

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**“SISTEMA WEB PARA EL CONTROL Y SEGUIMIENTO DE  
INVENTARIO  
CASO: FARMACIA ALPASO”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POSTULANTE: MARIA LUISA ROJAS FEBRERO  
TUTOR METODOLOGICO: Ph. D. JAVIER HUGO REYES PACHECO  
ASESOR: Ph. D. YOHONI CUENCA SARZURI**

**LA PAZ – BOLIVIA  
2017**



UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## **DEDICATORIA**

A Dios por bendecirme cada día, más de lo que me merezco siendo la luz de mi camino.

A mis padres Gregorio Rojas y Catalina Febrero por su amor, cariño, sacrificio y apoyo incondicional en toda mi carrera.

A mis hermanos por estar siempre presentes en cada momento que los necesite.

A mis amigos que estuvieron ahí para brindarme su apoyo incondicional en todo momento.

## **AGRADECIMIENTOS**

A la Universidad Mayor de San Andrés por haberme acogido en sus aulas de estudio, a la Facultad de Ciencias Puras y Naturales y especialmente a la carrera de Informática por brindarme una educación de excelencia que impactó mi vida en el ámbito profesional.

Agradecer al Ph. D. Javier Hugo Reyes Pacheco, mi tutor metodológico, por su experiencia profesional, quien me brindo su tiempo, su dedicación, su colaboración en cada momento de consulta en el desarrollo del presente trabajo.

También expresar un especial agradecimiento y respeto al Ph. D. Yohoni Cuenca Sarzuri, mi asesor, por su colaboración, paciencia, el tiempo brindado en la revisión y corrección del presente trabajo y por brindarme su ayuda en el planteamiento de la estructura de este Proyecto.

## **RESUMEN**

Con la evolución de las nuevas tecnologías y siendo utilizado en la optimización de varias tareas que puedan mejorar el manejo de información y control de inventarios, se puede dar razones para tener un sistema web para obtener con mayor facilidad y rapidez los datos requeridos.

Se pudo observar que estas nuevos avances tecnológicos están ausentes en la FARMACIA ALPASO, y se da la iniciativa de desarrollar un sistema que mejore y optimice los requerimientos, con el fin de tener un espacio que permite el control eficiente de productos y aumente la difusión de información comercial. Además se cuenta con un ambiente de consulta generalizada que apoye al personal e informe a los clientes.

Para el diseño y desarrollo del sistema Web se utiliza la metodología OOHDMD, esta metodología ofrece el diseño de aplicaciones hipermédia y Web, para su implementación es necesario aplicar cinco fases, que van desde el análisis de requerimientos hasta su implementación.

Para la implementación se hace uso del lenguaje de programación PHP, con un gestor de base de datos MySQL. Además se utiliza el framework Bootstrap front-end para que la interfaz del sistema sea adaptable a dispositivos (responsive).

Para el análisis de costo se utiliza el método COCOMO II, es la evolución mejorada de COCOMO, permitiendo calcular el costo total de desarrollo del sistema. Más adelante, se llevan a cabo las pruebas de caja blanca y caja negra, el cual muestran que los elementos del sistema funcionen de manera adecuada, para el análisis de calidad del sistema web se utiliza el método Web-Site QEM que analizará la usabilidad, funcionalidad, confiabilidad y eficiencia, con el resultado final se muestra el nivel de aceptabilidad del sistema web.

**Palabra claves:** Sistema Web, framework, diseño, desarrollo

## **ABSTRACT**

With the evolution of new technologies and being used in the optimization of several tasks that can improve the handling of information and control of inventories, can give reasons to have a web system to obtain more easily and quickly the required data.

It was observed that these new technological advances are absent in the FARMACIA ALPASO, and the initiative is given to develop a system that improves and optimizes the requirements, in order to have a space that allows the efficient control of products and increases the diffusion of Commercial information. In addition, there is a generalized consultation environment that supports staff and informs clients.

For the design and development of the Web system the OOHDM methodology is used, this methodology offers the design of hypermedia and Web applications, for its implementation it is necessary to apply five phases, ranging from the analysis of requirements to its implementation.

For the implementation, the PHP programming language is used, with a MySQL database manager. In addition, the Bootstrap frond-end framework is used so that the system interface can be adapted to devices (responsive).

For the cost analysis the COCOMO II method is used, it is the improved evolution of COCOMO, allowing to calculate the total cost of system development. Later, the white box and black box tests are carried out, which show that the elements of the system work properly, for the quality analysis of the web system the Web-Site QEM method is used to analyze the usability, functionality, reliability and efficiency, with the final result the level of acceptability of the web system is shown.

**Keyword:** Web system, framework, design, development

# **INDICE GENERAL**

**Pág.**

## **CAPITULO I**

### **MARCO INTRODUCTORIO**

<b>1.1.</b> Introducción.....	1
<b>1.2.</b> Antecedentes.....	2
<b>1.2.1.</b> Antecedentes Institucionales.....	2
<b>1.2.2.</b> Antecedentes De Proyectos Similares.....	3
<b>1.3.</b> Planteamiento Del Problema.....	4
<b>1.3.1.</b> Problema Central.....	4
<b>1.3.2.</b> Problemas Específicos.....	4
<b>1.4.</b> Definición De Objetivos.....	5
<b>1.4.1.</b> Objetivo General.....	5
<b>1.4.2.</b> Objetivos Específicos.....	5
<b>1.5.</b> Justificación.....	5
<b>1.5.1.</b> Justificación Económica.....	5
<b>1.5.2.</b> Justificación Social.....	6
<b>1.5.3.</b> Justificación Tecnológica.....	6
<b>1.6.</b> Alcances y Límites.....	6
<b>1.6.1.</b> Alcances.....	6
<b>1.6.2.</b> Límites.....	6
<b>1.7.</b> Aportes.....	7
<b>1.8.</b> Metodología y herramientas de implementación.....	7
<b>1.8.1.</b> Metodologías.....	7
<b>1.8.2.</b> Herramienta de diseño.....	7

## CAPITULO II

### MARCO TEORICO

<b>2.1.</b> Marco institucional.....	9
<b>2.1.1.</b> Organigrama Farmacia “Al paso”.....	10
<b>2.2.</b> Ingeniería de Software.....	11
<b>2.2.1.</b> Metodologías de desarrollo.....	11
<b>2.2.2.</b> Manifiesto ágil o metodologías agiles.....	12
<b>2.2.3.</b> Metodología SCRUM.....	13
<b>2.2.4.</b> Principales elementos de la metodología Scrum.....	15
<b>2.2.4.1.</b> Herramientas y practicas.....	16
<b>2.2.4.2.</b> Procesos.....	18
<b>2.2.4.3.</b> Roles y responsabilidades.....	20
<b>2.3.</b> Metodología de diseño.....	22
<b>2.3.1.</b> Metodología de diseño OOHDMD.....	23
<b>2.3.1.1.</b> Obtención de requerimientos.....	24
<b>2.3.1.2.</b> Diseño conceptual.....	25
<b>2.3.1.3.</b> Diseño navegacional.....	26
<b>2.3.1.4.</b> Diseño de interfaces abstractas.....	28
<b>2.3.1.5.</b> Implementación.....	29
<b>2.4.</b> Calidad de software.....	30
<b>2.4.1.</b> Modelo de calidad ISO 9126.....	31
<b>2.4.2.</b> Características de la norma ISO 9126.....	31
<b>2.4.2.1.</b> Funcionalidad.....	31
<b>2.4.2.2.</b> Confiabilidad.....	32
<b>2.4.2.3.</b> Usabilidad.....	33
<b>2.4.2.4.</b> Eficiencia.....	33
<b>2.4.2.5.</b> Mantenibilidad.....	34
<b>2.4.2.6.</b> Portatilidad.....	35
<b>2.5.</b> Modelo para el análisis de costos.....	35

<b>2.5.1.</b> Cocomo II.....	35
<b>2.5.1.1.</b> Modelo de diseño temprano.....	37
<b>2.5.1.2.</b> Puntos de función no ajustados.....	37
<b>2.5.1.3.</b> Puntos de función ajustados.....	38
<b>2.5.1.4.</b> Estimación del esfuerzo.....	40
<b>2.5.1.5.</b> Estimación del cronograma.....	44
<b>2.5.1.6.</b> Costo total del sistema.....	45
<b>2.6.</b> Metodología de evaluación de calidad de sitios web WEB-SITEQEM.....	45
<b>2.6.1.</b> Fase de definición y especificación de los requerimientos de calidad.....	45
<b>2.6.2.</b> Fase diseño e implementación de la evaluación elemental.....	46
<b>2.6.3.</b> Fase de diseño e implantación de la evaluación global.....	49
<b>2.6.4.</b> Fase de análisis de resultados y recomendaciones.....	50

## **CAPITULO III**

### **MARCO APlicativo**

<b>3.1.</b> Introducción.....	52
<b>3.2.</b> Proceso de desarrollo.....	52
<b>3.2.1.</b> Obtención de requerimientos .....	52
<b>3.2.1.1.</b> Identificación de roles y tareas.....	52
<b>3.2.2.2.</b> Especificación de escenarios.....	54
<b>3.2.2.3.</b> Especificación de casos de uso.....	57
<b>3.2.2.4.</b> Especificaciones de UIDs.....	62
<b>3.3.</b> Pre – game.....	66
<b>3.3.1.</b> Planificación.....	66
<b>3.4.</b> Game .....	67
<b>3.4.1.</b> Diseño conceptual.....	68
<b>3.4.2.</b> Diseño navegacional.....	70
<b>3.4.3.</b> Diseño abstracto de la interfaz .....	71
<b>3.4.4.</b> Implementación.....	74

## **CAPITULO IV**

### **ANALISIS DE COSTOS, CALIDAD DE SOFTWARE**

<b>4.1.</b> Introducción.....	80
<b>4.2.</b> Análisis de costos.....	80
<b>4.2.1.</b> Calculo de Punto Función no ajustado.....	80
<b>4.2.2.</b> Calculo de los puntos función ajustados.....	81
<b>4.2.3.</b> Estimación del esfuerzo.....	84
<b>4.2.4.</b> Estimación del cronograma.....	88
<b>4.2.5.</b> Costo total del sistema.....	89
<b>4.3.</b> Calidad de software.....	90
<b>4.3.1.</b> Evaluación elemental.....	90
<b>4.3.2.</b> Evaluación global.....	95
<b>4.3.3.</b> Análisis y Resultados.....	100

## **CAPITULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

<b>5.1.</b> Conclusiones.....	101
<b>5.2.</b> Recomendaciones.....	102

## **BIBLIOGRAFIA**

# ÍNDICE DE FIGURAS

<b>Figura 2.1</b> Organigrama de la Farmacia “Alpaso”.....	10
<b>Figura 2.2</b> Ciclo de vida de Scrum.....	14
<b>Figura 2.3</b> Fases de Scrum.....	18
<b>Figura 2.4</b> Roles y responsabilidad en Scrum.....	22
<b>Figura 2.5.</b> Fases del método OOHDM.....	24
<b>Figura 2.6.</b> Ejemplo del diseño conceptual.....	26
<b>Figura 2.7</b> Ejemplo de diseño navegacional.....	28
<b>Figura 2.8</b> Ejemplo de diseño de interfaz abstracta.....	29
<b>Figura 2.9</b> Ejemplo de implementación de OOHDM.....	30
<b>Figura 2.10</b> factores que se toman para el análisis de costo.....	36
<b>Figura 3.1</b> Escenario de identificación de usuario.....	54
<b>Figura 3.2</b> Escenario de Registro de Regente.....	54
<b>Figura 3.3</b> Escenario de Registro de clientes.....	55
<b>Figura 3.4</b> Escenario de Registro de laboratorio.....	55
<b>Figura 3.5</b> Escenario de Registro de productos.....	56
<b>Figura 3.6</b> Escenario de Publicación de productos.....	56
<b>Figura 3.7</b> Escenario de administración del Administrador.....	56
<b>Figura 3.8</b> Escenario de administración del Regente.....	57
<b>Figura 3.9</b> Escenario de administración del Cliente.....	57
<b>Figura 3.10</b> Diagrama de casos de uso general del sistema.....	58
<b>Figura 3.11</b> Diagrama de casos de uso Registro de Regente.....	59
<b>Figura 3.12</b> Diagrama de casos de uso Registro de Cliente.....	59
<b>Figura 3.13</b> Diagrama de casos de uso Registro de Cliente.....	60
<b>Figura 3.14</b> Diagrama de casos de uso procesos.....	61
<b>Figura 3.15</b> Diagrama de casos de uso Reportes.....	61
<b>Figura 3.16</b> Diagrama de casos de uso Administración de usuarios.....	62
<b>Figura 3.17</b> UID Autentificación de usuarios.....	63

<b>Figura 3.18</b> UID Registro de Usuario Regente.....	63
<b>Figura 3.19</b> UID Registro de Usuario Cliente.....	64
<b>Figura 3.20</b> UID Registro de Laboratorio y productos.....	64
<b>Figura 3.21</b> UID procesos de administrador y regente.....	65
<b>Figura 3.22</b> UID Reportes de administrador y regente.....	65
<b>Figura 3.23</b> Diagrama Entidad – Relación.....	68
<b>Figura 3.24</b> Diagrama de clases del sistema.....	69
<b>Figura 3.25</b> Diseño navegacional.....	70
<b>Figura 3.26</b> Diseño página principal del sistema.....	71
<b>Figura 3.27</b> Acceso de usuario.....	72
<b>Figura 3.28</b> Registro de usuarios/clientes.....	73
<b>Figura 3.29</b> Interfaz de usuarios/clients.....	73
<b>Figura 3.30</b> Interfaz registro de productos .....	74
<b>Figura 3.31</b> Interfaz de usuario del sitio web.....	75
<b>Figura 3.32</b> Página de inicio del sistema.....	76
<b>Figura 3.33</b> Interfaz de usuario Registro de cliente.....	77
<b>Figura 3.34</b> Interfaz de usuario cuenta administrador .....	77
<b>Figura 3.35</b> Interfaz de usuario cuenta administrador .....	78
<b>Figura 3.36</b> Interfaz de usuario cuenta cliente.....	78
<b>Figura 3.37</b> formulario de agregación de productos.....	79

# ÍNDICE DE TABLAS

<b>Tabla 2.1</b> puntos función sin ajustar.....	37
<b>Tabla 2.2</b> Factores de complejidad.....	38
<b>Tabla 2.3</b> líneas de código por cada punto función.....	39
<b>Tabla 2.3</b> Factores de escala.....	41
<b>Tabla 2.4</b> Valores de los factores de escala.....	42
<b>Tabla 2.5</b> Factores de ajuste.....	43
<b>Tabla 2.6</b> Valores de los factores de ajuste obtenidos.....	44
<b>Tabla 3.1</b> Requerimientos.....	66
<b>Tabla 3.2</b> Requerimientos de hardware y software.....	67
<b>Tabla 3.3</b> Sprint.....	67
<b>Tabla 4.1</b> Puntos función sin ajustar.....	80
<b>Tabla 4.2</b> Factores de complejidad.....	81
<b>Tabla 4.3</b> Líneas de código por cada punto función.....	83
<b>Tabla 4.4</b> Valores de los factores de escala.....	85
<b>Tabla 4.5</b> Valores de los factores de escala.....	85
<b>Tabla 4.6</b> Factores de ajuste.....	86
<b>Tabla 4.7</b> Valores de Factores de ajuste.....	87
<b>Tabla 4.8</b> Resultados de preferencias elementales de usabilidad .....	90
<b>Tabla 4.9</b> Resultados de preferencias elementales de funcionalidad.....	92
<b>Tabla 4.10</b> Resultados de preferencias elementales de confiabilidad.....	94
<b>Tabla 4.11</b> Resultados de preferencias elementales de eficiencia.....	95
<b>Tabla 4.12</b> Resultados global de Usabilidad.....	96
<b>Tabla 4.13</b> Resultados global de Funcionalidad.....	97
<b>Tabla 4.14</b> Resultados global de Confiabilidad.....	99
<b>Tabla 4.15</b> Resultados global de Eficiencia.....	99
<b>Tabla 4.16</b> Resultados global .....	100

## CAPITULO I

### MARCO INTRODUCTORIO

#### 1.9. INTRODUCCION

Actualmente se observa un gran avance tecnológico y a la vez muchas innovaciones informáticas, por lo cual la necesidad de automatizar las actividades diarias se hace cada vez más requerida buscando implementar mejores soluciones para el área de informática y de computación. Esto representa una gran ventaja para el usuario, dado que puede acceder a distintas áreas de su interés desde su equipo y/o celular.

De la misma forma con el paso del tiempo fueron surgiendo nuevas necesidades abarcando actividades distintas a la informática, como ser un sistema de control de compra y venta de un supermercado, un control de planilla de personal en fin el propósito sería suplir actividades realizadas por el usuario, por aplicaciones informáticas que realizan las actividades con eficiencia.

También podemos mencionar que con forma vaya avanzando la tecnología hoy en día podemos almacenar gran cantidad de datos en la web, se podría decir que anteriormente se realizaban base de datos con un tamaño determinado para el almacenamiento de sus datos que obligaban a la depuración de los datos más antiguos, hoy en día gracias al almacenamiento en la nube podemos guardas gran cantidad de datos.

En el entorno actual, cada vez más competitivo y con menores márgenes, las organizaciones buscan continuamente oportunidades de mejora que las haga más competitivas. En este sentido, cada vez son más conscientes de la importancia de la gestión de almacenes (y la gestión logística en general) como parte esencial a la hora de aportar más valor a sus clientes para un eficaz funcionamiento.

La “FARMACIA ALPASO”, requiere un sistema adecuado para la toma de decisiones, respecto al control de ventas por día, mes y año. Control de stock para realizar nuevos pedidos, y un sistema de consultas donde busque el producto requerido.

Bajo estas circunstancias se desea desarrollar un sistema web que será útil para la mejor administración de la “FARMACIA ALPASO”,

## **1.10. Antecedentes**

### **1.10.1. Antecedentes Institucionales**

La “FARMACIA ALPASO”, tiene un carácter de desarrollo social, para la comunidad que tiene como finalidad mejorar el bienestar de los ciudadanos, a través de una cartera de servicios y de productos, cuyos rasgos diferenciales son la calidad, seguridad y el mejor precio del Mercado.

El 15 de Abril de 2017 nace la “FARMACIA ALPASO”, con la adquisición de la primera sucursal ubicada en la Av.: A1 # 1012 y contaba con una clientela leal que depositaba en ella la confianza del despacho de productos. Esta a su vez ya era reconocida en la zona por aquellos profesionales de la salud quienes confiaban en el servicio que brindaba esta farmacia y la importante labor que la misma desarrollaba en la comunidad de EL ALTO.

La empresa al ser nueva en el área de farmacéutica tiene como propósito crecer y poder convertirse en una cadena de red de farmacias en toda la ciudad y el país.

En este sentido que se identificó la necesidad de establecer un Sistema Web Para El Control Y Seguimiento De Inventario, que regularice y permita el eficaz control de registro y distribución del los insumos farmacéuticos, además de proporcionar un registro de compra y venta diaria par poder estar pendientes de las fechas de caducidad de cada insumo farmacéutico.

### **1.10.2. Antecedentes De Proyectos Similares**

#### Trabajos Similares

- “**Sistema de Control de Farmacia para un centro de salud de nivel I-3**”, Arroyo Olivera, José Carlos, 2013, Universidad Peruana de Ciencias Aplicadas, Facultad De Ingeniería, Carrera De Ingeniería De Software.

Resumen: La Memoria del Proyecto recoge toda la información relevante acerca del desarrollo del Sistema de Control de Farmacia (SISCOFARMA). En el primer capítulo del presente documento se explica el contexto del proyecto SISCOFARMA. Esto incluye información acerca del Ministerio de Salud y de la categorización de los centros de salud del MINSA para luego centrarse en el nivel I-3 de complejidad, (Arroyo, 2013).

- “**Sistema informático de inventario y facturación de mercadería con entorno web en la imprenta y gráficas chone**”, Ing. Geomara Juliana Lucas Chávez, 2013, Escuela Superior Politécnica Agropecuaria De Manabí Manuel Félix López.

Resumen: El objetivo primordial de esta investigación fue crear un sistema informático web con la finalidad de mejorar el control de inventario y facturación de mercadería en la Imprenta y Gráficas Chone, con el firme propósito de agilizar los procesos y así facilitar las tareas en dicha institución. El modelo de software que se empleó para la elaboración de la aplicación fue el modelo UWE (UMLbased Web Engineering/Ingeniería de la Web basada en UML), el cual permitió la recopilación de información para definir los requisitos y la arquitectura del sistema. Se elaboraron los casos de uso y los diagramas de flujos de datos que ayudaron a describir lo que el sistema debe hacer. (Chávez, 2013).

## **1.11. Planteamiento Del Problema**

La base de toda empresa comercial es la compra y venta de bienes o servicios; de aquí la importancia del manejo del inventario por parte de la misma. Este manejo registrable permitirá a la empresa mantener el control oportunamente, así como también conocer al final del período contable un estado confiable de la situación económica de la Farmacia.

“FARMACIA ALPASO”, es una empresa que se dedica a la distribución de productos farmacéuticos, por lo cual es necesario llevar a cabo el control del inventario de los productos y a su vez llevar la contabilidad de la empresa, realizando informes de actividad diarias.

### **1.11.1. Problema Central**

¿Cómo mejorar el proceso de registro de inventarios, supervisión y control de suministros farmacéuticos y también mejorar el control adecuado de registros de movimientos diarios finalidad de optimizar sus procesos de atención al cliente?

### **1.11.2. Problemas Específicos**

Se pudo identificar los siguientes problemas secundarios:

- La variedad de productos farmacéuticos, ocasiona en algunos casos una confusión a la hora de ofrecerlas a clientes.
- El tiempo que se demora en realizar la búsqueda de un producto dentro de almacenes si no se tiene registrado el stock de los suministros
- Si no contara con un registro diario de ventas y compras, no se sabría si hubo ingresos o egresos.

## **1.12. Definición De Objetivos**

### **1.12.1. Objetivo General**

“Implementar un sistema informático con entorno web con la finalidad de mejorar la supervisión y control de registros, y controles diarios de ingresos y egresos de los suministros para la farmacia ALPASO”.

### **1.12.2. Objetivos Específicos**

- Facilitar el ingreso de la información para que los datos se mantengan actualizados y contar con información real de la operación.
- Optimizar la búsqueda de un producto utilizando una base de datos que almacene la información de todos los tipos de medicamentos
- Ofrecer a los clientes una rápida y eficaz atención al momento de hacer sus compras.
- Optimizar los reportes diarios de compra y venta.
- Crear la base de datos que se acople a los requerimientos del sistema.
- Implementar el sistema con entorno web.

## **1.13. Justificación**

### **1.13.1. Justificación Económica**

El desarrollo del sistema experto resulta un beneficio en términos de ahorro para la “FARMACIA ALPASO”, ya que acelerara el tiempo de atención al cliente, y así poder realizar más movimiento económico, también tener un control de compra y venta que permitan saber si hubo ganancias o déficit diariamente.

### **1.13.2. Justificación Social**

El sistema web es un apoyo al administrador de la “FARMACIA ALPASO”, ya que coadyuvará con el registro de suministros y tener un control de todo lo que tenemos en almacenes, y proporcionar una mejor atención a los clientes sin generar demoras ni molestias a los clientes.

### **1.13.3. Justificación Tecnológica**

El proyecto coopera con todas la necesidades que se requiere para mejorar la administración de suministros y la atención al cliente, con respecto a la búsqueda rápida y segura de todos los suministros, y obtener un informe diario de compra y venta, también los pedidos que se requieren, en fin una gran cantidad de tarea se realizaban por escrito y tomaría mucho tiempo.

## **1.14. Alcances y Límites**

### **1.14.1. Alcances**

- El sistema de control podrá generar la búsqueda de un producto y mostrarnos su existencia en los distintos tipos de laboratorios que lo distribuyen.
- El sistema de control podrá realizar informes diarios de entra y salida de suministros.
- El sistema de control podrá notificar los pedidos que se requieren si estos faltaran en almacenes.
- Se realizara un interfaz gráfico amigable, para que el usuario pueda realizar el almacenamiento de datos fácilmente.

### **1.14.2. Límites**

- El sistema no está destinado proporcionar diagnósticos médicos, para la distribución de medicamentos que se venden bajo receta medica

## **1.15. Aportes**

La realización de un sistema web de control de inventarios, mejorara la atención al cliente, un control de registros en almacén con datos actualizados diariamente para una fácil administración de la farmacia ALPASO VISAR, ya que se tiene una manera morosa de registros realizados manualmente que resulta moroso; es por ello que el desarrollo del Sistema web es muy conveniente.

## **1.16. Metodología y herramientas de implementación**

### **1.16.1. Metodologías**

Analizando el mundo en que vivimos y el avance de la nueva tecnología vemos que el desarrollo de software va en crecimiento de forma considerable, donde la formación y los nuevos sistemas de información requieren nuevas metodologías y herramientas de análisis, diseño e implementación de software, planteando soluciones y respuestas inmediatas, para la toma de decisiones en las empresas de nuestra ciudad.

Por lo cual es de suma importa elegir la metodología adecuada, así como las herramientas de diseño e implementación, es por ello que la metodología ágil Scrum nos proporciona todas las bases para llegar al éxito de la elaboración del sistema.

La metodología ágil Scrum nos brinda disciplina, donde se encuentra los artefactos y las guías para la documentación e implementación de manera fácil y eficiente de desarrollo de software, todas estas guías coadyuvan con las respectivas fases con las cuales cuenta esta metodología.

### **1.16.2. Herramienta de diseño**

Para la parte de diseño se utilizara OOHDM Object Oriented Hypermedia Design Method: OOHDM este método propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas:

- **diseño conceptual**, uso de diagramas que permitan expresar con claridad el comportamiento, la estructura y las relaciones entre dichos conceptos.
- **diseño navegacional**, compone de objetos construidos a partir de objetos conceptuales, y constituyen en general los elementos canónicos de las aplicaciones hipermédia tradicionales: nodos, enlaces, anclas y estructuras de acceso.
- **diseño de interfaces abstractas**, se define la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación.
- **Implementación**, en esta fase, se debe implementar el diseño, (ITM, sf).



## CAPITULO II

### MARCO TEORICO

#### 2.1. Marco institucional

FARMACIA “ALPASO” es una microempresa Boliviana que apenas cumplirá su primer año de servicio, dedicándose a la distribución de insumos farmacéuticos, proporcionando asesoramiento al cliente a la hora de adquirir un producto farmacéutico.

La FARMACIA “ALPASO” cuenta con el registro de comercio, órgano concesionado a la fundación para el desarrollo empresarial FUNDAEMPRESA

#### MISIÓN

Mejorar la atención de las necesidades de la sociedad en materia de salud, proporcionando un servicio y productos de calidad, siempre procurando la máxima satisfacción del cliente. Para ello contamos con un personal muy cualificado, capaz de proporcionar la mejor solución para cada problema, siempre con el mejor trato posible hacia el cliente y prescribiendo el producto más adecuado. Una de nuestras mayores preocupaciones es que el cliente quede completamente satisfecho del trato recibido, e intentar mejorar para que día a día podamos ofrecer un servicio eficaz y de calidad.

#### VISIÓN

Convertirnos en la mejor farmacia social comunitaria, formando el mejor equipo humano trabajando con profesionalismo, honestidad y compromiso para así llegar a ser la mejor cadena de farmacias a nivel nacional. Trascender más allá de la tradicionalidad del negocio para proyectarnos y reflejarnos en el desarrollo de nuestra comunidad y se distinga por proporcionar un servicio de calidad y por el elevado grado de satisfacción del cliente.

## **VALORES**

- Trabajo en equipo.
- Excelencia.
- Responsabilidad.
- Actitud positiva.
- Lealtad a nuestra Misión.
- Satisfacer necesidades del cliente.
- Procurar una atención farmacéutica personalizada y de calidad.
- Proporcionar servicios innovadores y de calidad.
- Trabajo en equipo y compromiso.
- Clima agradable tanto entre nuestro personal como con el cliente.

### **2.1.1. ORGANIGRAMA FARMACIA “ALPASO”**

En la figura 2.1 se muestra el organigrama de la FARMACIA “ALPASO”



**Figura 2.1 Organigrama de la FARMACIA “ALPASO”**  
**Fuente [Elaboración propia]**

## **2.2. Ingeniería de Software**

La Ingeniería de Software es aquella disciplina que se ocupa del desarrollo, la operación y el mantenimiento del software, que ha traído consigo propuestas diferentes para mejorar los resultados del proceso de construcción.

Cabe destacarse que es preciso estudiar tanto los principios como las metodologías para llevar a cabo estas acciones mencionadas, en tanto, la disposición de ese conocimiento es lo que permitirá el diseño y la construcción de programas informáticos con los cuales se pueda operar de modo satisfactorio en las diversas computadoras personales (Sommerville, 2005).

Las metodologías tradicionales hacen énfasis en la planeación del proyecto, mientras las metodologías agiles hacen énfasis en la adaptación del proceso, de manera simultánea el tema de modelos que sirven para para el mejoramiento de los procesos de desarrollo de software ocupa un lugar importante en la búsqueda de la metodología adecuada para implementar un software de calidad en cualquier contexto de desarrollo.

Entonces, la ingeniería de software implica un trabajo integral, es decir, se produce un análisis del contexto, se diseña el proyecto, se desarrolla el correspondiente software, se efectúan las pruebas para asegurar su correcto funcionamiento y finalmente se implementa el sistema (DefABC, 2010).

### **2.2.1. Metodologías de desarrollo**

En ingeniería de software cuando se hace referencia al desarrollo de software, se está hablando del desarrollo de programas, los cuales deben cumplir una serie de etapas o fases, para poder funcionar con otros métodos ya establecidos en otras disciplinas de ingeniería.

Su objetivo principal es exponer un conjunto de técnicas clásicas y modernas de modelado de sistemas que hagan posible desarrollar un software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistema (Condef, 2016).

Según (Letelier 2006) existen distintas propuestas de metodología para desarrollar software , tradicionalmente estas metodologías se centran en el control de proceso, estableciendo rigurosamente las actividades, herramientas y notaciones, dado estas reglas las metodologías se caracterizan por ser rígidos y dirigidos por la documentación que se genera en cada uno de los proyectos desarrollados.

Estos métodos no son muy adecuados para muchos de los proyectos actuales donde el entorno de sistema es dinámico y donde es necesario reducir los tiempos de desarrollo. Por tal motivo y necesidad nace el término ágil aplicado al desarrollo de software.

Tras esta reunión se creó The Agile Alliance<sup>3</sup>, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil.

### **2.2.2. Manifiesto ágil o metodologías agiles**

El manifiesto ágil hace énfasis en cuatro valores principales que deben soportar el desarrollo de software:

- *contar con un recurso humano* para garantizar el éxito.
- *Software funcionando por encima de la documentación* es decir los documentos deben ser cortos y limitarse a lo fundamental, dando prioridad al contenido sobre la forma de presentación.
- *la colaboración del cliente* por encima de la negociación del contrato es decir que incluye de manera directa y comprometida al cliente o usuario en el equipo de trabajo.
- *La respuesta al cambio por encima del seguimiento* de un plan: debe ser flexible para poder adaptarse a los cambios que puedan surgir.

Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno (UTP, 2007).

Según (UTO, 2007) , los valores anteriores inspiran doce principios del manifiesto ágil que diferencian un proceso ágil con uno tradicional, entre ellos existen dos que son generales y resumen gran parte del manifiesto ágil. Estos dos principios son:

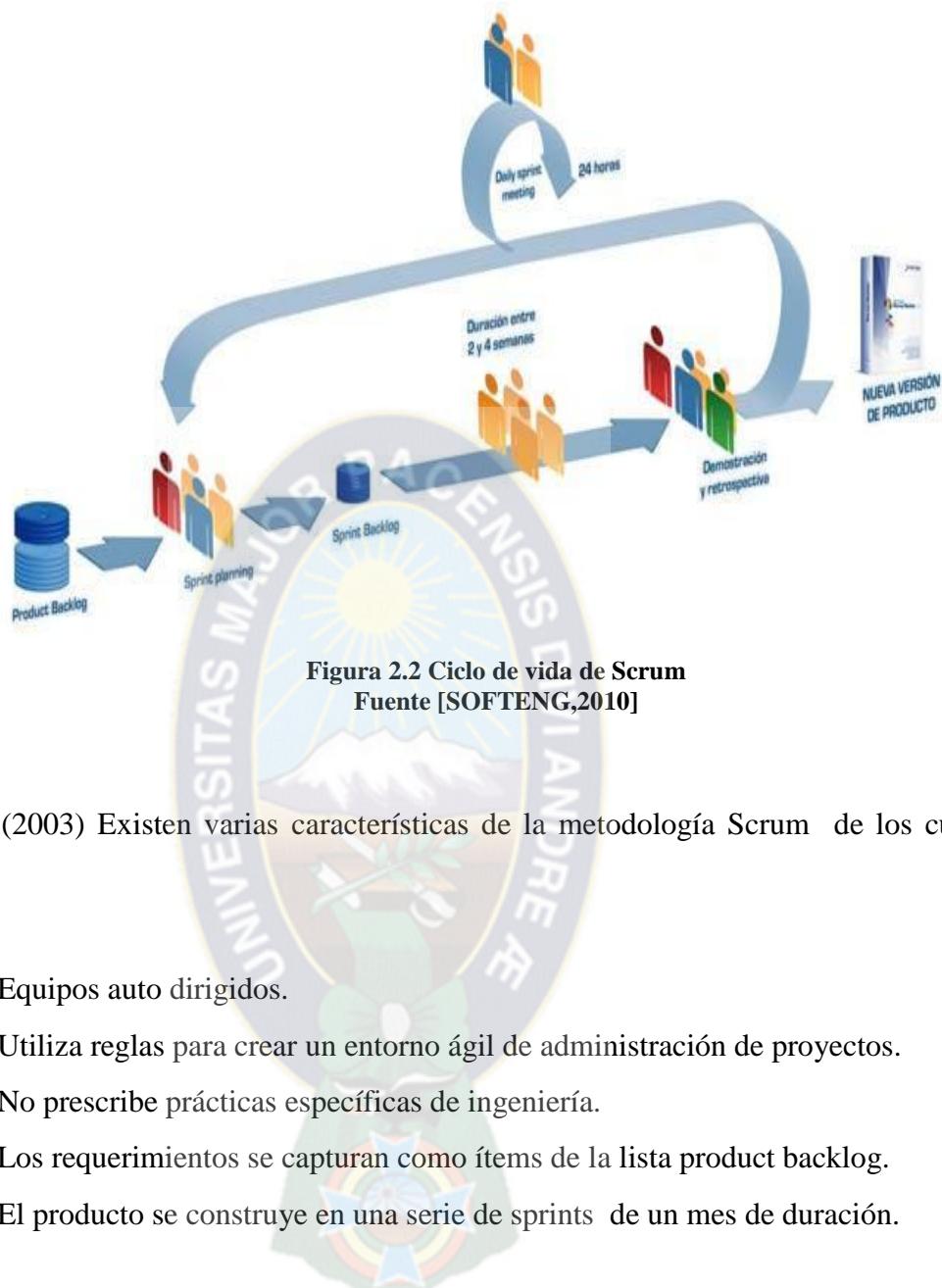
- Los cambios en los requisitos son bienvenidos, incluso tarde en el desarrollo, los procesos agiles aprovechan los cambios como ventajas competitiva del cliente.
- Entregar software que funciona con frecuencia, desde un par de semanas hasta un par de meses, con preferencia de escalas de tiempo cortas con el menor intervalo de tiempo posible.

### **2.2.3. Metodología SCRUM**

Scrum es un framework de desarrollo ágil de software. El trabajo es estructurado en ciclos de trabajo llamados sprints, iteraciones de trabajo con una duración típica de dos a cuatro semanas. Durante cada sprint, los equipos eligen de una lista de requerimientos de cliente priorizados, llamados historias de usuarios, para que las características que sean desarrolladas primero sean las de mayor valor para el cliente. Al final de cada sprint, se entrega un producto potencialmente funcional, distribuible y comercializable.

Scrum es flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa, Se basa en construir primero la funcionalidad de mayor valor para el cliente (TOP, 2011).

El ciclo de vida definido por Scrum es incremental e interactivo y se caracteriza por ser muy adaptable ver Figura 2.2.



Según Peralta (2003) Existen varias características de la metodología Scrum de los cuales son:

- Equipos auto dirigidos.
- Utiliza reglas para crear un entorno ágil de administración de proyectos.
- No prescribe prácticas específicas de ingeniería.
- Los requerimientos se capturan como ítems de la lista product backlog.
- El producto se construye en una serie de sprints de un mes de duración.

Estas características se basan en la teoría de control de procesos empíricos, el empirismo asegura que el conocimiento procede de la experiencia y de la toma de decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control de riesgo. Según (Vila, 2007) Tres pilares soportan toda la implementación del control de procesos empíricos: *Transparencia, Inspección y Adaptación*.

**a) Transparencia:**

Implica dar visibilidad a todo lo que está pasando, ya que los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La Reunión de Planificación proporciona visibilidad al Equipo Scrum acerca de aquello que va a hacer en el sprint; el Scrum Diario, proporciona visibilidad sobre las tareas diarias, los impedimentos y cómo marcha el trabajo; la Revisión del Sprint ofrece visibilidad sobre los logros, resultados y el progreso. Por último, la Retrospectiva del Sprint contribuye con la inspección y la adaptación del proceso.

**b) Inspección:**

Scrum promueve la inspección frecuente de los Artefactos y del progreso para identificar y corregir las variaciones no deseadas. La inspección tiene lugar durante: la Reunión de Planificación de Sprint, el Scrum Diario, la Revisión del Sprint y la Retrospectiva del Sprint.

**c) Adaptación:**

La adaptación implica hacer los ajustes en los procesos y artefactos para minimizar la desviación. Aunque la Retrospectiva del Sprint es el momento que formalmente se reserva para llevar plantar las mejoras a implementar durante el sprint siguiente.

#### **2.2.4. Principales elementos de la metodología Scrum**

Entre sus principales elementos (Peralta, 2003), que utiliza la metodología Scrum se encuentran las siguientes:

- Herramientas y prácticas.
- Proceso.
- Roles y responsabilidades.

#### **2.2.4.1. Herramientas y prácticas**

Scrum no requiere ni provee prácticas de Ingeniería. En lugar de eso, especifica prácticas y herramientas de gerencia que se aplican en sus distintas fases para evitar el caos originado por la complejidad e imposibilidad de realizar predicciones.

##### **a) Product Backlog List (Lista de registro de productos)**

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para un Sprint.

La Product Backlog List puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. Con la restricción de que solo puede cambiarse entre Sprints. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Existe un rol asociado con esta lista y es el de Product Owner. Si alguien quiere realizar cualquier modificación sobre la lista por ejemplo: agregar o incrementar la prioridad de sus elementos tiene que convencer al Product Owner (Peralta, 2003).

##### **b) Sprints**

Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

El objetivo de un Sprint debe ser expresado en pocas palabras para que sea fácil de recordar y esté siempre presente en el equipo. Es posible definir una serie de restricciones que el equipo deba aplicar durante un Sprint.

Un Sprint tiene una duración planificada de entre una semana y un mes. No es posible introducir cambios durante el Sprint, por lo tanto para planificar su duración hay que pensar en cuanto tiempo puedo comprometerme a mantener los cambios fuera del Sprint. Dependiendo del tamaño del sistema, la construcción de un release puede llevar entre 3 y 8 Sprints. Por otra parte podrían formarse equipos para desarrollar en forma paralela distintos grupos de funcionalidad.

Las actividades que se desarrollan Durante el Sprint son: Sprint Planning Meeting, Sprint Backlog, Daily Scrum Meetings y Sprint Review Meeting. En la siguiente gráfica se pueden ver las prácticas involucradas en un Sprint (Peralta, 2003).

#### c) **Sprint Backlog**

Es el punto de entrada de cada Sprint. Es una lista que tiene los ítems de la Product Backlog List que van a ser implementados en el siguiente Sprint.

Los ítems son seleccionados por el Scrum Team, el Scrum Master y el Product Owner en la Sprint Planning Meeting a partir de la priorización de los ítems y los objetivos que se marcaron para ese Sprint. A partir de los objetivos a cumplir durante el Sprint el Scrum Team determina que tareas debe desempeñar para cumplir el objetivo. De esto surge el Sprint Backlog. Es importante destacar que es el equipo quien se organiza para alcanzar el objetivo.

El Manager no asigna tareas a los individuos y tampoco toma decisiones por el equipo. El equipo puede agregar nuevas tareas o remover tareas innecesarias en cualquier momento si lo considera necesario para cumplir el objetivo. Pero el Sprint Backlog solo puede ser modificado por el equipo. Las estimaciones se actualizan cada vez que aparece nueva información (Peralta 2003).

#### d) **Stabilization Sprints**

En estos Sprints el equipo se concentra en encontrar defectos, no en agregar funcionalidad. Suelen aplicarse cuando se prepara un producto para el release. Son

útiles cuando se están realizando pruebas beta, se está introduciendo a un equipo en la metodología de Scrum o cuando la calidad de un producto no alcanza los límites esperados. No fueron definidos por Scrum pero han sido recomendados por su utilidad al aplicar esta metodología.

#### e) Scrum of Scrums o MetaScrum

Los equipos de Scrum suelen tener entre 5 y 10 personas, sin embargo está metodología ha sido aplicada en proyectos que involucran más de 600 personas. Esto ha sido llevado a cabo dividiendo a los accionistas en equipos de pequeños de hasta 10 personas aproximadamente. Y definiendo jerárquicamente personas que pertenecen a dos equipos, es decir, además de su rol específico dentro de un equipo tienen el rol de enlace en un equipo superior (Peralta, 2003).

##### 2.2.4.2. Procesos

De acuerdo con (Trigas,sf). Scrum consta de tres fases: Pregame, Development y Postgame, como se muestra en la figura 2.3.

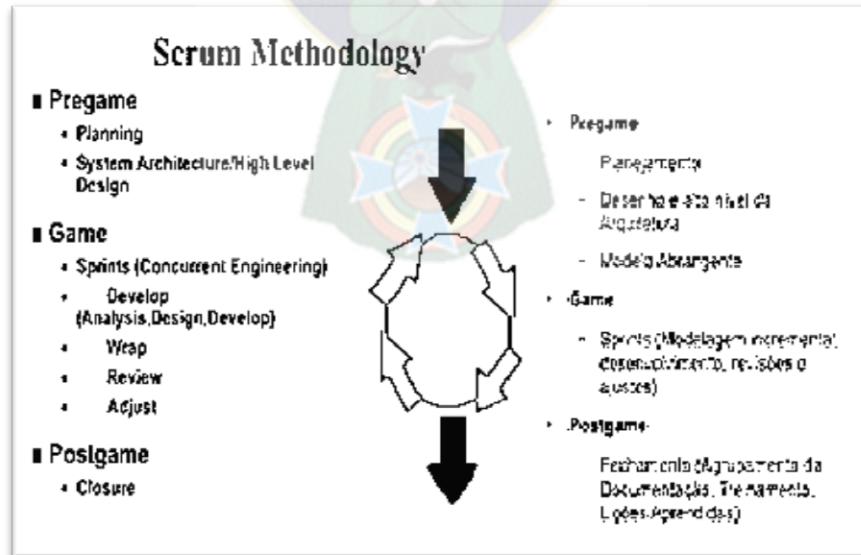


Figura 2.3 Fases de Scrum  
Fuente [Elaboración propia]

### a) Pre-Game

La fase de Pre-game incluye dos subfases: Planning y Architecture.

- **Planning**

Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La Product Backlog List es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems.

- **Architecture / High level Design**

El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la Product Backlog List. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los elementos que aparecen en la lista Product Backlog y el impacto que pueden tener estos cambios. Se sostiene una Design Review Meeting para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión. Se preparan planes preliminares sobre el contenido de cada release.

### b) Game

En esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los Sprints.

Dentro de variables de entorno encontramos: *tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación*. En lugar de tenerlas en

consideración al comienzo del desarrollo, Scrum propone controlarlas constantemente para poder adaptarse a los cambios en forma flexible (Trigas,sf).

### c) Post-Game

Se centra en la integración de los incrementos producidos y liberando el sistema en el entorno de usuario. De acuerdo con (Calzada, 2015) Las tareas que se llevan a cabo en esta etapa son:

- Integración de los incrementos producidos durante los sprints.
- Pruebas de todo el sistema.
- Preparación de la documentación del usuario.
- Preparación de la formación y el material de comercialización.
- Capacitar a los usuarios y operadores del sistema.
- Sistema de conversión / empaquetado.
- Las pruebas de aceptación.

#### 2.2.4.3. Roles y responsabilidades

Los roles y responsabilidades dentro del proceso de desarrollo de la metodología Scrum, de acuerdo con (Calzada, 2015), viene dado por el Scrum Master, Product Owner, Scrum Team, Customer y Management, descritos a continuación.

##### a) Scrum Master

Es un rol de administración que debe asegurar que el proyecto se está llevando a cabo de acuerdo con las prácticas, valores y reglas de Scrum y que todo funciona según lo planeado. Su principal trabajo es remover impedimentos y reducir riesgos del producto. Este rol suele ser desempeñado por un Gerente de Proyecto o Líder de equipo.

**b) Product Owner**

Es el responsable del proyecto, administra, controla y comunica la Backlog List. Es el responsable de encontrar la visión del producto y reflejarla en la Backlog List. Generalmente esta persona puede ser el Product Manager, Marketing, Internal Customer,

**c) Scrum Team**

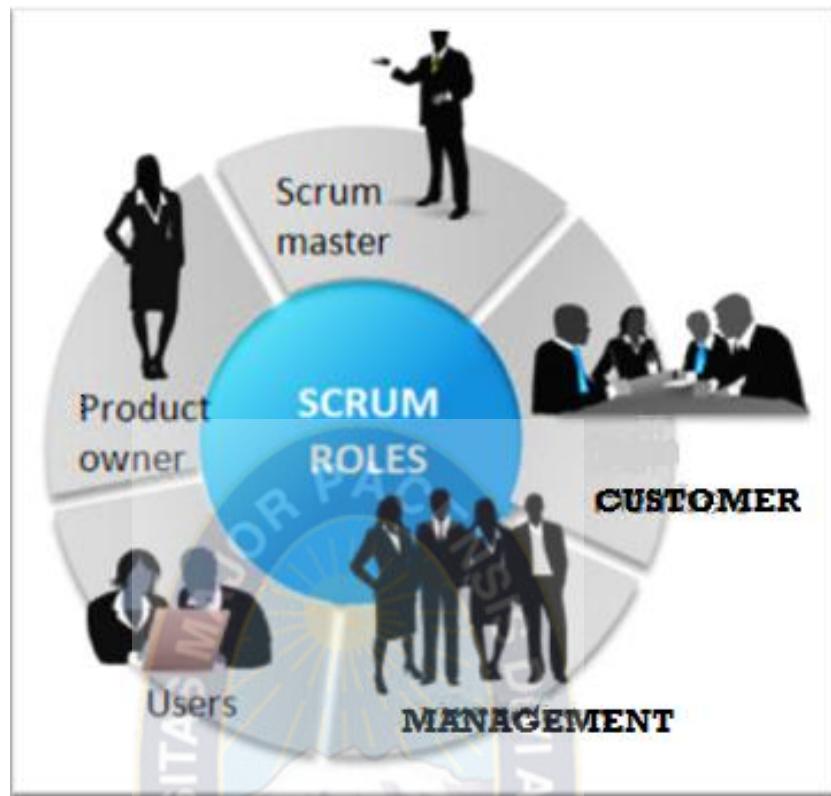
Es el equipo del proyecto que tiene la autoridad para decidir como organizarse para cumplir con los objetivos de un Sprint. Sus tareas son: Effort Estimation (Estimar Esfuerzo), crear el Sprint Backlog, revisar la Product Backlog List y sugerir obstáculos que deban ser removidos para cumplir con los items que aparecen.

Típicamente es un equipo de entre 5 y 10 personas cada una especializada en algún elemento que conforma los objetivos a cumplir, por ejemplo: Programadores, Diseñadores de Interfaz de usuario, etc. La dedicación de los miembros del equipo debería ser full-time con algunas excepciones. La membresía solo puede cambiar entre sprints (no durante).

**d) Customer** El cliente participa en las tareas que involucran la lista Product Backlog.**e) Management**

Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. Participa en la selección de objetivos y requerimientos y en la selección del Scrum Owner. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Scrum Master en la reducción de la Product Backlog List.

En la figura 2.4 se puede observar los tipos de roles que se utilizan al aplicar la metodología Scrum.



**Figura 2.4** Roles y responsabilidad en Scrum  
Fuente [Elaboración propia]

### 2.3. Metodología de diseño

Las metodologías tradicionales de ingeniería de software, o las metodologías para sistemas de desarrollo de información, no contienen una buena abstracción capaz de facilitar la tarea de especificar aplicaciones hipermedia. El tamaño, la complejidad y el número de aplicaciones crecen en forma acelerada en la actualidad, por lo cual una metodología de diseño sistemática es necesaria para disminuir la complejidad y admitir evolución y reusabilidad. Producir aplicaciones en las cuales el usuario pueda provechar el potencial del paradigma de la navegación de sitios web, mientras ejecuta transacciones sobre bases de información, es una tarea muy difícil de lograr.

En primer lugar, la navegación posee algunos problemas. Una estructura de navegación robusta es una de las claves del éxito en las aplicaciones hipermedia. Si el usuario entiende

dónde puede ir y cómo llegar al lugar deseado, es una buena señal de que la aplicación ha sido bien diseñada. Construir la interfaz de una aplicación web es también una tarea compleja; no sólo se necesita especificar cuáles son los objetos de la interfaz que deberían ser implementados, sino también la manera en la cual estos objetos interactuarán con el resto de la aplicación. En hipermedia existen requerimientos que deben ser satisfechos en un entorno de desarrollo unificado (UML) (Rossainz, s.f.).

### 2.3.1. Metodología de diseño OOHDM

Object-Oriented Hypermedia Design Method (OOHDM) es un método orientado a modelos para el desarrollo de aplicaciones Web. Este método permite a los diseñadores especificar una aplicación Web mediante el uso de varios meta-modelos especializados: conceptual, navegación y de interfaz de usuario. Cada meta-modelo pone foco en diferentes aspectos de una aplicación. Una vez que estos modelos han sido especificados para una aplicación dada, es posible generar código en tiempo de ejecución que implemente la aplicación; es decir, los diseños de la aplicación.

OOHDM utiliza mecanismos de abstracción y composición diferentes en un framework orientado a objetos, para permitir por un lado, una descripción concisa de elementos de información correspondientes al negocio subyacente y, por el otro lado, la especificación de escenarios de navegación complejos en base a los datos del modelo conceptual y transformaciones de interfaz para hacer perceptible la información indicada en el modelo anterior. Los principios de OOHDM han sido aplicados sobre una variante del método, SHDM, en la cual el modelo de datos es utilizado está basado en RDF y RDFS (Urbieta, 2012).

De acuerdo con Urbieta (2012), OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cinco fases que se muestran en la figura 2.5:



**Figura 2.5.** Fases del método OOHDM  
**Fuente** [Elaboración propia]

### 2.3.1.1. Obtención de requerimientos

La obtención de requerimientos nos llevará a la comprensión, de cuál será el efecto que tendrá el sistema en la entidad, que es lo que quieren los usuarios y cómo interactúan estos usuarios con el sistema. Para el modelado de los requerimientos se utilizan diagramas para ilustrarlos en forma que sea relativamente fácil de entender. OOHDM propone dividir esta etapa en cinco sub etapas.

#### a) Identificación de roles y tareas

En esta etapa se identificaran los roles que cumplen cada uno de los usuarios. Después para cada rol se identificaran las tareas que debe soportar el sistema.

#### b) Especificaciones de escenario

En esta sub etapa, cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea. Los escenarios son descripciones narrativas de cómo será el sistema utilizado.

### c) Especificación de caso de uso

En esta sub etapa se presentan los casos de uso, donde se muestra cómo interactúan los usuarios (actores) en el sistema. La descripción de un caso de uso tiene que incluir la información presentada en todos los escenarios existentes.

### d) Especificación de diagramas de interacción de usuarios (UID's)

**OOHDM** propone utilizar UIDs, estas son herramientas que nos permiten representar gráficamente la interacción del usuario y el sistema. La elipse nos muestra la información intercambiada entre el usuario y el sistema, la flecha indica que el procesamiento del sistema se produce antes de que se presente la siguiente información. Una línea con una bola negra en un extremo representa las operaciones que no requieren interconexión

### e) Validación de caso de uso y UIDs

En esta sub parte, el desarrollador interactúa con cada usuario para validar los caso de uso y los UIDs, el usuario intercede solo en aquellos casos de uso y UIDs en los que participa

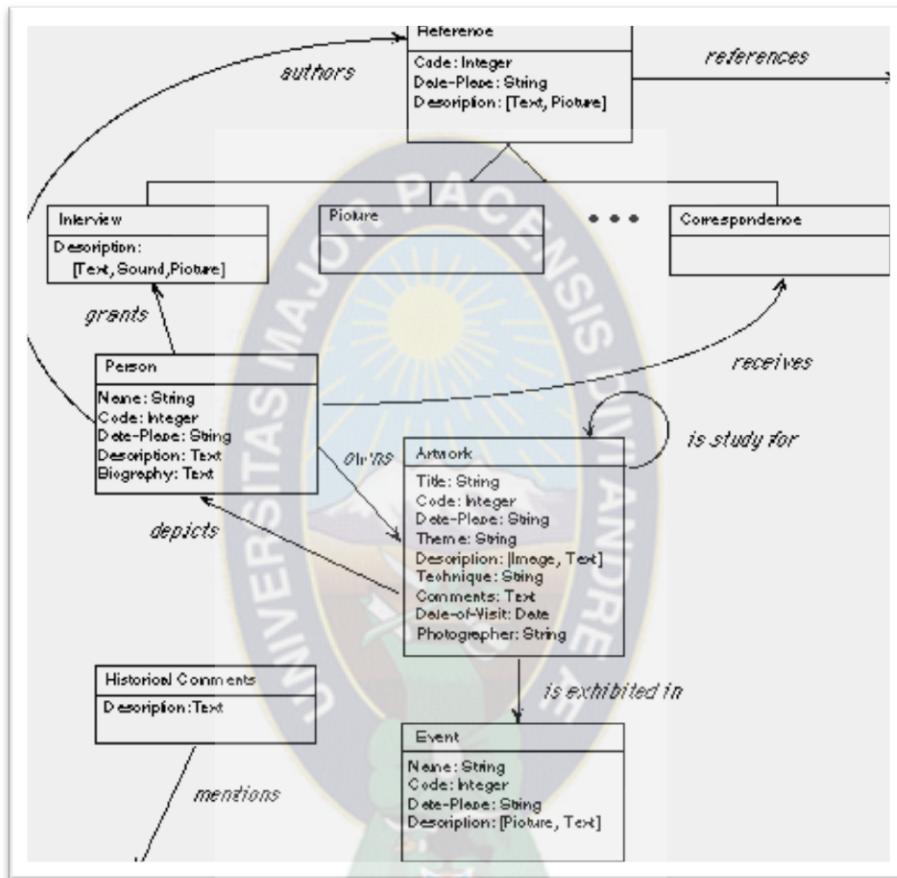
#### 2.3.1.2. Diseño conceptual

En esta fase se construye un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos. En las aplicaciones hipermédia convencionales, cuyos componentes de hipermédia no son modificados durante la ejecución, se podría usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). De este modo, en los casos en que la información base pueda cambiar dinámicamente o se intenten ejecutar cálculos complejos, se necesitará enriquecer el comportamiento del modelo de objetos En OOHDM.

El esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos

pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real (Rossainz, s.f.).

Para un ejemplo de este punto de diseño se tiene la figura 2.6



**Figura 2.6.** Ejemplo del diseño conceptual  
Fuente [ACM, 2013]

### 2.3.1.3. Diseño navegacional

En esta fase se debe tomar en cuenta que la generación de aplicaciones Web fue pensada para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipermedia. En OOHDM, la navegación es considerada un paso crítico en el diseño de aplicaciones.

Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual.

De acuerdo con Urbieta (2012) El diseño de navegación es expresado en dos esquemas:

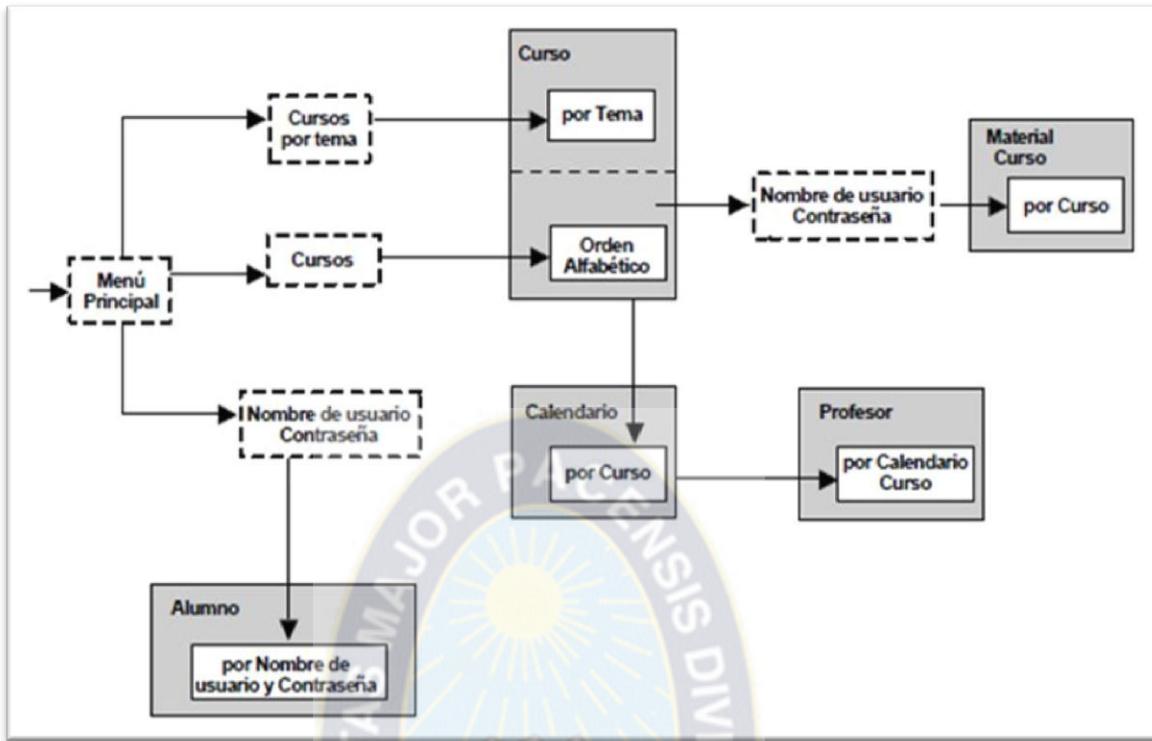
**a) Esquema de clases navegacionales**

En OOHDM existe un conjunto de tipos predefinidos de clases navegacionales: nodos, enlaces y estructuras de acceso. La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipertexto, y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos. La principal estructura primitiva del espacio navegacional es la noción de contexto navegacional.

**b) Esquema de contextos navegacionales.**

Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos, y otros contextos navegacionales (contextos anidados) tal y como se muestra como ejemplo en la figura 2.7. Pueden ser definidos por comprensión o extensión, o por enumeración de sus miembros. Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada.

Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como métodos (comportamiento) cuando se navega en un particular contexto(Urbieta, 2012).



**Figura 2.7** Ejemplo de diseño navegacional  
Fuente [Pinto, 2015]

#### 2.3.1.4. Diseño de interfaces abstractas

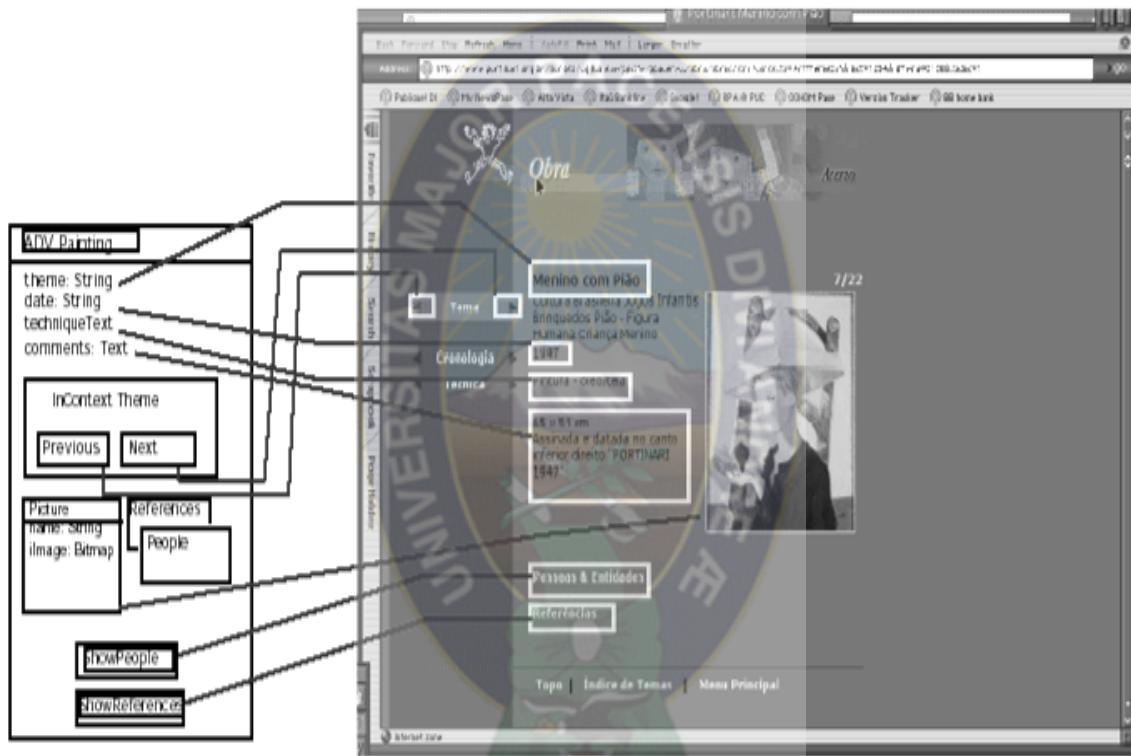
Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, qué transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas. Una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.

El aspecto de la interfaz de usuario de aplicaciones interactivas (en particular las aplicaciones web) es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar.

En OOHMD se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia. El modelo de interfaz ADVs (Vista de Datos Abstracta) especifica

la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación (Rossainz, s.f.).

En la figura 2.8 se pueden observar un ejemplo de como debería ser el análisis en este punto de diseño.

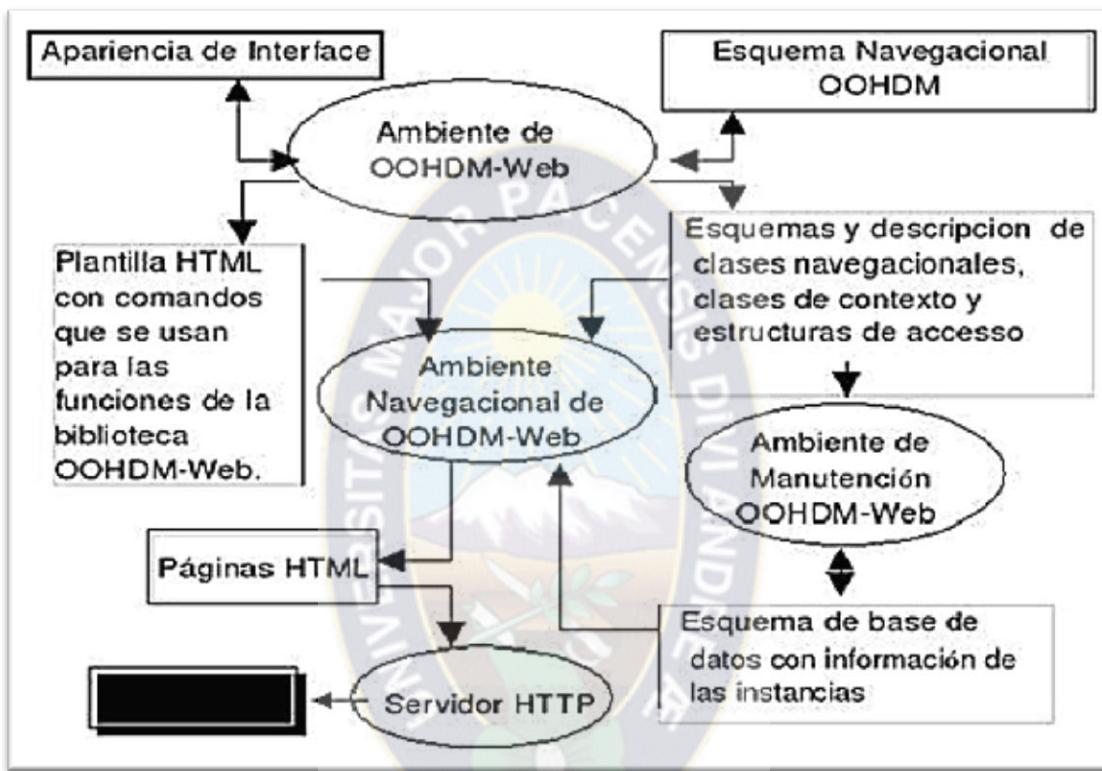


**Figura 2.8** Ejemplo de diseño de interfaz abstracta  
Fuente [Lamarca, 2013]

### 2.3.1.5. Implementación

En esta fase, el diseñador debe implementar el diseño como se puede ver en la figura 2.9. Hasta ahora, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase es tenido en cuenta el entorno particular en el cual se va a correr la aplicación. Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los

ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada (Lamarca, 2013).



**Figura 2.9** Ejemplo de implementación de OOHDMD  
Fuente [Pinto, 2015]

#### 2.4. Calidad de software

La calidad del software es el conjunto de cualidades medibles y específicas que varía de un sistema a otro, dependiendo de tipo de software que se va a desarrollar, para determinar su utilidad y existencia. Este desarrollo debe ser confiable, mantenable y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de utilización y durante las etapas del ciclo de vida del software (SENA, 2015).

#### **2.4.1. Modelo de calidad ISO 9126**

La norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “Information technology–Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software (Borbon, 2013).

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: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de sub características que permiten profundizar en la evaluación de la calidad de productos de software.

#### **2.4.2. Características de la norma ISO 9126**

De acuerdo a Borbon (2013). La norma ISO 9126 cuenta con 6 características generales los cuales son:

##### **2.4.2.1. Funcionalidad**

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas, donde la funcionabilidad de divide en cinco criterios.

- a) **Adecuación:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
- b) **Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

- c) **Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.
- d) **Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados
- e) **Conformidad de la funcionalidad:**  
La capacidad del software de cumplir los estándares referentes a la funcionalidad.

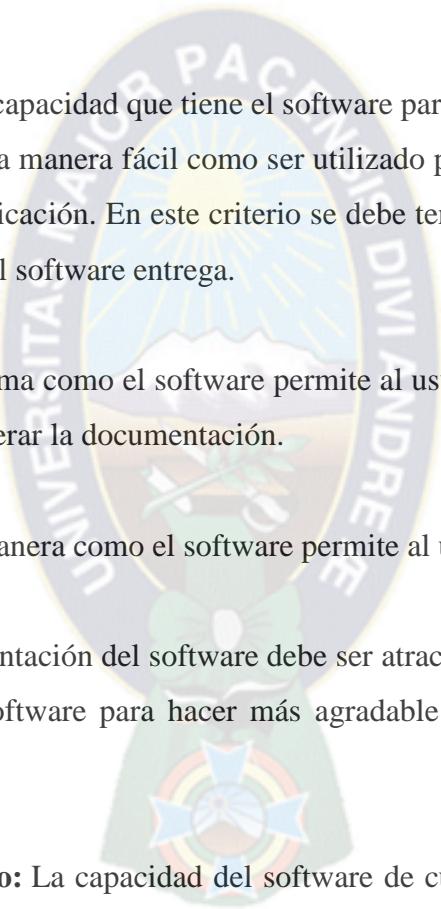
#### 2.4.2.2. Confiabilidad

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. En este caso al confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida. La confiabilidad se divide en cuatro criterios.

- a) **Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente en el disco duro donde esta almacenando los datos.
- b) **Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- c) **Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
- d) **Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

#### **2.4.2.3. Usabilidad**

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. La usabilidad se divide en cinco criterios.

- 
- a) Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
  - b) Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
  - c) Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.
  - d) Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
  - e) Conformidad de uso:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

#### **2.4.2.4. Eficiencia**

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo. La eficiencia tiene tres criterios.

- a) **Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas.
- b) **Utilización de recursos:** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas.
- c) **Conformidad de eficiencia:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

#### 2.4.2.5. Mantenibilidad

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales. Se divide en cinco criterios.

- a) **Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- b) **Cambiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- c) **Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.
- d) **Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- e) **Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

#### **2.4.2.6. Portatilidad**

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera cinco criterios:

- a) **Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados hardware o sistemas operativos, sin que implique reacciones negativas ante el cambio.  
Incluye la escalabilidad de capacidad interna
- b) **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- c) **Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.
- d) **Reemplazabilidad:** La capacidad que tiene el software para ser remplazado por otro software del mismo tipo, y para el mismo objetivo.
- e) **Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

### **2.5. Modelo para el análisis de costos**

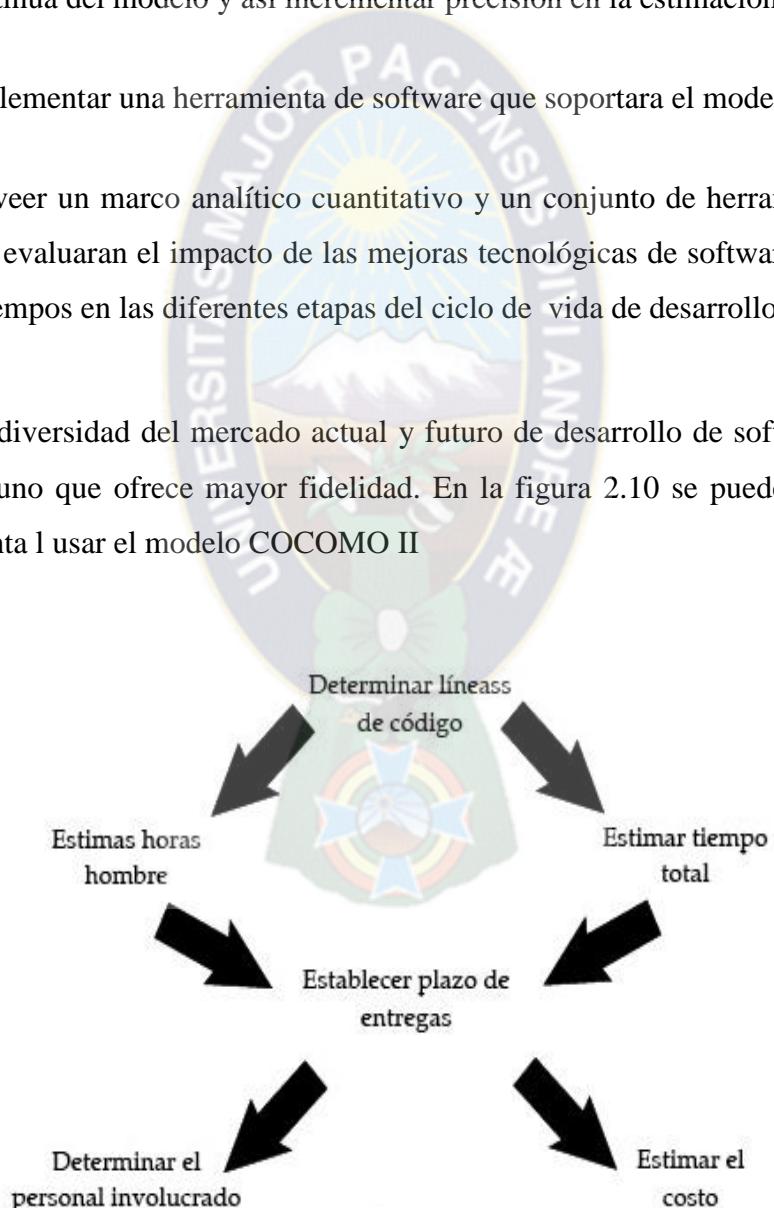
#### **2.5.1. Cocomo II**

El modelo COCOMO II es la evolución exhaustiva del modelo original COCOMO, este nuevo modelo permite realizar estimaciones en función del tamaño de software y del conjunto de factores de costo y de escala. Los factores de costo incluyen aspectos relacionados con la naturaleza del sistema, equipo y características propias del proyecto, los factores de escala incluyen una parte de la escala producida a medida que un proyecto de software incrementa su tamaño.

Los objetivos que se tomaron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de 1 década del 90 como a las futuras.
- Construir una base datos de proyectos de software que permitirá la calibración continua del modelo y así incrementar precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo

En respuesta a diversidad del mercado actual y futuro de desarrollo de software, surgen submodelos, cada uno que ofrece mayor fidelidad. En la figura 2.10 se puede ver los puntos a tomarse en cuenta al usar el modelo COCOMO II



**Figura 2.10** factores que se toman para el análisis de costo

Fuente [Aparicio, 2012]

### 2.5.1.1. Modelo de diseño temprano

El modelo se utiliza en las primera etapas del desarrollo de las cuales se evalúan las alternativas de hardware y software. Al tener poca información en estas etapas, hace que concuerde el uso de PUNTOS FUNCION, para estimar tamaño y el uso de un número reducido factores de costo.

### 2.5.1.2. Puntos de función no ajustados

**Tabla 2.1 puntos función sin ajustar**

Fuente [Pressman, 2010]

TIPO DE ELEMENTO	DIFICULTAD	PESO	CANTIDAD	TOTAL PUNTOS	TOTAL ELEMENTOS
ENTRADAS	Simple	3	E1	Peso*E1	$\sum (Peso * E_i)$
	Media	4	E2	Peso*E2	
	Compleja	6	E3	Peso*E3	
	Total puntos de función entrada				
SALIDAS	Simple	4	S1	Peso*S1	$\sum (Peso * S_i)$
	Media	5	S2	Peso*S2	
	Compleja	7	S3	Peso*s3	
	Total puntos de función salida				
CONSULTAS	Simple	3	C1	Peso*C1	$\sum (Peso * C_i)$
	Media	4	C2	Peso*C2	
	Compleja	6	C3	Peso*C3	
	Total puntos de función consulta				
ARCHIVOS	Simple	7	A1	Peso*A1	$\sum (Peso * A_i)$
	Media	10	A2	Peso*A2	
	Compleja	15	A3	Peso*A3	
	Total puntos de función archivo				
<b>TOTAL PUNTOS DE FUNCION NO AJUSTADOS</b>					$\sum \sum (Peso * X_i)_j$

Donde

**PFNA** son los puntos función no ajustados

$$PFNA = \sum \sum (Peso * X_i)_j$$

### 2.5.1.3. Puntos de función ajustados

el modelo utiliza puntos función ajustado (PFA) como base para medir el tamaño y con los datos calculados en la siguiente ecuación.

$$PFA = PFNA * FA$$

$$FA = 0.65 + 0.01 \sum F_i$$

Dónde:

**FA** es el factor de ajuste

$F_i$  Corresponde a los pesos asignados a los siguientes factores.

**Tabla 2.2** Factores de complejidad  
**Fuente** [Pressman, 2010]

CARACTERISTICA	PESO
Comunicación de datos	$F_1$
Procesamiento distribuido de datos	$F_2$
Rendimiento	$F_3$
Configuraciones fuertemente utilizadas	$F_4$
Frecuencia de transacciones	$F_5$
Entrada de datos on-line	$F_6$
Eficiencia del usuario final	$F_7$
Actualizaciones on-line	$F_8$

Procesamiento complejo	$F_9$
Reusabilidad	$F_{10}$
Facilidad de instalación	$F_{11}$
Facilidad de operación	$F_{12}$
Instalación de distintos lugares	$F_{13}$
Facilidad de cambio	$F_{14}$
TOTAL	$\sum F_i$

Los pesos se consideran dentro de una escala de 0 a 5, donde: 0 es no influencia, 1 es incidental, 2 es moderado, 3 es medio, 4 es significativo, 5 es esencial.

Los puntos función tienen que ser convertidos a líneas de código (KSLOC) considerando el lenguaje de implementación.

$$KSLOC = \frac{PF * LLP}{1000}$$

Dónde:

*LLP* Son las líneas de código requeridos por puntos de función en un lenguaje.

**Tabla 2.3** líneas de código por cada punto función  
**Fuente** [Pressman, 2010]

	LENGUAJE	LCD/PF
<b>Access</b>		38
<b>Ada 95</b>		49
<b>AI Shell</b>		49
<b>APL</b>		32
<b>Assembly – Basic</b>		320
<b>Basic – ANSI</b>		64
<b>Basic – Compilet</b>		91
<b>Basic – Visual</b>		32

<b>C</b>	128
<b>C++</b>	55
<b>Cobol (ANSI85)</b>	91
<b>Base de datos</b>	40
<b>4ta generación de lenguaje</b>	20
<b>Nivelar lenguaje</b>	64
<b>Jovial</b>	107
<b>Machine code</b>	640
<b>Modula 2</b>	80
<b>PHP</b>	29
<b>PERL</b>	27
<b>Prolog</b>	64
<b>Query -default</b>	13
<b>Report Generator</b>	80
<b>Second Generation language</b>	107
<b>Spreadsheeet</b>	46
<b>Third Generation Language</b>	80
<b>USR_5</b>	1
<b>Visual basic 5.0</b>	29

#### 2.5.1.4. Estimación del esfuerzo

El modelo ajusta el esfuerzo nominal usando siete factores de costo. La ecuacion para el cálculo del esfuerzo es la siguiente:

$$PM_{ajustado} = PM_{nominal} * \prod EM_i$$

$$PM_{nominal} = A * SIZE^B$$

$$B = 0.91 + 0.01 * \sum W_i$$

Dónde:

**PM<sub>nominal</sub>** es el esfuerzo nominal ajustado por 7 factores, que reflejan aspectos propios del proyecto.

**SIZE** es el tamaño de software expresado en miles de líneas de código fuente (KSLOC).

**A** es una constante que captura los efectos lineales sobre el esfuerzo ( $A=2.45$ ).

**B** es el factor de ahorro o gasto relativo de escala.

**W<sub>i</sub>** Corresponde a los valores de los siguientes factores de escala.

- **PREC:** Experiencia previa de la organización con este tipo de proyectos.
- **FLEX:** Refleja el grado de flexibilidad en el proceso de desarrollo.
- **RESL:** Refleja la amplitud de análisis de riesgo que se lleva a cabo.
- **TEAM:** Refleja la relación entre los miembros del equipo de desarrollo
- **PMAT:** Refleja la madurez del proceso de la organización.

**Tabla 2.3 Factores de escala**

Fuente [Pressman, 2010]

ACTORES DE ESCALA	MUY BAJO	BAJO	NORMAL	ALTO	MUY ALTO	EXTRA ALTO
<b>PREC</b>	Completa mente sin precedente s 6.20	Práctica mente sin preceden tes 4.96	Casi sin precedentes 3.72	Algo familiar 2.48	Muy familiar 1.24	Completa mente familiar 0.00
<b>FLEX</b>	Riguroso 5.07	Relajació n ocasional 4.05	Algo de relajación 3.04	Conformid ad general 2.03	Algo de conformid ad 1.01	Metas generales 0.00
<b>RESL</b>	Poco 20% 7.07	Poco 40% 5.65	A menudo 60% 2.24	Generalme nte 75% 2.83	En su mayor parte 90% 1.41	Por completo 100% 0.00

<b>TEAM</b>	Interacciones muy difíciles 5.48	Algo de dificultad en las interacciones 4.38	Interacciones básicamente cooperativas 3.29	Bastante cooperativo 2.19	Altamente cooperativo 1.10	Completas interacciones 0.00
<b>PMAT</b>	SW-CMM Nivel 1 7.80	SW.CMM M Nivel I Alto 6.24	SW-CMM Nivel 2 4.68	SW-CMM Nivel 3 3.12	SW-CMM Nivel 4 1.56	SW-CMM Nivel 5 0.00
<b>Peso medio respuestas “SI” para el cuestionario de madurez CMM</b>						

**Tabla 2.4** Valores de los factores de escala  
**Fuente** [Pressman, 2010]

FACTORES DE ESCALA	VALOR
<b>PREC</b>	$W_1$
<b>FLEX</b>	$W_2$
<b>RESL</b>	$W_3$
<b>TEAM</b>	$W_4$
<b>PMAT</b>	$W_5$
<b>Resultado</b>	$\sum W_i$

- **EM<sub>i</sub>** Corresponde a los factores de ajuste que tienen un efecto multiplicativo sobre el esfuerzo.

Los siguientes puntos describen los factores de ajuste en categorías:

#### Del producto

- **RCPX:** fiabilidad y confiabilidad del proyecto
- **RUSE:** Requerimiento de reusabilidad

### De la plataforma

- **PDIF:** Dificultad de la plataforma

### Del personal

- **PERS:** Capacidad del personal
- **PREX:** Experiencia del personal

### Del proyecto

- **FCIL:** Facilidad para el desarrollo
- **ACED:** Esfuerzo del calendario

Tabla 2.5 Factores de ajuste  
Fuente [Pressman, 2010]

DRIVERS	EXTRA BAJO	MUY BAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
<b>RCPX</b>	0.73	0.81	0.98	1	1.30	1.74	2.38
<b>RUSE</b>	-	-	0.95	1	1.07	1.15	1.24
<b>PDIF</b>	-	-	0.87	1	1.29	1.81	2.61
<b>PERS</b>	2.12	1.62	1.26	1	0.83	0.63	0.50
<b>PREX</b>	1.59	1.33	1.12	1	0.87	0.71	0.62
<b>FCIL</b>	1.43	1.30	1.10	1	0.87	0.73	0.62
<b>SCED</b>	-	1.43	1.14	1	1	1	-

**Tabla 2.6** Valores de los factores de ajuste obtenidos  
**Fuente** [Pressman, 2010]

DRIVERS	VALOR
<b>RCPX</b>	$EM_1$
<b>RUSE</b>	$EM_2$
<b>PDIF</b>	$EM_3$
<b>PERS</b>	$EM_4$
<b>PREX</b>	$EM_5$
<b>FCIL</b>	$EM_6$
<b>SCED</b>	$EM_7$
<b>Resultado</b>	$\prod EM_i$

#### 2.5.1.5. Estimación del cronograma

La ecuación inicial es:

$$TDEV = [3.67 * PM^{(0.28+0.2*(B+1.01))}] * \left[ \frac{SCED\%}{100} \right]$$

Dónde:

**TDEV** es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad.

**PM** Es el esfuerzo expresado en meses persona.

**B** Es el factor de escala

**SCED%** Es el porcentaje de comprensión/expansión del cronograma.

Para el número de personas (NP) requeridas se utiliza lo siguiente ecuación:

$$NP = \frac{PN_{ajustado}}{TDEV}$$

### **2.5.1.6. Costo total del sistema**

El costo del desarrollo (CD) se obtiene aplicando la siguiente ecuación:

$$CD = TDEV * PS * NP$$

Dónde:

*PS* Es el sueldo promedio de un desarrollador de software

## **2.6. Metodología de evaluación de calidad de sitios web WEB-SITEQEM**

Esta metodología propone un enfoque sistemático, disciplinado y cuantitativo que se adecua a la evaluación, comparación y análisis de calidad de sistemas de información centradas en la Web. Además, incluye un conjunto de fases, actividades, productos, modelos y constructores de proceso. Uno de los objetivos principales de la evaluación y comparación de calidad de artefactos Web, radica en comprender el grado de cumplimiento de un conjunto de características y sub características con respecto a los requerimientos de calidad establecidos. De este modo, otro aporte interesante consiste en la definición de características, sub características y atributos cuantificables considerados dominios de aplicaciones Web.

Segun Olsina (2002) esta metodología comprende una serie de fases principales las cuales son:

- Fase de definición y especificación de los requerimientos de calidad.
- Fase diseño e implementación de la evaluación elemental.
- Fase de diseño e implementación de la evaluación global.
- Fase de análisis de resultados y recomendaciones.

### **2.6.1. Fase de definición y especificación de los requerimientos de calidad**

Esta comprende actividades y procedimientos sirven para modelar y especificar los requerimientos de calidad. A partir de un proceso de evaluación, con el fin de analizar,

comparar, mejorar y comprender atributos y características de artefactos Web, los requerimientos deben responder a necesidades de un perfil de usuario o dominios establecidos.

De las características de calidad ISO derivamos las sub características y de estas derivamos los atributos. A cada atributo cuantificable del dominio empírico, le asociamos una variable en el dominio numérico, esta variable toma un valor real que será medido.

### **2.6.2. Fase diseño e implementación de la evaluación elemental**

Esta fase comprende actividades, modelos, técnicas, heurísticas y herramientas que sirven para definir una base de criterios de evaluación para los atributos cuantificables y realizar un proceso de medición. Para cada atributo medible o cuantificable  $A_i$  del árbol de requerimientos, debemos asociar y determinar una variable  $X_i$ , que tendrá un valor numérico a partir de un proceso de medición. Demás, para cada variable  $X_i$  computada, por medio de un criterio elemental, producirá una preferencia elemental  $I_E i$ . Este resultado final es un indicador elemental que es interpretado como el grado o porcentaje del requerimiento del usuario satisfecho para el atributo  $A_i$ .

Para cada variable de calidad medida  $X_i$ ,  $i=1,\dots,n$  se define una función que representa el criterio elemental. Un criterio elemental es una correspondencia del valor de la variable de calidad  $X_i$  en el valor de preferencia (8º indicador) elemental de calidad  $I_E i$ . En términos generales, el valor medido de la variable es un número real:  $X_i \in R \in R$

El valor de la preferencia de calidad elemental es también un número real pero perteneciente al intervalo unitario I, de manera que:  $I_E i \in I = [0,1]$ .

con frecuencia, en vez de usar el intervalo unitario es útil emplear la escala de [0,100]. En este sentido se interpreta a la preferencia como el porcentaje del requerimiento satisfecho.

Desde el punto de vista analítico, el criterio elemental se define como la función:

$$F_i: R \rightarrow I \text{ en donde } I_E i = F_i(X_i), X_i \text{ min} \leq X_i \leq X_i \text{ max}$$

Sea el tiempo  $T_i$  que representa el tiempo de respuesta promedio. Entonces en criterio elemental para dicho atributo puede ser definido como.

$$\begin{aligned} IE_i = 1(0 \text{ } 100\%) & \quad \text{si } T_i \leq T_{i \min} \\ IE_i = 0(0 \text{ } 0\%) & \quad \text{si } T_i \geq T_{i \max} \\ IE_i = (T_{i \max} - T_i) / (T_{i \max} - T_{i \min}) & \quad \text{si } T_{i \min} \leq T_i \leq T_{i \max} \end{aligned}$$

Un criterio absoluto se emplea para determinar la preferencia absoluta de un atributo de un artefacto. Un criterio relativo consiste en la determinación de los indicadores relativos de los sistemas comparados sin evaluar la calidad de cada sistema de un modo individual e independiente.

### Criterios elementales absolutos con variable continua

- **Criterio de variable única** Este es un criterio elemental común, se asume que la variable  $X$  es única y continua. Con el fin de determinar el criterio elemental, el primer paso consiste en definir el rango de valores de interés para la evaluación de la variable continua. El siguiente paso, consiste en determinar las coordenadas de los puntos más relevantes y su preferencia de calidad.
- **Criterio de variable** Este es un criterio elemental que se suele utilizar para evaluar la relación entre dos atributos con criterios absolutos de un mismo sistema.
- **Criterio de Multi-variables continuas** En este tipo de criterio, la variable  $X$  es resultante de algunas otras variables y constantes (el valor  $X$  corresponde a una métrica indirecta).
- **Criterio de preferencia de calidad directa** Este tipo de criterio es subjetivo y basado en la experiencia y criterio de los evaluadores. Desde el punto de vista de la precisión y objetividad, es el peor criterio, debido que se pueden introducir errores de

valoración intencionales y/o involuntarios. El criterio de la variable X se mapea en una preferencia trivial cuyas coordenadas son:  $\text{CrE (Xi)}=\{(0,0),(100,100)\}$ .

### Criterios elementales absolutos con variable discreta

- **Criterio binario** Este criterio es el más simple de los criterios discretos y absolutos. El criterio para la variable X se mapea una preferencia elemental cuyas coordenadas son:

$$\text{CrE (Xi)} = \{(0,0),(1,100)\}.$$

En donde un valor de  $X_i = 0$  se interpreta como la ausencia del atributo de calidad; en cambio un valor de  $X_i = 1$ , se interpreta como la presencia o disponibilidad del mismo.

- **Criterio de Multi-nivel.** Este criterio es una generalización del criterio binario. La variable discreta puede tomar más de los valores, cada uno de los cuales se corresponde a una preferencia de calidad. La variable X se mapea en valores de preferencia cuyas coordenadas son:

$$\text{CrE (Xi)} = \{(0,0),(1,60),(2,100)\}.$$

Donde  $X_i = 0$  se interpreta como la ausencia del atributo de calidad, en cambio un valor de  $X_i = 1$ , se interpreta como la presencia parcial, finalmente un valor  $X_i = 2$ , se interpreta como la total.

- **Criterio de Multi-nivel definido como subconjunto.** Este criterio es uno Multi-nivel definido como un subconjunto de los números naturales. La variable discreta puede tomar más de dos valores, cada uno de los cuales se corresponde a una preferencia de calidad, la variable X se mapea en valores de preferencia cuyas coordenadas son:

$$\text{CrE } (\text{Xi}) = \{(0,0), (1,100), (2,100)\}$$

Donde  $\text{Xi}$  tiene los siguientes valores:

0 = ausencia de mecanismo de búsqueda restringida.

1 = búsqueda básica: mecanismo de búsqueda por nombre/apellido.

2 = 1 + búsqueda extendida o avanzada: mecanismo de búsqueda por tipo de producto o entre otros.

- **Criterio Multi-variables discretas.** Este criterio permite agrupar varias variables discretas y modelar el resultado en una única variable  $X$ . De este modo se puede reducir la cantidad de criterios elementales. Sea el conjunto de variables discretas  $D_1, \dots, D_n$ , entonces se puede definir en una variable auto compuesta  $X$ , también discreta, como función de las anteriores.

$$X = F(D_1, \dots, D_n), \text{ y } X \in \{X_1, \dots, X_n\}$$

### 2.6.3. Fase de diseño e implantación de la evaluación global

Esta fase comprende actividades, modelos, procedimientos y herramientas para determinar los criterios de agregación de las preferencias de calidad elemental, para producir  $n$  preferencias de calidad elemental para los  $n$  atributos considerados en el árbol de requerimientos. Aplicando el mecanismo de agregación paso a paso, las preferencias elementales se pueden agrupar para producir al final un esquema de agregación. Las preferencias de calidad parcial y global se pueden obtener mediante cálculo conforme al modelo de agregación y puntaje. Se puede utilizar el modelo de Agregación Lógica de Preferencias (LSP) para estructurar y computar la calidad global de los sitios Web.

Entre las características generales de LSP, están:

- Es un modelo de agregación y puntaje para evaluar sistemas complejos

- Sus resultados representan el grado de satisfacción de los usuarios conforme a los requerimientos de calidad establecidos.
- Es una generalización de los modelos y técnicas de puntaje aditivo y lineal.
- Tiene sus fundamentos en principios, modelos matemáticos y de lógica.

La función utilizada para calcular la calidad se denomina media potencia pesada la cual es:

$$IG(r) = (P_1 * IE_1^r + P_2 * IE_2^r + \dots + P_m * IE_m^r)^{1/r}$$

$$-\alpha \leq r \leq +\alpha$$

Dónde:

$P_i$  Es el peso que está sujeto a los siguientes valores

$$(P_1 + P_2 + \dots + P_m) = 1 ; P_i > 0; i = 1 \dots m$$

$IE_I$  Es el indicador elemental que esta entre

$$0 \leq IE_I \leq 1$$

La preferencia resultante  $IG_i$  tiene un valor entre:

$$\text{Min}(IE_1, IE_2, \dots, IE_m) \leq IG_i \leq \text{Max}(IE_1, IE_2, \dots, IE_m)$$

#### 2.6.4. Fase de análisis de resultados y recomendaciones

Esta fase comprende actividades de análisis y comparación de las preferencias de calidad elemental, parcial y global que justifica los resultados. A partir de las metas establecidas y el punto de vista de usuario a evaluar, el proceso culmina con las conclusiones y recomendaciones del caso. Por otra parte se utilizan herramientas y mecanismos de análisis y documentación para facilitar la interpretación de los datos, su registro y seguimiento.

Los datos numéricos e información mostrada en tablas, gráficos y otros documentos, pueden ser eficientemente empleados en actividades de análisis y toma de decisiones. Por lo tanto, un nivel de aceptabilidad satisfactorio indica que los valores están en el rango de 60 a 100%. Un nivel a aceptabilidad marginal indica que los valores de la preferencia caen en el rango de 40 a 60%. Finalmente un valor de preferencia es insatisfactorio cae en rango de 0% y el 40% de los requerimientos de calidad.



## CAPITULO III

### MARCO APLICATIVO

#### 3.1. Introducción

Según la ingeniería de software, el diseño y desarrollo de todo producto software existe una serie de actividades que deben realizarse en un orden determinado y que abarca no solo su producción, sino también su explotación y mantenimiento, esto se llama ciclo de vida o proceso de desarrollo de software.

Para la implementación de este proyecto es necesario estar en constante comunicación con el cliente que participan en el uso y administración del sistema, con el fin de identificar las tareas, escenarios, casos de uso diagramas de interacción de usuario.

#### 3.2. Proceso de desarrollo

Para el proceso de desarrollo de software se utiliza un modelo de proceso incremental de Scrum y se complementa con la metodología de diseño OOHDM, para hacer el seguimiento del sistema.

##### 3.2.1. Obtención de requerimientos

###### 3.2.1.1. Identificación de roles y tareas

La siguiente descripción muestra los usuarios y las tareas definidas para el sistema.

**Rol:** ADMINISTRADOR (Propietario)

**Tareas:**

- El administrador tiene acceso a toda la información de la página web
- El administrador puede registrar a usuarios con privilegios directo desde su cuenta de administrador.
- El administrador podrá ver toda la información sin restricción de todos los datos del sistema y modificarlos según sus requerimientos.
- El administrador podrá ver los registros de compra y venta de la fecha que requiera.

**Rol:** REGENTE (Encargado)

**Tareas:**

- El Regente tiene acceso a toda la información de la página web, excepto a realizar modificaciones en las características de los insumos
- El Regente podrá realizar actualizaciones en el stock de los productos y no podra modificar su características
- El regente podrá realizar un reporte diario de los movimientos de compra y venta de insumos.
- El regente podrá adicionar o eliminar desde su cuenta a clientes y modificar sus datos.

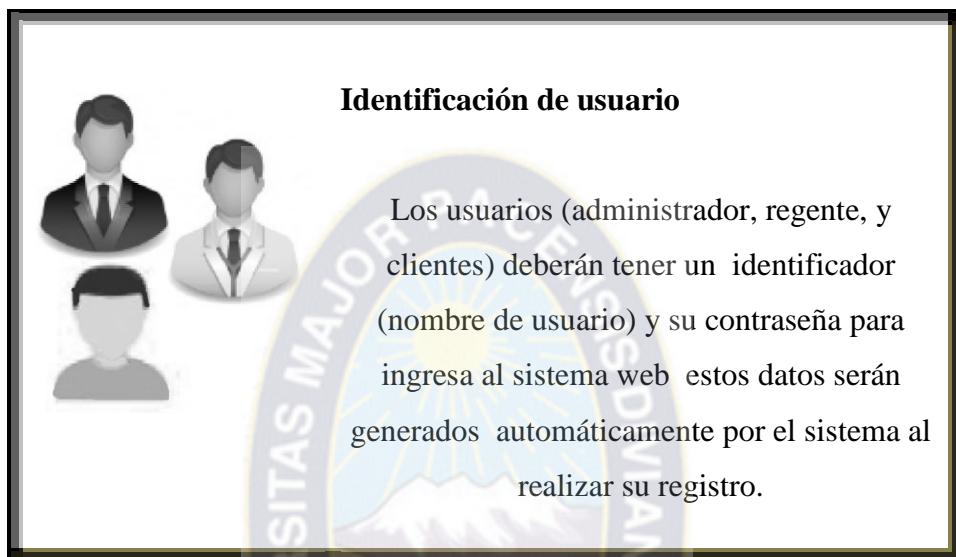
**Rol:** USUARIO (Cliente)

**Tareas:**

- El Usuario tiene acceso a toda la información de la página web.
- El Usuario podrá registrarse en cualquier momento en la web.
- El Usuario podrá modificar sus datos en el sistema.
- El Usuario visualiza sus puntos acumulados por las compras realizadas.
- El Usuario podrá realizar una lista de insumos e imprimirlo.
- El Usuario podrá realizar búsqueda de insumos y ver sus características.

### 3.3.2.2. Especificación de escenarios

Los siguientes escenarios muestran textualmente como será utilizado el sistema con respecto a los roles y tareas ya descritos anteriormente.



**Figura 3.1** Escenario de identificación de usuario

**Fuente:** [Elaboración propia]



**Figura 3.2** Escenario de Registro de Regente

**Fuente:** [Elaboración propia]



**Figura 3.3** Escenario de Registro de clientes

**Fuente:** [Elaboración propia]

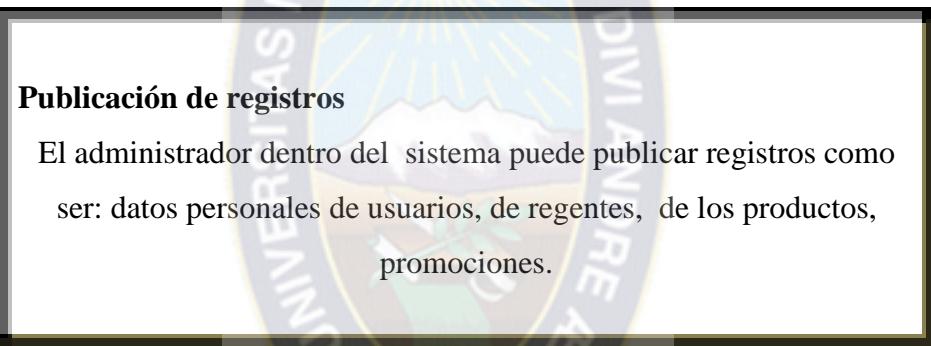


**Figura 3.4** Escenario de Registro de laboratorio

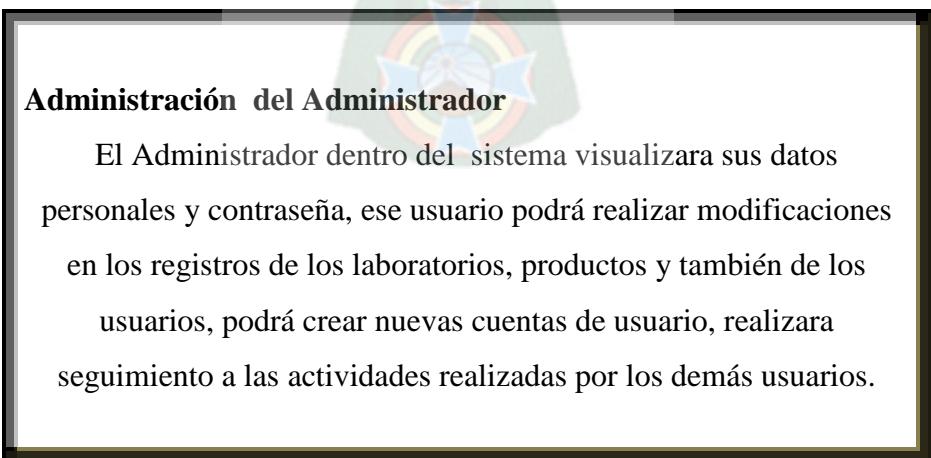
**Fuente:** [Elaboración propia]



**Figura 3.5 Escenario de Registro de productos**  
**Fuente:** [Elaboración propia]



**Figura 3.6 Escenario de Publicación de productos**  
**Fuente:** [Elaboración propia]



**Figura 3.7 Escenario de administración del Administrador**  
**Fuente:** [Elaboración propia]

### **Administración del Regente**

El Regente dentro del sistema no tendrá los mismos privilegios que el administrador pero podrá registrar a nuevos clientes y realizar modificaciones a sus datos, también podrá ver los reportes diarios del movimiento realizado en la farmacia.

**Figura 3.8** Escenario de administración del Regente

**Fuente:** [Elaboración propia]

### **Administración del Cliente**

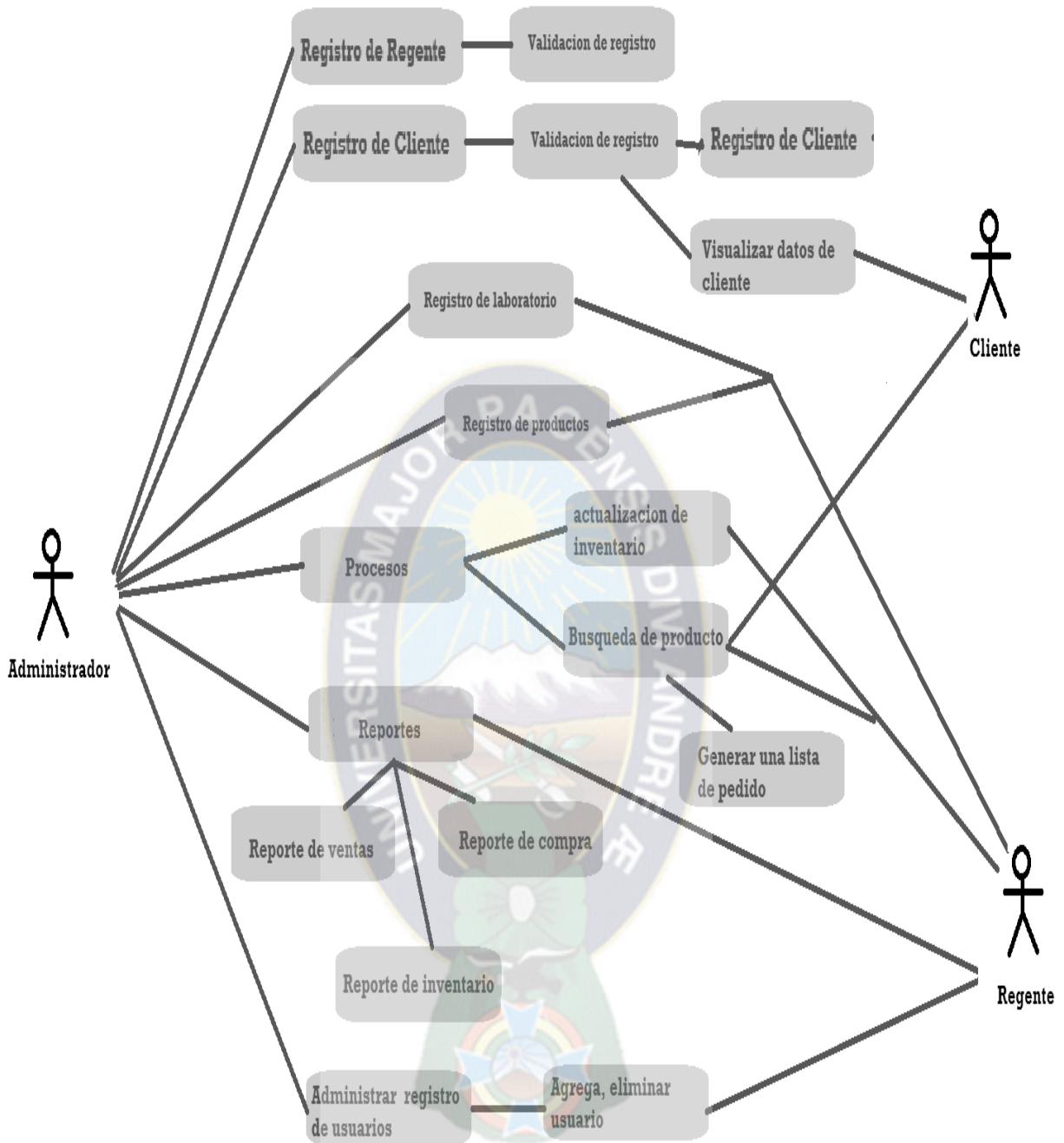
El Cliente dentro del sistema podrá modificar sus datos personales y modificar su contraseña, también podrá acceder toda información que el sistema proporciona , visualizar toda la información de los productos.

**Figura 3.9** Escenario de administración del Cliente

**Fuente:** [Elaboración propia]

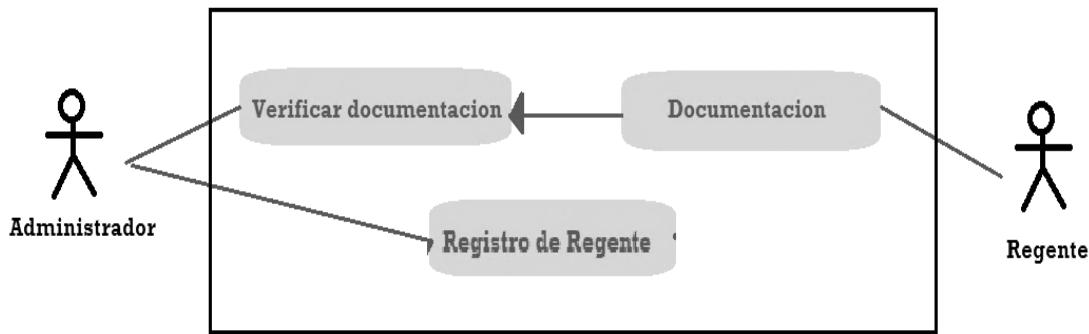
#### **3.3.2.3. Especificación de casos de uso**

En la figura 3.10 se muestra el diagrama de caso de uso general del sistema, donde se identificaran los procesos y actividades que realizan los usuarios (administrador, regentes, clientes) y las acciones respecto al sistema web de Farmacias Alpaso.



**Figura 3.10** Diagrama de casos de uso general del sistema  
**Fuente:** [Elaboración propia]

En la figura 3.11 se muestra al administrador y la manera en la que interactúa con el sistema respecto al registro de regentes, clientes, laboratorios y productos.



**Figura 3.11** Diagrama de casos de uso Registro de Regente  
**Fuente:** [Elaboración propia]

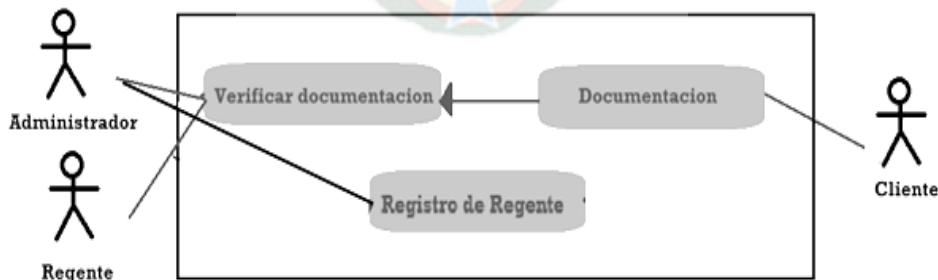
En el caso de uso registro de regente que se ve en la figura 3.10 se tiene las siguientes características:

*Actores:* Administrador y Regente

*Propósito:* El Administrador registra a nuevos usuarios (regentes) para que ellos puedan administrar el sistema.

Para que el regente pueda ser registrado tiene que presentar la documentación requerida por el administrador.

En la figura 3.121 se tiene el caso de uso de registro de cliente



**Figura 3.12** Diagrama de casos de uso Registro de Cliente  
**Fuente:** [Elaboración propia]

En este caso de uso **registro de usuario cliente**, podemos ver que los actores son: Administrador, Regente y cliente.

El cliente puede registrarse online en cualquier momento, no es necesario que presente una documentación. En el caso de que lo realice en las instalaciones de la farmacia será requerido su documento de identidad.

En figura 3.13 se muestra el caso de uso de registro de la descripción de nuevos laboratorios, y el de los productos.

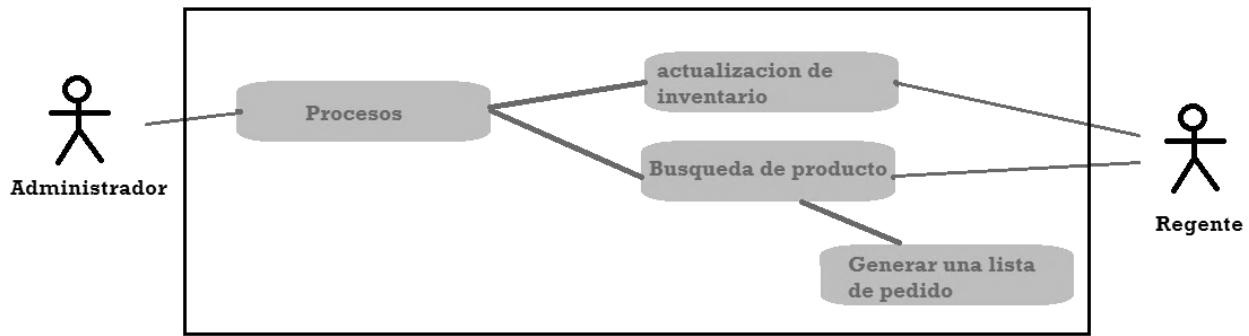


**Figura 3.13** Diagrama de casos de uso Registro de Cliente  
**Fuente:** [Elaboración propia]

Este caso de uso nos describe tanto el usuario de tipo administrador es el único que puede realizar el registro de un nuevo laboratorio.

En el caso de realizar un nuevo registro de productos los usuarios que los pueden realizar son el administrador y regente.

En la figura 3.14 se puede ver los procesos que pueden realizar tanto el administrador y el regente.



**Figura 3.14** Diagrama de casos de uso procesos

Fuente: [Elaboración propia]

Este caso de uso nos muestra los procesos que pueden realizar los usuarios (administrador y regente), pero en este caso el administrador podrá realizar actualizaciones en el stock de almacenes y realizar una lista de nuevos pedido, realizando una búsqueda específica de un producto, mientras tanto el regente solo podrá realizar un inventario por día para presentar su informe diario de movimiento.

En la figura 3.15 se puede ver el caso de realizar reportes tanto de administrador como del regente.



**Figura 3.15** Diagrama de casos de uso Reportes

Fuente: [Elaboración propia]

El administrador y el regente podrán hacer sus reportes diarios, mensuales y anuales de ventas, compras e inventarios. Pero solo el administrador tendrá la opción de visualizar los reportes de los demás usuarios tipos clientes.

En la figura 3.16 se puede observar el diagrama de casos de uso de los permisos de eliminar y registra usuarios nuevos.



**Figura 3.16** Diagrama de casos de uso Administración de usuarios

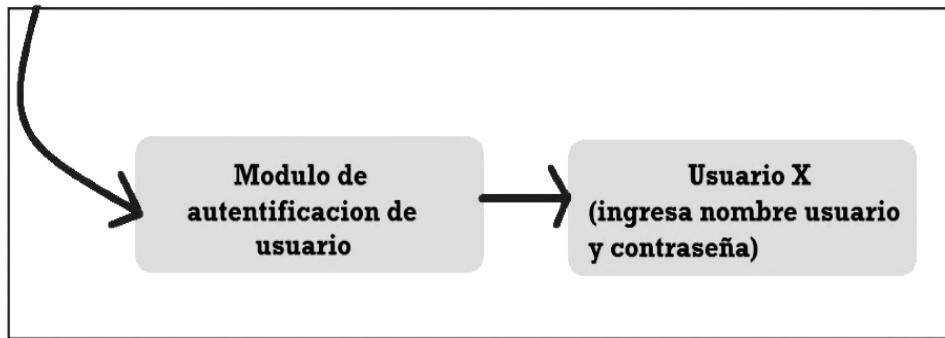
Fuente: [Elaboración propia]

Podemos ver las actividades que pueden realizar el administrador y el gerente, para la función de administración de usuarios el administrador tendrá todos los permisos para realizar cualquier modificación o cambio de todos los usuarios, pero los permisos del regente serán que podrá realizar modificaciones y cambios a los usuarios de tipo cliente únicamente.

### 3.3.2.4. Especificaciones de UIDs

Si bien la especificación de casos se la realizó con UML, como también las historias de usuarios, OOHDM propone utilizar los diagramas de interacción de usuarios (UIDs) para mostrar el intercambio de información entre los usuarios (administrador, regente y cliente) y el sistema web.

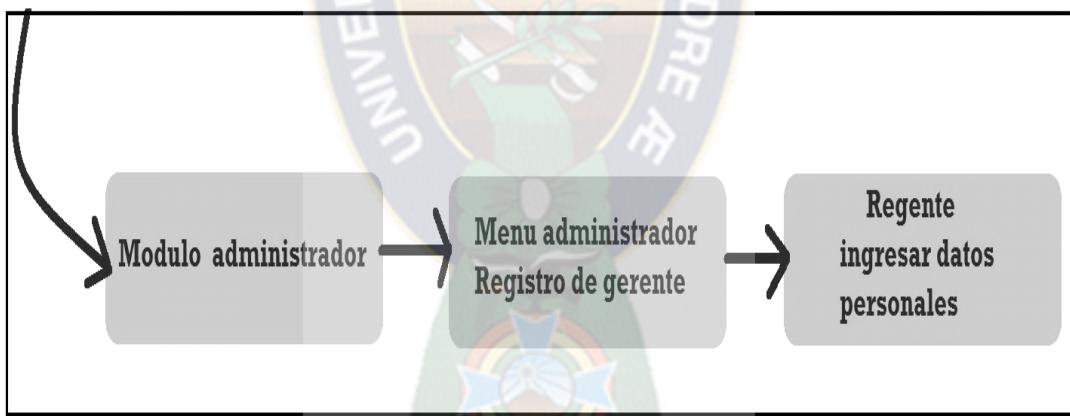
Para la interacción del sistema a con los clientes, para su autentificación estos módulos se detallan en la figura 3.17.



**Figura 3.17** UID Autentificación de usuarios

Fuente: [Elaboración propia]

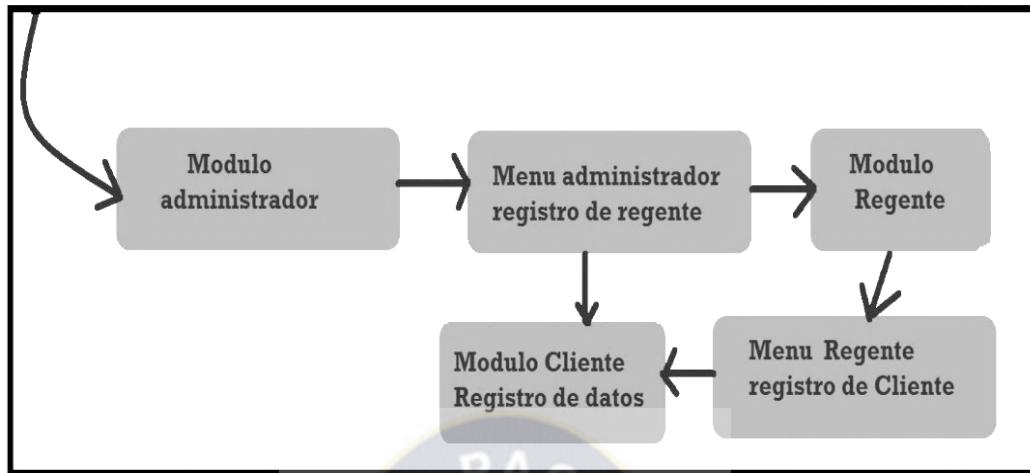
El diagrama de la figura 3.18, corresponde al caso de uso registro de Regente como se muestra en la figura 3.11, la primera interacción presenta el módulo de administrador, la segunda interacción muestra el menu administrador general, y la tercera presenta el objeto de regente que contiene un conjunto de atributos.



**Figura 3.18** UID Registro de Usuario Regente

Fuente: [Elaboración propia]

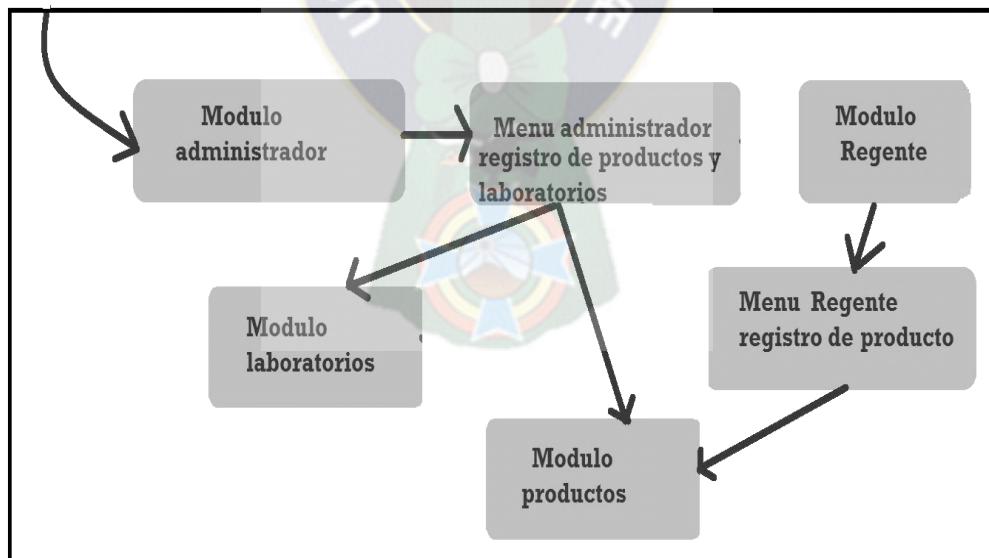
En la siguiente figura 3.19 detallaremos el caso de uso de registro de cliente muestra en la figura 3.12, la primera interacción presenta al módulo Usuario/regente, a este le sigue la interacción menu Regente que contiene el registro de clientes que tiene diferentes atributos.



**Figura 3.19 UID Registro de Usuario Cliente**

**Fuente:** [Elaboración propia]

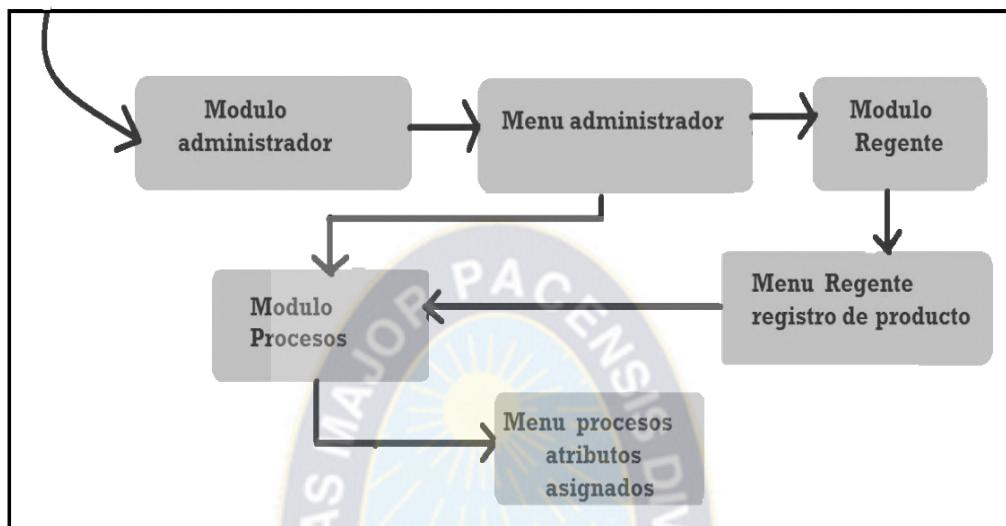
El diagrama de la figura 3.20 detallaremos el caso de uso de registro de laboratorios y productos que se muestra en la figura 3.13, mostrando el modulo administrador, menu administrador, modulo regente menu regente, modulo registro de laboratorios y finalizando con el módulo de registro de productos.



**Figura 3.20 UID Registro de Laboratorio y productos**

**Fuente:** [Elaboración propia]

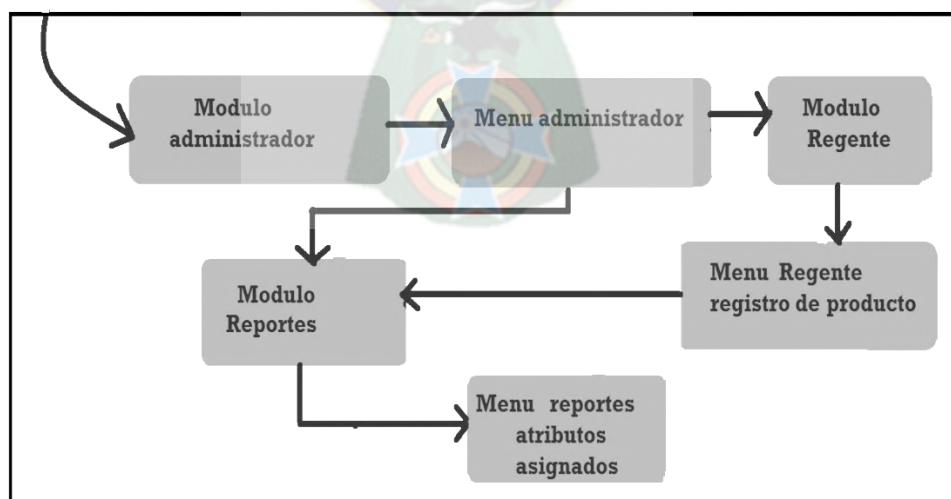
El diagrama de la figura 3.21 se detallara el caso de uso de procesos como se muestra en la figura 3.14, las interacciones que representan son el menu de administrador, menu de procesos, y sus diferentes atributos, como también menu de regente.



**Figura 3.21** UID procesos de administrador y regente

**Fuente:** [Elaboración propia]

En el diagrama de la figura 3.22 se detallaran el casos de uso de los reportes como se ve en la figura 3.15, las interacciones que se realizan en este son: módulo de administrador, menu administrador, módulo de regente, menu de regente, módulo de procesos y sus atributos correspondientes.



**Figura 3.22** UID Reportes de administrador y regente

**Fuente:** [Elaboración propia]

### **3.4. Pre – game**

#### **3.4.5. Planificación**

Como primera etapa del Pre-Game de Scrum se tiene la planificación donde se creó la lista priorizada a partir del análisis de requerimientos de la farmacia AlPaso realizadas anteriormente. Estos son los requerimientos funcionales que describen las interacciones entre el sistema y el entorno, estos son representados en la tabla 3.1.

**Tabla 3.1 Requerimientos**

**Fuente:** [Elaboración propia]

CODIGO	REQUERIMIENTO
RI_1	Diseñar la base de datos, que permita almacenar los datos de los laboratorios y los productos, que permita almacenar los proceso que se vayan a realizar, también almacenarlos reportes generados diariamente.
RI_2	Se deben manejar roles de usuarios de tipo administrador, regente y clientes son sus respectivos permisos
RI_3	La inserción, modificación y eliminación de la cuentas de usuario
RI_4	Reportes de las ventas, compras y modificación de las cantidad de stock en almacenes.
RI_5	Registrar las ventas, compras y modificaciones por dia
RI_6	Realizar procesos de búsqueda en la base de datos de productos requeridos por el cliente
RI_7	Realizar pedidos desde la página web

Los requerimientos de hardware y software que permiten que el sistema funcione correctamente se detallan en la tabla 3.2

**Tabla 3.2** Requerimientos de hardware y software

Fuente: [Elaboración propia]

CODIGO	REQUERIMIENTO
RH_1	Pc de escritorio, Tablet, o un teléfono celular android
RH_2	Contar con conexión a internet
RH_3	Tener instalado Adobe pdf, en la pc de escritorio, Tablet y/o celular.

### 3.5. Game

Se procede a la programación de trabajo y tiempo requerido para poder realizar el diseño de la estructura del sistema, a esto lo llamamos el primer sprint tomando las siguientes características.

- a) **Reunión y planificación del sprint I** Diseñar la base de datos, que permita almacenar los datos de los laboratorios y los productos, que permita almacenar los proceso que se vayan a realizar, también almacenarlos reportes generados diariamente.
- b) **Scrum diario del Sprint I** El desarrollo del Sprint mediante la aplicación de la metodología de diseño OOHDM del módulo de diseño conceptual donde se realizara en el diseño de la base datos analizando los requerimientos antes mencionados, en la tabla 3.3 se detallan las tareas del Sprint I.

**Tabla 3.3** Sprint I

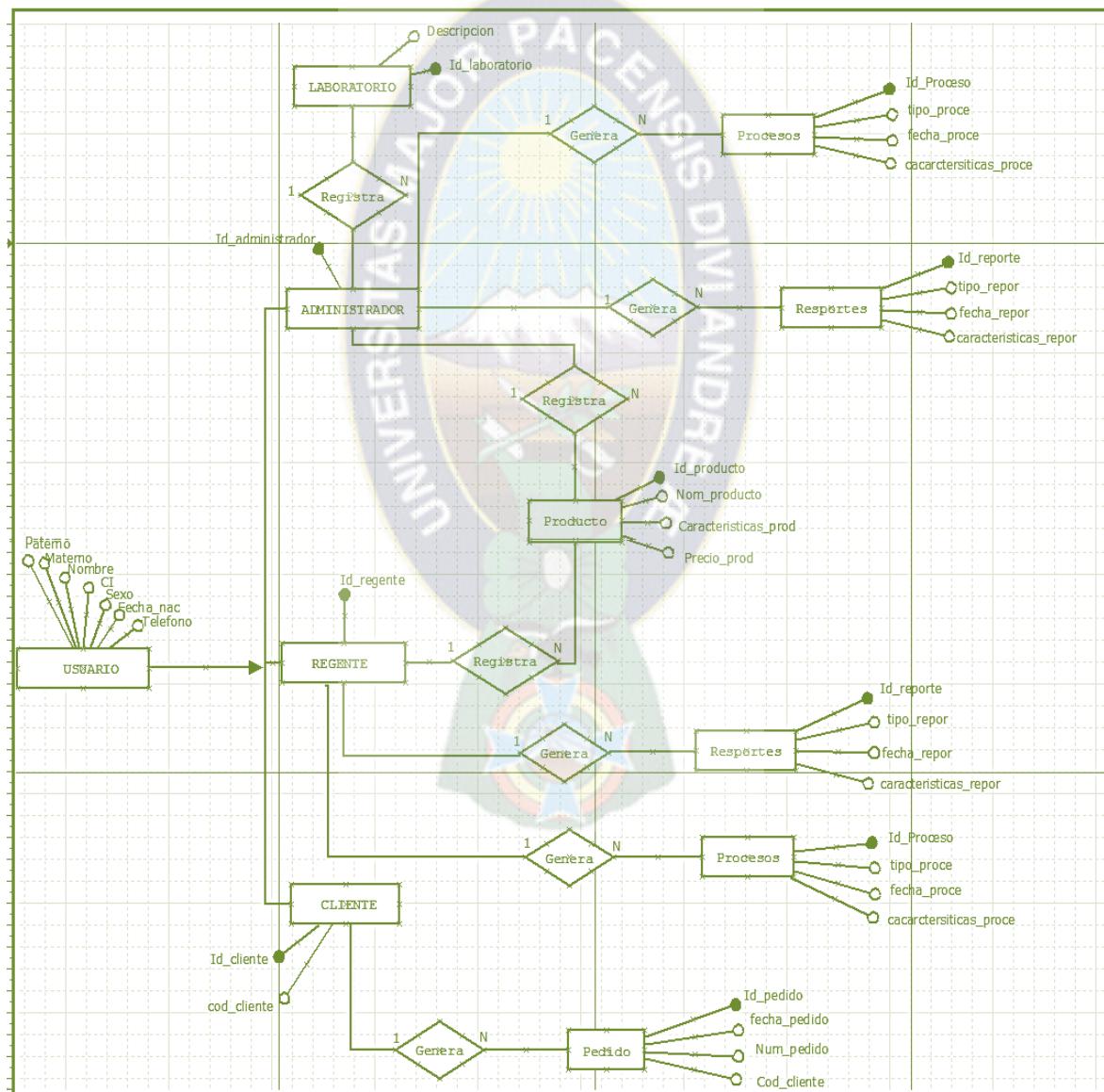
Fuente: [Elaboración propia]

ID	PRIMERA ITERACION TAREAS	SPRINT 1	INICIO Días	DURACION	
					estado
1	Realizar el diseño de diagrama de entidad-relación	Diseño	2		Terminado
2	Realizar el diseño de base de datos	Diseño	2		Terminado
3	Construir la base datos en el gestor de base datos	Desarrollo	5		Terminado

c) **Trabajo y desarrollo del Sprint I** En este proyecto se tiene estructura de datos complejas, por el cual es necesario crear una interfaz con una base de datos. El diagrama Entidad - Relación representa en su mayoría de todos los datos que se manejan en el sistema.

### 3.5.1. Diseño conceptual

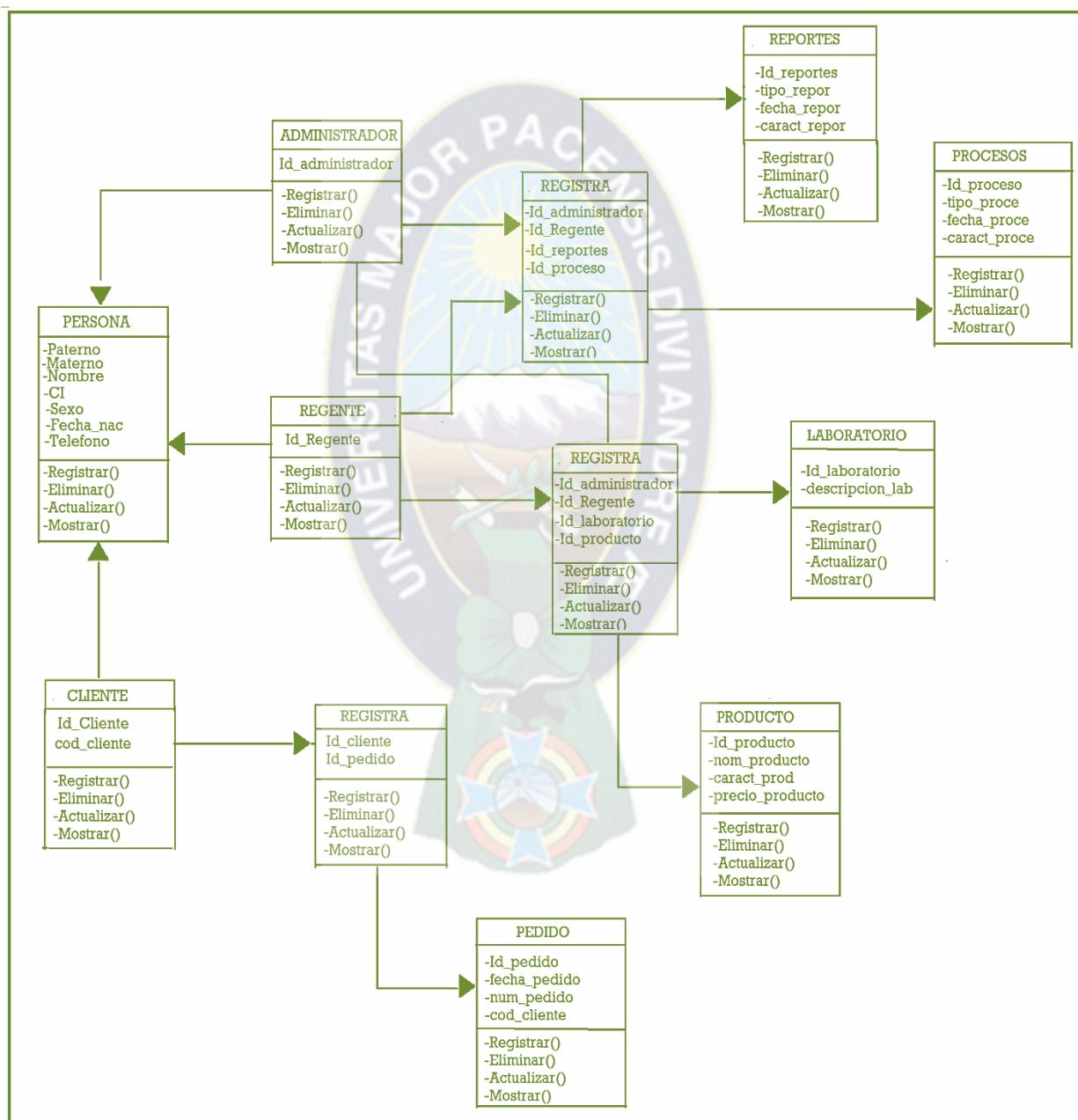
En la figura 3.23 se muestra el diseño Entidad –Relación de la base de datos



**Figura 3.23** Diagrama Entidad - Relación  
Fuente: [Elaboración propia]

Para el diseño conceptual se representan los objetos que manipularán el sistema, las operaciones (métodos) que se aplicarán a los objetos, las relaciones algunas de ellas jerárquicas entre los objetos.

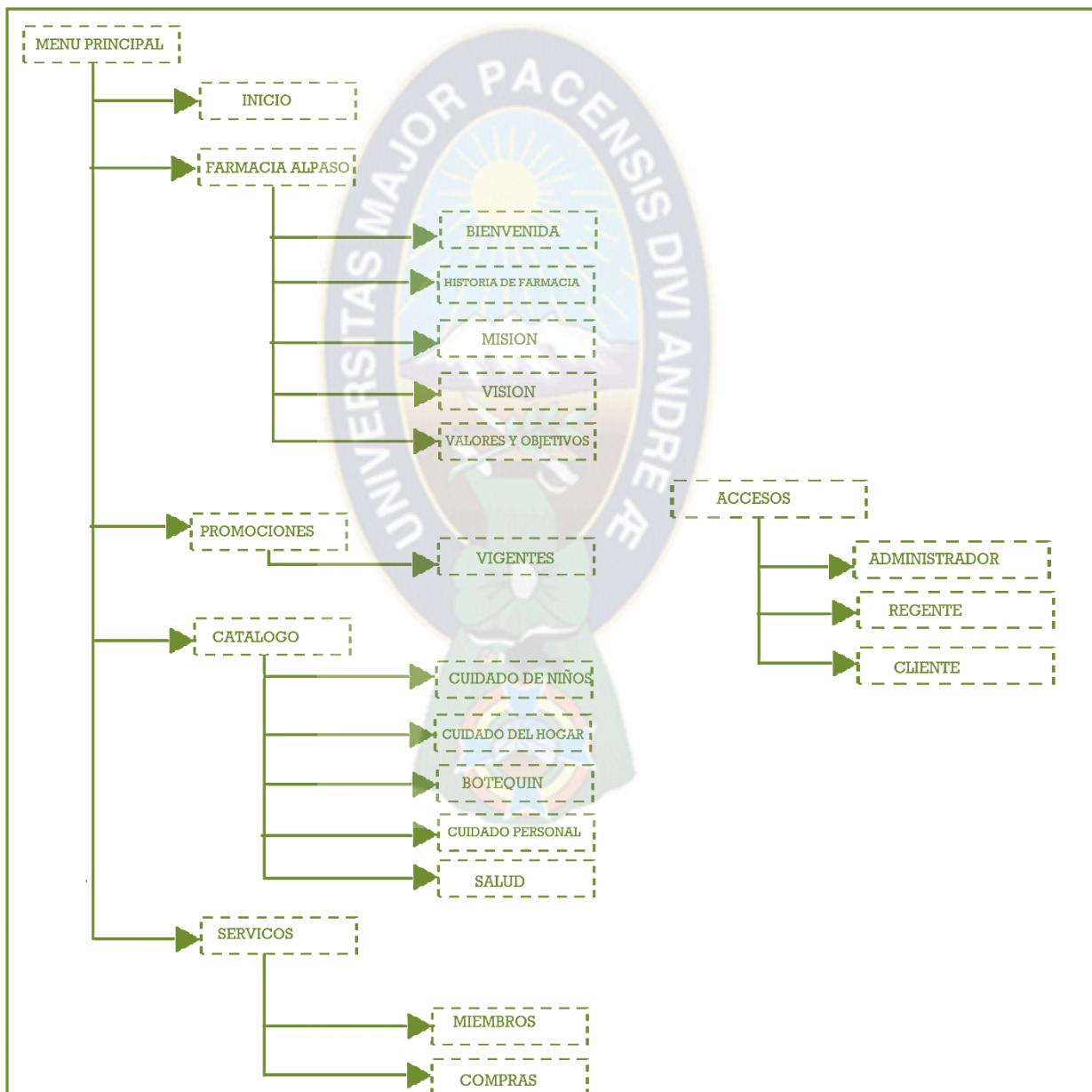
En la figura 3.24 se muestra el Diagrama de clases del sistema.



**Figura 3.24** Diagrama de clases del sistema  
**Fuente:** [Elaboración propia]

### 3.5.2. Diseño navegacional

En el diseño navegacional se presenta la estructura de navegación de la página principal del sitio web, en el cual muestra todas las estructuras de acceso que las componen, siendo así el menu y los accesos para clientes, regentes, y el administrador. En la figura 3.25 se puede observar el diseño navegacional.



**Figura 3.25** Diseño navegacional  
Fuente: [Elaboración propia]

### 3.5.3. Diseño abstracto de la interfaz (Sprint II)

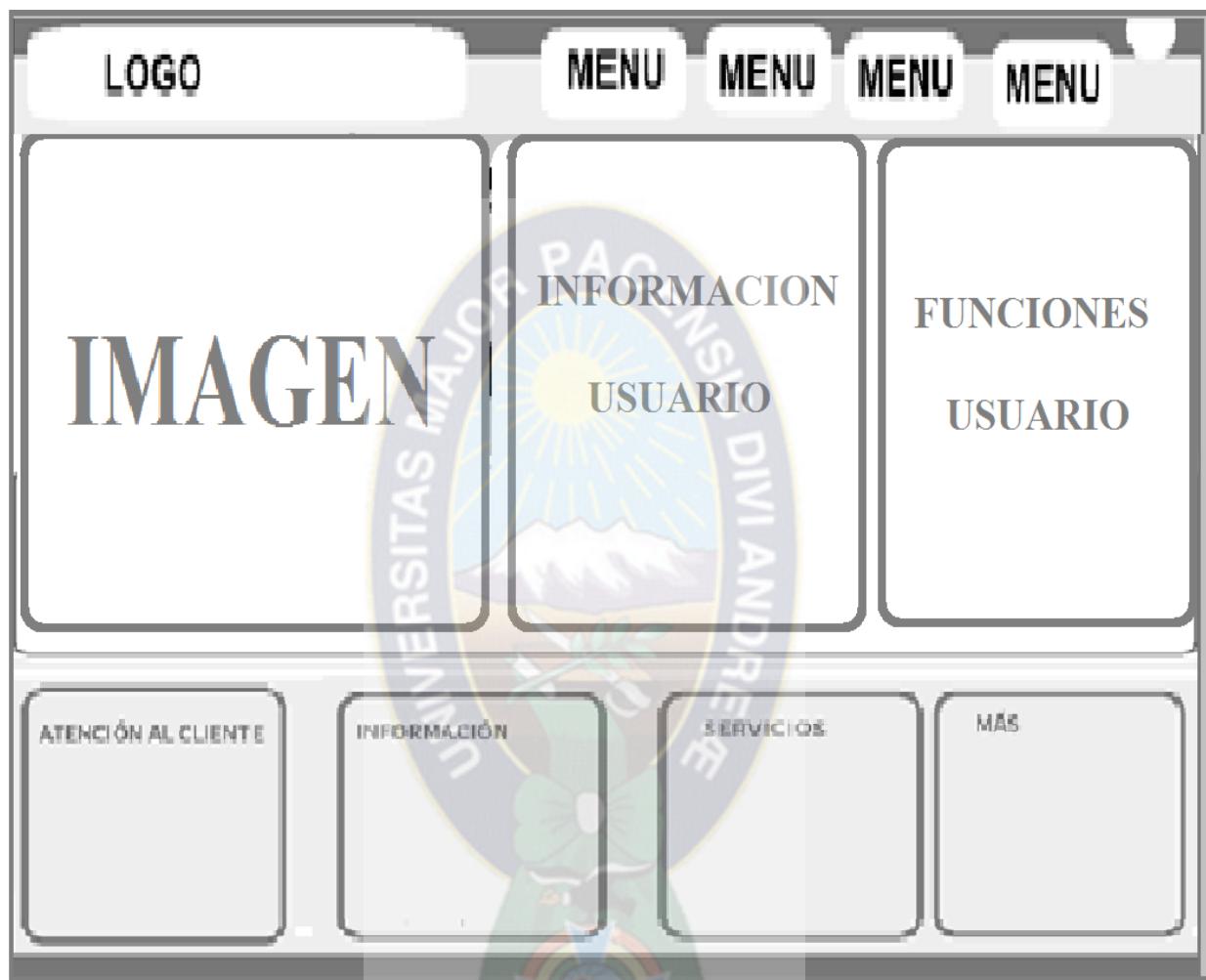
Para representar la interfaz del usuario, se utiliza el modelo de vista de datos abstracta (ADVs), donde se especifica la organización de todos los elementos que componen el sistema.

En la figura 3.26 se muestra la página principal del sistema que contiene un encabezado, menú de navegación, barra de navegación con la identificación de usuario junto a la presentación de imágenes o Slideshow, una barra de promociones de productos y por último en pies de página.



**Figura 3.26** Diseño página principal del sistema  
**Fuente:** [Elaboración propia]

En la figura 3.27 presenta el modelo de la página al momento de ingresar al sistema, utilizando la cuenta administrador o regente.



**Figura 3.27** Acceso de usuario  
Fuente: [Elaboración propia]

En la figura 3.28 presenta una página modal donde un nuevo cliente puede registrarse, solamente llenando el formulario. Solo para usuarios de tipo clientes.



**Figura 3.28** Registro de usuarios/clientes

Fuente: [Elaboración propia]

En la figura 3.29 presenta el modelo de la página al momento de ingresar al sistema, utilizando la cuenta cliente.



**Figura 3.29** Interfaz de usuarios/clientes

Fuente: [Elaboración propia]

En la figura 3.30 se puede ver el panel del formulario de registro de nuevos productos, al momento de seleccionar la opción de adicionar.

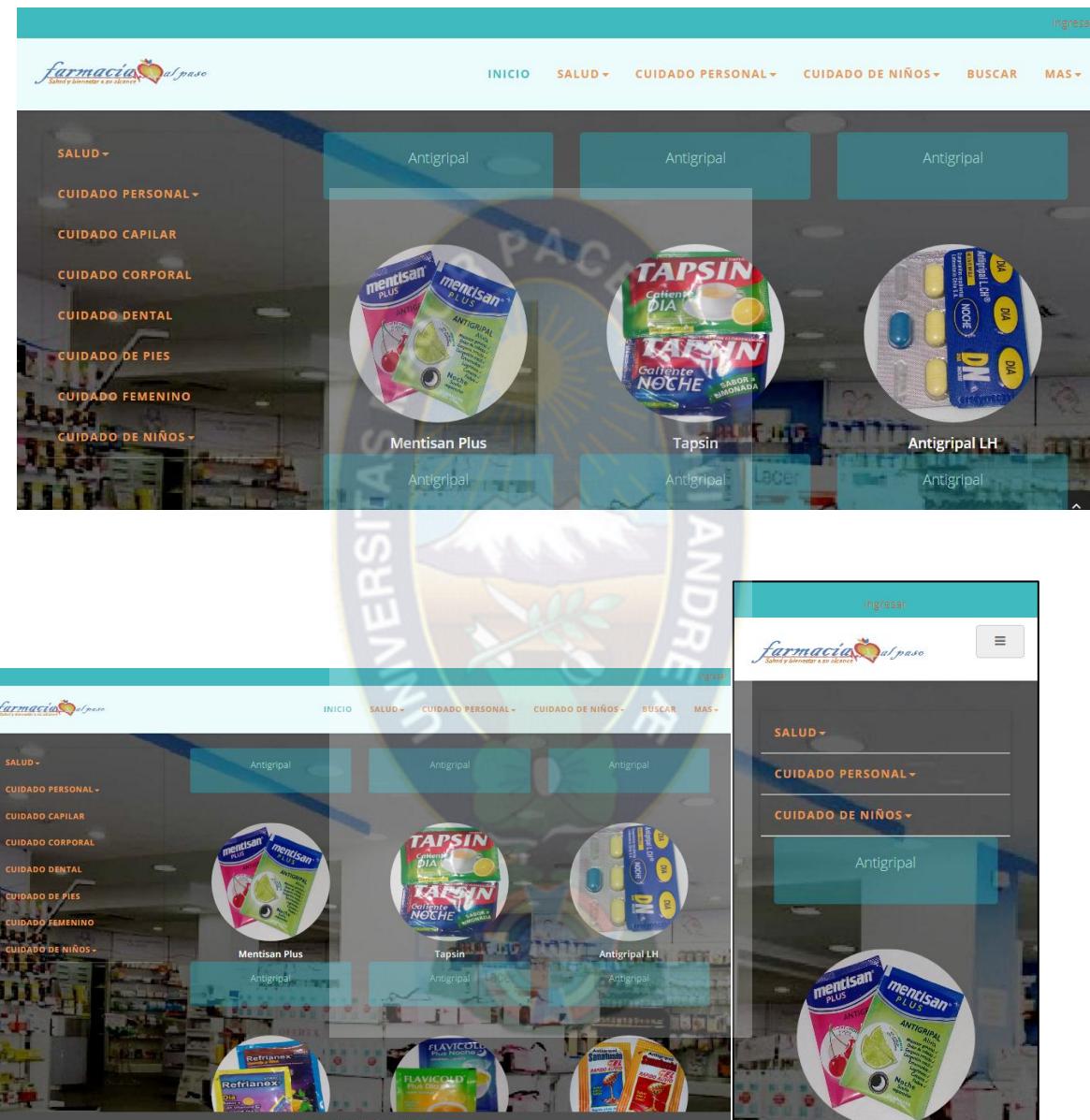
The figure shows a user interface for adding new products. At the top, there is a header with a logo on the left and four 'MENU' buttons. The main title 'FORMULARIO DE ADICIÓN DE PRODUCTOS' is displayed prominently in the center. Below the title is a large rectangular input field. Underneath this input field are four smaller rectangular input fields arranged vertically. At the bottom of the page, there is a footer with four buttons labeled 'ATENCIÓN AL CLIENTE', 'INFORMACIÓN', 'SERVICIOS', and 'MÁS'. A watermark of a university crest is visible in the background.

**Figura 3.30** Interfaz registro de productos  
Fuente: [Elaboración propia]

#### 3.5.4. Implementación

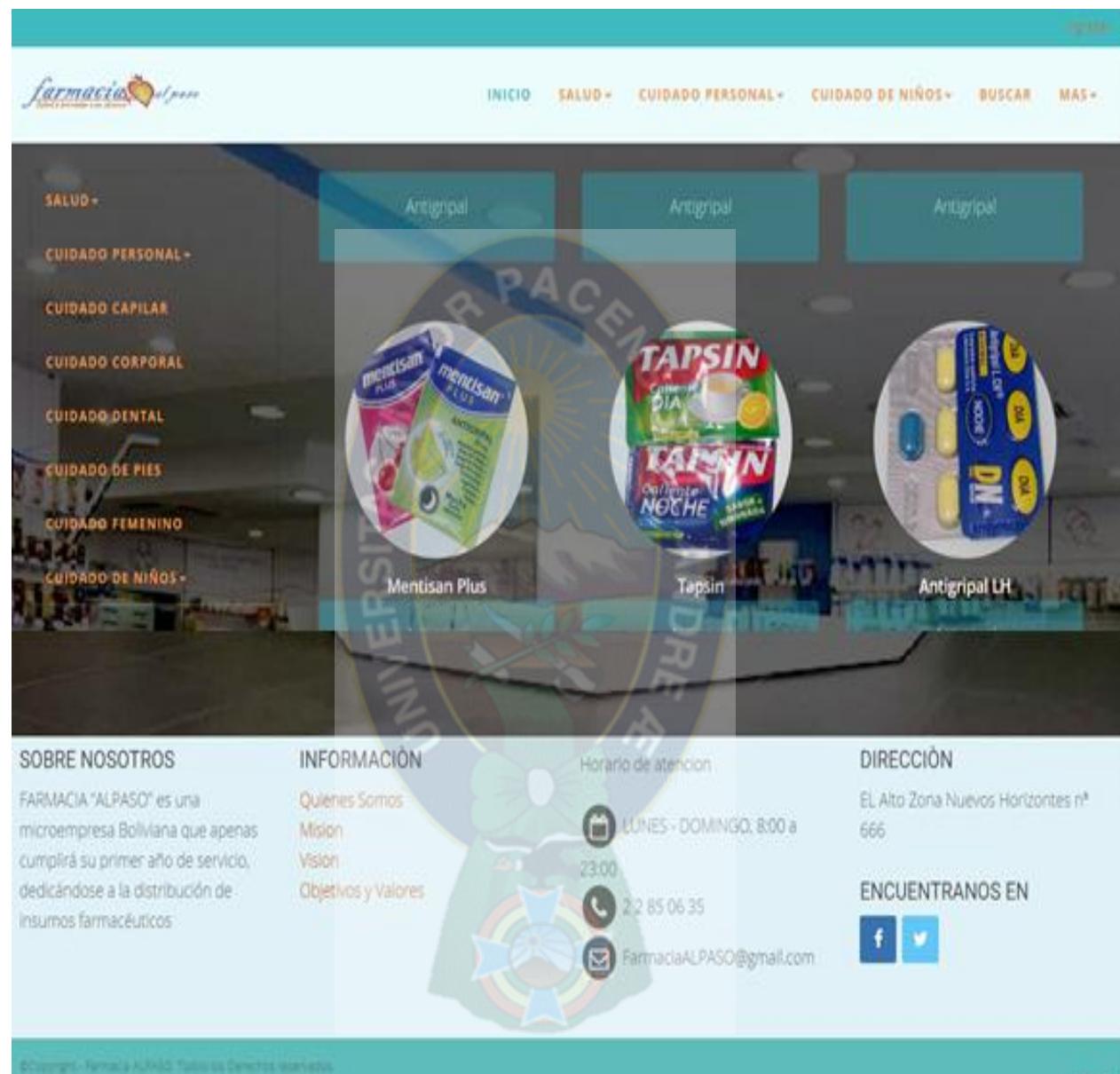
En la implementación del sistema, se muestra la interfaz del sitio web, desarrollada específicamente para la optimización en el control de datos de la farmacia al paso,

En la figura 3.31 muestra la adaptabilidad del sistema de seguimiento para tres distintos dispositivos.



**Figura 3.31** Interfaz de usuario del sitio web  
Fuente: [Elaboración propia]

La interfaz de la página principal del sitio web, utilizando en computadora de escritorio se puede ver en la figura 3,32



**Figura 3.32** Página de inicio del sistema  
**Fuente:** [Elaboración propia]

La figura 3.33 muestra la interfaz que carga después de seleccionar la opción de registrarse para el cliente,

**Figura 3.33** Interfaz de usuario Registro de cliente  
**Fuente:** [Elaboración propia]

En la figura 3.34 se puede observar el tipo de usuario administrador y las operaciones que el mismo puede realizar

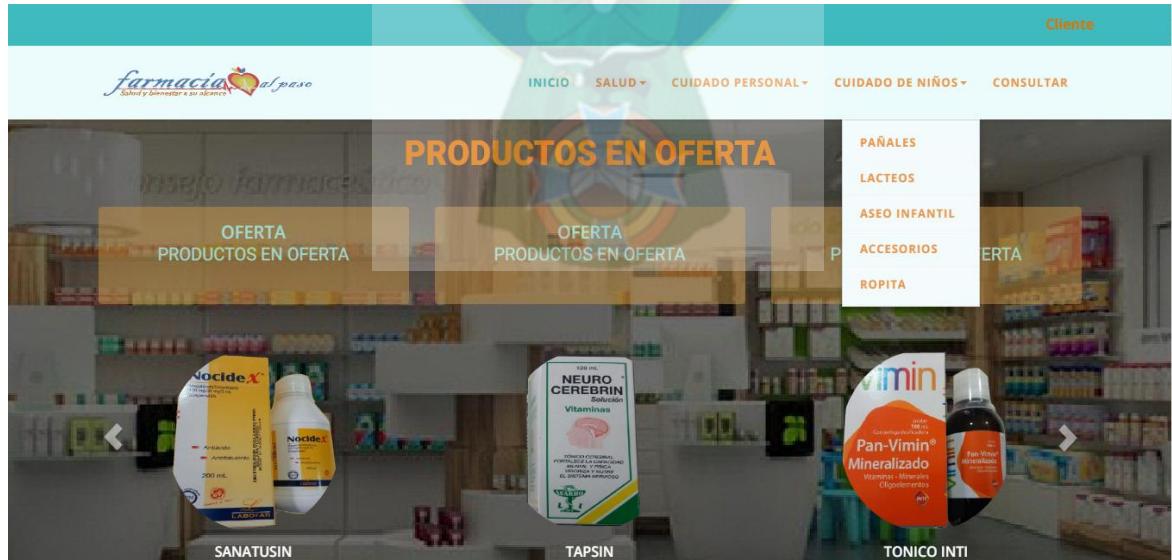
**Figura 3.34** Interfaz de usuario cuenta administrador  
**Fuente:** [Elaboración propia]

En la figura 3.35 se puede observar el tipo de usuario REGENTE y las operaciones que el mismo puede realizar.



**Figura 3.35** Interfaz de usuario cuenta administrador  
**Fuente:** [Elaboración propia]

En la figura 3.36 se puede observar el tipo de usuario CLIENTE y las operaciones que el mismo puede realizar.



**Figura 3.36** Interfaz de usuario cuenta cliente  
**Fuente:** [Elaboración propia]

En la figura 3.37 se puede observar el formulario de agregación de nuevos productos esto es realizado únicamente por el administrador

The screenshot shows a web-based application for managing a pharmacy. At the top, there's a navigation bar with 'Adicionar' on the left and 'Administrador' on the right. Below the bar, the logo 'farmacia al paso' is visible. On the right side of the header, there's a link labeled 'INICIO'. The main content area is titled 'ADICIONE PRODUCTO'. It contains several input fields grouped into pairs: 'CODIGO' and 'CANTIDAD', 'NOMBRE GENERICO' and 'PRECIO UNITARIO COMPRA', 'NOMBRE' and 'PRECIO UNITARIO VENTA', and 'LABORATORIO' and 'Laboratorio'. Each pair consists of a text input field followed by a dropdown menu. At the bottom center of the form is a blue button labeled 'ADICIONAR PRODUCTO'.

**Figura 3.37** formulario de agregación de productos  
Fuente: [Elaboración propia]

## CAPITULO IV

### ANALISIS DE COSTOS, CALIDAD DE SOFTWARE

#### 4.4. Introducción

Es importante tener la planificación o estimación del costo de proyecto para ello se utilizó el método COCOMO permitiendo realizar las estimaciones en función del tamaño de software y un conjunto de factores de coste. Al final se encuentra el costo total de desarrollo de sistema. También se realizaran pruebas de caja blanca y baja negra, la caja blanca permite garantizar que los elementos del sistema se hayan integrado de manera adecuada, la de caja negra permite verificar que los procesos funcionen y que se puedan buscar los errores. Y para la prueba de calidad de software se utilizara la metodología Web-SiteQEM el cual sirve para la evaluación de calidad de sitios web y la aceptabilidad del sistema.

#### 4.5. Análisis de costos

##### 4.5.1. Calculo de Punto Función no ajustado

**Tabla 4.1** Puntos función sin ajustar  
**Fuente:** Elaboración Propia

TIPO DE ELEMENTO	DIFICULTAD	PESO	CANTIDAD	TOTAL PUNTOS	TOTAL ELEMENTOS
ENTRADAS	Simple	3	10	30	
	Media	4	2	12	
	Compleja	6	1	6	
	Total puntos de función entrada				
SALIDAS	Simple	4	1	4	
	Media	5	2	10	
	Compleja	7	1	7	
	Total puntos de función salida				

CONSULTAS	Simple	3	2	6
	Media	4	6	24
	Compleja	6	5	30
	Total puntos de función consulta			60
ARCHIVOS	Simple	7	2	14
	Media	10	4	40
	Compleja	15	5	75
	Total puntos de función archivo			129
<b>TOTAL PUNTOS DE FUNCION NO AJUSTADOS (PFNA)</b>		<b>258</b>		

#### 4.5.2. Calculo de los puntos función ajustados

Para calcular los puntos de función se toman en cuenta factores de complejidad, es necesario evaluar 14 aspectos con una escala del 0-5, donde 0 es no influencia, 1 es incidental, 2 es moderado, 3 es medio, 4 es significativo y 5 es esencial.

**Tabla 4.2 Factores de complejidad**  
**Fuente:** Elaboración Propia

CARACTERISTICA	PESO
Comunicación de datos	4
Procesamiento distribuido de datos	5
Rendimiento	4
Configuraciones fuertemente utilizadas	3
Frecuencia de transacciones	3
Entrada de datos on-line	3
Eficiencia del usuario final	5
Actualizaciones on-line	0
Procesamiento complejo	4
Reusabilidad	3

Facilidad de instalación	3
Facilidad de operación	2
Instalación de distintos lugares	0
Facilidad de cambio	2
<b>TOTAL</b>	<b>41</b>

El factor de ajuste (**FA**) se obtiene calculando la siguiente ecuación:

$$FA = 0.65 + 0.01 \sum F_i$$

Dónde:

$F_i$  Corresponde a los pesos asignados.

Reemplazando en la ecuación se tiene:

$$FA = 0.65 + 0.01 * 41$$

$$FA = 1.06$$

Los puntos de función ajustados (PFA) se obtienen calculando la siguiente ecuación:

$$PFA = PFNA * FA$$

Reemplazando los valores en la ecuación se tiene

$$PFA = 258 * 1.06$$

$$PFA = 273.48$$

La tabla 4.3 muestra la equivalencia de líneas de código y puntos de función

**Tabla 4.3** Líneas de código por cada punto función  
**Fuente:** Elaboración Propia

LENQUAJE	LCD/PF
<b>Access</b>	38
<b>Ada 95</b>	49
<b>AI Shell</b>	49
<b>APL</b>	32
<b>Assembly - Basic</b>	320
<b>Basic - ANSI</b>	64
<b>Basic – Compilet</b>	91
<b>Basic - Visual</b>	32
<b>C</b>	128
<b>C++</b>	55
<b>Cobol (ANSI85)</b>	91
<b>Base de datos</b>	40
<b>4ta generación de lenguaje</b>	20
<b>Nivelar lenguaje</b>	64
<b>Jovial</b>	107
<b>Machine code</b>	640
<b>Modula 2</b>	80
<b>PHP</b>	29
<b>PERL</b>	27
<b>Prolog</b>	64
<b>Query -default</b>	13
<b>Report Generator</b>	80
<b>Second Generation language</b>	107
<b>Spreadsheeet</b>	46
<b>Third Generation Language</b>	80
<b>USR_5</b>	1
<b>Visual basic 5.0</b>	29

Los puntos función convertida a miles de líneas de código (KSLOC) se obtienen calculando la siguiente ecuación.

$$KSLOC = \frac{PF * LLP}{1000}$$

Dónde:

*LLP* : Líneas de código requeridos por putos función en un lenguaje (PHP , LLP =29).

Reemplazando en la ecuación de tiene:

$$KSLOC = \frac{273.48 * 29}{1000}$$

$$KSLOC = 7.93092$$

#### 4.5.3. Estimación del esfuerzo

Para la estimación del esfuerzo nominal  $PN_{Nominal}$  se utiliza la siguiente ecuación:

$$PM_{Nominal} = A * SIZE^B$$

Dónde:

**A:** Constante de calibración, su valor es 2.45

**SIZE** (Tamaño) = KSLOC

**B:** Factor de ahorro o gasto relativo de escala.

$$B = 0.91 + 0.01 * \sum W_i$$

Dónde:

**$W_i$**  Es el factor de escala

En la tabla 4.4 se puede observar los valores de los factores de escala.

**Tabla 4.4** Valores de los factores de escala  
**Fuente:** Elaboración Propia

ACTORES DE ESCALA	MUY BAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
<b>PREC</b>	6.20	4.96	3.72	2.48	1.24	0
<b>FLEX</b>	5.07	4.05	3.04	2.03	1.01	0
<b>RESL</b>	7.07	5.65	4.24	2.83	1.41	0
<b>TEAM</b>	5.48	4.38	3.29	2.19	1.10	0
<b>PMAT</b>	7.80	6.24	4.68	3.12	1.56	0

En la tabla 4.5 se muestra los valores obtenidos

**Tabla 4.5** Valores de los factores de escala  
**Fuente:** [Elaboración Propia]

FACTORES DE ESCALA	NIVEL	VALOR
<b>PREC</b>	Bajo	4.96
<b>FLEX</b>	Alto	2.03
<b>RESL</b>	Nominal	4.24
<b>TEAM</b>	Alto	2.19
<b>PMAT</b>	Nominal	4.68
Resultado		18.1

El valor de  $\sum W_i = 18.1$ , se reemplaza en la siguiente ecuación:

$$B = 0.91 + 0.01 * \sum W_i$$

$$B = 0.91 + 0.01 * 18.1$$

$$B = 1.091$$

Con el valor obtenido, calcularemos el valor del esfuerzo nominal.

$$PM_{Nominal} = A * SIZE^B$$

$$PM_{Nominal} = 2.45 * 7.93092^{1.091}$$

$$PM_{Nominal} = 23.46001$$

Para el esfuerzo ajustado  $PM_{ajustado}$  en Personas/Mes se utiliza la siguiente ecuación;

$$PM_{ajustado} = PM_{Nominal} * \prod EM_i$$

Dónde:

$EM_i$  Son los valores que toman los drivers de coste, los cuales serán tomados de la siguiente tabla:

**Tabla 4.6** Factores de ajuste  
Fuente: [Elaboración Propia]

DRIVERS	EXTRA BAJO	MUY BAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
RCPX	0.73	0.81	0.98	1	1.30	1.74	2.38
RUSE	-	-	0.95	1	1.07	1.15	1.24
PDIF	-	-	0.87	1	1.29	1.81	2.61

<b>PERS</b>	2.12	1.62	1.26	1	0.83	0.63	0.50
<b>PREX</b>	1.59	1.33	1.12	1	0.87	0.71	0.62
<b>FCIL</b>	1.43	1.30	1.10	1	0.87	0.73	0.62
<b>SCED</b>	-	1.43	1.14	1	1	1	-

En la tabla 4.7 se pueden ver los valores obtenidos al analizar el sistema

**Tabla 4.7** Valores de Factores de ajuste  
**Fuente:** [Elaboración Propia]

FACTORES DE ESCALA	NIVEL	VALOR
<b>RCPX</b>	Alto	1.30
<b>RUSE</b>	Bajo	0.95
<b>PDIF</b>	Alto	1.29
<b>PERS</b>	Nominal	1
<b>PREX</b>	Nominal	1
<b>FCIL</b>	Alto	0.87
<b>SCED</b>	Nominal	1
Resultado		1.39

Reemplazando en la ecuación se tiene

$$PM_{ajustado} = PM_{Nominal} * \prod EM_i$$

$$PM_{ajustado} = 23.46001 * 1.39$$

$$PM_{ajustado} = 32.61$$

$$PM_{ajustado} = 33 \text{ Persona /mes}$$

#### 4.5.4. Estimación del cronograma

El tiempo de desarrollo (**TDEV**) se calcula de la siguiente ecuación:

$$TDEV = [3.67 * PM^{(0.28+0.2*(B-1.01))}] * [\frac{SCED\%}{100}]$$

Dónde:

**SCED%** : Es el porcentaje en que se calcula podrá aumentar o disminuir el calendario.

(**SCED% = 100**)

Reemplazando en la ecuación se tiene:

$$TDEV = [3.67 * 32.61^{(0.28+0.2*(1.091-1.01))}] * [\frac{100}{100}]$$

$$TDEV = 10.30$$

$$TDEV = 11 \text{ meses}$$

La cantidad de personal (NP) se obtiene calculando la siguiente ecuación:

$$NP = \frac{PM_{ajustado}}{TDEV}$$

Reemplazando en la ecuación se tiene:

$$NP = \frac{33}{11}$$

$$NP = 3 \text{ personas}$$

#### 4.5.5. Costo total del sistema

El costo total de desarrollo (CD) del sistema se calcula de la siguiente ecuación:

$$CD = TDEV * PS * NP$$

Donde PS es el sueldo promedio de un programador

$$CD = 11 * 2088 * 3$$

$$CD = 68904 \text{ Bs}$$

El proyecto tendrá un costo de Bs 68 904 , será desarrollada aproximadamente en 11 meses y será necesario la participación de 3 personas , que tendrán el sueldo promedio de Bs 2088.

## 4.6. Calidad de software

Para evaluar la calidad del sistema web se proporciona la información adecuada sobre los datos necesarios referente a la calidad del producto, permitiendo el cumplimiento de los objetivos del proyecto. Para poder valorar la calidad del sistema web se aplicó la metodología Web-Site QEM de evaluación de sitios web.

### 4.6.1. Evaluación elemental

A continuación se muestran los tipos de criterio elemental utilizados:

**Criterio Binario, Discreto y Absoluto:** No Disponible 0≈0 y Disponible 1≈100

**Criterio Multi-Nivel, Discreto y absoluto:** No Disponible 0≈0, Parcialmente disponible 1≈60 y totalmente disponible 2≈100.

**Criterio de preferencia directa, continuo y Absoluto:** Sujeto a la subjetividad del evaluador.

**Criterio de variable normalizada, continuo y absoluto:** BL= número de X encontrados, TL= Número total de enlaces, Y = $100(BL*100)/TL)*10$ , donde Y < 0 entonces Y = 0, además X =Enlaces Rotos o enlaces inválidos.

En la tabla 4.8 se pueden observar los resultados de preferencias elementales de Usabilidad

**Tabla 4.8** Resultados de preferencias elementales de usabilidad  
**Fuente:** [Elaboración Propia]

CODIGO	ATRIBUTOS SUBCARACTERISTICA	DEFINICION	CRITERIO	
			ELEMENTAL	IEi %
1.1.1.1	Mapa de sitio	Un mapa del sitio es una representación con componentes gráficos, que muestran la estructuras	Binario Discreto Absoluto	100
1.1.1.2	Tabla de contenido	Es la información que permite estructurar el contenido de todo	Multi-nivel Discreto	60

		el sitio web, permitiendo Absoluto navegación directa.			
<b>1.1.2</b>	Mapa de imagen	Un mapa de imagen como Multi-nivel áreas etiquetables que proveen Discreto puntos de partida directa a Absoluto contenidos, o que permiten visitas guiadas.	0		
<b>1.2.1</b>	Calidad de ayuda	Se refiere a la información de apoyo orientada al usuario.	Multi-nivel Discreto Absoluto	0	
<b>1.2.2.1</b>	Ultima actualización global	Permite verificar 1 a última fecha de actualización del sistema	Binario Discreto Absoluto	0	
<b>1.2.2.1</b>	Última actualización restringida	Permite verificar la última fecha de actualización de los sub sitios o páginas del sistema	Binario Discreto Absoluto	0	
<b>1.2.3.1</b>	Directorio Email	Es un sitio donde se agrupan las direcciones electrónicas para el envío de correos	Binario Discreto Absoluto	100	
<b>1.2.3.2</b>	Directorio Te-Fax	Es un sitio donde se agrupan las direcciones de contactos vía telefónica	Multi-nivel Discreto Absoluto	60	
<b>1.2.4</b>	Retroalimentación	Mide si existe un intercambio de información de usuario al sistema o viceversa	Multi-nivel Discreto Absoluto	0	
<b>1.3.1</b>	Cohesividad al agrupar los objetos de control principal	Indica si los enlaces que posee el sistema, están agrupados adecuadamente.	Preferencia Directa Continuo Absoluto	100	
<b>1.3.2.1</b>	Permanencia de	Representa la permanencia	Binario	100	

	controles Directos	directa de controles del menu principal que permiten la navegación	Discreto	Absoluto
<b>1.3.3.1</b>	Uniformidad en el color de enlaces	Se refiere a todos los enlaces que tiene el sitio web	Binario Discreto	100 Absoluto
<b>1.3.3.2</b>	Uniformidad en el estilo global	Se refiere a todo el sitio que tiene uniformidad de colores y fuentes	Multi-nivel Discreto Absoluto	100

En la tabla 4.9 se pueden observar los resultados de preferencias elementales de Funcionalidad

**Tabla 4.9** Resultados de preferencias elementales de funcionalidad  
**Fuente:** [Elaboración Propia]

CODIGO	ATRIBUTO CARACTERISTICA	DEFINICION	CRITERIO ELEMENTAL	IEi %
<b>2.1.1.1</b>	Búsqueda restringida	Búsqueda de un elemento en particular	Binario Discreto Absoluto	100
<b>2.1.1.2</b>	Búsqueda global	Búsqueda general del sistema	Multi-nivel Discreto Absoluto	0
<b>2.1.2</b>	Mecanismo de recuperación	Se refiere al nivel de personalización y nivel de retroalimentación en la recuperación	Multi-nivel Discreto Absoluto	0
<b>2.2.1.1</b>	Orientación	Se refiere las pistas visuales que indica con precisión donde se encuentra el usuario al navegar en el sitio	Preferencia Directa Continuo Absoluto	70
<b>2.2.1.2</b>	Promedio de enlaces por pagina	Da una idea sobre la cantidad de puntos de partida en una página promedio que indica	Multi-nivel Discreto Absoluto	60

		como una página esta interconectada		
<b>2.2.2.1</b>	Permanencia y estabilidad en la presentación de los controles contextuales	Es la permanencia y estabilidad de los controles de navegación de un sub sitio	Multi-nivel Discreto Absoluto	100
<b>2.2.2.2</b>	Nivel de desplazamiento	Nivel de desplazamiento que el usuario debe realizar para ajustar la interface	Multi-nivel Discreto Absoluto	60
<b>2.3.1.1</b>	Información de la farmacia	Este atributo indica si están presentes los datos relevantes de la farmacia	Binario Discreto Absoluto	100
<b>2.3.1.2</b>	Información de registro	Se refiere si la información se encuentra presente o ausente, en cuanto al registro de nuevos productos	Binario Discreto Absoluto	100
<b>2.3.1.3</b>	Información de cliente	Se refiere si la información se encuentra presente o ausente en cuanto a los datos de los clientes	Binario Discreto Absoluto	100
<b>2.3.1.4</b>	Información de reportes	Se refiere si la informaciones encuentra presente o ausente en cuanto a la generación de informes diarios	Binario Discreto Absoluto	100
<b>2.3.2.2</b>	Servicio de páginas web	Se refiere si la información se encuentra presente o ausente, en cuanto a la aprobación del curso	Binario Discreto Absoluto	100
<b>2.3.2.4</b>	Servicio FTP	Se refiere si se encuentra presente o ausente, en cuanto a la página web.	Binario Discreto Absoluto	100

En la tabla 4.10 se pueden observar los resultados de preferencias elementales de confiabilidad

**Tabla 4.10** Resultados de preferencias elementales de confiabilidad

**Fuente:** [Elaboración Propia]

CODIGO	ATRIBUTO	DEFINICION	CRITERIO	ELEMENTAL	IEi%
<b>3.1.1.1</b>	Enlaces rotos	Mide la cantidad de enlaces que están rotos (ausentes o pendiente)	Variable Normalizada Continuo Absoluto		100
<b>3.1.1.2</b>	Enlaces inválidos	Mide los enlaces que no son validos	Variable Normalizada Continuo Absoluto		100
<b>3.1.1.3</b>	Enlaces implementados	Mide los enlaces implementados en el sitio web	Multi-nivel Discreto Absoluto		100
<b>3.1.2.1</b>	Deficiencias o cualidades ausentes debido a diferentes navegadores	Mide si existe una mala presentación del sitio debido a los navegadores.	Preferencia Directa Continuo Absoluto		60
<b>3.1.2.2</b>	Deficiencias o resultados inesperados independientes de browsers	Mide si existe una mala presentación del sitio web que afecte al navegador	Preferencia Directa Continuo Absoluto		60
<b>3.1.2.3</b>	Nodos destinos (inesperadamente) en construcción	Mide si existen paginas en construcción (proceso de implementación)	Preferencia Directa Continuo Absoluto		100
<b>3.1.2.4</b>	Nodos muertos (sin enlaces de retorno)	Mide si existe páginas web sin funcionamiento	Preferencia Directa Continuo Absoluto		100

En la tabla 4.11 se pueden observar los resultados de preferencias elementales de Eficiencia

**Tabla 4.11** Resultados de preferencias elementales de eficiencia

**Fuente:** [Elaboración Propia]

CODIGO	ATRIBUTO	DEFINICION	CRITERIO ELEMENTAL	IEi%
4.1.1	Páginas de acceso rápido	Mide el tamaño de todas las páginas (estáticas) considerando todos sus componentes gráficos, tabulares o textuales	Multi-nivel Discreto Absoluto	60
4.2.1.1.	Soporte a versión solo texto	Representa la accesibilidad a la información que está en las páginas más relevantes que la página sea editada en una versión de solo texto	Multi-nivel Discreto Absoluto	60
4.2.1.2	Legibilidad al desactivar la propiedad	Se presenta l desactivar la propiedad imagen del navegador, la página debe ser entendible	Multi-nivel Discreto Absoluto	100

#### 4.6.2. Evaluación global

para esta evaluación , se emplea el Modelo de agregación lógica de preferencias (LSP) que utiliza la función denominada media potencia pesad que sirve para calcular la calidad de sub características, característica, características de alto nivel y finalmente la calidad global del sistema. La función es la siguiente:

$$IG(r) = (P_1 * IE_1^r + P_2 * IE_2^r + \dots + P_m * IE_m^r)^{1/r}$$

Dónde:

El valor de  $r$  esta entre  $-\alpha \leq r \leq +\alpha$

$P_i$  Es el peso que está sujeto a los valores siguientes:

$$(P_1 + P_2 + \dots + P_m) = 1$$

$$P_i > 0$$

$$i = 1 \dots m$$

$IE_i$  Es el indicador elemental que esta entre:

$$0 \leq IE_i \leq 1$$

La preferencia resultante  $IG_i$  tiene un valor entre:

$$\min(IE_1, IE_2, \dots, IE_m) \leq IG_i \leq \max(IE_1, IE_2, \dots, IE_m)$$

En la tabla 4.12 se muestra los resultados obtenidos para la característica de **Usabilidad**:

**Tabla 4.12** Resultados global de Usabilidad  
Fuente: [Elaboración Propia]

CARACTERISTICAS Y SUBCARACTERISTICAS	PESO	VALORES
<b>1.USABILIDAD</b>		<b>81.50</b>
<b>1.1 COMPRENSIBILIDAD GLOBAL DEL SITIO</b>	<b>0.3</b>	<b>73.75</b>
<b>1.1.1 Esquema de organización global</b>	0.8	82.46
1.1.1.1 Mapa de sitio	0.5	100
1.1.1.2 Tabla de contenido	0.5	60
<b>1.1.2 Mapa de imagen</b>	0.2	0
<b>1.2 MECANISMO DE AYUDA Y RETROALIMENTACIÓN EN LÍNEA</b>	<b>0.2</b>	<b>71.48</b>

<b>1.2.1 Calidad de ayuda</b>	0.2	0
<b>1.2.2 Indicador de Ultima actualización</b>	0.2	0
1.2.2.1 Última actualización global	0.5	0
1.2.2.1 Última actualización restringida	0.5	0
<b>1.2.3 Directorio de Direcciones</b>	0.4	89.89
1.2.3.1 Directorio Email	0.7	100
1.2.3.2 Directorio Te-Fax	0.3	60
<b>1.2.4 Retroalimentación</b>	0.2	0
<b>1.3 ASPECTOS DE INTERFACES Y ESTÉTICOS</b>	<b>0.4</b>	<b>93.07</b>
<b>1.3.1 Cohesividad al agrupar los objetos de control principal</b>	0.3	100
<b>1.3.2 Permanecía y estabilidad en la presentación de los controles principales</b>	0.3	70.71
1.3.2.1 Permanencia de controles Directos	0.5	100
1.3.2.2 Permanencia de controles indirectos	0.5	0
<b>1.3.3 Aspectos de estilo</b>	0.4	100
1.3.3.1 Uniformidad en el color de enlaces	0.5	100
1.3.3.2 Uniformidad en el estilo global	0.5	100
<b>1.4 MISCELÁNEAS</b>	<b>0.1</b>	<b>0</b>
<b>1.4.1 Soporte a lenguaje extranjero</b>	0.4	0
<b>1.4.2 Indicador de Resolución de pantalla</b>	0.6	0

En la tabla 4.13 se muestra los resultados obtenidos para la característica de **funcionalidad**:

**Tabla 4.134** Resultados global de Funcionalidad  
**Fuente:** [Elaboración Propia]

CARACTERISTICAS Y SUB CARACTERISTICA	PESO	VALOR
<b>2. FUNCIONALIDAD</b>		<b>79.98</b>
<b>2.1 ASPECTOS DE BÚSQUEDA Y RECUPERACIÓN</b>	<b>0.3</b>	<b>64.81</b>
<b>2.1.1 Mecanismo de Búsqueda en el sitio web</b>	0.7	77.46

2.1.1.1 Búsqueda restringida	0.6	100
2.1.1.2 Búsqueda global	0.4	0
<b>2.1.2 Mecanismo de recuperación</b>	0.3	0
<b>2.2 ASPECTOS DE NAVEGACIÓN Y EXPLORACIÓN</b>	<b>0.3</b>	<b>69.73</b>
<b>2.2.1 Navegabilidad</b>	0.4	67.16
2.2.1.1 Orientación	0.7	70
2.2.1.2 Promedio de enlaces por pagina	0.3	60
<b>2.2.2 Objetos de control Navegacional</b>	0.3	89.89
2.2.2.1 Permanencia y estabilidad en la presentación de los controles contextuales	0.7	100
2.2.2.2 Nivel de desplazamiento	0.3	60
<b>2.2.3 Predicción Navegacional</b>	0.3	0
<b>2.3 ASPECTOS DEL DOMINIO ORIENTADO A LA INSTITUCION</b>	<b>0.4</b>	<b>93.61</b>
<b>2.3.1 Relevancia de contenido</b>	0.7	97.40
2.3.1.1 Información de la farmacia	0.2	100
2.3.1.2 Información de registro	0.35	100
2.3.1.3 Información de cliente	0.15	100
2.3.1.4 Información de reportes	0.2	100
<b>2.3.2 Servicios On-line</b>	0.3	84.09
2.3.2.1 Información Aranceles	0.25	0
2.3.2.2 Servicio de páginas web	0.25	100
2.3.2.3 servicio de grupo de Noticia	0.25	0
2.3.2.4 Servicio FTP	0.25	100

En la tabla 4.14 se muestra los resultados obtenidos para la característica de **Confiabilidad**:

**Tabla 4.14** Resultados global de Confiabilidad

Fuente: [Elaboración Propia]

CARACTERISTICAS Y SUB CARACTERISTICAS	PESO	VALOR
<b>3. CONFIABILIDAD</b>		<b>93.02</b>
<b>3.1 NO DEFICIENCIA</b>		93.02
<b>3.1.1 Errores de enlaces</b>	0.7	100
3.1.1.1 Enlaces rotos	0.4	100
3.1.1.2 Enlaces inválidos	0.3	100
3.1.1.3 Enlaces implementados	0.3	100
<b>3.1.2 Errores o deficiencias Varias</b>	0.3	74.23
3.1.2.1 Deficiencias o cualidades ausentes debido a diferentes navegadores	0.4	60
3.1.2.2 Deficiencias o resultados inesperados independientes de browsers	0.4	60
3.1.2.3 Nodos destinos (inesperadamente) en construcción	0.1	100
3.1.2.4 Nodos muertos (sin enlaces de retorno)	0.1	100

En la tabla 4.15 se muestra los resultados obtenidos para la característica de **Eficiencia**:

**Tabla 4.15** Resultados global de Eficiencia

Fuente: [Elaboración Propia]

CARACTERISTICAS Y SUB CARACTERISTICAS	PESO	VALOR
<b>4 EFICIENCIA</b>		<b>64.94</b>
<b>4.1 PERFORMANCIA</b>	0.7	60
4.1.1 Páginas de acceso rápido	1	60
<b>4.2 ACCESIBILIDAD</b>	0.3	75.20
4.2.1 Accesibilidad de Informacion	0.7	89.89
4.2.1.1. Soporte a versión solo texto	0.3	60
4.2.1.2 Legibilidad al desactivar la propiedad	0.7	100
4.2.2 Accesibilidad de ventana	0.3	0

De acuerdo a la valoración de calidad global del sistema web, aplicando la metodología WEB-SITE QEM . los resultados globales de Usabilidad, Funcionalidad, Confiabilidad y Eficiencia se detallan en la tabla 4.16.

**Tabla 4.16** Resultados global

Fuente: [Elaboración Propia]

CARACTERISTICAS DE MAS ALTO NIVEL	PESO	VALOR
1 USABILIDAD	0.25	81.50
2 FUNCIONALIDAD	0.25	79.98
3 CONFIABILIDAD	0.25	93.02
4 EFICIENCIA	0.25	64.94
<b>CALIDAD GLOBAL</b>		<b>81.64%</b>

#### 4.6.3. Análisis y Resultados

De acuerdo a la metodología el nivel de aceptabilidad **Satisfactorio** esta en el rango de 60 % a 100%, un nivel de aceptabilidad **Marginal** está en el rango de 40% a 60%. Finalmente, un valor de preferencia **Insatisfactorio** esta entre el rango de 0% a 40%.

El resultado global obtenido para la usabilidad del sistema es de 81.50%, para funcionalidad del sistema es de 79.98, para la confiabilidad del sistema 93.02% y el resultado para la eficiencia del sistema se obtuvo un valor de 64.94, esto nos dice que estos valores se encuentran dentro del margen de aceptabilidad satisfactorio, porque se encuentran dentro del rango de 60% a 100%.

Observando la tabla 4.17 se puede decir que el resultado de calidad global es de 81.64% como este se encuentra dentro del rango de 60% a 100%, se puede afirmar que el sistema web de control y seguimiento de inventario tiene un nivel de aceptabilidad SATISFACTORIO.

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

Durante la implementación de este sistema web de seguimiento y control de inventario de la Farmacia Al paso, se cumplieron con los siguientes propósitos:

- Se logró implementar una base de datos donde se almacenaran toda la información de los productos de la farmacia.
- El sistema de seguimiento y control de inventarios mejoro la administración de los ingresos y egresos de la farmacia.
- Se incorporó el módulo de reportes diarios de las ventas, compras y sus demás movimientos.
- Se logró tener una base datos con información actualizada y abierta a nuevas modificaciones.
- El sistema mejoro el proceso de atención al cliente al optimizar las operaciones de búsqueda y venta de medicamentos.

A continuación se detallan los procesos logrados para los tipos de usuario.

El usuario **Administrador** podrá verificar si un producto tendrá que ser reemplazado ya que los insumos farmacéuticos cuentan con fecha de caducidad, también podrá realizar el registro de nuevos usuarios, podrá también ver los reportes diarios de los regentes, y actualizar la base de datos.

El usuario **Regente** podrá verificar si un producto deberá ser reemplazarlo y notificarlo al administrador para que lo reemplace en la base de datos, podrá realizar registros de nuevos usuarios solo de tipo cliente , generar reportes diarios de movimiento.

El usuario **Cliente** realizar una cotización de los productos que vaya a requerir y verificar si aún se tiene disponible, también realizar el canje de sus puntos acumulados por las compras realizadas, podrá acceder a las promociones que la farmacia realiza-

Es importante mencionar que el sistema web se implementó de forma responsive, adaptable para Tablet y teléfonos inteligentes.

## 5.2. Recomendaciones

Teniendo en cuenta que la tecnología se va desarrollando a paso agigantados y en la mayoría de los casos suplantando las tareas cotidianas

Se recomienda lo siguiente. Hacer el mantenimiento adecuado del sistema, esto con el fin de permitir mejorar en el sistema y así no presentar problemas.

Actualizar el sistema periódicamente, esto con el fin de mejorar las funcionalidades del sistema web de control y seguimiento.

Implementar el módulo de facturación de ventas para tener un libro contable dentro del sistema y mejorar la administración financiera de la farmacia.

Implementar el módulo de búsqueda de productos en diferentes sucursales , esto en un futuro, para cuando la farmacia vaya creciendo y sea una cadena farmacéutica.

## BIBLIOGRAFIA

Chávez, (2013), *Sistema Informático De inventario Y Facturación De Mercadería Con Entorno Web En La Imprenta Y Gráficas Chone*. Escuela Superior Politécnica Agro pecuaria De Manabí Manuel Félix López.

Arroyo, (2013). *Sistema de Control de Farmacia para un centro de salud de nivel I-3”*, Arroyo Olivera, José Carlos, 2013, Perú. Universidad Peruana de Ciencias Aplicadas, Facultad De Ingeniería, Carrera De Ingeniería De Software.

ITM (sf). *Metodologías para el desarrollo de aplicaciones Web*, ITM Instituto Tecnológico de Matehuala, Recuperado de <https://programacionwebisc.wordpress.com/2-5/metodologias-para-el-desarrollo-de-aplicaciones-web/>

Sommerville (2005), *Ingeniería De Software*, Séptima Edición Hoja 6. Recuperado de [http://zeus.inf.ucv.cl/~bcrawford/AULA\\_ICI\\_3242/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf](http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI_3242/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf)

DefABC (2010). *Aabc Ingenieria De Software*. Recuperado de <https://www.definicion nabc.com/tecnologia/ingenieria-de-software.php>

Letelier (2006). *Metodología agiles para el desarrollo de software*. Recuperado de [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)

Condef (2016). *Definición metodología de desarrollo de software*. RECUPERADO DE <http://conceptodefinicion.de/metodologia-de-desarrollo-de-software/>

UTP (2007). *Universidad Tecnológica de Pereira. del manifiesto ágil sus valores y principios*. Recuperado de <http://DialnetDelManifiestoAgilSusValoresYPrincipios>

TOP (2011). *Metodología de desarrollo Scrum*. Recuperado de <http://www.aplicandoscrum.com/scrum/>

SOFTENG, (2010) *metodologia scrum*. Recuperado de <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

Peralta (2003) *Metodología De Desarrollo Agil Scrum*, Adriana Peralta. Recuperado de <http://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>

Vila, (2007) .*Los Tres Pilares De Scrum*. Recuperado de <http://managementplaza.es/blog/los-tres-pilares-de-scrum/>

Trigas, (sf). *Metodología Scrum*. Recuperado de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Calzada, (2015) *Sistema Web De Gestión Documental Digital Y Administración De Archivos*. Recuperado de <http://repositorio.umsa.bo/bitstream/handle/123456789/8704/T.3044.pdf?sequence=1>

Rossainz, (s.f.). Mario Rossainz López, *Metodologías De Desarrollo De Aplicaciones Web*. Recuperado de [http://rossainz.cs.buap.mx/ingWeb/apuntes/IngWeb\\_Cap2.pdf](http://rossainz.cs.buap.mx/ingWeb/apuntes/IngWeb_Cap2.pdf)

Urbíeta, (2012), Mario Matías Urbíeta, *Metodología dirigida por modelos para el diseño de Funcionalidad Volátil en aplicaciones Web*. Recuperado de [http://postgrado.info.unlp.edu.ar/Carreras/Doctorado/Tesis/Urbíeta\\_Mario\\_Matía](http://postgrado.info.unlp.edu.ar/Carreras/Doctorado/Tesis/Urbíeta_Mario_Matía)

Lamarca, (2013). María Jesús Lamarca *Lapuente modelo OOHDM*, Recuperado de <http://www.hipertexto.info/documentos/oohdm.htm>.

(SENA, 2015), *Modelos De Calidad En El Desarrollo De Software*. Recuperado de  
[https://senaintro.blackboard.com/bbcswebdav/institution/semillas/228106\\_2\\_VIRTUAL-2015/contenido/oaaps/oaap10/aa2/oa\\_calidad/oa.pdf](https://senaintro.blackboard.com/bbcswebdav/institution/semillas/228106_2_VIRTUAL-2015/contenido/oaaps/oaap10/aa2/oa_calidad/oa.pdf)

(Garzas, 2012). Estandarizar la evaluación de la calidad de software, recuperado de <http://www.javiergarzas.com/2012/10/iso-9126-iso-25000-2.html>.

BORBÓN, (2013). Norma de evaluación iso/iec 9126. Recuperado de <http://activida reconoci miento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-soiec- 9126.html>

(Aparicio, 2012). El modelo COCOMO para estimar costes en un proyecto de software,  
Recuperado de <http://www.eoi.es/blogs/cesaraparicio/2012/05/06/el-modelo-cocomo-para-estimar-costes-en-un-proyecto-de-software/#>

COSC95. COCOMO University of Southern California. Recuperado de [http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)