

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



**PROYECTO DE GRADO**

**“SISTEMA WEB DE CONTROL DE PERSONAL, VENTAS Y FALLAS  
TÉCNICAS”**

**CASO: “SISCOM” SISTEMA INTEGRAL DE SERVICIOS EN  
COMPUTACIÓN**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA  
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

**POSTULANTE: MARITZA PAXI JARRO**

**TUTOR METODOLOGICO: M. Sc. FRANZ CUEVAS QUIROZ**

**ASESOR: LIC. GROVER ALEX RODRIGUEZ RAMIREZ**

**LA PAZ – BOLIVIA**

**2016**



**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 ESTRICTAMENTE 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 mis padres Roberto Paxi S. y Martha Jarro P.,  
que siempre me inculcaron el interés por el estudio.*

*A mis hermanos Alvaro, Jose Alfredo y Rosa Maria  
porque siempre me brindaron su apoyo y siempre  
estuvieron con migo.*

*A una persona especial que siempre estuvo a mi  
lado SFJL.*

*A todos mis seres queridos.*

## **AGRADECIMIENTOS**

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

*A mi familia por haberme brindado su apoyo estos últimos años.*

*Al M. Sc. Franz Cuevas Quiroz, por sus recomendaciones y colaboración para el desarrollo y culminación de este proyecto.*

*A la Lic. Grover Alex Rodriguez por su guía, observaciones, colaboración y tiempo proporcionado para el desarrollo y conclusión del presente proyecto.*

*A todos los Docentes de la carrera de Informática por haberme impartido sus conocimientos.*

*Al Señor. Jaime Antonio Megia, por su tiempo y colaboración para el desarrollo del proyecto.*

## **RESUMEN**

En la actualidad existen instituciones que cada día generan mucha información, en el caso de la empresa SISCOM cada día genera informes de ventas, reporte de caja , reporte de fallas técnicas y control de personal, esta institución realiza múltiples servicios de informática y diseño gráfico.

El presente proyecto tiene como finalidad facilitar el proceso de controlar al personal por ejemplo viendo las horas de llegada , salida y días que no vino a trabajar, también el sistema realizar el proceso de control de ventas donde registra la venta del día, mes y año para hacer un inventario para ver cuánto gana la empresa por sus productos o cuanto pierde, para finalizar también está la parte del proceso de control de fallas técnicas donde se pretende dejar como un reporte al gerente que maquinas está fallando para hacer las dichas reparaciones.

En la parte introductoria se muestra los antecedentes y actividades que realiza la empresa. También se muestra el análisis de los problemas y los objetivos propuestos.

Para el desarrollo del proyecto se aplicó la metodología de desarrollo ágil SCRUM apoyándose en la metodología de desarrollo UWE para el modelado del diseño.

El Sistema es un producto de calidad de acuerdo a la metodología de evaluación de calidad de sitios web llamada Web-site QEM.

Finalmente se puede concluir que los objetivos planteados fueron alcanzados y que el Sistema cumple con los requerimientos establecidos por el cliente.

## **ABSTRACT**

Currently there are institutions that every day generate much information, in the case of the company SISCOM every day generates sales reports, report cash report technical failures and personnel control, this institution performs many computer services and graphic design.

This project aims to facilitate the process of controlling staff for example watching the hours of arrival and departure days did not come to work, also the system perform the control process sales which records the sale of the day, month and year to make an inventory to see how much you earn the company for its products or how much lost, to end is also part of process control technical failures which aims to leave as a report to the manager that machine is failing to make such repairs.

In the introductory part of the background and activities of the company it is shown. analysis of the problems and the objectives are also shown.

For the project development methodology SCRUM agile development relying on UWE development methodology for modeling design it was applied.

The system is a quality product according to the assessment methodology quality site called Web-site QEM.

Finally it can be concluded that the proposed objectives were achieved and that the system meets the requirements set by the customer.

# INDICE

<b>CAPÍTULO I .....</b>	<b>1</b>
<b>MARCO INTRODUCTORIO.....</b>	<b>1</b>
1.1. Introducción.....	1
1.2. Antecedentes.....	2
1.2.1. Antecedentes institucionales.....	2
1.2.2. Antecedentes de proyectos similares .....	2
1.3. Planteamiento del problema .....	3
1.3.1. Problema central .....	3
1.3.2. Problemas secundarios.....	4
1.4. Definición de objetivos.....	4
1.4.1. Objetivo general.....	4
1.4.2. Objetivos específicos .....	4
1.5. Justificación .....	5
1.5.1. Justificación económica.....	5
1.5.2. Justificación social.....	5
1.5.3. Justificación tecnológica.....	5
1.6. Alcances y limites.....	6
1.6.1. Alcances.....	6
1.6.2. Limites .....	6
1.7. Aportes .....	7
1.7.1. Práctico .....	7
1.7.2. Teórico.....	7
1.8. Metodología y herramientas de implementación.....	7
1.8.1. Metodologías .....	7
1.8.2. Herramienta de implementación.....	8
1.8.2.1. Diagrama UML.....	8
1.8.2.2. Lenguaje de programación PHP .....	9
1.8.2.3. Gestor de base de datos MSQl .....	9
<b>CAPÍTULO 2 .....</b>	<b>10</b>
<b>MARCO TEÓRICO.....</b>	<b>10</b>
2.1. Marco institucional .....	10
2.2 Metodología SCRUM.....	12
2.3. Principales características.....	13
2.4. Principales elementos de la metodología.....	14
2.4.1. Herramientas y prácticas.....	14
2.4.2. El proceso .....	16
2.3.3 Pre-Game .....	17
2.3.4 Game.....	17
2.3.5 Post-Game .....	18
2.3.6 Roles y responsabilidades.....	19

2.4. Diseño.....	20
2.4.1 UML .....	20
2.5 Calidad de software .....	27
2.6 Funcionalidad .....	27
2.7 Confiabilidad .....	29
2.8 Mantenimiento.....	29
2.9 Certificación del software.....	30
2.10 Medición del software .....	30
2.7. Costo y beneficio .....	31
2.7.1 COCOMO.....	31
Modelos de estimación .....	31
2.8. Seguridad.....	34
2.9 Controles de acceso .....	34
2.10 Identificación y autenticación.....	34
2.11 Roles .....	35
<b>CAPÍTULO III .....</b>	<b>36</b>
<b>MARCO APLICATIVO .....</b>	<b>36</b>
3.1 Introducción.....	36
3.2 Proceso de desarrollo.....	36
3.3 Pre- Game .....	38
3.3.1 Planificación .....	38
3.4 Game.....	40
3.4.1. Primera iteración (SPRINT 1) .....	40
3.4.2. Segunda iteración (SPRINT 2) .....	42
3.4.3. Tercera iteración (SPRINT 3).....	50
3.4.4. Cuarta iteración (SPRINT 4) .....	55
3.4.2. Quinta Iteración (SPRINT 5).....	60
3.5 Post- Game .....	64
3.5.1 Diseño de las interfaces graficas .....	64
<b>CAPÍTULO IV.....</b>	<b>69</b>
<b>CALIDAD Y SEGURIDAD.....</b>	<b>69</b>
4.1 Introducción.....	69
4.2. Calidad.....	69
4.2.1 Caso de estudio.....	69
4.2.2 Tabla de características y atributos.....	70
4.2.3 Evaluación elemental.....	71
4.2.3.1 Mediciones elementales.....	72
4.2.4 Evaluación global .....	74
4.2.5 Análisis de resultados .....	75
4.3 Seguridad .....	75
4.3.1 Bases de la seguridad informática .....	75



4.3.2 Mecanismos básicos de seguridad .....	76
<b>CAPITULO V .....</b>	<b>78</b>
<b>COSTOS Y BENÉFICIOS .....</b>	<b>78</b>
5.1. Introducción .....	78
5.2. Punto función .....	78
5.2. COCOMO .....	81
5.2.1. Costos de la elaboración del proyecto .....	82
5.2.2. Costos del desarrollo del software .....	82
5.2.3. Costos de la implementación del sistema .....	84
<b>CAPITULO VI.....</b>	<b>85</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>85</b>
6.1. Conclusiones.....	85
6.2. Recomendaciones .....	86
BIBLIOGRAFÍA .....	87

## INDICE DE FIGURAS

Figura 1 Ubicación de la empresa SISCOM .....	10
Figura 2 Organigrama de la empresa SISCOM.....	12
Figura 3 Esquema general de Scrum .....	13
Figura 4 Roles que el equipo debe realizar.....	16
Figura 5 El procesos de Scrum .....	16
Figura 6 Ejemplo de Pila de Sprint con hoja de calculo.....	18
Figura 7 Modelado de casos de uso.....	21
Figura 8 Diagrama de clases.....	22
Figura 9 Diagrama de secuencia.....	22
Figura 10 Diagrama de estado .....	23
Figura 11 Diagrama de paquetes .....	24
Figura 12 Etapas de Scrum.....	36
Figura 13 Diagrama de casos de uso general del sistema.....	37
Figura 14 Modelo la base de datos .....	41
Figura 15 Diagrama de casos de uso- Administración de usuario .....	44
Figura 16 Diagrama de casos de uso - Autenticación de usuario .....	44
Figura 17 Diagrama de navegación Administración de usuarios .....	46
Figura 18 Diagrama de navegación Autenticación.....	47
Figura 19 Modelo de presentación - Autenticación de usuario .....	47
Figura 20 Diagrama de casos de uso – control de personal.....	52
Figura 21 Diagrama de navegación de control de personal.....	53
Figura 22 Modelo de presentación – control de personal.....	54
Figura 23 Diagrama de casos de uso - control de ventas.....	57
Figura 24 Diagrama de navegación de control de ventas .....	58
Figura 25 Modelo de presentación – control de ventas .....	59
Figura 26 Diagrama de casos de uso – control de fallas técnicas.....	61
Figura 27 Diagrama de navegación de control de personal.....	63
Figura 28 Modelo de presentación - Autenticación de usuario .....	63
Figura 29 Interface – Inicio de sesión .....	65
Figura 30 Interface - Página de inicio del sistema.....	65
Figura 31 Interface - Administración de usuarios .....	66
Figura 32 Memoria del sistema y carga de la CPU .....	67
Figura 33 Solicitud de apertura y transferencia de datos.....	68
Figura 34 Peticiones del usuario.....	68
Figura 35 Rango de aceptabilidad de preferencia de calidad .....	71

## INDICE DE TABLAS

Tabla 1 Ejemplo de laPila de Producto .....	17
Tabla 2 Factor de ponderación .....	28
Tabla 3 Constantes para calcular distintos aspectos de costos .....	32
Tabla 4Constantes a reemplazar en la formula.....	33
Tabla 5 Definición de actores .....	38
Tabla 6 Requerimientos.....	38
Tabla 7 Requerimientos para instalar el sistema .....	39
Tabla 8 Requerimiento de programas.....	39
Tabla 9 Sprint 1 .....	40
Tabla 10 Sprint 2 .....	43
Tabla 11 Detalle de casos de uso Administración de usuario .....	45
Tabla 12 Detalle de caso de uso Autenticación de usuario.....	45
Tabla 13 Pruebas de funcionalidad - Inicio de sesión .....	48
Tabla 14Pruebas de funcionalidad - Registro de usuario .....	49
Tabla 15 Pruebas de funcionalidad - Administración de usuario.....	49
Tabla 16 Sprint 3 .....	51
Tabla 17 Detalle de casos de uso Administración de usuario .....	52
Tabla 18 Pruebas de funcionalidad – registro de personal .....	55
Tabla 19 Sprint 4 .....	56
Tabla 20 Detalle caso de control de ventas .....	57
Tabla 21 Pruebas de funcionalidad – registro de ventas .....	59
Tabla 22 Sprint 5 .....	60
Tabla 23 Detalle caso de uso control de fallas técnicas .....	62
Tabla 24 Pruebas de funcionalidad – registro de fallas técnicas .....	64
Tabla 25 Requerimientos de calidad .....	70
Tabla 26 Computo de las preferencias parciales .....	72
Tabla 27 Calidad global.....	74
Tabla 28 Cálculo de puntos función.....	79
Tabla 29 Escala de rangos para hallar el punto función .....	80
Tabla 30 Cuestionario para el ajuste de complejidad.....	80
Tabla 31 Costos de elaboración del proyecto.....	82
Tabla 32 Factor LCD/PF de lenguajes de programación .....	82
Tabla 33 Tipos de proyectos de software .....	83

### 1.1.Introducción

La tecnología avanza rápidamente hoy en día las empresas no pueden dejar de lado los sistemas, en especial aquellas que manejan gran cantidad de datos, porque crece la complejidad al controlar y hacer seguimiento de toda esa información.

Analizando y viendo nuestra sociedad que todo se estas sistematizando y el avance de la tecnología va rápida en el campo de la informática, existe un notable interés tanto en las empresas públicas como privadas que todo sea sistematizado y de mejorar el rendimiento de su empresa y teniendo un control de las tareas que se realizan.

La Empresa SISCOSM requiere de un sistema adecuado para la toma de decisiones respecto al control de personal, control de ventas, control de falla y el control de las tareas que se deja por entregar.

Bajo estas prespectativas se desea desarrollar un sistema de control de personal, control de ventas, control de fallas técnicas y el control de entregas de trabajo, este sistema será útil para la mejor administración la empresa SISCOSM.

Es así, que el presente proyecto de grado pretende resolver varios problemas que se observa al realizar el seguimiento del manejo de la empresa SISCOSM; aplicando técnicas y métodos para su diseño, con un lenguaje de codificación y plataforma de software de alto nivel para su implementación.

De esta manera el sistema nos permitirá obtener reportes periódicamente, respecto a cómo va funcionando la empresa para una mejor administración.

## 1.2. Antecedentes

### 1.2.1. Antecedentes institucionales

La empresa “SISCOM” es una institución donde brindan múltiples servicios en Sistemas Informáticos y diseño gráfico, Actualmente SISCOM se encuentra ubicado en la ciudad de El Alto Plaza Ballivian Av. Alfonso Ugarte N° 45, emplea a más de 10 personas tomado turno y tarde.

Las funciones que realiza cada día son:

- Controlar a diario el ingreso de la personal hora entrada y salida manualmente.
- Cada mes se hace una planilla de pagos al personal manualmente.
- Control de las ventas del día, mes y año se realiza manualmente almacenando en un programa de Excel creado cada día un libro de ventas.
- Control de fallas técnicas cuanto algún equipo está mal se registra en un cuaderno manualmente.
- Reporte de caja se registra manualmente.

### 1.2.2. Antecedentes de proyectos similares

Los trabajos citados a continuación fueron realizados en la carrera de informática de la universidad Mayor de San Andrés:

- **Sistema de Administración de Personal Unidad De Talento Humano Hospital Argramont.**

Este sistema realiza el control de horarios de entrada y salida de los empleados, la actualización sus hojas de vida, el registro de contratos mixtos ya sea básico o porcentaje y el movimiento que tiene en la institución como se altas, bajas, rotación, permisos y vacación.

Para el desarrollo del dicho proyecto se utilizó la metodología RUP, (Proceso Unificado De Rational), metodología estándar para la construcción completa del ciclo de ingeniería de software, esta metodología es apoyada con UML (Lenguaje Unificado Del Modelo) (Calle ,2011).

- **Sistema de Control y Seguimiento de Activos Fijos para la Empresa Autoventa Intercambio.**

El sistema tiene como objetivo implementar un sistema informático que mejore la administración y control de activos de fijos, promoviendo la racionalidad en la distribución de estos bienes a los servicios de la entidad para que cumpla sus funciones a cabalidad.

Para el desarrollo del proyecto se utiliza la metodología RUP (Proceso Unificado Rational), como metodología para diseñar el ciclo de vida del sistema y las herramientas tecnológicas utilizadas son un gestor de base de datos y Msql , en el lenguaje PHP y este sistema está desarrollado para Windows XP y Windows Server NT (Cauna ,2012).

- **Sistema de Control de Personal, Caso: División de Acciones y Control UMSA**

El sistema de control de personal coadyuvara en la organización y toma de decisiones otorgando un manejo adecuado y eficiente de la información de manera que toda la información generada por las asistencias en diferentes predios se integrara una sola base de datos que administrada por el departamento de recursos humanos. El sistema y los datos serán centralizados en un servidor al cual tendrán acceso todos los funcionarios y usuarios dados de alta, desde cualquier punto de la institución y del radio urbano.

La metodología usada para este proyecto es Rup(rational unified process ) las herramientas principales para desarrollar este proyecto son : JDevelope para programación y PostgresSQL como gestor de base de datos además de que estas herramientas son software libre (Nina ,2011).

### **1.3.Planteamiento del problema**

#### **1.3.1. Problema central**

La empresa hace todos sus controles en forma manual y es demoroso y a algunas veces que el personal llega a la hora que quiere y en la parte de control de ventas no sabe lo que le produce ganancias o pérdidas por que el reporte que se deja es en general solo la renta lograda al día nada más por otro lado también están las fallas de algunas máquina que no están funcionando normal

y el personal se olvida de avisar de las fallas técnicas y por ende el dueño no sabe que está ocurriendo en su empresa.

¿De qué manera se podrá realizar el control del personal, el control de ventas y el control de fallas técnicas para la Empresa SISCOM?

### **1.3.2. Problemas secundarios**

- Los registros de la hora de llegada y salida de la persona son manualmente y hay algunas veces que el personal se olvida anotarse o anotan datos no correctos.
- La planilla de pagos al personal no hace el descuento correcto porque es muy complicado de controlar al personal.
- En la parte de ventas del día, mes y año algunas veces al sacar las ganancias por mes, día y año son muy demorosos y se pierde algunos datos.
- Los informes de caja algunas veces no son registradas correctamente y puede que haya pérdidas para la empresa.
- Los reportes de inventario se hace manualmente por tanto no se tiene los datos adecuados de pérdidas y ganancias de la empresa.

## **1.4. Definición de objetivos**

### **1.4.1. Objetivo general**

Desarrollar e implementación de un sistema web para el control de personal, control de ventas y las fallas técnicas para la empresa SISCOM.

### **1.4.2. Objetivos específicos**

- Diseñar e implementar de una Base de Datos.
- Implementar tecnología de arquitectura de software establecer una política de trabajo donde los controles de las ventas sean mejores.
- Elaborar un registro de control de fallas técnicas de las maquinas que necesita reparación

- Desarrollar el módulo de reportes de fallas técnicas, personal y ventas para una mejor información de la empresa.
- Elaborar un registro de control de ventas para ver en que se optime más ganancias.
- Diseñar una planilla de control de personal donde se pueda controlar la llegada y la salida del personal.

## **1.5. Justificación**

### **1.5.1. Justificación económica**

El sistema Web permitirá que la Empresa SISCOM reduzca sus pérdidas económicas que resultan de la mala administración de sus ventas y el pago de salario del personal, teniendo una mejor información sobre cómo se va a descontar al personal por los atrasos o los días faltantes.

La facilidad de realizar el pago de salarios mediante una planilla donde se verá el desempeño del personal y el gerente podrá tener un mejor control del personal.

### **1.5.2. Justificación social**

El presente proyecto se convierte en un instrumento que facilite el control y la toma de decisiones, permitiendo a la empresa SISCOM contar con un mejor proceso a la hora de tomar decisiones.

Adicionalmente, los clientes son los beneficiados por que recibirán una buena atención porque habrá más control en las áreas de la empresa.

### **1.5.3. Justificación tecnológica**

El proyecto es justificable técnicamente debido a las facilidades en cuanto a tecnología cuenta la empresa SISCOM, tanto en hardware como en software son reutilizados para la implementación del presente proyecto.

Es importante mencionar que el desarrollo del proyecto es realizado en su totalidad, con herramientas de software libre, lo cual permite un costo mínimo en el proyecto.



## 1.6. Alcances y límites

### 1.6.1. Alcances

En este sistema de control de personal, fallas técnicas y ventas su alcance será local orientado a la empresa “SISCOM”.

La aplicación se encargará de automatizar y optimizar los procesos. El sistema realizará las funciones siguientes:

- **Módulo control de personal:** Realiza altas bajas y modificaciones, para tener un registro de las personas que trabajan en la empresa.
- **Módulo control de fallas técnicas:** Efectúa el registro de todas las máquinas la que presentan fallas tiene para que así el dueño pueda arreglar la maquina o hacer soporte técnico.
- **Módulo control de ventas:** Muestra una planilla de control de ventas que se realizan al día o al turno que esta el personal
- **Planilla de control de personal:** En la planilla de control de personal va a tener una forma de controlar si el personal cumple con los horarios, también se va a registrar salida y llegada del personal.
- **Módulo de reportes:** Muestra los reportes del personal, ventas y fallas técnicas de acuerdo a la necesidad del cliente.

### 1.6.2. Límites

El sistema está limitado al manejo de la información del control de ventas donde se registre las ventas y después se genere un reporte, en la parte del control de personal se registrara al personal y se hará una planilla de control de llegadas y salidas del personal y se generar un reporte y por otro lado también estará el control de fallas técnicas que hay en la empresa y necesitan ser registradas para que el dueño pueda hacer mantenimiento de las maquinas.

## **1.7.Aportes**

### **1.7.1. Práctico**

En realidad el aporte más destacado será el mismo sistema de información aplicada en esta área que no ha sido desarrollada hasta el momento, si bien existen muchos paquetes que se tienen las mismas características.

### **1.7.2. Teórico**

El aporte teórico es el desarrollo del sistema utilizando la Metodología Scrum para una mejor organización y para el modelado de diagramas UML para el diseño del sistema para luego programarla en el Lenguaje Php utilizando un gestor de base de datos Mysql y todo esto debe de estar en la web para a el dueño de la empresa se conecte al internet inmediatamente pueda ver su reporte de como estuvo el día en la empresa

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

### **1.8.1. 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 información 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 sociedad.

Las metodologías y estándares utilizadas en el desarrollo de software, cada día van proporcionando las guías para poder conocer el camino que debemos recorrer, desde su inicio hasta antes de comenzar con la implementación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento y la entrega del mismo en un tiempo determinado.

Es de suma importancia elegir la metodología adecuada, así como las herramientas de diseño e implementación, es por ello que la metodología ágil Scrum nos proporcionara todas las bases para llegar al éxito de la elaboración del sistema.

Para el presente Proyecto de Grado, se utilizara la metodología ágil Scrum, metodología del proceso de ingeniería de software que proporciona un enfoque disciplinario para asignar tareas y responsabilidad dentro de una organización de desarrollo.

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

### **1.8.2. Herramienta de implementación**

Para el desarrollo de este proyecto se utiliza herramientas de diseño y programación, estas herramientas se mencionan a continuación.

Herramientas se mencionan a continuación:

1. **Diagramas UML**, para la parte del diseños
2. **Lenguaje de programación PHP**, para la programación del proyecto.
3. **Gestor de base de datos MsqL**, para el almacenamiento de datos.

#### **1.8.2.1. Diagrama UML**

UML es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y es frecuentemente usada por analistas funcionales (aquellos que definen qué debe hacer un programa sin entrar a escribir el código) y analistas-programadores (aquellos que dado un problema, lo estudian y escriben el código informático para resolverlo en un lenguaje como Java, C#, Python o cualquier otro). Por tanto si estás dando tus primeros pasos en programación, te recomendaríamos que te olvides de UML hasta que tengas unos conocimientos mínimos como uso de condicionales, bucles, y conocimiento de la programación orientada a objetos. Esto es solo una recomendación, en realidad prácticamente cualquier persona puede usar UML, incluso podría usarse para realizar esquemas o documentación de procesos que no tengan que ver con la informática. (UML216)

### **1.8.2.2.Lenguaje de programación PHP**

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

¿Cómo trabaja PHP? El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo <http://www.aprenderaprogramar.com/index.php> en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática. (PH216)

### **1.8.2.3.Gestor de base de datos MSQl**

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. (MQ216)

#### 2.1. Marco institucional

Es una empresa SISCOM es una empresa Boliviana que se dedica a múltiples servicios informáticos y diseño como por ejemplo realiza mantenimiento de computadoras ,trabajos en ofimática, hace diseños de todo tipo , en si da soluciones que nos permiten realizar nuestras labores con mayor rapidez y eficacia, cambiado la forma en que vivimos Inclusive aquellos que no les gusta la tecnología, de vez en cuando disfrutan de las ventajas de ciertos servicios informáticos y admiran la calidad de los servicios que se obtienen.

Actualmente SISCOM se encuentra ubicado en la ciudad de El Alto Plaza Ballivian Av. Alfonso Ugarte N° 45, ver en la figura 1.



**Figura 1 Ubicación de la empresa SISCOM**

Fuente: [Sv12016]

La empresa SISCOM emplea a más de 10 personas tomado turno mañana y tarde.

Cuenta con el registro de comercio, órgano concesionado a la Fundación Para El Desarrollo Empresarial – Fundaempresa. Mediante contrato de concesión de fecha 17 de diciembre de 2001, en merito a las facultades establecidas en el decreto supremo 26215 y en cumplimiento a las normas establecidas en el código de comercio, con Objeto de brindar servicios de informática

También cuenta con el Certificado De Funcionamiento de actividad económica con un PATENTE N° 1511314562 y PMC N° 10259221.

SISCOM trabaja según los principios de responsabilidad social, buscando ser una empresa responsable donde el cliente este satisfecho con el servicio que brinda.

SISCOM cree en potencial porque está atento a la tecnología para ver la necesidad que puede tener el cliente, actualizándose constantemente y viendo que productos están en el mercado más que todo en la parte de diseño gráfico.

### **Misión**

Ser una empresa social, íntegramente responsable y sostenible, que busca que el cliente este satisfecho y mejorar su calidad de vida con la tecnología.

### **Visión**

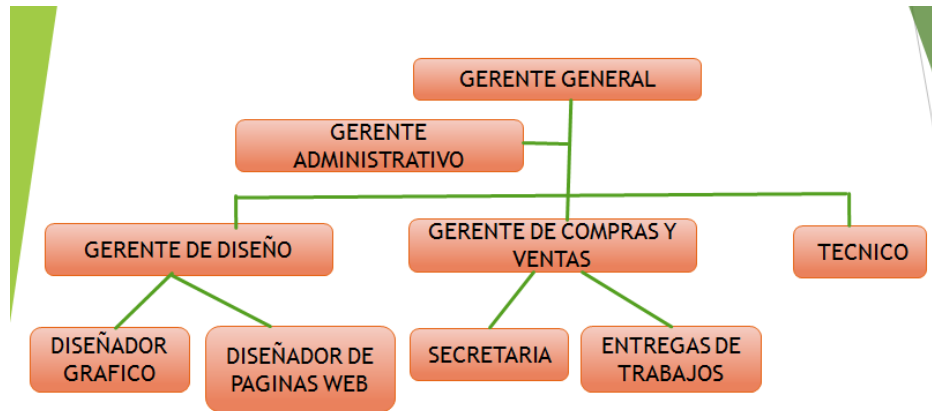
Ser un emprendimiento y ejemplo para la sociedad buscando mejorar la calidad de vida del cliente que es nuestra mayor prioridad, ofreciendo nuestros servicios informáticos.

### **Objetivos**

- Siempre ir actualizándonos con la tecnología para brindar una mejor atención al cliente.
- Mejorar la calidad de vida de los clientes
- Fidelizar el personal a través de un programa de comunicación.

### **Organigrama**

En la figura 2 se muestra el organigrama de la empresa SISCOM



**Figura 2 Organigrama de la empresa SISCOM**

Fuente [Elaboración propia]

### Funciones

- Controlar a diario el ingreso del personal hora entrada y salida manualmente, lo que hacen es anotar cada día en un cuaderno de control de personal
- Cada mes se hace una planilla de pagos al personal manualmente
- Control de las ventas del día, mes y año se realiza manualmente almacenando en un programa de Excel creado cada día un libro de ventas.
- Control de fallas técnicas cuanto algún equipo está mal se registra en un cuaderno manualmente
- Reporte de caja se registra manualmente

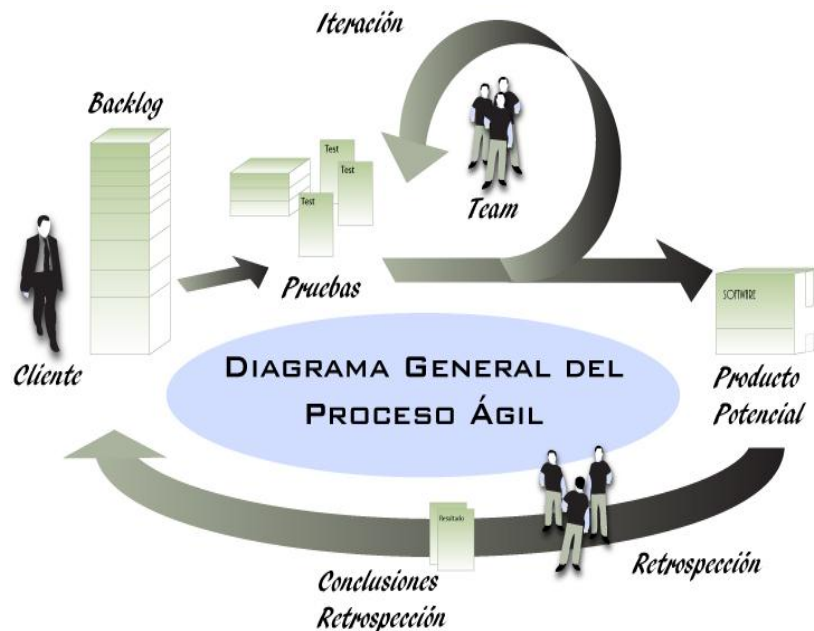
### 2.2 Metodología SCRUM

Scrum es un proceso ágil para desarrollar software que fue aplicado por primera vez por Ken Schwaber y Jeff Sutherland, quienes lo documentaron en detalle en el libro Agile Software DevelopmentwithScrum. Esta metodología centra su atención en las actividades de Gerencia y no especifica prácticas de Ingeniería. Fomenta el surgimiento de equipos autodirigidos cooperativos y aplica inspecciones frecuentes como mecanismo de control. (Peralta, 2003)

Scrum parte de la base de que los procesos definidos funcionan bien solo si las entradas están perfectamente definidas y el ruido, ambigüedad o cambio es muy pequeño. Por lo tanto, resulta

ideal para proyectos con requerimientos inestables, ya que fomenta el surgimiento de los mismos (Peralta, 2003).

El ciclo de vida definido por Scrum es incremental iterativo y se caracteriza por ser muy adaptable, ver Figura 3.



**Figura 3 Esquema general de Scrum**

Fuente [Sv2016]

### 2.3.Principales características

Existen varias características de la metodología de desarrollo Scrum, a continuación se listará las principales características (Peralta, 2003):

- Equipos autodirigidos
- Utiliza reglas para crear un entorno agile de administración de procesos
- No prescribe practicas específicas de ingeniería
- Los requerimientos se capturan como ítems de la lista ProductBacklog
- El producto se construye en una serie de Sprints de un mes de duración



## **2.4.Principales elementos de la metodología**

### **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 (Peralta, 2003).

#### **a) ProductBacklogList**

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.

El ProductBacklogList 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 (Peralta, 2003).

#### **b) Sprint**

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.

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 reléase 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 Figura 2.4 se pueden ver las prácticas involucradas en un Sprint (Peralta, 2003).

### **c) Sprint Backlog**

Es una lista que tiene los ítems de la ProductBacklogList que van a ser implementados en el siguiente Sprint y es el punto de entrada de cada Sprint.

Los ítems son seleccionados por el ScrumTeam, el Scrum Master y el ProductOwner 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 ScrumTeam determina que tareas debe desempeñar para cumplir 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 una nueva información (Peralta, 2003).

### **d) Stabilization Sprint**

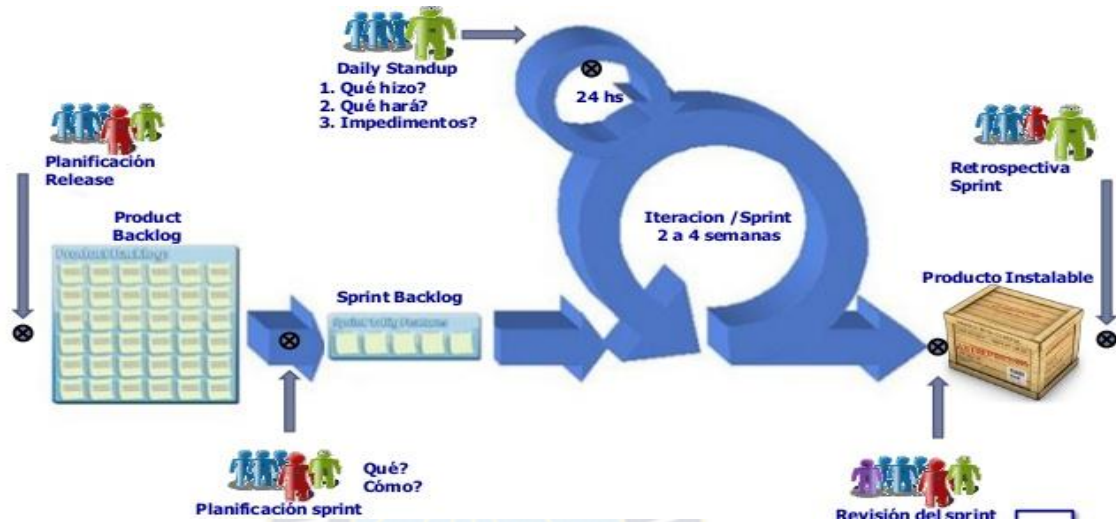
En estos Sprints el equipo se concentra en encontrar defectos, no en agregar funcionalidad. Suelen aplicarse cuando se prepara un producto para el release. Son útiles cuando se están realizando pruebas beta, se está introduciendo a un equipo en la metodología de Scrum o cuando la calidad de un producto no alcanza los límites esperados.

No fueron definidos por Scrum pero han sido recomendados por su utilidad al aplicar esta metodología (Peralta, 2003).

### **e) Scrum of Scrums o Metascrum**

Los equipos de Scrum suelen tener entre 5 y 10 personas, sin embargo esta metodología ha sido aplicada en proyectos que involucran a más de 600 personas. Esto ha sido llevado a cabo dividiendo a los accionistas en equipos 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 donde tienen que tomar en cuenta la pila de productos, los Scrum diario y sprint luego la demostración como se de en la figura 4 (Peralta, 2003).

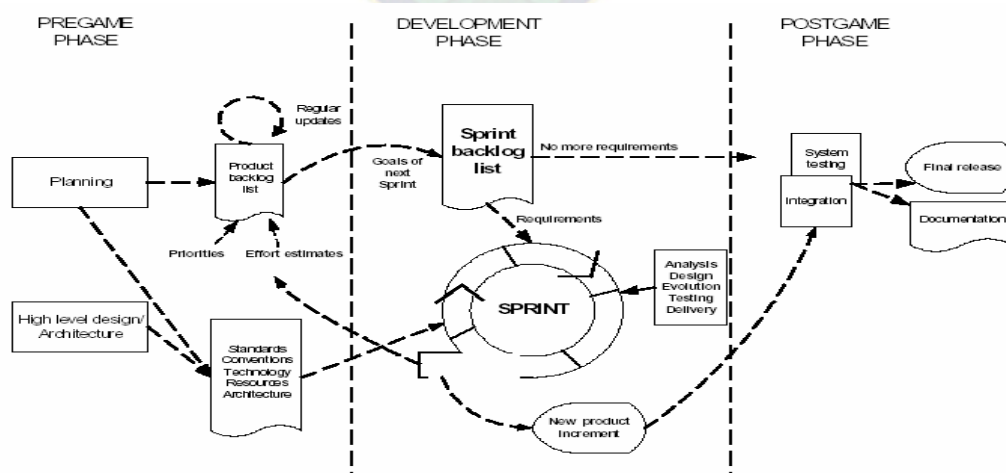


**Figura 4 Roles que el equipo debe realizar**

Fuente [PROYSCRUM]

## 2.4.2. El proceso

Scrum consta de tres fases: Pregame, Development y Postgame que se describen en la Figura 5. La fase del Pregame incluye dos subfases: Planning y Architecture (Peralta, 2003).



**Figura 5 El procesos de Scrum**

Fuente [PROYSCRUM]

### 2.3.3 Pre-Game

Es la primera fase del proceso de Scrum donde se realiza el *planning*(planeación), *architecture*(arquitectura) (Peralta, 2003)

- **Planning**

Consiste en la definición del sistema que será construido. Para esto se crea la lista ProductBacklog 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. Un ejemplo de pila de producto se ilustra en la tabla 1 (Peralta, 2003).

**Tabla 1 Ejemplo de la Pila de Producto**

Fuente: [Peralta, 2003]

Id	Prioridad	Descripción	Est.	Por
1	Muy alta	Plataforma tecnológica	30	AR
2	Muy Alta	Interfaz de usuario	40	LM
3	Muy Alta	Un usuario se registra en el sistema	40	LM
4	Alta	El operador define el flujo y textos de un expediente	60	AR
5	Alta	xxx	999	CC

- **Architecture/High level design**

El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la ProductBacklogList. Se sostiene una reunión de revisión de diseño (DesignReview Meeting) para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión (Peralta, 2003)

### 2.3.4 Game

La fase de Development también llamada GamePhase es la parte ágil de Scrum:

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. Un ejemplo de una pila de Sprint se ilustra en la Figura 6 (Peralta, 2003)

PROYECTO					Sáb 07 Ene	Dom 08 Ene	Lun 09 Ene	Mar 10 Ene	Mié 11 Ene
Inicio	Fin	Jornada							
7-ene-12	1-abr-12	40 hs							
Tareas pendientes					15	15	14	14	11
Horas pendientes					172	162	148	142	124
Fecha de Cierre					12-ene	12-ene	12-ene	13-ene	16-ene

PILA DEL PRODUCTO					OBJETIVO DEL SPRINT				
Categoría	Tarea	Responsable	Estimado en horas	Estado	Crear y publicar versión básica del sitio web público				
Diseño	Crear diseño de base de datos	Juan	24	Completo	24	16	8	4	
Diseño	Validar diseño de base de datos	Pedro	4	Completo	4	4	4	4	
Desarrollo	Contratar servicio de hosting	Pedro	4	Completo	4	2			
Desarrollo	Crear layout y estilos de sitio web	Maria	16	Activo	8	8	4	2	
Desarrollo	Crear página principal	Maria	24	Pendiente	24	24	24	24	24
Desarrollo	Mostrar resúmenes de noticias por sección	Juan	16	Pendiente	16	16	16	16	8
Desarrollo	Crear banners de publicidad	Luis	24	Pendiente	24	24	24	24	24
Desarrollo	Visualizar un Artículo	Luis	8	Pendiente	8	8	8	8	8
Desarrollo	Imprimir un Artículo	Luis	4	Pendiente	4	4	4	4	4

Figura 6 Ejemplo de Pila de Sprint con hoja de calculo

Fuente [PROYSCRUM]

### 2.3.5 Post-Game

Cierre: Esta fase trata sobre la preparación para la liberación, incluyendo la documentación final, las pruebas preparadas antes de la liberación y la liberación. El sistema está listo para ser liberado y es en esta etapa en la que se realiza la integración, pruebas del sistema y documentación (Peralta, 2003).

### **2.3.6 Roles y responsabilidades**

#### **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 (Peralta, 2003).

#### **ProductOwner**

Es el responsable del proyecto, administra, controla y comunica la BacklogList. Es el responsable de encontrar la visión del producto y reflejarla en la BacklogList. Generalmente esta persona puede ser el Product Manager, Marketing, InternalCustomer(Peralta, 2003).

#### **ScrumTeam**

Es el equipo del proyecto que tiene la autoridad para decidir cómo organizarse para cumplir con los objetivos de un Sprint. Sus tareas son: EffortEstimation (Estimar Esfuerzo), crear el Sprint Backlog, revisar la ProductBacklogList y sugerir obstáculos que deban ser removidos para cumplir con los ítems que aparecen (Peralta, 2003).

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 usuarios. La dedicación de los miembros del equipo debería ser full-time con algunas excepciones. La membresía solo puede cambiar entre Sprints (no durante) (Peralta, 2003).

#### **Customer**

El cliente participa en las tareas que involucran la lista ProductBacklogList.

#### **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

ScrumOwner. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Scrum Master en la reducción de la ProductBacklogList (Peralta, 2003).

## **2.4. Diseño**

### **2.4.1 UML**

El Lenguaje Unificado de Modelado pre-escibe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. (MSO15)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura re-visión.

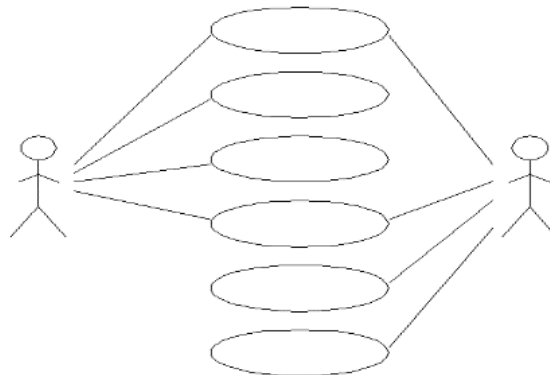
Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones y otros)
- Relaciones: Relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.[MSO15]

#### 2.4.1.1 DIAGRAMA DE CASOS DE USO

El diagrama de casos de usos representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. En la figura 7., se muestra un ejemplo de casos de uso, donde se muestran los actores y las operaciones que pueden realizar.[MSO15]



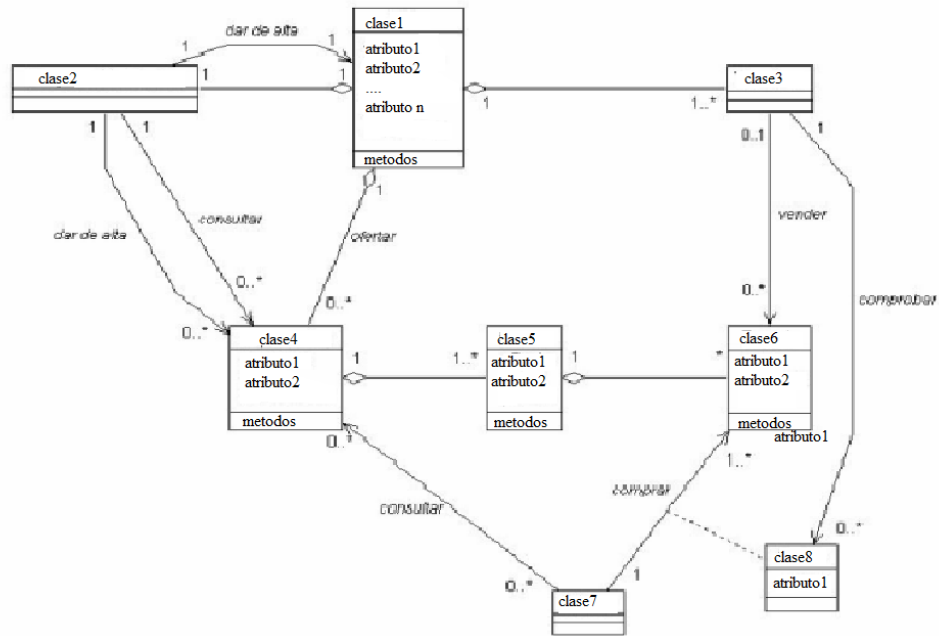
**Figura 7 Modelado de casos de uso**

Fuente: [MSO15]

#### 2.4.1.2 DIAGRAMA DE CLASES

El diagrama de clases muestra un conjunto de clases, interfaces y sus relaciones. Este es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. En la figura 8, se muestran las clases globales, sus atributos y los métodos que la acompañan.(MSO15)



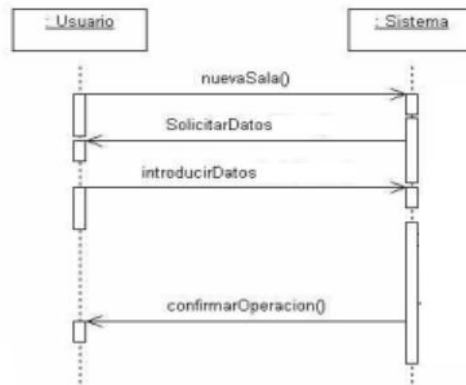


**Figura 8 Diagrama de clases**

Fuente: [MSO15]

### 2.4.1.3 DIAGRAMA DE SECUENCIA

En el diagrama de secuencia se muestra la interacción de los objetos que componen un sistema de forma temporal. La figura9., muestra la interacción de crear una secuencia.(MSO15)

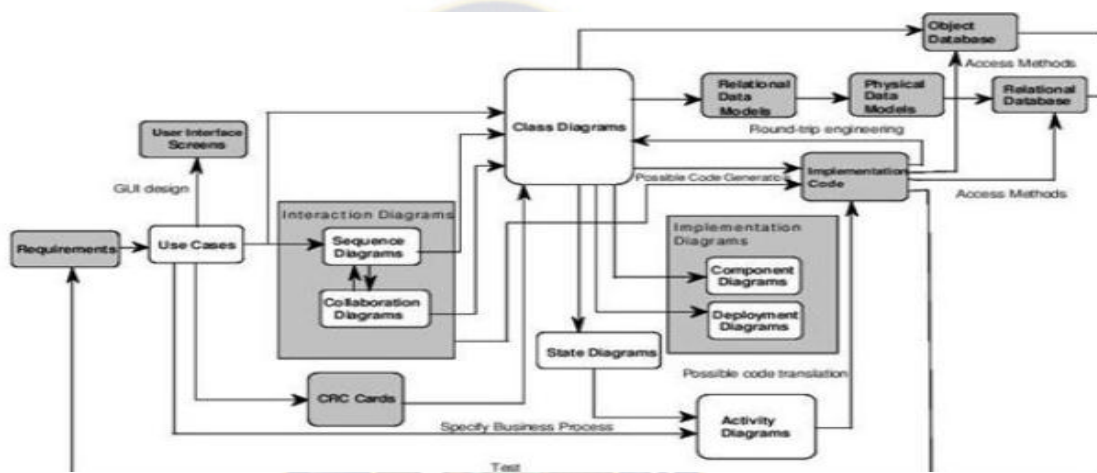


**Figura 9 Diagrama de secuencia**

Fuente: [MSO15]

### 2.4.1.3 DIAGRAMA DE ESTADOS

El comportamiento en tiempo real de cada clase que tiene comportamiento dinámico y significativo, se modela usando un Diagrama de Estado. El diagrama de actividad puede ser usado también aquí, esta vez como una extensión del diagrama de estado, para mostrar los detalles de las acciones llevadas a cabo por los objetos en respuesta a eventos internos. La figura 10, representa en una forma gráfica el diagrama de estado. [MSO15]

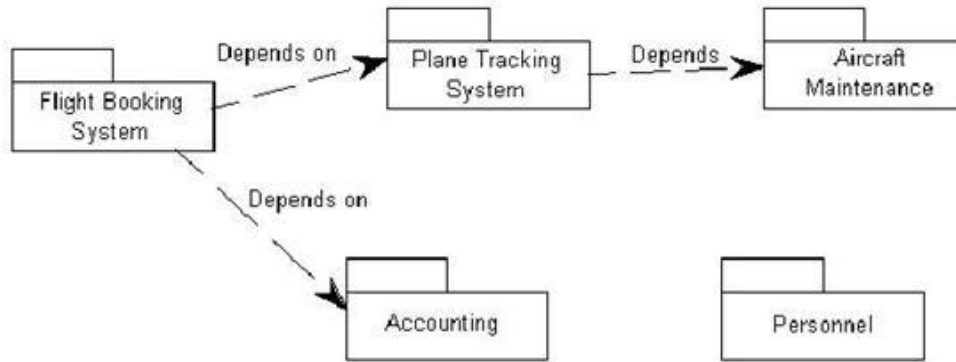


**Figura 10 Diagrama de estado**

Fuente: [MSO15]

### 2.4.1.4 DIAGRAMA DE PAQUETES

UML introduce la noción de un paquete como el ítem universal para agrupar elementos, permitiendo a los modeladores subdividir y categorizar sistemas. Los paquetes pueden ser usados en cualquier nivel, desde el nivel más alto, donde son usados para subdividir el sistema en dominios, hasta el nivel más bajo, donde son usados para agrupar casos de uso individuales, clases, o componentes. La figura 11, representa gráficamente los paquetes en una forma gráfica. (MSO15)



**Figura 11 Diagrama de paquetes**

Fuente: [MSO15]

## 2.4.2 UWE

Para presentar la parte de diseño utilizaremos UWE

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web.

Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos.

Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web.

De esta manera, se obtiene una notación UML adecuada aun dominio en específico a la cual se le conoce como Perfil UML.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Además de estar considerado como una extensión del estándar UML, también se basa en otros estándares como por ejemplo: XMI como modelo de intercambio de formato, MOF para la meta-modelado, los principios de modelado de MDA, el modelo de transformación del lenguaje QVT y XML (MTODUWE).

### **Actividades de modelado de UWE.**

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web.

### **Etapas o sub-modelos de UWE**

- Modelo de Casos de Uso

- Modelo de contenido
- Modelo de estructura
- Modelo Abstracto
- Modelo de Adaptación
- Modelo de flujo de presentación
- Modelo de ciclo de vida del objeto
- Modelo Lógico-Conceptual

UWE apunta a construir un modelo conceptual de una aplicación Web, procurano hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web.

La construcción de este modelo lógico-conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos.

El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web (MTODUWE).

### **Modelo de Navegación**

Consta de la construcción de dos modelos de navegación, el modelo de espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionaran (MTODUWE).

### **Modelo de presentación**

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario (MTODUWE).

## **2.5 Calidad de software**

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

La calidad es la aptitud de un producto o servicio para satisfacer las necesidades del usuario.

Es la cualidad de todos los productos, no solamente de equipos sino también de programas.

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta (CDS12).

La calidad del software se define como: concordancia con los requisitos de rendimiento explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente (Pressman,2010).

La calidad del software consiste en aquellos procedimientos, técnicas e instrumentos aplicados por entes capacitados para garantizar que un producto cumpla o supere un nivel mínimo aceptable para su comercialización, si es que así se lo ha planteado, lo que hasta el momento no se tiene estándares del software, no es lo mismo que las pruebas del sistema, estos son implícitos al momento de hacer las pruebas de integración final del sistema (Pressman,2010).

## **2.6 Funcionalidad**

El punto función es una métrica orientada a la función. Es una medida indirecta del software y del proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa. Los puntos de función se calculan llenando los datos en la tabla 3 para el cálculo se determinan cinco características de dominios de información:

**Tabla 2 Factor de ponderación**

Fuente: [Elaboración propia]

Parámetros de medición	Cuenta	Factor de ponderación						
			Simple	Medio	Complejo			
Número de entradas de usuario		x	5	4	3	=		
Número de salidas de usuario		x	6	9	3	=		
Número de peticiones de usuario		x	1	13	0	=		
Número de archivos		x	1	9	5	=		
Número de interfaces externas		x	6	4	0	=		
Cuenta total	→							

- El número de entradas de usuarios. Se cuenta cada entrada de usuario que proporciona al software de diferentes datos orientados a la aplicación. Las entradas deben diferenciar de las peticiones, estas pueden ser el ingreso de datos o control de flujo de la información, y otros datos importantes. Dichas entradas se llevan a cabo mediante las siguientes interfaces.
- Pantalla de inicio del sistema, pantalla de control de permisos, grupos y usuarios, Pantalla de control de registros de producción.
- Número de salidas de usuario. Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. Las salidas de usuario representan los informes, pantallas, mensajes de error. En nuestro ámbito tenemos como salidas: los mensajes de error, mostrar los registros de producción, y otras interfaces que se desarrollan en el software.
- Número de peticiones de usuario. Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta inmediata del software en forma de salida interactiva. Se cuenta por separado cada petición, en este caso una petición viene a ser: la consulta de rendimientos, de pH, de horas pasteurizadas, horas de fabricación y otras consultas desarrolladas en el software.
- Número de archivos. Se cuenta cada archivo lógico maestro, es decir un grupo lógico de datos que pueden ser parte de una gran base de datos o un archivo independiente. Se cuenta con un archivo de configuración del sistema, el cual permite hacer el enlace con el servidor Postgres, además de configurar los clientes.

## 2.7 Confiabilidad

Para determinar la confiabilidad del sistema se especifica el instante en el que comienza a funcionar determinado por  $t_0=0$ . A partir de este momento se observa el trabajo del sistema hasta que se introduzca una falla en el instante  $T$  que se va aproximando a una variable aleatoria continua, que nos determina la confiabilidad en términos probabilísticos. Entonces se tiene las siguientes probabilidades:

$$P(T \leq t) = F(t) \quad \dots(1) \text{Probabilidad de fallas}$$

$$P(T > t) = 1 - F(t) \quad \dots(2) \text{Probabilidad de trabajo sin fallas}$$

Se calcula estas probabilidades mediante la distribución exponencial. Donde, punto función, resultado obtenido en el cálculo realizado en un punto anterior a este.

$$\text{El } F(t) \text{ está dada por: } F(t) = \text{Puntofunción} * e^{(-\lambda * t)}$$

## 2.8 Mantenimiento

Se calcula el índice de madurez del software (IMS), estableciendo los cambios que ocurrieron con la versión del producto.

MT: Nro. De módulos en la versión actual.

Fc: Nro. De módulos en la versión actual que se han cambiado.

Fa: Nro. De módulos en la versión actual que se han añadido.

Fa: Nro. De módulos en la versión actual que se han eliminado.

$$\text{El IMS está dada por:} \quad \text{IMS} = (\text{MT} - (\text{Fc} + \text{Fa} + \text{Fe})) / \text{MT}$$



## 2.9 Certificación del software

Consecuencia de un proceso que es asegurar la calidad pero nunca es el objetivo final. La calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos deben ser correctos y estar en función de la normalización ISO 9000.

La normativa ISO 9000 pone a disposición de un auditor o certificador los procesos internos, de forma que este indique si cumple o no la normativa al 100%, audita el sistema. Si los resultados son positivos se emite la certificación y cada cierto tiempo se tiene que renovar. La certificación es costosa, a consecuencia de costes que ocasionan la lejanía y el tiempo de duración de proceso (aprox. 6 meses). Se certifica la empresa y la metodología para el desarrollo de la aplicación (CDS12).

## 2.10 Medición del software

En el software lo que se mide son atributos propios del mismo, se descompone un atributo general en otros más simples de medir, a veces se mide bien o mal ya que la descomposición del atributo genérico de calidad en otros sub-atributos se torna irreal, se mide con datos estadísticos no avalados, es imposible decir que la medición se hace en forma correcta (CAFU14).

El concepto de medida va de más a menos, va de lo general a lo concreto y lo concreto es asociado a la métrica, cuya combinación te daría el nivel de calidad o seguridad de tu producto. Las ciencias bien estructuradas se basan en medidas bien hechas, se basan en la matemática. Los tipos de medidas son:

- Número de errores durante un periodo determinado.
- Fallo en la codificación o diseño de un sistema que causa que el programa no funcione correctamente o falle.
- Tamaño de un producto informático (líneas de código).
- Métrica de punto función (IBM): relaciona funcionalidades que ofrece estimación de costes y esfuerzos (CAFU14).

## 2.7. Costo y beneficio

### 2.7.1 COCOMO

El Modelo Constructivo de Costos (COCOMO, por su acrónimo del inglés ConstructiveCostModel) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres sub-modelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado (COSC95).

Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en líneas de código principalmente.

#### Modelos de estimación

Las ecuaciones que se utilizan son:

$$E = a(Kl)^b * m(X), \text{ en persona-mes}$$

$$Tdev = c(E)^d, \text{ en meses}$$

$$P = E/Tdev, \text{ en personas}$$

Dónde:

E : es el esfuerzo requerido por el proyecto, en persona-mes

Tdev : es el tiempo requerido por el proyecto, en meses

P : es el número de personas requerido por el proyecto

a, b, c y d : son constantes con valores definidos en una tabla, según cada sub-modelo

Kl : es la cantidad de líneas de código, en miles.

m(X): Es un multiplicador que depende de 15 atributos (COSC95).

A la vez, cada sub-modelo también se divide en modos que representan el tipo de proyecto, y puede ser:

**Modo orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).

**Modo semilibre o semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

**Modo rígido o empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla (COSC95).

### Modelo básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla (ver tabla 3) de constantes para calcular distintos aspectos de costes:

**Tabla 3 Constantes para calcular distintos aspectos de costos**

Fuente: [COSC95]

<b>MODO</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>D</b>
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semi - Orgánico</b>	3.00	1.12	2.50	0.35
<b>Empotrado</b>	3.60	1.20	2.50	0.32

Estos valores son para las fórmulas:

Personas necesarias por mes para llevar adelante el proyecto (**MM**) =  $a \cdot (Kl)^b$

Tiempo de desarrollo del proyecto (**TDEV**) =  $c \cdot (MM)^d$

Personas necesarias para realizar el proyecto (**CosteH**) =  $MM/TDEV$

Costo total del proyecto (**CosteM**) = CosteH \* Salario medio entre los programadores y analistas.

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno (COS95).

### **Modelo intermedio**

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula son (ver tabla 4):

**Tabla 4 Constantes a reemplazar en la fórmula**

Fuente: [COS95]

<b>MODO</b>	<b>A</b>	<b>b</b>
<b>Orgánico</b>	3.20	1.05
<b>Semi – Orgánico</b>	3.00	1.12
<b>Empotrado</b>	2.80	1.20

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño, mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste (COS95).

### **Modelo Detallado**

Presenta principalmente dos mejoras respecto al anterior:

Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, y otros. Tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra (COSC95).

De esta manera haciendo el uso del software COCOMO y utilizándose las fórmulas mencionadas anteriormente, se logra estimar los costos, los recursos humanos y el tiempo, el software se encuentra publicado en la WEB y es de uso libre (COSC95).

## **2.8. Seguridad**

Siendo que la información es el activo más importante que posee cualquier sistema, deben existir técnicas más allá de la seguridad física, que la aseguren.

Estas técnicas las brinda la seguridad lógica (SSI12).

La seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y solo se permita acceder a ellos a las personas autorizadas para hacerlo. Para tal efecto se consideran los siguientes aspectos:

## **2.9 Controles de acceso**

Estos controles pueden implementarse en el sistema operativo, sobre los sistemas de aplicación, en base de datos, en un paquete específico de seguridad y en cualquier otro utilitario.

Constituyen una importante ayuda para proteger al sistema operativo de la red, al sistema de aplicación y demás software de la utilización o modificaciones no autorizadas; para mantener la integridad de la información (restringiendo la cantidad de usuarios y procesos con acceso permitido) y para resguardar la información confidencial de accesos no autorizados (SSI12).

## **2.10 Identificación y autenticación**

Es la primera línea de defensa para la mayoría de los sistemas computarizados, permitiendo prevenir el ingreso de personas no autorizadas. Es la base para la mayor parte de los controles de

acceso y para el seguimiento de las actividades de los usuarios.

Se denomina identificación al momento en que el usuario se da a conocer en el sistema; y autenticación a la verificación que realiza el sistema sobre esta identificación (SSI12).

### **2.11 Roles**

El acceso a la información también puede controlarse a través de la función o rol de usuario que requiere dicho acceso. Algunos ejemplos de roles serían los siguientes:

- Programador, líder de proyecto.
- Gerente de un área, Administrador del sistema.



#### 3.1 Introducción

Según la ingeniería de software, en 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 denomina ciclo de vida o proceso de desarrollo de software.

El presente proyecto enfoca la problemática que existe en la parte de control de personal, control de ventas y control de fallas técnicas que tiene la empresa.

Analizando los datos que tienen almacenados manualmente por tal motivo se desarrolla un sistema de control de personal, ventas y fallas técnicas, a fin de mejorar el control de las actividades de la empresa.

#### 3.2 Proceso de desarrollo

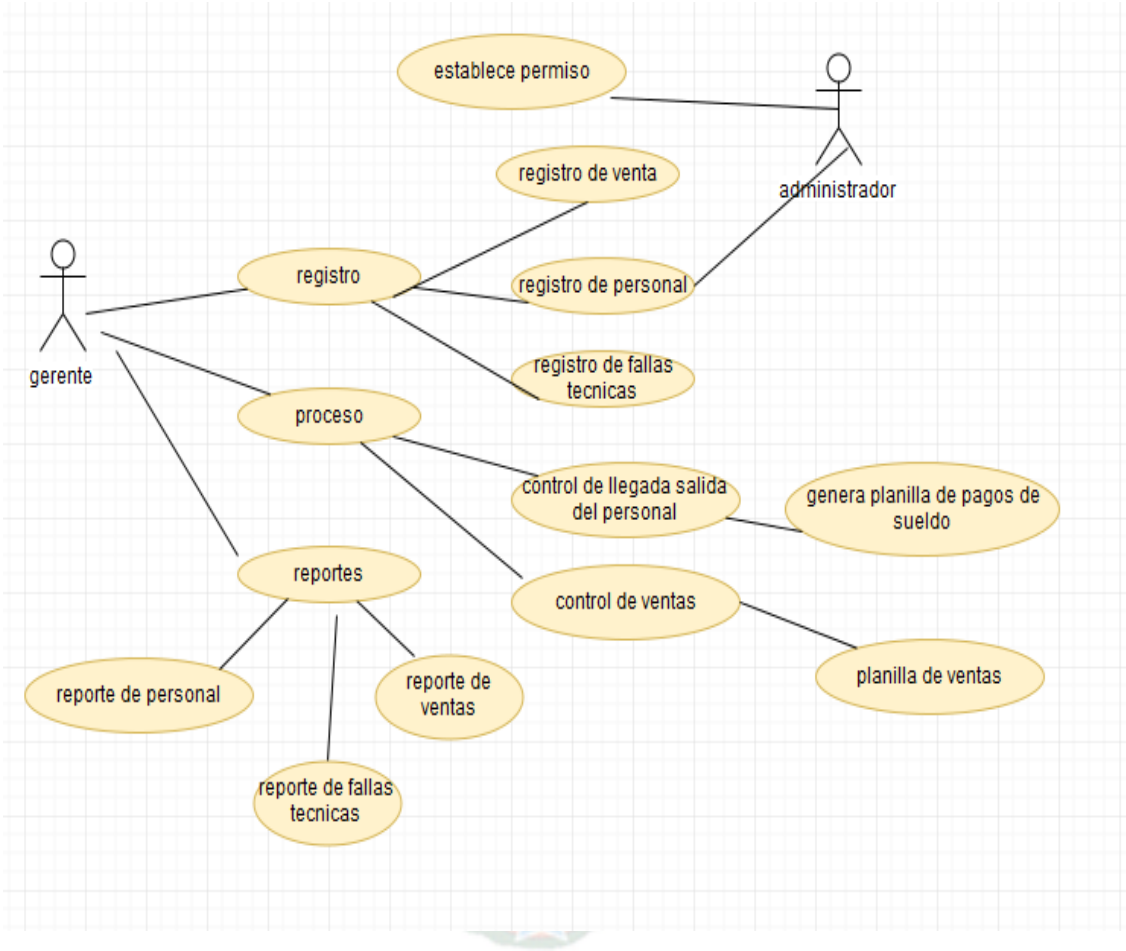
Para el proceso de desarrollo se utiliza un modelo de proceso incremental de SCRUM y se complementa con la metodología UWE para hacer el seguimiento del diseño, en la Figura 10 se puede apreciar gráficamente el modelo de proceso que se utiliza en el presente proyecto de grado.



**Figura 12 Etapas de Scrum**

Fuente [PROYSCRUM]

Antes de comenzar con la fase del Pre-game se observó los procesos y actividades que realizan en la empresa SISCOM el cual se muestra en la Figura 13 mediante un diagrama de casos de uso general del Sistema.



**Figura 13 Diagrama de casos de uso general del sistema**

Fuente: [Elaboración propia]

En la Tabla 5 se muestra la definición de los actores y sus respectivos roles para el Sistema.



**Tabla 5 Definición de actores**

Fuente: [Elaboración propia]

<b>ACTOR</b>	<b>ROLES O FUNCIONES</b>
Administrador	<ul style="list-style-type: none"><li>✓ Administrar usuarios</li><li>✓ Registrar personal</li><li>✓ Registrar ventas</li><li>✓ Registrar las fallas técnicas de la empresa</li><li>✓ Ver reportes</li><li>✓ Configuración de ventas</li></ul>
Gerente	<ul style="list-style-type: none"><li>✓ Registrar personal</li><li>✓ Registrar ventas</li></ul>

### 3.3 Pre- Game

#### 3.3.1 Planificación

En la planificación se creó la lista priorizada ProductBaklog a partir del análisis de requerimientos de la empresa SISCOM. Los requerimientos funcionales que describen las interacciones entre el sistema y el entorno son representados en la tabla 6.

**Tabla 6 Requerimientos**

Fuente: [Elaboración propia]

<b>CÓDIGO</b>	<b>REQUERIMIENTOS</b>
RF1	Diseñar la base de datos, que permita almacenar al personal, las ventas realizadas. Y un reporte de las fallas técnicas que tiene la empresa.
RF2	Se debe manejar roles de usuarios, grupos de usuarios y permisos
RF3	La inserción, modificación, eliminación del personal
RF4	El registro del personal de la empresa

RF5	Reporte de la planilla de control de cada personal
RF6	La inserción, modificación, eliminación de las ventas realizadas
RF7	Registro de las ventas que hace la empresa
RF8	Reporte de la ventas realizadas por día , mes
RF9	La inserción, modificación , eliminación de las fallas técnicas
RF10	Registro de las fallas técnicas que puede tener la empresa
RF11	Reporte de las fallas técnicas que tenía la empresa

Los requerimientos de hardware que permiten que el sistema funcione correctamente se describen en la tabla 7.

**Tabla 7 Requerimientos para instalar el sistema**

Fuente: [Elaboración propia]

<b>CÓDIGO</b>	<b>REQUERIMIENTO PARA LA INSTALACIÓN</b>
RH1	Pc de escritorio o laptop
RH2	Memoria 521 MB o superior
RH3	Espacio de disco duro 10 GB de 32 b
RH4	Procesador Pentium 4 o superior de 32 b
RH5	Pantalla, teclado, mouse, impresora compatible con el RH1

Los requerimientos de software que permite que el sistema funcione correctamente se describen en la tabla 8

**Tabla 8 Requerimiento de programas**

Fuente: [Elaboración propia]

<b>CÓDIGO</b>	<b>REQUERIMIENTO DE PROGRAMAS</b>
Rs1	SO Windows XP o superior de 32 b
Rs2	Microsoft Office 2007 o superior
Rs3	Java 1.7. o superior
Rs4	Adobe pdf

### 3.4 Game

#### 3.4.1. Primera iteración (SPRINT 1)

##### a) Reunión y planificación del sprint 1

**RF1:** Diseñar la base de datos, que permita almacenar al personal, las ventas realizadas. Y un reporte de las fallas técnicas que tiene la empresa.

##### b) Scrum diario del sprint 1

El desarrollo del Sprint mediante la aplicación de la metodología UWE del módulo de diseño de base de datos, a continuación se detalla la lista de tareas para el Sprint 1. Ver Tabla 9

**Tabla 9 Sprint 1**

Fuente:[Elaboración propia]

PRIMERA ITERACIÓN		SPRINT	INICIO	DURACIÓN
		1	02-01-2016	10 días
ID	Tareas	Tipo	días de trabajo	Estado
1.1	Analizar los requerimientos	Análisis	2	Terminado
1.2	Realizar el modelo de base de datos	Diseño	2	Terminado
1.3	Construir la base de datos en gestor de base de datos	Desarrollo	4	Terminado

c) Trabajo y desarrollo del sprint 1

Diagrama de base de datos ver figura 14.

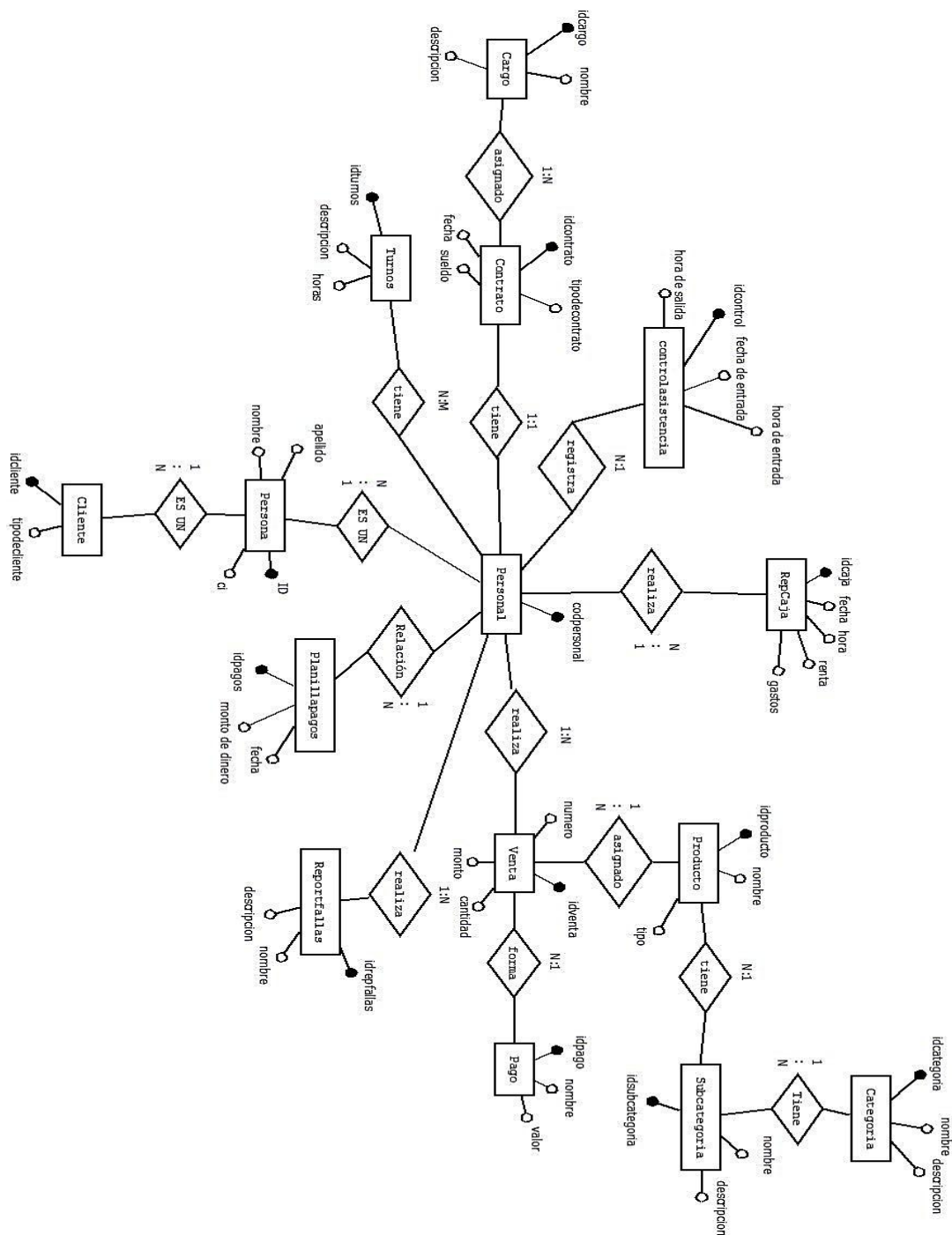


Figura 14 Modelo la base de datos

Fuente: [Elaboración propia]

## Modelo relacional de la base de datos

**Controlasistencia** (*idcontrol*, horaentrada, fecha,horasalida, **codpersonal**)

**RepCaja** (*idcaja*, fecha, hora, renta, gastos, **codpersonal**)

**Cargo** (*idcargo*, nombre, descripcion)

**Contrato** (*idcontrato*, tipodecontrato, fecha, sueldo, **idcargo**)

**Turnos** (*idturnos*, descripciondehoras)

**Persona** (*id*, ci, nombre, apellido)

**Cliente** (*idcliente*, tipodecliente, **id persona**)

**Planillapagos** (*idpagos*, monto, fecha, **codpersonal**)

**Personal**(*codpersonal*)

**Reportefallas** (*idrepfallas*, nombre, descripción, **codpersonal**)

**Venta** (*idventa*, numero, monto, cantidad, **codpersonal**, **idpago**, **idproducto**)

**Pago** (*idpago*, nombre, valor)

**Producto** (*idproducto*, nombre, tipo,**idsubcategoria**)

**Subcategoría** (*idsubcategoria*, nombre, descripción, **idcategoria**)

**Categoría** (*idcategoria*, nombre, descripción)

### 3.4.2. Segunda iteración (SPRINT 2)

#### a) Reunión y planificación del sprint 2

**RF2:** Se debe manejar roles de usuarios, grupos de usuarios y permisos

#### b) Scrum diario del sprint 2

El desarrollo del Sprint mediante la aplicación de la metodología UWE del módulo de administración de usuarios y autenticación de usuario, a continuación se detalla la lista de tareas para el Sprint 2. Ver Tabla 10

**Tabla 10 Sprint 2**  
Fuente:[Elaboración propia]

PRIMERA ITERACIÓN		SPRINT	INICIO	DURACIÓN
		2	02-02-2016	20 días
ID	Tareas	Tipo	días de trabajo	Estado
2.1	Análisis de requerimientos para la administración de usuarios y autenticación de usuario	Análisis	1	Terminado
2.2	Diseñar el modelado de administración de usuario y autenticación	Diseño	1	Terminado
2.3	Diseñar la interface para la administración de usuarios	Diseño	1	Terminado
2.4	Diseñar la interface para la autenticación de usuario	Diseño	3	Terminado
2.5	Desarrollar el módulo de administración de usuarios	Desarrollo	4	Terminado
2.6	Desarrollar el módulo de autenticación de usuario	Desarrollo	5	Terminado
2.7	Pruebas del módulo de administración de usuarios y módulo de autenticación de usuarios		1	Terminado

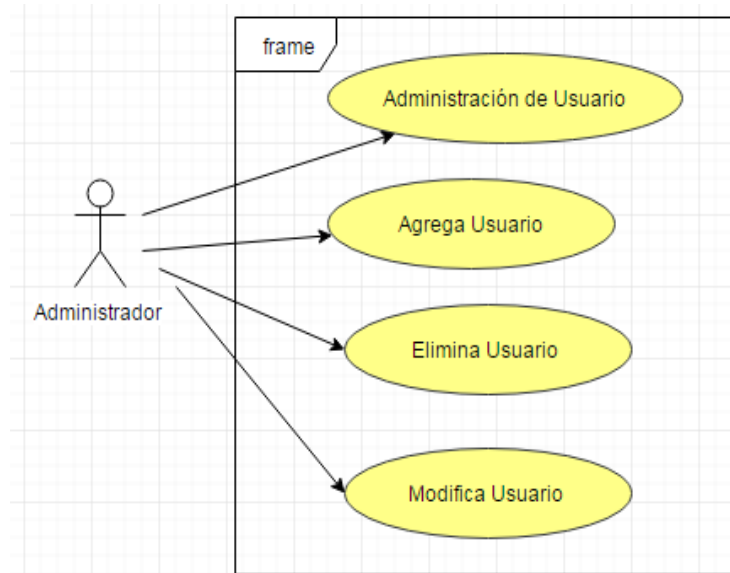
En el Sprint 2 se desarrollaron las siguientes funcionalidades para el Sistema:

- Administración de la base de datos
- Administración de usuarios
- Autenticación de usuario

**c) Trabajo y desarrollo del sprint 2**

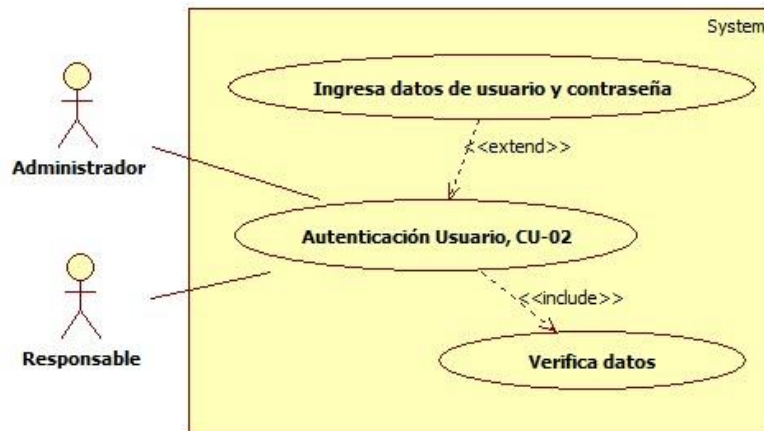
**Modelo de requerimientos sprint 2**

En el modelo de requerimientos para el Sprint 1 se elabora los casos de uso que describen el proceso de administración de usuario y la autenticación de usuario. Ver Figura 13 y 14.



**Figura 15 Diagrama de casos de uso- Administración de usuario**

Fuente:[Elaboración propia]



**Figura 16 Diagrama de casos de uso - Autenticación de usuario**

Fuente:[Elaboración propia]

La descripción extendida de los casos de uso para proceso de administración de usuarios y autenticación de usuario se detalla en la Tabla 11 y Tabla 12

**Tabla 11 Detalle de casos de uso Administración de usuario**

Fuente:[Elaboración propia]

<b>NOMBRE</b>	Administración usuarios	<b>CÓDIGO:</b> CU-01
<b>ACTORES</b>	Administrador	
<b>DESCRIPCIÓN</b>	Se describe el proceso de registro, modificación y eliminación de usuarios	
<b>FLUJO PRINCIPAL</b>	<b>EVENTO ACTOR</b>	<b>EVENTO SISTEMA</b>
	<ul style="list-style-type: none"> <li>• Digita los datos de usuario</li> <li>• Solicitar nuevo registro para usuario</li> <li>• Solicitar modificación de datos de usuario</li> <li>• Solicitar eliminación de usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Validar datos del usuario</li> <li>• Registra nuevo usuario</li> <li>• Registra actualización de datos de usuario</li> <li>• Elimina usuario</li> </ul>
<b>PRECONDICIÓN</b>	El administrador debe haber iniciado sesión	
<b>POST-CONDICIÓN</b>	Procesos de administración de usuarios realizado con éxito	
<b>PRESUNCIÓN</b>	El que realiza la administración de usuarios debe ser el Administrador del sistema	

**Tabla 12 Detalle de caso de uso Autenticación de usuario**

Fuente:[Elaboración propia]

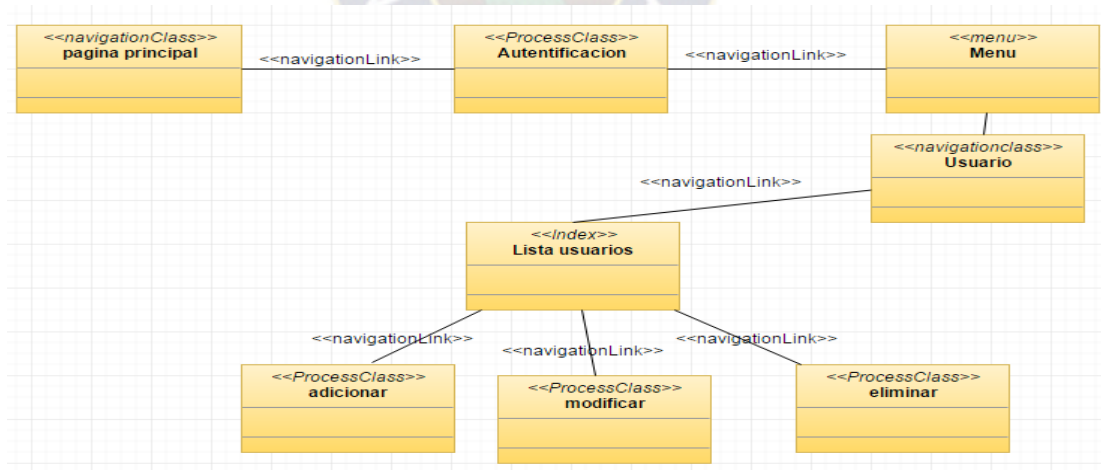
<b>NOMBRE</b>	Autenticación de usuario	<b>CÓDIGO:</b> CU-02
<b>ACTORES</b>	Administrador o Responsable	
<b>DESCRIPCIÓN</b>	Se describe el proceso de autenticación de usuario	
<b>FLUJO PRINCIPAL</b>	<b>EVENTO ACTOR</b>	<b>EVENTO SISTEMA</b>
	<ul style="list-style-type: none"> <li>• Ingresa al sistema</li> <li>• Digita el nombre de usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Despliega formulario de autenticación</li> <li>• Verifica datos</li> </ul>



	<ul style="list-style-type: none"> <li>• Digita la contraseña de usuario</li> <li>• Solicita ingresar al sistema</li> </ul>	<ul style="list-style-type: none"> <li>• ingresados</li> <li>• Verifica que el usuario este registrado en la base de datos del sistema</li> </ul>
<b>PRECONDICIÓN</b>	El administrador o responsable debe haber iniciado sesión	
<b>POST-CONDICIÓN</b>	Ingreso de forma correcta al usuario	
<b>PRESUNCIÓN</b>	El que ingresa al sistema debe estar registrado como usuario en la base de datos del sistema	

### Modelo de navegación para la administración de usuarios

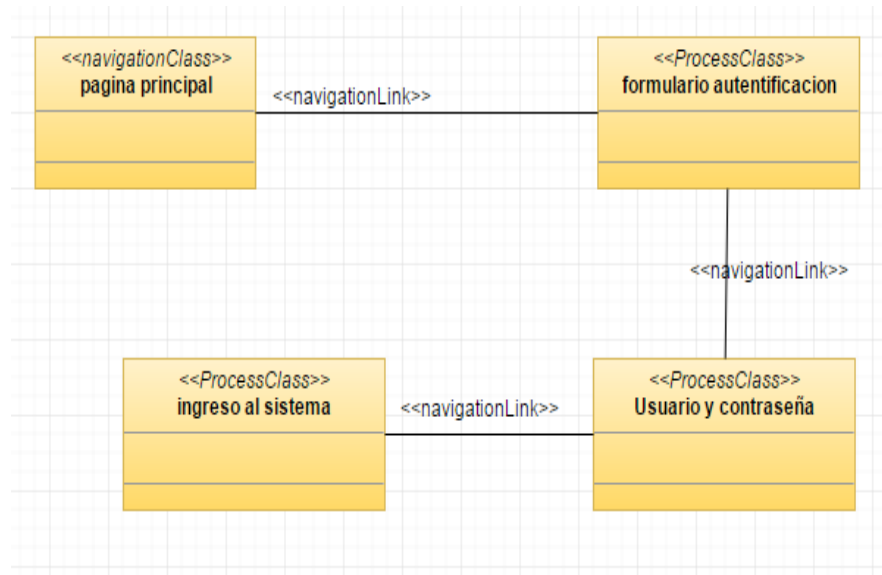
En la Figura 17 se puede observar el diseño de navegación para el módulo de administración de usuarios. Donde se muestra el panel de navegación del administrador del sistema.



**Figura 17 Diagrama de navegación Administración de usuarios**

Fuente [Elaboración propia]

En la Figura 18 se muestra el diseño de navegación para la autenticación de usuario este usuario puede ser el administrador o responsable de la empresa SISCOM.

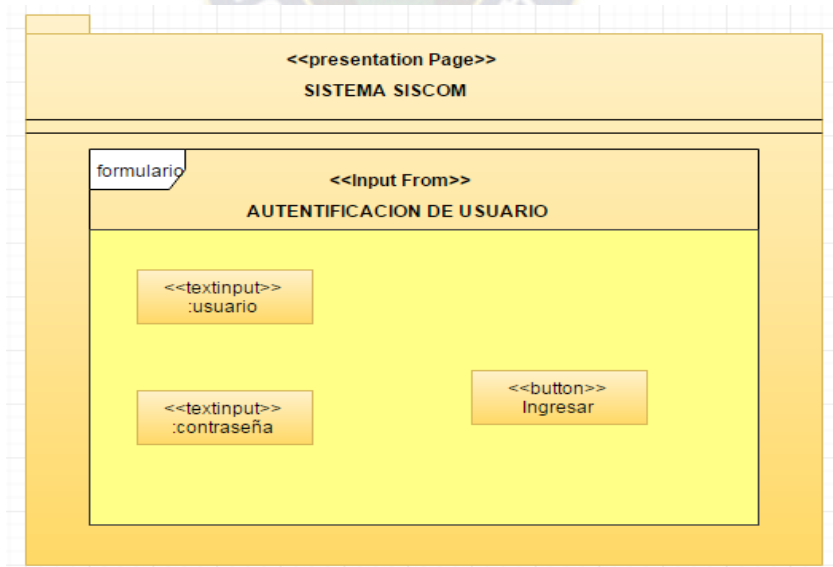


**Figura 18 Diagrama de navegación Autenticación**

Fuente [Elaboración propia]

### Modelo de presentación para la autenticación de usuario

En la Figura 19 se puede observar todos los componentes de la interfaz gráfica para la autenticación de usuario registrados en el Sistema Web.



**Figura 19 Modelo de presentación - Autenticación de usuario**

Fuente [Elaboración propia]

#### d) Revisión sprint 2

En esta etapa se revisó el cumplimiento de las tareas planificadas, como se puede observar en la Tabla 13

Para verificar el producto entregable del Sprint 2, se realizaron las siguientes pruebas de funcionalidad.

**Tabla 13 Pruebas de funcionalidad - Inicio de sesión**

Fuente: [Elaboración propia]

<b>PRUEBA 1.1</b>	<b>OPERACIÓN: INICIO DE SESIÓN</b>
<b>Precondición:</b> <ul style="list-style-type: none"><li>- Conexión a la área local</li><li>- Estar registrado en la base de datos del Sistema web</li></ul>	
<b>Datos/Proceso:</b> <ul style="list-style-type: none"><li>- Digitar el nombre de usuario</li><li>- Digitar la contraseña</li><li>- Presionar el botón ingresar</li></ul>	
<b>Resultados esperados:</b> <p>En caso de iniciar sesión con éxito, dependiendo a la jerarquía de usuario lo llevara a la página principal del Sistema Web.</p> <p>En caso del ingreso de datos incorrecto o datos no registrados en la base de datos, el sistema debe mostrar el mensaje de verificar datos.</p>	
<b>Post-condición:</b> <p>Inicio de sesión con éxito</p>	
<b>Resultados obtenidos:</b> <p>Se obtuvieron los resultados esperados</p>	

Las pruebas de funcionalidad que se hicieron para el registro de usuarios se muestran en la Tabla 14.

**Tabla 14 Pruebas de funcionalidad - Registro de usuario**

Fuente: [Elaboración propia]

<b>PRUEBA 1.2</b>	<b>OPERACIÓN: REGISTRO DE USUARIOS</b>
<b>Precondición:</b>	
<ul style="list-style-type: none"> <li>- Conexión a la área local</li> <li>- El usuario debe haber iniciado sesión</li> </ul>	
<b>Datos/Proceso:</b>	
<ul style="list-style-type: none"> <li>- Presionar en el menú la opción de configuraciones y luego usuarios</li> <li>- Digitar todos los datos en el formulario</li> <li>- Presionar el botón guardar</li> <li>- Verificar el registro de usuarios en la lista de usuarios</li> </ul>	
<b>Resultados esperados:</b>	
<p>En caso de registro de usuario con éxito se actualiza automáticamente la lista de usuarios.                  En caso de que en el proceso de registro de usuario se ingresen datos incorrectos se mostrara un mensaje de alerta.</p>	
<b>Post-condición:</b>	
<ul style="list-style-type: none"> <li>- Registro de usuario con éxito</li> <li>- Actualización automática de la lista de usuarios</li> </ul>	
<b>Resultados obtenidos:</b>	
<p>Registra usuarios de manera exitosa y muestra la lista de usuarios.</p>	

Las pruebas de funcionalidad para el proceso de administración de usuarios se detallan en la Tabla 15.

**Tabla 15 Pruebas de funcionalidad - Administración de usuario**

Fuente: [Elaboración propia]

<b>PRUEBA 1.3</b>	<b>OPERACIÓN: ADMINISTRACIÓN DE USUARIOS</b>
<b>Precondición:</b>	

<ul style="list-style-type: none"> <li>- Conexión a la área local</li> <li>- El usuario debe haber iniciado sesión</li> </ul>
<p><b>Datos/Proceso:</b></p> <ul style="list-style-type: none"> <li>- Presionar en el menú la opción de configuraciones y luego usuarios</li> <li>- Seleccionar usuario para la actualización de datos o para dar de baja</li> <li>- Presionar el botón de modificar o el botón de dar de baja</li> </ul>
<p><b>Resultados esperados:</b></p> <p>En caso de actualizar los datos de usuarios el sistema debe verificar que los datos sean ingresados de forma correcta y mostrar la lista actualizada de usuarios</p> <p>En caso de dar de baja al usuario el sistema debe modificar el estado de usuario a inactivo</p> <p>En caso de problemas en la administración de sistemas el sistema muestra un mensaje de alerta.</p>
<p><b>Post-condición:</b></p> <ul style="list-style-type: none"> <li>- Actualización de datos de usuario de forma exitosa</li> <li>- Dar baja a usuario realizado con éxito</li> <li>- Lista actualizada de usuarios activos</li> </ul>
<p><b>Resultados obtenidos:</b></p> <p>Realiza el proceso de administración de usuarios de manera exitosa.</p>

### 3.4.3. Tercera iteración (SPRINT 3)

#### a) Reunión y planificación del sprint 3

**RF3:** La inserción, modificación, eliminación del personal

**RF4:** El registro del personal de la empresa

**RF5:** Reporte de la planilla de control de cada personal

## b) Scrum diario del sprint 3

Para el Sprint 3 se realizó el análisis de los requerimientos y el modelado del módulo de control de personal. En la Tabla 15 se lista las tareas planificadas para el Sprint 3.

**Tabla 16 Sprint 3**

Fuente: [Elaboración propia]

SEGUNDA ITERACIÓN		Sprint	Inicio	Duración
		3	03-03-2016	13 días
ID	Tareas	Tipo	días de trabajo	
3.1	Análisis de requerimientos para el módulo de registro del personal	Análisis	1	Terminado
3.2	Construir el modelado para el módulo de registro del personal	Diseño	1	Terminado
3.3	Diseñar la interfaz para el registro del nuevo personal	Diseño	4	Terminado
3.4	Desarrollar el módulo de registro de personal	Desarrollo	6	Terminado
3.5	Pruebas del módulo del registro al personal	Pruebas	1	Terminado

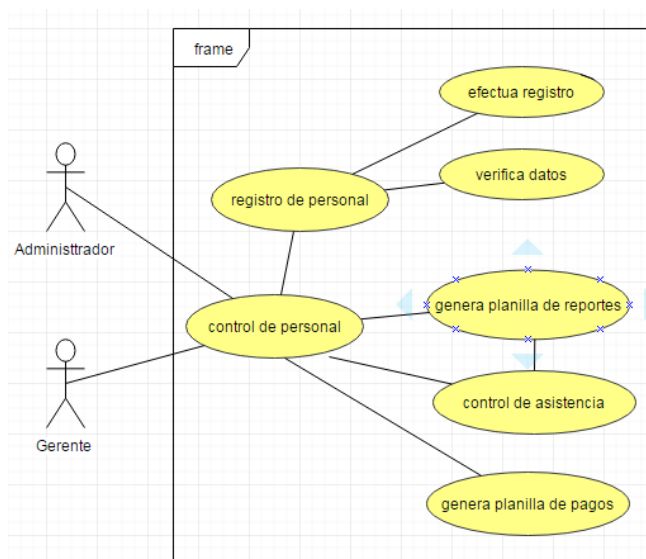
En el Sprint 3 se desarrolló el módulo de control de personal con las siguientes funcionalidades:

- Registro del personal
- Adiciona al personal
- Modifica y elimina

## c) Trabajo y desarrollo del sprint 3

### Modelo de requerimientos sprint 3

En el modelo de requerimientos para el Sprint 3 se elabora los casos de uso que describen el proceso de control de personal. Ver Figura 20.



**Figura 20 Diagrama de casos de uso – control de personal**

Fuente [Elaboración propia]

La descripción extendida de los casos de uso para proceso de control de personal se detalla en la Tabla 17.

**Tabla 17 Detalle de casos de uso Administración de usuario**

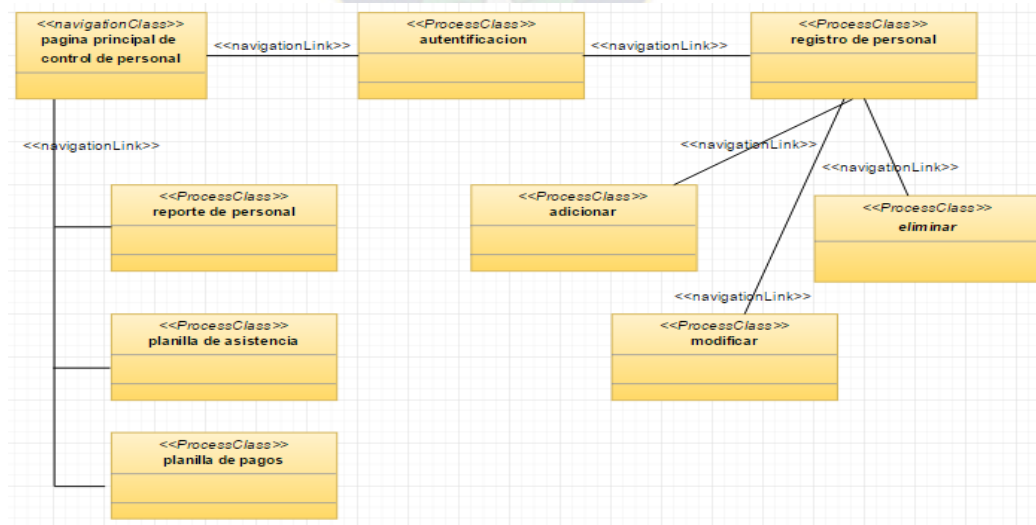
Fuente [Elaboración propia]

<b>NOMBRE</b>	Administración usuarios	<b>CÓDIGO:</b> CU-01
<b>ACTORES</b>	Administrador	
<b>DESCRIPCIÓN</b>	Se describe el proceso de control de personal viendo las horas de llega salida y fecha también está la parte de la planilla de pagos	
<b>FLUJO PRINCIPAL</b>	<b>EVENTO ACTOR</b>	<b>EVENTO SISTEMA</b>
	<ul style="list-style-type: none"> <li>Entre a la página de</li> </ul>	<ul style="list-style-type: none"> <li>Valida sus datos para</li> </ul>

	control de personal <ul style="list-style-type: none"> <li>• Efectúa su registro de asistencia</li> <li>• Por otro lado también está la parte de control de planilla de pagos</li> </ul>	el registro de asistencia <ul style="list-style-type: none"> <li>• Valida datos para la planilla de pagos</li> </ul>
<b>PRECONDICIÓN</b>	El administrador debe haber iniciado sesión	
<b>POST-CONDICIÓN</b>	Procesos de control de personal es realizado con éxito	
<b>PRESUNCIÓN</b>	El que realiza el control de personal debe ser el Administrador del sistema	

### Modelo de navegación para la administración de usuarios

En la Figura 21 se puede observar el diseño de navegación para el módulo de control de personal. Donde se muestra el panel de navegación del administrador del sistema.



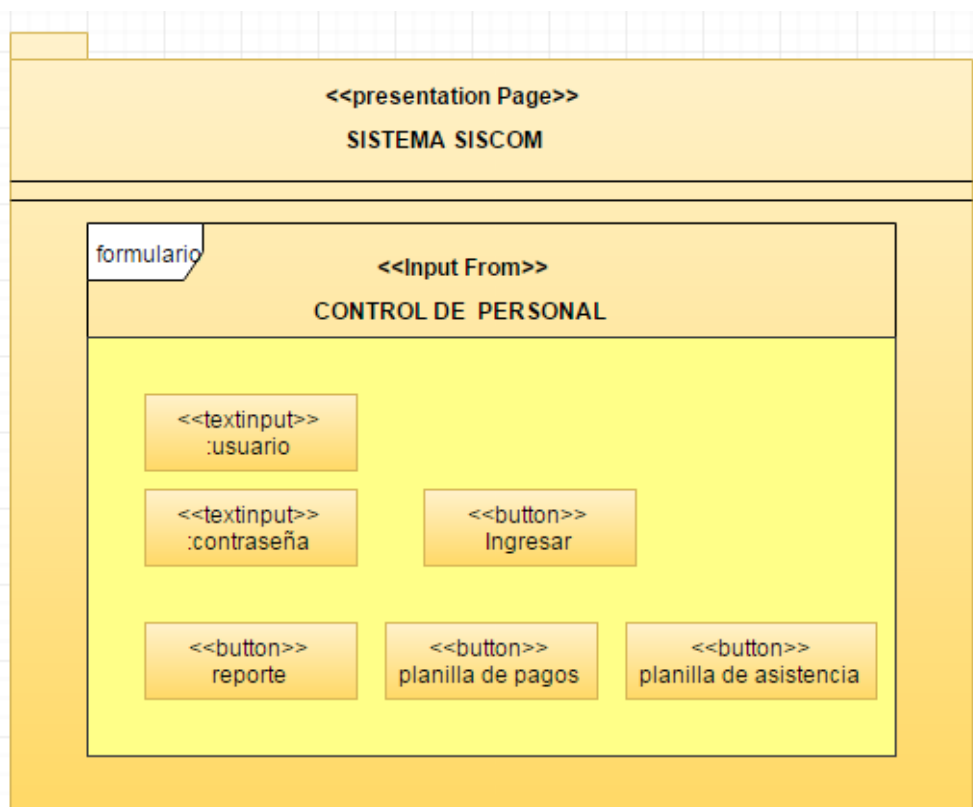
**Figura 21 Diagrama de navegación de control de personal**

Fuente [Elaboración propia]



## Modelo de presentación para la autenticación de usuario

En la Figura 22 se puede observar todos los componentes de la interfaz gráfica para el control de personal en el Sistema Web.



**Figura 22 Modelo de presentación – control de personal**

Fuente [Elaboración propia]

### d) Revisión sprint 3

En esta etapa se revisó el cumplimiento de las tareas planificadas en el Sprint 3.

Las pruebas de funcionalidad del módulo de registro de personal se detallan en la Tabla 18

**Tabla 18 Pruebas de funcionalidad – registro de personal**

Fuente [Elaboración propia]

<b>PRUEBA 2.1</b>	<b>OPERACIÓN: Registro al personal</b>
<b>Precondición:</b> <ul style="list-style-type: none"><li>• Conexión a la área local</li><li>• Iniciar sesión</li></ul>	
<b>Datos/Proceso:</b> <ul style="list-style-type: none"><li>• Presionar en la parte de la configuración</li><li>• Entrar a personal</li><li>• Presionar el botón adicional nuevo personal</li></ul>	
<b>Resultados esperados:</b> <p>Se registra al personal , en caso de no cumplir algún requisito para el registro del personal muestra un mensaje</p>	
<b>Post-condición:</b> <p>Registro de personal realizado con éxito.</p>	
<b>Resultados obtenidos:</b> <p>Registro de personal realizado manera exitosa.</p>	

#### **3.4.4. Cuarta iteración (SPRINT 4)**

##### **a) Reunión y planificación del sprint 4**

**RF6:** La inserción, modificación, eliminación de las ventas realizadas

**RF7:** Registro de las ventas que hace la empresa

**RF8:** Reporte de la ventas realizadas por día, mes

##### **b) Scrum diario del sprint 4**

A continuación se muestra en la Tabla 20 las tareas planificadas para el Sprint 4

**Tabla 19 Sprint 4**

Fuente [Elaboración propia]

<b>TERCERA ITERACIÓN</b>		<b>Sprint</b>	<b>Inicio</b>	<b>Duración</b>
		<b>4</b>	<b>22-04-16</b>	<b>19 días</b>
<b>ID</b>	<b>Tareas</b>	<b>Tipo</b>	<b>días de trabajo</b>	<b>Estado</b>
4.1	Análisis de requerimientos para el módulo de control de ventas	Análisis	2	Terminado
4.2	Construir el modelado para el módulo de control de ventas	Diseño	2	Terminado
4.3	Diseñar la interfaz del módulo de control de ventas	Diseño	5	Terminado
4.4	Desarrollo del módulo de control de ventas	Desarrollo	9	Terminado
4.5	Pruebas del módulo del control de ventas	Prueba	1	Terminado

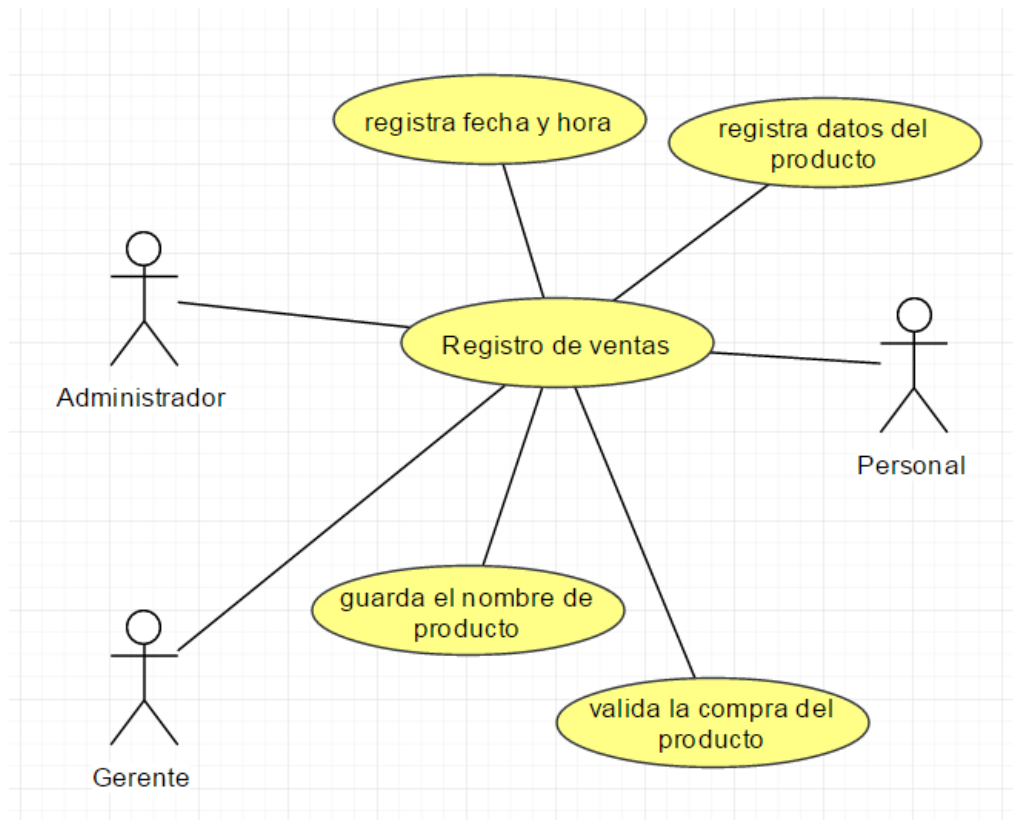
En el Sprint 4 se desarrollaron las siguientes funcionalidades para el Sistema:

- Búsqueda de ventas realizadas
- Registro de ventas de productos

#### **c) Trabajo y desarrollo del sprint 4**

##### **Modelo de requerimientos sprint 4**

En el modelo de requerimientos para el control de ventas, se tiene la descripción y el comportamiento del sistema cuando se realiza el control de ventas, se elabora el caso de uso que describe el proceso de control de ventas .Ver Figura 23



**Figura 23 Diagrama de casos de uso - control de ventas**

Fuente [Elaboración propia]

La descripción extendida de los casos de unos para proceso de venta se detalla en la Tabla 20

**Tabla 20 Detalle caso de control de ventas**

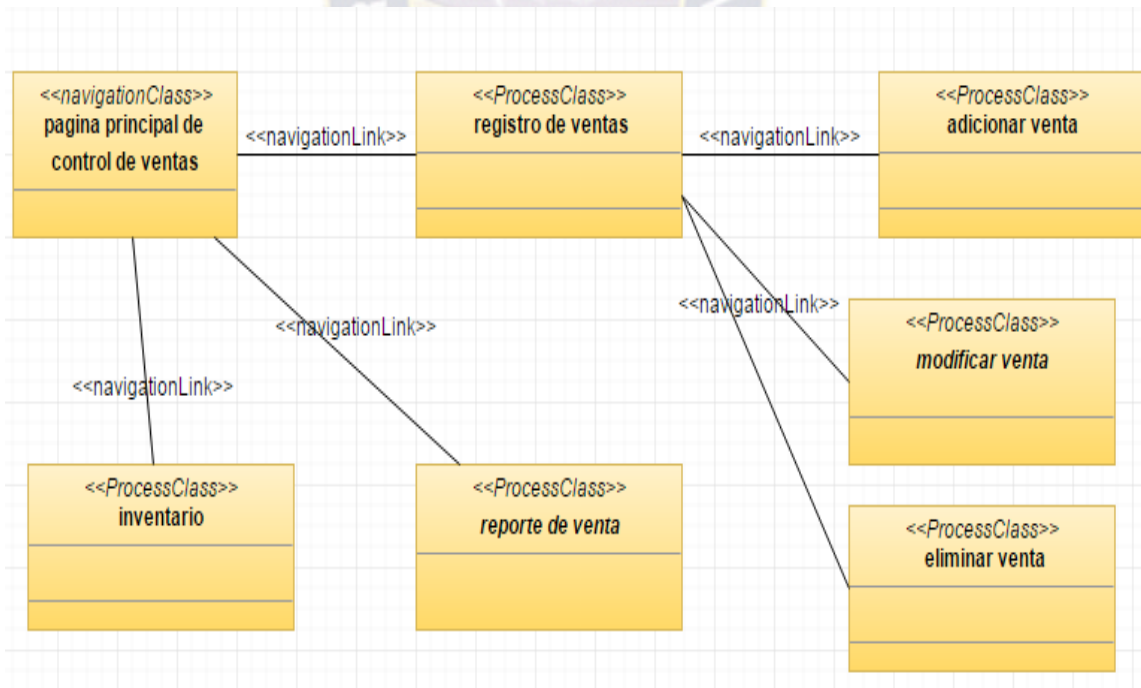
Fuente [Elaboración propia]

<b>NOMBRE</b>	Control de ventas	<b>CÓDIGO:</b> CU-03
<b>ACTORES</b>	Administrador, gerente y personal	
<b>DESCRIPCIÓN</b>	Se describe el proceso de venta	
<b>FLUJO PRINCIPAL</b>	<b>EVENTO ACTOR</b>	<b>EVENTO SISTEMA</b>
	<ul style="list-style-type: none"> <li>• Ingresa a la pantalla de ventas</li> <li>• Registra el producto que es vendido</li> <li>• Digita el personal nombre</li> </ul>	<ul style="list-style-type: none"> <li>• Despliega formulario de registro de ventas</li> <li>• Valida los datos ingresados</li> <li>• Guarda la venta</li> </ul>

	<p>del personal quien está haciendo la venta</p> <ul style="list-style-type: none"> <li>• Solicita la validación de los datos del producto.</li> <li>• Solicite que se guarde dicha compra</li> </ul>	
<b>PRECONDICIÓN</b>	El administrador o responsable debe haber iniciado sesión	
<b>POST-CONDICIÓN</b>	Venta guardada de manera exitosa	
<b>PRESUNCIÓN</b>	La venta debe ser guardada en la base de datos y el personal debe iniciar sesión.	

### Modelo de navegación para la administración de usuarios

En la Figura 24 se puede observar el diseño de navegación para el módulo de control de ventas. Donde se muestra el panel de navegación del administrador del sistema.

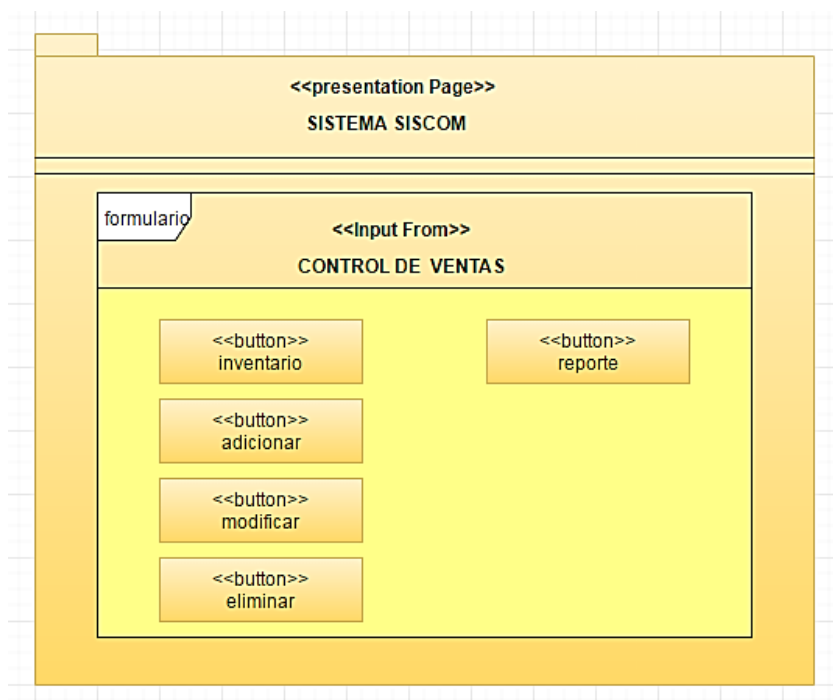


**Figura 24 Diagrama de navegación de control de ventas**

Fuente [Elaboración propia]

## Modelo de presentación para la autenticación de usuario

En la Figura 25 se puede observar todos los componentes de la interfaz gráfica para la el control de ventas en el Sistema Web.



**Figura 25 Modelo de presentación – control de ventas**

Fuente [Elaboración propia]

### d) Revisión sprint 4

Las pruebas de funcionalidad del módulo de registro de ventas se detallan en la Tabla 17

**Tabla 21 Pruebas de funcionalidad – registro de ventas**

Fuente [Elaboración propia]

<b>PRUEBA 2.1</b>	<b>OPERACIÓN: Registro de ventas</b>
<b>Precondición:</b> <ul style="list-style-type: none"><li>- Conexión a la área local</li><li>- Iniciar sesión</li></ul>	
<b>Datos/Proceso:</b> <ul style="list-style-type: none"><li>- Presionar en la parte de la configuración</li></ul>	

<ul style="list-style-type: none"> <li>- Entrar a la parte de ventas</li> <li>- Presionar el botón adicionar</li> </ul>
<p><b>Resultados esperados:</b></p> <p>Se registra la venta , en caso de no cumplir algún requisito para el registro de las ventas muestra un mensaje</p>
<p><b>Post-condición:</b></p> <ul style="list-style-type: none"> <li>- Registro de ventas realizado con éxito.</li> </ul>
<p><b>Resultados obtenidos:</b></p> <p>Registro de ventas realizadas de manera exitosa.</p>

### 3.4.2. Quinta Iteración (SPRINT 5)

#### a) Reunión y planificación del sprint 5

**RF9:** La inserción, modificación, eliminación de las ventas realizadas

**RF10:** Registro de las ventas que hace la empresa

**RF11:** Reporte de la ventas realizadas por día, mes

#### b) Scrum diario del sprint 5

En la Tabla 22 se puede observar las tareas planificadas en el Sprint 4.

**Tabla 22 Sprint 5**

Fuente [Elaboración propia]

CUARTA ITERACIÓN		Sprint	Inicio	Duración
		5	1-05-16	16 días
ID	Tareas	tipo	días de trabajo	Estado
5.1	Análisis de requerimientos para el módulo de control de fallas	Análisis	6	Terminado

5.2	Construir el modelado para el módulo de control de fallas	Diseño	4	Terminado
5.3	Diseñar la interfaz para el módulo del módulo de fallas técnicas	Diseño	4	Terminado
5.4	Pruebas del módulo de fallas técnicas y reportes	Prueba	2	Terminado

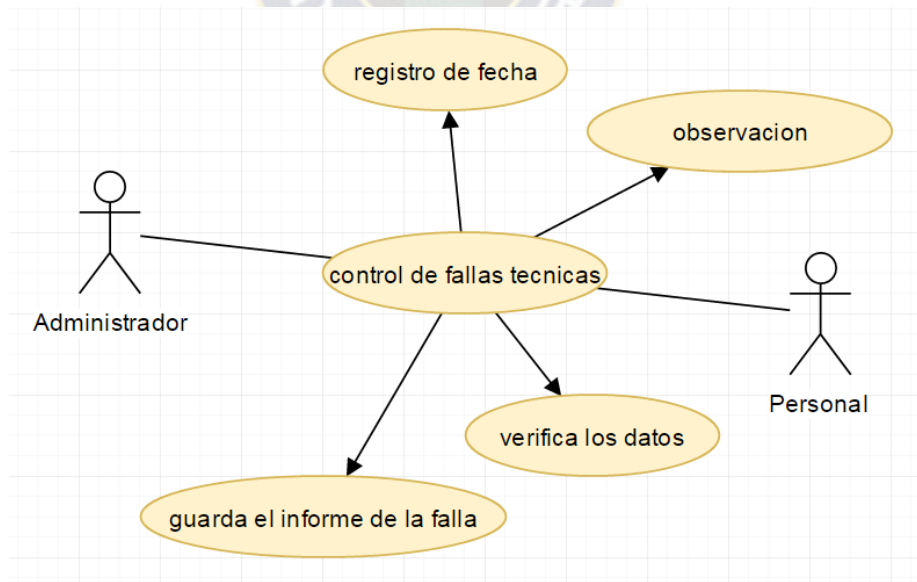
En el Sprint 4 se desarrollaron las siguientes funcionalidades para el Sistema:

- Administración de fallas técnicas
- Registro de las fallas técnicas
- Generación de reportes

**c) Trabajo y desarrollo del sprint 5**

**Modelo de requerimientos sprint 5**

En el modelo de requerimientos para el control de fallas técnicas , se tiene la descripción y el comportamiento del sistema cuando se realiza el control de fallas técnicas de la empresa elaborando el siguiente diagrama de casos de uso. Ver Figura 26



**Figura 26 Diagrama de casos de uso – control de fallas técnicas**

Fuente [Elaboración propia]



La descripción extendida de los casos de unos para proceso de control de fallas técnicas se detalla en las siguientes tablas. Ver Tabla 23

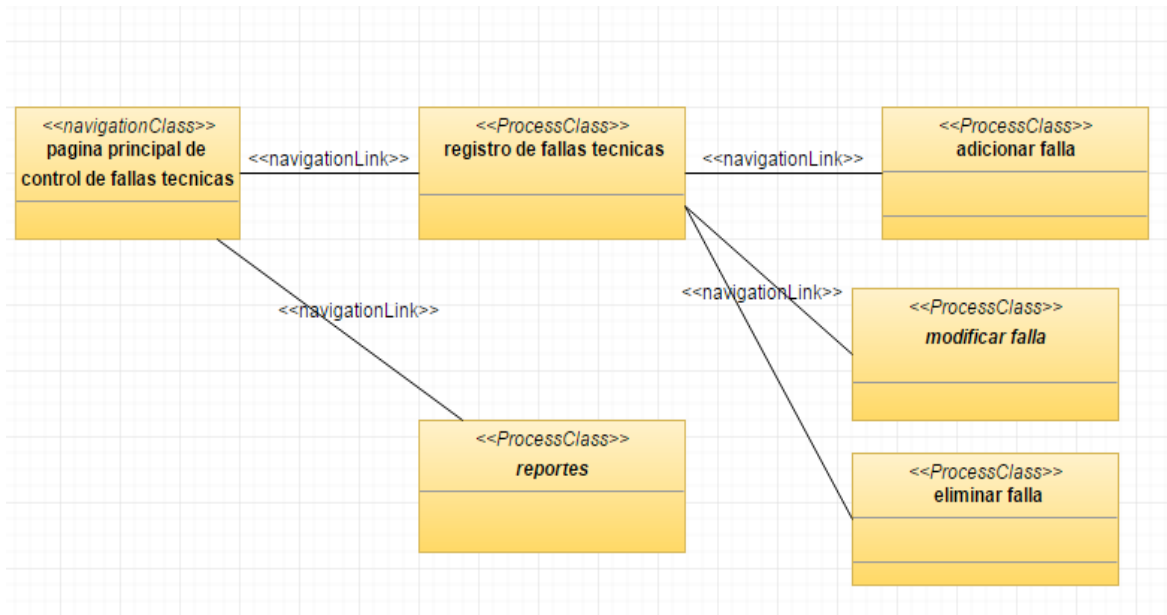
**Tabla 23 Detalle caso de uso control de fallas técnicas**

Fuente [Elaboración propia]

<b>NOMBRE</b>	Control de fallas técnicas	<b>CÓDIGO:</b> CU-04
<b>ACTORES</b>	Administrador , personal	
<b>DESCRIPCIÓN</b>	Se describe el proceso de control de fallas técnicas	
<b>FLUJO PRINCIPAL</b>	<b>EVENTO ACTOR</b>	<b>EVENTO SISTEMA</b>
	<ul style="list-style-type: none"> <li>• Ingresar al formulario</li> <li>• Añadir los datos en el formulario de registro de fallas técnicas</li> <li>• Introducir al personal que está reportando la falla técnica</li> </ul>	<ul style="list-style-type: none"> <li>• Abrir el formulario de registro de fallas técnicas</li> <li>• Introducir los datos que solicita el sistema.</li> <li>• Introducir el nombre del personal que está reportando la falla.</li> <li>• Guardar la falla técnica en la base de datos.</li> </ul>
<b>PRECONDICIÓN</b>	El administrador o el personal de turno es responsable de reportar la falla	
<b>POST-CONDICIÓN</b>	Reporte de fallas técnicas guardada con éxito	
<b>PRESUNCIÓN</b>	El que realiza el proceso de fallas técnicas debe estar registrada en la base de datos con responsabilidad del personal de turno o el administrador.	

### Modelo de navegación para la administración de usuarios

En la Figura 27 se puede observar el diseño de navegación para el módulo de control de fallas técnicas. Donde se muestra el panel de navegación del administrador del sistema.

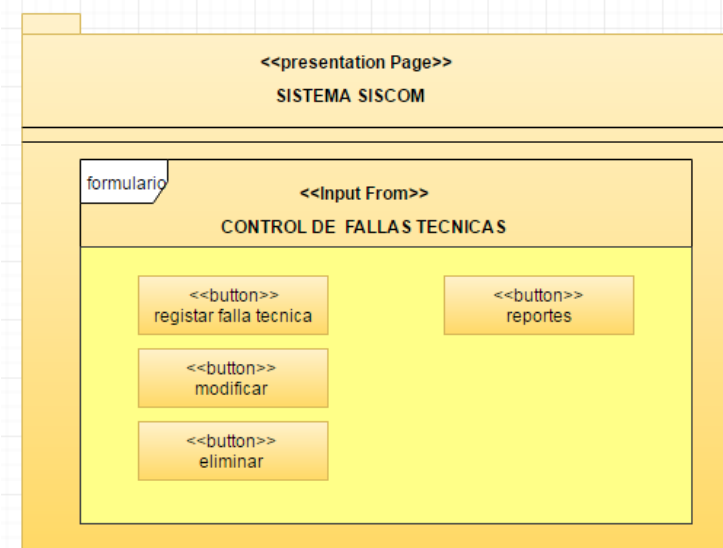


**Figura 27 Diagrama de navegación de control de personal**

Fuente [Elaboración propia]

### Modelo de presentación para la autenticación de usuario

En la Figura 28 se puede observar todos los componentes de la interfaz gráfica para el control de fallas técnicas en el Sistema Web.



**Figura 28 Modelo de presentación - Autenticación de usuario**

Fuente [Elaboración propia]

#### d) Revisión sprint 5

Tabla 24 Pruebas de funcionalidad – registro de fallas técnicas

Fuente [Elaboración propia]

<b>PRUEBA 2.1</b>	<b>OPERACIÓN: Registro de fallas técnicas</b>
<b>Precondición:</b> <ul style="list-style-type: none"><li>• Conexión a la área local</li><li>• Iniciar sesión</li></ul>	
<b>Datos/Proceso:</b> <ul style="list-style-type: none"><li>• Presionar en la parte de la configuración</li><li>• Entrar a registro de fallas técnicas</li><li>• Presionar el botón adicional fallas técnicas</li></ul>	
<b>Resultados esperados:</b> <p>Se registrar las fallas técnicas , en caso de no cumplir algún requisito para el registro de las fallas técnicas muestra un mensaje</p>	
<b>Post-condición:</b> <ul style="list-style-type: none"><li>- Registro de fallas técnicas realizado con éxito</li></ul>	
<b>Resultados obtenidos:</b> <p>Registro de fallas técnicas realizado de manera exitosa.</p>	

### 3.5 Post- Game

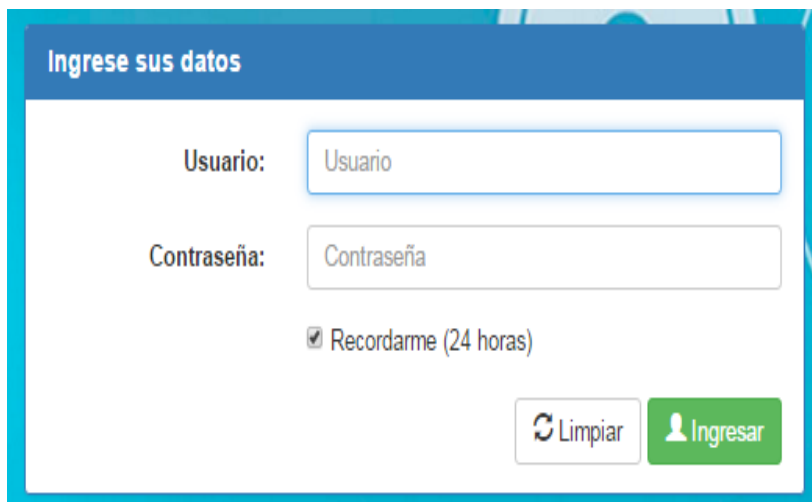
Durante cada Sprint se hace el cierre y la entrega de los productos. En esta fase se muestran todos los productos entregables que resultado de cada Sprint.

#### 3.5.1 Diseño de las interfaces graficas

El diseño de interfaces graficas fue desarrollado en cada iteración y las pruebas de funcionalidad fueron desarrolladas durante la revisión de cada Sprint. A continuación se listara las interfaces graficas del Sistema Web.

### a) Interface: Inicio de sesión

La Figura 29 muestra el panel de inicio de sesión, pueden iniciar sesión todos los usuarios activos y registrados en la base de datos, deben ingresar el nombre de usuario y su contraseña.



**Figura 29 Interface – Inicio de sesión**

Fuente: [Elaboración propia]

### b) Interface: Página de inicio del sistema

Una vez que el usuario inicie sesión de manera correcta el sistema muestra la pantalla de inicio de sistema como se muestra en la Figura 30

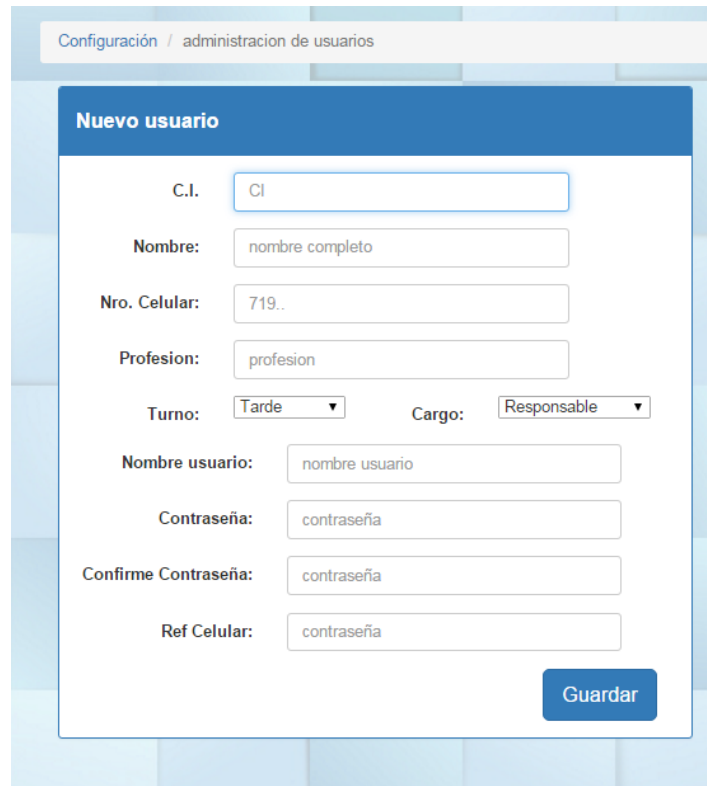


**Figura 30 Interface - Página de inicio del sistema**

Fuente: [Elaboración propia]

### c) Interface: Administración de usuarios

En la Figura 20 se muestra el panel de administración de usuarios, donde puede crear un nuevo usuario llenando el formulario con los datos del usuario. También se puede modificar datos del usuario y eliminar un usuario.



The image shows a web interface for user administration. At the top, there is a breadcrumb trail: "Configuración / administracion de usuarios". Below this is a blue header for the form titled "Nuevo usuario". The form contains several input fields and dropdown menus:

- C.I.: A text input field containing "CI".
- Nombre: A text input field containing "nombre completo".
- Nro. Celular: A text input field containing "719..".
- Profesion: A text input field containing "profesion".
- Turno: A dropdown menu with "Tarde" selected.
- Cargo: A dropdown menu with "Responsable" selected.
- Nombre usuario: A text input field containing "nombre usuario".
- Contraseña: A text input field containing "contraseña".
- Confirme Contraseña: A text input field containing "contraseña".
- Ref Celular: A text input field containing "contraseña".

A blue "Guardar" button is located at the bottom right of the form.

**Figura 31 Interface - Administración de usuarios**

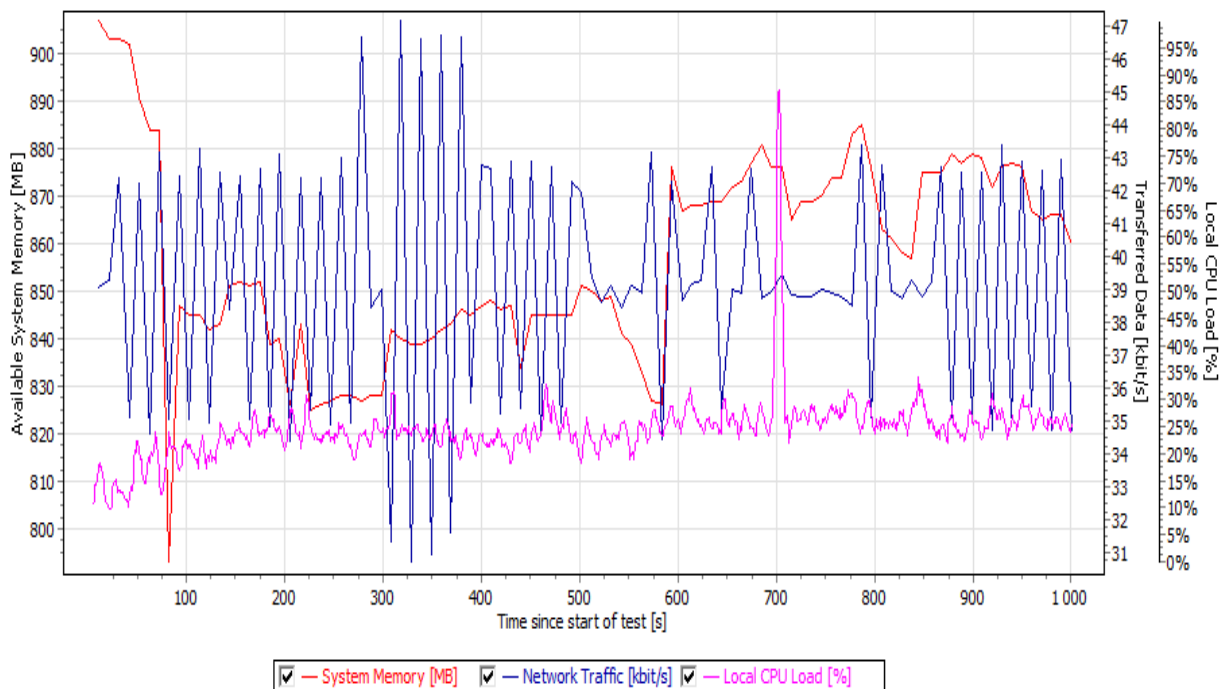
Fuente: [Elaboración propia]

### 3.5.2. Pruebas De Stress

Las pruebas de Stress sirven para encontrar el volumen de datos o de tiempo en que la aplicación comienza a fallar o es incapaz de responder a las peticiones que realiza el usuario. Son pruebas que superan los límites esperados. Las pruebas fueron realizadas con la herramienta Webserver stress tool.

### a) Transferencia de datos

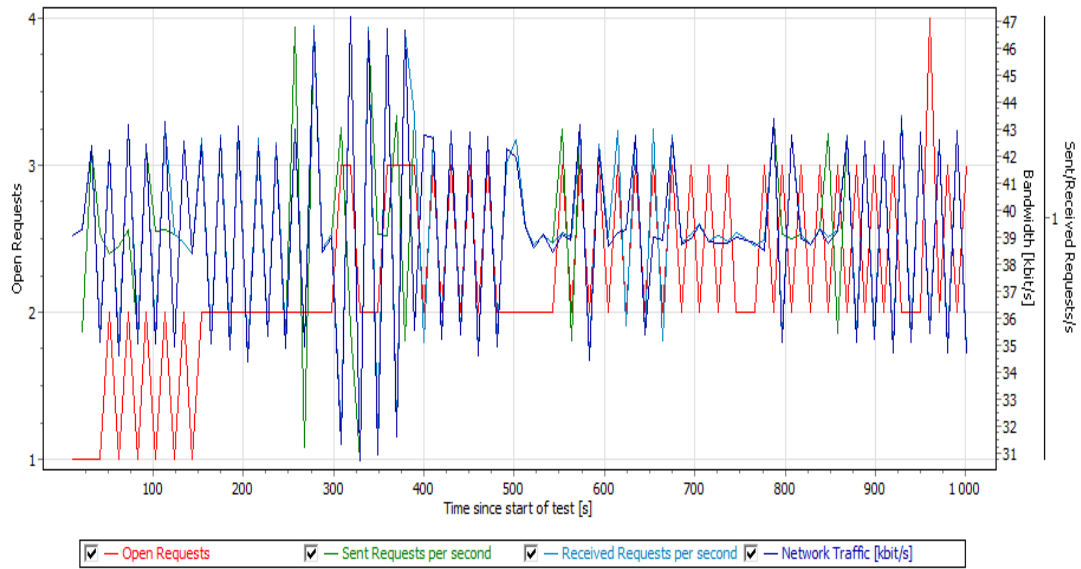
En la Figura 32 se muestra el tiempo de conexión de los usuarios al sistema y se observa que el uso de memoria puede llegar a 100% de su uso, y después de unos segundos el uso de la memoria va disminuyendo.



**Figura 32 Memoria del sistema y carga de la CPU**  
Fuente [Elaboración propia]

La siguiente Figura 33 se observa la transferencia de datos y el tiempo de respuesta del sistema a las peticiones de usuario.

- Línea roja: indica los tiempos de petición del usuario
- Línea verde: indica el tiempo de respuesta del servidor
- Línea celeste: representa el tiempo de recibo de respuesta del servidor
- Línea azul: representa el tráfico de datos kb/s del servidor

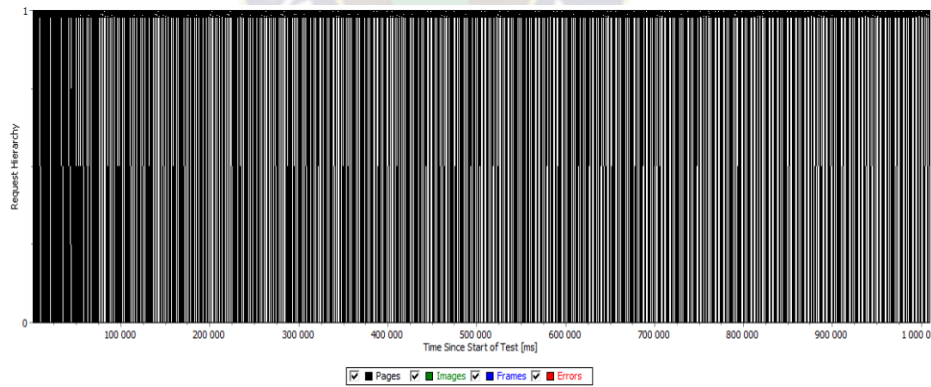


**Figura 33 Solicitud de apertura y transferencia de datos**

Fuente [Elaboración propia]

**b) Peticiones de usuario**

En la Figura 34 se muestra las peticiones de los usuarios expresados en milisegundos, y se observa que existen 0 errores en peticiones del Sistema durante 10000 [ms].



**Figura 34 Peticiones del usuario**

Fuente [Elaboración propia]

#### 4.1 Introducción

Para valorar la calidad del Sistema Web se proporciona la información adecuada sobre los datos necesarios referentes 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 la cual sirve para la evaluación de calidad de sitios web.

La seguridad es uno de los aspectos más importantes, siendo el software desarrollado para la web y tomando en cuenta que todos los datos almacenados son de vital importancia. Por lo tanto para este capítulo se utilizó como base el UNE-ISO /IEC 17779.

#### 4.2. Calidad

Pressman define a la calidad de software como: Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan.

##### 4.2.1 Caso de estudio

Usamos la metodología Web-site QEM en diferentes fases del proceso de desarrollo. Para el caso de estudio, en nuestro caso se hace la siguiente hipótesis: “el sistema web de control de personal, ventas y fallas técnicas , satisfacen en general los requerimientos de calidad en consideración de perfil de usuario y al menos el punto crítico de aceptabilidad del 60 % de la preferencia global, conforme a los requerimientos de calidad acordados”

- Se pudo establecer dos perfiles de usuario:
- Usuario Administrador
- Usuario Responsable



## 4.2.2 Tabla de características y atributos

En esta fase se definirán los requerimientos de calidad dados en la Tabla 25.

En cuanto al enfoque de modelo de calidad empleado sigue el mismo modelo de calidad mixto. Se parte de un conjunto prescrito de características, se acuerdan subcaracterísticas de niveles inferiores de los atributos (los criterios de medición) y las relaciones entre atributos, subcaracterísticas y características.

Por otra parte, el perfil de usuario seleccionado para este estudio fue el de usuario Administrador

**Tabla 25 Requerimientos de calidad**

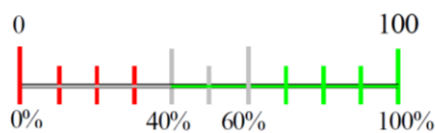
Fuente: [Elaboración propia]

<p><b>1. Usabilidad</b></p> <p>1.1 Compresibilidad global del sitio</p> <p>1.1.1 Esquema organizacional global</p> <p>1.1.1.2 Índice global</p> <p>1.2 Calidad de etiquetado</p> <p>1.2.1 Etiquetado textual</p> <p>1.2.1.1 Etiquetado con iconos</p> <p>1.3 Calidad de ayuda</p> <p>1.3.1 Ayuda explicatoria acerca del sistema</p> <p>1.4 Aspectos de interfaces y estéticos</p> <p>1.4.1 Uniformidad en los estilos</p> <p>1.4.2 Uniformidad en el estilo global</p> <p>1.4.3 Preferencia estética</p> <p><b>2. Funcionalidad</b></p>	<p>2.3 Navegabilidad global</p> <p>2.3.1 Permanencia y estabilidad en la presentación de controles</p> <p>2.4 Nivel de desplazamiento</p> <p>2.4 Desplazamiento horizontal</p> <p><b>3. Confiabilidad</b></p> <p>3.1 No deficiencia</p> <p>3.1.1 Errores de enlace</p> <p>3.2 Errores o diferencias varias</p> <p>3.2.1 Número de deficiencias debido a diferentes navegadores</p> <p>3.2.2 Numero de deficiencias o resultados inesperados</p> <p><b>4. Eficiencia</b></p> <p>4.1 Accesibilidad de Información</p> <p>4.1.1 Legibilidad clara de texto</p>
--	---

2.1 Aspectos de búsqueda	4.1.2 Legibilidad global
2.1.1 Mecanismos de búsqueda en el sitio	4.2 Performancia
2.1.1.1 Búsqueda específicas	4.2.1 Páginas rápidas
2.2 Aspectos de navegación y exploración	
2.2.1 Navegabilidad local	
2.2.2 Orientación	
2.2.2.1 Indicador del camino	
2.2.2.2 Etiqueta de la posición actual	

#### 4.2.3 Evaluación elemental

A partir del árbol de calidad antes esquematizado, y para cada atributo cuantificable  $A_i$  se asocia y determina la variable  $X_i$ , que tomara un valor a partir de un proceso de medición. Además para el rango de valores acordados para la variable  $X_i$ , por medio de un criterio elemental se hace corresponder una preferencia elemental. Ver Figura 32



**Figura 35 Rango de aceptabilidad de preferencia de calidad**

Fuente: [Elaboración propia]

La elección del tipo de criterio de evaluación elemental resulta de importancia en consideración de los niveles de precisión depende del grado de criticidad de los componentes del producto. Dos tipos básicos de criterios elementales son absolutos y relativos

### 4.2.3.1 Mediciones elementales

Para la medición de los atributos, se utilizaron los siguientes criterios elementales:

$$CVN : IE = \frac{X}{Y} * 100 \text{ con } X = \sum \text{Puntajes maximos} , Y = \sum \text{Puntajes obtenidos}$$

$$CN : IE = \frac{X}{Y} * 100 \text{ con } X = \text{Cantidad total de datos para la variable y } Y \\ = \text{cantidad total}$$

$$CB : IE = 0 \text{ si no existe, } IE = 1 \text{ si existe}$$

*CPD: Sujeto a objetividad del observador*

$$CMN : IE = 0 \approx 0 \text{ ausente}$$

$$CMN : IE = 1 \approx 60 \text{ presente parcial}$$

$$CMN : IE = 2 \approx 100 \text{ presente}$$

La Tabla 26 se muestra algunos de los valores obtenidos para las preferencias de calidad elemental.

**Tabla 26** Computo de las preferencias parciales

Fuente: [Elaboración propia]

ATRIBUTOS	CRITERIO ELEMENTAL X <sub>i</sub>	IE <sub>i</sub> (%)
<b>1. Usabilidad</b>	<b>CVN</b>	<b>90.2</b>
1.1 Compresibilidad global del sitio	CVN	95
1.1.1 Esquema organizacional global	CVN	96
1.1.1.2 Índice global	CVN	94
1.2 Calidad de etiquetado	CPD	96
1.2.1 Etiquetado textual	CPD	95
1.2.1.1 Etiquetado con iconos	CPD	90

1.3 Calidad de ayuda	CVN	86
1.3.1 Ayuda explicatoria acerca del sistema	CPD	91
1.4 Aspectos de interfaces y estéticos	CVN	94.5
1.4.1 Uniformidad en los estilos	CMN	2≈100
1.4.2 Uniformidad en el estilo global	CMN	2≈100
1.4.3 Preferencia estética	CMN	2≈100
1.5 Retroalimentación	CVN	90
1.5.1 Formularios de entrada	CPN	90
1.5.2 Reportes	CPN	90
<b>2. Funcionalidad</b>	<b>CVN</b>	<b>90.5</b>
2.1 Aspectos de búsqueda	CVN	94
2.1.1 Mecanismos de búsqueda en el sitio	CVN	95
2.1.1.1 Búsqueda específicas	CVN	96
2.2 Aspectos de navegación y exploración	CVN	95
2.2.1 Navegabilidad local	CVN	95
2.2.2 Orientación	CVN	90
2.2.2.1 Indicador del camino	CB	1≈100
2.2.2.2 Etiqueta de la posición actual	CB	1≈100
2.3 Navegabilidad global	CVN	96
2.3.1 Permanencia y estabilidad en la presentación de controles	CVN	95
2.4 Nivel de desplazamiento	CVN	85
2.4 Desplazamiento horizontal	CB	1≈100
<b>3. Confiabilidad</b>	<b>CVN</b>	<b>90</b>
3.1 No deficiencia	CVN	90
3.1.1 Errores de enlace	CVN	100
3.1.2 Enlaces rotos	CMN	2≈100

3.2 Errores o diferencias varias	CMN	2≈100
3.2.1 Número de deficiencias debido a diferentes navegadores	CMN	2≈100
3.2.2 Numero de deficiencias o resultados inesperados	CMN	1≈60
<b>4. Eficiencia</b>	<b>CVN</b>	<b>90</b>
4.1 Accesibilidad de Información	CPD	90
4.1.1 Legibilidad clara de texto	CPD	90
4.1.2 Legibilidad global	CPD	90
4.2 Performancia	CVN	90
4.2.1 Páginas rápidas	CVN	90

#### 4.2.4 Evaluación global

De acuerdo a la valoración de calidad global del Sistema Web, aplicando la metodología Website QEM el valor de calidad global total es de 90.2 %.

Los resultados de los valores de las preferencias de calidad para las características de más alto nivel, y valores finales. Ver Tabla 27

**Tabla 27 Calidad global**

Fuente: [Elaboración propia]

<b>Características de más alto nivel</b>	<b>Preferencia</b>
1. Usabilidad	90.2
2. Funcionabilidad	90.5
3. Confiabilidad	90
4. Eficiencia	90
<b>Calidad global</b>	90.2

#### **4.2.5 Análisis de resultados**

Observando la tabla 27 podemos decir que el Sistema Web de control de personal, ventas y fallas técnicas tiene una preferencia de calidad global satisfactoria del 90.2% en todas sus características de más alto nivel. Por tanto se puede decir que el Sistema Web cumple con los requisitos de calidad.

#### **4.3 Seguridad**

Para la seguridad del sistema se utilizó como base el UNE-ISO /IEC 17779, en cuyos principales puntos para la seguridad de un Sistema son:

- Confidencialidad
- Integridad
- Disponibilidad
- Autenticidad
- Auditabilidad

##### **4.3.1 Bases de la seguridad informática**

###### **4.3.1.1 Confidencialidad**

En términos de seguridad de la información, la confidencialidad hace referencia a la necesidad de ocultar o mantener secreto sobre determinada información o recurso. El objetivo de la confidencialidad es, entonces, prevenir la divulgación no autorizada de la información.

El mecanismo que garantice la confidencialidad es la Criptografía, para garantizar la confidencialidad en el sistema se utilizó la encriptación Asimétrica Existen dos claves, una para cifrar el mensaje y otra para descifrarlo, generalmente la primera es pública, es decir, solo la conoce el emisor, en cambio la segunda se denomina Privada y solo la posee a quien van dirigidos los mensajes enviados entre los que disponen de la clave pública, por tanto, solo el poseedor de la clave privada podrá leer los mensajes (desencriptarlos).

La criptografía simétrica es más vulnerable que la asimétrica por el hecho de usar una única llave, por otro lado en la encriptación simétrica es más rápida que la asimétrica y esto la favorece ya que el tiempo de descifrado es bastante importante.

Por tanto las contraseñas de cada usuario del sistema serán controlada a través de una librería creada para PHP donde maneja los roles de usuario.

#### **4.3.1.2 Integridad**

En términos de seguridad de la información, la integridad hace referencia a la fidelidad de la información o recurso, y normalmente se expresa en lo referente a prevenir el cambio impropio o desautorizado.

La integridad hace referencia a:

- La integridad de los datos (el volumen de la información)
- La integridad del origen (la fuente de los datos, llamada autenticación)

Para mantener la integridad del Sistema se han definido perfiles de usuario para el acceso con roles específicos de acuerdo al nivel del usuario, garantizando así la integridad en los datos, para que personal no autorizado no tenga acceso a modificar, eliminar datos sin autorización.

#### **4.3.1.3 Disponibilidad**

En términos de seguridad de la información, la disponibilidad hace referencia a que la información del sistema debe permanecer accesible a elementos autorizados.

Para asegurar la disponibilidad del Sistema, se implementó el uso de variables de sesión permitiendo el acceso a la información autorizada sin restricciones.

### **4.3.2 Mecanismos básicos de seguridad**

#### **4.3.2.1 Autenticidad**

Este atributo nos permite asegurar el origen de la información, es decir validar la identidad del usuario para esto el sistema debe verificar su nombre de usuario y la contraseña, para la

seguridad en la autenticación se inscripto la contraseña, una vez validado el usuario y la contraseña el sistema permite el acceso a la página principal del Sistema.

#### **4.3.2.2 Auditabilidad**

Este atributo nos permite determinar qué acciones se han ejecutado en el Sistema Web, quien lo ha ejecutado, que fecha se ejecutó, y cuáles son los registros afectados. Para esto se lleva un registro de todas las acciones ejecutadas.





### **5.1. Introducción**

Es importante tener la planificación o estimación del costo del proyecto para ello el objetivo de este capítulo es el de mostrar el análisis del costo y beneficio mediante el método COCOMO.

Antes de empezar con la estimación del costo del proyecto se hallara el punto función del Sistema de control de personal, control ventas y fallas técnicas caso “SISCOM”

### **5.2. Punto función**

La técnica de la medición del tamaño en punto-función consiste en asignar una cantidad de “puntos” a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos, siempre tratando de considerarlo desde el punto de vista del usuario. Los puntos función se calculan completando la tabla 28.

Se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla. Los valores de los dominios se definen de la forma siguiente:

- **Número de entradas de usuario.** Cada entrada externa origina en un usuario o es transmitida desde otra aplicación y proporciona distintos datos orientados a la aplicación o información de control.
- **Número de salidas de usuario.** Se cuenta cada salida orientada a la aplicación que proporciona el usuario, en este contexto la salida se refiere a informes, pantallas, mensajes, error entre otros.
- **Número de peticiones del usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.

- **Número de archivos lógicos internos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser parte de un gran base de datos o un archivo independiente).
- **Numero de archivos de interfaz externos.** Se cuentan todas las interfaces legibles por la máquina, por ejemplo archivos de datos de cinta disco que se utiliza para transmitir información a otro sistema.

Las organizaciones que utilizan métodos de punto función desarrollan criterios para determinar si una entrada en particular es simple, media, o compleja. Para calcular el punto función (PF), se utiliza la relación siguiente.

$$PF = \text{cuenta total} * (0.65 + 0.01 * \sum_{i=1}^{14} F_i)$$

Donde *cuenta total*, es la suma de todas las entradas de PF obtenidas de la tabla anterior.

**Tabla 28 Cálculo de puntos función**

Fuente: [Elaboración propia]

Parámetros de medición	Cuenta	Factor de ponderación			Resultados
		Simple	Medio	Complejo	
Número de entradas de usuarios	32	3	4	6	128
Número de salidas de usuario	5	4	5	7	25
Número de peticiones de usuario	14	3	4	6	42
Número de archivos	27	7	10	15	405
Número de interfaces externas	1	5	7	10	5
<b>Cuenta total</b>					<b>605</b>

Cada pregunta es respondida usando una escala de rangos desde 0 a 5 que tiene como valores la siguiente Tabla 29

**Tabla 29 Escala de rangos para hallar el punto función**

Fuente: [Elaboración propia]

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

$F_i$  son los factores de ajustes de valores basados en las respuestas a las siguientes preguntas de la Tabla 30

**Tabla 30 Cuestionario para el ajuste de complejidad**

Fuente: [Elaboración propia]

<b><math>F_i</math></b>	<b>Factores de ajuste</b>	<b>Valor</b>
1	¿Requiere el sistema copias de seguridad?	5
2	¿Se requiere comunicación de datos?	5
3	¿Existe funciones de procesamientos distribuido?	4
4	¿Es crítico el rendimiento?	2
5	¿Se ejecutará el sistema en entorno operativo existente y utilizado?	4
6	¿Se requiere entrada de datos?	5
7	¿Requiere la entrada de datos que las transiciones de entrada hagan sobre múltiples pantallas u operaciones?	2
8	¿Se utilizan los archivos maestros de forma interactiva?	5

9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	4
10	¿Es complejo el procesamiento interno?	4
11	¿Se ha diseñado el código para ser reutilizable?	5
12	¿Están incluidas en el diseño la conversión y la instalación	3
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14	¿Se ha diseñado la aplicación para facilitar los cambios y ser fácilmente utilizada por el usuario?	5
$\Sigma(F_i)$		57

Aplicando la fórmula de Punto función se tiene:

$$PF = \text{cuenta total} * (0.65 + 0.01 * \sum_{i=1}^{14} F_i)$$

$$PF = 738.1$$

Por tanto concluimos que tiene una complejidad de manejo de datos es de 738,1 es una forma de representa cuanto mide el sistema.

## 5.2. COCOMO

Uno de los métodos para realizar estimaciones del costo de proyecto es COCOMO (COConstructiveCOstMOdel) orientado a los puntos función. Para la estimación del costo total del sistema se tomara en cuenta: costo de elaboración del proyecto, costos de software desarrollado, costos de la implementación del sistema.

### 5.2.1. Costos de la elaboración del proyecto

Los costos de elaboración del proyecto son obtenidos del estudio del sistema en la etapa de análisis, estos costos se detallan en la Tabla 31

**Tabla 31 Costos de elaboración del proyecto**

<b>Descripción</b>	<b>Costo (Bs)</b>
Análisis y diseño del proyecto	2000
Material de escritorio	300
Otros	550
<b>Total</b>	<b>2850</b>

### 5.2.2. Costos del desarrollo del software

Se utilizara el punto función PF=738.1, a continuación realizaremos la conversión de punto función a miles de líneas de código mediante la siguiente tabla 32

**Tabla 32 Factor LCD/PF de lenguajes de programación**

<b>Lenguaje</b>	<b>Nivel</b>	<b>Factor LDC/PF</b>
Java	6	53
Visual Basic	7	46
ASP	9	36
C++	9.5	34
PHP	9	12
Ensamblador	10	320
C	9	150

$$LDC = PF * \text{Factor LDC/PF}$$

$$LDC = 738 * 12 = 8856$$

$$KLDC = 8.85$$

Ahora se aplicara las formulas básicas de esfuerzo, tiempo calendario y personal requerido.

$$E = a_b (KLCD)^{b_b}$$

$$D = c_b(E)^{d_b}$$

Dónde E: Esfuerzo aplicado en personas por mes

D: Tiempo de desarrollo en mes

KLCD: Número estimado de líneas de código distribuidas (en miles)

En la tabla 33 se muestran los tipos de proyecto de software.

**Tabla 33 Tipos de proyectos de software**

Fuente: [Elaboración propia]

Proyecto de software	a <sub>b</sub>	b <sub>b</sub>	c <sub>b</sub>	d <sub>b</sub>
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

**Orgánico:** Relativamente sencillos, en los que se trabajan en equipos pequeños con experiencia

**Semiacoplados:** Proyectos intermedios (en tamaño y complejidad) en los que participan equipos con variados niveles de experiencia

**Empotrados:** Proyectos que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringido

El proyecto al que se adecua es al Semiacoplado y el cálculo realizado para el factor de ajuste del esfuerzo FAE= 0.53, por lo tanto se obtendrá lo siguiente:

$$E = a_b(KLCD)^b_b * FAE$$

$$E = 3*(8.85)^{1.12}*0.53 = 21,76 \text{ [personas/mes]}$$

$$D = 2.5*(21.76)^{0.35} = 7.35 \text{ [meses]}$$

$$\text{Numero de programadores} = E/D = 21.76/7.35 = 2.96 \text{ [programadores]}$$

Por lo tanto se necesitarían 3 programadores para el desarrollo del proyecto. El costo salario por programador es de 4150 Bs. Por lo tanto con este dato la estimación del costo del software se calculará con la siguiente formula:

$$\begin{aligned} \text{Costo del software} &= \text{Numero de programadores} * \text{Salario de programador} \\ &= 3*4150 = 12450 \text{ Bs} \end{aligned}$$

Por lo tanto el costo de desarrollo del software de 2850 Bs + 12450 Bs = 15300 Bs

### 5.2.3. Costos de la implementación del sistema

En la empresa SISCOM, cuenta con un equipo de computación para la instalación de un servidor local.

El costo de la instalación del servidor y las configuraciones necesarias es de 300 Bs

**Por lo tanto el costo total del sistema es de 2850 Bs + 12450 Bs + 300 Bs =15600 Bs**

### CONCLUSIONES Y RECOMENDACIONES

---

#### 6.1. Conclusiones

Durante este trabajo se diseñó, desarrollo un sistema para el control de persona, ventas y fallas técnicas, cumpliendo con los siguientes propósitos:

- Se logró Implementar una base de datos donde se almacena más de mil datos para los movimientos del día a día de la empresa.
- El sistema del control de personal, ventas y fallas técnicas agilizo el proceso de la empresa obteniendo un mayor rendimiento.
- El sistema elabora planillas de control de personal donde se registrar la hora de entra, hora de salida y fecha.
- Registra ventas cronológicamente para ver las ganancias que tiene la empresa.
- Se incorporó el módulo de fallas técnicas para que la empresa tenga un informe de que máquinas están en mal estado y necesiten mantenimiento.
- Obtiene informes confiables, donde se puedan hacer comparaciones con datos pasados, de esta manera pronosticar los rendimientos, y toma buenas decisiones.

El diseño e implementación se lo realizo de una forma que se puedan adherir fácilmente a otros subsistemas.

De esta manera se cumple las expectativas y requerimientos de la empresa

“Desarrollar un sistema de control de personal, ventas y fallas técnicas”.

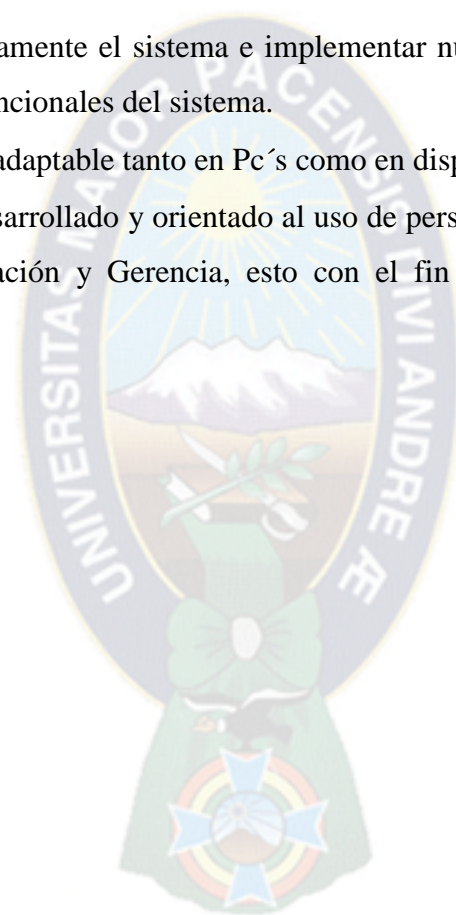


## 6.2. Recomendaciones

Viendo que la tecnología se va desarrollando a pasos agigantados cada vez mas sobre todo en el área de la informática

Por tal motivo se recomienda:

- Hacer el mantenimiento adecuado del sistema, esto con el fin de permitir mejoras funcionales del sistema.
- Actualizar periódicamente el sistema e implementar nuevos módulos, esto con el fin de permitir mejoras funcionales del sistema.
- Que el sistema sea adaptable tanto en Pc´s como en dispositivos móviles
- El software está desarrollado y orientado al uso de personas que tengan conocimiento del área de Administración y Gerencia, esto con el fin de mantener la integridad de la información.



## BIBLIOGRAFÍA

- Calle B. (2011), *Sistema de control y seguimiento de activos fijos para la empresa autoventa intercambio*. Mención ingeniería de sistemas informáticos. La Paz, Bolivia. Universidad mayor de San Andrés, Facultad de Ciencias Puras y Naturales, carrera informática.
- Cauna R. (2012), *sistema de control de personal caso: división de acciones y control UMSA*. Mención ingeniería de sistemas informáticos. La Paz, Bolivia. Universidad mayor de San Andrés, Facultad de Ciencias Puras y Naturales, carrera informática.
- Citlali N. (2014), *UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un método en caso de estudio*. Revista Latinoamericana de Ingeniería de Software. Universidad Autónoma de Yucatán. México.
- Chiavenato I. (1994), *Administración de Recursos Humanos*, 2da Edición. Mcgrawhill, México
- CDS12. *Calidad de Software*, (Consultado Abril, 2015)  
[http://es.wikipedia.org/wiki/Calidad\\_de\\_software](http://es.wikipedia.org/wiki/Calidad_de_software)
- CLD10. *Cliente Servidor de dos capas*, (Consultado Enero, 2015)  
<http://www.monografias.com/trabajos89/cliente-servidor-dos-capas/cliente-servidor-dos-capas.shtml>
- CAFU14. *Calidad de Software, funcionalidad y usabilidad*, (Consultado Abril, 2015)  
<http://albertolacalle.com/hci/funcionalidad-usabilidad.htm>
- CBML. *Curso breve de marco lógico (visión general del curso)*, (extraído el 08 de Octubre de 2014)  
[http://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.solucionesong.org%2Fimg%2Fforos%2F4ce164c4dd0de%2FMarco\\_Logico.ppt&ei=zfs1VJv3PPPkATfsYK4BQ&usq=AFQjCNHXLx0MJeOWJqSSe9pnDYLNBw-IA&sig2=2U\\_NiNQyVITugG1ToMHwRA&bvm=bv.76943099,d.cWc&cad=rja](http://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.solucionesong.org%2Fimg%2Fforos%2F4ce164c4dd0de%2FMarco_Logico.ppt&ei=zfs1VJv3PPPkATfsYK4BQ&usq=AFQjCNHXLx0MJeOWJqSSe9pnDYLNBw-IA&sig2=2U_NiNQyVITugG1ToMHwRA&bvm=bv.76943099,d.cWc&cad=rja)
- COSC95. *COCOMO, University of Southern California*. (Consultado Marzo, 2015)  
[http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)
- Ferré, (2015), *Desarrollo Orientado a objetos con UML*. Facultad de informática - UPM
- Kendall&K. (1997), *Análisis y diseño de sistemas*. New York, Services of new England Inc.
- MQ216. *Gestor de base de datos* (consultado septiembre, 2016)  
[https://www.ecured.cu/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos)
- MSO15. *Modelado de sistemas con UML*, (Consultado Enero, 2015).

- <http://www.monografias.com/Modelado de Sistemas con UML>  
MTODUWE. *Proyecto de grado de ingeniería de sistemas*, (Consultado septiembre ,2016)  
<http://proyctogradoingenieriasistemas.blogspot.com/2015/03/metodologia-uwe-uml-uml-based-web.html>
- Nina G (2011),*sistema de administración de personal unidad de talento Humano Hospital AGRAMONT*. Mención ingeniería de sistemas informáticos. La Paz, Bolivia. Universidad mayor de San Andrés, Facultad de Ciencias Puras y Naturales, carrera informática.
- Pressman R. (2010).*Ingeniería Del Software un enfoque práctico*. Séptima edición. Editorial McGrawHill. México
- Peralta A. (2003),*Metodología Scrum*. Universidad ORT Uruguay.
- Pérez H. (2010),*Propuesta de Análisis y Diseño basada en UML y UWE para la migración de arquitectura de software centralizada hacia internet*. Trabajo de graduación. Universidad de San Carlos de Guatemala.
- PH216. *Diseño de página web* (consultado septiembre,2016)  
<http://php.net/manual/es/intro-what-is.php>
- PCB15.*Pruebas de caja Blanca*,(Consultado Abril,2015)  
[http://es.wikipedia.org/wiki/Pruebas\\_de\\_caja\\_blanca](http://es.wikipedia.org/wiki/Pruebas_de_caja_blanca)
- PROYSCRUM. *Ejemplo de proyectos con Scrum*, (Consultado octubre, 2016)  
<http://docplayer.es/705796-Gestion-de-proyectos-scrum-manager-v-2-5.html>
- SSI12. *Seguridad en un sistema de Información*,(Consultado Abril, 2015)  
<http://www.monografias.com/trabajos/seguinfo/seguinfo.shtml>
- UML216. *Aprenderaprogramar.com*, (consultado septiembre , 2016)  
[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=688:i-que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46:lenguajes-y-entornos&Itemid=163](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:i-que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46:lenguajes-y-entornos&Itemid=163)

# DOCUMENTACIÓN

