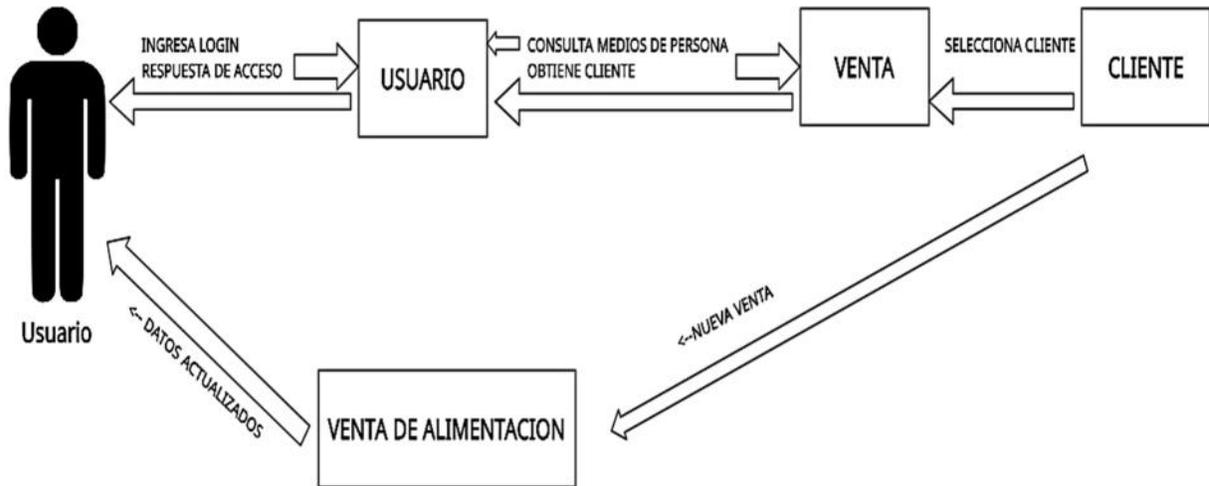
**Realiza Suscripción.** Los usuarios serán: Encargado de Circulación.

**Figura 3.12: Diagrama de Ingreso: Realiza Ingreso Fuente: Elaboración propia**



**Registra Ventas.** Los usuarios serán: Encargado de Ventas.

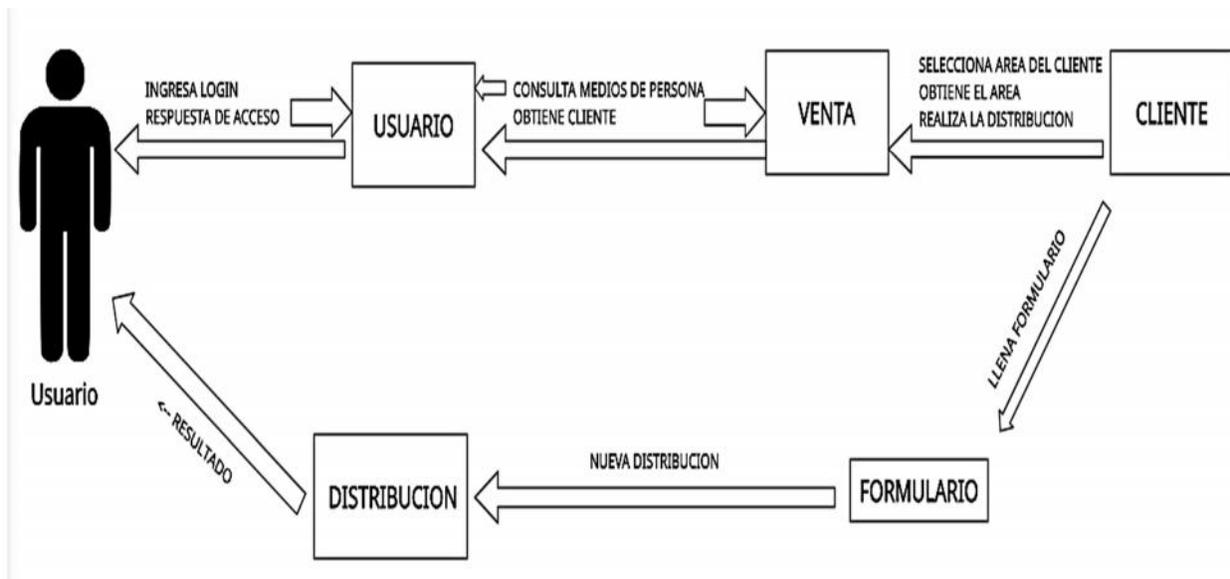
**Figura 3.13: Diagrama de colaboración: Registra Ventas**



**Fuente: Elaboración propia**

**Registro Administrador.** Los usuarios serán: Encargado de Administración

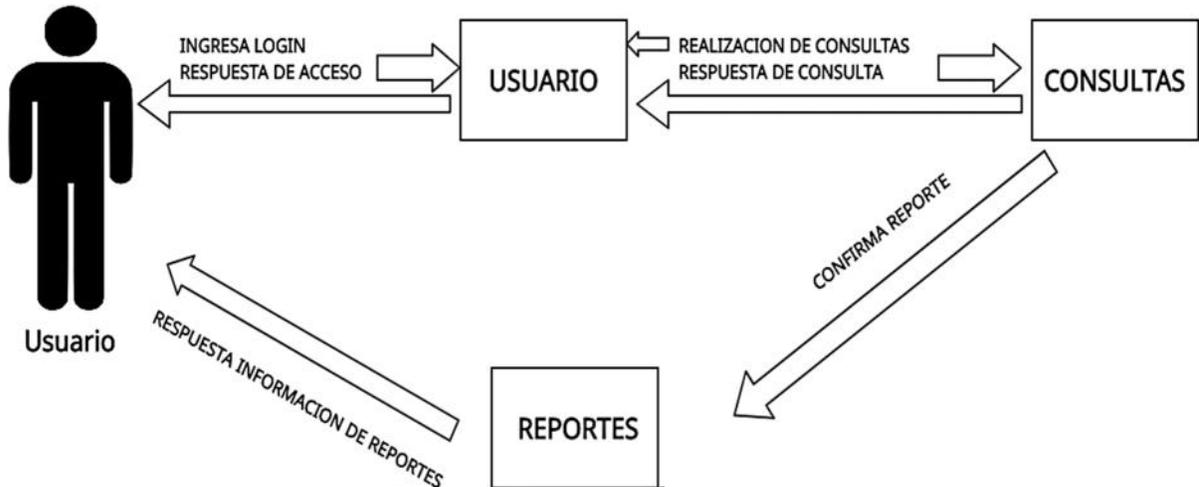
**Figura 3.14: Diagrama de Colaboración: Registra Distribución**



**Fuente: Elaboración propia**

**Generación de Reportes. Los usuarios serán:**(Encargado de Circulación, Encargado de Ventas y Gerencial) Figura 3.15

**Figura 3.15: Diagrama de Colaboración: Generación de Reportes**



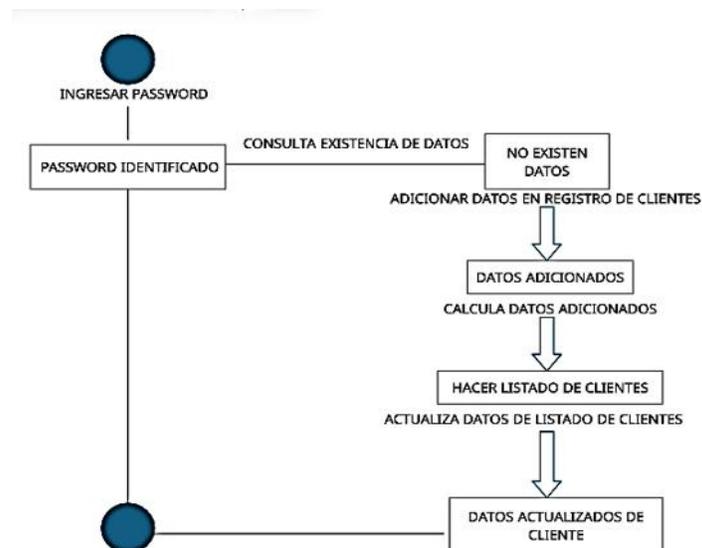
**Fuente: Elaboración propia**

### 3.4.6 Diagrama de Estados

Estos diagramas serán elaborados de acuerdo a los casos de uso esenciales y a los diagramas de colaboración, para saber las transiciones y estados que tienen cada caso de uso.

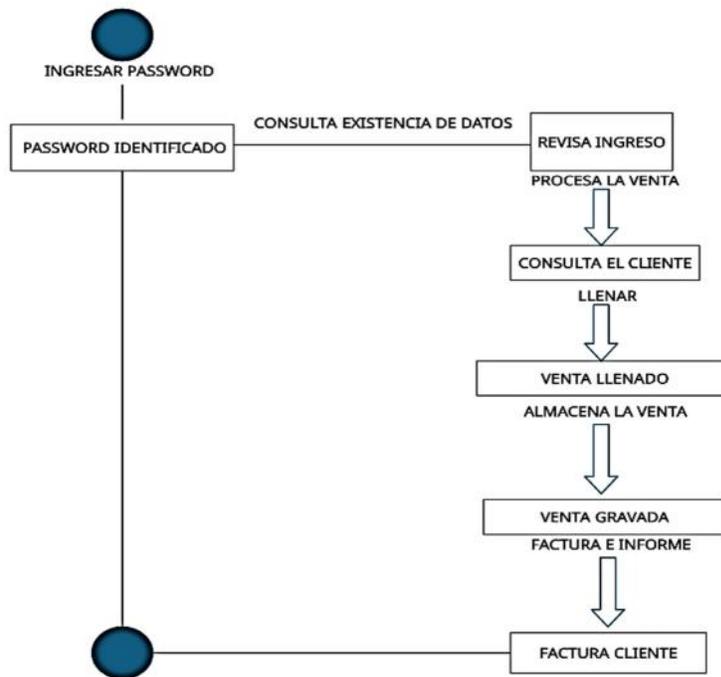
**Realiza Ingreso.** Los usuarios serán: Encargado de Circulación.

**Figura 3.16: Diagrama de Estados: Realiza Ingreso Fuente: Elaboración propia**



**Registra Ventas.** Los usuarios serán: Encargado de Ventas.

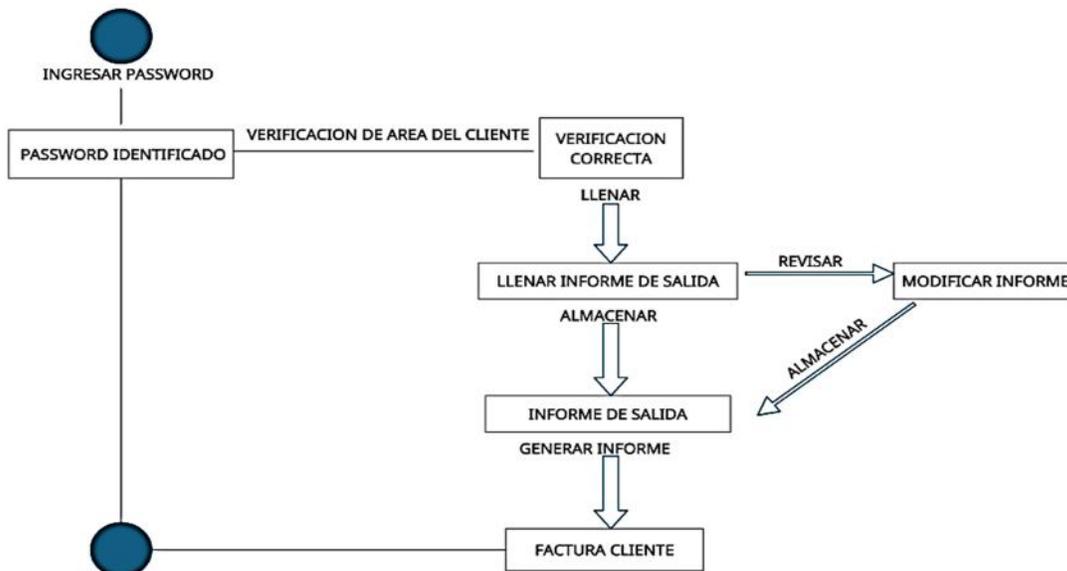
**Figura 3.17: Diagrama de Estados: Registra Ventas**



**Fuente: Elaboración propia**

**Registro Administrador.** Los usuarios serán: Encargado de Administración

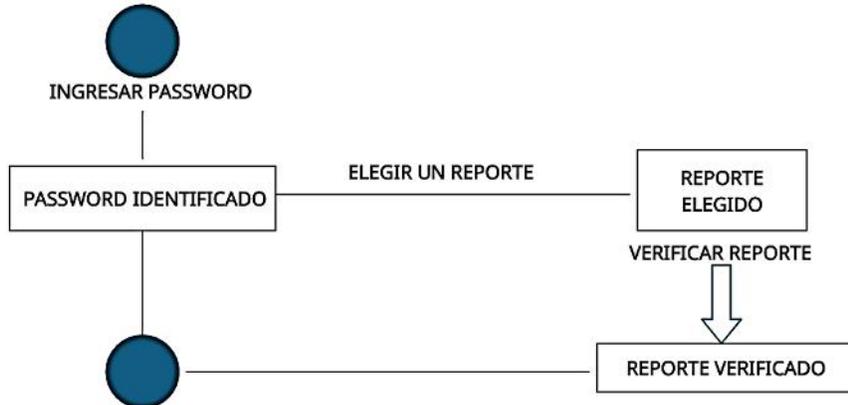
**Figura 3.18: Diagrama de Estados: Registra Administrador**



**Fuente: Elaboración propia**

**Generación de Reportes.** Los usuarios serán: Encargado de Circulación, Encargado de Ventas y Gerencial.

**Figura 3.19: Diagrama de Estados: Generación de Reportes**



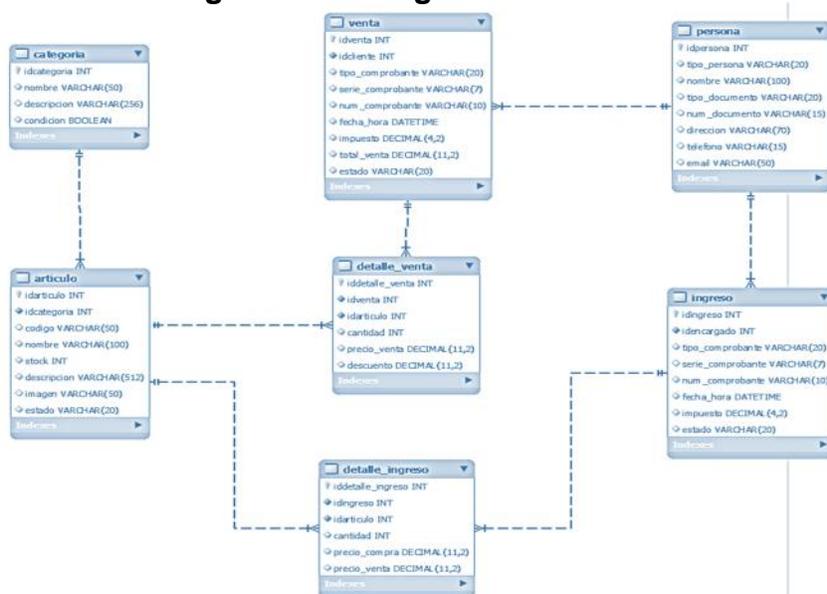
**Fuente: Elaboración propia**

### 3.4.7 Diagrama de Clases

El diagrama de clases describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones que existen.

En la figura 3.20 siguiente se muestra el diseño de las clases correspondiente al presente proyecto.

**Figura 3.20: Diagrama de Clases**



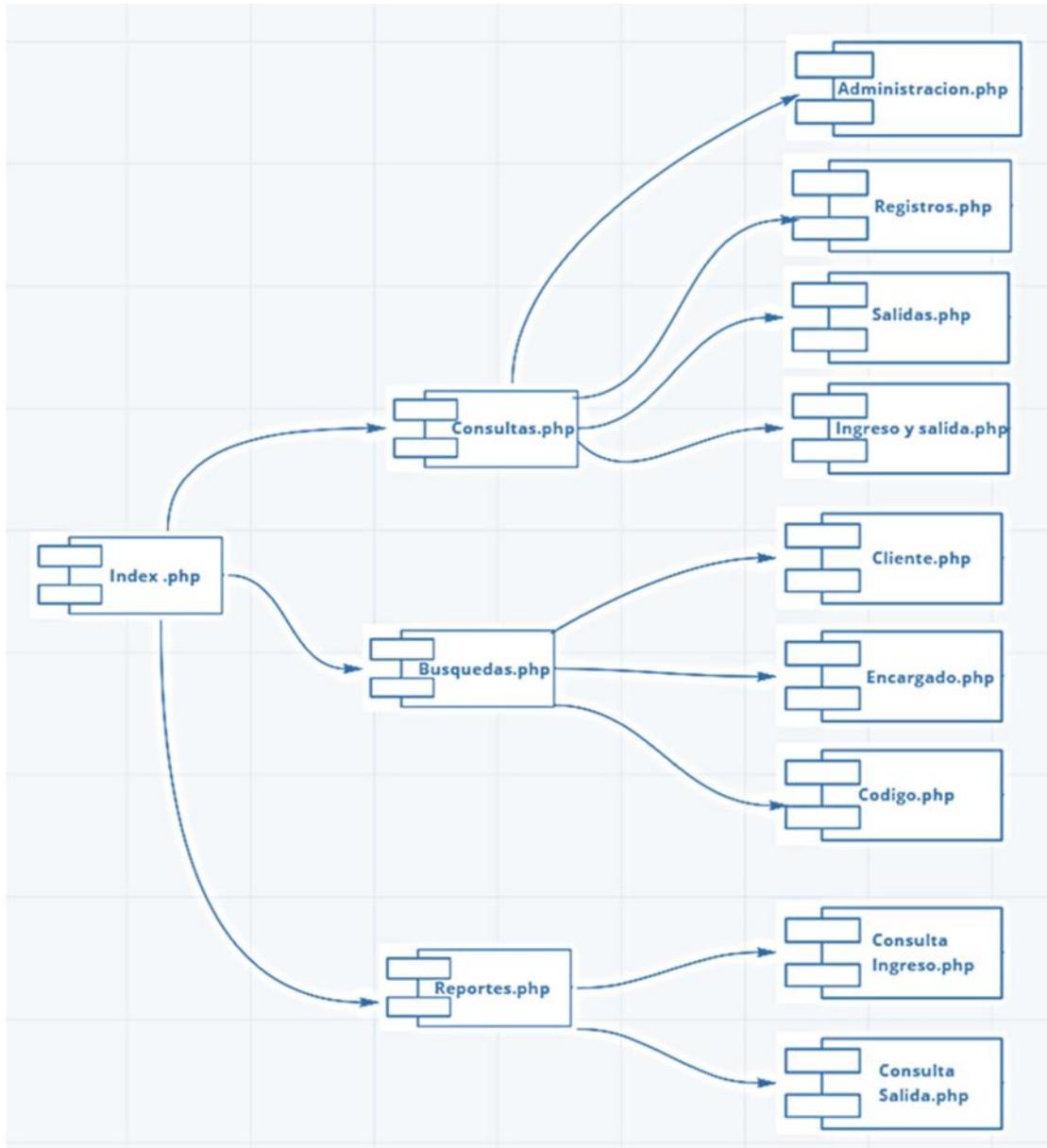
**Fuente: Elaboración propia**

### 3.5 FASE DE CONSTRUCCION

En esta fase se explicará el diseño e implantación del proyecto tomando en cuenta la especificación de los componentes, de los nodos y de la interfaz. Para esto se describirá el diagrama de componentes.

#### 3.5.1 Diagrama de Componentes

Figura 3.21: Diagrama de Componentes



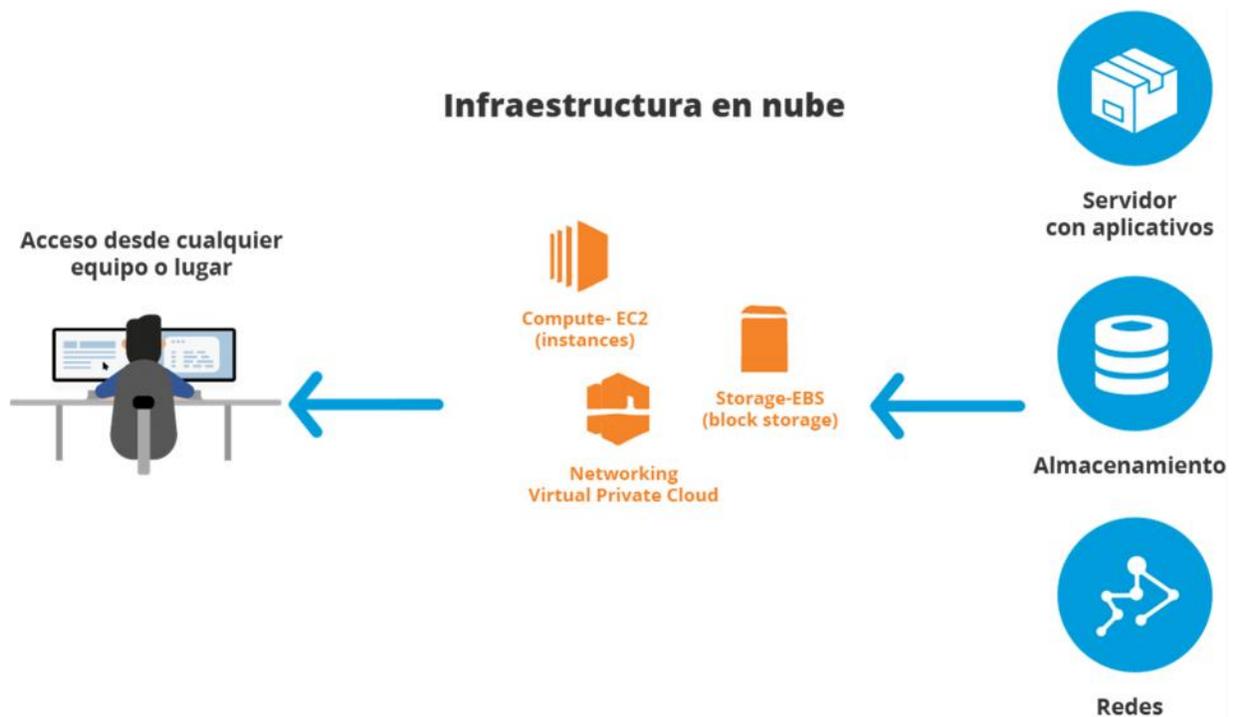
Fuente: Elaboración propia

### 3.5.2 Arquitectura del Sistema

La arquitectura que se desarrolla para el presente proyecto es el de encargado/servidor, el cual tendrá un servidor en la nube con el motor de base de datos.

La ubicación del servidor y los clientes se encuentran referenciados en la Figura 3.22

**Figura 3.22: Arquitectura del sistema**



**Fuente: Elaboración propia**

### 3.5.3 Implementación del sistema

En la implementación del sistema establecemos todas las interfaces con el usuario, esta actividad al igual que las actividades de captura de requisitos y diseño del sistema fue realizada iterativamente.

#### Ingreso al sistema

El usuario antes de ingresar al sistema se autentifica con un nombre de usuario y contraseña. En la figura 3.23 se observa la primera pantalla de autenticación.

**Figura 3.23: Inicio de Sesión**

**Administrador**

Ingrese sus datos de Acceso

Usuario 

Password 

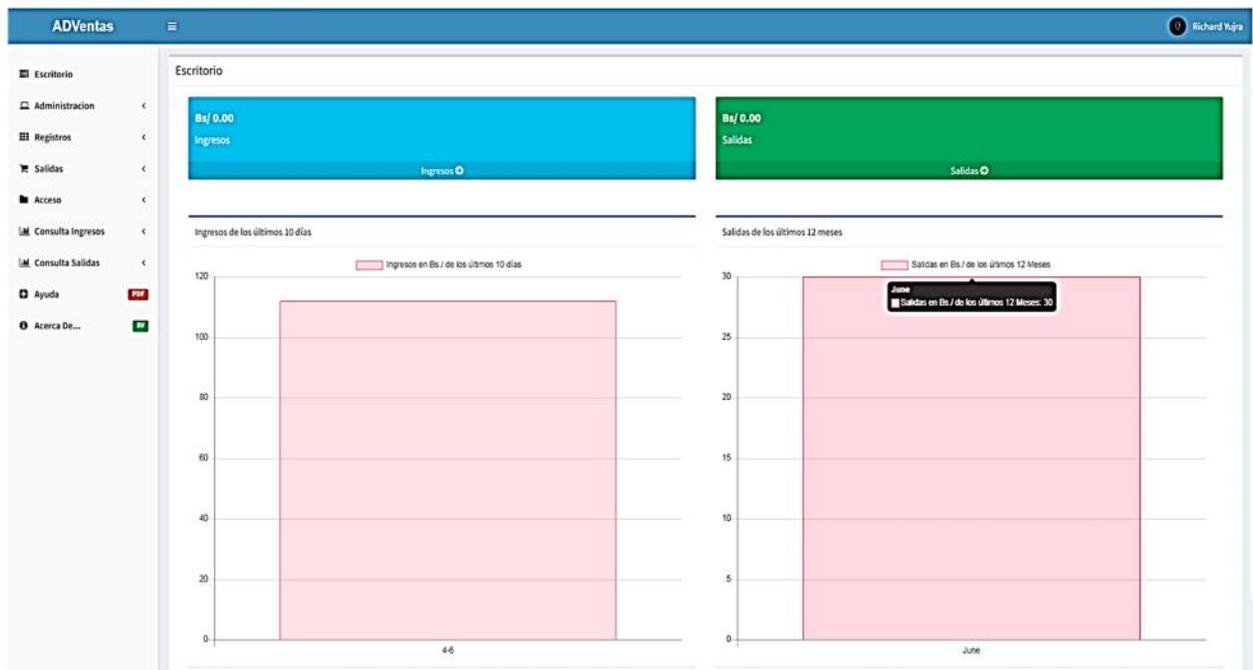
**Ingresar**

[Olvidé mi password](#)

**Fuente: Elaboración propia**

En la Figura 3.24 nos muestra el prototipo de una ventana principal, en la cual se encuentra activa la opción del menú ir a escritorio donde también muestra la estadística de los ingresos y salidas.

**Figura 3.24: Pantalla principal**



**Fuente: Elaboración propia**

Desde esta pantalla, el administrador realiza el ingreso de la alimentación

**Figura 3.25: Pantalla de Ingreso de Alimentación**

The screenshot shows the 'Alimentación' form in the ADVentas system. The form is titled 'Alimentación' and has a 'Reporte' button. It contains the following fields and controls:

- Nombre(\*):** A text input field with the placeholder 'Nombre'.
- Descripción:** A text input field with the placeholder 'Descripción'.
- Código:** A text input field with the placeholder 'Código Barras'.
- Tipo de Dieta(\*):** A dropdown menu with 'DIETA NORMAL' selected.
- Imagen:** An upload field with a 'Seleccionar archivo' button and the text 'Ninguno archivo selec.'.
- Buttons:** 'Generar' (green), 'Imprimir' (blue), 'Guardar' (blue), and 'Cancelar' (red).

At the bottom of the page, there is a copyright notice: 'Copyright © 2023-2025 Servialimut. Todos los derechos reservados.' and the version number 'Version 1.0.1'.

**Fuente: Elaboración propia**

Una vez registrado la información de alimentación se muestra el registro el cual puede ser editado e imprimir el reporte.

**Figura 3.26: Características de la suscripción**

The screenshot shows the 'Alimentación' form in the ADVentas system, now filled with data. The form is titled 'Alimentación' and has a 'Reporte' button. It contains the following fields and controls:

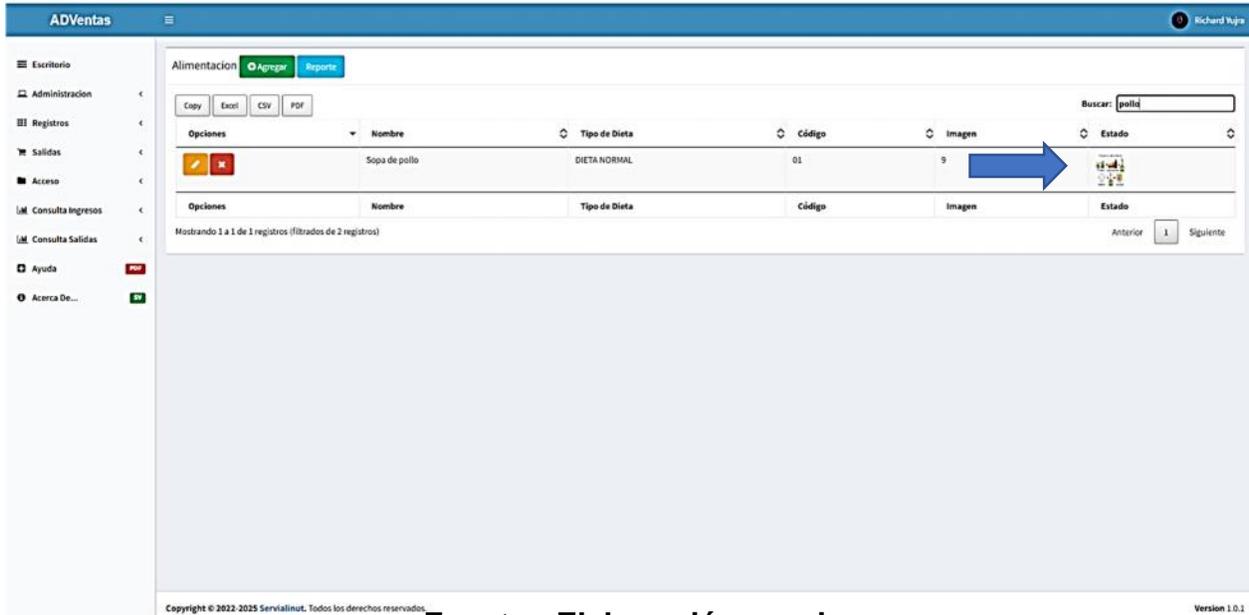
- Nombre(\*):** A text input field containing 'Sopa de pollo'.
- Descripción:** A text input field containing 'sopa de pollo hervido con verduras'.
- Código:** A text input field containing '01'.
- Tipo de Dieta(\*):** A dropdown menu with 'DIETA NORMAL' selected.
- Imagen:** An upload field with a 'Seleccionar archivo' button and the text 'Ninguno archivo selec.'. Below the field, there is a preview of a food image with the text 'Proceso de Carga Dental'.
- Buttons:** 'Generar' (green), 'Imprimir' (blue), 'Guardar' (blue), and 'Cancelar' (red).

Below the 'Código' field, a barcode is generated for the code '01'. At the bottom of the page, there is a copyright notice: 'Copyright © 2023-2025 Servialimut. Todos los derechos reservados.' and the version number 'Version 1.0.1'.

**Fuente: Elaboración propia**

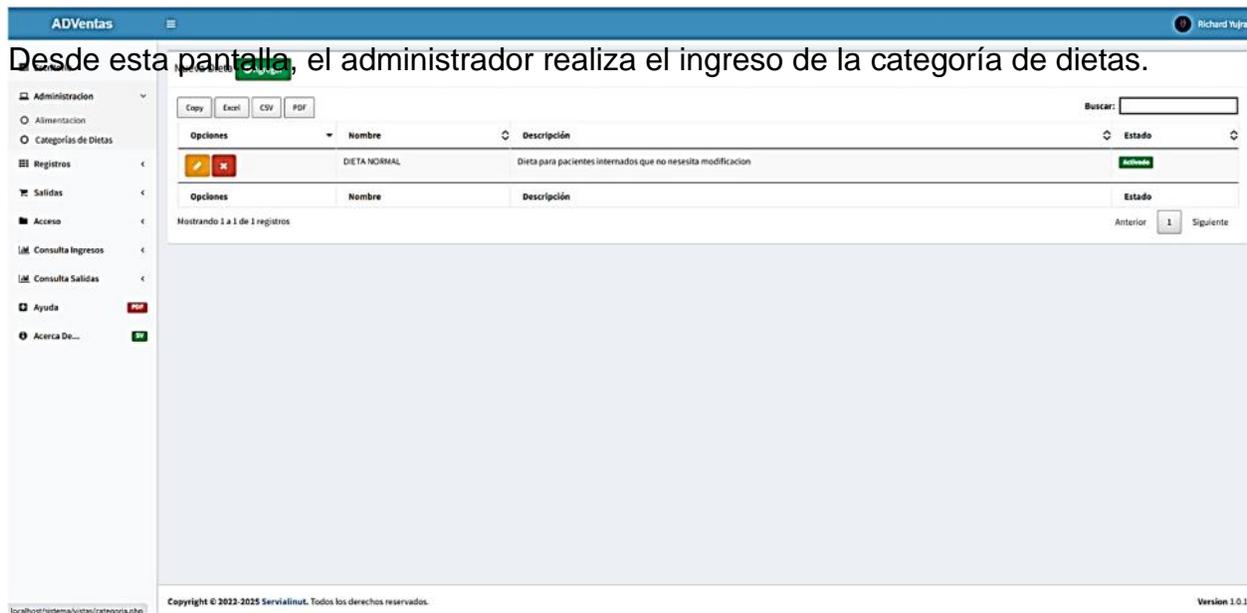
En esta pantalla se realiza las búsquedas de la información de la alimentación para la realización de las actualizaciones.

**Figura 3.27: Búsqueda de Clientes**



Fuente: Elaboración propia

**Figura 3.28: Pantalla de Categoría de Dietas**



Fuente: Elaboración propia

Desde esta pantalla, el administrador realiza el ingreso de la categoría de dieta.

**Figura 3.29: Pantalla Nueva Dieta**

ADVentas

Richard Fuja

Escritorio

Administración

Alimentación

Categorías de Dietas

Registros

Salidas

Acceso

Consulta Ingresos

Consulta Salidas

Ayuda

Acerca De...

Nueva Dieta

Nombre de la Dieta:

Nombre

Descripción:

Descripción

Guardar Cancelar

Copyright © 2022-2025 Servialinut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

Una vez registrado la información de la categoría se muestra el registro el cual puede ser editado y luego se guarda actualizando datos.

**Figura 3.30: Pantalla Editar Dieta**

ADVentas

Richard Fuja

Escritorio

Administración

Registros

Salidas

Acceso

Consulta Ingresos

Consulta Salidas

Ayuda

Acerca De...

Nueva Dieta

Nombre de la Dieta:

DIETA NORMAL

Descripción:

Dieta para pacientes internados que no necesita modificación

Guardar Cancelar

↑

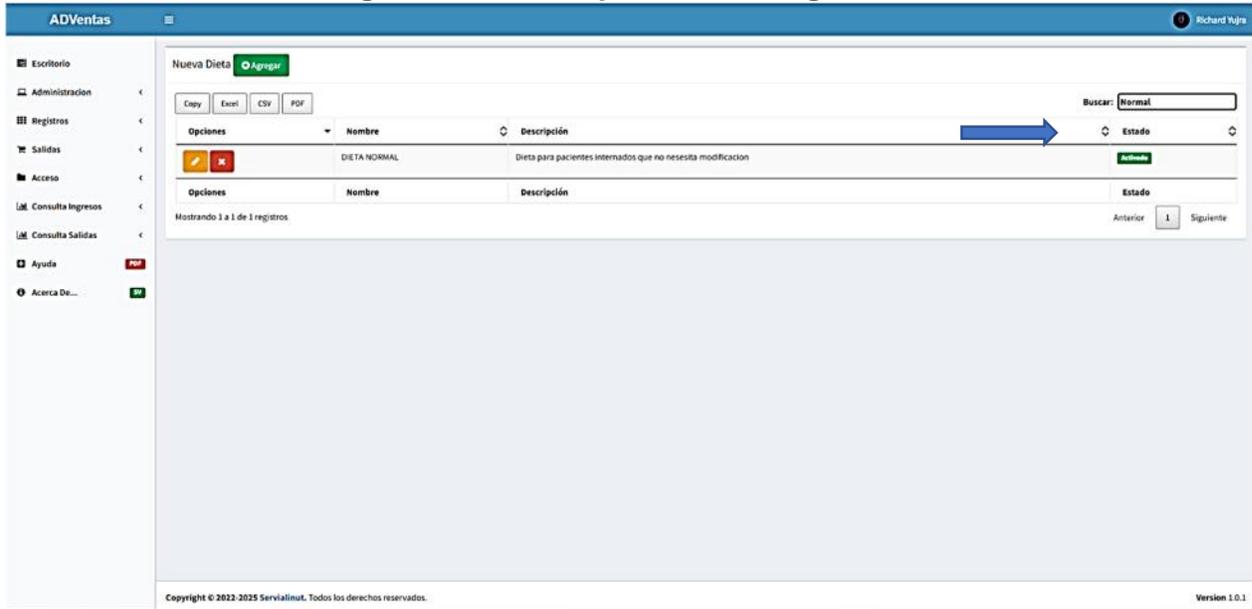
↑

Copyright © 2022-2025 Servialinut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

En esta pantalla se realiza las búsquedas de la información de la categoría para la realización de las actualizaciones.

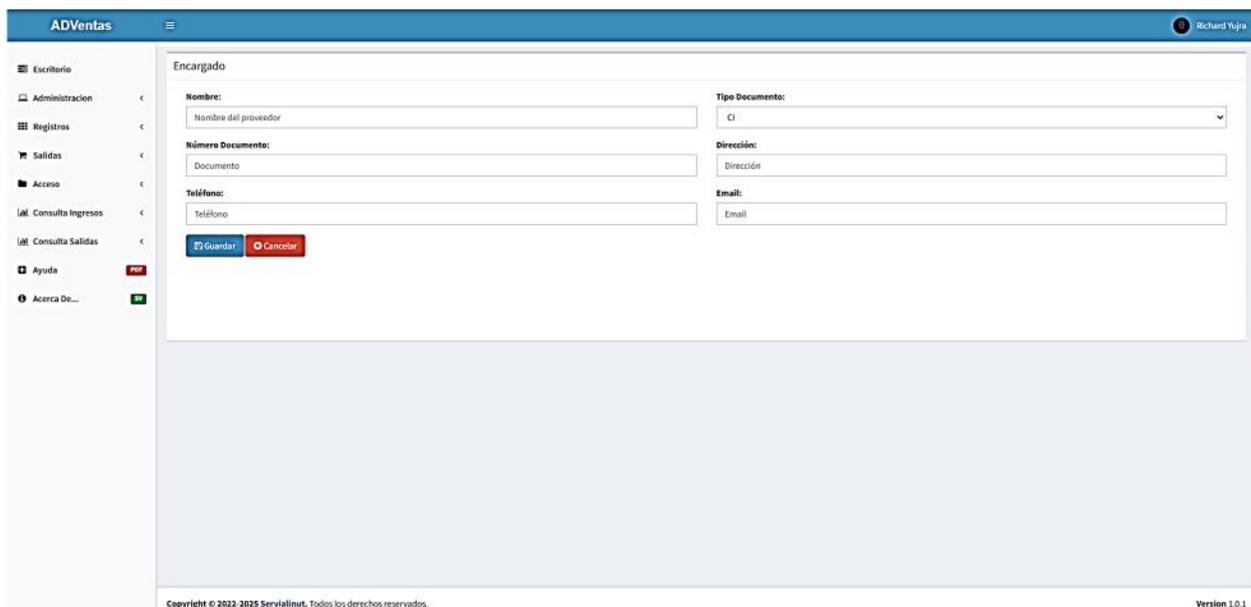
**Figura 3.31: Búsqueda de Categoría**



**Fuente: Elaboración propia**

Desde esta pantalla, Registros realiza el llenado de Encargado.

**Figura 3.32: Nuevo Encargado**



**Fuente: Elaboración propia**

Una vez registrado la información del Encargado se muestra el registro el cual puede ser editado y luego se guarda actualizando datos.

**Figura 3.33: Pantalla Editar Encargado**

ADVentas Richard Yujra

Encargado

Nombre: Randy Richard Chipana

Tipo Documento: CI

Número Documento: 9989131

Dirección: Zona/Tos Andes

Teléfono: 76223571

Email: richirandy6625@gmail.com

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

En esta pantalla se realiza las búsquedas de la información de los Encargados para la realización de las actualizaciones.

**Figura 3.34: Búsqueda de Encargado**

ADVentas Richard Yujra

Encargado

Opciones	Nombre	Documento	Número	Teléfono	Email
<input type="button" value="Editar"/>	Randy Richard Chipana	CI	9989131	76223571	richirandy6625@gmail.com

Mostrando 1 a 1 de 1 registros Anterior 1 Siguiente

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

Desde esta pantalla, Registros realiza el llenado de Ingreso.

**Figura 3.35: Nuevo Ingreso**

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

una vez registrado la información del Ingreso se muestra el registro el cual no puede ser editado y luego se guarda actualizando datos.

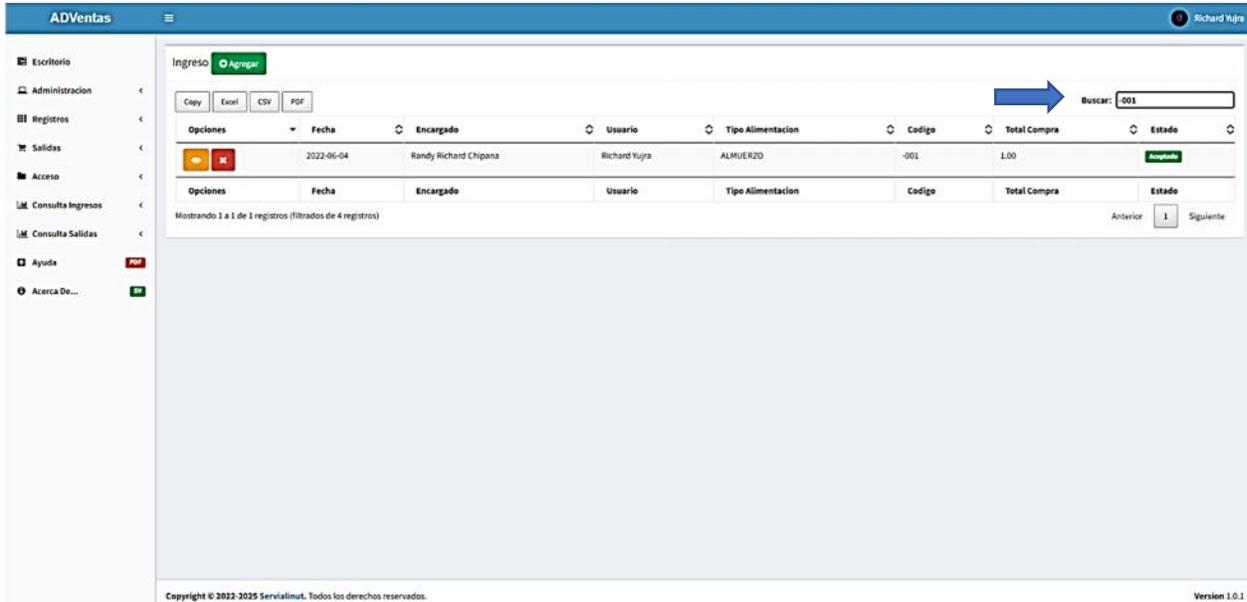
**Figura 3.36: Pantalla detalle Ingreso**

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

En esta pantalla se realiza las búsquedas de la información del ingreso esta puede ser buscado por código y nombre actualizando la pantalla.

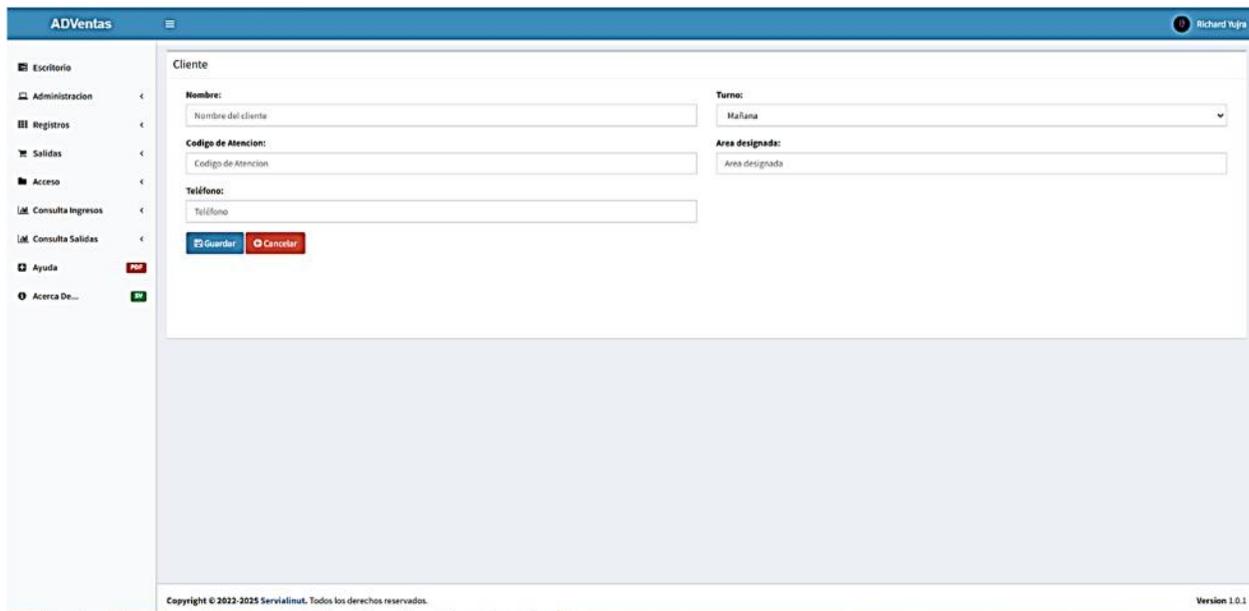
**Figura 3.37: Búsqueda de Encargado**



**Fuente: Elaboración propia**

Desde esta pantalla, Salidas realiza el llenado de Cliente

**Figura 3.38: Nuevo Cliente**



**Fuente: Elaboración propia**

una vez registrado la información del Cliente se muestra el registro el cual puede ser editado y luego se guarda actualizando datos

**Figura 3.39: Pantalla editar Cliente**

The screenshot shows the 'Editar Cliente' form in the ADVentas system. The form is titled 'Cliente' and contains the following fields and controls:

- Nombre:** A text input field containing 'Personal de Seguridad'.
- Turno:** A dropdown menu with 'Mañana' selected.
- Código de Atención:** A text input field containing '010'.
- Área designada:** A text input field containing 'Puerta Principal'.
- Teléfono:** A text input field containing '2861236'.
- Buttons:** 'Guardar' (Save) and 'Cancelar' (Cancel).

The left sidebar shows the navigation menu with options like 'Escritorio', 'Administración', 'Registros', 'Salidas', 'Acceso', 'Consulta Ingresos', 'Consulta Salidas', 'Ayuda', and 'Acercas De...'. The footer includes 'Copyright © 2022-2025 Servialimut. Todos los derechos reservados.' and 'Version 1.0.1'.

**Fuente: Elaboración propia**

En esta pantalla se realiza las búsquedas de la información del Cliente esta puede ser buscado por código y nombre actualizando la pantalla.

**Figura 3.40: Búsqueda de Encargado**

The screenshot shows the 'Búsqueda de Encargado' screen in the ADVentas system. The search bar contains '010' and a blue arrow points to the search button. The table below shows the search results:

Opciones	Nombre	Designación	Código	Teléfono
 	Personal de Seguridad	MAÑANA	010	2861236

The footer includes 'Copyright © 2022-2025 Servialimut. Todos los derechos reservados.' and 'Version 1.0.1'.

**Fuente: Elaboración propia**

Desde esta pantalla, Salidas realiza el llenado de la Venta

**Figura 3.41: Nuevo Venta**

ADVentas

Richard Vujra

Venta

Cliente[\*]: Personal de Seguridad Fecha[\*]: 11/06/2022

Turno[\*]: Mañana Código: 02 + Agregar Alimentación

Opciones	Alimentación	Cantidad	Precio Venta	Descuento	Subtotal
X	Jugo de papaya	1	9,00	0	9
TOTAL					S/. 9

Guardar Cancelar

Copyright © 2022-2025 Servialnut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

una vez registrado la información de la Venta se muestra el registro el cual no puede ser editado y luego se guarda actualizando datos.

**Figura 3.42: Pantalla detalle Venta**

ADVentas

Richard Vujra

Venta

Cliente[\*]: Personal de Seguridad Fecha[\*]: 04/06/2022

Turno[\*]: Mañana Código: 02

Opciones	Artículo	Cantidad	Precio Venta	Descuento	Subtotal
	Jugo de papaya	2	9.00	0.00	18.00
TOTAL					S/. 18

Cancelar

Copyright © 2022-2025 Servialnut. Todos los derechos reservados. Version 1.0.1

**Fuente: Elaboración propia**

Acceso a Permisos, en esta parte los tipos de usuarios son: Administrador, Encargado de Ingresos, Encargado de Salidas (ventas).

**Figura 3.43: elegir permisos**

ADVentas

Richard Vujia

Usuario

Nombre(\*):  
Nombre

Tipo Documento(\*):  
Ci

Número(\*):  
Documento

Dirección:  
Dirección

Teléfono:  
Teléfono

Email:  
Email

Cargo:  
Cargo

Login (\*):  
Login

Clave (\*):  
Clave

Imagen:  
Seleccionar archivo | Ninguno archivo selec.

Permisos:  
 Escritorio  
 Administración  
 Registros  
 Salidas  
 Acceso  
 Consulta Ingresos  
 Consulta Salidas

Guardar Cancelar

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

Fuente: Elaboración propia

Se tiene las siguientes opciones al acceso de permisos dependiendo al tipo de usuario.

**Figura 3.44: Permisos del Administrador**

ADVentas

Richard Vujia

Usuario

Nombre(\*):  
Angela Quecaña

Tipo Documento(\*):  
Ci

Número(\*):  
99095624

Dirección:  
Zona/San Jose de yunguyo

Teléfono:  
77231248

Email:  
angelaquecana@gmail.com

Cargo:  
Vendedor

Login (\*):  
Angela123

Clave (\*):  
.....

Imagen:  
Seleccionar archivo | Ninguno archivo selec.

Permisos:  
 Escritorio  
 Administración  
 Registros  
 Salidas  
 Acceso  
 Consulta Ingresos  
 Consulta Salidas

Guardar Cancelar

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

Fuente: Elaboración propia

### 3.6 FASE DE TRANSICIÓN

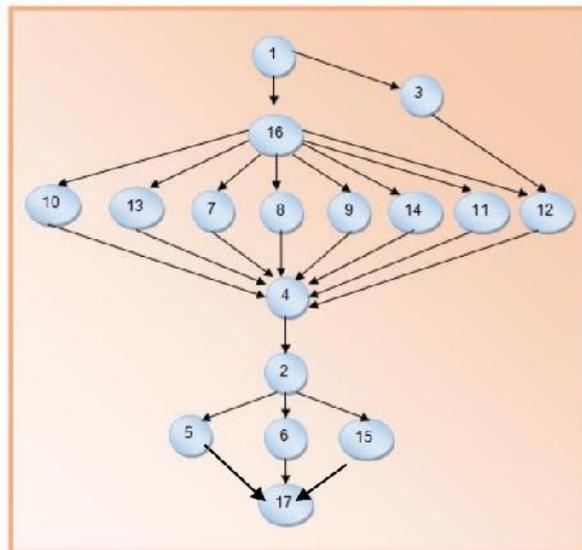
En esta fase luego de haber concluido con la fase de implementación y de haber realizado las pruebas necesarias dentro de su desarrollo se obtendrá el producto finalizado y listo para experimentarlo en el entorno de los usuarios finales, y dejar que manipulen el sistema con datos reales, y dar un compás de espera para ver si el usuario no tiene ningún inconveniente, o más aún si el sistema tiene algún error, o provoca una salida no deseada dar el pronto mantenimiento de dicho sistema.

#### 3.6.1 Caja Blanca

Son pruebas estructurales, conociendo el código y siguiendo su estructura lógica, se pueden diseñar destinadas a comprobar que el código hace correctamente lo que diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones. Ejemplos típicos de ello son las pruebas unitarias.

Se tomaron puntos vulnerables para evaluaciones como la estructura de la base de datos, las validaciones, manejo de errores. También fueron evaluados que los principales procesos funcionen correctamente acorde a los requisitos del usuario.

Esta prueba se orienta al cálculo de las regiones que deben ser consideradas como parte independientes del sistema, y estableciendo cuales son las entidades que se ejecutan en cada una de las regiones, asegurando así que cada región se ejecuta al menos una vez. De forma general se emplea el diseño del sistema para elaborar el grafo del programa de la siguiente forma:



### 3.7 METRICAS DE CALIDAD

#### 3.7.1 Introducción

Para evaluar este proyecto se aplican las métricas de calidad según el estándar ISO 9126, evaluándose la fiabilidad, funcionabilidad, portabilidad, usabilidad y mantenimiento del sistema.

#### 3.7.2 Fiabilidad

Durante la entrega de los prototipos realizados en las iteraciones se encontraron las siguientes deficiencias. A los mismos que necesitaron de un tiempo de servicio determinado.

Se logra levantar la siguiente información que se muestra en la siguiente tabla

**Tabla 3.5 Fiabilidad**

Tiempo de servicio	Número de peticiones	Fallas encontradas	Probabilidad de fallo	Tiempo medio entre fallos
8 horas	25	1	0.04	8
16 horas	50	2	0.04	8
32 horas	80	3	0.0375	10.7
64 horas	160	5	0.03125	12.8
<b>TOTAL</b>			0.149	39.5

Por lo tanto, el valor promedio de fallas producidas en un tiempo de servicio (PFTS) es de:

$$\text{PFTS} = \frac{0.04 + 0.04 + 0.0375 + 0.03125}{4} = \frac{0.149}{4} = 0.03725$$

Lo que indica que el sistema es promedio puede presentar aproximadamente 37 fallas cada mil peticiones.

El tiempo promedio en presentar estas fallas (TMF) está dado por:

$$\text{TMF} = \frac{8 + 8 + 10.7 + 12.8}{4} = \frac{39.5}{4} = 9.875$$

Lo cual indica que el sistema presenta una falla en promedio cada 9.875 horas en el que se hace el uso del sistema.

Entonces se tiene:

$$(1 - \text{PFTS}) \% = (1 - 0.03725) \% = (0.9628) = 96.28\%$$

Lo que nos indica que el sistema de control y administración de la empresa SERVIALINUT tiene la capacidad de ser utilizado libre de errores con un 96.28% de seguridad y con una probabilidad de que suceda una falla cada 25 peticiones.

**Tabla 3.6 Análisis de Fiabilidad**

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
<b>Fiabilidad</b>	96.28 %	Resultado obtenido durante la etapa de las iteraciones	Con el transcurso de tiempo de uso del sistema en las futuras versiones mejoradas se pretende incrementar aún más la fiabilidad

Es conveniente hacer notar que estos datos fueron levantados en etapa de iteraciones y con el transcurso de tiempo el mismo el mismo tendrá a no presentar fallas haciendo de la fiabilidad una calidad de software, es así que se vuelve a levantar los datos en la finalización del proyecto según la metodología y se obtiene los siguientes datos:

**Tabla 3.7 Fiabilidad a la finalización del proyecto**

Tiempo de servicio	Número de peticiones	Fallas encontradas	Probabilidad de fallo	Tiempo medio entre fallos
<b>8 horas</b>	25	1	0.04	8
<b>16 horas</b>	50	1	0.02	16

<b>32 horas</b>	80	1	0.0125	32
<b>64 horas</b>	160	2	0.0125	32
<b>TOTAL</b>			0.085	88

$$PFTS = \frac{0.04 + 0.02 + 0.0125 + 0.0125}{4} = \frac{0.085}{4} = 0.03725$$

$$(1 - PFTS) \% = (1 - 0.03725) \% = (0.96275) = 96.28 \%$$

Lo que nos indica que el sistema web de seguimiento y control administrativo de la empresa SERVIALINUT tiene capacidad de ser utilizado libre de errores con un 96.28 % de seguridad.

**Tabla 3.8 Análisis de Fiabilidad a la finalización del proyecto**

CARACTERISTICAS	VALOR	VALOR FINALIZACION	OBSERVACION
<b>Fiabilidad</b>	96.28 %	97.87 %	Se puede ver que existe una mejora en la fiabilidad

### 3.7.3 Funcionabilidad

Se hace uso de las métricas de adecuación funcional y completitud de la implementación funcional que nos ayuda a medir la implementación funcional y que tan adecuadas son las funciones evaluadas.

#### 3.7.3.1 Completitud de la implementación funcional

En total el presente proyecto cuenta con historias de usuario. El desarrollo del proyecto se hizo mediante iteraciones, de tal manera que podemos obtener la completitud de la implementación funcional de la siguiente manera:

En la primera iteración se implementó 3 historias de usuario que son: Registro de usuarios, registro de encargados, registro de ingresos

$$X = 1 - \frac{A \text{ (HU implementadas)}}{B \text{ (HU definidas en fase de exploración)}} = 1 - \frac{3}{7} = 0.57$$

Lo que nos indica que falta un 57 % del sistema por implementar.

En la segunda iteración se implementó historias de usuario: Verifica el estado de los detalles al ingresar registros sumadas con la primera iteración tenemos aquí 4 historias de usuario.

$$X = 1 - \frac{A \text{ (HU implementadas)}}{B \text{ (HU definidas en fase de exploración)}} = 1 - \frac{4}{7} = 0.43$$

En la segunda iteración se implementó historias otras 2 historias de usuario: Verifica el estado de los detalles de ingresos y salidas sumadas con la primera iteración tenemos aquí 6 historias de usuario.

$$X = 1 - \frac{A \text{ (HU implementadas)}}{B \text{ (HU definidas en fase de exploración)}} = 1 - \frac{6}{7} = 0.14$$

En la cuarta iteración se implementó la restante historia de usuario: Genera reportes de ingresos, salidas, clientes, encargados y detalles. Con esta historia final de X es cero. Lo que indica que las historias de usuario descritas en la fase de exploración fueron resueltas al finalizar del proyecto

**Tabla 3.9 Análisis de Funcionalidad**

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
<b>Fiabilidad</b>	71.4 %	Resultado obtenido durante la etapa de las iteraciones	Con el trascurso de tiempo de uso del sistema en las futuras versiones se pretende mejorar la adecuación funcional de las historias de usuario.

A la finalización del proyecto se vuelve a medir la adecuación funcional y se obtiene un solo problema de que los funcionarios deberían poder ser garantes de otros funcionarios, el nuevo resultado es:

$$X = 1 - \frac{A \text{ (HU donde se detectaron problemas)}}{B \text{ (HU evaluadas)}} = 1 - \frac{1}{7} = 0.857$$

**Tabla 3.9 Mejora de la Funcionalidad a la finalización del proyecto**

CARACTERISTICAS	VALOR	VALOR FINALIZACION	OBSERVACION
<b>Fiabilidad</b>	71.4 %	85.7 %	Se puede ver que existe una mejora en la funcionabilidad

### Consistencia operacional en el uso

Viene dado de la siguiente manera:

$$X = 1 - \frac{A \text{ (# funciones que se encontraron inconsistencias)}}{B \text{ (# usos por usuario en el periodo de prueba)}} = 1 - \frac{2}{15} = 0.867$$

Existe un 86.7 % de satisfacción del usuario en consistencia operacional de uso del sistema.

**Tabla 3.10 Análisis de los datos de la fiabilidad**

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
<b>Fiabilidad</b>	95.2 %	Resultado obtenido durante la etapa de las iteraciones	Con el trascurso de tiempo de uso del sistema en las futuras versiones se pretende que se tienda a incrementar el valor de la consistencia operacional

<b>Consistencia operacional en el uso</b>	86.7 %	Resultado obtenido durante la etapa de iteraciones	Se pretende incrementar con el transcurso del tiempo y uso del sistema en las futuras versiones la satisfacción del usuario en la consistencia
---	--------	--	--

### 3.8 MANTENIMIENTO

**Capacidad para ser analizado:** El sistema web tiene la capacidad para ser diagnosticado para identificar partes que han sido modificados si fuera el caso.

**Confiabilidad:** Debido a su arquitectura MVC (Modelo Vista Controlador) y la capacidad del software que permite que una determinada modificación sea implementada, es posible con el código fuente debidamente documentado siendo posible ampliar el sistema con futuras funciones.

Obteniendo un resumen de todas las mediciones se tiene:

**Tabla 3.10 Resumen métricas de calidad**

<b>Medida</b>	<b>Valor</b>
Fiabilidad	97.87 %
Funcionabilidad	85.70 %
Usabilidad	93.30 %
Portabilidad	100 %
<b>PROMEDIO</b>	<b>94.22 %</b>

Todos los valores obtenidos están dentro de los márgenes de los estándares de calidad ISO

9126. por lo tanto se concluye con la valoración que se obtuvo alcanzo una calidad del 94.22 % lo cual representa un grado de satisfacción.

### 3.9 ANALISIS DE COSTOS

Costo de desarrollo del software, costo de capacitación del usuario y costo del software base para la construcción del sistema.

#### Costo de desarrollo del software

La obtención de este precio se realizó sobre la base del método de estimación empírica COCOMO básico, que estima el esfuerzo E expresada en personas-mes y el tiempo de duración del desarrollo del software T mediante las siguientes ecuaciones

$$\begin{aligned}K_m = E &= a_i S_k^{b_i} & K_m = E &= 3.0 * 5^{1.12} \\ & & &= 3.0 * 6.07 \\ T_d &= 2.5 K_m^{0.35} & &= 18.21\end{aligned}$$

Donde E y K son el número de personas necesarias por mes, S el tamaño expresado en miles de líneas de código fuente, y a, b son obtenidos mediante una tabla de coeficientes.

Dadas las características del software se pueden afirmar que el mismo está comprendido entre grupo de sistemas Semi encargado.

Modelos	Vistas	Controladores	TOTAL
1829	1690	1938	5457

En este caso se toma un valor de 5 mil líneas de código estimadas

Dado estos valores se tiene:

Luego

$$\begin{aligned}T_d &= 2.5 * 18.21^{0.35} \\ &= 2.5 * 2.76 \\ &= 6.9\end{aligned}$$

De qui se obtiene una estimación de esfuerzo de 18 personas-mes en un tiempo de 6 meses, se obtiene el total de personas necesarias.

$$P = \frac{K_m}{T_d} = \frac{18.21}{6.9} = 2.64 \cong 3$$

Considerando el sueldo de un desarrollador depende de experiencia del mismo se da un valor según la oferta de los programadores en el mercado actual boliviano, 3500 Bs al mes El costo estimado del software es:

$$\begin{aligned} \text{COSTO DE DESARROLLO} &= 3 \times 3500 \\ &= 10500 \text{ Bs} \end{aligned}$$

#### **Costo de capacitación de los usuarios**

Este costo esta dado por producto del tiempo de capacitación, dos horas de 50Bs la hora), por el número de días de capacitación que en este caso fueron 5, hacen un total de

$$\begin{aligned} \text{COSTO DE CAPACITACION} &= 5 \times 50 \\ &= 250\text{Bs} \end{aligned}$$

#### **Costo del software base para el desarrollo del sistema**

Los precios de licencia del lenguaje de programación y el motor de base de datos no se contemplan ya que son libres.

$$\text{COSTO DE SOTFWARE BASE} = 0 \text{ Bs}$$

Haciendo un resumen de costos se obtiene el costo estimado del proyecto

**Tabla 4.13 costo del proyecto**

COSTO DE DESARROLLO	10500
COSTO DE CAPACITACION	250
COSTO DE SOTFWARE	0
<b>TOTAL</b>	<b>10750</b>

Es decir que el costo estimado del proyecto es de 10750 Bs

# **4 CAPITULO IV**

## **CONCLUSIONES Y RECOMENDACIONES**

### **4.1 CONCLUSIONES**

Después de terminar el desarrollo del sistema para la empresa SERVIALINUT se llegaron a las siguientes conclusiones.

- ) Se logro automatizar la información de los clientes rápida y accesible
- ) Se logro brindar información oportuna y actualizada de los clientes y sus servicios.
- ) Se genero el registro de inventario de menús más accesible.
- ) Se implemento la información sobre los servicios que ofrece la empresa a los clientes.
- ) Se implemento datos estadísticos para tener un informe detallado al fin de mes.
- ) Se estableció la designación de roles para una mejor atención

El sistema web para el control de administración de la empresa SERVIALINUT ha sido desarrollado bajo las especificaciones planteadas cumpliendo así con los objetivos generales.

### **4.2 RECOMENDACIONES**

Las recomendaciones que se pueden realizar para futuros proyectos basados en este están referidas a continuación:

El sistema también puede crear reportes tanto como registros de cada encargado.

Se puede sugerir una implementación de control de almacén para que la empresa cuente con un informe de los víveres que tiene y así poder evitar pérdidas de su servicio.

## 5 BIBLIOGRAFIA V

- Limachi Machaca, K. B. (2018). Sistema web para el control y seguimiento de ventas de productos artesanales caso: Bolivia Tech Hub (Doctoral dissertation).
- Peralta, T., & Briggith, A. (2018). Aplicación web para el control y seguimiento del rendimiento académico en estudiantes de secundaria de la Institución Educativa Privada Jesus el Maestro
- Vega Macías, E. J., & Villón Pesantes, D. E. (2021). Desarrollo de un prototipo en ambiente web para la gestión de control y seguimiento de los mantenimientos realizados a los vehículos en una lubricadora (Bachelor's thesis)
- LISSETTI, L. M. S. (2022). SISTEMA WEB PARA EL CONTROL DE RECEPCIÓN DE PEDIDOS Y SEGUIMIENTO DE RUTAS EN LA INDUSTRIA TRANSFORMADORA DE PLÁSTICO
- CUEVAS COYO, G. C. (2020). SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y SEGUIMIENTO ACADÉMICO.
- CONDORI QUISPE, L. F. (2020). SISTEMA DE INFORMACIÓN WEB DE CONTROL Y SEGUIMIENTO DE ASOCIADOS.
- Merino, J. P. (2012). Calificación.
- Meucci, J. C. (2008). SEGURIDAD.
- Olsina, D. L. (2003). Metricas de Calidad.
- Olsina, R. S.-D. (2008). Metodología UWE.
- Peña, I. S. (2014). Partes que Componen el sistema.
- Peña, J. M. (2011). Calidad de Software.
- Peño, J. M. (2015). Puebas de Sotfware.
- Pérez, J. E. (2008). Ajax.
- Pressman. (2010). En METODO DE ESTIMACION COCOMO II.
- Pressman, R. (2010). El Modelo COCOMO II.
- Prieto, R. M. (2017). LA NORMA ISO/IEC 9126. Obtenido de
- Redondo, F. (2018). Postman.

# ANEXOS

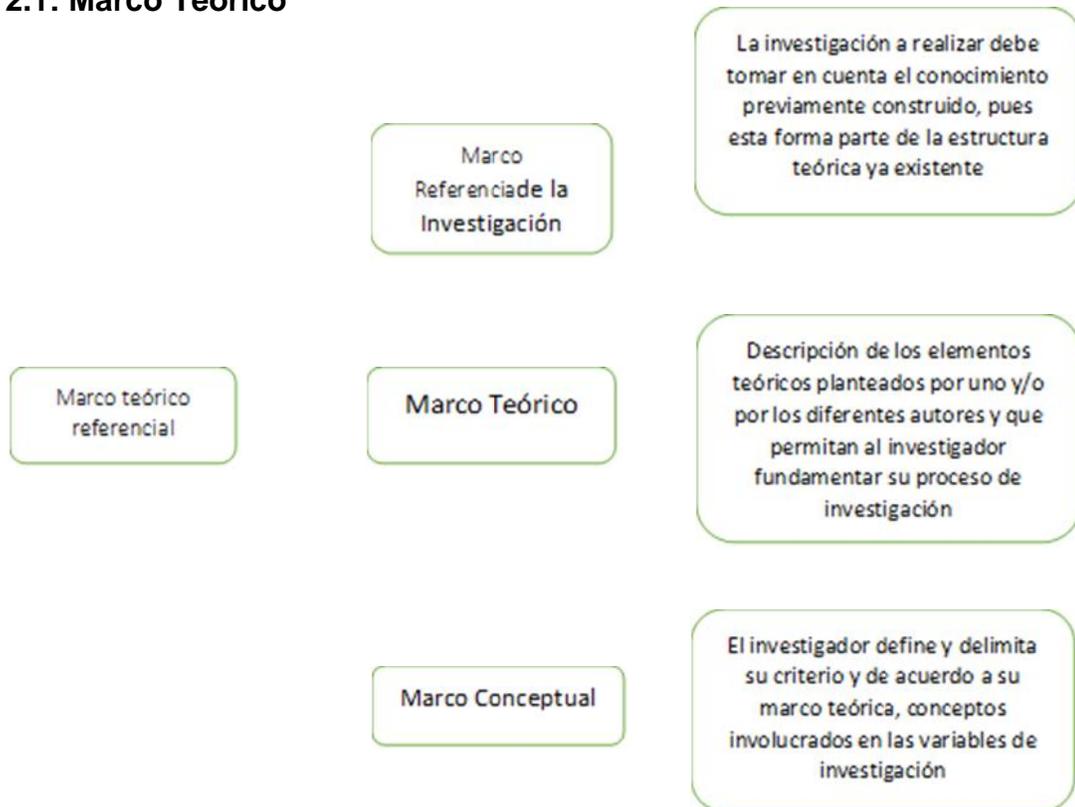
## ANEXOS A – ARBOL DE PROBLEMAS



## ANEXOS B – OBJETIVOS ESPECIFICOS



**Figura 2.1: Marco Teórico**



**Figura 2.2: Esquema Proceso Unificado Ágil**

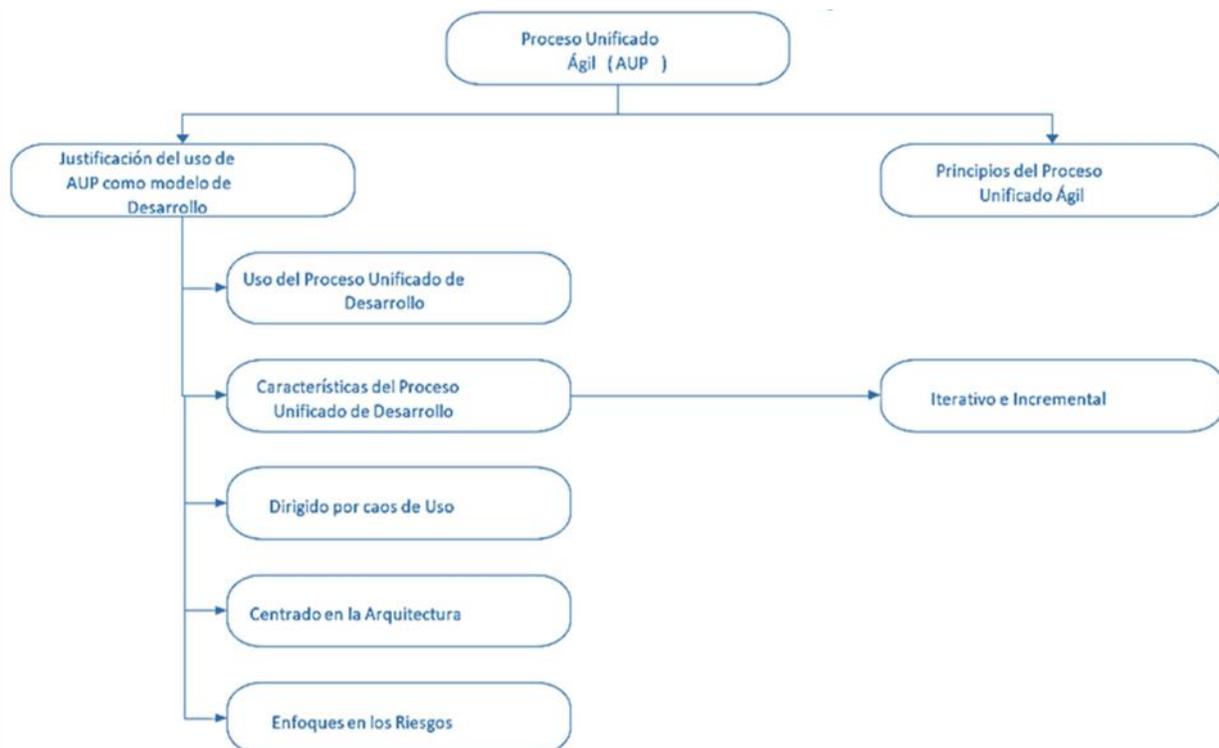


Figura 2.3: (Ciclo de Vida AUP) Fuente-Scott Ambler o (Agüe Inified Process AUP)

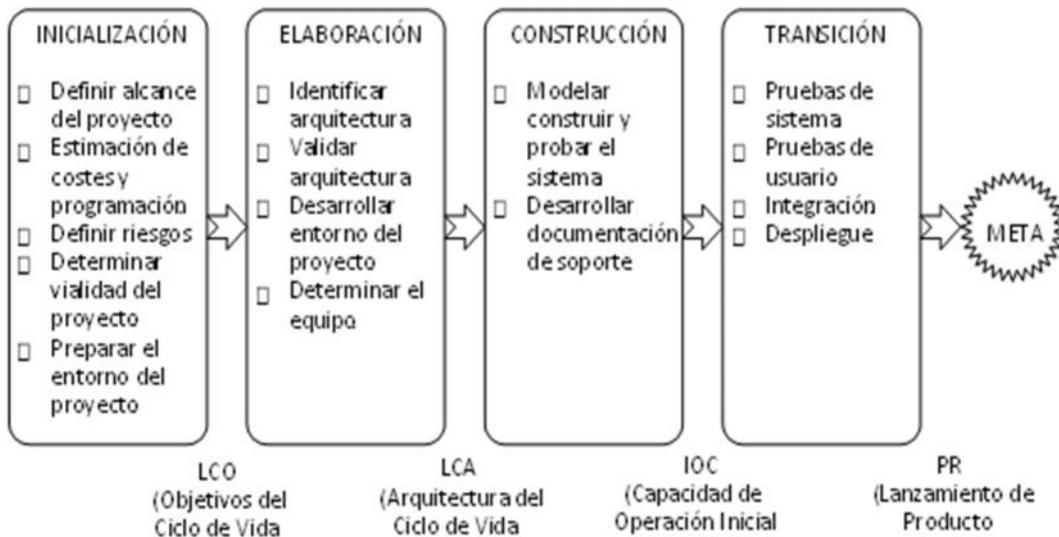
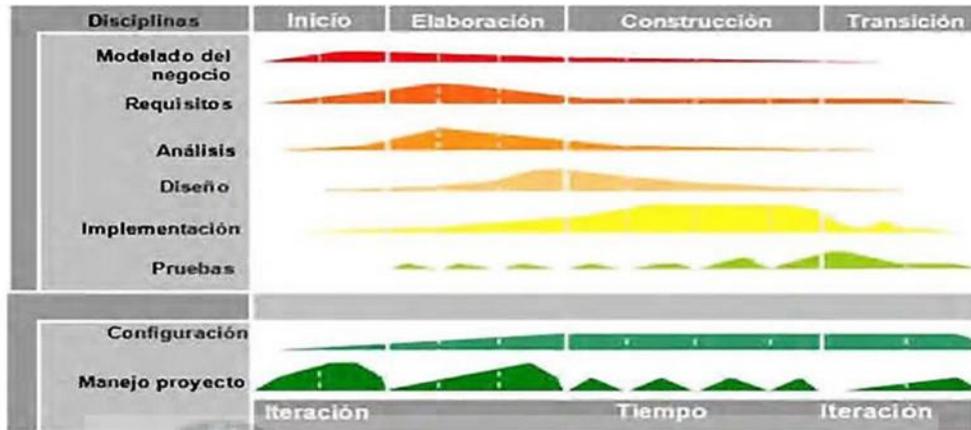
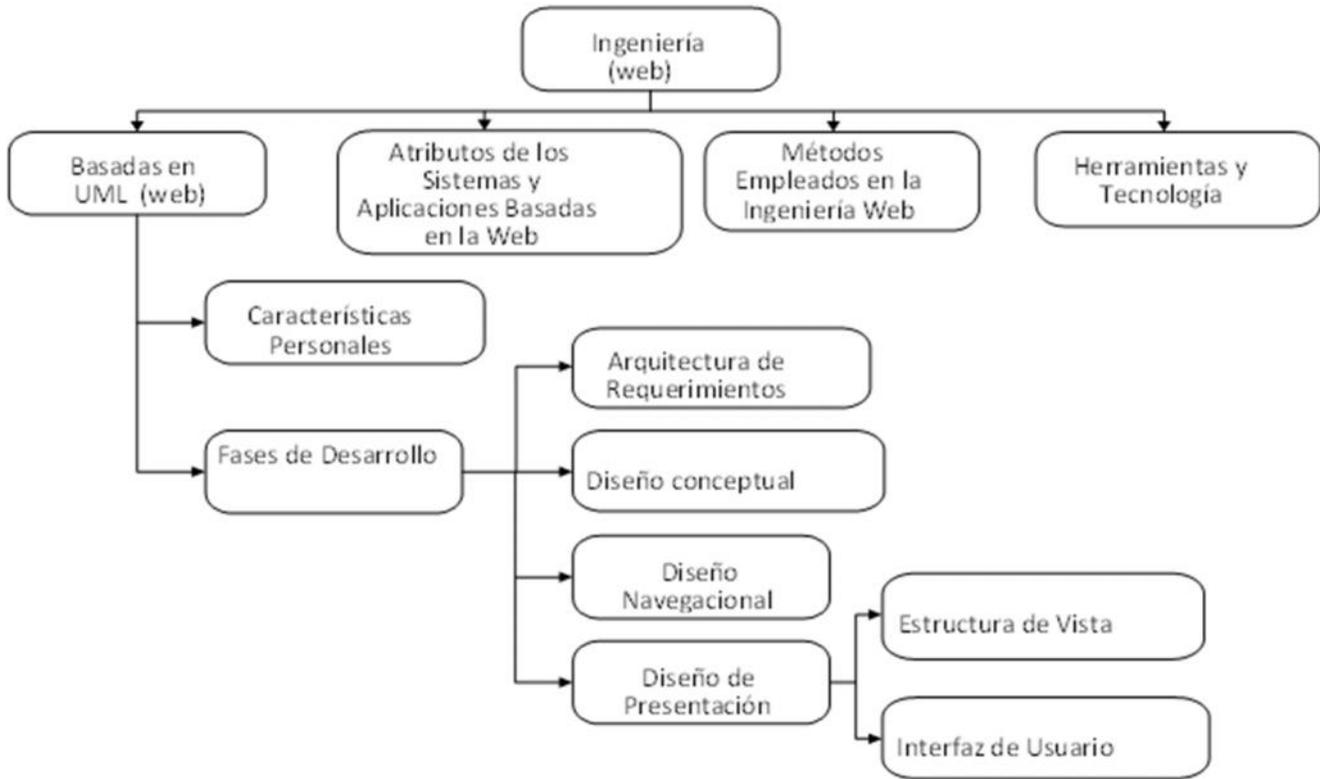


Figura 2.4: Fases del Proceso Unificado Ágil

**Figura 2.5: Esquema de Desarrollo-Ingeniería Web**



**Figura 2.6: Ingeniería Web Basada en UML**

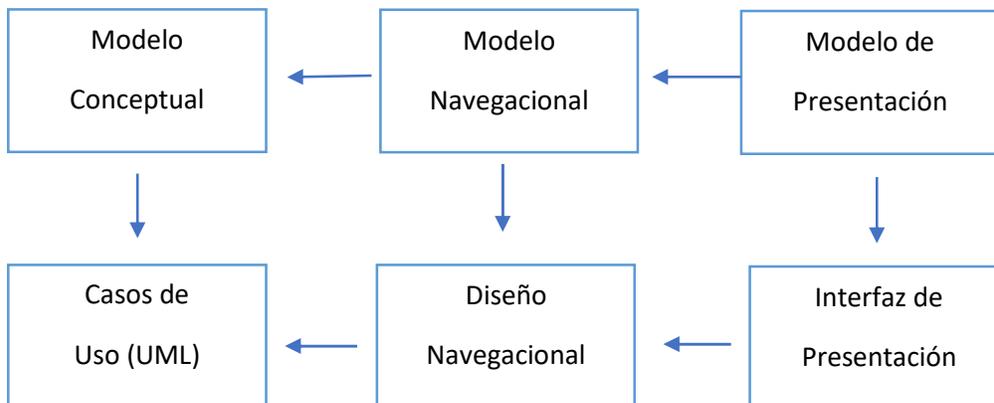


Figura 2.7: (IWeb Basado en UML) fuente Jesse James Garret 2000

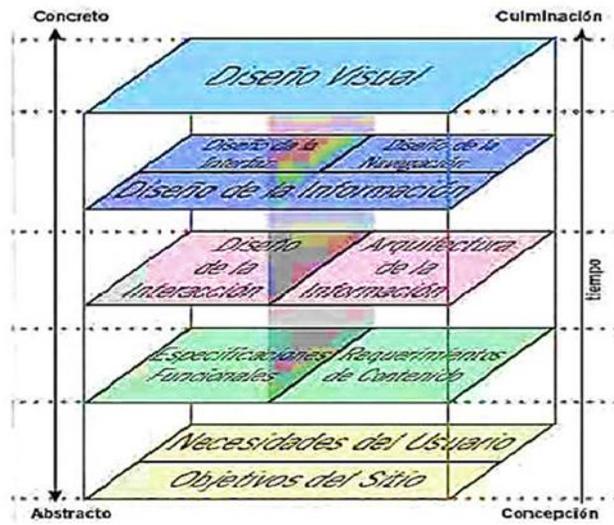
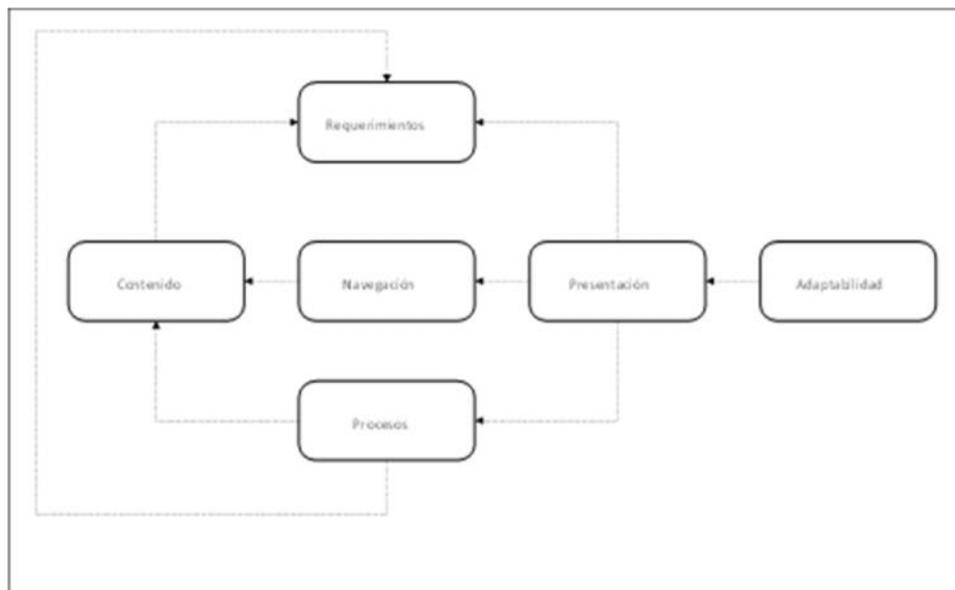


Figura 2.8: (Fases de Desarrollo UWE) Fuente-Nora Koch



**Figura 2.9: (Fases del UWE) Fuente-Nora Koch**



**Figura 3.1: Organigrama de la Empresa Servialinut**

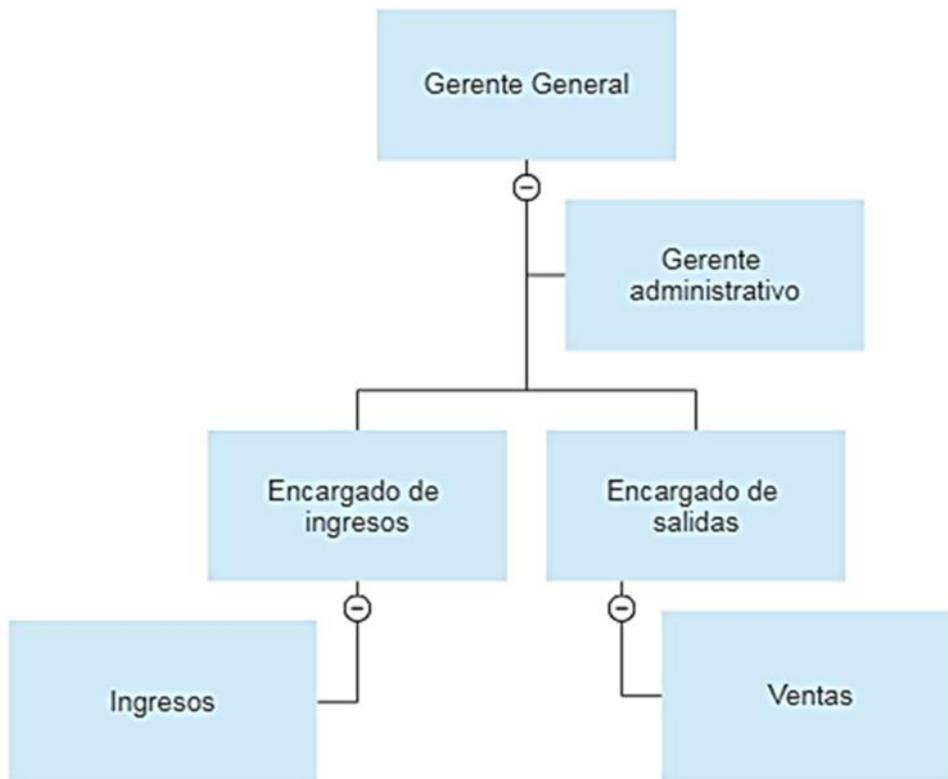
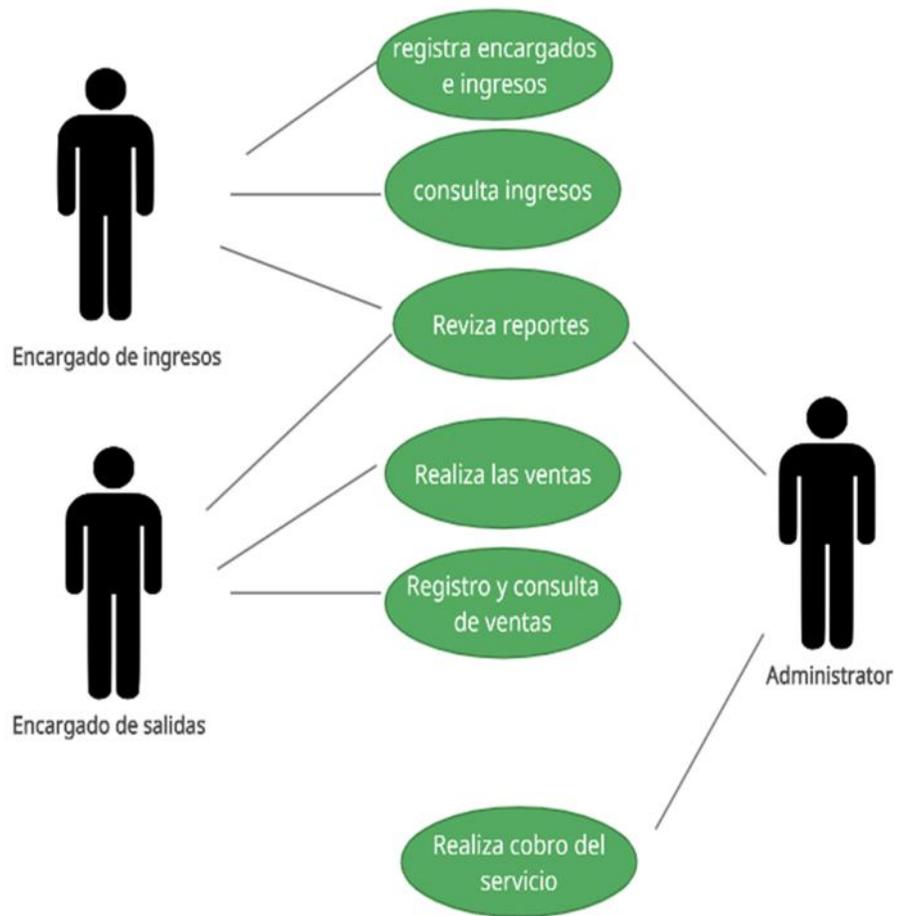


Figura 3.2: Modelo de negocio



**Figura 3.3: Modelo de Caso de Uso**



**Figura 3.4: Caso de Uso: Realiza Suscripción**



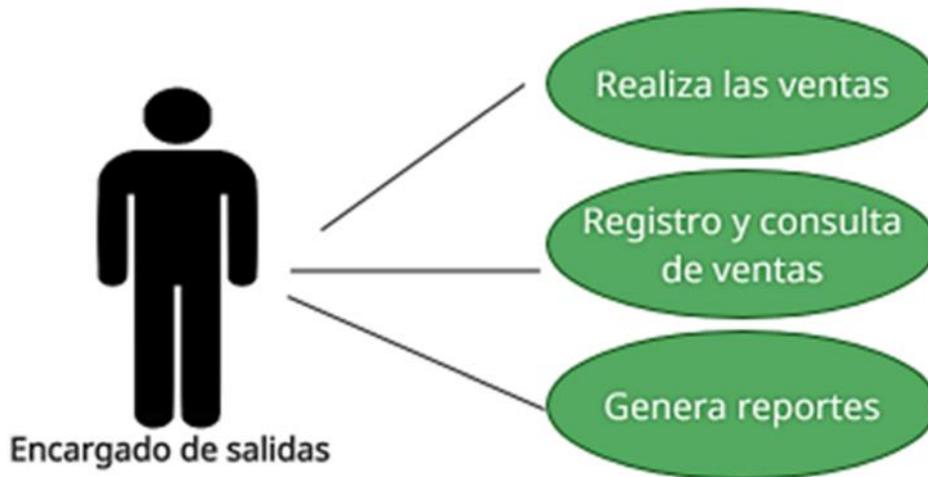
<b>Caso de Uso: Realiza el ingreso</b>	
Actores:	Encargado de ingreso
Propósito:	Realiza el registro, verificación y actualización de datos.
Resumen:	El encargado ingresa al sistema, realiza el registro de datos del cliente, posteriormente realiza el ingreso y finalmente la actualización de los datos ya registrados.
Tipo: Referencias	Primario
Referencias Cruzadas	
<b>Curso Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos  a) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. b) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Introduce los datos del cliente	4. Registra los datos introducidos
5. Confirma datos introducidos	6. Guarda datos
7. Se elige la opción detalle del cliente.	8. Mostrar el formulario y actualiza automáticamente los clientes
9. Realizar los cambios respectivos del cliente	10. Siempre que se haga un cambio. este se almacenará en el informe de actualización de clientes.
11. Guardar	12. Guardara la actualización
13. Para Salir de la tabla apretar cancelar	

**Tabla 3.1: Caso de Uso: Realiza el Ingreso**

**Fuente: Elaboración propia**

**Figura 3.5: Caso de Uso: Registra Ventas**

Fuente: Elaboración propia

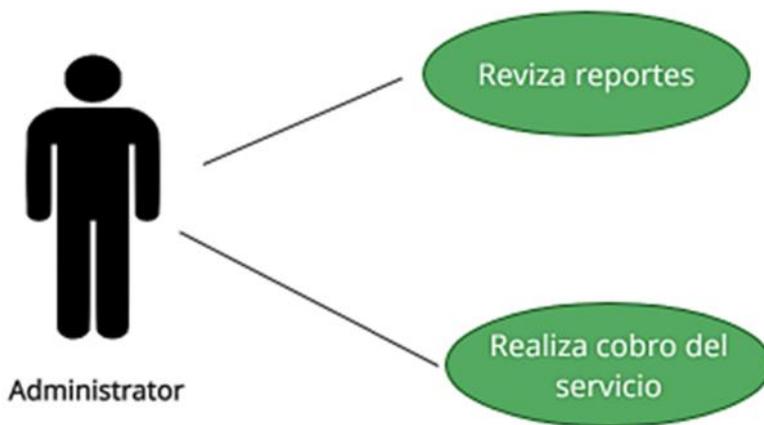


Caso de Uso: Registra ventas	
Actores:	Encargado de ventas
Propósito:	Realizar el registro de las ventas
Resumen:	El encargado de ventas ingresa al sistema, realiza el registro de los datos de las ventas de los medios de prensa, verifica medios de prensa más vendidos y posteriormente realiza el cobro de los servicios
Tipo: Referencias	Primario
Referencias Cruzadas	

Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos  e) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. d) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Elige la opción ventas	4. Nos muestra la interfaz de formulario de ventas para la adición de datos
5. Introduce los datos de la venta	6. Registra los datos introducidos
7. Confirma datos introducidos	8. Guarda datos
9. Se quiere realizar alguna otra operación volver al menú principal	10. saldrá al menú principal
11. Para salir del sistema apretar cancelar.	

**Tabla 3.2: Caso de Uso: Registra Ventas**

**Figura 3.6: Caso de Uso: Administrador**

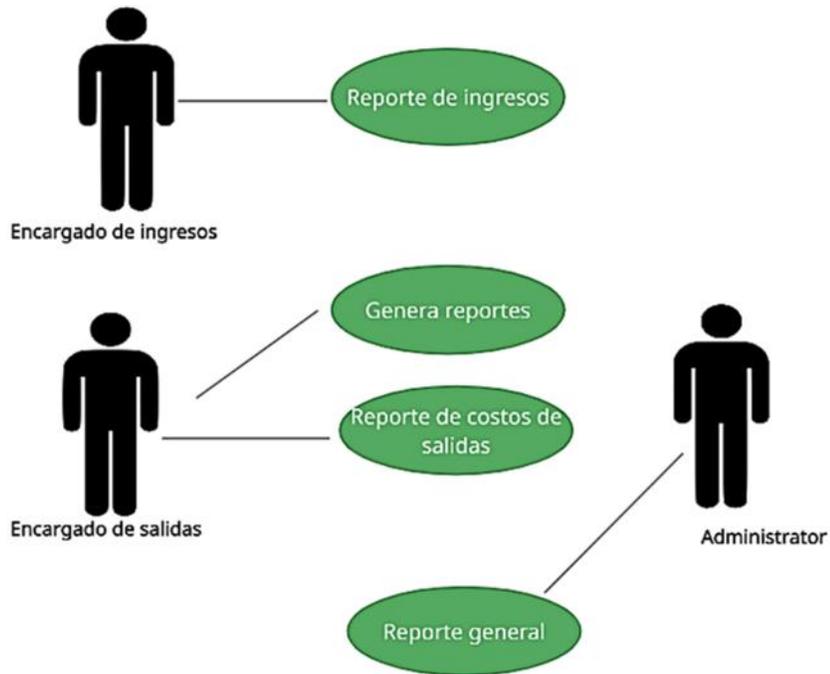


<b>Caso de Uso: Registro Administrativo</b>	
Actores:	Administración
Propósito:	Realizar el registro de las distribuciones
Resumen:	El encargado de administración ingresa al sistema, realiza el registro de la distribución de los tipos de alimentación que se proporcionara en el día, posteriormente realiza el informe de la distribución que se realiza en la empresa.
Tipo: Referencias	Primario
Referencias Cruzadas	

<b>Curso Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos
	a) Los datos son correctos, el sistema nos muestra la interfaz del menú principal.
	b) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Elige la opción Ingreso	4. Nos muestra la interfaz de formulario de ingreso para la adición de datos
5. Introduce los datos de la distribución de ingresos	6. Registra los datos introducidos
7. Confirma datos introducidos	8. Guarda datos
9. Apretar el botón imprimir formulario	10. Mandara el sistema a imprimir el formulario
11. Se quiere realizar alguna otra operación volver al menú principal	12. Saldrá al menú principal
13. Para salir del sistema apretar cancelar.	

**Tabla 3.3: Caso de Uso: Registro Administrativo Fuente:**

**Figura 3. 7: Caso de Uso: Generación de Reportes**



<b>Caso de Uso: Generación de Reportes</b>	
Actores:	Encargado de ingresos, Encargado de ventas y administración
Propósito:	Mostrar los distintos reportes para poder realizar el control de la información que se tiene.
Resumen:	El encargado de Ingresos, Encargado de ventas y Administración una vez que se decida qué tipo de información requiere, selecciona un reporte, posteriormente actualiza la información, con lo que se tiene la información, que puede ser impresa o que puede ser visualizada por pantalla
Tipo: Referencias	Primario
Referencias Cruzadas	

Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Introduce su nombre y password	2. El sistema constata si los datos introducidos son correctos  a) Los datos son correctos, el sistema nos muestra la interfaz del menú principal. b) Los datos son incorrectos, el sistema nos mostrara una nota de precaución al usuario.
3. Elige la opción reportes	4. El sistema visualiza la información a través de una pantalla
5. Elige si se imprime el reporte	
6. Para salir del sistema apretar cancelar.	

Tabla 3.4: Caso de Uso: Generación de Reportes

Figura 3.8: Diagrama de Secuencia: Realiza Ingreso

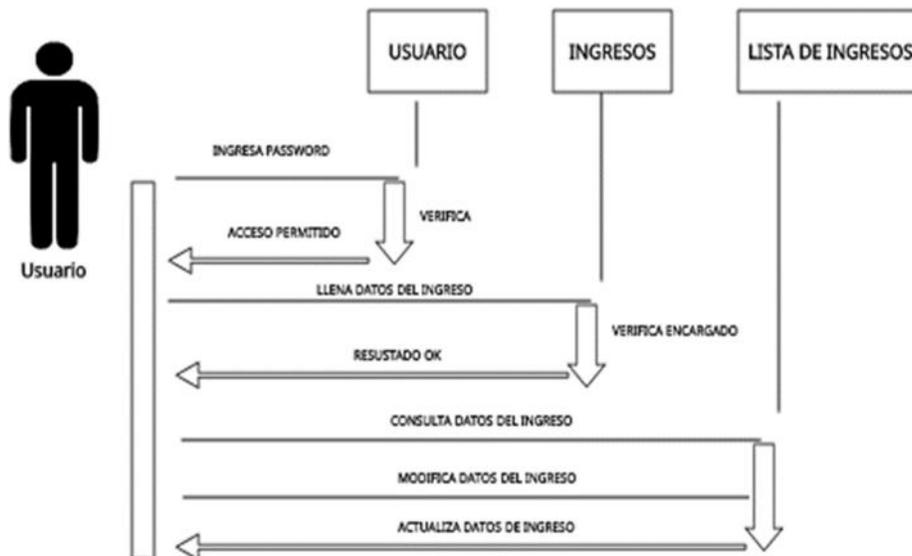


Figura 3.9: Diagrama de Secuencia: Registra Ventas

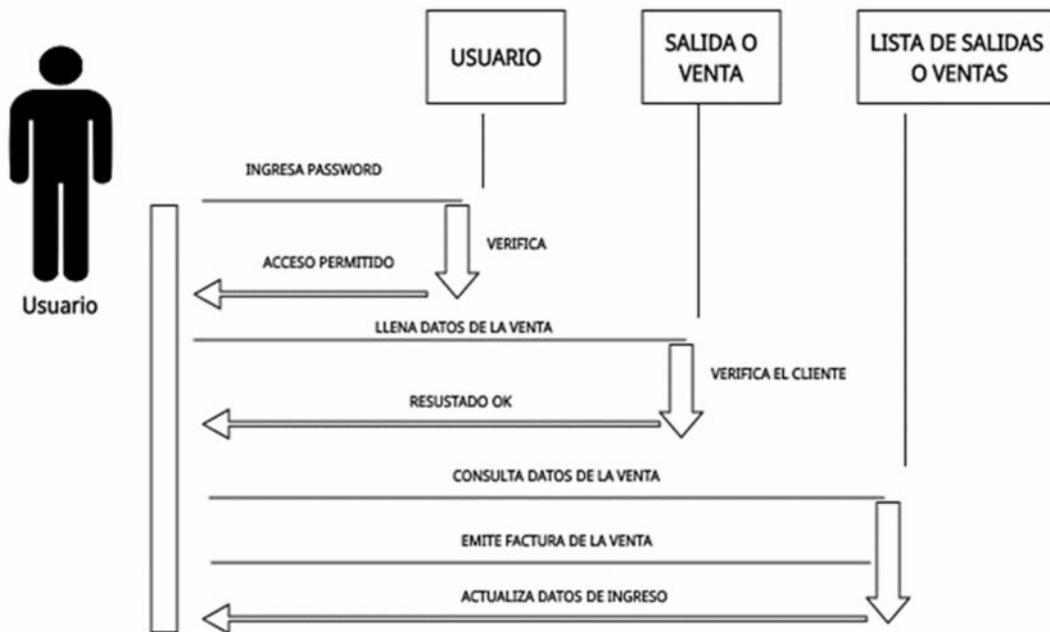


Figura 3.10: Diagrama de Secuencia: Registra Distribución

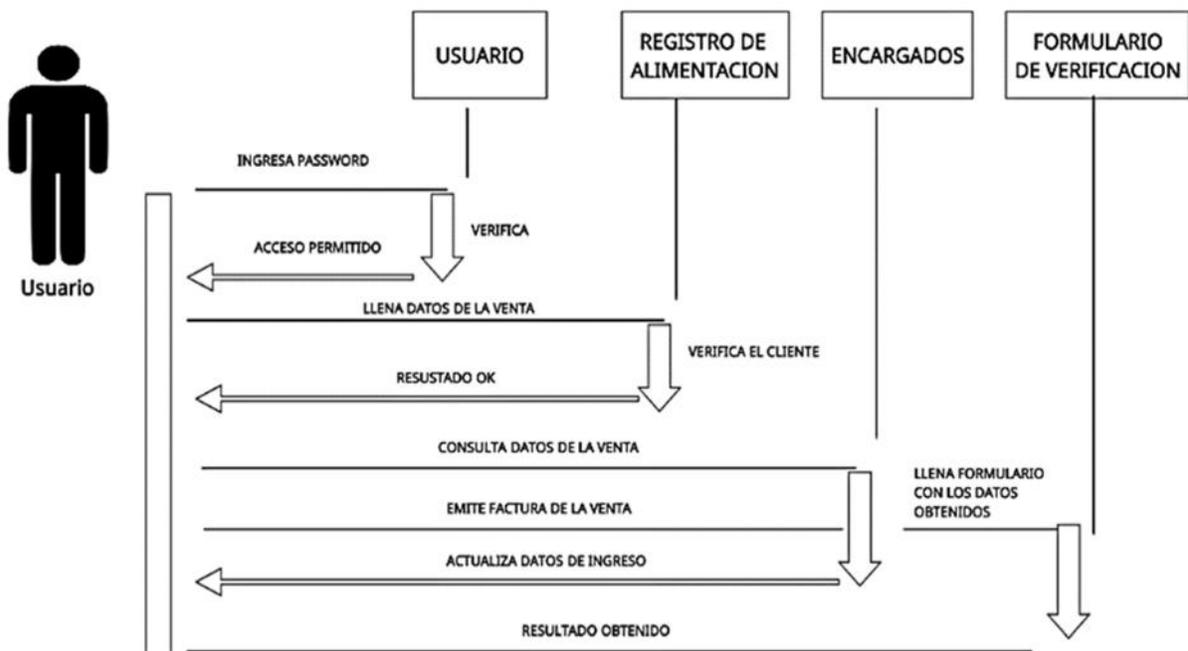


Figura 3.11: Diagrama de Secuencia: Generación de Reportes

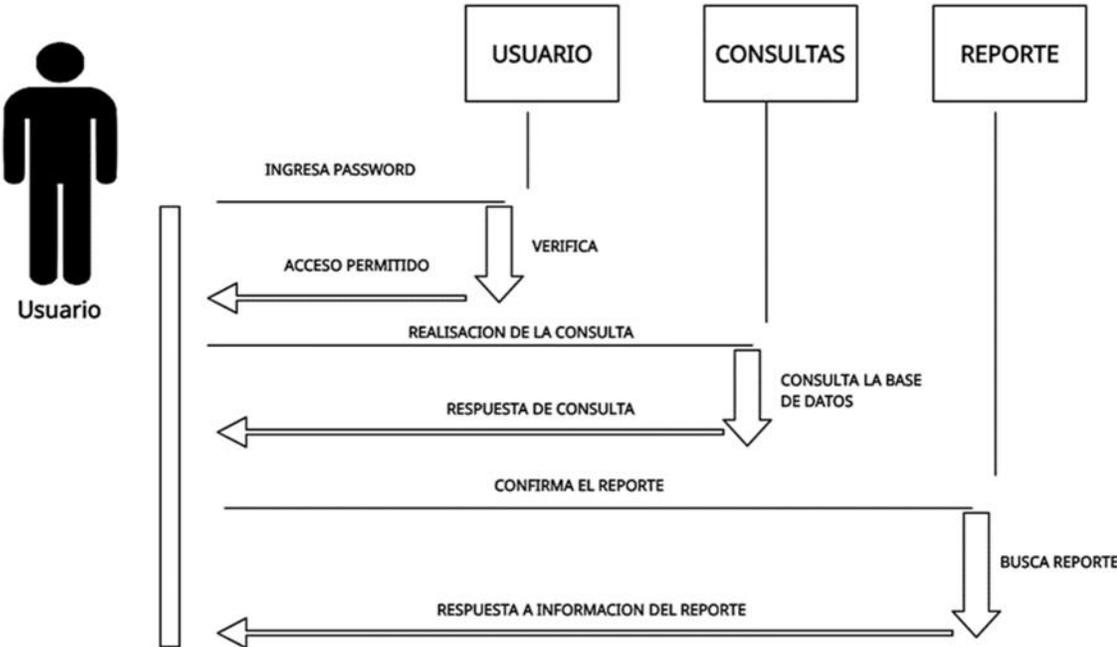


Figura 3.12: Diagrama de Ingreso: Realiza Ingreso

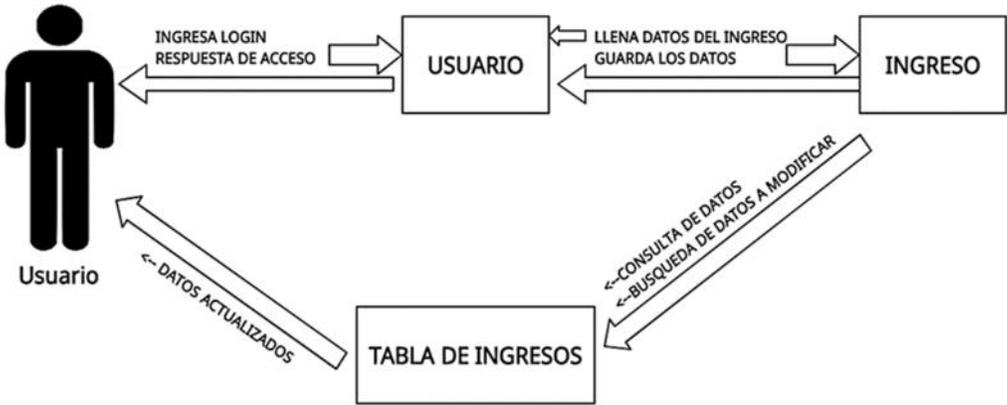


Figura 3.13: Diagrama de colaboración: Registra Ventas

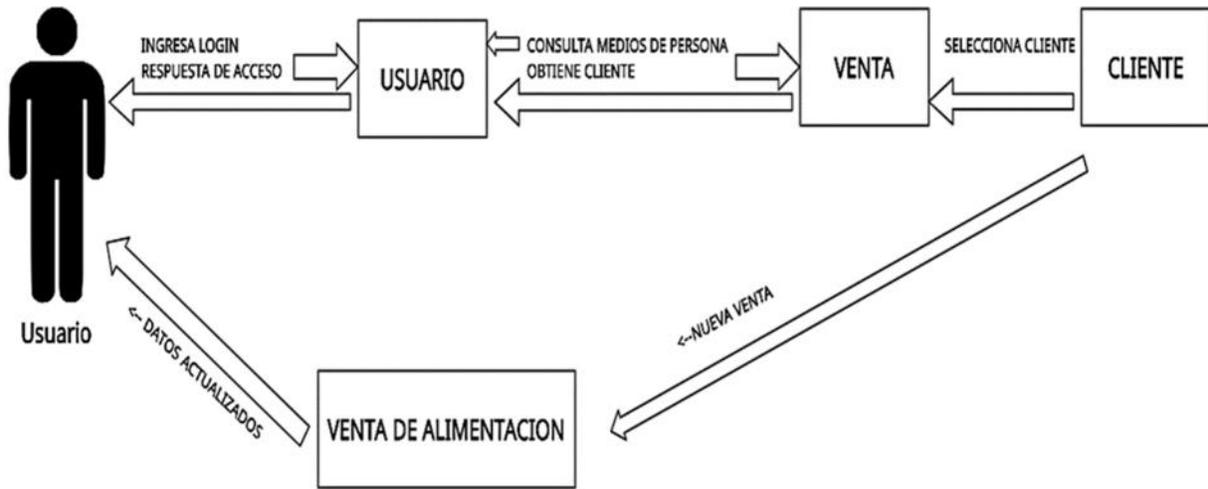


Figura 3.14: Diagrama de Colaboración: Registra Distribución

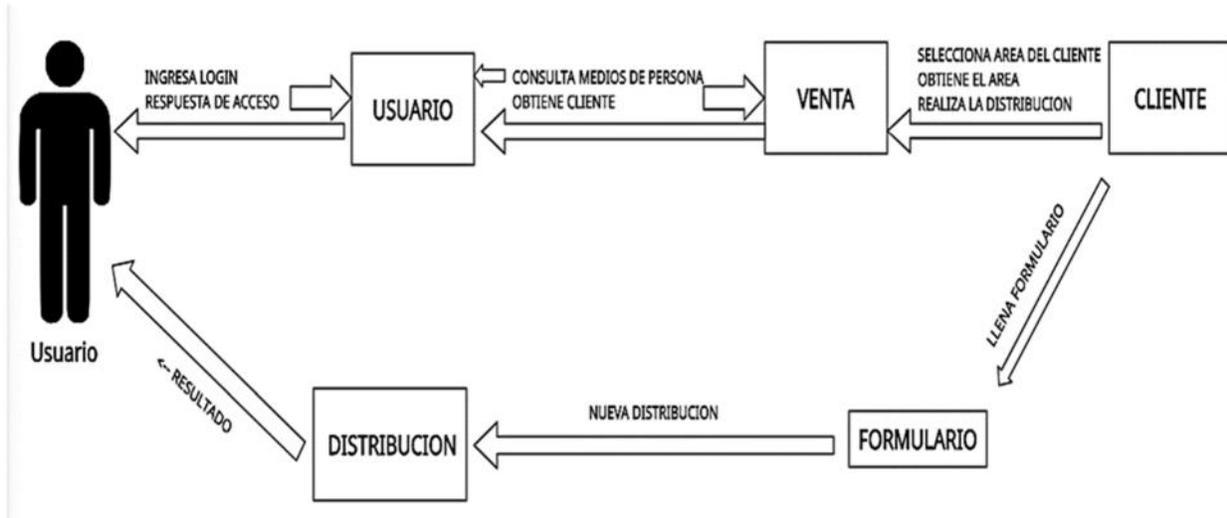


Figura 3.15: Diagrama de Colaboración: Generación de Reportes



Figura 3.16: Diagrama de Estados: Realiza Ingreso

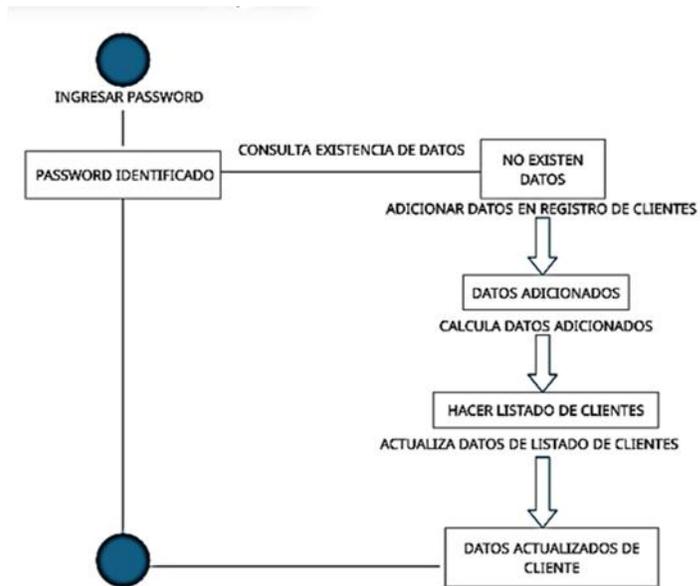


Figura 3.17: Diagrama de Estados: Registra Ventas

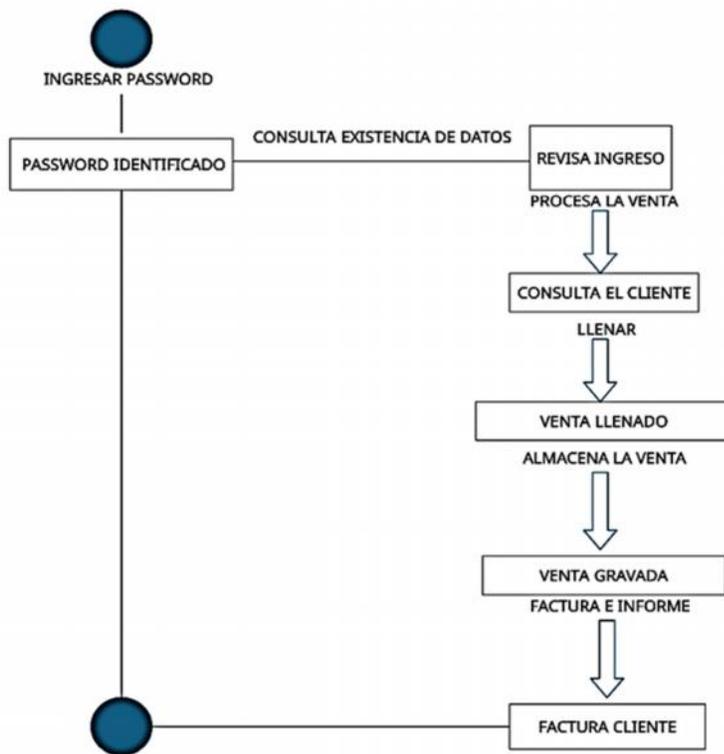


Figura 3.18: Diagrama de Estados: Registra Administrador

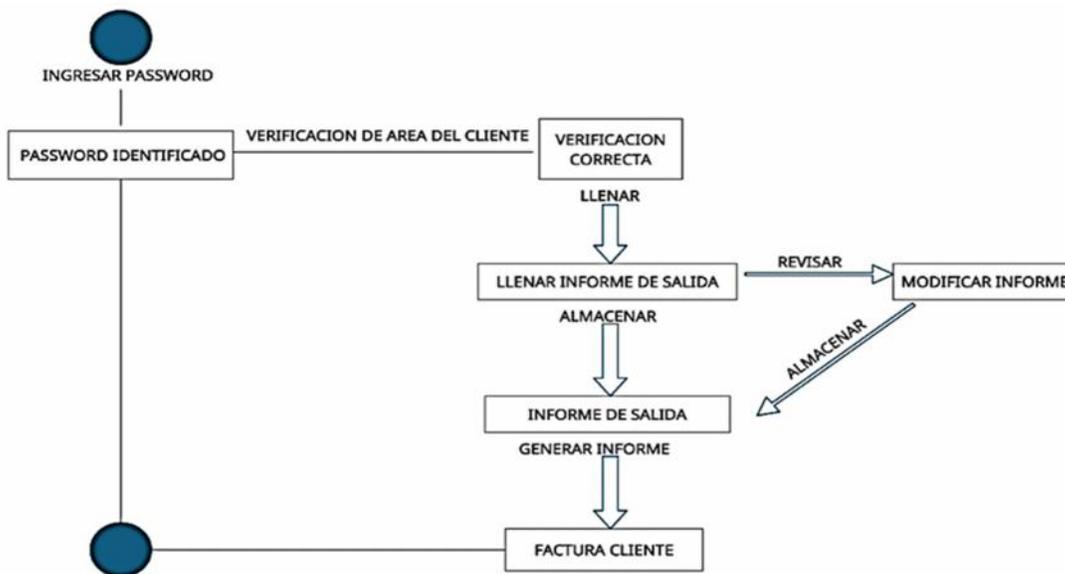


Figura 3.19: Diagrama de Estados: Generación de Reportes

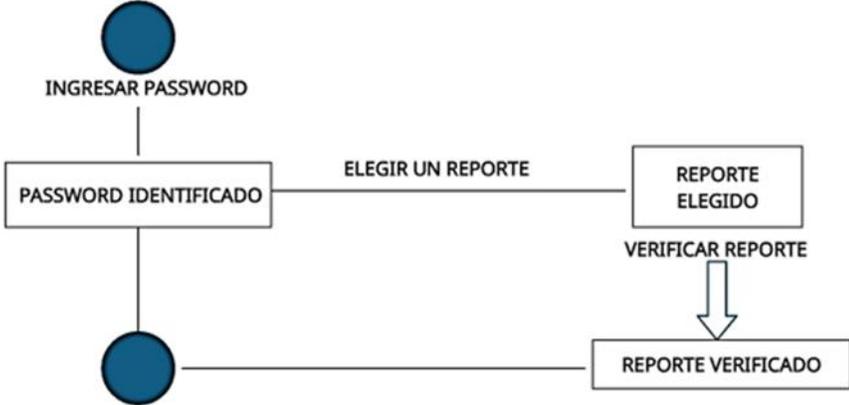


Figura 3.20: Diagrama de Clases

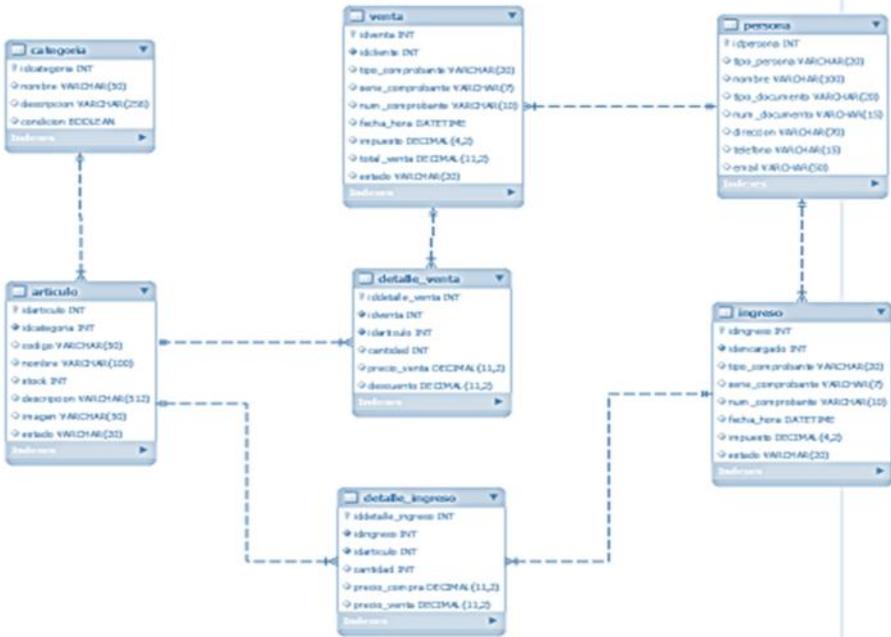


Figura 3.21: Diagrama de Componentes

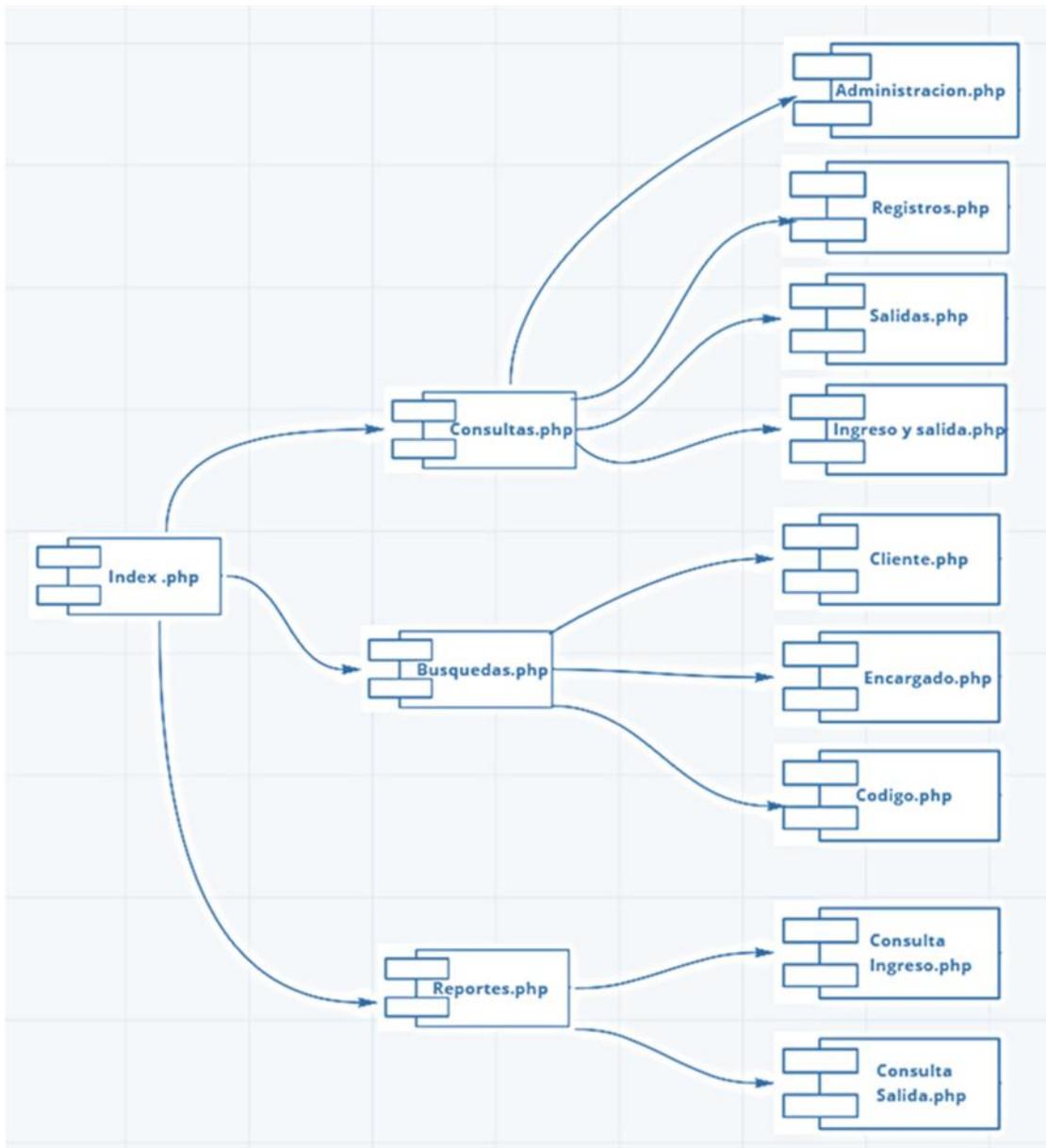


Figura 3.22: Arquitectura del sistema

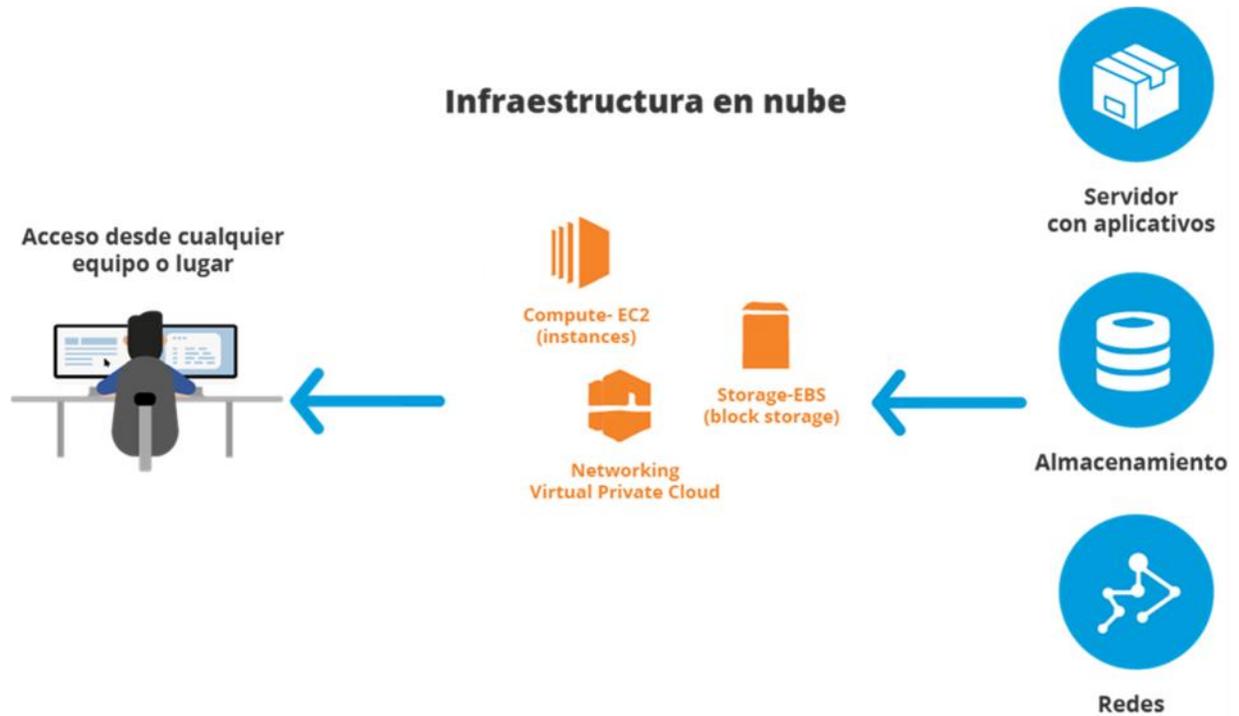


Figura 3.23: Inicio de Sesión

Administrador

Ingrese sus datos de Acceso

Usuario

Password

Ingresar

[Olvidé mi password](#)

Figura 3.24: Pantalla principal

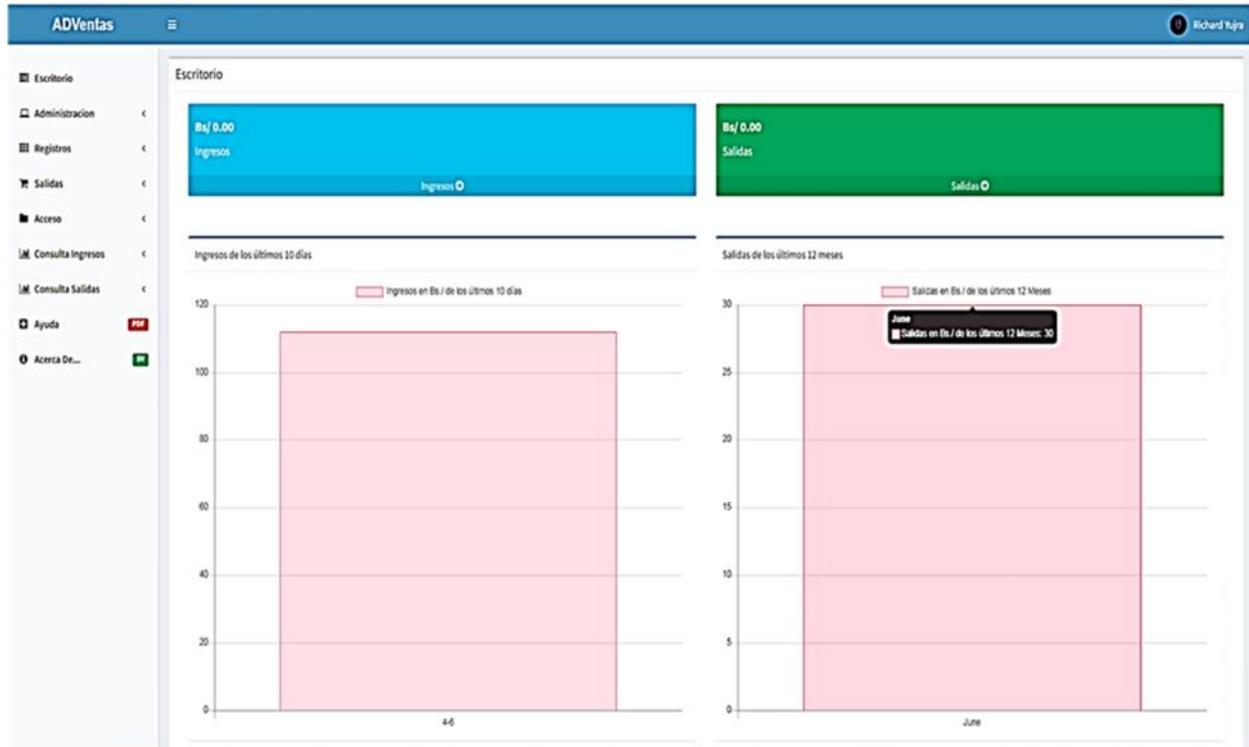


Figura 3.25: Pantalla de Ingreso de Alimentación

The screenshot shows the 'Alimentación' (Feeding) form in the ADVentas system. The form includes fields for 'Nombre(\*)' (Name), 'Descripción' (Description), and 'Código' (Code). A 'Tipo de Dieta(\*)' (Diet Type) dropdown menu is set to 'DIETA NORMAL'. An 'Imagen' (Image) field is present with a 'SELECCIONAR ARCHIVO' button and the text 'Ninguno archivo selec.'. At the bottom, there are buttons for 'Generar', 'Imprimir', 'Guardar', and 'Cancelar'. The footer contains the copyright notice 'Copyright © 2022-2025 Servidim4. Todos los derechos reservados.' and the version 'Versión 1.0.1'.

Figura 3.26: Características de la suscripción

ADVentas Richard Fuja

Alimentación **Reporte**

Nombre(\*): Sopa de pollo

Tipo de Dieta(\*): DIETA NORMAL

Descripción: sopa de pollo hervido con verduras

Imagen:  Ninguno archivo selec.

Código: 01

Copyright © 2022-2025 Servilainut. Todos los derechos reservados. Version 1.0.1

Figura 3.27: Búsqueda de Clientes

ADVentas Richard Fuja

Alimentación **Agregar** **Reporte**

Opciones	Nombre	Tipo de Dieta	Código	Imagen	Estado
<input type="button" value="✖"/>	Sopa de pollo	DIETA NORMAL	01	9	

Mostrando 1 a 1 de 1 registros (filtrados de 2 registros) Anterior 1 Siguiente

Copyright © 2022-2025 Servilainut. Todos los derechos reservados. Version 1.0.1

Figura 3.28: Pantalla de Categoría de Dietas

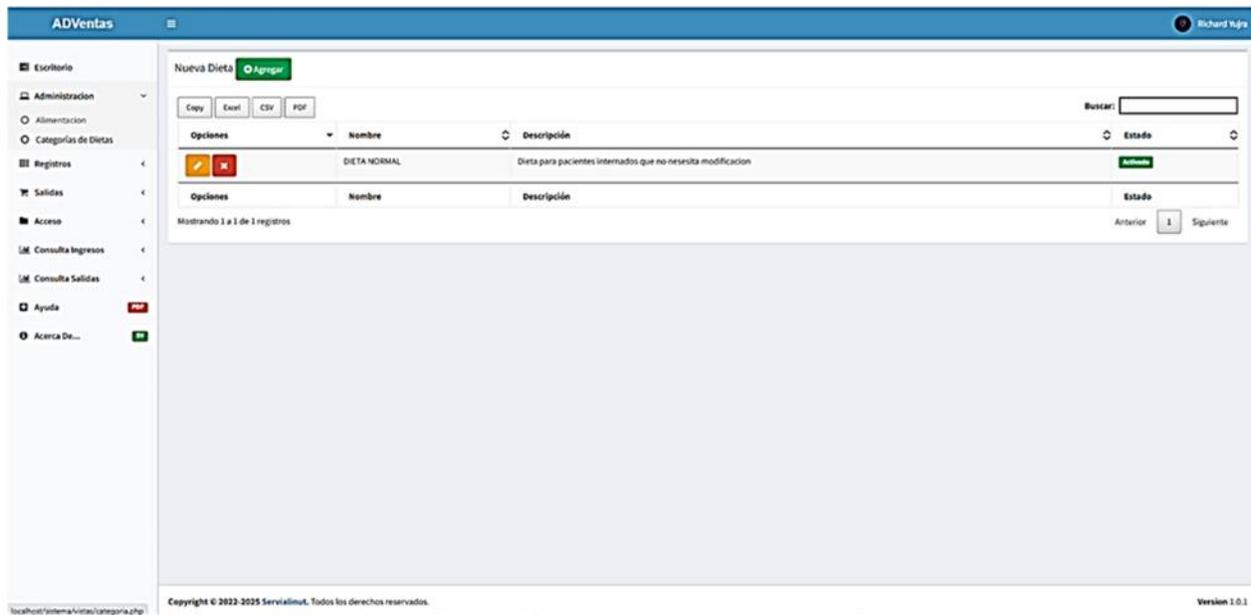


Figura 3.29: Pantalla Nueva Dieta

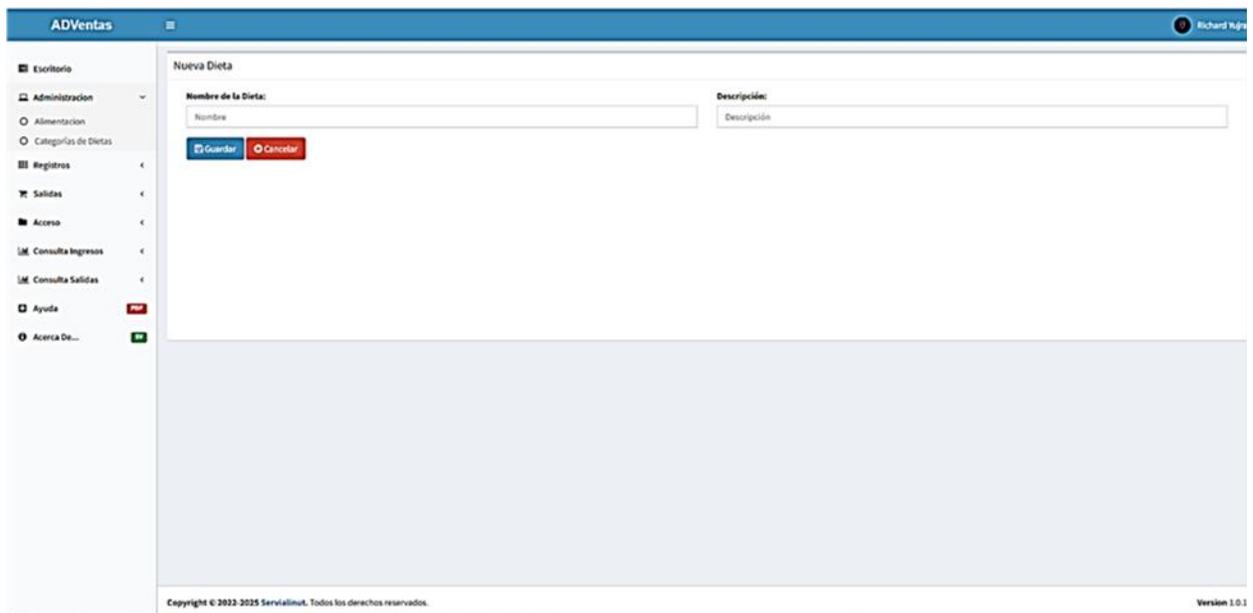


Figura 3.30: Pantalla Editar Dieta

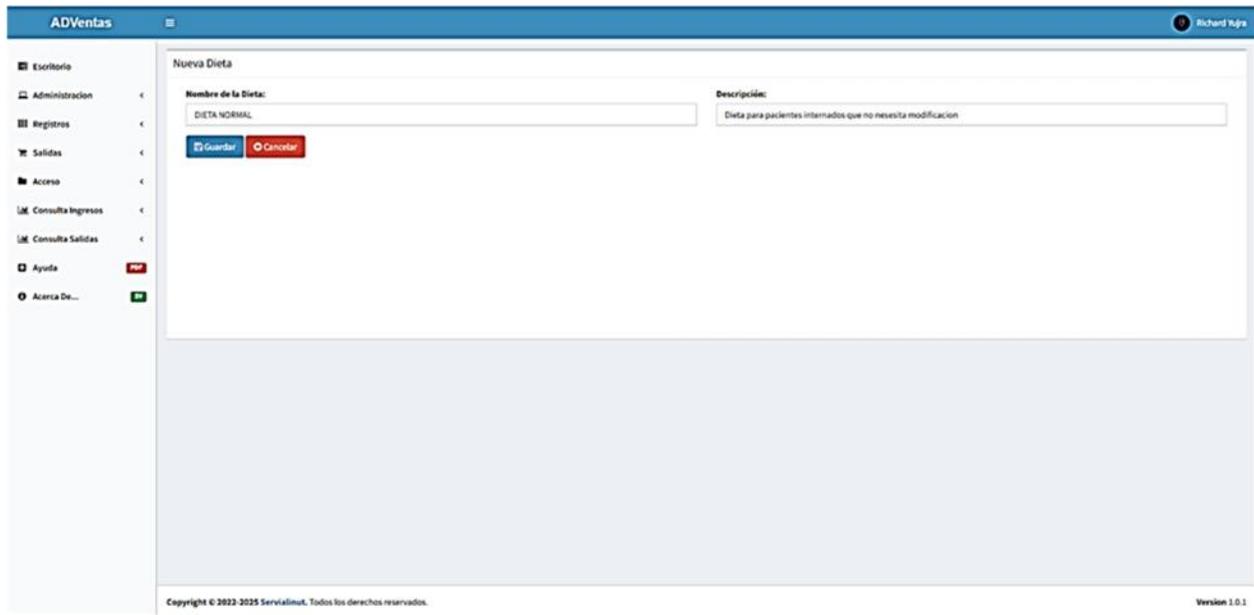


Figura 3.31: Búsqueda de Categoría

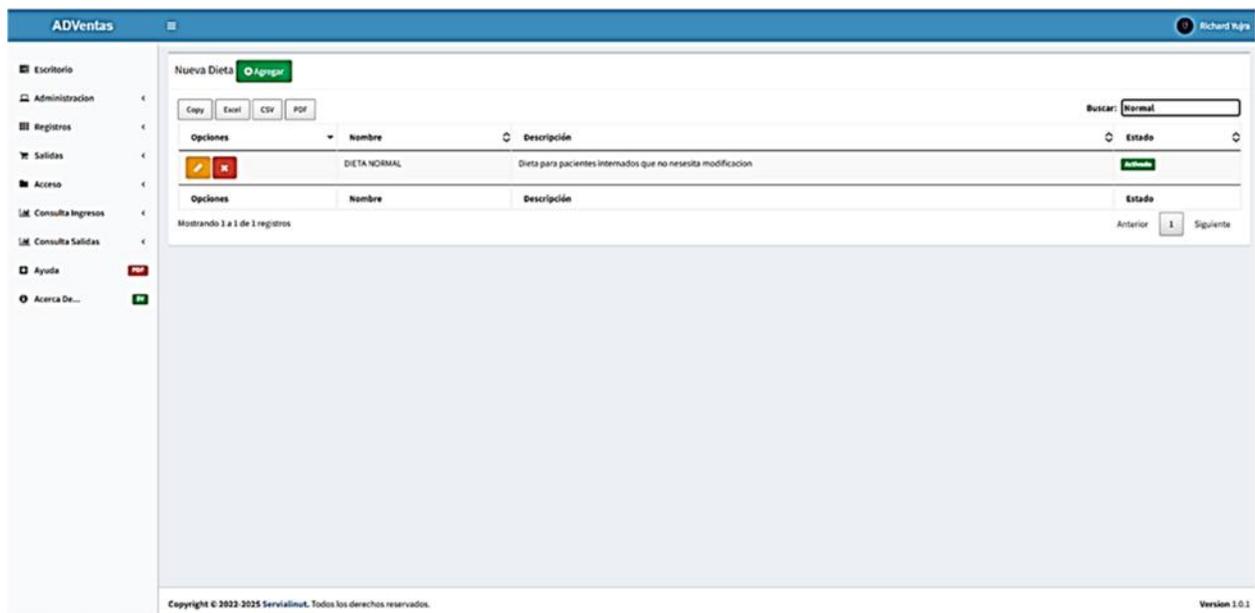


Figura 3.32: Nuevo Encargado

The screenshot shows the 'Nuevo Encargado' (New Assignee) form in the ADVentas system. The form is located in the main content area, with a sidebar on the left containing navigation options like 'Escritorio', 'Administración', 'Registros', 'Salidas', 'Acceso', 'Consulta Ingresos', 'Consulta Salidas', 'Ayuda', and 'Acerca De...'. The form itself has a title 'Encargado' and contains several input fields: 'Nombre' (with a sub-field 'Nombre del proveedor'), 'Número Documento' (with a sub-field 'Documento'), 'Teléfono', 'Tipo Documento' (a dropdown menu), 'Dirección', and 'Email'. Below the input fields are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel). At the bottom of the page, there is a copyright notice 'Copyright © 2022-2025 Servialnut. Todos los derechos reservados.' and the version number 'Version 1.0.1'.

Figura 3.33: Pantalla Editar Encargado

The screenshot shows the 'Nueva Dieta' (New Diet) screen in the ADVentas system. The screen has a title 'Nueva Dieta' with a green 'Agregar' (Add) button. Below the title are buttons for 'Copy', 'Excel', 'CSV', and 'PDF'. There is a search bar with the text 'Buscar: Normal'. The main content is a table with columns for 'Opciones', 'Nombre', 'Descripción', and 'Estado'. The table contains one row with the following data: 'DIETA NORMAL', 'Dieta para pacientes internados que no necesita modificación', and 'Activa'. Below the table, there is a footer that says 'Mostrando 1 a 1 de 1 registros' and navigation buttons for 'Anterior' and 'Siguiente'. At the bottom of the page, there is a copyright notice 'Copyright © 2022-2025 Servialnut. Todos los derechos reservados.' and the version number 'Version 1.0.1'.

Opciones	Nombre	Descripción	Estado
	DIETA NORMAL	Dieta para pacientes internados que no necesita modificación	Activa

Figura 3.34: Búsqueda de Encargado

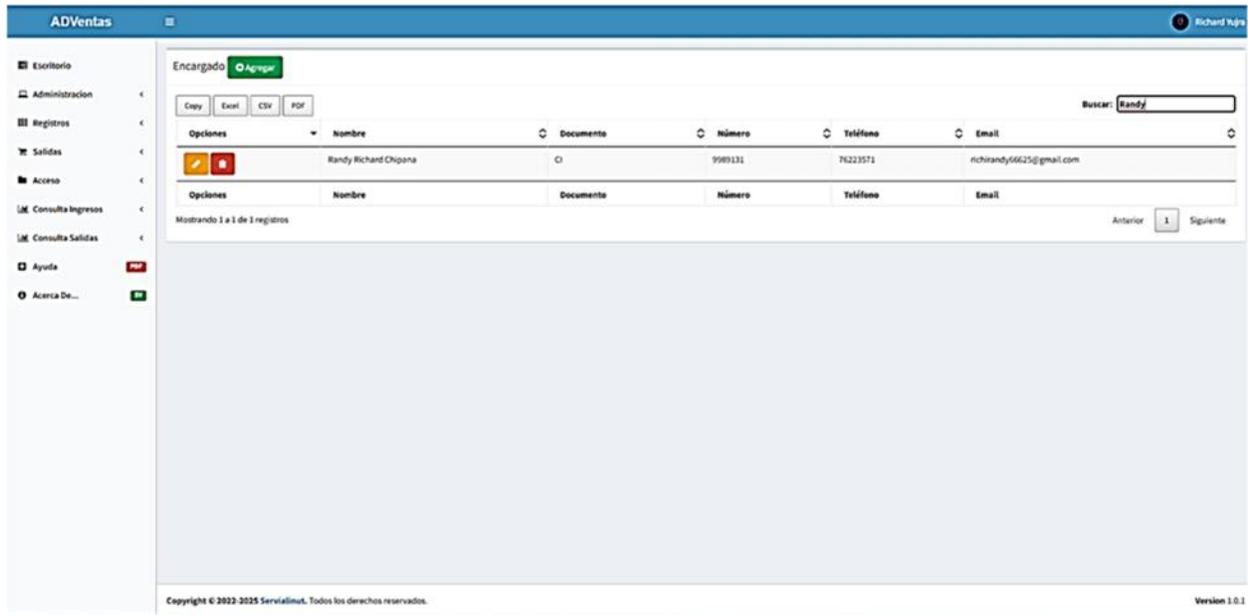


Figura 3.35: Nuevo Ingreso

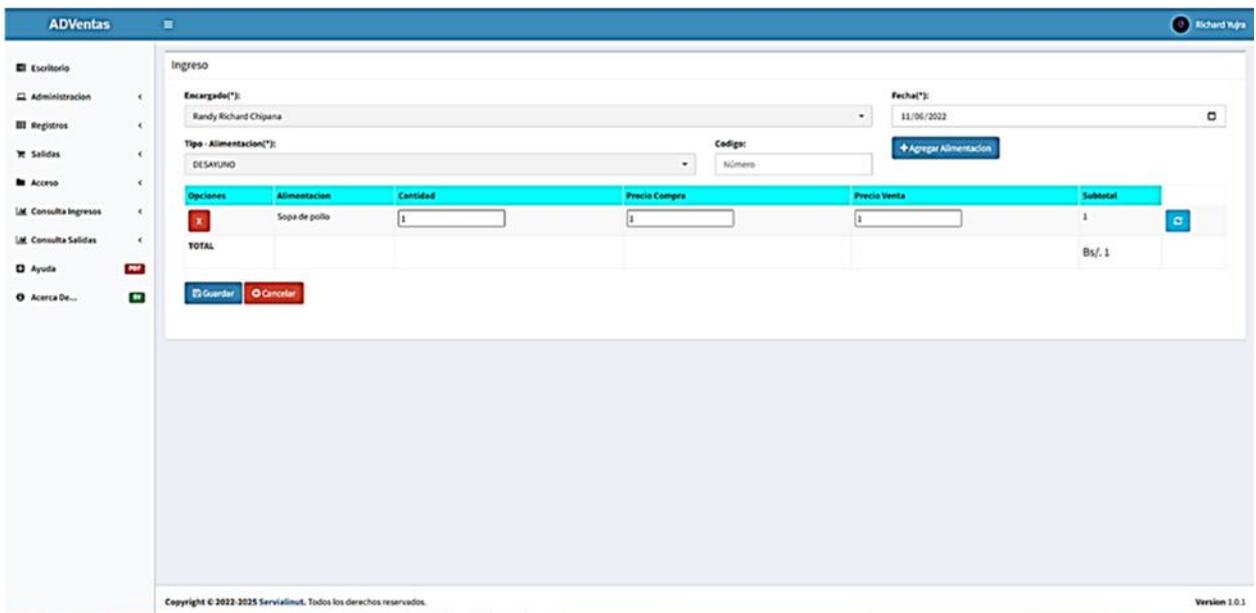


Figura 3.36: Pantalla detalle Ingreso

ADVentas Richard Nuja

**Ingreso**

Encargado(\*): Randy Richard Chipana Fecha(\*): 04/06/2022

Tipo Alimentación(\*): DESAYUNO Codigo: 02

Opciones	Artículo	Cantidad	Precio Compra	Precio Venta	Subtotal
	Jugo de papaya	12	5.00	9.00	60
<b>TOTAL</b>					<b>5/.60</b>

Cancelar

Copyright © 2022-2025 Servialnut. Todos los derechos reservados. Version 1.0.1

Figura 3.37: Búsqueda de Encargado

ADVentas Richard Nuja

**Ingreso** Agregar

Copy Excl CSV PDF Buscar: 001

Opciones	Fecha	Encargado	Usuario	Tipo Alimentación	Codigo	Total Compra	Estado
<span style="color: orange;">+</span> <span style="color: red;">-</span>	2022-06-04	Randy Richard Chipana	Richard Nuja	ALMUERZO	001	1.00	Pagado

Mostrando 1 a 1 de 1 registros (filtrados de 4 registros) Anterior 1 Siguiente

Copyright © 2022-2025 Servialnut. Todos los derechos reservados. Version 1.0.1

Figura 3.38: Nuevo Cliente

ADVentas Richard Rojas

Escritorio  
Administracion  
Registros  
Salidas  
Acceso  
Consulta Ingresos  
Consulta Salidas  
Ayuda  
Acerca De...

Cliente

Nombre: Nombre del cliente

Turno: Mañana

Codigo de Atencion: Codigo de Atencion

Area designada: Area designada

Telefono: Telefono

Guardar Cancelar

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

Figura 3.39: Pantalla editar Cliente

ADVentas Richard Rojas

Escritorio  
Administracion  
Registros  
Salidas  
Acceso  
Consulta Ingresos  
Consulta Salidas  
Ayuda  
Acerca De...

Cliente

Nombre: Personal de Seguridad

Turno: Mañana

Codigo de Atencion: 010

Area designada: Puerta Principal

Telefono: 2861236

Guardar Cancelar

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

Figura 3.40: Búsqueda de Encargado

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

Figura 3.41: Nuevo Venta

Copyright © 2022-2025 Servialimut. Todos los derechos reservados. Version 1.0.1

Figura 3.42: Pantalla detalle Venta

ADventas

Richard Fuja

Escritorio

Administración

Registros

Salidas

Acceso

Consulta Ingresos

Consulta Salidas

Ayuda

Acercía De...

Venta

Cliente(\*)  
Personal de Seguridad

Fecha(\*)  
04/06/2022

Turno(\*)  
Mañana

Codigo  
02

Opciones	Artículo	Cantidad	Precio Venta	Descuento	Subtotal
	Jugo de papaya	2	9.00	0.00	18.00
TOTAL					5,18

Cancelar

Copyright © 2022-2025 Servialimut. Todos los derechos reservados.

Version 1.0.1

Figura 3.43: elegir permisos

ADventas

Richard Fuja

Escritorio

Administración

Registros

Salidas

Acceso

Consulta Ingresos

Consulta Salidas

Ayuda

Acercía De...

Usuario

Nombre(\*)  
Nombre

Tipo Documental(\*)  
CI

Dirección:  
Dirección

Email:  
Email

Login (\*)  
Login

Número(\*)  
Documento

Teléfono:  
Telefono

Cargo:  
Cargo

Clave (\*)  
Clave

Image:  
SELECCIONAR ARCHIVO | ninguno archivo selec.

Permisos:

- Escritorio
- Administración
- Registros
- Salidas
- Acceso
- Consulta Ingresos
- Consulta Salidas

Guardar Cancelar

Copyright © 2022-2025 Servialimut. Todos los derechos reservados.

Version 1.0.1

**Figura 3.44: Permisos del Administrador**

The screenshot shows the 'ADVentas' user management interface. The 'Permisos' section is expanded, showing a list of permissions with checkboxes:

- Escritorio
- Administración
- Registros
- Salidas
- Acceso
- Consulta Ingresos
- Consulta Salidas
- Ayuda
- Acerca De...

Other visible fields include: Nombre (\*): Angela Quecacha; Tipo Documento(\*): CI; Dirección: Zona/San Jose de yunguyo; Email: angelaquecana@gmail.com; Login (\*): Angra123; Número(\*): 90995624; Teléfono: 7723248; Cargo: Vendedor; and a profile picture placeholder.

Tiempo de servicio	Número de peticiones	Fallas encontradas	Probabilidad de fallo	Tiempo medio entre fallos
8 horas	25	1	0.04	8
16 horas	50	2	0.04	8
32 horas	80	3	0.0375	10.7
64 horas	160	5	0.03125	12.8
<b>TOTAL</b>			0.149	39.5

**Tabla 3.5 Fiabilidad**

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
<b>Fiabilidad</b>	96.28 %	Resultado obtenido durante la etapa de las iteraciones	Con el trascurso de tiempo de uso del sistema en las futuras versiones mejoradas se pretende incrementar aún más la fiabilidad

**Tabla 3.6 Análisis de Fiabilidad**

Tiempo de servicio	Número de peticiones	Fallas encontradas	Probabilidad de fallo	Tiempo medio entre fallos
<b>8 horas</b>	25	1	0.04	8
<b>16 horas</b>	50	1	0.02	16
<b>32 horas</b>	80	1	0.0125	32
<b>64 horas</b>	160	2	0.0125	32
<b>TOTAL</b>			0.085	88

**Tabla 3.7 Fiabilidad a la finalización del proyecto**

CARACTERISTICAS	VALOR	VALOR FINALIZACION	OBSERVACION
<b>Fiabilidad</b>	96.28 %	97.87 %	Se puede ver que existe una mejora en la fiabilidad

**Tabla 3.8 Análisis de Fiabilidad a la finalización del proyecto**

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
<b>Fiabilidad</b>	71.4 %	Resultado obtenido durante la etapa de las iteraciones	Con el trascurso de tiempo de uso del sistema en las futuras versiones se pretende mejorar la adecuación funcional de las historias de usuario.

**Tabla 3.9 Análisis de Funcionalidad**

CARACTERISTICAS	VALOR	VALOR FINALIZACION	OBSERVACION
<b>Fiabilidad</b>	71.4 %	85.7 %	Se puede ver que existe una mejora en la funcionabilidad

**Tabla 3.9 Mejora de la Funcionalidad a la finalización del proyecto**

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
<b>Fiabilidad</b>	95.2 %	Resultado obtenido durante la etapa de las iteraciones	Con el transcurso de tiempo de uso del sistema en las futuras versiones se pretende que se tienda a incrementar el valor de la consistencia operacional
<b>Consistencia operacional en el uso</b>	86.7 %	Resultado obtenido durante la etapa de iteraciones	Se pretende incrementar con el transcurso del tiempo y uso del sistema en las futuras versiones la satisfacción del usuario en la consistencia

**Tabla 3.10 Análisis de los datos de la fiabilidad**

Medida	Valor
Fiabilidad	97.87 %
Funcionabilidad	85.70 %
Usabilidad	93.30 %
Portabilidad	100 %
<b>PROMEDIO</b>	<b>94.22 %</b>

**Tabla 3.10 Resumen metricas de calidad**

**Tabla 4.13 costo del proyecto**

COSTO DE DESARROLLO	7500
COSTO DE CAPACITACION	250
COSTO DE SOTFWARE	0
<b>TOTAL</b>	<b>7750</b>