UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA INFORMÁTICA

PROYECTO DE GRADO

SISTEMA DE INFORMACION Y SEGUIMIENTO DE PROYECTOS CASO: UNIDAD DESCONCENTRADA SUSTENTAR

MINISTERIO DE DESARROLLO RURAL Y TIERRAS

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

POSTULANTE : *UNIV. CRISPIN ACHO ACHACOLLO*

TUTOR : *LIC. EUFREN LLANQUE QUISPE*

REVISOR : *M. Sc. CARLOS MULLISACA CHOQUE*

LA PAZ – BOLIVIA
2009

RESUMEN

En la actualidad instituciones públicas dedicadas a la gestión de proyectos para el desarrollo del país es la Unidad Desconcentrada SUSTENTAR dependiente del Ministerio de Desarrollo Rural y Tierras – MDRyT. Uno de los Ministerios orientados al manejo de proyectos en diferentes comunidades de nuestro territorio.

Unida Desconcentrada SUSTENTAR no cuenta con un Sistema de Información automatizada y menos con un portal Web que pueda respaldar de manera óptima el trabajo que realiza, ya sea para seguimiento de proyectos que desarrolla en las diferentes comunidades, provincias, municipios, departamentos, etc. que tiene la misión de llevar en los diferentes sectores que el país requiera.

El Sistema de Información y Seguimiento de Proyectos es el encargado de seguimiento a diferentes proyectos que se gestiona mediante la Unidad Desconcentrada SUSTENTAR en los diferentes tipos de proyectos como pequeños, medianos y grandes proyectos en los diferentes sectores a nivel nacional.

La metodología utilizada para el análisis es RUP conjuntamente con el UML que permite la construcción de los modelos de análisis, modelos de diseño en un ciclo iterativo incremental, esto para la fase de inicio y elaboración llegando hasta un diseño final del sistema.

En la implementación del sistema se toma en cuenta la autenticación de administración, la actualización de datos del proyecto como Altas, Bajas y Modificación, así también se encuentra parte del personal quienes debe tener su autenticación para gestionar el sistema. Para el usuario externo es la navegación del portal Web que se pueda verificar en el desarrollo de cada proyecto para la toma de decisiones del mismo.

ÍNDICE

Pág.

CAPÍTULO I

MARCO REFERENCIAL

1.1. Introducción……..……. 1

1.2. Antecedentes……... 1

1.3. Planteamiento del problema…….... 6

1.3.1. Definicición del problema……. 7

1.4. Objetivos…… 7

1.4.1. Objetivo general…….…….... 7

1.4.2. Objetivos específicos…..…….. 8

1.5. Aportes…… 8

1.5.1. Aporte Teórico…….…….... 9

1.6. Justificación…… 10

1.6.1. Justificación Social…..……..... 10

1.6.2. Justificación Técnica……..... 10

1.5.3. Justificación Económica……..……... 10

1.7. Alcances y Límites…..……..... 11

1.7.1. Límites…..……... 11

1.8. Metodología…..…….……..... 11

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción……..…… 13

2.2. Conceptos Generales…… 14

2.2.1. Sistema de Información…… 14

2.2.2. Internet y la Web……. 19

2.3. Marco Conceptual……..... 20

2.3.1. El Paradigma: Orientado a Objetos…..……... 20

2.4. Metodología de Análisis y Diseño……..... 22

2.4.1. Proceso Unificado del Desarrollo de Software……. 22

2.4.2. Ciclo de vida..... 24
2.4.3. Diseño Navegacional..... 25
2.5. El lenguaje de Modelado UML..... 28
2.5.1. Modelo Cliente – Servidor..... 31
2.5.2. Base de Datos Relacional..... 33
2.6. Análisis..... 34
2.7. Diseño..... 38
2.7.1. Modelo y Diseño de la Base de Datos..... 40
2.8. Requerimiento de Software y Hardware..... 41
2.9. Implementación..... 41
2.9.1. Pruebas..... 42
2.9.2. Mantenimiento del Sistema..... 44
2.10. Estándares Aplicables..... 44
2.11. Factores y Métricas de Calidad..... 45
2.11.1. Factores de Calidad ISO 9126..... 45

CAPÍTULO III

MARCO APLICATIVO

3.1. Introducción..... 50
3.2. Recopilación de Necesidades..... 50
3.3. Análisis..... 53
3.3.1. Análisis del Sistema Actual..... 53
3.3.2. Análisis del Nuevo Sistema..... 57
3.4. Modelo Conceptual..... 62
3.5. Diagrama de Secuencia..... 65
3.6. Diagrama de Colaboración 67
3.7. Diagrama de Paquetes..... 68
3.8. Diagrama de Clases..... 68
3.9. Diagrama de Componentes..... 69
3.10. Diagrama Navegacional..... 70
3.11. Diagrama de Distribución..... 71

3.12. Modelo de Base de Datos..... 72
3.13. Diseño e Implementación..... 72
3.14. Modelo de Implementación de Sistemas..... 73
3.15. Diseño de Interfaces..... 74
3.16. Seguridad..... 77
3.16.1. Seguridad Lógica..... 78
3.16.2. Seguridad Física..... 80
3.17. Implantación..... 80
3.18. Fase de Implementación..... 81
3.19. Calidad de Software..... 81
3.19.1.1. Funcionalidad..... 81
3.19.1.2. Completitud de la implementación funcional..... 81
3.19.1.3. Fase de Inicio..... 81
3.19.1.4. Fase de Elaboración..... 82
3.19.2.1. Fase de Construcción..... 82
3.19.2.2. Adecuación Funcional..... 83
3.19.2.3. Fiabilidad..... 84
3.19.3.1. Levantamiento de Defectos..... 84
3.19.3.2. Densidad de Defectos..... 84
3.19.4. Facilidad de Mantenimiento..... 85
3.19.5. Portabilidad..... 86
3.19.5.1. Facilidad de Instalación..... 86
3.19.6. Usabilidad..... 86
3.19.6.1. Completitud de la Descripción..... 87
3.19.6.2. Consistencia Operacional en uso..... 87

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones..... 89
4.2. Recomendaciones..... 90

ÍNDICE DE FIGURAS

FIGURAS

Figura 1. Procedimiento de programas y proyectos..... 3

Figura 2. Proceso de trámite..... 4

Figura 3. Estructura de una Red de Petri..... 9

Figura 2.1. Automatización de procesos..... 16

Figura 2.2. Proceso de Desarrollo de Software..... 22

Figura 2.3. Clase índice..... 27

Figura 2.4. Clase Vuelta Guiada..... 27

Figura 2.5. Clase Consulta..... 28

Figura 2.6. Simbología UML..... 29

Figura 2.7. Simbología UML..... 30

Figura 2.8. Modelo de Tres Capas..... 32

Figura 2.9. Representación de Actores..... 35

Figura 2.10. Diagrama de Casos de Uso..... 35

Figura 2.11. Diagrama de Clases..... 36

Figura 2.12: Diagrama de Estados..... 36

Figura 2.13. Diagrama de Secuencia..... 37

Figura 2.14. Diagrama de Colaboración..... 37

Figura 2.15. Diagrama de Actividades..... 38

Figura 2.16. Diagrama de componentes..... 39

Figura 2.17. Diagrama de Distribución..... 39

Figura 2.18. Prototipo de Interfaz..... 40

Figura 2.19. Diagrama del sistema..... 40

Figura 2.20. Modelo Espiral..... 43

Figura 3.1. Área de Unidad Desconcentrada Sustentar..... 51

Figura 3.2. Actores del sistema Unidad Desconcentrada Sustentar..... 53

Figura 3.3. Diagrama de casos de Unidad Desconcentrada SUSTENTAR..... 55

Figura 3.4. Diagrama de actividades sistema actual..... 56

Figura 3.5. Identificación de actores..... 57
Figura 3.6. Diagrama de casos de uso general..... 59
Figura 3.7. Diagrama de casos de uso general para Jefe Técnico..... 60
Figura 3.8. Modelo Conceptual..... 64
Figura 3.9. Modelo Conceptual..... 65
Figura 3.10. Diagrama de Secuencia. Registro de proyectos..... 66
Figura 3.11. Diagrama de Secuencia. Seguimiento de Proyectos..... 66
Figura 3.12. Diagrama de colaboración. Seguimiento de proyecto..... 67
Figura 3.13. Diagrama de colaboración. Registro de Proyectos..... 67
Figura 3.14. Diagrama de paquetes..... 68
Figura 3.15. Diagrama de Clases..... 69
Figura 3.16. Diagrama de Componentes..... 70
Figura 3.17. Diagrama Navegacional..... 71
Figura 3.18. Diagrama de componentes. Distribución del sistema..... 71
Figura 3.19. Diagrama de Entidad Relación..... 72
Figura 3.20. Diagrama Jerárquico..... 74

MARCO REFERENCIAL

CAPITULO I

1.1. INTRODUCCIÓN

En la actualidad la información es una herramienta indispensable, en cualquier entidad ya sea esta pública o privada, la tecnología informática ha llegado a ser el recurso con mayor auge en las organizaciones, resulta muy difícil imaginar que una organización pueda mantener su crecimiento y estabilidad en un medio cada vez mas competitivo sin apoyo de la tecnología informática y sin los sistemas de información adecuados. Se puede decir que depende de una buena información el éxito o fracaso de una entidad.

En nuestro país (Bolivia), el Ministerio de Desarrollo Rural y Tierras - MDRyT ve la necesidad de contar con un sistema de información y seguimiento de proyectos, que colabore con el trabajo de Proyecto de Desarrollo Rural y Tierras considerando las leyes, reglamentos y normas que rigen en dicho Ministerio.

Entre las principales funciones del Ministerio se encuentran, la de tener un control de los proyectos que se desarrollan en las diferentes comunidades del área rural, con la finalidad de facilitar y agilizar sus tramites, así proporcionar información oportuna tanto a las organizaciones locales como nacionales.

De ahí surge la iniciativa de realizar el presente trabajo, en el que se pretende desarrollar un Sistema de Información y Seguimiento de Proyectos para la Unidad Desconcentrada Sustentar dependiente del MDRyT que gestione la información generada en cada proceso para llevar a cabo un proyecto.

1.2. ANTECEDENTES

Se puede definir PROYECTO como un conjunto de actividades interdependientes orientadas a un fin específico, con una duración predeterminada. Completar con

éxito el PROYECTO, significa cumplir con los objetivos dentro de las especificaciones técnicas, de costo y de plazo de terminación. A un conjunto de Proyectos orientados a un objetivo superior se denomina PROGRAMA, y un conjunto de Programas constituye un PLAN, como corresponde generalmente a los grandes Planes Nacionales [Mendoza 2005].

El MDRyT alberga una gran cantidad de funciones por su amplia estructura, donde la información se considera un recurso valioso que debe administrarse de manera adecuada para el logro de sus objetivos como Ministerio al servicio del país.

La unidad SUSTENTAR se constituye en el marco del Decreto Supremo No. 29315 de 17 de octubre de 2007, es la unidad encargada de poner en marcha el proceso de apoyo a las organizaciones sociales con inversiones a través de transferencias directas no reembolsables del Tesoro General de la Nación (TGN) y de la cooperación internacional para su ejecución y el manejo integral, diversificado para la elaboración de proyectos.

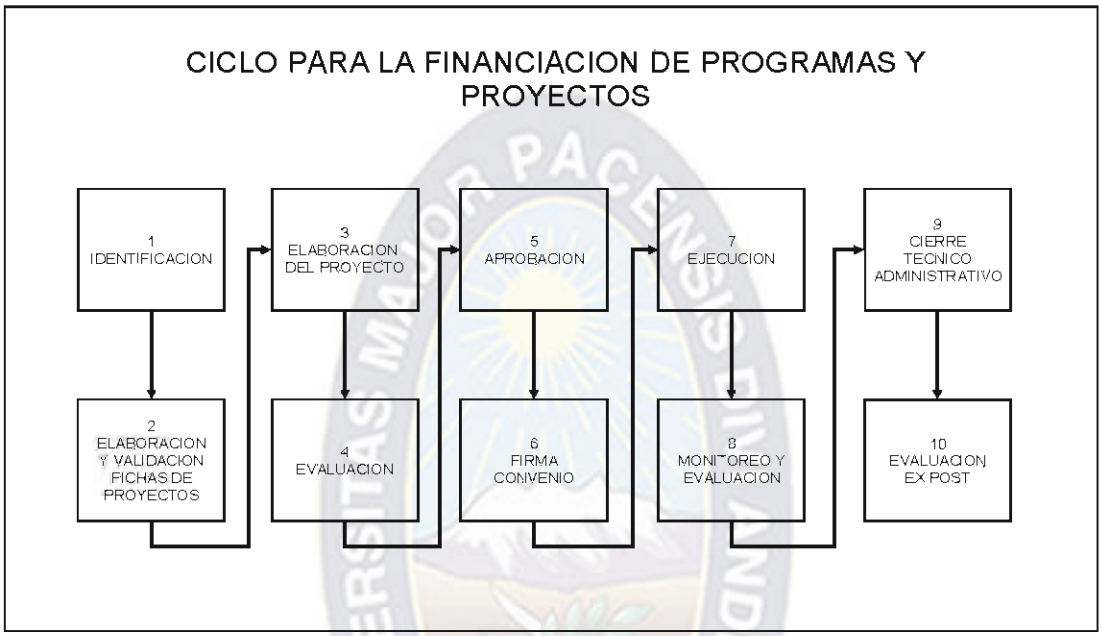
La Unidad tiene las siguientes características:

- Órganos de ejecución, seguimiento y evaluación;
- Se encuentra bajo dependencia directa del Ministro de Desarrollo Rural y Tierras que es dependencia funcional del Viceministro de Biodiversidad, Recursos Forestales y Medio Ambiente;
- Tienen independencia de gestión administrativa, financiera, legal y técnica, sobre la base de la normativa interna del Ministerio;

Esta dirigida por un Coordinador General con nivel equivalente a un Director General Ejecutivo del MDRyT, designado por Resolución Ministerial con atribuciones para atender, resolver y definir los asuntos de su competencia. Las funciones y servicios que presta Unidad Desconcentrada SUSTENTAR se muestran en el [Anexo A].

Se establece el siguiente procedimiento básico de los programas y proyectos que concurren a la Unidad Desconcentrada SUSTENTAR, que se muestra en la siguiente figura 1.

Figura. 1. Procedimiento de programas y proyectos



Fuente: [UD SUSTENTAR]

1- Identificación de la demanda

2- Elaboración y validación de fichas de proyectos

3- Elaboración del proyecto

4- Evaluación de los proyectos:

(i) financiera,

(ii) económica,

(iii) Social,

(iv) Institucional,

(v) Legal,

(vi) ambiental.

5- Presentación y aprobación del proyecto por el Comité de aprobación, seguimiento y revisión de programas y proyectos de SUSTENTAR

6- Firma de convenio de financiamiento

7- Ejecución

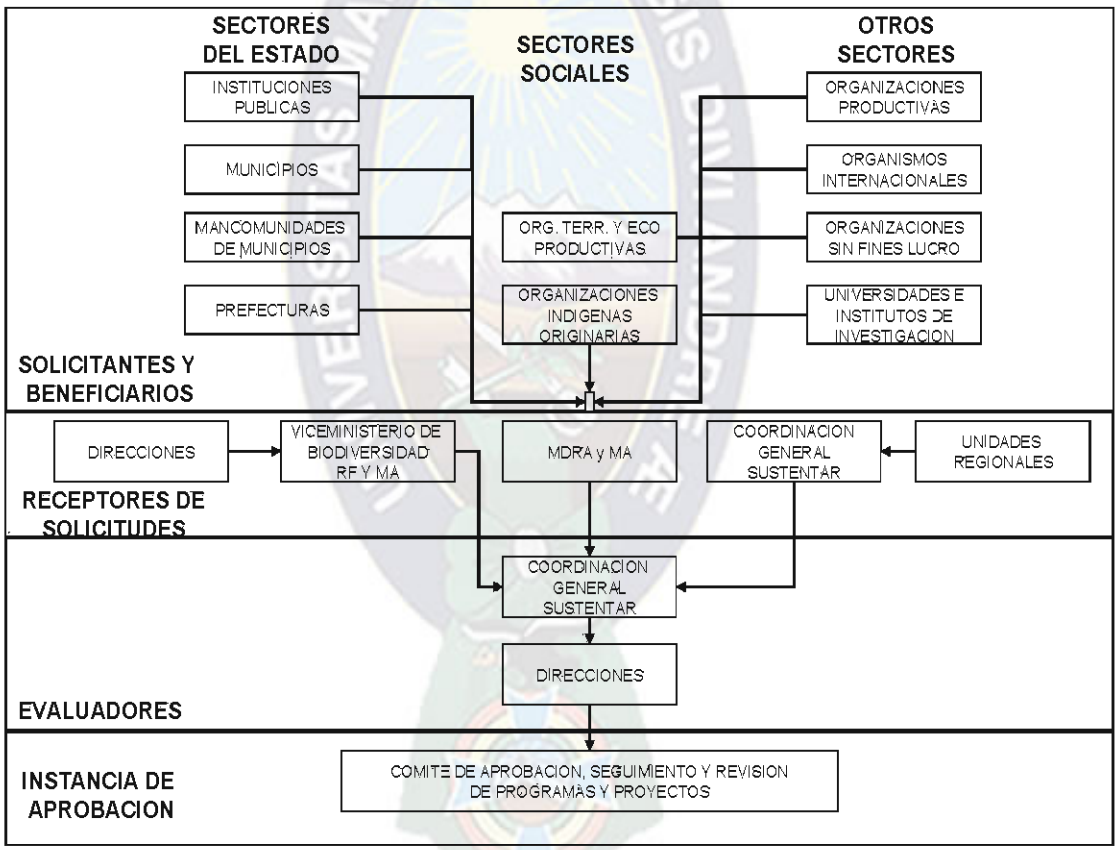
8- Monitoreo y evaluación

9- Cierre técnico y administrativo.

10.- Evaluación ex post (después de la entrega de la obra).

Para el proceso de trámites se sigue el siguiente procedimiento: Presentación, recepción, evaluación y aprobación de programas y proyectos que se muestra en la Figura 2.

Figura 2. Proceso de trámite



Fuente: [Unidad Desconcentrada SUSTENTAR]

Esta instancia esta conformada por el Coordinador General y por tres unidades:

- Unidad de Proyectos
- Unidad de Planificación y Monitoreo
- Unidad Administrativa y Financiera

Estos niveles de gestión tienen las siguientes atribuciones generales:

- El Coordinador General es la máxima autoridad de SUSTENTAR, responsable de dirigir, coordinar e implementar los proyectos de los componentes del programa SUSTENTAR y coordinar acciones con las unidades ejecutoras de programas y proyectos del MDRyT.

- La Unidad de Proyectos es la instancia técnica de SUSTENTAR encargada del apoyo a la formulación, evaluación y ejecución de los programas y proyectos de esta entidad.

- La Unidad de Planificación y Monitoreo, es la instancia técnica de SUSTENTAR encargada de la planificación, seguimiento, evaluación ex ante, durante y ex post de los proyectos implementados en la entidad.

- La Unidad Administrativa y Financiera, es la instancia responsable de dirigir y llevar adelante toda la administración de los recursos financieros de SUSTENTAR y de sus proyectos, en el marco de la normativa legal vigente.

También hay que mencionar algunos proyectos que se desarrollaron a medida para empresas o instituciones encargadas de administración de proyectos, los cuales contribuyen como referencia para el presente trabajo entre los cuales podemos mencionar los siguientes:

- Proyecto de Grado de la Carrera de Informática, con el Título: "SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE PROYECTOS DE LA EMPRESA TERACORP S.R.L. TECNOLOGÍA Y CONSULTORA", realizado por Mendoza Lliully Ruth Mery, en el año 2005. El siguiente trabajo es el que gestiona la calidad y garantiza la correcta elaboración de proyectos tomando en cuenta el control de actividades, recursos humanos y económicos que

intervienen en el desarrollo de un proyecto, brindando información precisa y actualizada para la toma de decisiones.

- Proyecto de Grado de la Carrera de Informática, con el Título: "SISTEMA INTEGRADO DE CONTROL Y SEGUIMIENTO DE PROYECTOS MUNICIPALES AMDEPAZ", realizado por Herrera Tola Justo Erland, en el año 2006. El presente sistema permite integrar, apoyar en el control y permitir el seguimiento de proyectos municipales relacionados a actividades de representación, asistencia técnica y fortalecimiento institucional.

- Proyecto de grado de la carrera de Informática, con el título:"SISTEMA DE INFORMCACIÓN PARA EL CONTROL Y SEGUIMIENTO DE PROYECTOS VIA WEB", realizado por Ibarra García Jhonny Wilson.

- Proyecto de grado de la carrera de Informática, con el título:"SISTEMA DE CONTROL Y SEGUIMIENTO DE LOS PROCESOS DE FISCALIZACIÓN INTEGRAL_GMLP", realizado por Ocaña Gallardo Geraldine en el año 2006. El presente sistema es el encargado registrar, control y seguimiento de los procesos de la unidad de fiscalización integral de la Sub Alcaldía San Antonio GMLP, de maneara que se logre la optimización de estos procesos, orientados a mejorar la eficiencia de la gestión municipal.

1.3. PLANTEAMIENTO DEL PROBLEMA

En la Unidad Desconcentrada SUSTENTAR, se encuentran las siguientes problemas que se mencionan a continuación:

- La falta de monitoreo de los diferentes proyectos en ejecución, debido a que la información no se puede actualizar, ya que no cuentan con un sistema para realizar dicho proceso.

- Se llenan los formularios de trámites manualmente, el cual conlleva mucho tiempo y demora el trabajo.

- La documentación del proyecto esta archivado en hojas impresas, imposibilitando el desarrollo y seguimiento del proyecto.

- No se procesa de manera oportuna la revisión de cada proceso para el seguimiento del proyecto, lo que representa deficiencias en su control.

- Falta de un portal Web para hacer conocer a la población la disponibilidad de proyectos en proceso de ejecución.

Lo anteriormente expuesto se observa en el Árbol de Problemas en [Anexo B].

1.3.1 DEFINICIÓN DEL PROBLEMA

Por tanto, se formula la siguiente interrogante:

¿Es posible sistematizar procesos de control y seguimiento de proyectos manuales, que se identifican en la Unidad Desconcentrada SUSTENTAR para tener una información adecuada, precisa para la toma de decisiones?

1.4. OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e implementar el Sistema de Información y Seguimiento de Proyectos en la Unidad Desconcentrada SUSTENTAR, que garantice su correcto monitoreo, tomando en cuenta las actividades, el desarrollo de proyecto, facilitando información precisa y confiable que coadyuve a la toma de decisiones.

1.4.2 OBJETIVOS ESPECÍFICOS

El Sistema tiene como objetivos específicos que se describe a continuación:

- Automatizar el sistema de seguimiento, monitoreo y control de diferentes proyectos (nacional, departamental, regional, municipal y local).
- Desarrollar e implementar un portal Web para difundir las actividades de la Unidad Desconcentrada SUSTENTAR.
- Verificar la factibilidad de Proyectos desde portal Web.
- Modificación de proyectos desde cualquier punto del país (de uso exclusivo del personal de Unidad Desconcentrada SUSTENTAR).
- Generar reportes de proyectos en ejecución, que sea útil para la toma de decisiones.

1.5. APORTES

Los aportes que brinda este proyecto con el nuevo Sistema de Información de Proyectos son los siguientes:

Ayudar de manera eficiente, en los diferentes niveles del proceso de la gestión de proyectos.

Se logrará la sistematización del proceso de gestión de proyectos de acuerdo a los requerimientos de la Unidad Desconcentrada SUSTENTAR usando mecanismos de gestión de proyectos.

Facilitar reportes que coadyuven a la correcta toma de decisiones.

1.5.1 **APORTE TEÓRICO**

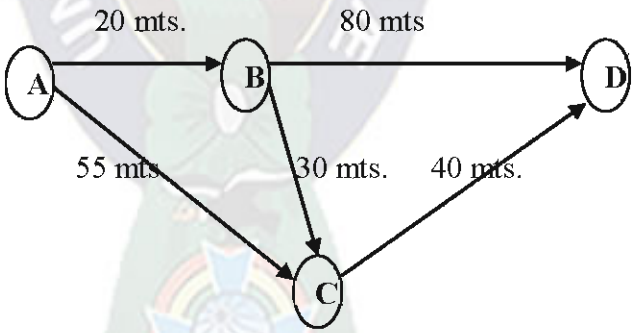
Para lograr los diferentes alcances que se proponen se utilizarán técnicas, metodologías mencionadas a continuación:

- El sistema que se presenta permitirá cubrir las necesidades de seguimiento de los diferentes programas y proyectos que son ejecutados o se encuentran en fase de planificación.

- Establecer una red interna que permita la integración de los datos para lograr su actualización en tiempo real.

- La clasificación de los proyectos se realizara por redes de PETRI, en donde cada nodo representa un proceso por el cual pasa un proyecto desde su concepción hasta su ejecución, lo que permitirá una búsqueda confiable y eficiente. Lo que se muestra en la Figura 3.

Figura 3. Estructura de una Red de Petri



Fuente: [Elaboración Propio]

1.6. JUSTIFICACIÓN

El desarrollo del Sistema de Información y Seguimiento de Proyectos para el seguimiento y monitoreo de proyectos de la Unidad Desconcentrada SUSTENTAR, justifica su existencia por las siguientes razones:

1.6.1. JUSTIFICACIÓN SOCIAL

El Sistema de Información y Seguimiento de Proyectos apoyará la gestión de proyectos, permitiendo a las comunidades muy distantes de las ciudades capitales tendrán la posibilidad a acceder a una página Web para el trámite de proyecto.

La implementación de este tipo de tecnología constituye un cambio organizacional importante en los diferentes ámbitos en el desarrollo de las comunidades del área rural.

El Sistema permitirá gestionar los proyectos en la parte administrativa de la Unidad Desconcentrada SUSTENTAR de forma eficiente y rápida.

1.6.2. JUSTIFICACIÓN TÉCNICA

El desarrollo del presente proyecto se justifica técnicamente, por el uso de alta tecnología de comunicación: Existen los medios necesarios para realizar el análisis, diseño, implementación, y mantenimiento del Portal Web para su funcionamiento en el Internet. La Unidad Desconcentrada SUSTENTAR, cuenta con los requerimientos tecnológicos de hardware y software están agrupados en: Equipos computacionales y software de base.

1.6.3. JUSTIFICACIÓN ECONÓMICA

La inversión en el portal tendrá una reducción de costos ya que se utilizará para el desarrollo Software gratuito en su gran mayoría.

El funcionamiento del Sistema permitirá una mejora de la gestión de proyectos de la Unidad Desconcentrada SUSTENTAR aumentando su eficiencia, y reduciendo costos.

1.7. ALCANCES Y LÍMITES

El presente proyecto está dirigido al seguimiento y monitoreo de proyectos, con acceso a una base de datos del sistema vía Web, con el objetivo de mejorar la correcta administración y gestión de los proyectos que se va ejecutar desde la Unidad Desconcentrada SUSTENTAR.

Sus alcances proporcionarán una herramienta que brinde información veraz, confiable y oportuna como soporte para la toma de decisiones de la institución.

1.7.1. LÍMITES

El Sistema tiene como límite a las siguientes:

- Se podrá acceder al sistema vía Web desde un Internet, y si no cuenta con una red de Internet no será viable el proceso.
- Su operatividad será a nivel nacional.
- La modificación de proyectos será de uso exclusivo del personal Unidad Desconcentrada SUSTENTAR.

1.8. METODOLOGÍA

Las metodologías de desarrollo ayudan a comprender la realidad del sistema a desarrollar, entender el contexto, entender sus partes y sus interrelaciones; en síntesis apoya a adquirir un conocimiento pertinente respecto a una realidad. Sabemos perfectamente que todo conocimiento conlleva a un riesgo de error e ilusión, en la más mínima transmisión de información de un emisor a un receptor existe siempre el riesgo de errar, concientes de esta realidad se debe procurar que

una metodología que minimice el nivel de error, favoreciendo la comunicación del equipo de desarrollo.

En la investigación más reciente la propuesta para utilizar el modelo de redes de Petri para analizar el DFD y validar la exactitud de la especificación. Dentro de las principales razones para hacerlo de esta manera están:

- Las redes de Petri se pueden expresar tanto gráfica como algebraicamente.

- Las representaciones gráficas de transiciones y plazas de una red de Petri corresponden a las componentes proceso y flujo de datos de los DFD's, respectivamente.

- Las redes de Petri pueden representarse en diagramas jerárquicos para modelar la naturaleza multinivel de los DFD's.

Como se ve, las interpretaciones que se le pueden dar a una red de Petri no tienen necesariamente que ser relacionadas con computación.

MARCO TEÓRICO

CAPÍTULO II

2.1 INTRODUCCIÓN

En el presente capítulo se darán a conocer los aspectos de las metodologías, herramientas y técnicas que serán utilizadas en cada etapa del desarrollo del proyecto que se dividirán en diferentes fases como son el análisis, diseño, implementación y pruebas del sistema a implementar.

A continuación se hace referencia de los términos de proyecto y seguimiento. "**El proyecto** no es mas que un modelo del emprendimiento a ser realizado, con las previsiones de recursos, de tiempo de ejecución y de resultados esperados". [IBARROLA, 1972]

Las Naciones Unidas definen proyecto como el "conjunto de antecedentes que permita estimar las ventajas y desventajas económicas que se derivan de asignar ciertos recursos de un país para la producción de determinados bienes o servicios".

Según [Senn 1992] Sistema de información se define como: "las finalidades de los sistemas de información, como las de cualquier otro sistema dentro de una organización, son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas". A partir de lo que viene a ser el avance tecnológico se ha podido cambiar los procesos manuales y mecánicos por procesos automáticos, en nuestro caso, aquellos que son realizados por el computador.

Entrando en más detalle se define lo que es un **Sistema de Procesamiento de Transacciones TPS**, que es el tipo de sistemas a desarrollar, y diremos que son sistemas de información computarizados desarrollados para procesar una gran

cantidad de datos para transacciones rutinarias de los negocios tales como nómina de inventarios. Los TPS eliminan el tedioso de las transacciones operacionales necesarias y reducen el tiempo que alguna vez se requirió para ejecutar manualmente, aunque la gente debe alimentar datos a los sistemas.

Sistema de Información es el conjunto de elementos interrelacionados y orientados a establecer un flujo continúo y eficiente de comunicación que sirva para satisfacer los requerimientos de distintos usuarios tanto internos como externos. [IBAÑEZ, 2006].

Todo sistema de información está formado por subsistemas que incluyen hardware, software, medios de almacenamiento de datos para archivos y base de datos.

2.2. CONCEPTOS GENERALES

2.2.1 SISTEMA DE INFORMACIÓN

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

El equipo computacional: El hardware necesario para que el sistema de información pueda operar.

El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: Entrada, almacenamiento, procesamiento y salida de información.

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son

datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfases automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáners, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

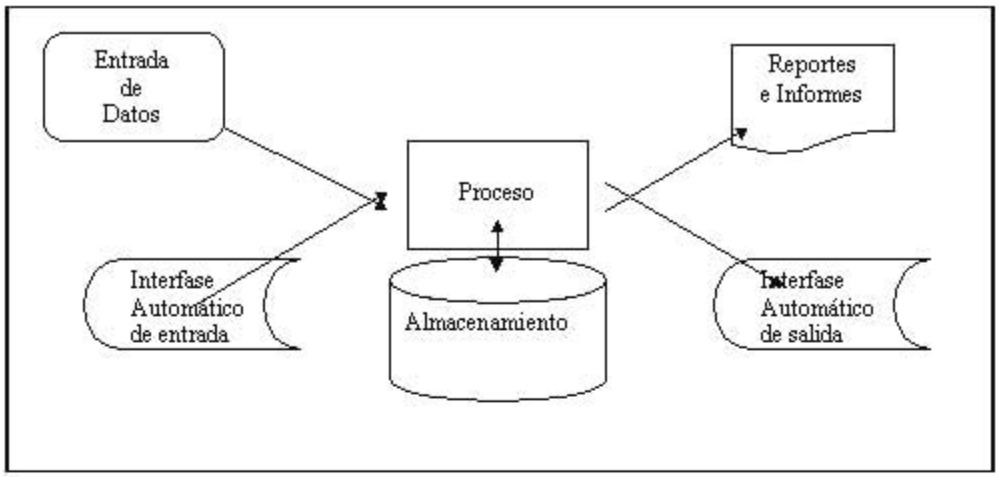
Salida de Información: La salida es la capacidad de un Sistema de personal para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas entre otros.

Tipos y Usos de los Sistemas de Personal

Durante los próximos años, el Sistema de cumplirá tres objetivos básicos dentro de la organización:

1. Automatización de procesos operativos.
2. Proporcionar información que sirva de apoyo al proceso de toma de decisiones, (como se muestra en la figura 2.1).

Figura 2.1. Automatización de procesos



Fuente:[diseño propio]

3. Lograr ventajas competitivas a través de su implantación y uso.

Los Sistemas de Personal logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc.. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

A continuación se mencionan las principales características de estos tipos de Sistemas de Información.

Sistemas Transaccionales. Sus principales características son:

- A través de éstos suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.

- Con frecuencia son el primer tipo de Sistemas de Información que se implanta en las organizaciones. Se empieza apoyando las tareas a nivel operativo de la organización.

- Son intensivos en entrada y salid de información; sus cálculos y procesos suelen ser simples y poco sofisticados.

- Tienen la propiedad de ser recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.

- Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables.

Sistemas de Apoyo de las Decisiones. Las principales características de estos son:

- Suelen introducirse después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, genera poca información como resultado, pero puede realizar muchos cálculos durante su proceso.

- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser Sistemas de Información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.

- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivos y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un Sistema de Compra de Materiales que indique cuándo debe hacerse un pedido al proveedor o un Sistema de Simulación de Negocios que apoye la decisión de introducir un nuevo producto al mercado.
- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.

Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios, modelos de inventarios, etc.

Sistemas Estratégicos. Sus principales características son:

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse in house, es decir, dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Típicamente su forma de desarrollo es a base de incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.

- Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los Sistema Estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el uso de cajeros automáticos en los bancos en un Sistema Estratégico, ya que brinda ventaja sobre un banco que no posee tal servicio. Si un banco nuevo decide abrir sus puertas al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.

- Apoyan el proceso de innovación de productos y proceso dentro de la empresa debido a que buscan ventajas respecto a los competidores y una forma de hacerlo en innovando o creando productos y procesos.

2.2.2. INTERNET Y LA WEB

Internet es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que la Internet, independientemente de su extensión o de que sea pública o privada.

Al contrario de lo que se piensa comúnmente, **Internet** no es sinónimo de World Wide Web. Ésta es parte de aquella, siendo la World Wide Web uno de los muchos servicios ofertados en la red Internet. La Web es un sistema de información mucho más reciente que emplea la red Internet como medio de transmisión...

Algunos de los servicios disponibles en Internet aparte de la Web son el acceso remoto a otras máquinas (telnet y ssh), transferencia de archivos (FTP), correo electrónico (e-mail), boletines electrónicos (news o grupos de noticias), conversaciones en línea (chat).

- Un **servicio Web** (en inglés *Web service*) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones;
- Accesible desde cualquier aplicación;
- Por cualquier lenguaje de programación;
- Desde cualquier plataforma;
- Usando estándares abiertos;

Ventajas de los servicios Web

- Aportan **interoperabilidad** entre aplicaciones de software;

- Los servicios Web **fomentan los estándares** y protocolos **basados en texto** (más humanos y accesibles);
- Al apoyarse en **HTTP**, permiten acceder a cualquier sistema conectado a la red (http usa el puerto 80);
- Permiten el uso de **servicios integrados** cambiando el de varias compañías y varios softwares;

Permiten la **interoperabilidad** entre plataformas de distintos fabricantes por medio de protocolos estándar.

2.3. MARCO CONCEPTUAL

2.3.1. EL PARADIGMA: ORIENTADO A OBJETOS

El paradigma Orientado a Objetos es un conjunto de teorías, estándares y métodos que juntos representan un medio de organización del conocimiento. La orientación a objetos muestra un nuevo pensamiento sobre como modelar los objetos para un sistema de computación y sobre cómo se estructura la información dentro de la computadora.

Las ventajas que representa el paradigma son:

- Mejora la calidad del software generado;
- Acorta el tiempo de desarrollo;
- Aumenta la productividad;
- Se puede reutilizar software generado;

En programación orientada a objetos se utilizan conceptos propios como: objeto, encapsulamiento, ocultación, métodos, clases, herencias, polimorfismo, sub-clases y relaciones.

A continuación se explican los más relevantes:

- **Objeto,** es una entidad real o abstracta que pertenece a una clase (o clases) que define sus estados y comportamientos para con otros objetos. Un objeto consta de estado y comportamiento, es decir, de atributos y operaciones;

- **Herencia,** es una de las características que más se destaca de la programación orientada a objetos, la herencia es la capacidad de adquirir y conservar características "comunes" de una generación a otra. Otras clases pueden beneficiarse de las características de las "de las antecesoras" de manera de ahorrar tiempo en la hora de desarrollar e implementar un proyecto;

- **Encapsulamiento,** es un principio de abstracción de diseño encargado de ocultar la información, datos o funciones especiales a los usuarios. El principio de ocultación de información dice que los aspectos privados de un objeto (atributos) no pueden ser modificados ni capturados a no ser que lo sean mediante uno de los servicios (también llamados métodos u operaciones);

- **Abstracción,** proporciona la programación Orientada a Objetos se basa en la posibilidad de encapsulación consiste en agrupar bajo la misma entidad los datos y las funciones o métodos que trabajan con esos datos. Esta propiedad permite distinguir a un objeto de los demás, observando sus características y comportamientos, pensando en qué es y no en cómo se codificaría en un lenguaje. Con la abstracción se destaca lo importante y se ignora lo irrelevante, es decir, hay ocultamiento de información;

- **Polimorfismo,** los objetos encapsulan, es decir, oculta la funcionalidad interna de sus operaciones de otros objetos.

2.4. METODOLOGÍA DE ANÁLISIS Y DISEÑO

La metodología a ser utilizada en el desarrollo del trabajo está enfocado al proceso de desarrollo de software RUP (Proceso Unificado del Desarrollo de Software) que utiliza técnicas del desarrollo a través del UML (Lenguaje Unificado de Modelado) [www.creangel.com/uml.pdf]

2.4.1. PROCESO UNIFICADO DEL DESARROLLO DE SOFTWARE

Como muestra en figura 2.2 un proceso de desarrollo de software en el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. El RUP es un proceso de desarrollo que puede enfocarse a distintos sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organización y diferentes tamaños de proyectos.

Figura 2.2. Proceso de Desarrollo de Software

Fuente: [Acarapi, 2004]

La metodología RUP consta de tres partes fundamentales que son:

- Dirigido por casos de uso.
- Centrado en la Arquitectura.
- Iterativo e incremental.

Dirigido por casos de uso

Para el mejor desarrollo se debe capturar las necesidades de los usuarios a fin de diseñar una implementación funcional que se ajuste a las necesidades y para ello es necesario que la descripción de los requisitos sea comprensible y en esto nos ayudan los casos de uso a través de los cuales recopilamos información acerca de la interacción que tiene los usuarios en este caso actores con el sistema. Un caso de uso es una secuencia de acciones que el sistema lleva a cabo para ofrecer un resultado de valor a algún actor.

En la etapa de Análisis y Diseño que son otros modelos en los que se transforma el modelo de Caso de Uso, son estructuras compuestas por documentos definidos en el UML [www.creangel.com/uml.pdf] como son el modelo conceptual, diagramas de iteración, diagramas de clases, etc.

La implementación hace referencia a los componentes físicos del sistema como elementos de la Base de Datos, código fuente, etc. Un componente es un aporte físico y reemplazable del sistema que cumple y proporciona la realización de un conjunto de interfaces, estos componentes pueden ser reemplazados por otros siempre y cuando el nuevo proporcione requiera las mismas interfaces.

Un proceso centrado en la Arquitectura.

Este término hace referencia a un conjunto de decisiones significativas acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de las cuales se componen el sistema con su respectivo comportamiento y las interacciones entre esos elementos y la composición de esos elementos estructurales.

La necesidad de una arquitectura radica en poder comprender el sistema, es decir que todos los que están involucrados con su desarrollo puedan entender el problema al cual va enfocado el sistema de software para satisfacer las demandas

individuales y de la organización mediante la utilización de los diagramas definidos por el UML [www.creangel.com/uml.pdf]

Un proceso iterativo e incremental.

Esta fase proporciona la estrategia para desarrollar un producto de software en pasos pequeños manejables:

- Planificar un poco.
- Especificar, diseñar e implementar un poco.
- Integrar, probar y ejecutar un poco cada iteración.
- Una vez concluido un poco, continuamos con el siguiente, en cada paso obtenemos retroalimentación que nos permite ajustar nuestros objetivos para el siguiente paso y así sucesivamente.

Con el método de desarrollo iterativo e incremental se divide el proyecto en un número de mini proyecto, siendo cada uno de ellos una iteración, cada iteración tiene todo lo que un proyecto de desarrollo de software: planificación, desarrollo en una serie de pasos (requisitos, análisis y diseño, implementación y pruebas) y la entrega final. Estas iteraciones difieren cada fase del desarrollo de software (inicio, elaboración, construcción, transición) los desafíos que afrontan los desarrolladores son diferentes en cada fase.

2.4.2. CICLO DE VIDA

El ciclo de desarrollo se divide en cuatro fases: inicio, elaboración, construcción y transición, con la secuencia de modelos se visualiza lo que sucede en cada fase.

a) Inicio.

Durante esta primera fase se desarrolla una descripción del sistema final a partir de una buena representación del análisis del sistema tomando en cuenta:

- Funciones principales del sistema para los usuarios más importantes
- Arquitectura del sistema

b) Elaboración.

En esta fase se especifica los detalles de los casos de uso y se diseña la arquitectura del sistema expresado en forma de vistas de todos los modelos del sistema, los cuales junto expresan al sistema entero.

c) Construcción.

La construcción es parte del producto para ser entregado a los usuarios viendo que el sistema cubra las necesidades del usuario.

d) Transición.

Cubre el periodo durante el cual el producto se convierte en versión beta.

En esta versión los usuarios prueban el producto e informan de defectos y deficiencias. Se corrigen los problemas e incorporan algunas de las mejoras sugerencias en una versión general dirigida a los usuarios. Esto lleva a las actividades como capacitación del usuario, ayuda, asistencia y corrección de los errores que se encuentra después de la entrega del software.

2.4.3. DISEÑO NAVEGACIONAL

El diseño navegacional especifica que objetos pueden ser visitados navegando a través del sistema y el modelo de estructura de navegación que define como se alcanzan estos objetos.

Se usan dos tipos de elementos para construir el modelo navegacional: clases de navegación y asociaciones de navegación:

- **Clases de navegación:** es una clase cuya instancia puede visitar el lector durante la navegación, las clases de navegación deben tener el mismo nombre que la correspondiente clase del modelo conceptual.

- **Asociaciones de navegación:** las asociaciones representan navegabilidad directa entre la clase de navegación inicial y la clase de navegación final, por lo que posee una semántica diferente que las asociaciones del modelo conceptual. En este modelo las asociaciones se representan por una flecha. Se etiqueta cada extremo de una asociación con un nombre de rol.

A continuación se presenta los pasos a seguir en la construcción del modelo de espacios de navegación:

i) Incluir clases del modelo conceptual que son relevantes para la navegación, como clases de navegación en el modelo de espacios de navegación.

ii) Guardar información de las clases omitidas como atributos de las clases en el espacio del modelo de navegación.

iii) Las asociaciones del modelo conceptual guardadas en el modelo de navegación.

iv) Las asociaciones adicionales basadas en la descripción de requisitos o en los descritos en el modelo conceptual.

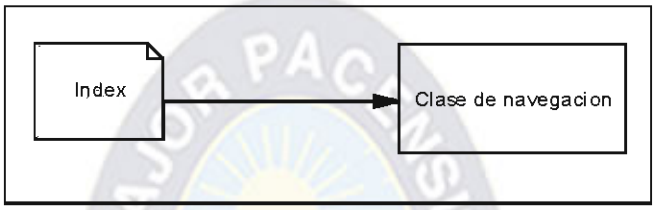
v) Añadir restricciones para especificar el espacio de navegación.

La descripción de la navegabilidad se las hace a través de las siguientes primitivas de navegabilidad:

a) Índice.

Un índice permite acceso directo a las instancias de una clase de navegación. Se modelan por un objeto compuesto que tiene un nombre que identifica cada instancia de una clase de navegación. La figura 2.3 muestra la siguiente notación.

Figura 2.3. Clase índice



Fuente: [Acarapi 2004]

b) Vuelta Guiada.

Estos elementos proporcionan acceso secuencial a instancias de una clase de navegación [Figura 2.4]. Las vueltas guiadas pueden ser controladas por el sistema o por el usuario.

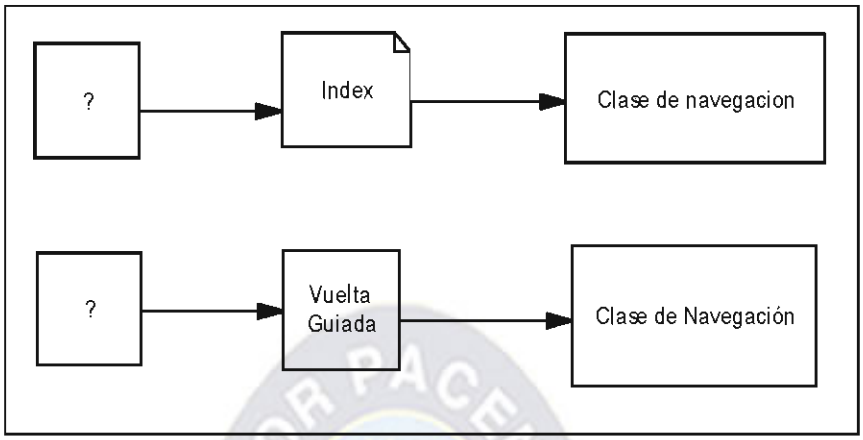
Figura 2.4. Clase Vuelta Guiada

Fuente: [Acarapi 2004]

c) Consulta.

Se modelan mediante una clase con una cadena de consultas como atributos. Esta cadena puede ser dada por una operación de selección como muestra la figura 2.5

Figura 2.5. Clase Consulta



Fuente: [Acarapi 2004]

2.5. EL LENGUAJE DE MODELADO UML

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 de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Objetivos de UML

- Definir un lenguaje de modelado visual;
- Fácil de aprender, rico en significado;
- Estándar, estable y confiable;
- Independiente de lenguaje de programación o procesos particulares.

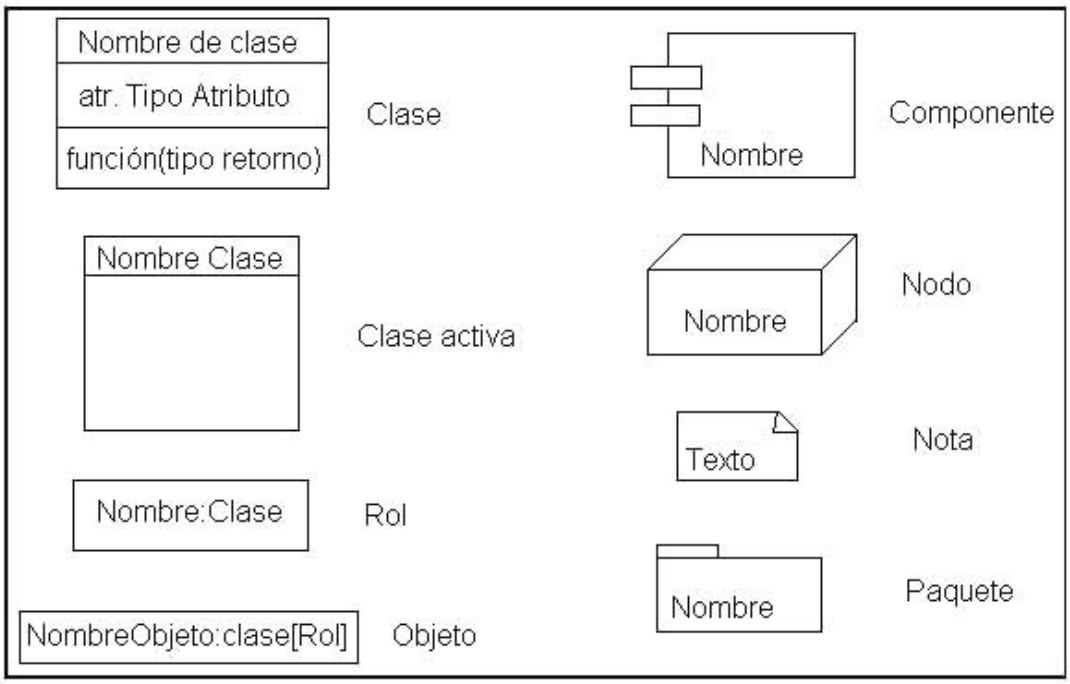
El **UML** es parte muy importante para el desarrollo de Software Orientado a Objetos y en el Proceso de Desarrollo de Software. Utiliza, en su mayor parte,

notaciones gráficas para expresar los procesos en el diseño de proyectos de Software.

Reglas de UML

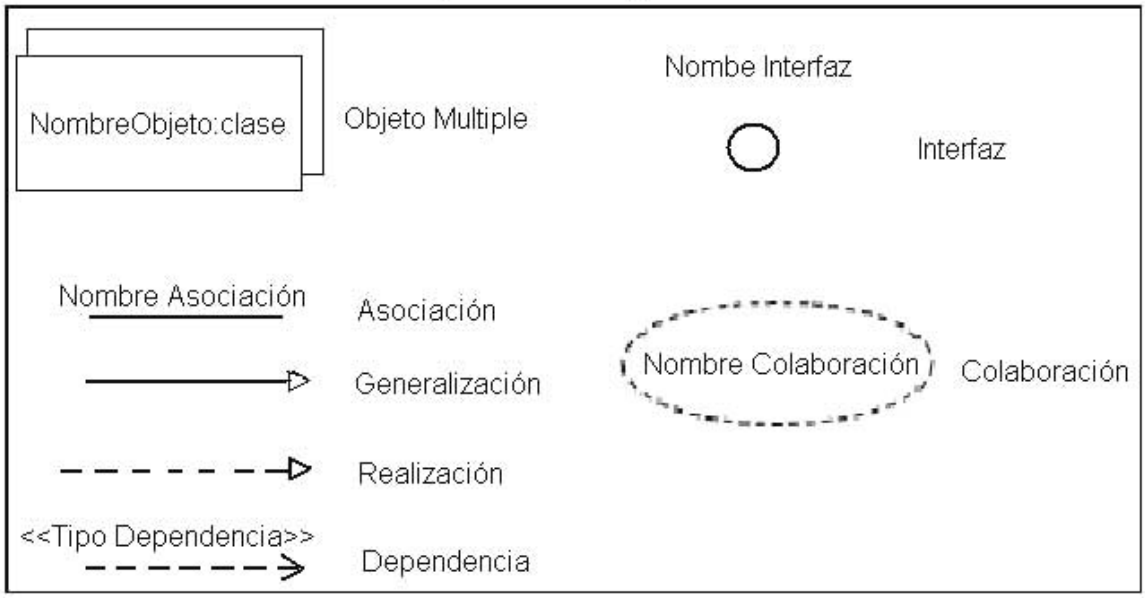
Las reglas en UML especifican a que debe parecerse un modelo bien formado, es decir un modelo que es semánticamente auto consistente y están con los modelos relacionados [Ramírez, 2005]. A continuación se presenta su simbología, en la figura 2.6 y 2.7

Figura 2.6. Simbología UML



Fuente: [Schmuller, 1997]

Figura 2.7. Simbología UML



Fuente: [Schmuller, 1997]

Diagramas de UML

Un diagrama es la representación gráfica de un conjunto de elementos y relaciones. Los diagramas se utilizan para visualizar el comportamiento de un sistema desde diferentes perspectivas. Estos diagramas de UML son los siguientes:

Diagramas de estructura

Muestran los elementos que deben existir en el sistema modelado:

- Diagramas de clases para modelar la estructura y visualizar las relaciones entre las clases que involucran en sistema.
- Diagrama de componentes para modelar el aspecto físico del sistema.
- Diagrama de objetos para modelar la estructura estática de los objetos en el sistema.
- Diagrama de despliegue para formar un sistema ejecutable.

- Diagrama de paquetes para organizar y analizar el avance del sistema.

Diagrama de Comportamiento

Muestra lo que debe suceder en el sistema:

- Diagrama de actividades para visualizar, especificar y modelar el comportamiento de los casos de uso, objetos y operaciones.
- Diagrama de casos de uso para modelar los procesos.
- Diagrama de estados para modelar el funcionamiento de los objetos.

Diagrama de Interacción

Muestra el flujo de control y los datos entre los elementos del sistema:

- Diagrama de secuencia para modelar la interacción de un conjunto de objetos en una aplicación a través del tiempo.
- Diagrama de comunicaciones para modelar el paso de mensajes entre objetos.

2.5.1. MODELO CLIENTE – SERVIDOR

El esquema cliente-servidor es un modelo de computación donde las aplicaciones relacionadas se divide entre dos o mas procesos que cooperan entre si. Los principales componentes del modelo son los clientes (interactúan con el usuario), los Servidores (proporciona un servicio al cliente y devuelven el resultado), y la infraestructura de comunicación (la cual proporciona los mecanismos básicos de direccionamiento y transporte).

Modelo de 3 capas

El modelo de 3 capas es un método que se utiliza en la ingeniería de software, para dividir una aplicación en diferentes capas, en modelo de tres capas se divide en: capa cliente, capa intermedia o aplicación y capa del servidor o datos del negocio, ver Figura 2.8.

El desarrollo del proyecto se la realiza a través del modelo de tres capas el cual presenta las siguientes ventajas:

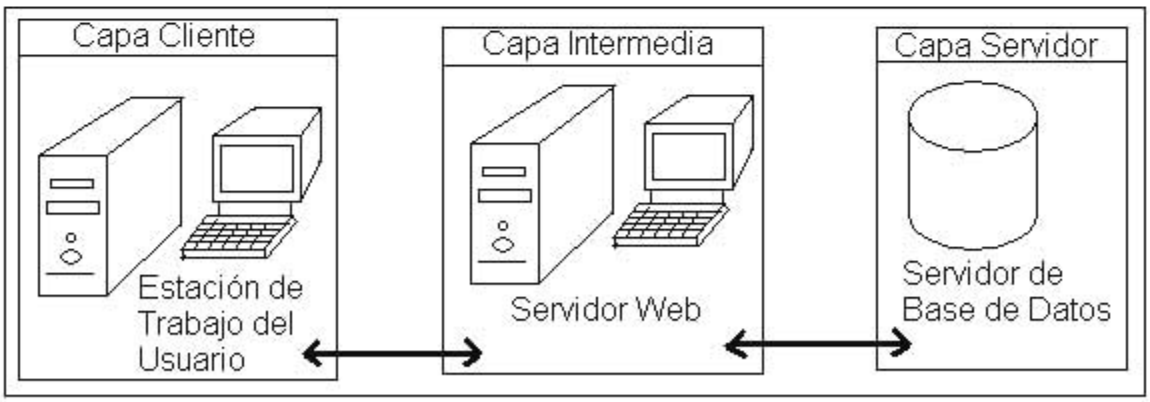
a) **Separación de funciones,** todo lo relacionado con la interfaz del usuario va en una capa, las reglas de negocio en otra y el manejo de datos en una tercera capa;

b) **Reutilización,** el código correspondiente a una capa puede ser reutilizado;

c) **Escalabilidad,** sabiendo donde está el código correspondiente a cada capa, pueden realizarse modificaciones dentro de una capa para mejorar o aumentar el tamaño del sistema de software, con un mínimo impacto en las capas restantes;

d) **Facilidad de mantenimiento,** mediante esta división, es mucho más sencillo localizar errores en el código o efectuar mejoras.

Figura 2.8. Modelo de Tres Capas



Fuente: [Elaboración propia]

2.5.2. BASE DE DATOS RELACIONAL

En una computadora existen diferentes formas de almacenar información. Esto da lugar a distintos modelos de organización de la base de datos: jerárquico, red, relacional y orientado a objetos.

Los sistemas relacionales son importantes porque ofrecen muchos tipos de procesos de datos, como ser: Simplicidad y generalidad, facilidad de uso para el usuario final, periodos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas [Stallings, 2000].

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no están determinados.
- Para cada campo existe un conjunto de valores posible.

Objeto – Relacional

El mapeo objeto – relacional es una técnica de programación para hacer corresponder datos entre el modelo orientado a objetos al modelo de datos relacional, los conceptos de clase e instancia siguen las siguientes definiciones [Elizondo, 2005]:

- En Entidad Relación (E/R) un tipo de entidad representa dos cosas:

- La intención, el nombre del tipo de entidad y la lista de sus atributos;
- La extensión o conjunto de entidades que forman parte del tipo de entidad;

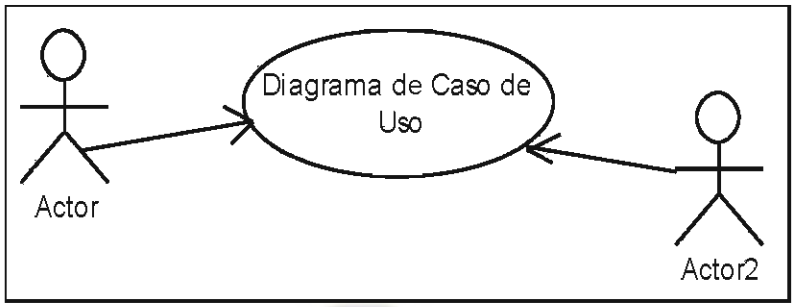
- Considerando lo anterior, una clase es un concepto equivalente al tipo de entidad;
- Las entidades de una clase se llaman objetos o instancias de esa clase;
- Cada objeto o instancia consta de un identificador predefinido que lo identifica. Dicho identificador del objeto no es ninguno de sus atributos;
- Los objetos o instancias de una clase cuentan con un conjunto de operaciones que se pueden realizar sobre ellos;
- Dichas operaciones pueden servir para especificar requisitos funcionales durante el diseño de la base de datos.

2.6. ANÁLISIS

Se profundiza la información obtenida en la Recopilación de Necesidades, el análisis del sistema se realizará con las siguientes etapas:

- **Comprensión del uso del sistema,** en la presente etapa se descubre los actores que iniciaran cada caso de uso del sistema, comprendiendo el uso que el usuario realiza en el sistema, los actores son los diferentes usuarios y el papel que representan dentro del sistema. Ver figura 2.9.

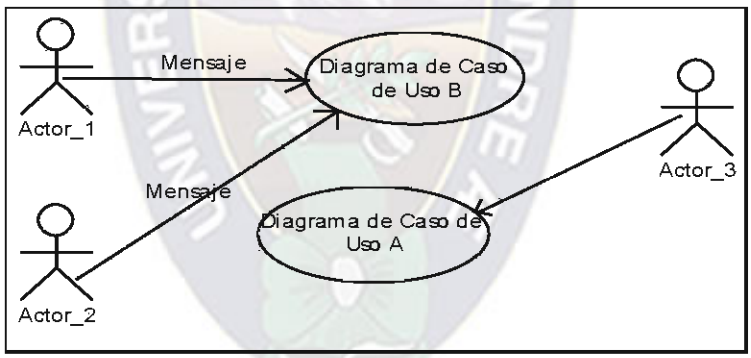
Figura 2.9. Representación de Actores



Fuente: [Elaboración Propia]

- **Diagrama de Casos de Uso,** un caso de uso representa todo lo que el usuario puede realizar dentro del sistema, en la actual etapa se hace realidad los casos de uso, analizando las secuencias de pasos para cada caso de uso, la notación que se utiliza para representar los casos de uso se encuentra en la Figura 2.10.

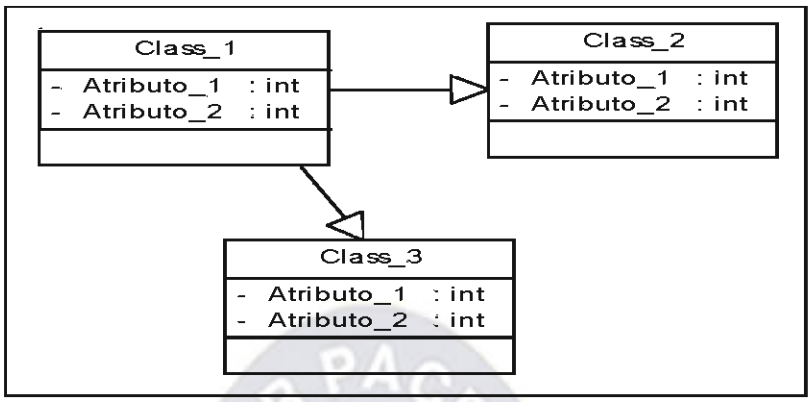
Figura 2.10. Diagrama de Casos de Uso



Fuente: [Elaboración propia]

- **Diagrama de clases,** es una colección de elementos (estáticos) declarativos de un modelo [Larman, 1999], en esta etapa se realiza el análisis, modelado y depuración de los diagramas de Clases, se debe de llenar los nombres de las asociaciones, clases abstractas, multiplicidades, generalizaciones y agregaciones, ver Figura. 2.11

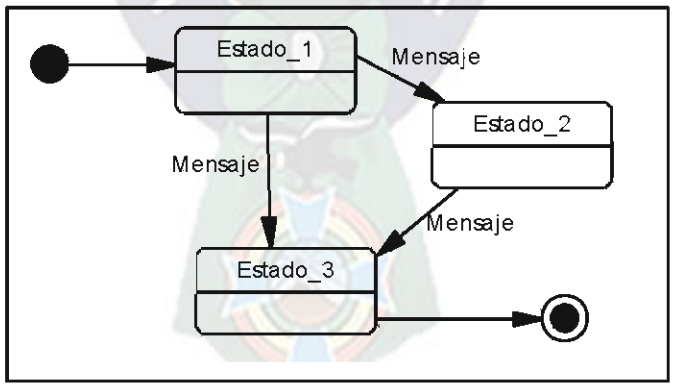
Figura 2.11. Diagrama de Clases



Fuente: [Elaboración propia]

- **Analizar cambios de estado en los objetos,** muchas veces el proyecto requiere la presentación de cambios de estado, por ello es importante realizar los diagramas de estado los cuales nos permiten describir el comportamiento de un objeto, mostrando la secuencia de estados por los que pasa a lo largo de su vida. En esta etapa se describen todos los estados posibles en los que pueden entrar un objeto en particular, ver Figura. 2.12

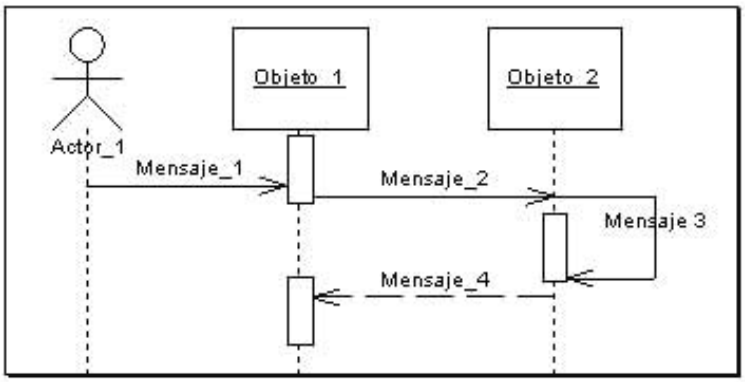
Figura 2.12: Diagrama de Estados



Fuente: [Elaboración propia]

- **Definir la comunicación entre objetos,** se encarga de modelar los objetos y permite ilustrar las acciones de los actores y las operaciones iniciadas por ellos. Un diagrama de secuencia representa la interacción entre las clases, se modela para cada caso de uso, ver Figura 2.13

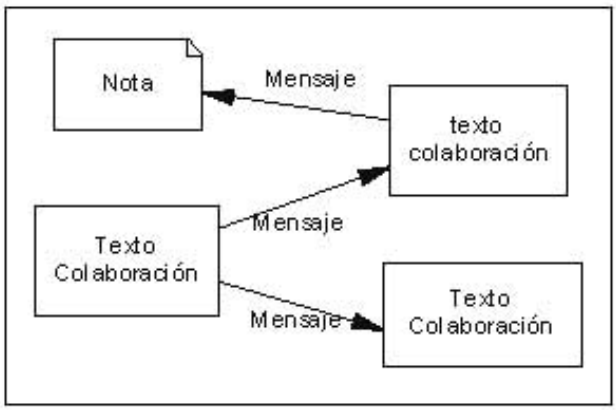
Figura 2.13. Diagrama de Secuencia



Fuente: [Elaboración propia]

- **Analizar la integración con los diagramas de colaboración,** se debe descubrir todos los detalles específicos del sistema, de ser necesario realizar los diagramas de distribución detallada. Los diagramas de colaboración permiten modelar interacciones entre objetos en el sistema y se centra a estudiar todos los efectos de un objeto durante un escenario, ver Figura 2.14.

Figura 2.14. Diagrama de Colaboración



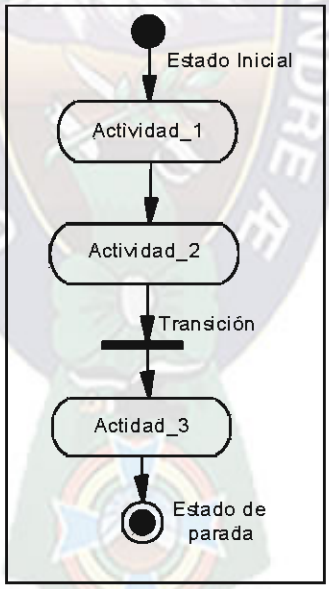
Fuente: [Elaboración propia]

2.7. DISEÑO

En este segmento se trabaja con los resultados del segmento de análisis para diseñar la solución, las tareas que se realizaran en el diseño son las siguientes:

- **Desarrollo y depuración de los diagramas de objetos,** se debe dar vida a los objetos mediante el análisis de cada operación y el desarrollo de un diagrama de actividades. El diagrama de actividades es un diagrama de flujo del proceso multi-propósito que se usa para modelar el comportamiento del sistema. Los diagramas de actividades se pueden utilizar para modelar una clase, un caso de uso o un método complicado, ver Figura 2.15

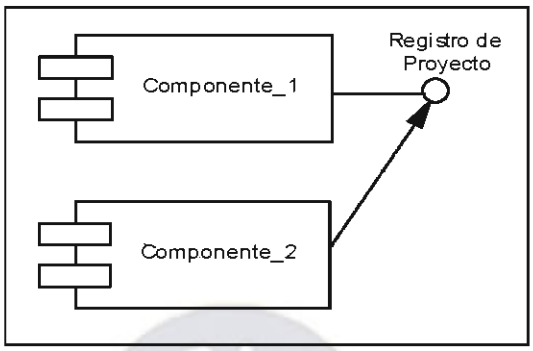
Figura 2.15. Diagrama de Actividades



Fuente: [Elaboración propia]

- **Desarrollo de diagramas de componentes,** son los diagramas de componentes, los cuales describen los elementos físicos del sistema y sus relaciones. Muestran las operaciones de realización incluyendo código fuente, binario y ejecutable, véase Figura 2.16.

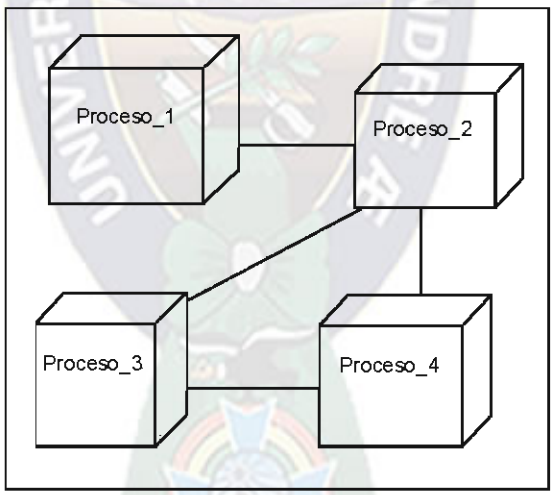
Figura 2.16. Diagrama de componentes



Fuente: [Elaboración propia]

- **Planeación de la distribución,** se desarrollara los diagramas de distribución los cuales muestran el despliegue de nodos (locales y remotos), en la organización del sistema, mostrando el lugar donde se encontraran los componentes, ver Figura. 2.17.

Figura 2.17. Diagrama de Distribución



Fuente: [Elaboración propia]

- **Diseño y prototipos de la interfaz del usuario,** se diseña las interfaces entre usuario y maquina con las que contara el proyecto, véase Figura. 2.18.

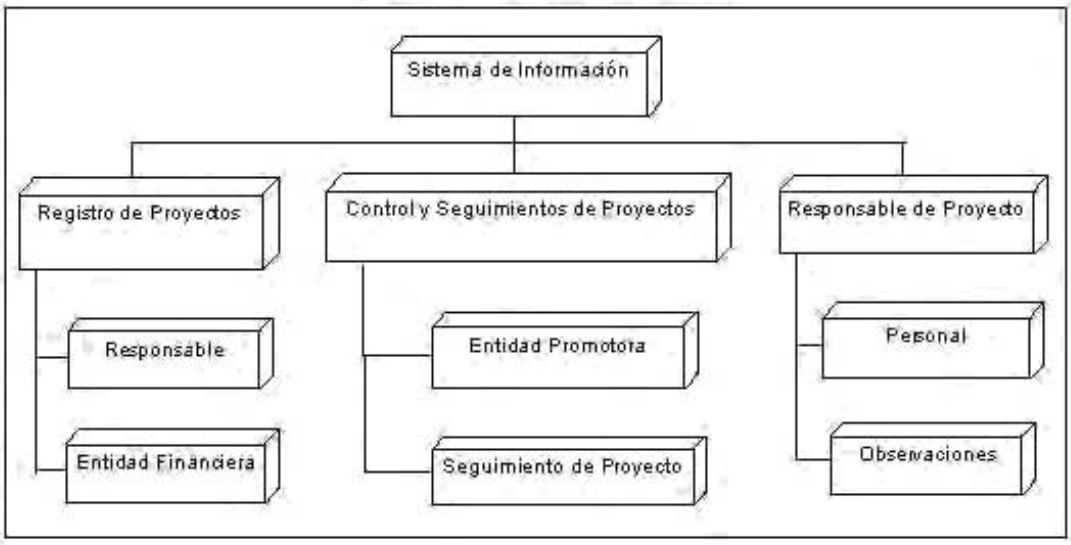
Figura 2.18. Prototipo de Interfaz

Fuente: [Elaboración propia]

2.7.1. MODELO Y DISEÑO DE LA BASE DE DATOS

Los sistemas pueden subdividirse en pequeños componentes o subsistemas, los cuales colaboran y ayudan a comprender mejor el sistema general. La Figura 2.19 muestra la división del sistema a desarrollar.

Figura 2.19. Diagrama del sistema



Fuente: [Elaboración propia]

Para el diseño de base de datos se utiliza la técnica de conversión al modelo entidad relación, tomando la información de los diagrama de clases, ver la correspondencia objeto relacional [Anexo C].

2.8. REQUERIMIENTO DE SOFTWARE Y HARDWARE

Para el desarrollo del presente proyecto se utilizan, un conjunto de herramientas de Software y Hardware, de manera que estas herramientas coadyuven en el desarrollo del sistema en sus diferentes etapas. Se utiliza las herramientas case, como PowerDesigner para el diseño del sistema, el entorno de desarrollo visual **html,** para la programación del software se utiliza el lenguaje **Php** y otras herramientas que se describen en el siguiente capítulo del presente documento.

Gestor de base de datos MySql es un servidor de base de datos multiusuario y es Cliente/Servidor que consta de un *deamon* en el servidor llamado MySql y muchos programas clientes a demás de diferentes bibliotecas, este es un gestor de base de datos preparado a fondo para Internet, su utilización es libre.

Servidor Web Apache es un software de código abierto que funciona sobre cualquier plataforma. Tiene capacidad de servir páginas estático como de contexto dinámico a través de otras herramientas que facilitan la actualización de los contenidos mediante base de datos, ficheros u otras fuentes de información.

2.9. IMPLEMENTACIÓN

Para realizar la implementación se debe agrupar todos los elementos que intervienen en el desarrollo del sistema, incluyendo al manual del sistema, archivos de configuración, archivos de datos, componentes software, etc.

El manual del sistema tiene la finalidad de proporcionar la información del sistema, a nivel de análisis de manera que permita hacer cambios, modificaciones y eliminaciones. El manual describe los siguientes casos: Caso de uso, base de

datos, modelo relacional, plataformas en que funciona el sistema y requerimientos de hardware y software.

La implementación es una colección de componentes y elementos de software, estos componentes incluyen: Ficheros ejecutables, ficheros de código fuente, y otro tipo de ficheros necesarios para la implementación y despliegue del sistema.

En esta sección se realiza las siguientes tareas:

- **Generación de código,** con los diagramas anteriormente desarrollados se generará el código del sistema;
- **Verificación del código,** mientras se desarrolla el código del sistema, se realizará pruebas de manera constante para avaluar el código generado;
- **Generación de interfaz del usuario,** se realiza la generación de la interfaz de usuario a partir de los diagramas de diseño;
- **Manual de usuario,** es un resumen de secciones y acciones que el usuario puede realizar con el sistema, expone de manera clara y concisa las siguientes funciones: instalación, inicio del sistema, realizar altas, realizar bajas, realizar modificaciones y realizar actualizaciones.

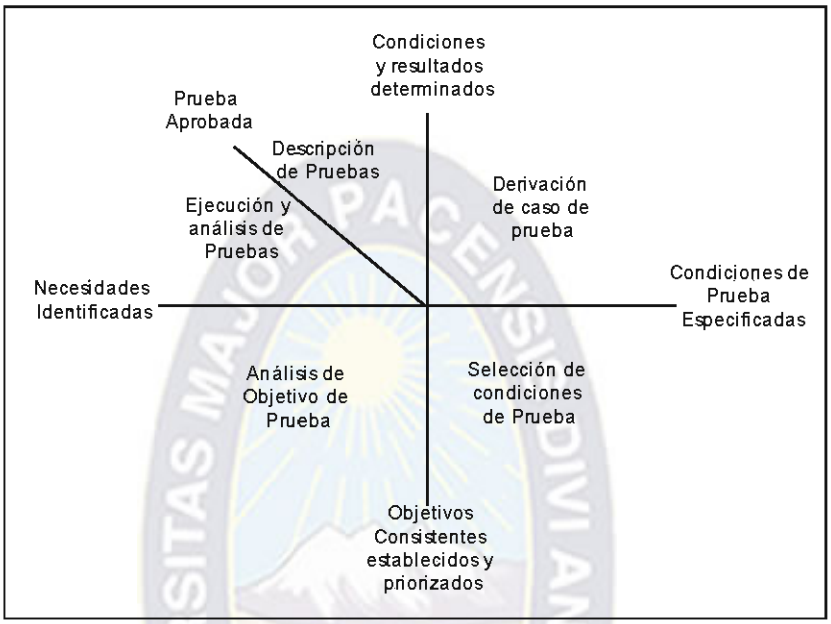
2.9.1. PRUEBAS

Las pruebas de Software, es un proceso usado para identificar posibles fallos de implementación, calidad, o usabilidad de un sistema. El objetivo de las pruebas es encontrar el mayor número posible de errores con una cantidad razonable de esfuerzo, aplicando sobre un lapso de tiempo relista [Presuman, 2002].

Para el presente proyecto se realizan las pruebas de Estrategia Espiral, el cual tiene la ventaja de ser flexible y maximiza la cantidad de prueba realizada durante el desarrollo del prototipo. El Modelo Espiral, permite un desarrollo estable de

pruebas de cada cambio del sistema y facilita las pruebas de bajo nivel, a medida que se crea el código, el cual se muestra en la siguiente Figura 2.20.

Figura 2.20. Modelo Espiral



Fuente: [Presman, 2002]

Fases del Modelo Espiral:

- **Planteamiento de Objetivos**, se identifican los objetivos específicos para cada fase del proyecto;
- **Identificación y reducción de riesgos**, los riesgos clave se identifican y analizan, y la información sirve para minimizar los riesgos;
- **Desarrollo y Validación**, se elige un modelo apropiado para la siguiente fase del desarrollo;
- **Planeación**, se revisa el proyecto y se trazan planes para la siguiente ronda del espiral.

2.9.2. MANTENIMIENTO DEL SISTEMA

Se basa en la corrección de errores o las adaptaciones que el sistema requiere a medida que esta evoluciona en su entorno, durante esta fase se encuentran cuatro tipos de cambio:

- **Mantenimiento Correctivo**, es aquel proceso orientado a la reparación de defectos existentes en un sistema software, ocurre cuando: El programa falla o aborta, un programa produce un resultado que no es acorde con los requisitos, los diseños y requisitos no están acordes con el software que los soporta y la documentación de usuario lleva a conclusiones erróneas al propio usuario hacia actividades que provoquen resultados incorrectos o fallas en el sistema;

- **Mantenimiento Adaptativo**, es el proceso para mejorar la funcionalidad del software, hardware y su documentación, el mantenimiento adaptativo mejora la productividad automatizado actividades;

- **Mantenimiento Perfectivo**, es un método de pulir o refinar la calidad del software y su documentación;

- **Mantenimiento Preventivo**, es el que se ejecuta para prevenir fallos antes de que éstos ocurran, es también llamado reingeniería del software. En esencial, el mantenimiento preventivo hace cambios en programas a fin de que se puedan corregir, adaptar y mejorar más fácilmente.

2.10. ESTÁNDARES APLICABLES

Hoy en día las compañías de todo el mundo industrializado reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general. La industria de desarrollo de software no es la excepción, por lo que en los últimos

años se han realizado intensos trabajos para aplicar los conceptos de calidad en el ámbito del software.

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar.

Entre los principales están la familia de normas ISO 9000, donde se establece distintas referencias a estas y se detallan a continuación las que se utiliza para este proyecto.

2.11. FACTORES Y METRICAS DE CALIDAD

La calidad del software se define como la concordancia de los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo el software desarrollado profesionalmente, es importante resaltar que la calidad se extiende a los atributos técnicos de los modelos de análisis, de diseño y codificación.

Es importante resaltar que la calidad se extiende a los atributos técnicos de los modelos de análisis, de diseño y de codificación. Los modelos que representan una alta calidad (en el sentido técnico) darán lugar a un software de una alta calidad desde el punto de vista del cliente.

2.11.1. Factores de Calidad ISO 9126

Este estándar ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. El estándar identifica cinco atributos clave de calidad.

• **Funcionalidad**

El grado en que el software satisface las necesidades indicadas por los siguientes atributos: Idoneidad, corrección, interoperatividad, conformidad y seguridad.

Tabla 1. Métricas para medir el factor de la funcionalidad

Factor: **Funcionalidad**		
Subcaracterística: **Conveniencia**		
Nombre	**Propósito de la métrica**	**Medición o fórmula de cálculo**
Completitud de la implementación funcional	¿Cuán completa es la es la implementación funcional?	**X = 1- A / B (1)** A = Número de casos de uso(o funciones) B = Número de casos de uso (o funciones) descritas en el Alcance del sistema final (fin de la fase de elaboración) **Y = 1 – A / C (2)** A = Número de casos de uso(o funciones) C= Número de casos de uso (o funciones) descritas en el Alcance del sistema al final de la fase inicial **Z = 1 – A / D (3)** A = Número de casos de uso(o funciones) no implementadas. D = Número de casos de uso(o funciones descritas en la especificación de Requerimientos)
Adecuación funcional	¿Cuán adecuadas son las funciones evaluadas?	**X = 1 – A / B (4)** A = Número de funciones (casos de uso) en las cuales se detectaron problemas en la evaluación. B = Número de funciones(casos de uso) evaluadas

• **Fiabilidad**

La fiabilidad viene dado por la medida de la subcaracterística de la madures, cuyas métricas son: Levantamiento de defectos (atributo para medir la calidad del proceso) y densidad de defectos.

Tabla 2. Métricas para medir el factor de la fiabilidad

Factor: **Fiabilidad**		
Subcaracterística: **Madurez**		
Métricas		
Nombre	**Propósito de la métrica**	**Medición o fórmula de cálculo**
Levantamiento de defectos (atributo para medir la calidad del proceso)	¿cuántos defectos han sido corregidos? ¿Cuál es la proporción de defectos corregidos?	**X = A** A = Número de defectos corregidos en diseño/codificación. **Y = A / B (5)** A = Número de defectos corregidos en diseño/codificación. B = Número de defectos detectados en las revisiones.
Densidad de defectos	¿Cuál es la proporción de defectos respecto al tamaño del producto?	**X = 1 – A / B (6)** A = Número de defectos que no fueron corregidos. B = Tamaño del producto NOTA: El tamaño del producto se mide en líneas de código.
Densidad de defectos	¿Cuántos defectos fueron detectados durante el periodo de prueba?	**X = A / B (7)** A = Número de defectos detectados. B = Tamaño del producto. NOTA: El tamaño del producto se mide en líneas de código.

• Facilidad de mantenimiento

La facilidad con que una modificación puede ser realizada. Esta indicada por los siguientes atributos: Facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Este factor de calidad viene dado según el estándar IEEE 982.1 – 1998 por la métrica del índice de madurez del software (IMS)que proporciona una indicación de la estabilidad del producto software (basada en los cambios que ocurren con cada versión del producto) que se determina con la siguiente formula

$$IMS = (Mt – (Fa + Fc + Fd)/Mt) \quad (8)$$

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.
Fd = número de módulos en la versión anterior que se han borrado en la versión actual.

A medida que el IMS se aproxima a 1 el producto se empieza a estabilizar. El IMS también se puede aplicar como métrica para la planificación de las actividades de mantenimiento del software. El tiempo medio para producir una versión de un producto software puede correlacionarse con el IMS desarrollándose modelos empíricos para el mantenimiento.

• Portabilidad

La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Este factor viene dado por la medida de la subcaracteristica de facilidad de instalación, cuyas métricas es la facilidad de instalación que se encuentra explicado en la tabla 3.

Tabla 3. Métricas para medir el factor de portabilidad

Factor: **Portabilidad**		
Subcaracterística: **Facilidad de instalación**		
Métricas		
Nombre	**Propósito de la métrica**	**Medición o fórmula de cálculo**
Facilidad de instalación	¿Puede el usuario o quien mantiene el software fácilmente instalar el software en un ambiente operacional?	**X = A / B (9)** A = Número de casos en que el usuario es exitoso en la operación de instalación. B = Número total de casos en que el usuario intenta ejecutar la operación de instalación. .

• **Usabilidad**

Grado en que el software es fácil de usar, tiene los siguientes subatributos. Facilidad de comprensión, facilidad de aprendizaje y operatividad.

Este factor viene dado por la medida de las subcaracterísticas de la capacidad de ser entendido (*Understandability)*, y la operabilidad cuyas métricas son: completitud de la descripción, consistencia operacional en el uso que se encuentra explicado en la tabla 4.

Tabla 4. Métricas para medir el factor de usabilidad

Factor: **Usabilidad**		
Subcaracterística: **Capacidad de ser entendido (Understandability)**		
Métricas		
Nombre	**Propósito de la métrica**	**Medición o fórmula de cálculo**
Completitud de la descripción	¿Qué proporción de las funciones (casos de uso)o tipos de funciones se describen en la descripción del producto (documentación de usuario, ayuda, etc.)?	**X = A / B (10)** A = Número de funciones (casos de uso) o tipos de funciones descritas en la descripción del producto. B = Número total de funciones (casos de uso) o tipos de funciones. NOTA: Esta medida indica si los usuarios potenciales entenderán la capacidad del producto luego de leer la descripción del producto.
Subcaracteristica: **Operabilidad**		
Nombre	**Propósito de la métrica**	**Medición o fórmula de cálculo**
Densidad de defectos	¿Cuán consistentes son los componentes de la interfase de usuario?	**X = 1 - A / B (11)** A = Número de funciones que el usuario encontró inaceptablemente inconsistentes según sus expectativas. B = Número de funciones usadas por el usuario durante el período de prueba. **Y = A / UOT (12)** A = Número de funciones que el usuario encontró inaceptablemente inconsistentes según sus expectativas. OUT = Tiempo de operación del usuario (durante el período de observación).

MARCO APLICATIVO

CAPÍTULO III

3.1. INTRODUCCIÓN

En el presente capítulo se desarrolla la aplicación del marco teórico, donde se especifican los requisitos del producto, desarrollo, construcción, implementación, pruebas y actividades de mantenimiento y métricas del sistema.

El presente proyecto tomo como objetivo de estudio a la Unidad Desconcentrada SUSTENTAR, que es una institución fiscal dependiente del Ministerio de Desarrollo Rural y Tierras, que gestiona, programas y proyectos.

3.2. RECOPILACIÓN DE NECESIDADES

Descubrir las necesidades es muy importante por ello en esta acción se realiza la recolección de información acerca de la función que cumple la institución.

Descubrir los procesos de negocio

A continuación se describe la estructura y composición de la Unidad Desconcentrada SUSTENTAR y las funciones que cumplen:

- **Comité de Aprobación, Seguimiento y Revisión de Programas y Proyectos,** es el encargado de Coordinar con otras autoridades nacionales, departamentales y locales las distintas actuaciones que se llevan a cabo, Aprobar los programas y/o proyectos a ser ejecutados.

- **La Unidad de Coordinación Nacional,** es la instancia encargada de aplicar las políticas y lineamientos generales del programa definidos por la MAE del MDRyT, la coordinación institucional y la administración de los recursos financieros de los proyectos destinados al programa

SUSTENTAR. Esta instancia estará conformada por el Coordinador General y por tres unidades:

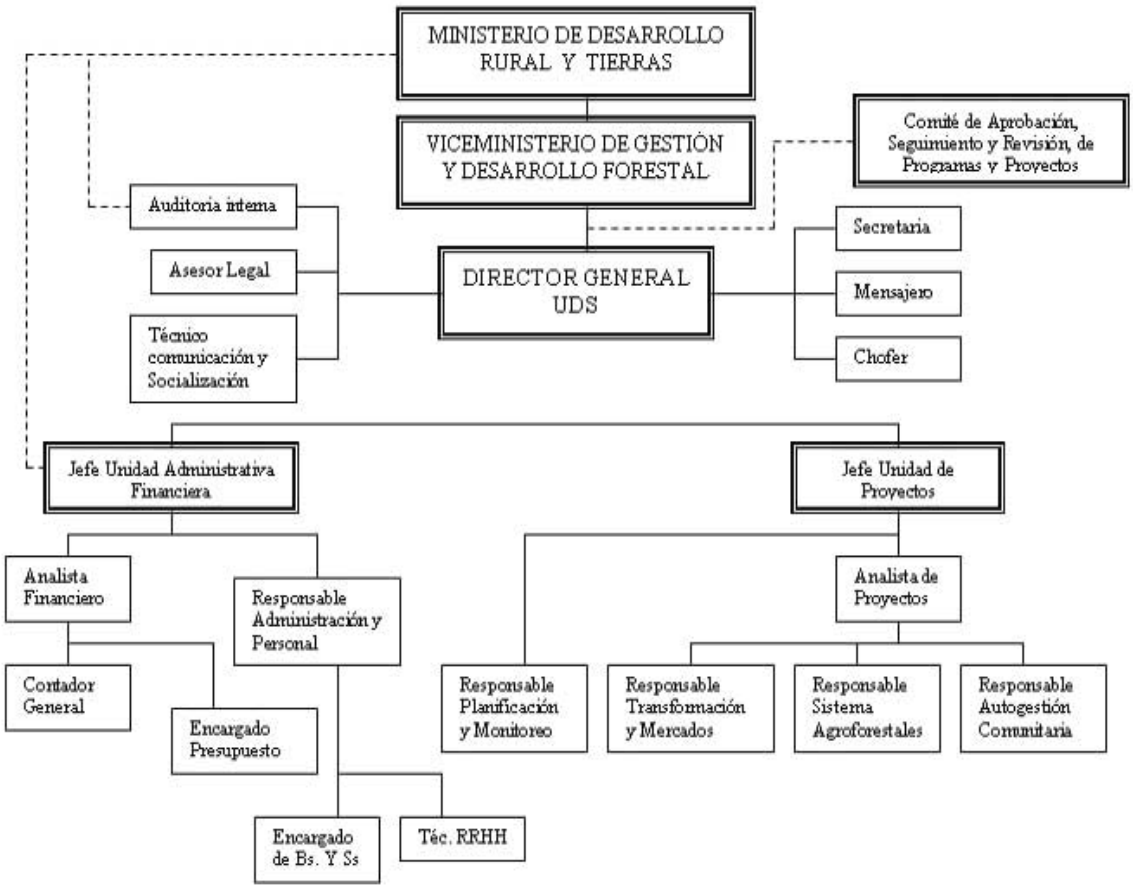
- Unidad de Proyectos
- Unidad de Planificación y Monitoreo
- Unidad Administrativa y Financiera

- **Unidades Regionales,** son los responsables de ejecutar programas y proyectos establecidos en los Planes Operativos de la Unidad Desconcentrada SUSTENTAR, en el ámbito de sus competencias.

Identificación de los sistemas cooperativos

En la siguiente Figura 3.1 se muestra la relación entre las áreas de Unidad Desconcentrada SUSTENTAR.

Figura 3.1. Área de Unidad Desconcentrada Sustentar.



Fuente: [Elaboración propia]

Identificación del producto

El Software tiene como nombre Sistema de Información y Seguimiento de Proyectos SIS-PROY.

- **¿Qué hará el sistema?**

El sistema a desarrollarse tendrá los siguientes objetivos: Registrar proyectos de **sector del estado** instituciones públicas, municipios, mancomunidades de municipios y prefecturas, **sectores sociales** organizaciones indígenas, **otros sectores** organizaciones productivas, organizaciones sin fines de lucro.

El sistema permite obtener reportes de proyectos de los distintos sectores como historial y seguimiento de los proyectos.

El sistema realiza el seguimiento y control del desarrollo de los proyectos en diferentes etapas que se desarrolla del proyecto.

- **¿Qué no hará el sistema?**

El sistema no realiza aquellos procesos donde intervenga el procedimiento de documentos físicos del trámite de proyecto.

- **Beneficios**

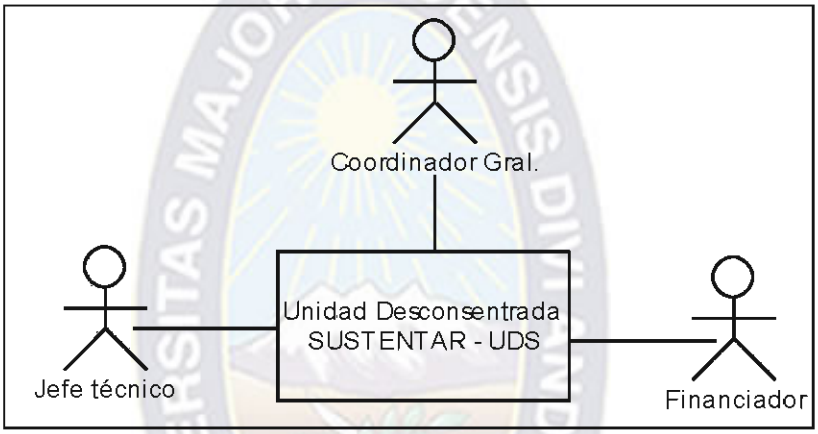
Este producto de software ayuda en las actividades básicas del control de proyectos; ayuda en el seguimiento con datos eficientes del proyecto.

3.3. ANÁLISIS

3.3.1. ANÁLISIS DEL SISTEMA ACTUAL

En la siguiente Figura 3.2 se muestra los actores que intervienen en el actual sistema de la Unidad Desconcentrada SUSTENTAR, pueden interactuar varios actores.

Figura 3.2. Actores del sistema Unidad Desconcentrada Sustentar



Fuente: [Elaboración propia]

Una de las técnicas utilizadas para recopilar datos de la información de Unidad Desconcentrada SUSTENTAR. Es la entrevista, la cual proporciona una información cualitativa.

A continuación se detalla el funcionamiento de Unidad Desconcentrada SUSTENTAR con las entrevistas realizadas.

Actor: Jefe Técnico

Descripción: Es el encargado de todo el trámite realizando las siguientes funciones:

- Recibir los fólderes de trámite.
- Brinda información general de trámite del proyecto.
- Registra las solicitudes de trámite del proyecto.
- Realiza la recepción de documentos.
- Administra y ordena de acuerdo a los sectores los trámites.

Actor: Coordinador Gral.

Descripción: Es el encargado de revisar los proyectos y control de aprobación, realiza las siguientes funciones:

- Verifica el presupuesto de cada carpeta;
- Realiza el seguimiento, revisión de programas y proyectos;
- Gestiona los recursos de acuerdo al proyecto;
- Manda una respuesta al proyecto;

Financiador **Actor:** Financiador
Descripción: Es el encargado de revisar cada carpeta de acuerdo a los sectores de tramite, realiza las siguientes funciones: • Revisa las carpetas de trámite de proyectos; • Revisa el costo del proyecto en tramite; • Revisa fecha y hora del proyecto;

Diagramas de Casos de Uso

En la Figura 3.3 se muestra el modelo de casos de uso del negocio identificado.

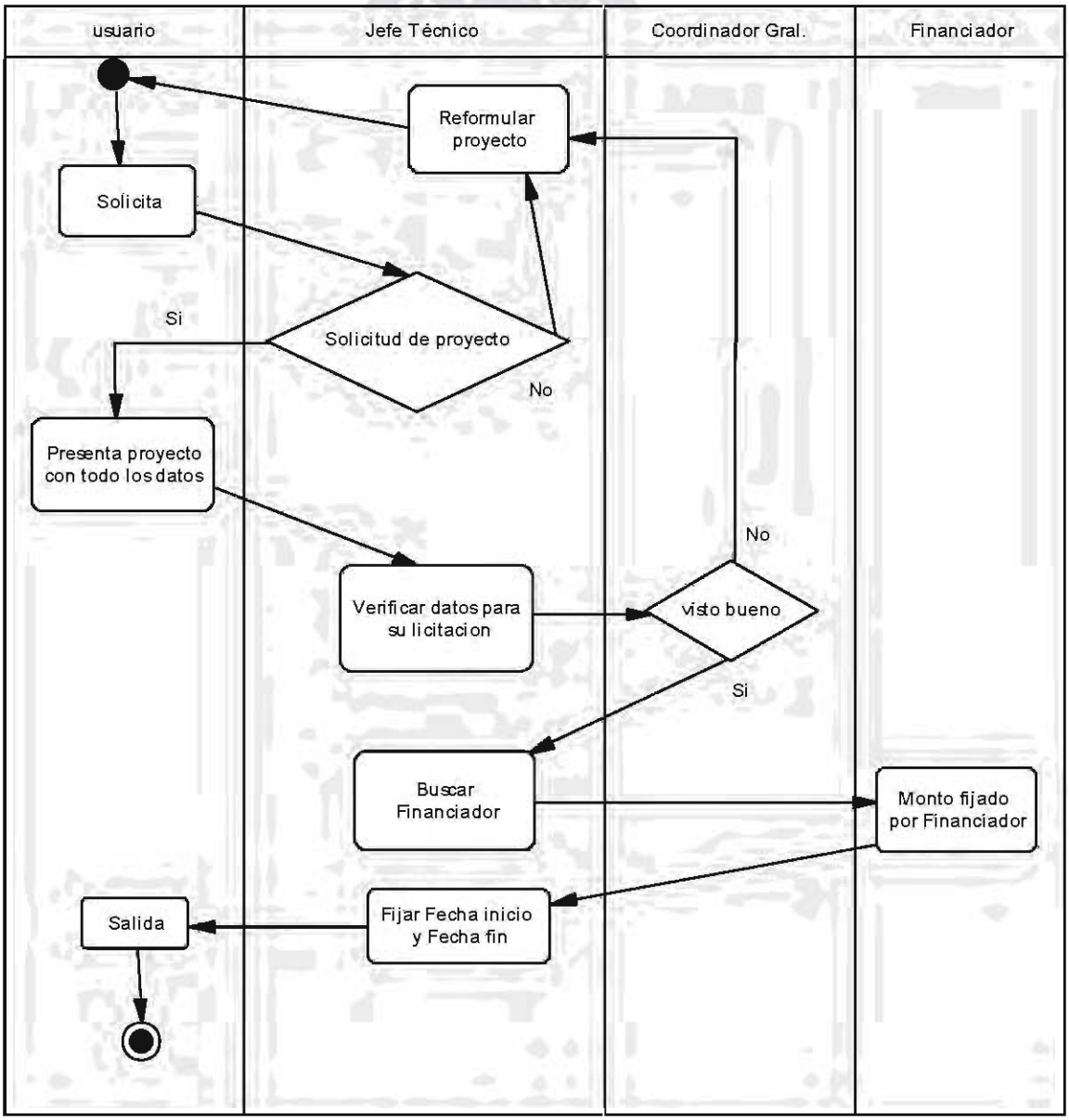
Figura 3.3. Diagrama de casos de Unidad Desconcentrada SUSTENTAR.

Fuente: [Elaboración propia]

Elaboración de los Cambios del Estado de Objeto

Los diagramas de actividades indican como se ejecuta el trabajo, proporcionando una descripción de los actores que se realizan en la Unidad Desconcentrada SUSTENTAR. Se muestra en la Figura 3.4 el diagrama de actividades relacionado con el sistema actual.

Figura 3.4. Diagrama de actividades sistema actual



Fuente: [Elaboración propia]

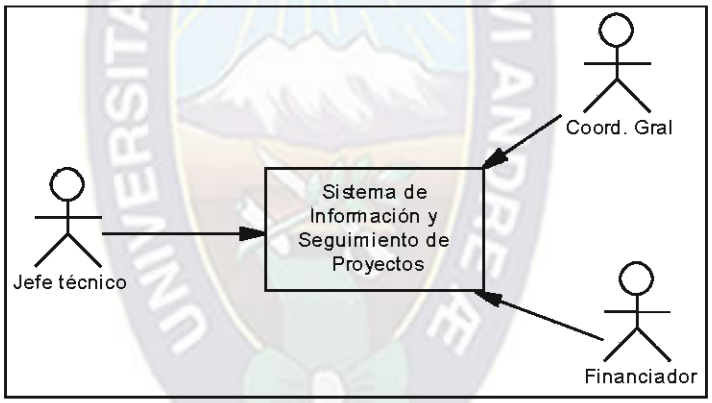
El sistema manual que se lleva a cabo en la Unidad Desconcentrada SUSTENTAR es para proporcionar información de licitación de proyectos y seguimiento de proyectos en los diferentes sectores.

Para consultas e informaciones deben acudir a los cuadernos de registro, haciendo el proceso muy lento.

3.3.2. ANÁLISIS DEL NUEVO SISTEMA

Como una primera aproximación identificamos a los actores que interactúan con el sistema, como se ve en la Figura 3.5.

Figura 3.5. Identificación de actores



Fuente: [Elaboración propia]

Se define a los actores y los roles que desempeñan dentro del sistema. Los actores en la descripción de los requisitos son: Recepcionista, Comité de Aprobación, Evaluador.

Actor: Jefe Técnico
Descripción: Se encarga de registrar nuevos proyectos.
Caso de Uso: Autenticación de recepcionista Registrar proyectos Registro de Tramites Recepción de documentos Ordenar Proyectos Altas, Bajas y Modificaciones de los proyectos

Actor: Coordinador General
Descripción: Se encarga de verificar los proyectos para aprobar
Caso de Uso: Autenticación de comité de aprobación Verificar Carpeta Revisión de Proyectos Gestión de Proyectos Ordenar por fechas

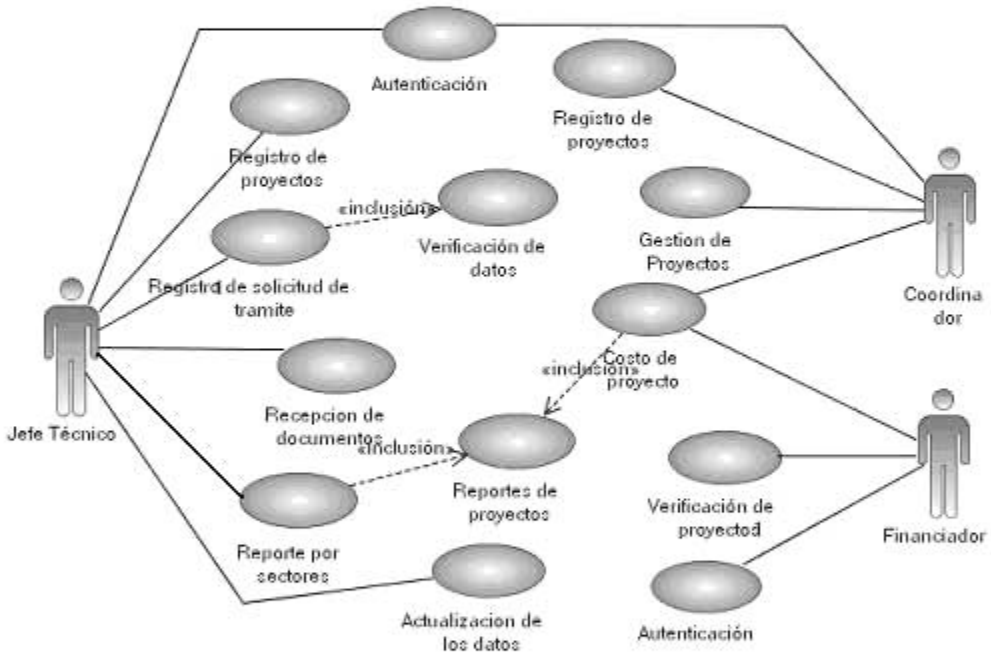
Actor: Financiador
Descripción: Se encarga de financiar los proyectos que cumplen los requisitos

Caso de Uso:
Autenticación de financiador
Revisión de proyectos
Costo de Proyectos

Diagramas de Casos de Uso

El diagrama de casos de uso muestra una parte de la funcionalidad general del sistema. La Figura 3.6 muestra el modelo de casos de uso general identificado, en el cual se observa las interacciones que hay entre un actor y un caso de uso.

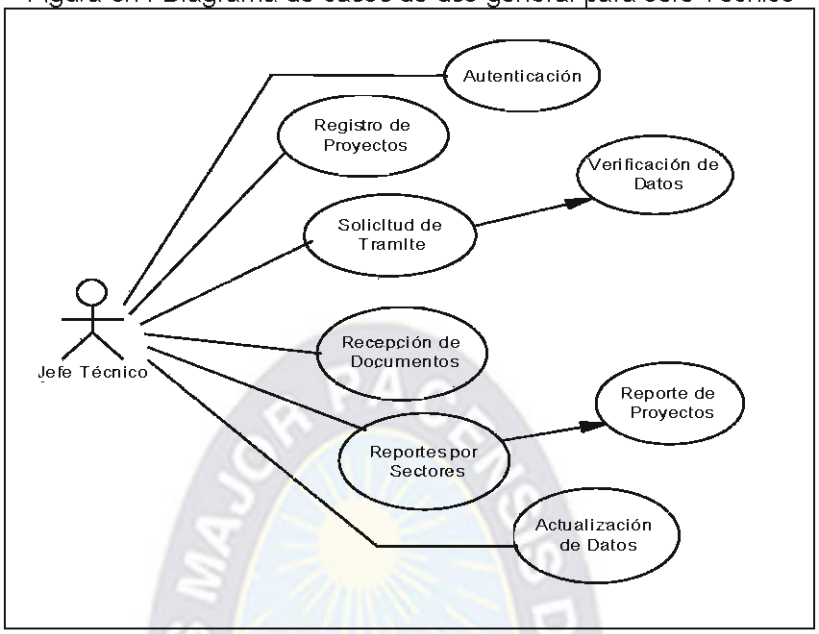
Figura 3.6. Diagrama de casos de uso general



Fuente: [Elaboración propia]

En la Figura 3.7 se muestra el diagrama de casos de uso en general para el paquete Jefe técnico. La descripción de los restantes casos de uso se encuentra en el Anexo D.

Figura 3.7. Diagrama de casos de uso general para Jefe Técnico



Fuente: [Elaboración propia]

A continuación se describe la funcionalidad de los diagramas de casos de uso presentados. Para ello, se emplea una ficha de descripción.

Nombre de Caso de Uso: "Autenticación de usuario"

Actores: Jefe Técnico, Coordinador, Financiador

Tipos: Primario de carácter esencial

Descripción:

Pre-condición: El usuario entra en la opción identificación de usuario.

Post-condición: El usuario en reconocido por la aplicación.

Flujo de evento: El caso de uso empieza cuando el usuario visualiza la interfaz que permite la intersección de nombre de usuario y contraseña. En esta interfaz el usuario introduce sus datos de 'usuario' y 'Contraseña' para entrar al sistema.

Nombre de Caso de Uso: "Registro de Proyectos"

Actores: Jefe Técnico

Tipo: Primario de carácter real

Descripción:

Pre-condición: El usuario entra en la opción registro de nuevo proyecto.

Post-condición: El evento es reconocido por la aplicación.

Flujo de evento: El caso de uso empieza cuando el usuario visualiza la interfaz que le permite realizar el registro de un proyecto, el usuario ingresa los datos en el formulario de registro de proyecto y finalmente guarda los datos.

Nombre de Caso de Uso: "Solicitud de Tramite"

Actores: Jefe Técnico

Tipo: Primario de carácter real

Descripción:

Pre-condición: El usuario entra en la opción verificar proyecto.

Post-condición: El evento es reconocido por la aplicación.

Flujo de evento: El caso de uso empieza cuando el usuario visualiza la interfaz que le permite seleccionar gestión de proyectos, para luego poder crear un "nuevo proyecto".

Nombre de Caso de Uso: "Recepción de Documentos"

Actores: Jefe Técnico

Tipo: Primario de carácter real

Descripción:

Pre-condición: El usuario entra en la opción de gestión de proyecto.

Post-condición: El evento es reconocido por la aplicación.

Flujo de evento: El caso de uso empieza cuando el usuario visualiza la interfaz que le permite realizar el registro de los proyectos. El usuario realiza el registro de proyectos en el formulario de proyectos.

Nombre de Caso de Uso: "Reporte por sectores"

Actores: Jefe Técnico

Tipo: Primario de carácter real

Descripción:

Pre-condición: El usuario entra en la opción Reportes.

Post-condición: El evento es reconocido por la aplicación.

Flujo de evento: El caso de uso empieza cuando el usuario visualiza la interfaz

que le permite ver los proyectos registrados, el usuario selecciona un proyecto para ver los detalles del proyecto seleccionado.

Nombre de Caso de Uso: "Actualización de Datos"

Actores: Jefe Técnico

Tipo: Primario de carácter real

Descripción:

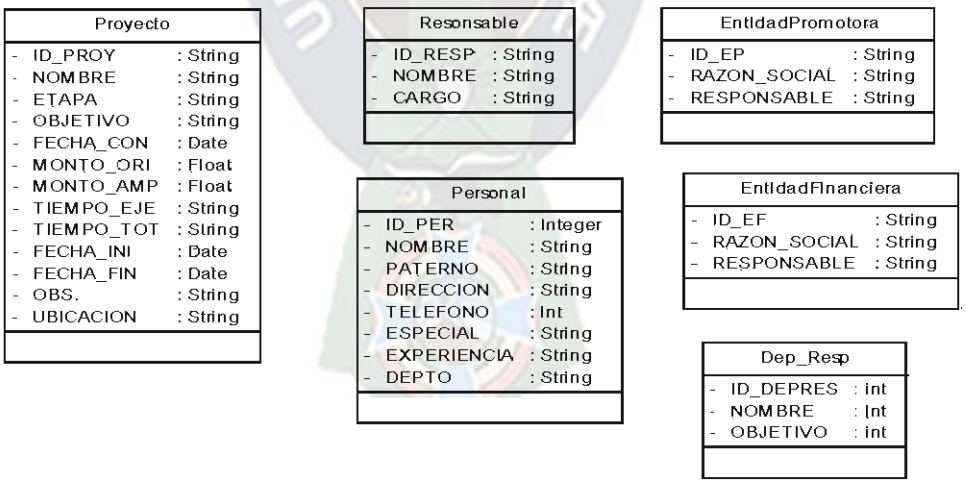
Pre-condición: El usuario entra en la opción Proyectos.

Post-condición: El evento es reconocido por la aplicación.

Flujo de evento: El caso de uso empieza cuando el usuario visualiza la interfaz de proyectos para revisar los datos y actualizar los detalles observados.

3.4. MODELO CONCEPTUAL

En el modelo conceptual se presentan los conceptos más importantes en el análisis. Los conceptos del modelo conceptual con sus respectivos atributos o valores lógicos definidos son:



A continuación una breve explicación de los atributos de conceptos:

ID_PROY: Identidad del proyecto que debe ser introducido por el usuario

NOMBRE:	Nombre del proyecto
ETAPA:	Etapa del proyecto
OBJETIVO:	Objetivo del proyecto
FECHA_CON:	Fecha de conclusión
MONTO_ORI:	Monto original
MONTO_AMP:	Monto mas ampliación
TIEMPO_EJE:	Tiempo de Ejecución
TIEMPO_TOT:	Tiempo Total
FECHA_INI:	Fecha de inicio
FECHA_FIN:	Fecha de final
OBS:	Observación del proyecto
UBICACIÓN:	Ubicación del proyecto
ID_RESP:	Identidad del proyecto
NOMBRE:	Nombre de persona que incluye sus apellidos
CARGO:	Cargo que ocupa el responsable de departamento
ID_EP:	Identidad de la Entidad Gestora
NOM_ENTPROM:	Nombre de la entidad gestora
RESPONSABLE:	Nombre del responsable
ID_EF:	Identidad de la entidad financiera
NOM_ENTFINA:	Nombre de la entidad financiera
RESPONSABLE:	Nombre del responsable
ID_DEPRES:	Identidad del departamento responsable
NOMBRE:	Nombre del responsable
OBJETIVO:	Objetivo que persigue el departamento responsable
ID_PER:	Identidad del Personal
DIRECCION:	Dirección del personal

TELEFONO: Teléfono del personal

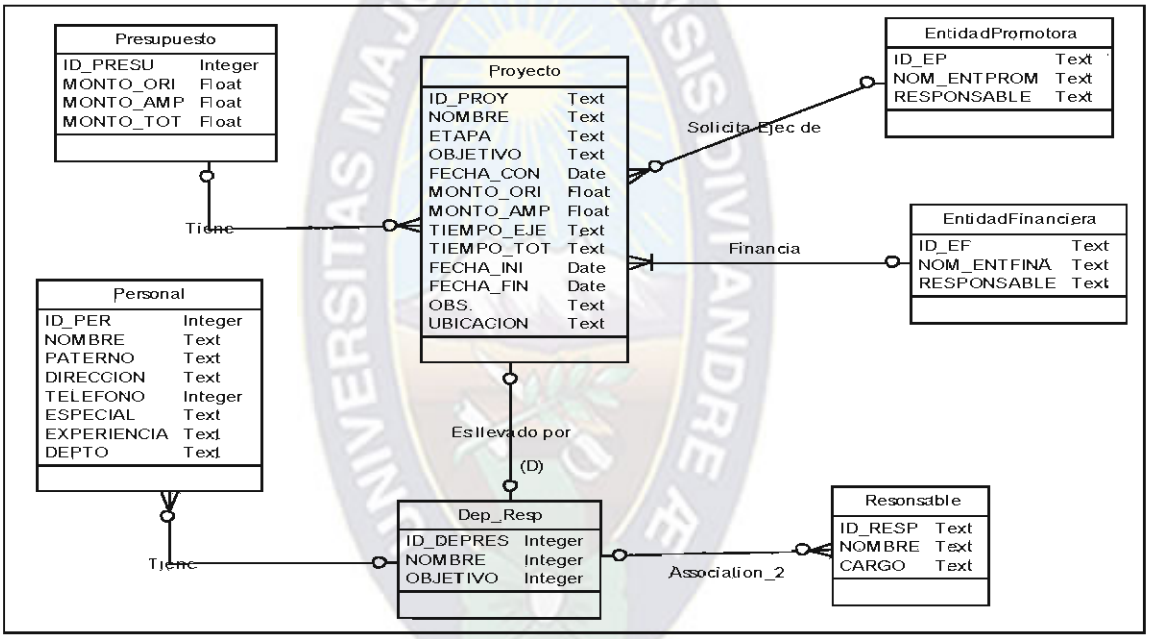
ESPECIAL: Especialidad del personal

EXPERIENCIA: Años de experiencia del personal

DEPTO: Lugar geográfico de residencia del profesional externo

Definida los conceptos con sus respectivos atributos, pasamos a definir el modelo conceptual que se muestra en la figura 3.8

Figura 3.8. Modelo Conceptual



Fuente: [Elaboración propia]

Elaboración de los Cambios de Estado de Objetos

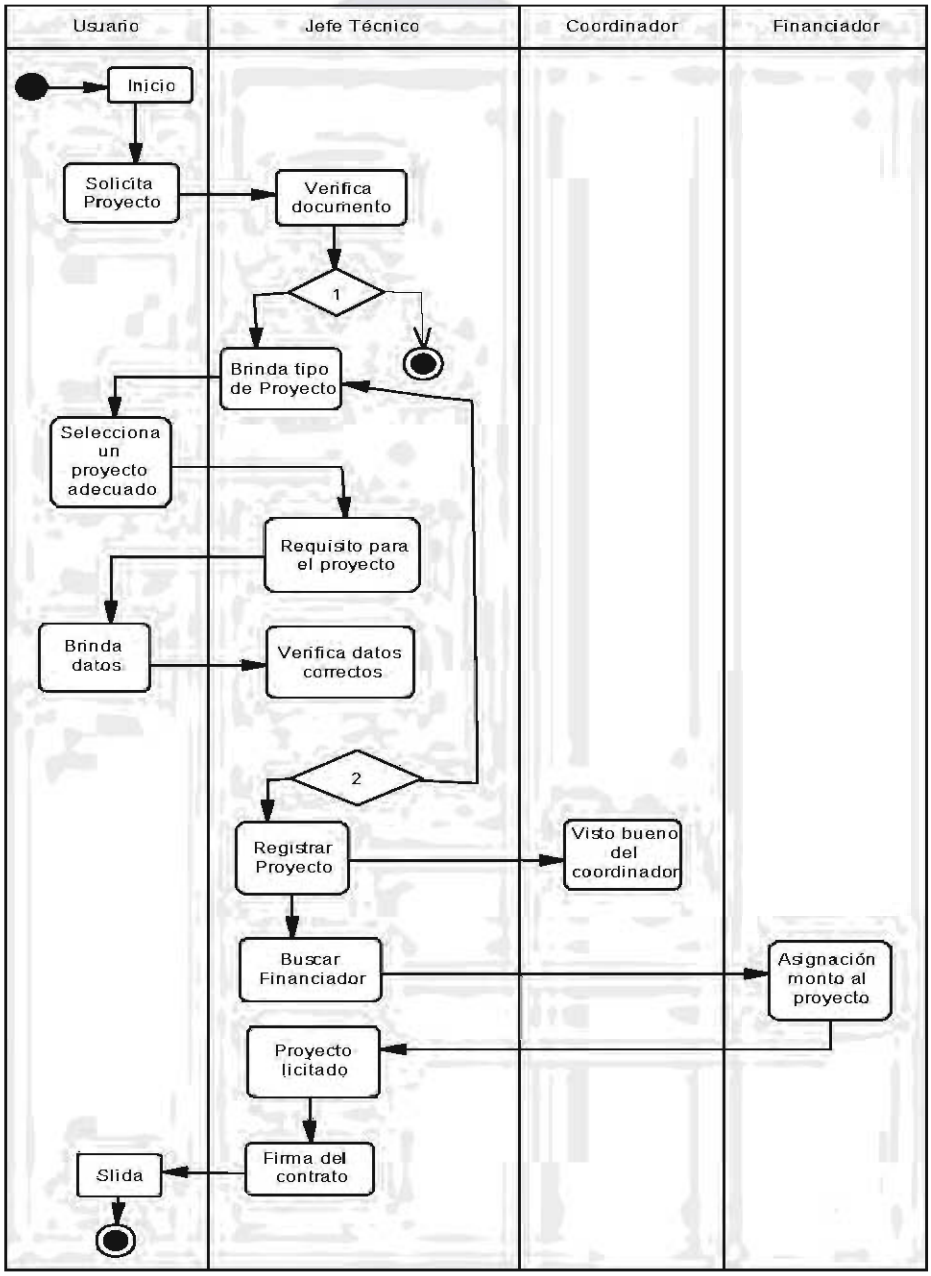
Los diagramas de actividades son en esencia diagramas de flujo, con algunos elementos adicionales que le permiten expresar conceptos como la concurrencia y la división del trabajo [Elizondo, 2005].

Los diagramas de actividades nos indican como se ejecuta el trabajo, proporcionando una descripción detallada de cada uno de los actores que realiza la Unidad Desconcentrada SUSTENTAR. En la figura 3.9 se muestra el diagrama de actividad relacionado con el sistema.

3.5 . DIAGRAMA DE SECUENCIA

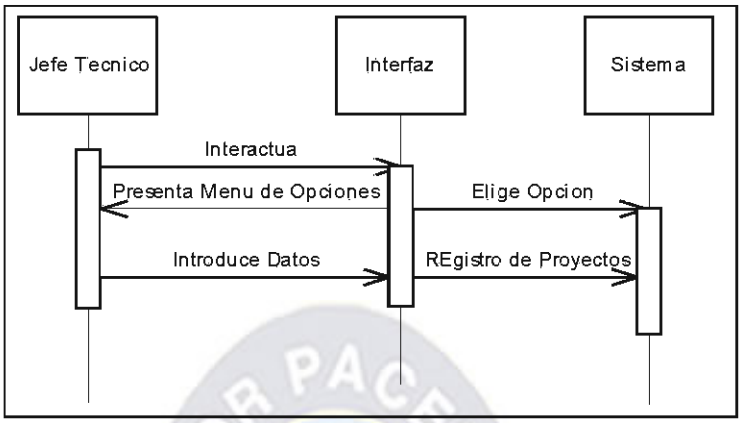
Los diagramas de secuencia son los primeros diagramas dinámicos cuyo propósito es el de exponer y visualizar el sistema en ejecución, es decir el flujo de los eventos de los actores al sistema para una mejor comprensión del sistema.

Figura 3.9. Modelo Conceptual



Fuente: [Elaboración propia]

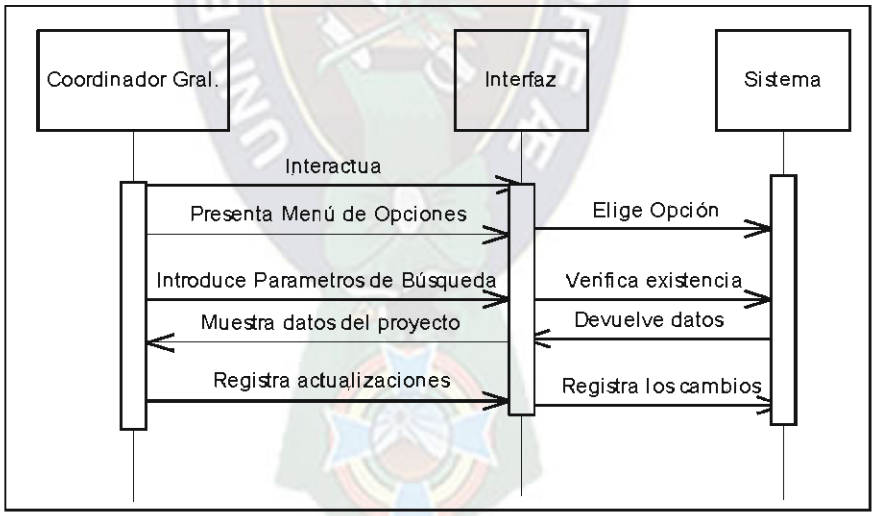
Figura 3.10. Diagrama de Secuencia. Registro de proyectos



Fuente: [Elaboración propia]

La figura 3.10 describe el diagrama de secuencia del proceso de registro de proyectos en la que el primer paso es que el usuario interactúa con el sistema a través de un interfaz que le permite introducir datos necesarios para tal efecto y finalmente el sistema guarda esta información.

Figura 3.11. Diagrama de Secuencia. Seguimiento de Proyectos



Fuente: [Elaboración propia]

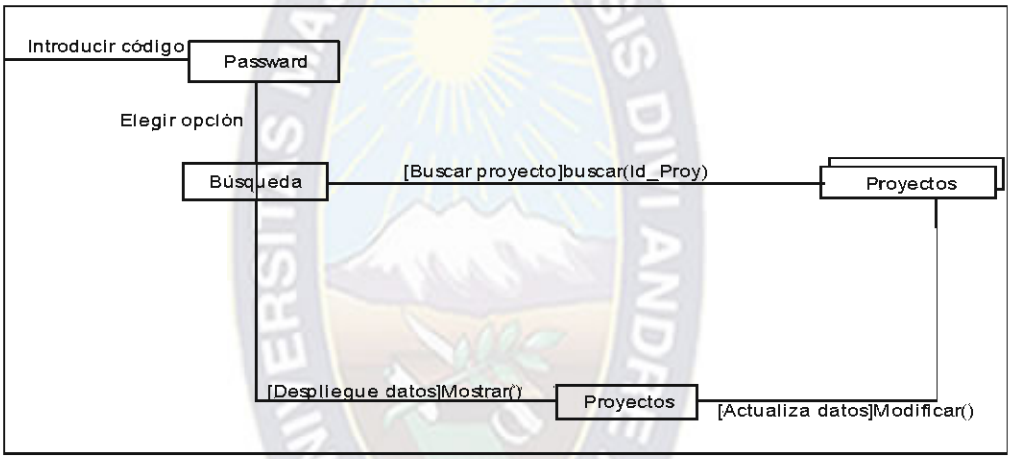
En la figura 3.11 se presenta el diagrama de secuencia de Seguimiento de Proyectos en la que primeramente, a través del interfaz de usuario, éste digita el código de proyecto a ser actualizado, luego de ser encontrado por el sistema es

mostrado por pantalla los datos que se tienen de dicho proyecto para luego sea modificados por el usuario y grabados por el sistema.

3.6. DIAGRAMAS DE COLABORACIÓN

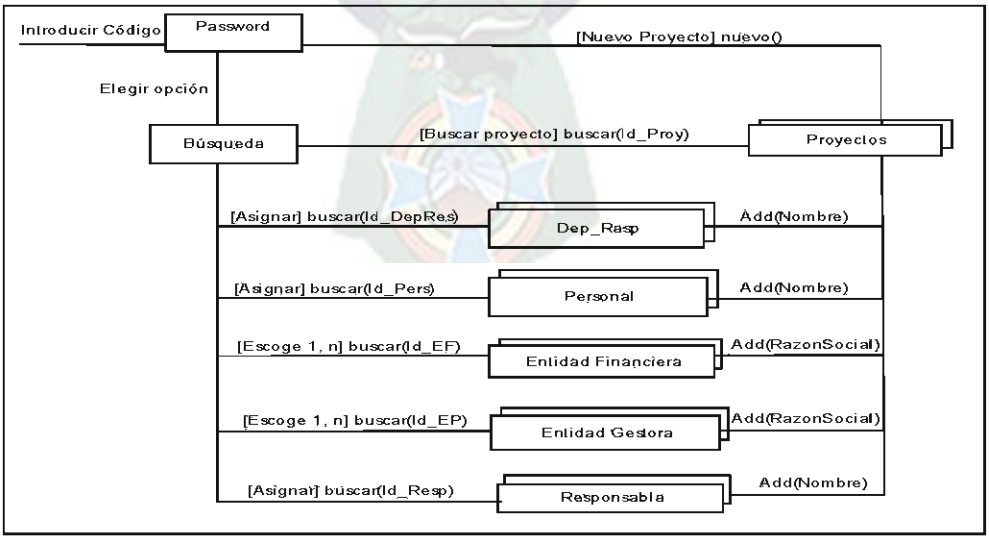
La figura 3.12 y figura 3.13 dan una referencia mucho más detallada de lo que el sistema hace en cada proceso mostrando las operaciones que este realiza para ejecutar las funciones necesarias, todo a través del contexto de un grupo de objetos y las interacción entre objetos.

Figura 3.12. Diagrama de colaboración. Seguimiento de proyecto



Fuente: [Elaboración propia]

Figura 3.13. Diagrama de colaboración. Registro de Proyectos

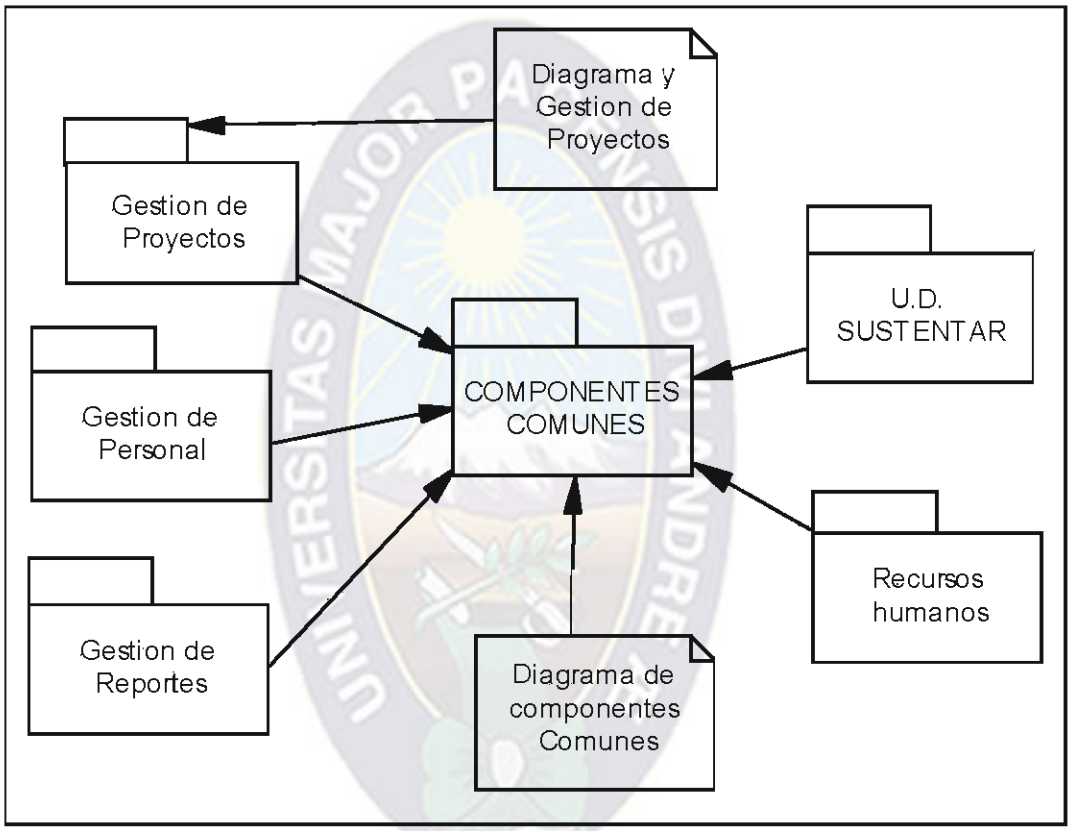


Fuente: [Elaboración propia]

3.7. DIAGRAMA DE PAQUETES

La figura 3.14 muestra la disposición de las partes integrantes de la aplicación y las dependencias entre los distintos módulos de la aplicación.

Figura 3.14. Diagrama de paquetes

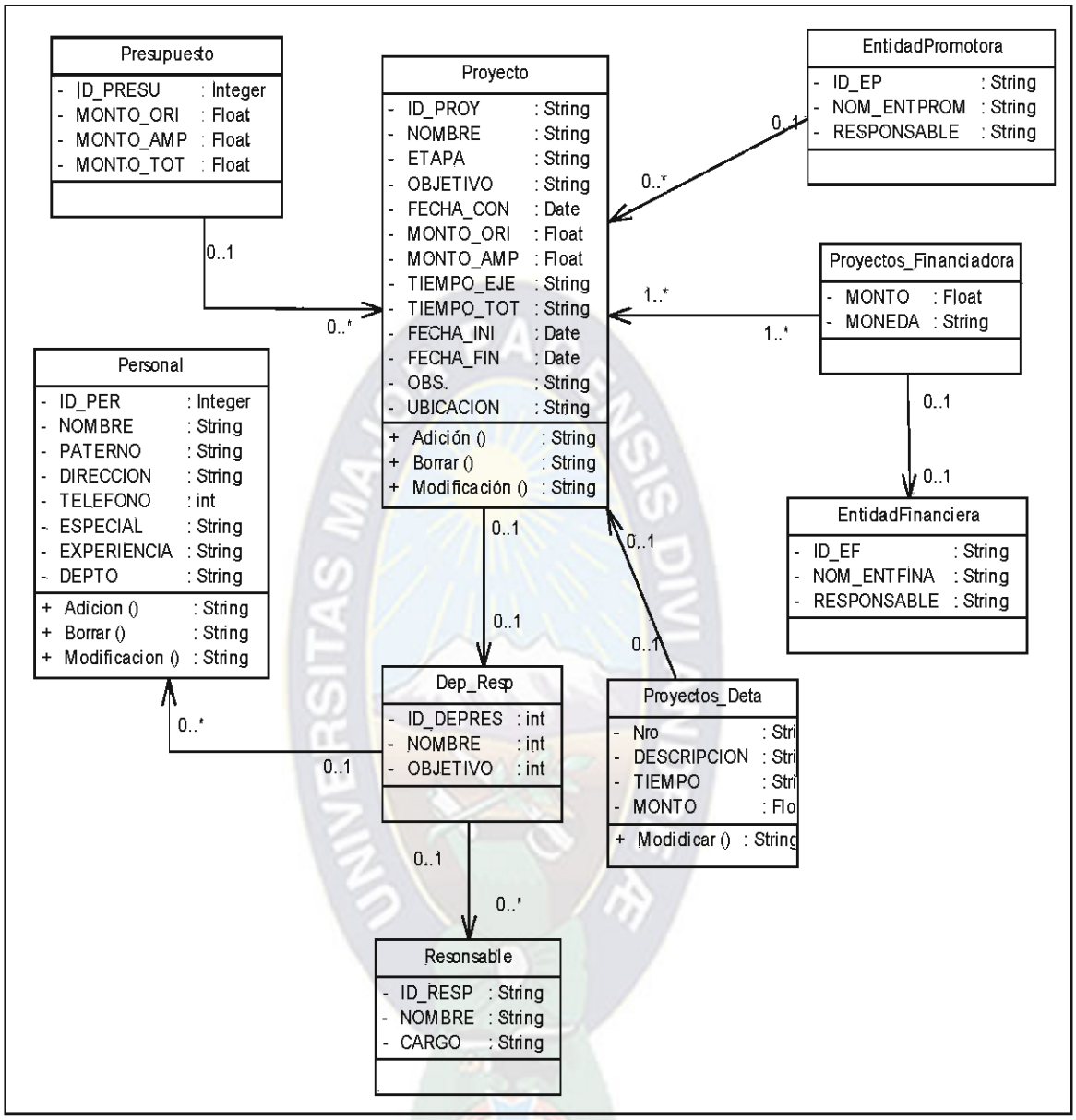


Fuente: [Elaboración propia]

3.8. DIAGRAMA DE CLASES

La figura 3.15 permite identificar las clases del software que interactúan para poder llevar a cabo los procesos del sistema, a diferencia del modelo conceptual, que solo es una abstracción del dominio que permite definir funciones que tiene cada clase.

Figura 3.15. Diagrama de Clases

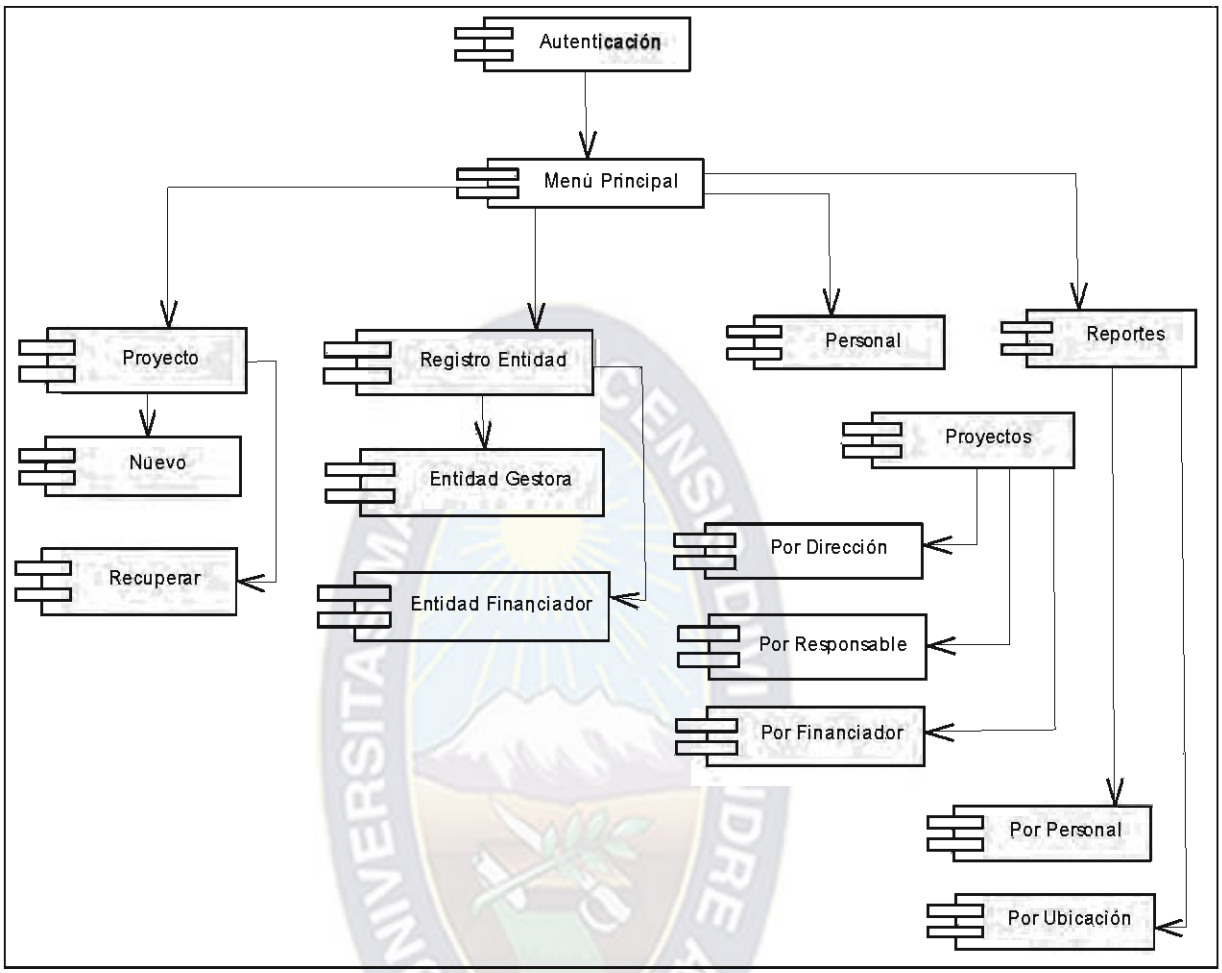


Fuente: [Elaboración propia]

3.9. DIAGRAMA DE COMPONENTES

En la figura 3.16 se muestra la disposición de las partes integrantes de la aplicación y las dependencias entre los distintos módulos de la aplicación.

Figura 3.16. Diagrama de Componentes

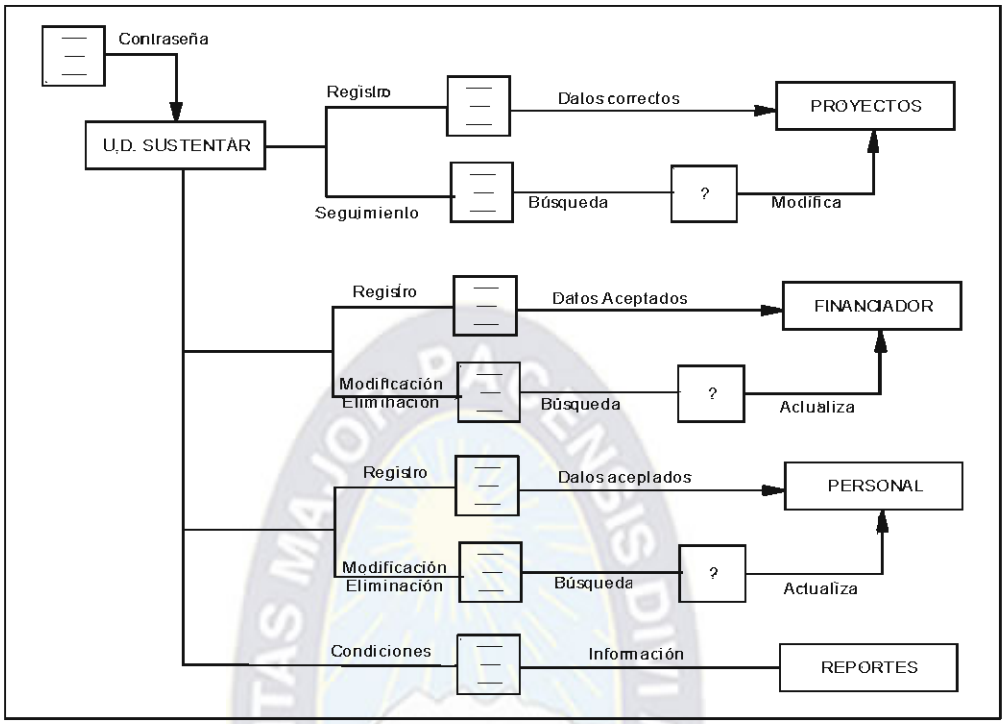


Fuente: [Elaboración propia]

3.10. DIAGRAMA NAVEGACIONAL

Este es un modelo de estructura de navegación que describe como aparecen los elementos de acceso como índices, visitas guiadas y consultas descritas en el capítulo anterior [Figura 3.17]

Figura 3.17. Diagrama Navegacional

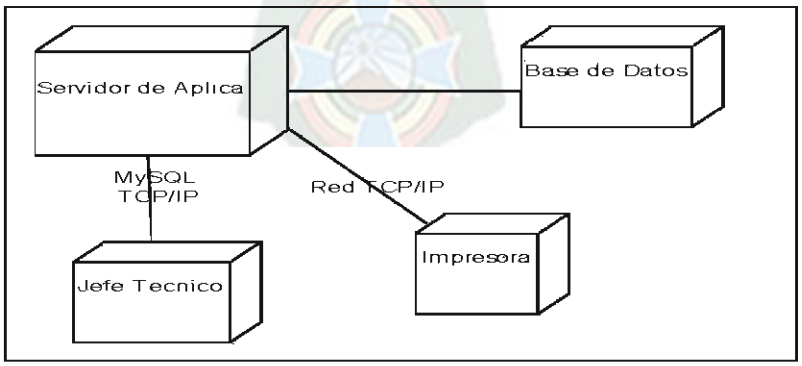


Fuente: [Elaboración propia]

3.11. DIAGRAMA DE DISTRIBUCIÓN

En el diagrama de distribución se diseña la parte lógica y física del sistema, donde interactúan los componentes y hardware del sistema. La figura 3.18 muestra el diagrama de despliegue del sistema.

FIgura 3.18. Diagrama de componentes. Distribución del sistema

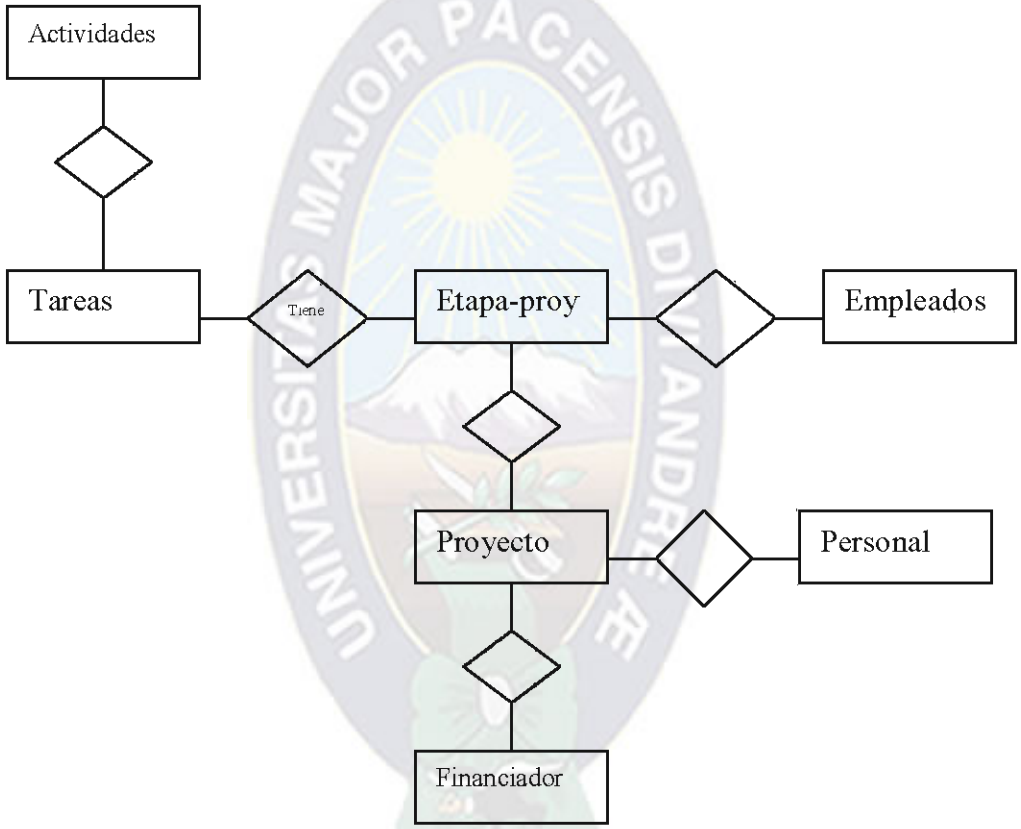


Fuente: [Elaboración propia]

3.12. MODELO DE BASE DE DATOS

El modelo entidad / relación del nuevo sistema está elaborado con las relaciones y entidades que han sido encontradas en el análisis, en la cual cada entidad y relación tiene su respectivo identificador, figura 3.19

Figura 3.19. Diagrama de Entidad Relación



Fuente: [Elaboración propia]

3.13. DISEÑO E IMPLEMENTACIÓN

Se emplean diferentes modelos, de la misma manera que se hizo en la etapa de análisis, de los cuales los más importantes son el modelo de implementación del sistema y el modelo de implementación de programas.

3.14. MODELO DE IMPLEMENTACIÓN DE SISTEMAS

Se establece todos los elementos necesarios para ensamblar y hacer disponible el sistema físico, incluyendo al manual del sistema, archivos de configuración, archivos de datos, componentes software.

Interfaces del usuario

En la presente sección se muestran los diferentes procesos entre el ordenador y el usuario, además se exponen las necesidades y características del programa como zonas de selección, iconos y botones del sistema.
El sistema presenta un entorno gráfico amigable y fácil de usar, brindando contenido en forma de texto e imágenes.

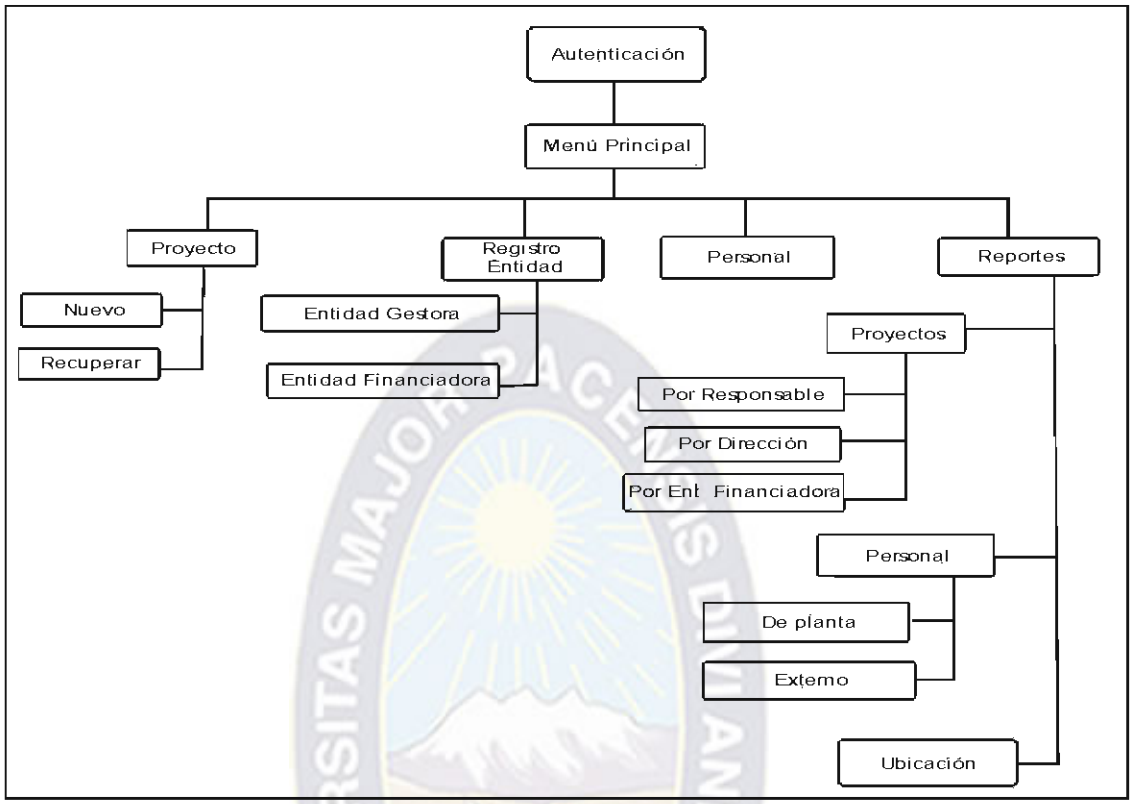
Menús

Este tipo de interfaz proporciona al usuario una lista de opciones en el monitor del sistema, el cual se presenta en la pantalla principal o la primera pantalla. El usuario final esta limitado solo a las opciones desplegables del diseño.

Diagrama de interfaz jerárquico

En la figura 3.20 se muestra la estructura jerárquica del sistema orientado a la interfaz de usuario, que permite orientar desplegar el software o cumplir las funciones que fue diseñado en el capítulo anterior, los cuales se muestran a continuación.

Figura 3.20. Diagrama Jerárquico



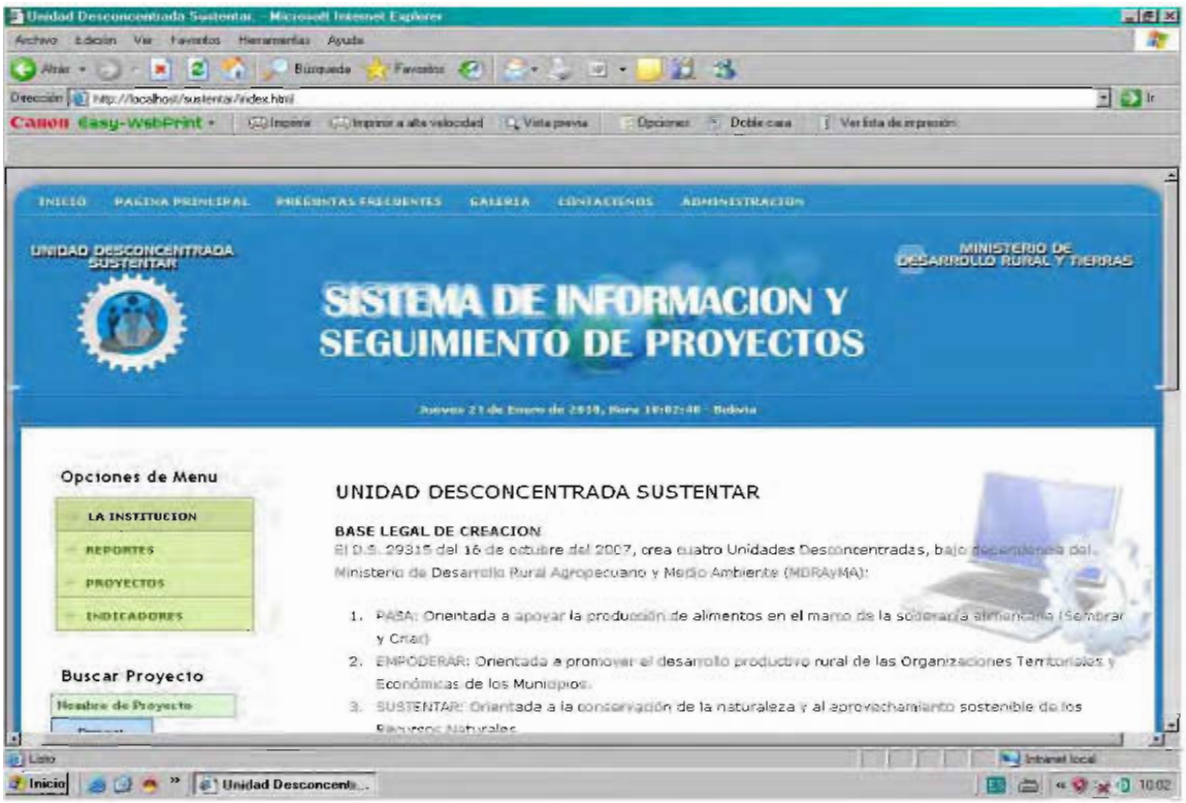
Fuente: [Elaboración propia]

3.15. DISEÑO DE INTERFACES

A continuación se describe de forma general el diseño de interfaz de la página Web para usuario externos de la Institución que se muestra en la grafica 1.

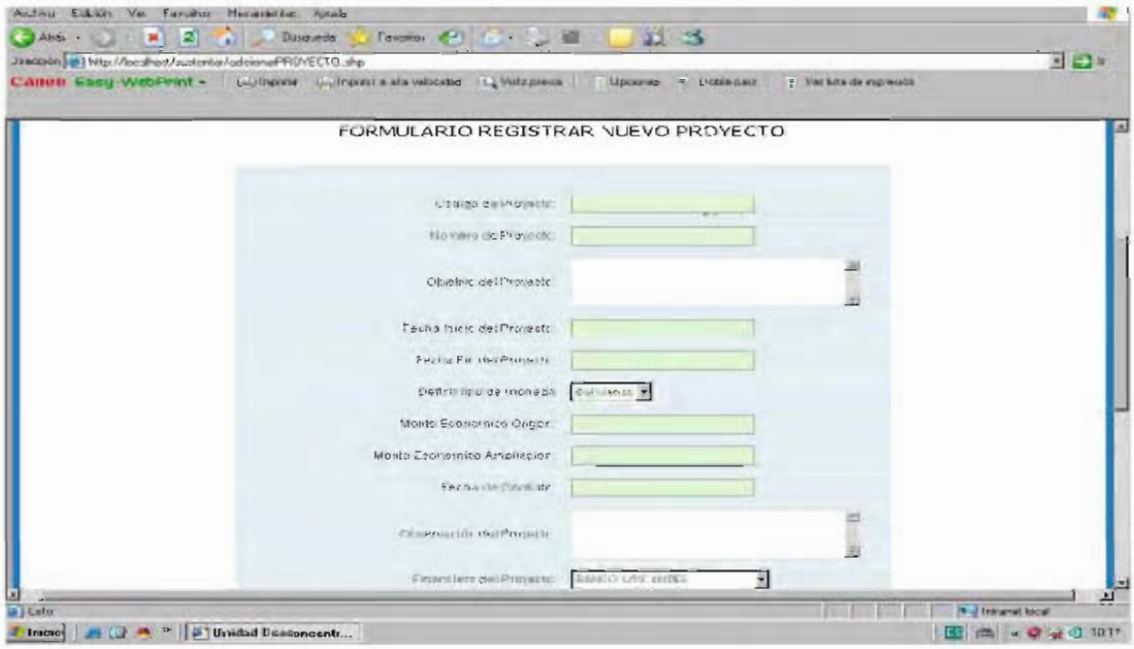
Pantalla Principal de Registro de Proyectos: En la siguiente figura se muestra la pantalla principal de registro de proyectos, donde el Responsable de Proyectos y el Encargado de Seguimiento, podrán interactuar con el sistema, que se muestra en la gráfica 2.

Gráfica 1. Diseño Web



Fuente: [Elaboración propia]

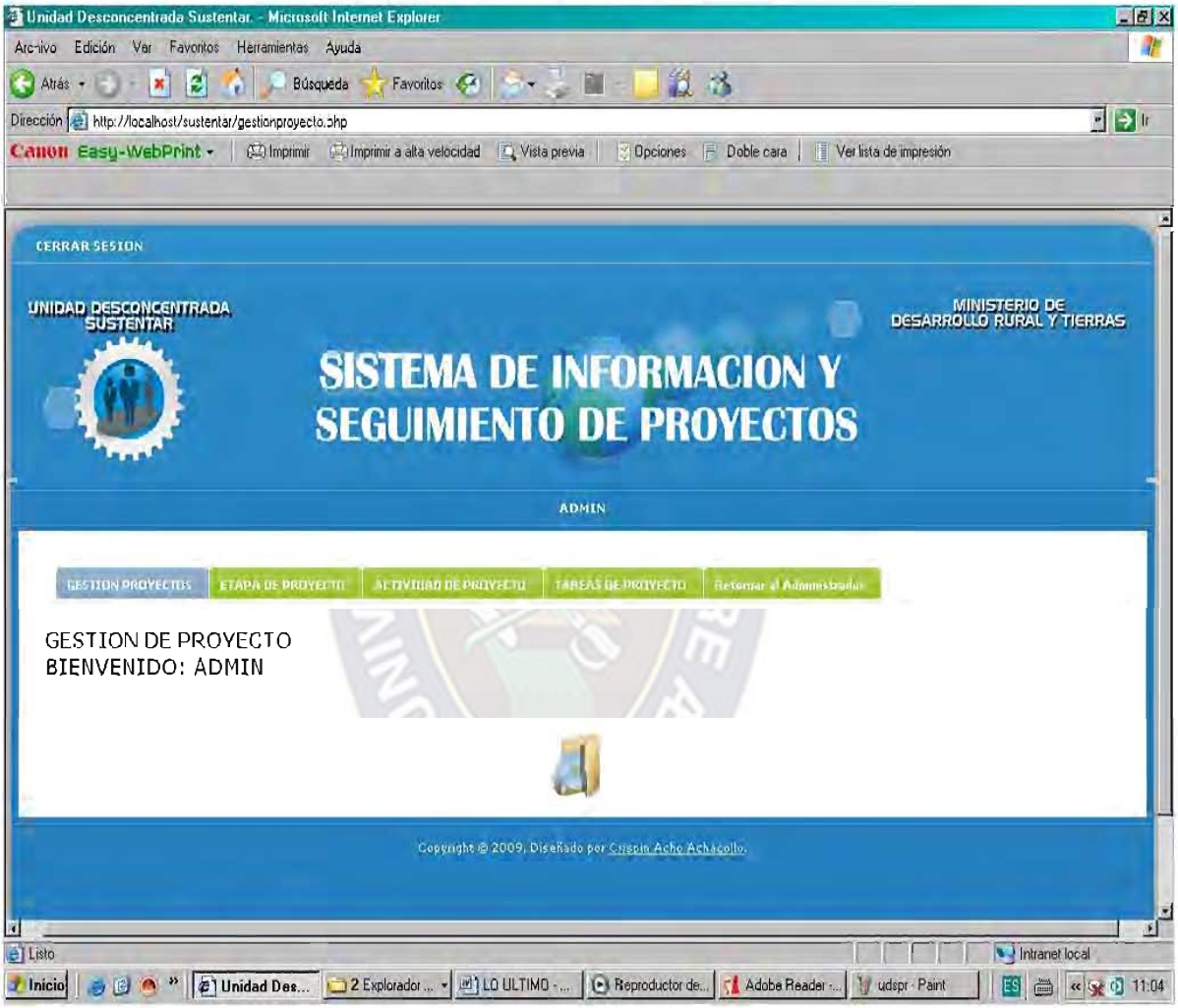
Gráfica 2. Registro de proyectos



Fuente: [Elaboración propia]

Pantalla Principal de Seguimiento de Proyectos: En la siguiente gráfica 3 se muestra la pantalla principal del Encargado de Seguimiento, donde se adscribe el seguimiento correspondiente referente a los diferentes proyectos que se registro anteriormente.

Gráfica 3. Registro de proyectos

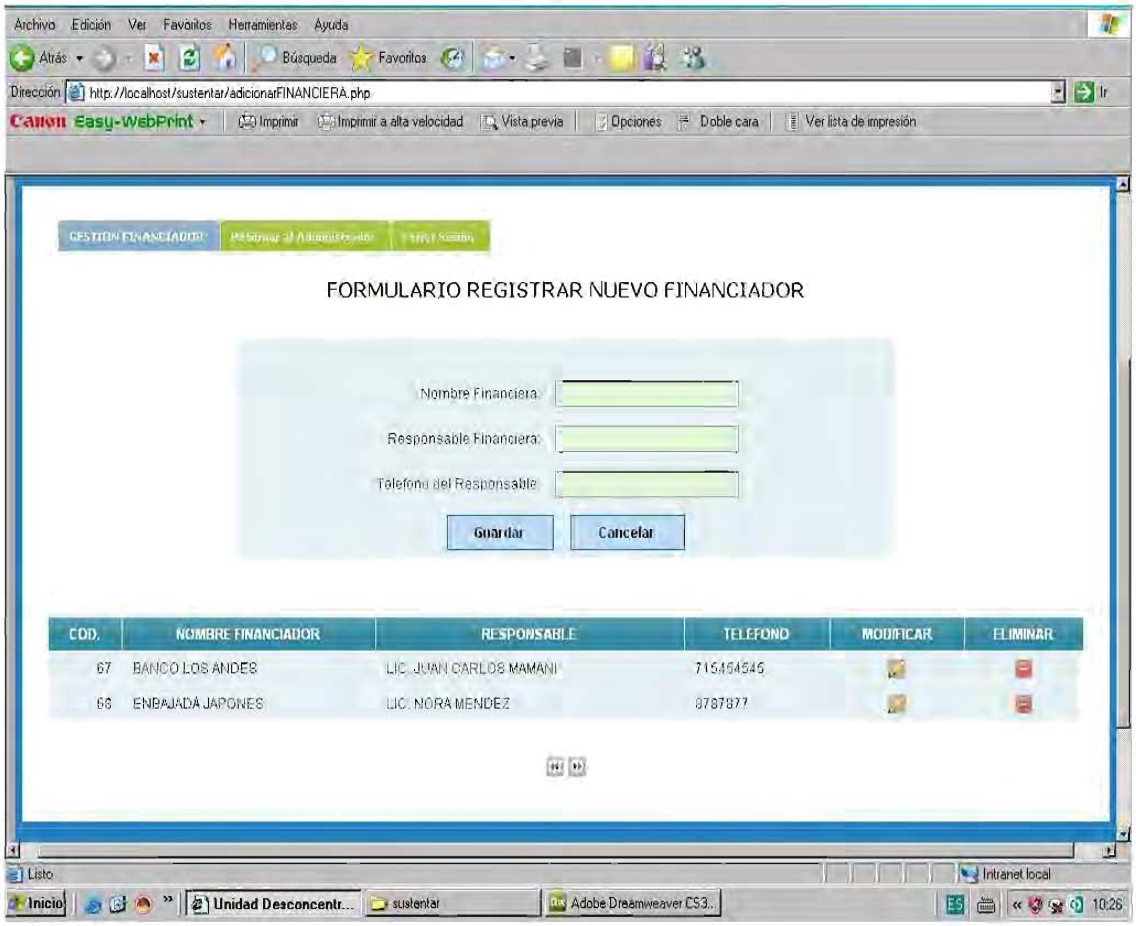


Fuente: [Elaboración propia]

En el presente diseño de interfaz de usuario depende de las necesidades y características que tiene este mismo, y el control que se desea aplicar al sistema.

La siguiente gráfica 4 nos muestra el registro de financiero del proyecto donde el operador debe registrar dependiendo del proyecto que esta financiando.

Gráfica 4. Registro de proyectos



Fuente: [Elaboración propia]

3.16. SEGURIDAD

Las medidas de seguridad tiene el fin de controlar el acceso de los usuarios al sistema mediante la asignación de contraseña a través de los cuales se les asignan privilegios para acceder a cierta parte de la información almacenada.

En caso de que el inicio de sesión no se lleve adelante, no será posible entrar al sistema en su conjunto, ninguna ventana estará disponible.

La seguridad tiene dos conceptos importantes: Seguridad lógica y seguridad física.

3.16.1. Seguridad Lógica

En la que podemos resaltar la seguridad lógica con md5 () este función es propio de php, que utiliza la encriptación porque esta conectado a Internet en cual requiere una seguridad máxima, hoy en día hay muchos hackers para vulnerar la información, para tal efecto cada usuario tendrá su propia contraseña de acceso al sistema.

Uno de los aspectos mas importantes dentro de un sistema es la seguridad de la información, en este sentido la seguridad lógica consiste en la **Aplicación de barreras y procedimientos que resguarden el acceso a los datos y solo permitan acceder a las personas que están registradas por el sistema**.

a) Control de Acceso

La seguridad de acceso que ofrece el sistema esta dada por el uso de contraseñas (Password), estableciendo un medio de control al sistema. En la gráfica 4 se muestra la pantalla de Identificación de usuario.

b) Identificación y Autentificación

Se denomina Identificación al momento que el usuario se da a conocer en el sistema, por ejemplo como mostramos en la gráfica 5 el responsable se da a conocer al sistema introduciendo el Usuario y Contraseña.

La Autentificación se refiere al hecho que el sistema realiza la verificación correspondiente a la identificación anteriormente establecida.

Gráfica 5. Acceso al sistema

Fuente: [Elaboración propia]

c) Roles

Los roles o funciones de los usuarios están dadas de la siguiente manera:

- **Responsable de Proyectos**

Este es el Usuario que establece el control correspondiente a los proyectos que tienen seguimiento, también se encarga de registrar proyectos que tienen licitación.

- **Encargado de Seguimientos**

Este es el usuario que realiza seguimientos sobre los proyectos, registra los seguimientos a los proyectos y elabora un historial correspondiente a seguimiento de proyectos.

3.16.2. Seguridad Física

Que hace referencia al hardware el cual se encuentra bien establecido pues el suelo en el que se encuentra es de madera, la temperatura es favorable, no existen incidencia de las inclemencias del tiempo y cada componente se halla bien distribuido a fin de no ocasionar cortes de energía eléctrica o de producirse estática en los mismos.

Las principales amenazas que se prevén en la seguridad física son:

- Desastres Naturales, incendios accidentales, tormentas e inundaciones.
- Acciones Hostiles.

Puesto que las instalaciones de la Dirección de la Unidad Desconcentrada SUSTENTAR se encuentran instaladas en el piso 4, el riesgo de inundaciones es minino, las instalaciones se encuentran previstas para riesgos que contemplen inundaciones, ya que la infraestructura mantiene fuera de riesgo.

En caso de incendios, la dirección cuenta con extintores instalados, esta prohibido fumar dentro de las oficinas.

A veces solamente es necesario recurrir al sentido común para tomar previsiones en las distintas áreas de la dirección, en cuanto a asegurar las dependencias y oficinas de la dirección para evitar posibles robos o sabotajes de parte de ajenos a la dirección.

3.17. Implantación

Según las especificaciones del diseño, se llevo a cabo la codificación e integración de submodulos del sistema en lenguaje de programación PHP utilizando como administrador la base de datos MySQL.

3.18. Fase de Implementación

El objeto de esta fase es el de asegurar de que el software disponible para los usuarios finales, incluyan pruebas del producto como parte de la preparación para ser entregado y este sometido a los distintos cambios que requieran en la funcionalidad o en caso de aumentar nuevos módulos. Consiste en decidir si los objetivos se cumplieron a cabalidad y si se debe comenzar otro ciclo de vida.

3.19. CALIDAD DE SOFTWARE

La calidad del software es hacer uso de todos los requerimientos, procedimientos, técnicas e instrumentos, para que el producto de software cumpla los estándares predefinidos, durante el ciclo de desarrollo del producto basado en ISO 9126 que tiene los siguientes criterios de calidad: **Funcionalidad, Fiabilidad, Facilidad de Mantenimiento, Portabilidad y Usabilidad**, que han sido descritos con detalle en el capitulo II, juntamente con las formulas que se utiliza.

3.19.1.1. Funcionalidad

Para medir la funcionalidad se tomo en cuenta las métricas *completitud de implementación funcional y adecuación funcional* que nos ayuda a medir cuan completa es la implementación funcional.

3.19.1.2. Completitud de la implementación funcional

Esta métrica se evaluó al término de cada fase de la metodología RUP utilizando las formulas mencionadas comenzamos:

3.19.1.3. Fase de Inicio

En la fase de inicio no se ha implementado ningún caso de uso.

3.19.1.4. Fase de Elaboración

En la fase de diseño se implemento 3 casos de uso de los 10 que existe:

- Registro de Personal.
- Registro de trámite de Proyectos.
- Registro del ingreso de un trámite del Formulario Financiera.

A= 3; B=10

X = 1- A / B (1)

Reemplazando en la fórmula 1 se tiene: X = 1 – 3 / 10 = 0,7
En términos de porcentaje falta el 70% por implementar.

En el flujo de trabajo de diseño del sistema se implementaron 4 casos de usos más:

- Registro del informe validación de datos
- Registro del informe de Regularización

A=7; C=10

Y = 1- A / C (2)

Reemplazando en la formula 2 se tiene: Y = 1 – 7 / 10 = 0,3
En términos de porcentaje falta el 30% por implementar.

3.19.2.1. Fase de Construcción

Por ultimo se implemento todo los casos de uso es decir los 10.

- Consultas

• Reportes

• Conclusión de proyectos

A=10; D=10

Z = 1- A / D (3)

Reemplazando en la formula 3 se tiene: Z = 1 – 10 / 10 = 0.

En términos de porcentaje es 0%, eso quiere decir que ya no falta ningún caso de uso y que se realizo el 100% de los casos de uso descritos en la fase de inicio.

3.19.2.2. Adecuación funcional

Se detectaron problemas en los siguientes casos de uso:

• Generación de consultas de proyectos

• En las consultas por fecha

A=2; B=10

X = 1- A / B (4)

Reemplazando en la formula 4 se tiene: X = 1 – 2 / 10 = 0.8

En términos de porcentaje se tiene un 80% de la adecuación de los casos de uso, estos datos fueron encontrados en las pruebas que se hicieron en la fase de transición. Se logro subsanar todos estos errores, pero para darnos un parámetro; sí en lo posterior existe algún error, se da como inicio los errores hallados es este flujo de trabajo.

Después de subsanar los errores de acuerdo a los requerimientos especificados y adecuado a las actividades de los usuarios, por consiguientes es funcional.

3.19.3. Fiabilidad

La fiabilidad es medida la subcaracteristica de la madurez que tiene 2 métricas que son: *Levantamiento de defectos* y *densidad de defectos* que ayuda a medir los defectos hallados y la proporción de defectos respecto al tamaño del producto (líneas de código).

3.19.3.1. Levantamiento de defectos

Se calcula el número de defectos encontrados y corregidos en la etapa de diseño/ codificación se tiene.

A = 40 defectos corregidos; B = 42 defectos encontrados

Y = A / B (5)

Reemplazando en la formula 5 se tiene: Y = 40 / 42 = 0.95
Por lo tanto se corrigió un 95% de defectos en el diseño/codificación.

3.19.3.2. Densidad de defectos

La densidad de defectos se mide por la siguiente formula:

A = Número de defectos que no fueron corregidos (A - B de la anterior formula)

B = 2261 líneas de código (tamaño del producto)

X = 1 - A / B (6)

Reemplazando en la formula 6 se tiene: X = 1 – (42 - 40) / 2261 = 0.99

Se puede ver que la densidad de defectos es casi nula por tanto existe un 99% de efectividad de corrección de errores.

Nota.- Aclarando que se esta tomando en cuenta los errores que no fueron corregidos en el diseño/codificación, y que en el flujo de trabajo de pruebas fueron corregidos.

Ahora se halla el número de defectos que fueron encontrados en el periodo de prueba.

A = 15 Número de defectos detectados

B = 2261 líneas de código

X = A / B (7)

Reemplazando en la formula 7 se tiene: X = 15/2261 = 0.006

Por tanto el porcentaje de defectos que se ha encontrado en el sistema SIS-PROY. es del 0.6%.

En todo el proceso de pruebas se detectaron todos estos defectos, Luego fueron corregidos en su totalidad.

Tenemos un alto índice de corrección de errores que es 95%, con una efectividad de corrección de errores de un 99% y por ultimo se tiene un error por línea de código de 0.6%

Por los resultados obtenidos, el sistema SIS-PROY. tiene un alto grado de fiabilidad.

3.19.4. Facilidad de mantenimiento

Este factor se ha medido mediante las métricas del índice de madurez del sistema (IMS) que mide la *estabilidad del producto*.

Mt =10, Fc =2, Fa = 0, Fd = 0 número de módulos en la versión anterior que se han cambiando en la versión actual.

IMS = (Mt – (Fa + Fc + Fd)/Mt) (8)

Reemplazando en la formula 8 se tiene: IMS = (10 – (0+2+0)/10) = 0.8

Esta formula se aplico consecutivamente en cada versión del sistema, donde la versión beta (ultima versión) dio como resultado de un 80% de facilidad de mantenimiento.

3.19.5. Portabilidad

Este factor de calidad es calculado mediante la métrica *facilidad de instalación* que básicamente calcula el porcentaje de los usuarios o quien mantiene el software de instalar el software en un ambiente operacional.

3.19.5.1. Facilidad de instalación

A= 4 número de casos en que el usuario es exitoso en la operación de instalación.

B= 5 número total de casos en que el usuario intenta ejecutar la operación de instalación.

X = A / B	**(9)**

Reemplazando en la formula 9 se tiene: X= 4/5 = 0.8

Por tanto existe un 80% de facilidad de instalación del sistema SIS-PROY. por tanto el sistema es portable.

3.19.6. Usabilidad

El factor de calidad de usabilidad en la mayoría de sus métricas existe el subjetivismo, pero se tiene las siguientes métricas que nos servirá a decidir cuan usable es el sistema.

3.19.6.1. Completitud de la descripción

Que indica que proporción de las funciones (casos de uso) se describen en la descripción del producto (documentación, ayuda, etc.)

A= 9 Número de funciones (casos de uso)

B= 10 Número total de funciones (casos de uso)

X = A / B	**(10)**

Reemplazando en la formula (10) se tiene: X = 9/10 = 0.9

Por tanto existe un 90% de entendimiento de la parte de los usuarios con respecto a la capacidad del producto, luego de leer la descripción del producto.

3.16.6.2. Consistencia Operacional en el uso

Cuan consistentes son los componentes de la interfaz de usuario.

A = 1 Función inaceptable

B = 9 Funciones usadas

X = 1 - A / B	**(11)**

Reemplazando en la formula 11 se tiene: X = 1 – 1/ 9 = 0.89

El usuario detecto un 11% del sistema que era inaceptablemente en el periodo de prueba.

A= 0, funciones que el usuario encontró inconsistentes según sus expectativas

UOT Tiempo de operación del usuario (durante el periodo de observación)

Y = A / UOT	**(12)**

Reemplazando en la formula 12 se tiene: Y = 0/24 hrs.

Y= 0

Después de haber resuelto todas las inconsistencias que tenia el sistema, el usuario quedo satisfecho con la consistencia operacional del uso del sistema SIS-PROY.

Por tanto los resultados obtenidos en las métricas del factor de calidad de usabilidad vemos que el sistema es usable en todas sus funciones.

CONCLUSIONES Y RECOMENDACIONES

CAPITULO IV

4.1. CONCLUSIONES

Luego de haber cumplido con los requerimientos y necesidades de Unidad Desconcentrada SUSTENTAR, con los objetivos trazados en el marco referencial, se logró satisfactoriamente la meta trazada, realizando un sistema confiable y de fácil manipulación con el usuario, el cual permite realizar el registro y seguimiento de proyectos, las entidades Financiador y Gestoras como también el registro del profesional externo, facilitando de esta manera contar con información precisa y confiable, para el control y seguimiento de proyectos.

Por lo que se llega a la conclusión de que:

- El sistema de seguimiento de información y control de diferentes proyectos se llegó a concretar.
- Se logró desarrollar e implementar un portal Web para difundir las actividades de la Unidad Desconcentrada SUSTENTAR.
- Se evidencia verificar la factibilidad de proyectos desde portal Web.
- Se modifican los proyectos desde cualquier punto del país (de uso exclusivo del personal de Unidad Desconcentrada SUSTENTAR).
- Genera reportes de proyectos en ejecución, que sea útil para la toma de decisiones.

4.2. RECOMENDACIONES

Luego de haber concluido el presente proyecto de grado es necesario realizar acciones que permitan mantener la madurez del sistema por ello se recomienda:

- Ampliar el sistema con módulos de seguimiento y control de proyectos en las diferentes unidades que tiene MDRyT;
- Incorporar de un subsistema que realice el control de inventarios;
- Llevar el sistema a una plataforma de Linux por ser una entidad pública para ahorrar los costos de la compra de licencia del Sistema Operativo Windows.
- Desarrollar un módulo de contabilidad y fiscalización de los proyectos en desarrollo.

BIBLIOGRAFÍA

[Chaparro, 1993] Chaparro Aguilar David "Tesis de Maestría en Ciencias ITEMS campus Cuermavaca" Redes de Petri (Un Editor Gráfico) Julio 1993.

[Larman, 1999] UML y Patrones, Larman Caring, México 1999 1ra Edición.

[Loayza, 2004] Mario Loayza, "Apuntes de Taller de Sistemas de Información", Bolivia, 2004

[Pressman, 2002] Presman Roger S., "Ingenieria del Software – Un Enfoque Práctico", 5ta. Ed. McGraw Hill, Madrid, 2002

[Senn, 1992] James A Senn, Análisis y Diseño de Sistemas de Información. 1ra Edición, México McGraw Hill.

[Kendall & Kendall, 1997] Kendall Kenneth, Kendall Julie, "Análisis y Diseño de Sistemas", 3ra. Ed. Prentice may Hispanoamericana, 1997

[Schmuller, 1997] Aprendiendo UML en 24 horas, Joseph Schmuller, Ediciones Pretice, España, pp103.

[Elizondo, 2005] Entidad-Relación extendido y modelado de objetos, Elizondo, 2005.

[Acarapi, 2004] Acarapi Cristina, Proyecto de grado Sistema integrado para el seguimiento y control de Prestamos de material bibliográfico para la biblioteca de la carrera de informática, Carrera de Informática, UMSA, Laz - Bolivia

[Mendoza, 2005] Mendoza Lliully Ruth Mery, Proyecto de Grado de la carretera de Informática, con el Título: "Sistema de Información para la Gestión de Proyectos de la Empresa teracorp s.r.l. Tecnología y Consultora", 2005.

[Ibarra, 2007] Ibarra Garcia Jhonny Wilson Proyecto de Grado "Sistema de Información para el control y seguimiento de Proyectos vía Web" caso: Gobierno Municipal de Mecapaca 2007.

[Tompson, 2006] Mónica Thompson: "Conceptos de Proyectos",2006 http//www.promonegocios.net/proyecto/concepto-proyecto.html

www.creangel.com/uml.pdf acceso: 24/11/2009, hrs. 16:30

Anexo A

FUNCIONES:

- Administrar programas y proyectos dentro del área de las competencias del MDRAyMA
- Ejecutar los proyectos de los programas que le sean asignados en el marco de sus competencias
- Realizar el seguimiento y evaluación de los programas y proyectos ejecutados, estableciendo sus resultados.
- Firmar convenios con Sectores Productivos, Prefecturas, Municipios, Universidades, Organizaciones Territoriales y Económicas Campesinas, Indígenas – Originarias y Otras, para la ejecución de programas y proyectos; previa recomendación del Comité de Seguimiento y Revisión, y autorización del Viceministro del cual dependa la Unidad, misma que deberá ser aprobada por el Ministro(a) de DRAyMA.
- Adquirir bienes y contratar servicios que contribuyan a cumplir con los objetivos de los programas y proyectos.
- Utilizar mecanismos de administración financiera que convenga al desarrollo de los programas y proyectos, entre los que están las gestiones para obtener financiamiento de organismos nacionales e internacionales y estructurar sistemas de financiamiento diversos, buscando la complementariedad y concurrencia financiera y técnica entre los programas y proyectos entre los actores vinculados a niveles local, regional, departamental, nacional e internacional.
- Remitir información periódica de los avances y el desarrollo de los programas y proyectos a la Unidad de Planificación del MDRAyMA
- Otras que sean necesarias para el cumplimiento de los objetivos de los programas y proyectos, previa autorización de la MAE del MDRAyMA mediante Resolución Expresa.

SERVICIOS QUE PRESTA SUSTENTAR:

a) **Financia proyectos de pre-inversión**
La UD SUSTENTAR canaliza recursos de los financiadores para proyectos de pre-inversión, solicitados por Comunidades Campesinas, Indígenas-Originarias, Organizaciones Económico-Productivas, Organizaciones

Campesinas, Organizaciones Indígenas-Originarias, Instituciones Públicas, Gobiernos Municipales, Mancomunidades de Municipios, Prefecturas, Universidades e Instituciones de Investigación.

b) **Financia programas y proyectos de inversión**

La UD SUSTENTAR canaliza recursos de los financiadores a programas y proyectos de inversión, solicitados por Comunidades Campesinas, Indígenas-Originarias, Organizaciones Económico-Productivas, Organizaciones Campesinas, Organizaciones Indígenas-Originarias, Instituciones Públicas, Gobiernos Municipales, Mancomunidades de Municipios, Prefecturas, Universidades e Instituciones de Investigación.

c) **Transfiere conocimientos y experiencias calificadas**

Con el fin de generar, complementar y fortalecer las capacidades técnicas y financieras en la preparación y ejecución de proyectos, la UD SUSTENTAR transfiere conocimientos y experiencias calificadas Comunidades Campesinas, Indígenas-Originarias, Organizaciones Económico-Productivas, Organizaciones Campesinas, Organizaciones Indígenas-Originarias, Instituciones Públicas, Gobiernos Municipales, Mancomunidades de Municipios, Prefecturas, Universidades e Instituciones de Investigación, Organizaciones Internacionales y sin fines de lucro.

d) **Gestiona recursos financieros y no financieros para programas y proyectos**

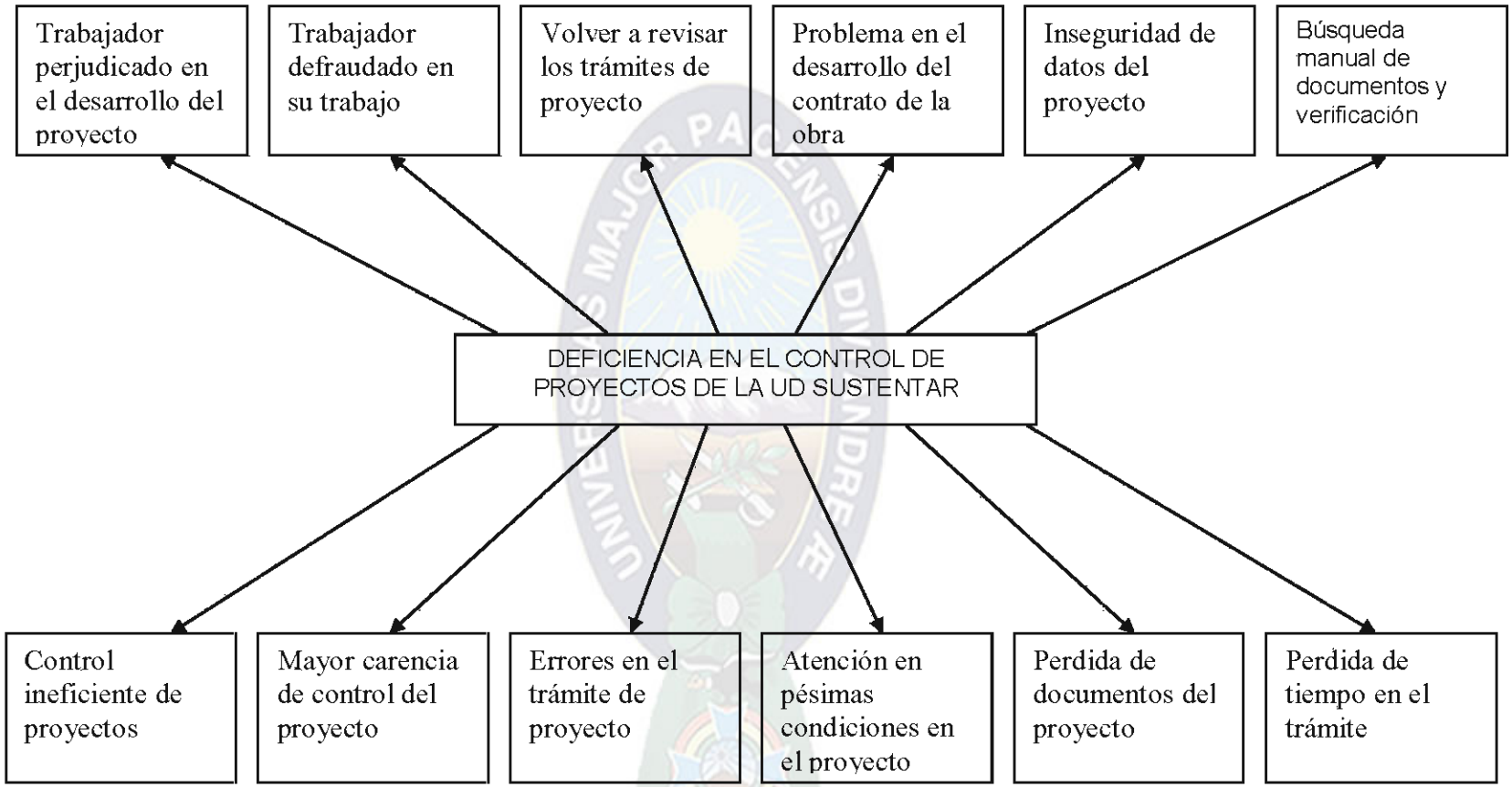
La UD SUSTENTAR toma contacto con organismos internacionales, y de financiamiento de manera directa o a través de convenios suscritos con otras organizaciones, instituciones y otros organismos para gestionar recursos financieros y no financieros.

e) **Estructura sistemas de financiamiento**

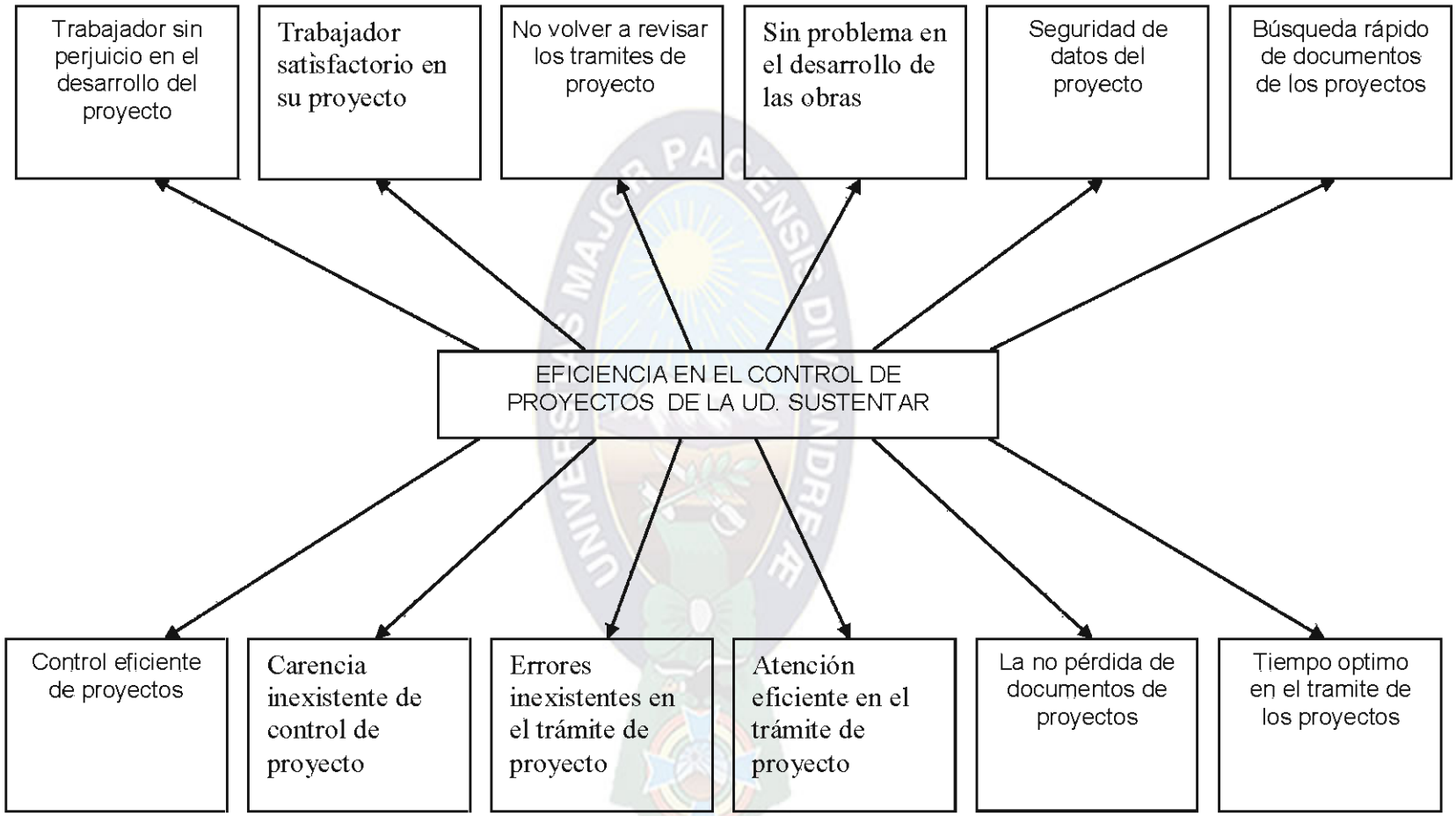
La UD SUSTENTAR diseña Sistemas de Financiamiento diversos para buscar complementariedad y concurrencia financiera y técnica entre los programas y proyectos administrados por la UD SUSTENTAR u otros actores vinculados, con la aplicación de los mismos a nivel local, regional, departamental, nacional e internacional.

ANEXO B

ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVOS



ANEXO C

CORESPONDENCIA DE LOS DIAGRAMAS DE CLASES A LOS DIAGRAMAS ENTIDAD/RELACIÓN

El modelo E/R es un modelo que ayuda a desarrollar aplicaciones "tradicionales" como ser: procesamiento de los datos en un negocio o industria. Actualmente hay base de datos con requisitos más complejos como multimedia, telecomunicaciones, minería de datos, etc. para tales casos la Base de datos se utiliza con mayor frecuencia el modelo orientado de a objetos con notación UML.

UML (Lenguaje de Modelado Unificado) se está haciendo cada vez más presente en ingeniería del software. UML cubre el diseño de base de datos y la especificación detallada de aplicaciones. Para ello emplea varios tipos de diagramas (casos de uso, de clases de secuencia, etc.).

UML usa diagramas de clases para el diseño de base de datos, además de atributos y relaciones, también se incluyen operaciones sobre objetos los cuales pueden servir para especificar los requisitos funcionales del diseño de la base de datos.

El mapeo objeto-relacional es una técnica de programación para hacer corresponder datos entre el modelo orientado a objetos al modelo de datos relacional, los conceptos de clase e instancia sigue las siguientes definiciones:

En E/R un tipo de entidad representa dos cosas:

- La intensión, el nombre del tipo de entidad y la lista de sus atributos,
- La extensión o conjunto de entidades que forman parte del tipo de entidad,
- Considerando lo anterior, una clase es un concepto equivalente al tipo entidad,
- Las entidades de una clase se llaman objetos o instancias de esa clase.
- Cada objeto o instancia consta de un identificador predefinido que lo identifica. Dicho identificador del objeto no es ninguno de sus atributos.
- Los objetos o instancias de una clase cuentan con un conjunto de operaciones que se pueden realizar sobre ellos.
- Dichas operaciones pueden servir para especificar requisitos funcionales durante el diseño de la base de datos.

Berza nos ofrece algunas reglas para el mapeo objeto relacional:

Figura 1.1: Representación del Modelo UML

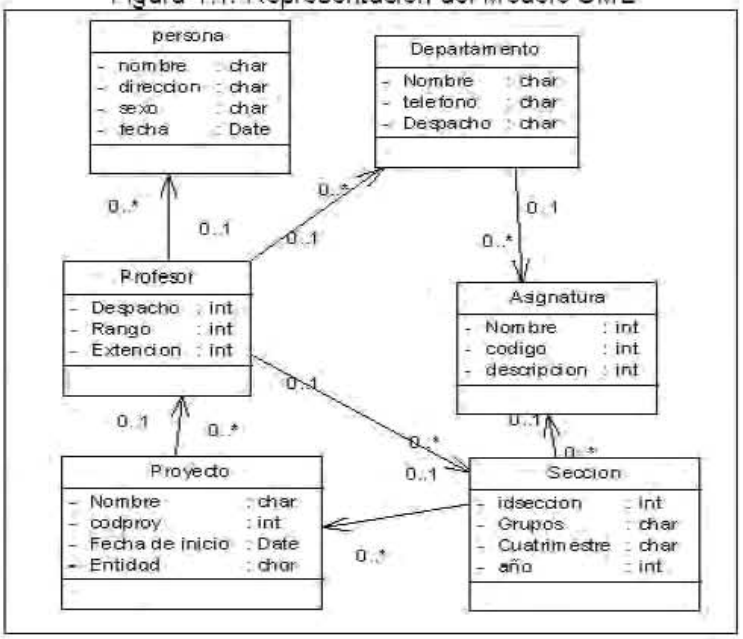
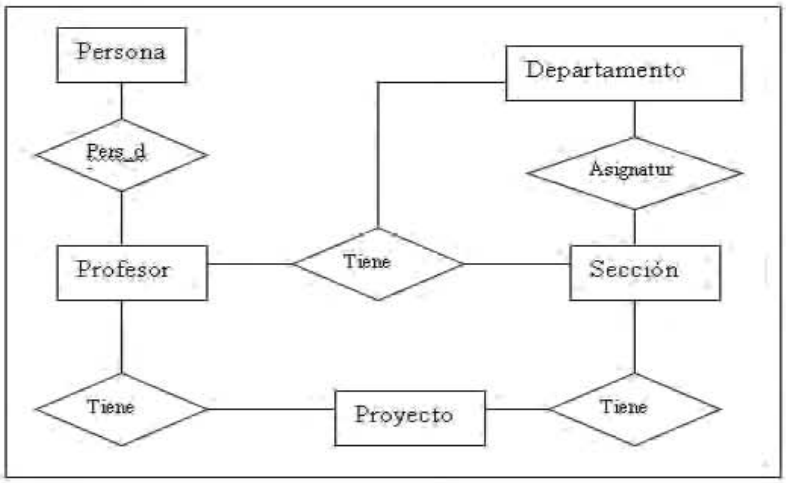


Figura 1.2 Representación del Modelo Entidad Relación

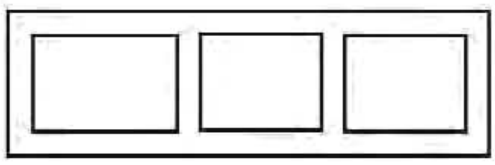


Reglas de Correspondencia del Modelo UML al E/R

Tipo de entidad

Grupo de objetos que tienen las mismas propiedades y que en la organización para la que va servir la BD tiene una existencia independiente, bien sea física o abstracta.

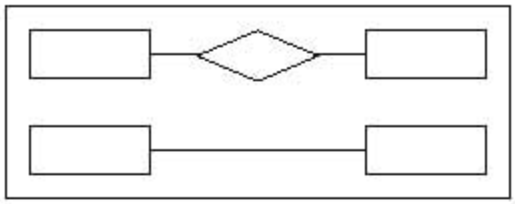
Notación



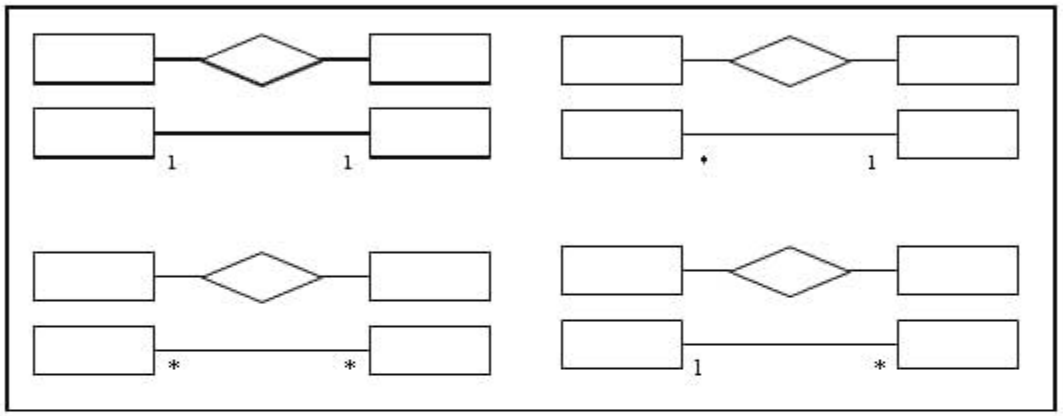
Tipo de relación

Asociación que establece entre tipos de entidad para representar un conjunto de relaciones que establecen entre las ocurrencias de esos tipos de entidad

Notación



Representación de la cardinalidad máxima de una relación



Claves de una relación

Las claves nos permiten diferenciar entre sí las distintas entidades que podríamos aplicar de la siguiente forma a las relaciones:

Las claves de las relaciones vienen definidas por las claves de las entidades relacionadas:

- Relaciones de muchos a muchos (N:M): La clave primaria será la unión de las claves primarias de las entidades participantes en la relación.
- Relaciones uno a muchos (1:N): La clave primaria de la entidad que interviene en la relación con paridad N.
- Relaciones uno a uno (1:1): Las claves primarias de las entidades participantes son claves candidatas de la relación entre entidades.
-

Entidades fuertes y entidades débiles

Un tipo de entidad es fuerte si la existencia de sus ocurrencias no depende de ningún otro tipo. En caso contrario, se dice que el tipo de entidad es débil.

DIAGRAMAS DE CASO DE USO

Diagrama de casos de uso para el paquete Coordinador

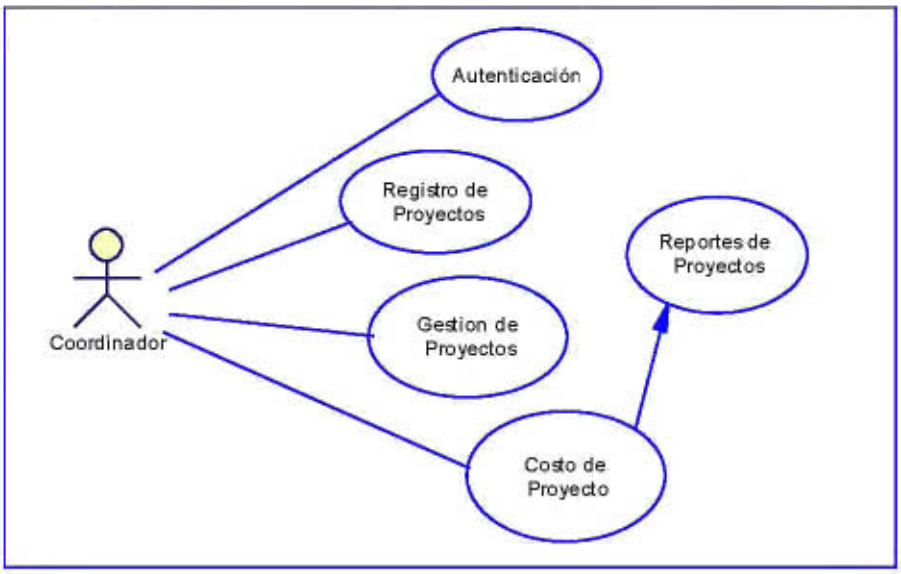
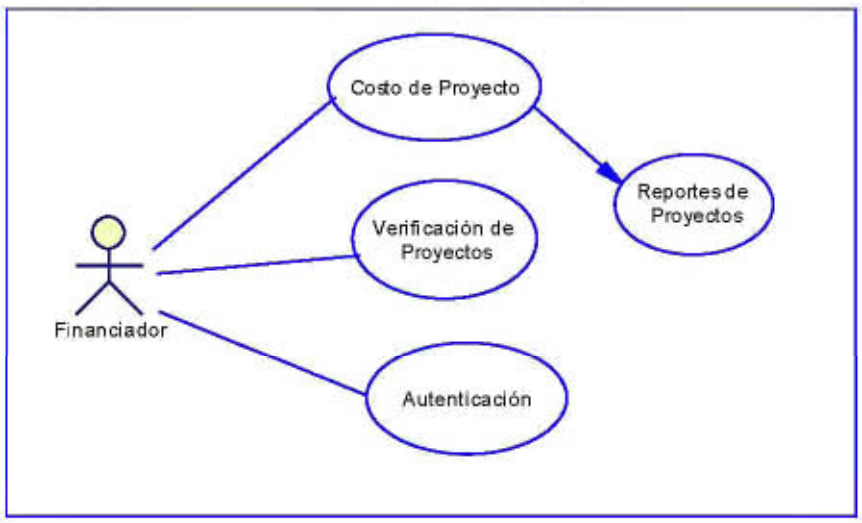



Diagrama de casos de uso para el paquete Financiador