

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



PROYECTO DE GRADO

Sistema de control de activos fijos S.I.C.A.F.
Caso: Seguro Social Universitario (S.S.U.)

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA
MENCION INGENIERIA DE SISTEMAS

POSTULANTE : LEONARDO JESÚS CÁCERES SARAVIA
TUTOR : M. Sc. FRANZ CUEVAS QUIROZ
REVISOR : LIC. CARMEN ROSA HUANCA QUISBERT

LA PAZ - BOLIVIA
2009

*A mi familia: Clemente Cáceres
Guarachi, Ana Saravia de Cáceres,
Andrea y Cecilia Cáceres,
especialmente a mi abuela Antonia
Guarachi.*

*Por todo su apoyo, comprensión y
amor.*

Agradecimientos

- *A Dios principalmente por permitirme vivir*
- *Al M. Sc Franz Cuevas Quiroz y a la Lic. Carmen Rosa Huanca Quisbert a quienes agradezco el apoyo y la guía y los consejos brindados por la realización y culminación exitosa del presente trabajo*
- *Al Ing. Guillermo Salinas R. por la colaboración brindada en esta última etapa .*
- *A los docentes de la carrera de informática por compartir sus conocimientos en las aulas de la universidad.*
- *A todos los amigos compañeros por las experiencias compartidas.*
- *A Lizeth Carmen por toda su ayuda y colaboración incondicional.*
- *A mi hermanita Andrea por su apoyo y cariño.*
- *Por ultimo agradecer el amor cariño y apoyo incondicional brindados por parte de mis padres :*

Clemente Cáceres G. y Ana J. Saravia .P

RESUMEN

El presente proyecto titulado "Sistema De Control de Activos Fijos" (SICAF) surge por la necesidad de recopilar, almacenar, procesar y difundir la información relacionada con el manejo de activos fijos del Seguro Social Universitario y tiene por objetivo principal: Desarrollar e implementar un sistema de información que permita el seguimiento y control de activos fijos reduciendo el tiempo en el proceso de manejo de información de activos fijos del S.S.U.

Los activos fijos se denominan como los "activos que producen utilidades", ya que generalmente son estos los que dan base a la capacidad de la empresa para generar utilidades.

Sin planta y equipo la empresa no podría realizar su tarea diaria, ni elaborar los productos que le produce ingresos.

Administración de activos fijos:

Como las inversiones en activos fijos representan desembolsos importantes de efectivo hecho por las firmas manufactureras, estas deben prestarle gran atención a las decisiones que se tomen con respecto al valor de compra del activo y a las posibles erogaciones futuras que se deba hacer para su instalación, mantenimiento, operación etc.

Como se sabe los activos fijos tienen una vida útil mayor a un año, es por ello que estos pueden representar compromisos financieros de largo plazo para la empresa.

A medida que el tiempo pasa los activos fijos se van volviendo obsoletos, por ello según su vida útil deben depreciarse, para recuperar el valor de estos en el futuro y ajustándolos por inflación, a continuación se presenta un ejemplo de esto.

INDICE

CAPITULO I

1.1.	Introducción	1
1.2.	Antecedentes	1
1.3.	Planteamiento Del Problema De Investigación	4
1.3.1.	Planteamiento Del Problema	4
1.3.2.	Formulación Del Problema	5
1.4.	Objetivos De Investigación	5
1.4.1.	Objetivo General	5
1.4.2.	Objetivos Específicos	5
1.5.	Justificación De La Investigación	6
1.6.	Técnicas, Metodología Y Herramientas	6
1.7.	Lenguaje De Modelado Uml	9
1.8.	Alcances	10

CAPITULO II

2.1	<i>Activos Fijos</i>	11
2.1.1	Clasificación	11
2.1.2	Depreciación	12
2.2	Ingeniería Del Software Orientada A Objetos	12
2.3	Programación Orientada A Objetos	13
2.3.1	Estructura De Un Objeto	14
2.3.2	Beneficios Que Se Obtienen Del Desarrollo Con Poo	14
2.3.3	Análisis Orientado A Objetos	15
2.3.4	Diseño Orientado A Objetos (Doo)	16
2.4	Lenguaje Unificado De Modelado	17
2.4.1	Diagramas Uml	20
2.5	Proceso De Desarrollo	23
2.6	Definición De Actividades	24
2.6.1	Fase De Concepción E Inicio	26
2.6.2	Fase De Elaboración	29
2.6.3	Fase De Construcción	31
2.6.4	Fase De Transición	33
2.7	Diseño De La Base De Datos Relacional – Extensión Informal Del Uml	33

2.7.1	Un Análisis Acerca De La Transformación	36
2.8	Calidad	42
2.8.1	Métricas De Calidad	42
2.8.2	Factores de calidad ISO-9126	43
2.9	<i>Arquitectura Del Sistema</i>	43

CAPITULO III

3.1	<i>Plan De Desarrollo De Software</i>	45
3.2	<i>Fase De Inicio</i>	46
3.2.1	<i>Modelado Del Negocio</i>	46
3.2.2	Diagrama De Actividades	50
3.3	Modelado De Requisitos	52
3.3.1	Requisitos Funcionales	52
3.4	Requisitos No Funcionales	53
3.5	Glosario De Términos	54
3.6	Modelo Conceptual	56
3.7	Fase De Elaboración	57
3.7.1	Modelo De Casos De Uso	57
3.7.1.1	Casos De Uso Esenciales	57
3.7.1.2	Diagramas De Casos De Uso Expandidos	58
3.8	Modelo De Diseño	64
3.8.1	Diagramas De Secuencia	64
3.9	Contratos	67
3.10	Diagramas De Estado	69
3.11	Diagramas De Clase	71
3.12	Desarrollo De La Arquitectura Del Sistema	72
3.13	Transformación Del Modelo OO Al Modelo Er	73
3.14	Modelo Relacional	74
3.15	Esquemas De Base De Datos	75
3.16	Modelo De Despliegue Del Sistema	77
3.16.1	Diagrama De Componentes	77
3.16.2	Diagrama De Despliegue	79
3.17	Fase De Construcción	80
3.17.1	Implementación De Interfaces	80

CAPITULO IV

4.1	Facilidad del mantenimiento	83
4.2	Portabilidad	83
4.3	Usabilidad	84

CAPITULO V

5.1	Conclusiones	85
5.2	Recomendaciones	85
	Bibliografía	86

ANEXOS:

<i>ANEXO A: MÉTODOS DE DEPRECIACIÓN</i>	89
<i>ANEXO B: CASO DE USO EXPANDIDO</i>	92
<i>ANEXO C: DIAGRAMAS DE SECUENCIA</i>	97
<i>ANEXO D: CONTRATOS</i>	100
<i>ANEXO E: ESQUEMAS DE LA BASE DE DATOS</i>	120



INDICE DE TABLAS

Tabla 2.1 conceptos clave del UP	26
Tabla 2.2 Descripción de los hitos de las fases	27
Tabla2.3 disciplinas de la fase de inicio	28
Tabla2.4 Artefactos de la fase de Inicio	29
Tabla 2.5 Categoría de las funciones del sistema	29
Tabla2.6 disciplinas de la fase de elaboración	30
Tabla2.7 Disciplinas de la fase de elaboración	31
Tabla2.8 elementos del glosario	31
Tabla2.9 Disciplinas de la fase de Construcción	32
Tabla2.10 Artefactos de la fase de construcción	33
Tabla 3.1 tiempos estimados en las etapas del UP	45
Tabla.3.2 Definición de requisitos funcionales del sistema	53
Tabla: 3.3 glosario de términos	56
Tabla: 3.4 caso de uso Asignación de Activos Fijos	59
Tabla: 3.5 caso de uso Registrar Activo Fijo	60
Tabla: 3.6 caso de uso Registrar proveedor	61
Tabla: 3.7 caso de uso Registrar baja de Activos Fijos	62
Tabla: 3.8 caso de uso Devolución De Activos Fijos	63
Tabla: 3.9 Contrato Encargado de A.F. selecciona el funcionario	67
Tabla: 3.10 Contrato Se despliegan los datos del empleado	67
Tabla: 3.11 Contrato Selección de activos disponibles	68
Tabla: 3.12 Contrato Registro de la asignación	68
Tabla: 3.13 Contrato Se muestra un reporte de la nueva asignación	69
Tabla.3.14 Definición de la tabla de Activos Fijos	75
Tabla.3.15 Definición de la tabla de Asignaciones	75
Tabla.3.16 Definición de la tabla Baja de Activos Fijos	76
Tabla.3.17Definición de la tabla Cuenta de Activos Fijos	76
Tabla.3.18 Definición de la tabla Devoluciones de Activos Fijos	76

INDICE DE FIGURAS

Fig.1.1 Términos orientados a la planificación del UP	7
Fig1.2 Disciplinas del UP	8
Fig.2.1 Símbolo para expresar la generalización	19

Fig.2.2 Símbolo para expresar la asociación	19
FIG.2.3 Símbolo para expresar la realización	19
Fig.2.4 Símbolo para expresar la dependencia	20
Fig. 2.5 Diagrama de casos de uso. [JOS2005]	22
Fig2.6 Diagrama de clases. [JOS2005]	22
Fig. 2.7Diagrama de secuencia. [JOS2005]	23
Fig. 2.8 Arquitectura del UP. [LARMAN 2003]	24
Figura2.9 Notación UML para modelar E-R	36
Figura 2.10: arquitectura cliente servidor	44
Fig. 3.1 Modelos de casos de uso del negocio	49
Fig.3.2 Diagrama de actividades “Adquisición de bienes”	50
Fig.3.3 Diagrama de actividades “Solicitud de Bienes”	51
Fig.3.4 Diagrama de actividades “Ingreso de Bienes”	51
Fig.3.5 Modelo Conceptual asociado al seguimiento y control de Activos Fijos	56
Fig. 3.6 Diagrama de Casos de Uso para el sistema de activos fijos	58
Fig.3.7 Diagrama de Secuencia Asignación de Activos Fijos	64
Fig.3.8 Diagrama de Secuencia Registrar activo fijo	65
Fig.3.9Diagrama de Secuencia Registrar proveedor	65
Fig.3.10 Diagrama de Secuencia Registrar baja de activos fijos	66
Fig.3.11 Diagrama de Secuencia Devolución de activos fijos	66
Fig.3.12 Diagrama de Estado Asignación de Activos fijos	69
Fig.3.13 Diagrama de Estado Asignación de Activos fijos	70
Fig.3.14 Diagrama de Estado Asignación de Activos fijos	70
Fig. .3.15 Diagrama de clases del sistema SICAF	71
FIG.3.16 Intranet Del Seguro Social Universitario	72
Fig. .3.17 Diagrama Entidad Relación del sistema SICAF	73
Fig. .3.18 Modelo Relacional del sistema SICAF	74
Fig.3.19 Diagrama de componentes sistema SICAF	77
Fig.3.20 Diagrama de componentes sistema SICAF	78
Fig.3.21 Diagrama de despliegue sistema SICAF	79
Fig.3.22 Interfaz Activos Fijos	80
Fig.3.23 Interfaz Funcionarios	81
Fig.3.24 Interfaz Proveedores	81
Fig.3.25 Interfaz Modificar Empleado	82
Fig.3.26 Interfaz Nuevo Empleado	82

1.1. INTRODUCCION

Los cambios permanentes que ocurren en nuestra sociedad son causa de constantes necesidades que muchas veces son difíciles de satisfacerse con recursos disponibles. Es así, que muchas organizaciones a menudo recurren a herramientas de alta tecnología, que les permita satisfacer esas demandas, lo que les permite seguir vigentes, en un mercado donde es competente aquel que cuenta con recurso de tecnología avanzada, entre los que la ciencia informática juega un papel preponderante.

El Sistema de Información y Control de Activos Fijos (S.I.C.A.F), denominado así este proyecto, tiene el objetivo de administrar y controlar en forma eficiente los activos fijos del Seguro Social Universitario; aplicando tecnologías y modelos científicos para su diseño, además de utilizar un lenguaje de codificación y plataforma de software de alto nivel para su implementación. En función a ello, el sistema nos permite obtener información lo que garantiza su utilidad, confiabilidad y precisión, lo que permitirá realizar en forma oportuna cualquier toma de decisiones.

De esta manera, este trabajo integra en una sola vía la teoría con la práctica, estableciendo los lineamientos que nos permitan aplicar el conocimiento adquirido para resolver los diferentes problemas de la Unidad de activos fijos del SSU.

Por tanto el presente proyecto estudiara el Subsistema de manejo de bienes (Módulo de activos fijos) para el S.S.U. en el cual aprovechando al máximo sus recursos tecnológicos con los que cuenta y así dar solución a los problemas que se suscitan en la unidad de activos fijos, coadyuvando a la toma de decisiones y por sobre todo dar cumplimiento a los objetivos trazados por la institución.

1.2. ANTECEDENTES

La informática se ha utilizado, principalmente en los niveles Secundarios y Terciario de la atención medica, es decir en los hospitales, una de las razones es que el manejo de la información constituye el 25 a 30 por ciento del costo de las actividades.

Los hospitales se han beneficiado de instrumentos de informática versátiles que permiten atender tanto a múltiples demandas de las necesidades de índole administrativa y medica.

Los sistemas de información para hospitales tienen dos componentes funcionales:

- Procedimiento de información dentro de un departamento.
- Comunicación entre departamentos.

Insalud por su parte en 1984 publica un modelo de gestión hospitalaria en el que propone la dirección por objetos y puntualiza que resulta imprescindible instrumentar una serie de medidas, componentes definitorios de este modelo. Entre ellas la creación de una infraestructura de la información mínima y homologa entre las instituciones de salud.

Desde entonces no resulta tarea fácil acometer la transformación con cierto grado de garantía, pues obliga a realizar un gran esfuerzo en la mejora de la información lo cual imposibilita la evaluación y análisis de la actividad, costes incurridos y resultados, etc.

El objetivo básico de introducir la informática en los hospitales es la de mejorar el tiempo de procedimiento, la calidad de la información de que disponen los directores, gerentes o personas encargadas de la institución. Estas personas deben decidir constantemente que hacer (incertidumbre en la acción), como hacerlo (incertidumbre en el método), cuando (incertidumbre en los resultados). Con información confiable y oportuna se puede reducir estas incertidumbres, como afecto inmediato brindar un buen servicio a los pacientes y demás el uso eficiente de los recursos de la institución.

“En nuestro medio existen dos hospitales que cuentan con un sistema de información Administrativa, el Hospital de la Fundación San Gabriel y el Hospital de la Caja Petrolera, en los cuales no se da relevancia al control de los Activos Fijos”.

En el SSU actualmente se encuentran en desarrollo sistemas que cubrirán toda la gestión hospitalaria, que se encuentra realizando la empresa SOLUCORP hasta el momento tienen dos sistemas en funcionamiento que son.

- “Sistema de afiliación del SSU”. Software cuyo objetivo principal es de afiliar a los asegurados para que después de esto ellos puedan recibir atención médica como asegurados del S.S.U.

- “Sistema de fichaje SSU” un software que utiliza un lector de tarjetas magnéticas y una pantalla touch screen en la que los asegurados toman una ficha para ser atendidos luego por el encargado de fichaje.

Los proyectos en desarrollo por la empresa SOLUCORP son:

- *consulta externa.*
- *Historia clínica.*
- *Laboratorio clínico.*
- *Hospitalización.*
- *Imageneología.*
- *Emergencia.*
- *Enfermería.*
- *Administración del sistema (los nueve subsistemas anteriores mencionados)*

Algunos proyectos relacionados al seguimiento y control de activo fijos en la Carrera de Informática son:

- ✓ “Sistema de Registro y Asignación de Activos Fijos para la SPVS”, elaborado por Angelica Apaza Cocarico (2004), haciendo uso de la metodología de desarrollo de software RUP, para la parte de análisis, diseño, implementación y diagramas hace uso de la herramienta case Enterprise Architect y como lenguaje de modelado UML.
- ✓ “Sistema de Control de activos Fijos”, realizado por Juana Cano Huanaco (2005) empleando la metodología RUP en el análisis y el diseño del sistema poniendo énfasis en que el sistema brinde rapidez y eficiencia en los procesos.
- ✓ “Control y Seguimiento de Activos fijos del Servicio Exterior Vía Web-Ministerio de Relaciones y Cultos”, realizado por Mario Canchillo Zanga (2006) el objetivo de tal tiene dar control vía web, empleando la metodología Orientada a Objetos de Larman.
- ✓ “Sistema de Control de Activos Fijos Para El gobierno Municipal de El Alto”, realizado por Epifanio Mamani Calle (2007) hecho usando la metodología RUP, empleando UML en el modelado Orientado a objetos.

1.3. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACION

1.3.1. PLANTEAMIENTO DEL PROBLEMA

A través de la propia observación de la realidad, mediante referencias, encuestas e informes de los responsables y directivos del hospital, se concluye la sobrecarga de tareas monótonas y de los medios necesarios, la escasa o tímida innovación de procedimientos y tecnologías.

El Control de Activos Fijos, es una tarea extremadamente vulnerable cuando no se controla. Cuando no se contabilizan en forma correcta, pueden llegar a generar consecuencias impositivas para el S.S.U.

No se cuenta con un sistema de información automatizado el cual brinde información significativa debido a que fundamentalmente la información está sujeta a mecanismos manuales los cuales dan lugar a procedimientos repetitivos, lentos y faltos de oportunidad.

Con lo cual la dirección, el administrador y personas involucradas de las diferentes unidades del hospital reciben y emiten información general o específica y en muchos de los casos no oportunas lo cual le quita significación para una apropiada toma de decisiones.

Por esta razón se considera necesaria la aplicación de un sistema de control de Activos Fijos que a través de herramientas coherentes y adecuadas apoyen la labor Administrativa a fin que se pueda desarrollar sus actividades de mejor manera y en un menor tiempo.

A continuación detallaremos los problemas detectados:

- No se tiene un control adecuado de los activos Fijos.
- El proceso de Depreciación y Actualización de Activos Fijos es manual y lento.
- La codificación del Activo Fijo es manual.
- Cuando se quiere saber la ubicación física de un activo específico se tiene que revisar mucha documentación o buscar en los distintos ambientes del hospital lo cual demanda pérdida de tiempo.
- Cuando se requiere información acerca de activos fijos para controlar se tiene que revisar inmensas de documentación por gestiones.

- Al realizar el Inventario Físico (conteo físico) no se tiene un previo listado actualizado de los activos fijos por ambiente, por ítem, por lo tanto cuando se realiza el control se procede a registrar todo cada vez que realice este trabajo.

1.3.2. Formulación del problema

¿El sistema de control de activos fijos reducirá el tiempo de procesamiento de la información de la Depreciación, actualización y transferencia de los datos de activos fijos del Hospital del SEGURO SOCIAL UNIVERSITARIO (S.S.U.)?

1.4. OBJETIVOS DE INVESTIGACION

1.4.1. OBJETIVO GENERAL

Diseñar e implementar un sistema de información para el control de los Activos Fijos para el Seguro Social Universitario

1.4.2. OBJETIVOS ESPECIFICOS

- Registrar toda la información de los Activos fijos del SSU.
- Almacenar los detalles de los movimientos de cada activo fijo dentro del SSU.
- Realizar la Bitácora de cada Activo Fijo, desde su ingreso, hasta que este ya no sea utilizado por el personal del SSU.
- Realizar la codificación automática de los Activos Fijos.
- Facilitar el control de inventario físico a través de la emisión de listados de Activos Físicos codificados y clasificados de acuerdo a una ordenación lógica y de fácil consulta.
- Mejorar la calidad de información mediante la obtención de informes confiables y oportunos con el SICAF.
- Plantear la codificación de los ambientes para la rápida ubicación de los Activo Físico.

1.5. JUSTIFICACIÓN DE LA INVESTIGACION

La incorporación de un Sistema de Control de Activos Fijos será una herramienta útil de dirección administrativa la misma que permitirá al nivel directivo estar informado.

Con la implementación de un sistema, la unidad de activos fijos y el S.S.U misma a través de sus funcionarios serán beneficiados en sentido de que la información que manejen será más confiable y oportuna. Además el hecho de contar con un sistema será capaz de satisfacer a todas las necesidades de información, motivara al personal de la unidad a prestar mayor interés.

Con el sistema de control de activos fijos los usuarios resultan beneficiados, mejoran también los procesos que se realizan proporcionando una nueva herramienta a los usuarios y preservando la información en un medio que a diferencia de un medio físico no es degradable.

Los beneficios que el sistema brinda a la institución son:

- Preserva la documentación importante que se maneja.
- Los servicios que brinda son de mejor calidad y de manera eficaz.
- El tiempo de búsqueda se reduce.
- Se tiene un mejor control de la información.

1.6. TECNICAS, METODOLOGIA Y HERRAMIENTAS

Para el desarrollo del presente proyecto utilizaremos el modelo de procesos denominado "Proceso Unificado" o UP.

El proceso Unificado cuenta con las siguientes características:

- Iterativo
- Incremental

- Flexibilidad para el cambio
- Retro alimentación temprana

Ventajas del Proceso Unificado:

- Prevención de riesgos
- Comprobación de la calidad continua
- Orientado a casos de Uso
- Modelización visual
- Gestión de Requisitos
- Gestión de Configuración

Este modelo consta de las siguientes fases:

- ❖ **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.
- ❖ **Elaboración**, en esta etapa el objetivo es determinar la arquitectura optima.
- ❖ **Construcción**. En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ❖ **Transición**. El objetivo es llegar a obtener el realce del proyecto.

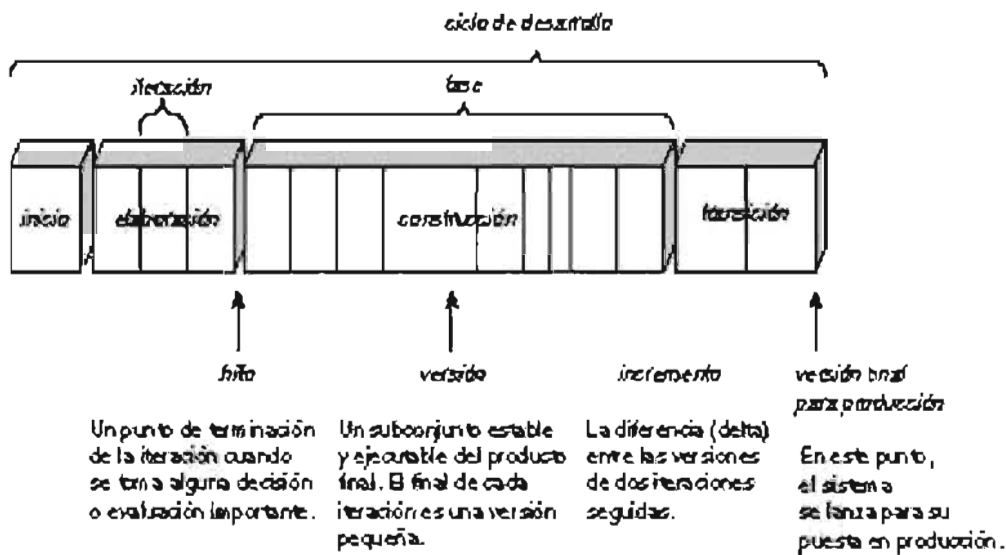


Fig.1.1 Términos orientados a la planificación del UP
Fuente Craig Larman 2003

El Proceso Unificado cuenta también con las siguientes disciplinas:

- Modelado de Negocio.
- Requisitos.
- Diseño.
- Implementación.
- Prueba
- Despliegue
- Gestión de Configuración
- Entorno

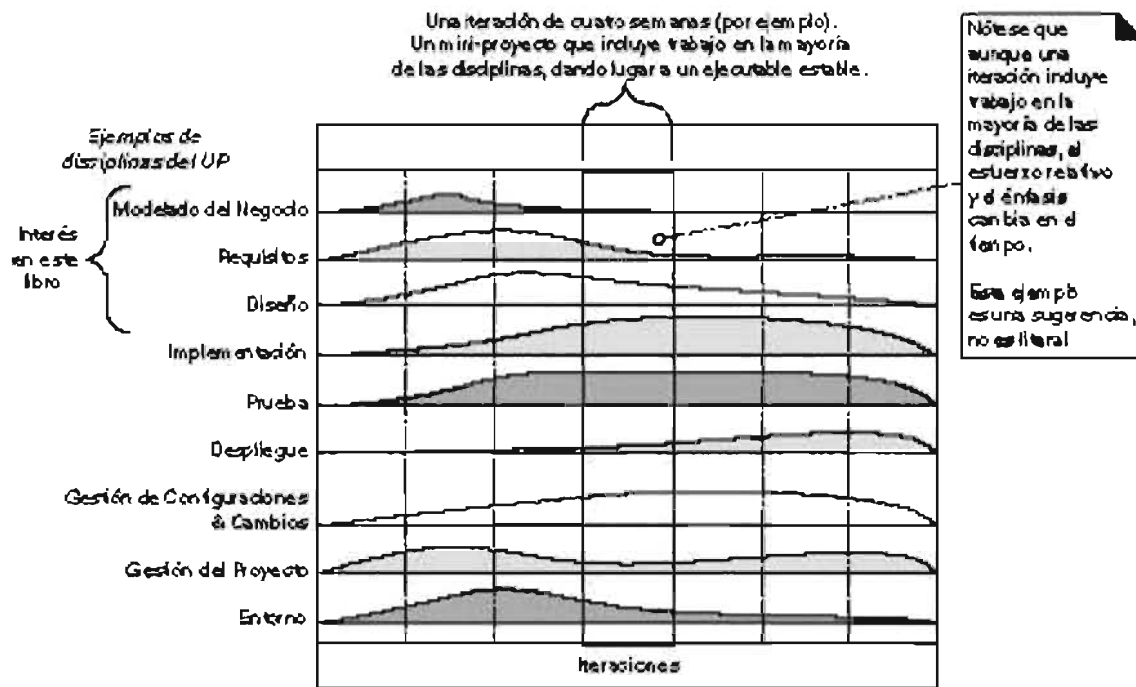


Fig1.2 Disciplinas del UP
Fuente Craig Larman 2003

Para el desarrollo de todos los modelos del UP se utilizara el Lenguaje Unificado de Modelado (UML).

1.7. 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; aún cuando todavía no es un estándar oficial, está respaldado por el OMG (Object Management Group). 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.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar.

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

Diagramas de estructura enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- [[Diagrama de componentes] OBSOLETO HOY EN DIA]
- Diagrama de objetos
- Diagrama de estructura compuesta
- Diagrama de despliegue
- Diagrama de paquetes

Diagramas de comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

Diagramas de Interacción, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación
- Diagrama de tiempos
- Diagrama de vista de interacción

1.8. ALCANCES

El presente proyecto de grado no presenta la solución total de los problemas administrativos del Seguro Social Universitario, lo que se pretende es desarrollar un sistema de control de Activos Fijos que mejore el procedimiento.

Una vez implantado el sistema gozará de las siguientes características:

- *Personal:* Se tendrá un estricto control de los inventariadores (usuarios de los activos fijos) del sistema en cuanto a todos los accesos operaciones con el sistema.
- *Movimiento:* Permitirá realizar un control de los movimientos de los activos fijos (Altas transferencias, Reposiciones, Modificaciones, Bajas)
- *Valoración:* Se concentrará en los procesos de Depreciación, Actualización y revaloraciones.
- *Conteo Físico:* Estará relacionada a la verificación física e de los activos fijos en cuanto a su ubicación y condiciones materiales.
- *Mantenimiento:* Permitirá el control de mantenimiento preventivo y correctivo de los equipos médicos.

En el presente capitulo se describe la teoría que fundamenta la realización del proyecto de grado titulado Sistema de Información y Control de Activos Fijos.

La teoría que se describe trata de dar pautas de las técnicas, metodológicas, herramientas y marco administrativo (leyes, decretos, procedimiento) que serán utilizados y aplicados. Sin embargo para obtener una explicación profunda de cada uno de los temas se incluye amplia información en la bibliografía presentada.

2.1. Activos Fijos

Los activos fijos son bienes de la empresa que tienen una vida relativamente larga y no están para la venta dentro de las actividades normales de la empresa y solo están para su servicio.

2.1.1. CLASIFICACION

Los activos fijos de una empresa se clasifican en dos grandes grupos principales

- Bienes inmuebles:

Los activos fijos de una empresa se clasifican como bienes inmuebles cuando no son trasladables, es decir, que no se pueden mover de un sitio a otro.

- Bienes inmuebles :

Los activos fijos de una empresa se clasifican como bienes muebles cuando son trasladables, es decir, que se pueden mover de un sitio a otro.

La inmensa mayoría de los activos fijos de una empresa son bienes muebles, esto es, que se pueden trasladar o mover de un lugar a otro.

De los activos fijos muebles de la empresa se destacan:

- ✓ Equipos electrónicos
- ✓ Equipamiento
- ✓ Muebles y enceres
- ✓ Hardware
- ✓ Maquinaria
- ✓ Software
- ✓ Herramientas

2.1.2. DEPRECIACION

- La depreciación es la pérdida de valor que sufre un bien de uso a través del tiempo por el servicio que presta, por inclemencias climatológicas u obsolescencia.
- Con el fin de calcular la depreciación de las propiedades, planta y equipo es necesario estimar el tiempo de vida útil y su valor residual. Se entiende por vida útil el lapso durante el cual se espera que estos activos contribuyan a la generación de ingresos de la empresa.
- Para determinarla es necesario conocer ciertos factores como las especificaciones de la fábrica, el deterioro que sufren por el uso la obsolescencia por avances tecnológicos, la acción de alguno factores naturales y los cambios en la demanda de bienes y servicios a cuya producción o servicio contribuyen.
- El valor residual o valor de salvamento es aquel precio que la empresa estima que tendrá el activo una vez terminada su vida útil.
- Existen básicamente tres tipos de depreciación
 - Deterioro físico o reducción de su utilidad debido al uso, medio ambiente, averías, conservación inadecuada, etc.
 - Obsolescencia funcional o reducción de su utilidad por singularidades o deficiencias del diseño arquitectónico que impiden la optimización de la rentabilidad del inmueble, es decir, por causas intrínsecas de la propiedad.
 - Depreciación económica o merma de su utilidad en virtud de causas de carácter económica extrínsecas a la propiedad. En general, se debe a no ubicarse en lugar apropiado para su uso, tipología y categoría.
- La depreciación física afecta solo a la edificación (a todo menos al suelo) y puede estimarse partiendo de la antigüedad y de la vida económica del edificio.

Para una mayor comprensión de el manejo de activos fijos detallamos estos en el **ANEXO A**.

2.2. INGENIERIA DEL SOFTWARE ORIENTADA A OBJETOS

La ingeniería de software orientada a objetos (ISOO) tiene como principal protagonista al objeto, pero una forma en describirlo en términos sencillos sería “el mundo está rodeado de objetos. Estos objetos están presentes en la naturaleza, en las entidades hechas por el hombre en los negocios y en los productos que usamos.”

Una ventaja de estos objetos es que pueden ser clasificados, descritos, organizados, manipulados y creados. De ahí es que nace la Visión Orientada a Objetos para el desarrollo de sistemas informáticos basados en objetos, utilizando para ello una abstracción del entorno mismo.

2.3. PROGRAMACION ORIENTADA A OBJETOS

Actualmente una de las áreas más candentes en la industria y en el ámbito académico es la orientación a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El elemento fundamental de la OOP es, como su nombre lo indica, el **objeto**. Podemos definir un objeto como **un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización**.

Dado que la OOP se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo. [Larman-1999].

2.3.1. ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

- ✓ **RELACIONES**
- ✓ **PROPIEDADES**
- ✓ **MÉTODOS**

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las **relaciones** permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las **propiedades** distinguen un objeto determinado de los restantes que forman parte de la misma organización y tienen valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los **métodos** son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia. [JGCR2005]

2.3.2. BENEFICIOS QUE SE OBTIENEN DEL DESARROLLO CON POO

Día a día los costos del Hardware decrecen. Así surgen nuevas áreas de aplicación cotidianamente: procesamiento de imágenes y sonido, bases de datos multimediales, automatización de oficinas, ambientes de ingeniería de software, etc. Aún en las aplicaciones *tradicionales* encontramos que definir interfaces hombre-máquina "a-la-Windows" suele ser bastante conveniente.

Lamentablemente, los costos de producción de software siguen aumentando; el mantenimiento y la modificación de sistemas complejos suele ser una tarea trabajosa; cada aplicación, (aunque tenga aspectos similares a otra) suele encararse como un proyecto nuevo, etc.

Todos estos problemas aún no han sido solucionados en forma completa. Pero como los objetos son portables (teóricamente) mientras que la herencia permite la reusabilidad del código orientado a objetos, es más sencillo modificar código existente porque los objetos no interactúan excepto a través de mensajes; en consecuencia un cambio en la codificación de un objeto no afectará la operación con otro objeto siempre que los métodos respectivos permanezcan intactos. La introducción de tecnología de objetos como una herramienta conceptual para analizar, diseñar e implementar aplicaciones permite obtener aplicaciones más modificables, fácilmente extendibles y a partir de componentes reusables. Esta reusabilidad del código disminuye el tiempo que se utiliza en el desarrollo y hace que el desarrollo del software sea más intuitivo porque la gente piensa naturalmente en términos de objetos más que en términos de algoritmos de software. [JGCR-2005]

2.3.3. ANALISIS ORIENTADO A OBJETOS

Se puede definir al Análisis Orientado a Objetos (AOO) como el proceso que se utiliza para modelar el dominio del problema identificado y especificando un conjunto de objetos semánticos que interactúan y se comportan de acuerdo a los requisitos del sistema. [JGCR2005]

El AOO se fundamenta en un conjunto de cinco principios básicos:

- a) Modelar el dominio de la información.
- b) Describir la función del módulo.
- c) Representar el comportamiento del modelo.
- d) Dividir el Modelo para mostrar más detalles.
- e) Observar que los modelos iniciales representan la esencia del problema mientras que los últimos aportan detalles de la implementación.

El propósito del AOO es definir todas las clases, además de las relaciones y comportamientos asociados a ellas, que son relevantes al problema a ser resuelto.

Para cumplirlo se deben ejecutar las siguientes tareas:

- a) Los requisitos básicos del usuario deben comunicarse entre el cliente y el ingeniero de software
- b) Identificar las clases definiendo los atributos y los métodos
- c) Especificar una jerarquía de clases
- d) Representar las relaciones conexiones, objeto a objeto.
- e) Modelar el comportamiento del objeto.
- f) Repetir iterativamente las tareas 1 a la 5 hasta completar el modelo.

2.3.4. DISEÑO ORIENTADO A OBJETOS (DOO)

El Diseño Orientado a Objetos transforma el producto obtenido en la anterior etapa o sea el modelo de análisis en un modelo de diseño que sirve como un anteproyecto para la construcción de software. A diferencia de los métodos convencionales de diseño del software, el DOO constituye un tipo de diseño que logra un cierto número de diferentes niveles de modularidad. Los componentes principales del sistema están organizados en módulos denominados subsistemas.

El DOO debe describir la organización de datos específicos, de atributos y los detalles procedimentales de las operaciones individuales. Esta representación fragmentada de datos y algoritmos de un sistema orientado a objetos colaboran a una modularidad general.

La naturaleza única del DOO descansa en su capacidad para apoyarse en cuatro conceptos importantes del diseño clásico del software:

- Abstracción, consiste en ignorar aquellos aspectos del problema en estudio que no son relevantes para centrarse en aquellos que si son importantes.
- Ocultación de la información, este concepto está íntimamente asociado al encapsulamiento en donde los datos y las operaciones unen en un paquete
- Independencia Funcional, en el cual se buscan los módulos u operaciones que sean capaces de realizar o especializarse en una tarea en particular.

- Modularidad, este concepto permite la reutilización y facilita la verificación y depuración de los mismos. Los objetos son módulos naturales ya que corresponden a una representación lógica de la realidad.

Para los sistemas orientados a objetos es posible definir un diseño en pirámide con las siguientes cuatro capas:

- Subsistema. Contiene una representación de cada uno de los subsistemas que le permiten al software conseguir los requisitos definidos por el cliente e implementar la infraestructura técnica que los soporta.
- Clases y Objetos. Contienen la jerarquía de clases que permiten crear el sistema utilizando generalizaciones y especializaciones mejor definidas incrementalmente. También contiene representaciones de diseño para cada objeto.
- Mensajes. Contiene los detalles que permiten a cada objeto comunicarse con sus colaboradores. Establece las interfaces externas e internas para el sistema.
- Responsabilidades. Contiene las estructuras de datos y el diseño algorítmico para todos los atributos y operaciones de cada objeto.
[JGCR2005]

2.4. LENGUAJE UNIFICADO DE MODELADO

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

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

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

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

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

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

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

Elementos

Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)

Hay cuatro tipos de elementos:

1. **Los Elementos estructurales:** Representan partes materiales o conceptuales del sistema, los elementos estructurales principales de UML son:
 - ◆ La clase.
 - ◆ La interfaz.
 - ◆ La colaboración.
 - ◆ El caso de uso.
 - ◆ Los componentes.
 - ◆ Los nodos

2. **Los elementos de comportamiento:** Son las partes dinámicas de los modelos UML.

Los elementos de comportamiento básico son:

Las interacciones, que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular, para alcanzar un propósito específico.

La máquina de estados, que especifica las secuencias de estados por las que pasa un objeto o una interacción durante su vida en respuesta a eventos, junto con sus reacciones a estos eventos. [JOS2005]

3. Los elementos de agrupación: Son los paquetes, y sirven para organizar otros elementos de los modelos UML. Un paquete es un mecanismo conceptual de propósito general para organizar elementos en grupos. Por ejemplo, se pueden incluir en un paquete varias clases y luego asignar ese paquete a un componente, con esto lo que se está haciendo es asignar todas las clases al componente a través de un paquete. [JOS2005]

4. De anotación: Los elementos de anotación son partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clarificar o hacer observaciones sobre cualquier elemento de un modelo. [JOS2005]

Relaciones

Relacionan los elementos entre sí. Los tipos fundamentales de relaciones son los siguientes:

- ◆ Generalización: Conecta a elementos vinculados por una relación padre/hijo (generalización / especialización). Gráficamente viene dada por una línea acabada en punta de flecha vacía que va del elemento especializado (el hijo) al general (el padre) como se muestra en la figura 2.1.

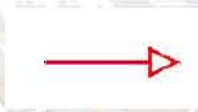


Fig.2.1 Símbolo para expresar la generalización
Fuente: Larman [1999]

- ◆ Asociación: Es una relación estructural entre dos elementos. Opcionalmente puede también acabar en punta de flecha para indicar la dirección de navegabilidad o incluir ciertos adornos que aumentan su expresividad como se muestra en la figura 2.2.



Fig.2.2 Símbolo para expresar la asociación
Fuente: Larman [1999]

- ◆ Realización: Es una relación semántica entre clasificadores, en donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Es la relación que hay entre una especificación y su implementación ver figura 2.3.



FIG.2.3 Símbolo para expresar la realización
Fuente: Larman [1999]

- Dependencia: Es una relación semántica entre dos elementos, en la cual un elemento requiere la presencia de otro para su implementación o funcionamiento. Un cambio en el elemento requerido (el independiente) puede afectar al del otro elemento (el dependiente).

Se representa como una línea discontinua entre el elemento dependiente, y el independiente acabada en punta de flecha que señala al elemento independiente (esto es, el elemento dependiente indica el elemento del que depende) ver figura 2.4.



Fig.2.4 Símbolo para expresar la dependencia
Fuente: Larman [1999]

2.4.1. DIAGRAMAS UML

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

Los diagramas más interesantes (y los más usados) son los de casos de uso, clases y secuencia, por lo que nos centraremos en éstos. Para ello, se utilizará ejemplos de un sistema de venta de entradas de cine por Internet.

El *diagrama de casos de usos* representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. En la **figura 2.5** se muestra un ejemplo de casos de uso, donde se muestran tres actores (los clientes, los taquilleros y los jefes de taquilla) y las operaciones que pueden realizar (sus roles).

El *diagrama de clases* muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. En la **figura 2.6** se muestran las clases globales, sus atributos y las relaciones de una posible solución al problema de la venta de entradas.

En el *diagrama de secuencia* se muestra la interacción de los objetos que componen un sistema de forma temporal. Siguiendo el ejemplo de venta de entradas, la **figura 2.7** muestra la interacción de crear una nueva sala para un espectáculo.

El resto de diagramas muestran distintos aspectos del sistema a modelar. Para modelar el comportamiento dinámico del sistema están los de *interacción*, *colaboración*, *estados* y *actividades*. Los *diagramas de componentes* y *despliegue* están enfocados a la implementación del sistema.

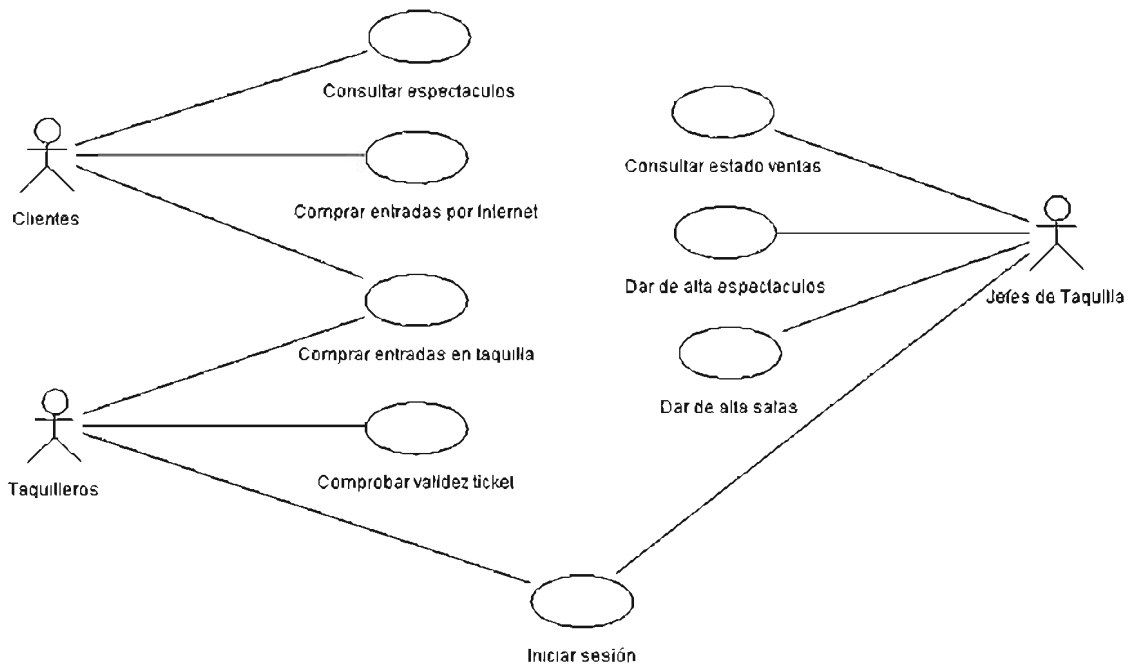


Fig 2.5 Diagrama de casos de uso. [JOS2005]

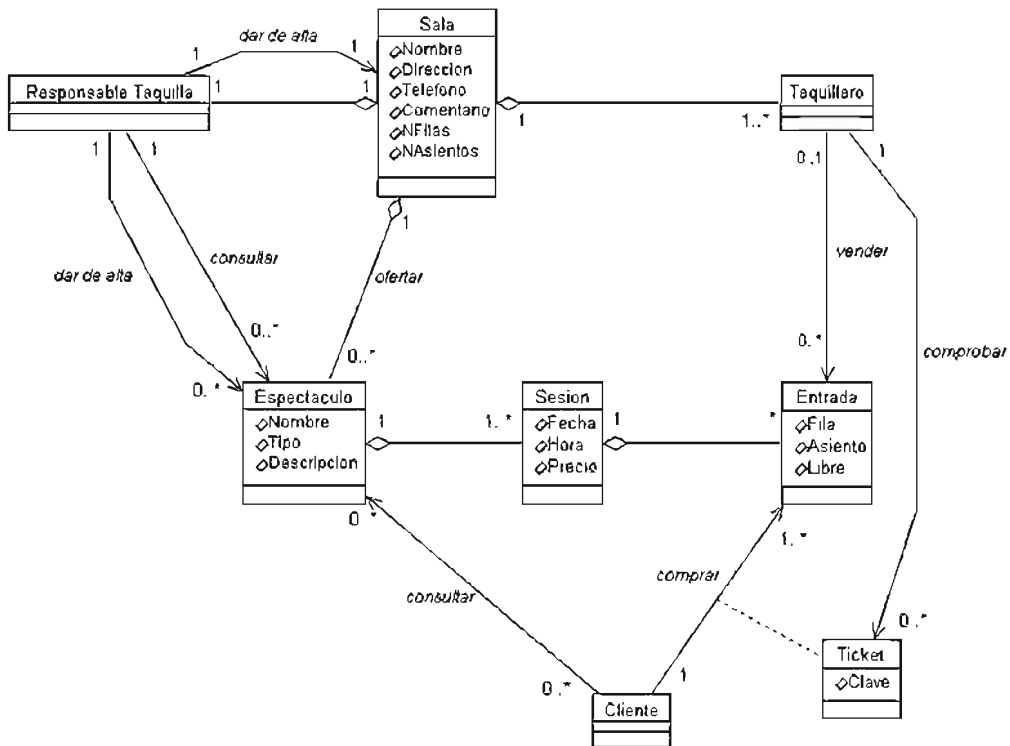


Fig2.6 Diagrama de clases. [JOS2005]

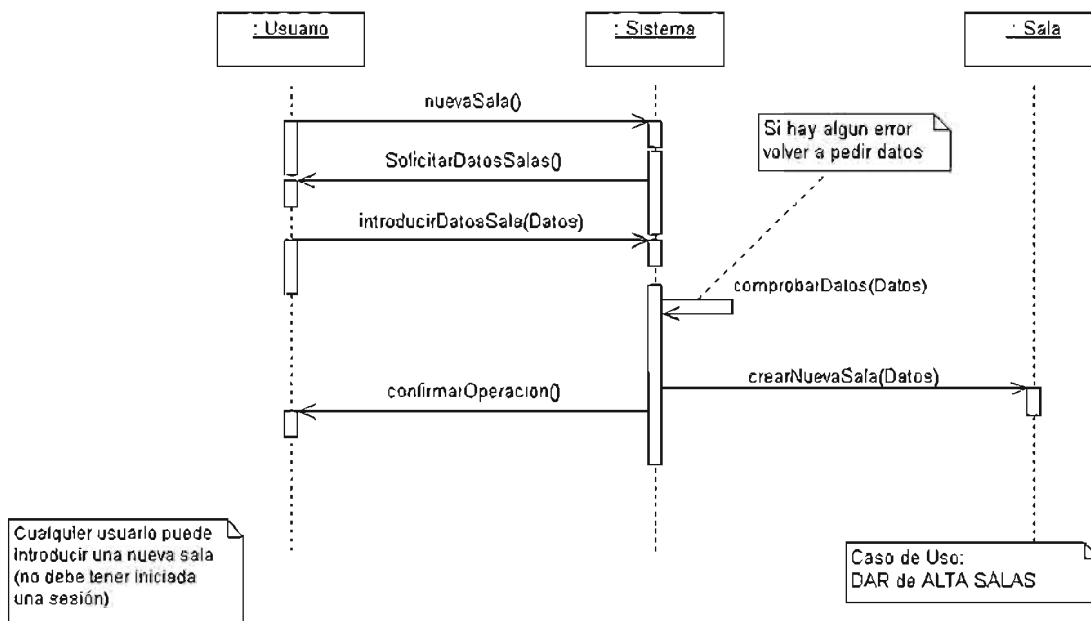


Fig. 2.7 Diagrama de secuencia. [JOS2005]

2.5. PROCESO DE DESARROLLO

Aunque UML es bastante independiente del proceso de desarrollo que se siga, los mismos creadores de UML han propuesto su propia metodología de desarrollo, denominada el *Proceso Unificado de Desarrollo*.

El **Proceso Unificado** está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidos. Además, el Proceso Unificado utiliza el UML para expresar gráficamente todos los esquemas de un sistema software. Pero, realmente, los aspectos que definen este Proceso Unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura:

- *Dirigido por casos de uso*: Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.
- *Centrado en la arquitectura*: En la arquitectura de la construcción, antes de construir un edificio éste se contempla desde varios puntos de vista: estructura, conducciones

eléctricas, fontanería, etc. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.

- *Iterativo e incremental*: Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases. Cada recorrido por las fases se denomina iteración en el proyecto en la que se realizan varios tipos de trabajo (denominados flujos). Además, cada iteración parte de la anterior incrementado o revisando la funcionalidad implementada. Se suele denominar proceso. **Ver fig2.8**

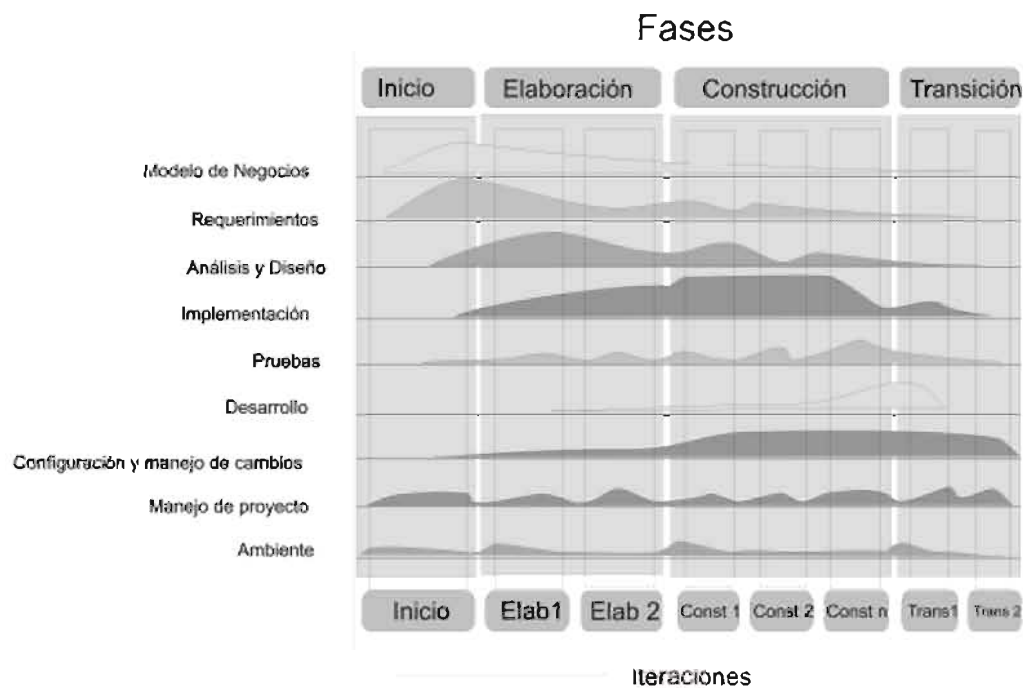


Fig. 2.8 Arquitectura del UP. [LARMAN 2003]

2.6. DEFINICION DE ACTIVIDADES

RUP o UP identifica actividades con la que define una forma efectiva de trabajar para los equipos de desarrollo de software estas son:

- a) **Gestión de Requisitos.** UP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de Casos de Uso y escenarios para representar los requisitos
- b) **Desarrollo de Software Iterativo.** Desarrollo del producto mediante iteraciones con hitos bien definidos, en las cuales se repiten las actividades pero con distinto énfasis, según la fase del proyecto.
- c) **Desarrollo Basado en Componentes.** La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o se desarrollan sus componentes.
- d) **Modelo Visual.** UP utiliza UML que es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema software. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. El modelo visual también ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. En resumen el modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software.
- e) **Verificación Continua de la calidad.** Es importante que la calidad de todos los artefactos se evalúe en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración. En esta verificación las pruebas juegan un papel fundamental y se integran a lo largo de todo el proceso. Para todos los artefactos no ejecutables las revisiones e inspecciones también deben ser continuas.
- f) **Gestión de Cambios.** El cambio es un factor de riesgo crítico en los proyectos de software. Los artefactos software cambian no solo debido a acciones de mantenimiento posteriores a la entrega del producto, sino que durante el proceso

de desarrollo, especialmente importantes por su posible impacto son sus cambios en los requisitos.

El Proceso Unificado es un proceso por que “define quien está haciendo que, cuando lo hace y como alcanzar cierto objetivo, en este caso el desarrollo de software”. Los conceptos clave del Proceso Unificado son:

Fase e iteraciones	¿Cuándo se hace?
Flujos de trabajo de Procesos	¿Qué se está haciendo?
Artefactos modelos reportes documentos	¿Quién lo hace?
Trabajador	¿Quién lo Hace?

Tabla 2.1 conceptos clave del UP
Fuente: Presman [sexta edición]

2.6.1. FASE DE CONCEPCION E INICIO

El objetivo de esta fase es elaborar una buena planificación del proyecto, realizando un análisis de negocio hasta lograr una vista general para la puesta en marcha del proyecto. Esta planificación se lograra a través de, la especificación de los objetivos, definir la funcionalidad y capacidades del producto, definir los límites y los alcances, definir la arquitectura inicial sujeta a cambios, determinar los riesgos y posibles medidas para reducirlos, estimar costos y definir una agenda preliminar.

Esta frase comprende:

- **Plan de desarrollo de Software**, se define la organización de las cuatro fases e iteraciones que se desarrollan en cada una de ellas, se describe también los hitos que marca el final de cada fase, como se puede apreciar en la siguiente tabla

FASE	HITO
Inicio	Se desarrollara los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto visión. Los principales casos de uso serán identificados y será un refinamiento del Plan de Desarrollo del Proyecto. La aceptación de Cliente/Usuario del artefacto visión y el plan de desarrollo marca el final de esa frase.
Elaboración	Se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y/o críticas de los sistemas). Al final de esta fase, todos los casos de uso deben estar analizados y diseñados (en el modelo análisis/diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el modelo de análisis y diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesaria la planificación para asegurar el cumplimiento de los objetivos.
Construcción	Se terminaran de analizar y diseñar todos los casos de uso, refinando el modelo de análisis/Diseño. El producto Software se construye en más de dos iteraciones en las primeras se obtiene una versión del producto las cuales se validan con el cliente y se comienza la elaboración del material de apoyo al usuario. El hito que marca fin de esta fase son los módulos terminados del producto, realizando sobre ellos pruebas de integración y funcionalidad de acuerdo a los requerimientos esta integración se realiza en la última iteración
Transición	Se libera el producto software para su entrega al usuario para su uso real con tareas de instalación y configuración incluyendo el entrenamiento de los usuarios. En la primera iteración se realiza las pruebas con datos reales y en cada una de las funciones. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el materia de apoyo al usuario, La finalización de entrenamiento de los usuarios y la presentación del producto final probado (corregido en caso de falla) en la primera iteración

Tabla 2.2 Descripción de los hitos de las fases
Fuente: [I. Jacobson, Grady Booch, J. Rumbaugh, Ed Addison Wesley, 1999]

- **Informe Preliminar de Investigación**, se realiza un análisis con motivos, alternativas y necesidades de la empresa u organización, por lo que se opto o se justifica el desarrollo del sistema.
- **Definir los requerimientos**, se recopila los requerimientos que tiene la institución, a través de los problemas que presenta, los cuales serán resueltos a partir de la solución propuesta en el desarrollo del sistema.
- **Registro de términos del Glosario**, se desarrolla un diccionario con los conceptos y especificaciones que contienen toda la información concerniente al sistema y sus reglas de desarrollo.
- **Definir la Arquitectura del Sistema** de acuerdo al estudio de requerimientos y soluciones se plantea un bosquejo de la estructura física-lógica del sistema.
- **Perfeccionamiento del plan** con lo desarrollado anteriormente, se revisa y modifica la planificación realizada al principio.

De acuerdo a la metodología propuesta la fase de inicio se desarrolla de acuerdo a las disciplinas como se muestra en la siguiente figura:

Las tareas que se deben desarrollar en esta fase están de acuerdo a los flujos de trabajo o disciplinas logrando desarrollar especialmente el modelado de Negocio y Requisitos, obteniendo la base del desarrollo que se describe en la tabla.

DICIPLINA	ACTIVIDAD
REQUISITOS	<ul style="list-style-type: none"> ◆ Recopila los requisitos iniciales(características del sistema) ◆ Comprende el comienzo del sistema. ◆ Representa los requisitos con casos de uso ◆ Recopila los requisitos no funcionales
ANALISIS	<ul style="list-style-type: none"> ◆ Análisis de la arquitectura ◆ Análisis de los casos de uso
DISEÑO	<ul style="list-style-type: none"> ◆ Esbozo de la arquitectura
IMPLEMENTACION	<ul style="list-style-type: none"> ◆ Prototipo desechable
PRUEBAS	<ul style="list-style-type: none"> ◆ No se realizan pruebas en esta fase

Tabla2.3 disciplinas de la fase de inicio
Fuente [JAC1999]

De acuerdo a lo señalado en la tabla anterior los artefactos que se obtienen en esta fase de acuerdo a las disciplinas son:

ARTEFACTO	DESCRIPCION
Visión	Objetivos y restricciones
Especificación Adicional	Requisitos no funcionales
Modelado de Casos de Uso de alto nivel y esenciales	Describe los requisitos funcionales y esenciales en los que se describe la secuencia de eventos entre los actores y el sistema
Modelado Inicial del Dominio	Define el contexto
Modelado inicial de Prototipo	Esbozo Inicial
Prototipos	Validar la tecnología
Plan de desarrollo	Recursos para el desarrollo
Lista de riesgos	Riesgos posibles
Análisis de negocio	Beneficios
Casos de desarrollo	Como se aplica el UP al proyecto

Tabla2.4 Artefactos de la fase de Inicio
Fuente [JAC1999]

Se describe también las funciones del sistema que son las acciones o procesos a ser realizados para alcanzar el objetivo que presenta el proyecto, estas funciones pueden ser organizadas de dos tipos como se muestra en la siguiente tabla:

CATEGORIA DE FUNCION	SIGNIFICADO
EVIDENTE	Debe realizarse y el usuario deberá saber que se ha realizado
OCULTA	Debe realizarse, aunque no es visible para los usuarios. Esto se aplica en muchos servicios subyacentes, como guardar información en un mecanismo persistente de almacenamiento. Las funciones ocultas a menudo se omiten (erróneamente) durante el proceso de obtención de requerimientos.

Tabla 2.5 Categoría de las funciones del sistema
Fuente: [Larman 2003]

2.6.2. FASE DE ELABORACION

El objetivo principal de esta fase es la de realizar una planificación del proyecto en función de los recursos disponibles, así como un estudio más profundo de los requisitos de la

funcionalidad y del dominio del problema, se define también en una primera iteración la arquitectura básica para luego consolidarla a partir de las siguientes iteraciones. En cuanto a los requisitos, estos son detallados lo más posible para poder plasmarlos en la arquitectura del sistema tomando en cuenta los riesgos identificados en la fase de inicio, estimando costos y tiempos de ejecución.

Esta fase está dada por más de una iteración para lograr un diseño básico lo más completa y detallado posible, de acuerdo a las disciplinas los objetivos en cada uno esta detallado en la siguiente tabla:

DISCIPLINA	OBJETIVO
REQUISITOS	<ul style="list-style-type: none"> ◆ Encontrar los casos de uso y actores de manera detallada ◆ Determinar la prioridad de los casos de uso ◆ Construir el modelo de casos de uso y los prototipos de las interfaces de usuario.
ANALISIS	<ul style="list-style-type: none"> ◆ Análisis de la arquitectura, de los casos de uso, de las clases y de los paquetes.
DISEÑO	<ul style="list-style-type: none"> ◆ Diseño de la arquitectura (general y subsistemas) y de los casos de uso (por subsistemas)
IMPLEMENTACION	<ul style="list-style-type: none"> ◆ Implementación de la arquitectura base (para los casos de uso)
PRUEBAS	<ul style="list-style-type: none"> ◆ Planificar y diseñar las pruebas ◆ Realizar pruebas de integración y del sistema

Tabla2.6 disciplinas de la fase de elaboración
Fuente [JAC1999]

De acuerdo a lo señalado en la tabla anterior los artefactos que se obtienen en esta tabla son:

ARTEFACTO	DESCRIPCION
Modelo de Casos de Uso	La mayoría de los casos de uso
Modelo del Dominio	Conceptos del dominio
Modelo de Análisis	Diagrama de clases Diagrama de iteración
Modelo de Diseño	Diagramas y paquetes de clases
Arquitectura del sistema	Esbozo general del diseño que se utilizara en el sistema
Modelo de Implementación	Incluye diagramas de implementación y código fuente
Modelo de pruebas	Que

	debe probarse y sobre que artefactos
Prototipos de interfaz de Usuario	Todo lo relacionado con la interfaz
Modelo de Datos	Esquema de base de datos

Tabla2.7 Disciplinas de la fase de elaboración
Fuente [Larman 1999]

REGISTRO DE TERMINOS EN EL GLOSARIO

El glosario es también un diccionario de datos que incluye todos los términos que se requieren explicación para mejorar la comunicación y aminorar el riesgo de interpretación de términos. El significado uniforme de términos da como resultado una buena comprensión del problema que ocurrirá en el desarrollo de las aplicaciones, siendo utilizado como un documento referencial que debe estar de acuerdo a la siguiente tabla:

TERMINO	CATEGORIA	COMENTARIO
Nombre del termino	Caso de uso/atributo/tipo	Breve descripción del termino

Tabla2.8 elementos del glosario
Fuente [Larman 1999]

2.6.3. FASE DE CONSTRUCCION

La fase de construcción se desarrolla al igual que las anteriores a través de iteraciones en el desarrollo del sistema. Esta fase puede ser lo más extensa y de acuerdo a sus objetivos, el número de iteraciones dependerá de la planificación de la construcción del mismo. Uno de los objetivos de esta fase es refinar la arquitectura básica de manera incremental conforme se vaya construyendo, además de manera paralelamente se realiza la documentación del sistema en cuanto a su construcción y manejo. Al empezar la fase se debe fijar los criterios de evaluación de acuerdo a los requerimientos especificados en los casos de uso detallados en la fase de elaboración. De acuerdo a lo mencionado, la mayor parte del trabajo será: realizar la programación y pruebas de los módulos que se vayan desarrollando de acuerdo a las disciplinas y los objetivos de cada uno se van detallando en la siguiente tabla:

DISCIPLINA	OBJETIVOS
Requisitos	<ul style="list-style-type: none"> ● Contemplar los casos de uso y el detalle de los mismos ● Desarrollar prototipos de interfaz de usuario.
Análisis	<ul style="list-style-type: none"> ● Análisis de los casos de uso añadidos
Diseño	<ul style="list-style-type: none"> ● Diseño de los casos de uso añadidos
Implementación	<ul style="list-style-type: none"> ● Implementación de la Arquitectura ● Implementación de clases y subsistemas ● Integración de los módulos del sistema
Pruebas	<ul style="list-style-type: none"> ● Planificar y diseñar las pruebas ● Realizar pruebas de integración ● Realizar pruebas de sistema ● Evaluar las pruebas

Tabla2.9 Disciplinas de la fase de Construcción
Fuente [JAC1999]

De acuerdo a lo señalado en la tabla anterior los artefactos que se obtienen en esta fase de acuerdo a las disciplinas son:

ARTEFACTO	DESCRIPCION
Modelo de casos de Uso	Conjunto de artefactos producidos hasta el momento
Modelo de Análisis	
Modelo de Diseño	
Modelo de Pruebas	
Arquitectura del sistema	
Modelo de implementación	Incluye el código fuente
Modelo de pruebas	
Sistema Ejecutable	Versión con capacidad operativa inicial

Manual de Usuario	Versión Inicial
Análisis del sistema	Situación Actual del Proyecto
Plan de Apoyo	Plan para la fase de transición

Tabla2.10 Artefactos d la fase de construcción
Fuente [JAC1999]

2.6.4. FASE DE TRANSICION

La fase de transición es la fase final de desarrollo del sistema en la que se libera el producto y se entrega al usuario para su uso real, al empezar esta fase se establece criterios de evaluación que estarían basados en los requisitos detallados en los casos de uso de alto nivel y los casos de uso detallados, así como en el diseño del sistema. De acuerdo a esta evaluación también se revisara y complementara la documentación del sistema y el manual de usuario con la información obtenida anteriormente, para aplicar métricas de calidad que nos ayuden a determinar la calidad de software.

El trabajo en esta fase será específicamente, la configuración, instalación, entrenamiento y soporte. Todas estas tareas también se realizaran en iteraciones en las cuales se tomaran en cuenta las disciplinas.

El esquema de esta fase es distinto al resto de las fases con los siguientes objetivos:

- Preparar la versión de pruebas de aceptación a partir de la versión inicial.
- Instalación de la primera versión en lugares elegidos
- Plan de riesgos en caso de resultados inesperados (como ser: fallos de /en componentes de un caso de uso y problemas de fondo)
- Adaptación del prototipo a entornos variados

2.7. DISEÑO DE LA BASE DE DATOS RELACIONAL – EXTENSION INFORMAL DEL UML

El diagrama de clase presenta un mecanismo de implementación neutral para modelar los aspectos de almacenamiento de datos del sistema. Las clases persistentes, sus atributos y sus relaciones pueden ser implementados directamente en una base de datos orientada a objetos. Aun así en el entorno de desarrollo actual, la base de datos relacional es el método más usado para el almacenamiento de datos.

El diagrama de clases de UML se puede usar para modelar algunos aspectos del diseño de base de datos relacionales pero no cubre toda la semántica involucrada en el modelo relacional. Para capturar información, un diagrama Entidad Relación se utilizara como una extensión a UML.

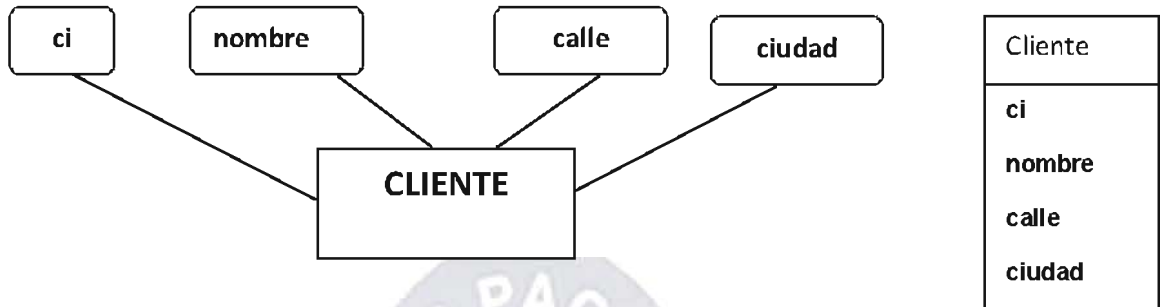
El Diagrama de clases se puede usar para modelar una estructura lógica de la base datos independientemente de si es orientada a objetos o relacional, con clases representando tablas y atributos de clase representando columnas. Si una base de datos relacional es el método de implementación escogido entonces el diagrama de clases puede ser referenciado a un diagrama de relación de entidad lógico.

Las clases persistentes y sus atributos hacen referencia directamente a las entidades lógicas y a sus atributos, el modelador dispone de varias opciones sobre como inferir asociaciones en relaciones entre entidades. Las relaciones de herencia son referenciadas directamente o súper-sub relaciones entre entidades en un diagraman de Entidad Relación.

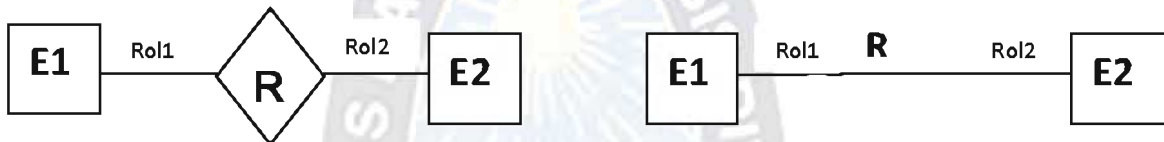
Ya en el DER, el modelador puede empezar el proceso de determinar como el modelo relacional encaja y que atributos son claves primarias, claves secundarias, y claves externas basadas en relaciones con otras entidades. La idea es construir un modelo lógico que sea conforme a las reglas de normalización de datos.

Al implementar el diseño relacional, la estrategia está encaminada a ser referencia al diagrama de relación de entidad lógico a un diagrama físico que represente el objetivo el RDBMS. El diagrama físico puede ser normalizado para lograr un diseño de base datos que tiene tiempos eficientes de acceso a los datos. Las relaciones súper-sub entre entidades se resuelven por las estructuras de tablas actuales. Además el diagrama físico se usa para modelar propiedades específicas de cada fabricante para el RDBMS.

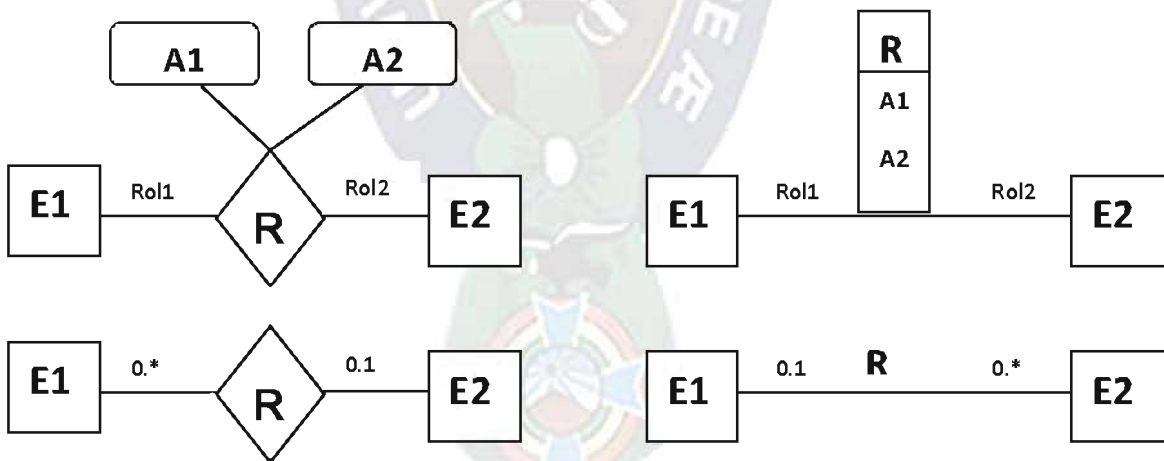
a) ENTIDADES Y ATRIBUTO



b) RELACIONES



c) CARDINALIDAD ENTRE ENTIDADES



d) GENERALIZACION Y ESPECIFICACION

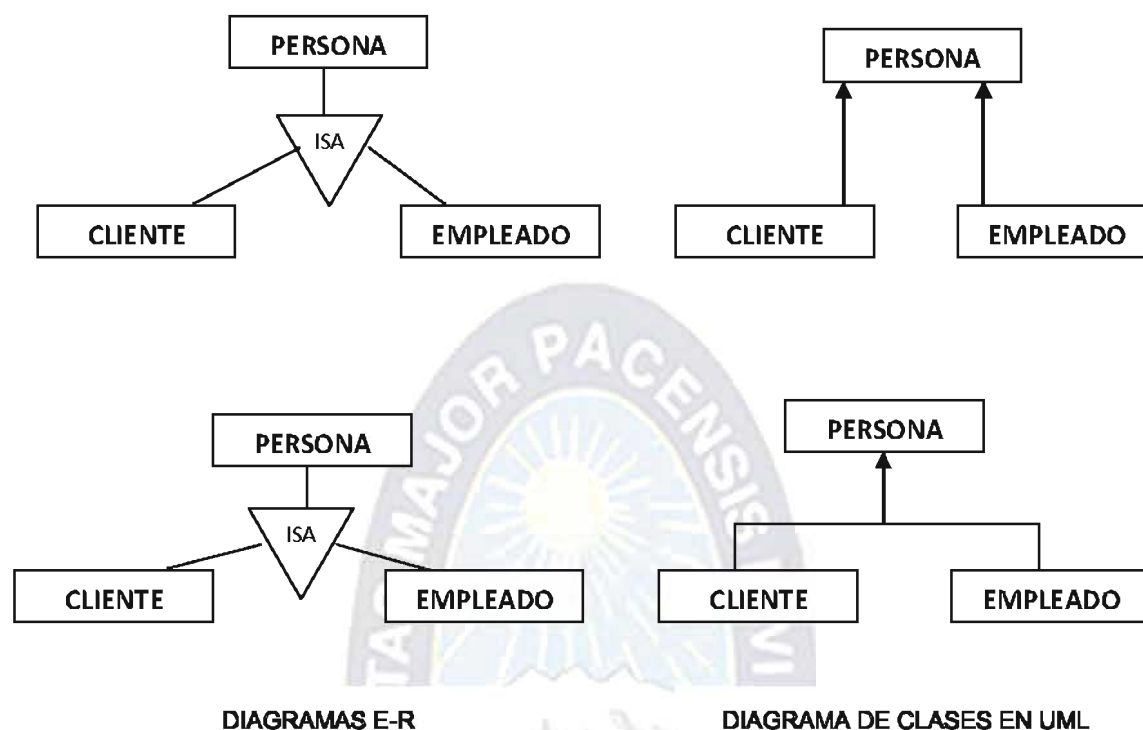


Figura2.9 Notación UML para modelar E-R
Fuente [ABC, 2008]

2.7.1. UN ANÁLISIS ACERCA DE LA TRANSFORMACIÓN DE UN MODELO UML EN OBJETOS DE UNA BASE DE DATOS OBJETO-RELACIONAL.

A continuación se analizan las transformaciones posibles para convertir un Diagrama de Clases de UML (Unified Modeling Language) en un modelo conceptual de una base de datos objeto-relacional (ORDBMS).

Una transformación es un “mapeo” que permite partir de un modelo que pertenece a un dominio determinado (Diagrama de Clases) y arribar a un modelo que pertenece a otro dominio (Modelo Conceptual). Aplicando una serie de transformaciones uno puede simplificar y optimizar dicho modelo. No existe una metodología para la transformación de los modelos planteados, como existe para el caso del modelo relacional, que está basada en la teoría de la normalización de relaciones. Tampoco es posible plantear un único modo de hacer este mapeo. Las posibilidades son amplias, dadas las capacidades del

modelo objeto-relacional. El mapeo que se genere puede ir entre los dos extremos: relacional puro y objeto puro, con toda una gama intermedia que ofrece un mapeo combinando las dos tecnologías. El mapeo final seleccionado tiene un fuerte impacto en las propiedades del sistema, y puede afectar notablemente el costo, robustez y flexibilidad del mismo, de ahí la importancia del trabajo realizado. [DEP2003]

Se emplea UML porque es el lenguaje de modelado que se ha transformado en estándar en los últimos años para generar los modelos de datos de las aplicaciones. UML está basado en el paradigma orientado a objetos y posee una serie de modelos que permiten plasmar con facilidad los requerimientos de una aplicación, generar su diseño e implementarlo. La parte de UML que está relacionada con los datos se denomina Diagrama de Clases. A partir de este diagrama y los requerimientos del sistema se puede lograr el modelo conceptual de una Base de Datos, aplicando las transformaciones que más se adapten al sistema en estudio. [DEP2003]

La base del pensamiento objeto-relacional es la habilidad para incorporar mayores niveles de abstracción en los modelos de datos, y acercar la propuesta relacional a la del paradigma orientado a objetos. Esta idea representa un cambio importante en la forma de modelar los mismos. Las bases de datos relacionales corrientes usualmente son modelos altamente normalizados con poca abstracción. Cada “cosa de interés” es instanciada como una tabla relacional. Como resultado, los sistemas frecuentemente requieren de cientos de tablas de base de datos e igual número de módulos de pantallas y reportes. Los módulos de programas están basados directamente en estas tablas, con el flujo de trabajo del usuario instanciado, solamente a través de la forma en que los cientos de módulos interactúan. Sin embargo, la incorporación del pensamiento objeto-relacional, proveyendo una serie de abstracciones distintas, potencian sus capacidades, con las que se esperan obtener las siguientes ventajas:

- ◆ Construir sistemas de manera más rápida, sencilla y a menor costo,
- ◆ Construir sistemas más flexibles y robustos, de manera que las modificaciones que se realicen en el sistema y los nuevos requisitos que vayan surgiendo sean más simples de acomodar y de menor costo. [DEP2003]

MATERIALES Y MÉTODOS

1. Colección y análisis de los requerimientos: en esta etapa se debe tener en cuenta las necesidades de los usuarios, analizar el ambiente de trabajo actual y el uso futuro de la información.

2. Diseño del Esquema Conceptual: a partir del análisis de requerimientos se diseña el modelo de los datos de la base, identificando entidades, atributos, relaciones, cardinalidades. Se identifican además las transacciones que se ejecutarán sobre la base de datos (transacciones de actualización, de recuperación, de eliminación) y se las diseña.

Se utiliza UML para la definición de los diagramas de clases y las diferentes relaciones entre las mismas.

3. Elección de un DBMS: se elige una Base de Datos Objeto Relacional, que incluyen el manejo de tipos de datos complejos, colecciones, relaciones de herencia, composición y asociación. Particularmente se trabajara con SQL Server, teniendo en cuenta factores como disponibilidad, factores técnicos, económicos y organizacionales.

4. Mapeo del Modelo de Objetos: se realiza la transformación del modelo de datos generado en la etapa 2, teniendo en cuenta el DBMS elegido. Los resultados de esta etapa son las sentencias DDL en el lenguaje del DBMS seleccionado. La ejecución de esta etapa y las adopciones realizadas son las que se comunicarán en este trabajo.

5. Diseño del Esquema Físico: se definen las estructuras de almacenamiento y los caminos de acceso para los datos de la base de datos.

6. Implementación de la Base de Datos y Tuning: esta etapa, puede afectar las etapas anteriores, fundamentalmente las 4 y 5, ya que es muy probable la introducción de cambios en los mapeos y en el diseño del esquema físico para que se pueda obtener una Base de Datos flexible y con tiempos de respuesta aceptables. [DEP2003]

Asociación:

Una asociación es la relación que se encuentra más frecuentemente en los modelos y que los diseñadores deberían utilizar como último recurso, es decir que primero deben explorar si no existe otra forma de relación entre las clases.

Esto es así porque la asociación no indica cómo actúa una clase sobre la otra sino que más bien representa el hecho que las clases relacionadas tienen algo que ver entre ellas. Esta forma de relación está caracterizada por la cardinalidad, que puede ser 1:1(uno-uno),

1: n (1-muchos) y n: m (muchos-muchos), y por la existencia, que puede ser obligatoria u opcional. La primera indica cuantos objetos de una clase están relacionados con los objetos de la otra clase, mientras que la existencia indica si todos los objetos de una clase deben estar obligatoriamente relacionados con algún objeto de la otra clase o no. El modelo relacional plantea un mapeo de la asociación que se basa en la definición de claves primarias y foráneas. Para estos casos el sistema de Base de Datos mantiene la integridad referencial entre las clases. El modelo objeto-relacional posibilita un mapeo que permite incluir una referencia o un arreglo de referencias a objetos en cada una de las clases que están ligadas. La ventaja de esta transformación, comparada con la relacional, es que en este caso no es necesario realizar operaciones de joins para averiguar cuáles son los objetos que se relacionan de ambas clases, permitiendo además, una navegación más rápida y natural entre los objetos. La desventaja de este mapeo es que la integridad de las referencias se debe programar ya que el sistema no provee aún un sustento para las mismas. Para nuestro caso, decidimos optar por esta última transformación, dado que es importante contar con las **ventajas** a nivel de manipulación de la información más que la integridad referencial, dado que las relaciones de asociación planteadas en nuestro caso no son críticas en cuanto al mantenimiento de la integridad referencial, por lo que no es necesario plantear un gasto extra de programación. Lo que finalmente se resolvió para este tipo de relaciones es lo siguiente:

- Si la relación es uno a uno se representa mediante un atributo de referencia en una de las clases intervinientes (relación unidireccional), o en las dos clases (relación bidireccional), según las necesidades de la aplicación. En general se ubica el atributo en la clase principal, a través de la cual se resuelve la referencia. En nuestro trabajo, en la asociación *especie - morfología*, la referencia se puso en especie, debido a que es una de las **clases** principales o más relevantes del sistema. Una especie tiene una morfología y cada morfología pertenece a exactamente una especie.
- Con la cardinalidad es uno a muchos, existen varias opciones: definir en la clase correspondiente al “muchos” una referencia a la otra clase, o definir un arreglo de referencias en la clase correspondiente al “uno” de la relación.

También existe la posibilidad de definir ambas opciones ya que esto permite mayor flexibilidad. [DEP2003]

- Si la relación es muchos a muchos, la solución es definir en una o en las dos clases un arreglo de referencias a la otra. Esta última forma representaría la relación en ambas direcciones y es más fácil obtener objetos a partir de cualquiera de las clases, pero a su vez es más costosa de mantener. Las referencias se recuperan como un todo y no en base a algún criterio.

Composición

La relación de composición especifica que un objeto de la clase A “está compuesto” por un objeto de la clase B. Los objetos de la clase B no pueden existir de manera independiente, si el objeto de la clase A desaparece el objeto de la clase B que lo compone también desaparecerá. Esto es lo que lo diferencia de la agregación simple, en la agregación, que se describe en la siguiente sección, un objeto de la clase B puede componer un objeto de la clase A pero puede existir a su vez de manera independiente. Una relación de este tipo no está contemplada en el modelo entidad relación desde el cual se mapea a una base de datos relacional. Se la representa a través de asociaciones que no reflejan exactamente el sentido de la composición. Existen dos maneras de representar una relación de composición en el paradigma objeto-relacional, mediante:

- a) tablas anidadas y
- b) arreglos.

Una tabla anidada implica que un atributo tabla de objetos de una clase puede ser una tabla de objetos de otra clase. Si se elimina, por ejemplo una fila de la tabla contenedora, también desaparecerá la tabla contenida, completamente. Las tablas anidadas tienen la ventaja que no imponen un límite en la cantidad de objetos que pueden anidarse, por un lado, pero plantea problemas de navegación y búsqueda de datos, dado que su acceso no es tan directo. Además, una tabla contenida en otra no es accesible por objetos que sean independientes y esto plantea problemas si se quiere establecer una asociación o algún otro tipo de relación con una tabla contenida. Un arreglo, por otro lado, si bien es fácil de consultar y permite mayores facilidades de navegación entre los objetos, tiene la dificultad de tener que limitar su tamaño, además la administración del contenido en cuanto a inserciones, eliminaciones y actualizaciones ya no se puede hacer tan fácil ni de manera directa. [DEP2003]

Las relaciones de composición que se presentaban en nuestro modelo están relacionadas directamente con la taxonomía.

Dadas las características de estas relaciones, se adoptaron las siguientes transformaciones:

- Si la relación es uno a uno, en la clase contenedora se crea un atributo del tipo contenido.
- Si la relación es uno a muchos, en el contenedor se crea una tabla anidada. Como ejemplos se puede presentar la estructura taxonómica, donde se tiene una clase que está compuesta por órdenes, éstos a su vez están compuestos por familias y éstas por géneros. [DEP2003]

Agregación

Como se expresó en la descripción de la composición, la agregación es una relación similar sólo que expresa una ligazón más débil, el objeto componente si bien compone a otro, puede existir de manera independiente. Esta relación tampoco está comprendida en el modelo entidad relación. Un ejemplo clásico que se emplea para ilustrar este concepto es la relación entre las clases persona y comisión, donde toda comisión está constituida por personas, pero las personas existen de manera independiente a la comisión. Otra propiedad de una agregación, y que se diferencia de otras relaciones que se pueden presentar en un modelo de objetos, es que el objeto padre no puede concebirse sin la existencia de los hijos (una comisión no tiene sentido sin las personas que la componen). Las alternativas para representar una agregación en el modelo objeto-relacional son las mismas que las de composición: tablas anidadas o arreglos de objetos.

Otra alternativa en este caso es la de agregar un arreglo de referencias y administrar la integridad de las mismas por medio de codificación. El empleo de un arreglo de objetos tiene la dificultad de definir el tamaño, que dadas las características de la aplicación podrían variar entre extremos muy amplios dificultando entonces su administración. Se decidió por el arreglo de referencias por no ser tan crítica la administración de su tamaño. [DEP2003]

Herencia

El manejo de herencia ya se había incorporado al modelo Entidad-Relación, y sus mapeos al modelo relacional están perfectamente definidos. Ninguna de estas transformaciones es natural, y la navegación y búsqueda de información en las tablas resultantes se presenta como adecuada. Las Bases de Datos Objeto-Relacionales ya presentan el concepto de herencia de clases y herencia de tablas y proveen un mecanismo de transformación directo y natural que nosotros adoptamos.

El empleo de la metodología de UML y de las Bases de Datos Objeto-Relacionales para el modelado de un Sistema de Información se está convirtiendo en un estándar en los últimos años. No existe una metodología que apoye la transformación de objetos de un modelo en objetos del otro. El diseñador de las aplicaciones debe hacer un análisis de cuáles son los mapeos que más se adaptan a los requerimientos del sistema que desarrolla. [DEP2003]

2.8. CALIDAD

A la hora de definir la calidad se debe diferenciar entre la calidad de producto software y la calidad del proceso de desarrollo de este (calidad de diseño y de fabricación) no obstante las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo ya que la calidad del primero va a depender entre otros aspecto de estos.

La ausencia de defectos, la aptitud, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados con el concepto de calidad de software. Sin embargo la calidad de software debe ser construida desde el comienzo no es algo que pueda ser añadido después.

2.8.1. METRICAS DE CALIDAD

La Calidad del software se define como, concordancia con los requisitos funcionales y de rendimiento explícitamente establecido con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera en todo software desarrollado profesionalmente.

Se definen en tres puntos:

- a) Los requisitos del software son la base de las medidas de la calidad. La falta de la concordancia con los requisitos es una falta de la calidad.
- b) Unos estándares especificados definen un conjunto de criterios de desarrollo que guían de manera en que se hacen la ingeniería de software. Si no se siguen los criterios, ha de existir seguramente poca calidad.

- c) Existe un conjunto de requisitos implícitos que a menudo no se nombran (tal es el caso de la facilidad de mantenimiento). Si el software cumple con sus requisitos explícitos pero falta en los implícitos, la calidad del software no será fiable.

La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según los clientes que la pidan.

2.8.2. FACTORES DE CALIDAD ISO-9126

El estándar ISO-9126 ha sido desarrollado en un intento de identificar los atributos clave de la calidad para el software. El estándar identifica seis atributos clave de calidad los cuales se describen a continuación:

- ◆ **Funcionalidad.** El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.
- ◆ **Confiabilidad.** Cantidad de tiempo que el software está disponible para su uso. Esta referido con los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- ◆ **Usabilidad.** Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- ◆ **Eficiencia.** Grado en que el software hace optimo el uso de recursos de sistema. Esta indicado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- ◆ **Facilidad de mantenimiento.** La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
- ◆ **Portabilidad.** La facilidad con la que el software puede ser llevado de un entorno a otro. Esta referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

2.9 ARQUITECTURA DEL SISTEMA

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes.

Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).

Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor.

Ambas partes deben estar conectadas entre sí mediante una red.

Una representación gráfica de este tipo de arquitectura sería la siguiente.

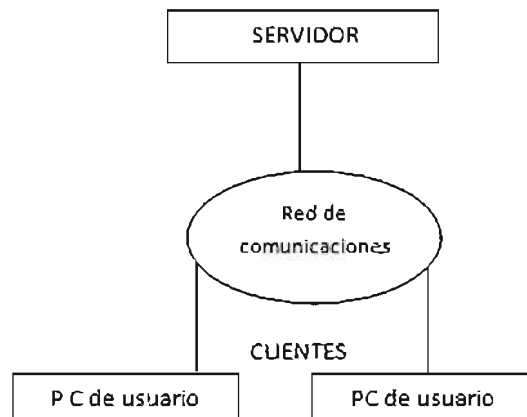


Fig.2.10 arquitectura cliente servidor
Fuente. [DWEB2007]

Este tipo de arquitectura es la más utilizada en la actualidad, debido a que es la más avanzada y la que mejor ha evolucionado en estos últimos años.

Podemos decir que esta arquitectura necesita tres tipos de software para su correcto funcionamiento:

- Software de gestión de datos: Este software se encarga de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente este software se aloja en el servidor.
- Software de desarrollo: este tipo de software se aloja en los clientes y solo en aquellos que se dedique al desarrollo de aplicaciones.
- Software de interacción con los usuarios: También reside en los clientes y es la aplicación gráfica de usuario para la manipulación de datos, siempre claro a nivel usuario (consultas principalmente).

A parte de estos existen más aplicaciones software para el correcto funcionamiento de esta arquitectura pero ya están condicionados por el tipo de sistema operativo instalado, el tipo de red en la que se encuentra, etc.

Según la metodología planteada, el proyecto será desarrollado según la línea del proceso del proceso unificado conocido comúnmente como UP, esta metodología utiliza el Lenguaje Unificado de Modelado para su notación, el cual representa todos los esquemas de un sistema de software.

El proceso unificado de desarrollo describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica rápida para diseñar y probar el sistema de acuerdo a los requerimientos y la arquitectura. El proceso de desarrollo UP se describe en: fases, actividades, artefactos, trabajadores y flujos de trabajo que guiaran este proyecto.

3.1 PLAN DE DESARROLLO DE SOFTWARE

El objetivo principal de esta primera parte es desarrollar un plan de desarrollo de software basado en la metodología UP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos que serán generados. Tomaremos en cuenta que el estudio preliminar se lo realizó en la primera parte de este documento, concretamente en la identificación del problema, objetivo general, objetivos específicos, justificaciones, alcances y límites.

En esta sección se presenta la organización en fases e iteraciones. El desarrollo del proyecto se llevará a cabo en base a las cuatro fases detalladas anteriormente con una o más iteraciones en cada una de ellas. La siguiente tabla de muestra la distribución de tiempos y el número de iteraciones de cada fase.

Id	TAREA	Comienzo	Fin	Duración	May 2009		Jun 2009			Jul 2009			Ago 2009			Sep 2009			Oct 2009			Nov 2009																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
					01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03

3.2 FASE DE INICIO

3.2.1 MODELADO DEL NEGOCIO

Basado en el estudio preliminar de la institución se procede a identificar y describirá cada uno de los procesos de negocios, determinando el tipo de información actividades roles y reglas del negocio implicadas. Con esto lo que se pretende es comprender toda la actividad de la institución relacionada con el sistema a implantar.

3.2.1.1 IDENTIFICACION DE PROCESOS DEL NEGOCIOS

El modelado de negocios se inicia identificando los principales objetivos estratégicos de la institución (Unidad de activos fijos del SSU) y los procesos de mayor relevancia relacionados a estos objetivos.

- a) Realizar de forma eficiente todo el proceso de administración de los Activos Fijos del SSU.
- b) Realizar inventario anual actualizado.
- c) Realizar Inventarios Individuales de Responsabilidades.
- d) Realizar Acta de transferencias y de entrega.
- e) Solicitud de compra

3.2.1.2 IDENTIFICACION DE ACTORES DEL NEGOCIO

A continuación se identifica y describe los actores del negocio correspondientes a la Empresa.

- **Encargado de Activos Fijos**, Tiene la función de controlar todo el movimiento de los activos existentes en SSU desde su recepción, registro, asignación, baja, etc. A su vez también está a cargo del cálculo de actualización y depreciación además de la emisión de reportes y actas respectivas.
- **Gerente Administrativo financiero** Es a Quien se emite los reportes sobre el movimiento y estado de los activos fijos. Revisa y firma la conformidad de devuelve al encargado de activos fijos

- **Funcionario** es el personal que pertenece a las distintas unidades o direcciones de la SSU. Es también a quien se le asigna el activo y dispone de los mismos para la realización de las funciones y responsable directo del activo.
- **Administrador del sistema**, es el personal que se encuentra a cargo de las funciones especializadas de administración del sistema mismo que se encuentra en la dirección administrativa.
- **Proveedor**, es la persona física o jurídica que se encarga de proveer de activos fijos cuando requiera la SSU.
- **Unidad**, son las direcciones que se encuentran en el SSU y que interactúan con la unidad encargada de activos fijos.
- **Cotizador**, es la persona encargada de obtener las cotizaciones de los activos fijos solicitados por el SSU.
- **Encargado de Bienes y Servicios** Realizada la compra, remite a la unidad de activos fijos copia de solicitudes de cada unidad en las que se incluyen especificaciones técnicas de los bienes adquiridos, mediante procedimientos y normas vigentes. Confirmando la entrega del bien.

3.2.1.3 CASOS DE USO DEL NEGOCIO

La identificación de los casos de Uso así también como su respectiva descripción se la detalla a continuación.

Solicitar pedido de activos Fijos.- El funcionario realiza la solicitud de activo al encargado de activos fijos y este último en caso de existir carencia de bienes solicitados remitirá la solicitud de bienes al encargado de bienes y servicios.

Adquirir Activo Fijo.- El Encargado de Bienes y Servicios es la que se encarga de la cotización y la respectiva compra de los activos fijos para luego delegar estos bienes al encargado de activos fijos.

Remitir Solicitud de Activos Fijos.- El Encargado de activos fijos remite la solicitud de activos a la dirección de Bienes y Servicios del S.S.U.

Asignar Activo Fijo.- El Encargado de activos fijos del S.S.U. realiza la asignación de activos fijos al funcionario solicitante. Este último será el custodio y responsable del bien.

Clasificar Activo Fijo.- El Encargado de activos fijos es la persona que clasifica los bienes según el grupo y subgrupos contables al cual corresponda el activo fijo indicado.

Actualizar Inventario.- El encargado de activos fijos después de la adquisición de estos actualiza el inventario.

Devolución de Activos.- El funcionario del S.S.U. procede a la devolución del o de los ACTIVOS FIJOS que están en su custodia para entregarlos al encargado de activos fijos del S.S.U.

Recepción del Activo Fijo.- El encargado de activos Fijos se encarga de verificar las características del activo posteriormente es codificado y registrado.

Codificación.- El encargado de activos realiza la codificación correlativa una vez que el activo se haya recepcionado.

Actualización y Depreciación.- El encargado de activos fijos realiza al cierre de gestión la actualización y depreciación de todos los activos

Baja del Activo.- El activo es dado de baja por el encargado de activos fijos y es registrado cuando ya no es apto para las funciones que desempeña.

Devoluciones.- El responsable del bien o custodio devuelve el activo a su cargo, el encargado de activos fijos recepciona y registra la devolución.

3.2.1.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO

A continuación se realiza la identificación de los diferentes procesos de negocio de la entidad. El siguiente grafico muestra los casos de uso del modelo de negocio además se desglosan estos en modelos de objetos respectivos

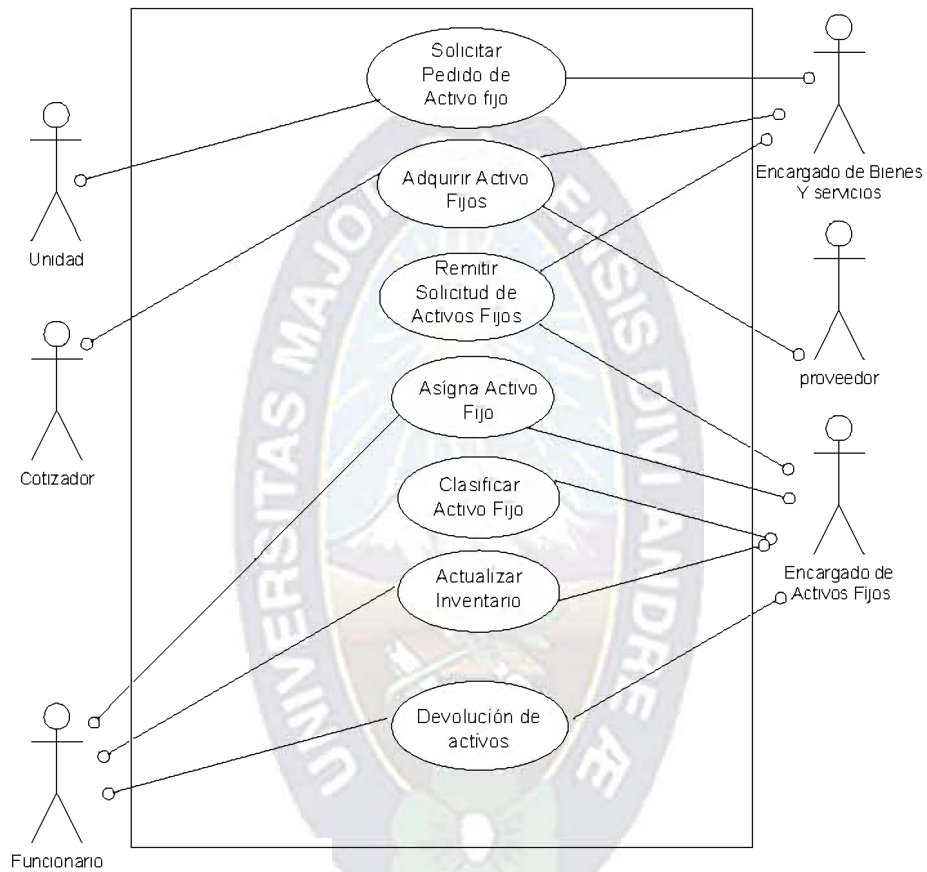


Fig. 3.1 Modelos de casos de uso del negocio
Fuente: Elaboración Propia

3.2.2 DIAGRAMA DE ACTIVIDADES

El diagrama de actividades muestra el flujo de actividades entre los elementos del dominio mismo que se puede apreciar en el siguiente grafico.

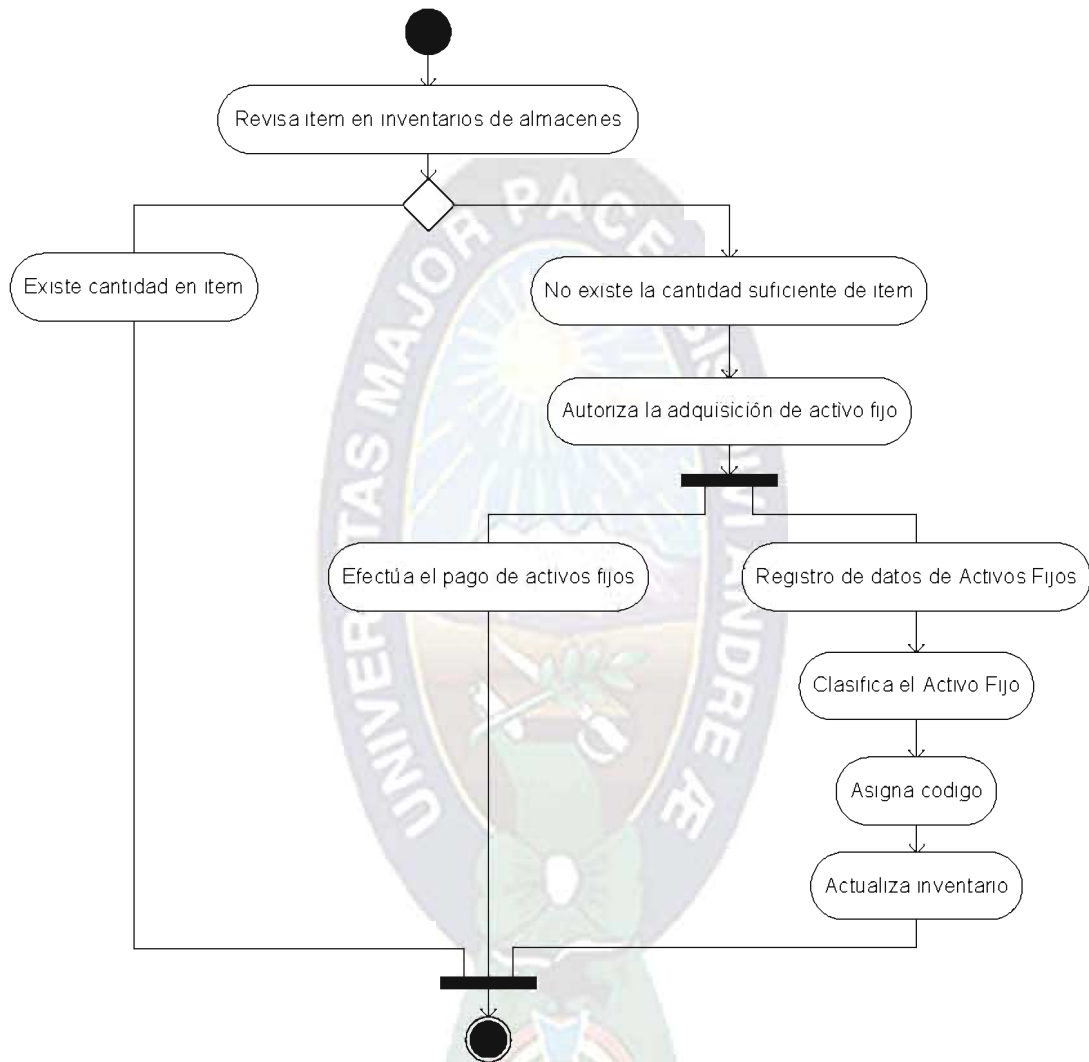


Fig.3.2 Diagrama de actividades "Adquisición de bienes"
Fuente: Elaboración Propia

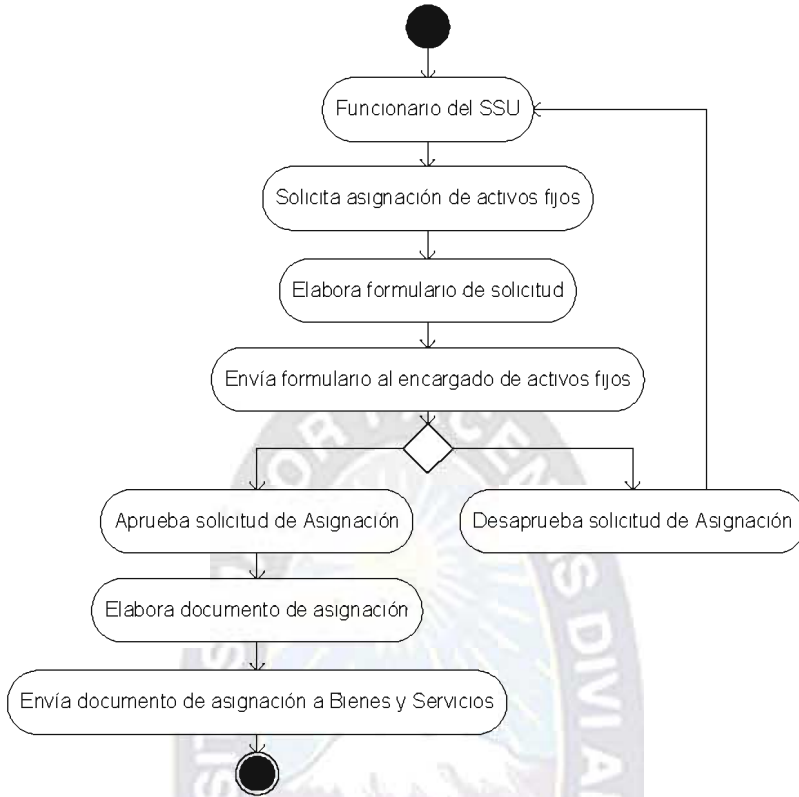


Fig.3.3 Diagrama de actividades "Solicitud de Bienes"
 Fuente: Elaboración Propia



Fig.3.4 Diagrama de actividades "Ingreso de Bienes"
 Fuente: Elaboración Propia

3.3 MODELADO DE REQUISITOS

El modelo de requisitos es donde se establecen los requisitos funcionales y no funcionales del sistema, mismos que facilitan el análisis y diseño del sistema.

3.3.1 REQUISITOS FUNCIONALES

La siguiente tabla detalla las funciones elementales o requerimientos que deberán ser satisfechos para el usuario final por el sistema a implementar.

Nro.	FUNCION	CATEGORIA
R1	Control de ingreso para usuario	Evidente
R2	Registro de Usuarios habilitados	Evidente
R3	Registro de ingreso de activos fijos	Evidente
R4	Asignación de código Identificador de activo fijo	Oculto
R5	Registro de datos complementarios de activo fijo	Evidente
R6	Captura de información sobre el activo registrado	Evidente
R7	Asignación de activos fijos a unidad o personal solicitante	Evidente
R8	Registro de proveedores	Evidente
R9	Muestra la descripción y el precio de activos fijos registrados	Evidente
R10	Registro de devolución de activos fijos	Evidente
R11	Informe de activos fijos por unidad y dirección	Evidente
R12	Verificación de la existencia bienes para la asignación	Evidente
R13	Realizar el cálculo de actualizaciones en valor de los activos fijos	Evidente
R14	Realizar el cálculo de depreciaciones en valor de los activos fijos	Evidente

R15	Obtener el valor actualizado de los activos fijos en custodia	Evidente
R16	Generar el informe financiero de los activos fijos registrados	Evidente
R17	Realizar los cálculos de actualización depreciación y valor actual en distintos tipos de moneda	Oculto
R18	Obtener reporte de las asignaciones realizadas de los activos fijos por responsable	Evidente
R19	Registrar la baja de activos fijos y su correspondiente destino	Evidente
R20	Generar el historial del seguimiento de los activos fijos asignados	Evidente
R21	Generar informes de proveedores	Evidente
R22	Generar informe de activos por grupo fecha de ingreso o monto adquirido	Evidente
R23	Generar informes de mantenimiento	Evidente

Tabla.3.2 Definición de requisitos funcionales del sistema
Fuente: Elaboración propia

3.4 REQUISITOS NO FUNCIONALES

Para el correcto funcionamiento el sistema informático que se está desarrollando deberá cumplir con los siguientes requerimientos no funcionales en el momento de implantar el sistema, mismos que se detallan a continuación:

- Windows como sistema operativo al lado del usuario del sistema.
- Microsoft net frameworks 3.1 al lado del usuario del sistema.
- En el lado de servidor deberá estar instalado el gestor de base de datos Microsoft SQL server 2005.
- La información almacenada deberá ser capaz de ser accedida en otros equipos donde se encuentre instalada la aplicación, conectados dentro de la SSU.
- La aplicación no deberá consumir demasiado tiempo a la hora de acceder a la misma.

3.5 GLOSARIO DE TERMINOS

El glosario de términos define e incluye todos los términos además de una descripción textual de los elementos del modelo, esto para evitar que exista la posibilidad de ambigüedad en la comprensión de los términos utilizados.

Se define la siguiente tabla de términos utilizados con el fin de registrar todos los términos de usados en las fases siguientes.

TERMINOS	CATEGORIA	DESCRIPCION
Activo Fijo	TIPO	Bien de uso con el cual se cuenta dentro de la institución.
Actualización de activo	CASO DE USO	Es la valoración de una activo fijo de acuerdo a una determinada fecha y cotización deSSU su fecha de adquisición
Ajuste de Activo	CASO DE USO	Es la valoración de un determinado activo fijo de acuerdo las normas de contabilidad
Asignación de activo fijo	Caso de uso	Es la asignación de un bien a un funcionario dentro de una unidad de dirección.
Depreciación de activo	Caso de uso	Es el cálculo de pérdida de valor que sufre un bien a causa del uso que se le da conforme el tiempo transcurrido.
Edificio	Tipo	Es un tipo de activo fijo
Equipo de computación	Tipo	Es un tipo de activo fijo
Equipo de comunicación	Tipo	Es un tipo de activo fijo
Funcionario	Actor	Es una determinada persona física cuya relación es laboral dentro la institución.

Proveedor	Actor	Es una determinada persona física cuya relación con la institución es la de proveer bienes a la misma.
Encargado de activos	Actor	Es una determinada persona física cuya relación dentro la institución es la de realizar gestionar y controlar el manejo de bienes de la institución.
Gerente administrativo financiero	Actor	Es una determinada persona física cuya relación dentro la institución es la de realizar la gestión administrativa financiera en la institución.
Historial	Tipo	Es el registro de todos los procesos a los cuales fue sometido un determinado bien a lo largo de su vida útil.
Registro de proveedores	Casos de uso	Es la recopilación de los datos correspondientes a los proveedores y que el sistema requiere para gestionar el origen de los bienes
Solicitud de asignación de activos	Casos de uso	Es el proceso por el cual se inicia la solicitud y posterior asignación de de los bienes solicitados al encargado de bienes.
Saldo de artículos	Tipo	Es el correspondiente informe de cantidades existentes de los bienes por parte del encargado de activos fijos.
Baja de activos fijos	casos de uso	Es el proceso que se efectúa para el retiro de un bien en función de una determinada causa detectada.
Adición por revaluó técnico	casos de uso	

Clasificación de activos fijos	casos de uso	
Grupo	Tipo	
Sub grupo	Tipo	

Tabla: 3.3 glosario de términos
Fuente: Elaboración Propia

3.6 MODELO CONCEPTUAL

Una parte de la investigación del dominio del problema consiste en identificar los conceptos que los conforman. La base sobre la cual se puede identificar estos conceptos es en la especificación de requisitos además del conocimiento que se tenga sobre el entorno en el cual se basan los mismos.

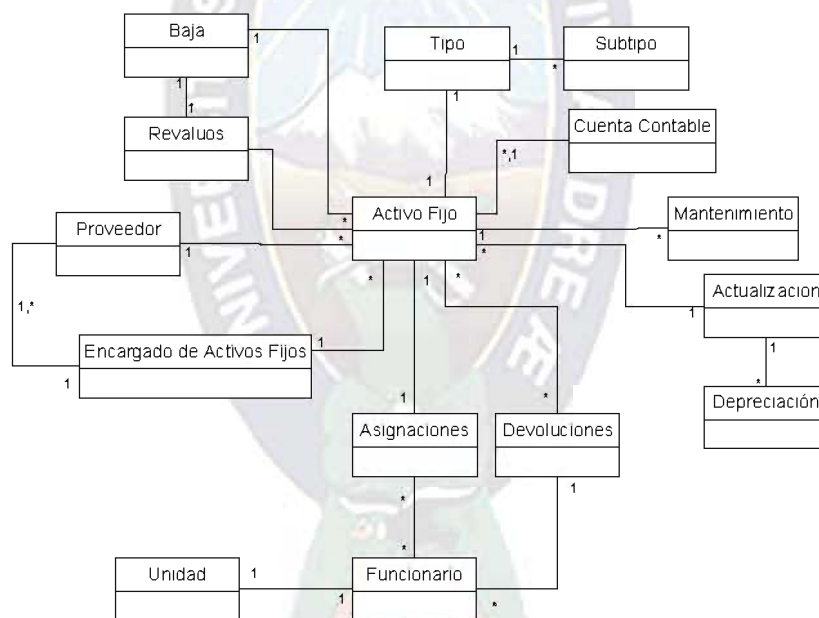


Fig.3.5 Modelo Conceptual asociado al seguimiento y control de Activos Fijos
Fuente: Elaboración Propia

3.7 FASE DE ELABORACION

3.7.1 MODELO DE CASOS DE USO

3.7.1.1 CASOS DE USO ESENCIALES

En función de los casos de uso ya especificados en la etapa inicial además de los nuevos casos de uso, mismos que se describen a continuación corresponderán a los casos de uso esenciales y serán definidos en formato expandido para una mejor comprensión.

- Registrar activos fijos el encargado de activos fijos o el usuario quien tiene el privilegio de registro son los responsables de ingresar los activos al sistema por medio del registro. Para que esta funcionalidad pueda efectuarse se deberá tener ya identificado la clasificación de activos, su correspondiente valor contable, el proveedor correspondiente y el documento respaldatorio
- Asignar activos fijos
- Generar actas de asignación
- Devolución de activos fijos
- Generar actas de devolución
- Calcular actualización y depreciación
- Baja de activos fijos
- Adicionar activos revaluados
- Actualización de cotizaciones
- Actualizar datos del proveedor
- Administrar sistema el administrador es el encargado de realizar el mantenimiento y/o actualización de la base de datos y de los sistemas de los cuales está a cargo

Control de accesos el control de accesos está en función a los privilegios asignados en el modulo al cual desea acceder. Este privilegio es asignado por el administrador del sistema

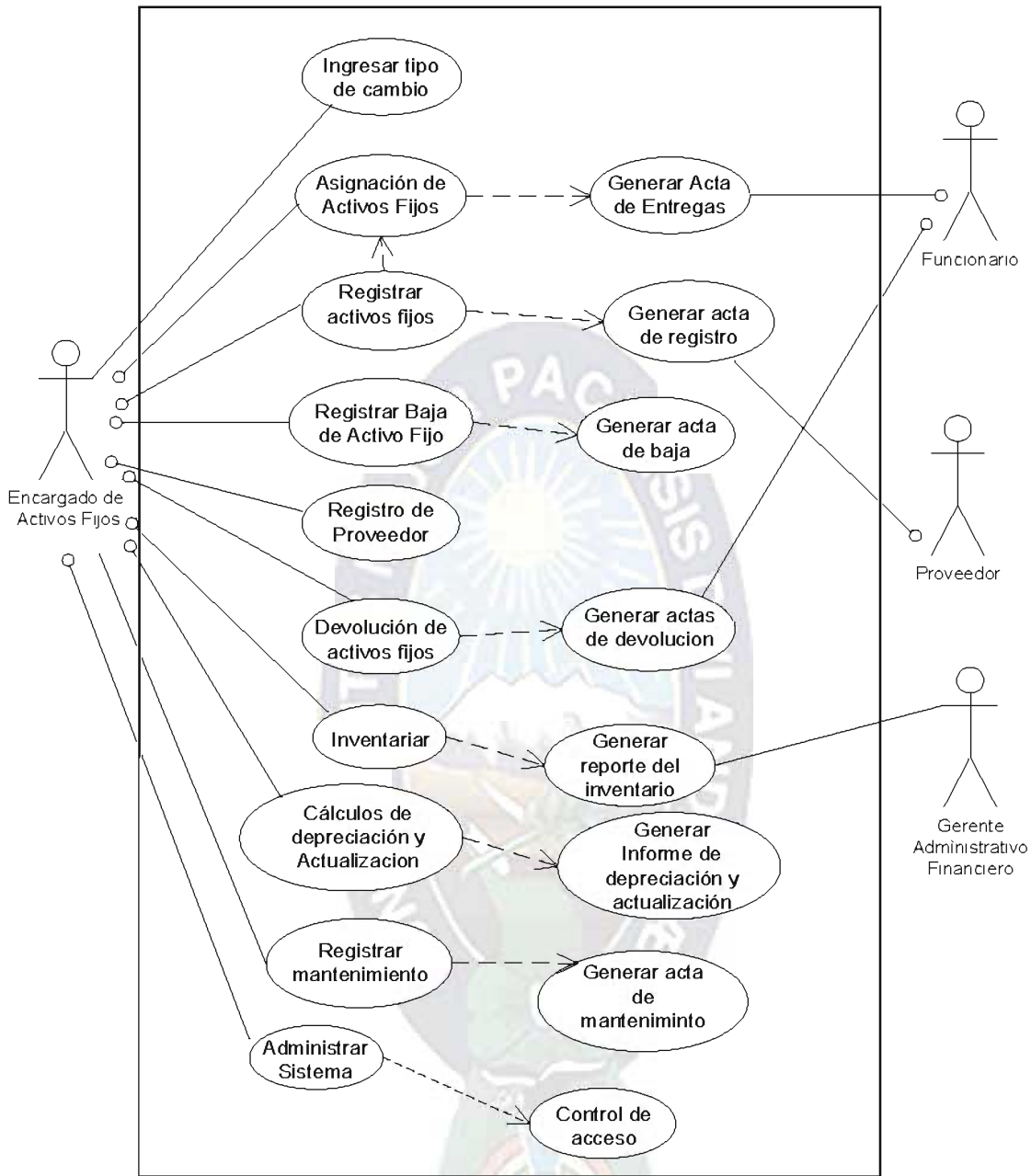


Fig. 3.6 Diagrama de Casos de Uso para el sistema de activos fijos
Fuente: Elaboración Propia

3.7.1.2 DIAGRAMAS DE CASOS DE USO EXPANDIDOS

El siguiente diagrama corresponde al caso de uso esencial del “sistema de información de seguimiento de activos fijos“. A partir de esta especificación se detalla los diagramas de casos de uso expandido.



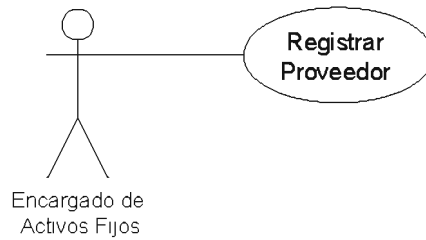
NOMBRE DEL CASO DE USO	ASIGNACIÓN DE DE ACTIVOS FIJOS	
ACTORES	Encargado de Activos Fijos, Funcionario del S.S.U.	
RESUMEN	El encargado de Ac. Fijos en función del formulario de solicitud de activos procede a la asignación del activo como constancia de esa asignación se imprime un acta de asignación correspondiente.	
PROPOSITO	Asignar un activo a un funcionario.	
REFERENCIAS CRUZADAS	R7,R12,R18	
FLUJO NORMAL DE EVENTOS	ACTORES 1.- Encargado de A.F. selecciona el funcionario. 3.- Encargado Elige o Busca Activos disponibles. 4.- Realizar Asignación.	SISTEMA 2.- Devuelve datos del empleado. 5.- Genera reporte de asignación.
FLUJO ALTERNATIVO	- En 1 si no se encuentra funcionario se deberá actualizar la tabla de empleados. - En 3 los bienes disponibles son los que se despliegan en caso de no encontrar deberá verificar el estado del bien.	

Tabla: 3.4 caso de uso Asignación de Activos Fijos
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Registrar Activo Fijo	
ACTORES	Encargado de Activos Fijos, Proveedor	
RESUMEN	Se registra el bien con sus características respectivas para luego entregar un comprobante al proveedor que lo trae.	
PROPOSITO	Registrar Los datos correspondientes a un determinado Activo Fijo en la base de datos.	
REFERENCIAS CRUZADAS	R4,R5,R6	
FLUJO NORMAL DE EVENTOS	ACTORES 1.- El encargado elige la opción nuevo Activo Fijo. 3.- Se Elige grupo y subgrupo. 4.- Se llena datos correspondientes a respectivo bien. 5.-Una vez confirmados los datos se completa el registro. 8.- Se le entrega una copia del acta de registro.	SISTEMA 2.- El sistema carga los datos correspondientes al Registro. 6.- El sistema Almacena los datos del nuevo Activo. 7.- Se genera el acta de registro.
FLUJO ALTERNATIVO	- En 3 si no se encuentra el grupo y el subgrupo al cual pertenece el activo se deberá adicionar el grupo i el subgrupo. - en 4 si los datos del proveedor no existen se debe actualizar el registro de proveedores.	

Tabla: 3.5 caso de uso Registrar Activo Fijo
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Registrar Proveedor	
ACTORES	Encargado de Activos Fijos	
RESUMEN	Se recopila datos del proveedor que el sistema requiere para el origen del activo.	
PROPOSITO	Registrar un proveedor para saber la procedencia del bien	
REFERENCIAS CRUZADAS	R8	
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- El encargado elige la opción registrar proveedor	
	2.- El encargado llena todos los campos para el registro	
	3.- Termina el registro	4.- Mensaje de confirmación de registro
FLUJO ALTERNATIVO	- En 2 para hacer efectivo el registro todos los campos deben estar llenos con excepción del correo que puede ser nulo	

Tabla: 3.6 caso de uso Registrar proveedor
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Registrar Baja de Activo Fijo	
ACTORES	Encargado de Activo Fijo	
RESUMEN	Se procede a dar de baja al activo según en la condición que se encuentre	
PROPOSITO	Efectuar la baja de activo Fijo en condiciones malas	
REFERENCIAS CRUZADAS	R19	
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- El encargado ingresa al sistema.	2.- El sistema carga datos de los activos que no tienen baja
	3.- El encargado elige el activo a ser dado de baja	
	4.- El encargado llena los datos requeridos por el sistema	
	5.- El encargado termina el registro	6.- El sistema muestra el reporte de baja
FLUJO ALTERNATIVO	- En 2 si el bien no se encuentra se procederá a la verificación del código.	

Tabla: 3.7 caso de uso Registrar baja de Activos Fijos
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Devolución de Activos Fijos	
ACTORES	Encargado de activos fijos, Funcionario	
RESUMEN	El funcionario Devuelve el bien que el encargado de Activos Fijos le asigno anteriormente.	
PROPOSITO	Efectuar la devolución del activo fijo, generando el acta correspondiente.	
REFERENCIAS CRUZADAS	R10	
FLUJO NORMAL DE EVENTOS	ACTORES 1.- El funcionario Solicita la devolución del bien 3.-El encargado de activos fijos elige el activo a ser devuelto. 4.- El encargado de Activos Fijos Realiza la devolución.	SISTEMA 2.- El sistema carga los datos de funcionario 5.- El sistema muestra el reporte de devolución.
FLUJO ALTERNATIVO	- En 2 el funcionario no se encuentra registrado losa bienes no podrán ser mostrados y se debe verificar el código del funcionario	

Tabla: 3.8 caso de uso Devolución De Activos Fijos
Fuente: Elaboración propia

Para una mayor descripción de estos casos de uso, en el ANEXO B se pueden encontrar los restantes casos de uso que corresponden al sistema de información SICAF.

3.8 MODELO DE DISEÑO

3.8.1 DIAGRAMAS DE SECUENCIA

En los siguientes diagramas de secuencia se observa las operaciones proporcionadas por el SICAF mismos que son demandadas por los actores del sistema. En estos diagramas se muestra la interacción de los actores que fueron identificados y mencionados en este documento con los objetos del diseño a través de la secuencia de mensajes entre los mismos

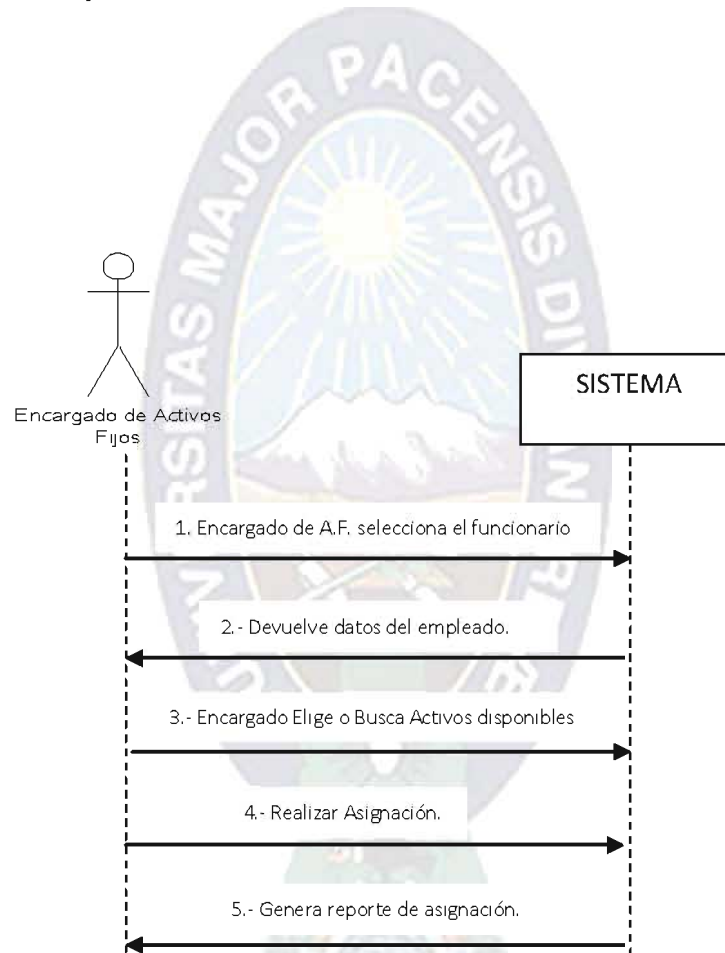


Fig.3.7 Diagrama de Secuencia Asignación de Activos Fijos
Fuente: Elaboración Propia

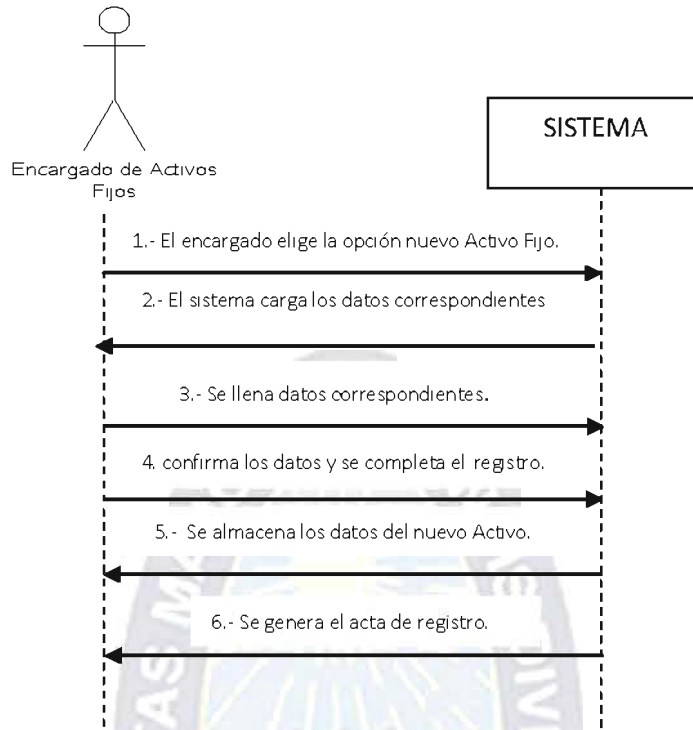


Fig.3.8 Diagrama de Secuencia Registrar activo fijo
Fuente: Elaboración Propia

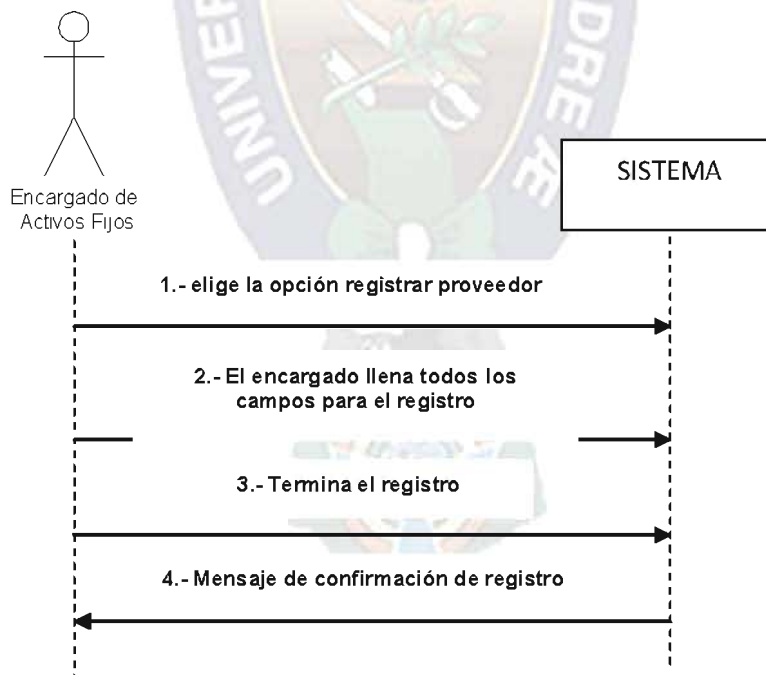


Fig.3.9 Diagrama de Secuencia Registrar proveedor
Fuente: Elaboración Propia

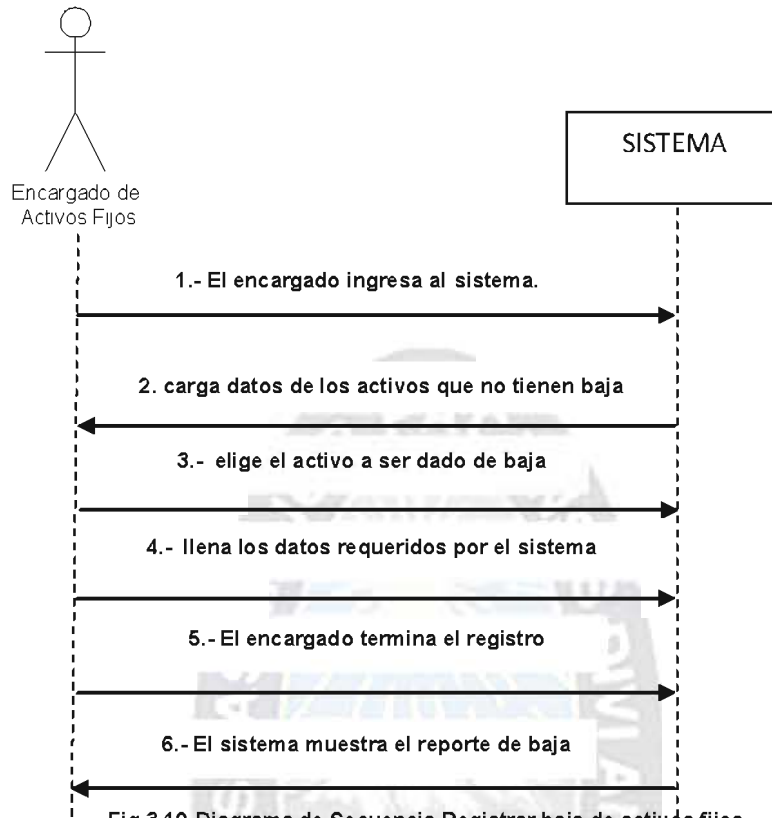


Fig.3.10 Diagrama de Secuencia Registrar baja de activos fijos
Fuente: Elaboración Propia

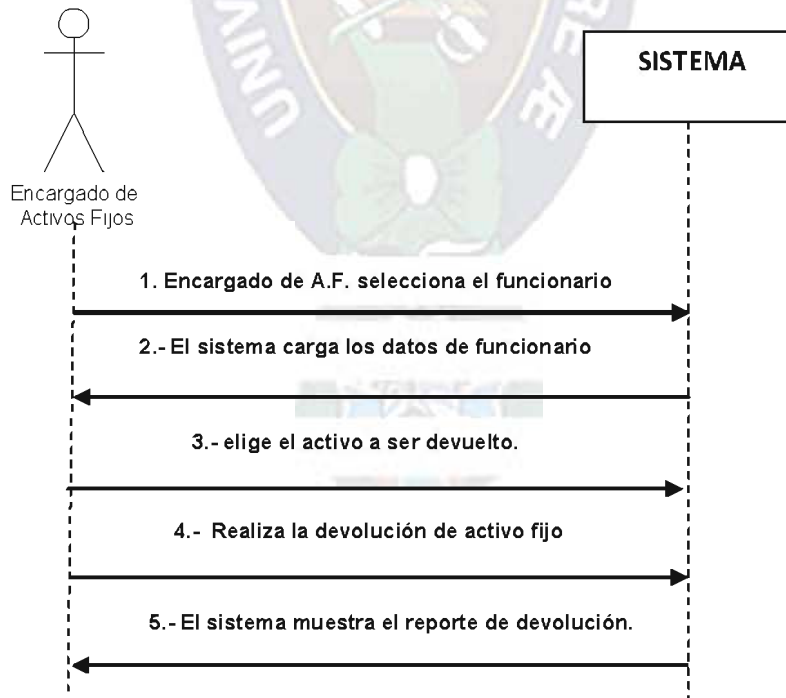


Fig.3.11 Diagrama de Secuencia Devolución de activos fijos
Fuente: Elaboración Propia

En el ANEXO C se pueden encontrar los restantes diagramas de secuencia que corresponden al sistema de información SICAF.

3.9 CONTRATOS

- ASIGNACIÓN DE ACTIVOS FIJOS

CONTRATO	
NOMBRE	Encargado de A.F. selecciona el funcionario
RESPONSABILIDAD	Se verifican datos del funcionario
TIPO	Sistema
REFERENCIAS	R7
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 3.9 Contrato Encargado de A.F. selecciona el funcionario
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se despliegan los datos del empleado
RESPONSABILIDAD	Se muestran los datos del funcionario a asignar
TIPO	Sistema
REFERENCIAS	R7
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 3.10 Contrato Se despliegan los datos del empleado
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Selección de activos disponibles
RESPONSABILIDAD	Se muestran activos disponibles para la asignación
TIPO	Sistema
REFERENCIAS	R7
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 3.11 Contrato Selección de activos disponibles
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Registro de la asignación
RESPONSABILIDAD	Se guardan los datos de la nueva asignación
TIPO	Sistema
REFERENCIAS	R7
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 3.12 Contrato Registro de la asignación
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se muestra un reporte de la nueva asignación.
RESPONSABILIDAD	Se genera un informe de asignación.
TIPO	Sistema

REFERENCIAS	R7
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 3.13 Contrato Se muestra un reporte de la nueva asignación
Fuente: Elaboración propia

Para una mayor descripción de estos contratos , en el ANEXO D se pueden encontrar los restantes contratos que corresponden al sistema de información SICAF

3.10 DIAGRAMAS DE ESTADO

Los siguientes diagramas de estado nos permiten observar los estados y los eventos más interesantes de entre los objetos definidos que intervienen en el modelo como también el comportamiento del sistema de información a partir de los casos de uso de alto nivel del Sistema SICAF detallado en el punto anterior (definición de los diagramas de uso esenciales)



Fig.3.12 Diagrama de Estado Asignación de Activos fijos
Fuente: Elaboración Propia

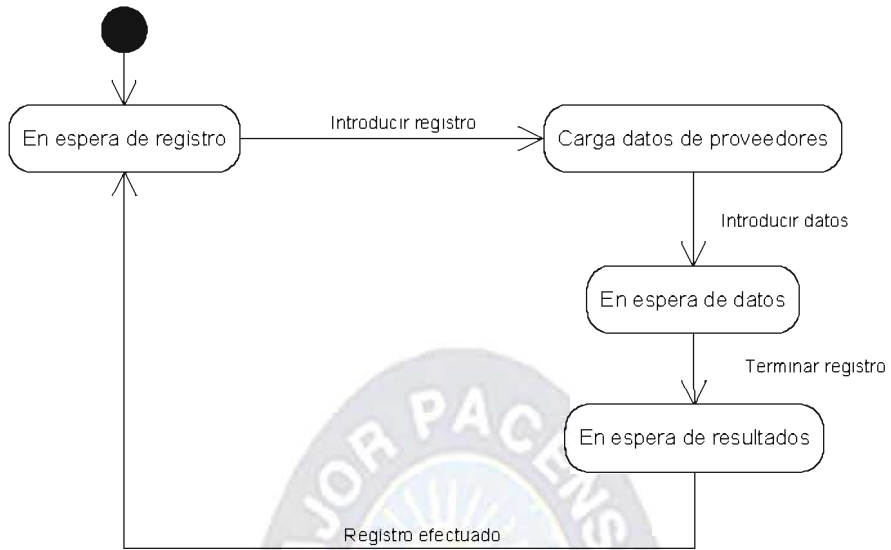


Fig.3.13 Diagrama de Estado Asignación de Activos fijos
Fuente: Elaboración Propia

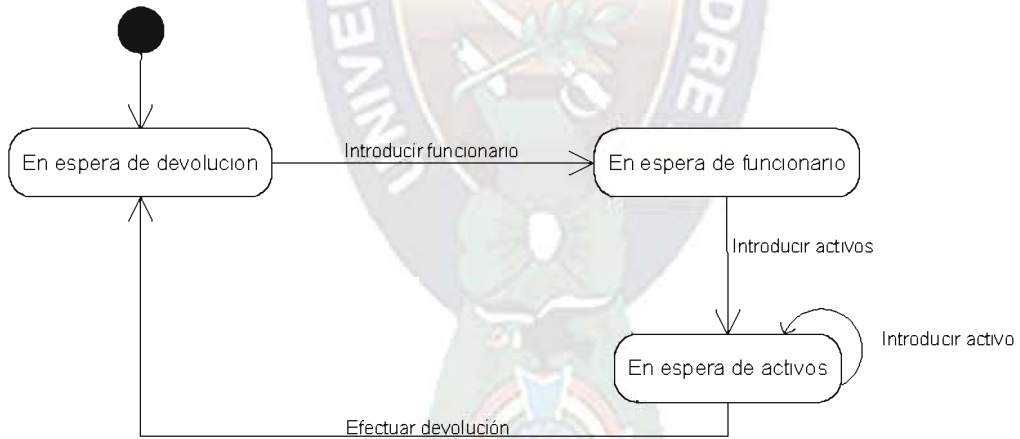


Fig.3.14 Diagrama de Estado Asignación de Activos fijos
Fuente: Elaboración Propia

3.11 DIAGRAMA DE CLASES

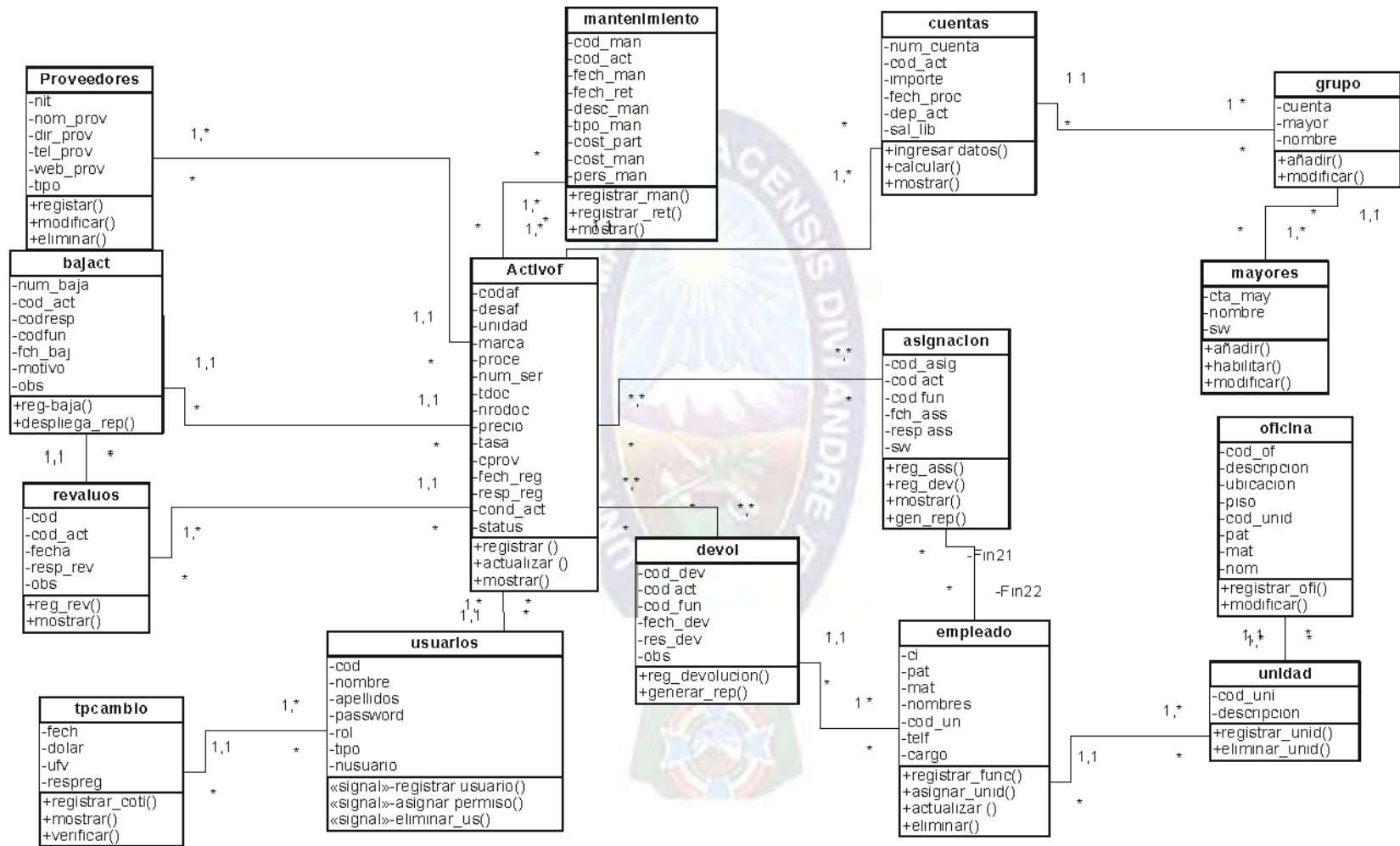


Fig. .3.15 Diagrama de clases del sistema SICAF
Fuente: Elaboración propia

3.12 DESARROLLO DE LA ARQUITECTURA DEL SISTEMA

El diseño de la arquitectura diseñada para el Sistema es el de Cliente/Servidor, el cual consta de un Servidor de Base de Datos SQL Server 2005 al cual se conectan todas las terminales mediante intranet (red local), la unidad de activos fijos tendrá acceso al Sistema como se muestra en la siguiente figura.

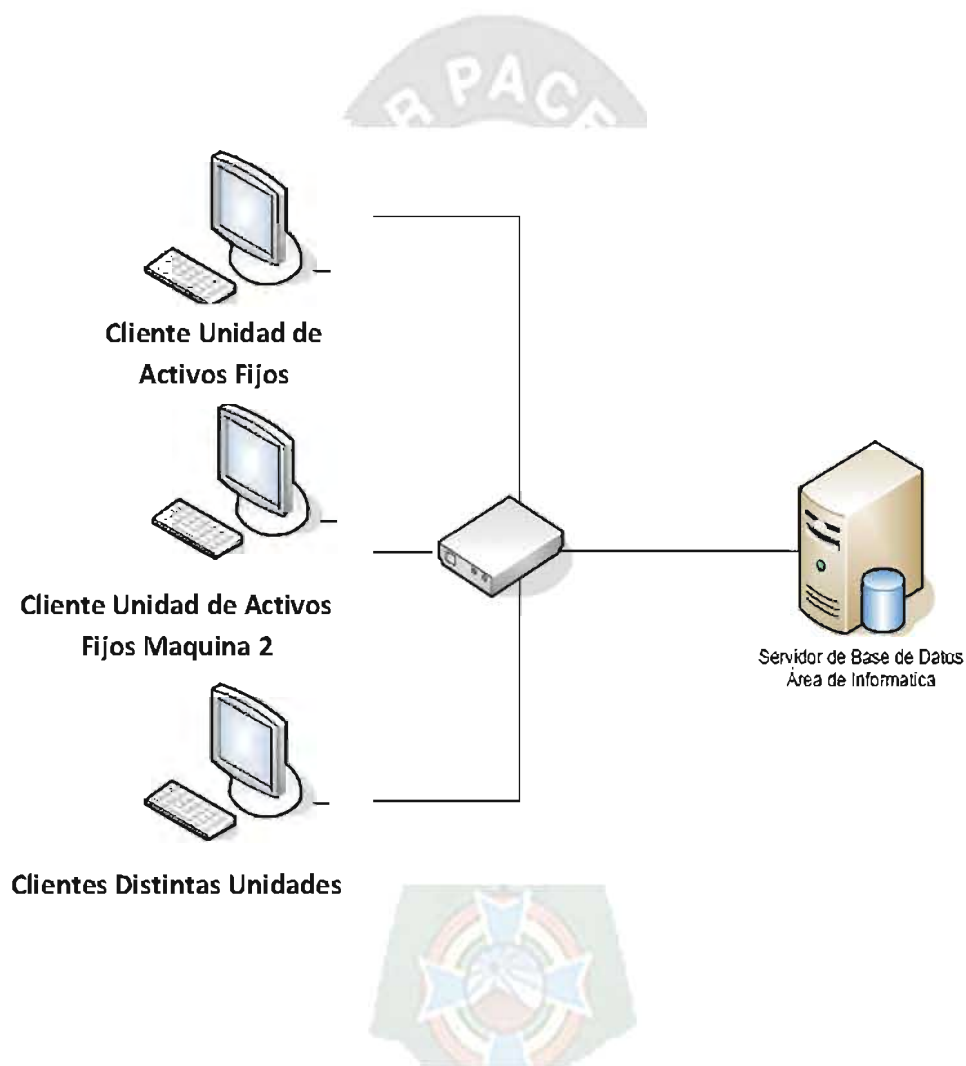


FIGURA 3.16 INTRANET DEL SEGURO SOCIAL UNIVERSITARIO
FUENTE: Elaboración Propia

3.13 TRANSFORMACION DEL MODELO OO AL MODELO ER

El proceso de transformación del diagrama de clases definido para el sistema de Activos Fijos se detalla a continuación.

- Clases.- clase fuerte: activo fijo, cuentas, tipo de cambio, funcionario y unidad. Clase débil: Usuario, y grupo.
- Asociaciones.- asignaciones, devoluciones y documentos son entidades con atributos de referencia.
- Atributos.- los atributos se mantienen en cada entidad
- Generalización.- se realiza la transformación para la clase grupos que reproduce todos los atributos

Dentro de la base de datos del sistema de información SICAF las tablas que son necesarias alimentar con datos son: activo fijo, Proveedor, Mayores, grupo, usuario, empleado.

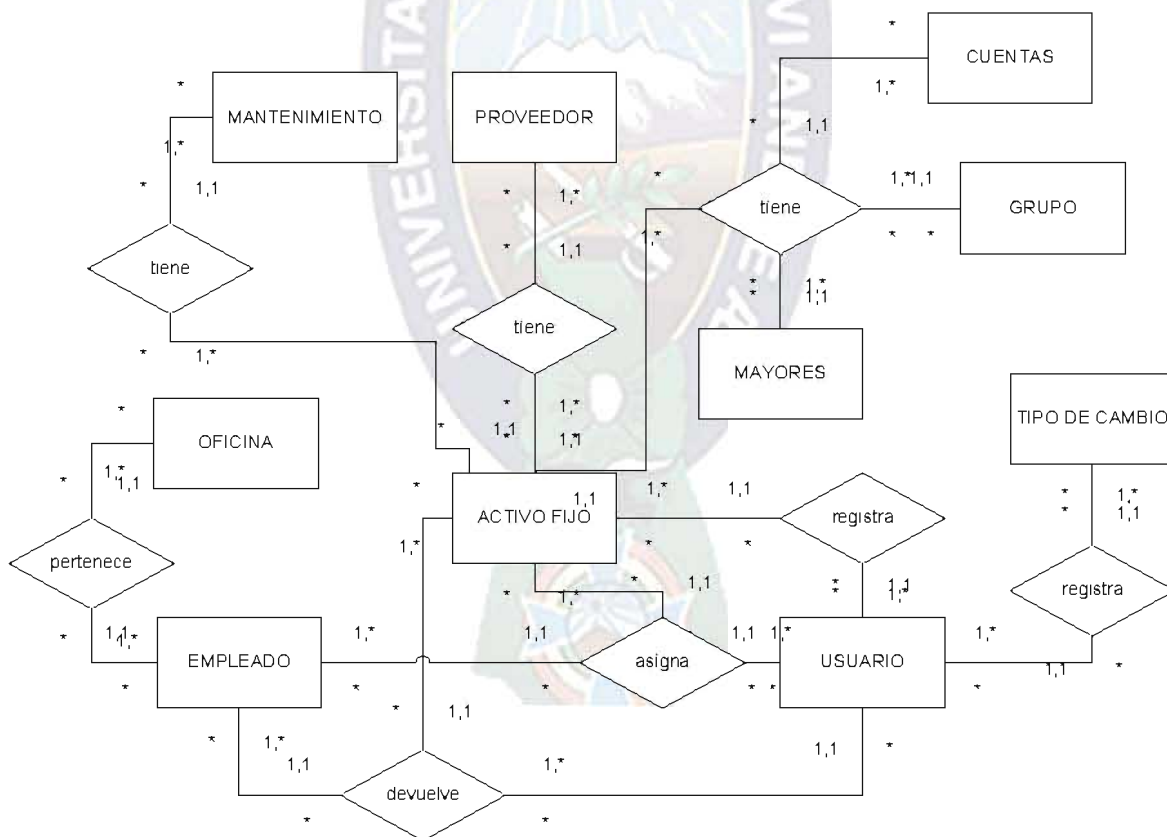


Fig. .3.17 Diagrama Entidad Relación del sistema SICAF
Fuente: Elaboración propia

3.14 MODELO RELACIONAL

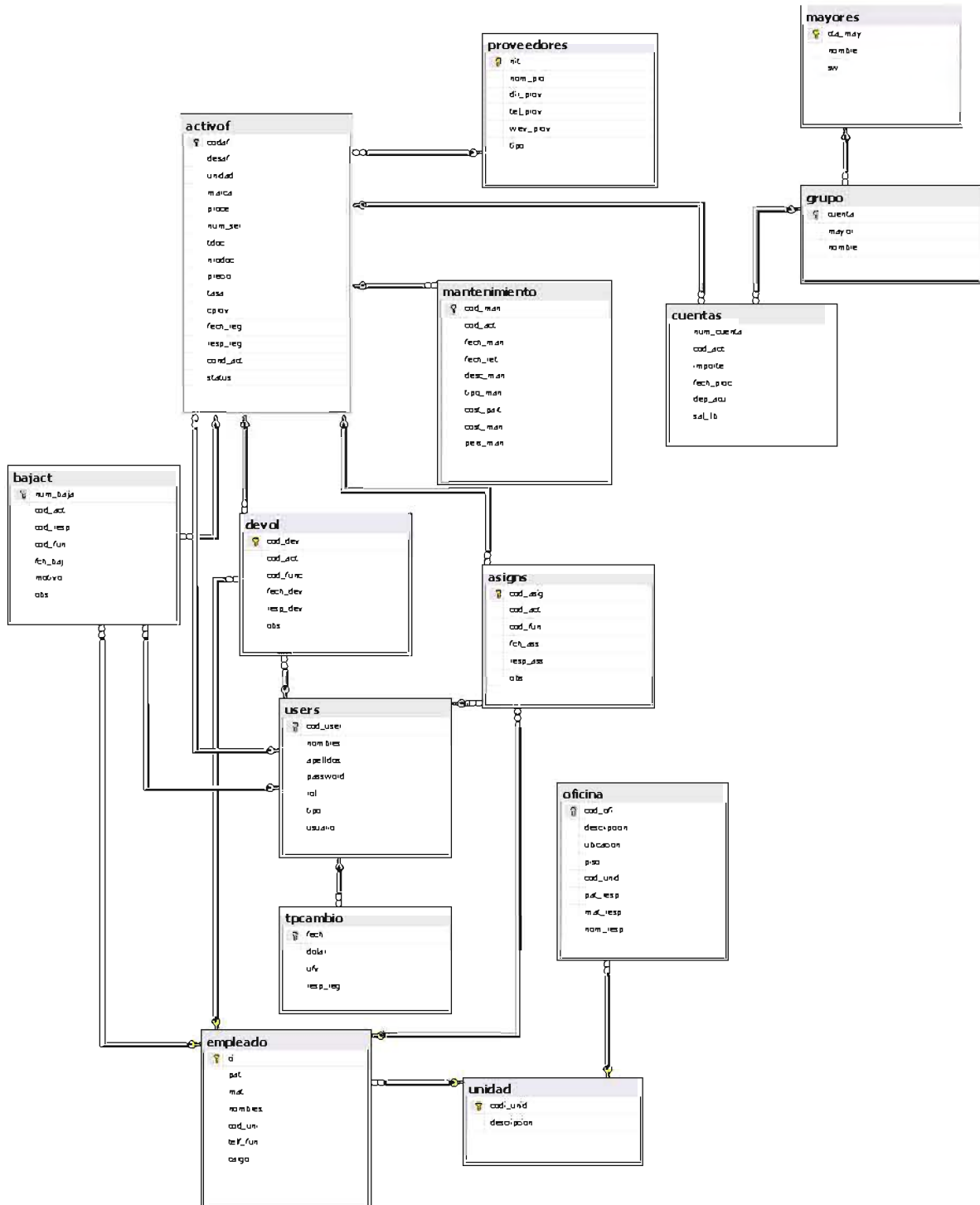


Fig. 3.18 Modelo Relacional del sistema SICAF
Fuente: Elaboración propia

3.15 ESQUEMAS DE BASE DE DATOS

A continuación se muestra algunas tablas del modelo relacional que fueron normalizadas hasta la cuarta forma normal.

tabla de los activos fijos

activof				
	Nombre de columna	Tipo de datos	Permitir valores nulos	Descripción
	codaf	varchar(10)	<input type="checkbox"/>	codigo del activo
	desaf	varchar(200)	<input type="checkbox"/>	descripcion del activo
	unidad	varchar(20)	<input type="checkbox"/>	unidad de medida de cada activo
	marca	varchar(20)	<input checked="" type="checkbox"/>	marca del activo
	proce	varchar(15)	<input checked="" type="checkbox"/>	procedencia del activo
	num_ser	varchar(15)	<input checked="" type="checkbox"/>	#de serie del activo
	tdoc	varchar(15)	<input type="checkbox"/>	tipo de documento de compra del activo
	nrodoc	varchar(15)	<input type="checkbox"/>	numero de documento de compra
	precio	decimal(10...	<input type="checkbox"/>	precio de adquisicion del activo
	tasa	decimal(5, 3)	<input type="checkbox"/>	porcentage de vida util e j 4 años $100/4=25.0$
	cprov	varchar(15)	<input type="checkbox"/>	codigo del proveedor
	feh_reg	datetime	<input type="checkbox"/>	fecha de adquisicion
	resp_reg	varchar(10)	<input type="checkbox"/>	responsable del registro
	cond_act	varchar(25)	<input type="checkbox"/>	condicion del activo
	status	varchar(20)	<input type="checkbox"/>	estado del activo
			<input type="checkbox"/>	

Tabla.3.14 Definición de la tabla de Activos Fijos

Fuente: Elaboración propia

tabla de asignaciones

asigns *				
	Nombre de columna	Tipo de datos	Permitir valores nulos	Descripción
	cod_asig	int	<input type="checkbox"/>	cod de la asignacion generada automaticamente por la base de datos
	cod_act	varchar(10)	<input type="checkbox"/>	codigo del activo
	cod_fun	varchar(10)	<input type="checkbox"/>	cod del funcionario asignado
	fch_ass	datetime	<input type="checkbox"/>	fecha de asignacion
	resp_ass	varchar(10)	<input type="checkbox"/>	responsable de la asignacion
	obs	varchar(100)	<input checked="" type="checkbox"/>	observaciones
			<input type="checkbox"/>	

Tabla.3.15 Definición de la tabla de Asignaciones

Fuente: Elaboración propia

tabla de baja de activos


bajact				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 num_baja	int	<input type="checkbox"/>		numero correlativo de bajas
cod_act	varchar(10)	<input type="checkbox"/>		codigo del activo dado de baja
cod_resp	varchar(10)	<input type="checkbox"/>		codigo del responsable de la baja
cod_fun	varchar(10)	<input type="checkbox"/>		codigo del funcionario quien custodiaba el activo
fch_baj	datetime	<input type="checkbox"/>		fecha de la baja
motivo	varchar(300)	<input type="checkbox"/>		motivo de la baja
obs	varchar(200)	<input checked="" type="checkbox"/>		observaciones de la baja
		<input type="checkbox"/>		

Tabla.3.16 Definición de la tabla Baja de Activos Fijos
Fuente: Elaboración propia

tabla de cuentas

cuentas				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
num_cuenta	varchar(11)	<input type="checkbox"/>		#de cuenta del activo
cod_act	varchar(10)	<input type="checkbox"/>		codigo del activo
importe	decimal(10...	<input type="checkbox"/>		importe revalorizado de la fecha
feh_proc	datetime	<input checked="" type="checkbox"/>		fecha del proceso
dep_acu	decimal(10...	<input checked="" type="checkbox"/>		depreciación acumulada a la fecha
sal_lb	decimal(10...	<input checked="" type="checkbox"/>		saldo en libros a la fecha
		<input type="checkbox"/>		

Tabla.3.17 Definición de la tabla Cuenta de Activos Fijos
Fuente: Elaboración propia

tabla de devoluciones


devol				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 cod_dev	int	<input type="checkbox"/>		num correlativo de devoluciones
cod_act	varchar(10)	<input type="checkbox"/>		codigo del activo devuelto
cod_func	varchar(10)	<input type="checkbox"/>		cod del funcionario que devuelve
feh_dev	datetime	<input type="checkbox"/>		fecha de devolución
resp_dev	varchar(10)	<input type="checkbox"/>		responsable de la devolución
obs	varchar(100)	<input checked="" type="checkbox"/>		observaciones
		<input type="checkbox"/>		

Tabla.3.18 Definición de la tabla Devoluciones de Activos Fijos
Fuente: Elaboración propia

En el ANEXO E se pueden encontrar los restantes tablas que corresponden al sistema de información SICAF

3.16 MODELO DE DESPLIEGUE DEL SISTEMA

3.16.1 DIAGRAMA DE COMPONENTES

El siguiente diagrama muestra la disposición de las partes integrantes del sistema de información SICAF y las dependencias entre los distintos módulos de la aplicación.

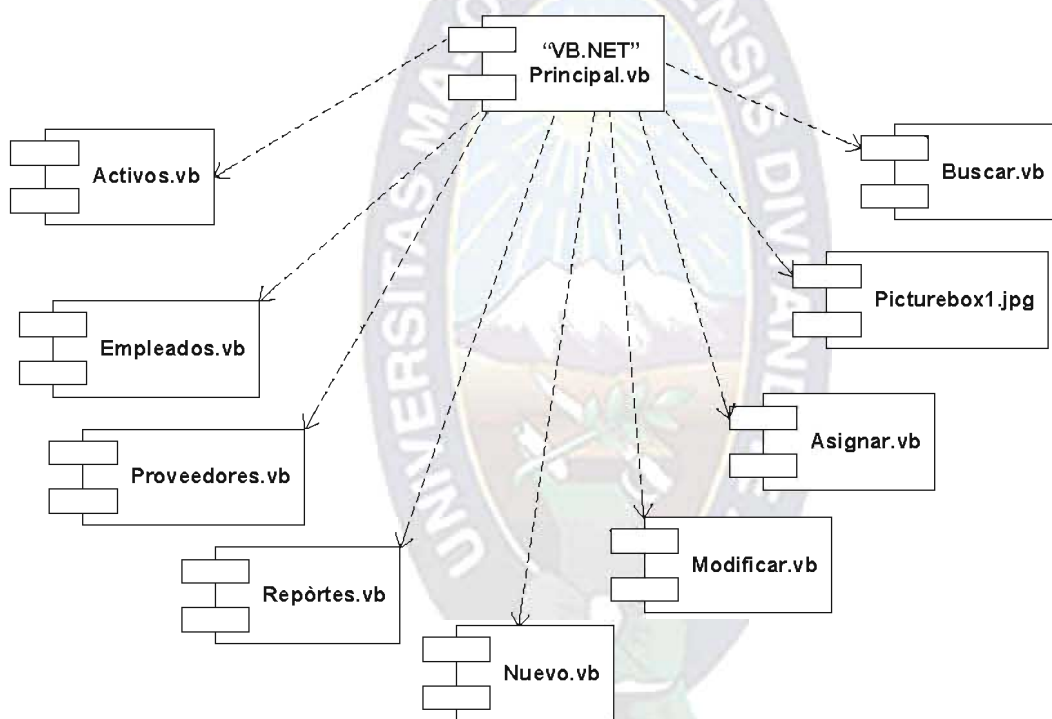


Fig.3.19 Diagrama de componentes sistema SICAF
Fuente: Elaboración propia

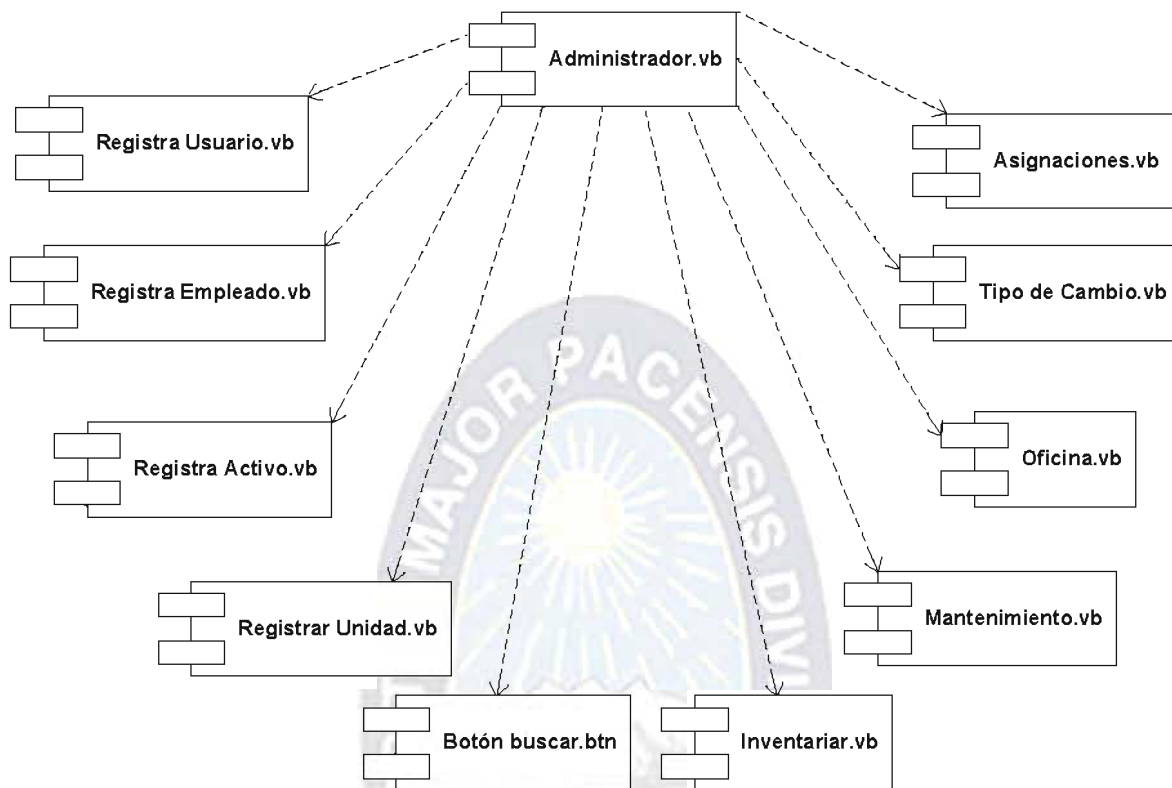


Fig.3.20 Diagrama de componentes sistema SICAF
Fuente: Elaboración propia

3.16.2 DIAGRAMA DE DESPLIEGUE

El sistema de información SISGAF se implementa en un entorno de red con un servidor central y cuatro terminales correspondiendo a: la unidad administrativa, financiera, contabilidad, encargado de activos fijos y la unidad de tecnologías de la información

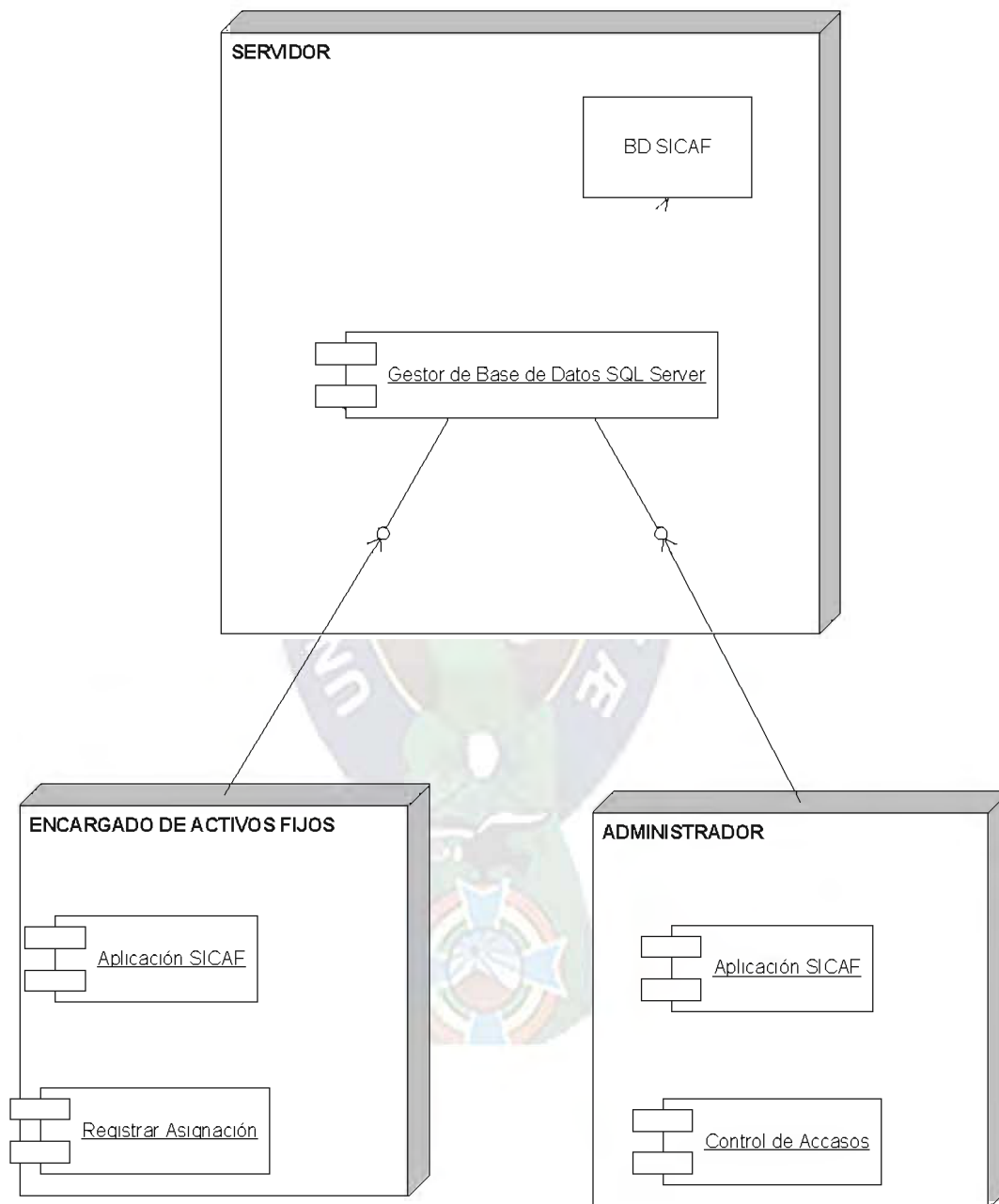


Fig.3.21 Diagrama de despliegue sistema SICAF
Fuente: Elaboración propia

3.17 FASE DE CONSTRUCCION

En la fase de construcción las clases de la fase de diseño son convertidas a código fuente en un lenguaje de programación

3.17.1 IMPLEMENTACION DE INTERFACES



Fig.3.22 Interfaz Activos Fijos
Fuente: Elaboración propia



Fig.3.23 Interfaz Funcionarios
Fuente: Elaboración propia

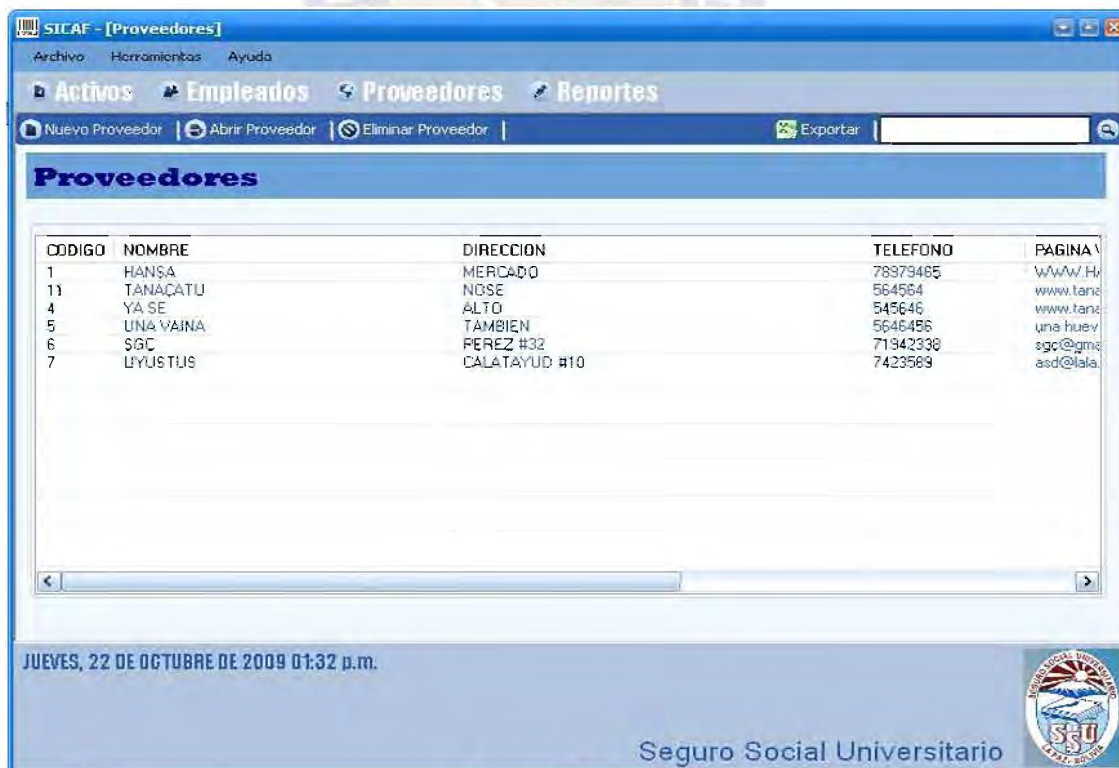


Fig.3.24 Interfaz Proveedores
Fuente: Elaboración propia

Datos Empleado

GUARDAR Y SALIR

INFORMACION DEL EMPLEADO

NOMBRE:

PATERNO:

MATERNO:

TELEFONO:

CARGO:

CODIGO DEL EMPLEADO

CODIGO:

UNIDAD

CODIGO:

DESCRIPCION

ACTIVOS ASIGNADOS

CODIGO	DESCRIPCION	FECHA DE ASIGNACION	RESPONSABLE DE ASIGNACION
asd123	asdfgh	02/10/2009	dormilona
as123	pc corel2duo	02/09/2009	LEONARDO

Fig.3.25 Interfaz Modificar Empleado
Fuente: Elaboración propia

Datos Empleado

GUARDAR Y SALIR

INFORMACION DEL EMPLEADO

NOMBRE:

PATERNO:

MATERNO:

TELEFONO:

CARGO:

CODIGO DEL EMPLEADO

CODIGO:

UNIDAD

CODIGO:

DESCRIPCION

ACTIVOS ASIGNADOS

CODIGO	DESCRIPCION	FECHA DE ASIGNACION	RESPONSABLE DE ASIGNACION
--------	-------------	---------------------	---------------------------

Fig.3.26 Interfaz Nuevo Empleado
Fuente: Elaboración propia

El contenido del capítulo abarca todo lo relacionado a la calidad del sistema, mediante las métricas de calidad de software detallada en el capítulo II que hace referencia a la ISO/IEC 9126 que es un estándar que establece los factores de calidad para un sistema software.

En la calidad no es necesariamente llegar al objetivo perfecto más por el contrario es la necesidad y suficiencia para contexto de uso en el momento de manejo de los usuarios, se tomo los siguientes criterios de calidad. Funcionalidad, Fiabilidad, Facilidad de Mantenimiento, Usabilidad y Portabilidad. Los mismos los desarrollaremos a continuación.

4.1 FACILIDAD DE MANTENIMIENTO

La facilidad de mantenimiento está asociada a la detección y corrección de errores, a los cambios debido a los requerimientos del usuario a las adaptaciones requeridas a medida que evoluciona el software.

El mantenimiento del software cuenta con más esfuerzo que cualquier otra actividad de ingeniería del software. La facilidad de mantenimiento es la habilidad con la que se puede corregir un programa si se encuentra un error, se puede adaptar si su entorno cambia u optimizar si el cliente desea un cambio de requisitos.

Para tales efectos el mantenimiento del sistema de control de activos fijos estará a cargo de mi persona durante el primer año.

4.2 PORTABILIDAD

De acuerdo a los factores de calidad *“la portabilidad es el esfuerzo necesario para transferir una aplicación de un entorno Sistema Hardware y/o Software a otro”*

[PRESSMAN, 2003].

El sistema de Digitalización por estar diseñado para un entorno de acceso a red, mide la portabilidad en: el lado del Servidor, y en el lado del Cliente:

- **Portabilidad en el lado del Servidor.-** en cuando al Hardware requiere de un ordenador con las siguientes características: Procesador 1.7 GHz (o superior), Memoria RAM 512 (como mínimo), Memoria en Video 32 MB. En el Software, el sistema funciona sin ninguna dificultad con un Sistema Operativo Windows Server 2003, como servidor se utilizara, el gestor de Base de Datos es SQL 2005.
- **Portabilidad en el lado del Cliente.-** el equipo tecnológico que corresponde a los clientes de la aplicación de deberán consistir como mínimo en: Procesador 1.7 GHz o Pentium D, Memoria RAM 125 MB, Tarjeta de Video de 64 MB.

Los requerimientos necesarios relacionados en el Software son los siguientes: Sistema Operativo Windows XP SP2, Acrobat Reader, Microsoft net frameworks 2.0.

4.3 USABILIDAD

El estándar ISO-9126 define la Usabilidad como *“la capacidad de un producto de software de facilitar a usuarios específicos alcanzar metas específicas con eficacia, productividad, seguridad y satisfacción en un contexto específico de uso”*. Añade que *“calidad en uso es la visión de calidad de los usuarios de un ambiente conteniendo software y es medida sobre los resultados de usar el software en el ambiente, antes que las propiedades del software en sí mismo”*.

El sistema de control de activos fijos con relación a la usabilidad tiene la característica de ser comprensible además de disponer de la facilidad de uso para los usuarios finales como el encargado de Activos fijos u otro con privilegios limitados que cumplan funciones relacionadas con el manejo de activos fijos. La usabilidad y facilidad de uso se logro conforme se hacían las entregas del software en las cuales se han hecho las modificaciones necesarias conforme a las peticiones del usuario.

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Con la implementación del Sistema de Información y Control de Activos fijos SICAF para el Seguro Social Universitario La Paz, se logro cumplir con el objetivo principal del presente proyecto de grado proporcionando con ello una herramienta de trabajo reduciendo los tiempos en el flujo administrativo además de ordenar y centralizar la información ayudando con ello a los funcionario encargados del manejo de activos fijos del SSU.

La implementación del sistema de información permitió: que el flujo de la información, la generación de información de los activos fijos y los reportes de actualización y depreciación de los bienes del SSU sean confiables, precisos, oportunos y coherentes. Además se logro la automatización de los procesos manuales existentes (como ser: registró, asignación, devoluciones, cálculos y generación de reportes) con la implementación de procedimientos que efectúan estos procesos manuales.

5.2 Recomendaciones

Un sistema de control de activos fijos lograria ser más completo si unimos otros sistemas relacionádos con el area de activos fijos como sistema de Almacenes, Sistema contable y tal vez un sistema de seguridad que identifique un activo cuando sale de la institución sin el permiso respectivo.

El presente proyecto tiene planes a futuro conectarse con las partes mencionadas anteriormente pero para dicho proyecto se requiere un monto de inversión que nos permita construir dichos sistemas, es recomendable implantar estos sistema para tener un control absoluto de cuentas, seguridad de los activos y almacenes de los mismos

BIBLIOGRAFIA

REFERENCIAS BIBLIOGRAFICAS

- [Presman2005] Pressman, R.: “Ingeniera De Software”, 2005, 5ta. Edicion, Editorial. McGraw Hill.
- [Presman-2006] Pressman, R.: “Ingeniera De Software”, 2006, 6ta. Edicion, Editorial. McGraw Hill.
- [Larman1999] Larman Craig.: “UML Y Patrones “, 1999 Primera Edicion, Editorial. Prentice Hall, México 1999.
- [Larman2003] Larman Craig.: “UML Y Patrones “, 2000, Editorial. Prentice Hall, México 2003.
- [JAC1999] I. Jacobson, Grady Booch, J. Rumbaugh, Ed Addison Wesley “EL Proceso Unificado de Desarrollo Software”, 1999.
- [Fun2006] Funes Orellana, j. 2006: “EL ABC DE LA CONTABILIDAD” 3ra edición, Cochabamba- Bolivia agosto 2006.
- [Fun2008] Funes Orellana, j. 2008: “CONTABILIDAD INTERMEDIA” 3ra edición, Cochabamba- Bolivia agosto 2007.
- [Teran1997] Teran Gandarillas, G. 1997: “CONTABILIDAD BASICA” 1ra edición, La paz – Bolivia 1997.
- [JGCR2005] José García Fanjul – Claudio de la Riva Álvarez “Introducción al Análisis Orientado a Objetos con UML” 2005.
- [JOS2005] José Jaime López Rabadán “Análisis para el proceso de diseño de Sistemas” 2005.

REFERENCIAS Web

- *Administración de activos fijos e intangibles.*

<http://www.fepade.edu.sv/cra/istitutos/infravenz/docs/apremat/tt/tercer%20a%F1o/AC%20III%20m%F3dulo%201.pdf>

- [DEP2003] Departamento de Sistemas Facultad regional Santa Fe (UTN)resumen :E-011-2003

- TRATAMIENTO CONTABLE EN LA ADMINISTRACIÓN DE ACTIVOS FIJOS

<http://www.cgr.gov.bo/PortalCGR/Uploads/2008/cencap/admcentral/09Tratamontable.pdf>

- [DWEB2007] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.htm>





ANEXO A: MÉTODOS DE DEPRECIACIÓN

Existen varios métodos de depreciación entre ellas tenemos:

- Línea recta.
- Por Rendimiento.
 - ✓ Por volumen de producción.
 - ✓ Por horas de trabajo.
- Números de dígito o suma de los dígitos.
 - ✓ Creciente.
 - ✓ Decreciente.
- Saldo Decreciente.

1 Método de Línea Recta:

Este es un método simple y el más utilizado en las operaciones comerciales. Permite hacer la distribución total de la depreciación por partes iguales a través de todos los periodos de su vida útil de su bien particular, su fórmula es como sigue:

$$D = \frac{C - VR}{N}$$

Denominaciones:

D = Depreciación Anual

C = Costo

VR = Valor Residual o desecho

N = Años de Vida útil del activo

2 Método Por rendimiento:

Mediante este método la depreciación se realiza en función al rendimiento del bien de uso que puede medirse a través del volumen de producción durante la vida útil o en función de la cantidad de horas probables de trabajo, por tanto se puede aplicar en función de:

- a) Volumen de Producción, y
- b) Hojas de trabajo

2.1 Método de Volumen de Producción

En este método le depreciación se calcula en directa relación al rendimiento, ósea al volumen de la producción del activo en cada periodo. La distribución del costo se realiza sobre la vida probable en la unidad de producción.

Denominación:

P = Número estimado de unidades de producción durante la vida útil del activo

Cdu = Cuota de depreciación por unidad

Su Formula es:

$$Cdu = \frac{C - VR}{p}$$

2.2 Método de Horas de Trabajo :

Este método es similar al aplicado volumen de producción con la única diferencia de que se utiliza las horas de trabajo en vez del volumen de producción, de modo que es necesario conocer la cuota de depreciación por cada hora de trabajo y la información técnica relacionado con las horas de trabajo y la información técnica relacionado con las horas de trabajo estimado durante la vida útil de la vida del bien de uso ; por lo que es aplicable en forma preferencial a las maquinarias y equipos industriales por su relación con los costos de producción.

Su fórmula es:

$$Cuh = \frac{C - VR}{nh}$$

Denominaciones.

nh = número estimado de horas de trabajo.

Cuh = Cuota de depreciaciones por hora de trabajo

2 METODO DE SUMA DE NÚMEROS Y DIGITOS

Consiste en obtener la suma de los números dígitos de acuerdo al tiempo de vida útil de los bienes de usoy utilizar esta sumatoria como denominador.

2.1 Método Decreciente

De acuerdo con este método la depreciación se calcula en forma descendente, de modo que, en los primeros años la depreciación resulta mayor que en los últimos años, con el fin de compensar el aumento de los gastos de reparación en los activos a medida que transcurre su vida útil.

La fórmula para determinar la suma de los dígitos es como sigue:

$$S = N * \frac{N + 1}{2}$$

2.2 Método Creciente

La depreciación se aplica en forma creciente, de modo que las primeras gestiones el bien de uso no requiere el costo de mantenimiento que los últimos años, por tanto, la depreciación de los años será inferior que los últimos años.

3 Método del saldo Decreciente

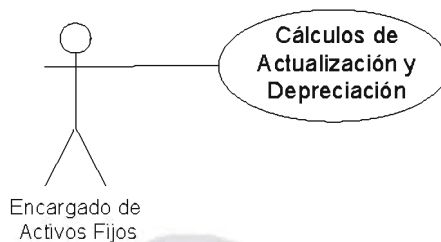
Para su cálculo primero se obtiene el porcentaje de depreciación para luego aplicar sobre los saldos de cada periodo.

Denominaciones:

% S/S = Porcentajes a aplicar a los saldos del activo fijo.

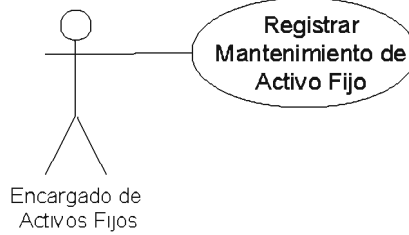
$$\% \frac{S}{S} = 1 - \sqrt[n]{\frac{VR}{C}} * 100$$

ANEXO B: CASO DE USO EXPANDIDO



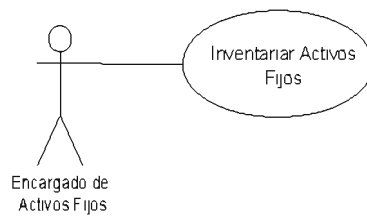
NOMBRE DEL CASO DE USO	Calculo de depreciación y actualización	
ACTORES	Encargado de activos fijos	
RESUMEN	Efectúa el procedimiento de actualización y depreciación en función al tipo de cambio.	
PROPOSITO	Conocer los valores netos de los activos	
REFERENCIAS CRUZADAS	R13,R14, R15, R17	
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- El encargado ingresa al formulario respectivo.	2.- Se muestra la fecha del último proceso.
	3.- El encargado de activos selecciona la fecha para el nuevo proceso.	4.-Se carga el tipo de cambio.
	5.-Se verifica y se termina el proceso.	6.- Se genera un reporte de actualización y depreciación.
FLUJO ALTERNATIVO	<ul style="list-style-type: none"> - En 2 si no se encuentra la fecha se verificara el tipo de cambio y fecha. - En 4 si no existe el tipo de cambio a la fecha se debe verificar el mismo. 	

Tabla: 1 Anexo B: caso de uso Cálculo de Depreciación y actualización
Fuente: Elaboración propia



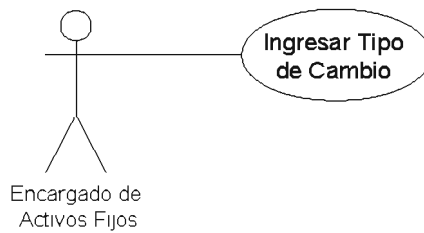
NOMBRE DEL CASO DE USO	Registrar mantenimiento de activo fijo	
ACTORES	Encargado de activos fijos	
RESUMEN	El activo fijo se encuentra en mal estado o cambia al status del mismo se lleva a un mantenimiento.	
PROPOSITO	Registrar o tener un historial de los mantenimientos realizados a un bien.	
REFERENCIAS CRUZADAS	R23	
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- El encargado ingresa al formulario respectivo.	2.-Se cargan los datos respectivos.
	3.- El encargado llena los datos respectivos.	4.- Se termina el proceso.
		5.- Se imprime el reporte.
FLUJO ALTERNATIVO	- En 1 si no se encuentra el activo se debe verificar el código del mismo.	

Tabla: 2 Anexo B: caso de uso Registrar mantenimiento de activo Fijo
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Inventariar activos fijos	
ACTORES	Encargado de activos fijos	
RESUMEN	Se realiza un inventariado de los activos a un respectivo funcionario.	
PROPOSITO	Se realiza el inventariado para verificar los datos existentes del bien.	
REFERENCIAS CRUZADAS	R20	
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- El encargado introduce el código del funcionario correspondiente.	2.- El sistema carga los bienes asignados al funcionario.
	3.- El encargado introduce código de barras del activo.	4.- El sistema verifica el código de barras existente en la lista de bienes asignadas al funcionario.
	5.- El encargado actualiza los datos.	
		6.- saca reporte del inventario
FLUJO ALTERNATIVO	<ul style="list-style-type: none"> - En 1 el funcionario debe estar registrado. - En 4 el sistema enviara un mensaje de error si el código del bien no se encuentra entre los bienes asignados al funcionario. 	

Tabla: 3 Anexo B: caso de uso Inventariar activos fijos
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Ingresar tipo de cambio	
ACTORES	Encargado de activos fijos	
RESUMEN	Ingresar el tipo de cambio respectivo al día actual	
PROPOSITO	Tener a disposición los tipos de cambio con actualización diaria	
REFERENCIAS CRUZADAS	Ninguno	
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- Ingresar al sistema	2.- Muestra formulario de tipo de cambio
	3.- Ingresar el tipo de cambio de moneda	
		4.- Registrar el ingreso de tipo de cambio de la moneda
		5.- Confirmar los datos del tipo de cambio
	6.- encargado tiene acceso a los procesos de activos fijos.	
FLUJO ALTERNATIVO	<ul style="list-style-type: none"> - En 1 el encargado de activos fijos debe realizar su correspondiente autenticación, para luego proceder a la verificación de los tipos de cambio a la fecha actual. - En 2 y 3 puede que se haya registrado ya el tipo de cambio a la fecha actual, entonces el sistema no permite ingresar tipo de cambio y el usuario accede a los procesos de activos fijos. 	

Tabla: 4 Anexo B: caso de uso Ingresar Tipo De Cambio
Fuente: Elaboración propia



NOMBRE DEL CASO DE USO	Administrar el Sistema	
ACTORES	Encargado de activos fijos	
RESUMEN	El administrador del sistema es el encargado de controlar el ingreso al sistema de los usuarios. Solo pueden ingresar los usuarios registrados además que cada usuario tiene sus respectivos privilegios.	
PROPOSITO	Controlar los ingresos al sistema además de comprobar los privilegios asignados al usuario registrado	
REFERENCIAS CRUZADAS		
FLUJO NORMAL DE EVENTOS	ACTORES	SISTEMA
	1.- El encargado introduce su contraseña.	2.- Se verifica la existencia del usuario.
	3.- El usuario accede al sistema de activos fijos	
	4.- El usuario realiza las actividades y funciones correspondientes	
FLUJO ALTERNATIVO	<ul style="list-style-type: none"> - En 1 si la contraseña del usuario son incorrectos se despliega el correspondiente mensaje de error - En 2 se verifican los roles de cada usuario. 	

Tabla: 5 Anexo B caso de uso Administrar el sistema
Fuente: Elaboración propia

ANEXO C: DIAGRAMAS DE SECUENCIA



Fig.1 Anexo C: Diagrama de Secuencia Calculo de depreciación y Actualización
Fuente: Elaboración Propia

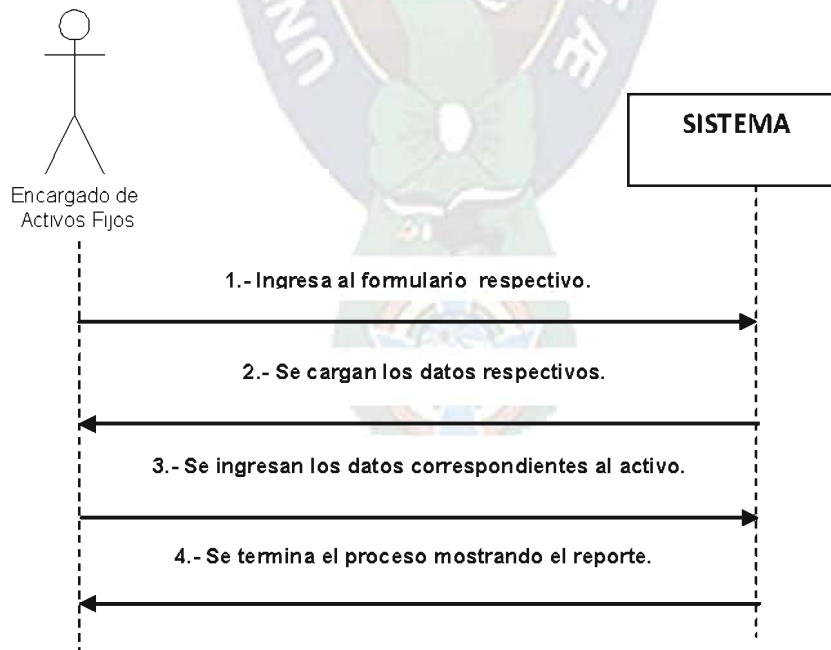


Fig.2 Anexo C: Diagrama de Secuencia Registrar Mantenimiento de Activo Fijo
Fuente: Elaboración Propia

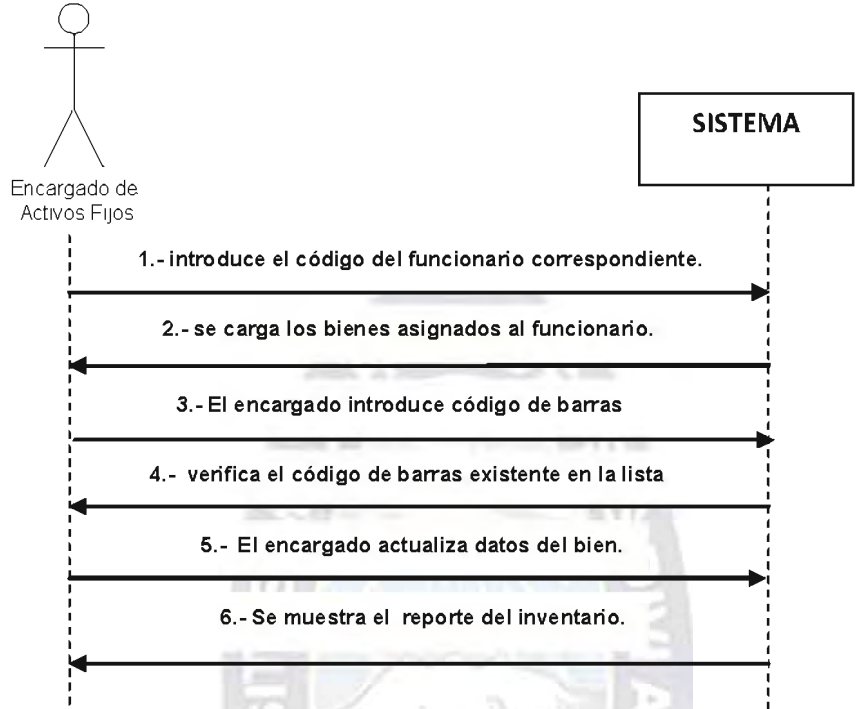


Fig.3 Anexo C: Diagrama de Secuencia Inventariar Activo Fijo
Fuente: Elaboración Propia

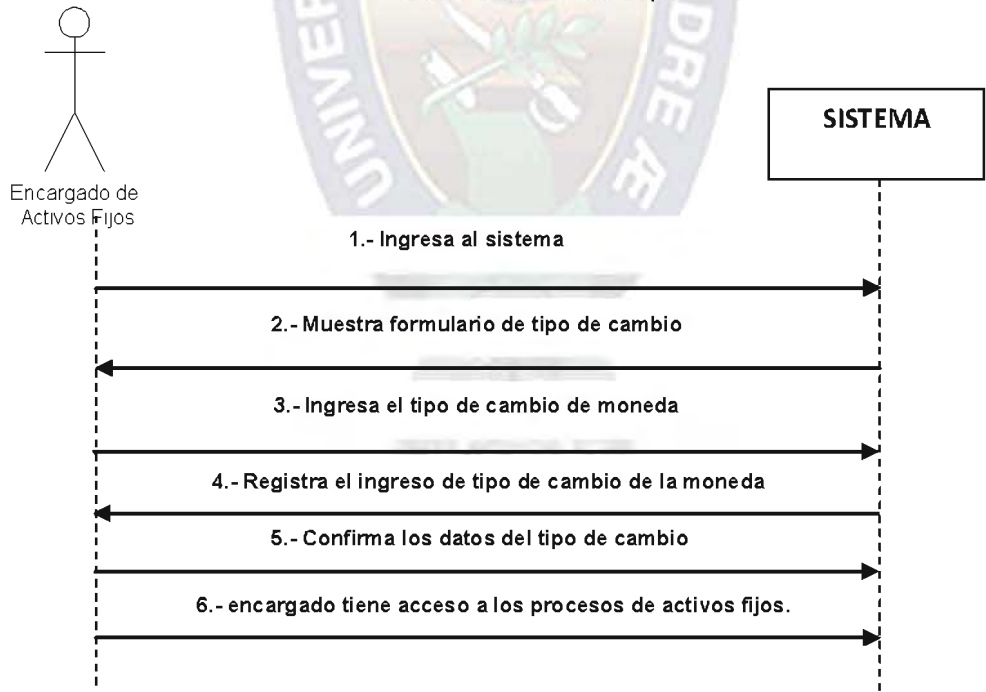


Fig.4 Anexo C: Diagrama de Secuencia Ingresar Tipo de Cambio
Fuente: Elaboración Propia

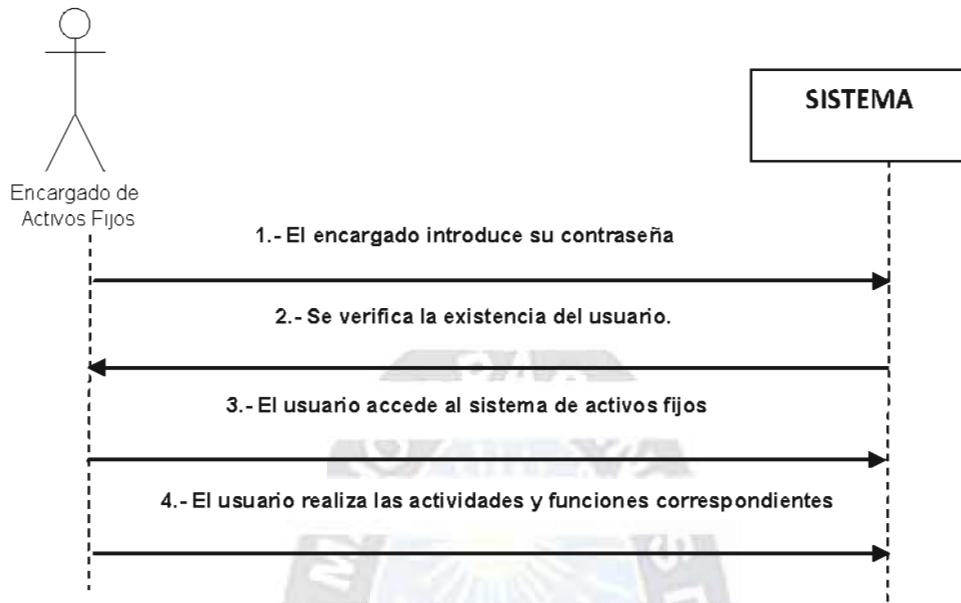


Fig.5 Anexo C: Diagrama de Secuencia Administrar el sistema
Fuente: Elaboración Propia

ANEXO D: CONTRATOS

- REGISTRAR ACTIVO FIJO

CONTRATO	
NOMBRE	El encargado elige la opción nuevo Activo Fijo
RESPONSABILIDAD	Ingresa datos del nuevo activo fijo
TIPO	Sistema
REFERENCIAS	R4,R5
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 6 Anexo D: El encargado elige la opción nuevo Activo Fijo
Fuente: Elaboración propia

CONTRATO	
NOMBRE	El sistema carga los datos correspondientes
RESPONSABILIDAD	Muestra los datos correspondientes al nuevo registro
TIPO	Sistema
REFERENCIAS	R4,R5
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 7 Anexo D: Contrato El sistema carga los datos correspondientes
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se llenan datos correspondientes al activo
RESPONSABILIDAD	Se registra los datos del activo
TIPO	Sistema
REFERENCIAS	R4,R5
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 8 Anexo D: Contrato Se llenan datos correspondientes al activo
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se confirman los datos y se confirma el registro
RESPONSABILIDAD	Se muestra un mensaje de confirmación
TIPO	Sistema
REFERENCIAS	R4,R5
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 9 Anexo D: Contrato Se confirman los datos y se confirma el registro
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se almacena los datos del nuevo activo
RESPONSABILIDAD	Se registra los datos del activo en la base de datos

TIPO	Sistema
REFERENCIAS	R4,R5
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla:10 Anexo D: Contrato Se almacena los datos del nuevo activo
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se genera el acta de registro
RESPONSABILIDAD	Despliega informe con todos los datos del nuevo activo
TIPO	Sistema
REFERENCIAS	R4,R5
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 11 Anexo D: Contrato Se genera el acta de registro
Fuente: Elaboración propia

- Registrar Proveedor

CONTRATO	
NOMBRE	Elige opción registrar proveedor
RESPONSABILIDAD	Despliega el formulario de registrar proveedor.
TIPO	Sistema
REFERENCIAS	R8

PRECONDICION	No debe existir registro del proveedor.
POSCONDICION	El registro del proveedor estará validado

Tabla: 12 Anexo D: Contrato registrar proveedor
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Llena todos los datos para el registro
RESPONSABILIDAD	Llenar formulario de registro de proveedor
TIPO	Sistema
REFERENCIAS	R8
PRECONDICION	No debe existir registro del proveedor.
POSCONDICION	El registro del proveedor estará validado

Tabla: 13 Anexo D: Llena todos los datos para el registro
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Termina el registro
RESPONSABILIDAD	Guarda registro en la base de datos.
TIPO	Sistema
REFERENCIAS	R8
PRECONDICION	No debe existir registro del proveedor.
POSCONDICION	El registro del proveedor estará validado

Tabla: 14 Anexo D: contrato Termina el registro
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Mensaje de confirmación de registro
RESPONSABILIDAD	Despliega mensaje de confirmación registro del proveedor.
TIPO	Sistema
REFERENCIAS	R8
PRECONDICION	No debe existir registro del proveedor.
POSCONDICION	El registro del proveedor estará validado

Tabla: 15 Anexo D: contrato Mensaje de confirmación de registro
Fuente: Elaboración propia

- Registrar Baja de Activo Fijo

CONTRATO	
NOMBRE	El encargado ingresa al sistema
RESPONSABILIDAD	Despliega los formularios del sistema
TIPO	Sistema
REFERENCIAS	R19
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 16 Anexo D: contrato El encargado ingresa al sistema
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Carga activos que no están en baja
RESPONSABILIDAD	Despliega lista de activos en uso
TIPO	Sistema
REFERENCIAS	R19
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 17 Anexo D: contrato Carga activos que no están en baja
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Elije el activo a ser dado de baja
RESPONSABILIDAD	Captura los datos del presente activo
TIPO	Sistema
REFERENCIAS	R19
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 18 Anexo D: contrato Elije el activo a ser dado de baja
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Registrar Baja de Activo Fijo
RESPONSABILIDAD	Llenar formulario para la respectiva baja

TIPO	Sistema
REFERENCIAS	R19
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 19 Anexo D: contrato **Registrar Baja de Activo Fijo**
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Termina el registro
RESPONSABILIDAD	Se guardan datos del registro
TIPO	Sistema
REFERENCIAS	R19
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 20 Anexo D: contrato Termina el registro
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se genera al acta de la respectiva baja
RESPONSABILIDAD	Despliega informe con todos los datos del nuevo activo
TIPO	Sistema
REFERENCIAS	R19
PRECONDICION	El usuario debe estar suscrito en el sistema.

POSCONDICION	La cuenta de Usuario estará validado
---------------------	--------------------------------------

Tabla: 21 Anexo D: contrato Se genera al acta de la respectiva baja
Fuente: Elaboración propia

- Devolución de Activos Fijos

CONTRATO	
NOMBRE	El Encargado de A.F. selecciona el funcionario respectivo
RESPONSABILIDAD	Se captura los datos para la devolución del Activo
TIPO	Sistema
REFERENCIAS	R10
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La devolución estará registrada.

Tabla: 22 Anexo D: contrato El Encargado de A.F. selecciona el funcionario respectivo
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Carga los datos del funcionario
RESPONSABILIDAD	Muestra activos asignados al funcionario
TIPO	Sistema
REFERENCIAS	R10
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La devolución estará registrada.

Tabla: 23 Anexo D: contrato Carga los datos del funcionario
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Elige activo a ser devuelto
RESPONSABILIDAD	Se capturan los datos del activo
TIPO	Sistema
REFERENCIAS	R10
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La devolución estará registrada.

Tabla: 24 Anexo D: contrato Elige activo a ser devuelto
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Registro de la devolución
RESPONSABILIDAD	Se llena los datos de la respectiva devolución
TIPO	Sistema
REFERENCIAS	R10
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La devolución estará registrada.

Tabla: 25 Anexo D: contrato Registro de la devolución
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Reporte de la devolución
RESPONSABILIDAD	Se despliega el informe de devolución

TIPO	Sistema
REFERENCIAS	R10
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La devolución estará registrada.

Tabla: 26 Anexo D: contrato Reporte de la devolución
Fuente: Elaboración propia

- **Calculo de depreciación y actualización**

CONTRATO	
NOMBRE	El encargado ingresa al formulario respectivo
RESPONSABILIDAD	Se muestra el formulario correspondiente.
TIPO	Sistema
REFERENCIAS	R13,R14, R15, R17
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 27 Anexo D: contrato El encargado ingresa al formulario respectivo
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se carga la fecha del último proceso
RESPONSABILIDAD	Se muestra la fecha del último proceso registrado
TIPO	Sistema
REFERENCIAS	R13,R14, R15, R17

PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 28 Anexo D: contrato Se carga la fecha del último proceso
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se elige la fecha del nuevo proceso
RESPONSABILIDAD	Se verifica el tipo de cambio
TIPO	Sistema
REFERENCIAS	R13,R14, R15, R17
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 29 Anexo D: contrato Se elige la fecha del nuevo proceso
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se carga el tipo de cambio
RESPONSABILIDAD	Se despliega el tipo de cambio
TIPO	Sistema
REFERENCIAS	R13,R14, R15, R17
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 30 Anexo D: contrato Se carga el tipo de cambio
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se termina el proceso
RESPONSABILIDAD	Se registran los nuevos datos
TIPO	Sistema
REFERENCIAS	R13,R14, R15, R17
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 31 Anexo D: contrato Se termina el proceso
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se genera el reporte del registro
RESPONSABILIDAD	Se despliega el informe respectivo
TIPO	Sistema
REFERENCIAS	R13,R14, R15, R17
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 32 Anexo D: contrato Se genera el reporte del registro
Fuente: Elaboración propia

- Registrar mantenimiento de activo fijo

CONTRATO	
NOMBRE	Ingresar al formulario respectivo
RESPONSABILIDAD	Se carga el formulario correspondiente
TIPO	Sistema
REFERENCIAS	R23
PRECONDICION	Se verificara el código del bien.
POSCONDICION	Se creara un historial sobre los mantenimientos del bien

Tabla: 33 Anexo D: contrato Ingresar al formulario respectivo
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Caga de los datos respectivos
RESPONSABILIDAD	Se muestran las opciones correspondientes
TIPO	Sistema
REFERENCIAS	R23
PRECONDICION	Se verificara el código del bien.
POSCONDICION	Se creara un historial sobre los mantenimientos del bien

Tabla: 34 Anexo D: contrato Caga de los datos respectivos
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se registran los datos
RESPONSABILIDAD	Llenar formulario para su respectivo mantenimiento del bien
TIPO	Sistema
REFERENCIAS	R23
PRECONDICION	Se verificara el código del bien.
POSCONDICION	Se creara un historial sobre los mantenimientos del bien

Tabla: 35 Anexo D: contrato Se registran los datos
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se muestra el reporte en pantalla
RESPONSABILIDAD	Se despliega el informe de mantenimiento
TIPO	Sistema
REFERENCIAS	R23
PRECONDICION	Se verificara el código del bien.
POSCONDICION	Se creara un historial sobre los mantenimientos del bien

Tabla: 36 Anexo D: contrato Se muestra el reporte en pantalla
Fuente: Elaboración propia

- Inventariar activos fijos

CONTRATO	
NOMBRE	Se introduce el código del funcionario

RESPONSABILIDAD	Verificar las asignaciones del mismo
TIPO	Sistema
REFERENCIAS	R20
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	Los datos del bien o los bienes en custodia serán mostrados y verificados

Tabla: 37 Anexo D: contrato Se introduce el código del funcionario
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se muestran los bienes asignados al funcionario
RESPONSABILIDAD	Informar al encargado de los bienes del funcionario correspondiente
TIPO	Sistema
REFERENCIAS	R20
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	Los datos del bien o los bienes en custodia serán mostrados y verificados

Tabla: 38 Anexo D: contrato Se muestran los bienes asignados al funcionario
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se introduce el código del bien
RESPONSABILIDAD	Verificar el bien asignado al funcionario respectivo
TIPO	Sistema

REFERENCIAS	R20
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	Los datos del bien o los bienes en custodia serán mostrados y verificados

Tabla: 39 Anexo D: contrato Se introduce el código del bien
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Verificar el código del bien
RESPONSABILIDAD	Si el bien pertenece al funcionario este es aceptado por el sistema
TIPO	Sistema
REFERENCIAS	R20
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	Los datos del bien o los bienes en custodia serán mostrados y verificados

Tabla: 40 Anexo D: contrato Verificar el código del bien
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se actualizan los datos del bien
RESPONSABILIDAD	Modificar su estado o condición del bien
TIPO	Sistema

REFERENCIAS	R20
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	Los datos del bien o los bienes en custodia serán mostrados y verificados

Tabla: 41 Anexo D: contrato Se actualizan los datos del bien
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se muestra un reporte del inventario
RESPONSABILIDAD	Despliegue de informe del bien inventariado
TIPO	Sistema
REFERENCIAS	R20
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	Los datos del bien o los bienes en custodia serán mostrados y verificados

Tabla: 42 Anexo D: contrato Se muestra un reporte del inventario
Fuente: Elaboración propia

- Ingresar tipo de cambio

CONTRATO	
NOMBRE	Ingresar al sistema
RESPONSABILIDAD	Autenticar al usuario

TIPO	Sistema
REFERENCIAS	Ninguno
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 43 Anexo D: contrato Ingresar al sistema
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se despliega el formulario
RESPONSABILIDAD	Actualizar el tipo de cambio a la fecha
TIPO	Sistema
REFERENCIAS	Ninguno
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 44 Anexo D: contrato Se despliega el formulario
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Registrar el tipo de cambio
RESPONSABILIDAD	Registro del nuevo tipo de cambio
TIPO	Sistema
REFERENCIAS	Ninguno

PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 45 Anexo D: contrato Registrar el tipo de cambio
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Registra el tipo de cambio
RESPONSABILIDAD	Se guardan los datos
TIPO	Sistema
REFERENCIAS	Ninguno
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 46 Anexo D: contrato Registra el tipo de cambio
Fuente: Elaboración propia

CONTRATO	
NOMBRE	Se despliega un mensaje de confirmación
RESPONSABILIDAD	La base de datos a aceptado el tipo de cambio
TIPO	Sistema
REFERENCIAS	Ninguno
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 47 Anexo D: contrato Se despliega un mensaje de confirmación
Fuente: Elaboración propia

CONTRATO	
NOMBRE	El encargado tiene acceso al sistema
RESPONSABILIDAD	Una vez aceptado el tipo de cambio el sistema se habilita
TIPO	Sistema
REFERENCIAS	Ninguno
PRECONDICION	El usuario debe estar suscrito en el sistema.
POSCONDICION	La cuenta de Usuario estará validado

Tabla: 48 Anexo D: contrato El encargado tiene acceso al sistema
Fuente: Elaboración propia



ANEXO E: ESQUEMAS DE LA BASE DE DATOS

tabla del empleado


empleado *				
Nombre de columna	Tipo de datos	Permitir valores nulos	Descripción	
 ci	varchar(10)	<input type="checkbox"/>	codigo unico del empleado	
pat	varchar(20)	<input type="checkbox"/>	apellido paterno del empleado	
mat	varchar(20)	<input checked="" type="checkbox"/>	apellido pmaterno del empleado	
nombres	varchar(30)	<input type="checkbox"/>	nombre o nombres de empleado	
cod_uni	varchar(3)	<input type="checkbox"/>	codigo de la unidad del empleado	
telf_fun	varchar(10)	<input checked="" type="checkbox"/>	telefono o #interno del empleado	
cargo	varchar(100)	<input type="checkbox"/>	cargo que ocupa el empleado	
		<input type="checkbox"/>		

Tabla.49 Anexo E Definición de la tabla de Empleados

Fuente: Elaboración propia

tabla de grupo de cuentas

grupo *				
Nombre de columna	Tipo de datos	Permitir valores nulos	Descripción	
 cuenta	varchar(11)	<input type="checkbox"/>	# de cuenta mayor	
mayor	varchar(11)	<input type="checkbox"/>	# de la cuenta mayor a la que pertenece	
nombre	varchar(50)	<input type="checkbox"/>	nombre del grupo	
		<input type="checkbox"/>		

Tabla.50 Anexo E Definición de la tabla de Grupo

Fuente: Elaboración propia

tabla de mantenimiento de los activos


mantenimiento *				
Nombre de columna	Tipo de datos	Permitir valores nulos	Descripción	
 cod_man	varchar(5)	<input type="checkbox"/>	cod del mantenimiento	
cod_act	varchar(10)	<input type="checkbox"/>	cod del activo a realizar su mantenimiento	
feh_man	datetime	<input type="checkbox"/>	fecha del mantenimiento	
feh_ret	datetime	<input type="checkbox"/>	fecha de conclusion del mantenimiento	
desc_man	varchar(200)	<input type="checkbox"/>	descripcion del mantenimiento	
tipo_man	varchar(15)	<input type="checkbox"/>	mantenimiento, inspeccion, reparacion	
cost_part	decimal(5, 5)	<input checked="" type="checkbox"/>	costo de las partes compradas del mantenimiento	
cost_man	decimal(5, 5)	<input checked="" type="checkbox"/>	costo del mantenimiento	
pers_man	varchar(100)	<input type="checkbox"/>	nombre de la persona que se encarga del mantenimiento	
		<input type="checkbox"/>		

Tabla.51 Anexo E Definición de la tabla de Mantenimiento

Fuente: Elaboración propia

tabla de mayores


mayores *				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 cta_may	varchar(11)	<input type="checkbox"/>		# de cuenta mayor
nombre	varchar(50)	<input type="checkbox"/>		nombre de la cuenta mayor
sw	varchar(2)	<input type="checkbox"/>		dato que indica si la cuenta mayor se actualiza y deprecia o...
		<input type="checkbox"/>		

Tabla.52 Anexo E Definición de la tabla de Mayores
Fuente: Elaboración propia

tabla de oficinas


oficina *				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 cod_ofi	varchar(4)	<input type="checkbox"/>		codigo de la oficina a la que pertenece la ubicacion
descripcion	varchar(100)	<input type="checkbox"/>		descripcion de la oficina
ubicacion	varchar(1)	<input type="checkbox"/>		ubicacion como el 3hospital, 2obrajes, 1policonsult...
piso	varchar(2)	<input type="checkbox"/>		pisos de -2 a 16
cod_unid	varchar(3)	<input type="checkbox"/>		codigo de la unidad
pat_resp	varchar(20)	<input type="checkbox"/>		paterno del responsable de la oficina
mat_resp	varchar(20)	<input type="checkbox"/>		materno del responsable de la oficina
nom_resp	varchar(30)	<input type="checkbox"/>		nombre del responsable de la oficina
		<input type="checkbox"/>		

Tabla.53 Anexo E Definición de la tabla de Oficinas
Fuente: Elaboración propia

tabla de proveedores


proveedores *				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 nit	varchar(15)	<input type="checkbox"/>		numero de nit del proveedor
nom_pro	varchar(30)	<input type="checkbox"/>		nombre del proveedor
dir_prov	varchar(50)	<input type="checkbox"/>		direccion del proveedor
tel_prov	varchar(10)	<input type="checkbox"/>		telefono del proveedor
wev_prov	varchar(50)	<input checked="" type="checkbox"/>		pag web del proveedor ó correo electronico
tipo	varchar(50)	<input type="checkbox"/>		tipo de provision que trae
		<input type="checkbox"/>		

Tabla.54 Anexo E Definición de la tabla de Proveedores
Fuente: Elaboración propia

tabla de tipo de cambios


tpcambio *				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 fech	datetime	<input type="checkbox"/>		fecha de ingreso del tipo de cambio
dolar	decimal(10...	<input type="checkbox"/>		tipo de cambio en dolares
ufv	decimal(10...	<input type="checkbox"/>		tipo de cambio en ufvs
resp_reg	varchar(10)	<input type="checkbox"/>		responsable del registro
		<input type="checkbox"/>		

Tabla.55 Anexo E Definición de la tabla de Tipos de Cambio
Fuente: Elaboración propia

tabla de unidad


unidad *				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 codi_unid	varchar(3)	<input type="checkbox"/>		codigo de la unidad
descripcion	varchar(100)	<input type="checkbox"/>		nombre de la uni...
		<input type="checkbox"/>		

Tabla.56 Anexo E Definición de la tabla de Unidad
Fuente: Elaboración propia

tabla de usuarios del sistema


users *				
Nombre de columna	Tipo de datos	Permitir valores nulos		Descripción
 cod_user	varchar(10)	<input type="checkbox"/>		codigo del usuario CI
nombres	varchar(30)	<input type="checkbox"/>		nombres
apellidos	varchar(30)	<input type="checkbox"/>		apellidos
password	varchar(15)	<input checked="" type="checkbox"/>		password de ingreso al sistema
rol	varchar(50)	<input type="checkbox"/>		rol que tiene el usuario
tipo	char(1)	<input type="checkbox"/>		tipo de usuario Activo o Pasivo
usuario	varchar(20)	<input type="checkbox"/>		nombre de usuario de ingreso al sistema
		<input type="checkbox"/>		

Tabla.57 Anexo E Definición de la tabla de Usuarios
Fuente: Elaboración propia