

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



**PROYECTO DE GRADO**

**SISTEMA DE INFORMACION DE PROGRAMACIÓN DE OPERACIONES  
ANUAL DEL SERVICIO DEPARTAMENTAL DE SALUD LA PAZ**

Para optar al Título de Licenciatura en Informática  
Mención: Ingeniería de Sistemas Informáticos

**POSTULANTE:** Gonzalo Felix Quisbert Nina  
**TUTOR METODOLÓGICO:** Lic. Javier Hugo Reyes Pacheco  
**ASESOR:** M.Sc. Carlos Mullisaca Choque

**LA PAZ – BOLIVIA**

**2012**

## **DEDICATORIA**

A Dios por la vida presente

A mi madre, Mica, por su confianza incondicional

A mis hermanos, Jhonny, Mauricio, Cristina y Alvaro, por su apoyo

A mi esposa, Carmen, e hijas, Lorena y Dafne, que son mi inspiración para seguir adelante.

## **AGRADECIMIENTOS**

Al Servicio Departamental de Salud La Paz

- Dr. Gustavo Marconi Ojeda - Director
- Dr. Pedro Mamani Mamani - Ex Director
- Dr. Ramon Cordero Cuevas - Jefe Unidad de Planificación y Desarrollo Organizacional
- Dra. Teresa Villagomez Fernandez - Responsable de Proyectos y Convenios
- Dr. Wilfredo Tancara – Ex Director SERES
- Dr. Angel Revollo Castellón – Ex Jefe Unidad de Planificación SERES
- Sr. Erick Troche Quispe – Técnico de Planificación (Hermano)
- Lic. Rocio Vargas – Responsable de Telecomunicaciones
- Sr. William Quisbert – Tecnico SNIS

A la Carrera de Informática, en especial a:

- Lic. Javier Hugo Reyes Pachecho
- M.Sc. Carlos Mullisaca Choque

# INDICE

## CAPÍTULO I

### MARCO CONTEXTUAL

1.1 INTRODUCCION .....	1
1.2 ANTECEDENTES.....	2
1.3 TRABAJOS SIMILARES .....	3
1.4 ANALISIS Y PLANTEAMIENTO DEL PROBLEMA .....	3
1.5 OBJETIVOS .....	5
1.5.1 Objetivo General.....	5
1.5.2 Objetivos Especificos .....	5
1.6. JUSTIFICACION .....	5
1.7. LIMITES Y ALCANCES.....	6
1.8 METODOLOGIA .....	6
1.9 APORTES .....	7

## CAPÍTULO II

### MARCO TEÓRICO

2.1 REFERENCIAS IMPORTANTES EN LA LEY N° 1178.....	8
2.2 REFERENCIAS IMPORTANTES DE LAS NORMAS BASICAS SPO.....	9
2.3 METODOLOGIA PROCESO UNIFICADO .....	14
2.3.1 CARACTERISTICAS DE LA METODOLOGIA UP.....	14
2.3.2 FASES DEL PROCESO UNIFICADO .....	16
2.3.3 MODELADO .....	16
2.4 LENGUAJE UNIFICADO DE MODELADO .....	18
2.5 METODO PARA DISEÑO DE SITIOS WEB (WSDM) .....	19
2.5.1 Fases de WSDM.....	19
2.6 LENGUAJE Y SOPORTE DE IMPLEMENTACION .....	22
2.7 SEGURIDAD.....	23

2.8 CALIDAD DE SOFTWARE.....	24
2.8.1 FACTORES DE LA CALIDAD ISO 9126 .....	24
2.8.2 METRICAS DE SOFTWARE.....	25
2.8.2.1 Funcionalidad .....	25
2.8.2.2 Mantenibilidad .....	28
2.8.2.3 Confiabilidad.....	28
2.8.2.4 Portabilidad .....	30
2.8.2.4 Usabilidad .....	30

### **CAPÍTULO III**

#### **DISEÑO DEL SISTEMA**

3.1. ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN.....	31
3.2 ETAPAS DE LA METODOLOGÍA UP-WSDM .....	31
3.2.1 Modelado del Negocio .....	33
3.2.1.1 Diagrama de casos de uso .....	33
3.2.1.2 Modelo de Objetos .....	34
3.2.2 Modelado de Usuarios.....	35
3.2.2.1 Clasificación de Usuarios.....	35
3.2.2.2 Descripción de Usuarios .....	30
3.2.3-1Descripción de Requerimientos .....	36
3.2.3 Modelado de Requisitos.....	36
3.2.3.1 Diagramas de Casos de Uso.....	38
3.2.3.2 Diagramas de Actividad.....	39
3.2.4 Modelado de Análisis.....	41
3.2.4.1 Diagramas de Secuencia .....	42
3.2.5 Modelado de Diseño .....	43
3.2.5.2 Modelo Relacional .....	45
3.2.6 Modelado de Navegación.....	46
3.2.6.1 Estructura del Sitio Web .....	46
3.2.6.2 Clases Navegacionales.....	46

3.2.7 Modelado de Implementación.....	47
3.2.7.2 Pantallas del Sistema.....	49
3.2.8 Modelado de Despliegue.....	52
3.2.8.1 Diagrama de Despliegue .....	45
3.2.9 Modelado de Pruebas .....	53
3.2.9.1 Pruebas de Caja Blanca.....	53
3.2.9.2 Pruebas de Caja Negra .....	55
3.2.9.3 Seguridad del Sistema.....	55

## **CAPÍTULO IV**

### **EVALUACIÓN DEL SISTEMA**

4.1 CALIDAD DE SOFTWARE .....	57
4.1.1 Funcionalidad.....	57
4.1.2 Mantenibilidad .....	59
4.1.3 Confiabilidad.....	60
4.1.4 Portabilidad .....	62
4.1.5 Usabilidad .....	63

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

5.1 CONCLUSIONES .....	64
5.2 RECOMENDACIONES .....	64

<b>BIBLIOGRAFIA</b> .....	65
---------------------------	----

### **ANEXOS**

## **Resumen**

Las organizaciones han reconocido la importancia de las tecnologías de la información para automatizar procesos administrativos, con la finalidad de obtener la optimización de esfuerzo laboral y tiempo.

En la Unidad de Planificación del Servicio Departamental de Salud La Paz (SEDES), se dirige el proceso de planificación de corto plazo, enmarcado en el Sistema de Programación de Operaciones (SPO) de la Ley de Control Gubernamentales (Ley N° 1178), el SPO traduce los objetivos y planes estratégicos de cada entidad, en resultados concretos; en tareas específicas a ejecutar; en procedimientos a emplear y en medios y recursos a utilizar, todo ello en función del tiempo y del espacio, de ahí que surge la necesidad de contar con una herramienta que ayude a aplicar la Programación de Operaciones Anual en el SEDES, en sus procesos de programación, seguimiento y evaluación de operaciones.

Para obtener el sistema informático una vez identificado las necesidades de la Unidad de Planificación se utilizaron las siguientes metodologías y herramientas:

Para identificar y ordenar los procesos de programación de operaciones, se utilizó la complementación de la metodología Proceso Unificado y la Metodología para Diseño de Sitios Web (WSDM).

Para la programación se utilizó el lenguaje de programación PHP, con el gestor de base de datos Mysql, soportada por un Servidor Apache.

A la finalización del proceso de construcción obtenemos un software de calidad que proporcione información oportuna, confiable y veraz que será utilizada por la Dirección

del SEDES La Paz, mediante la Unidad de Planificación, para coadyuvar a tener una mejor gestión del SEDES.

# CAPITULO I

## 1.1 INTRODUCCION

El presente Proyecto de Grado, más allá de servir, como un trabajo para optar al Título de Licenciatura, tiene la inquietud de aportar a una institución pública, el soporte a la aplicación de uno de sus sistemas de administración, mediante el uso de tecnologías de la información y comunicación, reconociendo la importancia de la información en el tiempo presente.

“Las organizaciones han reconocido, desde hace mucho, la importancia de administrar recursos principales tales como la mano de obra y las materias primas. La información se ha colocado en un lugar adecuado como recurso principal. Los tomadores de decisiones están comenzando a comprender que la información no es sólo un subproducto de la conducción, sino que a la vez alimenta a los negocios y puede ser el factor crítico para la determinación del éxito o fracaso de éstos” [1].

Las entidades del sector público aplican los sistemas de administración y control, establecidos en la Ley N° 1178, para lograr la administración eficaz y eficiente de los recursos públicos, generar información que muestre transparencia de gestión, lograr que todo servidor público asuma plena responsabilidad por el desempeño de sus funciones y desarrollar la capacidad administrativa.

El Sistema de Programación de Operaciones (SPO) tiene como objetivo traducir los objetivos y planes estratégicos de la entidad, en resultados concretos a alcanzar en el corto plazo, en un instrumento, que es el Programa Operativo Anual <sup>1</sup> (POA), donde se inscribe tareas específicas a ejecutar, procedimientos a emplear, y medios y recursos a utilizar, en función del tiempo y espacio [2].

---

<sup>1</sup> Proceso de Planificación de corto plazo, vale decir la programación de operaciones a ser realizadas en un año con inclusión de recursos humanos, activos, insumos y materiales, plasmado en un documento oficial (Ver anexos Formularios de programación)

El Servicio Departamental de Salud de (SEDES) La Paz, es una institución pública desconcentrada, dependiente del Gobierno Autónomo del Departamento de La Paz, que implementa la Política Departamental de Salud en el marco de la Constitución Política del Estado, bajo los principios de Equidad, Calidad, Intersectorialidad, Interculturalidad y Participación Social [3].

El SEDES La Paz como una entidad pública del Estado Plurinacional de Bolivia, aplica los sistemas de administración definidos en la Ley N°1178, y específicamente el SPO, con la Programación de operaciones anual, para la concreción de sus objetivos institucionales.

El presente proyecto tiene el propósito de desarrollar un Sistema de Información que facilite la Programación de operaciones anual del SEDES La Paz.

## 1.2 ANTECEDENTES

En el marco del Plan Nacional de Desarrollo, Plan Nacional de Salud, Plan Territorial Autónomo del Departamento de La Paz, Plan Estratégico Institucional, Ley N°1178, las normas básicas del SPO y el Reglamento específico del programa de operaciones; El SEDES La Paz realiza su POA, el mismo que es resultado de la compatibilización de los Programas operativos anuales de las áreas organizacionales<sup>2</sup>.

La Unidad de Planificación es la encargada de dirigir el proceso de planificación para la programación de operaciones, de acuerdo a la normativa vigente, como de realizar el seguimiento y evaluación de los mismos, brindando información para el análisis de la situación.

La Unidad de Planificación para el proceso de la programación de operaciones anual en la entidad define las siguientes actividades:

- Coordinación con Dirección y jefaturas de unidad para el establecimiento de lineamientos y objetivos institucionales.
- Socialización de lineamientos y objetivos institucionales en áreas organizacionales.

---

<sup>2</sup> Área organizacional es una generalización de unidad, programa, área o coordinadora de red, componentes del organograma del SEDES La Paz (Ver anexos: Organograma del SEDES)

- Facilitación de formulación del POA en áreas organizacionales en formularios preestablecidos
- Compatibilización y consolidación de POA's de áreas organizacionales en un solo POA institucional
- Seguimiento mensual a la ejecución de operaciones por responsables de las áreas organizacionales en formularios preestablecidos, diseñado en libros de Excel.
- Compatibilización y consolidación del seguimiento mensual a la ejecución de operaciones en un solo informe
- Evaluación de áreas organizacionales e institucional con indicadores de eficacia, en función a la ejecución de operaciones programadas
- Almacenamiento en medio magnético e impreso de POA's

### 1.3 TRABAJOS SIMILARES

Para el presente Proyecto de Grado, se tomaron como referencia los siguientes trabajos:

- *SISTEMA DE CONTROL Y SEGUIMIENTO DE PROYECTOS, CASO: OFICIALIA MAYOR DE PROTECCIÓN SOCIAL, DIRECCIÓN DE EDUCACIÓN, GOBIERNO MUNICIPAL DE EL ALTO* [4]. La programación se realizó con el lenguaje de programación PHP, bajo los sistemas operativos Linux y Windows. El sistema permite realizar el control y seguimiento de proyectos de manera optima, centralizando la información en un servidor.
- *SISTEMA DE CONTROL Y SEGUIMIENTO A PROGRAMAS OPERATIVOS ANUALES Y (POA'S) PARA EL VICEMINISTERIO DE BIODIVERSIDAD, RECURSOS FORESTALES Y MEDIO AMBIENTE* [5]. El trabajo muestra la medición de la calidad en base al método de cuantificación Web-site QEM que establece criterios, elementos inherentes en la medición de aplicaciones web.

### 1.4 ANALISIS Y PLANTEAMIENTO DEL PROBLEMA

Al realizar el estudio y análisis por medio de entrevistas, formularios y memorias se han detectado los siguientes problemas:

- Programación de operaciones de las áreas organizacionales con insuficiente relación lógica respecto a los objetivos de gestión institucionales
- Envío retrasado de documento de POA de áreas organizacionales a la Unidad de Planificación
- Insuficiente difusión del documento del POA a áreas organizacionales
- Dificultad en la consolidación de la programación de operaciones de áreas organizacionales
- Dificultad en el correcto llenado de formularios establecidos del POA por responsables de áreas organizacionales
- Desorden en el almacenamiento de información documentada de programación de operaciones
- Insuficiente manejo del programa de aplicación de Hojas electrónicas por parte de responsables de áreas organizacionales
- Dificultad en el seguimiento a la ejecución de operaciones programados
- Insuficiente relación en la programación de objetivos de gestión, operaciones, responsables de operaciones, medios materiales, servicios y presupuesto
- Los formularios establecidos tienen una insuficiente relación lógica en la programación de operaciones
- Insuficiente programación de operaciones en base a partidas de gasto presupuestario
- Deficiente evaluación de ejecución a la programación de operaciones
- Utilización de diferentes formatos de formularios de programación y seguimiento de operaciones por responsables de áreas organizacionales
- Dificultades de entrega oportuna de informes de seguimiento a la ejecución de operaciones de áreas organizacionales que tienen su funcionamiento en distintos lugares geográficos

#### PLANTEAMIENTO DEL PROBLEMA

Por lo que se considera la necesidad de plantear el siguiente problema:

¿El desarrollo de un Sistema de Información, optimizará la Programación de operaciones anual del SEDES La Paz?

## 1.5 OBJETIVOS

### 1.5.1 Objetivo General

Desarrollar un Sistema de Información que permita optimizar la Programación de operaciones anual del SEDES La Paz.

### 1.5.2 Objetivos Específicos

- Desarrollar una aplicación Web de programación de operaciones
- Desarrollar una Base de Datos de programación de operaciones
- Desarrollar interfaces de Sistema, funcionales a las necesidades del usuario
- Desarrollar los procesos de programación, seguimiento y evaluación de operaciones

## 1.6 JUSTIFICACIÓN

### 1.6.1 Justificación Tecnológica

En la actualidad una parte del desarrollo de la información de negocios, social, cultural u otros, está dispuesta sobre un conjunto descentralizado de redes de comunicación interconectadas de alcance mundial, denominada Internet.

El SEDES La Paz como una institución de alcance departamental y con dependencias en distintos lugares geográficos, necesita hacer uso de tecnologías de comunicación como el Internet, Intranet, Telefonía IP y otros, para el normal desenvolvimiento de actividades.

El Sistema de información de programación de operaciones del SEDES La Paz, se definirá como una aplicación Web, donde los usuarios podrán utilizarlo accediendo a un servidor web a través de Internet o Intranet mediante un navegador.

### 1.6.2 Justificación Económica

El presupuesto destinado al funcionamiento de las entidades públicas es muy limitado, por lo que se hace necesaria una correcta priorización de operaciones y asignación coherente de gasto presupuestario.

El Sistema de información de programación de operaciones del SEDES La Paz, facilitará la programación del presupuesto dispuesto para la ejecución de operaciones.

### **1.6.3 Justificación Social**

La implementación del Sistema de información de programación de operaciones procura apoyar la gestión administrativa del SEDES La Paz y por ende a un mejor servicio de sus usuarios.

### **1.7 LIMITES Y ALCANCES**

El Sistema de información de programación, seguimiento y evaluación de operaciones del SEDES La Paz funcionará en el marco de la Ley N° 1178 [2], en su Subsistema de Programación de Operaciones y subsistemas interrelacionados y en la Resolución Suprema N° 225557 [6], que aprueba las Normas Básicas del Sistema de Programación de Operaciones.

### **1.8 METODOLOGÍA**

En la identificación de problemas, objetivos se utilizó la técnica del Árbol de Problemas y Objetivos

La planeación del proyecto se realizó con el método del Marco Lógico.

El desarrollo de software utilizará la metodología del Proceso Unificado con el Lenguaje Unificado de Modelado (UML).

La programación del software del Sistema de información se realizará en el Lenguaje de programación PHP.

El contenido de las páginas dinámicas tendrá el soporte de un programa servidor Web: Apache y la Base de datos se implementará en MySQL.

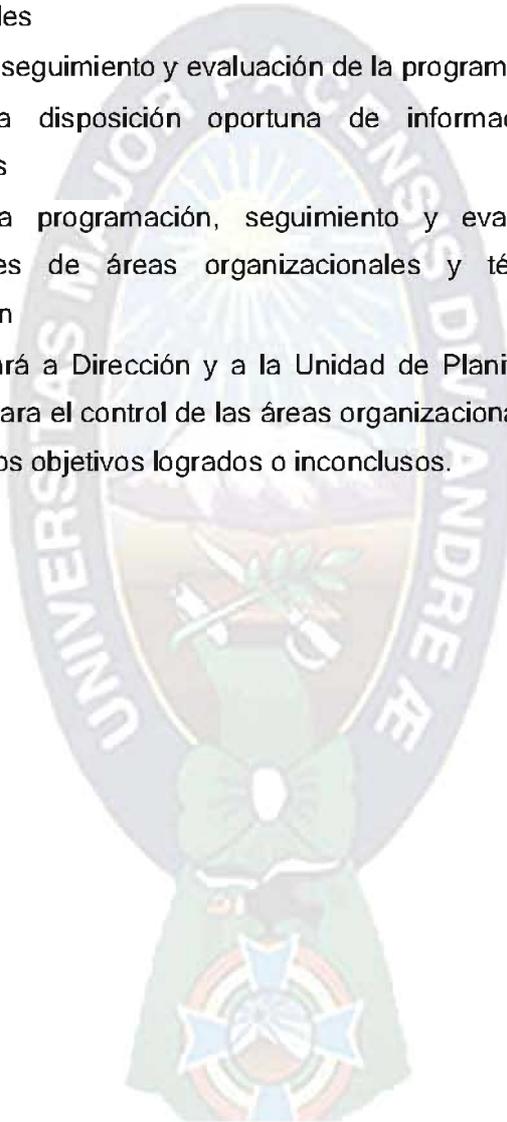
El diseño de la aplicación Web utilizará el Método para Diseño de Sitios Web (WSDM en sus siglas en inglés)

La evaluación de la calidad de software seguirá los atributos de calidad definidos en la ISO 9126.

## 1.9 APORTE

La implementación del Sistema de Programación de Operaciones del SEDES La Paz:

- Centralizará la información de programación de operaciones en una Base de Datos
- Mejorará la programación de operaciones en relación a los objetivos de gestión institucionales
- Mejorará el seguimiento y evaluación de la programación de operaciones
- Mejorará la disposición oportuna de información de programación de operaciones
- Facilitará la programación, seguimiento y evaluación de operaciones a responsables de áreas organizacionales y técnicos de la Unidad de Planificación
- Proporcionará a Dirección y a la Unidad de Planificación de una herramienta gerencial, para el control de las áreas organizacionales y toma de decisiones de acuerdo a los objetivos logrados o inconclusos.



# CAPITULO II

## MARCO TEORICO

El presente capítulo describe algunos de los conceptos y definiciones que a lo largo del desarrollo del proyecto serán empleados.

El proyecto se sustenta en el Marco legal de la Ley N° 1178, de Administración y Control Gubernamentales [2] y normas Básicas de Sistema de Programación de Operaciones [8].

### 2.1 REFERENCIAS IMPORTANTES EN LA LEY N° 1178

**Artículo 1º.-** La presente ley regula los sistemas de Administración y de Control de los recursos del Estado y su relación con los sistemas nacionales de Planificación e Inversión Pública, con el objeto de:

- a) Programar, organizar, ejecutar y controlar la captación y el uso eficaz y eficiente de los recursos públicos para el cumplimiento y ajuste oportuno de las políticas, los programas, la prestación de servicios y los proyectos del Sector Público;
- b) Disponer de información útil, oportuna y confiable asegurando la razonabilidad de los informes y estados financieros;
- c) Lograr que todo servidor público, sin distinción de jerarquía, asuma plena responsabilidad por sus actos rindiendo cuenta no sólo de los objetivos a que se destinaron los recursos públicos que le fueron confiados sino también de la forma y resultado de su aplicación,
- d) Desarrollar la capacidad administrativa para impedir o identificar y comprobar el manejo incorrecto de los recursos del Estado.

**Artículo 2º.-** Los sistemas que se regulan son:

- a) Para programar y organizar las actividades:

- Programación de Operaciones.
- Organización Administrativa.
- Presupuesto.

b) Para ejecutar las actividades programadas:

- Administración de Personal.
- Administración de Bienes y Servicios.
- Tesorería y Crédito Público.
- Contabilidad Integrada.

c) Para controlar la gestión del Sector Público:

- Control Gubernamental, integrado por el Control Interno y el Control Externo Posterior.

**Artículo 3º.-** Los sistemas de Administración y de Control se aplicarán en todas las entidades del Sector Público, sin excepción.

**Artículo 6º.-** El Sistema de Programación de Operaciones traducirá los objetivos y planes estratégicos de cada entidad, concordantes con los planes y políticas generados por el Sistema Nacional de Planificación, en resultados concretos a alcanzar en el corto y mediano plazo; en tareas específicas a ejecutar; en procedimientos a emplear y en medios y recursos a utilizar, todo ello en función del tiempo y del espacio. Esta programación será de carácter integral, incluyendo tanto las operaciones de funcionamiento como las de ejecución de preinversión e inversión. El proceso de programación de inversiones deberá corresponder a proyectos compatibilizados con las políticas sectoriales y regionales, de acuerdo con las normas del Sistema Nacional de Inversión Pública.

## **2.2 REFERENCIAS IMPORTANTE DE LA NORMAS BASICA DEL SISTEMA DE PROGRAMACIÓN DE OPERACIONES**

**Artículo 1. Concepto y objeto del Sistema de Programación de Operaciones**

El Sistema de Programación de Operaciones es un conjunto ordenado y coherente de principios, disposiciones normativas, procesos e instrumentos técnicos, que tiene por objeto la elaboración del Programa de Operaciones Anual según los programas y proyectos definidos en el marco del Plan General de Desarrollo Económico y Social de la República.

Asimismo tiene por objeto la generación y obtención de información confiable, útil y oportuna, para el seguimiento y evaluación de los objetivos y operaciones previstos en el Programa de Operaciones Anual.

### **Artículo 3. Ámbito de Aplicación de las Normas Básicas**

Las presentes Normas Básicas son de aplicación obligatoria en todas las entidades y órganos públicos comprendidos en el alcance de los Artículos 3° y 4° de la Ley 1178 de Administración y Control Gubernamentales.

### **Artículo 6. Principios del Sistema de Programación de Operaciones**

Los principios en los que se sustenta el Sistema de Programación de Operaciones, además de la eficacia, eficiencia y oportunidad, establecidos en la Ley 1178, son:

- a) Integralidad y Universalidad: El Programa de Operaciones Anual de cada entidad debe contemplar todos los objetivos resultantes del cumplimiento de las competencias asignadas.
- b) Sostenibilidad: El Programa de Operaciones Anual de cada entidad debe sujetarse a las posibilidades reales de su financiamiento.
- c) Transparencia: El Programa de Operaciones Anual de cada entidad debe ser expresado en términos claros, y difundido tanto al interior de cada entidad como para conocimiento público.
- d) Flexibilidad: El Programa de Operaciones Anual de cada entidad, podrá ser objeto de ajustes o modificaciones.

### **Artículo 15. Elaboración y contenido del Programa de Operaciones Anual**

En cada entidad y órgano público la elaboración del Programa de Operaciones Anual debe sujetarse a los recursos que se estimen disponer para el ejercicio fiscal, a los

objetivos y programas contenidos en el Plan Estratégico Institucional, y a las políticas públicas definidas para cada gestión fiscal.

La elaboración del Programa de Operaciones Anual comprende:

- a) El Análisis de Situación.
- b) La definición de los objetivos de gestión para cada ejercicio fiscal.
- c) La determinación de las operaciones de funcionamiento e inversión correspondientes.
- d) La definición de indicadores de gestión.
- e) La determinación de los recursos requeridos para la ejecución de las operaciones previstas en el POA.

#### **Artículo 17 Determinación de los Objetivos de Gestión Anual**

- I. En cada entidad y órgano público deben identificarse los Objetivos de Gestión que se pretenden alcanzar en una determinada gestión fiscal, según y los objetivos definidos en el Plan Estratégico Institucional y considerando los resultados del Análisis de Situación Interna y Externa.
- II. Los Objetivos de Gestión deben ser medibles cuantitativamente.
- III. Cuando estos Objetivos de Gestión Institucionales demanden la realización de varios procesos, o cuando demanden la generación de distintos productos, deberán ser desagregados al interior de cada entidad y órgano público, en Objetivos de Gestión Específicos.

#### **Artículo 18 Indicadores de Gestión**

- I. Cada entidad y órgano público debe establecer los indicadores de eficiencia y eficacia para cada Objetivo de Gestión Institucionales y Específicos.
- II. Los indicadores de eficacia deberán establecerse ponderando el grado de cumplimiento de los objetivos, respecto a los resultados programados para el período a evaluar.
- III. Los indicadores de eficiencia deberán establecerse ponderando los resultados alcanzados respecto a los insumos o recursos utilizados para su cumplimiento.
- IV. Los indicadores de eficacia y eficiencia deberán contemplar básicamente las siguientes características:

- a) Homogeneidad, los resultados deberán ser comparables a través del tiempo.
- b) Competencia, los indicadores deberán corresponder al bien o servicio que se trate y no a otros productos relacionados con aquel.
- c) Oportunidad, los indicadores deberán estar disponibles dentro de plazos que permitan una oportuna evaluación y toma de decisiones.

#### **Artículo 19 Determinación de Operaciones**

- I. En cada entidad y órgano público deberán estructurarse las operaciones a desarrollar para alcanzar los Objetivos de Gestión Institucionales o Específicos establecidos en el Programa de Operaciones Anual.
- II. La determinación de operaciones deberá incluir:
  - a) El área ó unidad organizacional responsable de realizar la operación.
  - b) El periodo de tiempo, término y finalización, establecido para la ejecución de cada operación.

#### **Artículo 20. Determinación de Recursos humanos y de bienes y servicios**

Para cada Objetivo de Gestión Institucional o Específico deberá identificarse la cantidad de recursos humanos y de bienes y servicios requeridos para su desarrollo.

El requerimiento deberá considerar:

- a) El análisis de los recursos existentes.
- b) La disponibilidad de recursos para la gestión y las políticas públicas definidas sobre el uso de estos recursos.
- c) El Cronograma de requerimientos, definiendo los plazos máximos para la obtención de los insumos, materiales, activos fijos, servicios, u otros que permitan el cumplimiento de los objetivos y operaciones.

#### **Artículo 21. Articulación POA - Presupuesto**

El Programa de Operaciones Anual se articulará con el Presupuesto, vinculando los Objetivos de Gestión Institucionales o Específicos con las categorías programáticas del Presupuesto.

## **Artículo 22. Seguimiento de la Ejecución del Programa de Operaciones Anual**

- I. Cada entidad y órgano público deberá elaborar y emitir en forma periódica, la información de ejecución del Programa de Operaciones Anual, a nivel Institucional y por cada Objetivo de Gestión Institucional o Específico, según los formularios de registro que para este fin disponga el Ministerio de Hacienda, vinculando esta información con la ejecución del Presupuesto.
- II. Esta información de ejecución deberá ser difundida tanto al interior de cada entidad, como para conocimiento público. Asimismo deberá ser presentada al Ministerio de Hacienda, en los plazos que éste determine, y cuando sea requerida por el H. Congreso Nacional o por la Contraloría General de la República.

## **Artículo 23. Evaluación de los resultados**

- I. Cada entidad y órgano público deberá evaluar el cumplimiento del Programa de Operaciones Anual, ponderando los resultados alcanzados respecto a los resultados programados y a la ejecución del Presupuesto. La evaluación deberá contemplar:
  - a) El Análisis de Situación, o los factores o circunstancias que hubieran incidido en el cumplimiento de los resultados.
  - b) El grado de cumplimiento de los Objetivos de Gestión, según los indicadores previstos y el cronograma de su realización.
  - c) El grado de cumplimiento de los Objetivos de Gestión y su vinculación con los recursos asignados.
- II. En caso de detectarse desviaciones respecto a la programación, cada entidad y órgano público deberá disponer las medidas correctivas necesarias que permitan cumplir con los objetivos de gestión previstos.

## 2.3 METODOLOGIA PROCESO UNIFICADO [7]

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (véase la Figura 2.1).



Figura 2.1 Un proceso de desarrollo de software  
Fuente: [6]

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema, de hecho UML es una parte esencial del Proceso Unificado - sus desarrollos fueron paralelos.

Los aspectos definitorios del Proceso Unificado se resumen en tres frases claves

- Dirigidos por casos de uso
- Centrado en la arquitectura
- Iterativo e incremental

### 2.3.1 CARACTERISTICAS DE LA METODOLOGIA UP

#### DIRIGIDO POR CASOS DE USO

Los casos de uso modelan los requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema y guían el proceso de desarrollo (diseño, implementación y prueba). Basándose en los casos de uso se crea una serie de modelos de diseño e implementación. De este modo, no solo inician el proceso de desarrollo sino que proporcionan un hilo conductor.

#### CENTRADO EN LA ARQUITECTURA

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permiten tener una visión común entre todos los involucrados

(desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

## ITERATIVO E INCREMENTAL

Para hacer más manejable un proyecto se recomienda dividirlo en ciclos, en cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un miniproyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. El Proceso Unificado divide el proceso en cuatro fases: inicio, elaboración, construcción y transición, dentro las cuales se realizan varias iteraciones en número variable según el proyecto, y en las que se hace un mayor o menor hincapié en las distintas actividades. Una iteración puede realizarse por medio de una cascada, que se divide en flujos de trabajo: requisitos, análisis, diseño, implementación y prueba.



Figura 2.2 Esfuerzo en actividades según fase del proyecto

Fuente: [6]

### 2.3.2 FASES DEL PROCESO UNIFICADO

El Proceso Unificado consta de cuatro fases que son importantes para el desarrollo del producto:

- Inicio
- Elaboración
- Construcción
- Transición

#### INICIO

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar qué recursos deben ser asignados al proyecto

## **ELABORACION**

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

## **CONSTRUCCIÓN**

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta Fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

## **TRANSICION**

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

### **2.3.3 MODELADO**

#### **MODELADO DEL NEGOCIO**

Con este flujo de trabajo pretendemos llegar a un mejor entendimiento de la organización donde se va a implantar el producto.

## **MODELADO DE REQUISITOS**

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que construyamos.

En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

## **MODELADO DE ANÁLISIS**

En la fase de diseño conceptual, debe alcanzarse una idea clara de cómo será la interfaz de usuario y las relaciones con esta para desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior. La interfaz determinará en gran medida la percepción que el usuario tendrá de la aplicación.

## **MODELADO DE DISEÑO**

En esta fase se desarrollan diagramas que colaboraran para entregar a los usuarios un producto amigable, entendible y fácil de manejar.

## **IMPLEMENTACION**

En la fase de implementación o producción, se crea el producto final. Llegados a este punto, a grandes rasgos, es cuando debe empezarse a programar, lo cual implica haber escogido el o los lenguajes de programación que mejor se adapten a nuestro proyecto, las bases de datos correspondientes que se precisen, la tecnología que garantice el éxito, etc. Se desarrollan los gráficos y textos definitivos y el sitio debe ser codificado.

En esta etapa corresponde exactamente a la que se describiría en la Ingeniería del Software clásica, puesto que la Ingeniería de la Usabilidad no trata de cómo programar un producto iterativo, sino de la metodología para conseguir un producto usable.

## 2.4 LENGUAJE UNIFICADO DE MODELADO [9]

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Vista	Diagramas	Conceptos principales
Vista estática	Diagrama de clases	Clase, asociación, generalización Dependencia, realización, interfaz
Vista de casos de uso	Diagrama de casos de uso	Caso de uso, actor, asociación, extensión, inclusión, generalización de casos de uso
Vista de implementación	Diagrama de componentes	Componente, interfaz, dependencia, realización
Vista de despliegue	Diagrama de despliegue	Nodo, componente, dependencia, localización
Vista de maquinas de estado	Diagrama de estados	Estado, evento, transición, acción

Vista de actividad	Diagrama de actividad	Estado, actividad, transición de terminación, división, unión
Vista de interacción	Diagrama de secuencia Diagrama de colaboración	Interacción, objeto, mensaje, activación Colaboración, interacción, rol de colaboración, mensaje
Vista de gestión del modelo	Diagrama de clases	Paquete, subsistema y modelo

Tabla 2.3 Vistas y diagramas de UML

Fuente: [8]

## 2.5 MÉTODO PARA DISEÑO DE SITIOS WEB (WSDM) [11]

El Método para diseño de sitios web (WSDM) es una propuesta en la que el sistema se define en base a los grupos de usuarios. Fue propuesto por De Troyer y Leune en 1997 y se compone de cuatro fases.

Los autores de WSDM dividen los sitios web en dos grupos: Kiosco web y Aplicación web. Las que pertenecen al primer tipo ofrecen al usuario una determinada información y les permite navegar hacia ella. Las aplicaciones web englobarían a aquellos sistemas de información interactivos cuya interfaz de usuario es un conjunto de páginas web. Tras esta división, los autores de WSDM especifican que su propuesta es una propuesta orientada a la primera de ellas, no resultando adecuada para las aplicaciones web.

### 2.5.1 FASES DE WSDM

El modelo de diseño de sitios web se divide en cinco fases: especificación de la misión, modelo de usuario, diseño conceptual, diseño de la implementación e implementación. A su vez, el modelo de usuario se divide en dos subfases: clasificación y descripción. Por su parte el diseño conceptual se divide en otras subfases: modelado de objetos y diseño navegacional.

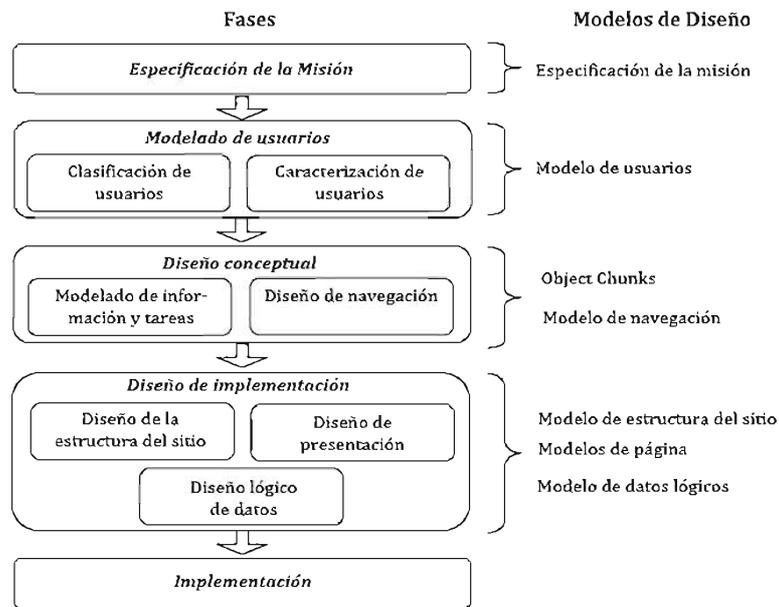


Figura 2.4 Fases de la Metodología WSDM  
Fuente: [11]

## FASE 1 - MODELADO DE USUARIOS

En esta primera fase se intenta detectar los perfiles de usuarios que se van a presentar en la aplicación. Para ello, se deben realizar dos tareas:

- Clasificación de usuarios: en este paso se deben identificar y clasificar a los usuarios que va a tener el sistema. Para ello, los autores de WSDM proponen que se estudie el entorno de la organización donde se vaya a implantar el sistema y los procesos que se vayan a generar. De esta forma, se debe generar un diagrama similar al diagrama de contexto.
- Descripción de los grupos de usuarios: en esta segunda etapa se describen con más detalles a los grupos de usuarios detectados en la etapa anterior. Para ello, se debe elaborar un diccionario de datos, en principio con formato libre, en el que se indique los requisitos de almacenamiento de información para cada grupo de usuarios y sus características.

## FASE 2 - DISEÑO CONCEPTUAL

La fase de diseño conceptual se hace una diferenciación de los grupos de usuarios desde el comienzo y proponen realizar primero un modelo conceptual y luego un modelo de navegación.

Así, esta fase en WSDM se divide en las siguientes dos etapas:

- Modelado de objetos

El objetivo de esta subfase es modelar formalmente los requisitos de información expresados en la etapa de descripción de la fase anterior. Para ello, se debe obtener un modelo de clases para cada grupo de usuarios, a cada modelo se le denominará diagrama de objetos de usuarios. Al conjunto de todos los diagramas de todos los usuarios se le denomina modelo de objetos de usuarios. La nomenclatura de clases se representa mediante la propuesta de OMT (en las últimas versiones ya asume la de UML).

- Diseño navegacional

Esta fase tiene como objetivo el conseguir un modelo para representar las posibilidades de navegación del sistema. El modelo propuesto por WSDM es bastante sencillo y se basa en representar la navegación a través de un conjunto de pistas de navegación. Una pista de navegación expresa cómo un usuario concreto puede navegar hacia una determinada información.

Cada componente de información se corresponde con una perspectiva de los tipos de objetos descritos en la subfase anterior. Es decir, una visión particular de la información. Los componentes de información pueden estar relacionados entre ellos a través de enlaces. Los componentes de navegación son conjuntos de enlaces y las entidades externas representan una referencia hacia un elemento externo al sistema.

En WSDM se propone un algoritmo para construir el modelo navegacional a partir del modelo de clases para un determinado usuario, cada uno de estos modelos es un contexto.

### FASE 3 - DISEÑO DE LA IMPLEMENTACIÓN

Una vez definido el modelo es necesario diseñar la interfaz y el entorno de usuario del sistema.

#### **FASE 4 - IMPLEMENTACIÓN**

En esta fase se pretende conseguir la aplicación ejecutable final en base a los resultados de las fases anteriores. De nuevo tampoco se propone ninguna técnica en concreto para ello.

### **2.6 LENGUAJE Y SOPORTE DE IMPLEMENTACION**

#### **PHP [12]**

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor*. Fue creado originalmente por Rasmus Lerdorf en 1994; Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

#### **MYSQL [13]**

**MySQL** es un sistema de gestión de bases de datos relacional, multihilo y multiusuario de Sun Microsystems.

#### **SERVIDOR WEB [14]**

El **servidor HTTP Apache** es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1<sup>2</sup> y la noción de sitio virtual. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

## 2.7 SEGURIDAD [16]

### Identificación y Autenticación

Para poder acceder a un sistema informático se puede proceder a identificar y autenticar al usuario, con el fin de comprobar si se trata de un usuario autorizado a acceder a los recursos del mismo, mediante el uso de un identificador (ID) y asociar al identificador de usuario una palabra de paso, contraseña o password.

### Cortafuegos

Un mecanismo utilizado para proporcionar seguridad en una red está dado por el uso de un Cortafuegos (Firewall), que es un dispositivo que combina hardware y software para aislar una red local de Internet, el objetivo de un cortafuegos es restringir el flujo de información entre dos redes, la local e Internet.

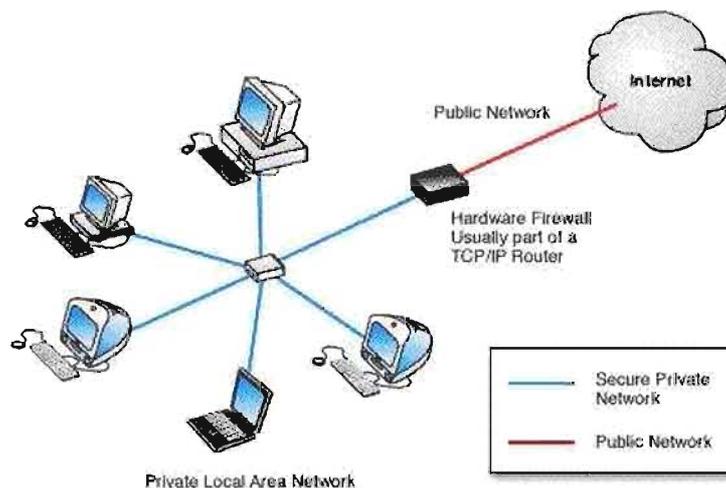


Figura 2.5 Cortafuegos  
Fuente: [17]

## 2.8 CALIDAD DE SOFTWARE

La Calidad de Software es el 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.

### 2.8.1 FACTORES DE LA CALIDAD ISO 9126 [10]

El estándar ISO 9126 se desarrollo con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad:

**Funcionalidad.** Grado en el que el software satisface las necesidades planteadas según establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

**Confiabilidad.** Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.

**Usabilidad.** Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.

**Eficiencia.** Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos.

**Facilidad de recibir mantenimiento.** Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.

**Portabilidad.** Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.



de la complejidad del software. Los valores de dominio de información se definen en la forma siguiente:

Número de entradas externas (EE). Cada entrada externa se origina de un usuario o se transmite desde otra aplicación, y proporciona distintos datos orientados a aplicación o información de control. Con frecuencia, las entradas se usan para actualizar archivos lógicos internos (ALI). Las entradas deben distinguirse de las consultas, que se cuentan por separado.

Número de salidas externas (SE). Cada salida externa es datos derivados dentro de la aplicación que ofrecen información al usuario. En este contexto, salida externa se refiere a reportes, pantallas, mensajes de error, etc. Los ítems de datos individuales dentro de un reporte no se cuentan por separado.

Número de consultas externas (CE). Una consulta externa se define como una entrada en línea que da como resultado la generación de alguna respuesta de software inmediata en la forma de una salida en línea.

Número de archivos lógicos internos (ALI). Cada archivo lógico interno es un agrupamiento lógico de datos que reside dentro de la frontera de la aplicación y se mantiene mediante entradas externas.

Número de archivos de interfaz externos (AIE). Cada archivo de interfaz externo es un agrupamiento lógico de datos que reside fuera de la aplicación, pero que proporciona información que puede usar la aplicación.

Una vez recolectados dichos datos, la tabla de la figura xxxx se completa y un valor de complejidad se asocia con cada conteo.

Para calcular puntos de función (PF), se usa la siguiente relación:

$$PF = \text{conteo total} \times [0,65 + 0,01 \times \sum(F_i)]$$

Donde conteo total es la suma de todas las entradas PF obtenidas.

Valor de dominio de información	Conteo	Factor ponderado			Resultado
		Simple	Medio	Complejo	
Entradas Externas (EE)	C1	3	4	6	
Salidas Externas (SE)	C2	4	5	7	
Consultas Externas (CE)	C3	3	4	6	
Archivos lógicos internos (ALI)	C4	7	10	15	
Archivos de interfaz externos (AIE)	C5	5	7	10	
				<b>Conteo total</b>	

Tabla 2.1: Calculo de puntos de función

Fuente: [10]

Los  $F_i$  ( $i=1$  a  $14$ ) son factores de ajuste de valor (FAV) con base en respuestas a las siguientes preguntas:

1. ¿El sistema requiere respaldo y recuperación confiables?
2. ¿Se requieren comunicaciones de datos especializados para transferir información hacia o desde la aplicación?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿El desempeño es crucial?
5. ¿El sistema correrá en un entorno operativo, existente enormemente utilizado?
6. ¿El sistema requiere entrada de datos en línea?
7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?
8. ¿Los archivos lógicos internos se actualizan en línea?
9. ¿Las entradas, salidas, archivos o consultas son complejos?
10. ¿El procesamiento interno es complejo?
11. ¿El código se diseña para ser reutilizable?
12. ¿La conversión y la instalación se incluyen en el diseño?
13. ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?
14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?

Cada una de estas preguntas se responde usando una escala que varía de 0 (no importante o aplicable) a 5 (absolutamente esencial).

El valor obtenido PF es sometido a la siguiente comparación:

$300 < PF$  : Optimo

$200 < PF \leq 300$  : Bueno

$100 < PF \leq 200$  : Suficiente

PF <= 100 : Deficiente

### 2.8.2.2 Mantenibilidad [10]

Para medir la mantenibilidad del sistema se utilizan los índices de madurez del software (IMS) según el IEEE 982.1-1988, este nos proporciona una indicación de la estabilidad basado en los cambios presentados en cada versión durante el desarrollo del sistema.

$$\text{IMS} = (\text{MT} - (\text{Fc} + \text{Fa} + \text{Fe}) / \text{MT}$$

Donde:

MT = Número de módulos en la versión actual

Fc = Número de módulos en la versión actual que se han cambiado

Fa = Número de módulos en la versión actual que se han añadido

Fe = Número de módulos en la versión actual que se han eliminado

A medida que el sistema se aproxima a 1 el producto se pone más estable según la siguiente relación:

75% <= IMS <= 100% es Optima

50% <= IMS < 75% es Buena

25% <= IMS < 50% es Suficiente

0% <= IMS < 25% es Deficiente

### 2.8.2.3 Confiabilidad [18]

En términos estadísticos se define como la probabilidad de operación libre de fallos del sistema en un entorno determinado y durante un tiempo.

El nivel de confiabilidad del sistema está estrictamente relacionado con la cantidad de errores que lanza el sistema durante el tiempo de ejecución y no en tiempo real (tiempo de reloj). Se expresa en una escala de 0 a 1, en el que el sistema será muy confiable si el valor obtenido de confiabilidad es cercano a 1 y viceversa. Es decir que a menos errores que genere el sistema el sistema será más confiable ante los ojos del usuario.

La confiabilidad del sistema denotado por  $R(t)$  está basado en la confiabilidad de cada uno de sus módulos en un determinado tiempo  $t$  se define como la probabilidad de que su tiempo para fallar excede a  $t$ ; es decir:

$$R(t) = P[T > t] = 1 - F(t)$$

Considerando que el tiempo  $T$  para fallar es una variable exponencial, se tiene

$$R(t) = 1 - (1 - e^{-\lambda t})$$

Donde:

$$R(t) = e^{-\lambda t}$$

$R(t)$ : Confiabilidad del sistema en el tiempo  $t$

$\lambda$ : Probabilidad que se presente una falla durante un intervalo de tiempo

$t$ : Periodo de operaciones en tiempo  $t$

Para encontrar el valor de la confiabilidad del sistema a partir del diagrama de transferencia de todos los módulos que lo conforman, se hace uso de lo siguiente:

a) Si  $n$  componentes, que funcionan independiente, están conectados en serie, y el  $i$ -ésimo componente tiene confiabilidad  $R_i(t)$ , entonces la confiabilidad del sistema completo,  $R(t)$  está dado por

$$R(t) = R_1(t) * R_2(t) * \dots * R_n(t)$$

b) Si  $n$  componentes que funcionan independientemente, actúan en paralelo y la  $i$ -ésima componente tiene confiabilidad  $R_i(t)$ , entonces la confiabilidad del sistema está dado por:

$$R(t) = 1 - [1 - R_1(t)] * [1 - R_2(t)] * \dots * [1 - R_n(t)]$$

#### **2.8.2.4 Portabilidad**

Portabilidad es el esfuerzo necesario para transferir el programa de un entorno de sistema hardware y/o software a otro, la factibilidad de instalación se calcula con:

$$X = A / B$$

Donde:

A: N° de intentos exitosos

B: N° de intentos

#### **2.8.2.5 Usabilidad**

Para determinar el factor de usabilidad haremos uso de tres métricas las cuales ayudaran a decidir cuan usable es el sistema, estas métricas son: completitud de la descripción, cuyo propósito es mostrar que proporción de las funciones (casos de uso) es entendible.

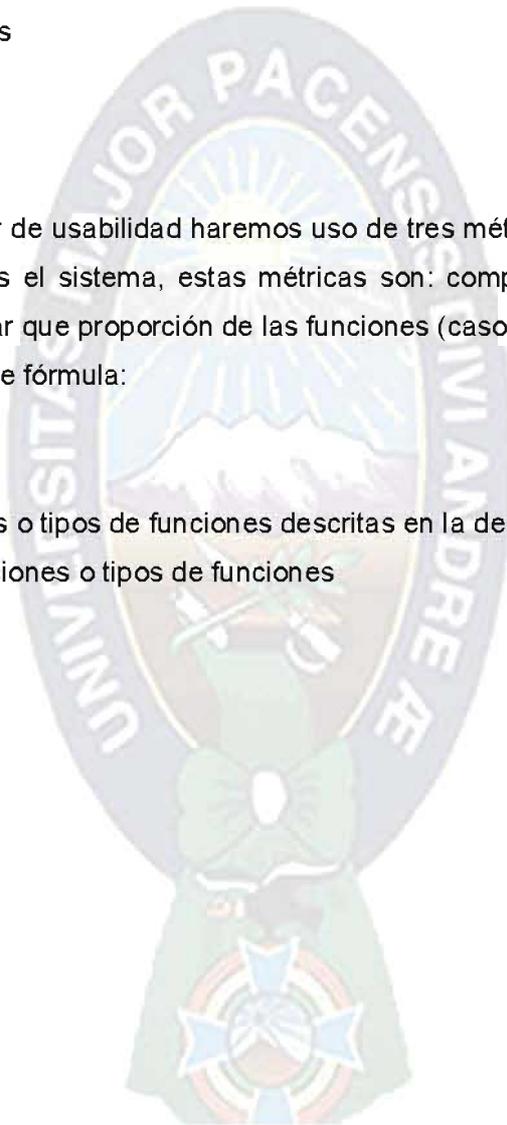
Esta dada por la siguiente fórmula:

$$X = A / B$$

Donde:

A = Número de funciones o tipos de funciones descritas en la descripción del producto

B = Número total de funciones o tipos de funciones



# Capítulo III

## MARCO APLICATIVO

### 3.1. Análisis, Desarrollo e Implementación

Analizando los procesos del Sistema de Programación de Operaciones realizados en el SEDES La Paz, direccionados por la Unidad de Planificación y Desarrollo Organizacional, se planteo la construcción de una herramienta que facilite la programación de operaciones anual.

Luego de haber realizado una investigación acerca de las metodologías más destacadas, se vio por conveniente utilizar la metodología PU complementadas con WSDM dentro de las fases del PU, considerando las ventajas de ambos métodos para resolver el problema planteado, además de la utilización de la herramienta de descripción de UML.

La metodología PU comprende las siguientes fases:

- FASE DE INICIO
- FASE DE ELABORACIÓN
- FASE DE CONSTRUCCION
- FASE DE TRANSICION

A continuación se presenta las etapas del modelo que se propone para el diseño del sistema.

### 3.2 Etapas de la Metodología PU-WSDM

En las fases de Inicio y Elaboración, se indican y describen cada una de las etapas de la metodología PU-WSDM, mismas que constituyen la configuración desde la perspectiva de artefactos, que se propone para este proyecto son los siguientes:

- Modelado del Negocio
- Modelado de Usuarios
- Modelado de Requisitos
- Modelado de Análisis
- Modelado de Diseño
- Modelado de Navegación
- Modelado de Implementación
- Modelado de Despliegue
- Modelado de Pruebas

Flujos de Trabajo Funcionales	FASES			
	Inicio	Elaboración	Construcción	Transición
Modelado del Negocio		Diag. de Casos de Uso del Negocio Modelo de Objetos del Negocio		
Modelado de Usuarios		Clasificación de los Usuarios Descripción de los Usuarios		
Modelado de Requisitos		Diag. de Casos de Uso del Sistema Diagrama de Actividades		
Modelado de Análisis		Diagrama de Secuencias		
Modelado Diseño		Diagrama de clases Modelo Relacional		
Modelado de Navegación		Estructura del Sitio Web Clases Navegacionales		
Modelado de Implementación		Diagrama de Componentes Pantallas del Sistema		
Modelado de Despliegue		Diagrama de Despliegue Modelo de despliegue		
Modelado de Pruebas		Prueba de Caja Blanca Prueba de Caja Negra		

Figura 3.1 Fases de la metodología PU - WSDM  
Fuente: Elaboración propia

### 3.2.1 Modelado del Negocio

El propósito del modelo del negocio es definir una infraestructura de información adecuada, que permita contar con un espacio que dé a conocer las fuentes de información generadas por los diferentes actores en la gestión (instituciones y organizaciones generadoras de información)

Un modelo comprende un conjunto de paquetes que contienen los elementos del mismo, es una descripción completa de un sistema.

La siguiente figura representa el diagrama del modelado de negocio del “Sistema de programación de operaciones” con todos los subsistemas que intervienen en el sistema.

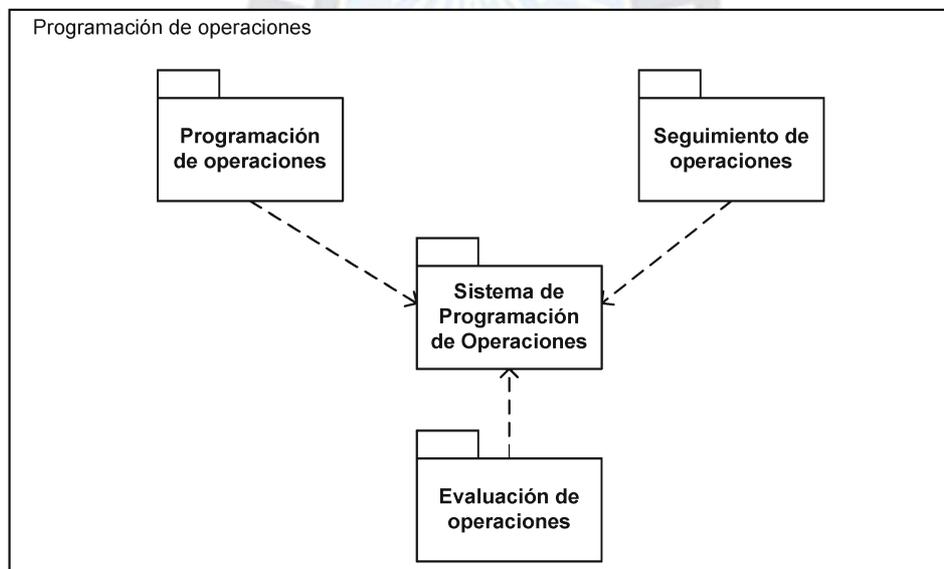


Fig. 3.2. Modelado del negocio  
Fuente: Elaboración propia

#### 3.2.1.1 Diagrama de casos de uso

Los casos de uso más importantes para el sistema de manera general son los que se detallan a continuación:

- Programación de operaciones
- Seguimiento de operaciones
- Evaluación de operaciones

## Casos de Uso del Negocio



Fig. 3.3 Diagrama de casos de uso del negocio  
Fuente: Elaboración propia

### 3.2.1.2 Modelo de Objetos

Los modelos de objetos del negocio están asociados a cada uno de los casos de uso.



Fig. 3.4 Modelo de objetos de programación y seguimiento de operación  
Fuente: Elaboración propia

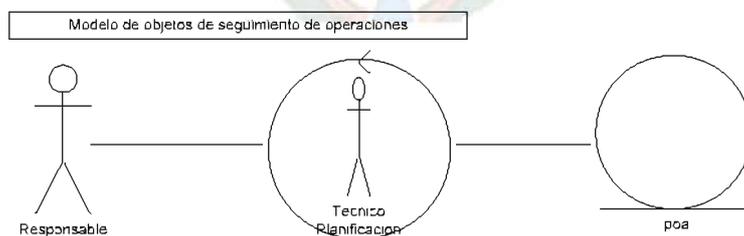


Fig. 3.5 Modelo de objetos de programación y seguimiento de operación

Fuente: Elaboración propia

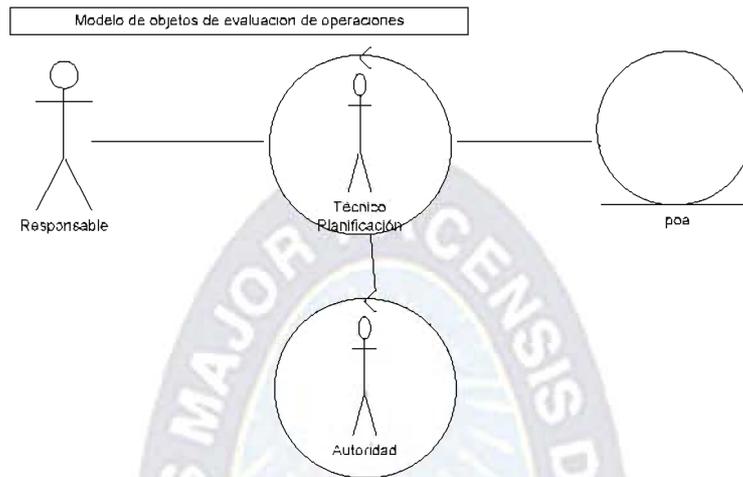


Fig. 3.6 Modelo de objetos de programación y seguimiento de operación  
Fuente: Elaboración propia

### 3.2.2 Modelado de Usuarios

El modelado de usuarios se basa en definir clases o perfiles de los usuarios agrupados según sus limitaciones de acceso, distinguiendo entre tipos de limitaciones similares o equivalentes, y así poder diseñar con el objetivo de satisfacer las necesidades propias de cada grupo de usuarios. Este modelado se realiza en base a la información extraída de los usuarios.

#### 3.2.2.1 Clasificación de Usuarios

La Institución interactúa con distintos usuarios para un mejor entendimiento lo vemos en la siguiente figura:

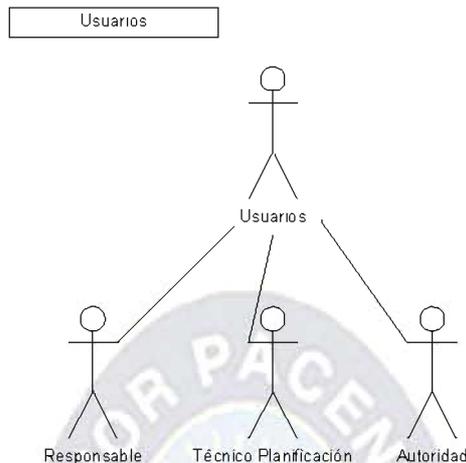


Fig.3.7 Diagrama de Usuarios del Sistema  
Fuente: Elaboración propia

### 3.2.2.2 Descripción de Usuarios

Dependiendo del perfil del tipo de Usuario que sea, le mostrarán unos datos u otros.

Su perfil de usuario, en cuyo caso pueden ser:

- Un responsable es el encargado de una Unidad o un área organizacional, que solo tiene los permisos de dar de alta, modificar o consultar información de su área.
- El Técnico de planificación es el Administrador del Sistema, es el que tiene los permisos para dar de alta, modificar o consultar cualquier información.
- La Autoridad es una abstracción, que puede corresponder al Director, Coordinador de dirección o Jefe Unidad de Planificación y Desarrollo Organizacional, los que podrán consultar información resumen relevante.

### 3.2.3 Modelado de Requisitos

El propósito del modelado de requisitos es analizar y definir las necesidades y características requeridas para desarrollar el sistema.

### 3.2.4 Descripción de Requerimientos

El Sistema de Programación de Operaciones (SPO) permite al responsable de un área organizacional realizar:

- el acceso al SPO de acuerdo a un identificador y contraseña
- la presentación del SPO de acuerdo al área organizacional
- la programación de operaciones a realizar en referencia al objetivo de gestión institucional a cumplir
- la programación de tareas a realizar en referencia a las operaciones definidas
- la programación de tiempos de cumplimiento de las tareas definidas
- la programación del presupuesto de acuerdo a la tarea a realizar
- el registro de cumplimiento de tareas programadas
- la consulta de programación de operaciones del área
- la impresión de programación de operaciones del área
- la impresión de seguimiento de operaciones del área

El SPO permite a la autoridad realizar:

- el acceso al SPO de acuerdo a un identificador y contraseña
- la consulta de programación de operaciones de áreas
- la evaluación de cumplimiento de la programación de operaciones de áreas
- la impresión de programación de operaciones de áreas
- la impresión de la evaluación de cumplimiento de la programación de operaciones de áreas

El SPO permite al Técnico en planificación realizar:

- el acceso al SPO de acuerdo a un identificador y contraseña
- el registro de objetivos institucionales y resultados de acuerdo a la gestión
- el registro del análisis de situación interno y externo
- el registro de los suministros
- el mantenimiento del análisis de situación, objetivos institucionales, resultados, operaciones, tareas, tiempo de cumplimiento, ejecución de cumplimiento y justificación
- la consulta de programación de operaciones de áreas
- la consulta de seguimiento de operaciones de áreas
- la evaluación de cumplimiento de la programación de operaciones de áreas
- la impresión de programación de operaciones de áreas

- la impresión de la evaluación de cumplimiento de la programación de operaciones de áreas

Los detalles de cómo el sistema cubre los requerimientos se basará básicamente en el modelo de casos de uso.

### 3.2.3.1 Diagrama de Casos de Uso

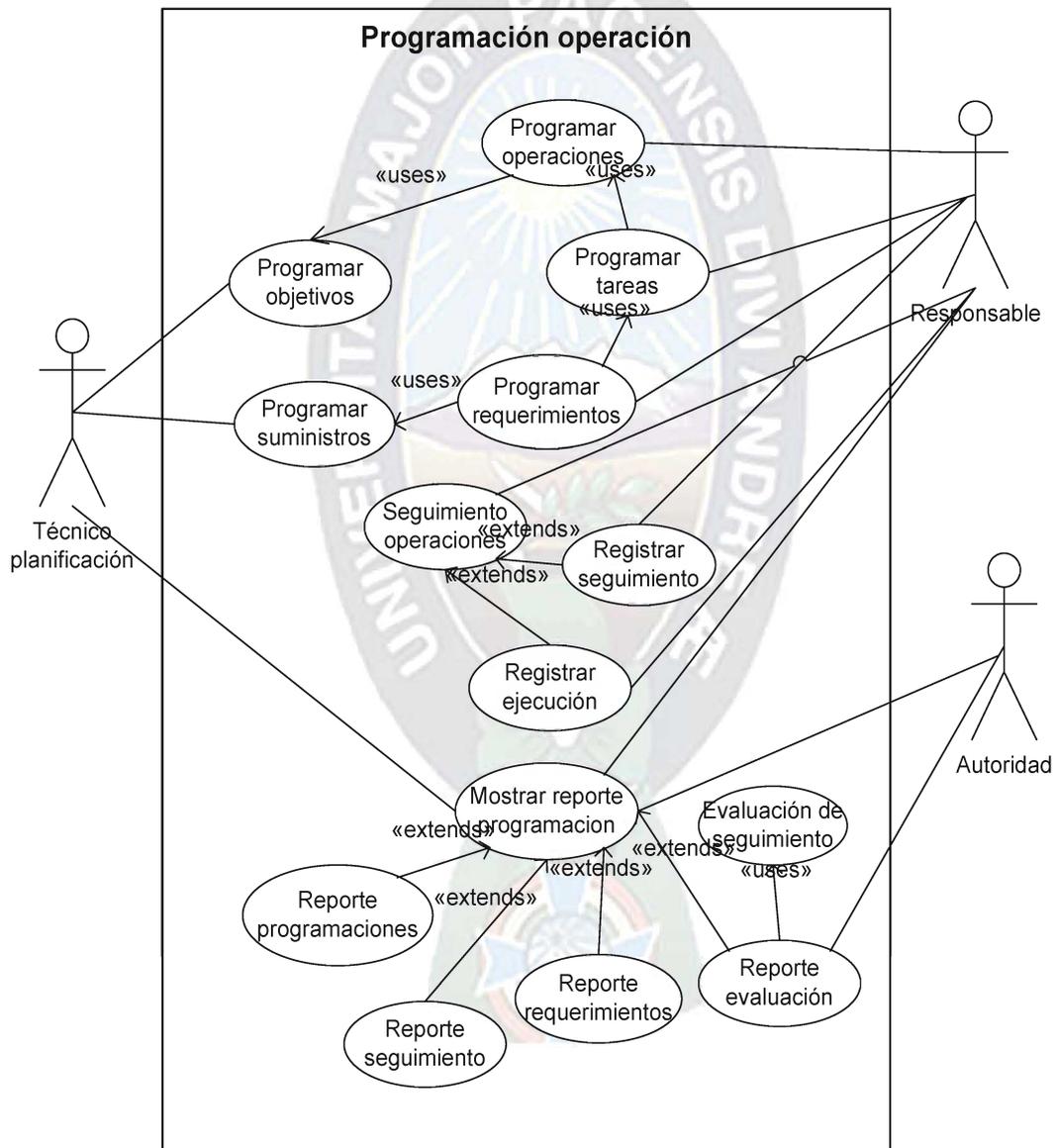


Fig. 3.8 Caso de Uso de Programación de operación  
Fuente: Elaboración propia

### 3.2.3.2 Diagramas de Actividad

Los diagramas de actividades son tareas que deben ser realizadas, cada actividad puede ser seguida por otra actividad. Esta notación de UML es similar al diagrama de flujo que muestra el flujo de control entre actividades; esto implica modelar los pasos secuenciales de un proceso computacional.

A continuación se muestran los diferentes diagramas de actividades del sistema:

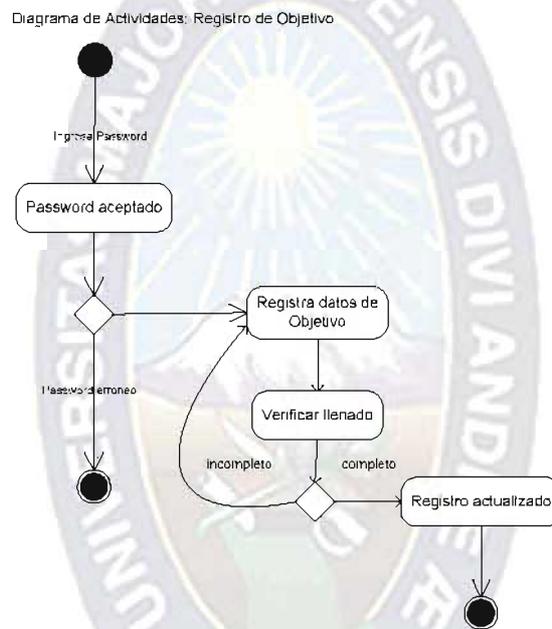


Fig. 3.9 Diagrama de Actividades: Registro de Objetivo  
Fuente: Elaboración propia

Diagrama de Actividades Registro de tareas

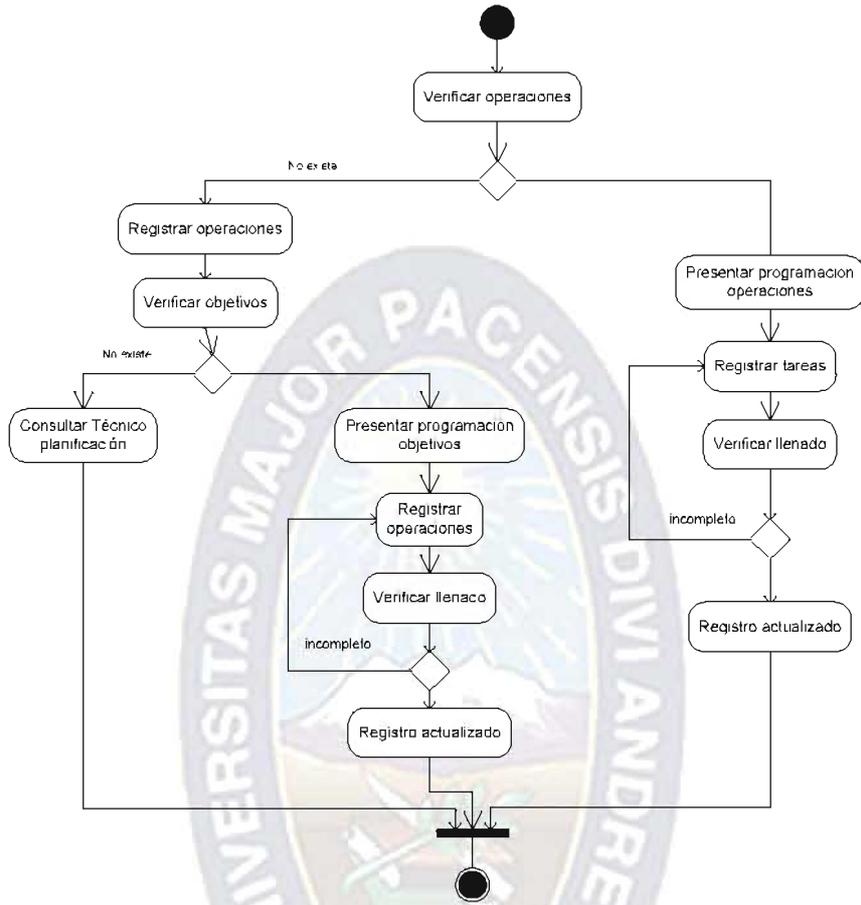
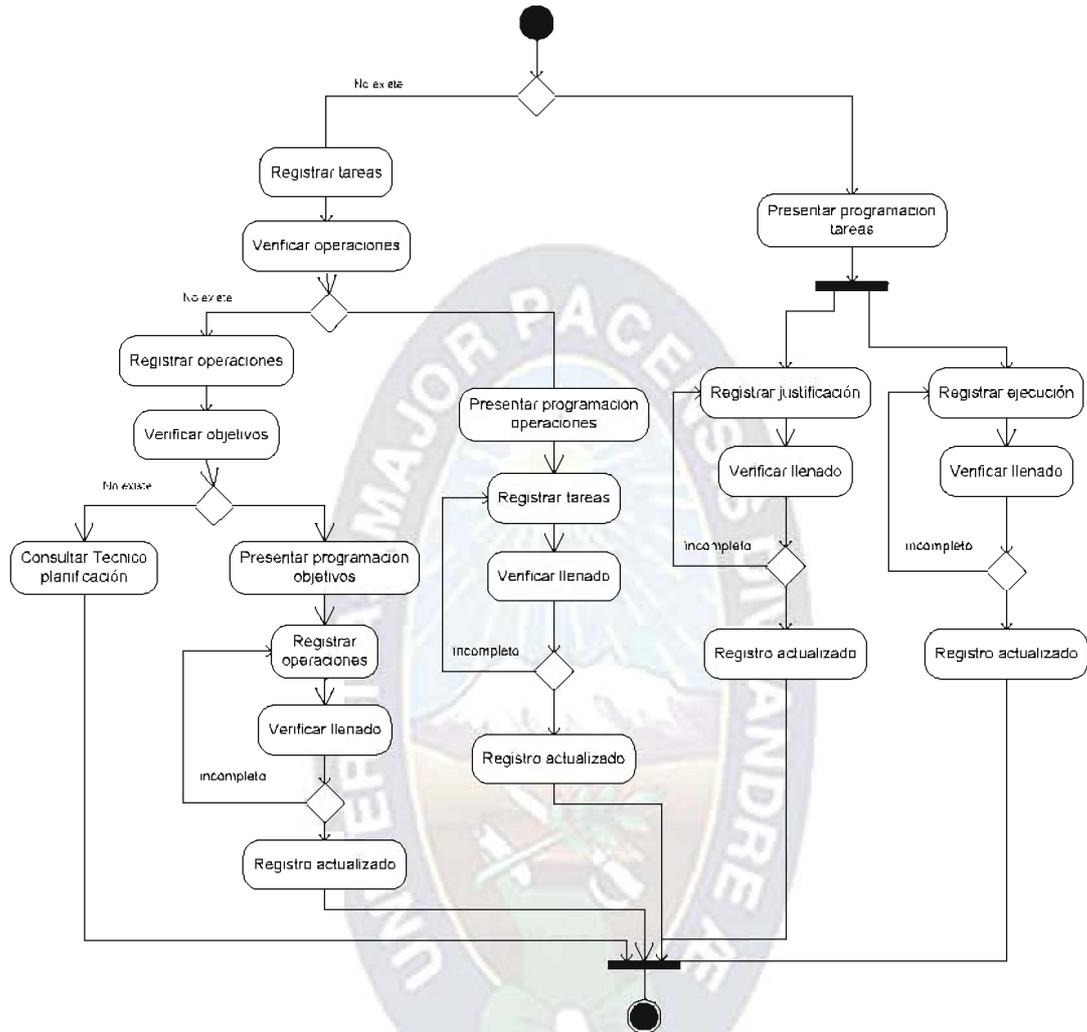


Fig.3.10 Diagrama de Actividades: Registro de operaciones  
Fuente: Elaboración propia

Diagrama de Actividades Registro de seguimiento



Fi

Figura.3.11 Diagrama de Actividades: Registro de seguimiento

Fuente: Elaboración propia

### 3.2.4 Modelado de Análisis

El modelado de análisis ayuda a refinar los requisitos (casos de uso) con más detalle y nos permite refinar sobre los aspectos internos del sistema.

Se logra una aproximación a la fase de diseño. Para el análisis se utilizarán diagramas que servirán de modelo.

### 3.2.4.1 Diagramas de Secuencia

Los diagramas de secuencia dan una descripción gráfica de la interacción del actor con las operaciones que da origen el sistema. Los diagramas de secuencia detallan las etapas en las cuales va a desarrollarse el sistema, por lo que a continuación mostramos los diagramas de secuencia correspondiente a los casos de uso representados en forma gráfica.

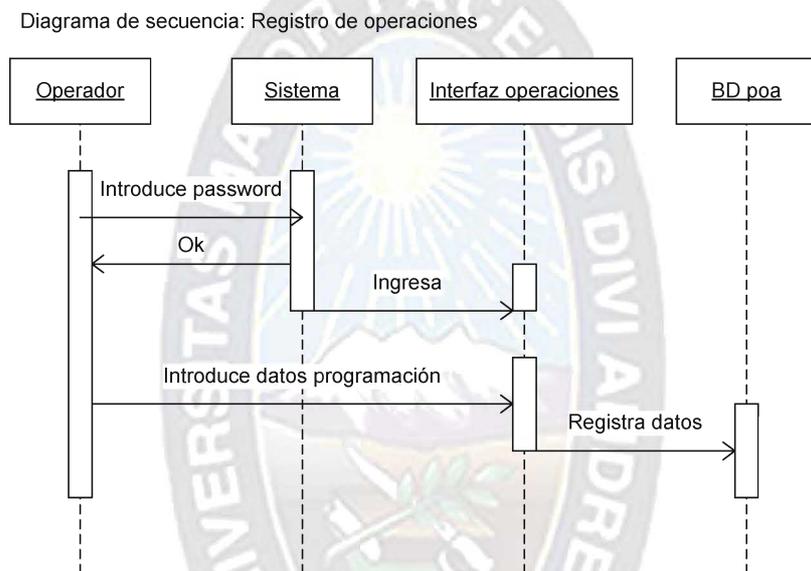


Fig. 3.12 Diagrama de Secuencia: Registro de operaciones base  
Fuente: Elaboración propia

Diagrama de secuencia: Registro de ejecución

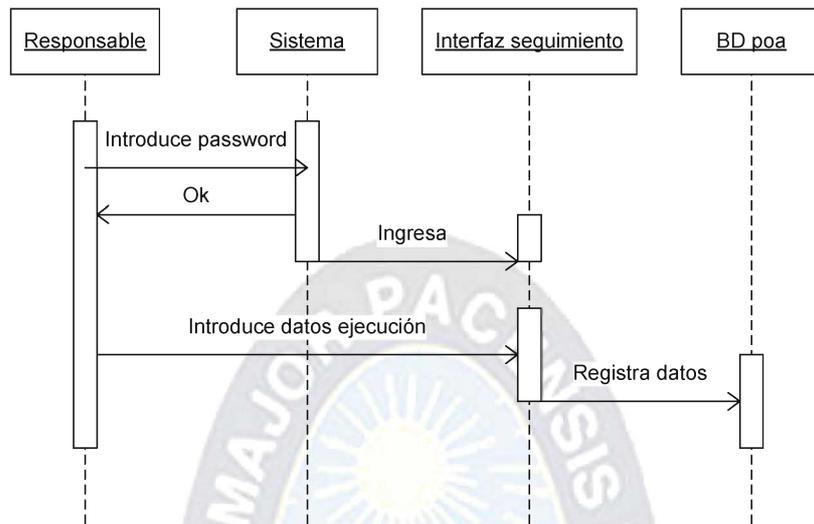


Fig. 3.13. Diagrama de Secuencia: Registro de ejecución operaciones  
Fuente: Elaboración propia

### 3.2.5 Modelado de Diseño

Es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, tienen impacto en el sistema a considerar. Además el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación.

A continuación se presentan los modelos definidos en RUP como modelo de datos y modelo de análisis / diseño. Constará de un diagrama de clases en el que se muestran tan sólo las clases generadas a partir de los casos de uso incorporados a la aplicación y de un modelo de datos (modelo relacional) donde se muestran las entidades que participan en las relaciones definidas en el proyecto.

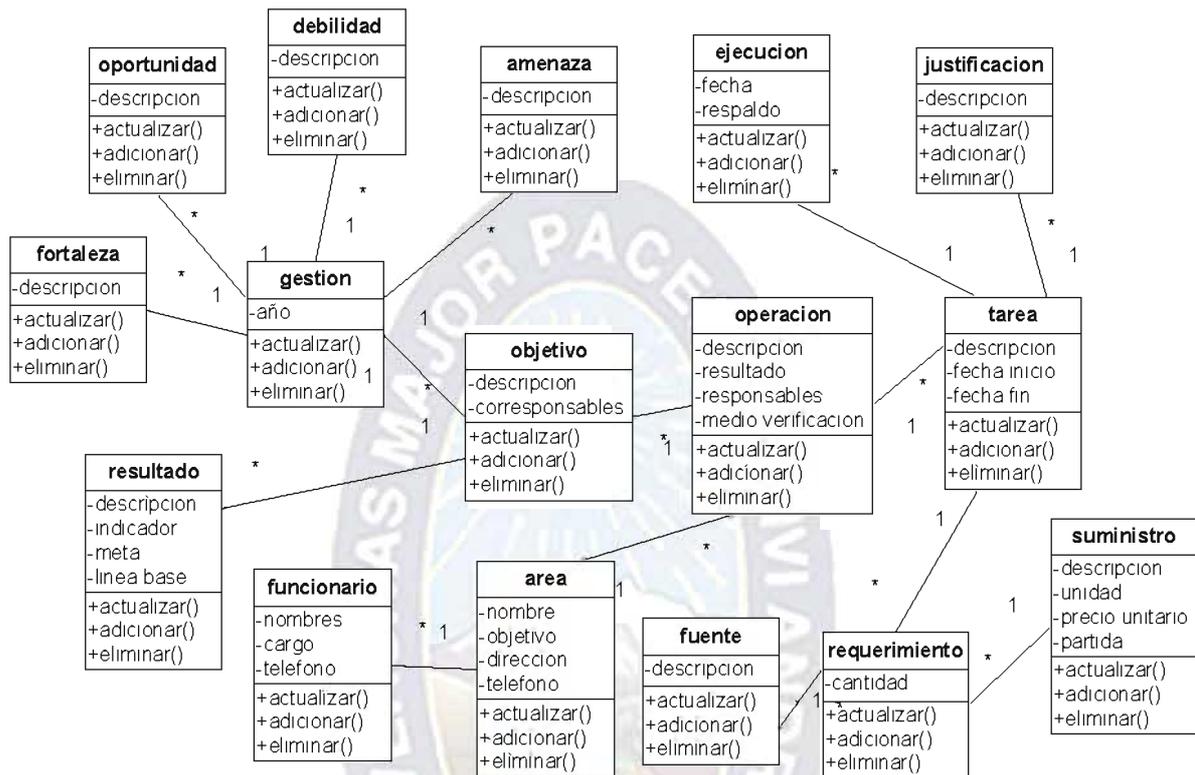


Fig. 3.14. Diagrama de Clases  
Fuente: Elaboración propia

### 3.2.5.2 Modelo Relacional

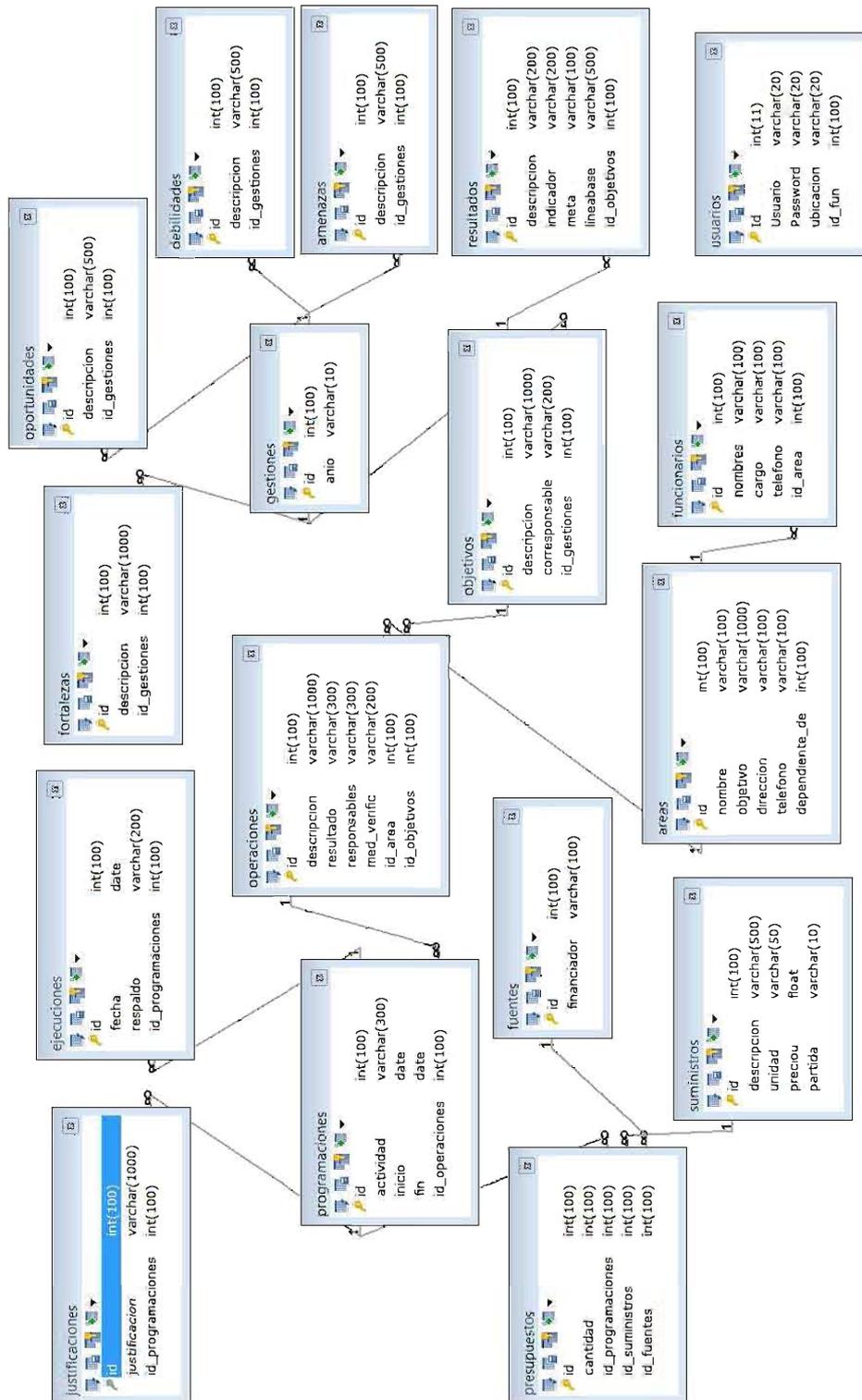


Fig. 3.15. Modelo Relacional  
Fuente: Elaboración propia

### 3.2.6 Modelado de Navegación

En esta Etapa se realiza el diseño de páginas que componen la aplicación. Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual.

#### 3.2.6.1 Estructura del Sitio Web

La navegación de los usuarios define exactamente la orden de aparición de las pantallas, se puede habilitar o no las opciones de un menú dependiendo del tipo de usuario que sea.



Fig. 3.16 Estructura del sitio Web  
Fuente: Elaboración Propia

#### 3.2.6.2 Clases Navegacionales

Se presenta el esquema de clase navegacional para el usuario visitante donde se muestra la visión de navegación enfocada hacia el usuario y las formas de navegación que tiene el mismo en el sitio web .



Fig. 3.17 Esquema de clase navegacional visión Responsable  
Fuente: Elaboración Propia

### 3.2.7 Modelado de Implementación

Describe como los diagramas de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente ejecutable, etc. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros.

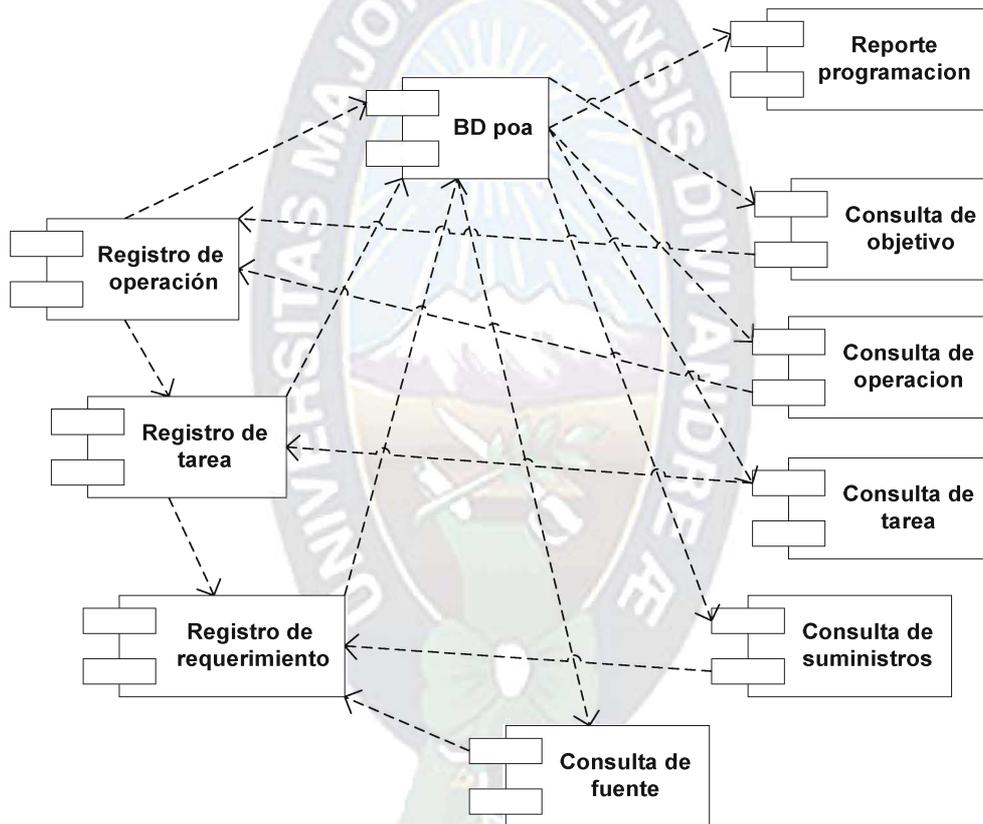


Fig. 3.18 Diagrama de Componentes: Programación de operaciones

Fuente: Elaboración propia

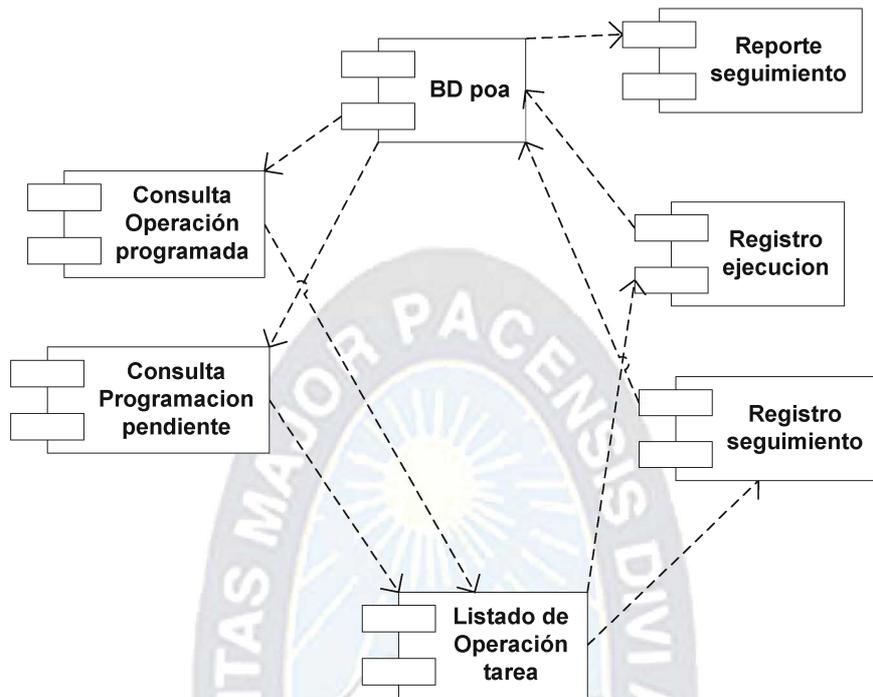


Fig. 3.19 Diagrama de Componentes: Seguimiento de operaciones  
Fuente: Elaboración propia

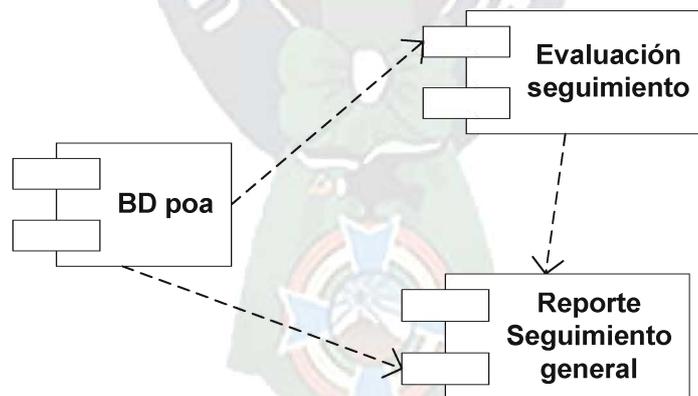


Fig. 3.20 Diagrama de Componentes: Evaluación y reporte de seguimiento general  
Fuente: Elaboración propia

### 3.2.7.2 Pantallas del Sistema

En este punto se presentan algunas de las interfaces del sistema.

La siguiente figura muestra el ingreso al sistema y permite controlar, verificar y mantener la seguridad del sistema de acuerdo a los permisos y contraseñas asignados por el usuario autorizado.

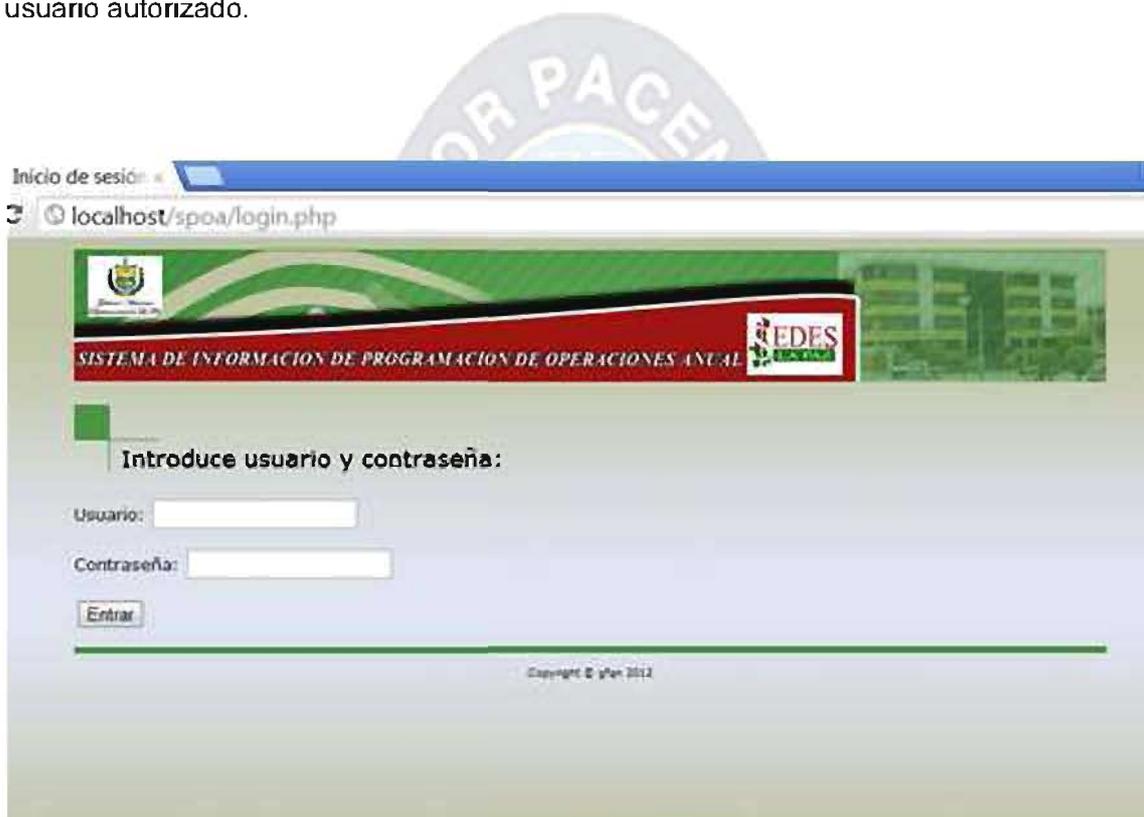


Fig. 3.21 Pantalla de ingreso al sistema  
Fuente: Elaboración propia

En la siguiente figura se observa la interfaz gráfica del usuario responsable, mostrando un menú horizontal de operaciones Principal, Programación, Ejecución, Modificación y Reportes.



Fig. 3.22 Interfaz Gráfica Responsable  
Fuente: Elaboración propia

En la siguiente figura se observa la interfaz gráfica del usuario operador donde en un menú horizontal define operaciones de base operacional, modificación y reportes.



Fig. 3.23 Interfaz Gráfica del usuario operador  
Fuente: Elaboración propia

La Interfaz del módulo de modificación del Sistema, permite realizar las operaciones fundamentales de modificación y eliminación de operaciones, tareas, tiempos, ejecuciones.

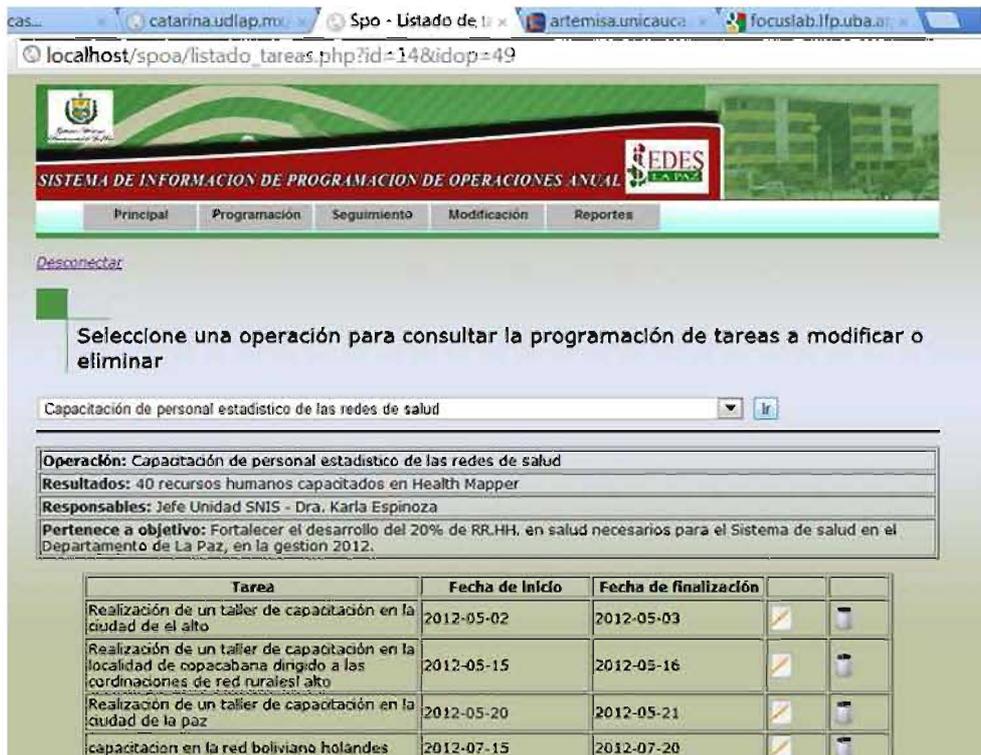


Fig. 3.24 Pantalla de listado de tareas para su posible modificación o eliminación  
Fuente: Elaboración propia



Fig. 3.25 Pantalla selección de tareas programadas para registro de ejecución  
Fuente: Elaboración propia

Los reportes que obtiene el sistema son en formato pdf dando la seguridad correspondiente a dicho reporte donde tiene las facilidades de guardar e imprimir.

Sistema de Programación de Operaciones - Reporte de Programación de Operaciones

**UNIDAD DEPARTAMENTAL DEL SISTEMA DE INFORMACION EN SALUD**  
**RESPONSABLE:** Karla Espinoza  
**CARGO:** Jefe Unidad Departamental del Sistema de Información en Salud

**OPERACION 1:** Capacitación de personal establico de las redes de salud  
**RESULTADO:** 40 recursos humanos capacitados en Health Mapper  
**RESPONSABLES:** Jefe Unidad SNIS - Dra. Karla Espinoza  
**MEDIO DE VERIFICACIÓN:** 40 certificados de capacitación extendidos por el MByD - SEDES  
**PERTENECE AL OBJETIVO:** Fortalecer el desarrollo del 20% de RR.HH. en salud necesarios para el Sistema de salud en el Departamento de La Paz, en la gestión 2012.

Nº	Título	Fecha Inicio	Fecha Fin	Fecha Reporte	Documento	Avance
1	Realización de un taller de capacitación en la ciudad de el alto	2012-05-02	2012-05-03	2012-06-06	Informe CITE:SNIS14302012	
2	Realización de un taller de capacitación en la localidad de capacita dirigido a las comunidades de red rural alba	2012-05-15	2012-05-16	2012-05-18	Informe CITE:SNIS15092012	
3	Realización de un taller de capacitación en la ciudad de la paz	2012-05-20	2012-05-21	2012-06-29	Informe CITE:SNIS15092012	
4	capacitación en la red boliviano holandés	2012-07-15	2012-07-20			
5	capacitación en la red los andes	2012-06-01	2012-07-20			

EL AVANCE FISICO DE LA OPERACIÓN 1 ES DE 60 %

**OPERACION 2:** Ajuste del Manual de Organización y Funciones del personal dependiente del SNIS SEDES LA PAZ.  
**RESULTADO:** Desempeño de funciones del personal del SNIS mejorado  
**RESPONSABLES:** Jefe Unidad SNIS - Dra. Karla Espinoza  
**MEDIO DE VERIFICACIÓN:** Informe técnico

Fig. 3.26 Pantalla Reporte de seguimiento de ejecución de operaciones  
 Fuente: Elaboración propia

### 3.2.8 Modelado de Despliegue

Es la etapa del desarrollo que describe la configuración del sistema para su ejecución en un ambiente del mundo real. Para el despliegue, se deben tomar decisiones sobre los parámetros de la configuración, funcionamiento, asignación de recursos, distribución y concurrencia .

#### 3.2.8.1 Diagrama de Despliegue

Define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos hardware sobre los cuales pueden ejecutarse los elementos software. Con frecuencia conocemos como será la arquitectura física del sistema antes de comenzar su desarrollo. Por tanto, podemos modelar los nodos y las conexiones del modelo de despliegues tan pronto como comience el flujo de trabajo de los requisitos.

Diagrama de Despliegue: Sistema de Programación de operaciones

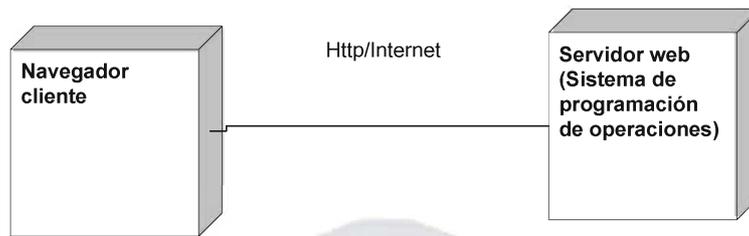


Fig. 3.27 Diagrama de Despliegue: Sistema de programación de operaciones

Fuente: Elaboración propia

### 3.2.9 Modelado de Pruebas

La prueba debe ocurrir durante todo el ciclo de vida: podemos probar la funcionalidad de los primeros prototipos; probar la estabilidad, cobertura y rendimiento de la arquitectura; probar el producto final, etc.

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

Según Pressman, se realizan dos tipos de pruebas: las pruebas de caja blanca y las pruebas de caja negra.

#### 3.2.9.1 Pruebas de Caja Blanca

La prueba de camino básico es una técnica de prueba de caja blanca que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental.

#### Complejidad Ciclomática

La complejidad ciclomática es una métrica de software que proporciona una medida cuantitativa de la complejidad lógica de un programa.

La complejidad ciclomática se basa en la teoría gráfica y se calcula de una de tres maneras que son:

El número de regiones corresponde a la complejidad ciclomática.

La complejidad ciclomática  $V(G)$ , de una gráfica de flujo,  $G$  se define como:

$$V(G) = A - N + 2$$

Donde:

$A$  = Número de aristas.

$N$  = Número de nodos de la gráfica de flujo.

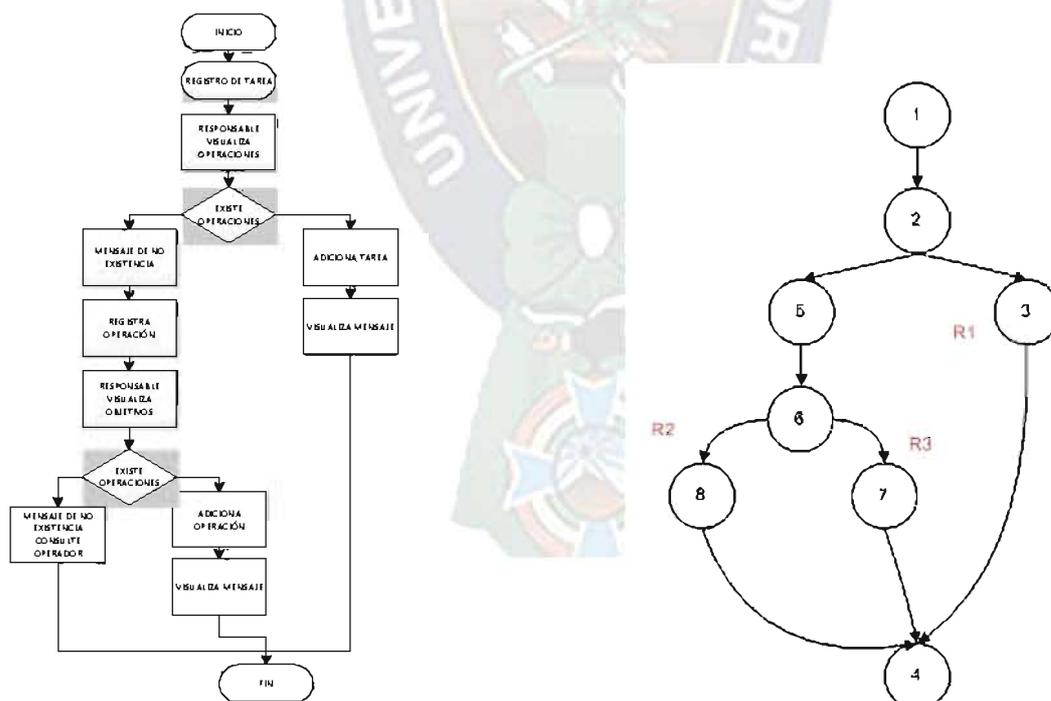
La complejidad ciclomática  $V(G)$ , de una gráfica de flujo  $G$  también se define como:

$$V(G) = NP + 1$$

Donde:

$NP$  = Es el número de nodos predicado incluidos en la gráfica de flujo  $G$

Para realizar esta prueba se tomara en cuenta el diseño procedimental del módulo de Registro de Asistencia y el diseño del grafo respectivo como muestra la siguiente figura.



De acuerdo al grafo se obtiene los siguientes caminos independientes

Camino1: 1-2-3-4

Camino2: 1-2-5-6-7-4

Camino3: 1-2-5-6-8-4

Ahora se halla la Complejidad Ciclomática

$$V(G) = \# \text{ de Regiones} = 3$$

$$V(G) = A - N + 2 = 9 - 8 + 2 = 3$$

$$V(G) = NP + 1 = 2 + 1 = 3$$

Por tanto la complejidad ciclomática es 3, esto quiere decir que existen 3 caminos independiente para el proceso de índice de establecimientos.

### **3.2.9.2 Pruebas de Caja Negra**

Este tipo de pruebas demuestran la funcionalidad operativa del sistema. Esto se puede realizar mediante las pruebas de corrida del programa realizadas para verificar el funcionamiento correcto de cada uno de los módulos del sistema.

En el presente proyecto las pruebas de caja negra se realizan durante todo el proceso de construcción del software, mediante la verificación y corrección del código fuente utilizando las pruebas de corrida para encontrar las posibles fallas y poder solucionar hasta encontrar los resultados esperados.

La funcionalidad del sistema se verifica con las pantallas de corrida.

### **3.2.9.3 Seguridad del Sistema**

El Sistema de Programación de Operaciones solicita un identificador de usuario y una contraseña para poder iniciarse, esta información está protegida mediante un código único de 32 caracteres, generados por un algoritmo de encriptación (MD5).

La aplicación para el administrador, utiliza autenticación del usuario mediante identificador y contraseña, tiene implementado una comprobación de sesión de usuario para restringir el acceso de personas no autorizadas.

El Sistema de Programación de Operaciones se encuentra instalado y prestando sus servicios en la Red LAN del SEDES, con la protección de un Cortafuegos (Firewall) Shorewall.



# Capítulo IV

## EVALUACIÓN DEL SISTEMA

### 4.1 Calidad de Software

En la medición de la calidad de un producto de software se emplean modelos que especifican la calidad mediante definición de un conjunto de atributos o características que pueden ser medidos mediante métricas, las cuales analizaremos a continuación.

Para el presente proyecto se tomarán en cuenta los siguientes atributos de calidad:

- Funcionalidad
- Mantenibilidad
- Confiabilidad
- Portabilidad
- Usabilidad

#### 4.1.1 Funcionalidad

El punto de función se determina mediante la sumatoria del producto entre la cuenta y el Factor Ponderado de la Tabla No. 4.1.1

PARÁMETROS DE MEDICIÓN	FACTORES PONDERADOS				
	CUENTA		Ponderación		
NUM. DE ENTRADAS	25	x	3	=	75
NUM. DE SALIDAS	34	x	4	=	136
NUM. DE PETICIONES	19	x	3	=	57
NUM. ARCHIVOS	13	x	7	=	91
NUM. DE INTERFACES EXTERNAS	3	x	5	=	15
CUENTA TOTAL					374

**Tabla No. 4.1: Métricas de Punto Función**  
**Fuente: Elaboraciones Propia**

Para calcular el Factor de Ajuste se debe dar valores empíricos a cada una de las catorce preguntas que se formulan en la siguiente tabla.

ESCALA	NO INFLUENCIA	INCIDENTAL	MODERAD	MEDIO	SIGNIFICATIVO	ESENCIAL
	0	1	2	3	4	5
<b>FACTOR</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
¿Requiere el sistema copias de seguridad y de recuperación fiable?					x	
¿Se requiere comunicación de datos?					x	
¿Es crítico el rendimiento?		x				
¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?				x		
¿Requiere el sistema entrada interactiva?				x		
¿Se actualizaran los archivos maestros de forma interactiva?			x			
¿Son Complejas las entradas, las salidas, los archivos ó las peticiones?			x			
¿Es complejo el proceso interno?				x		
¿Están incluidas en el diseño la conversión y la instalación?				x		
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?					x	
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?					x	

**Tabla No. 4.2: Matriz para el Factor de Ajuste**  
Fuente: Elaboraciones Propia

Por lo tanto El Factor de Ajuste es igual a:

$$\sum Fi = 4 + 4 + 1 + 3 + 3 + 2 + 2 + 3 + 3 + 4 + 4 = 33$$

Reemplazando a la ecuación para un nivel de confianza del 65%

$$\text{Ec. : PF} = \text{Cuenta total} * (0.65 + 0,01 * \sum Fi)$$

Se tiene:

$$PF_{\text{Real}} = \text{Cuenta}_{\text{total}} * [0.65 + (0.01 * \sum Fi)]$$

$$PF_{\text{Real}} = 374 * [0,65 + (0,01 * 33)]$$

$$PF_{\text{Real}} = 366,5$$

Ahora calculamos para un nivel de confianza del 100%

$$PF_{\text{Esperado}} = \text{Cuenta}_{\text{total}} * [1 + (0.01 * \sum Fi)]$$

$$PF_{\text{Esperado}} = 355,3 * [1 + (0.01 * 33)]$$

$$PF_{\text{Esperado}} = 487,4$$

El porcentaje de funcionalidad será como sigue:

$$\%PF = PF_{\text{real}} / PF_{\text{esperado}} = 366,5 / 487,4 = 0.75$$

Por lo tanto se tiene una funcionalidad de 75%

#### 4.1.2 Mantenibilidad

Para medir la mantenibilidad del sistema se utilizan los índices de madurez del software (IMS)

$$IMS = (MT - (Fc + Fa + Fe)) / MT$$

Donde:

MT = Número de módulos en la versión actual

Fc = Número de módulos en la versión actual que se han cambiado

Fa = Número de módulos en la versión actual que se han añadido

Fe = Número de módulos en la versión actual que se han eliminado

A medida que el sistema se aproxima a 1 el producto se pone más estable según la siguiente relación:

75% ≤ IMS ≤ 100% es Optima

50% ≤ IMS < 75% es Buena

25% ≤ IMS < 50% es Suficiente

0% ≤ IMS < 25% es Deficiente

Reemplazando valores:

$$IMS = (31 - (6 + 2 + 1)) / 31 * 100 = 71\%$$

Donde:

$$MT = 31 \quad Fc = 6 \quad Fa = 2 \quad Fe = 0$$

El valor obtenido está en el segundo intervalo, con esto se puede afirmar que el sistema tiene una mantenibilidad buena.

### 4.1.3 Confiabilidad

En términos estadísticos se define como la probabilidad de operación libre de fallos del sistema en un entorno determinado y durante un tiempo.

El nivel de confiabilidad del sistema está estrictamente relacionado con la cantidad de errores que lanza el sistema durante el tiempo de ejecución y no en tiempo real (tiempo de reloj). Se expresa en una escala de 0 a 1, en el que el sistema será muy confiable si el valor obtenido de confiabilidad es cercano a 1 y viceversa. Es decir que a menos errores que genere el sistema el sistema será más confiable ante los ojos del usuario.

La confiabilidad del sistema denotado por  $R(t)$  está basado en la confiabilidad de cada uno de sus módulos en un determinado tiempo  $t$  se define como la probabilidad de que su tiempo para fallar excede a  $t$ ; es decir:

$$R(t) = P[T > t] = 1 - F(t)$$

Considerando que el tiempo  $T$  para fallar es una variable exponencial, se tiene

$$R(t) = 1 - (1 - e^{-\lambda t})$$

Donde:

$$R(t) = e^{-\lambda t}$$

$R(t)$ : Confiabilidad del sistema en el tiempo  $t$

$\lambda$ : Probabilidad que se presente una falla durante un intervalo de tiempo

$t$ : Periodo de operaciones en tiempo  $t$

Para encontrar el valor de la confiabilidad del sistema a partir del diagrama de transferencia de todos los módulos que lo conforman, se hace uso de lo siguiente:

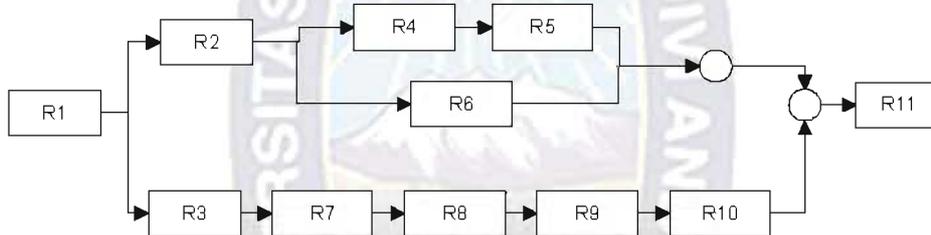
a) Si  $n$  componentes, que funcionan independiente, están conectados en serie, y el  $i$ -ésimo componente tiene confiabilidad  $R_i(t)$ , entonces la confiabilidad del sistema completo,  $R_s(t)$  está dado por

$$R_i(t) = R_1(t) * R_2(t) * \dots * R_n(t)$$

b) Si  $n$  componentes que funcionan independientemente, actúan en paralelo y la  $i$ -ésima componente tiene confiabilidad  $R_i(t)$ , entonces la confiabilidad del sistema está dado por:

$$R_s(t) = 1 - [1 - R_1(t)] * [1 - R_2(t)] * \dots * [1 - R_n(t)]$$

Ahora mediremos la fiabilidad de nuestro sistema, considerando los módulos del mismo.



Donde:

- R1: Ingreso
- R2: Programación base operacional
- R3: Programación operacional
- R4: Programación de objetivos
- R5: Programación de resultados
- R6: Programación de suministros
- R7: Programación de operaciones
- R8: Programación de tareas
- R9: Programación de requerimientos
- R10: Seguimiento de operaciones
- R11: Reportes

Para hallar el tiempo medio de fallo probamos cada uno de los módulos del programa.

Módulos	$\lambda$ tasa de constantes de Prueba de fallo	t periodo de Operaciones en tiempo t	R (t) es la fiabilidad de un componente en el tiempo t
R1	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R2	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R3	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R4	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R5	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R6	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R7	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R8	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R9	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R10	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99
R11	$\lambda = 0.001$	t = 8 horas	R (t) = 0.99

De esta manera utilizando los teoremas 1 y 2 obtenemos:

$$R12=R4*R5=0.98$$

$$R13=\{1-[(1-R12)*(1-R6)]\}=1$$

$$R14=R2*R13=0.99$$

$$R15=R3*R7*R8*R9*R10=0,95$$

$$R16=\{1-[(1-R14)*(1-R15)]\}=1q$$

$$R17=R1*R16*R11=0.98$$

$$\text{Confiabilidad} = R(t) * 100\% = 98\%$$

De acuerdo con el resultado obtenido podemos decir que el Sistema de programación de Operaciones del Servicio Departamental de Salud La Paz presenta una confiabilidad del 98 por ciento entonces podemos afirmar que es un sistema fiable.

#### 4.1.4 Portabilidad

Portabilidad es el esfuerzo necesario para transferir el programa de un entorno de sistema hardware y/o software a otro, la factibilidad de instalación se calcula con:

$$X = A / B = 8 / 9 = 0,89$$

Existe un 89% de que el usuario instale el software con éxito, entonces la portabilidad es óptima.

#### 4.1.5 Usabilidad

Para determinar el factor de usabilidad haremos uso de tres métricas las cuales ayudaran a decidir cuan usable es el sistema, estas métricas son: completitud de la descripción, cuyo propósito es mostrar que proporción de las funciones (casos de uso) es entendible.

Esta dada por la siguiente fórmula:

$$X = A / B$$

Donde:

A = Número de funciones o tipos de funciones descritas en la descripción del producto

B = Número total de funciones o tipos de funciones

Reemplazando en la formula tenemos:

$$X = 11/12 = 92$$

Es decir, existe un 92% de entendimiento de parte de los usuarios con respecto a la capacidad del producto, después de leer la descripción del mismo.

# Capítulo V

## CONCLUSIONES Y RECOMENDACIONES

En este capítulo se dará las conclusiones pertinentes y algunas recomendaciones que son necesarias, las mismas se describen a continuación.

### 5.1 Conclusiones

- Se cumplió con todos objetivos definidos en el presente proyecto.
- Se ordenaron los procesos de programación, seguimiento y evaluación de operaciones de acuerdo a los requerimientos planteados.
- Se centralizó la información de programación de operaciones en una Base de datos logrando unificar la información de programación de operaciones.
- La automatización de los procesos aliviana la carga de esfuerzo laboral al técnico de la unidad de planificación, que cuando este realizaba el seguimiento y evaluación a la programación de operaciones de manera manual, esta tarea tenía un tiempo aproximado de una semana de consolidación y evaluación, sobre información inconsistente.

### 5.2 Recomendaciones

- Se recomienda utilizar el sistema de programación de operaciones para complementar con informaciones del Sistema Nacional de Información en Salud
- Se recomienda ampliar y complementar la automatización de procesos del Sistema de Presupuesto y Sistema de Recursos Humanos.

## BIBLIOGRAFÍA

1. **Kendall, Kenneth E. y Kendall, Julie E.** *ANÁLISIS Y DISEÑO DE SISTEMAS*. Naulcalpan de Juárez, Estado de México : PRENTICE HALL HISPANOAMERICANA S.A., 1997.
2. **Gubernamentales, Ley de Administración y Control.** Contraloría General del Estado. [En línea] 20 de Julio de 1990. <http://www.cge.gob.bo/PortalCGR/uploads/ley1178.pdf>.
3. **SEDES, Unidad de Planificación y Desarrollo Organizacional -.** *Marco Estratégico SEDES La Paz*. [Documento] La Paz : s.n., 2011.
4. **Condori, R.** *Sistema de Control y Seguimiento de Proyectos, Caso: Oficialía Mayor de Protección Social, Dirección de Educación, Gobierno Municipal de El Alto*. La Paz : UMSA, 2011.
5. **Huanca, H.** *Sistema de Control y Seguimiento a Programas Operativos Anuales y (POA's) para el Viceministerio de Biodiversidad, Recursos forestales y Medio ambiente*. La Paz : UMSA, 2009.
6. **James, R., Jacobson I. y Booch G.** *El Proceso Unificado de Desarrollo de Software*. España : Addison Wesley, 2000. *El Lenguaje Unificado de Modelado Manual de Referencia*. España : Addison Wesley, 2000.
7. **Troyer, De.** *Metodo de Diseño de Sitios Web WSDM*. 1997.
8. **Hacienda, Ministerio de.** Normas Básicas del Sistema de Programación de Operaciones. *Resolución Suprema N° 225557*. La Paz Bolivia : s.n., 2006.
9. **Weitzenfeld, A.** *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. Mexico : Thomson Editores S.A., 2005.
10. **Pressman, R.** *Ingeniería de Software, Un enfoque práctico*. Mexico : Mc Graw Hill S.A., 2010
11. WSDM. [En línea] <http://www7.scu.edu.au/1853/com1853.htm>
12. PHP. [En línea] <http://es.wikipedia.org/wiki/PHP>.
13. MySQL. [En línea] <http://es.wikipedia.org/wiki/MySQL>.
14. Apache. [En línea] [http://es.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://es.wikipedia.org/wiki/Apache_HTTP_Server).
15. iso9126. [En línea] <http://iso25000.com/index.php/iso-ieee-9126.html>.

16. Seguridad [En línea] <http://www.uji.es/CA/departaments/icc>

17. Cortafuegos [En línea] <http://www.adrformacion.com/cursos/seguint/leccion3/tutorial1.html>

18. Confiabilidad [En línea]

[http://focuslab.lfp.uba.ar/public/CursoTErrores2k4/Monografias2005/Ana\\_E\\_Luna.pdf](http://focuslab.lfp.uba.ar/public/CursoTErrores2k4/Monografias2005/Ana_E_Luna.pdf)

