

UNIVERSIDAD AMAZÓNICA DE PANDO

ÁREA DE CIENCIA Y TECNOLOGÍA

CARRERA DE INGENIERÍA INFORMÁTICA



PROYECTO DE GRADO

“SISTEMA DE ADMINISTRACIÓN DE ACTIVOS FIJOS PARA EL GOBIERNO AUTÓNOMO MUNICIPAL DE COBIJA”

Trabajo de Titulación presentado en conformidad a los requisitos para obtener el Título de Ingeniero en Sistemas Informáticos.

ELABORADO: UNIV. HÉCTOR ROLANDO CHIPANA CHAMBI

TUTOR COLECTIVO: MSC. LIC. HUMBERTO FERNANDEZ

ASESOR: ING. JOSÉ EDGAR BALDERRAMA M.

Cobija - Pando - Bolivia
2012

DEDICATORIA

Este trabajo está dedicado a Dios Padre todo poderoso, por ser la Luz de mi camino y por todo lo que me ha dado en esta vida.

*Y sobre todo este trabajo está dedicado a mis queridos padres, por haberme dado la vida y por ser mi constante guía en el transcurso de mi vida y por ser **la inspiración en estos cinco años de estudio y esfuerzo.***

AGRADECIMIENTOS

- *A Dios todo poderoso por haberme iluminado para realizar el siguiente trabajo de investigación el cual es de gran ayuda para mi vida profesional.*
- *A mis queridos Padres por su constante aliento, apoyo moral y cariño.*
- *A mi Familia: Por su permanente incentivo, por brindarme su amor incondicional puro y verdadero, por haber depositado su total y entera confianza.*
- *A mi Tutor y Asesor, Msc. Lic. Humberto Fernández y al Ing. José E. Balderrama M. , por los consejos y acertadas sugerencias brindadas.*
- *A los Señores Docentes de la Universidad Amazónica de Pando, quienes hicieron posible mi formación académica.*
- *A mis compañeros universitarios, por los años de estudio compartidos.*

RESUMEN

Las Instituciones Públicas dependientes del Estado Plurinacional de Bolivia, para llevar a cabo sus actividades, utilizan ciertos bienes. A estos bienes que se usan, se les denomina "Activos Fijos". Gracias a estos bienes, se genera utilidades dentro las Instituciones a lo largo de la vida útil del bien adquirido. Una de las instituciones que se ha beneficiado con estos bienes, es el Gobierno Autónomo Municipal de Cobija.

De acuerdo al análisis de la investigación realizada se llego a plantear los siguientes problemas presentados en la Unidad de Activos Fijos dependiente del Gobierno Autónomo Municipal de Cobija, en el cual se determino el problema principal: "Inadecuado Sistema de Seguimiento y Control de Activos Fijos del Gobierno Autónomo Municipal de Cobija" Por lo cual se planteo el objetivo principal para mejorar la Administración de los Activos Fijos, mediante la "Implementación de un Sistema de Información de Activos Fijos". Aplicando La Metodología XP basado en Iteraciones lo cual permite trabajar de manera ágil permitiendo alcanzar con todos los objetivos planteados en el menor tiempo posible.

Con la Implementación del Sistema de Administración de Activos Fijos del Gobierno Autónomo Municipal de Cobija se garantiza una eficiente administración de los Activos Fijos.

Palabras Clave: Activos Fijos, Sistema de Información, Metodología de Desarrollo XP.

INDICE.-

1.- CAPITULO I: INTRODUCCIÓN	2
1.1.- ANTECEDENTES	2
1.2.- PLANTEAMIENTO DEL PROBLEMA	4
1.3.- OBJETIVOS	5
1.4.- JUSTIFICACIÓN	5
1.5.- METODOLOGÍAS	7
1.6.- ALCANCES	10
1.7.- APORTES	12
2.- CAPITULO II: MARCO TEORICO	14
2.1.- ANTECEDENTES TEÓRICOS, LEGALES Y CONTEXTUALES	14
2.1.1.- ASPECTOS TEÓRICOS	14
2.1.2.- ASPECTOS LEGALES	17
2.1.3.- ASPECTOS CONTEXTUALES	25
2.2.4.- ACTIVIDADES QUE REALIZA UN SISTEMA DE INFORMACIÓN	26
2.3.- METODOLOGÍAS DE DESARROLLO DE SOFTWARE	32
2.4.- METODOLOGÍA XP	36
2.5.- ELEMENTOS DE LA IMPLEMENTACIÓN DE LOS SISTEMAS DE INFORMACIÓN	39
3.- CAPITULO III: DESARROLLO	57
3.1.- PLANIFICACIÓN GENERAL DEL PROYECTO	57
3.2.- DIAGRAMA DE CASOS DE USO GENERAL DEL SISTEMA	62
3.3.- HISTORIAS DE LOS USUARIOS	63
3.4.- PLAN DE ENTREGA	63
4.- CAPITULO IV: CONCLUSIONES Y RECOMENDACIONES	92
4.1.- CONCLUSIONES	92
4.2.- RECOMENDACIONES	92
BIBLIOGRAFÍA	93
ANEXOS	94



CAPÍTULO I

INTRODUCCION



1.- CAPITULO I: INTRODUCCIÓN

1.1.- ANTECEDENTES

En nuestro País, desde años atrás, las instituciones públicas, se rigen en base a leyes y normativas para llevar a cabo sus actividades, con la finalidad de alcanzar ciertos objetivos trazados que contribuyan al buen funcionamiento y administración.

En el Gobierno Autónomo Municipal de Cobija, en su estructura jerárquica, se encuentra la unidad de Activos Fijos y comprende una tarea muy importante, que es la de administrar todos los bienes muebles e inmuebles del Municipio de Cobija.

Dentro de la Unidad de Activos Fijos del Gobierno Autónomo Municipal de Cobija, desde años atrás, se administra los activos mediante un sistema llamado SIAF (Sistema de Información de Activos Fijos), la misma que fue elaborada bajo requerimiento del Órgano Rector del Ministerio de Hacienda, que es la autoridad normativa del Sistema de Contabilidad Integrada, establecido por el artículo 22 de la Ley No.1178 de Administración y Control Gubernamental el 20 de julio de 1990, en ese entonces, se creó con la finalidad de facilitar un medio sencillo de registro del movimiento de Activos, y a su vez permitía procesar y emitir información según Principios Generales y Normas básicas de Contabilidad Gubernamental. Las mismas que con el transcurso del tiempo fueron quedando fuera de funcionamiento debido a que las normativas y leyes han ido evolucionando con el transcurso del tiempo.

Hoy en día, en la Unidad de Activos Fijos dentro del Gobierno Municipal de Cobija, los registros de los activos se guardan mediante planillas de Excel. Se codifican de acuerdo al clasificador de activos fijos, por grupos contables, cuentas principales y sub cuentas para posteriormente introducirlas al SIAF.

Dentro de los requerimientos de la Unidad de Activos Fijos son:

- Intentar ubicar los activos fijos de una manera rápida y eficiente.
- Migrar la información de los activos fijos a otros sistemas operativos.
- Actualizar la información del t/c del 1999 al 2006 en \$us. A t/c Ufv del 2007 a la fecha.

- Administrar los activos fijos mediante código de barras. Con reportes adecuados e importantes para toma de decisiones (Ejm. revaluos técnicos, reposición de activos, mantenimiento, actas, baja, etc.)

Las autoridades del Gobierno Autónomo Municipal de Cobija, en busca de mejorar la administración de los activos fijos se vieron con la necesidad de contar con un sistema que responda a las normativas y leyes que regulan la administración de Activos Fijos. Por lo que se pretende subsanar ahora, son las regularizaciones que emite el Gobierno Central del Estado Plurinacional de Bolivia, en base al Decreto Supremo 0181, que trata sobre las Normas Básicas del Sistema de Administración de Bienes y Servicios. Conforme a los nuevos requerimientos dentro de la Unidad de Activos Fijos del Gobierno Autónomo Municipal de Cobija.

De acuerdo a investigaciones realizadas en esta área se han podido obtener ciertos sistemas de información de activos fijos que de alguna manera han contribuido al desarrollo de las instituciones (Nacional/Internacional).

“Sistema Simec de Activos Fijos” Fue elaborado por la empresa SIMEC SOFT LIMITADA que se encarga de brindar soluciones al control Administrativo y Financiero, este software fue implementado en Cochabamba-Bolivia el año 2003, el cual menciona que da una amplia gama de facilidades en el control y administración de los activos fijos.

“Sistema de Activos Fijos ASA” Fue elaborado por la empresa ASAVEN. Este software fue realizado en Caracas-Venezuela. También menciona que su sistema está programado en Lenguaje Visual Basic, y puede realizar consultas, reportes y procesos en cuanto al control administrativo de los activos fijos.

“Sistema Integrado de Información Financiera SIIF” Fue elaborado en Bolivia, para la administración financiera de control interno y externo de las adquisiciones de bienes, fondos en custodia, participación popular, crédito público, flujo de caja y ejecución presupuestaria.

“Sistema de Información para la Gestión Financiera del Estado SIGFE” Fue elaborado en Chile, es un sistema de información financiera pública, que permite la

administración de sus bienes en todos los niveles jerárquicos, realiza un análisis de los distintos procesos que tiene, optimizando la gestión y su control; y a su vez brinda un apoyo a la toma de decisiones de la administración del Estado.

Es así que estos sistemas de información de activos fijos nos sirven como una referencia para tomar en cuenta en el presente proyecto, pero no en su totalidad, debido a que fueron desarrolladas en un entorno desmedido, es decir, que se abstraen de algunas leyes y normativas de su país.

1.2.- PLANTEAMIENTO DEL PROBLEMA

Una vez realizado el análisis del funcionamiento de la Unidad de los Activos Fijos dentro del Gobierno Autónomo Municipal de Cobija, se ha podido evidenciar ciertas falencias en la administración de los Activos Fijos, como ser:

- Demora en la obtención de la información.
- Datos desactualizados.
- Redundancia de datos.
- Procesos inadecuados para el seguimiento y control de los activos fijos.

Por esta razón se formula el siguiente problema principal:

“INADECUADO SISTEMA DE SEGUIMIENTO Y CONTROL DE ACTIVOS FIJOS DEL GOBIERNO AUTONOMO MUNICIPAL DE COBIJA”

Este problema principal ocasiona:

- Que la Unidad de activos fijos tenga una información descentralizada e incompleta.
- Que exista errores en el procesamiento de la información.
- Que exista demora de reportes para la toma de decisiones.
- Que exista retrasos en la ejecución de seguimiento y control de los activos.
- Pérdida de credibilidad en la administración de los activos fijos.

(VER ANEXO ÁRBOL DE PROBLEMAS)

1.3.- OBJETIVOS

1.3.1.- OBJETIVO GENERAL

Mejorar la Administración de los Activos Fijos mediante la implementación de un Sistema de Información de Activos Fijos en base a la metodología XP, para contribuir al proceso de la toma de decisiones de la Unidad de los Activos Fijos del Gobierno Autónomo Municipal de Cobija.

1.3.2.- OBJETIVO ESPECIFICO

- Determinar los requerimientos del sistema en base a la fase de la Planificación.
- Análisis de las especificaciones del sistema.
- Diseñar la estructura lógica del sistema.
- Diseño de la Base de Datos.
- Desarrollar módulos del sistema a partir de las especificaciones del diseño.
- Realizar pruebas del sistema y evaluar la calidad del sistema aplicando las métricas de calidad de la norma ISO 9126.
- Implantación del sistema de administración de Activos Fijos del Gobierno Autónomo Municipal de Cobija.

1.4.- JUSTIFICACIÓN

El crecimiento en la cantidad de los bienes muebles e inmuebles del Gobierno Autónomo Municipal de Cobija, y el volumen de información que la unidad de Activos Fijos posee actualmente, hace imprescindible el uso de la tecnología de la información que le permita operar con mayor eficiencia en la administración de los Activos Fijos. Por lo que este proyecto tiene por objetivo la implementación de un sistema de información de administración de activos fijos que permita automatizar el manejo y control de la información de una manera sistémica logrando alcanzar una mayor calidad y eficiencia en su administración.

1.4.1.- JUSTIFICACIÓN ECONÓMICA

El desarrollo del sistema de administración de activos fijos, tendrá un costo mínimo para su realización, contemplando así, la minimización en la reducción de gastos, tiempos, recursos humanos y materiales permitiendo agilizar el procesamiento de la información referidos al manejo de la administración de los Activos Fijos, esto con el fin de contribuir al Gobierno Autónomo Municipal de Cobija, y específicamente a la Unidad de Activos Fijos.

El costo mínimo que se considera para el desarrollo del proyecto, es para la compra de accesorios y material necesario para la implementación y validación del sistema de administración de activos fijos, como ser material para la capacitación al personal de la unidad de Activos Fijos en el manejo y administración del sistema, documentación del sistemas entre otros.

Estos factores ya mencionados presentan un beneficio económico para la institución del Gobierno Autónomo Municipal de Cobija, puesto que permite una mejor gestión eficiente de los activos fijos.

1.4.2.- JUSTIFICACIÓN SOCIAL

Este proyecto propone impulsar la implementación de las TIC'S en la administración pública el cual viene impulsando desde el Gobierno Central, el mismo favorecerá a todos los funcionarios públicos que asumen toda responsabilidad de dirigir y administrar las instituciones públicas de manera eficiente.

Con la implementación del sistema de administración de bienes del estado, nos permitirá en gran manera llevar una administración pública de una manera eficiente en todos los niveles posibles.

Es así que llevar una administración eficiente, permitirá al gobierno autónomo municipal de cobija mostrar una mayor credibilidad, responsabilidad, transparencia y confianza, el cual beneficia a todo la población.

1.4.3.- JUSTIFICACIÓN TÉCNICA

El sistema de administración de activos fijos a desarrollarse, mejorara su rapidez en cuanto a los reportes en tiempo real y al instante, estableciendo una importante y

eficiente administración de los bienes del estado. Así como también el sistema contará con políticas de seguridad, con claves personales para cada usuario del sistema, brindando una seguridad a la base de datos. De igual manera también podrá realizar respaldos de la base de datos.

El Hardware y Software mínimo necesario para el funcionamiento del sistema de administración de activos fijos es el siguiente:

- Micro-procesador: Core 2 Duo.
- Memoria RAM: 512 Mb.
- Tarjeta de Red: 10/100 Base – T
- Disco Duro: 100 Gb.
- Sistema Operativo: Linux
- Gestor de Base de Datos: MySQL
- PAL
- Lector de Código de Barras
- Impresora de Código de Barras

El sistema de administración de activos fijos trabajara sobre una red LAN, bajo la arquitectura Cliente-Servidor, el cual permite conectar varias computadoras clientes a un servidor, al mismo tiempo.

En otros puntos, el sistema de administración de activos fijos será implementado sobre software libre con código cerrado.

1.5.- METODOLOGÍAS

El método de desarrollo de software que se empleara, será la metodología XP (Programación Extrema).

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Es un método de desarrollo de software, que no está dedicado a un único lenguaje determinado. Es ágil y determinado, y no se rige por una metodología estándar, si no que coge aquellas características más destacadas, las mejores de cada una y se adapta a los requisitos en cualquier punto de vida del Software.

Esta metodología (XP) está orientada a objetos y es una de las alternativas en cuanto a productos a corto plazo.

1.5.1.- FASES DE LA METODOLOGÍA XP

Fase: Planificación del Proyecto.

La fase planificación del proyecto es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán llevar a cabo las historias de los usuarios, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas.

Fase: Diseño.

La metodología XP hace especial énfasis en los diseños simples y claros. El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

Mientras la primera fase de la planificación del proyecto se encuentre en producción, en la fase del Diseño se elaborara más diseños para encontrar un diseño adecuado para el software y así realizar con éxito la codificación y las pruebas.

Fase: Codificación.

Es necesario codificar y plasmar nuestro diseño a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar nuestra fase del diseño, y para hacer comunes las ideas, y por tanto para aprender y mejorar.

Crear un test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado

para ese fin. Se puede dividir la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco conseguiremos un desarrollo que cumpla todos los requisitos especificados.

Fase: Pruebas.

En la fase de las pruebas nos da a conocer como resultado final, que todos los requerimientos que se han solicitado han sido implementados satisfactoriamente.

Las pruebas dan la oportunidad de saber si lo que se ha desarrollado es lo que en realidad se tenía en mente. Las pruebas indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.

A continuación se representara mediante una tabla, un resumen de la Metodología XP:

Etapas	Metodología XP	Herramientas	Productos
1	Planificación del Proyecto	Historias de Usuario	Definir las historias de los usuarios con los clientes.
		Release Planning	Se indicaran las historias de los usuarios que se crearan para cada versión del programa.
2	Diseño del Sistema	Casos de Uso	Documentación del análisis y diseño del sistema.
		Uso de Tarjetas "CRC"	
		UML, Jdeveloper, Visio, etc	Diseños simples y sencillos del sistema mediante prototipos.

3	Codificación	Netbeans, PHP, HTML, MySQL, etc.	Módulos construidos y funcionando de acuerdo al diseño.
4	Pruebas	Técnica de la Caja Negra	Documentación de las pruebas de Caja Negra.
		Técnica de la Caja Blanca	Documentación de las pruebas de Caja Blanca.

TABLA 1: Tabla Resumen de la Metodología XP

FUENTE: Elaboración Propia

1.6.- ALCANCES

De acuerdo a estudio y análisis realizado en la unidad de Activos Fijos del G.A.M.C. El proyecto se limitará a las siguientes funcionalidades:

- Módulo Administración Parámetros del Sistema de Activos Fijos
- Módulo Administración Usuarios
- Módulo Administración de Activos Fijos
- Módulo Reportes
- Modulo Seguridad

Cada módulo mencionado constara de los siguientes componentes:

MÓDULOS	COMPONENTES
Módulo Administración Parámetros del Sistema de Activos Fijos.	<ul style="list-style-type: none"> • Unidades (Direcciones, Oficinas y Responsables) • Usuarios • Cargos • Rubros (Categorías) • Tipos de Entrada • Tipos de Salida

	<ul style="list-style-type: none"> • Tipos de Documentos • Ubicación geográfica
Módulo Administración Usuarios	<ul style="list-style-type: none"> • Usuarios del Sistema • Usuarios Habilitados
Módulo Administración Activos Fijos	<ul style="list-style-type: none"> • Activos Fijos Donaciones • Activos Fijos Adquisición • Activos Fijos Bajas • Activos Fijos Transferencias • Activos Fijos Revaluó • Activos Fijos Duplicados • Activos Fijos Grupos y Auxiliares Contables • Activos Fijos Oficinas y Responsables • Activos Fijos Unidades Administrativas
Módulo de reportes	<ul style="list-style-type: none"> • Buscar Activos Fijos • Reporte por Código de Activo • Reporte por Oficina y Responsable • Reporte por Grupo y Auxiliar Contable • Código de Barras • Depreciaciones
Modulo seguridad	<ul style="list-style-type: none"> • Autenticación de Usuarios • Protección contra ataques (Sql Injection, XSS)

El sistema se implementara bajo la arquitectura de Cliente – Servidor, el cual permite conectarse a varios usuarios al mismo tiempo a un mismo servidor y llevar diferentes procesos en él.

1.7.- APORTES

Con el presente proyecto se pretende alcanzar los siguientes aportes:

El presente proyecto será un gran aporte en el campo de innovación tecnológica con el uso de las nuevas tecnologías de información y comunicación en la administración pública, implementando un gestor generador de código de barras el cual facilitará la administración de los activos fijos de una manera óptima.

Al emprender la investigación sobre los activos y su movimiento en las entidades públicas, se considera implementar un sistema como una herramienta que facilite la administración de los activos fijos logrando una mejor confiabilidad, credibilidad, viabilidad, y transparencia, así como también brindar un apoyo fundamental a lo que es la toma de decisiones dentro de la unidad de Activos Fijos del G.A.M.C.



CAPÍTULO II

MARCO TEORICO



2.- CAPITULO II: MARCO TEORICO

2.1.- ANTECEDENTES TEÓRICOS, LEGALES Y CONTEXTUALES

2.1.1.- ASPECTOS TEÓRICOS

2.1.1.1.- ACTIVOS FIJOS

Los Activos son recursos económicos de propiedad de una **entidad u organización** y que de ellos se espera, beneficien las operaciones futuras de la misma.

El valor del activo se determina en base al costo de adquisición del artículo más todas las erogaciones necesarias para el funcionamiento del mismo.

Los activos con vida útil de más de un año que se adquieren para ser utilizados en la operación de una entidad o un negocio, con el fin de obtener ingresos se les conocen como **activos fijos**.

Los activos pueden tener y no tener forma física, a estos activos se clasifican en dos grupos:

a) Activos Tangibles:

Es aquel activo que es perceptible a los sentidos, tiene forma física y material.

Este tipo de activos a excepción de los terrenos, sirven a las compañías, empresas, organizaciones, etc., durante un número limitado de años.

b) Activos Intangibles:

El activo intangible es aquel que no posee sustancia o forma física. Los tipos más conocidos de activos intangibles son los fondos de comercio y patentes.

Estos activos a veces se adquieren por cantidades sustanciales de dinero, su valor para el comprador reside en el uso que pueda hacer de ellos.

2.1.1.2.- OBJETIVO DE LOS ACTIVOS FIJOS

El objetivo de los activos fijos es:

- El uso o usufructo de los mismos en beneficio de la entidad económica.
- La producción de artículos o uso de la misma.
- La prestación de servicio de la entidad al público en general.
- La adquisición de estos bienes denota el propósito de utilizarlos y no venderlos en el curso normal de las operaciones de la entidad, aunque con el paso del tiempo los activos se deterioran y pueden venderse, donarse o simplemente darse de baja para la adquisición de otros nuevos.

2.1.1.3.- DEPRECIACIÓN DE LOS ACTIVOS FIJOS

Existen diferentes métodos de depreciación de los cuales cada organización o entidad utilizara los que más les convenga:

Métodos Lineales:

- Línea Recta
- Unidades Producidas

Métodos Acelerados:

- Doble salto decreciente
- Suma de años dígitos

(Contabilidad Financiera. López Méndez. 1999).

2.1.1.4.- CONTABILIZACIÓN DE LOS ACTIVOS FIJOS

Las cuentas contables que se utilizan para los activos fijos en una organización o entidad, varían de acuerdo al sistema de contabilización que estarían utilizando, muchas de estas cuentas dependen de las políticas del gobierno de estado en la administración de los recursos del estado nacional y la **clasificación de los activos fijos** del mismo.

Para tener un concepto más sencillo, la contabilización de los activos se realizan de forma automática al momento de crear una solicitud de compra de un activo se le

imputa una cuenta contable, así como a un usuario y un centro de costos de acuerdo a la gerencia en la que pertenece el solicitante.

Una vez realizada dicha solicitud y autorizada de manera electrónica se procede a realizar una orden compra, la cual se libera de forma electrónica y se imprime para proporcionar al proveedor que surtirá dicha requisición de compra o de activo fijo, y por último el proveedor anexara dicha orden de compra a su factura de pago.

2.1.1.5.- RESGUARDO

Es el control de un formato en el cual contiene todos los datos de uno o varios activos como son, su serie, marca, inventario, modelo, así como la localización de su usuario puesto, gerencia a la que pertenece, centro de costo, y ubicación geográfica.

Además, este sirve para crear un responsabilidad del cuidado y resguardo o custodia de los activos asignados por la organización para el cumplimiento de sus funciones de acuerdo al puesto de trabajo.

2.1.1.6.- CONCILIACIÓN

Sirve para realizar una comparación entre los activos físicos de una o varias clases, así mismo esta se puede generar por un tiempo determinado o por un centro de costos específicos.

Esta conciliación nos sirve para saber si todo lo que se ha comprado se ha asignado o se está utilizando, así mismo sirve para determinar las diferencias por bajas, obsoletos o reutilizables y poder determinar el uso de estos.

También sirve para llevar un control de la depreciación de acuerdo a los centros de costos por gerencias por otro lado, para determinar si todos los activos que nos enviaron del corporativo están cargados contablemente o nos faltan o nos cargaron de más y esto impacta en resultados por las depreciaciones, presupuestos, etc.

2.1.2.- ASPECTOS LEGALES

2.1.2.1.- LEY 1178

La **LEY 1178** de Administración y Control Gubernamentales fue aprobada y consolidada por el consenso político el 20 de Julio de 1990.

Esta ley 1178, establece un modelo de administración y control para regular el funcionamiento de las entidades del Sector Público.

Según la **LEY 1178**, el **Sistema de Administración de Bienes y Servicios (SABS)** es un conjunto de normas de carácter administrativo, técnico y jurídico, que regulan en forma interrelacionada con el **Sistema de Administración de Activos Fijos** y otros subsistemas de administración, con la finalidad de tener control de los recursos del estado, manejo y disposición de Bienes y Servicios de las entidades públicas.

2.1.2.2.- SISTEMA DE ADMINISTRACIÓN DE ACTIVOS FIJOS

Según **Decreto Supremo N° 0181**, la Administración de Activos Fijos se divide en dos grupos:

Administración de Activos Fijos Muebles

La administración de activos fijos muebles, es la función administrativa que comprende actividades y procedimientos relativos al ingreso, asignación, mantenimiento, salvaguarda, registro y control de bienes de uso en las entidades públicas.

Objetivo

Tiene por objetivo lograr la racionalidad en la distribución, uso y conservación de los activos fijos muebles de las entidades públicas.

Alcance

Las disposiciones contenidas en este capítulo, se aplicaran a todos los activos fijos muebles de propiedad de la entidad y los que estén a su cargo o custodia.

Administración de Activos Fijos Inmuebles

La administración de activos fijos inmuebles, es la función administrativa que comprende actividades y procedimientos inherentes al uso, conservación, salvaguarda, registro y control de edificaciones, instalaciones y terrenos.

Objetivo

Tiene por objetivo lograr la racionalidad en el uso y conservación de las edificaciones, instalaciones y terrenos de las entidades públicas, preservando su integridad, seguridad y derecho propietario.

Alcance

Las disposiciones de este capítulo se aplicaran a todos los bienes inmuebles de propiedad de la entidad pública y los que están a su cargo o custodia.

2.1.2.3.- CLASIFICACIÓN DE LOS ACTIVOS FIJOS

De acuerdo al **Clasificador Presupuestario 2012**, proveniente del **Ministerio de Economía y Finanzas Públicas** tiene la siguiente clasificación:

40000 ACTIVOS REALES

41000 INMOBILIARIOS

41100 Edificios

41200 Tierras y Terrenos

41300 Otras Adquisiciones

43000 MAQUINARIA Y EQUIPO

43100 EQUIPO DE OFICINA Y MUEBLES

43110 Equipo de Oficina y Muebles

43120 Equipo de Computación

43200 MAQUINARIA Y EQUIPO DE PRODUCCION

43300 EQUIPO DE TRANSPORTE, TRACCION Y ELEVACION

43310 Vehículos Livianos para Funciones Administrativas
43320 Vehículos Livianos para Proyectos de Inversión Pública
43330 Maquinaria y Equipo de Transporte de Tracción
43340 Equipo de Elevación
43400 EQUIPO MEDICO Y DE LABORATORIO
43500 EQUIPO DE COMUNICACION
43600 EQUIPO EDUCACIONAL Y RECREATIVO
43700 OTRA MAQUINARIA Y EQUIPO

49000 OTROS ACTIVOS FIJOS
49100 ACTIVOS INTANGIBLES
49300 SEMOVIENTES Y OTROS ANIMALES
49400 ACTIVOS MUSEOLOGICOS Y CULTURALES
49900 OTROS ACTIVOS FIJOS

2.1.2.4.- VALOR ECONÓMICO DEL BIEN

Para la asignación del valor económico a los diferentes bienes, se deberá considerar los siguientes puntos:

- **Desgaste del Bien:** En esta parte se deberá tener mucho cuidado al tomar en cuenta las características físicas del bien, siendo un requisito principal el de cumplir durante un año o más, todas las funciones para las que ha sido destinado.
- **Valor Económico del Bien:** En este punto se ha visto conveniente asignar el valor de 1.00 Bs. A los muebles que ya tienen su vida útil concluida y que tengan desperfectos irreparables o que en el transcurso de su vida útil sufren un daño pero que siguen siendo utilizados.

- **Ejemplo:**

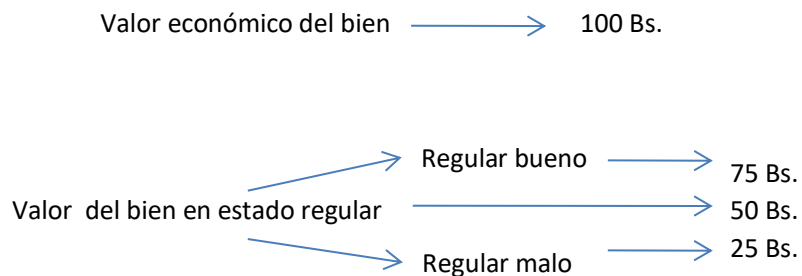


FIGURA 1. Representación del Valor Económico del Bien

FUENTE: Elaboracion Propia

2.1.2.5.- INCORPORACIÓN Y DESINCORPORACIÓN

- **Incorporaciones:** Se entiende por incorporaciones a toda adición de activos debido a:

- ✓ **Compra**
- ✓ **Traspaso**
- ✓ **Permutas**
- ✓ **Donaciones**
- ✓ **Omisión de Inventarios**
- ✓ **Adición y Mejoras**

- **Desincorporaciones:** Se entiende por desincorporación a todo retiro de Activo Fijo por los siguientes motivos:

- ✓ **Remate o Venta**
- ✓ **Traspaso**
- ✓ **Permuta**
- ✓ **Venta**
- ✓ **Donaciones**

2.1.2.6.- CONTROL DE INVENTARIOS

Con la finalidad de mantener registros permanentes y detallados, retiros, bajas, etc., de los activos fijos que pertenecen al Gobierno Municipal. Y al mismo tiempo impedir el manejo incorrecto de los activos, es necesario el levantamiento físico.

La responsabilidad del manejo, control y custodia, estará a cargo única y exclusivamente de la **Oficialía Mayor Administrativa y Financiera** a través de la **Unidad de Bienes y Servicios**, como lo indica el **Art. 2** (De la Responsabilidad en la Administración de Activos Fijos) del **reglamento específico** del Sistema de Inventariación de Activos Fijos. (**SIAF**).

- **Paso I: Revisión de Inventarios Anteriores**
- **Paso II: No existencia de Inventarios Anteriores**
- **Paso III: Verificación Física de los Activos**
- **Paso IV: Elaboración del Nuevo Inventario**
- **Paso V: Incorporación del Inventario al SIAF**

2.1.2.7.- CODIFICACIÓN DE LOS ACTIVOS FIJOS

A través de la apropiación de un código alfanumérico de identificación, el presente codificador tiene por objetivo estandarizar todos los procedimientos de control y registros, para salvaguardar los bienes que forman parte del activo fijo.

Ejemplo: Modelo de codificación de activos fijos del G.A.M.C. (Gobierno Autónomo Municipal de Cobija).

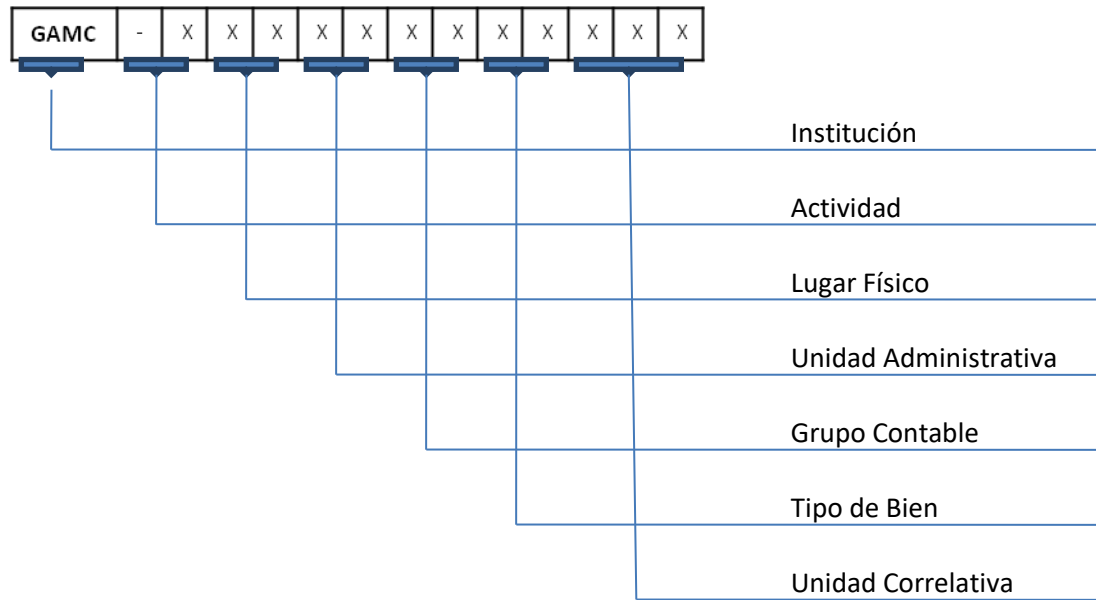


FIGURA 1.2. Representación del Modelo de Codificación de Activos Fijos del G.A.M.C.

FUENTE: Elaboración Propia

2.1.2.8.- DEPRECIACIÓN DE LOS ACTIVOS FIJOS

Es la distribución del costo u otro valor básico del activo tangible, menos el valor de salvamento si hubiera, a través de la vida útil probable de la unidad y en forma sistemática y racional [FUN, 1998].

El importe de la depreciación no debe deducirse directamente del costo del activo, si no debe acreditarse a una cuenta complementaria como: “depreciación acumulada”, por dos razones que se indican continuación:

- a) La depreciación contribuye una pérdida estimada del valor de un bien de uso (Activo Fijo) tangible y su importe no es exacto, si no aproximado.
- b) Es el costo del activo menos la depreciación acumulada, de esta manera el valor que refleja el Balance General o Posición Financiera es el valor neto del activo a una determinada fecha.

Cálculo de depreciación: Para efectuar el cálculo de la depreciación de un bien deben tomarse en cuenta lo siguiente [TER, 1997]:

- ✓ Costo actualizado del bien
- ✓ Valor de desecho (si corresponde)
- ✓ Años de vida útil estimados
- ✓ Disposición legal

I. Costo Actualizado del bien: El costo actualizado del bien está constituido por el precio de compra (según factura menos el IVA) mas desembolsos incidentales (instalación, horarios de técnicos, pruebas, etc.) más si corresponde el costo financiero (intereses y comisiones sobre prestamos financieros) y adicionarle la re expresión de valor en función a las fluctuaciones en la cotización del dólar Estadounidense [TER, 1997].

II. Valor de desecho: Conocido también como valor de salvamento o valor de rescate, es el importe mínimo de un bien después de cumplir los años de vida útil estimados, que se espera obtener al quedar fuera de servicio (por venta o baja) [TER, 1997].

III. Años de Vida Útil Estimados: constituye el tiempo máximo de duración de los bienes nuevos, para prestar servicio, variando entre ellos por las características de cada uno de estos [TER, 1997].

IV. Disposición Legal: Para efectuar la depreciación de vehículos automotores se deberá tomar muy en cuenta lo tipificado en cuenta lo tipificado en Artículo 60, Capítulo II, Título IV Impuesto a la propiedad de bienes inmuebles y vehículos automotores del Texto Ordenado de la Ley N° 843 [TER, 1997].

2.1.2.9.- MÉTODO DE DEPRECIACIÓN

Se denomina método de depreciación a la mecánica utilizada para proporcionar el costo actualizado del bien tomando en cuenta sus años de vida útil estimados, horas de trabajo, unidades de producción o cualquier otro parámetro aceptado por normas contables [TER, 1997].

En el presente proyecto para la depreciación de los activos fijos se usara el siguiente método de depreciación.

Denominaciones:

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

D = Depreciación anual

C = costo

VR = Valor residual o desecho

n = Años de útil del activo

2.1.2.10.- REVALORIZACIÓN TÉCNICA DE ACTIVOS FIJOS

Es un procedimiento reconocido contablemente, a través del cual los peritos independientes asignan nuevos valores o determinan justiprecios a estos mas los correspondientes años de vida útil residual en función al estado de conservación [TER, 1997].

Los objetivos de la revalorización técnica de activos fijos son:

- ✓ Asignar nuevos valores a los bienes
- ✓ Asignar años de vida útil residual
- ✓ Cumplir con disposiciones legales y normas contables

2.1.2.11.- BAJA DE ACTIVOS FIJOS

Se denomina baja de bienes de uso al retiro de estos por encontrarse en condiciones no aptas para prestar servicio útil a una empresa. Se puede originar tal situación, pro

inclemencias climatológicas, por siniestros, por obsolescencia o por haber cumplido con su vida útil [TER, 1997].

Además; se deberá seguir el siguiente procedimiento:

- 1.- Cumplir con lo estipulado en el Decreto Supremo N° 0181
- 2.- Actualizar valores hasta la fecha cuando se realiza la baja.
- 3.- Efectuar cálculos de la depreciación y su registro hasta la fecha cuando se realiza la baja, tomando en cuenta si es revaluado o no.
- 4.- Determinar el valor residual para efectuar cuenta de gasto cuando no se recupere o caso contrario cuenta de activo (por cobrar) si existe la posibilidad de recuperación.
- 5.- Preceder a la preparación del registro contable.

2.1.3.- ASPECTOS CONTEXTUALES

2.1.3.1.- SISTEMA

[Mcl, 2000] Un sistema es un grupo de elementos que se integran con el propósito común de lograr un objetivo.

2.1.3.2.- SISTEMAS DE INFORMACIÓN

Un **sistema de información** se define como un “conjunto de procedimientos ordenados interrelacionados que obtiene, procesa, almacena, y distribuye información para apoyar la toma de decisiones y el control de la Institución”.

La **información** se define como una entidad que permite reducir la incertidumbre acerca de algún estado o suceso.

[Campos, 1974] dice que los sistemas de información constan de los siguientes elementos:

- **La Información (datos).**- Conjunto de datos estructurados según los mensajes a comunicar.

- **Los Beneficiarios de la Información.**- Los miembros de la organización y agentes relacionados con ella.
- **Los elementos soporte:** Proceso de tratamiento de información, sistemas de análisis de datos, procedimientos de comunicación o difusores de información y soportes de información.

2.1.3.3.- SISTEMAS DE INFORMACIÓN AUTOMATIZADO

Se dice **sistema de información automatizado** a aquellos sistemas que incorporan a su entorno las computadoras.

[O'Brien, 2001], define como “Una combinación organizada de personas, mecanismos físicos (Hardware), procedimientos e instrucciones de procesamiento de información (Software), canales de comunicación (Redes) y datos almacenados (recursos de datos) que reúne, transforma y disemina información en una organización”.

2.2.4.- ACTIVIDADES QUE REALIZA UN SISTEMA DE INFORMACIÓN

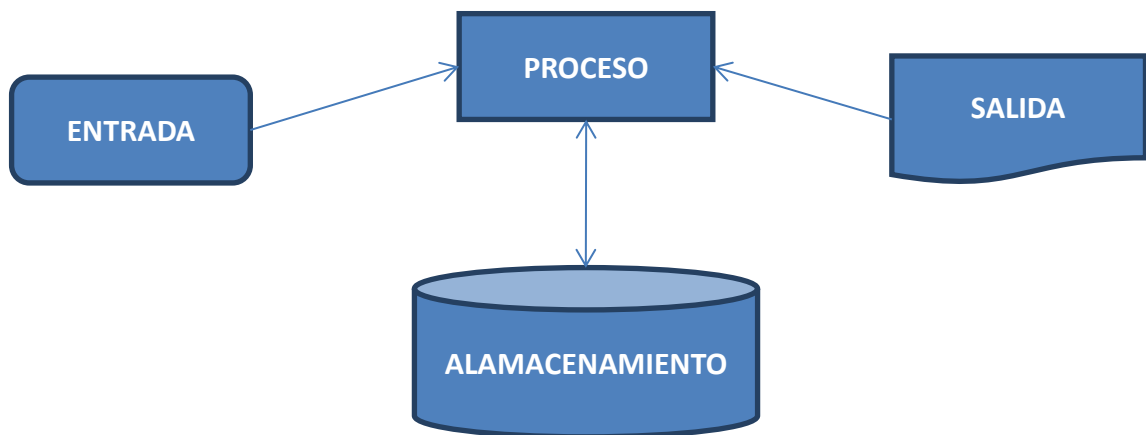


FIGURA 2. Representación grafica de las actividades de un S.I.

FUENTE: Elaboración Propia

[Peralta, 2008]. Un sistema de información realiza cuatro actividades básicas: **entrada, almacenamiento, procesamiento y salida de información.**

2.2.4.1.- Entrada de Información:

Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáneres, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

2.2.4.2.- Almacenamiento de información:

El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede salvaguardar la información transmitida al sistema de información. Esta información suele ser almacenada en estructuras de información denominadas archivos, Las unidades típicas de almacenamiento son los discos magnéticos o discos duros, los discos compactos (CD-ROM), entre otros.

2.2.4.3.- Procesamiento de Información:

Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

2.2.4.4.- Salida de Información:

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

2.2.5.- TIPOS DE SISTEMAS DE INFORMACIÓN

Con el incremento de la información y la automatización de cada proceso dentro las entidades u organizaciones, se ha dado lugar a la especialización para cada tipo de sistema de información dentro del nivel jerárquico que le corresponda, y esto con la finalidad de cumplir objetivos a un nivel optimó y eficiente.

2.2.5.1.- Sistema de Procesamiento de Transacciones

Este sistema es con frecuencia el primer tipo de especialidad que se implanta dentro de una organización, suelen ser recolectoras de información, y permiten mejorar el registro de las transacciones rutinarias eliminando las tareas tediosas. Los datos ingresados se almacenan en un archivo de datos o una base de datos, esta, puede servir de apoyo a otros sistemas de información.

2.2.5.2.- Sistema de Información Administrativa

A diferencia de los sistemas de procesamiento de transacciones estas hacen uso real de la información almacenada, es decir genera reportes estandarizados a partir de la base de datos implementada en los sistemas transaccionales, que sirven de apoyo a los mandos intermedios y toma de decisiones por parte de los altos ejecutivos.

Para poder ligar la información, los usuarios de un sistema de información gerencial, comparte una base de datos común. La base de datos guarda modelos que ayudan a los usuarios a interpretar y aplicar esos mismos datos. Los sistemas de información gerencial producen información que es utilizada en la toma de decisiones [**Ibidem, 2000**].

2.2.5.3.- Sistema de Apoyo Ejecutivo o Soporte de Decisiones

Son herramientas orientadas a usuarios de nivel ejecutivo, que permite monitorizar el estado de las variables de un área o unidad de la empresa a partir de la información interna y externa a la misma. Estos sistemas permiten a los ejecutivos obtener respuesta a problemas inesperados y relativamente excepcionales.

Entre los sistemas de información con mayor alcance en las entidades públicas son los Sistemas de Contabilidad, generalmente para poder contar con la información transparente y oportuna.

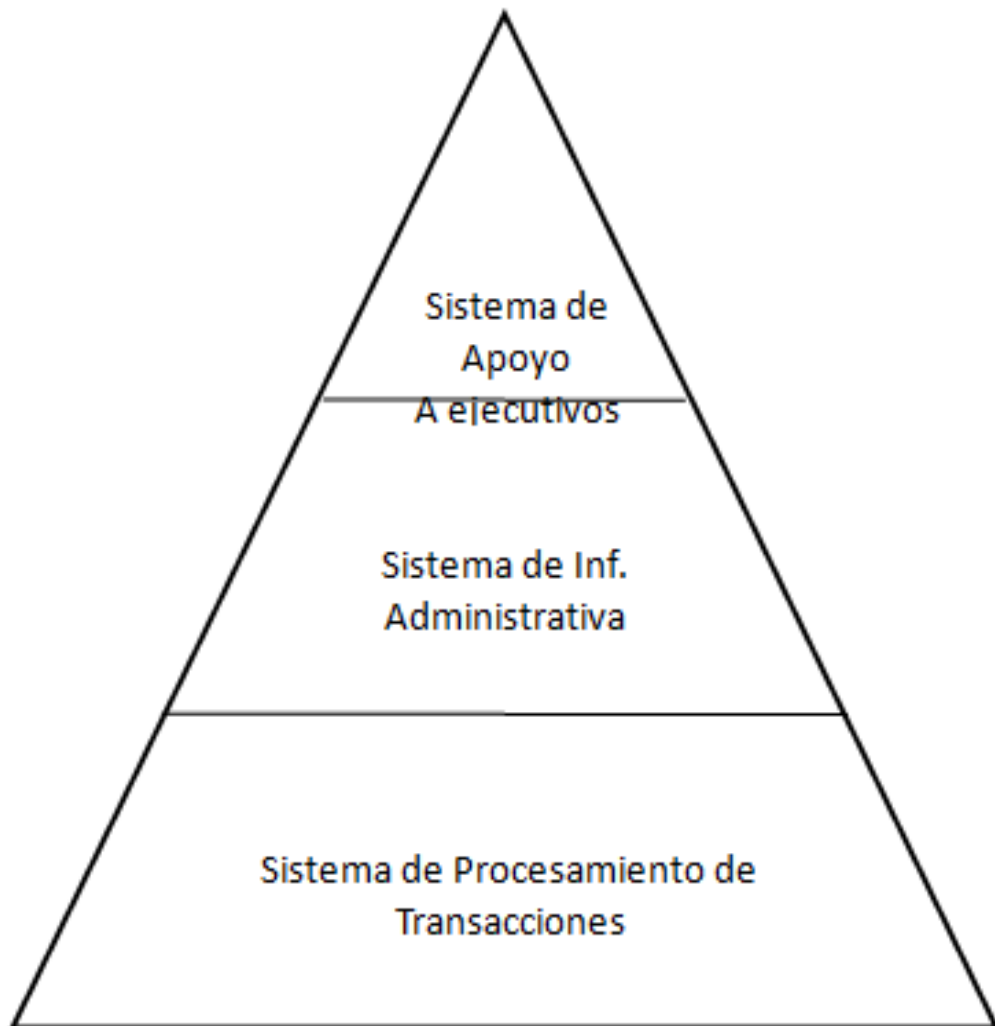


FIGURA 2.1.Especialización de los Sistemas de Información

FUENTE: Bedriñana A. (2004)

2.2.6.- SEGURIDAD DE SISTEMAS DE INFORMACIÓN

[FJRP, 2005], Los sistemas de información concentran los datos en archivos de computadoras, por ello, los datos automatizados son más susceptibles a destrucción, fraude, error y abuso. Cuando los sistemas de computación fallan o no funcionan como es debido, las compañías que dependen mucho de ellos experimentan la pérdida grave de su capacidad para operar.

Porque son vulnerables los sistemas de información: Cuando se almacenan grandes cantidades de datos en forma electrónica, estos son vulnerable a muchos tipos de amenazas. Su origen puede estar en factores técnicos, de organización y del entorno, combinados con las malas decisiones gerenciales.

Amenaza a los sistemas de información computarizados: Fallos de hardware, Incendio, Fallos de software, Problemas eléctricos, Acciones del personal, Errores de usuario, Penetración por terminales, Cambios de programas, Robo de datos, servicios, equipo, Problemas de telecomunicaciones.

Aspectos importantes para prevenir amenazas a los sistemas de información: Dentro de los aspectos más importantes tenemos tres puntos.

- **Protección del Hardware:** Asegurar la integridad y disponibilidad de los elementos del hardware, se tratara básicamente de medidas de prevención y, ocasionalmente, de recuperación físicas.

- **Protección de los Datos:** Asegurar la integridad de la información contra ataques, amenazas que pueden dar lugar a la destrucción de la información almacenada. Este aspecto es el que más trabaja un sistema de información, debido a que los datos se encuentran constantemente en actualización.

Seguridad SSL: El protocolo SSL (Secure Socket Layer), desarrollado por la empresa Netscape Communications, protege la información que se comparte entre el servidor y el cliente.

Gracias a los certificados SSL permiten que se forme un lazo de confianza protegiendo los datos del cliente mediante su encriptación y dejando disponible el acceso a la información sólo mediante claves.

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar.

SSL implica una serie de fases básicas:

- ✓ Negociar entre las partes el algoritmo que se usará en la comunicación.
- ✓ Intercambio de claves públicas y autenticación basada en certificados digitales.
- ✓ Cifrado del tráfico basado en cifrado simétrico.

- **Seguridad de los Usuarios:** Sin duda uno de los puntos más débiles de los sistemas de información.

Unos de los ataques relacionados con el personal son:

La Ingeniería Social (Manipulación de las personas donde realizan actos que normalmente nunca harían).

Fisgoneo (Espionaje de la actividad de los usuarios).

Suplantación de la identidad de un usuario legítimo (electrónicamente o en persona) cuando el control y la verificación de accesos es demasiado débil.

Basureo Obtención de la información abandonada.

2.3.- METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software.

Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software.

Estas se podrían clasificar en dos grandes grupos:

- Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas **Metodologías Pesadas**.
- Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas **Metodologías ligeras/ágiles**.

2.3.1.- METODOLOGÍAS PESADAS

[Ricardo Colusso, 2011]. Afirma que son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida. Una de las metodologías pesadas más conocidas y utilizadas es la **Metodología RUP**.

La forma tradicional de desarrollar software se basa en procesos predefinidos con documentación muy precisa, y una detallada planificación inicial que debe seguirse estrictamente. Esta forma de trabajar surgió naturalmente hace unos cincuenta años como una adaptación del manejo de proyectos de ingeniería, que era lo más parecido a desarrollar programas que se conocía en ese momento, y funcionó razonablemente bien

en un comienzo. También es necesario tener en cuenta que los ordenadores era enormemente caros, la mayor parte de la inversión informática se la llevaban los equipos y por esta razón los programas se hacían a medida para unas máquinas que se adquirirían, no lo olvidemos, para realizar unas tareas muy concretas.

Pero los proyectos de desarrollo de software en la actualidad incluyen desafíos muy diferentes a los que se presentan al construir puentes y casas, por lo que no sorprende que los métodos tradicionales de desarrollo de software estén en crisis.

Tradicionalmente los proyectos se dividen en etapas bien diferenciadas: Análisis de Factibilidad, Análisis de Requerimientos, Diseño, Programación, y Testeo.

Generalmente se trata de que haya retroalimentación (feedback en inglés) entre las etapas contiguas, de tal forma de que, por ejemplo, haya un momento en que se mejoren los Requerimientos en base a comentarios, sugerencias y necesidades de los responsables del Diseño. Sin embargo, esta forma de desarrollar software genera muy serios problemas, debido a que al comienzo del proyecto, que es cuando menos se conocen las características del problema que resolver, se toman las decisiones de mayor relevancia e impacto en el resto del proyecto.

Como las chances de tomar decisiones erróneas al comienzo del proyecto son generalmente mayores al principio de un proyecto que cuando ya se ha trabajado un tiempo, suele ocurrir que los proyectos desarrollados siguiendo el método tradicional no cumplen sus objetivos, no se terminan a tiempo, y resultan mucho más caros que lo presupuestado. En particular, esto ocurre con mayor frecuencia en los casos en que el grupo de desarrollo necesita crear algo totalmente nuevo o de características específicas que nadie ha creado aún . . . lo cual es cierto para la mayoría de los proyectos de software! , porque en caso contrario, la organización compraría directamente un producto o sistema ya desarrollado anteriormente por otros.

Como propuesta de solución a estos problemas han surgido una serie de “métodos ágiles” de desarrollo de software y manejo de proyectos en general.

2.3.2.- METODOLOGÍAS AGILES

[Juan Gabardini, 2011]. Menciona que la metodología ágil como tal, nace en febrero del 2001 en una reunión celebrada en Utah EEUU.

¿Por qué tantos proyectos de desarrollo de software no se terminan a tiempo, cuestan más que lo presupuestado originalmente, tienen problemas de calidad serios y generan menor valor que el esperado?

Este interrogante fue uno de los que se formularon los 17 profesionales expertos en el desarrollo de software cuando se reunieron en febrero de 2001 para analizar el problema y decidieron redactar un “Manifiesto Ágil”. Se trató de un compromiso público en buscar nuevas y mejores formas de desarrollar software poniendo énfasis en las personas y sus interacciones, la colaboración y la respuesta continua al cambio, explorando nuevas formas de hacer las cosas, y compartiendo experiencias — dando origen a una nueva comunidad de profesionales que explora sistemáticamente nuevas alternativas frente al modo tradicional de desarrollar software.

Principales ideas de la metodología ágil:

- ✓ Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- ✓ Se hace mucho más importante crear un producto software que funcione que escribir mucha documentación.
- ✓ El cliente está en todo momento colaborando en el proyecto.
- ✓ Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan.

Dentro de esta metodología de desarrollo de software consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo.

Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida

entre todos los participantes, también busca simplificar las soluciones implementadas y coraje para los múltiples cambios.

Una de estas Metodologías de Desarrollo de Software es la **XP (Programación Extrema)**.

Metodologías Agiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestamente internamente (por el equipo)	Impuestas externamente
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Pocos artefactos.	Más artefactos.
Pocos Roles.	Mas Roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.
Grupos pequeños y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.

FIGURA 3. Tabla Diferencias entre Metodologías Agiles y Metodologías no Agiles

FUENTE: Elaboración Propia

2.4.- METODOLOGÍA XP

XP (Programación Extrema), es una metodología para el desarrollo de software, clasificada como uno de los más ligeros, en contraposición con las metodologías pesadas como el **Proceso Unificado**.

[**Kent Beck, 2000**], fue el desarrollador de esta metodología, en el cual se basa en la simplicidad, la comunicación y la retroalimentación o reutilización de código.

Entre los objetivos de esta metodología se encuentran la **satisfacción del cliente**, además de potenciar al máximo el **trabajo en equipo (clientes, desarrolladores, jefes)**.

[**Beck, 2000**] argumenta que: “Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder, el problema es la incapacidad de adaptarnos a dicho cambio cuando este tiene lugar”.

2.4.1.- FASES DE LA METODOLOGÍA XP

La XP (Programación Extrema), define **4 fases** bien definidas que permiten cubrir todo el proceso de desarrollo de software: Planificación, Diseño, Desarrollo y Pruebas.

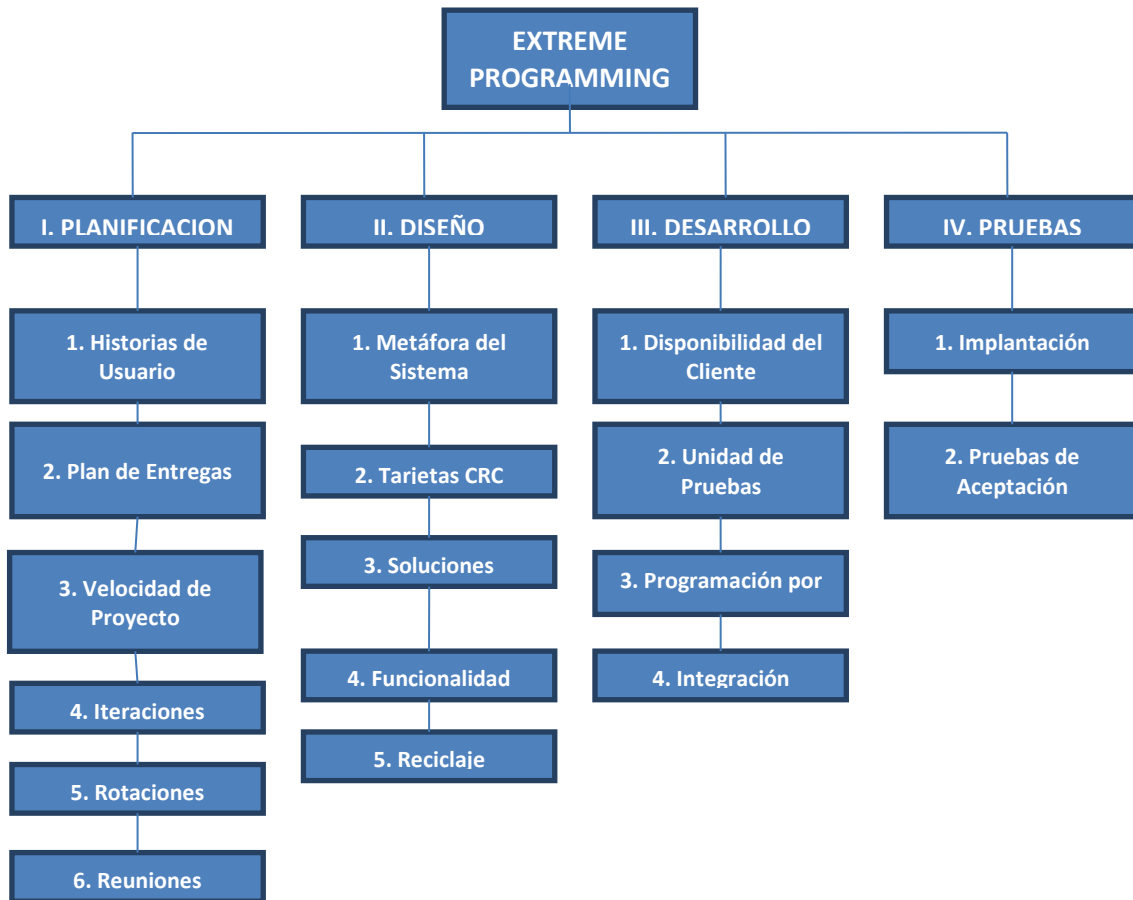


FIGURA 4. Fases de la Programación Extrema

FUENTE: Fernandez, Escribano. 2002

2.4.1.1.- FASE DE LA PLANIFICACIÓN

Es la etapa de interacción entre el cliente con la parte técnica. En esta fase el cliente dará a conocer lo realmente necesario y el orden de prioridad del proyecto a través de las **Historias de Usuario**, se define las versiones del software y el cronograma de actividades.

[Fernández, 2002], define como tres actividades a desarrollar:

- a) **Redactar las Historias de Usuarios:** Es donde se describe de manera sintetizada cierto proceso, en realidad tienen el mismo propósito que los **Casos de Uso** de la metodología **Proceso Unificado**.
- b) **Se crea un plan de entregas:** Es donde se planifica determinados tiempos requeridos para el desarrollo de las **Historia de Usuarios**.

- c) **Se divide el proyecto en iteraciones:** Es donde una iteración corresponde a una cierta cantidad de historias de usuario.

2.4.1.2.- FASE DE DISEÑO

Para la Fase de Diseño la Programación Extrema, define los siguientes principios:

- a) **Simplicidad:** Es donde se realiza las cosas de la forma más simple posible.
- b) **Tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración):** En estas tarjetas se describen **las clases, los métodos y la relación con otras clases.**

CLASE: Nombre de la Clase	
RESPONSABILIDAD:	COLABORACIÓN:
Método de la clase	Lista de clases con los que se relaciona

FIGURA 4.1. Tarjeta clase-responsabilidad-colaboración

FUENTE: Fernandez, Escribano. 2002

- c) **No se añade funcionalidad en las primeras etapas,** es decir, trabajar en lo que se ha fijado para el día.

2.4.1.3.- FASE DE DESARROLLO

En esta etapa se trabaja bajo los siguientes lineamientos:

- a) El cliente debe ser parte del proyecto, es decir está disponible cuando se lo necesite.
- b) Escribir el código de acuerdo a buenas prácticas de programación.
- c) Implementar la Base de Datos.

- d) Realizar las pruebas.

2.4.1.4.- FASE DE PRUEBAS

- a) Se efectuaran las siguientes pruebas:

1	Pruebas	Técnica de la Caja Negra	Documentación de las pruebas de Caja Negra. Ver ANEXO A
		Técnica de la Caja Blanca	Documentación de las pruebas de Caja Blanca. Ver ANEXO B
		Norma ISO 9126	Estándar internacional para la evaluación de la calidad del software. Ver ANEXO C

FIGURA 4.2. Tabla de Aplicación de Pruebas

FUENTE: Elaboración Propia

2.5.- ELEMENTOS DE LA IMPLEMENTACIÓN DE LOS SISTEMAS DE INFORMACIÓN

2.5.1.- ARQUITECTURA CLIENTE SERVIDOR

[Hurtado, 2006], define como un “Sistema donde el **Cliente** es una máquina que solicita un determinado servicio y se denomina **Servidor** a la maquina que lo proporciona”.

En este modelo de trabajo, las tareas son repetidas entre el equipo Servidor y el Cliente, son un típico ejemplo las aplicaciones que solicitan, a través de interfaces, información a un servidor de Bases de Datos alojado en otro equipo y que ambas están comunicadas por una red de datos.

2.5.2.- BASE DE DATOS

[Navarrete, 1999], dice que la Base de Datos es una **colección de información organizada, y relacionada entre ella**, esta, pueda estar constituido de gran cantidad de información, con posibilidad de acceder en cualquier momento y de manera concurrente, este acceso generalmente se realiza a través de instrucciones denominado consultas.

Una Base de Datos cuenta con las siguientes propiedades:

- a) Representación de algún aspecto del mundo real.
- b) Es un conjunto de datos lógicamente coherente e interrelacionado.
- c) Toda Base de Datos se diseña con un propósito específico.

2.5.3.- GESTOR DE BASE DE DATOS MySQL

MySQL es un sistema de gestión de bases de datos relacionales, multihilo y multiusuario con más de seis millones de instalaciones, desarrollada como software libre. MySQL es muy utilizado en aplicaciones web, como DRUPAL o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla.

Su popularidad como aplicación web está muy ligada al lenguaje de programación PHP, que a menudo aparece en combinación con MySQL.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- ✓ Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- ✓ Transacciones y claves foráneas.
- ✓ Conectividad segura.
- ✓ Replicación.
- ✓ Búsqueda e indexación de campos de texto.

2.5.4.- REDES DE DATOS

Las Redes de Datos son fuentes centrales de información y comunicación que tienen la posibilidad de compartir con carácter universal la información entre grupos de computadoras y usuarios.

Un componente vital de la era de la información, es la generalización del **ordenador o computadora personal (PC)** y de la **Red de área local (LAN)**, que durante la década de los 80 ha dado lugar a la posibilidad de acceder a información en bases de datos remotas.

2.5.5.- LENGUAJES DE PROGRAMACIÓN

[Araya, 2001]

Un lenguaje de programación se define como “una técnica de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora”.

2.5.5.1.- PROGRAMACIÓN ESTRUCTURADA

[Knuth, “The art of Computer Programming”, 1997].

La programación estructurada consiste en escribir un programa de acuerdo con unas reglas y un conjunto de técnicas. Las reglas son: el programa tiene un diseño modular, los módulos son diseñados descendentemente, cada módulo de programa se codifica usando tres estructuras de control (secuencia, selección e iteración); es el conjunto de

técnicas que han de incorporar: recursos abstractos; diseño descendente y estructuras básicas de control.

Descomponer un programa en términos de recursos abstractos consiste en descomponer acciones complejas en términos de acciones más simples capaces de ser ejecutadas en una computadora.

El diseño descendente se encarga de resolver un problema realizando una descomposición en otros más sencillos mediante módulos jerárquicos. El resultado de esta jerarquía de módulos es que cada módulo se refina por los de nivel más bajo que resuelven problemas más pequeños y contienen más detalles sobre los mismos.

Las estructuras básicas de control sirven para especificar el orden en que se ejecutarán las distintas instrucciones de un algoritmo. Este orden de ejecución determina el flujo de control del programa.

La programación estructurada significa:

- ✓ El programa completo tiene un diseño modular.
- ✓ Los módulos se diseñan con metodología descendente (puede hacerse también ascendente).
- ✓ Cada módulo se codifica utilizando las tres estructuras de control básicas: secuenciales, selectivas y repetitivas.
- ✓ Estructuración y modularidad son conceptos complementarios (se solapan).

2.5.5.2.- PROGRAMACIÓN ORIENTADA A OBJETOS

[Knuth, “The art of Computer Programming”, 1997].

La programación orientada a objetos aporta un nuevo enfoque a los retos que se plantean en la programación estructurada cuando los problemas a resolver son complejos. Al contrario que la programación procedimental que enfatiza en los algoritmos, la POO enfatiza en los datos. En lugar de intentar ajustar un problema al enfoque procedimental de un lenguaje, POO intenta ajustar el lenguaje al problema. La idea es diseñar formatos de datos que se correspondan con las características esenciales

de un problema. Los lenguajes orientados combinan en una única unidad o módulo, tanto los datos como las funciones que operan sobre esos datos. Tal unidad se llama objeto. Si se desea modificar los datos de un objeto, hay que realizarlo mediante las funciones miembro del objeto. Ninguna otra función puede acceder a los datos. Esto simplifica la escritura, depuración y mantenimiento del programa.

En el paradigma orientado a objetos, el programa se organiza como un conjunto finito de objetos que contienen datos y operaciones, que llaman a esos datos y que se comunican entre sí mediante mensajes.

Cuando se trata de resolver un problema con orientación a objetos, dicho problema no se descompone en funciones como en programación estructurada tradicional, caso de C, sino en objetos. El pensar en términos de objetos tiene una gran ventaja: se asocian los objetos del problema a los objetos del mundo real.

Propiedades fundamentales de la orientación a objetos: Existen diversas características ligadas a la orientación a objetos. Todas las propiedades que se suelen considerar, no son exclusivas de este paradigma, ya que pueden existir en otros paradigmas, pero en su conjunto definen claramente los lenguajes orientados a objetos. Estas propiedades son:

- ✓ **Abstracción (tipos abstractos de datos y clases):** La abstracción es la propiedad de los objetos que consiste en tener en cuenta sólo los aspectos más importantes desde un punto de vista determinado y no tener en cuenta los restantes aspectos.
- ✓ **Encapsulado de datos:** El encapsulado o encapsulación de datos es el proceso de agrupar datos y operaciones relacionadas bajo la misma unidad de programación. En el caso de que los objetos posean las mismas características y comportamiento se agrupan en clases (unidades o módulos de programación que encapsulan datos y operaciones).
- ✓ **Ocultación de datos:** La ocultación de datos permite separar el aspecto de un componente, definido por su interfaz con el exterior, de sus detalles internos de implementación. Los términos ocultación de la información

(information hiding) y encapsulación de datos (data encapsulation) se suelen utilizar como sinónimos, pero no siempre es así y muy al contrario son términos similares pero distintos.

- ✓ **Objeto:** Los objetos son las cosas físicas y conceptuales que encontramos en el universo alrededor de nosotros. Hardware, software, documentos, seres humanos, los conceptos son todos los ejemplos de los objetos.
- ✓ **Clases:** Las Clases son como plantillas o modelos que describen como se construyen ciertos tipos de Objeto. Cada vez que se construye un Objeto de una Clase, se crea una instancia de esa Clase ("instance"). Una Clase es una colección de Objetos similares y un Objeto es una instancia de una Clase. Se puede definir una Clase como un modelo que se utiliza para describir uno o más Objetos del mismo tipo.
- ✓ **Herencia:** Una característica muy importante de los Objetos y las Clases es la Herencia, una propiedad que permite construir nuevos Objetos (Clases) a partir de unos ya existentes. Esto permite crear "Sub-Clases" denominadas Clases Derivadas que comparten las propiedades de la Clase de la cual derivan (Clase base). Las Clases derivadas heredan código y datos de la clase base, asimismo incorporan su propio código y datos especiales. Se puede decir que la herencia permite definir nuevas Clases a partir de las Clases ya existentes.
- ✓ **Polimorfismo:** La propiedad de polimorfismo es aquella en que una operación tiene el mismo nombre en diferentes clases, pero se ejecuta de diferentes formas en cada clase. Así, por ejemplo, la operación abrir se puede dar en diferentes clases: abrir una puerta, abrir una ventana, abrir un periódico, abrir un archivo, abrir una cuenta corriente en un banco, abrir un libro, etc. En cada caso se ejecuta una operación diferente aunque tiene el mismo nombre en todos ellos "abrir". El polimorfismo es la propiedad de una operación de ser interpretada sólo por el objeto al que pertenece. Y en un sentido literal, Polimorfismo significa la cualidad de tener más de una forma.

2.5.5.3.- MODELO VISTA CONTROLADOR (MVC)

[Juan Pavón Mestras, 2008-2009]

Es un patrón de arquitectura de las aplicaciones de software.

Separa la lógica del negocio de la interfaz de usuario.

Facilita la evolución por separado de ambos aspectos.

Incrementa reutilización y flexibilidad.

El patrón MVC consta de tres partes:

- ✓ Un modelo.
- ✓ Varias Vistas.
- ✓ Varios Controladores.

Las vistas y controladores que suelen estar muy relacionados.

Los controladores tratan los eventos que se producen en la interfaz gráfica - vista).

Esta separación de aspectos de una aplicación da mucha flexibilidad al desarrollador.

Flujo de Control del Patrón MVC

- ✓ El usuario realiza una acción en la interfaz
- ✓ El controlador trata el evento de entrada
 - Previamente se ha registrado
- ✓ El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo (si no es una mera consulta)
- ✓ Se genera una nueva vista. La vista toma los datos del modelo
 - El modelo no tiene conocimiento directo de la vista
- ✓ La interfaz de usuario espera otra interacción del usuario, que comenzará otro nuevo ciclo

Ejemplo del Patrón MVC en aplicaciones web

- ✓ **Vista:** La pagina HTML (manejo de la parte visual)
- ✓ **Controlador:** Código que obtiene datos dinámicamente y genera el contenido HTML (manejo de las acciones).
- ✓ **Modelo:** La información almacenada en una base de datos o en XML junto con las reglas de negocio que transforman esa información (manejo de los datos).

2.5.5.4.- PHP

[Jim Winstead, 2001]

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

Una de las características más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP.

Las siguientes bases de datos que están soportadas actualmente son:

Adabas D	Empress	IBM DB2
Ingres	FrontBase	MySQL
Oracle (OCI7 and OCI8)	Unix dbm	Velocis
dBase	FilePro	Velocis
InterBase	mSQL	Informix
PostgreSQL	Sybase	ODBC

FIGURA 5. Tabla de bases de datos soportadas con PHP

FUENTE: Elaboración Propia

PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir sockets de red directos (raw sockets) e interactuar con otros protocolos.

2.5.5.5.- HTML5

[Castillo, 2007].

Las páginas web son documentos de texto, escritos con un lenguaje especial llamado HTML (Hypertext Markup Language). En este lenguaje, se utilizan etiquetas para codificar las distintas partes de la página, dar formato al texto, crear tablas, insertar imágenes o enlaces. Las etiquetas empiezan con el signo <, y acaban con >. Entre ambos signos se pone el nombre de la etiqueta (que identifica su función), así como atributos, que pueden modificar de algún modo su función. Las etiquetas se abren y se cierran. Al abrir se pone el nombre de la etiqueta (por ejemplo <HTML>), mientras que al cerrar se añade el signo / antes del nombre (</HTML>). Todo el contenido que se encuentra entre la apertura y cierre se ven afectados por la etiqueta. Las etiquetas se pueden escribir en mayúsculas o en minúsculas.

El **HTML5** (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje de programación “básico” de la World Wide Web, el HTML. Esta nueva versión pretende remplazar al actual (X)HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día.

Actualmente el HTML5 está en un estado BETA, aunque ya algunas empresas están desarrollando sus sitios webs en esta versión del lenguaje. A diferencia de otras versiones de HTML, los cambios en HTML5 comienzan añadiendo semántica y accesibilidad implícitas, especificando cada detalle y borrando cualquier ambigüedad. Se tiene en cuenta el dinamismo de muchos sitios webs (facebook, twenti, etc), donde su aspecto y funcionalidad son más semejantes a aplicaciones webs que a documentos.

2.5.5.6.- CSS3 (Cascading Style Sheets)

[Freddy Vega, Van Der Henst, 2011]

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Como ya mencionamos anteriormente el CSS es un lenguaje de hojas de estilo para controlar los documentos web como el HTML. En esta nueva versión, se desarrollo para trabajar con el HTML5, potenciando aun mucho mejor la WEB.

Las versiones nuevas de móviles, desktop de Facebook y otros sitios web, hacen un uso intensivo de CSS3 para diseño y animaciones.

2.5.5.7.- JAVASCRIPT

[Eguíluz, 2009].

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

En los foros web hablan sobre la extinción del flash, poniendo como una de las causas a JavaScript y Canvas, debido al gran avance con las animaciones y efectos web en tiempo real.

2.5.5.8.- MOOTOOLS

[AngelAlvarez, 2011].

Mootools es un framework Javascript, que sirve para crear fácilmente código Javascript independiente del navegador, de una forma rápida y directa.

Dicho de otra manera, que tal vez sea más fácil de entender, Mootools es un conjunto de librerías, también llamado API, que proveen clases de programación orientada a objetos en Javascript, para realizar una amplia gama de funcionalidades en páginas web, como trabajo con capas, efectos diversos, Ajax y mucho más. Con Mootools podemos programar todo tipo de scripts en el cliente rápidamente y sin preocuparnos de las distintas particularidades de cada navegador. Mootools está especialmente indicado para programar scripts complejos, que nos costaría mucho más trabajo de realizar si partiésemos de cero.

Existen diversos framework en Javascript, la mayoría ni he probado todavía. Podemos ver unos cuantos de ellos en el artículo Listado de distintos Framework Javascript. Mootools es una opción interesante, bastante difundida y que tiene una serie de ventajas como:

- **Ligero:** el framework no pesa demasiado en Kb y no el procesamiento carga poco al navegador.
- **Modular:** mootools se compone de diversos módulos y podemos seleccionar los que vamos a utilizar para incorporarlos en nuestras páginas web, dejando los otros para que no ocupen tiempo de descarga ni procesamiento.
- **Libre de errores:** podemos confiar en el sistema porque Mootools las herramientas de Mootools funcionan perfectamente, sin emitir errores en tiempo de ejecución.
- **Soportado por una amplia comunidad:** existen muchos desarrolladores que lo utilizan con éxito y han creado una serie de componentes adicionales ya listos para usar en nuestras páginas web, como calendarios, editores de texto, etc.

2.5.6.- ZEND FRAMEWORK

[Benjamín Gonzales - *Zend Technologies*] (2005-10) de: <http://manual.zfdes.com>

Introducción a Zend Framework:

Zend Framework es un framework de código abierto **para desarrollar aplicaciones web y servicios web con PHP5**. Zend Framework es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de Zend Framework es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse. Zend Framework ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como Zend_Auth y Zend_Acl, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes . También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los web services más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y StrikeIron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

Ventajas de un Framework:

- ✓ El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que "rellenar".
- ✓ Facilita la colaboración. Cualquiera que haya tenido que "pelearse" con el código fuente de otro programador (¡o incluso con el propio, pasado algún tiempo!) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.
- ✓ Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.

No necesito utilizar un Framework?

Es verdad que no es necesario que un programador necesite utilizar un Framework, puede que sea tan pequeña que no lo considere necesario, que no conozca ninguno que se adapte a sus necesidades, o simplemente no desee dedicar tiempo a seleccionar y utilizar uno.

Sin embargo, a medida que la aplicación crece, un programador competente procurará seguir unas determinadas pautas que le faciliten su trabajo de desarrollo y mantenimiento: separación de presentación y lógica, una sintaxis coherente, etc. La evolución natural será hacia que, de algún modo, se construirá su propio Framework.

Y en vez de definir un estándar, ¿por qué no utilizar uno ya definido?.

En conclusión, la utilización de un Framework en el desarrollo de una aplicación implica un cierto coste inicial de aprendizaje, aunque a largo plazo es factible que facilite tanto el desarrollo como el mantenimiento.

2.5.7.- PRUEBAS DE CALIDAD DE SOFTWARE

[Pressman, 2002].

Las pruebas son un elemento crítico para la calidad del software. La importancia de los costos asociados a los errores, promueve la definición y la aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiendo como validación del software el proceso, externo al equipo de desarrollo, que determina si el software satisface los requisitos, y verificación como el proceso interno que determina si los productos de una fase satisfacen las condiciones de dicha fase.

Este sistema, como cualquier otro en ingeniería, puede probarse de dos formas:

a) conociendo la función específica para la que fue diseñada, y b) conociendo el funcionamiento del producto.

El primer enfoque se centra en las llamadas pruebas de caja negra y el segundo en las pruebas de caja blanca.

Según lo definido por Pressman, las pruebas de caja negra se llevan a cabo sobre la interfaz del software. Se trata de demostrar que las funciones del software son operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado. Las pruebas de caja blanca se centran en la estructura lógica interna del software. Se basan en un examen detallado de los procedimientos y caminos lógicos del sistema.

Las pruebas realizadas al sistema se basan en el enfoque de **CAJA NEGRA** debido a las particularidades del sistema como lo es la arquitectura cliente-servidor, además de la dificultad exponencial de realizar pruebas de caja blanca basadas en el seguimiento de todos los caminos posibles. Pressman expone la dificultad de realizar pruebas de caja blanca a un sistema basado en cliente-servidor por los siguientes motivos aplicables a nuestros sistemas: la complejidad de las comunicaciones de red, la necesidad de servir múltiples instancias del sistema y la potencialidad de diferentes plataformas de hardware.

Las pruebas de caja negra buscan encontrar errores en cinco categorías [**Pressman, 2002**]:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores en estructuras de datos o en accesos a bases de datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de Inicialización y Terminación.

2.5.7.1.- PRUEBAS DE CAJA NEGRA

Como ya se había mencionado, las pruebas de caja negra se centran en los requisitos funcionales del software.

Partición Equivalente.- La partición equivalente es un método que divide el campo de entrada en un programa en clases de datos.

Una condición de entrada es un valor numérico específico, un rango de valores, un miembro de un conjunto de valores o lógica.

Una clase de equivalencia representa un conjunto de estados válidos y no válidos para una condición de entrada.

La prueba de partición equivalente se basa en evaluar las clases de equivalencia para una condición de entrada.

Identificando Clases de Equivalencia.- Se examina cada condición de entrada y se divide en dos o más grupos.

Se identifican dos tipos de clases:

- ✓ Clases de Equivalencia Válidas
- ✓ Clases de Equivalencia no Válidas

Condición de Entrada	Clases de Equivalencia Validas	Clases de Equivalencia No Validas

FIGURA 4.2.1. Tabla de Clases de Equivalencia

FUENTE: Elaboración Propia

Si la condición de entrada es un:

- Rango, se define una clase de equivalencia valida y dos no validas.
- Valor Especifico, se define una clase de equivalencia valida y dos no validas.
- Miembro de conjunto, se define una clase de equivalencia valida y otra no valida.
- Lógica, se define una clase de equivalencia valida y otra no valida.

La técnica de de Análisis de Valores Limites selecciona casos de prueba que ejerciten los valores límite.

Se derivan tantos casos de prueba a partir de las condiciones de entrada como con las de salida .

2.5.7.2.- PRUEBAS DE CAJA NEGRA (EJEMPLO)

Un usuario puede conectarse al banco por Internet y realizar una serie de operaciones bancarias. Una vez accediendo al banco con las consiguientes medidas de seguridad(clave de acceso y demás), se requiere la siguiente entrada:

- ✓ **Código del Banco.** En blanco o numero de tres dígitos. En este último caso, el primero de ellos tiene que ser mayor que **1**.
- ✓ **Código de Sucursal.** Un numero de cuatro dígitos. El primero de ellos es mayor de 0.

- ✓ **Número de Cuenta.** Numero de cinco dígitos.
- ✓ **Clave Personal.** Valor alfanumérico de cinco posiciones. Este valor se introducirá según la orden que se desee realizar.
- ✓ **Orden.** Puede estar en blanco o ser una de las dos cadenas siguientes:
 - Talonario
 - Movimientos

En el primer caso el usuario recibirá un talonario de cheques, mientras que en el segundo recibirá los movimientos del mes en curso. Si este código esta en blanco, el usuario recibirá los dos documentos.

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Código banco	Lógica (puede estar o no) Si está es Rango	1: En blanco 2: $100 \leq \text{Código banco} \leq 999$	3: Un valor no numérico 4: Código banco < 100 5: Código banco > 999
Código sucursal	Rango	6: $1000 \leq \text{Código sucursal} \leq 9999$	7: Código sucursal < 1000 8: Código sucursal ≥ 9999
Nº Cuenta	Valor	9: Cualquier número de cinco dígitos	10: Número de más de cinco dígitos 11: Número de menos de cinco dígitos
Clave	Valor	12: Cualquier cadena de caracteres alfanuméricos de 5 posiciones	13: Cadena de menos de cinco posiciones 14: Cadena de más de cinco posiciones
Orden	Conjunto, con comportamiento distinto	15: "" 16: "Talonario" 17: "Movimientos"	18: Cadena distinta de blanco y de las válidas

FIGURA 4.2.2. Tabla de Prueba de Caja Negra

FUENTE: Elaboración Propia



CAPÍTULO III

DESARROLLO



3.- CAPITULO III: DESARROLLO

El presente capítulo se describe el ciclo de vida del desarrollo del Sistema de Información de Activos Fijos a través de la Metodología XP, aplicando también las herramientas de UML para tal efecto, identificando RRHH y los roles de cada uno de ellos.

Al principio de este capítulo, se presenta la planificación general del sistema, obteniendo de esta manera un Plan de Entregas, posterior a ello, el sistema se desarrolla en cinco iteraciones, cada una de ellas contempla un plan de iteración, diseño del sistema, programación y pruebas.

3.1.- PLANIFICACIÓN GENERAL DEL PROYECTO

La planificación del proyecto contempla los siguientes aspectos:

- ✓ *Realización de un diagrama de la situación actual.* Contempla la identificación de usuarios, representación del flujo general de trabajo y la descripción del flujo de trabajo.
- ✓ *Recolección de las historias de usuario.* Es una descripción corta que escriben los usuarios tal y como ven las necesidades del sistema.
- ✓ *Detallar un plan de entrega.* Desglosa la creación de un plan de entregas, desglosado a partir del análisis de las historias de usuario.

3.1.1.- Diagnostico Preliminar

Sintetizando el funcionamiento del sistema anterior (sin la incorporación del Sistema de Información Automatizado), se presentan los principales elementos.

3.1.2.- Identificación de Usuarios - Roles (Actores y Tareas)

***Administrador del Sistema.-** Es el encargado de Administrar y manejar el sistema, teniendo un Total acceso al Sistema.

***Tipos de Encargados:**

***Encargado de Activos Fijos Administración Central.-** Representa a la persona encargado de llevar un control de inventarios, administrar los activos y elaborar reportes de acuerdo a requerimiento.

***Encargado de Activos Fijos Sector Salud.-** Representa a la persona encargado de llevar un control de inventarios, administrar los activos y elaborar reportes de acuerdo a requerimiento.

***Encargado de Activos Fijos Sector Educación.-** Representa a la persona encargado de llevar un control de inventarios, administrar los activos y elaborar reportes de acuerdo a requerimiento.

***Encargado de Activos Fijos Áreas Desconcentradas.-** Representa a la persona encargado de llevar un control de inventarios, administrar los activos y elaborar reportes de acuerdo a requerimiento.

3.1.3.- Flujo General de Trabajo

✓ *Diagrama de Procesos de Negocio (BPMN)*

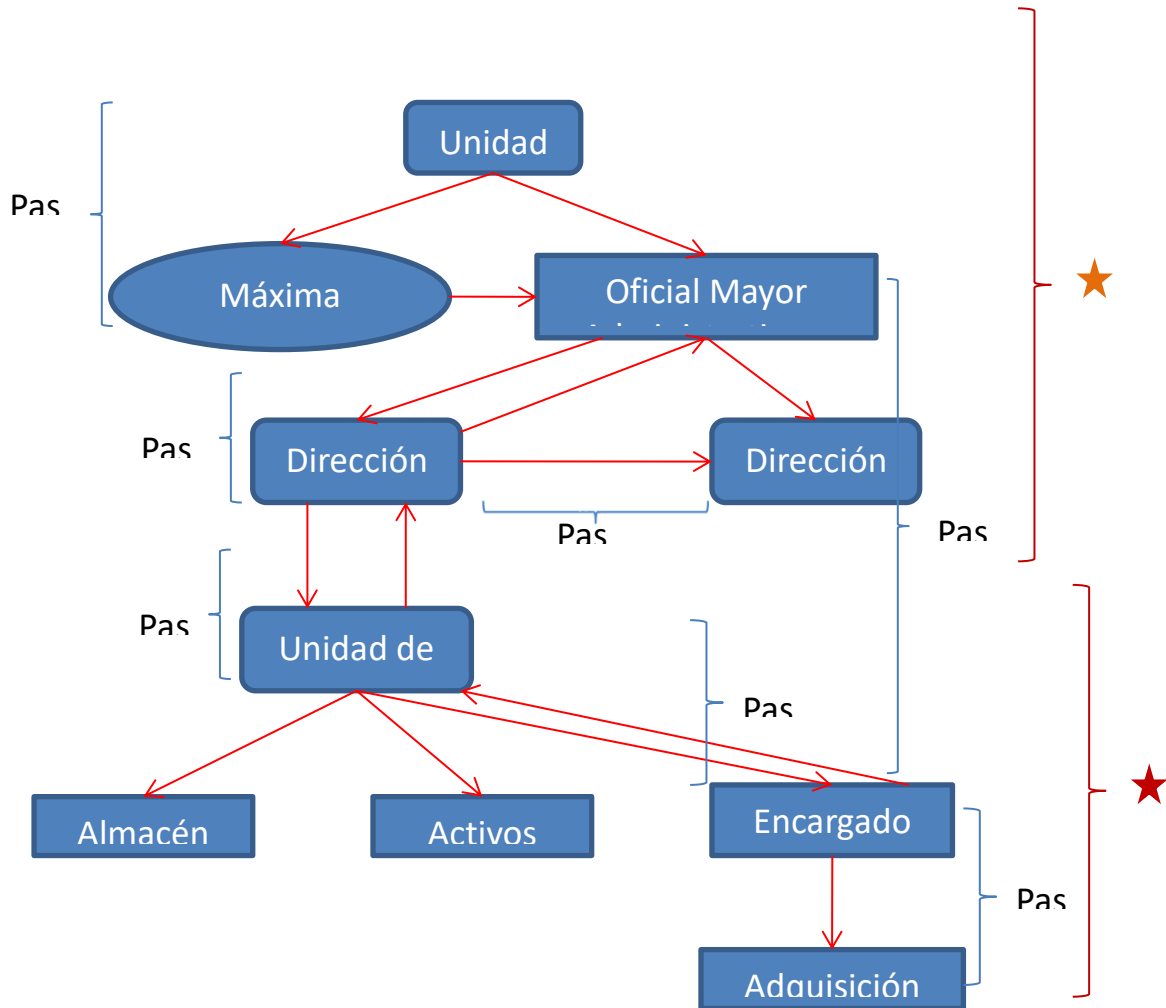


FIGURA 1. Representación del Diagrama de Procesos de Negocio (BPMN) de la adquisición de Activos Fijos del G.A.M.C.

FUENTE: Unidad de Bienes y Servicios G.A.M.C.

3.1.1. Flujo General del Proceso de Trabajo

- ✓ **Paso 1:** La **Unidad Solicitante**, deriva su requerimiento a la **Máxima Autoridad Ejecutiva** (Alcalde) y/o **Oficialía Mayor Administrativa y Financiera**.
- ✓ **Paso 1.1:** Si el requerimiento es entregado a la **Máxima Autoridad Ejecutiva** este lo deriva a la **Oficialía Mayor Administrativa Financiera**.
- ✓ **Paso 2:** La **Oficialía Mayor Administrativa Financiera** deriva la solicitud a la **Dirección Administrativa**.
- ✓ **Paso 3:** La **Dirección Administrativa**, verifica a través de la **Unidad de Presupuesto** si la **Unidad Solicitante** cuenta o no con presupuesto para la requerido.
- ✓ **Paso 4:** La **Unidad de Presupuesto**, certifica si cuenta con recursos y devuelve a la **Dirección Administrativa** y esta deriva a la **Unidad de Bienes y Servicios**, donde verifica si la solicitud es de algún activo o un material, y si la solicitud es con algún de estos bienes, se deriva a una de estas dos secciones que es; **Activo Fijo o Almacén** para su respectivo informe.
- ✓ **Paso 5:** Luego de la verificación de los documentos la **Unidad de Bienes y Servicios** deriva al encargado de **Cotización**, donde realiza la cotización e informa sobre su respectivo precio del bien y/o servicio a adquirir y el nombre del proveedor.
- ✓ **Paso 6:** Luego el informe del **Cotizador** es derivado a la **Oficialía Mayor Administrativa y Financiera** para su conocimiento de la compra o servicio e instruye a la Dirección Financiera efectuar la cancelación.
- ✓ **Paso 7:** Luego de la cancelación se coordina la entrega del bien y/o servicio con el encargado de **Cotización**, la **Unidad Solicitante** y el **Proveedor** donde se realiza un acta de entrega.

Nota: El flujo de trabajo general se divide en dos etapas del Proceso de Negocio (BPMN), y en una de ellas es donde se implementara el sistema.

★ Funcionamiento de la Parte Superior del flujo de trabajo

★ Funcionamiento de la Parte del Sistema de Activos Fijos

3.1.5.- Flujo del Sistema de la Unidad de Activos Fijos

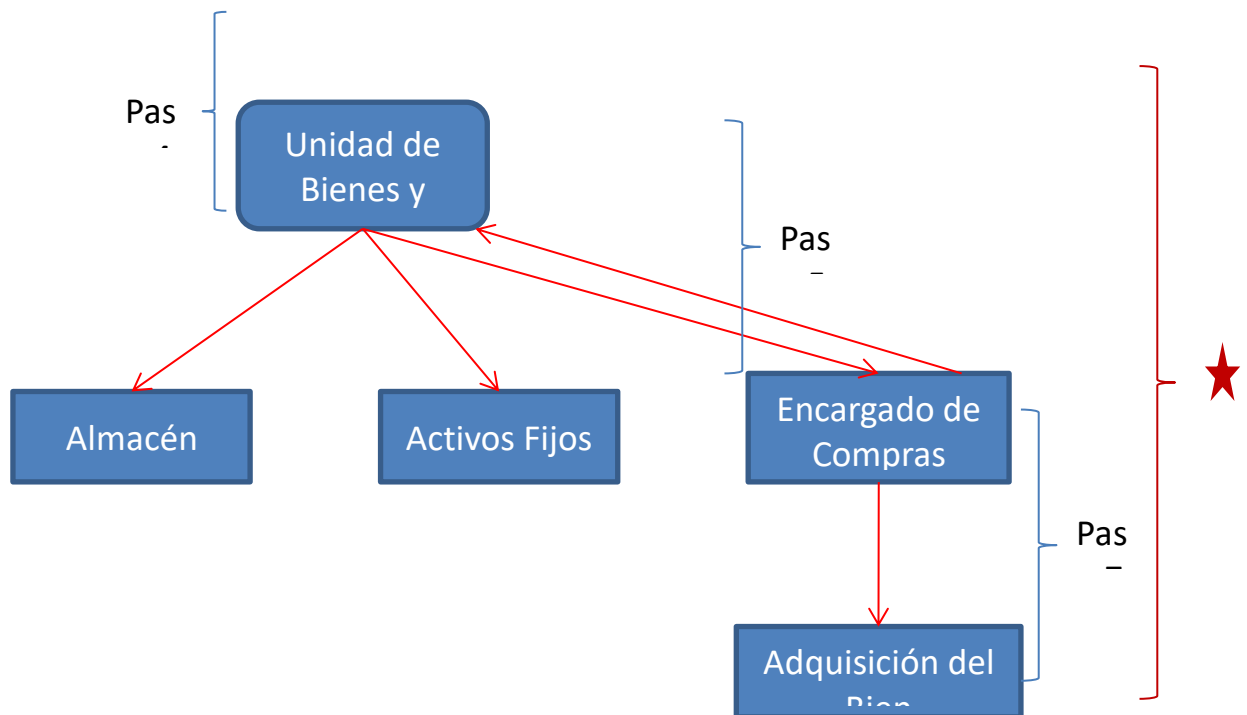


FIGURA 1.2. Organigrama de la Unidad de Activos Fijos del G.A.M.C.

FUENTE: Unidad de Activos Fijos

3.2.- DIAGRAMA DE CASOS DE USO GENERAL DEL SISTEMA

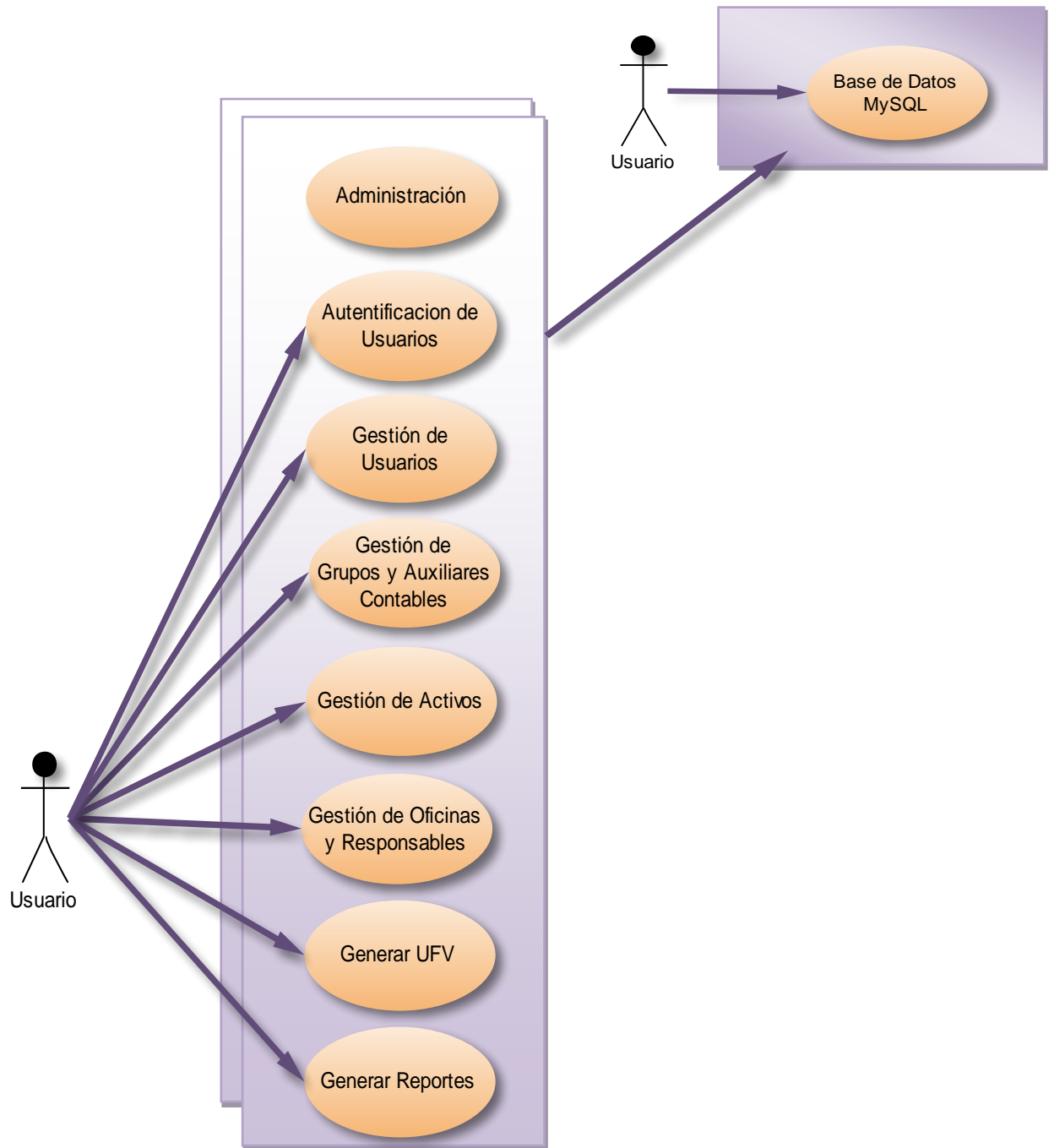


FIGURA 2. Diagrama de Casos de Uso General del Sistema

FUENTE: Elaboración Propia

3.3.- HISTORIAS DE LOS USUARIOS

Estas tarjetas son descritas por las historias de los usuarios identificados en el diagnostico preliminar y refleja de cómo ven ellos el sistema.

Para más detalle se podrá ver en análisis de requerimientos y diagrama de casos de uso de cada una de las Iteraciones.(Ver Anexo: Historias de Usuario)

3.4.- PLAN DE ENTREGA

Luego de un análisis de las Historias de Usuario se logra determinar un plan de entregas, y la división de las Historias de Usuario en Iteraciones.

3.4.1.- PRIMERA ITERACIÓN: Administración de Usuarios

De acuerdo al Plan General de Entregas (Tabla 1), se procede a realizar las actividades comprendidas en la Primera Iteración.

3.4.1.1.- Plan de la Primera Iteración

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Identificación y Autenticación de Usuarios	Ventana de Login del Sistema	Héctor R. Chipana	
	Ingreso de Usuario y Contraseña	Héctor R. Chipana	
	Mensaje de Error: "El usuario y password son incorrectos"	Héctor R. Chipana	
	Ventana del Menú Principal	Héctor R. Chipana	
	Cada Usuario tiene diferentes opciones de acuerdo al tipo de Rol de Usuario	Héctor R. Chipana	

TABLA 1. Plan de la Primera Iteración: Identificación y Autenticación de Usuarios

FUENTE: Elaboración Propia

➤ **Historia de Usuario: Gestión de Usuarios**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Gestión de Usuarios	Formulario de ingreso de datos	Héctor R. Chipana	
	Búsqueda de Usuarios	Héctor R. Chipana	
	Registro de Usuarios	Héctor R. Chipana	
	Modificar Usuarios	Héctor R. Chipana	
	Eliminar Usuarios	Héctor R. Chipana	
	Asignación de Roles a los Usuarios	Héctor R. Chipana	

TABLA 1.2. Plan de la Primera Iteración: Gestión de Usuarios

FUENTE: Elaboración Propia

➤ **Tarjeta Clase - Responsabilidad - Colaboración: Gestión de Usuario**

Clase: Formulario Gestión de Usuarios	
Responsabilidad: Solicitar Datos del Usuario; Guardar Datos del Usuario; Eliminar Datos del Usuario; Modificar Datos del Usuario; Buscar Datos del Usuario; Agregar Nuevo Usuario() Con Nuevo Formulario de Registro	Colaboración: Registrar Usuario Buscar Usuario Nuevo Usuario

TABLA 1.3. Tarjeta CRC: Gestión de Usuario

FUENTE: Elaboración Propia

➤ **Diagrama de Casos de Uso.- Gestión de Usuarios**

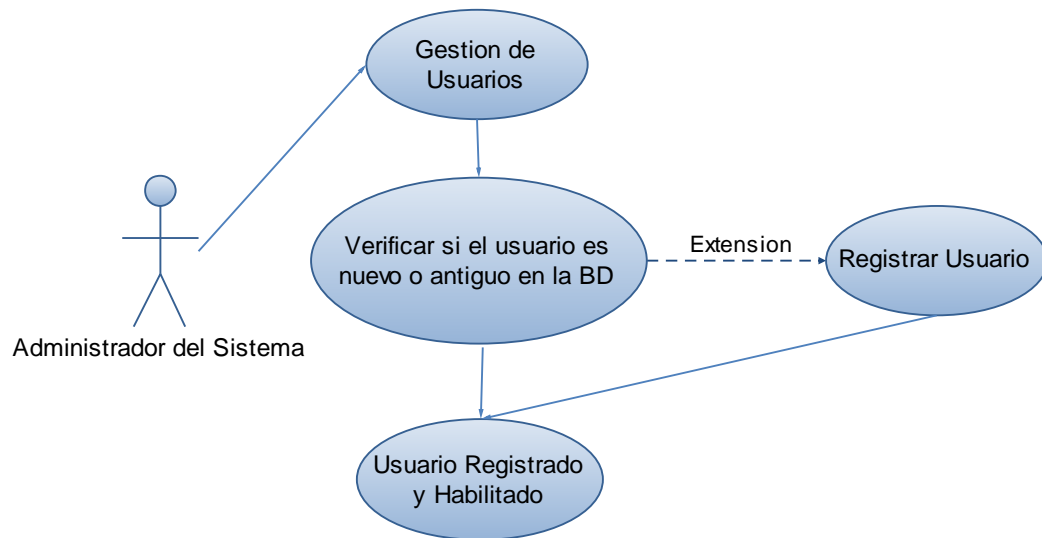


Figura 1: Caso de Uso: Gestión de Usuarios

Fuente: Elaboración Propia

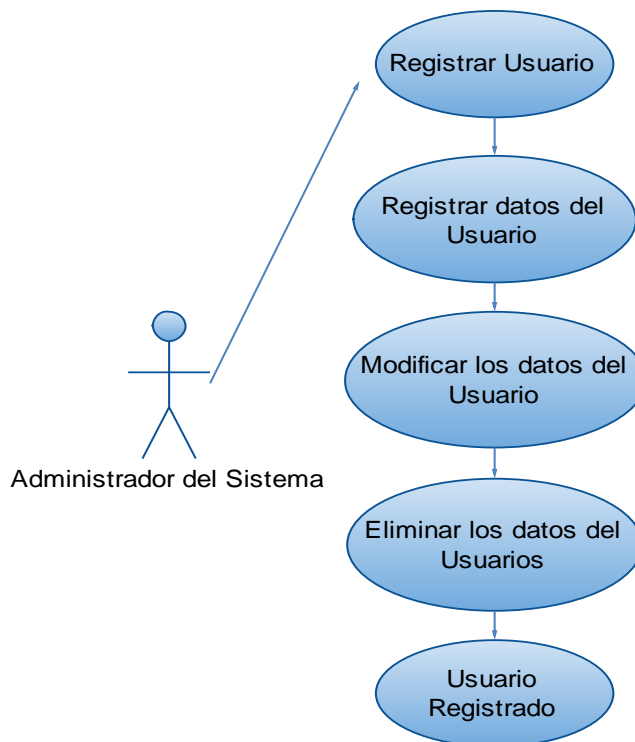


Figura 2: Caso de Uso: Registrar Usuario

Fuente: Elaboración Propia

➤ **Modelo Conceptual de la Base de Datos: Primera Iteración**

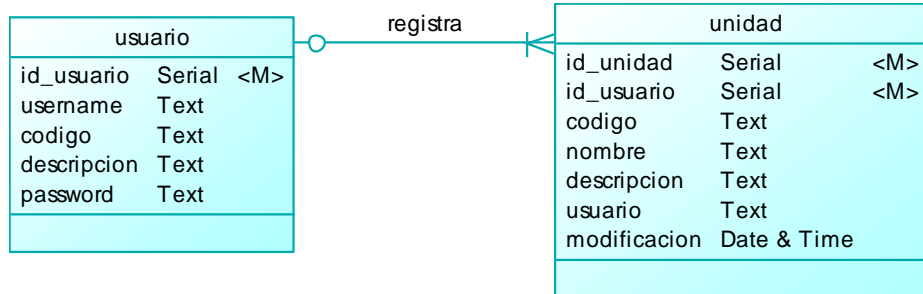


Figura 3. Modelo Conceptual de la Base de Datos de la Primera Iteración

Fuente: Elaboración Propia

➤ **Funcionalidad Mínima del Sistema: Formulario de Administración de Usuarios**

En este formulario se gestiona los Usuarios Correspondientes a la Administración del Sistema, según a requerimiento.

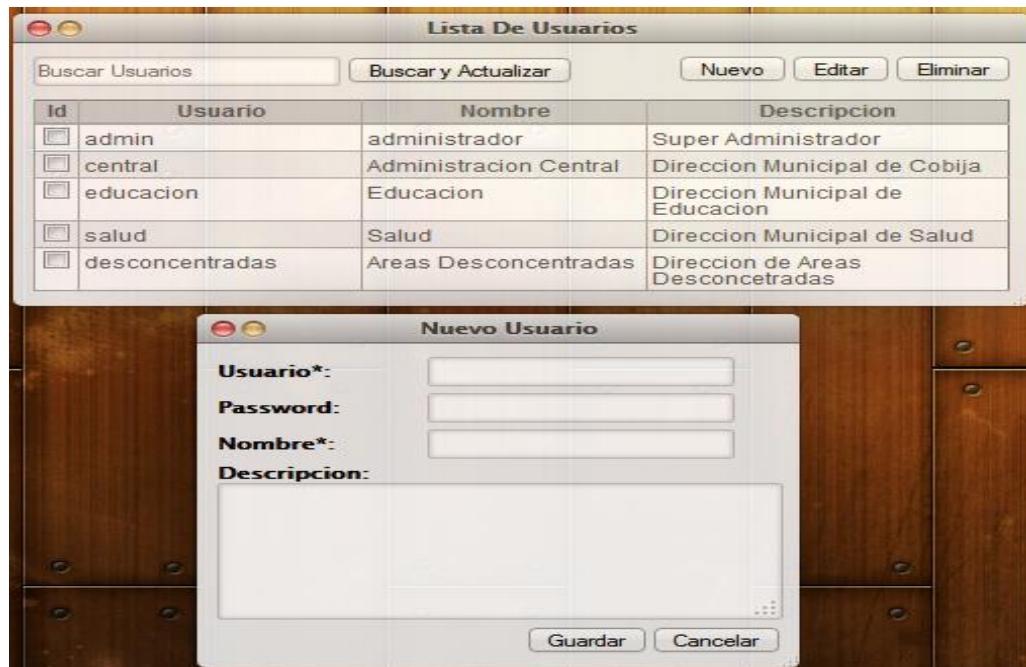


Figura 1: Formulario Administración de Usuarios

Fuente: Elaboración Propia

➤ **Pruebas del Subsistema de la Primera Iteración: Caso de Prueba N° 2:**
Gestión de Usuarios

- ✓ **Propósito:** Verificar el funcionamiento del formulario de Administración de Usuarios
- ✓ **Pre Requisitos:** El Usuario esta registrado, Rol de Usuario.
- ✓ **Datos de Prueba**
 - **Usuario**(central, ^*, ..%)
 - **Password**(*****, ^*, ?i)
 - **Nombre**(Administración Central, =&, ·2)
 - **Descripción**(Responsable Nivel 1, ·\$%, !^a)

Nota: Los Valores Con Negrilla son Validos.

PASOS: Preparatorio:

Menú Administración de Usuarios >> Realizar la búsqueda de los Usuarios>> Registrar los datos del Usuario >> Cargar los datos de los Usuarios

Resultados Esperados: Usuario Habilitado y Registrado con éxito.

➤ **Historia de Usuario: Gestión de Unidades Administrativas**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Gestión de Unidades Administrativas	Formulario de ingreso de datos	Héctor R. Chipana	
	Búsqueda de Unidades	Héctor R. Chipana	
	Registro de Unidades	Héctor R. Chipana	
	Modificar Unidades	Héctor R. Chipana	
	Eliminar Unidades	Héctor R. Chipana	

TABLA 1.4. Plan de la Primera Iteración: Gestión de Unidades Administrativas

FUENTE: Elaboración Propia

➤ **Tarjeta Clase - Responsabilidad - Colaboración: Gestión de Unidades Administrativas**

Clase: Formulario Gestión de Unidades Administrativas	
Responsabilidad: Solicitar Datos de la Unidad; Guardar Datos de la Unidad; Eliminar Datos de la Unidad; Modificar Datos de la Unidad; Buscar Datos de la Unidad; Agregar Nueva Unidad() Con Nuevo Formulario de Registro	Colaboración: Registrar Unidad Buscar Unidad Nueva Unidad

TABLA 1.5. Tarjeta CRC: Gestión de Unidades Administrativas

FUENTE: Elaboración Propia

➤ **Diagrama de Casos de Uso.- Gestión de Unidades Administrativas**

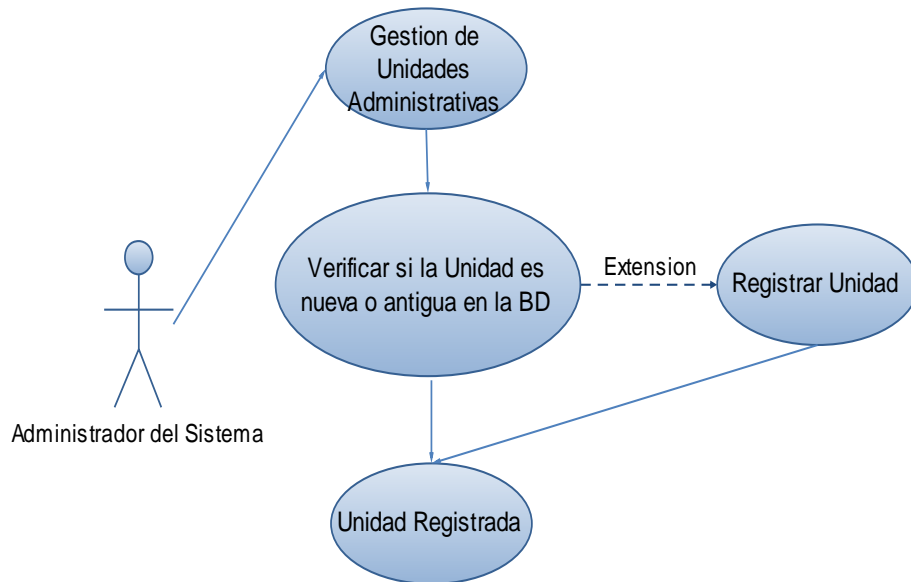


Figura 1: Caso de Uso: Gestión de Unidades Administrativas

Fuente: Elaboración Propia

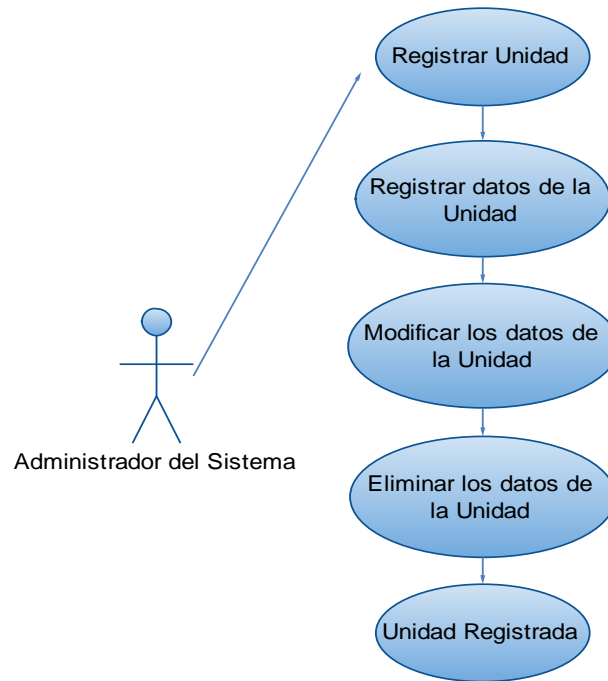


Figura 2: Caso de Uso: Registrar Unidad

Fuente: Elaboración Propia

➤ **Funcionalidad Mínima del Sistema: Formulario de Unidades Administrativas**

En este formulario se gestiona las Unidades Administrativas Correspondientes según a requerimiento.

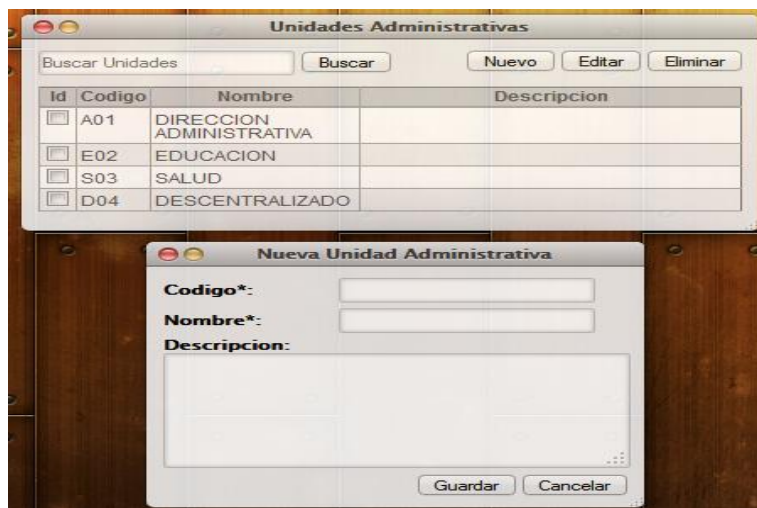


Figura 1: Formulario Unidades Administrativas

Fuente: Elaboración Propia

➤ **Pruebas del Subsistema de la Primera Iteración: Caso de Prueba N° 2:**

Gestión de Unidades Administrativas

- ✓ **Propósito:** Verificar el funcionamiento del formulario de Gestión de Unidades Administrativas.
- ✓ **Pre Requisitos:** La Unidad Administrativa esta registrado, Rol de Usuario.
- ✓ **Datos de Prueba**
 - **Código**(Z01, ^*, ..%)
 - **Nombre**(Sector Z, ^*, ?i)
 - **Descripción**(Nivel Z, =&, ·2)

Nota: Los Valores Con Negrilla son Validos.

d) PASOS: Preparatorio:

Menú Unidad Administrativa >> Realizar la búsqueda de la Unidad Administrativa >>

Registrar los datos de la Unidad Administrativa >> Cargar los datos de la Unidad Administrativa

Resultados Esperados: Unidad Administrativa Habilitada y Registrada con éxito.

3.4.2.- SEGUNDA ITERACIÓN: Administración de Activos

De acuerdo al Plan General de Entregas (Tabla 2), se procede a realizar las actividades comprendidas en la Segunda Iteración.

3.4.2.1.- Plan de la Segunda Iteración

➤ **Historia de Usuario: Gestión de Oficinas y Responsables**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Gestión de Oficinas y Responsables	Formulario de ingreso de datos	Héctor R. Chipana	
	Búsqueda de Oficinas y Responsables	Héctor R. Chipana	
	Registro de Oficinas y responsables	Héctor R. Chipana	
	Modificar Oficinas y Responsables	Héctor R. Chipana	
	Eliminar Oficinas y Responsables	Héctor R. Chipana	

TABLA 2.1. Plan de la Segunda Iteración: Gestión de Oficinas y Responsables

FUENTE: Elaboración Propia

➤ **Tarjeta Clase - Responsabilidad - Colaboración: Gestión de Oficinas y Responsables**

Clase: Formulario Gestión de Oficinas y Responsables	
Responsabilidad: Solicitar Datos de la Oficina; Solicitar Datos de la Responsable; Guardar Datos de la Oficina; Guardar Datos del Responsable; Eliminar Datos de la Oficina; Eliminar Datos del Responsable;	Colaboración: Registrar Oficina Registrar Responsable Buscar Oficina Buscar Responsable Nueva Oficina Nuevo Responsable

Modificar Datos de la Oficina; Modificar Datos del Responsable; Buscar Datos de la Oficina; Buscar Datos del Responsable; Agregar Nueva Oficina() Con Nuevo Formulario de Registro Agregar Nuevo Responsable() Con Nuevo Formulario de Registro	
--	--

TABLA 1.2. Tarjeta CRC: Gestión de Oficinas y Responsables

FUENTE: Elaboración Propia

➤ **Diagrama de Casos de Uso.- Gestión de Oficinas y Responsables**

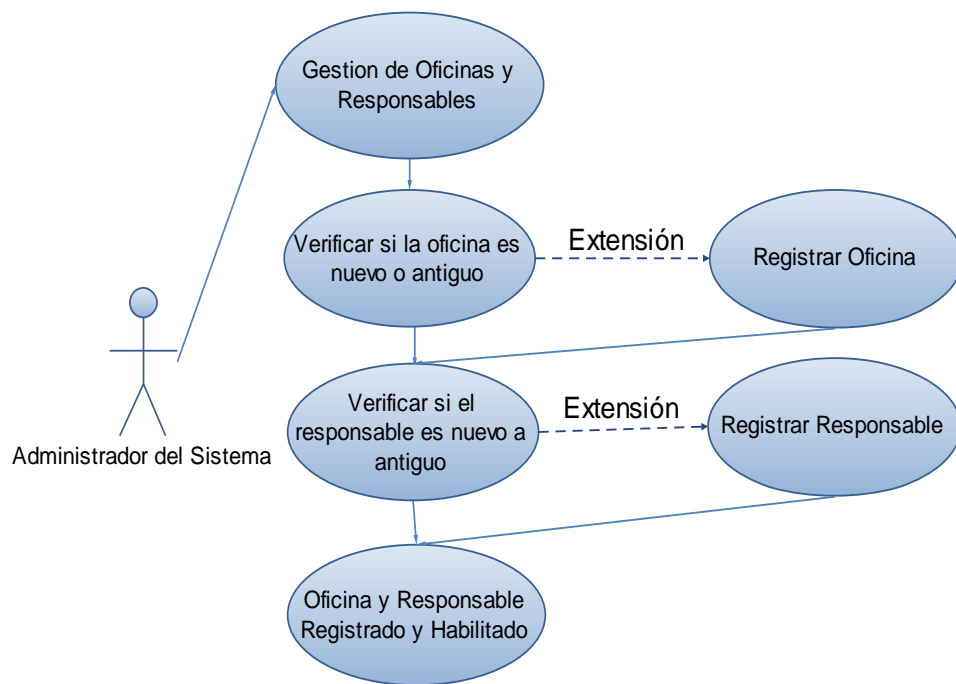


Figura 1: Caso de Uso: Gestión de Oficinas y Responsables

Fuente: Elaboración Propia

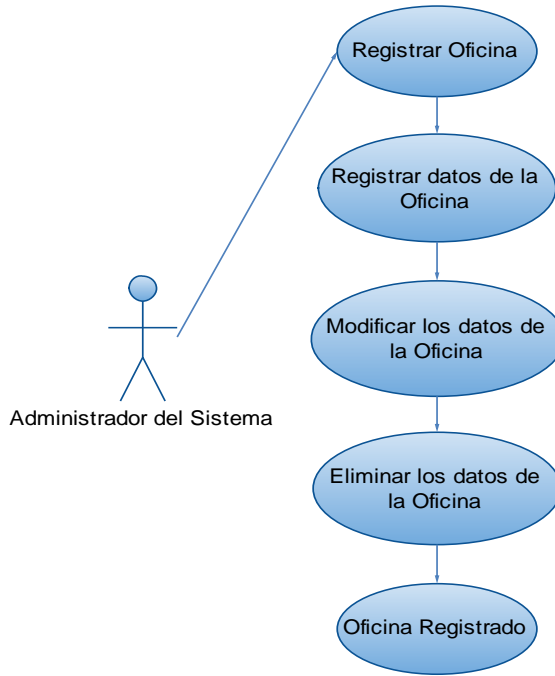


Figura 2: Caso de Uso: Registrar Oficina

Fuente: Elaboración Propia

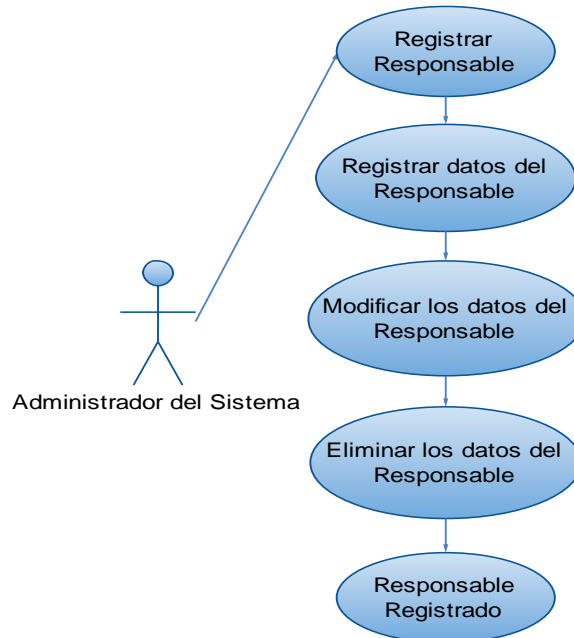


Figura 3: Caso de Uso: Registrar Responsable

Fuente: Elaboración Propia

➤ **Modelo Conceptual de la Base de Datos: Segunda Iteración**

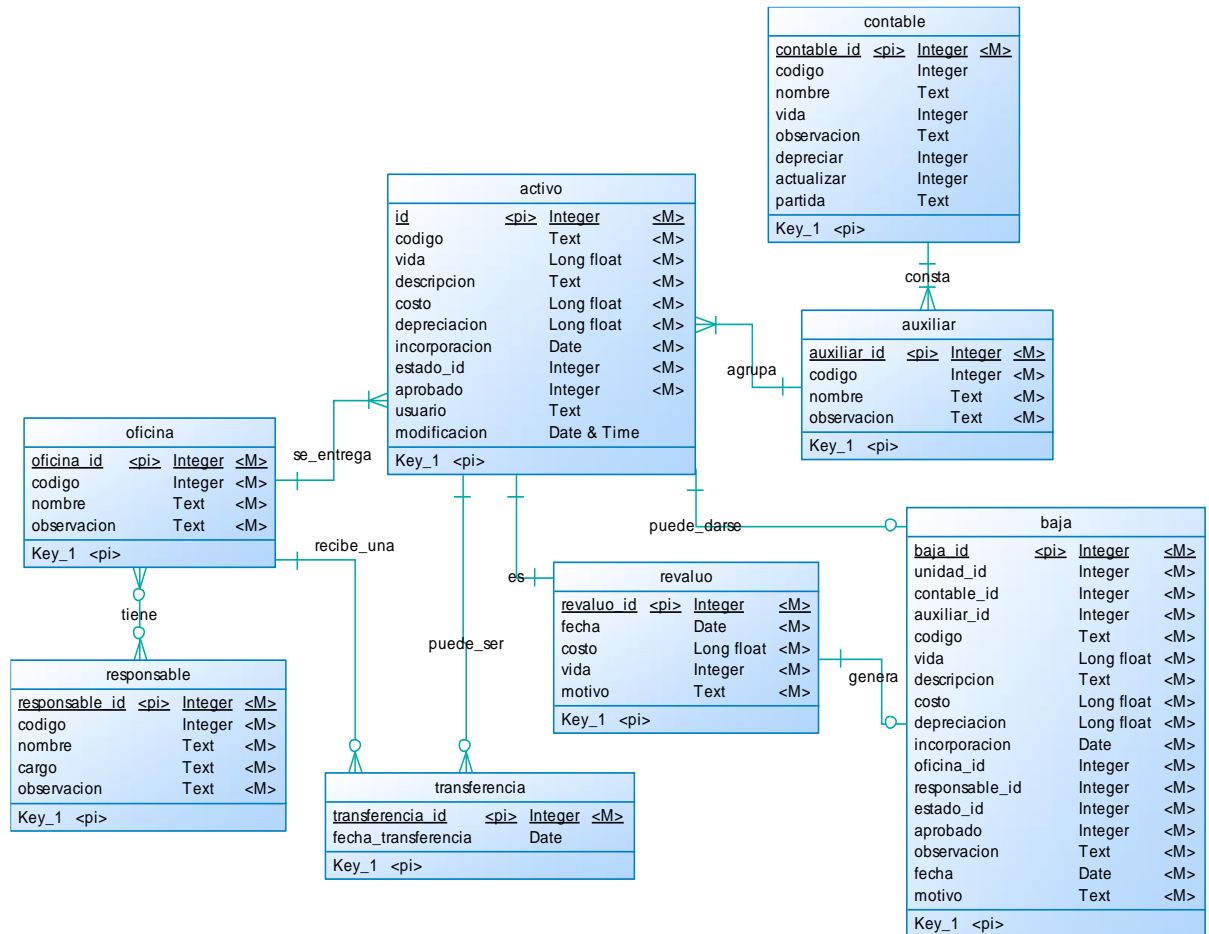


Figura 4. Modelo Conceptual de la Base de Datos de la Segunda Iteración

Fuente: Elaboración Propia

➤ **Funcionalidad Mínima del Sistema: Formulario de Oficinas y Responsables**

En estos dos formularios se gestiona las Oficinas y Responsables Correspondientes según a requerimiento.

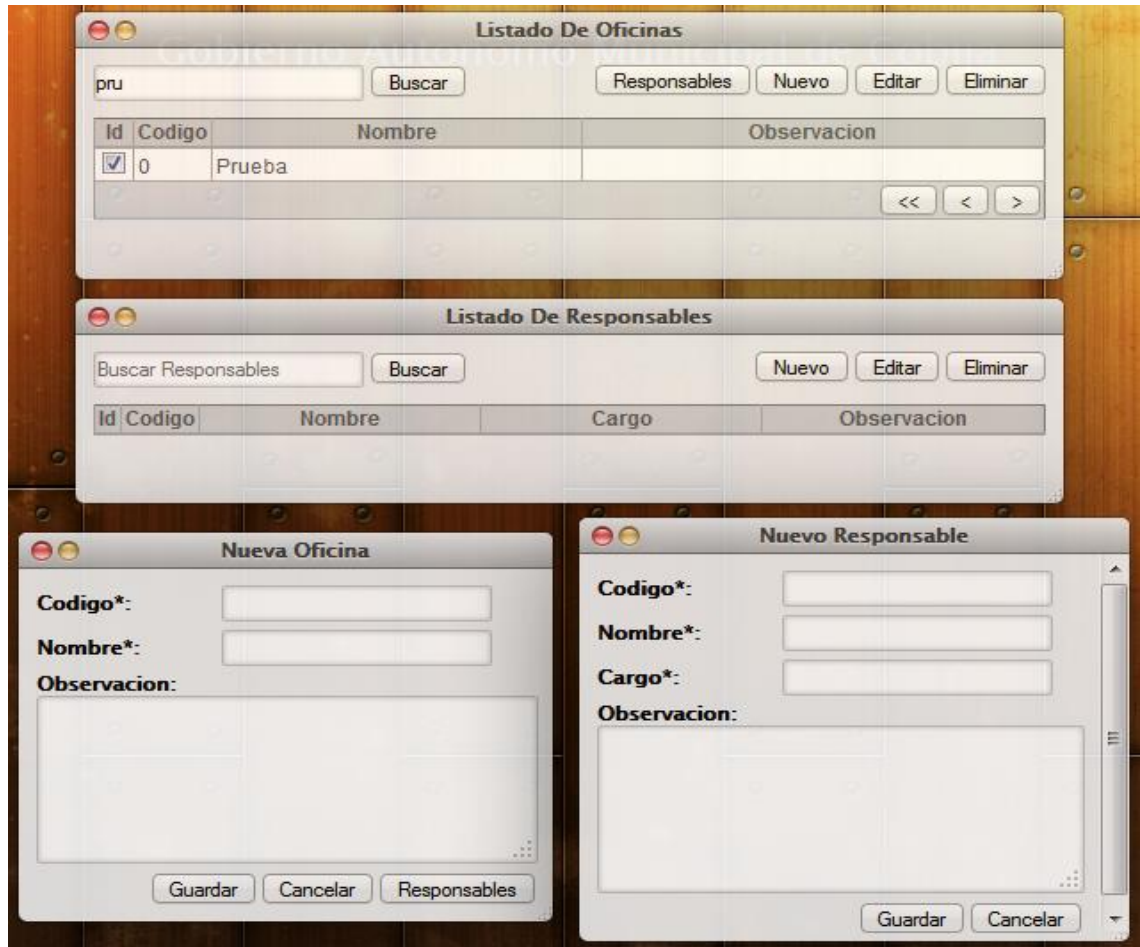


Figura 5: Formulario Oficinas y Responsables

Fuente: Elaboración Propia

- **Pruebas del Subsistema de la Segunda Iteración: Caso de Prueba N° 1: Gestión de Oficinas y Responsables**
- ✓ **Propósito:** Verificar el funcionamiento del formulario de Gestión de Oficinas y Responsables.
- ✓ **Pre Requisitos:** La Oficina y Responsable esta registrado, Rol de Usuario.
- ✓ **Datos de Prueba**
 - **Código Oficina**(45, ^*, ..%)
 - **Código Responsable**(03, ^*, ..%)
 - **Nombre Oficina**(Farmacia GAMC, ^*, ?i)

- **Nombre Responsable**(Sr. Julio Gonzales, ^*, ?i)
- **Observación Oficina**(Nivel 1, =&, ·2)
- **Observación Responsable**(Nivel 2, =&, ·2)

Nota: Los Valores Con Negrilla son Validos.

PASOS: Preparatorio:

Menú Oficinas y Responsables >> Realizar la búsqueda de la Oficina >> Realizar la búsqueda del Responsable >> Registrar los datos de la Oficina >> Registrar los datos del Responsable >> Cargar los datos de la Oficina >> Cargar los datos del Responsable

Resultados Esperados: Oficina y Responsable Habilitado y Registrado con éxito.

➤ **Historia de Usuario: Gestión de Grupos y Auxiliares Contables**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Gestión de Grupos y Auxiliares Contables	Formulario de ingreso de datos	Héctor R. Chipana	
	Búsqueda de Grupos y Auxiliares Contables	Héctor R. Chipana	
	Registro de Grupos y Auxiliares Contables	Héctor R. Chipana	
	Modificar Grupos y Auxiliares Contables	Héctor R. Chipana	
	Eliminar Grupos y Auxiliares Contables	Héctor R. Chipana	

TABLA 2.2. Plan de la Segunda Iteración: Gestión de Grupos y Auxiliares Contables

FUENTE: Elaboración Propia

➤ **Tarjeta Clase - Responsabilidad - Colaboración: Gestión de Grupos y Auxiliares Contables**

Clase: Formulario Gestión de Grupos y Auxiliares Contables	
<p>Responsabilidad: Solicitar Datos del Grupo; Solicitar Datos del Auxiliar; Guardar Datos del Grupo; Guardar Datos del Auxiliar; Eliminar Datos del Grupo; Eliminar Datos del Auxiliar; Modificar Datos del Grupo; Modificar Datos del Auxiliar; Buscar Datos del Grupo; Buscar Datos del Auxiliar; Agregar Nuevo Grupo() Con Nuevo Formulario de Registro Agregar Nuevo Auxiliar() Con Nuevo Formulario de Registro</p>	<p>Colaboración: Registrar Grupo Registrar Auxiliar Buscar Grupo Buscar Auxiliar Nueva Grupo Nuevo Auxiliar</p>

TABLA 1.3. Tarjeta CRC: Gestión de Grupos y Auxiliares Contables

FUENTE: Elaboración Propia

➤ **Diagrama de Casos de Uso.- Gestión de Grupos y Auxiliares Contables**

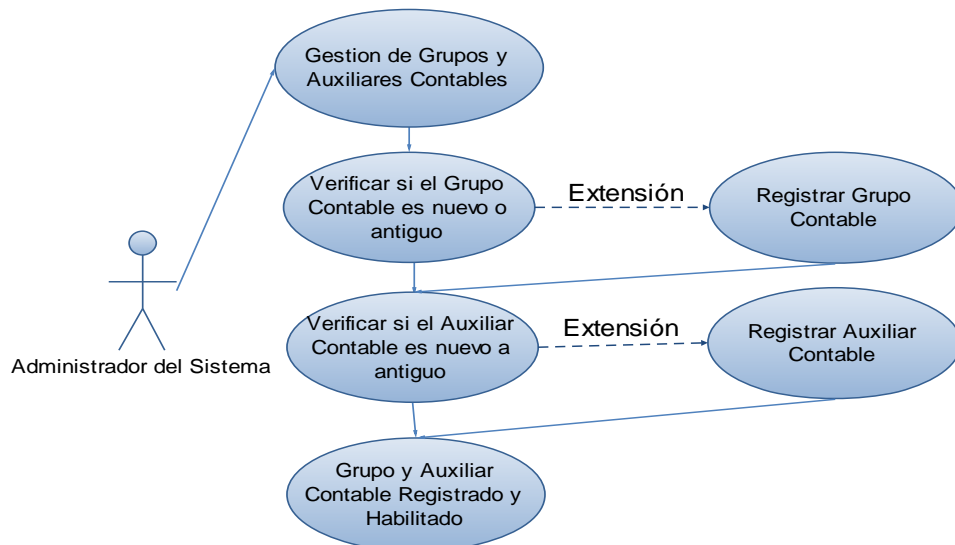


Figura 1: Caso de Uso: Gestión de Grupos y Auxiliares Contables

Fuente: Elaboración Propia

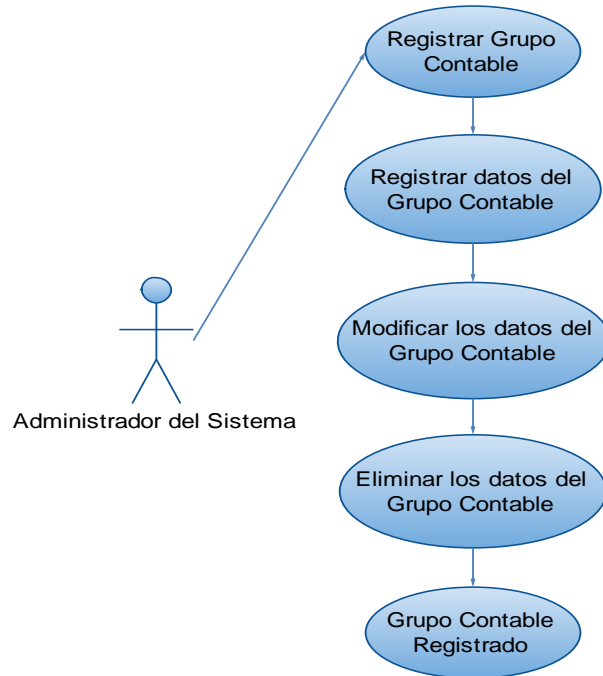


Figura 2: Caso de Uso: Registrar Grupo Contable

Fuente: Elaboración Propia

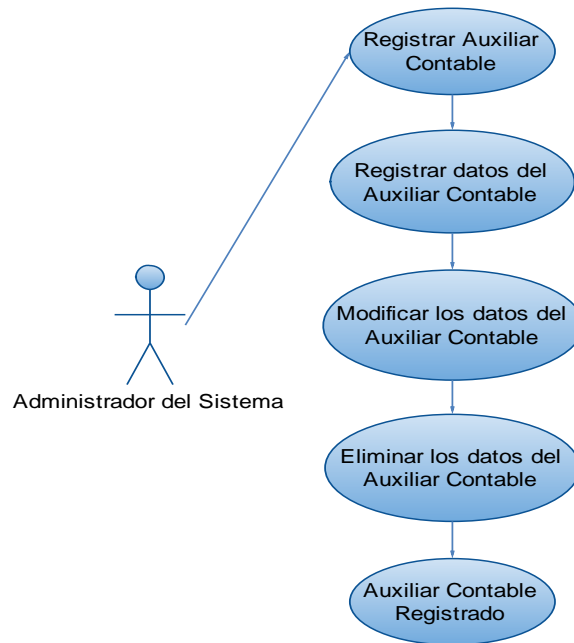


Figura 3: Caso de Uso: Registrar Auxiliar Contable

Fuente: Elaboración Propia

➤ **Funcionalidad Mínima del Sistema: Formulario de Grupos y Auxiliares Contables**

En estos dos formularios se gestiona los Grupos y Auxiliares Contables Correspondientes según a requerimiento.

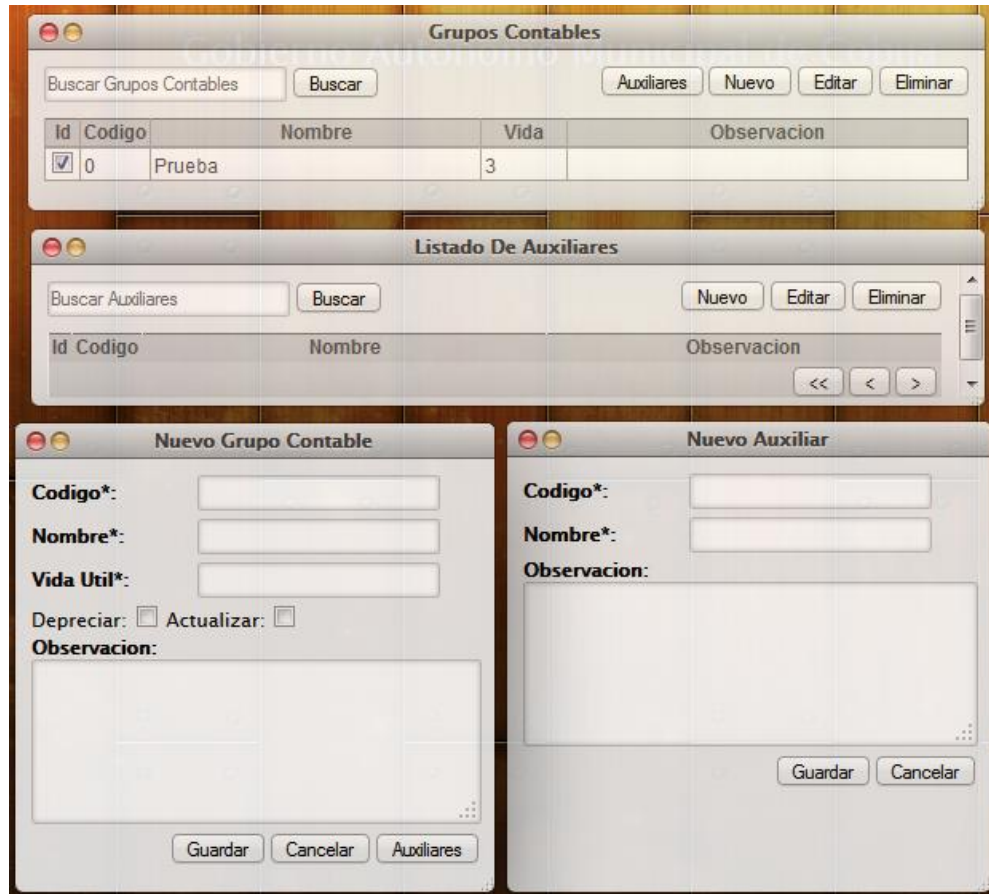


Figura 1: Formulario Grupos y Auxiliares Contables

Fuente: Elaboración Propia

➤ **Pruebas del Subsistema de la Segunda Iteración: Caso de Prueba N° 2:**

Gestión de Grupos y Auxiliares Contables

- ✓ **Propósito:** Verificar el funcionamiento del formulario de Gestión de Grupos y Auxiliares Contables.
- ✓ **Pre Requisitos:** El Grupo y Auxiliar Contable esta registrado, Rol de Usuario.
- ✓ **Datos de Prueba**
 - **Código Grupo**(45, ^*, ..%)
 - **Código Auxiliar**(03, ^*, ..%)

- **Nombre Grupo**(Vehículos, ^*, ?i)
- **Nombre Auxiliar**(Camión, ^*, ?i)
- **Vida Útil Grupo**(05, \$%, !")
- **Seleccionar Campos**(Depreciar, Actualizar)
- **Observación Grupo**(Grupo 1, =&, ·2)
- **Observación Auxiliar**(Auxiliar 2, =&, ·2)

Nota: Los Valores Con Negrilla son Validos.

d) PASOS: Preparatorio:

Menú Grupos y Auxiliares Contables >> Realizar la búsqueda del Grupo >> Realizar la búsqueda del Auxiliar >> Registrar los datos del Grupo >> Registrar los datos del Auxiliar >> Cargar los datos del Grupo >> Cargar los datos del Auxiliar

Resultados Esperados: Grupo y Auxiliar Contable Habilitado y Registrado con éxito.

➤ **Historia de Usuario: Gestión de Activos**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Gestión de Activos	Formulario de ingreso de datos	Héctor R. Chipana	
	Búsqueda de Activos	Héctor R. Chipana	
	Registro de Activos	Héctor R. Chipana	
	Modificar Activos	Héctor R. Chipana	
	Eliminar Activos	Héctor R. Chipana	
	Aprobar Activos	Héctor R. Chipana	
	Duplicar Activos	Héctor R. Chipana	
	Transferencia de Activos	Héctor R. Chipana	
	Revaluó de Activos	Héctor R. Chipana	
	Baja de Activos	Héctor R. Chipana	

TABLA 2.3. Plan de la Segunda Iteración: Gestión de Activos

FUENTE: Elaboración Propia

➤ **Tarjeta Clase - Responsabilidad - Colaboración: Gestión de Activos**

Clase: Formulario Gestión de Activos	
<p>Responsabilidad: Solicitar Datos del Activo; Solicitar Datos de la Oficina y Responsable; Solicitar Datos del Grupo y Auxiliar Contable; Guardar Datos del Activo; Eliminar Datos del Activo; Modificar Datos del Activo; Buscar Datos del Activo; Agregar Nuevo Activo() Con Nuevo Formulario de Registro</p>	<p>Colaboración: Registrar Activo Buscar Activo Aprobar Activo Duplicar Activo Transferir Activo Revaluó de Activo Baja de Activo Nueva Activo</p>

TABLA 1.4. Tarjeta CRC: Gestión de Activos

FUENTE: Elaboración Propia

➤ **Diagrama de Casos de Uso.- Gestión de Activos**

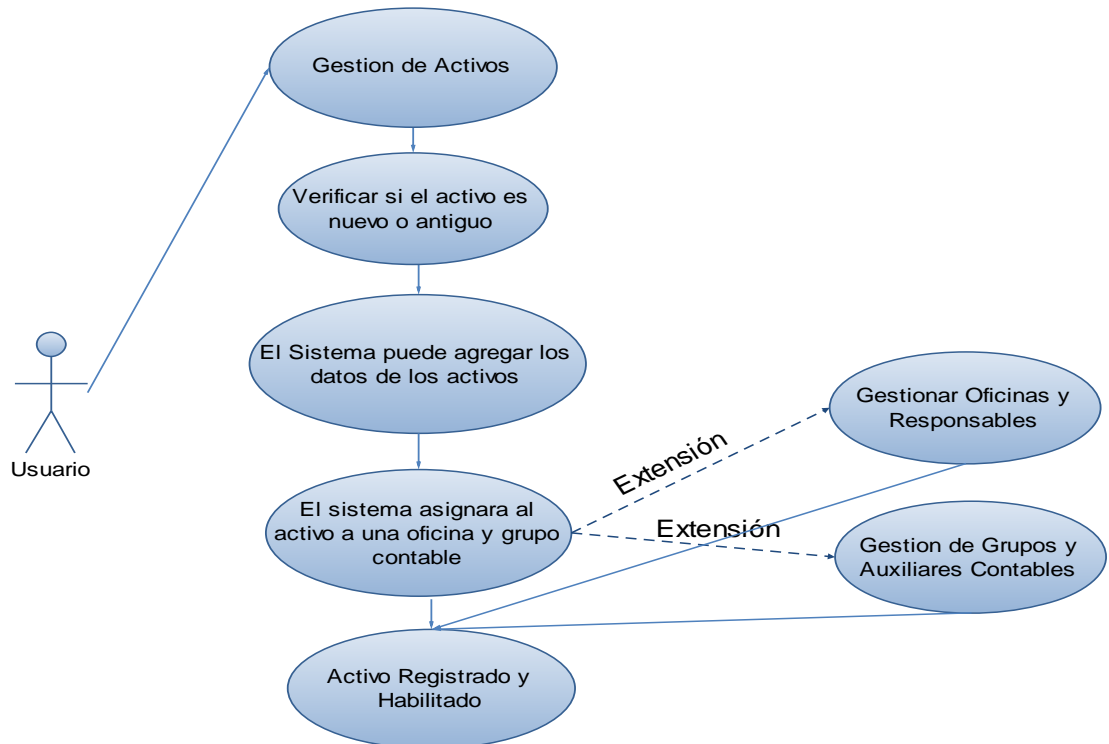


Figura 1: Caso de Uso: Gestión de Activos

Fuente: Elaboración Propia

➤ **Funcionalidad Mínima del Sistema: Formulario de Activos Fijos**

En estos dos formularios se gestiona los Activos Correspondientes según a requerimiento.

Busqueda porCodigo Elija una Oficina Elija un Responsable Elija un Grupo Cont. Elija un Auxiliar Buscar y Actualizar

No Ha Realizado Ninguna Busqueda...

Aprobar Duplicar Transferencia Reevaluo Baja Nuevo Editar Eliminar

Codigo: Incorporacion: 2012-12-03 Indice Ufv: 1

Descripcion Observacion

Oficina: ADMINISTRACION H.R.G. Costo Inicial: 0

Responsable: LIC. MARCELO PIEROLA Dep. Acu. Ini.: 0

Grupo: ACTIVOS INTANGIBLES Factor de Act.: 1.0

Auxiliar: SISTEMA BIOMETRICO Valor: 0

Estado: nuevo Aprobado: No

Vida Util: 5 % Depreciacion: 20 Dias Pasados: 0 Dep. Gestion: 0 Dep. Acumulada: 0 Valor Neto: 0

Calculados a: 2012-12-03 Indice Ufv: 1 Calcular

Cancelar Guardar

Duplicar Activo A011104EA001

NuevoCodigo: Cancelar Guardar

Transferencia Activo A011104EA001

Unidad: DIRECCION ADMINISTRATIVA

Oficina: DESPACHO H. ALCALDE

Responsable: DR. LUIS ADOLFO FLORES

Motivo:

Cancelar Guardar

Revaluación Activo A011104EA001

Fecha: 2007-04-12

Costo: 640

Vida: 10

Motivo:

Cancelar Guardar

Baja Activo A011104EA001

Fecha: 2012-12-03

Motivo:

Cancelar Guardar

Figura 1: Formulario Activos Fijos

Fuente: Elaboración Propia

➤ **Pruebas del Subsistema de la Segunda Iteración: Caso de Prueba N° 1:**

Gestión de Activos

✓ **Propósito:** Verificar el funcionamiento del formulario de Gestión de Activos.

✓ **Pre Requisitos:** El Activo esta registrado, Rol de Usuario.

✓ **Datos de Prueba**

- **Código Activo(A012304BM045, ^*, ..%)**
- **Fecha Incorporación(2012-12-03, ^*, ..%)**
- **Índice UFV(1, ^*, ?;)**
- **Descripción del Activo(Impresora HP, ^*, ?;)**
- **Observación del Activo(ninguna, =&, ·2)**
- **Seleccionar Oficina y Responsable**
- **Seleccionar Grupo y Auxiliar Contable**
- **Seleccionar Estado**
- **Costo Inicial(1455, !", %&)**
- **Depreciación Acumulada Inicial(0)**
- **Factor de Actualización(1.0)**
- **Valor (0)**
- **Aprobado("SI" o "No")**
- **Vida Útil(4)**
- **% Depreciación(0)**
- **Días Pasados(0)**
- **Depreciación Gestión(0)**
- **Depreciación Acumulada(0)**
- **Valor Neto(0)**
- **Calculados a (2012-12-03)**
- **Índice UFV(1)**

Nota: Los Valores Con Negrilla son Validos.

PASOS: Preparatorio:

Menú Activos Fijos >> Realizar la búsqueda de Activos >> Registrar los datos del Activo >> Cargar los datos de la Oficina y Responsable >> Cargar los datos del Grupo y Auxiliar Contable >> Cargar las Fechas >> Guardar Activo

Resultados Esperados: Activo Habilitado y Registrado con éxito.

3.4.3.- TERCERA ITERACIÓN: Control de Activos

De acuerdo al Plan General de Entregas (Tabla 3), se procede a realizar las actividades comprendidas en la Tercera Iteración.

3.4.3.1.- Plan de la Tercera Iteración

➤ **Historia de Usuario: Control de Activos**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Control de Activos (Verificación)	Identificar los activos con el lector de código de barras	Héctor R. Chipana	
	Mediante los códigos de barras etiquetados en los activos	Héctor R. Chipana	
	Generar Estado de los activos en el sistema	Héctor R. Chipana	

TABLA 3. Plan de la Tercera Iteración: Control de Activos (Verificación)

FUENTE: Elaboración Propia

➤ **Historia de Usuario: Generar UFV**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Generar UFV	Actualizar la UFV	Héctor R. Chipana	
	Búsqueda de UFV	Héctor R. Chipana	
	Ver UFV	Héctor R. Chipana	

TABLA 3.1. Plan de la Tercera Iteración: Generar UFV

FUENTE: Elaboración Propia

➤ **Modelo Conceptual de la Base de Datos: Generar UFV**

ufv		
id	Serial	<M>
fecha	Date	
valor	Decimal	

Figura 1. Modelo Conceptual de la Base de Datos de la Tercera Iteración

Fuente: Elaboración Propia

Caso de Uso: Generar UFV: Ver ANEXO.- Casos de Uso

3.4.4.- CUARTA ITERACIÓN: Reportes

De acuerdo al Plan General de Entregas (Tabla 4), se procede a realizar las actividades comprendidas en la Cuarta Iteración.

3.4.4.1.- Plan de la Cuarta Iteración

➤ **Historia de Usuario: Generar Reportes**

HIST. DE USUARIO	TAREAS	PROGRAM.	T. EST
Generar Reportes	Formulario de ingreso de datos	Héctor R. Chipana	
	Cargar Oficina y Responsable	Héctor R. Chipana	
	Cargar Grupo y Auxiliar Contable	Héctor R. Chipana	
	Seleccionar Fecha Inicio y Fecha Final	Héctor R. Chipana	
	Seleccionar Tipo de Reporte	Héctor R. Chipana	
	Generar Reporte	Héctor R. Chipana	

TABLA 4. Plan de la Cuarta Iteración: Generar Reportes

FUENTE: Elaboración Propia

➤ **Tarjeta Clase - Responsabilidad - Colaboración: Generar Reportes**

Clase: Formulario Generar Reportes	
Responsabilidad: Seleccionar Unidad; Seleccionar Oficina; Seleccionar Responsable; Seleccionar Grupo; Seleccionar Auxiliar; Seleccionar De Fecha; Seleccionar A Fecha; Seleccionar Tipo de Reporte;	Colaboración: Generar Reporte

TABLA 4. Tarjeta CRC: Generar Reporte

FUENTE: Elaboración Propia

➤ **Diagrama de Casos de Uso.- Generar Reporte**

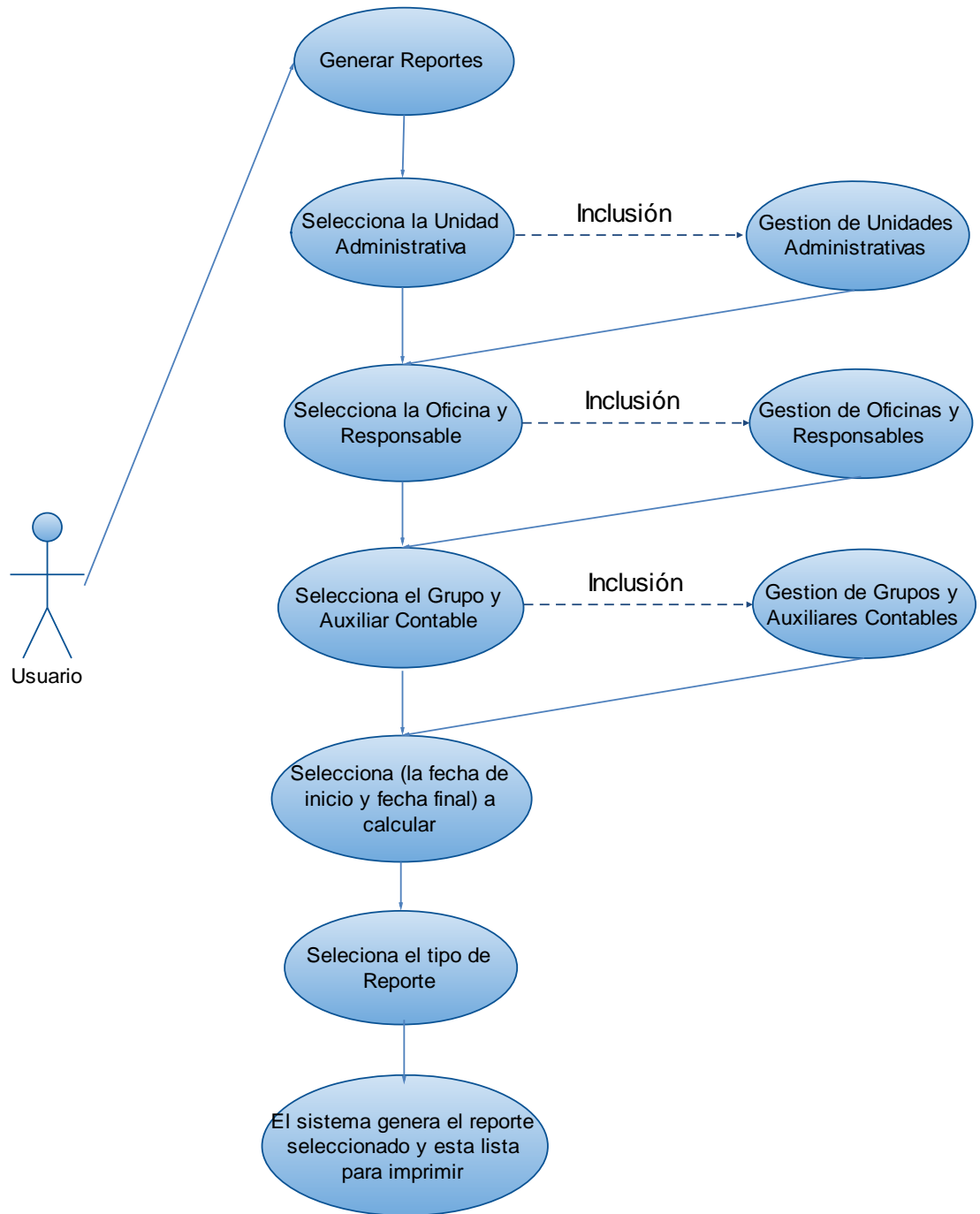


Figura 1: Caso de Uso: Generar Reporte

Fuente: Elaboración Propia

➤ **Funcionalidad Mínima del Sistema: Formulario Generar Reportes**

En este formulario se Genera los Reportes Correspondientes según a requerimiento.

The screenshot shows a web application window titled "Reporte De Activos Fijos". It contains several input fields and a list of report options. The fields are: "Unidad:" (dropdown menu with "Todos" selected), "Oficina:" (dropdown menu with "Todos" selected), "Responsable:" (dropdown menu with "Todos" selected), "Grupo:" (dropdown menu with "Todos" selected), "Auxiliar:" (dropdown menu with "Todos" selected), "De Fecha:" (text input with "2012-12-04"), "A Fecha:" (text input with "2012-12-04"), "Calculados A:" (text input with "2012-12-04"), and "Indice Ufv:" (text input with "1"). Below these fields is a list of report types under the heading "Reportes:". The first option, "Inventario Ordenado porCodigo de Activo", is highlighted in blue. Other options include "Inventario Ordenado por Grupo Contable", "Inventario Ordenado por Grupo Contable y Auxiliar", "Inventario Ordenado por Oficina", and "Inventario Ordenado por Oficina y Responsable". At the bottom right of the form is a button labeled "Generar Reporte".

Figura 1: Formulario Generar Reportes

Fuente: Elaboración Propia

➤ **Pruebas del Subsistema de la Segunda Iteración: Caso de Prueba N° 2:
Gestión de Grupos y Auxiliares Contables**

- ✓ **Propósito:** Verificar el funcionamiento del formulario de Reporte de Activos Fijos.
- ✓ **Pre Requisitos:** La Unidad Administrativa está registrada, La Oficina y Responsable está registrada, El Grupo y Auxiliar Contable esta registrado, Rol de Usuario.
- ✓ **Datos de Prueba**
 - **Seleccionar Unidad(Todos)**
 - **Seleccionar Oficina(Todos)**

- **Seleccionar Responsable(Todos)**
- **Seleccionar Grupo(Todos)**
- **Seleccionar Auxiliar(Todos)**
- **Seleccionar De Fecha(Todos)**
- **Seleccionar A Fecha(Todos)**
- **Seleccionar Tipo de Reporte(Inventario Ordenado)**
- **Generar Reporte**

Nota: Los Valores Con Negrilla son Validos.

PASOS: Preparatorio:

Menú Reportes y Estadísticas>> Seleccionar Unidad Administrativa >> Seleccionar Oficina y Responsable >> Seleccionar Grupo y Auxiliar Contable >> Seleccionar De Fecha >> Seleccionar A Fecha >> Seleccionar Reporte

Resultados Esperados: Reporte Generado exitosamente y listo para imprimirse.



CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES



4.- CAPITULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1.- CONCLUSIONES

Con la Implementación del Sistema de Administración de Activos Fijos en este proyecto de grado se alcanzaron los objetivos propuestos, en el cual se presenta algunos puntos sobresalientes del sistema.

- ✓ Se diseño la estructura lógica del sistema.
- ✓ Se ha implementado módulos del sistema a partir de las especificaciones del diseño.
- ✓ La información de los activos fijos se encuentran centralizados.
- ✓ Se logro mayor seguridad en la administración de activos fijos.
- ✓ Se logro implantar el lector de código de barras mejorando la búsqueda de activos.
- ✓ Se obtiene información confiable, y en tiempo real que coadyuva a una administración eficiente y a la toma de decisiones.
- ✓ Se realizo las respectivas pruebas de funcionamiento del sistema.
- ✓ Se evaluó la calidad del software mediante la ISO 9126.

4.2.- RECOMENDACIONES

Al culminar el proyecto de grado se planteo las siguientes recomendaciones.

- ✓ Se sugiere crear nuevos módulos en el sistema de administración de activos fijos como trabajos complementarios.
- ✓ Se debe de hacer mantenimientos al sistema de acuerdo a las nuevos requerimientos o cambios en las normativas.
- ✓ El Sistema de Administración de Activos Fijos nos permite reforzar de una manera óptima el trabajo en la administración de los bienes del municipio de Cobija, también es justo mencionar que tiene algunas carencias de información en los ajustes con el VICE MINISTERIO DE PRESUPUESTO Y CONTADURIA, sobre todo en las cuentas que se manejan, ya que esta es muy amplia y se debe tomar mucho cuidado a la hora de hacer procedimientos de conciliación con otras cuentas contables, de los cuales manejan otros sistemas. Ejemplo. (El SINCON).

BIBLIOGRAFÍA

Beck, Kent (2000). Extreme Programming Explained: EmbraceChange. Longman

Funes, Orellana, Juan (2003). ABC de la contabilidad. Universidad Mayor de San Simón, UMSS. Bolivia.

Fernández, Escribano, Gerardo (2002). Ingeniería de Software II. Universidad de Castilla La Mancha. España

Pressman,(2002), Pruebas de la Calidad del Software.

Navarrete, Carmen (2001), Introducción a la Base de Datos. UAM IBM Centro de Referencia Linux.

[WEB2], Sql Server 2005. www.microsoft.com/SqlServer.

ANEXOS



ANEXOS

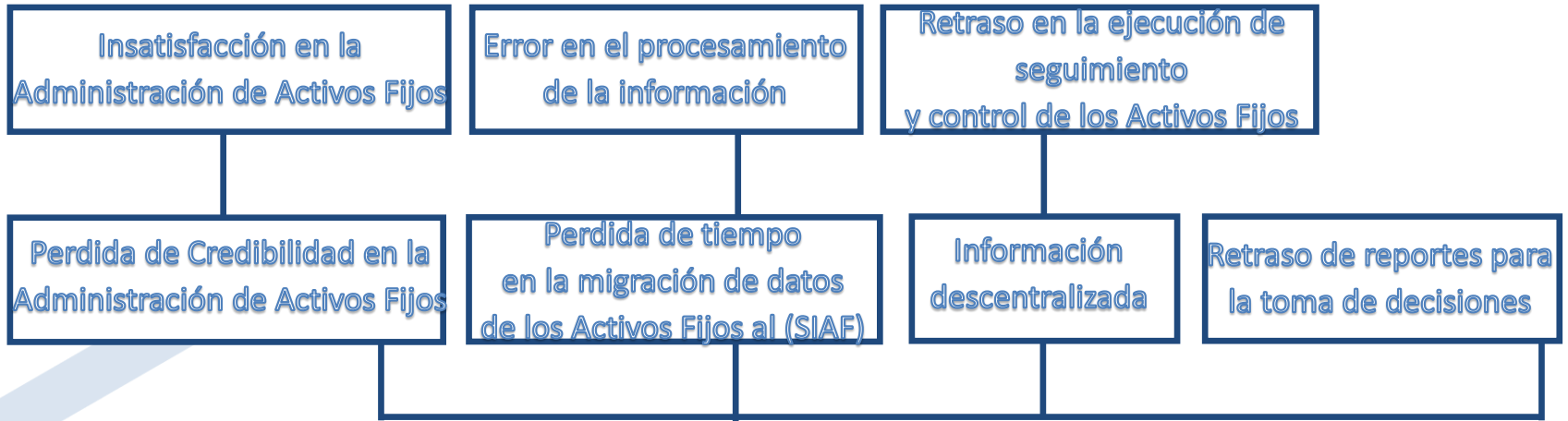
ÁRBOL DE

PROBLEMAS

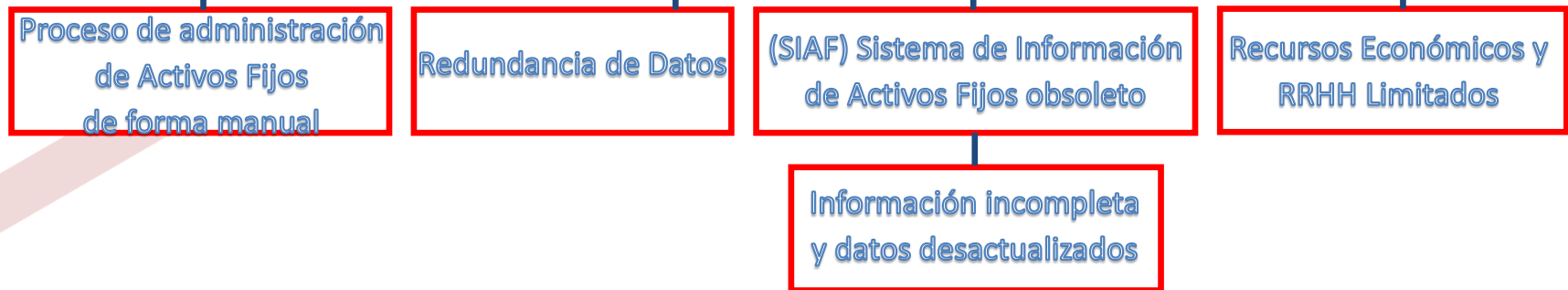


Árbol de Problemas

EFFECTOS



CAUSAS





ANEXOS

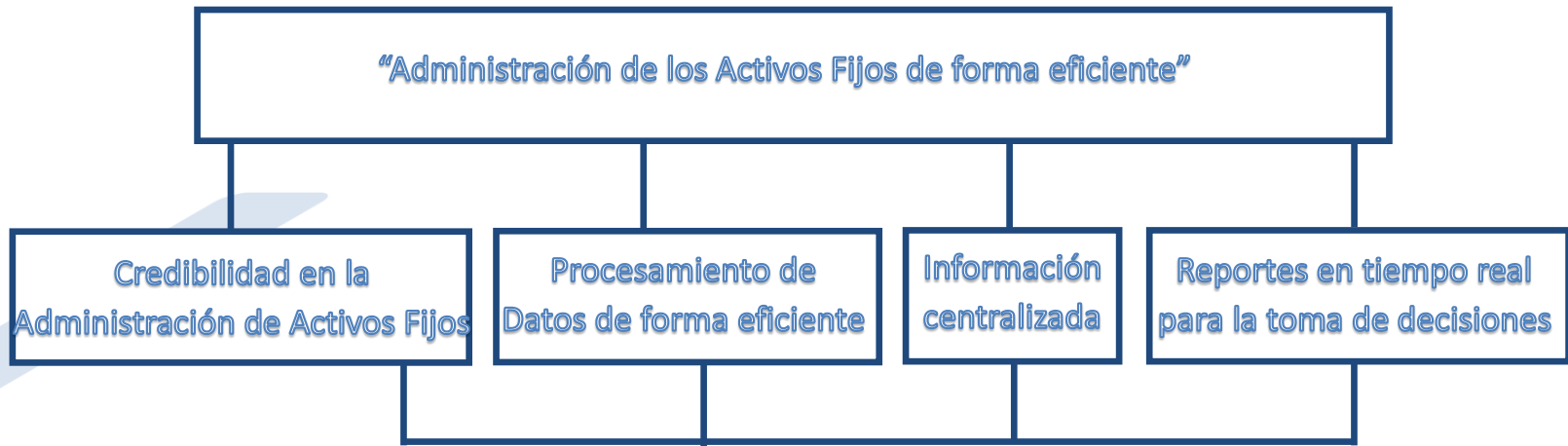
ÁRBOL DE

OBJETIVOS



Árbol de Objetivos

FINES



Desarrollar un sistema de información de Activos Fijos, para una administración eficiente de los Activos Fijos del Gobierno Autónomo Municipal de Cobija

MEDIOS





ANEXOS
PLAN DE
ENTREGAS



		Mes		1				2				3				4			
				Semana		1	2	3	4	1	2	3	4	1	2	3	4	1	2
N	HISTORIA																		
	PRIMERA ITERACION: ADMINISTRACION DE USUARIOS																		
1	Identificación y autenticación de usuarios																		
2	Gestión de Usuarios																		
3	Gestión de Unidades Administrativas																		
	ENTREGA																		
	SEGUNDA ITERACION: ADMINISTRACION DE ACTIVOS																		
4	Gestión de Grupos y Auxiliares contables																		
5	Gestión de oficinas y responsables																		
6	Gestión de Activos																		
	ENTREGA																		
	TERCERA ITERACION: CONTROL DE ACTIVOS																		
7	Control de Activos (Verificación)																		
8	Generar UFV																		
	ENTREGA																		
	CUARTA ITERACION: REPORTES																		
9	Generar Reportes																		
	ENTREGA																		

TABLA: Plan de Entregas
FUENTE: Elaboración Propia



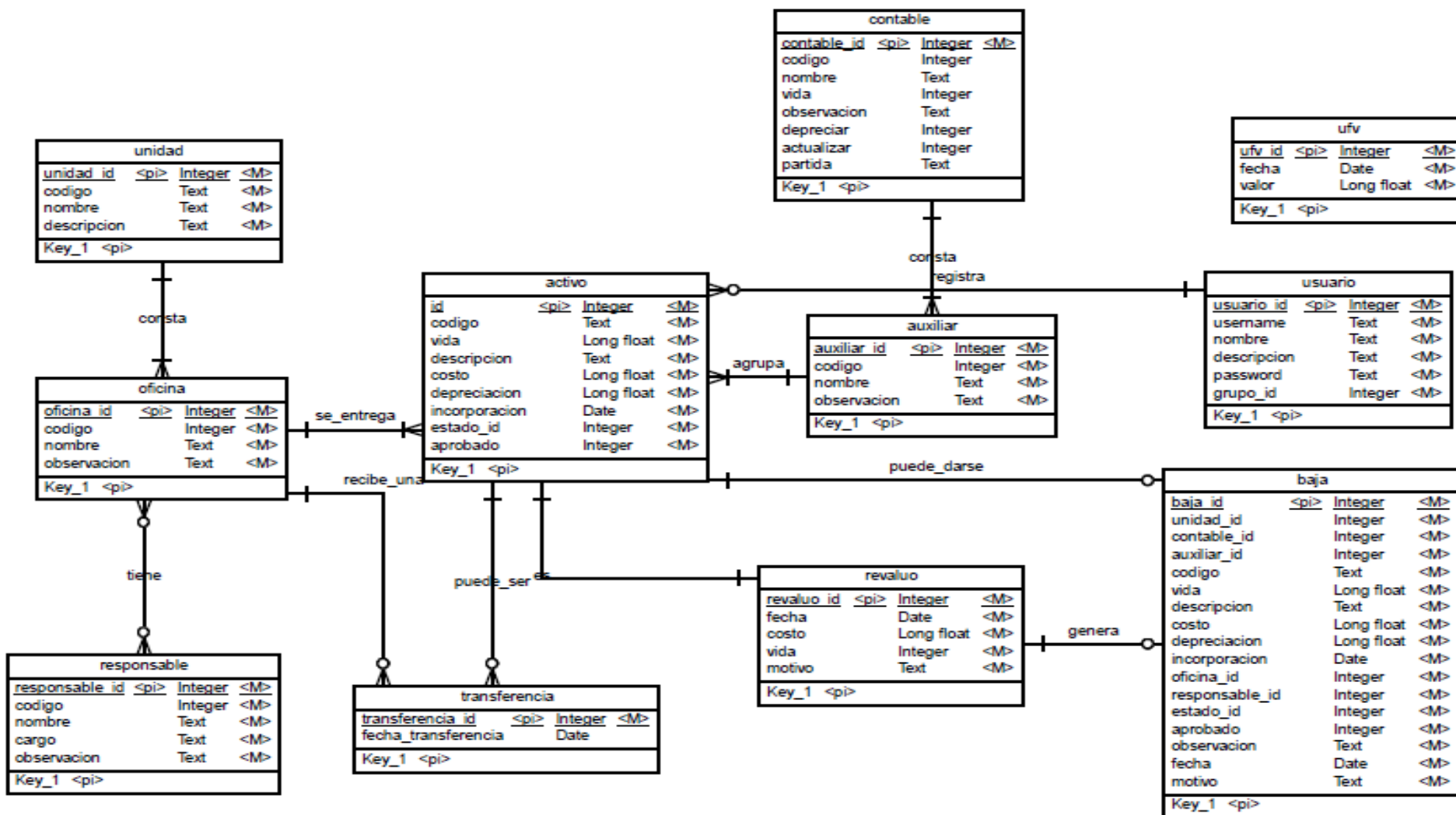
ANEXOS

DISEÑO DE LA

BASE DE DATOS



DISEÑO DE LA BASE DE DATOS





ANEXOS

HISTORIA DE LOS

USUARIOS



HISTORIA DE LOS USUARIOS

INGRESO DE ACTIVOS A LA UNIDAD DE ACTIVOS FIJOS

	HISTORIA DE USUARIO 1
USUARIO: Auxiliar de Activos Fijos	
NOMBRE DE LA HISTORIA: Ingreso de activos a la Unidad de Activos Fijos	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: 3	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: -La llegada de activos a la Unidad es acompañada con su correspondiente comprobante y factura, y esta es archivada en una carpeta al mes que le corresponda.	
Observaciones: -Una vez paralelamente se ingresa de activos a la Unidad se elabora su acta definitiva para hacer la entrega al responsable de dicho bien.	

Tabla 1. Historia de Usuario: Ingreso de Activos a la Unidad

Fuente: Elaboración Propia

AUTENTIFICACION DE DE USUARIOS

	HISTORIA DE USUARIO 2
USUARIO: General	
NOMBRE DE LA HISTORIA: Autenticación de Usuarios	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: 2	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	

Descripción: -Los usuarios deben autenticarse en el sistema con su nombre y password en el sistema.	
HISTORIA DE USUARIO 3	
Observaciones: USUARIO: General	
NOMBRE DE LA HISTORIA: Administración de Activos	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: General	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: -Los usuarios podrán realizar registros, modificaciones y eliminaciones, ajustes y estado de los activos.	
Observaciones:	

Tabla 2. Historia de Usuario N° 2: Autenticación de Usuarios

Fuente: Elaboración Propia

ADMINISTRACION DE ACTIVOS

Tabla 3. Historia de Usuario N° 3: Administración de Activos

Fuente: Elaboración Propia

OFICINAS Y RESPONSABLES

	HISTORIA DE USUARIO 5
USUARIO: General	
NOMBRE DE LA HISTORIA: Administración de Grupos y Auxiliares Contables	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: General	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: -Los usuarios podrán realizar registros, modificaciones y eliminaciones, ajustes y estado de los grupos y auxiliares contables.	
Observaciones:	

Tabla 4. Historia de Usuario N° 4: Administración de Oficinas y Responsables

Fuente: Elaboración Propia

ADMINISTRACIÓN DE GRUPOS Y AUXILIARES CONTABLES

	HISTORIA DE USUARIO 4
USUARIO: General	
NOMBRE DE LA HISTORIA: Administración de Oficinas y Responsables	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: General	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: -Los usuarios podrán realizar registros, modificaciones y eliminaciones, ajustes y estado de las oficinas y responsables.	
Observaciones:	

Tabla 5. Historia de Usuario N° 5: Administración de Grupos y Auxiliares Contables

Fuente: Elaboración Propia

GENERAR UFV

	HISTORIA DE USUARIO 7
USUARIO: General	
NOMBRE DE LA HISTORIA: Generar Reportes	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: General	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: -Los usuarios podrán generar reportes de acuerdo al tipo de reporte que desean seleccionar.	
Observaciones:	

Tabla 6. Historia de Usuario N° 6: GENERAR UFV

Fuente: Elaboración Propia

GENERAR REPORTES

	HISTORIA DE USUARIO 6
USUARIO: General	
NOMBRE DE LA HISTORIA: Generar UFV	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: General	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: -Los usuarios podrán realizar búsqueda de las UFV y también podrán actualizar las UFV.	
Observaciones:	

Tabla 7. Historia de Usuario N° 7: Generar Reportes

Fuente: Elaboración Propia

ADMINISTRACIÓN DE USUARIOS

Tabla 8. Historia de Usuario N° 8: Administración de Usuarios

Fuente: Elaboración Propia

	HISTORIA DE USUARIO 2
USUARIO: General	
NOMBRE DE LA HISTORIA: Administración de Usuarios	
PRIORIDAD EN LA TAREA: Alta	RIESGO EN EL DESARROLLO: Media
Puntos Estimados: General	INTERACCION ASIGNADA 1
PROGRAMDOR RESPONSABLE: HECTOR R. CHIPANA CH.	
Descripción: - Los usuarios podrán realizar registros, modificaciones y eliminaciones, ajustes y estado los usuarios del sistema.	
Observaciones:	



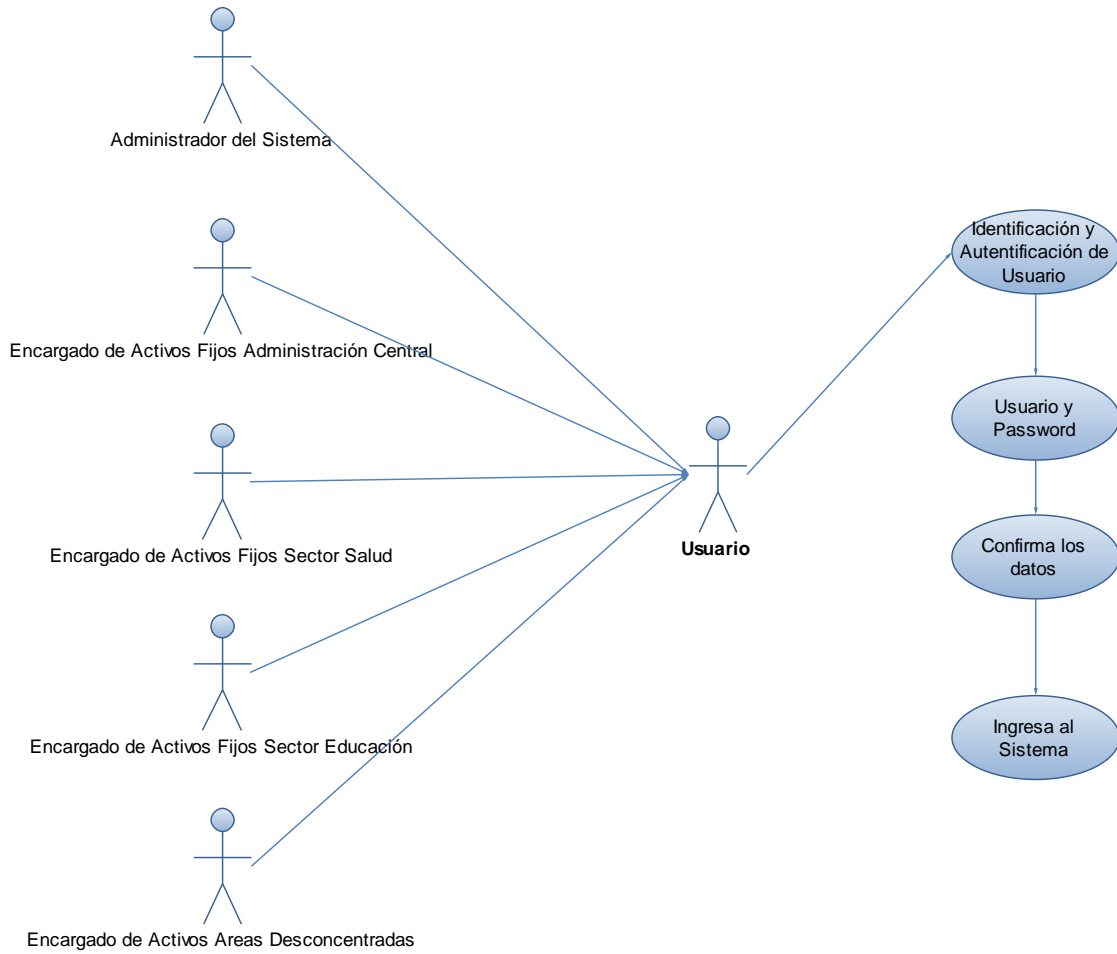
ANEXOS

**DIAGRAMA DE
CASOS DE USO**

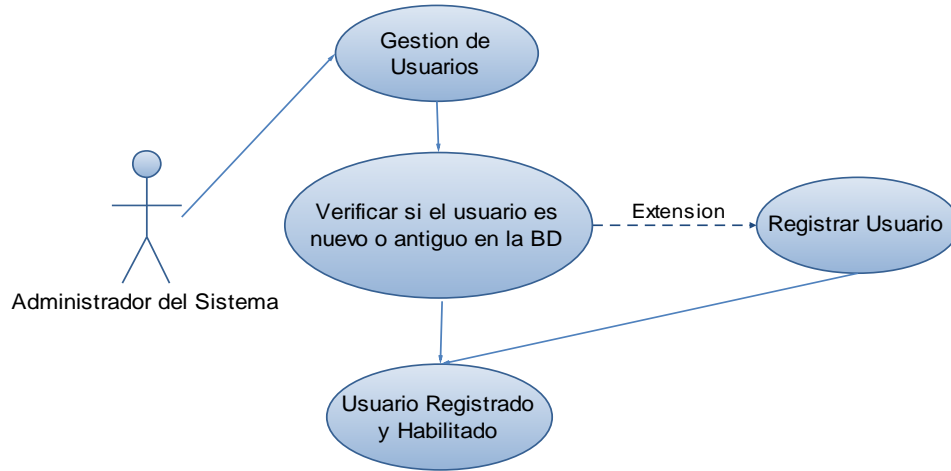


DIAGRAMA DE CASOS DE USO

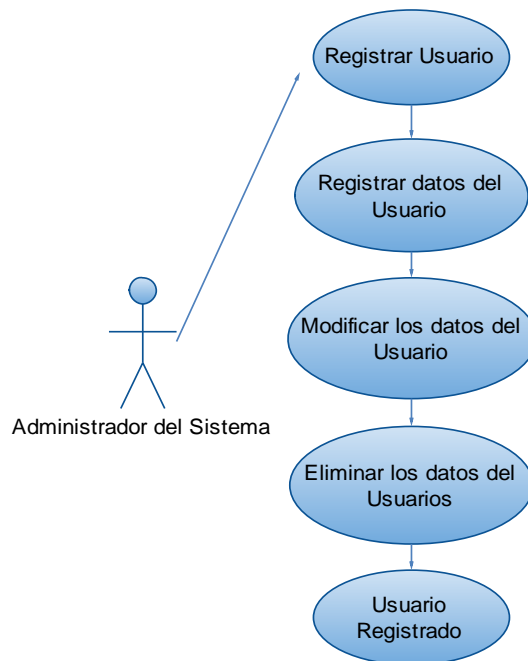
Identificación y Autenticación de Usuario



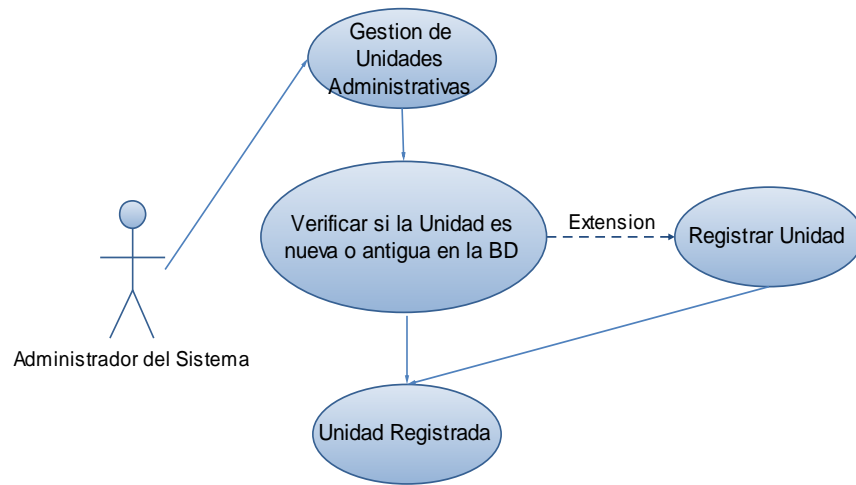
Gestión de Usuarios



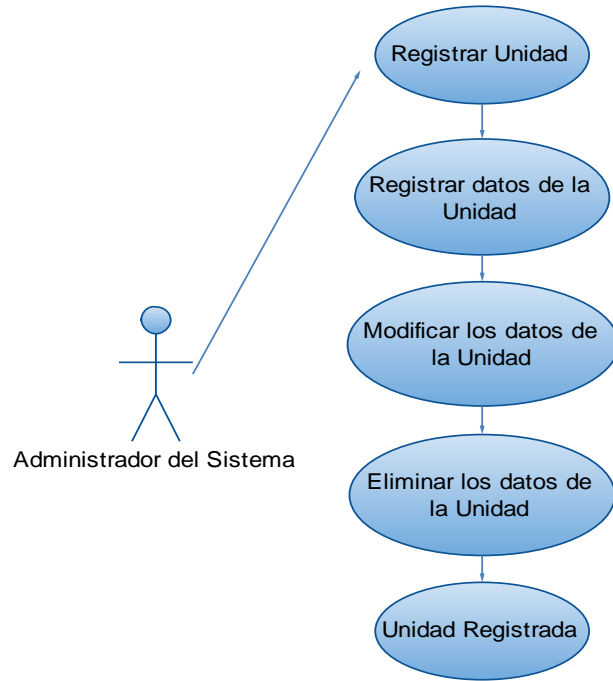
Registrar Usuario



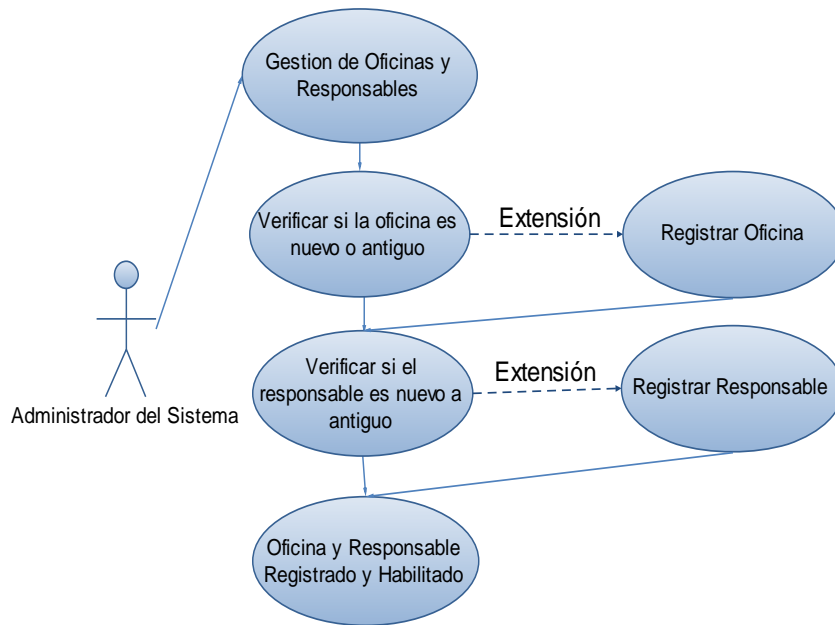
Gestión de Unidades Administrativas



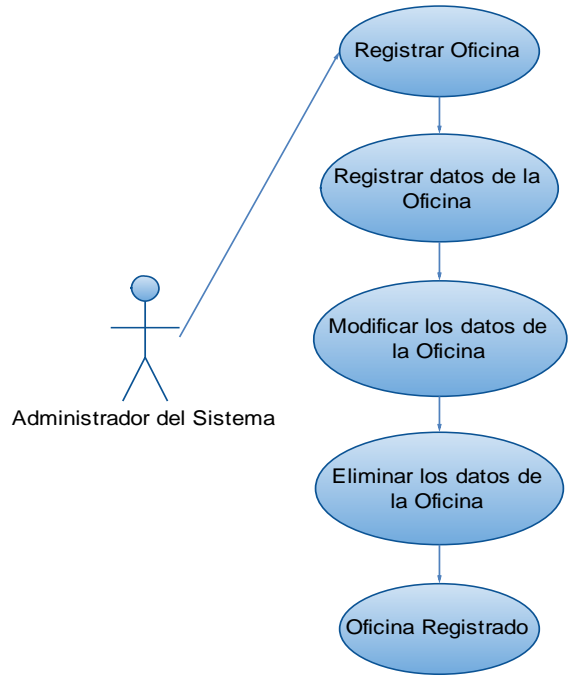
Registrar Unidad



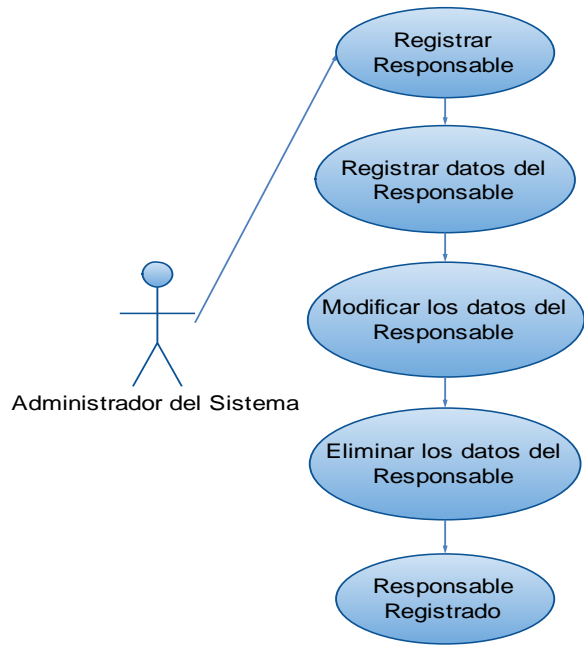
Gestión de Oficinas y Responsables



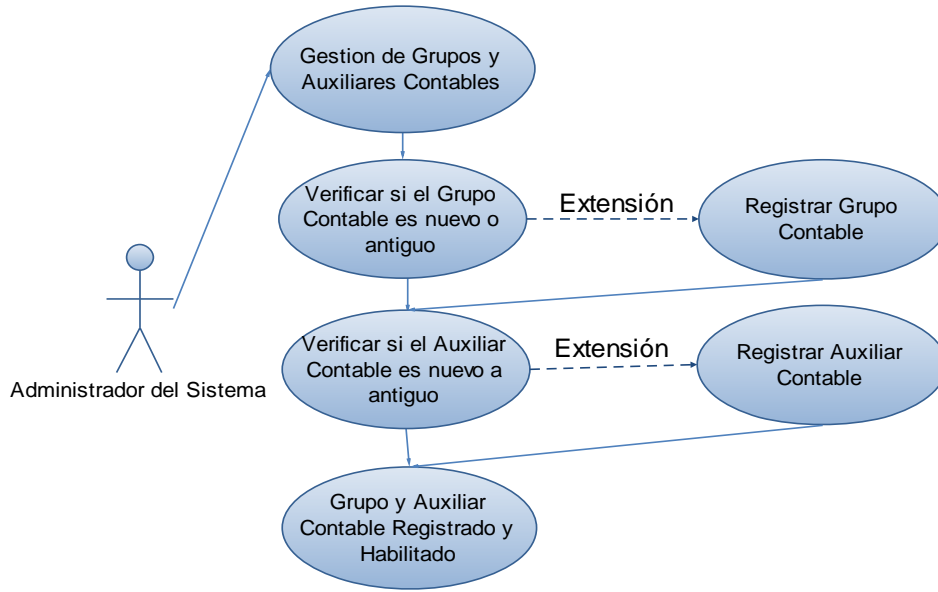
Registrar Oficina



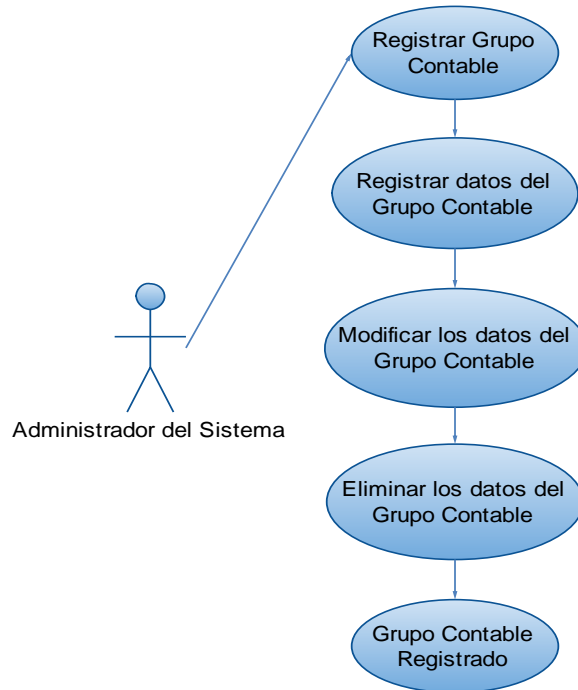
Registrar Responsable



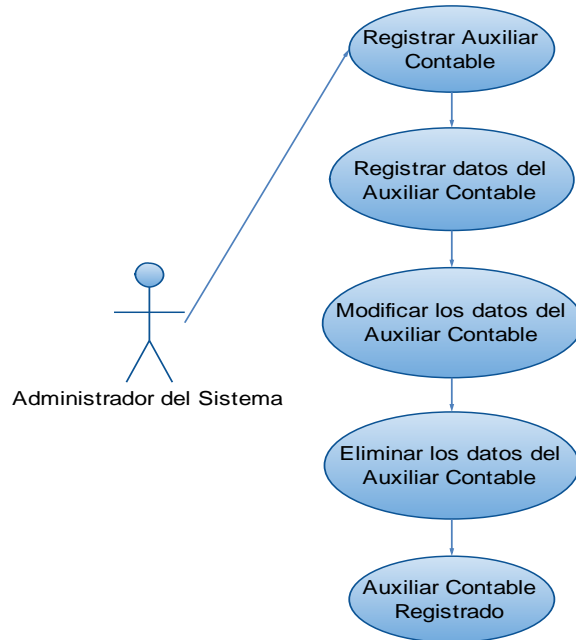
Gestión de Grupos y Auxiliares Contables



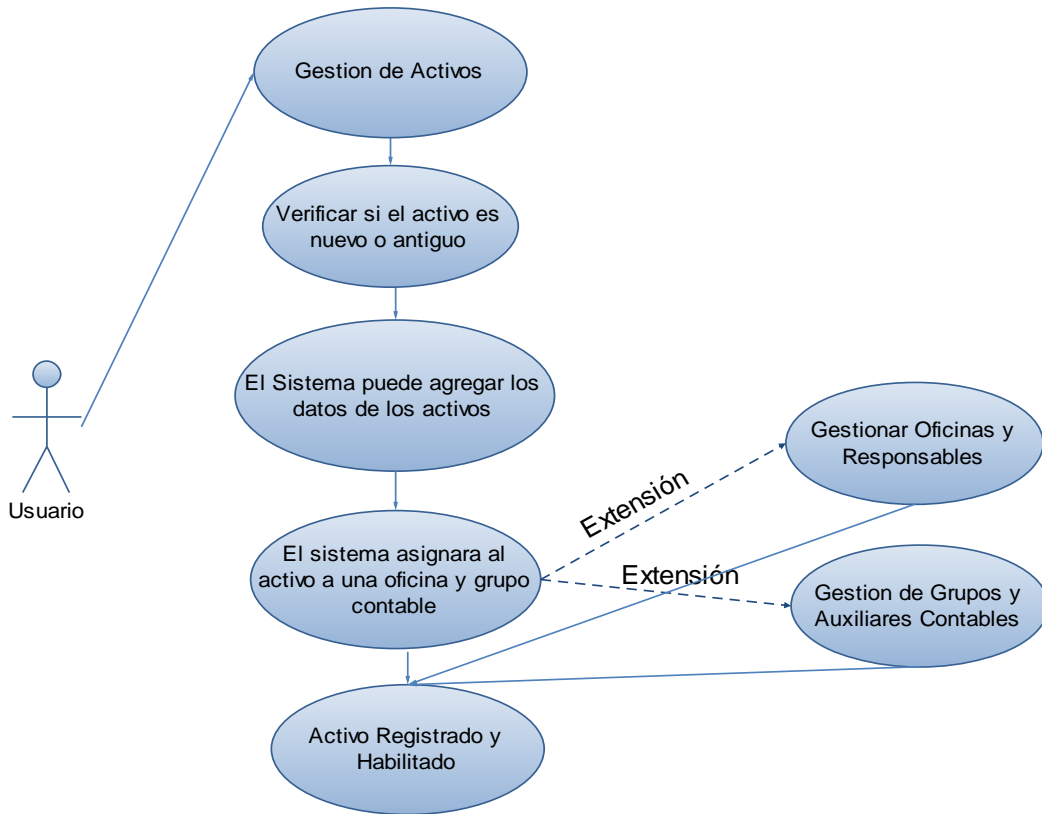
Registrar Grupo Contable



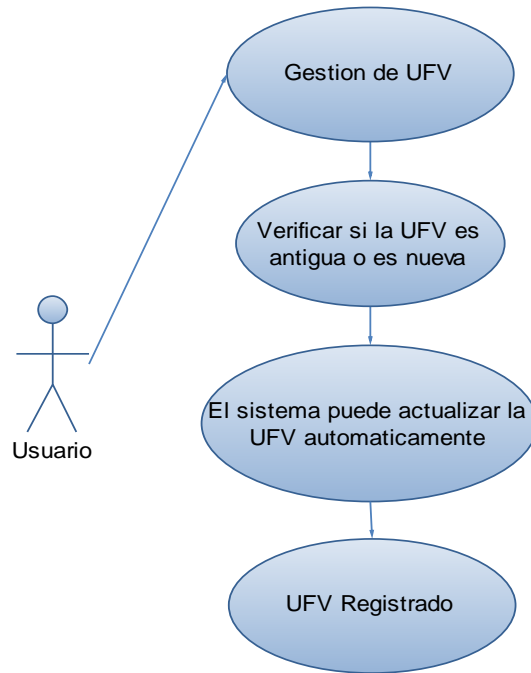
Registrar Auxiliar Contable



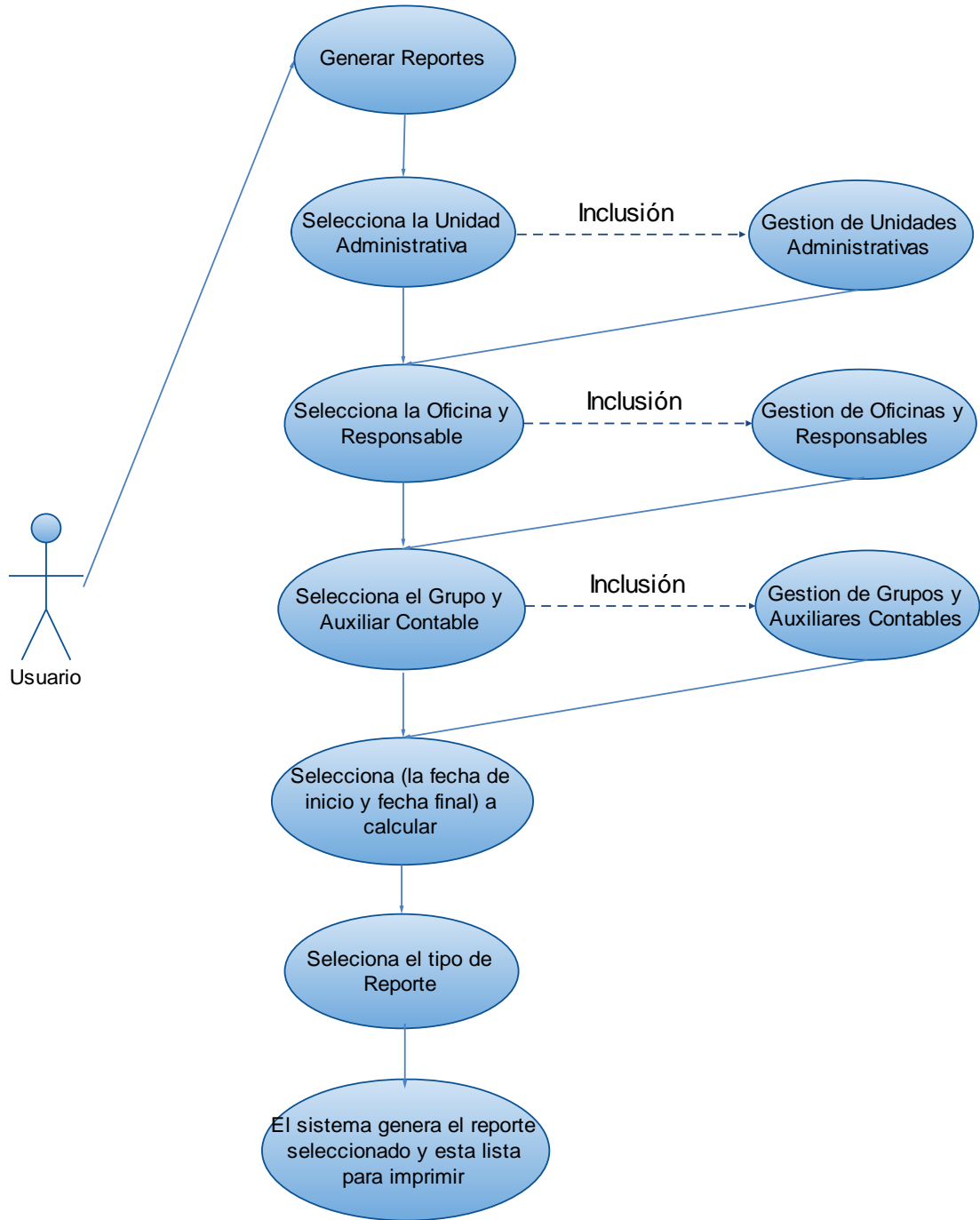
Gestión de Activos



Generar UFV



Generar Reportes





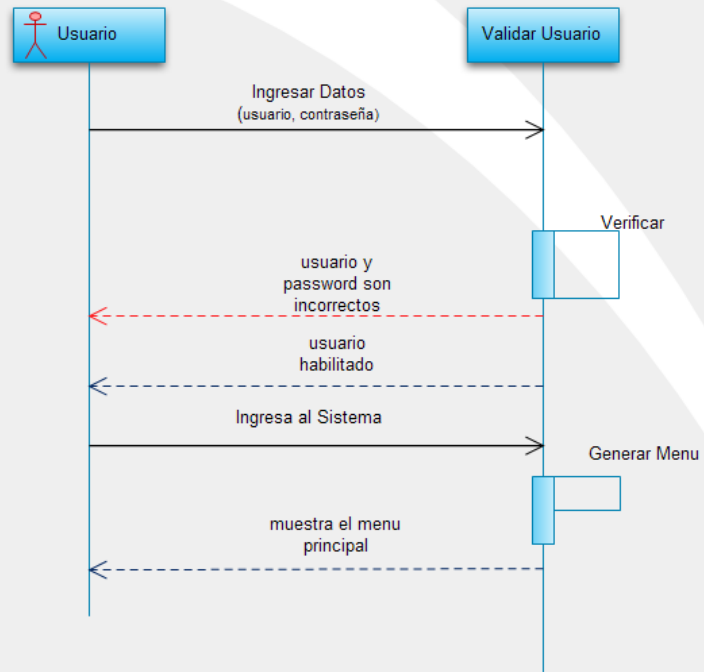
ANEXOS

DIAGRAMA DE

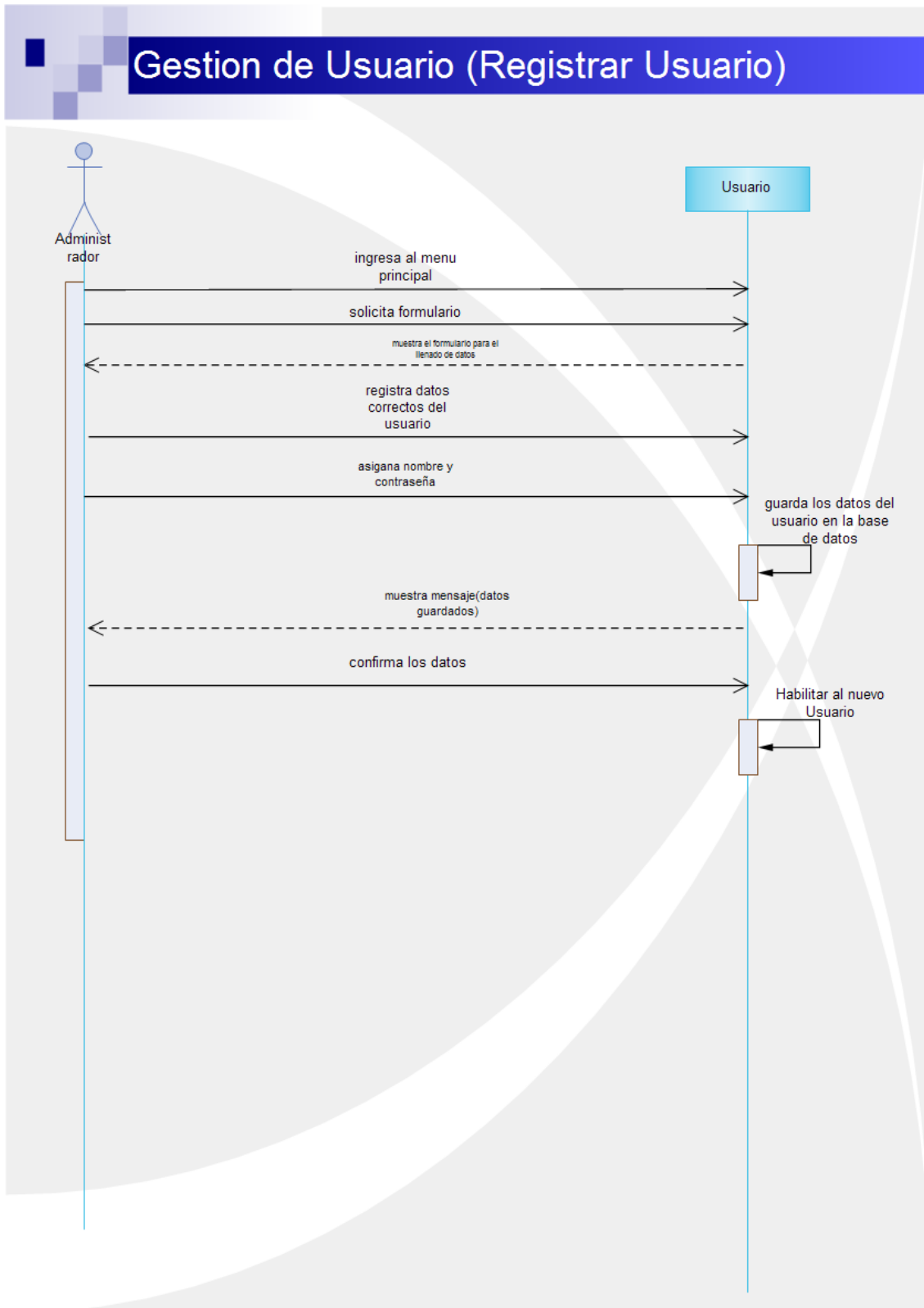
SECUENCIA



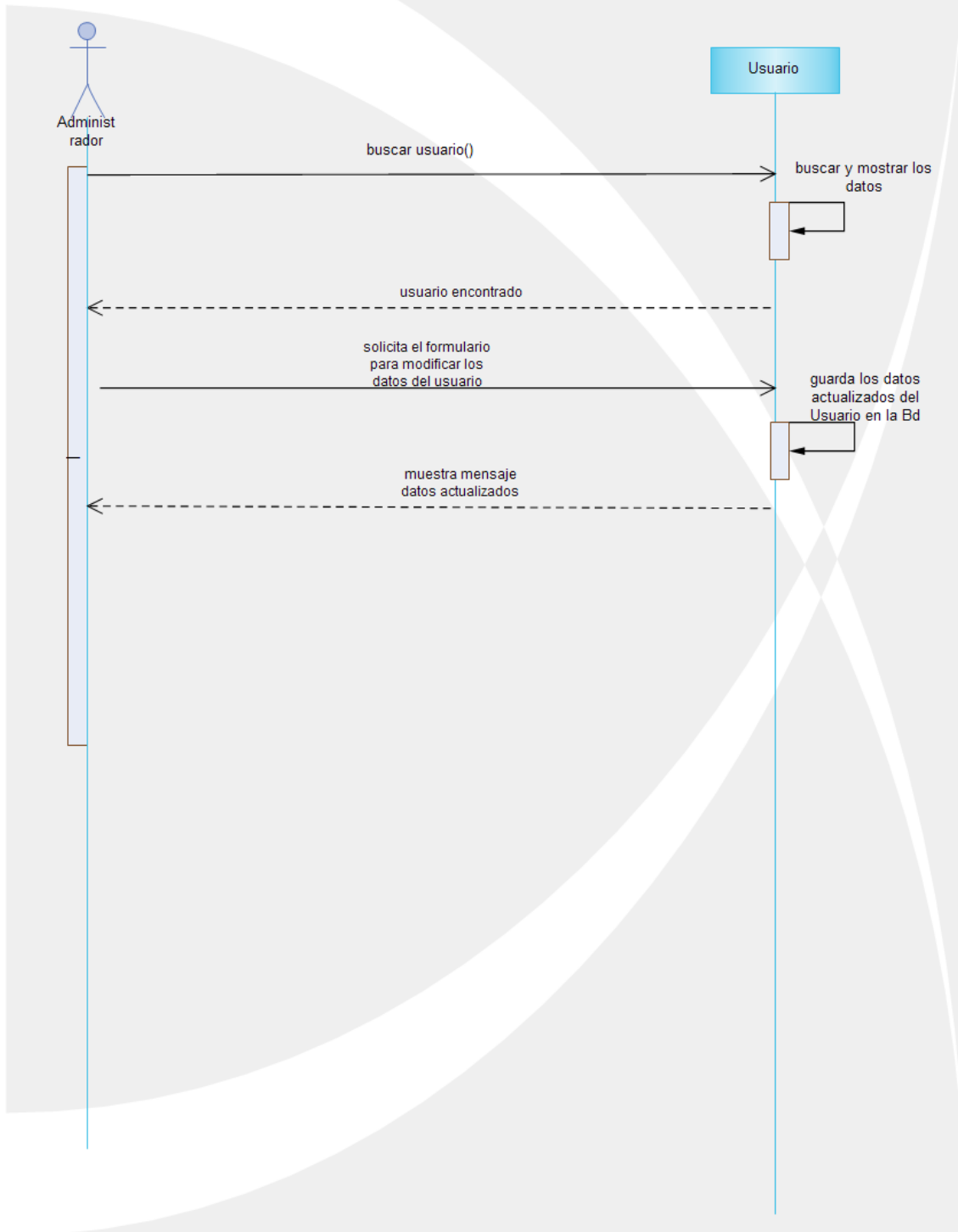
Autenticación de Usuario



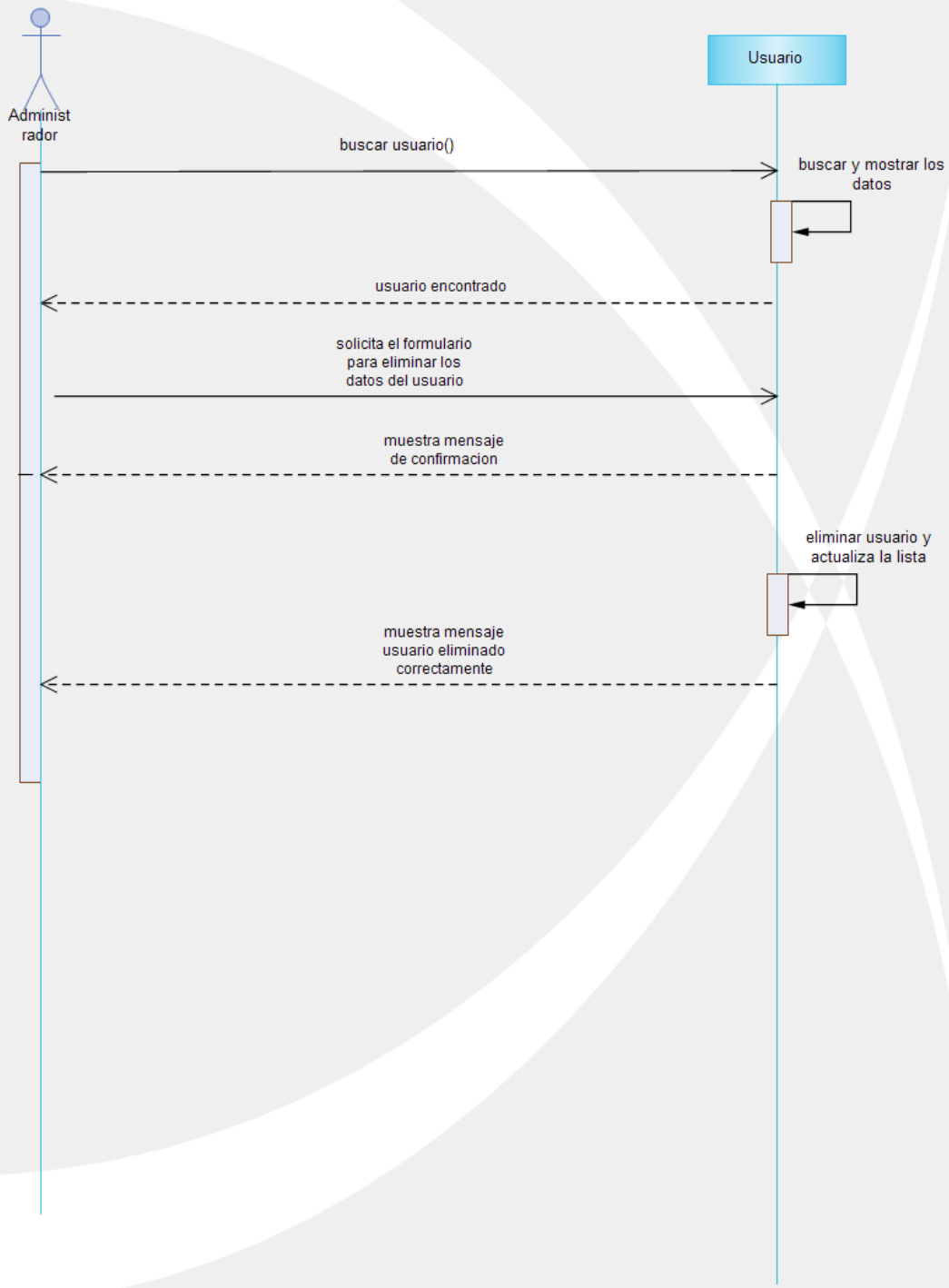
Gestion de Usuario (Registrar Usuario)



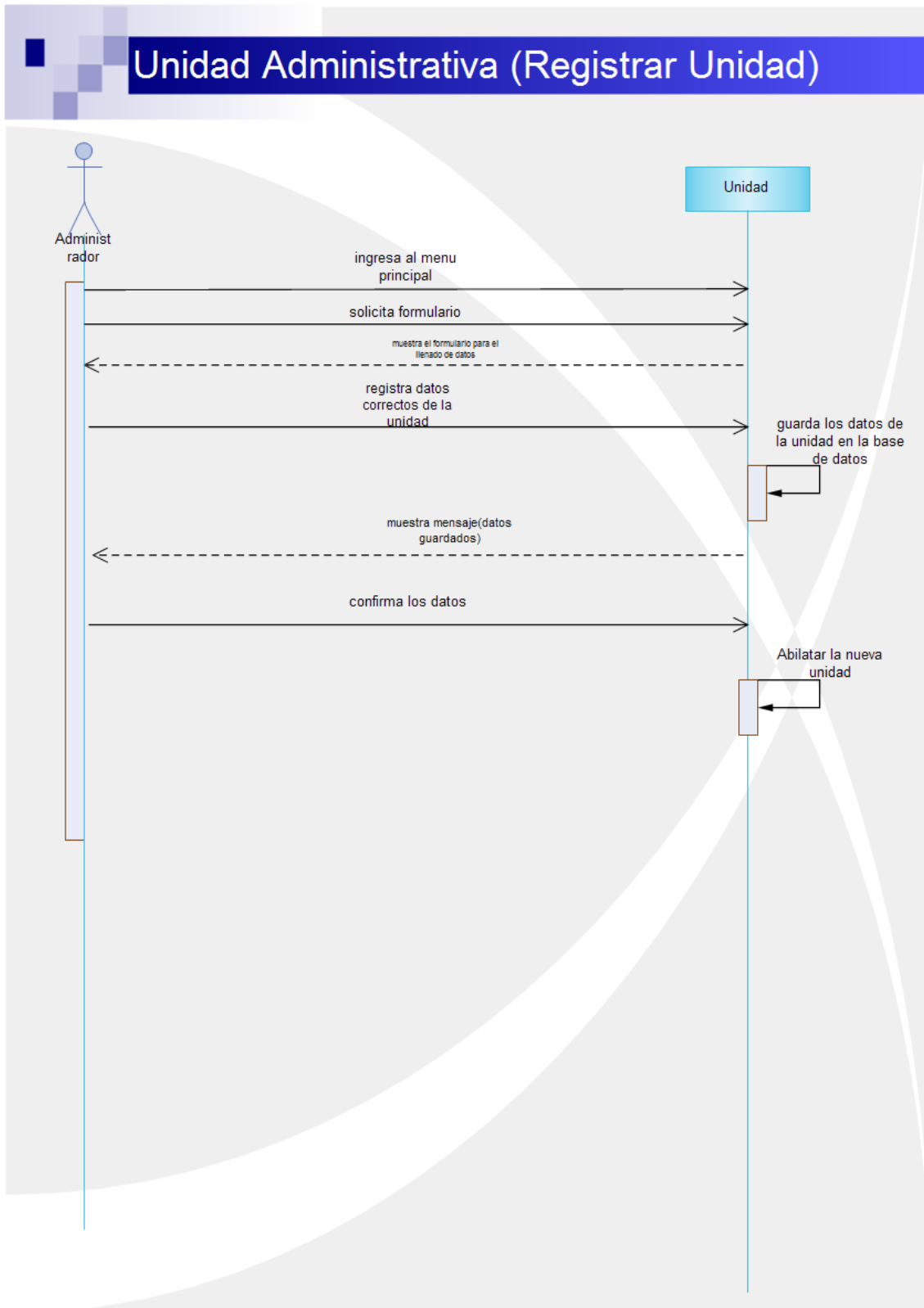
Gestion de Usuario (Modificar Usurio)



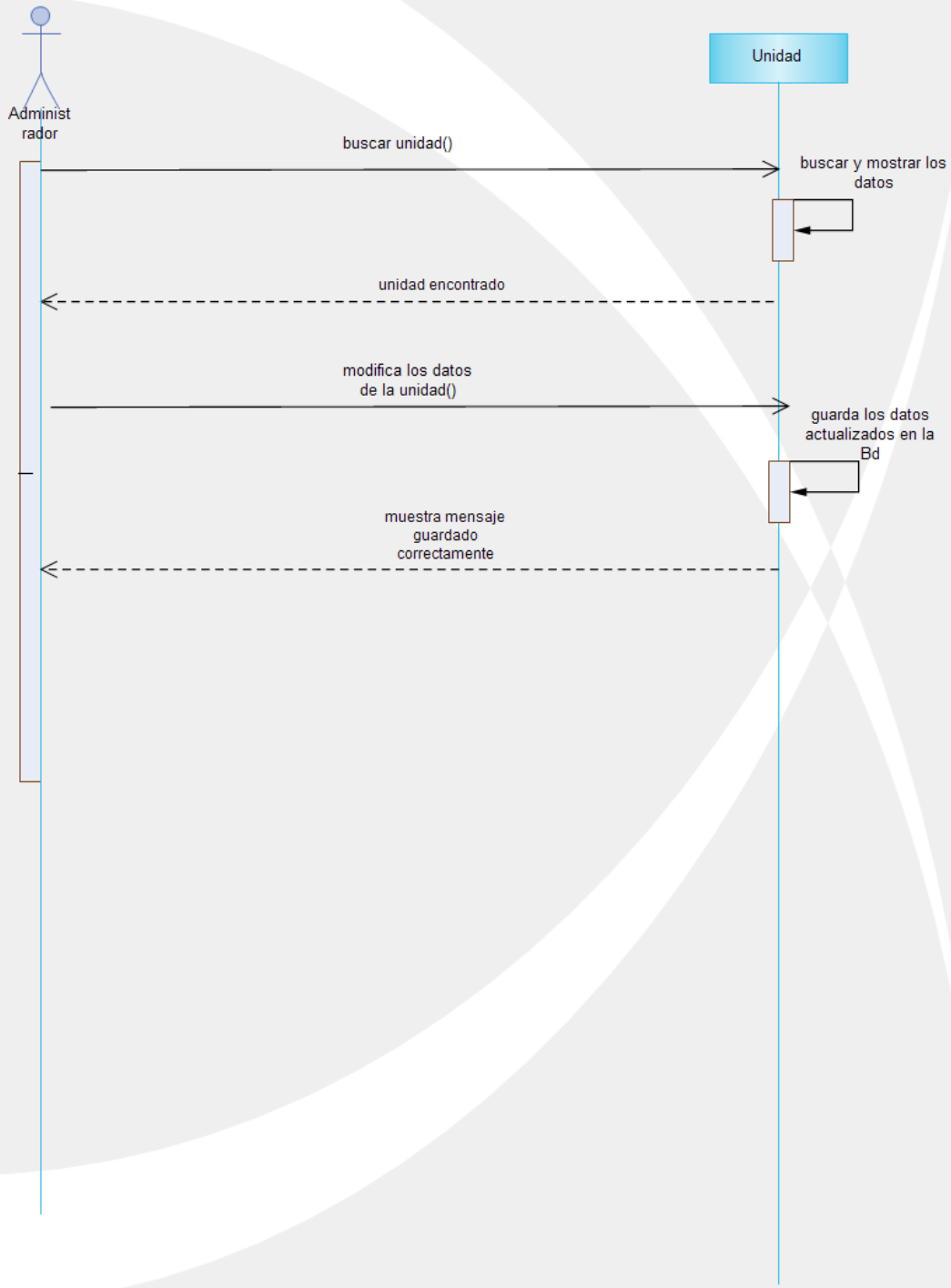
Gestion de Usuario (Eliminar Usuario)



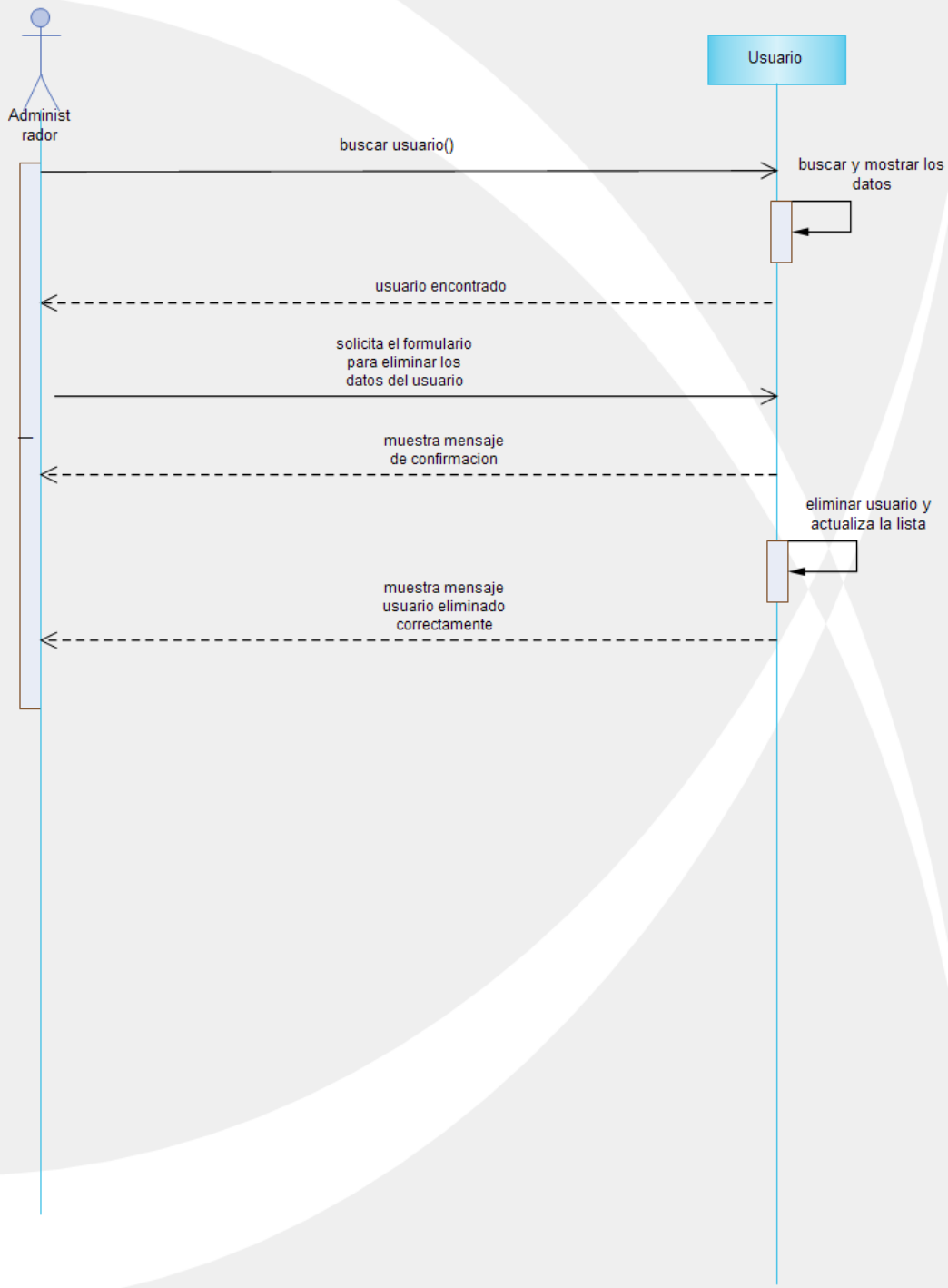
Unidad Administrativa (Registrar Unidad)



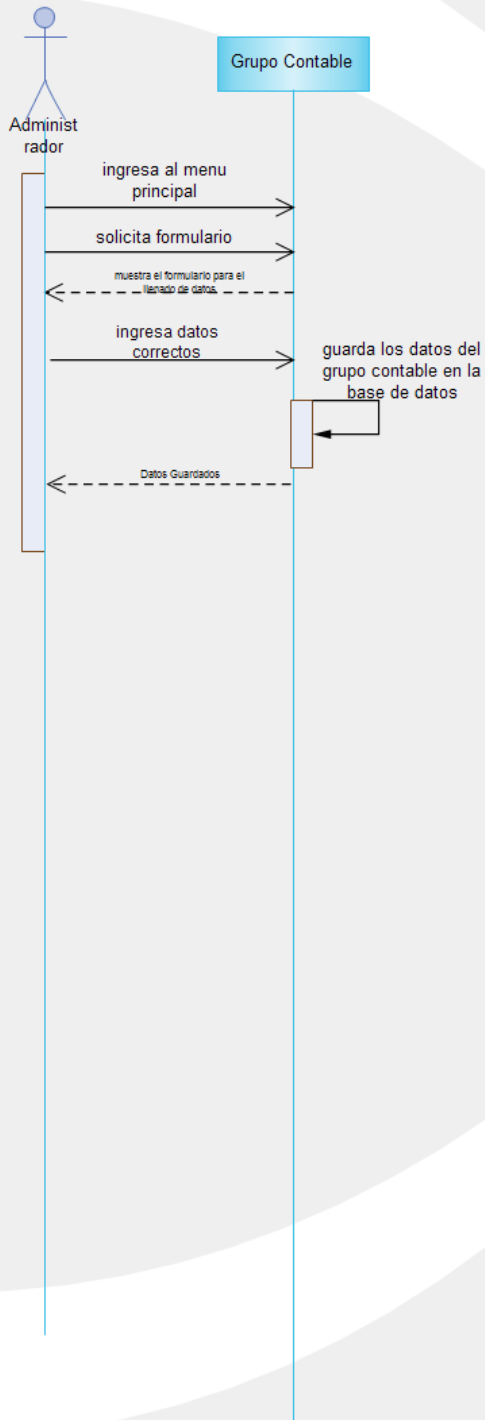
Unidades Administrativas(Modificar Unidad)



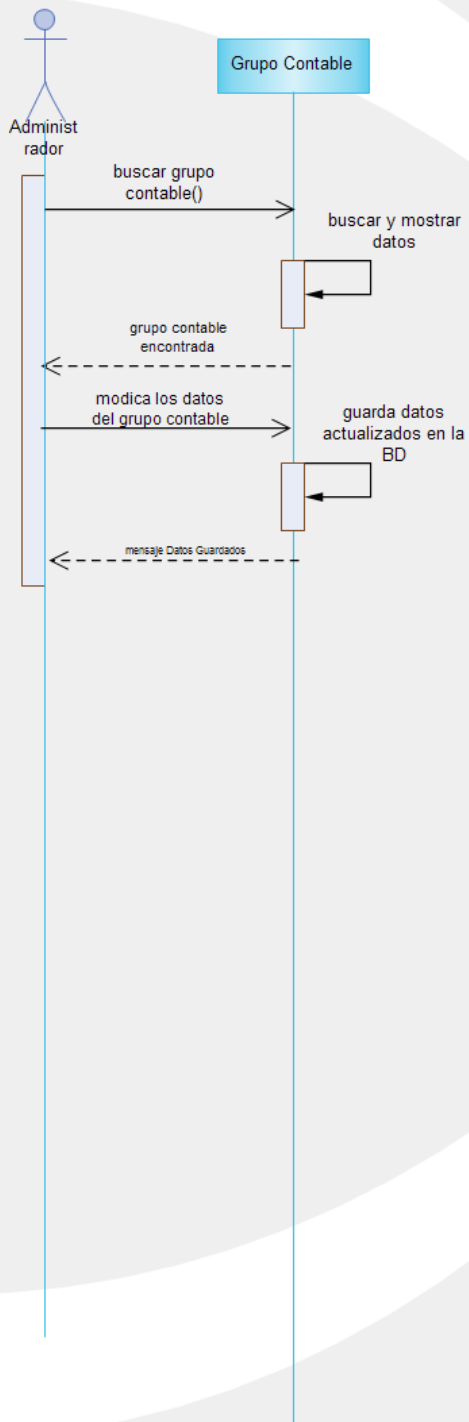
Gestion de Usuario (Eliminar Usuario)



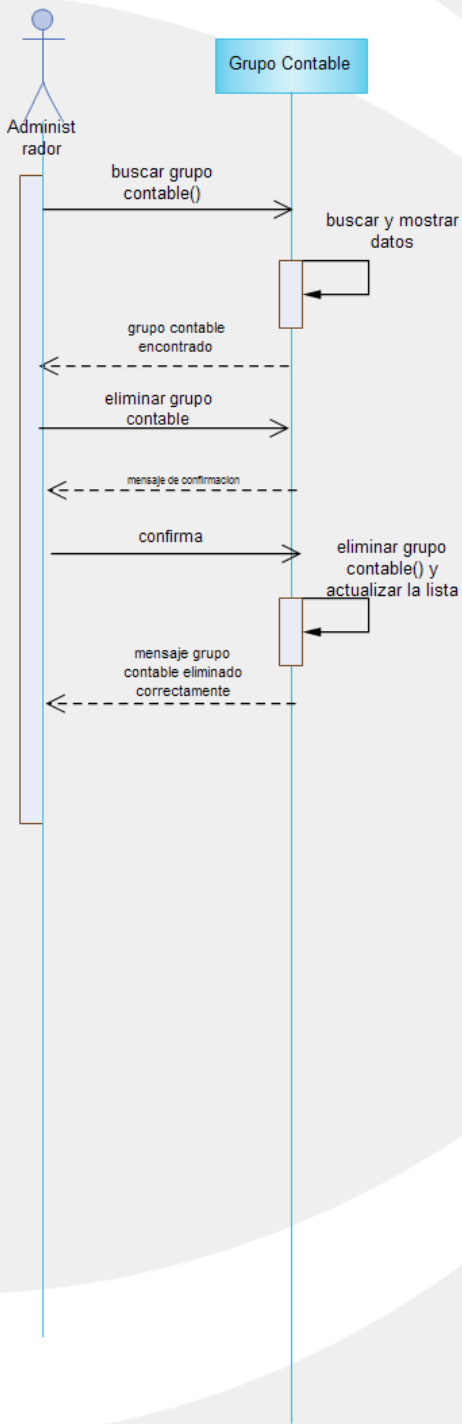
Gestion Grupos y Auxiliares Contables (Registrar Grupo Contable)



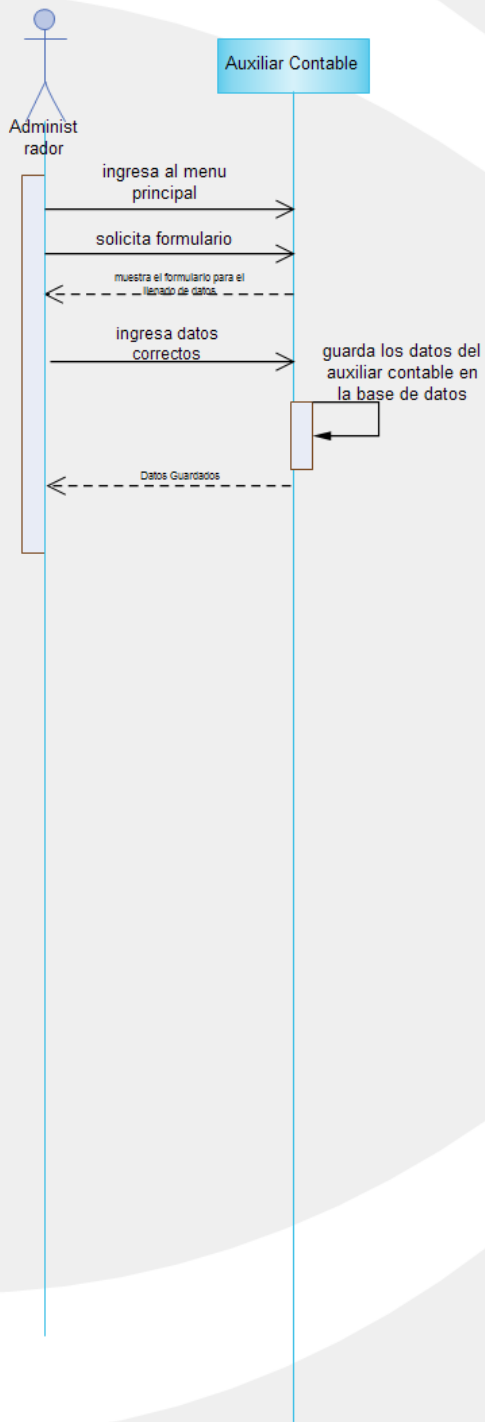
Gestion Grupos y Auxiliares Contables (Modificar Grupo Contable)



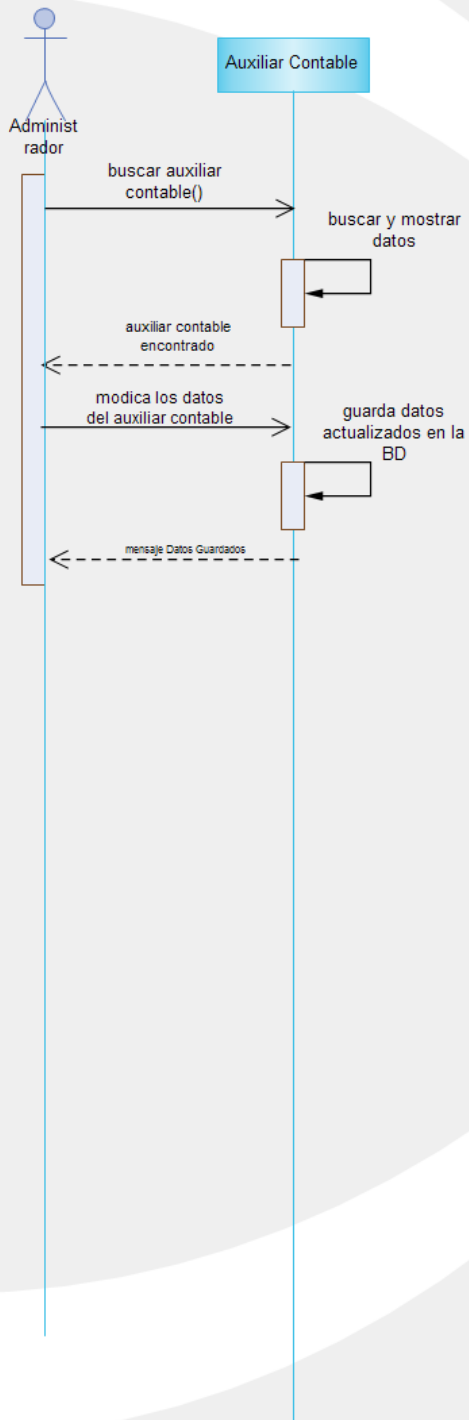
Gestion Grupos y Auxiliares Contables (Eliminar Grupo Contable)



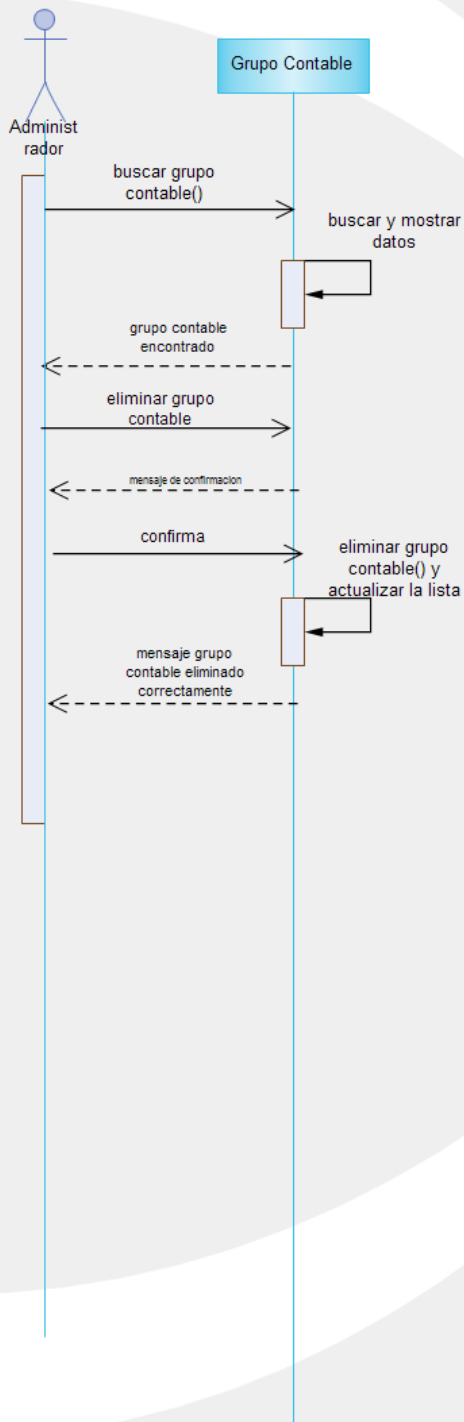
Gestion Grupos y Auxiliares Contables (Registrar Auxiliar Contable)



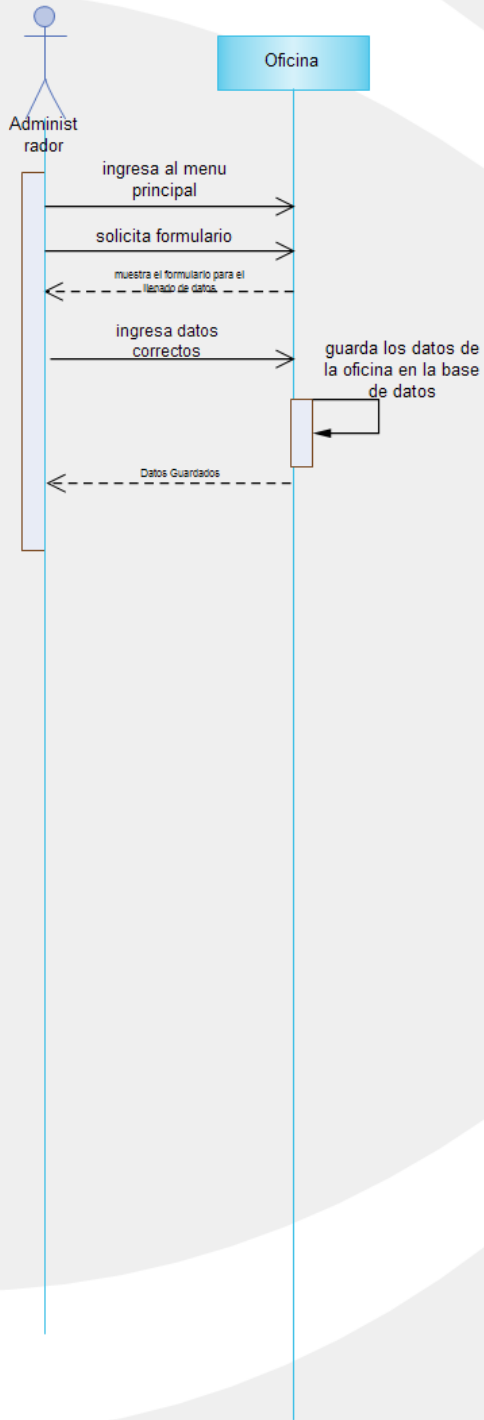
Gestion Grupos y Auxiliares Contables (Modificar Auxiliar Contable)



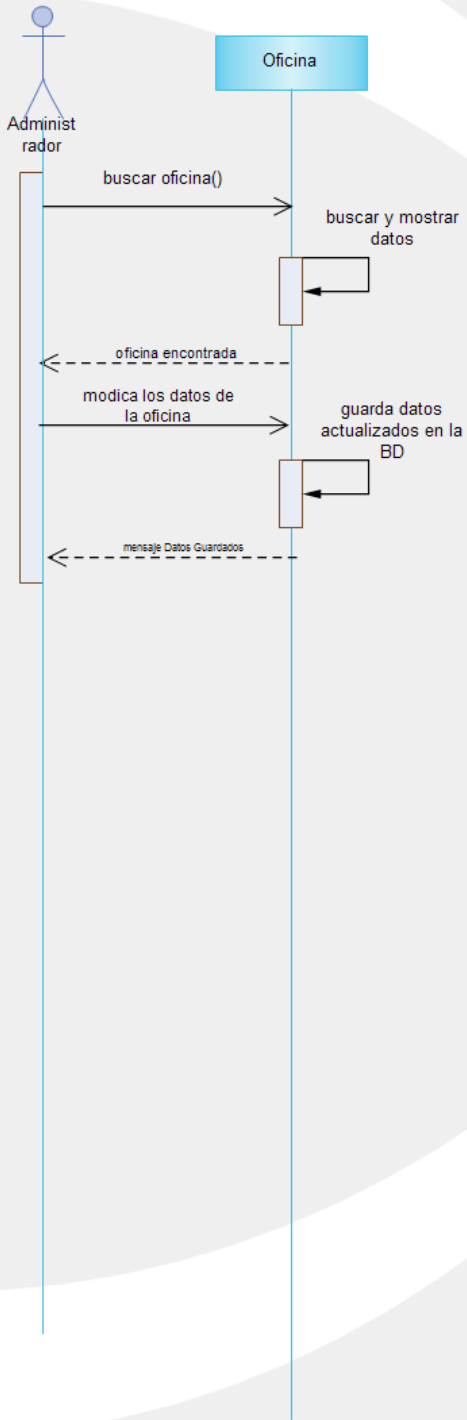
Gestion Grupos y Auxiliares Contables (Eliminar Grupo Contable)



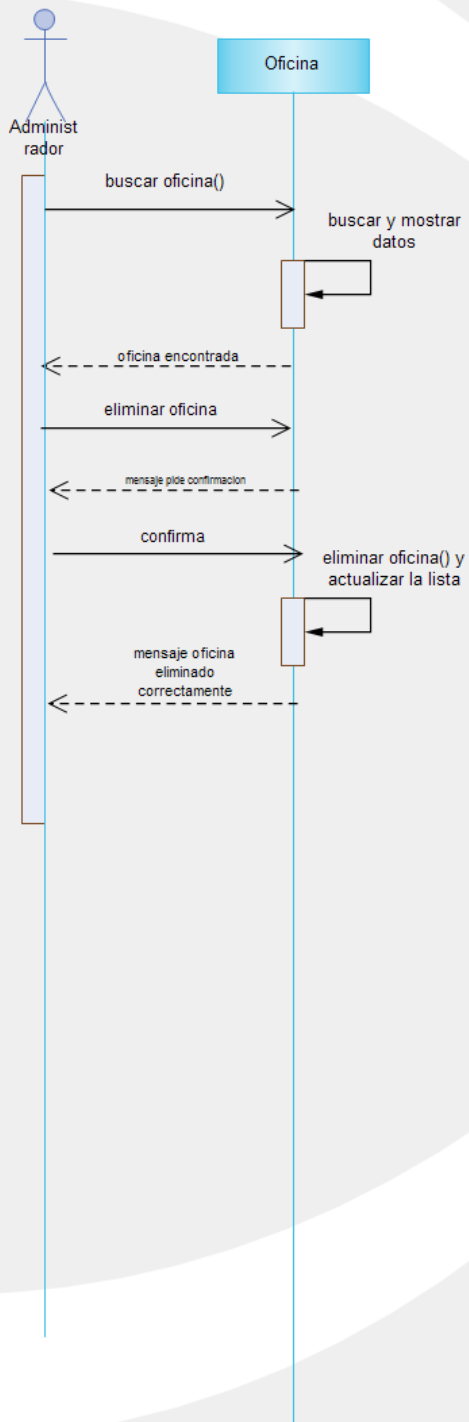
Gestion Oficina y Responsable (Registrar Oficina)



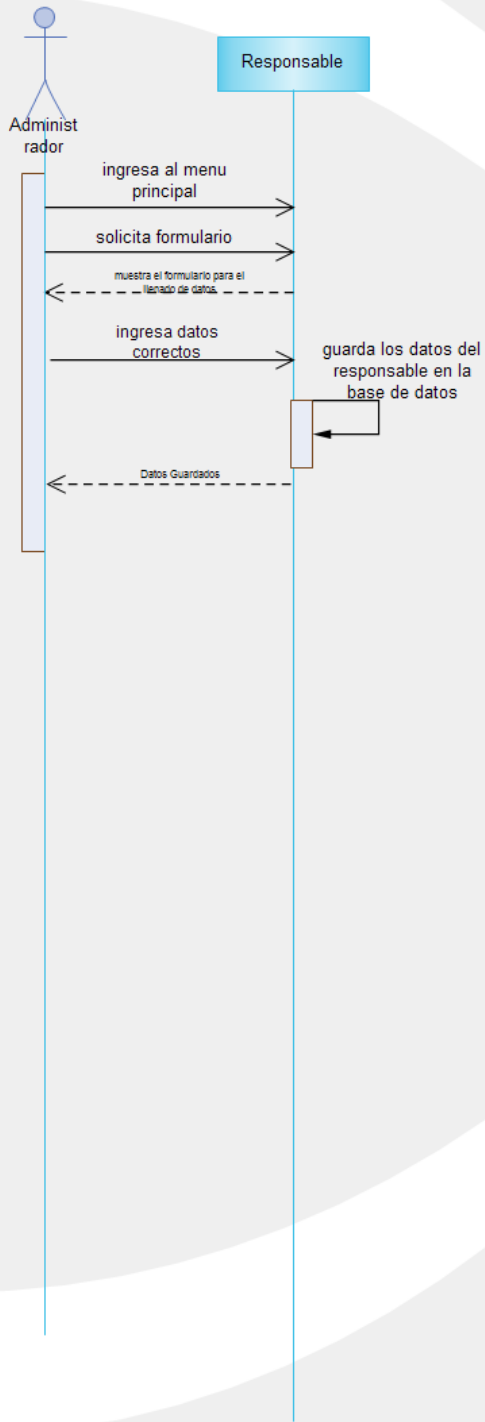
Gestion Oficina y Responsable (Modificar Oficina)



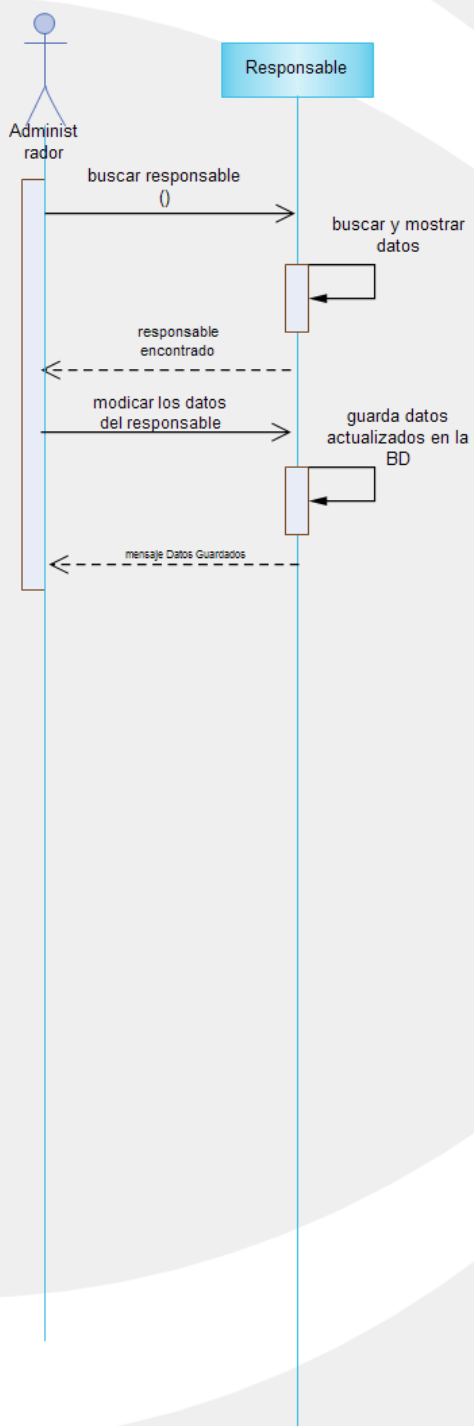
Gestion Oficina y Responsable (Eliminar Oficina)



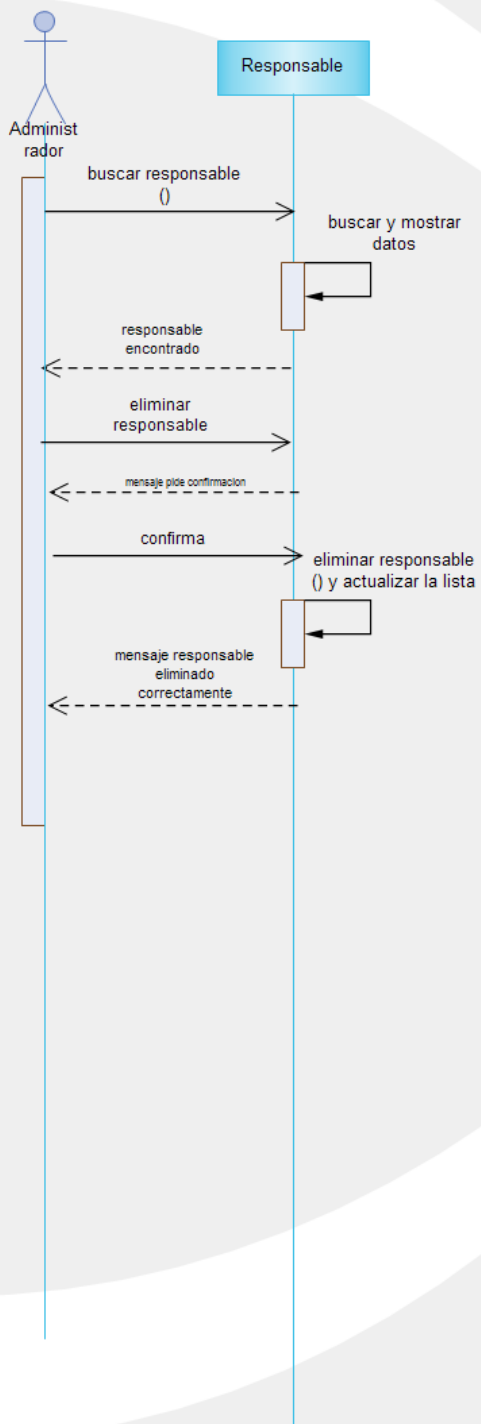
Gestion Oficina y Responsable (Registrar Responsable)



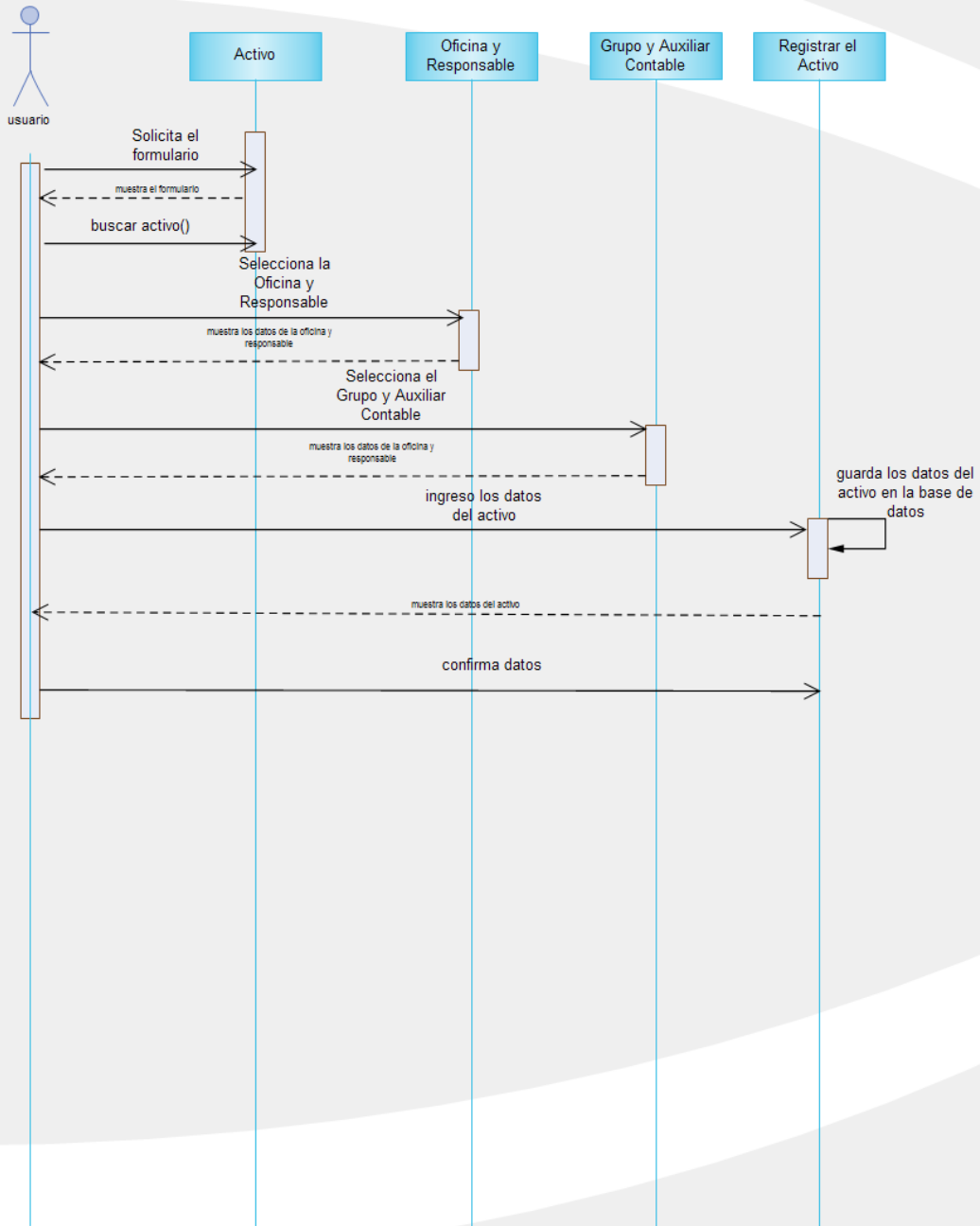
Gestion Oficina y Responsable (Modificar Responsable)



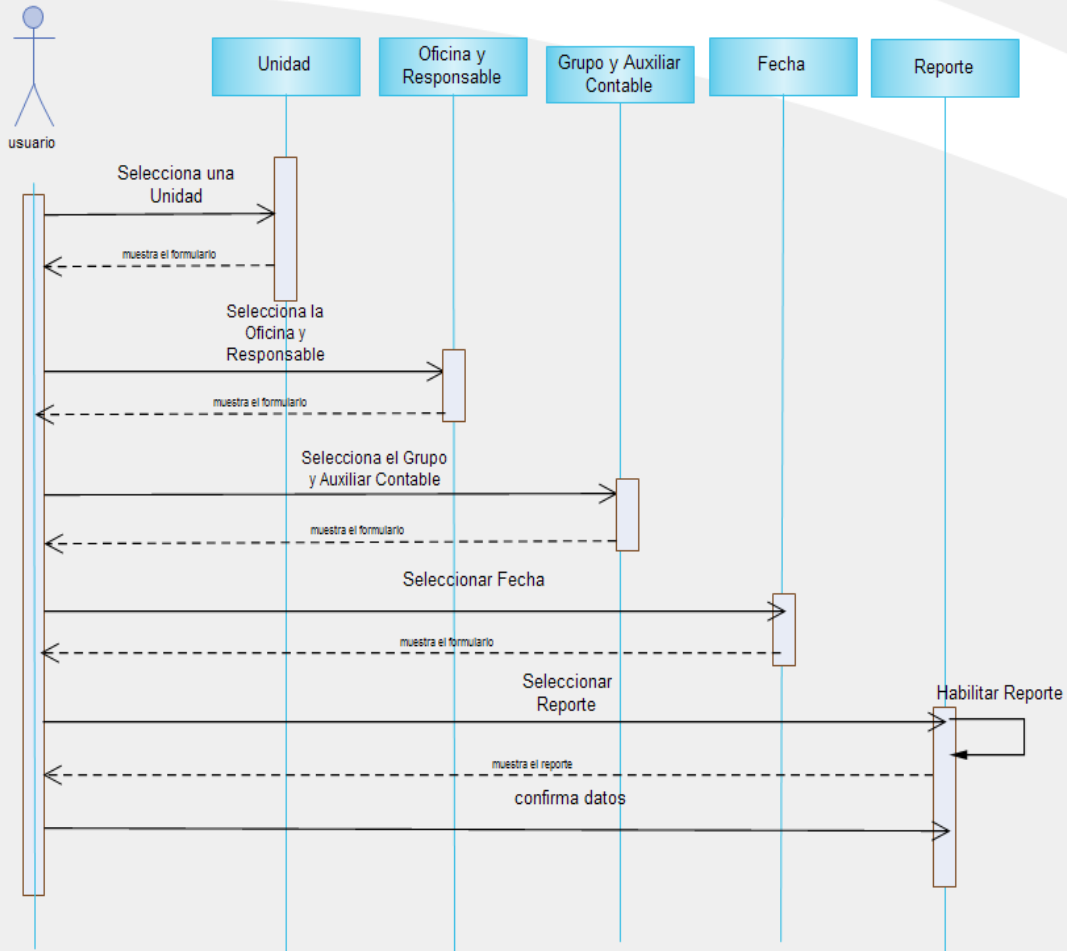
Gestion Oficina y Responsable (Eliminar Responsable)



Gestión de Activos



Generar Reporte





ANEXOS

EVALUACIÓN DEL SISTEMA MEDIANTE NORMA ISO 9126



**EVALUACIÓN DEL SISTEMA DE ACUERDO A LA NORMA ISO/IEC 9126 POR PARTE DE UN USUARIO
IDENTIFICADO COMO ADMINISTRADOR DEL SISTEMA**

DESCRIPCIÓN	VALOR DE CALCULO
MUY MALA	1
MALA	2
MEDIA	3
BUENA	4
MUY BUENA	5

REF	ATRIBUTO	CALIFICACIÓN
FUNCIONALIDAD		
1	El sistema realiza las tareas específicas en las historias de usuario?	
2	Cuando realiza una petición al sistema (ej. un reporte), esta, devuelve el o los valores que usted esperaba?	
3	El sistema pide un usuario y contraseña para acceder al sistema?	
4	El servidor de base de datos pide que se autentifique antes de ingresar?	
FIABILIDAD		
5	El registro de datos utilizando los formularios se encuentran debidamente validadas?	
6	En caso de existir un incidente que llegue a inutilizar el servidor, el tiempo de rehabilitación del sistema es menor a 1/2 día.	
USABILIDAD		
7	La interfaz de usuario le parece atractivo e intuitivo para su utilización?	
8	El sistema cuenta con los documentos (impreso, descripciones en la interfaz, etiquetas) necesarios para su aprendizaje?	
9	Puede usar el sistema haciendo solamente el uso del teclado, que permita mayor velocidad de procesamiento de información?	
EFICIENCIA		
10	El sistema responde de manera rápida a peticiones que realiza?	
MANTENIMIENTO		
11	Existe documentación del código fuente y la BD los cuales faciliten realizar modificaciones al sistema?	
12	Existen independencia entre la BD y el software?	
PORTABILIDAD		
13	Incompatibilidad con otros sistemas de información e interfieren la coexistencia?	
14	El sistema cuenta con un asistente que facilite una nueva instalación	
15	Es compatible con diferentes sistemas operativos?	