

**UNIVERSIDAD POLITÉCNICA
SALESIANA
FACULTAD DE INGENIERÍAS
SEDE QUITO-CAMPUS SUR
CARRERA DE INGENIERÍA DE SISTEMAS
MENCIÓN TELEMÁTICA**

**SISTEMA INFORMATICO PARA EL MANEJO Y CONTROL DE
INFORMACIÓN DE ACTIVOS FIJOS EN EL COLEGIO
PARTICULAR PRADO MEDIANTE LA UTILIZACION DE
TECNOLOGÍA MÓVIL**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SISTEMAS**

AMAYA PAZMIÑO MARCO ANTONIO

CRUZ LÓPEZ WILSON FERNANDO

DIRECTOR: ING. PATSY PRIETO

Quito, Enero 2011

DECLARACIÓN

Nosotros Marco Antonio Amaya Pazmiño y Wilson Fernando Cruz López, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Marco Antonio Amaya Pazmiño

Wilson Fernando Cruz López

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Marco Antonio Amaya Pazmiño y Wilson Fernando Cruz López.

ING. PATSY PRIETO

AGRADECIMIENTOS

A nuestra querida Universidad Politécnica Salesiana, a todos los Ingenieros y maestros que nos impartieron sus enseñanzas, experiencias y conocimientos para poder afrontar los problemas y riesgos que se presenten en nuestra vida profesional. A nuestras familias ya que todos y cada uno aportaron con su granito de arena y su apoyo para poder culminar esta etapa de nuestra formación profesional y personal.

DEDICATORIA

Fernando Cruz dedica a:

Al Señor Jesucristo, mi Señor y Dios, por enseñarme el camino correcto de la vida, guiándome y fortaleciéndome cada día con su Santo Espíritu.

A mi querida Esposa Lucia Anabel Gálvez Pérez, mi ayuda idónea, por su amor, paciencia, comprensión y motivación, sin lo que hubiese sido imposible lograr terminar estos estudios, como también a mi querida Hija María Paz Cruz y mi nuevo hijo que viene en camino-

A mis Padres, Hermano y Hermanas y sobrinos por creer y confiar siempre en mí, apoyándome en todas las decisiones que he tomado en la vida.

A mis maestros, en especial a la Ing. Patsy Prieto, por sus consejos y por compartir desinteresadamente sus amplios conocimientos y experiencia.

A mis compañeros y compañeras de clases, por el apoyo y motivación que de ellos he recibido.

Marco Amaya dedica a:

A Dios por todas las experiencias y vivencias que me ha dado en la vida, la enseñanza de cada día para seguir adelante siempre y darme los dos mejores regalos de mi vida, Damián y Esteban.

A mi madre que siempre está a mi lado y me dio todo su apoyo, amor y paciencia para poder culminar esta meta; todo esto es por ti, para ti mami.

A mi padre, que con su sabiduría y ejemplo ha sido mi mentor y admiración en la vida.

A mis hijos por ser mi inspiración en esta vida, por darme su cariño y amor incondicional que cada día me da la fuerza para luchar y seguir adelante.

A mis hermanas, quienes me dieron su ayuda siempre que necesite, sobre todo en el inglés.

Al Pepito y a la Maru quienes fueron una base importante en mi educación, el inicio de todo esto.

A todos los amigos(as); del barrio, de la U y del trabajo que aunque sigan pasando los años siempre están para los acolites de los problemas y las jodas de la vida.

CONTENIDO

CAPITULO 1.....	16
SUSTENTO TEÓRICO	16
1.1 GENERALIDADES	16
1.1.1 TEMA: “SISTEMA INFORMATICO PARA EL MANEJO Y CONTROL DE INFORMACIÓN DE ACTIVOS FIJOS EN EL COLEGIO PARTICULAR PRADO MEDIANTE LA UTILIZACION DE TECNOLOGÍA MÓVIL.”	16
1.1.2 PROBLEMA.....	16
1.1.3 JUSTIFICACION.....	16
1.1.4 OBJETIVO GENERAL	17
1.1.5 OBJETIVOS ESPECIFICOS	17
1.1.6 ALCANCE	17
1.1.7 HIPOTESIS.....	19
1.2 DESCRIPCIÓN DEL COLEGIO PARTICULAR PRADO	19
1.2.1 ORGANIGRAMA ESTRUCTURAL.....	21
1.2.2 DESCRIPCION DEL DEPARTAMENTO ADMINISTRATIVO.....	22
1.2.2 DESCRIPCION DEL DEPARTAMENTO DE COORDINACION DE ACTIVIDADES	26
1.2.2.1 DESCRIPCION DEL PERSONAL DEL DEPARTAMENTO ADMINISTRATIVO ..	22
1.3 CARACTERISTICAS DE LOS ACTIVOS FIJOS	26
1.3.1 DEFINICION DE LOS ACTIVOS FIJOS.....	26
1.3.2 CATEGORIAS DE LOS ACTIVOS FIJOS	27
1.3.3 CARACTERISTICAS DE LOS ACTIVOS FIJOS	27
1.3.4 DEPRECIACION DE LOS ACTIVOS FIJOS	28
1.3.4.1 DEPRECIACION FISCAL	28
1.3.4.2 METODOS DE DEPRECIACION	31
CAPITULO 2.....	34
SUSTENTO INFORMÁTICO	34
2.1 PROGRAMACIÓN ORIENTADA A OBJETOS.....	34
2.1.1 INTRODUCCION	34
2.1.2 OBJETOS	35
2.1.2.1 CARACTERISTICAS DE OBJETOS	36
2.1.2.1.1 HERENCIA	36
2.1.2.1.2 POLIMORFISMO	36
2.1.2.1.3 ENCAPSULAMIENTO	36
2.1.2.1.4 MENSAJE	37

2.1.2.1.5 ABSTRACCION	37
2.1.2.1.6 GENERALIZACION/ESPECIALIZACION	37
2.2 METODOLOGIA OMT	38
2.2.1 INTRODUCCION A OMT	38
2.2.2 DEFINICION DE OMT	39
2.2.2.1 VENTAJAS	40
2.2.2.2 DESVENTAJAS	40
2.3 DIAGRAMAS UML	41
2.3.1 INTRODUCCION A UML	41
2.3.2 DEFINICION DE UML	41
2.3.2.1 UTILIDAD	42
2.3.2.2 NOTACION	42
2.3.2.2.1 DIAGRAMA DE CLASES	42
2.3.2.2.2 DIAGRAMA DE OBJETOS	43
2.3.2.2.3 DIAGRAMA DE INTERACCION	44
2.3.2.2.4 DIAGRAMA DE ESTADOS	46
2.3.2.2.5 DIAGRAMA DE ACTIVIDAD	46
2.3.2.2.6 DIAGRAMA DE CASOS DE USO	47
2.3.2.2.7 DIAGRAMA DE COMPONENTES	48
2.3.2.2.7 DIAGRAMA DE DESPLIEGUE O EJECUCION	48
CAPITULO 3.....	50
ANALISIS Y DISEÑO DEL SISTEMA	50
3.1 ESPECIFICACIONES Y REQUISITOS DEL SOFTWARE	50
3.1.1 INTRODUCCION	50
3.1.1.1 PROPOSITO.....	50
3.1.1.2 ÁMBITO DEL SISTEMA.....	50
3.1.1.3 DEFINICION, ACRONIMOS, SIGLAS Y ABREVIATURAS	51
3.1.1.4 REFERENCIAS	52
3.1.2 DESCRIPCION GENERAL	52
3.1.2.1 PERSPECTIVA DEL PRODUCTO.....	52
3.1.2.2 FUNCIONES DEL PRODUCTO.....	54
3.1.2.3 CARACTERISTICAS DE USUARIO	55
3.1.2.4 LIMITACIONES GENERALES	57
3.1.3 REQUERIMIENTOS ESPECIFICOS.....	58
3.1.3.1 INTERFAZ DE USUARIO	59
3.1.3.1 REQUISITOS FUNCIONALES.....	58

3.1.3.3 INTERFACES DE HARDWARE.....	60
3.1.3.4 INTERFACES DE SOFTWARE	60
3.1.3.5 INTERFACES DE COMUNICACION	60
3.1.3.6 REQUISITOS DE EJECUCION.....	60
3.1.3.6.1 REQUERIMIENTOS DE DISEÑO, DESARROLLO E IMPLEMENTACION.....	60
3.1.3.6.2 REQUERIMIENTOS PARA EL DESARROLLO	60
3.1.3.6.2.1 SOFTWARE.....	60
3.1.3.6.2.2 HARDWARE	61
3.1.3.7 RESTRICCIONES LEGALES Y LICENCIAS.....	61
3.1.3.8 CUMPLIMIENTOS DE ESTANDARES	61
3.2 MODELO DE OBJETOS.....	63
3.2.1 MODELO DE CLASES	63
3.3 MODELO DINÁMICO	64
3.3.1 DIAGRAMAS DE INTERACCION	64
3.3.1.1 DIAGRAMAS DE SECUENCIA - ADMINISTRADOR DE LA SEGURIDAD	64
3.3.1.10 DIAGRAMA DE COLABORACION - Calculador de la Depreciación	73
3.3.1.11 DIAGRAMA DE SECUENCIA - Gestor de Activos Fijos.....	74
3.3.1.12 DIAGRAMA DE COLABORACION - Gestor de Activos Fijos	75
3.3.1.13 DIAGRAMA DE SECUENCIA - Sincronizador de Datos	76
3.3.1.14 DIAGRAMA DE COLABORACION - Sincronizador de Datos.....	77
3.3.1.2 DIAGRAMAS DE COLABORACION - ADMINISTRADOR DE LA SEGURIDAD	65
3.3.1.3 DIAGRAMA DE SECUENCIA - ADMINISTRADOR DE USUARIOS	66
3.3.1.4 DIAGRAMA DE COLABORACION – Administrador de Usuarios.....	67
3.3.1.5 DIAGRAMA DE SECUENCIA – Administrador de Categorías	68
3.3.1.6 DIAGRAMA DE COLABORACION- Administrador de Categorías	69
3.3.1.7 DIAGRAMA DE SECUENCIA - Organizador de Activos Fijos	70
3.3.1.8 DIAGRAMA DE COLABORACION - Organizador de Activos Fijos	71
3.3.1.9 DIAGRAMA DE SECUENCIA - Calculador de la Depreciación	72
3.3.2 DIAGRAMAS DE ESTADO	78
3.3.2.1 DIAGRAMA DE ESTADOS DEL OBJETO: Usuario	78
3.3.2.2 DIAGRAMA DE ESTADOS DEL OBJETO: Categoría.....	79
3.3.2.3 DIAGRAMA DE ESTADOS DEL OBJETO: Activo	80
3.3.3 DIAGRAMAS DE ACTIVIDAD.....	81
3.3.3.1 DIAGRAMA DE ACTIVIDAD: Validador de la Seguridad	81
3.3.3.2 DIAGRAMA DE ACTIVIDAD: Administrador de usuario.....	82
3.3.3.3 DIAGRAMA DE ACTIVIDAD: Administrador de Activos Fijos.....	83

3.3.3.4 Diagrama de Actividad: Organizador de Activos Fijos	84
3.3.3.5 DIAGRAMA DE ACTIVIDAD: Calculador de la depreciación.....	85
3.4 MODELO FUNCIONAL.....	86
3.4.1 DIAGRAMAS DE CASOS DE USO.....	86
3.4.1.1 MODELO FUNCIONAL: Perspectiva del Producto.....	86
3.4.1.2 MODELO FUNCIONAL: Validador de la Seguridad	87
3.4.1.3 MODELO FUNCIONAL: Administrador de Usuario	88
3.4.1.4 MODELO FUNCIONAL: Administrador de Activos Fijos	89
3.4.1.5 MODELO FUNCIONAL: Organizador de Activos Fijos.....	90
3.4.1.6 MODELO FUNCIONAL: Calculador de la Depreciación.....	91
3.4.1.7 MODELO FUNCIONAL: Gestor de Activos Fijos (Aplicación PALM).....	92
3.4.1.8 MODELO FUNCIONAL: Sincronizador de Datos	93
CAPITULO 4.....	94
IMPLEMENTACION Y PRUEBAS	94
4.1 HERRAMIENTAS DE IMPLEMENTACION.....	94
4.1.1 PUNTONET FRAMEWORK.....	94
4.1.1.1 ARQUITECTURA DEL .NET FRAMEWORK.....	94
4.1.1.2 BENEFICIOS DEL FRAMEWORK	96
4.1.2 VISUAL STUDIO .NET.....	96
4.1.2.1 VISUAL BASIC .NET	96
4.1.2.2 REFLEXIONES EN .NET	97
4.1.3 SQL SERVER 2005	97
4.1.3.1 ¿QUE ES SQL SERVER 2005?.....	97
4.1.3.2 PLATAFORMA DE DATOS DE SQL SERVER	99
4.1.4 CODIGO FUENTE	102
4.1.4.1 CODIGO DE UNA CLASE IMPLEMENTADA.....	102
4.1.4.2 SCRIPT DE CREACION DE LA BASE DE DATOS.....	104
4.2 PRUEBAS.....	116
4.2.1 PRUEBAS DE INTEGRACION	116
4.2.1.1 MODULO DE ACCESO AL SISTEMA CELLARACTIVE	116
4.2.1.2 MODULO ADMINISTRADOR DE ACTIVOS FIJOS	117
4.2.1.3 MODULO ORGANIZADOR DE ACTIVOS FIJOS	119
4.2.3.1.4 MODULO ADMINISTRADOR DE USUARIO.....	121
CAPITULO 5.....	155
CONCLUSIONES Y RECOMENDACIONES	155
5.1 CONCLUSIONES	155

5.2 RECOMENDACIONES.....	156
ANEXO 1	159
DESCRIPCIÓN DEL PROBLEMA	15
MANUAL DE CONFIGURACION DEL SISTEMA	123
MANUAL DE INSTALACIÓN DEL SISTEMA CELLARACTIVE	126
MANUAL DE USUARIO.....	131
REFERENCIAS BIBLIOGRAFICAS.....	158

INDICE DE FIGURAS Y DIAGRAMAS

Diagrama 3.1: Clases	63
Diagrama 3.10: Calculador de la Depreciación	72
Diagrama 3.11: Calculador de la Depreciación	73
Diagrama 3.12: Gestor de Activos Fijos	74
Diagrama 3.13: Gestor de Activos Fijos	75
Diagrama 3.14: Sincronizador de Datos.....	76
Diagrama 3.15: Sincronizador de Datos.....	77
Diagrama 3.16: Usuario	78
Diagrama 3.17: Categoría.....	79
Diagrama 3.18: Activo.....	80
Diagrama 3.19: Validador de la Seguridad.....	81
Diagrama 3.2: Administrador de la seguridad	64
Diagrama 3.20: Administrador de Usuario	82
Diagrama 3.21: Administrador de Activos Fijos.....	83
Diagrama 3.22: Organizador de Activos Fijos	84
Diagrama 3.23: Calculador de la Depreciación	85
Diagrama 3.24: Perspectiva del Producto	86
Diagrama 3.25: Validador de la Seguridad.....	87
Diagrama 3.26: Administrador de Usuario	88
Diagrama 3.27: Administrador de Activos Fijos.....	89
Diagrama 3.28: Organizador de Activos Fijos	90
Diagrama 3.29: Calculador de la Depreciación	91
Diagrama 3.3: Administrador de la seguridad	65
Diagrama 3.30: Gestor de Activos Fijos	92
Diagrama 3.31: Sincronizador de Datos	93
Diagrama 3.4: Administrador de Usuarios.....	66
Diagrama 3.5: Administrador de Usuarios.....	67
Diagrama 3.6: Administrador de Categorías	68
Diagrama 3.7: Administrador de Categorías	69
Diagrama 3.8: Organizador de Activos Fijos	70
Diagrama 3.9: Organizador de Activos Fijos	71
Fig. 3.1 PERSPECTIVA DEL PRODUCTO.....	53
Figura 1.1 Descripción del problema.....	15
Figura 1.2 Organigrama del Colegio Particular Prado	21

Figura 2.1 Análisis y Diseño Orientado a Objetos	38
Figura 2.10 Ejemplo de Diagrama de Componentes.....	48
Figura 2.11 Ejemplo de Diagrama de Ejecución	49
Figura 2.2 Metodología de Diseño	39
Figura 2.3 Representación de Clase en UML	43
Figura 2.4 Diagrama de Clases y Objetos.....	44
Figura 2.5 Secuencias de Mensajes y Flujos de Ejecución	45
Figura 2.6 Ejemplo de Diagrama de Secuencia	45
Figura 2.7 Ejemplo de Diagrama de Estado.....	46
Figura 2.8 Ejemplo de Diagrama de Actividad	47
Figura 2.9 Ejemplo de Diagrama de Casos de Uso	47
Figura 4.1: Arquitectura del Framework	95
Figura 4.2: Arquitectura del Framework	99

INDICE DE TABLAS

Tabla 1: Número de trabajadores del Colegio Particular Prado.....	19
Tabla 2: Categorías de los Activos Fijos	27
Tabla 3: Beneficios y Capacidades del Sistemas.....	51
Tabla 4: Roles y Privilegios de Usuario.....	57
Tabla 5: Casos de prueba Modulo de acceso al sistema	116
Tabla 6: Casos válidos Modulo de acceso al sistema	117
Tabla 7: Casos no válidos Modulo de acceso al sistema	117
Tabla 8: Casos de prueba Modulo administrador de activos	118
Tabla 9: Casos válidos Modulo administrador de activos.....	118
Tabla 10: Casos no válidos Modulo administrador de activos	119
Tabla 11: Casos de prueba Modulo organizador de activos.....	120
Tabla 12: Casos válidos Modulo organizador de activos.....	121
Tabla 13: Casos no válidos Modulo organizador de activos.....	121
Tabla 14: Casos de prueba Modulo administrador de usuario	122
Tabla 15: Casos válidos Modulo administrador de usuario	122
Tabla 16: Casos no válidos Modulo administrador de usuario	123

RESUMEN

Esta tesis está enfocada a controlar y supervisar todos los activos fijos que existen en el Colegio Particular Prado, utilizando equipos inalámbricos de fácil interacción. Con CellarActive se podrá controlar todos los movimientos de los activos fijos e ingresos por concepto de compras y supervisar el estado de los mismos. Finalmente, los procesos que el sistema CellarActive realiza se sujetan a una integración de datos que brinda la flexibilidad necesaria a los usuarios para el uso de la información.

PRESENTACIÓN

Este documento está dirigido a personas que estén involucradas en el control de activos fijos que deseen sacar información de sus movimientos en forma rápida y eficiente.

Muestra un enfoque de los procesos de adquisiciones que se realizan en una institución. Veremos las tecnologías utilizadas para registrar activos fijos en la institución educativa y los beneficios y desventajas de cada una de ellas.

En el primer capítulo presentamos un estudio del problema, los objetivos, una breve reseña sobre la institución y las características de cada activo fijo.

En el segundo capítulo nos centramos en todos los fundamentos tecnológicos para lograr un producto informático que pueda supervisar y controlar activos fijos a través de la información recopilada en el estudio del problema. Y conocer las características principales de las herramientas y/o servicios necesarios para obtener dicho producto.

En el capítulo de análisis y diseño presentamos la perspectiva del producto, las funciones que realizará el sistema, las interacciones que tendrá el usuario sobre el funcionamiento lógico del sistema de control de activos fijos. También detallamos las limitaciones, la fiabilidad, la confiabilidad, y los requisitos funcionales para poder poner en marcha el software obtenido.

En el cuarto capítulo se desarrolló pruebas de software basadas en una técnica para la comunicación entre el usuario final y el sistema CellarActive, usando los principales módulos que incorpora el software, para ello se realizó una integración entre dichos módulos.

El quinto capítulo detallamos las conclusiones y recomendaciones sobre el sistema CellarActive.

CAPITULO 1.

SUSTENTO TEÓRICO

1.1 GENERALIDADES

1.1.1 TEMA: “SISTEMA INFORMATICO PARA EL MANEJO Y CONTROL DE INFORMACIÓN DE ACTIVOS FIJOS EN EL COLEGIO PARTICULAR PRADO MEDIANTE LA UTILIZACION DE TECNOLOGÍA MÓVIL.”

1.1.2 PROBLEMA

En el Colegio Particular Prado en el departamento administrativo se observa que existe una falta de control de los Activos Fijos, ya que se desconoce de la ubicación real del bien.

No existe un historial exacto de los Activos Fijos durante su vida útil, ocasionando que se produzcan asientos contables errados por mala depreciación o revalorización. Creando un problema incremental y acumulativo.

Año a año la generación de los ajustes (Contables/depreciación/ubicación) se lo realiza manualmente ocasionando importantes pérdidas para la institución se le pide que cada inicio de año lectivo realice un reporte de los movimientos que han sufrido los activos, el mismo que está basado en el archivo de comprobantes emitidos para la ubicación de nuevos activos fijos y su reubicación, así que el tiempo invertido para la entrega del reporte es aproximadamente de 15 días, y la información contenida en este no es confiable por la pérdida de datos.

Se puede concluir que el principal problema, es que la información no es exacta y oportuna lo que ocasiona falta de control y aceptación de la información.

DESCRIPCIÓN DEL PROBLEMA

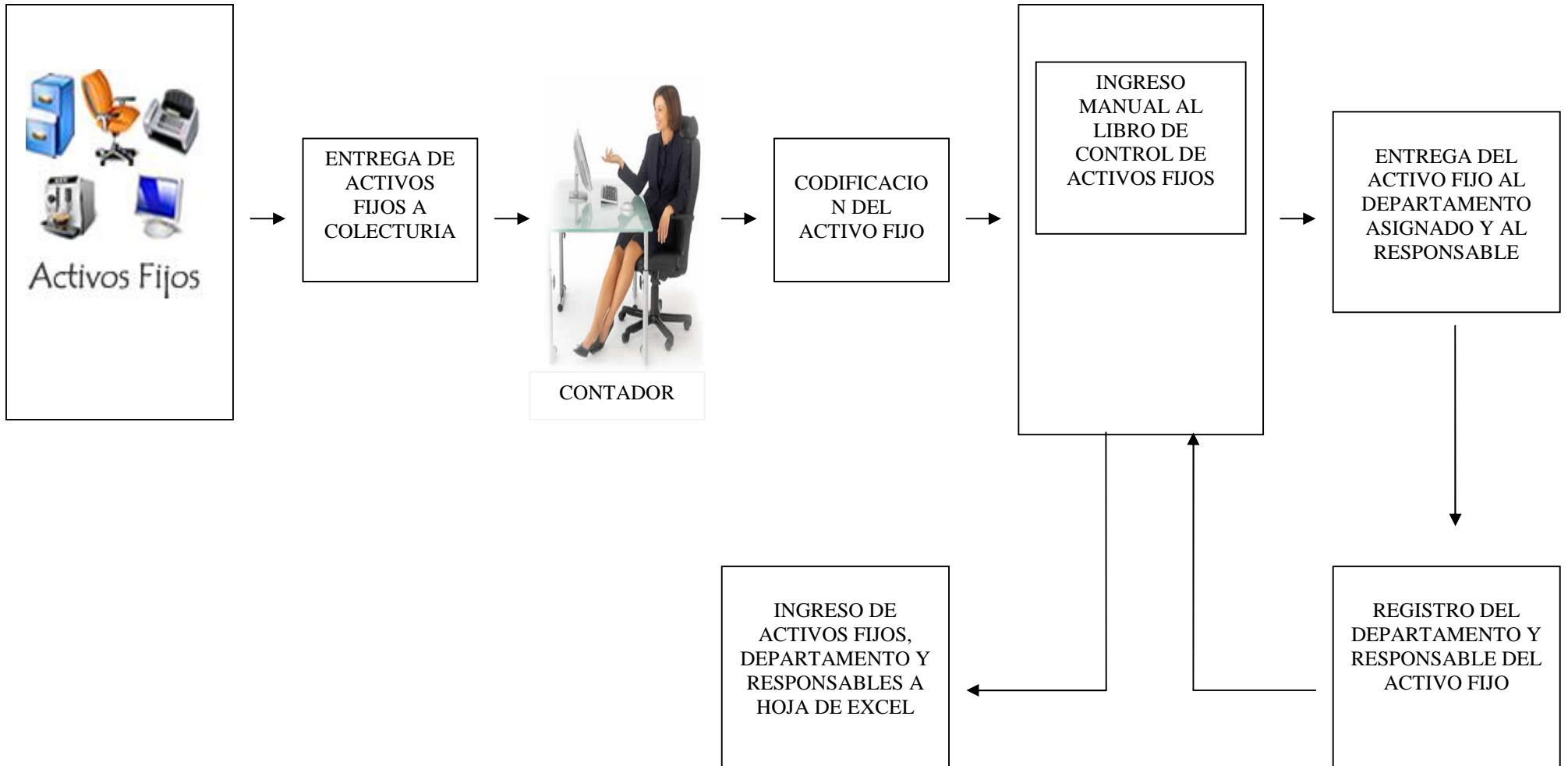


Figura 1.1 Descripción del problema

1.1.3 JUSTIFICACION

La importancia en el Manejo y Control de los Activos Fijos.

Los activos fijos tienen un peso determinante en el patrimonio de las instituciones y en la generación de ingresos; su manejo y correcta administración impactan sobre las decisiones que tome la Gerencia/Dirección.

Una administración efectiva de los activos fijos, además de garantizar un control y seguridad de los bienes, responde interrogantes tales como: gastos de depreciación técnica y contable; rendimiento de las inversiones; riesgo y coberturas de pólizas de seguro; reexpresión de estados financieros; financiamiento, garantías, incorporación y desincorporación de activos, entre otras.

Manejo y Control de los Activos Fijos a través de la Tecnología Móvil.

El departamento de Coordinación de actividades, es el encargado de los movimientos físicos de los activos fijos en la organización de la institución. Realizando actividades de control de nuevos ingresos de activos fijos, ubicación y bajas del activo que actualmente se lo realiza en registros manuales a través de formularios emitidos por el departamento de Administración. Al crear una nueva política en el manejo de estos procesos con la utilización de Tecnología Móvil en una Red Inalámbrica, a través de una interfase de fácil uso permitirá optimizar recursos como:

- Tiempo
- Recurso Humano
- Dinero

Por lo mencionado anteriormente el sistema es importante porque brindará un mejor control y manejo de la información produciendo reportes en tiempo real y acorde a las necesidades del usuario y de la institución, permitiendo una mejor toma de decisiones al momento de hacer una nueva adquisición de activos.

1.1.4 OBJETIVO GENERAL

Elaborar un Sistema capaz de facilitar Control y el Manejo de los distintos Activos Fijos.

1.1.5 OBJETIVOS ESPECIFICOS

- Análisis del sistema mediante la recolección de las necesidades de la institución y las perspectivas que tiene el cliente en base al sistema.
- Diseñar el sistema mediante la metodología OMT, aplicando el lenguaje UML.
- Realizar la implementación del sistema mediante el diagrama de componentes y de despliegue, los mismos que nos permitirán realizar la puesta en funcionamiento del sistema.
- Organizar el sistema de Control y Manejo de Activos fijos mediante una interfaz amigable, que posea las funcionalidades requeridas tales como la comunicación Wireless de los dispositivos móviles (Utilización de PALMS), fácil acceso y confiabilidad de los datos.
- Sistematizar mediante un Sistema Integral en el control de activos fijos y el seguimiento contable de Activos fijos, la información relevante para una adecuada administración de los mismos.
- Crear procedimientos adecuados, que permitan un apropiado control de la información, de tal forma que los Funcionarios Administrativos, Usuarios Externos se rijan a las normas y estándares establecidos por la Institución.
- Garantizar, la seguridad, integridad, respaldo, disponibilidad y confiabilidad de los datos, y de esa manera brindar un mejor servicio mediante la implementación de políticas y procedimientos que se ajusten adecuadamente a los requerimientos de los usuarios.

1.1.6 ALCANCE

De esta manera, se plantea una solución informática aplicando un sistema de manejo y control de información de activos fijos que permita, realizar un seguimiento de los movimientos de cada uno de los activos fijos de la institución, para aplicar un control sobre la adquisición de un nuevo activo, ubicación,

depreciación, revalorización, y baja del activo. Por último generar los reportes necesarios en tiempo real.

El manejo deberá ser amigable con los usuarios, contará con un manejo de perfil de usuarios. Así el sistema contará con los siguientes módulos:

MODULO 1

Registro de ingreso de Datos Activos Fijos

Este módulo se instalará en los dispositivos móviles (permitiendo), al usuario dirigirse al lugar donde se encuentran los activos fijos, permitiendo movilidad, al usuario. Aquí se registrará, la categoría del activo, tipo del activo, valor, medio por el cuál fue adquirido, y finalmente la persona responsable del cuidado del Activo Fijo.

MODULO 2

Control y Manejo de los Activos Fijos

A través de este módulo se podrá realizar un adecuado seguimiento de los diferentes estados que va adquiriendo el activo fijo, es necesario que se registren los traspasos (movilidad del activo) que sufre el activo fijo, las depreciaciones y revalorizaciones. Así como la necesidad de dar de baja un activo fijo.

MODULO 3

Gestor de la Información

El Gestor de la Información se encargará de realizar los diferentes reportes que permitirán ver el movimiento diario, mensual semestral o anual del Activo Fijo, debiendo tener varios tipos de reportes/informes como:

- Catálogo de Activos Fijos
- Clasificación de Activos Fijos por su Ubicación
- Precisión de los cálculos contables de depreciación de activos fijos
- Historial del Activos Fijos
- Existencias de Activos
- Valor Actual de Activos

1.1.7 HIPOTESIS

¿Es posible desarrollar un sistema de información para el manejo y control de los activos fijos?

¿Es posible que mediante la utilización de este sistema la dirección de la institución y el departamento administrativo puedan tener informes consistentes, integrales y a tiempo?

¿Es posible reemplazar el manejo manual en el desarrollo de las actividades para ingresos, transporte y desincorporar los activos fijos, por un sistema con tecnología móvil?

1.2 DESCRIPCIÓN DEL COLEGIO PARTICULAR PRADO

La Escuela Particular Prado comienza a funcionar el 5 de Agosto de 1996 está ubicado en Tumbaco en las calles Abdón Calderón 725 y Gonzalo Díaz de Pineda, bajo los principios de la educación laica, al servicio de la niñez estudiosa. Con niños creativos y críticos que utilizan la investigación, los talleres interdisciplinarios como estrategias de comunicación y aprendizaje.

	Número de trabajadores (promedio)	
	2007	2008
Condición		
Funcionarios	8	13
Profesores	10	20

Tabla 1: Número de trabajadores del Colegio Particular Prado ¹

a. Visión

“Somos una institución educativa con un personal administrativo y docente altamente calificado, una infraestructura amplia y acorde con los actuales requerimientos pedagógicos, cuyo propósito es dar a niños/as una educación de calidad desde la óptica: holística, procesal y sistemática con una metodología

¹ **Fuente:** Colegio Particular Prado

basada en el trabajo cooperativo, de modo que los alumnos lleguen a ser personas creativas, críticas y solidarias, al servicio de la sociedad”.

b. Misión

“La Unidad Educativa Particular Prado forma talentos con liderazgo, autonomía y criterio humanista, valores en las áreas científica, cultural y deportiva, con capacidad para acceder al conocimiento a través de la investigación y el uso adecuado de la tecnología, comprometidos con el ambiente y la sociedad”.

c. Objetivos Institucionales

Objetivo General: Ofertar servicios educativos de calidad en los niveles de Educación Básica, mediante la implementación de proyectos experimentales y de intervención; ejecución de procesos pedagógicos y administrativos orientados por el modelo pedagógico constructivista; que promuevan la investigación, el trabajo en equipo, la práctica de valores y propicien el desarrollo de capacidades científicas, culturales y deportivas.

Objetivos Específicos:

- Ampliar la cobertura educativa de la institución a Unidad Educativa, mediante la creación del nivel propedéutico, primero y segundo de bachillerato; construcción de ambientes y espacios pedagógicos adecuados para asegurar la formación integral de los y las estudiantes.
- Consolidar el Modelo Pedagógico Constructivista mediante el reajuste y sistematización anual y la concreción de procesos didácticos de aula para asegurar un aprendizaje funcional.
- Generar procesos de crecimiento personal y desarrollo profesional de docentes, administrativos, médico –odontológico y de apoyo mediante la formación en desempeño en temáticas afines a su área de trabajo para asegurar el mejoramiento continuo.
- Implantar propuestas educativas innovadoras, mediante el diseño, ejecución y monitoreo de proyectos educativos experimentales y de intervención o desarrollo, que propicien la solución de problemas educativos y que generen prácticas educativas permanentes acordes a las necesidades institucionales.

1.2.1 ORGANIGRAMA ESTRUCTURAL

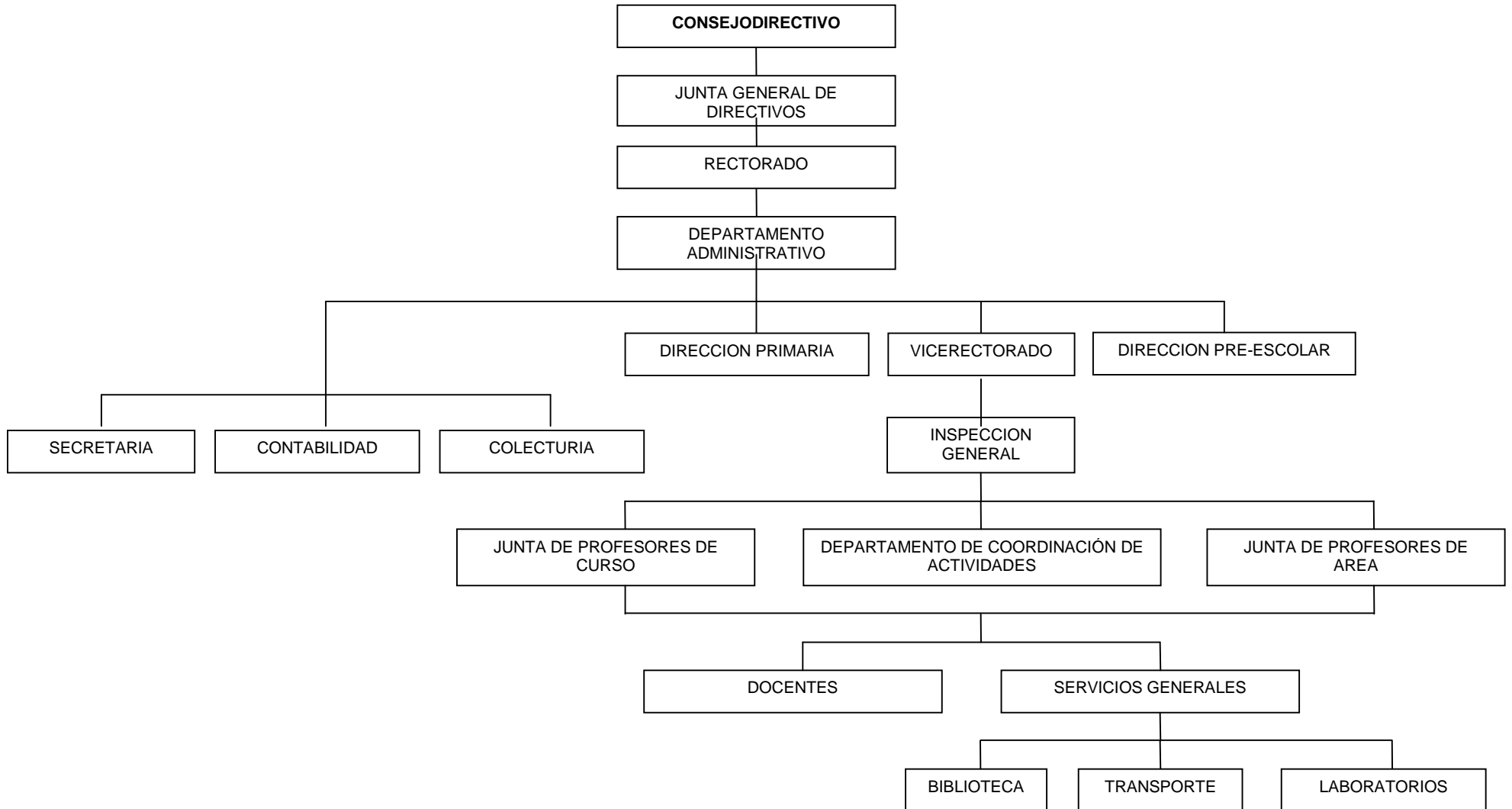


Figura 1.2 Organigrama del Colegio Particular Prado ²

² Fuente: Colegio Particular Prado

1.2.2 DESCRIPCION DEL DEPARTAMENTO ADMINISTRATIVO

1.2.2.1 DESCRIPCION DEL PERSONAL DEL DEPARTAMENTO ADMINISTRATIVO

Ver Anexo 1

- Gerente Financiera: La Gerente Financiera del Colegio Particular Prado es la Sra. Rosmery Gálvez.
- Secretaria: La Sra. Cecilia Coba encargada de contestar el teléfono y la encargada de llevar apuntes en las reuniones que realizan en todo el departamento, como también de recoger los informes de cada departamento lo que han hecho durante el mes.
- Colecturía: La Sra. Elena Rubio encargada de recibir por parte de las empresas ganadoras de las licitaciones uniformes, libros y útiles escolares.
- Contadora: la Sra. Lucia Arellano se encarga de controlar los asientos contables de la institución.

TAREAS Y FUNCIONES

Área: Departamento Administrativo y Financiero

TÍTULO DEL CARGO: GERENTE FINANCIERO

Descripción General

El ocupante del cargo deberá planificar, organizar, dirigir y controlar los procesos, proyectos, programas y acciones financieras encaminadas a la obtención de resultados positivos para la institución.

El Gerente Financiero deberá tomar decisiones financieras y efectuar el análisis, planeación, toma de decisiones sobre inversiones y financiamiento a corto y largo plazo, realizar el análisis de los pronósticos financieros y preparar los planes y presupuestos financieros de la empresa.

Además lleva una verificación rigurosa de los activos fijos que entran a la institución cada año, estos activos están en una hoja de Excel el cual imposibilita

hacer consultas de cada activo fijo que posee la institución, la depreciación de cada uno de ellos y los traspasos que se han realizado durante el año lectivo.

Tareas Principales:

- Efectúa el análisis y la planeación financiera.
- Toma decisiones de inversión.
- Toma decisiones de financiamiento.
- Transformar la información financiera a una forma útil para supervisar la condición financiera de la institución.
- Control de adquisición de Activos Fijos.
- Manejo de depreciaciones de Activos Fijos.

Tareas Secundarias

- Administra la política de crédito de la empresa, por medio de los informes que se le presenten.
- Administrar con responsabilidad la cartera de inversión.
- Maneja evaluaciones de rendimiento financiero.

Tareas Ocasionales

- Negocia presupuestos operativos.
- Arregla el financiamiento para las inversiones de activos aprobados.
- Coordina a los consultores como a los banqueros de inversiones y a los asesores legales.

TÍTULO DEL CARGO: CONTADOR

Administra un sistema contable óptimo para la entidad educativa.

Organizar la contabilidad de la institución con las normas establecidas que se ajusten a las políticas, principios y normas generalmente aceptadas.

Tareas Principales

- Desarrollar la contabilidad de la institución según las políticas, normas y procedimientos contables.
- Presentar balances mensuales.
- Realizar las transacciones oportunas de tal forma que haya una fluidez en las cuentas bancarias.
- Realizar los pagos mensuales y beneficios de nómina de la institución.
- Estudiar los valores que constituyen capital o que intervendrán en las operaciones diarias.
- Determinar las cuentas de la empresa y su significado; esto es, elaborar el Plan de Cuentas y preparar el Manual de Instrucciones.
- Fijar el sistema adecuado para calcular los precios de costos, reposición y venta, según corresponda.
- Indicar el número de libros, formularios y registros que deben llevarse, las informaciones que deben contener y su utilización.
- Estudiar e interpretar los resultados obtenidos.
- Depuración de cuentas.
- Pago de impuestos y tributos.

Tareas Secundarias

- Presentar información oportunamente cuando el Rector o Gerente Financiero lo soliciten.
- Realizar conciliaciones bancarias.

Tareas Ocasionales

- Establecer calendarios de pagos tanto a proveedores y empleados.
- El contador registra cada una de las transacciones.
- Manejo de Kardex en los departamentos que así lo requieran.

- Recibir y revisar facturas, órdenes de pago, órdenes de compra, ingresos y egresos de la organización.

TÍTULO DEL CARGO: SECRETARIA

Es la encargada de contestar el teléfono, como también la que lleva apuntes en las reuniones que realiza el departamento, como también recoger los informes de labores de su departamento.

Tareas Principales

- Realizar el informe estadístico anual de movimiento educativo del plantel y presentar al rector.
- Mantener debidamente encuadernados todos los documentos de archivo, clasificados por años y materiales.

TÍTULO DEL CARGO: COLECTORA

Tareas Principales

- Es la que se encarga de recibir en buen estado todos los materiales que el colegio necesite a través de las empresas ganadoras.
- Recaudar oportunamente, los ingresos propios del establecimiento, elaborar los partes diarios de recaudación y remitirlos a contabilidad.
- Depositar en forma inmediata e intacta los recursos financieros recaudados.
- Solicitar la autorización correspondiente para procesar a la emisión de especies valoradas al honorable Consejo Directivo.
- Custodiar y vender las especies valoradas.
- Solicitar y participar en bajas.
- Ejercer el control previo al desembolso, verificando que la documentación se encuentre completa y debidamente legalizada.

1.2.2 DESCRIPCION DEL DEPARTAMENTO DE COORDINACION DE ACTIVIDADES

- Presentar el plan anual de las actividades a las autoridades.
- Coordinar con las autoridades las acciones para la formación de clubes y Cooperativas estudiantiles.
- Fomentar las buenas relaciones sociales entre todo el personal del plantel.
- Responsabilizarse de las actividades, sociales que se proyecten durante el año lectivo, en las fechas de advenimiento.
- Elaborar la programación de fiestas patronales y poner en consideración del Consejo Directivo.
- Establecer las fuentes de financiamiento de las fiestas.
- Responsabilizarse de la ejecución de las fiestas patronales en coordinación con los diferentes estamentos de la institución.
- Presentar el informe económico y de las actividades de las fiestas patronales.
- Organizar programas sociales y culturales con los estudiantes y docentes del plantel.

1.3 CARACTERISTICAS DE LOS ACTIVOS FIJOS

1.3.1 DEFINICION DE LOS ACTIVOS FIJOS

El activo fijo está formado por bienes tangibles que han sido adquiridos o construidos para usarlos en el giro de la empresa, representan al conjunto de servicios que se recibirán en el futuro a lo largo de la vida útil de un bien adquirido³.

Los activos fijos se valorizan, en principio, al costo, de adquisición o construcción de los respectivos bienes. Dicho costo debe actualizarse posteriormente para reflejar los efectos de la inflación o depreciación.

³ **Chong, Esteban.** *Contabilidad Intermedia – Tomo I: Estados financieros y cuentas del activo; Universidad del Pacífico, 1992.*

1.3.2 CATEGORIAS DE LOS ACTIVOS FIJOS

TIPO DE ACTIVO FIJO	DEPRECIACION ANUAL	VIDA UTIL (EN AÑOS)
Inmuebles, (excepto terrenos), naves, aeronaves, barcazas y similares	5%	20
Instalaciones, maquinarias, equipos y muebles	10%	10
Vehículos, equipos de transporte y equipo caminero móvil	20%	5
Equipos de cómputo y software	33%	3

*Tabla 2: Categorías de los Activos Fijos*⁴

1.3.3 CARACTERISTICAS DE LOS ACTIVOS FIJOS

- El Activo Fijo debe ser físicamente tangible.
- Tener una vida útil relativamente larga (por lo menos mayor a un año o a un ciclo normal de operaciones, el que sea mayor).
- Sus beneficios deben extenderse, por lo menos, más de un año o un ciclo normal de operaciones, el que sea mayor. En este sentido, el activo fijo se distingue de otros activos (útiles de escritorio, por ejemplo) que son consumidos dentro del año o ciclo operativo de la empresa.
- Ser utilizado en la producción o comercialización de bienes y servicios, para ser alquilado a terceros, o para fines administrativos. En otras palabras, el bien existe con la intención de ser usado en las operaciones de la empresa de manera continua y no para ser destinado a la venta en el curso normal del negocio.

⁴ **Fuente:** Contabilidad General, Mercedes Bravo Valdivieso

1.3.4 DEPRECIACION DE LOS ACTIVOS FIJOS

Las depreciaciones se realizan con el fin de cargar a gastos el costo del activo, y que este valor será reembolsable en un futuro próximo, ya sea para la adquisición de otro activo, o el mejoramiento de él, dependiendo de las decisiones gerenciales de la empresa.

Así mismo tiene las siguientes finalidades:

- Lograr que cada ejercicio económico venga gravado por el total de gastos que le corresponden y han contribuido a formar el producto de éste.
- Valorar el desgaste anual de los elementos que forman su Activo Fijo.
- Constituir una reserva para reponer el valor inicial de los elementos que forman el Activo Fijo.

Para realizar una acertada depreciación se debe considerar los siguientes criterios: Determinar el costo inicial del elemento a renovar.

- Señalar la vida probable útil del elemento a depreciar o la producción prevista.
- Considerar el posible envejecimiento prematuro debido a la evolución de la técnica. A esta posibilidad se le da el nombre técnico de obsolescencia, que significa precisamente vejez prematura.
- Estimación del probable costo de sustitución del elemento amortizado por otro, cuando llegue la ocasión de realizarla.
- Valor residual o de desecho del elemento amortizado.
- Determinar el sistema de depreciación que se va a seguir.

1.3.4.1 DEPRECIACION FISCAL⁵

- Las depreciaciones de los activos fijos del respectivo negocio, en base de los siguientes porcentajes máximo de acuerdo al reglamento dictado por el SRI⁶.

⁵ Servicio de Rentas Internas

⁶ SRI: Servicio de Rentas Internas

- Inmuebles (excepto terrenos), naves, aeronaves, barcas y similares 5% anual.
 - Instalaciones, maquinarias, equipos y muebles 10% anual.
 - Vehículos, equipos de transporte y equipo caminero móvil 20% anual.
 - Equipos de cómputo y software 33% anual;
- Cuando el contribuyente haya adquirido repuestos destinados exclusivamente al mantenimiento de un activo fijo podrá, a su criterio, cargar directamente al gasto el valor de cada repuesto utilizado o depreciar todos los repuestos adquiridos, al margen de su utilización efectiva, en función a la vida útil restante del activo fijo para el cual están destinados, pero nunca en menos de cinco años. Si el contribuyente vendiere tales repuestos, se registrará como ingreso gravable el valor de la venta y, como costo, el valor que faltare por depreciar. Una vez adoptado un sistema, el contribuyente solo podrá cambiarlo con la autorización previa del respectivo Director Regional del Servicio de Rentas Internas;
 - En casos de obsolescencia, utilización intensiva, deterioro acelerado u otras razones debidamente justificadas, el respectivo Director Regional del Servicio de Rentas Internas podrá autorizar depreciaciones en porcentajes anuales mayores a los indicados, los que serán fijados en la resolución que dictará para el efecto. Para ello, tendrá en cuenta las Normas Ecuatorianas de Contabilidad y los parámetros técnicos de cada industria y del respectivo bien. Podrá considerarse la depreciación acelerada exclusivamente en el caso de bienes nuevos, y con una vida útil de al menos cinco años, por tanto, no procederá para el caso de bienes usados adquiridos por el contribuyente. Tampoco procederá depreciación acelerada en el caso de bienes que hayan ingresado al país bajo regímenes suspensivos de tributos, ni en aquellos activos utilizados por las empresas de construcción que apliquen para efectos de sus registros contables y declaración del impuesto el sistema de "obra terminada", previsto

en el Art. 29 de la Ley de Régimen Tributario Interno⁷. Mediante este régimen, la depreciación no podrá exceder del doble de los porcentajes señalados en letra a);

- Cuando se compre un bien que haya estado en uso, el adquirente puede calcular razonablemente el resto de vida útil probable para depreciar el costo de adquisición. La vida útil así calculada, sumada a la transcurrida durante el uso de anteriores propietarios, no puede ser inferior a la contemplada para bienes nuevos;
- Cuando el capital suscrito en una sociedad sea pagado en especie, los bienes aportados deberán ser valorados según los términos establecidos en la Ley de Compañías o la Ley de Instituciones del Sistema Financiero. El aportante y quienes figuren como socios o accionistas de la sociedad al momento en que se realice dicho aporte, así como los indicados peritos, responderán por cualquier perjuicio que sufra el Fisco por una valoración que sobrepase el valor que tuvo el bien aportado en el mercado al momento de dicha aportación. Igual procedimiento se aplicará en el caso de fusiones o escisiones que impliquen la transferencia de bienes de una sociedad a otra: en estos casos, responderán los indicados peritos evaluadores y los socios o accionistas de las sociedades fusionadas, escindidas y resultantes de la escisión que hubieren aprobado los respectivos balances. Si la valoración fuese mayor que el valor residual en libros, ese mayor valor será registrado como ingreso gravable de la empresa de la cual se escinde; y será objeto de depreciación en la empresa resultante de la escisión. En el caso de fusión, el mayor valor no constituirá ingreso gravable pero tampoco será objeto de depreciación en la empresa resultante de la fusión;
- Cuando un contribuyente haya procedido al revalúo de activos fijos, podrá continuar depreciando únicamente el valor residual. Si se asigna un nuevo valor a activos completamente depreciados, no se podrá volverlos a depreciar. En el caso de venta de bienes revaluados se considerará como

⁷ Decreto Ejecutivo No. 875, publicado en Registro Oficial 182 de 2 de Octubre del 2003.

ingreso gravable la diferencia entre el precio de venta y el valor residual sin considerar el revalúo;

- Los bienes ingresados al país bajo régimen de internación temporal, sean de propiedad y formen parte de los activos fijos del contribuyente y que no sean arrendados desde el exterior, están sometidos a las normas de los incisos precedentes y la depreciación será deducible, siempre que se haya efectuado el pago del impuesto al valor agregado que corresponda.

1.3.4.2 METODOS DE DEPRECIACION

Según NEC⁸ activos depreciables son los que:

- Se espera que sean usados durante más de un período contable.
- Tienen una vida útil limitada.
- Los posee una empresa para usarlos en la producción o prestación de bienes y servicios, para arrendarlos a otros o para fines administrativos.

Para depreciar los activos fijos se tiene que conocer el significado de los términos que se detallan a continuación:

- **Costo del activo fijo.-** es el valor del bien más todo lo que implica que este bien funcione. Ejemplo: transporte, seguro, instalación, capacitación de personal para su manejo.
- **Vida Útil:** Es la estimación que se da como tiempo de vida de un activo fijo. Al término de este tiempo el activo ya no presta servicio y hay que darle de baja por inservible.
- **Valor Residual:** Es el valor que se recupera después de la vida útil del activo. También se le llama VALOR DE DESECHO. Se puede decir también que es el valor en el que se pueden vender el activo cuando es desechado.

⁸ Norma Ecuatoriana de Contabilidad

La fórmula para el cálculo del valor residual⁹ es:

$$VR = \frac{VOriginal}{N}$$

Donde:

VR: Valor Residual

V.original: Valor Original

N: Número de años de vida útil

a. Método de línea recta¹⁰ (Depreciación constante)

Resulta de un cargo constante sobre la vida útil del activo. Este método ha devengado en el mes usado por su facilidad de aplicación. Su fórmula de cálculo es la siguiente:

$$DepreciaciónAnual = \frac{CostoDeAdquisiciónDelActivo - ValorResidualDelActivo}{VidaUtilDelActivo(EstimadaEnAños)}$$

b. Método de la Suma de los Dígitos

Mediante este método se estima una depreciación más rápida en su momento, para luego ir ascendiendo o descendiendo en intensidad. Se aplica especialmente en vehículos y maquinaria industrial.

El procedimiento para el cálculo es el siguiente:

- Se asigna a cada período de vida un dígito a contar desde uno.
- Se suman los dígitos.
- Se establece para cada período un quebrado, cuyo denominador precisamente será la suma de los dígitos y como numerador el dígito del año correspondiente.
- Este quebrado se multiplica por aquella diferencia entre (Valor del Activo menos el Valor Residual).

⁹ Fuente: Chong, Esteban, Contabilidad Intermedia

¹⁰ Fuente: Chong, Esteban, Contabilidad Intermedia

c. Método legal¹¹

Establece los límites máximos aceptados por el SRI como gasto deducible para el pago del impuesto a la renta. Se toma en cuenta los coeficientes anuales que guardan conformidad con las normas establecidas en el Reglamento de la Ley de Régimen Tributario.

Fórmula:

$$\text{DepreciaciónAnual} = (\text{ValorActual} - \text{Valor Residual})\%$$

d. Método por Unidades de Producción¹²

Como su nombre lo indica, el valor de la depreciación del activo fijo dependerá del número de kilómetros, o el número de unidades que se produzca a través de un activo fijo.

Fórmula:

$$\text{DepreciaciónPorUnidad} = \frac{\text{ValorActual} - \text{Valor Residual}}{\text{TotalDeUnidades}}$$

¹¹ **Fuente:** Chong, Esteban, Contabilidad Intermedia

¹² **Fuente:** Chong, Esteban, Contabilidad Intermedia

CAPITULO 2.

SUSTENTO INFORMÁTICO

2.1 PROGRAMACIÓN ORIENTADA A OBJETOS

2.1.1 INTRODUCCION

La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software

La Programación Orientada a Objetos permite a los programadores escribir software, de forma que esté organizado de la misma manera que el problema que trata de modelar.

La Programación Orientada a Objetos está organizada alrededor de "objetos" más que de "acciones", datos más que lógica. Históricamente, ha sido visto como un procedimiento lógico que toma los datos de entrada, los procesa y produce los datos de salida. El reto de la programación se centraba en escribir la lógica, y no en cómo definir los datos. La Programación Orientada a Objetos privilegia los objetos que se quiere manipular más que la lógica requerida para manipularlos.

Ventajas

La programación orientada a objetos pretende reflejar el mundo real, que se compone de objetos con los que (en general) sólo se puede realizar aquellas operaciones (métodos) previstos por los diseñadores del objeto. La programación orientada a objetos facilita la modularización del código, puesto que fuerza a que cada clase esté en ficheros (de cabecera y de código) diferentes. Por tanto, las aplicaciones estarán mejor organizadas y serán más claras de entender.

Por otra parte, se facilita el mantenimiento de las aplicaciones. Es posible hacer cambios en la definición de una clase (por ejemplo, un cambio en uno de los métodos usando un algoritmo más rápido) sin tener que cambiar el programa principal.

Las ventajas modularidad y facilidad de modificación son especialmente evidentes cuando el tamaño de la aplicación es grande. Si la aplicación es pequeña entonces probablemente no vale la pena usar la programación orientada a objetos. Por último, otra ventaja importante de la programación orientada a objetos es la facilidad para reutilizar código.

En resumen las ventajas de la POO son:

- Fomenta la reutilización y extensión del código.
- Facilita el mantenimiento del software.
- Permite crear sistemas más complejos.
- Agiliza el desarrollo de software.
- Facilita la creación de programas visuales.
- Facilita el trabajo en equipo.
- Relacionar el sistema al mundo real.

2.1.2 OBJETOS

“...Un objeto es una estructura de datos (cuyos componentes llamados atributos) y un conjunto de procedimientos y funciones (que denominados métodos) para manipular la estructura de datos. La única forma de manipular la información del objeto es a través de sus métodos.”¹³

Un objeto se caracteriza por varios conceptos:

- **Atributos:** estos son los datos que caracterizan al objeto. Son variables que almacenan datos relacionados al estado de un objeto.
- **Métodos (usualmente llamados funciones de miembro):** Los métodos de un objeto caracterizan su comportamiento, es decir, son todas las acciones (denominadas operaciones) que el objeto puede realizar por sí mismo. Estas operaciones hacen posible que el objeto responda a las solicitudes externas (o que actúe sobre otros objetos). Además, las operaciones están estrechamente ligadas a los atributos, ya que sus acciones pueden depender de, o modificar, los valores de un atributo.

¹³ <http://studies.ac.upc.edu>

- **Identidad:** El objeto tiene una identidad, que lo distingue de otros objetos, sin considerar su estado. Por lo general, esta identidad se crea mediante un identificador que deriva naturalmente de un problema (por ejemplo: un producto puede estar representado por un código, un automóvil, por un número de modelo, etc.).

2.1.2.1 CARACTERISTICAS DE OBJETOS

2.1.2.1.1 HERENCIA

“...La herencia es una de las diferencias clave entre sistemas convencionales y los sistemas OO. Una subclase Y hereda todos los atributos y operaciones asociadas con su superclase X. la reutilización se realiza directamente”¹⁴.

La herencia permite crear estructuras jerárquicas de clases donde es posible la creación de sub-clases que incluyan nuevas propiedades y atributos. Estas sub-clases admiten la definición de nuevos atributos, así como crear, modificar o inhabilitar propiedades. Es importante destacar que en cada nivel de la jerarquía de clases, pueden añadirse nuevos atributos y operaciones a aquellos que han sido heredados de niveles superiores de la jerarquía. De hecho, cada vez que se debe crear una nueva clase, el ingeniero del software tiene varias opciones:

2.1.2.1.2 POLIMORFISMO

La palabra polimorfismo proviene del griego y significa que posee varias formas diferentes. Este es uno de los conceptos esenciales de una programación orientada a objetos. Así como la herencia está relacionada con las clases y su jerarquía, el polimorfismo se relaciona con los métodos.

“... El polimorfismo permite que un número de operaciones diferentes tengan el mismo nombre. Esto reduce el acoplamiento entre objetos, haciendo a cada uno más independiente.”¹⁵

2.1.2.1.3 ENCAPSULAMIENTO

La encapsulación es un mecanismo que consiste en organizar datos y métodos de una estructura, conciliando el modo en que el objeto se implementa, es decir,

¹⁴ Roger Presuman, “Ingeniería del Software”, 5ª Edición, pag. 348

¹⁵ Roger Pressman, “Ingeniería del Software”, 5ª Edición, pág. 350

evitando el acceso a datos por cualquier otro medio distinto a los especificados. Por lo tanto, la encapsulación garantiza la integridad de los datos que contiene un objeto.

La encapsulación proporciona un número importante beneficios:

- Los detalles de implementación interna de datos y procedimientos están ocultos al mundo exterior. Esto reduce la propagación de efectos colaterales cuando ocurren cambios.
- Las estructuras de datos y las operaciones que las manipulan están mezcladas en una entidad sencilla; la clase. Esto facilita la reutilización de componentes.
- Las interfaces entre objetos encapsulados están simplificadas. Un objeto que envía un mensaje no tiene que preocuparse de los detalles de las estructuras de datos internas en el objeto receptor, lo que simplifica la interacción y hace que el acoplamiento del sistema tienda a reducirse.

2.1.2.1.4 MENSAJE

Un mensaje es simplemente una petición de un objeto a otro objeto para que éste se comporte de una manera determinada, ejecutando uno de sus métodos. La técnica de enviar mensajes se conoce como paso de mensajes.

2.1.2.1.5 ABSTRACCION

Significa representar propiedades esenciales sin incluir los detalles o la implementación de las mismas.

2.1.2.1.6 GENERALIZACION/ESPECIALIZACION

A veces sucede que dos clases tienen muchas de sus partes en común, lo que normalmente se abstrae en la creación de una tercera clase (*padre* de las dos) que reúne todas sus características comunes.

El ejemplo más extendido de este tipo de relaciones es la herencia, propiedad por la que una clase (*clase hija*) recoge aquellos métodos y atributos que una segunda clase (*clase padre*) ha especificado como "heredables".

Este tipo de relaciones es característico de la programación orientada a objetos.

En realidad, la generalización y la especialización son diferentes perspectivas del mismo concepto, la primera proviene del hecho de que la superclase generaliza a la subclase y la segunda hace alusión al hecho de que la subclase especializa a la superclase, una subclase puede anular una característica de una superclase definiendo una característica del mismo nombre, se hace para obtener un mejor rendimiento.

Al Análisis y Diseño orientado a objetos se lo puede resumir en el siguiente cuadro:

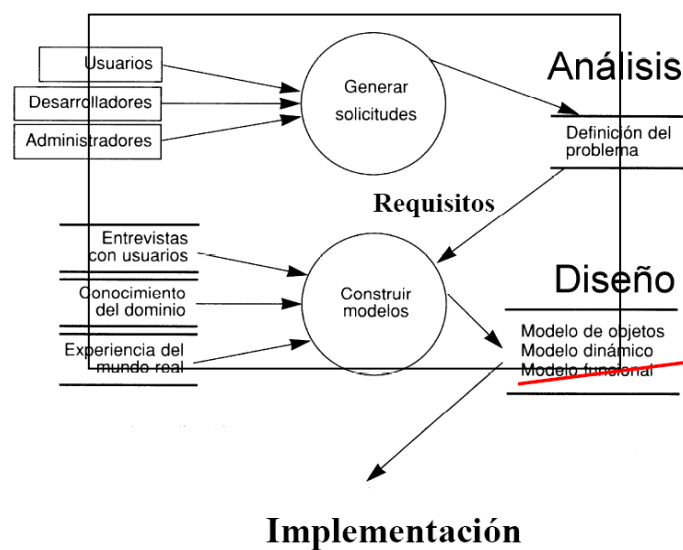


Figura 2.1 Análisis y Diseño Orientado a Objetos¹⁶

2.2 METODOLOGIA OMT

2.2.1 INTRODUCCION A OMT

OMT pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción , en el cual el análisis esta enfocado en el mundo real para un nivel de diseño, también coloca detalles particulares para modelado de recursos de computación. Esta Tecnología puede ser aplicada en varios aspectos de

¹⁶ **Fuente:** Cueva Lovelle Juan Manuel, *Introducción al UML*

implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos.

OMT esta construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones.

Desde que la comunidad de programación orientada a objetos tuvo la noción de incorporar el pensamiento de que los objetos son entidades coherentes con identidad estado y conducta, estos objetos pueden ser organizados por sus similitudes y sus diferencias, puestas en uso en herencia y polimorfismo.

2.2.2 DEFINICION DE OMT

OMT, es una metodología orientada a objetos muy difundida que se hace cargo de todo el ciclo de vida del software. Parte de la idea de utilizar los mismos conceptos y la misma notación a lo largo de todo el ciclo del software.

Divide el ciclo de vida del software en cuatro fases consecutivas: Análisis de objetos, Diseño del Sistema, Diseño de Objetos e Implementación. Se entiende a la Metodología del diseño en el siguiente cuadro:

Análisis	Modelo de Objetos Modelo Dinámico Modelo Funcional
Diseño a nivel Sistema	Subsistemas Hardware/Software Administración de datos
Diseño a nivel Objetos	Modelo de Objetos Modelo Dinámico Modelo Funcional

Figura 2.2 Metodología de Diseño¹⁷

a) Análisis

Lograr un modelo de la situación real que compete al sistema los requerimientos.

- El objetivo es eliminar la ambigüedad de una descripción informal y tener una descripción clara del problema y de los requerimientos.

¹⁷ **Fuente:** Sergio M Fernández Sastre, *Fundamentos del diseño y la programación orientada a objetos*

- No describe una solución.

b) Diseño a Nivel Sistema

Establecer el enfoque con el que se solucionará el problema:

- Establecer la Arquitectura del Sistema: organizar el sistema en subsistemas.
- Tomar una posición en relación a la administración de datos definir la arquitectura de hardware y software de base que se utilizará.
- Establecer criterios para definir prioridades.

c) Diseño a Nivel Objetos

- Especificar en detalle las clases y sus operaciones.
- Tomar en cuenta las decisiones realizadas en el Diseño a Nivel Sistema.
- Agregar al modelo del Análisis los objetos necesarios para obtener una solución.
- Especificar en forma detallada algoritmos e interfaces.
- Optimizar las estructuras descritas en el Análisis.

2.2.2.1 VENTAJAS

- Proporciona una serie de pasos perfectamente definidos al desarrollador a través del análisis, diseño e implementación.
- Tratamiento especial de la herencia.
- Facilita el mantenimiento dada la gran cantidad de información que se genera en el análisis.

2.2.2.2 DESVENTAJAS

- Hay pocos métodos para encontrar inconsistencias en los modelos.
- Interacción de objetos no soportada explícitamente en ninguna herramienta gráfica.
- Al ser un análisis iterativo es difícil de saber cuando comenzar con el diseño.

2.3 DIAGRAMAS UML

2.3.1 INTRODUCCION A UML

Las notaciones permiten formular ideas complejas en forma resumida y precisa. En los proyectos que involucran a muchos participantes, a menudo con diferentes conocimientos técnicos y culturales, la precisión y claridad son críticas conforme se incrementa rápidamente el costo de la falta de comunicación.

Para que una notación permita la comunicación precisa debe tener una semántica bien definida, debe ser muy adecuada para la representación de un aspecto dado de un sistema y debe ser bien comprendida por los participantes del proyecto. Se elige el UML como notación principal del desarrollo del sistema.

UML es una notación que se produjo como resultado de la unificación de la técnica de modelado de objetos (OMT, por sus siglas en inglés Rumbaugh 1991, Booch 1994, e ingeniería de software orientada a objetos Jacobson 1992). El UML ha sido diseñado para un amplio rango de aplicaciones. Por lo tanto, proporciona construcciones para un amplio rango de sistemas y actividades.

2.3.2 DEFINICION DE UML

El lenguaje para modelamiento (UML) “es un lenguaje para la especificación, visualización, construcción y documentación (...)”¹⁸.

El "UML" es un lenguaje de modelización genérico, independiente del proceso de construcción de software. No pretende definir el proceso de desarrollo. Pero asume la existencia de un proceso con estas características:

- Guiado por Casos de Uso
- Centrado en una Arquitectura
- Iterativo
- Incremental

¹⁸ Bernd Bruegge, Allen h. Dutoit, “Ingeniería del Software orientada a objetos”, Prentice Hall pag 54

2.3.2.1 UTILIDAD

UML pre-escribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML, diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros.

2.3.2.2 NOTACION

La notación de UML pretende ser un lenguaje común para el modelamiento de cualquier sistema.

Los diagramas principales de UML son:

Modelo de Objetos:

- Diagramas de clases
- Diagramas de objetos

Modelo Dinámico

- Diagramas de interacción
- Diagramas de estados
- Diagramas de actividad

Modelo Funcional

- Diagramas de casos de uso

Modelo de Implementación

- Diagrama de componentes
- Diagrama de despliegue

2.3.2.2.1 DIAGRAMA DE CLASES

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los

diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargará del funcionamiento y la relación entre uno y otro.

Los objetivos de realizar un diagrama de clases son:

- Describir las clases y las relaciones entre las mismas
- Describir la arquitectura de la solución

Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica, en el UML se lo representa así:

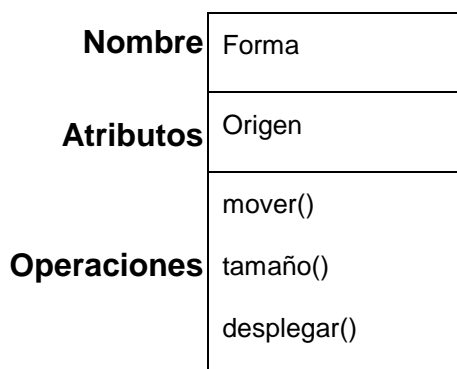


Figura 2.3 Representación de Clase en UML¹⁹

2.3.2.2 DIAGRAMA DE OBJETOS

Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto.

Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de

¹⁹ **Fuente:** Bernd Bruegge, Allen h. Dutoit, "Ingeniería del Software orientada a objetos"

un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase.

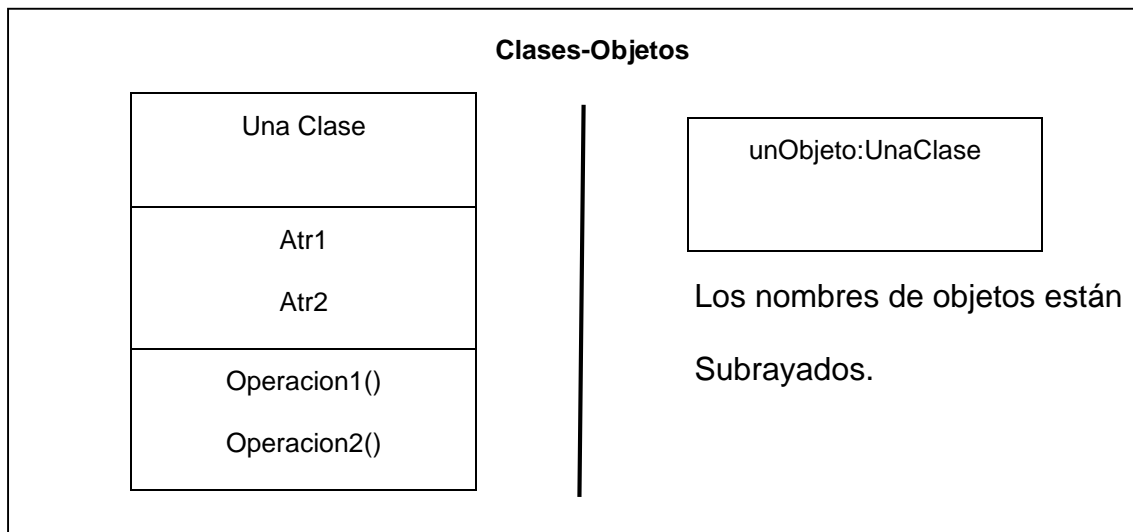


Figura 2.4 Diagrama de Clases y Objetos ²⁰

2.3.2.2.3 DIAGRAMA DE INTERACCION

Un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias (y las clases) del modelo de éstas. El punto de partida de las interacciones es el cumplimiento de las poscondiciones de los contratos de operación.

El UML define dos tipos de diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje: Diagramas de colaboración y Diagramas de secuencia.

a. Diagrama de Colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes

²⁰ **Fuente:** Coad P., *Object-Oriented Programming*

deben determinarse explícitamente mediante números de secuencia. Como en el siguiente ejemplo:

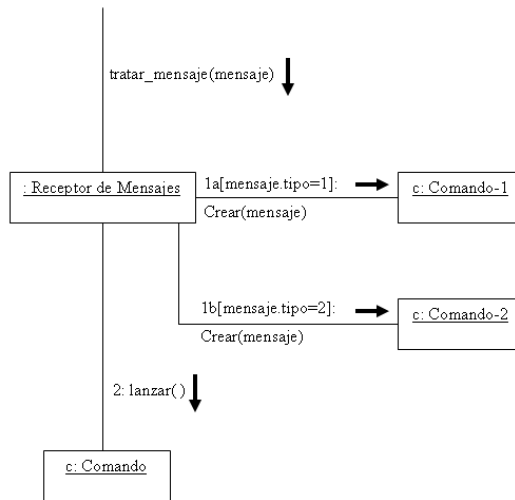


Figura 2.5 Secuencias de Mensajes y Flujos de Ejecución²¹

b. Diagrama de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se puede apreciar el movimiento en la siguiente figura:

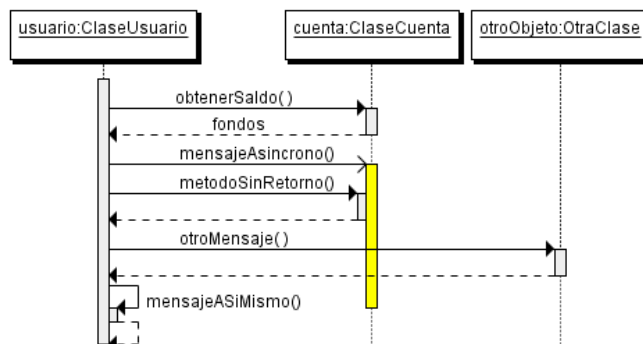


Figura 2.6 Ejemplo de Diagrama de Secuencia²²

²¹ **Fuente:** Coad P., *Object-Oriented Programming*

2.3.2.2.4 DIAGRAMA DE ESTADOS

Los diagramas de estados describen el comportamiento de un objeto individual como varios estados y transiciones entre esos estados. Un estado representa un conjunto particular de valores para un objeto.

El diagrama de estados y transiciones engloba todos los mensajes que un objeto puede enviar o recibir. En un diagrama de estados, un escenario representa un camino dentro del diagrama. Dado que generalmente el intervalo entre dos envíos de mensajes representa un estado, se pueden utilizar los diagramas de secuencia para buscar los diferentes estados de un objeto.

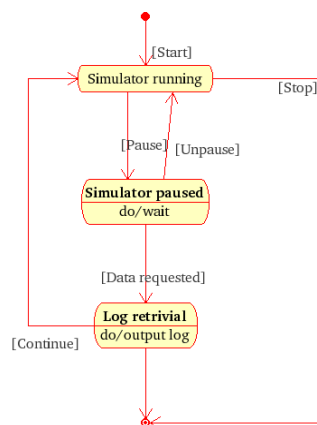


Figura 2.7 Ejemplo de Diagrama de Estado²³

2.3.2.2.5 DIAGRAMA DE ACTIVIDAD

Un diagrama de actividad describe un sistema desde el punto de vista de las actividades.

Un diagrama de actividad representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividad muestra el flujo de control general. Los Diagramas de Actividades están formados por un conjunto de Estados Activos y las transiciones entre ellos. Un *Estado Activo* indica que durante el período de tiempo en el que la instancia se encuentra en ese estado, está realizando una acción. Las transacciones son disparadas por la finalización de esa acción.

²² **Fuente:** http://es.wikipedia.org/wiki/Diagrama_de_secuencia

²³ **Fuente:** http://es.wikipedia.org/wiki/Diagrama_de_estados

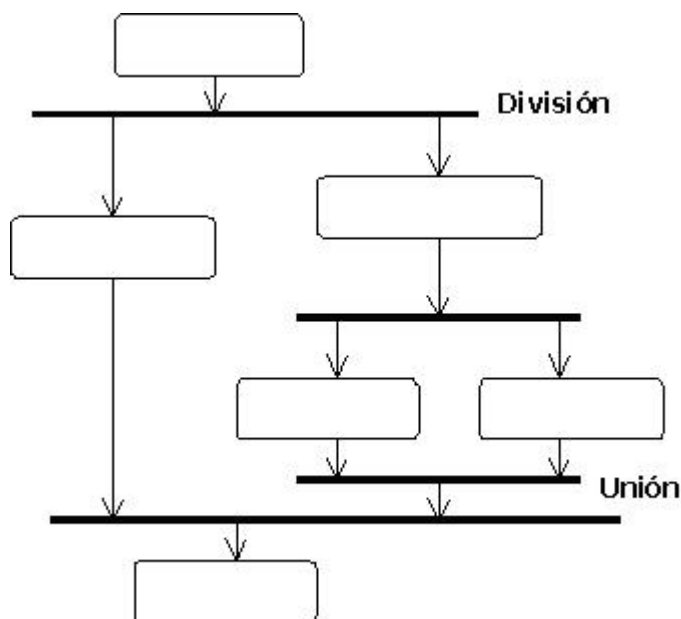


Figura 2.8 Ejemplo de Diagrama de Actividad²⁴

2.3.2.2.6 DIAGRAMA DE CASOS DE USO

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la Figura se muestra un ejemplo de Diagrama de Casos de Uso.

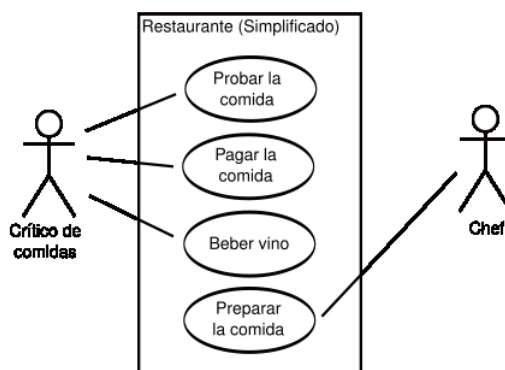


Figura 2.9 Ejemplo de Diagrama de Casos de Uso²⁵

²⁴ Fuente: http://es.wikipedia.org/wiki/UML#Tipos_de_Diagrama_en_UML

2.3.2.2.7 DIAGRAMA DE COMPONENTES

El diagrama de componentes es aquel que muestra las relaciones físicas entre los componentes de software y de hardware en el sistema entregado. Así, el diagrama de componentes, es un buen sitio para mostrar cómo se mueven los componentes y los objetos, dentro de un sistema distribuido. Muestran las dependencias entre los distintos componentes de software, incluyendo el código fuente, archivos binarios y ejecutables

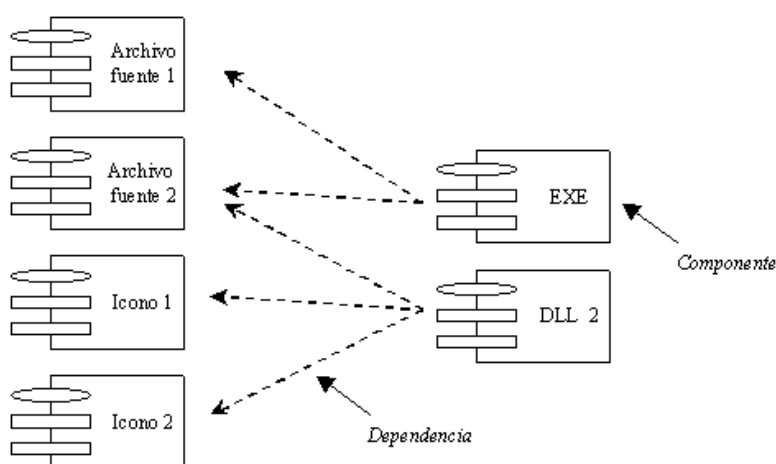


Figura 2.10 Ejemplo de Diagrama de Componentes²⁶

2.3.2.2.7 DIAGRAMA DE DESPLIEGUE O EJECUCION

El Diagrama de Despliegue o ejecución muestra la configuración de los elementos de procesamientos en tiempo de ejecución y los componentes de software, procesos y objetos que se ejecutan en ellos. Instancias de los componentes de software representan manifestaciones en tiempo de ejecución del código. Componentes que solo son utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes. Se puede visualizar a través del siguiente ejemplo:

²⁵ **Fuente:** http://es.wikipedia.org/wiki/UML#Tipos_de_Diagrama_en_UML

²⁶ **Fuente:** http://es.wikipedia.org/wiki/UML#Tipos_de_Diagrama_en_UML

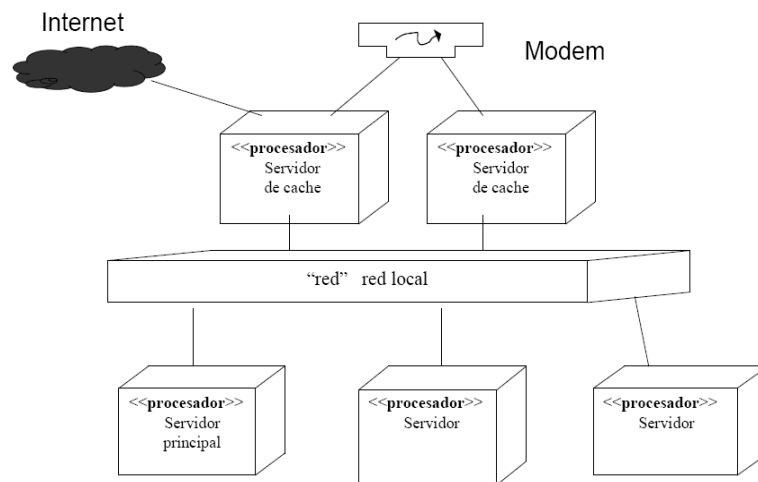


Figura 2.11 Ejemplo de Diagrama de Ejecución²⁷

²⁷ Fuente: Coad P., Object-Oriented-Programming

CAPITULO 3.

ANALISIS Y DISEÑO DEL SISTEMA

3.1 ESPECIFICACIONES Y REQUISITOS DEL SOFTWARE

3.1.1 INTRODUCCION

3.1.1.1 PROPOSITO

El propósito perseguido al documentar los requerimientos del sistema CELLARACTIVE (Sistema Informático para el Manejo y Control de Información de Activos Fijos) es la reunión organizada y clara de las necesidades operacionales y organizativas que se esperan una vez que el producto entre en el proceso de funcionamiento. Por otra parte el sistema pretende brindar información indispensable para los usuarios que permita el seguimiento de los Activos Fijos, para la toma adecuada de decisiones.

3.1.1.2 ÁMBITO DEL SISTEMA

La Gestión y Control de Activos, es un proceso, que bien ejecutado trae muchos beneficios.

- Para llegar a ello, se debe pensar en la elaboración de un Plan Formal del Trabajo.
- El proceso comienza en el Inventario Inicial de Activos, pasando por Gestión y Mantenimiento, finalizando en el control inmediato de activos fijos.
- La Gestión de Activos Fijos implica una filosofía o cultura particular de trabajo, que debe ser aceptada por la Alta Gerencia de la empresa, y adoptada por el personal involucrado en este proceso, para garantizar su éxito.

Beneficios	Capacidades Del Sistema
Atenderá a las necesidades en el ingreso , modificación y control de categorías de AF	Ser eficiente en la categorización clara de los AF
Control de AF	El departamento financiero podrá realizar un seguimiento desde el ingreso del AF a la institución, y sus respectivas valorizaciones o depreciaciones.
Registrar las diferentes ubicaciones que tiene el AF durante su vida útil dentro de la institución.	Permitirá a los usuarios tener un seguimiento del AF

Tabla 3: Beneficios y Capacidades del Sistemas

3.1.1.3 DEFINICON, ACRONIMOS, SIGLAS Y ABREVIATURAS

La descripción de las siglas es:

- **CELLARACTIVE:** sistema informático para el manejo y control de información de activos fijos.
- **SOFTWARE.-** Es el conjunto de instrucciones que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina.
- **IEEE.-** Instituto de Ingenieros Eléctricos y Electrónicos.
- **OMT.-** Técnica de Modelado de Objetos.
- **UML.-** Lenguaje de Modelado Unificado.
- **WINDOWS.-** Plataforma o sistema operativo donde la aplicación tendría su funcionamiento.
- **AF.-** Activo Fijo.

Las asignaciones de términos descriptivos son:

- **Usuario:** es la persona, que ingresa al sistema con el propósito de planificar un pedido u obtener información para tomar decisiones.

- **Control:** El proceso de medir los actuales resultados en relación con los planes, diagnosticando la razón de las desviaciones y tomando las medidas correctivas necesarias.
- **Categorías:** Una categoría, es una totalidad atributiva en la que ha sido posible concatenar, por cierres operatorios, unas partes con otras en círculos de radio más o menos amplio, intercomunicados entre sí

3.1.1.4 REFERENCIAS

Los libros que han servido de guía para la elaboración de este documento son:

- Principios Administración Financiera, Lawrence J. Gitman, Pearson Edición, 2003.
- Tesis Administración Financiera, Madura Jeff. Universidad San Francisco de Quito, 2001.
- Tesis Procesos Contable, Elizondo Arturo, Universidad San Francisco de Quito, 2000.
- Metodología, recodificación y control de activos fijos de una empresa de servicios petroleros. Javier Mauricio Guevara Albornoz ; Verónica Alexandra Olmedo Baldeón, PUCE, 2002

3.1.2 DESCRIPCION GENERAL

3.1.2.1 PERSPECTIVA DEL PRODUCTO

Las expectativas que genera el sistema son la flexibilidad y fácil manejo para los diferentes usuarios, pues al estar desarrollado en un entorno Windows permite su fácil manejo, acceso a datos y pantallas; para el acceso al sistema se puede utilizar mouse, teclado; en el dispositivo móvil PALM existe el acceso por código de barras.

En caso de equivocaciones el sistema permite las correcciones o eliminación de un proceso con el que el usuario no esté de acuerdo, siempre emitiendo señales de advertencia de lo que está a punto de efectuarse.

Si existe el caso de olvido de llenar algún dato necesario el sistema emite una alerta o advertencia, para que se ingrese o se seleccione una opción.

En resumen los beneficios que se obtiene con este sistema son:

- Optimización de la administración de los Activos Fijos.
- Catálogo único de Activos Fijos.
- Recepción rápida y ordenada.
- Automatización de los Traslados, Auditorías, Desincorporación con equipos móviles.
- Clasificación de activos fijos por Ubicación Física
- Valorización real.
- Precisión de los cálculos contables.
- Consultas “inteligentes” de Activos Fijos (Ej: ¿Qué Existe?, ¿Dónde Está?, ¿Valor Actual?, ¿Cómo se Encuentra?).
- El producto posee la siguiente funcionalidad a resolver como se muestra en la figura a continuación:

MODELO FUNCIONAL: PERSPECTIVA DEL PRODUCTO

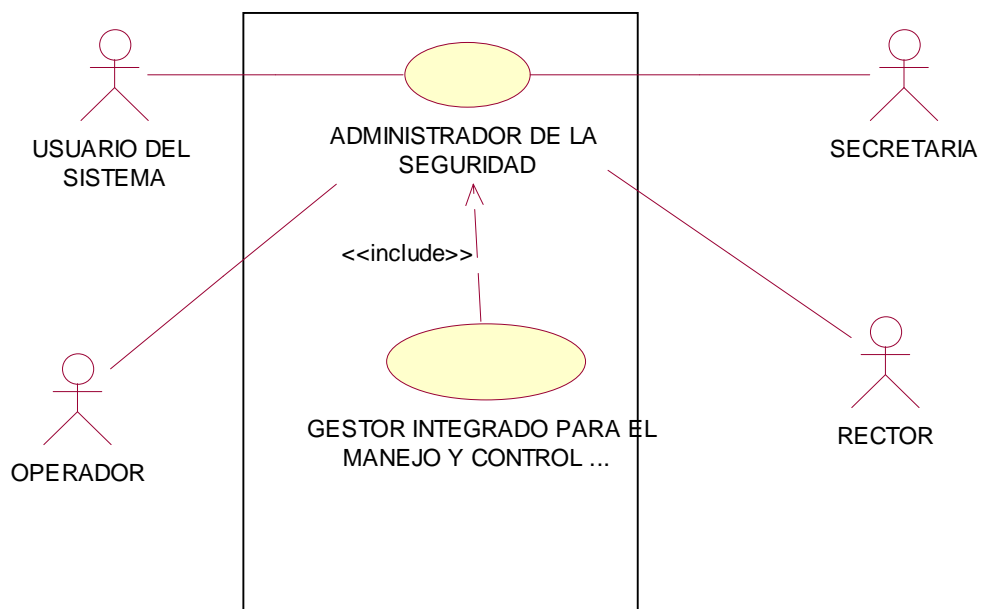


Fig. 3.1 PERSPECTIVA DEL PRODUCTO

3.1.2.2 FUNCIONES DEL PRODUCTO

Este sistema permitirá a los usuarios el ingreso al sistema por medio de una asignación de una clave y usuario, activando los respectivos roles y privilegios. Y una vez validado el ingreso, el usuario puede acceder al sistema a un entorno de fácil manejo. Además por medio del dispositivo móvil se puede ubicar los AF, de manera remota.

Las funciones de este sistema en el Gestor de Seguridad son:

- **Validador de la Seguridad:** Permite el fácil ingreso del usuario al sistema a través de un login y password, para el acceso al sistema, activando los roles y privilegios que cada usuario tienen autorizado.
- **Administrador de Usuario:** Admite al Administrador realizar los registros y/o actualizaciones de usuarios asignándoles los perfiles de los usuarios que interactúan con el sistema.

El Gestor Integrado para el Manejo y Control de AF:

- **Administrador de Activos Fijos:** Permite categorizar al AF, y a su vez, si existe un nuevo AF, se realiza su registro/actualización del mismo.
- **Organizador de AF:** Al recibir la solicitud de ubicar un AF, se tiene que aprobar la solicitud, y luego se registra la nueva ubicación a un departamento de la institución.
- **Calculador de la depreciación:** Permite realizar el cálculo de la revalorización o depreciación de un AF.
- **Gestor de AF:** Mediante la utilización del dispositivo móvil, permite legalizar la ubicación del AF, sin que esto implique que se deba trasladar el AF, sino el operador realiza el ingreso trasladándose hasta donde se encuentra el AF.
- **Sincronizador de Datos:** Este proceso alista al sistema y al dispositivo móvil para poder sincronizar los ingresos realizados desde la PALM a la base de datos del sistema.
- **Gestor de Reportes:** Básicamente permite solicitar los reportes necesarios para realizar un adecuado control y manejo de un AF.

3.1.2.3 CARACTERISTICAS DE USUARIO

Se debe tomar en cuenta que los usuarios que se tiene para el manejo del sistema deben ser como mínimo bachiller, y conocer del manejo de herramientas ofimáticas, para que puedan familiarizarse con el entorno en el cual está desarrollado el sistema CELLARACTIVE.

ROL	PRIVILEGIOS	S/N
Administrador	Validador de la Seguridad	S
	Administración de Usuarios	S
	Administrador de Activos Fijos	S
	Organizador de Activos Fijos	S
	Calculador de la depreciación	S
	Gestor de Activos Fijos(Aplicación palm)	S
	Sincronizador de datos	S
	Gestor de Reportes	S
Rector	Validador de la Seguridad	S
	Administración de Usuarios	N
	Administrador de Activos Fijos	N
	Organizador de Activos Fijos	N
	Calculador de la depreciación	N
	Gestor de Activos Fijos(Aplicación palm)	N
	Sincronizador de datos	N
	Gestor de Reportes	S

ROL	PRIVILEGIOS	S/N
Secretaria	Validador de la Seguridad	S
	Administración de Usuarios	N
	Administrador de Activos Fijos	S
	Organizador de Activos Fijos	S
	Calculador de la depreciación	S
	Gestor de Activos Fijos(Aplicación palm)	S
	Sincronizador de datos	N
	Gestor de Reportes	S
Jefe Financiero	Validador de la Seguridad	S
	Administración de Usuarios	N
	Administrador de Activos Fijos	S
	Organizador de Activos Fijos	N
	Calculador de la depreciación	S
	Gestor de Activos Fijos(Aplicación palm)	N
	Sincronizador de datos	N
	Gestor de Reportes	S

ROL	PRIVILEGIOS	S/N
Operador	Validador de la Seguridad	S
	Administración de Usuarios	N
	Administrador de Activos Fijos	N
	Organizador de Activos Fijos	N
	Calculador de la depreciación	N
	Gestor de Activos Fijos(Aplicación palm)	S
	Sincronizador de datos	S
	Gestor de Reportes	N

Tabla 4: Roles y Privilegios de Usuario

3.1.2.4 LIMITACIONES GENERALES

Para el adecuado desarrollo del sistema se debe tomar en cuenta los siguientes parámetros tanto de hardware como de software:

Hardware:

- Se dispone de los equipos pertenecientes a la empresa en la cual se está realizando el estudio. Pero al tratarse de una tesis de grado el limitante es el apoyo económico ya que este es mínimo, en vista de esta limitante, no se dispone del dispositivo móvil con lector de código de barras; lo que determinó que para el desarrollo del sistema se utilice un simulador del dispositivo para efectos de visualizar su funcionamiento.
- No existe una estandarización adecuada para la categorización de los AF.
- La resistencia al cambio por parte de algunos miembros de las instituciones.

3.1.3 REQUERIMIENTOS ESPECIFICOS

3.1.3.1 REQUISITOS FUNCIONALES

Los requerimientos serán especificados en base a los actores que son: Rector, Secretaria, Operador, Administrador, Jefe del departamento financiero.

- **Recepción efectiva del AF.**

La empresa requiere una recepción real y efectiva de los AF, que ingresan a la institución, es decir se debe registrar y categorizar a un AF. En esta función se debe especificar el tipo, la descripción, el valor, etc.

- **Flexibilidad en la Actualización**

Al inicio de cada año escolar se tiene una constante obtención de nuevos AF, necesarios para el adecuado funcionamiento de la institución, los mismos que son solicitados por los diferentes departamentos. Por esta razón el sistema debe permitir la constante actualización de nuevos AF.

- **Interactivo**

La facilidad de interactuar con el usuario es prioritaria, usar una comunicación amigable, la facilidad de poder acceder desde cualquier lugar de la institución sin que sea necesario trasladar el AF, de un lado a otro.

- **Facilidad en su Manejo**

Realizando un sistema de fácil acceso y con pantallas amigables y de fácil uso, evitará que se produzcan equivocaciones al ingresar los datos.

- **Exactitud**

El ingreso de los datos en todos los procesos no pueden contener equivocaciones, los ítems ingresados deben ser iguales a los presentados.

- **Organización**

El sistema debe permitir la navegación en el mismo en los siguientes ámbitos:

- Administrador de la seguridad.
- Gestor integrado para el Manejo y Control de los AF.

- **Velocidad de Respuesta**

Los usuarios requieren que las solicitudes que se realizan al sistema sean rápidas, para que los tiempos de operación no sean largos ni tediosos.

- **Seguridad en el Sistema.**

Al tener un Administrador de la seguridad que define la privacidad para cada usuario, así como el seguimiento de cada tarea que ejecuta, asigna responsabilidad de manejo a cada usuario.

- **Informes que se esperan del Sistema.**

- Catálogo de Activos Fijos.
- Clasificación de Activos Fijos por su Ubicación.
- Precisión de los cálculos contables de depreciación de activos fijos.
- Historial del Activos Fijos.
- Existencias de Activos.
- Valor Actual de Activos.

3.1.3.1 INTERFAZ DE USUARIO

Las interfaces de usuario deben presentarse con los siguientes requisitos:

- Deben tener un estándar en la ubicación de los botones de comandos, además las pantallas deben ser con un diseño similar, para que permita a los diversos usuarios una fácil adaptación al mismo.
- Los colores utilizados en las pantallas son formales es decir tonos que vayan en armonía al entorno Windows.
- El acceso a las interfaces son a través de mouse o teclado y el dispositivo móvil.
- Los diálogos de advertencia deben ser claros permitiendo al usuario un fácil entendimiento antes de realizar una acción.

3.1.3.3 INTERFACES DE HARDWARE

El CELLARACTIVE desarrollado requiere una conexión a red wireless, que está contemplada en la ejecución y en el diseño del sistema. Además se tiene la conexión con el dispositivo móvil, con el sincronizador de datos.

3.1.3.4 INTERFACES DE SOFTWARE

En el sistema se tiene la interfaz de software en el sincronizador de datos, que permite la conexión del sistema entre la base de datos en el servidor, y la que se encuentra en el dispositivo móvil.

3.1.3.5 INTERFACES DE COMUNICACION

- TCP/IP

3.1.3.6 REQUISITOS DE EJECUCION

Para que el sistema entre en funcionamiento los requisitos mínimos son:

3.1.3.6.1 REQUERIMIENTOS DE DISEÑO, DESARROLLO E IMPLEMENTACION

Los requerimientos necesarios para el diseño y la implementación son que existe un fácil acceso para la realización de los procesos de depreciación de activos fijos.

3.1.3.6.2 REQUERIMIENTOS PARA EL DESARROLLO

3.1.3.6.2.1 SOFTWARE

- SQL Server Express 2005.
- Visual Basic Express 2005.
- Rational Rose Enterprise Edition.
- Sistema Operativo Windows Xp.
- Sistema Operativo Windows Mobile 5.0 ó superior.

3.1.3.6.2.2 HARDWARE

- Computador Pentium IV ó superior.
- Espacio libre en disco 1GB.
- 512 MB en RAM mínimo.
- DVD Writer.
- Pda pocket pc con wi-fi.
- Router wireless.
- Tarjeta de red inalámbrica.

3.1.3.7 RESTRICCIONES LEGALES Y LICENCIAS

Para el desarrollo del sistema se requiere las licencias pagadas de los programas Rational Rose Enterprise Edition y Windows Xp, Sql Server 2005 y Visual Basic 2005 tienen licencias gratuitas ya que son versiones Express.

3.1.3.8 CUMPLIMIENTOS DE ESTANDARES

Estándares de análisis de requerimientos de acuerdo al documento de la IEEE830.

Estándares de Diseño:

Clases:

- Nombre de las clases inician con letras mayúsculas, haciendo referencia a lo que contiene.

Campos:

- Txt texto
- Lbl nombre del campo
- Lista lista

Variables: en cada variable se toma en consideración la función que va a desempeñar en el sistema para ponerle el nombre, en caso de ser un nombre muy extenso se toma las tres primeras letras de cada función.

Globales:

- Locales
- Procedimientos
- Funciones

Constantes:

- Id identificador

Para los contadores se han de denominar variables con letras i, j, k o con un nombre específico.

Interfaz Gráfica de Usuario:

- Frm_nombre de función de la Form
- Btn _nombre de acción del Button
- Listaview nombre lista.
- Rnd_nombre de la acción Radio Button

Mensajes: Cada mensaje contiene un texto claro con un cuadro de diálogo que aparece en el centro de la pantalla, cualquiera que sea el caso del mensaje.

- Exclamación
- Advertencia

Información: para este tipo de mensaje se utiliza la advertencia de Windows en la esquina inferior derecha de la pantalla.

3.2 MODELO DE OBJETOS

3.2.1 MODELO DE CLASES

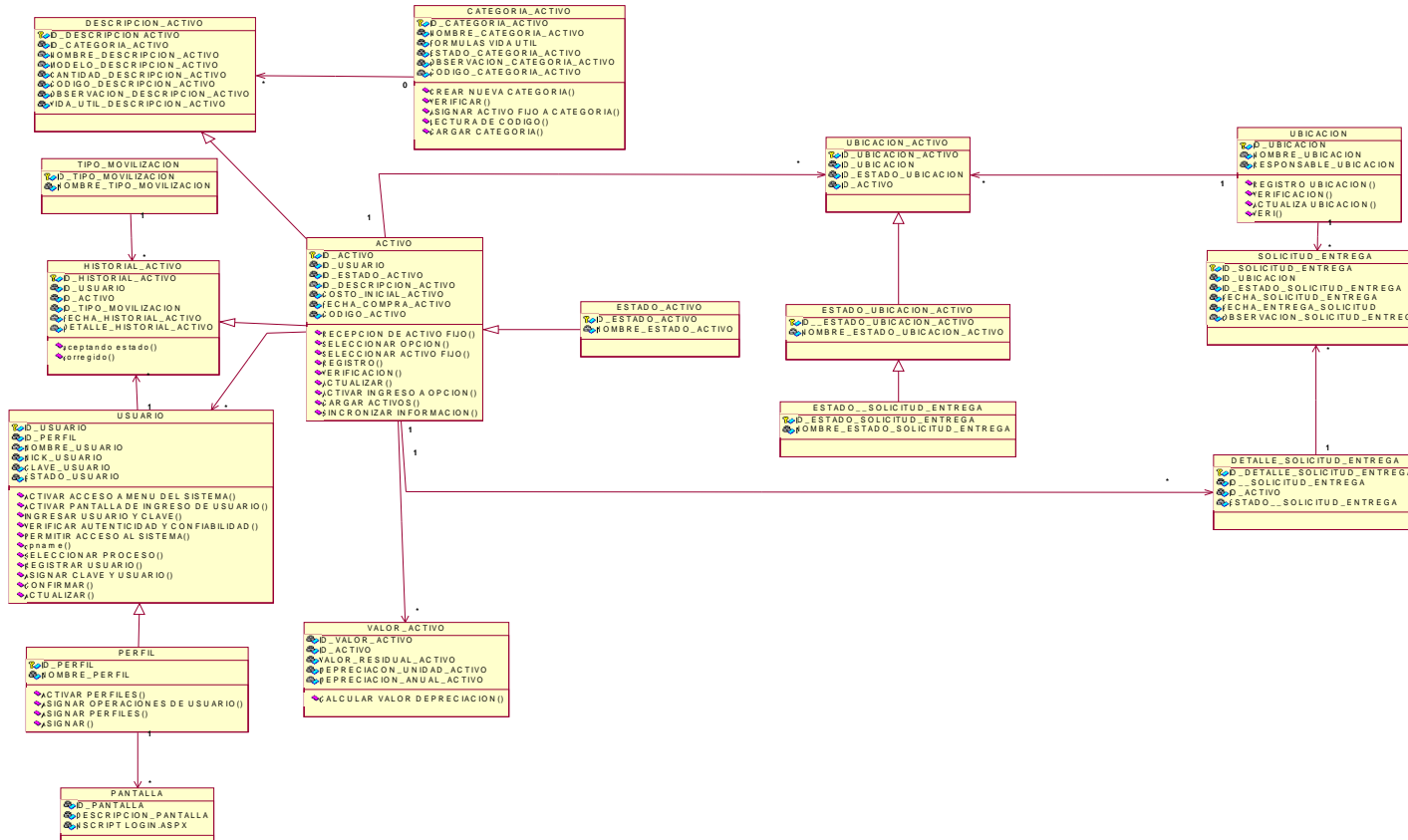


Diagrama 3.1: Clases

3.3 MODELO DINÁMICO

3.3.1 DIAGRAMAS DE INTERACCION

3.3.1.1 DIAGRAMAS DE SECUENCIA - ADMINISTRADOR DE LA SEGURIDAD

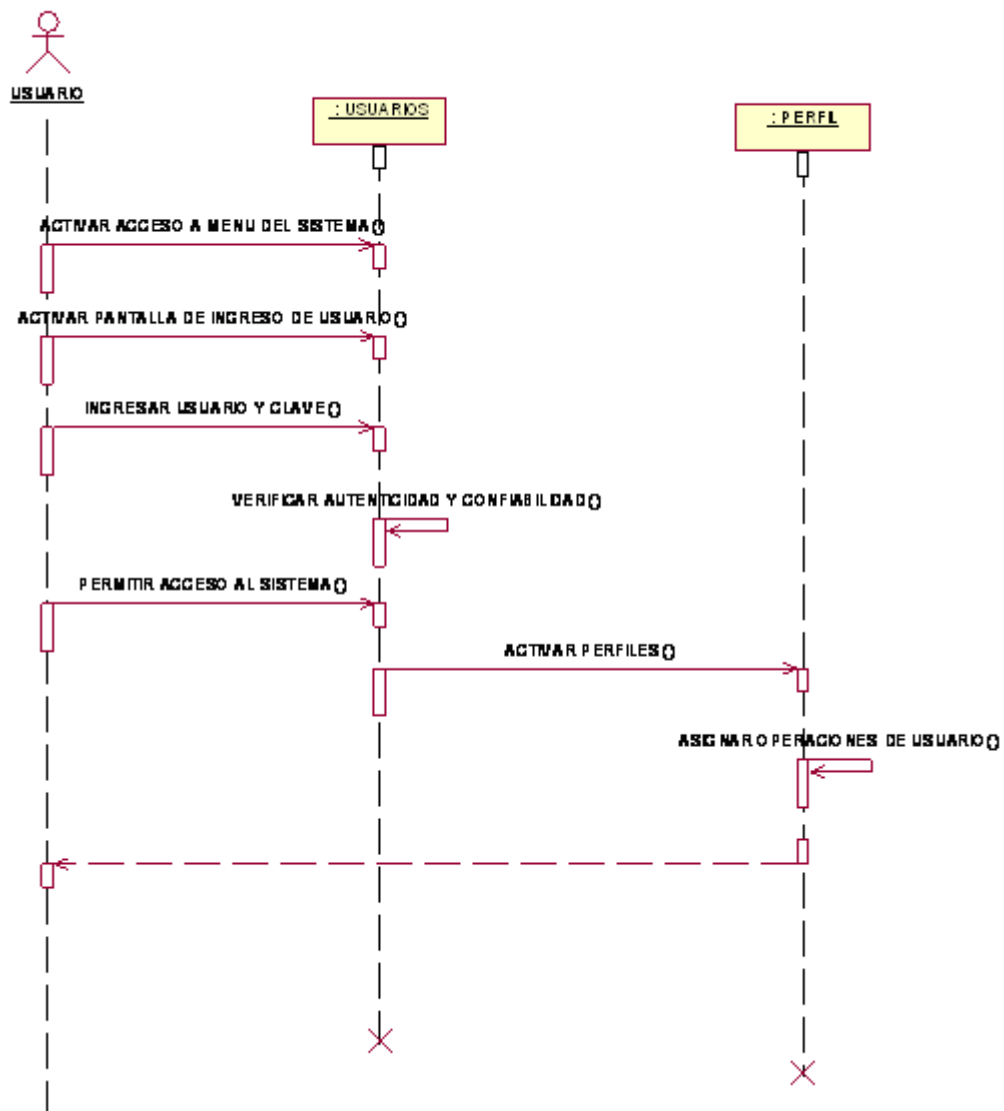


Diagrama 3.2: *Administrador de la seguridad*

Diagrama de Secuencia que interactúa entre el objeto usuario, y perfil, permitiendo el intercambio de mensajes para el ingreso del usuario, al sistema, verificando y validando la información para permitir el ingreso, y activar los perfiles del usuario.

3.3.1.2 DIAGRAMAS DE COLABORACION - ADMINISTRADOR DE LA SEGURIDAD

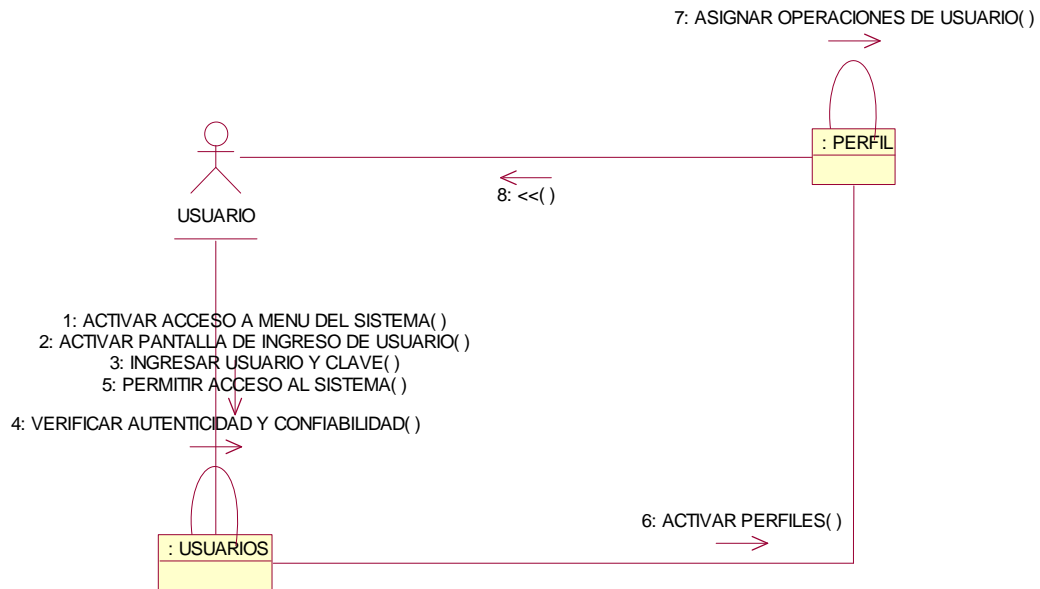


Diagrama 3.3: Administrador de la seguridad

Diagrama de colaboración generado a partir del diagrama de secuencia Administrador de la seguridad por la herramienta Rational Rose, que resalta la organización estructural de los objetos que intercambian mensajes.

3.3.1.3 DIAGRAMA DE SECUENCIA - ADMINISTRADOR DE USUARIOS

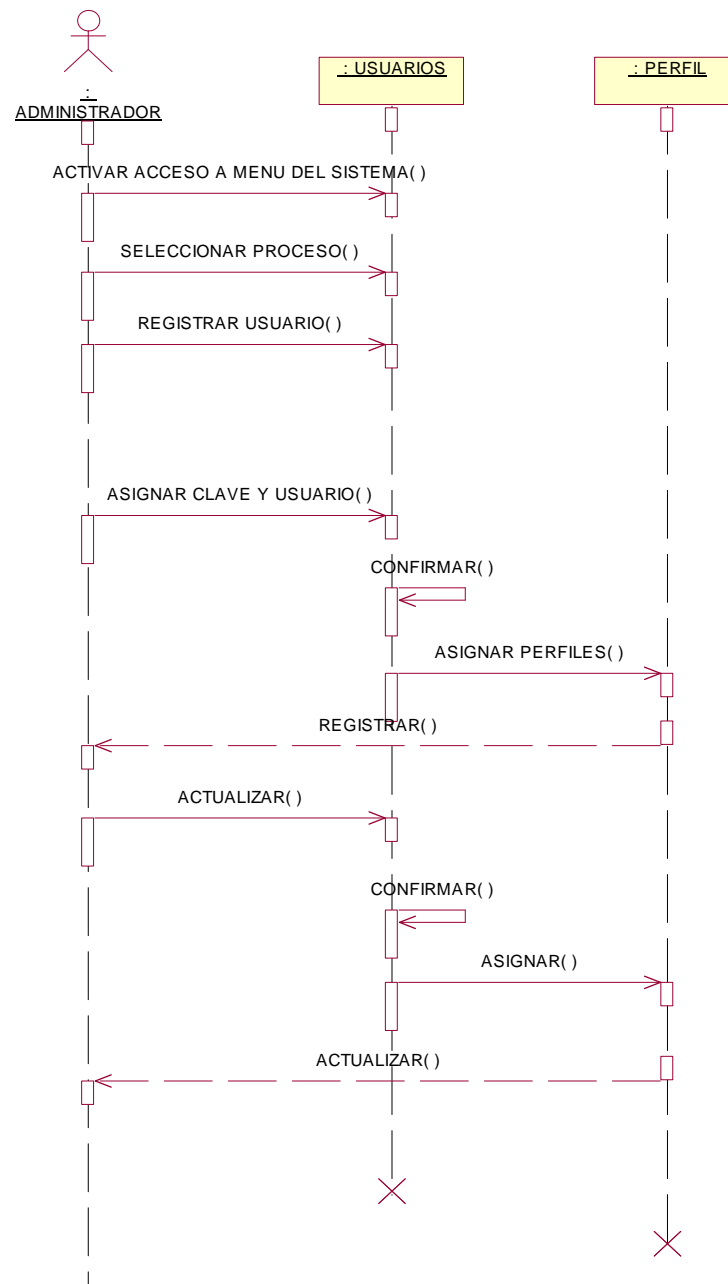


Diagrama 3.4: *Administrador de Usuarios*

Diagrama de secuencia Administrador de Usuarios que intercambia mensajes entre los objetos Usuarios y Perfil, permitiendo el acceso al sistema para registrar, un usuario o actualizarlo.

3.3.1.4 DIAGRAMA DE COLABORACION – Administrador de Usuarios

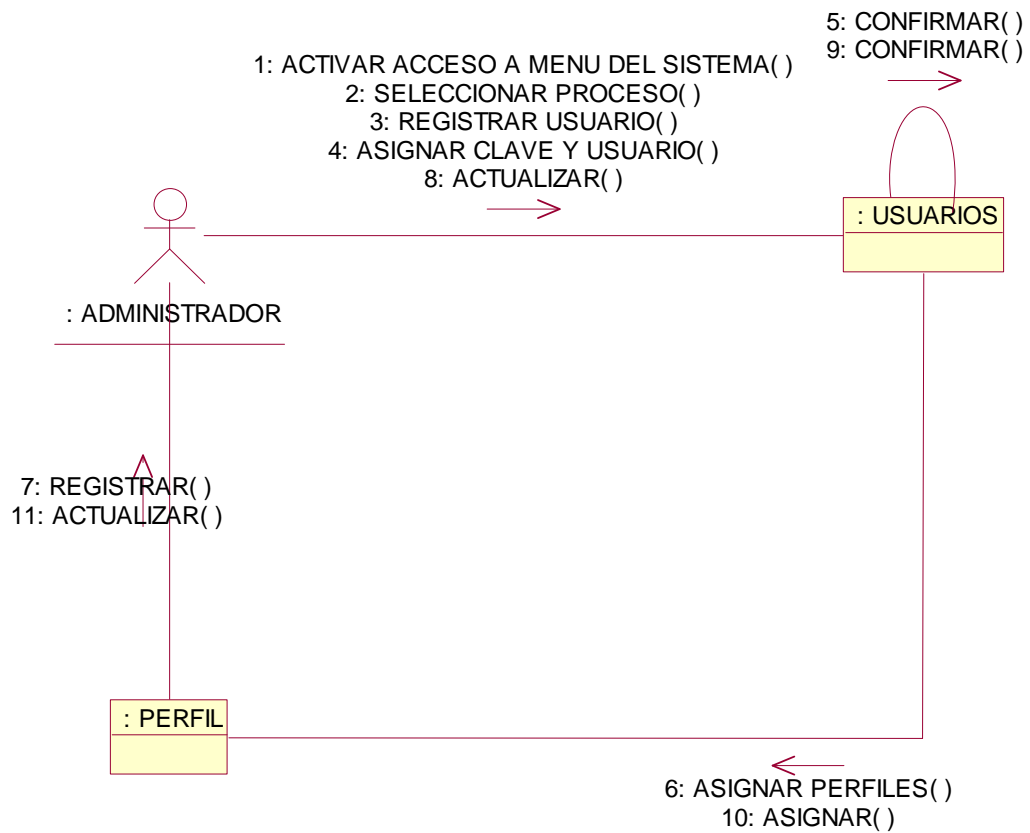


Diagrama 3.5: Administrador de Usuarios

Diagrama de colaboración generado a partir del diagrama de secuencia Administrador Usuarios por la herramienta Rational Rose, que resalta la organización estructural de los objetos Perfil y Usuarios que intercambian mensajes.

3.3.1.5 DIAGRAMA DE SECUENCIA – Administrador de Categorías

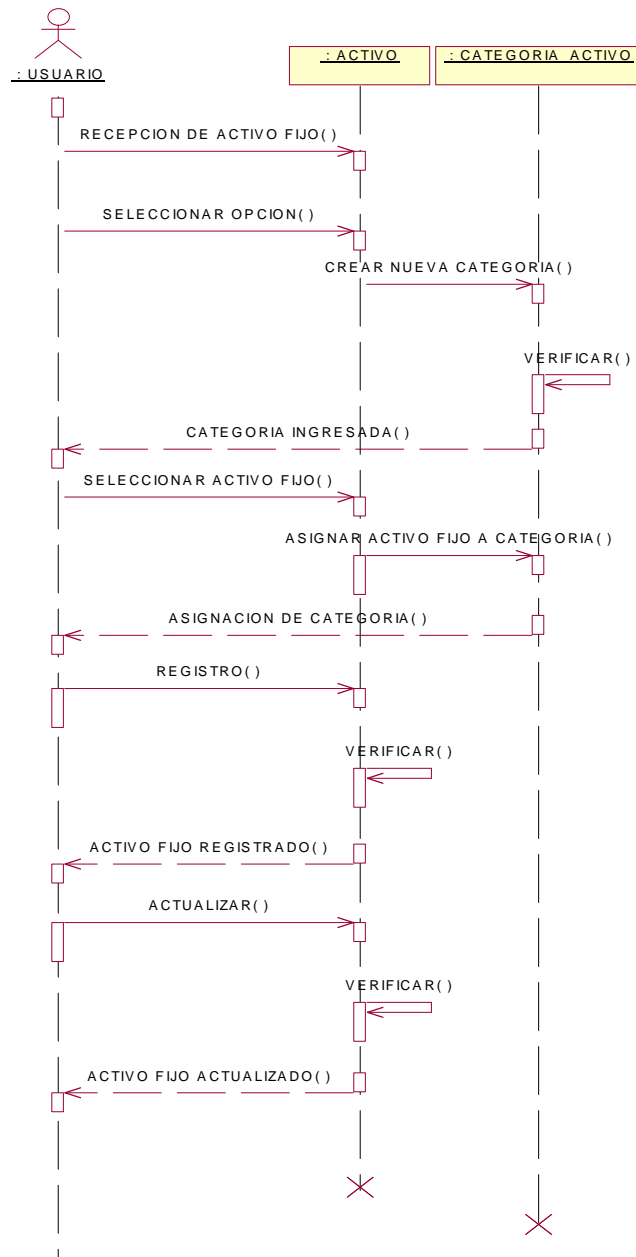


Diagrama 3.6: Administrador de Categorías

Diagrama de Secuencia Administrador de Categorías que intercambian mensajes entre los objetos Activo y Categoría de Activo, permitiendo la Recepción de un AF, creación de una categoría, y la respectiva asignación de una categoría a un AF, y el registro y la actualización de un AF.

3.3.1.6 DIAGRAMA DE COLABORACION- Administrador de Categorías

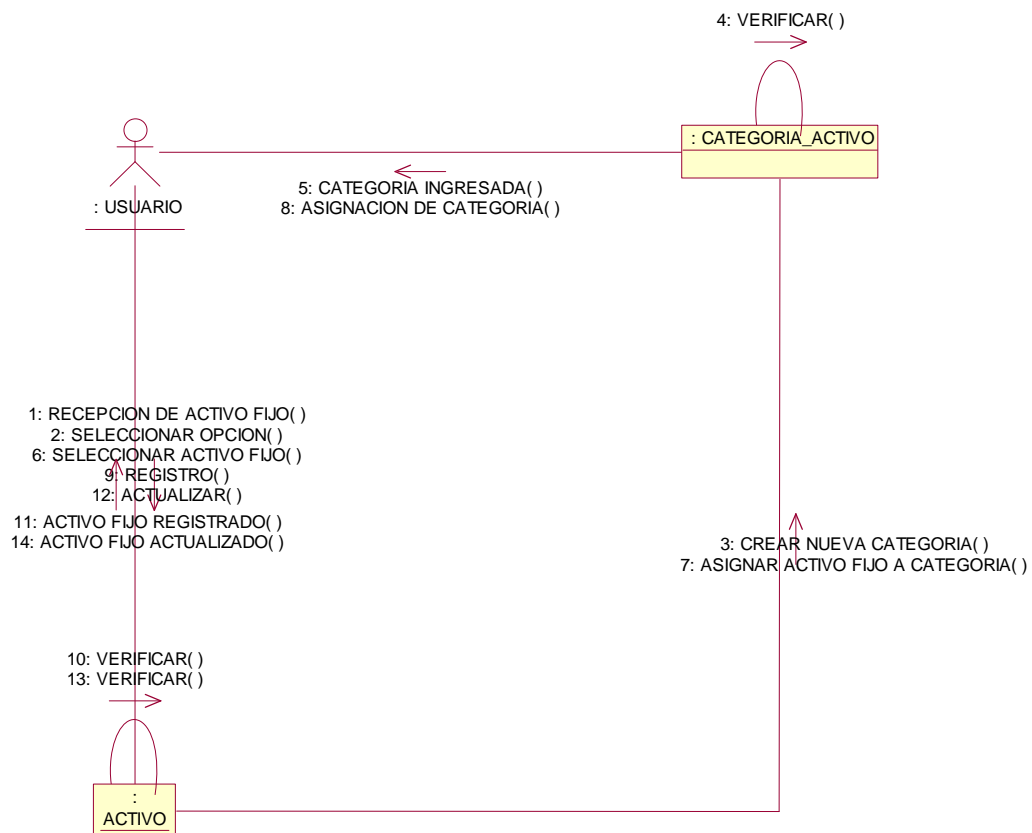


Diagrama 3.7: Administrador de Categorías

Diagrama de colaboración generado a partir del diagrama de secuencia Administrador de Categorías por la herramienta Rational Rose, que resalta la organización estructural de los objetos Activo, y Categoría de Activo que intercambian mensajes.

3.3.1.7 DIAGRAMA DE SECUENCIA - Organizador de Activos Fijos

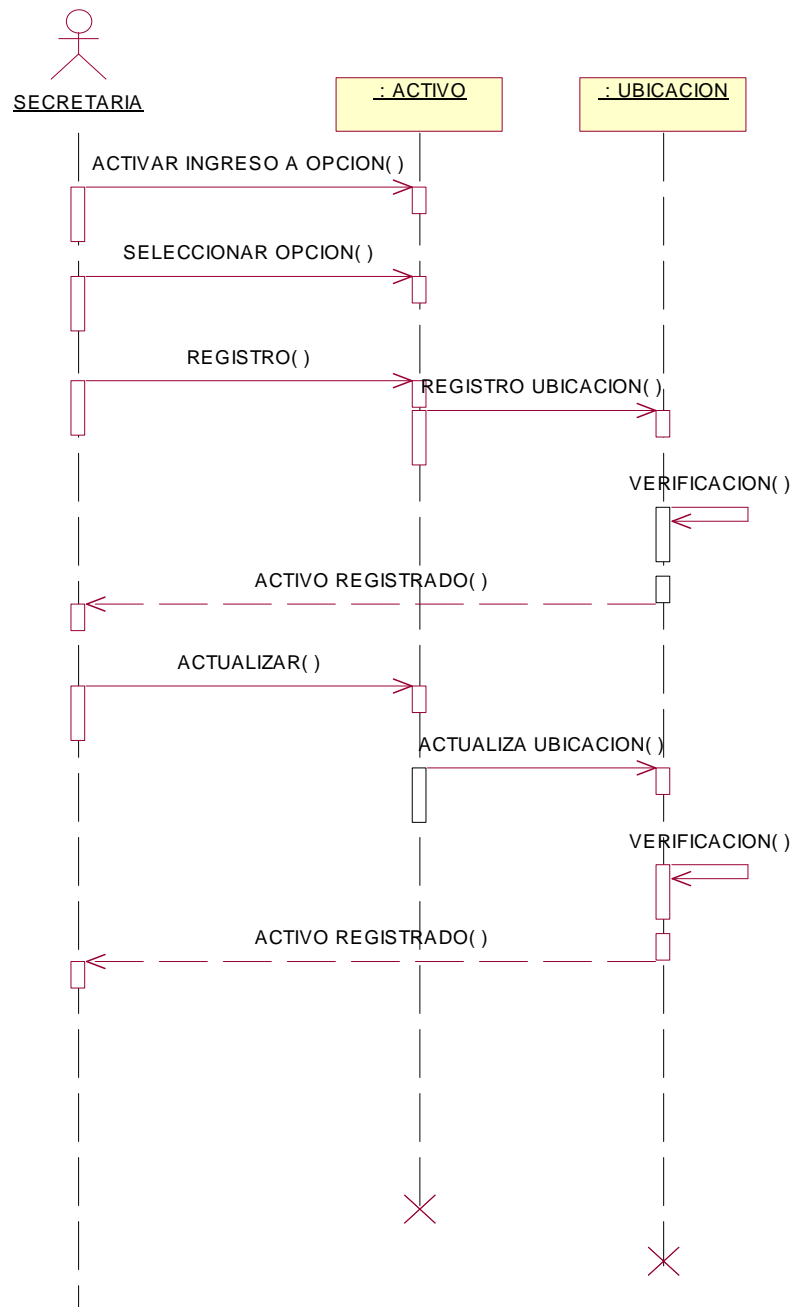


Diagrama 3.8: Organizador de Activos Fijos

Diagrama de Secuencia Organizador de Activos Fijos que intercambia mensajes entre los objetos Activo y Ubicación, que permite Activar el ingreso del AF, escoger una opción y registrar la ubicación, además de actualizar la ubicación.

3.3.1.8 DIAGRAMA DE COLABORACION - Organizador de Activos Fijos

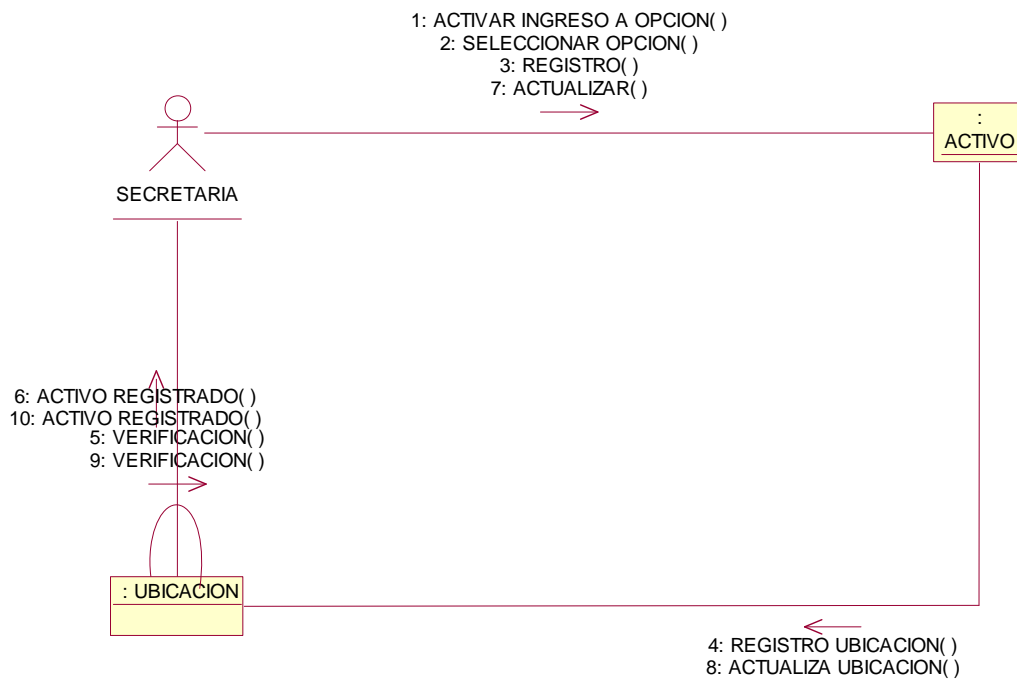


Diagrama 3.9: Organizador de Activos Fijos

Diagrama de colaboración generado a partir del diagrama de secuencia Organizador de Activos Fijos por la herramienta Rational Rose, que resalta la organización estructural de los objetos Activo y Ubicación que intercambian mensajes.

3.3.1.9 DIAGRAMA DE SECUENCIA - Calculador de la Depreciación

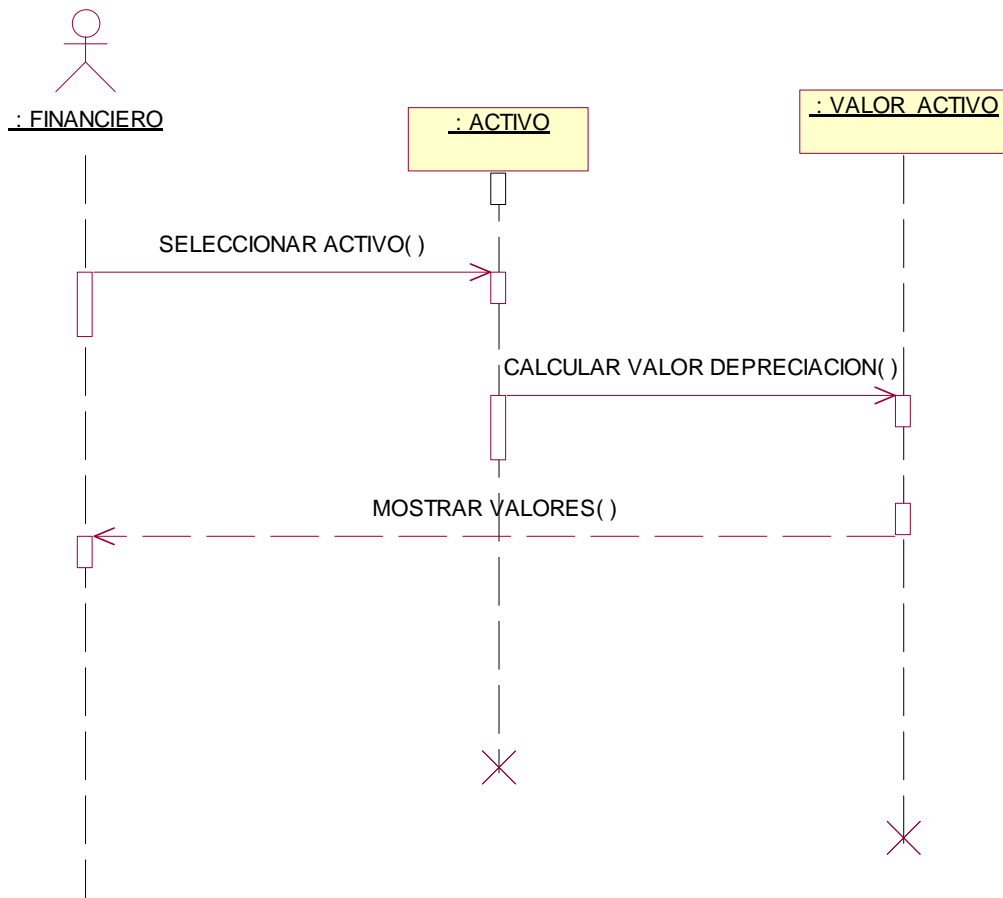


Diagrama 3.10: Calculador de la Depreciación

Diagrama de Secuencia Calculador de la Depreciación que intercambia mensajes entre los objetos Activo y Valor Activo, que permite seleccionar un activo, y calcular el valor de la depreciación.

3.3.1.10 DIAGRAMA DE COLABORACION - Calculador de la Depreciación

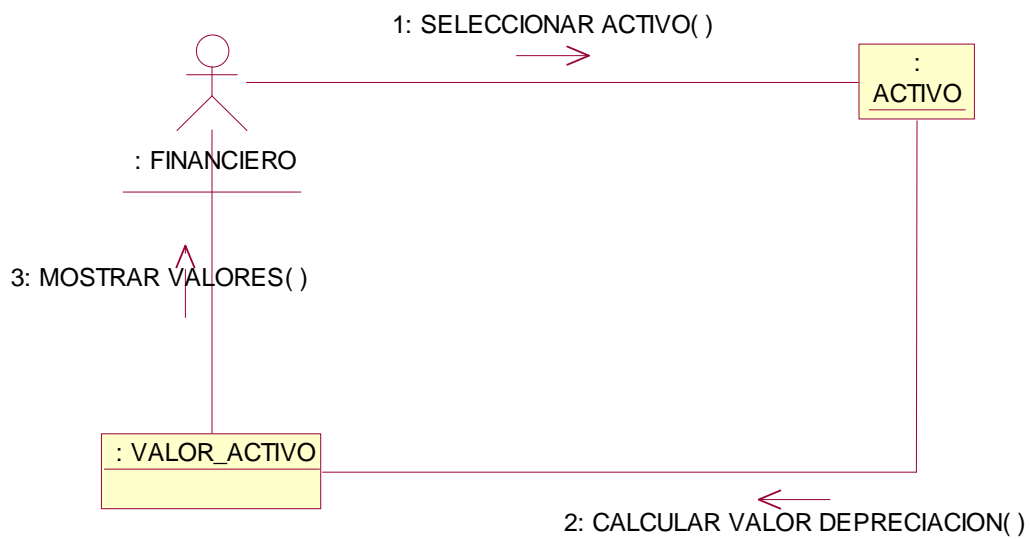


Diagrama 3.11: *Calculador de la Depreciación*

Diagrama de colaboración generado a partir del diagrama de secuencia Calculador de la Depreciación por la herramienta Rational Rose, que resalta la organización estructural de los objetos Activo y Valor Activo que intercambian mensajes.

3.3.1.11 DIAGRAMA DE SECUENCIA - Gestor de Activos Fijos

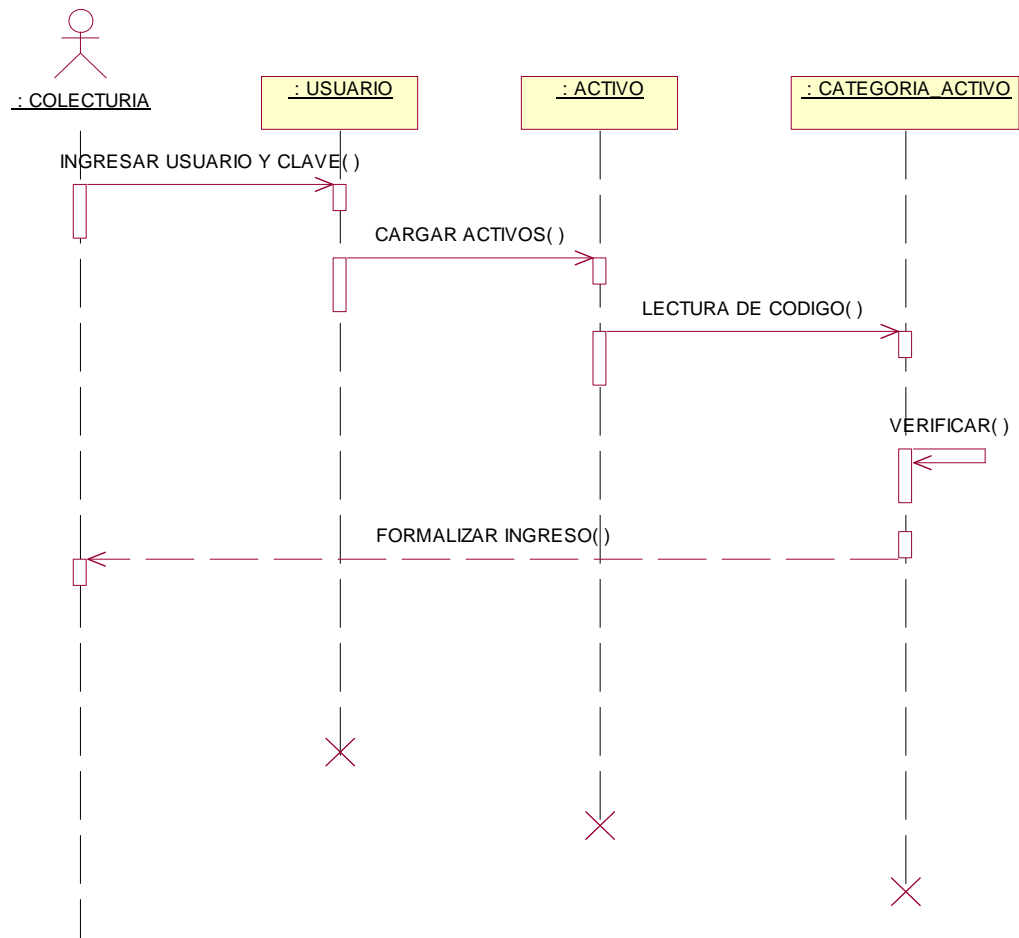


Diagrama 3.12: Gestor de Activos Fijos

Diagrama de Secuencia Gestor de Activos Fijos que intercambia mensajes entre los objetos Usuario, Activo y Categoría Activo, que una vez ingresado el usuario, permite cargar los activos, leer el código del activo, verificar, y formalizar un ingreso.

3.3.1.12 DIAGRAMA DE COLABORACION - Gestor de Activos Fijos

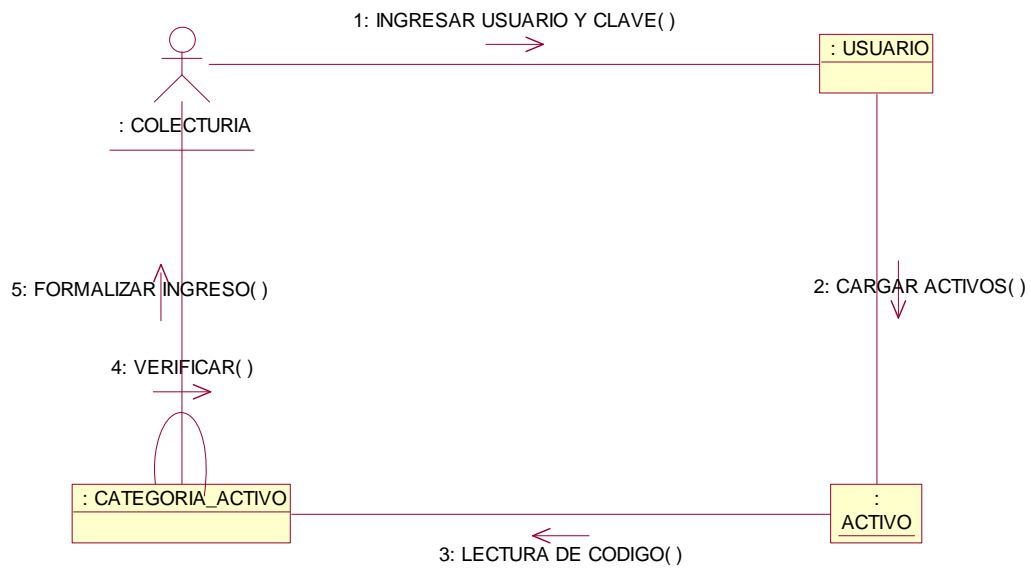


Diagrama 3.13: Gestor de Activos Fijos

Diagrama de colaboración generado a partir del diagrama de secuencia Gestor de Activos Fijos por la herramienta Rational Rose, que resalta la organización estructural de los objetos Usuario, Activo y Categoría Activo que intercambian mensajes.

3.3.1.13 DIAGRAMA DE SECUENCIA - Sincronizador de Datos

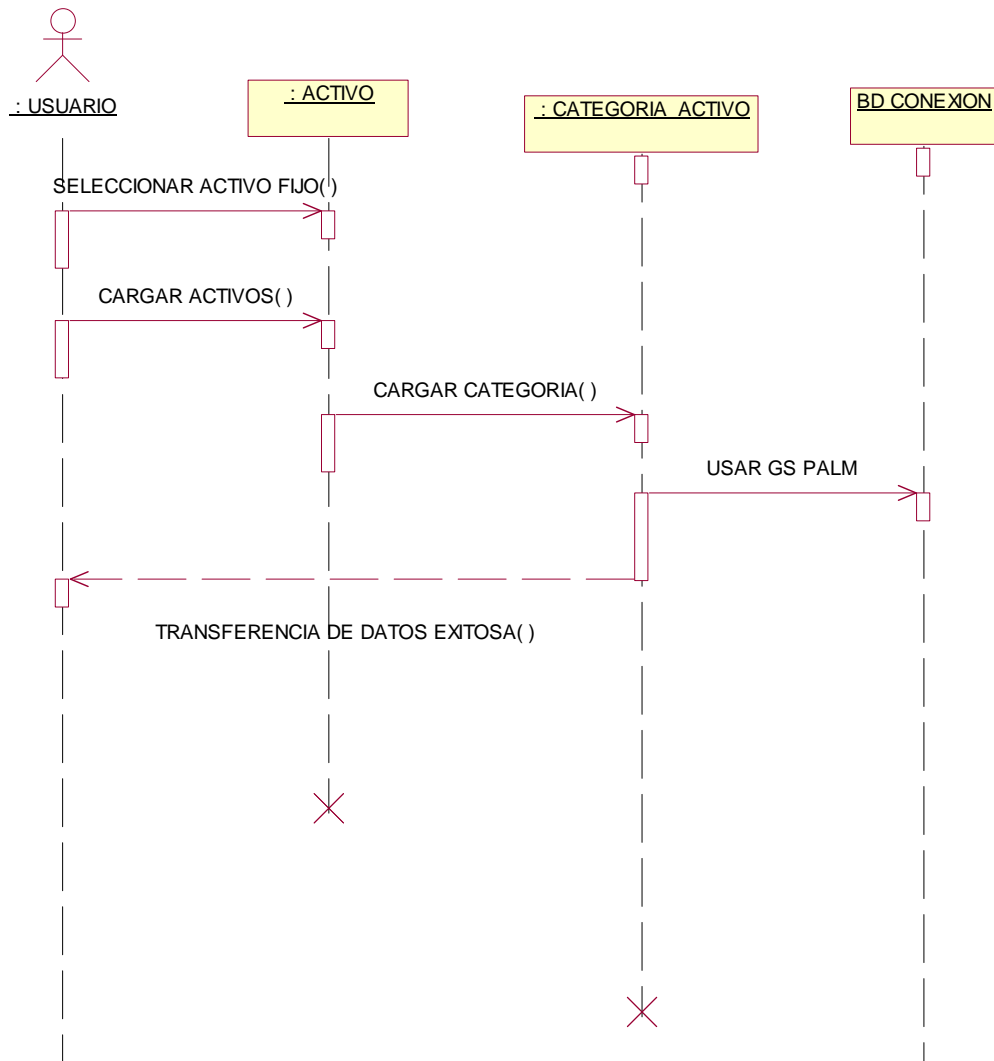


Diagrama 3.14: Sincronizador de Datos

Diagrama de Secuencia Sincronizador de Datos que intercambia mensajes entre los objetos Activo, Categoría Activo, y BD Conexión, que permite seleccionar un activo, Cargar el activo, Cargar la Categoría, utilizar la GS Palm, enviando la respuesta de transferencia exitosa.

3.3.1.14 DIAGRAMA DE COLABORACION - Sincronizador de Datos

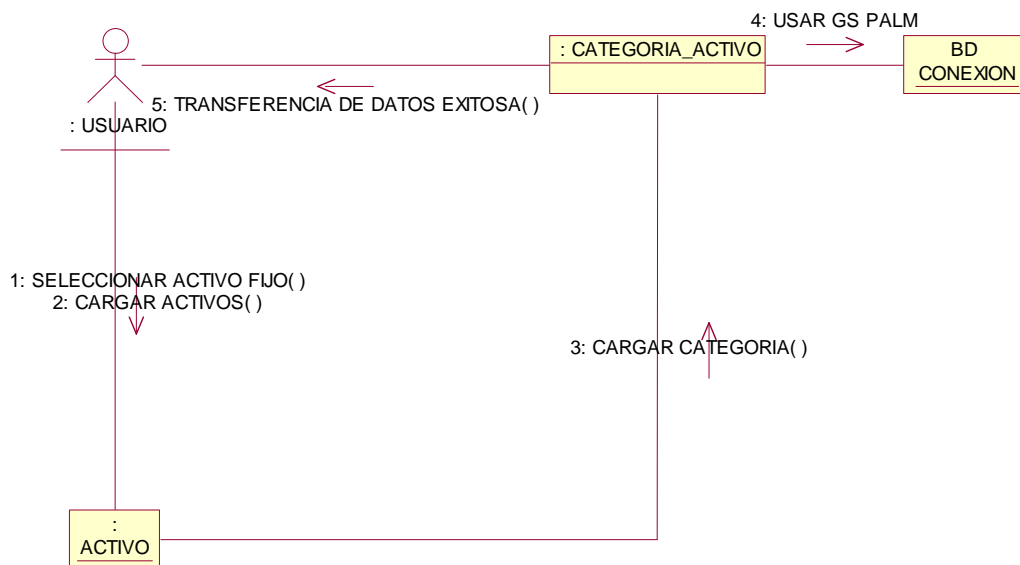


Diagrama 3.15: Sincronizador de Datos

Diagrama de colaboración generado a partir del diagrama de secuencia Sincronizador de Datos por la herramienta Rational Rose, que resalta la organización estructural de los objetos Activo, Categoría Activo y BD Conexión que intercambian mensajes.

3.3.2 DIAGRAMAS DE ESTADO

3.3.2.1 DIAGRAMA DE ESTADOS DEL OBJETO: Usuario

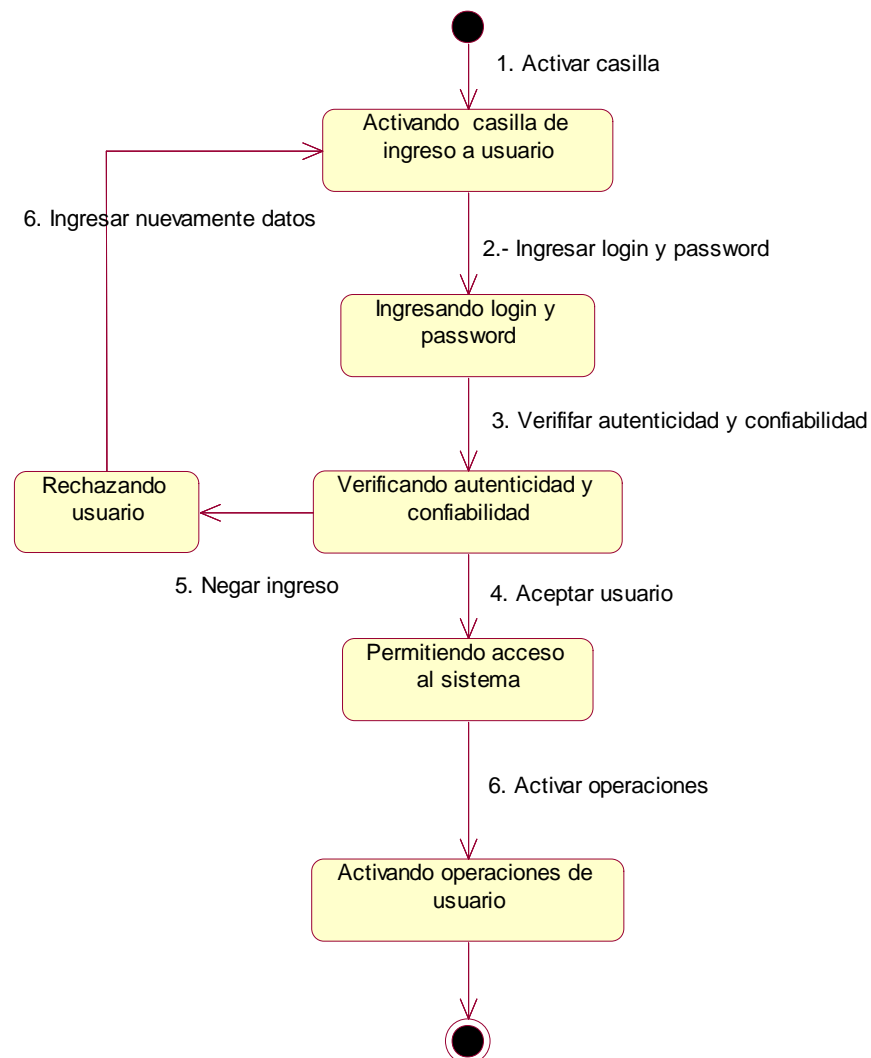


Diagrama 3.16: Usuario

Diagrama de Estado del Objeto Usuario, que permite ver el flujo de la información con el objeto usuario. Así, si la información no es la correcta para el ingreso del usuario, este debe reingresar los datos, para que pueda ingresar exitosamente, y se le activen las operaciones de usuario.

3.3.2.2 DIAGRAMA DE ESTADOS DEL OBJETO: Categoría

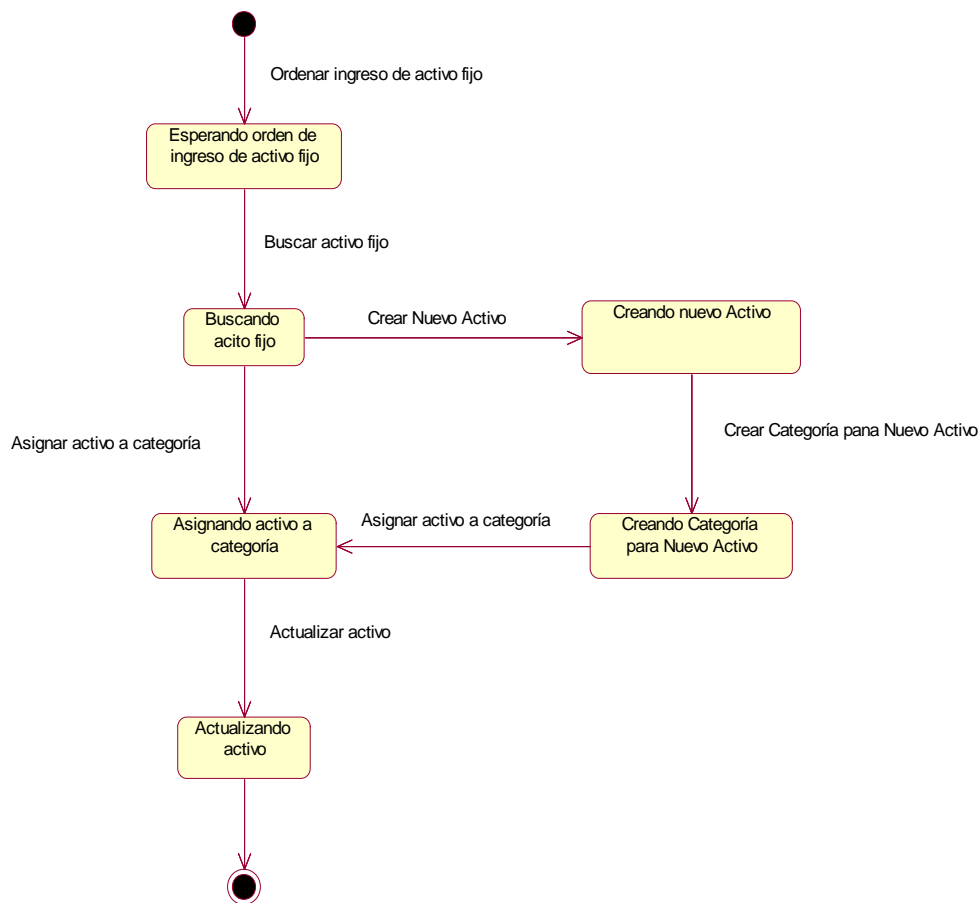


Diagrama 3.17: Categoría

Diagrama de Estado del Objeto Categoría, que permite ver el flujo de la información con el objeto categoría. Así, una vez que se busca un activo fijo, si este no ha sido ingresado se solicita el ingreso y creación de un nuevo activo y su respectiva categoría, de lo contrario, el curso normal de la información sería que se asigne la categoría al activo, y se realice la actualización del activo.

3.3.2.3 DIAGRAMA DE ESTADOS DEL OBJETO: Activo

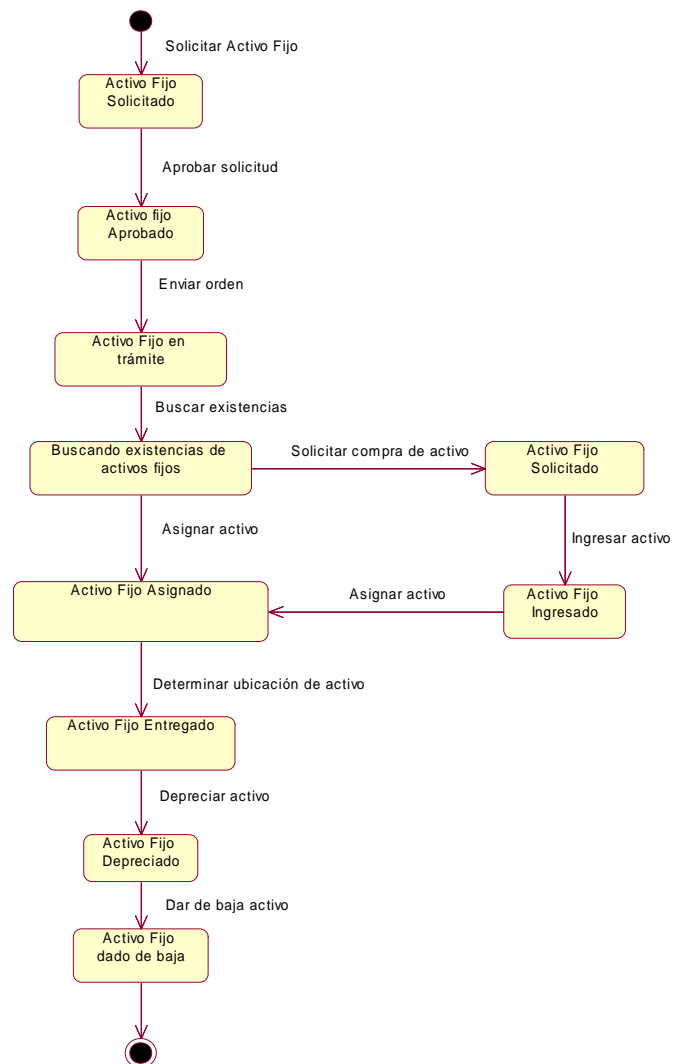


Diagrama 3.18: Activo

Diagrama de Estado del Objeto Activo, que permite ver el flujo de la información con el objeto Activo. Si se solicita un activo Fijo, se aprueba la solicitud, y el Activo Fijo entra en proceso de trámite, hasta verificar existencias, y si no fuera el caso, realizar una solicitud de compra, y el respectivo ingreso, una vez realizado este paso, se puede proceder a asignar el Activo Fijo, determinar la ubicación, y legalizar la entrega del activo, en este punto el activo fijo, también se puede depreciar, y ser dado de baja.

3.3.3 DIAGRAMAS DE ACTIVIDAD

3.3.3.1 DIAGRAMA DE ACTIVIDAD: Validador de la Seguridad

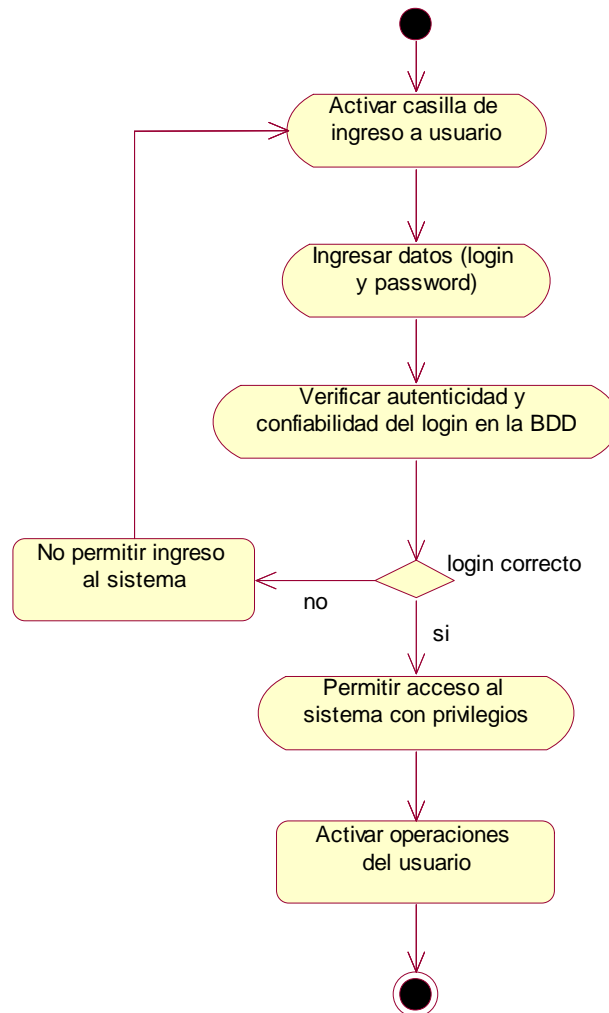


Diagrama 3.19: Validador de la Seguridad

Diagrama de Actividad del Validador de la seguridad, que permite ver el flujo de la información para el ingreso al sistema. Así, si la información no es la correcta para el ingreso del usuario, este debe reingresar los datos, para que pueda ingresar exitosamente, y se le activen las operaciones de usuario.

3.3.3.2 DIAGRAMA DE ACTIVIDAD: Administrador de usuario

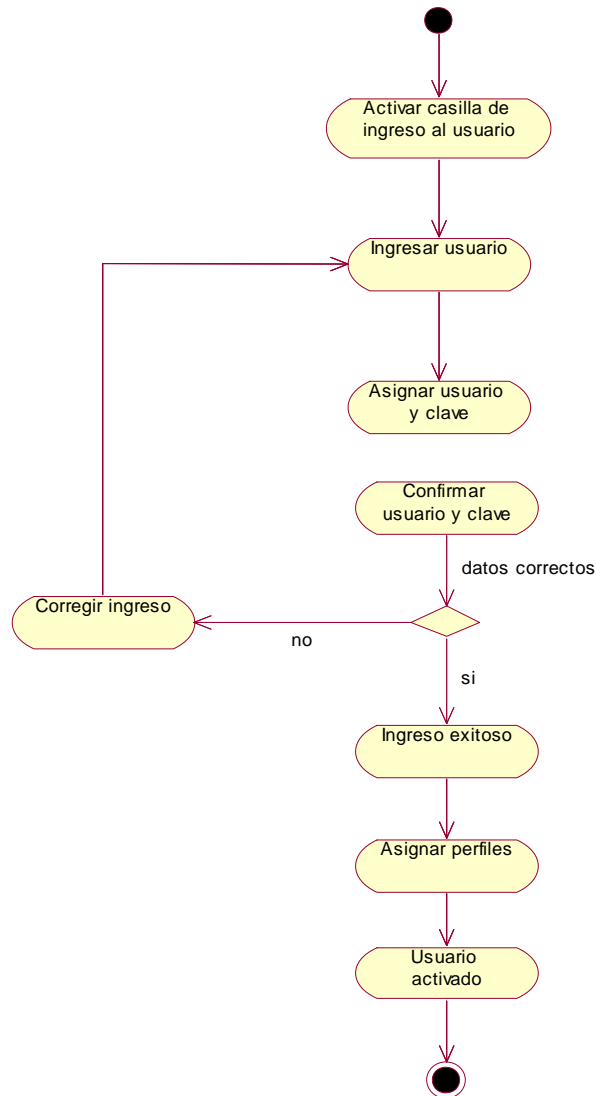


Diagrama 3.20: Administrador de Usuario

Diagrama de Actividad Administrador de Usuario, que permite ver el flujo de la información para la creación de un usuario y los perfiles a ser asignados para el mismo el ingreso al sistema.

3.3.3.3 DIAGRAMA DE ACTIVIDAD: Administrador de Activos Fijos

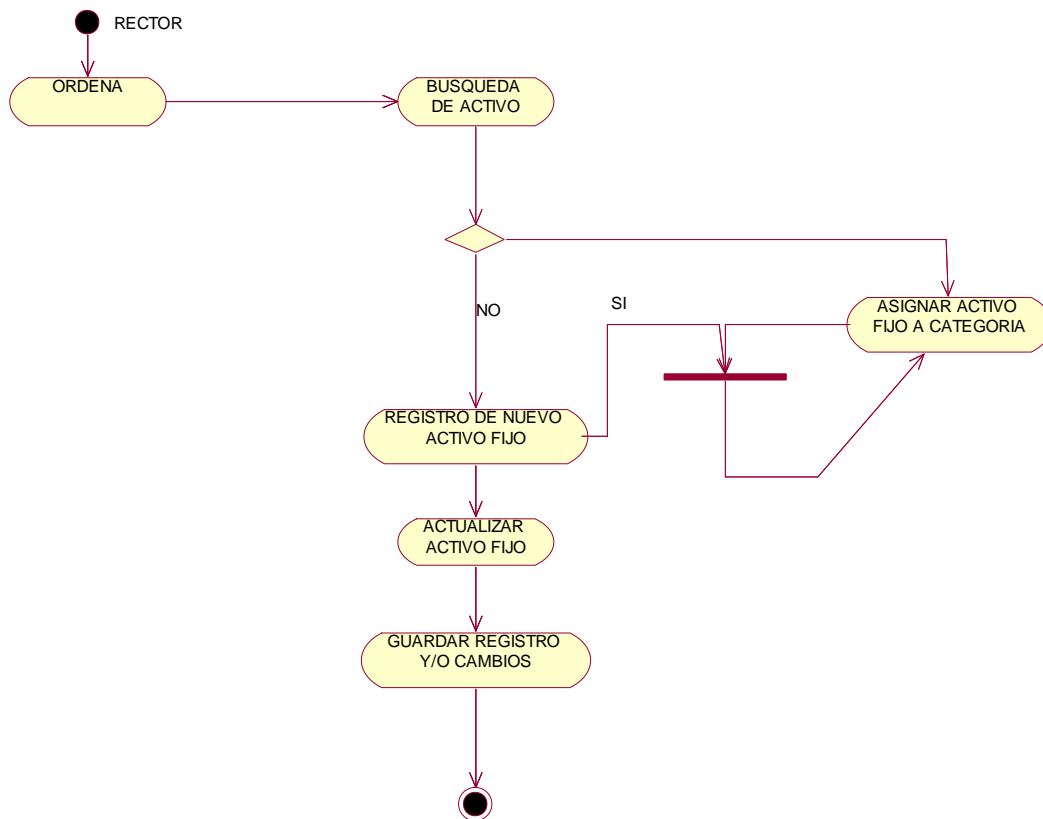


Diagrama 3.21: *Administrador de Activos Fijos*

Diagrama de Actividad del Validador de la seguridad, que permite ver el flujo de la información para el ingreso al sistema. Así, si la información no es la correcta para el ingreso del usuario, este debe reingresar los datos, para que pueda ingresar exitosamente, y se le activen las operaciones de usuario.

3.3.3.4 Diagrama de Actividad: Organizador de Activos Fijos

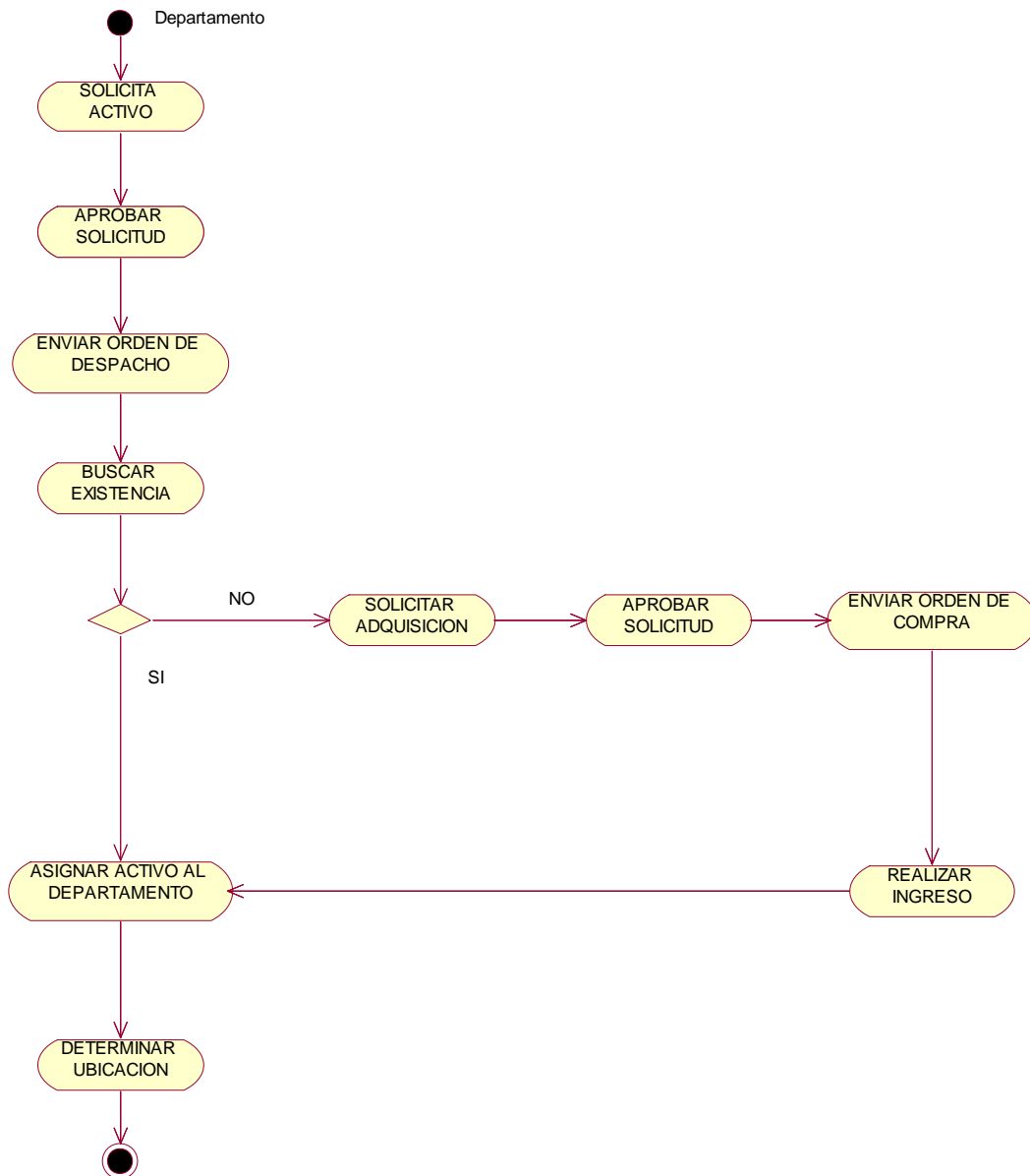


Diagrama 3.22: Organizador de Activos Fijos

3.3.3.5 DIAGRAMA DE ACTIVIDAD: Calculador de la depreciación

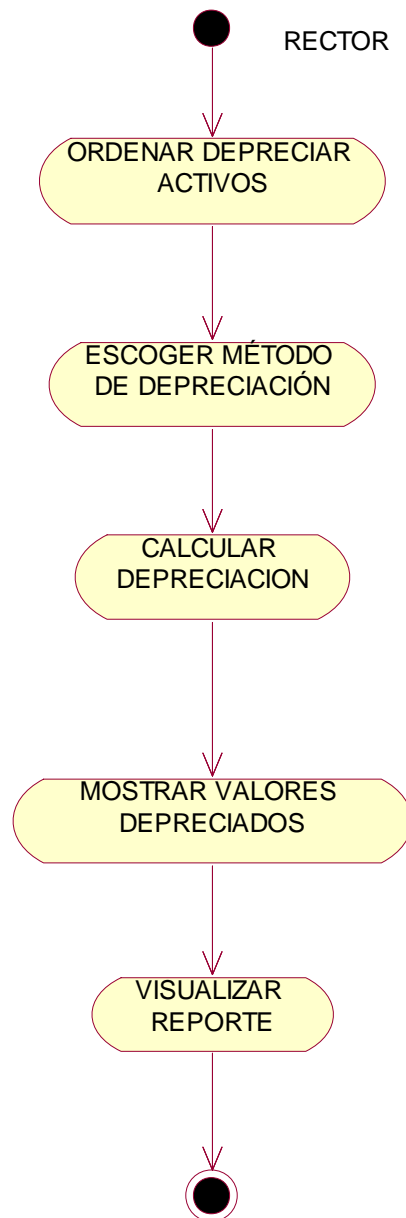


Diagrama 3.23: Calculador de la Depreciación

3.4 MODELO FUNCIONAL

3.4.1 DIAGRAMAS DE CASOS DE USO

3.4.1.1 MODELO FUNCIONAL: Perspectiva del Producto

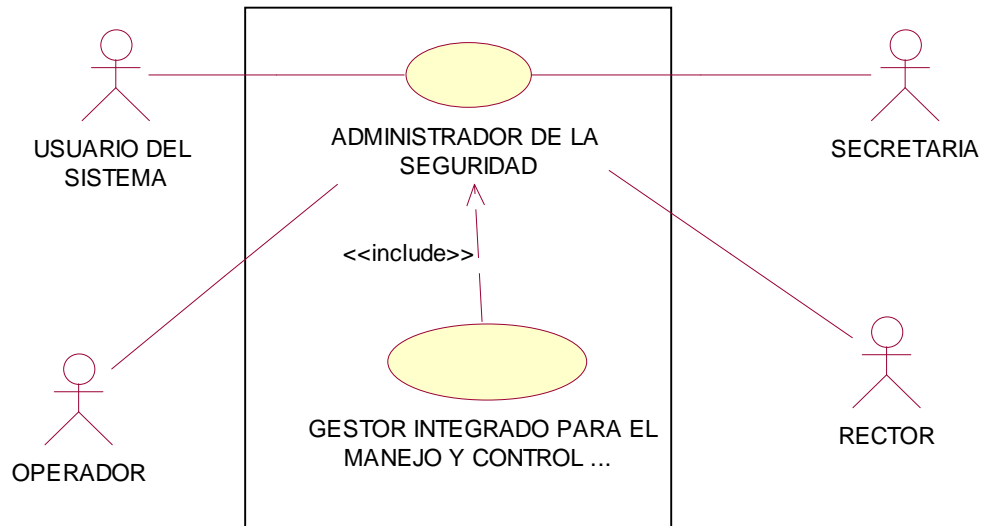


Diagrama 3.24: *Perspectiva del Producto*

3.4.1.2 MODELO FUNCIONAL: Validador de la Seguridad

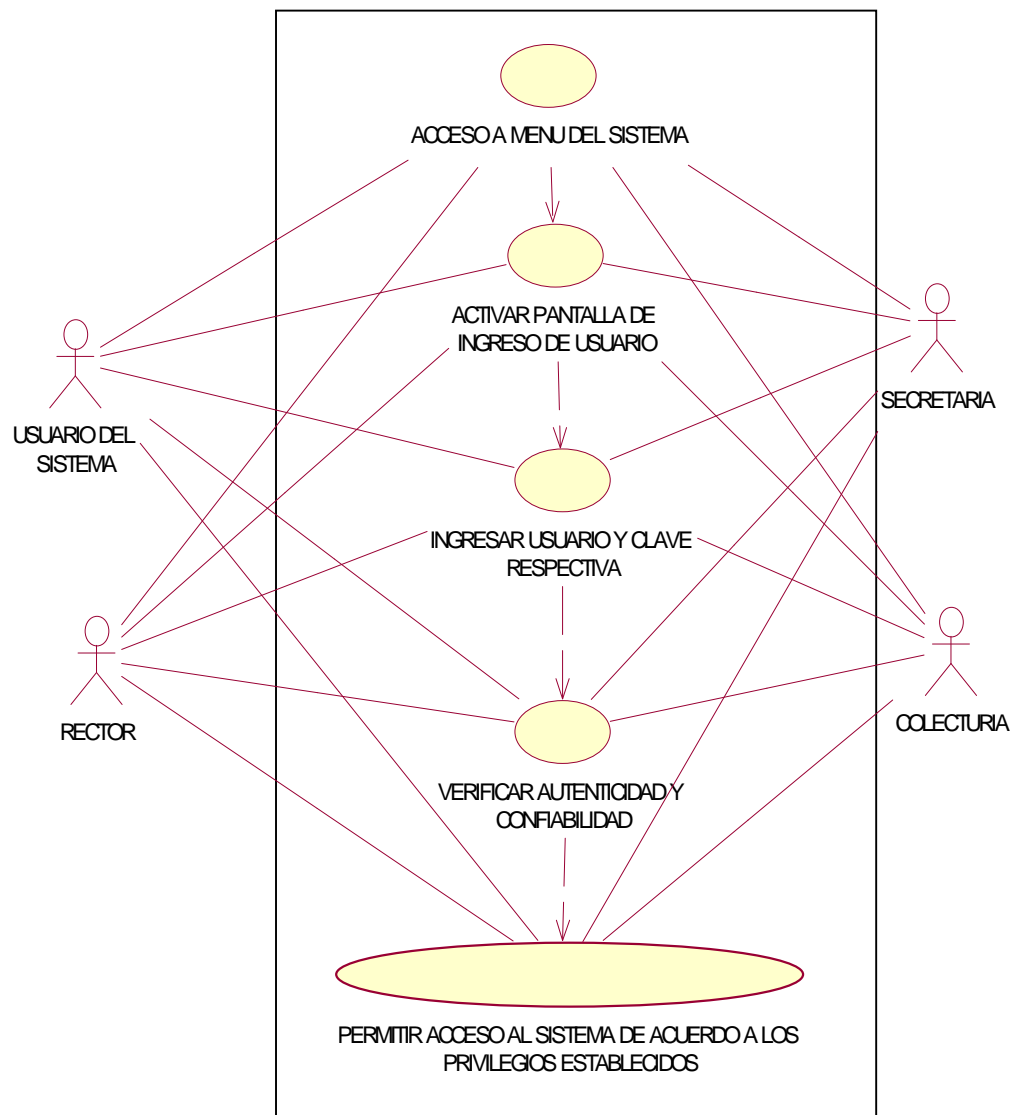


Diagrama 3.25: Validador de la Seguridad

3.4.1.3 MODELO FUNCIONAL: Administrador de Usuario

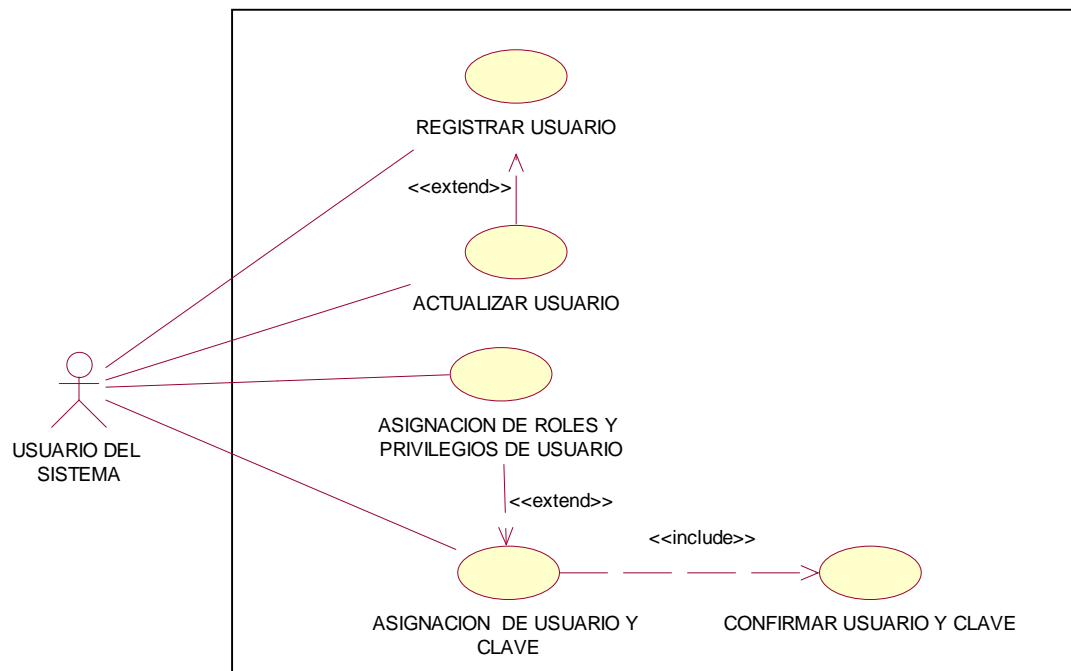


Diagrama 3.26: Administrador de Usuario

3.4.1.4 MODELO FUNCIONAL: Administrador de Activos Fijos

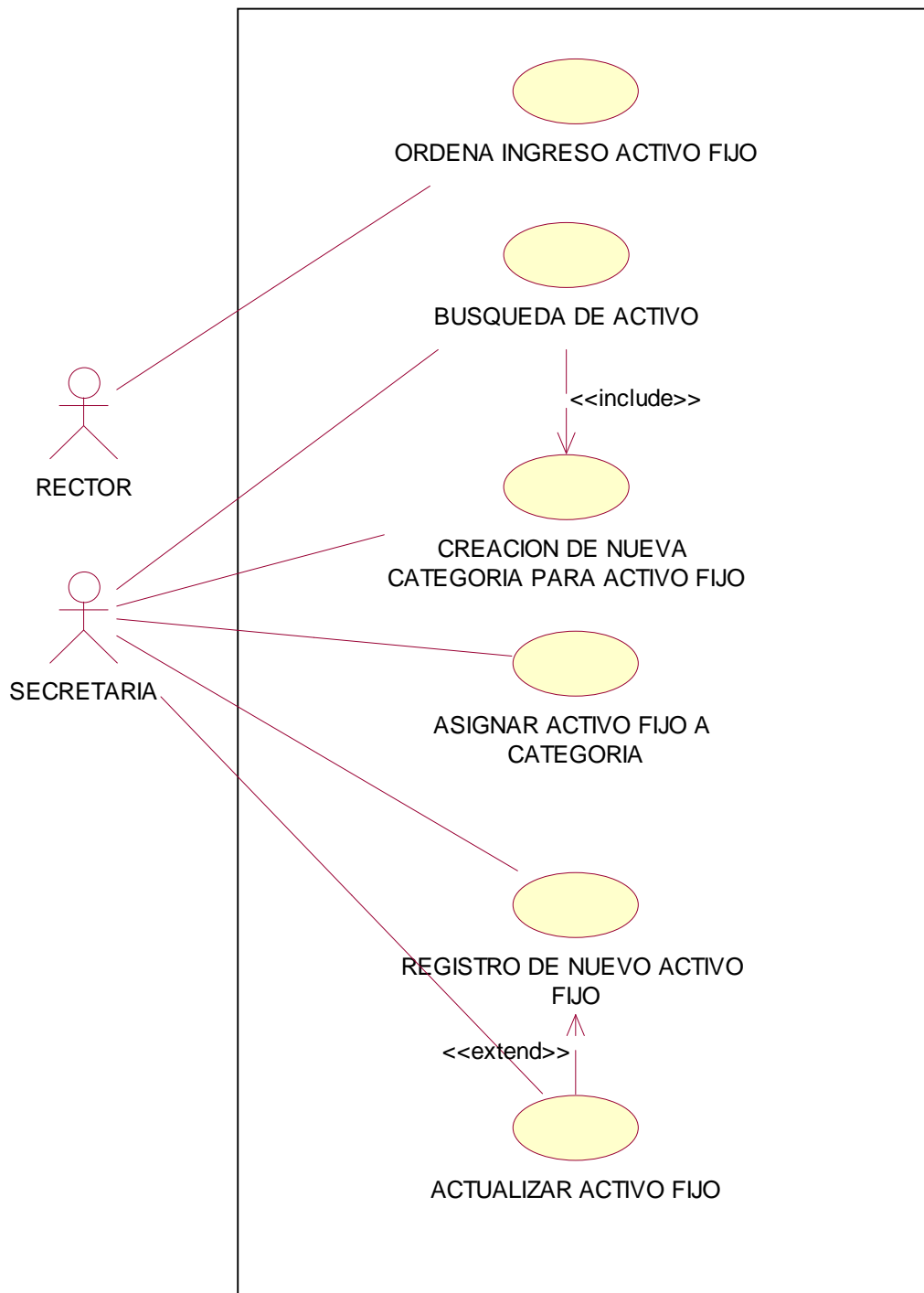


Diagrama 3.27: *Administrador de Activos Fijos*

3.4.1.5 MODELO FUNCIONAL: Organizador de Activos Fijos

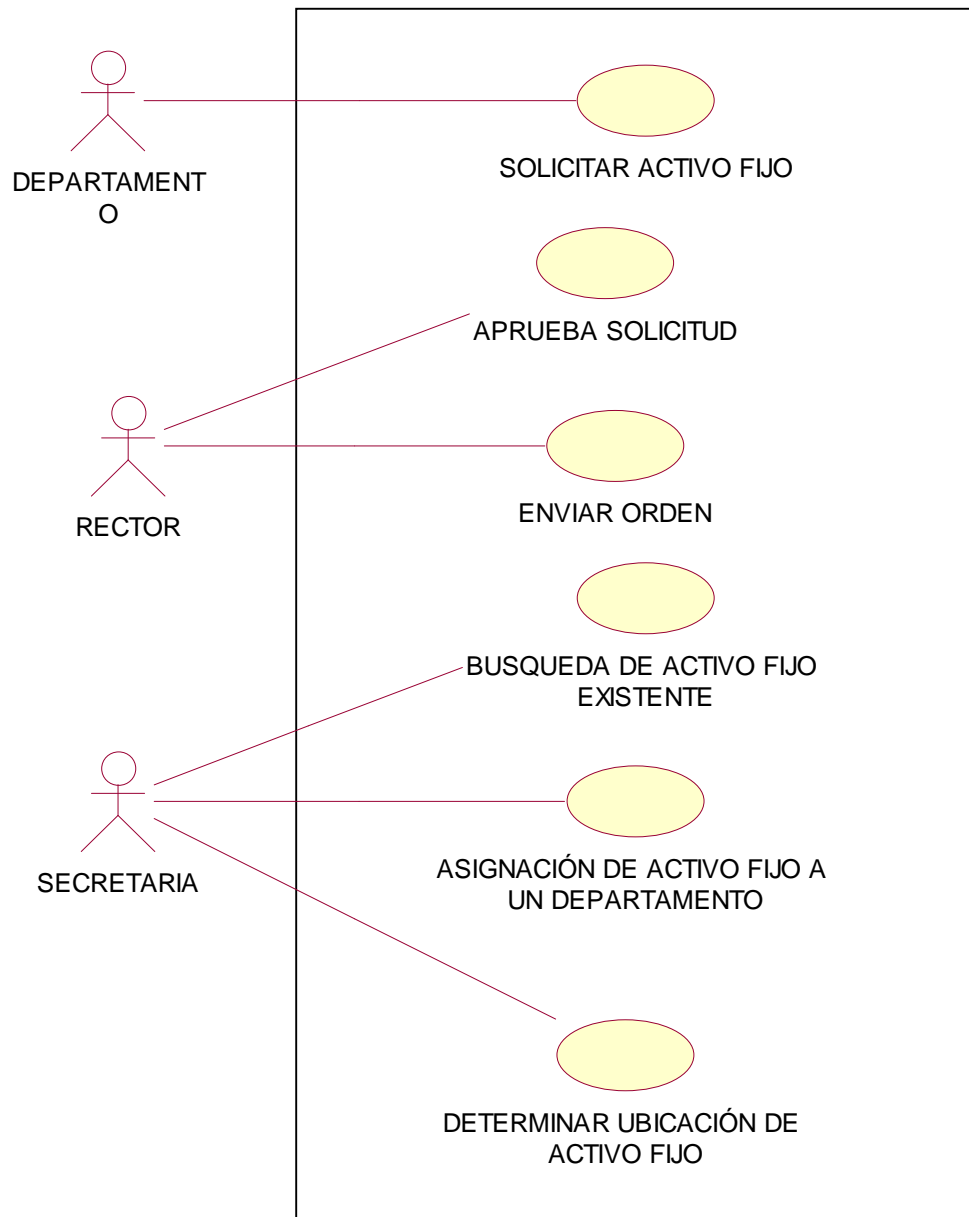


Diagrama 3.28: Organizador de Activos Fijos

3.4.1.6 MODELO FUNCIONAL: Calculador de la Depreciación

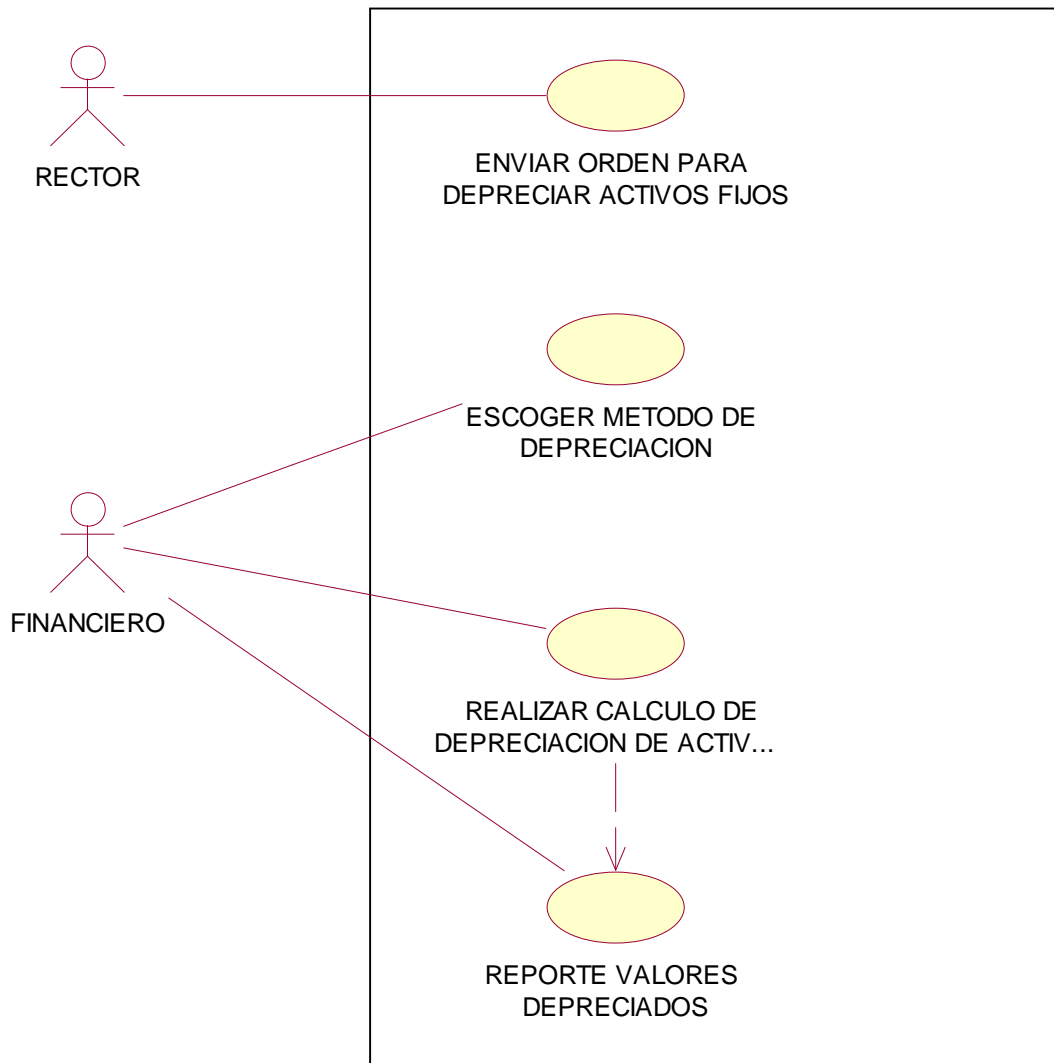


Diagrama 3.29: Calculador de la Depreciación

3.4.1.7 MODELO FUNCIONAL: Gestor de Activos Fijos (Aplicación PALM)

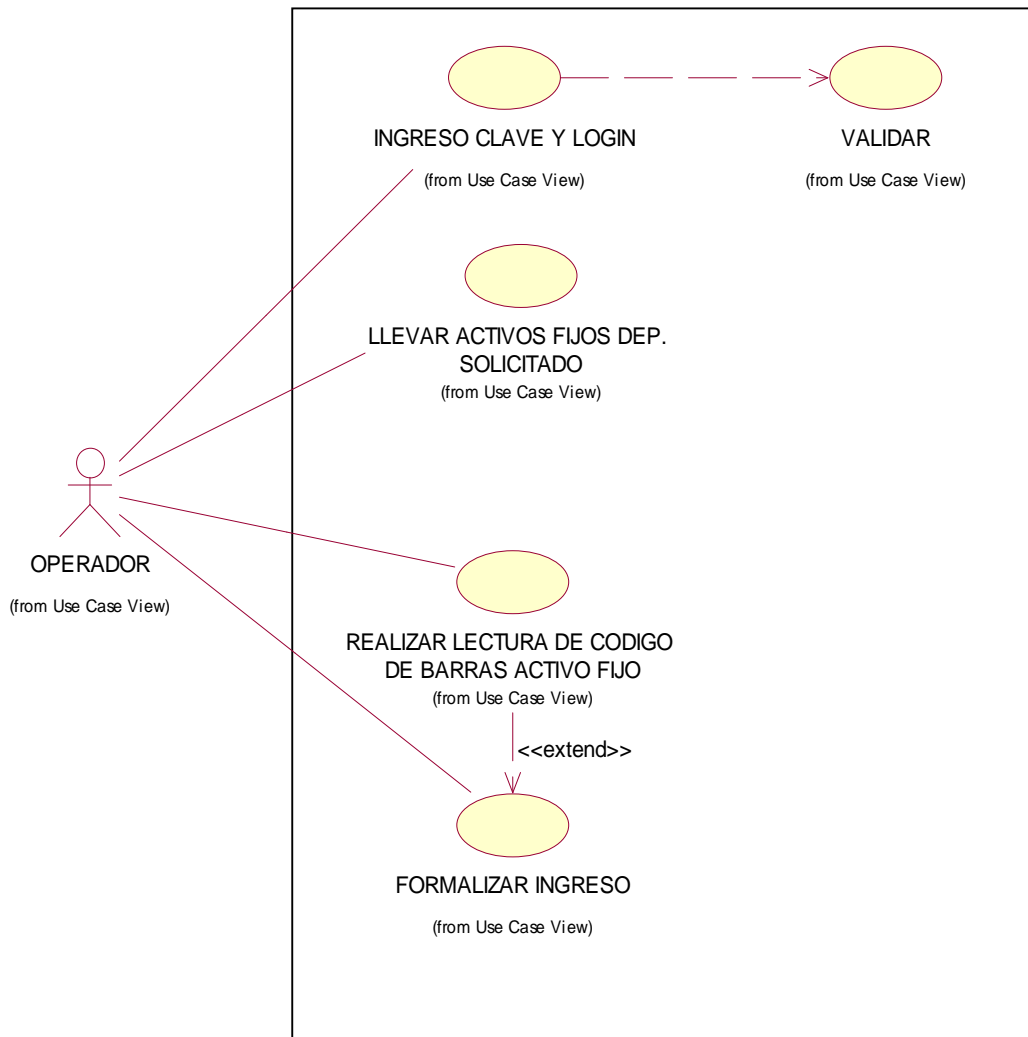


Diagrama 3.30: Gestor de Activos Fijos

3.4.1.8 MODELO FUNCIONAL: Sincronizador de Datos

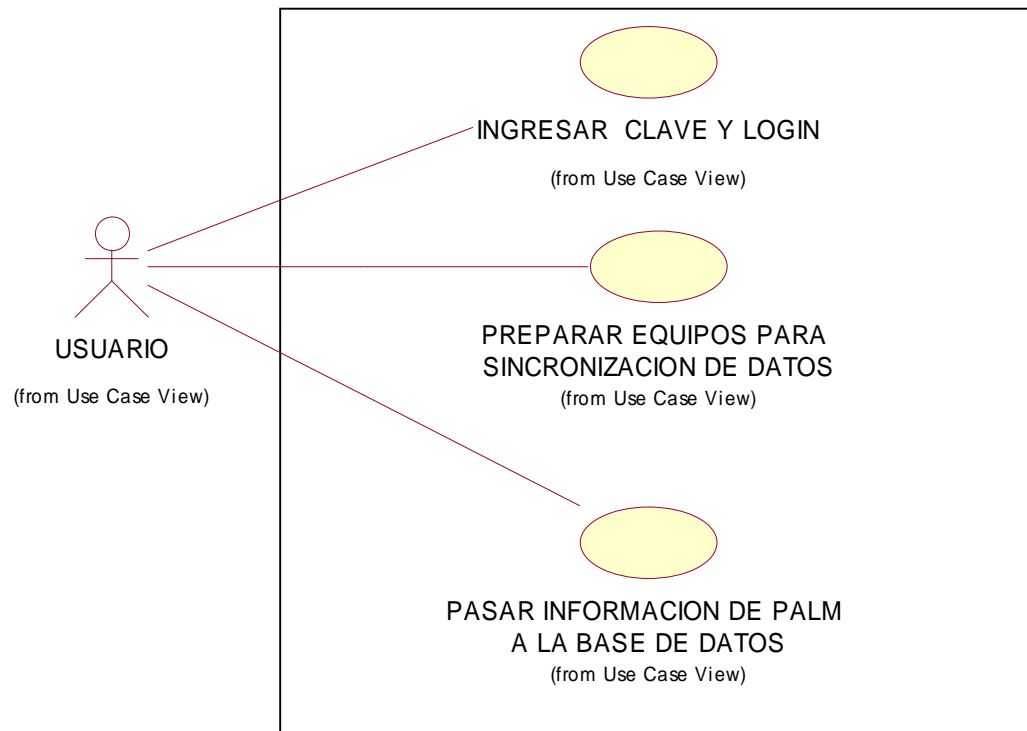


Diagrama 3.31: Sincronizador de Datos

CAPITULO 4.

IMPLEMENTACION Y PRUEBAS

4.1 HERRAMIENTAS DE IMPLEMENTACION

4.1.1 PUNTONET FRAMEWORK

.NET Framework constituye la plataforma y elemento principal sobre el que se asienta Microsoft .NET, es la pieza fundamental de todo este nuevo modelo de trabajo ya que proporciona las herramientas y servicios que necesitará en su labor habitual de desarrollo.

Funcionamiento del .NET Framework.- El Framework de .NET tiene dos componentes principales: el lenguaje común de corrida y la librería de clases.

Lenguaje común de corrida.- Cuando se compila una aplicación en .NET esta es traducida a un lenguaje común de corrida, llamado lenguaje intermedio Microsoft o MSIL²⁸, una vez compilado el ejecutor del lenguaje común administra la ejecución de la aplicación, este incluye la característica de compilación JIT²⁹, que traduce el lenguaje intermedio a lenguaje de máquina.

Librería de Clases.- Es una colección de clases reusable orientada a objetos, estas están organizadas por su funcionalidad.

4.1.1.1 ARQUITECTURA DEL .NET FRAMEWORK

El Framework esta compuesto por una infraestructura de componentes los cuales detallamos a continuación.

²⁸ **MSIL.-** Lenguaje Intermedio Microsoft Workshop Course 2559: Introducción a la programación con Visual Basic .NET utilizando Microsoft .NET

²⁹ **JIT.-** Just in Time (Justo a Tiempo) Workshop Course 2559: Introducción a la programación con Visual Basic .NET utilizando Microsoft .NET

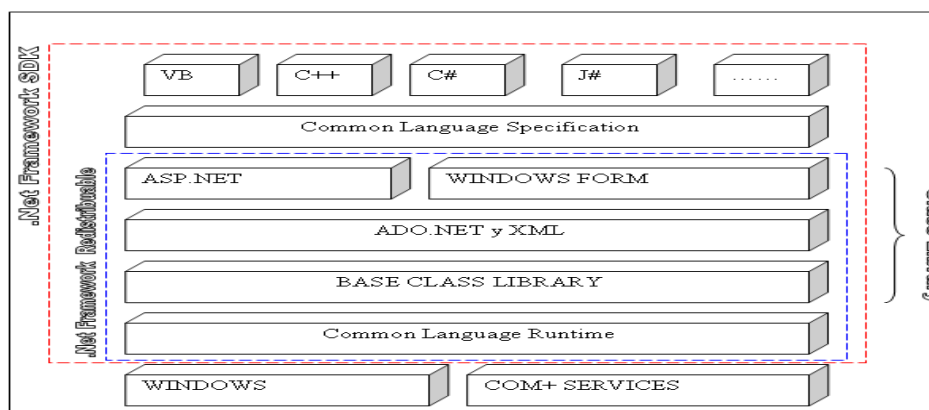


Figura 4.1: Arquitectura del Framework³⁰

Common Language Specification.- Es un conjunto de características en común que deben realizar todos los lenguajes de la plataforma para que puedan integrarse entre sí.

ASP.NET.- Cargar el entorno de un proceso preparado para gestionarse en la Web.

Windows Forms.- Contiene las clases necesarias para crear aplicaciones basadas en formularios y ventanas de Windows.

ADO.NET y XML.- Incluye una serie de mejoras y optimizaciones muy interesantes, que mejoran la experiencia de desarrollo y ejecución de aplicaciones que interactúan con base de datos. ADO.NET utiliza XML como formato universal de transmisión de datos.

Base Class Library.- Conjunto de clases reusables organizadas por su funcionalidad.

CLR (Common Language Runtime).- Representa el alma del .NET Framework y realiza la ejecución del código de las aplicaciones. Las características de este componente son:

³⁰ **Fuente:** Workshop Course 2559: Introducción a la programación con Visual Basic .NET utilizando Microsoft .NET

- Proporciona un desarrollo de aplicaciones más sencillo, gracias a que ya vienen implementadas funcionalidades que antes el programador tenía que desarrollar.
- Administra el código en tiempo de ejecución, es todo lo referente a la carga, disposición de memoria, recuperación de memoria no utilizada a través del recolector de memoria.
- Implementa características de Gestión a bajo nivel.
- Proporciona un sistema común de tipos para los lenguajes de entorno (CTS³¹)
- Gestiona la seguridad del código que es ejecutado.
- Facilita enormemente la distribución e instalación de aplicaciones.

4.1.1.2 BENEFICIOS DEL FRAMEWORK

- Se basa en estándares y prácticas Web.
- El Framework soporta todas las tecnologías de Internet.
- Ser extensible, la jerarquía del Framework no está oculta para el programador, es posible acceder y extender las clases de .net utilizando las inherencias.
- Ser de fácil uso para el desarrollador.

4.1.2 VISUAL STUDIO .NET

Visual Studio .NET es un ambiente integrado de desarrollo que ayuda a diseñar, desarrollar, eliminar errores y ejecutar de manera rápida soluciones basadas en .NET. Puede tener acceso a un conjunto común de herramientas, diseñadores y editores de cualquier lenguaje de programación de Visual Studio .NET.

4.1.2.1 VISUAL BASIC .NET

Visual Basic forma parte de las herramientas de programación de Visual Studio .NET. Visual Basic proporciona una herramienta más productiva y poderosa para

³¹ **CTS.-** (Common Type System) Sistema común de tipos, que permite definir el modo en que los tipos serán creados y manipulados por el entorno de ejecución de .NET Framework Workshop Course 2559: Introducción a la programación con Visual Basic .NET utilizando Microsoft .NET

asumir los retos actuales en el desarrollo de aplicaciones, en la cual se crea aplicaciones interactivas para Windows, Internet y para dispositivos móviles.

Visual Basic ofrece las siguientes características:

- Capacidades de diseño completamente orientado a objetos.
- Sub-procesamiento libre y acceso directo a Microsoft.Net Framework.
- Se integra con otros lenguajes de Microsoft de Visual Studio.NET.
- Mejora la seguridad de tipos.

4.1.2.2 REFLEXIONES EN .NET

La *reflexión* es uno de los pilares de .NET. Esta característica permite almacenar y obtener información en tiempo de ejecución sobre casi cualquier objeto o tipo presente en un módulo. Es gracias a esto que es posible implementar técnicas fundamentales como la recolección de basura o la serialización en distintos formatos. Y aunque es cierto que la mayoría de los entornos de programación modernos proporcionan algún tipo de RTTI (*runtime time information*, el pariente pobre de la reflexión), nunca antes se había visto un uso tan extenso y generalizado de este recurso como en .NET.³²

4.1.3 SQL SERVER 2005

4.1.3.1 ¿QUE ES SQL SERVER 2005?

SQL Server 2005 es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2005 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.

El motor de datos SQL Server 2005 constituye el núcleo de esta solución de administración de datos empresariales. Asimismo, SQL Server 2005 combina lo mejor en análisis, información, integración y notificación. Esto permite que su negocio cree y despliegue soluciones de BI rentables que ayuden a su equipo a

³² <https://www.marteens.com/trick56.htm>

incorporar datos en cada rincón del negocio a través de tableros de comando, escritorios digitales, servicios Web y dispositivos móviles.

SQL Server 2005 ofrece una plataforma de datos más confiable, segura y productiva para aplicaciones de unidad de negocios y analíticas.

- **Productividad del encargado del desarrollo:** SQL Server 2005 brinda un entorno de desarrollo de extremo a extremo que incluye diversas tecnologías nuevas que otorgan poder a los encargados del desarrollo y aumentan considerablemente la productividad del encargado del desarrollo.
- **Inteligencia empresarial:** Las capacidades globales analíticas, de integración y migración de datos de SQL Server 2005 permiten que las compañías amplíen el valor de sus aplicaciones existentes, sin perjuicio de la plataforma subyacente. Las soluciones de BI creadas en SQL Server 2005 ofrecen información crítica y oportuna a todos los empleados, permitiéndoles tomar mejores decisiones más rápidamente.

Hoy en día las organizaciones enfrentan numerosos desafíos de datos, tales como la necesidad de tomar decisiones más rápidas y más orientadas a datos, la necesidad de aumentar la productividad y flexibilidad del personal de desarrollo y presionan para reducir los presupuestos generales de informática (IT) a la vez que escalan la infraestructura para satisfacer las exigencias cada vez mayores.

Con la ampliación de las ventajas de SQL Server 2000, SQL Server 2005 ofrece una solución integrada de administración y análisis de datos que ayuda a las organizaciones de cualquier magnitud a realizar lo siguiente:

- Crear, desplegar y administrar aplicaciones empresariales más seguras, escalables y confiables.
- Maximizar la productividad de IT mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- Compartir datos en múltiples plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.

- Controlar los costes sin sacrificar el rendimiento, la disponibilidad, la escalabilidad o la seguridad.

4.1.3.2 PLATAFORMA DE DATOS DE SQL SERVER

SQL Server es una solución de datos: global, integrada y que habilita a los usuarios en toda la organización; mediante una plataforma más segura, confiable y productiva para datos empresariales y aplicaciones de BI. SQL Server 2005 provee herramientas sólidas y conocidas a los profesionales de IT, así como también a trabajadores de la información, reduciendo la complejidad de la creación, despliegue, administración y uso de aplicaciones analíticas y de datos empresariales en plataformas que van desde los dispositivos móviles hasta los sistemas de datos empresariales. A través de un conjunto global de características, la interoperabilidad con sistemas existentes y la automatización de tareas rutinarias, SQL Server 2005 ofrece una solución completa de datos para empresas de todos los tamaños.

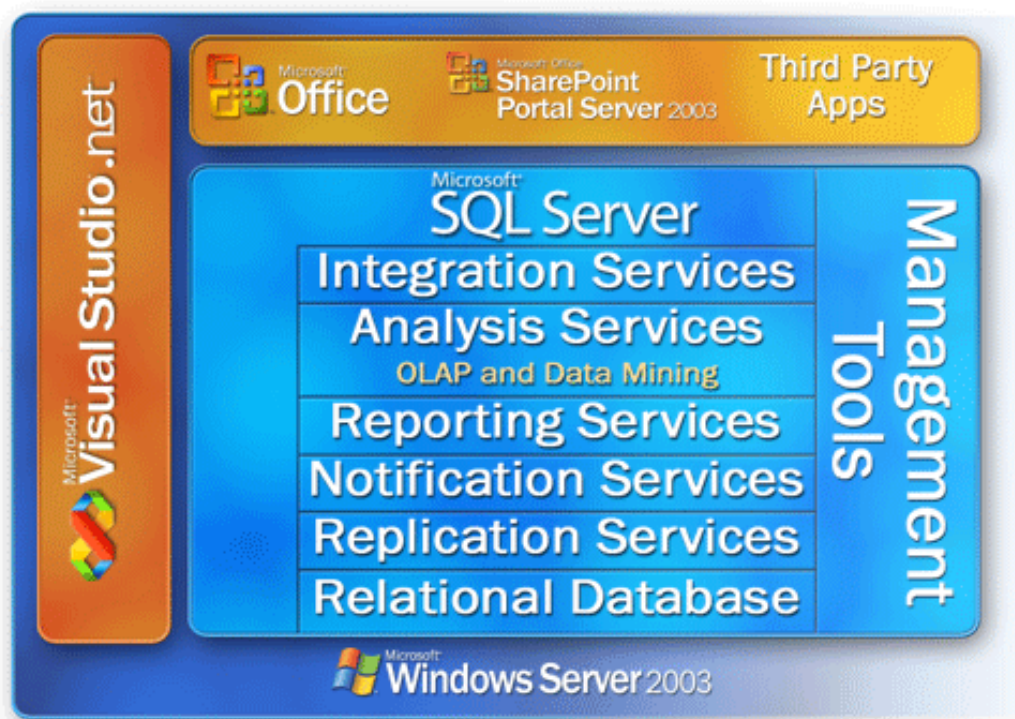


Figura 4.2: Arquitectura del Framework³³

³³ Fuente: Curso 5 estrellas Sql Server

La plataforma de datos SQL Server incluye las siguientes herramientas:

- **Base de datos relacional:** Un motor de base de datos relacional más segura, confiable, escalable y altamente disponible con mejor rendimiento y compatible para datos estructurados y sin estructura (XML).
 - **Servicios de réplica:** Réplica de datos para aplicaciones de procesamiento de datos distribuidas o móviles, alta disponibilidad de los sistemas, concurrencia escalable con almacenes de datos secundarios para soluciones de información empresarial e integración con sistemas heterogéneos, incluidas las bases de datos Oracle existentes.
 - **Notification Services:** Capacidades avanzadas de notificación para el desarrollo y el despliegue de aplicaciones escalables que pueden entregar actualizaciones de información personalizadas y oportunas a una diversidad de dispositivos conectados y móviles.
 - **Integration Services.Capacidades:** de extracción, transformación y carga (ELT) de datos para almacenamiento e integración de datos en toda la empresa.
 - **Analysis Services:** Capacidades de procesamiento analítico en línea (OLAP) para el análisis rápido y sofisticado de conjuntos de datos grandes y complejos, utilizando almacenamiento multidimensional.
 - **Reporting Services:** Una solución global para crear, administrar y proporcionar tanto informes tradicionales orientados al papel como informes interactivos basados en la Web.
 - **Herramientas de administración:** SQL Server incluye herramientas integradas para administración y optimización avanzada de bases de datos, así como también integración directa con herramientas tales como Microsoft Operations Manager (MOM) y Microsoft Systems Management Server (SMS). Los protocolos de acceso de datos estándar reducen drásticamente el tiempo que demanda integrar los datos en SQL Server con los sistemas existentes.
-

Asimismo, el soporte del servicio Web nativo está incorporado en SQL Server para garantizar la interoperabilidad con otras aplicaciones y plataformas.

- **Herramientas de desarrollo:** SQL Server ofrece herramientas integradas de desarrollo para el motor de base de datos, extracción, transformación y carga de datos, minería de datos, OLAP e informes que están directamente integrados con Microsoft Visual Studio.

4.1.4 CODIGO FUENTE

4.1.4.1 CODIGO DE UNA CLASE IMPLEMENTADA

```

Public Class frmRegistroActivo
Public activo As New RulerCellarActive.Activo
Private Sub btnCerrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCerrar.Click
    Me.DialogResult = Windows.Forms.DialogResult.Cancel
End Sub
Private Sub frmRegistroActivo_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Me.CargarDatos()
    Me.CargarCaracteristicas()
    ' Me.UbicaValor()
    If Not activo.ID = Nothing Then
        Me.CargarDatosActivo()
    End If
End Sub
Sub CargarDatosPrincipales()
    lumoGUI.ManagerControls.GetObjetToForm(Me, activo)
End Sub
Sub CargarDatos()
    lumoGUI.ManagerControls.GetComboBox(RulerCellarActive.Categoria_Activo.TraeCategoriasActivos(Ruler
CellarActive.Estados.Activos), Me.cmbCategoria)
    lumoGUI.ManagerControls.GetComboBox(RulerCellarActive.Descripcion_Activo.TraeDescripciones("",
Me.cmbCategoria.SelectedValue), Me.Descripcion)
    lumoGUI.ManagerControls.GetComboBox(RulerCellarActive.Proveedor.TraeListaProveedores(),
Me.Proveedor)
End Sub
Sub CargarDatosActivo()
    Try
        Me.cmbCategoria.SelectedValue = New
RulerCellarActive.Descripcion_Activo(Me.activo.Descripcion).Categoria
        lumoGUI.ManagerControls.GetObjetToForm(Me, Me.activo)
        Me.Descripcion.SelectedValue = Me.activo.Descripcion
        Me.Proveedor.SelectedValue = Me.activo.Proveedor
        Me.ValorCompra.Enabled = True
        Me.FechaCompra.Enabled = True
    Catch ex As Exception
    End Try
End Sub
Private Sub btnGuardar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnGuardar.Click
    Try
        Me.CaracteristicaValor()
        Me.activo.Usuario = Aplicacion.MiUsuario.ID
        lumoGUI.ManagerControls.SetFormToObject(Me, activo)
        Me.activo.Guardar()
        MsgBox("Activo Registrado Correctamente", MsgBoxStyle.Information)
        Me.DialogResult = Windows.Forms.DialogResult.OK
    Catch ex As Exception
        MsgBox("Error al registrar el activo", MsgBoxStyle.Critical, "CellarAcive")
    End Try
End Sub
Function CaracteristicaValor(Optional ByVal idActivo As Integer = 0)
    Dim ct As Control.ControlCollection = Me.pnlCaracteristicas.Controls
    Dim listaCarac As New ArrayList
    For Each cnt As Control In ct
        If TypeOf cnt Is TextBox Then
            Me.activo.AgregarCaracteristica(cnt.Tag, cnt.Text)
        End If
        If TypeOf cnt Is ComboBox Then

```

```

        Me.activo.AgregarCaracteristica(cnt.Tag, cnt.Text)
    End If
Next
Return Nothing
End Function
Private Sub cmbCategoria_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmbCategoria.SelectedIndexChanged
    Try
        lumoGUI.ManagerControls.GetComboBox(RulerCellarActive.Descripcion_Activo.TraeDescripciones("", Me.cmbCategoria.SelectedValue), Me.Descripcion)
        Me.CargarCaracteristicas()
    Catch ex As Exception
    End Try
End Sub
Sub CargarCaracteristicas()
    Try
        Dim cat As Integer = Me.cmbCategoria.SelectedValue
        Dim caracteristicas As ArrayList = RulerCellarActive.Caracteristicas.TraeCaracteristicas(cat, RulerCellarActive.Estados.Activos)
        Me.pnlCaracteristicas.Controls.Clear()
        Dim posx As Integer = 0
        Dim posy As Integer = 10
        For Each caracteristica As RulerCellarActive.Caracteristicas In caracteristicas
            Dim lbl As New Label
            lbl.Text = caracteristica.Nombre
            lbl.AutoSize = True
            lbl.Location = New Point(posx, posy)
            Dim caractivo As New RulerCellarActive.Activo_Caracteristica(Me.activo.ID, caracteristica.ID)
            Me.pnlCaracteristicas.Controls.Add(lbl)
            If caracteristica.Tipo = 1 Then
                Dim txt As New TextBox
                txt.Name = caracteristica.Nombre
                txt.Tag = caracteristica.ID
                txt.Location = New Point(posx + 200, posy)
                txt.Size = New Size(200, 21)
                txt.Text = caractivo.Valor
                Me.pnlCaracteristicas.Controls.Add(txt)
            ElseIf caracteristica.Tipo = 2 Then
                Dim cmb As New ComboBox
                cmb.Name = caracteristica.Nombre
                cmb.Location = New Point(posx + 200, posy)
                cmb.Tag = caracteristica.ID
                cmb.Size = New Size(200, 21)
                For Each v As RulerCellarActive.Valor_Caracteristica In RulerCellarActive.Valor_Caracteristica.TraeValores(caracteristica.ID)
                    cmb.Items.Add(v.Dato)
                Next
                lumoGUI.ManagerControls.GetComboBox(RulerCellarActive.Valor_Caracteristica.TraeValores(caracteristica.ID), cmb)
                cmb.Text = caractivo.Valor
                Me.pnlCaracteristicas.Controls.Add(cmb)
            End If
            posy += 30
        Next
    Catch ex As Exception
    End Try
End Sub
Private Sub ValorCompra_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ValorCompra.TextChanged
    If Not IsNumeric(Me.ValorCompra.Text) = True Then
        Me.ValorCompra.Text = ""
    End If
End Sub
End Class

```

4.1.4.2 SCRIPT DE CREACION DE LA BASE DE DATOS

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[ESTADO_ACTIVIVO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[ESTADO_ACTIVIVO](
    [ID_ESTADO_ACTIVIVO] [int] NOT NULL,
    [NOMBRE_ESTADO_ACTIVIVO] [varchar](30) NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_727DBE3F73E544409378FD2BC4E81AF5] DEFAULT (newid()),
    CONSTRAINT [PK_ESTADO_ACTIVIVO] PRIMARY KEY CLUSTERED
(
    [ID_ESTADO_ACTIVIVO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[ESTADO_SOLICITUD_ENTREGA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[ESTADO_SOLICITUD_ENTREGA](
    [ID_ESTADO_SOLICITUD_ENTREGA] [int] NOT NULL,
    [NOMBRE_ESTADO_SOLICITUD_ENTREGA] [varchar](40) NULL,
    CONSTRAINT [PK_ESTADO_SOLICITUD_ENTREGA] PRIMARY KEY CLUSTERED
(
    [ID_ESTADO_SOLICITUD_ENTREGA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[ESTADO_UBICACION_ACTIVIVO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[ESTADO_UBICACION_ACTIVIVO](
    [ID_ESTADO_UBICACION_ACTIVIVO] [int] NOT NULL,
    [NOMBRE_ESTADO_UBICACION_ACTIVIVO] [varchar](30) NULL,
    CONSTRAINT [PK_ESTADO_UBICACION_ACTIVIVO] PRIMARY KEY CLUSTERED
(
    [ID_ESTADO_UBICACION_ACTIVIVO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[GESTOR_CODIGOS]') AND type
in (N'U'))
BEGIN
CREATE TABLE [dbo].[GESTOR_CODIGOS](
    [ID_GESTOR_CODIGO] [int] NOT NULL,
    [VALOR_GESTOR_CODIGO] [varchar](30) NULL,
    CONSTRAINT [PK_GESTOR_CODIGOS] PRIMARY KEY CLUSTERED
(
    [ID_GESTOR_CODIGO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END

```

```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[PARAMETRO]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[PARAMETRO](
    [ID_PARAMETRO] [int] NOT NULL,
    [NOMBRE_PARAMETRO] [varchar](40) NULL,
    [DESCRIPCION_PARAMETRO] [varchar](200) NULL,
    [VALOR_PARAMETRO] [varchar](40) NULL,
    CONSTRAINT [PK_PARAMETRO] PRIMARY KEY CLUSTERED
(
    [ID_PARAMETRO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[PERFIL]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PERFIL](
    [ID_PERFIL] [int] NOT NULL,
    [NOMBRE_PERFIL] [varchar](40) NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_11E44535712B4B8ABC8EE8BF553FA59AA] DEFAULT (newid()),
    CONSTRAINT [PK_PERFIL] PRIMARY KEY CLUSTERED
(
    [ID_PERFIL] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[TIPO_MOVILIZACION]') AND
type in (N'U'))
BEGIN
CREATE TABLE [dbo].[TIPO_MOVILIZACION](
    [ID_TIPO_MOVILIZACION] [int] NOT NULL,
    [NOMBRE_TIPO_MOVILIZACION] [varchar](40) NULL,
    CONSTRAINT [PK_TIPO_MOVILIZACION] PRIMARY KEY CLUSTERED
(
    [ID_TIPO_MOVILIZACION] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[UBICACION]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[UBICACION](
    [ID_UBICACION] [int] NOT NULL,
    [NOMBRE_UBICACION] [varchar](60) NULL,
    [RESPONSABLE_UBICACION] [varchar](60) NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_F6B60D5A563C4C13846A02FFC731F120] DEFAULT (newid()),
    CONSTRAINT [PK_UBICACION] PRIMARY KEY CLUSTERED
(
    [ID_UBICACION] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[EMPRESA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[EMPRESA](
    [ID_EMPRESA] [int] NOT NULL,
    [NOMBRE_EMPRESA] [varchar](50) NULL,
    [DIRECCION_EMPRESA] [varchar](50) NULL,
    [TELEFONO_EMPRESA] [varchar](50) NULL,
    [IMAGEN_EMPRESA] [image] NULL,
    CONSTRAINT [PK_EMPRESA] PRIMARY KEY CLUSTERED
(
    [ID_EMPRESA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[LLAMADA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[LLAMADA](
    [ID_TIPO_LLAMADA] [int] NOT NULL,
    [NOMBRE_LLAMADA] [varchar](50) NULL,
    CONSTRAINT [PK_LLAMADA] PRIMARY KEY CLUSTERED
(
    [ID_TIPO_LLAMADA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[SEGUROS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[SEGUROS](
    [id_seguro] [int] NOT NULL,
    [nombre_seguro] [varchar](100) NULL,
    [estado_seguro] [int] NULL,
    CONSTRAINT [PK_SEGUROS] PRIMARY KEY CLUSTERED
(
    [id_seguro] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[PROVEEDOR]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[PROVEEDOR](
    [id_proveedor] [int] NOT NULL,
    [nombre_proveedor] [varchar](50) NULL,
    [apellido_proveedor] [varchar](50) NULL,
    [cedula_proveedor] [varchar](15) NULL,
    [empresa_proveedor] [varchar](50) NULL,
    [telefono_proveedor] [varchar](50) NULL,
    [telefono2_proveedor] [varchar](50) NULL,
    [direccion_proveedor] [varchar](150) NULL,
    [celular_proveedor] [varchar](12) NULL,

```

```

        [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_62B0B8C00CDE4367B7800C674F4C3464] DEFAULT (newid()),
CONSTRAINT [PK_PROVEEDOR] PRIMARY KEY CLUSTERED
(
        [id_proveedor] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[MOTIVO_MANTENIMIENTO]')
AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[MOTIVO_MANTENIMIENTO](
        [ID_MOTIVO_MANTENIMIENTO] [int] NOT NULL,
        [NOMBRE_MOTIVO_MANTENIMIENTO] [varchar](150) NULL,
CONSTRAINT [PK_MOTIVO_MANTENIMIENTO] PRIMARY KEY CLUSTERED
(
        [ID_MOTIVO_MANTENIMIENTO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CATEGORIA_ACTIVOS]') AND
type in (N'U'))
BEGIN
CREATE TABLE [dbo].[CATEGORIA_ACTIVOS](
        [ID_CATEGORIA_ACTIVOS] [int] NOT NULL,
        [NOMBRE_CATEGORIA_ACTIVOS] [varchar](60) NULL,
        [DEPRECIACION_CATEGORIA_ACTIVOS] [decimal](8, 2) NULL,
        [ESTADO_CATEGORIA_ACTIVOS] [int] NULL,
        [OBSERVACION_CATEGORIA_ACTIVOS] [varchar](200) NULL,
        [CODIGO_CATEGORIA_ACTIVOS] [varchar](4) NULL,
        [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_53D378FC258F421C934E02839E4B8A77] DEFAULT (newid()),
        [VIDA_UTIL_CATEGORIA_ACTIVOS] [int] NULL,
CONSTRAINT [PK_CATEGORIA_ACTIVOS] PRIMARY KEY CLUSTERED
(
        [ID_CATEGORIA_ACTIVOS] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[ACTIVOS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[ACTIVOS](
        [ID_ACTIVOS] [int] NOT NULL,
        [ID_USUARIO] [int] NOT NULL,
        [ID_ESTADO_ACTIVOS] [int] NOT NULL,
        [ID_DESCRIPCION_ACTIVOS] [int] NOT NULL,
        [FECHA_COMPRA_ACTIVOS] [datetime] NULL,
        [CODIGO_ACTIVOS] [varchar](20) NULL,
        [VALOR_COMPRA_ACTIVOS] [decimal](10, 3) NULL,
        [ID_PROVEEDOR] [int] NULL,
        [DOCUMENTO_ACTIVOS] [varchar](50) NULL,
        [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_BF26B37E8A28450684CD90F3376C6FDB] DEFAULT (newid()),
CONSTRAINT [PK_ACTIVOS] PRIMARY KEY CLUSTERED
(
        [ID_ACTIVOS] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[SOLICITUD_ENTREGA]') AND
type in (N'U'))
BEGIN
CREATE TABLE [dbo].[SOLICITUD_ENTREGA](
    [ID_SOLICITUD_ENTREGA] [int] NOT NULL,
    [ID_UBICACION] [int] NOT NULL,
    [ID_ESTADO_SOLICITUD_ENTREGA] [int] NOT NULL,
    [FECHA_SOLICITUD_ENTREGA] [datetime] NULL,
    [FECHA_ENTREGA_SOLICITUD_ENTREGA] [datetime] NULL,
    [OBSERVACION_SOLICITUD_ENTREGA] [varchar](200) NULL,
    [ID_USUARIO] [int] NULL,
    CONSTRAINT [PK_SOLICITUD_ENTREGA] PRIMARY KEY CLUSTERED
(
    [ID_SOLICITUD_ENTREGA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[UBICACION_ACTIVIVO]') AND
type in (N'U'))
BEGIN
CREATE TABLE [dbo].[UBICACION_ACTIVIVO](
    [ID_UBICACION_ACTIVIVO] [int] NOT NULL,
    [ID_UBICACION] [int] NOT NULL,
    [ID_ESTADO_UBICACION_ACTIVIVO] [int] NOT NULL,
    [ID_ACTIVIVO] [int] NOT NULL,
    [ID_EMPLEADO] [int] NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_87936CA3331049E882BABB37FEB4165A] DEFAULT (newid()),
    CONSTRAINT [PK_UBICACION_ACTIVIVO] PRIMARY KEY CLUSTERED
(
    [ID_UBICACION_ACTIVIVO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[USUARIO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[USUARIO](
    [ID_USUARIO] [int] NOT NULL,
    [ID_PERFIL] [int] NOT NULL,
    [NOMBRE_USUARIO] [varchar](30) NULL,
    [NICK_USUARIO] [varchar](30) NULL,
    [CLAVE_USUARIO] [varchar](30) NULL,
    [ESTADO_USUARIO] [int] NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_0D632373377B4900AEEC76F1768B5B35] DEFAULT (newid()),
    CONSTRAINT [PK_USUARIO] PRIMARY KEY CLUSTERED
(
    [ID_USUARIO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[PERMISO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PERMISO](
    [ID_PERMISO] [int] NOT NULL,
    [NOMBRE_PERMISO] [varchar](100) NULL,
    [PAGINA_PERMISO] [varchar](100) NULL,
    [ID_AUX_PERMISO] [int] NULL,
    [IMAGEN_PERMISO] [varchar](100) NULL,
    [ID_PERFIL] [int] NULL,
    [ID_TIPO_LLAMADA] [int] NULL,
    [NOMBRE_METODO_PERMISO] [varchar](400) NULL,
    [ORDEN_PERMISO] [int] NULL,
    CONSTRAINT [PK_PERMISO] PRIMARY KEY CLUSTERED
(
    [ID_PERMISO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLE_SOLICITUD_ENTREGA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLE_SOLICITUD_ENTREGA](
    [ID_DETALLE_SOLICITUD_ENTREGA] [int] NOT NULL,
    [ID_SOLICITUD_ENTREGA] [int] NOT NULL,
    [ID_ACTIVIVO] [int] NOT NULL,
    [ESTADO_SOLICITUD_ENTREGA] [int] NULL,
    CONSTRAINT [PK_DETALLE_SOLICITUD_ENTREGA] PRIMARY KEY CLUSTERED
(
    [ID_DETALLE_SOLICITUD_ENTREGA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[HISTORIAL_ACTIVIVO]') AND type
in (N'U'))
BEGIN
CREATE TABLE [dbo].[HISTORIAL_ACTIVIVO](
    [ID_HISTORIAL_ACTIVIVO] [int] NOT NULL,
    [ID_USUARIO] [int] NOT NULL,
    [ID_ACTIVIVO] [int] NOT NULL,
    [ID_TIPO_MOVILIZACION] [int] NOT NULL,
    [FECHA_HISTORIAL_ACTIVIVO] [datetime] NULL,
    [DETALLE_HISTORIAL_ACTIVIVO] [varchar](200) NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_44EF9117E6B6471CBB7D797840D5A071] DEFAULT (newsequentialid()),
    CONSTRAINT [PK_HISTORIAL_ACTIVIVO] PRIMARY KEY CLUSTERED
(
    [ID_HISTORIAL_ACTIVIVO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[EMPLEADO]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[EMPLEADO](
    [id_empleado] [int] NOT NULL,
    [nombre_empleado] [varchar](50) NULL,

```



```

        [telefono_empleado] [varchar](50) NULL,
        [direccion_empleado] [varchar](50) NULL,
        [profesion_empleado] [varchar](50) NULL,
        [cargo_empleado] [varchar](50) NULL,
        [ext_empleado] [varchar](50) NULL,
        [estado_empleado] [int] NULL,
        [id_ubicacion] [int] NULL,
        [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_4366CEFBFAE345A3852225039532372C] DEFAULT (newid()),
CONSTRAINT [PK_EMPLEADO] PRIMARY KEY CLUSTERED
(
        [id_empleado] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[PERMISO_USUARIO]') AND
type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PERMISO_USUARIO](
        [ID_PERMISO_USUARIO] [int] NOT NULL,
        [ID_USUARIO] [int] NULL,
        [ID_PERMISO] [int] NULL,
        [ACTIVO_PERMISO_USUARIO] [int] NOT NULL,
        CONSTRAINT [PK_PERMISO_USUARIO] PRIMARY KEY NONCLUSTERED
(
        [ID_PERMISO_USUARIO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[TRASPASO]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[TRASPASO](
        [ID_TRASPASO] [int] NULL,
        [ID_ACTIVO] [int] NULL,
        [ID_UBICACION_ORIGEN] [int] NULL,
        [ID_UBICACION_DESTINO] [int] NULL,
        [FECHA_TRASPASO] [datetime] NULL,
        [OBSERVACION_TRASPASO] [varchar](100) NULL,
        [ID_USUARIO] [int] NULL
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[VALOR_ACTIVO]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[VALOR_ACTIVO](
        [ID_VALOR_ACTIVO] [int] NOT NULL,
        [ID_ACTIVO] [int] NOT NULL,
        [VALOR_RESIDUAL_ACTIVO] [decimal](8, 2) NULL,
        [DEPRECIACION_UNIDAD_ACTIVO] [decimal](8, 2) NULL,
        [DEPRECIACION_ANUAL_ACTIVO] [decimal](8, 2) NULL,
        CONSTRAINT [PK_VALOR_ACTIVO] PRIMARY KEY CLUSTERED
(
        [ID_VALOR_ACTIVO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END

```

```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[ACTIVO_BAJA]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[ACTIVO_BAJA](
    [ID_ACTIVO_BAJA] [int] NOT NULL,
    [FECHA_ACTIVO_BAJA] [datetime] NULL,
    [ID_USUARIO] [int] NULL,
    [ID_ACTIVO] [int] NULL,
    [MOTIVO_ACTIVO_BAJA] [varchar](150) NULL,
    CONSTRAINT [PK_ACTIVO_BAJA] PRIMARY KEY CLUSTERED
(
    [ID_ACTIVO_BAJA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[ACTIVO_SEGURO]') AND type
in (N'U'))
BEGIN
CREATE TABLE [dbo].[ACTIVO_SEGURO](
    [id_activo_seguro] [int] NOT NULL,
    [id_activo] [int] NULL,
    [id_seguro] [int] NULL,
    CONSTRAINT [PK_ACTIVO_SEGURO] PRIMARY KEY CLUSTERED
(
    [id_activo_seguro] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[MANTENIMIENTO]') AND type in
(N'U'))
BEGIN
CREATE TABLE [dbo].[MANTENIMIENTO](
    [ID_MANTENIMIENTO] [int] NOT NULL,
    [ID_ACTIVO] [int] NULL,
    [ID_MOTIVO_MANTENIMIENTO] [int] NULL,
    [FECHA_MANTENIMIENTO] [datetime] NULL,
    [OBSERVACION_MANTENIMIENTO] [varchar](250) NULL,
    CONSTRAINT [PK_MANTENIMIENTO] PRIMARY KEY CLUSTERED
(
    [ID_MANTENIMIENTO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[ACTIVO_CARACTERISTICA]')
AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[ACTIVO_CARACTERISTICA](
    [ID_ACTIVO_CARACTERISTICA] [int] NULL,
    [ID_ACTIVO] [int] NULL,
    [ID_CARACTERISTICA] [int] NULL,
    [VALOR_CARACTERISTICA] [varchar](200) NULL
) ON [PRIMARY]

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[VALOR_CARACTERISTICA]')
AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[VALOR_CARACTERISTICA](
    [ID_VALOR_CARACTERISTICA] [int] NOT NULL,
    [ID_CARACTERISTICA] [int] NULL,
    [DATO_VALOR_CARACTERISTICA] [varchar](100) NULL,
    CONSTRAINT [PK_VALOR_CARACTERISTICA] PRIMARY KEY CLUSTERED
(
    [ID_VALOR_CARACTERISTICA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CARACTERISTICAS]') AND type
in (N'U'))
BEGIN
CREATE TABLE [dbo].[CARACTERISTICAS](
    [ID_CARACTERISTICA] [int] NOT NULL,
    [NOMBRE_CARACTERISTICA] [varchar](50) NULL,
    [TIPO_CARACTERISTICA] [varchar](5) NULL,
    [ID_CATEGORIA_ACTIVIVO] [int] NULL,
    [ESTADO_CARACTERISTICA] [int] NULL,
    CONSTRAINT [PK_CARACTERISTICAS] PRIMARY KEY CLUSTERED
(
    [ID_CARACTERISTICA] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[DESCRIPCION_ACTIVIVO]') AND
type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DESCRIPCION_ACTIVIVO](
    [ID_DESCRIPCION_ACTIVIVO] [int] NOT NULL,
    [ID_CATEGORIA_ACTIVIVO] [int] NOT NULL,
    [NOMBRE_DESCRIPCION_ACTIVIVO] [varchar](60) NULL,
    [MODELO_DESCRIPCION_ACTIVIVO] [varchar](60) NULL,
    [CANTIDAD_DESCRIPCION_ACTIVIVO] [int] NULL,
    [CODIGO_DESCRIPCION_ACTIVIVO] [char](4) NULL,
    [OBSERVACION_DESCRIPCION_ACTIVIVO] [varchar](200) NULL,
    [VIDA_UTIL_DESCRIPCION_ACTIVIVO] [int] NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL CONSTRAINT
[MSmerge_df_rowguid_59C009831ED64CDD9C56A76A1D138AE2] DEFAULT (newid()),
    CONSTRAINT [PK_DESCRIPCION_ACTIVIVO] PRIMARY KEY CLUSTERED
(
    [ID_DESCRIPCION_ACTIVIVO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVIVO_DESCRIPCI_DESCRIPC]') AND parent_object_id = OBJECT_ID(N'[dbo].[ACTIVO]'))
ALTER TABLE [dbo].[ACTIVO] WITH CHECK ADD CONSTRAINT [FK_ACTIVIVO_DESCRIPCI_DESCRIPC] FOREIGN
KEY([ID_DESCRIPCION_ACTIVIVO])
REFERENCES [dbo].[DESCRIPCION_ACTIVIVO] ([ID_DESCRIPCION_ACTIVIVO])
GO

```

```

IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_ESTADOACT_ESTADO_A]') AND parent_object_id = OBJECT_ID(N'[dbo].[ACTIVO]'))
ALTER TABLE [dbo].[ACTIVO] WITH CHECK ADD CONSTRAINT [FK_ACTIVO_ESTADOACT_ESTADO_A] FOREIGN
KEY([ID_ESTADO_ACTIVO])
REFERENCES [dbo].[ESTADO_ACTIVO] ([ID_ESTADO_ACTIVO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_PROVEEDOR]') AND parent_object_id = OBJECT_ID(N'[dbo].[ACTIVO]'))
ALTER TABLE [dbo].[ACTIVO] WITH CHECK ADD CONSTRAINT [FK_ACTIVO_PROVEEDOR] FOREIGN
KEY([ID_PROVEEDOR])
REFERENCES [dbo].[PROVEEDOR] ([id_proveedor])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_USUARIOXA_USUARIO]') AND parent_object_id = OBJECT_ID(N'[dbo].[ACTIVO]'))
ALTER TABLE [dbo].[ACTIVO] WITH CHECK ADD CONSTRAINT [FK_ACTIVO_USUARIOXA_USUARIO] FOREIGN
KEY([ID_USUARIO])
REFERENCES [dbo].[USUARIO] ([ID_USUARIO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_SOLICITU_ESTADOSOL_ESTADO_S]') AND parent_object_id =
OBJECT_ID(N'[dbo].[SOLICITUD_ENTREGA]'))
ALTER TABLE [dbo].[SOLICITUD_ENTREGA] WITH CHECK ADD CONSTRAINT
[FK_SOLICITU_ESTADOSOL_ESTADO_S] FOREIGN KEY([ID_ESTADO_SOLICITUD_ENTREGA])
REFERENCES [dbo].[ESTADO_SOLICITUD_ENTREGA] ([ID_ESTADO_SOLICITUD_ENTREGA])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_SOLICITU_UBICACION_UBICACION]') AND parent_object_id =
OBJECT_ID(N'[dbo].[SOLICITUD_ENTREGA]'))
ALTER TABLE [dbo].[SOLICITUD_ENTREGA] WITH CHECK ADD CONSTRAINT
[FK_SOLICITU_UBICACION_UBICACION] FOREIGN KEY([ID_UBICACION])
REFERENCES [dbo].[UBICACION] ([ID_UBICACION])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_UBICACION_ACTIVOXAB_ACTIVOXAB]') AND parent_object_id =
OBJECT_ID(N'[dbo].[UBICACION_ACTIVOXAB]'))
ALTER TABLE [dbo].[UBICACION_ACTIVOXAB] WITH CHECK ADD CONSTRAINT [FK_UBICACION_ACTIVOXAB_ACTIVOXAB]
FOREIGN KEY([ID_ACTIVOXAB])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIVOXAB])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_UBICACION_ESTADOXUB_ESTADO_U]') AND parent_object_id =
OBJECT_ID(N'[dbo].[UBICACION_ACTIVOXAB]'))
ALTER TABLE [dbo].[UBICACION_ACTIVOXAB] WITH CHECK ADD CONSTRAINT
[FK_UBICACION_ESTADOXUB_ESTADO_U] FOREIGN KEY([ID_ESTADO_UBICACION_ACTIVOXAB])
REFERENCES [dbo].[ESTADO_UBICACION_ACTIVOXAB] ([ID_ESTADO_UBICACION_ACTIVOXAB])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_UBICACION_UBICACION_UBICACION]') AND parent_object_id =
OBJECT_ID(N'[dbo].[UBICACION_ACTIVOXAB]'))
ALTER TABLE [dbo].[UBICACION_ACTIVOXAB] WITH CHECK ADD CONSTRAINT
[FK_UBICACION_UBICACION_UBICACION] FOREIGN KEY([ID_UBICACION])
REFERENCES [dbo].[UBICACION] ([ID_UBICACION])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_USUARIO_PERFILXUS_PERFIL]') AND parent_object_id = OBJECT_ID(N'[dbo].[USUARIO]'))
ALTER TABLE [dbo].[USUARIO] WITH CHECK ADD CONSTRAINT [FK_USUARIO_PERFILXUS_PERFIL] FOREIGN
KEY([ID_PERFIL])
REFERENCES [dbo].[PERFIL] ([ID_PERFIL])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PERMISO_LLAMADA]') AND parent_object_id = OBJECT_ID(N'[dbo].[PERMISO]'))
ALTER TABLE [dbo].[PERMISO] WITH CHECK ADD CONSTRAINT [FK_PERMISO_LLAMADA] FOREIGN
KEY([ID_TIPO_LLAMADA])
REFERENCES [dbo].[LLAMADA] ([ID_TIPO_LLAMADA])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_PERMISO_PERFIL]')
AND parent_object_id = OBJECT_ID(N'[dbo].[PERMISO]'))
ALTER TABLE [dbo].[PERMISO] WITH CHECK ADD CONSTRAINT [FK_PERMISO_PERFIL] FOREIGN
KEY([ID_PERFIL])
REFERENCES [dbo].[PERFIL] ([ID_PERFIL])
GO

```

```

IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLE__ACTIVOXDE_ACTIV0]') AND parent_object_id =
OBJECT_ID(N'[dbo].[DETALLE_SOLICITUD_ENTREGA]'))
ALTER TABLE [dbo].[DETALLE_SOLICITUD_ENTREGA] WITH CHECK ADD CONSTRAINT
[FK_DETALLE__ACTIVOXDE_ACTIV0] FOREIGN KEY([ID_ACTIV0])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIV0])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLE__SOLITUDXD_SOLICITU]') AND parent_object_id =
OBJECT_ID(N'[dbo].[DETALLE_SOLICITUD_ENTREGA]'))
ALTER TABLE [dbo].[DETALLE_SOLICITUD_ENTREGA] WITH CHECK ADD CONSTRAINT
[FK_DETALLE__SOLITUDXD_SOLICITU] FOREIGN KEY([ID_SOLICITUD_ENTREGA])
REFERENCES [dbo].[SOLICITUD_ENTREGA] ([ID_SOLICITUD_ENTREGA])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_HISTORIA_ACTIVOXHI_ACTIV0]') AND parent_object_id =
OBJECT_ID(N'[dbo].[HISTORIAL_ACTIV0]'))
ALTER TABLE [dbo].[HISTORIAL_ACTIV0] WITH CHECK ADD CONSTRAINT [FK_HISTORIA_ACTIVOXHI_ACTIV0]
FOREIGN KEY([ID_ACTIV0])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIV0])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_HISTORIA_TIPOMOVIL_TIPO_MOV]') AND parent_object_id =
OBJECT_ID(N'[dbo].[HISTORIAL_ACTIV0]'))
ALTER TABLE [dbo].[HISTORIAL_ACTIV0] WITH CHECK ADD CONSTRAINT
[FK_HISTORIA_TIPOMOVIL_TIPO_MOV] FOREIGN KEY([ID_TIPO_MOVILIZACION])
REFERENCES [dbo].[TIPO_MOVILIZACION] ([ID_TIPO_MOVILIZACION])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_HISTORIA_USUARIOXH_USUARIO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[HISTORIAL_ACTIV0]'))
ALTER TABLE [dbo].[HISTORIAL_ACTIV0] WITH CHECK ADD CONSTRAINT
[FK_HISTORIA_USUARIOXH_USUARIO] FOREIGN KEY([ID_USUARIO])
REFERENCES [dbo].[USUARIO] ([ID_USUARIO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_EMPLEADO_UBICACION]') AND parent_object_id = OBJECT_ID(N'[dbo].[EMPLEADO]'))
ALTER TABLE [dbo].[EMPLEADO] WITH NOCHECK ADD CONSTRAINT [FK_EMPLEADO_UBICACION] FOREIGN
KEY([id_ubicacion])
REFERENCES [dbo].[UBICACION] ([ID_UBICACION])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PERMISO__RELATIONS_PERMISO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[PERMISO_USUARIO]'))
ALTER TABLE [dbo].[PERMISO_USUARIO] WITH CHECK ADD CONSTRAINT
[FK_PERMISO__RELATIONS_PERMISO] FOREIGN KEY([ID_PERMISO])
REFERENCES [dbo].[PERMISO] ([ID_PERMISO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PERMISO__RELATIONS_USUARIO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[PERMISO_USUARIO]'))
ALTER TABLE [dbo].[PERMISO_USUARIO] WITH CHECK ADD CONSTRAINT
[FK_PERMISO__RELATIONS_USUARIO] FOREIGN KEY([ID_USUARIO])
REFERENCES [dbo].[USUARIO] ([ID_USUARIO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_TRASPASO_ACTIV0]')
AND parent_object_id = OBJECT_ID(N'[dbo].[TRASPASO]'))
ALTER TABLE [dbo].[TRASPASO] WITH CHECK ADD CONSTRAINT [FK_TRASPASO_ACTIV0] FOREIGN
KEY([ID_ACTIV0])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIV0])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_VALOR_AC_ACTIVOXVA_ACTIV0]') AND parent_object_id =
OBJECT_ID(N'[dbo].[VALOR_ACTIV0]'))
ALTER TABLE [dbo].[VALOR_ACTIV0] WITH CHECK ADD CONSTRAINT [FK_VALOR_AC_ACTIVOXVA_ACTIV0]
FOREIGN KEY([ID_ACTIV0])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIV0])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIV0_BAJA_ACTIV0]') AND parent_object_id = OBJECT_ID(N'[dbo].[ACTIVO_BAJA]'))
ALTER TABLE [dbo].[ACTIVO_BAJA] WITH CHECK ADD CONSTRAINT [FK_ACTIV0_BAJA_ACTIV0] FOREIGN
KEY([ID_ACTIV0])

```

```

REFERENCES [dbo].[ACTIVO] ([ID_ACTIVO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_SEGURO_ACTIVO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[ACTIVO_SEGURO]'))
ALTER TABLE [dbo].[ACTIVO_SEGURO] WITH CHECK ADD CONSTRAINT [FK_ACTIVO_SEGURO_ACTIVO]
FOREIGN KEY([id_activo])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIVO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_SEGURO_SEGUROS]') AND parent_object_id =
OBJECT_ID(N'[dbo].[ACTIVO_SEGURO]'))
ALTER TABLE [dbo].[ACTIVO_SEGURO] WITH CHECK ADD CONSTRAINT [FK_ACTIVO_SEGURO_SEGUROS]
FOREIGN KEY([id_seguro])
REFERENCES [dbo].[SEGUROS] ([id_seguro])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_MANTENIMIENTO_ACTIVO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[MANTENIMIENTO]'))
ALTER TABLE [dbo].[MANTENIMIENTO] WITH CHECK ADD CONSTRAINT [FK_MANTENIMIENTO_ACTIVO]
FOREIGN KEY([ID_ACTIVO])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIVO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_MANTENIMIENTO_MOTIVO_MANTENIMIENTO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[MANTENIMIENTO]'))
ALTER TABLE [dbo].[MANTENIMIENTO] WITH CHECK ADD CONSTRAINT
[FK_MANTENIMIENTO_MOTIVO_MANTENIMIENTO] FOREIGN KEY([ID_MOTIVO_MANTENIMIENTO])
REFERENCES [dbo].[MOTIVO_MANTENIMIENTO] ([ID_MOTIVO_MANTENIMIENTO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_CARACTERISTICA_ACTIVO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[ACTIVO_CARACTERISTICA]'))
ALTER TABLE [dbo].[ACTIVO_CARACTERISTICA] WITH CHECK ADD CONSTRAINT
[FK_ACTIVO_CARACTERISTICA_ACTIVO] FOREIGN KEY([ID_ACTIVO])
REFERENCES [dbo].[ACTIVO] ([ID_ACTIVO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACTIVO_CARACTERISTICA_CARACTERISTICAS]') AND parent_object_id =
OBJECT_ID(N'[dbo].[ACTIVO_CARACTERISTICA]'))
ALTER TABLE [dbo].[ACTIVO_CARACTERISTICA] WITH CHECK ADD CONSTRAINT
[FK_ACTIVO_CARACTERISTICA_CARACTERISTICAS] FOREIGN KEY([ID_CARACTERISTICA])
REFERENCES [dbo].[CARACTERISTICAS] ([ID_CARACTERISTICA])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_VALOR_CARACTERISTICA_CARACTERISTICAS]') AND parent_object_id =
OBJECT_ID(N'[dbo].[VALOR_CARACTERISTICA]'))
ALTER TABLE [dbo].[VALOR_CARACTERISTICA] WITH CHECK ADD CONSTRAINT
[FK_VALOR_CARACTERISTICA_CARACTERISTICAS] FOREIGN KEY([ID_CARACTERISTICA])
REFERENCES [dbo].[CARACTERISTICAS] ([ID_CARACTERISTICA])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_CARACTERISTICAS_CATEGORIA_ACTIVO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[CARACTERISTICAS]'))
ALTER TABLE [dbo].[CARACTERISTICAS] WITH CHECK ADD CONSTRAINT
[FK_CARACTERISTICAS_CATEGORIA_ACTIVO] FOREIGN KEY([ID_CATEGORIA_ACTIVO])
REFERENCES [dbo].[CATEGORIA_ACTIVO] ([ID_CATEGORIA_ACTIVO])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DESCRIPC_CATEGORIA_CATEGORI]') AND parent_object_id =
OBJECT_ID(N'[dbo].[DESCRIPCION_ACTIVO]'))
ALTER TABLE [dbo].[DESCRIPCION_ACTIVO] WITH CHECK ADD CONSTRAINT
[FK_DESCRIPC_CATEGORIA_CATEGORI] FOREIGN KEY([ID_CATEGORIA_ACTIVO])
REFERENCES [dbo].[CATEGORIA_ACTIVO] ([ID_CATEGORIA_ACTIVO])

```

4.2 PRUEBAS

Las pruebas de software se integran dentro de las diferentes fases del Ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

Uno de los puntos más importantes dentro del desarrollo de software son los escenarios de testing³⁴ que existen, pues en ellos se verifica los comportamientos del sistema para satisfacer los requerimientos establecidos por los clientes. Dentro de esta fase se pueden desarrollar diferentes tipos de prueba en función de los objetivos del mismo.

4.2.1 PRUEBAS DE INTEGRACION

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto.

Existen dos tipos de pruebas de integración, las pruebas de integración incremental ascendente y descendente.

4.2.1.1 MODULO DE ACCESO AL SISTEMA CELLARACTIVE

Valores de entrada

Clave o contraseña: es indispensable para el ingreso al sistema, debe ser numérico de tres a cuatro dígitos.

Valores de salida

S1 = Acceso al sistema

S2 = Clave incorrecta

Definición de casos de prueba

	Clases Válidas	Clases no válidas	Regla
Clave o contraseña	[111-9999] (1) 4 números (4)	<111 (2) >9999 (3) Alfabéticos (5)	Rango Miembro de un Conjunto

Tabla 5: Casos de prueba Modulo de acceso al sistema

³⁴ **Testing.**- desarrollo de un modelo de prueba.

Casos Válidos

Regla: Cada caso válido debe cubrir tantas clases válidas como sea posible.

Entrada:	Salida:
1234,1234	S1 (1)
9850,9850	S1 (4)

Tabla 6: Casos válidos Modulo de acceso al sistema

Casos no válidos

Regla: cada caso no válido debe cubrir una y sólo una clase no válida.

Entrada:	Salida:
00	S2 (2)
12120	S2 (3)
Abcd, ABCD	S2 (5)

Tabla 7: Casos no válidos Modulo de acceso al sistema

4.2.1.2 MODULO ADMINISTRADOR DE ACTIVOS FIJOS

Valores de entrada

Código.- Es un valor que se le asigna a las categorías, que puedan existir en el campo de activos fijos.

Nombre.- Es el nombre de la categoría de los activos fijos, debe ser alfabético de hasta sesenta dígitos.

Depreciación.- Es el valor en porcentaje, en el cual de acuerdo a la categoría de un activo fijo se deprecia.

Vida útil.- Valor en años, en el que la categoría de un activo fijo tiene.

Valores de salida

S3 = Código ingresado

S4 = Código no ingresado

S5 = Nombre registrado

S6 = Nombre no registrado

S7=Depreciación registrada

S8=Depreciación no registrada

S9=Vida útil registrada

S10=Vida útil no registrada

Definición de casos de prueba

	Clases Válidas	Clases no válidas	Regla
Código	1 número (6) Alfabéticos (7)	>9999 (8)	N.º de valores
Nombre	60 caracteres (9)	Numérico (10) >60caracteres(11)	N.º de valores
Depreciación	número(12)	>99999(13)	Rango
Vida Útil	número(14)	>99999(15)	Rango

Tabla 8: Casos de prueba Modulo administrador de activos

Casos Válidos

Entrada:	Salida:
1111	S3 (6)
A001	S3 (7)
Equipos de oficina	S5 (9)
100	S7 (12)
5	S9 (14)

Tabla 9: Casos válidos Modulo administrador de activos

Casos no válidos

Entrada:	Salida:
9999 ^a	S4 (8)
111111111111111111	S6(10)
Edificio.....	S6(11)
Entrada:	Salida:
10000	S8(13)
50000	S10(15)

Tabla 10: Casos no válidos Modulo administrador de activos

4.2.1.3 MODULO ORGANIZADOR DE ACTIVOS FIJOS

Valores de entrada

Categorías.- Es el nombre de la categoría a la cual va a pertenecer el activo que se va a registrar, debe ser alfabético de hasta sesenta dígitos.

Descripción.- Es la descripción más detallada del activo que se va a registrar, debe ser alfabética de hasta sesenta dígitos.

Proveedor.- Es la descripción de los proveedores de un activo determinado que se está registrando, debe ser alfabética de hasta sesenta dígitos.

Fecha de Compra.- Es el registro de la fecha en la cual se realizó la adquisición, no debe ser mayor a la fecha actual del sistema.

Valor de la compra.- Es el valor de compra del activo. Numérico.

Documento/Factura.- Es el número de la factura con la que se hizo la adquisición del activo. Valor alfanumérico de hasta 50 dígitos.

Valores de salida

S11 = Categoría registrada

S12 = Categoría no registrada

S13= Descripción registrada

S14= Descripción no registrada

S15=Fecha Registrada

S16= Fecha no Registrada

S17=Valor de compra Registrado

S18=Valor de compra no Registrado

S19=Documento/Factura Registrado

S20=Documento/Factura no Registrado

Definición de casos de prueba

	Clases Válidas	Clases no válidas	Regla
Categoría	60 caracteres (16)	Numérico (17) > 60 caracteres (18)	N.º de valores
Descripción	60 caracteres (19)	Numérico (20) > 60 caracteres (21)	N.º de valores
Fecha de Compra	Fecha de compra <Fecha actual(22)	Fecha de compra>Fecha actual(23)	Rango
Valor de compra	Numérico(24)	>999999999(25)	Valor
Documento/ Factura	Alfabético(26)	>50 caracteres(27)	No de valores

Tabla 11: Casos de prueba Modulo organizador de activos

Casos Válidos

Entrada:	Salida:
1-Terrenos	S11 (16)
0004-Router Cisco	S13(19)
15/07/09	S15(22)
256.00	S17(24)
S001-001-000256	S19(26)

Tabla 12: Casos válidos Modulo organizador de activos

Casos no Válidos

Entrada:	Salida:
51560....	S12 (17)
Terrenoooooooooooooooo.....s	S12 (18)
0004-Router Cisc.....o	S14(21)
2/08/2010	S16(23)
1999999999	S18(25)
S001-0001-0000000000....254	S20(27)

Tabla 13: Casos no válidos Modulo organizador de activos

4.2.3.1.4 MODULO ADMINISTRADOR DE USUARIO

Valores de entrada

Nombre de usuario.- Es el nombre del usuario que va a ingresar al sistema, debe ser alfabético de hasta treinta dígitos.

Nick.- Es el sobrenombre, que se le asigna a un usuario para que ingrese al sistema, debe ser alfabético de hasta treinta dígitos

Clave.- Es la clave del usuario que va ingresar al sistema, debe ser alfanumérico de hasta 30 dígitos.

Valores de salida

S21 = Nombre registrado

S22 = Nombre no registrado

S23 = Clave registrado

S24 = Clave no registrado

Definición de casos de prueba

	Clases Válidas	Clases no válidas	Regla
Nombre de usuario	30 caracteres (28)	Numérico (29) > 30 caracteres (30)	N.º de valores
Clave	30 caracteres (31)	< 30 caracteres (32)	Rango

Tabla 14: Casos de prueba Modulo administrador de usuario

Casos Válidos

Entrada:	Salida:
Carolina Cobos, 4578lm	S21 (28)
cCarol123,12524854	S23 (31)

Tabla 15: Casos válidos Modulo administrador de usuario

Casos no Válidos

Entrada:	Salida:
4871....., SA	S22 (29) (30)
Usuariosgerentegeneral, asdgf\$.1455	S24 (32)

Tabla 16: Casos no válidos Modulo administrador de usuario

MANUAL DE CONFIGURACION DEL SISTEMA

MANUAL DE CONFIGURACIÓN CELLARACTIVE

Propósito del manual

El siguiente manual de configuración básica del producto tiene como objetivo brindar la suficiente información para configurar correctamente las distintas opciones y parámetros que se requieren, para hacer la configuración de una forma correcta.

Este manual va dirigido al usuario final para que se ubiquen cuando se necesita realizar una instalación desde cero del producto.

Este conjunto de configuraciones debe ser realizado en cada una de las máquinas a las cuales se desee instalar la aplicación CELLARACTIVE.

CONFIGURACIÓN REGIONAL

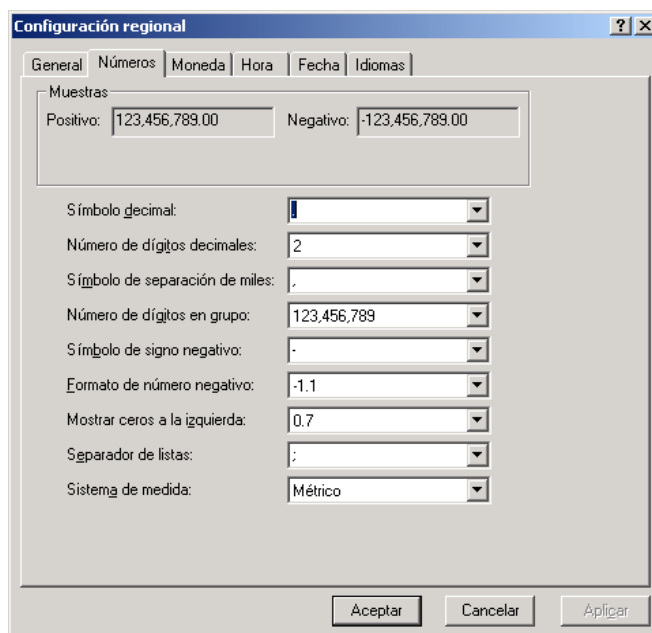
Esta configuración le indica a Windows como debe comportarse cuando tiene que interpretar fechas, números, números de tipo moneda.

La configuración regional debe ser configurada en cada estación.

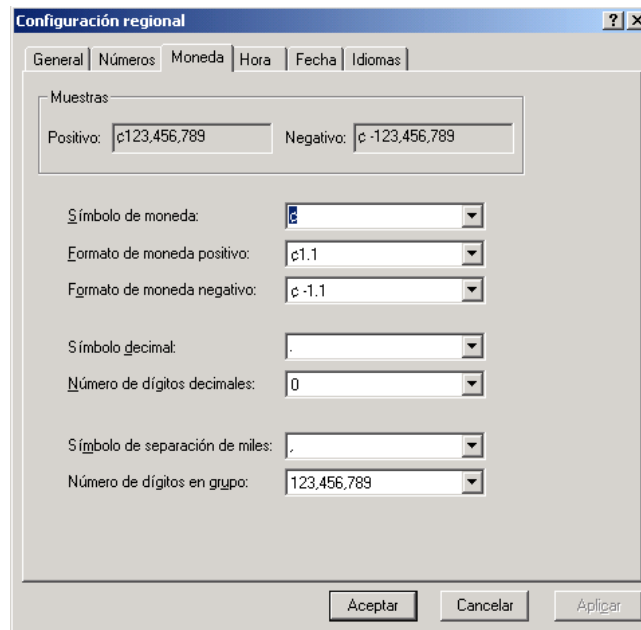
Números

Para símbolo decimal debe haber un punto (.).

Para símbolo de separación de miles debe ser una coma (,).



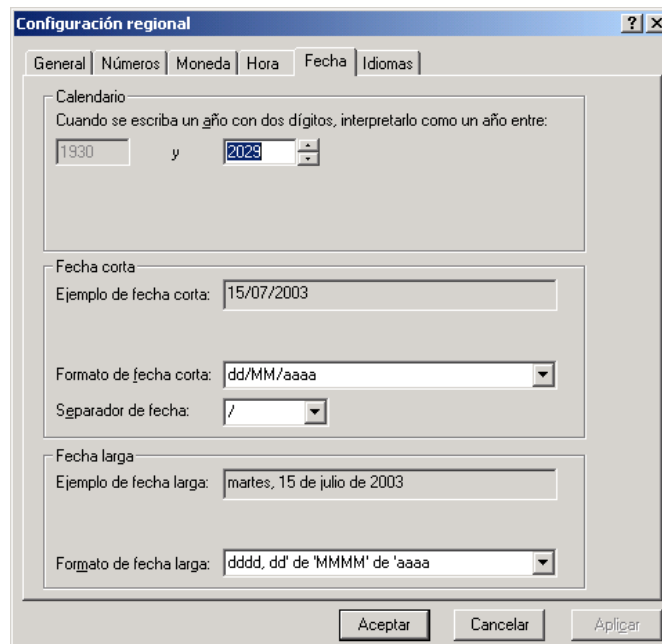
Moneda



Para símbolo decimal debe haber un punto (.).

Para símbolo de separación de miles debe ser una coma (,).

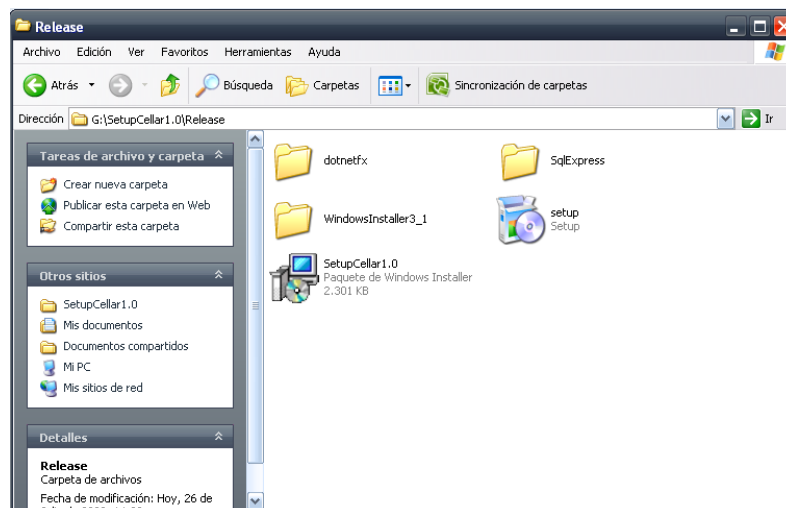
Fecha



Para formato de fecha corta debe ser dd/MM/aaaa o dd/MM/yyyy (para las versiones en inglés).

MANUAL DE INSTALACIÓN DEL SISTEMA CELLARACTIVE

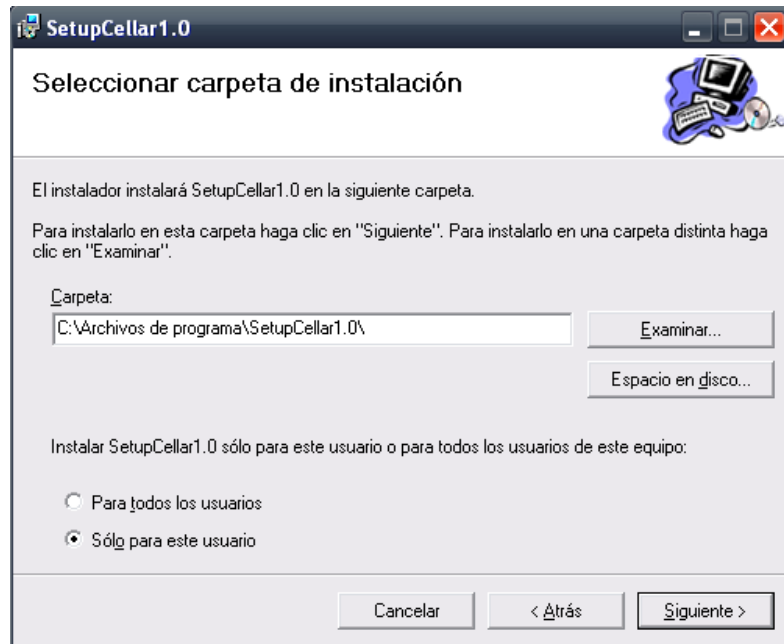
1. Abrimos la carpeta donde se encuentra el instalador, esta carpeta contendrá dos tipos de instaladores, el primero es un MSI el cual está diseñado para correr en Sistemas Operativos Vista, XP y 2000, el segundo instalador es una típica aplicación .EXE para que se ejecute en Sistemas operativos Windows 9x, Windows Me, y sus similares; además se encuentran carpetas con complementos como el framework 2.0, sql server express, librería de crystal reports, windows instaler. Esto para que, cuando instalamos nuestra aplicación si necesitamos de estos requisitos no nos toque instalarlos uno por uno, esto hace que cuando se ejecute el instalador el evalúe si se encuentran instalados los requisitos si no lo están el automáticamente los instalara.



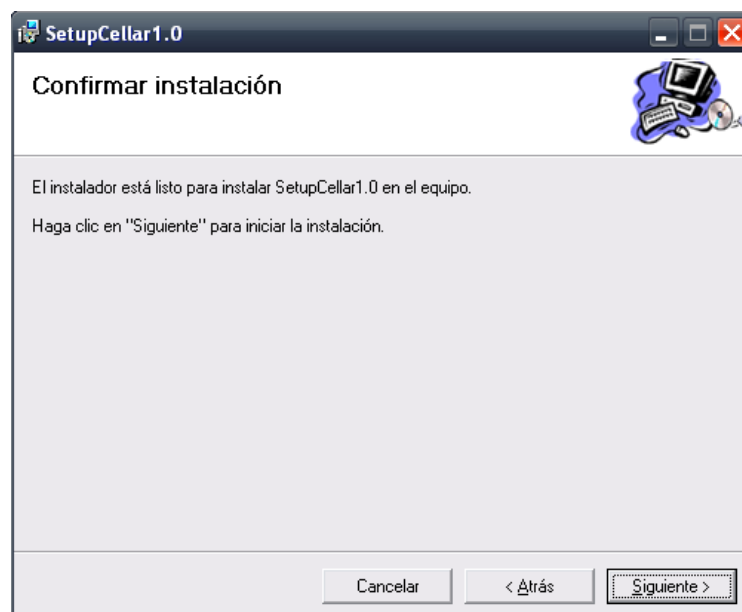
2. Ejecutamos el instalador, acorde a nuestra plataforma, como se muestra en la pantalla.



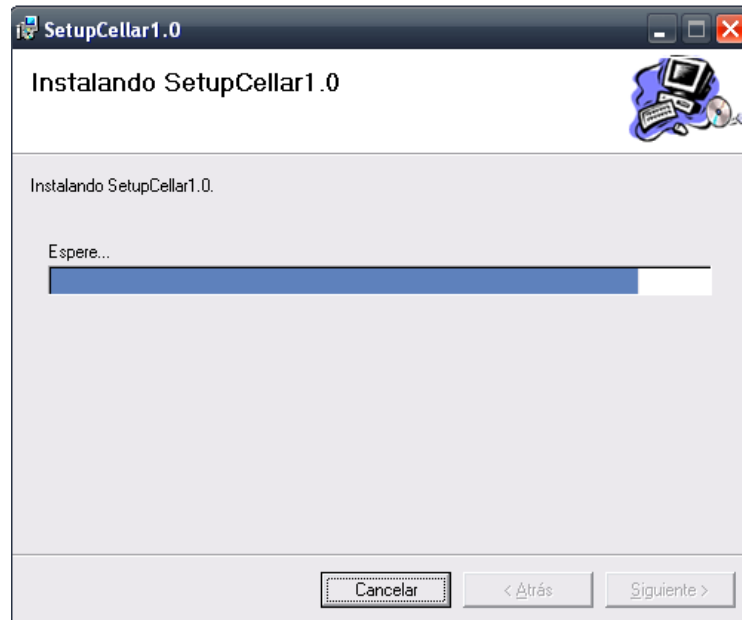
3. Pulsamos sobre el botón Siguiente, mostrándonos la siguiente pantalla:



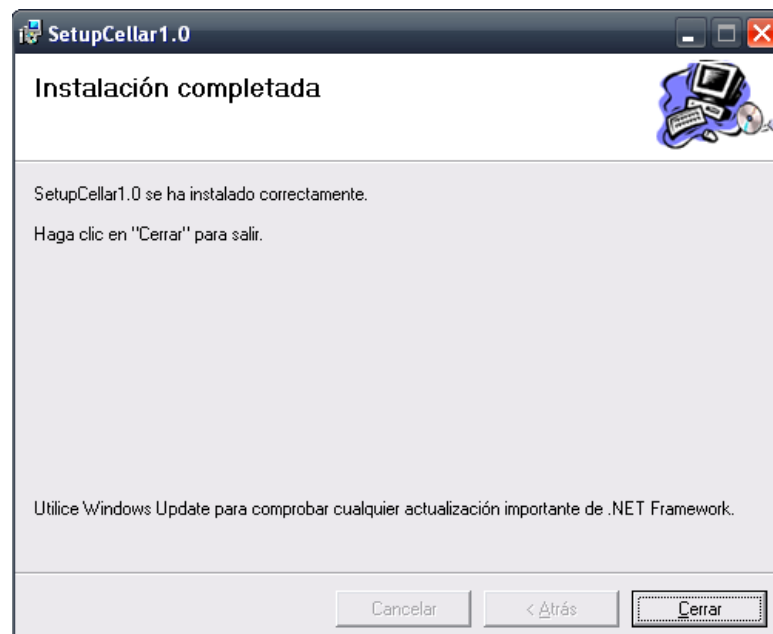
4. Si deseamos cambiar la ubicación en donde se instalará la aplicación, podemos cambiarla pulsando el botón Examinar. Para verificar si tenemos suficiente espacio en el disco duro podemos pulsar sobre el botón Espacio en Disco. Luego escogemos la opción Para todos los usuarios o Sólo para este usuario acorde a los usuarios que utilicen el equipo. Luego pulsamos el botón Siguiente presentándonos la siguiente pantalla.



5. Esta es solo una pantalla de confirmación de la instalación, para completar el proceso pulsamos sobre el botón Siguiente. Presentándonos la siguiente pantalla:



6. La pantalla anterior simplemente nos muestra el proceso de instalación de la aplicación, por lo solo se deberá esperar que se muestra la siguiente pantalla:



7. La pantalla anterior nos indica que la aplicación se instaló correctamente y finalmente pulsamos sobre el botón Cerrar para concluir la instalación.



8. Finalmente se creará un icono en el escritorio denominado CellarActive, Con el cual ya tendremos acceso directo a la aplicación CellarActive.

MANUAL DE USUARIO

Nomenclatura que usa CellarActive



Administración de Catálogos



Informes



Administración de Activos Fijos



Sincronización Móvil



Administración del Sistema



Categorías de Activos Fijos



Descripciones de Activos Fijos



Ubicaciones de Activos Fijos



Personal Encargado de Activos Fijos



Seguros de Activos Fijos



Proveedores de Activos Fijos



Activos, Entrega de Activos, Historial, Traspasos, Mantenimiento,
Depreciación



Módulo de Replicación



Perfiles



Permisos



Usuarios

CellarActive

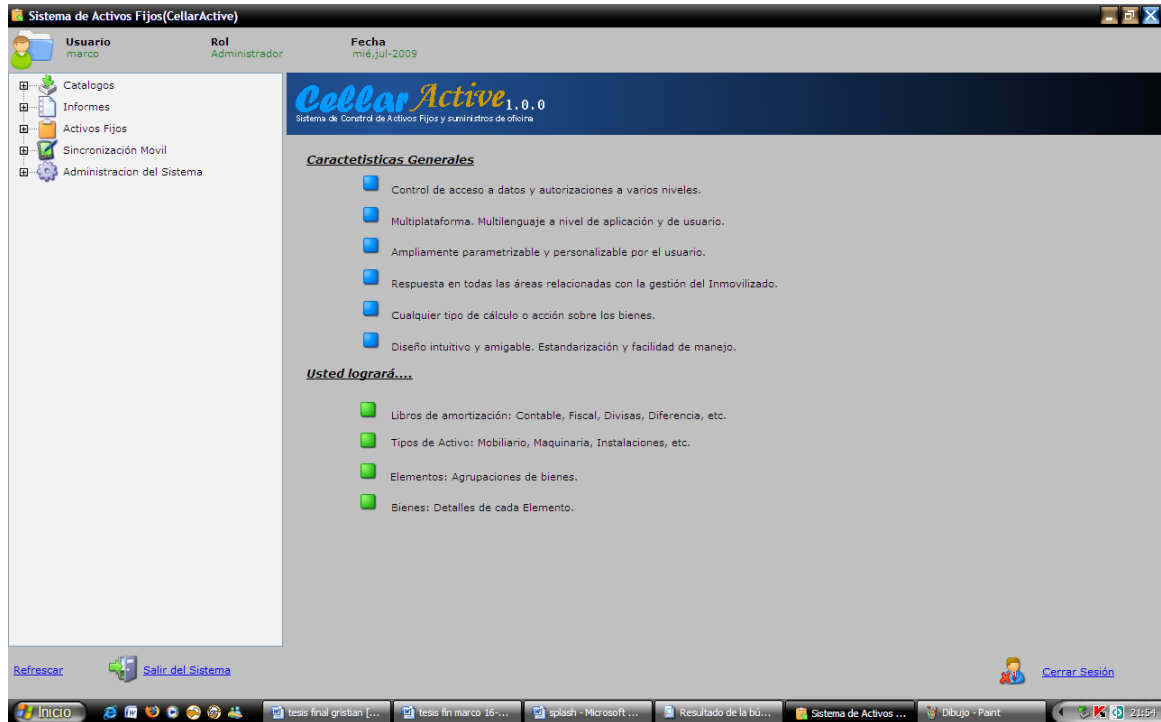
Pantalla de Bienvenida



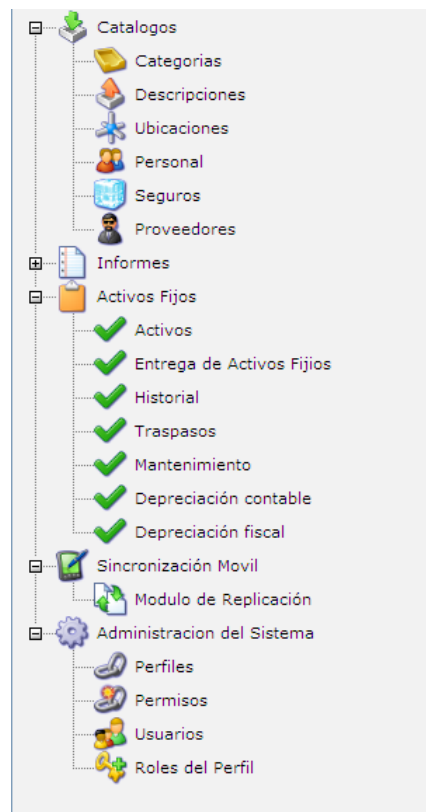
En esta pantalla se solicitará que ingresemos la contraseña asignada a cada usuario, para ingresar al sistema CellarActive 1.0.0

The screenshot shows a dialog box titled "Ingreso al Sistema - CellarActive". It contains a user icon on the left. The form has two input fields: "Nombre de usuario" with the text "marco" and "Contraseña" with masked characters "*****". At the bottom, there are two buttons: "Aceptar" and "Cancelar".

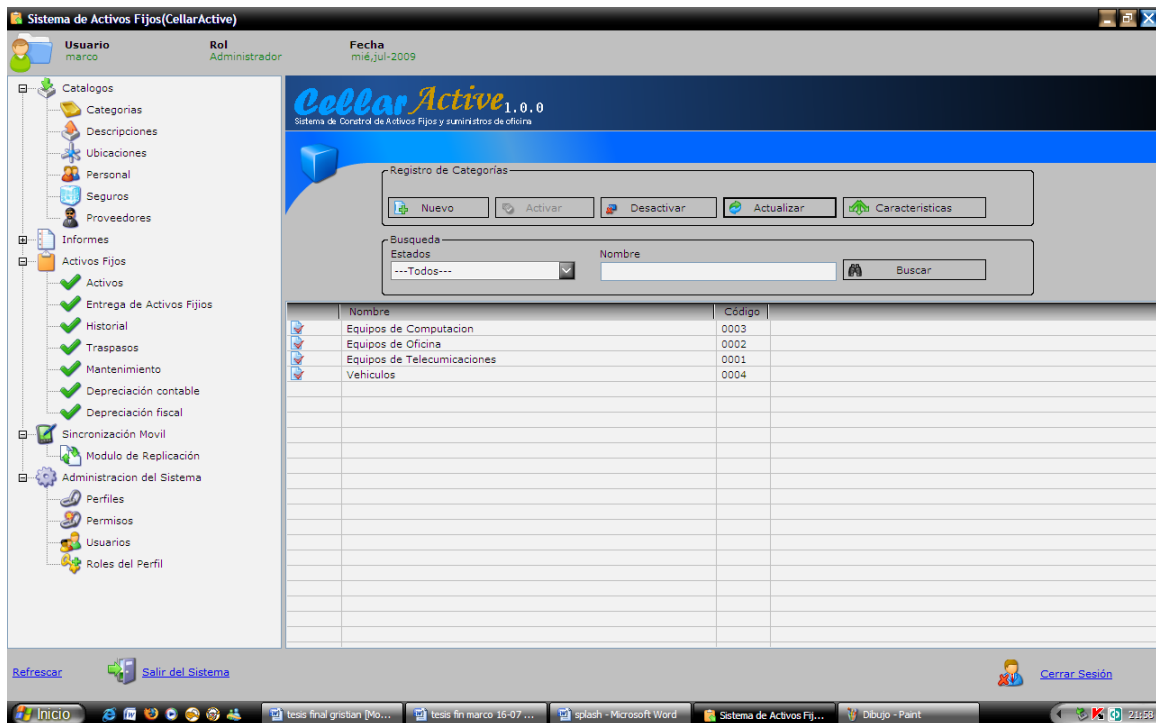
Una vez que hemos ingresado al sistema, este nos ofrecerá la siguiente pantalla:



En la cual podemos escoger las siguientes opciones:



1. Categorías: Permite el manejo de categorías de activos fijos



1.1 Nuevo: Permite agregar una nueva categoría de activos fijos

Registro de Categorías de Activos Fijos

Registro Categorías de Activos

Código: 6

Nombre: Equipos de Computación

Depreciación: 10 %

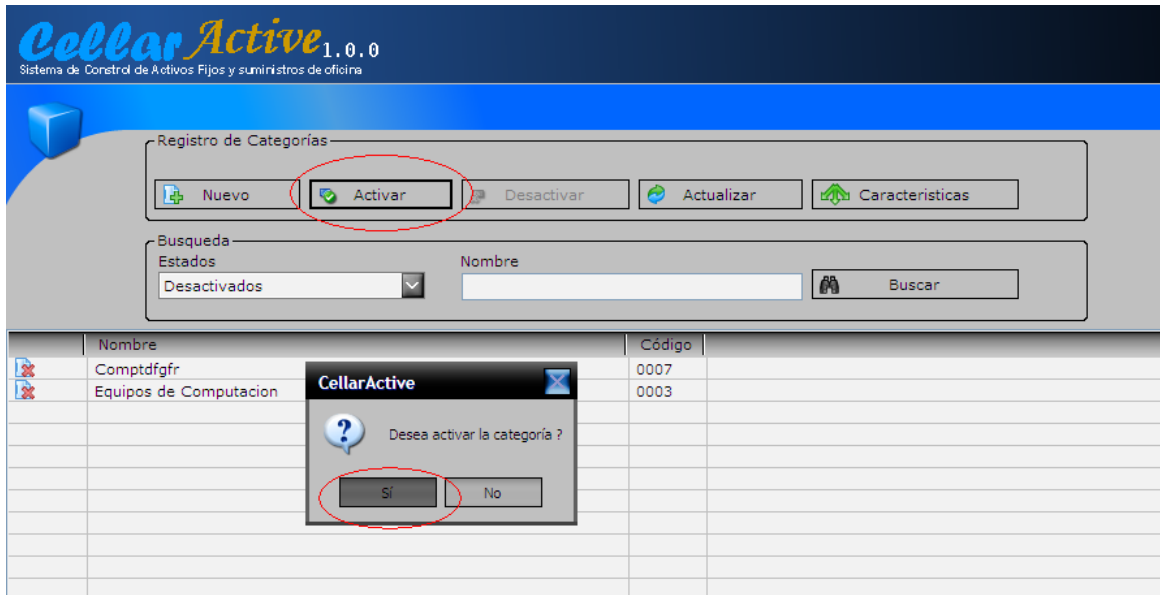
Observación:

Vida Util: 20

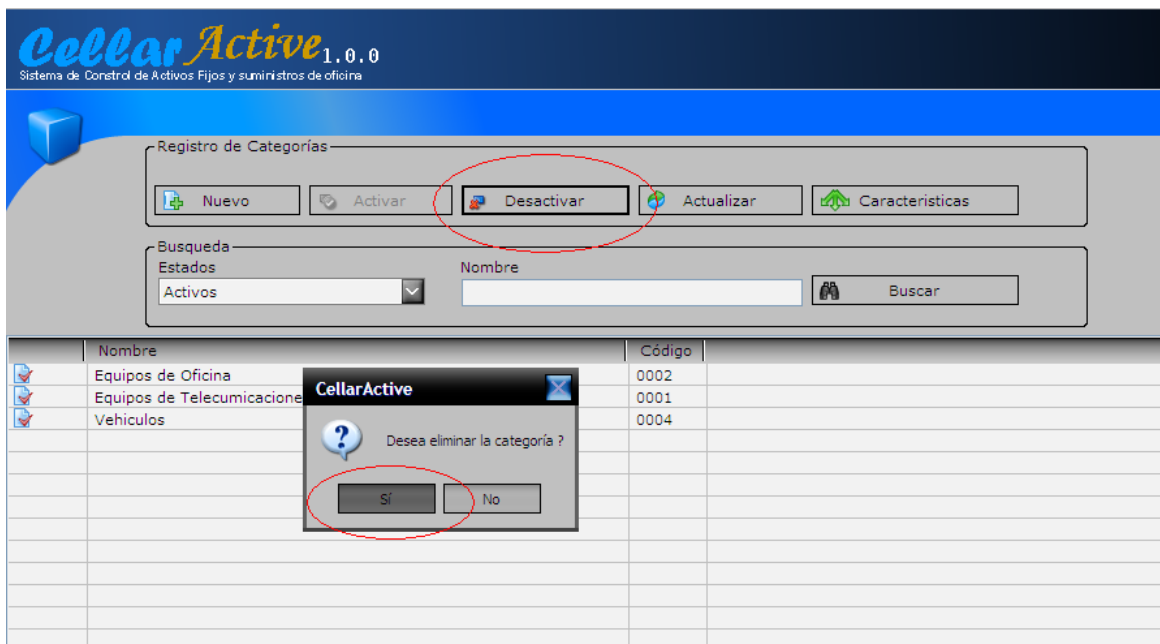
Estado: Activo

Aceptar Cerrar

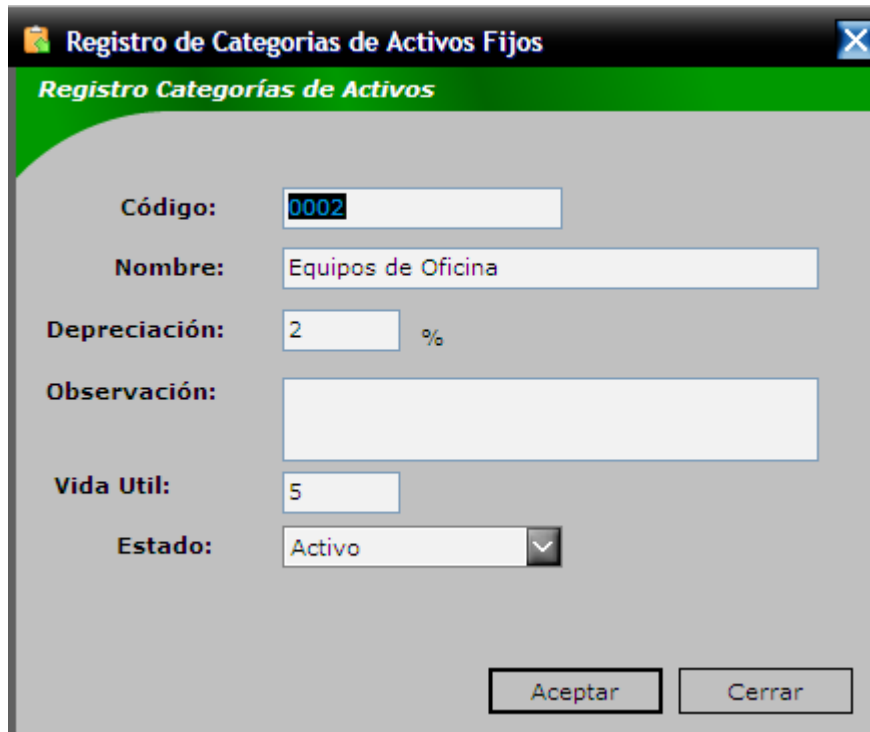
1.2 Activar: Permite activar una categoría de activo fijo que está desactivada



1.3 Desactivar: Permite desactivar una categoría de activo fijo que está activa



1.4 Actualizar: Permite modificar una categoría de activos fijos

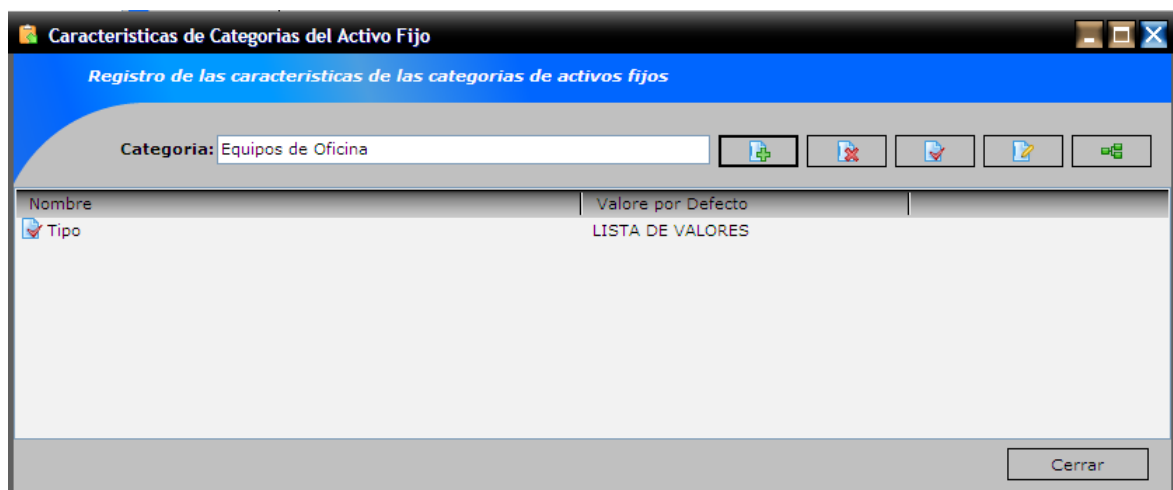


The screenshot shows a dialog box titled "Registro de Categorías de Activos Fijos". The main title bar is green and contains the text "Registro Categorías de Activos". The dialog has several input fields and a dropdown menu:

- Código:** A text box containing "0002".
- Nombre:** A text box containing "Equipos de Oficina".
- Depreciación:** A text box containing "2" followed by a percentage symbol "%".
- Observación:** A large empty text area.
- Vida Util:** A text box containing "5".
- Estado:** A dropdown menu currently showing "Activo".

At the bottom right, there are two buttons: "Aceptar" and "Cerrar".

1.5 Características: Permite agregar, activar, desactivar, modificar características de las categorías de activos fijos

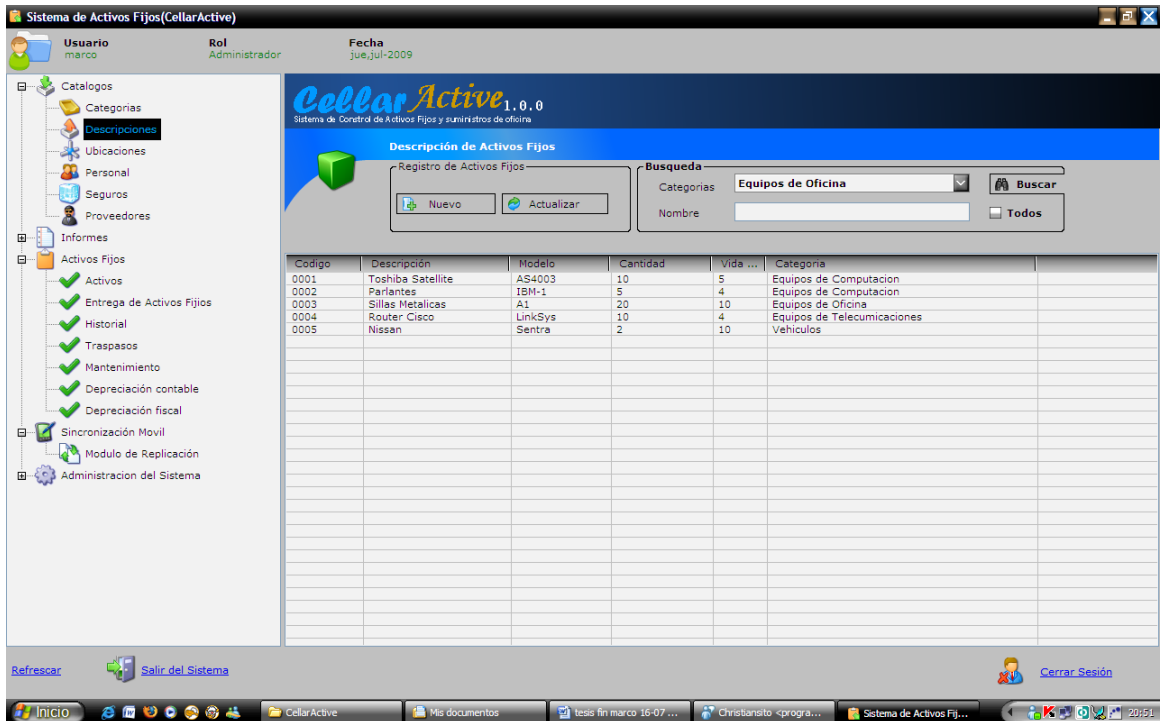


The screenshot shows a dialog box titled "Características de Categorías del Activo Fijo". The main title bar is blue and contains the text "Registro de las características de las categorías de activos fijos". The dialog has a search field and a list of characteristics:

- Categoría:** A text box containing "Equipos de Oficina".
- Lista de Características:** A table with columns "Nombre" and "Valore por Defecto". The table is currently empty, with the text "LISTA DE VALORES" centered below the header.

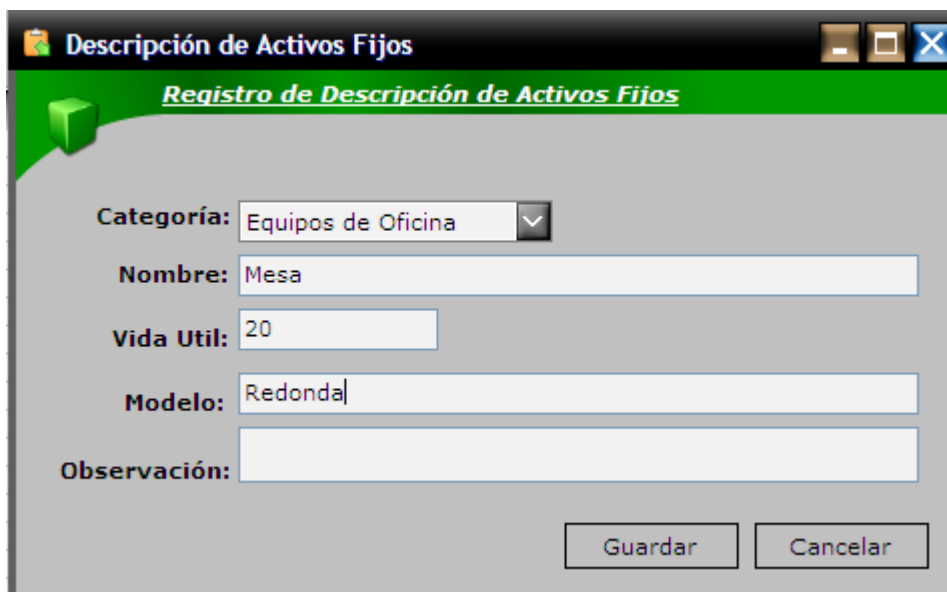
At the bottom right, there is a "Cerrar" button.

1. Descripciones: Permite registrar descripciones para los activos fijos.



Codigo	Descripción	Modelo	Cantidad	Vida ...	Categoría
0001	Toshiba Satellite	AS4003	10	5	Equipos de Computacion
0002	Parlantes	IBM-1	5	4	Equipos de Computacion
0003	Sillas Metalicas	A1	20	10	Equipos de Oficina
0004	Router Cisco	LinkSys	10	4	Equipos de Telecomunicaciones
0005	Nissan	Sentra	2	10	Vehiculos

1.1. Nuevo: Permite registrar una nueva descripción de activo fijo



Descripción de Activos Fijos

Registro de Descripción de Activos Fijos

Categoría: Equipos de Oficina

Nombre: Mesa

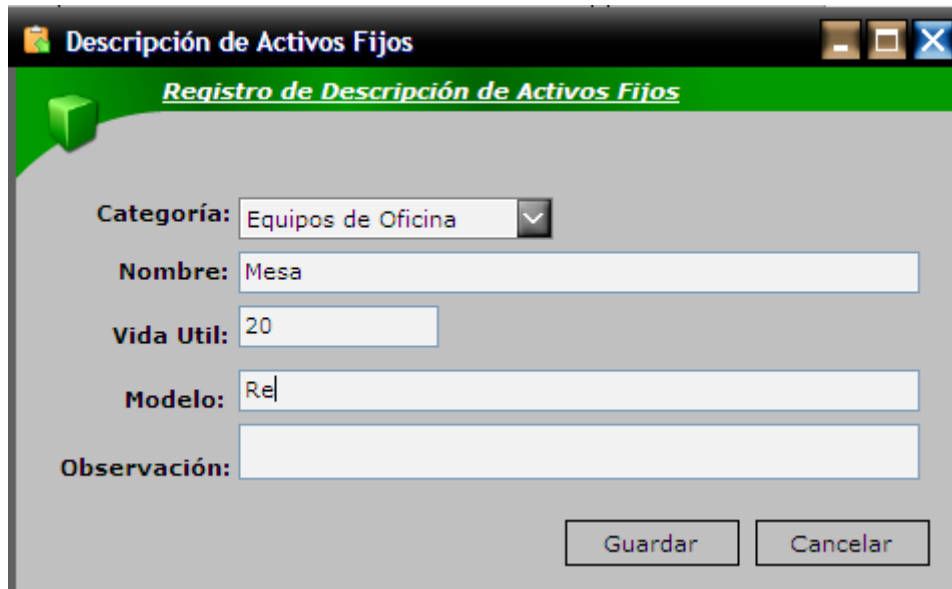
Vida Util: 20

Modelo: Redonda

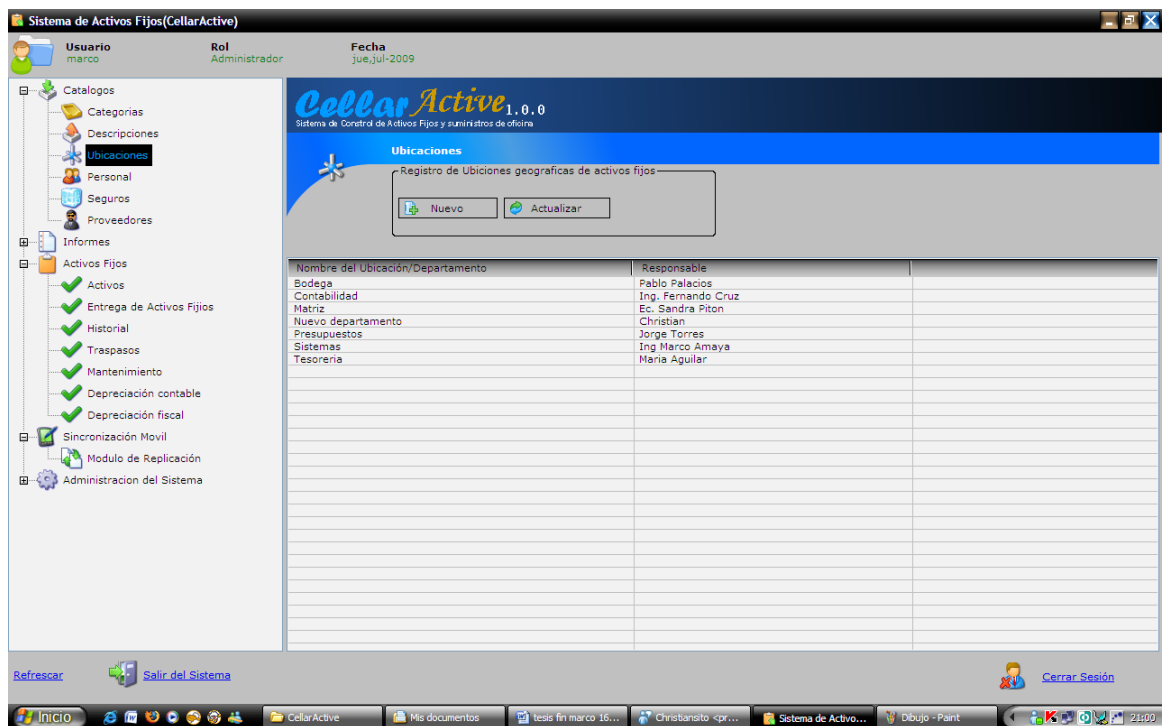
Observación:

Guardar Cancelar

1.2. Actualizar: Permite modificar una descripción de activo fijo



2. Ubicaciones: Permite el manejo de la ubicación donde se encuentra el activo fijo



2.1. Nuevo: Permite registrar una nueva ubicación y responsable del activo fijo

2.2. Actualizar: Permite modificar la información de la ubicación y del responsable del activo fijo

3. Personal: Permite el manejo del personal que utiliza el activo fijo

Nombre	Teléfono	Dirección	Profesión	Cargo	Exten...	Depa...
Christian Aguilar	2343	Las Casas	Ing Siste...	Desar...	092	2
Fernando Cruz		Solanda	Ing Siste...	Ing Si...		2
Pablo		Romero				5

3.1. Nuevo: Permite el registro de un empleado que utiliza el activo fijo

Empleado
Registro de Empleados

Nombre: Damián

Apellido: Suárez

Profesión: Contador

Cargo: Contador

Teléfono: 2858699 Extensión: 189

Departamento: Contabilidad

Aceptar Cerrar

3.2. Activar: Permite activar un empleado que está desactivado

Cellar Active 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Registro de Empleados

Nuevo **Activar** Desactivar Actualizar

Busqueda

Estados: Desactivados Nombre: Buscar

Nombre	Teléfono	Dirección	Profesión	Cargo	Exten...	Depa...
Fernando Cruz		Solanda	Ing Siste...	Ing Si...		2

3.3. Desactivar: permite desactivar un empleado activo en el sistema

Cellar Active 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Registro de Empleados

Nuevo Activar **Desactivar** Actualizar

Busqueda

Estados: Activos Nombre: Buscar

Nombre	Teléfono	Dirección	Profesión	Cargo	Exten...	Depa...
Christian Aguilar	2343	Las Casas	Ing Siste...	Desar...	092	2
Cristian López	2758967	Las casas	Ing. Civil	Asesor	185	5
Damián	2858699	Suárez	Contador	Conta...	189	3
Pablo		Romero				5
Wilson	2785990	Estacio	Arquitecto	Jefe	178	1

3.4. Actualizar: Permite modificar los datos del empleado que maneja el activo fijo

The screenshot shows a window titled 'Empleado' with a green header 'Registro de Empleados'. The form contains the following data:

Nombre:	Pablo Romero		
Dirección:	Pio XII		
Profesión:	Lic. Ciencias Exactas		
Cargo:	Profesor		
Teléfono:	2858697	Extensión:	123
Departamento:	Matriz		

Buttons: Aceptar, Cerrar

4. Seguros: Permite el manejo de los seguros que posee el activo fijo

The screenshot shows the 'Sistema de Activos Fijos(CellarActive)' interface. The top header displays 'Usuario: márcos', 'Rol: Administrador', and 'Fecha: jue,jul-2009'. The main content area is titled 'Registro de Seguros' and includes 'Nuevo' and 'Actualizar' buttons. Below is a table with the following data:

Nombre
Seguros Colonial
Copp Seguros
Seguro Madera
Aseguradora Hermanos Naranja

At the bottom, there are 'Refrescar' and 'Salir del Sistema' buttons, and a 'Cerrar Sesión' button in the bottom right corner.

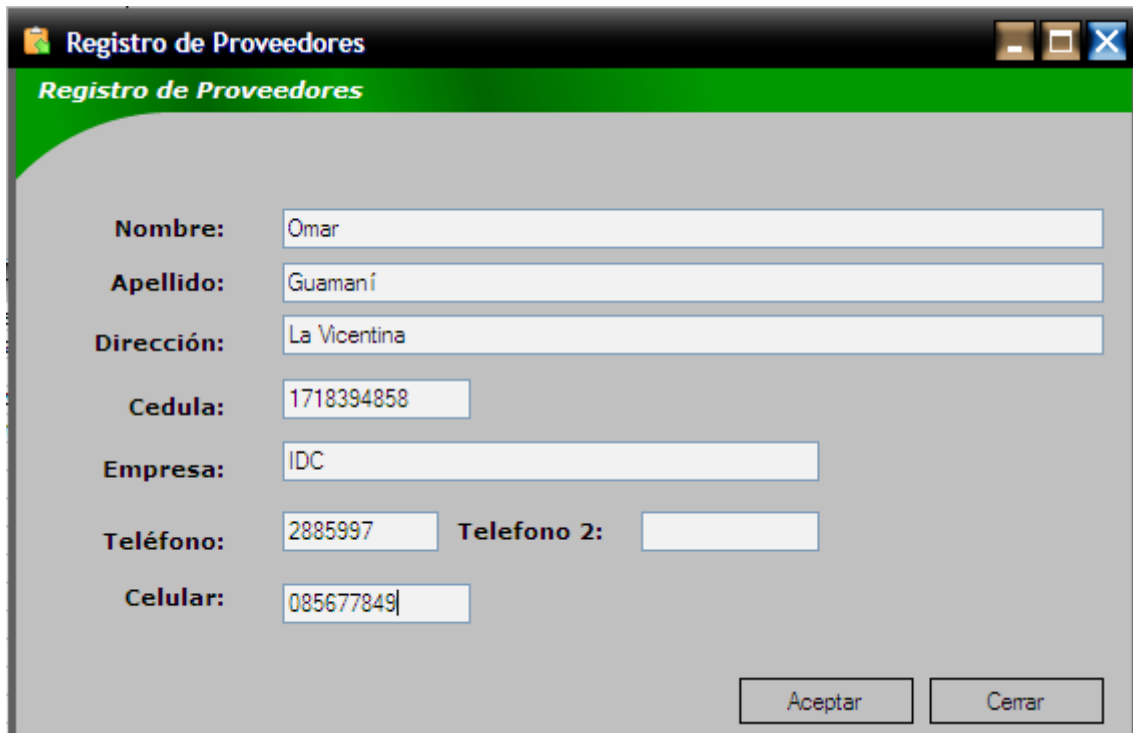
4.1. Nuevo: Permite registrar la aseguradora que da la garantía del activo fijo

4.2. Actualizar: Permite modificar el nombre de la aseguradora

5. Proveedores: Permite el manejo de las empresas que proveen de los activos fijos

Nombre	Apellido	Cedula	Empr...	Telef...	Telef...	Celular	Direccion
Christian Rodrigo	Poveda Aguilar	1111	PVNO...	00009	9090	99888	Brasilia 5
Espinoza	Espinoza	000	000	00	00	0	00
Jose	Jose	090	9090	909	90	90	90
Josue Aninal	Aguilar Poveda	099	ADAM...	24036...	098878	098655	Brasilia
Luis Almeida	Almeida Donoso	098887	KIA	24343	9867	09876...	8765
Omar	Guamani	17183...	IDC	28859...		08567...	La Vicentina

5.1. Nuevo: Permite registrar un proveedor

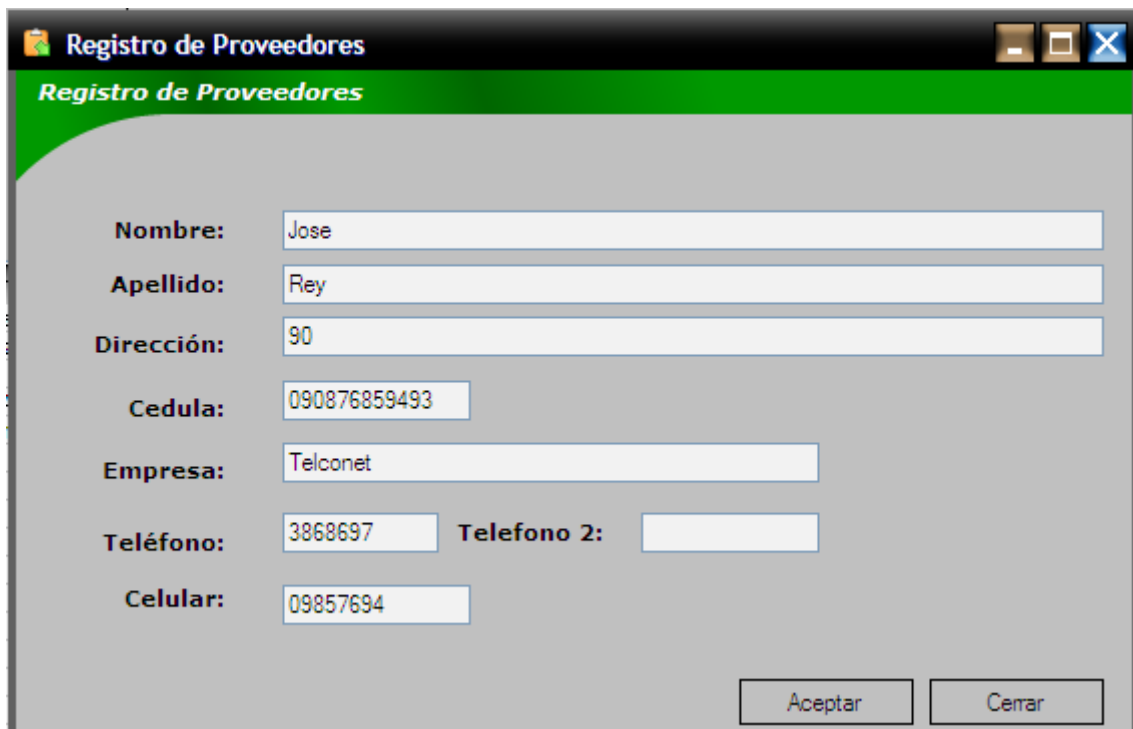


The screenshot shows a software window titled "Registro de Proveedores" with a green header bar. The form contains the following fields:

Nombre:	Omar		
Apellido:	Guamaní		
Dirección:	La Vicentina		
Cedula:	1718394858		
Empresa:	IDC		
Teléfono:	2885997	Telefono 2:	
Celular:	085677849		

Buttons: Aceptar, Cerrar

5.2. Actualizar: Permite modificar los datos de un proveedor



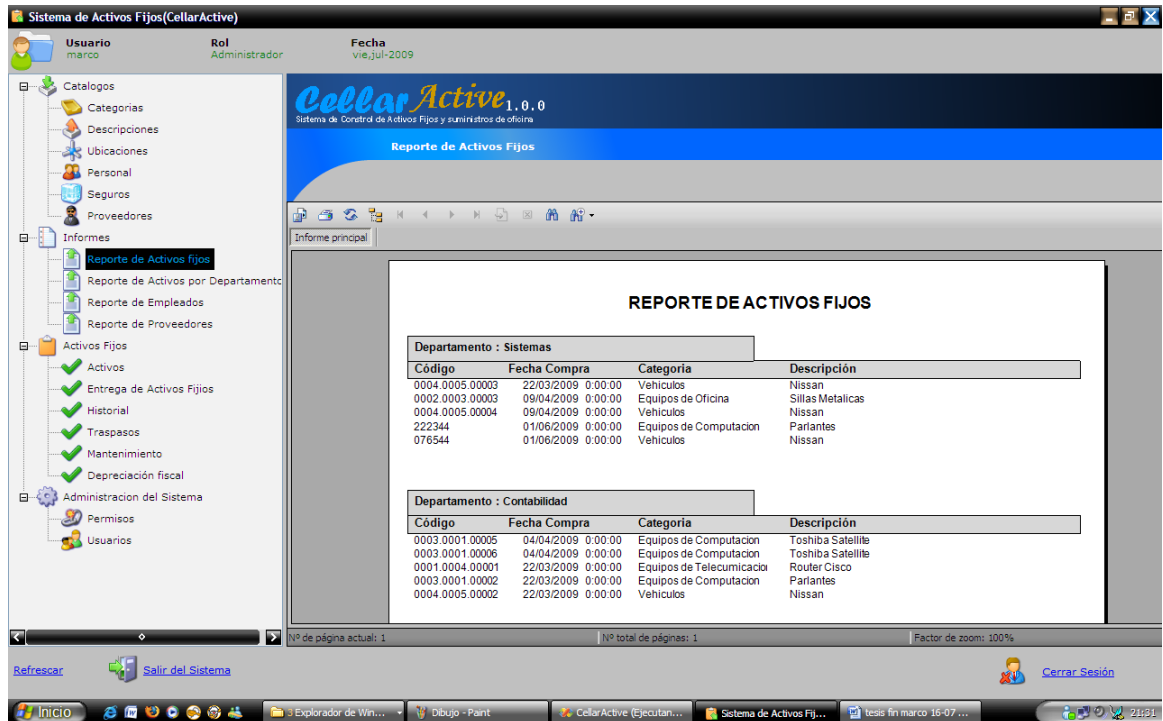
The screenshot shows the same software window "Registro de Proveedores" with the following updated data:

Nombre:	Jose		
Apellido:	Rey		
Dirección:	90		
Cedula:	090876859493		
Empresa:	Telconet		
Teléfono:	3868697	Telefono 2:	
Celular:	09857694		

Buttons: Aceptar, Cerrar

6. Informes: Muestra un detalle de diferente información del sistema

6.1. Reporte de activos fijos



Sistema de Activos Fijos(CellarActive)

Usuario: marcos Rol: Administrador Fecha: vie,jul-2009

Cellar Active 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Reporte de Activos Fijos

Informe principal

REPORTE DE ACTIVOS FIJOS

Departamento : Sistemas

Código	Fecha Compra	Categoría	Descripción
0004.0005.00003	22/03/2009 0:00:00	Vehiculos	Nissan
0002.0003.00003	09/04/2009 0:00:00	Equipos de Oficina	Sillas Metalicas
0004.0005.00004	09/04/2009 0:00:00	Vehiculos	Nissan
222344	01/06/2009 0:00:00	Equipos de Computacion	Parlantes
076544	01/06/2009 0:00:00	Vehiculos	Nissan

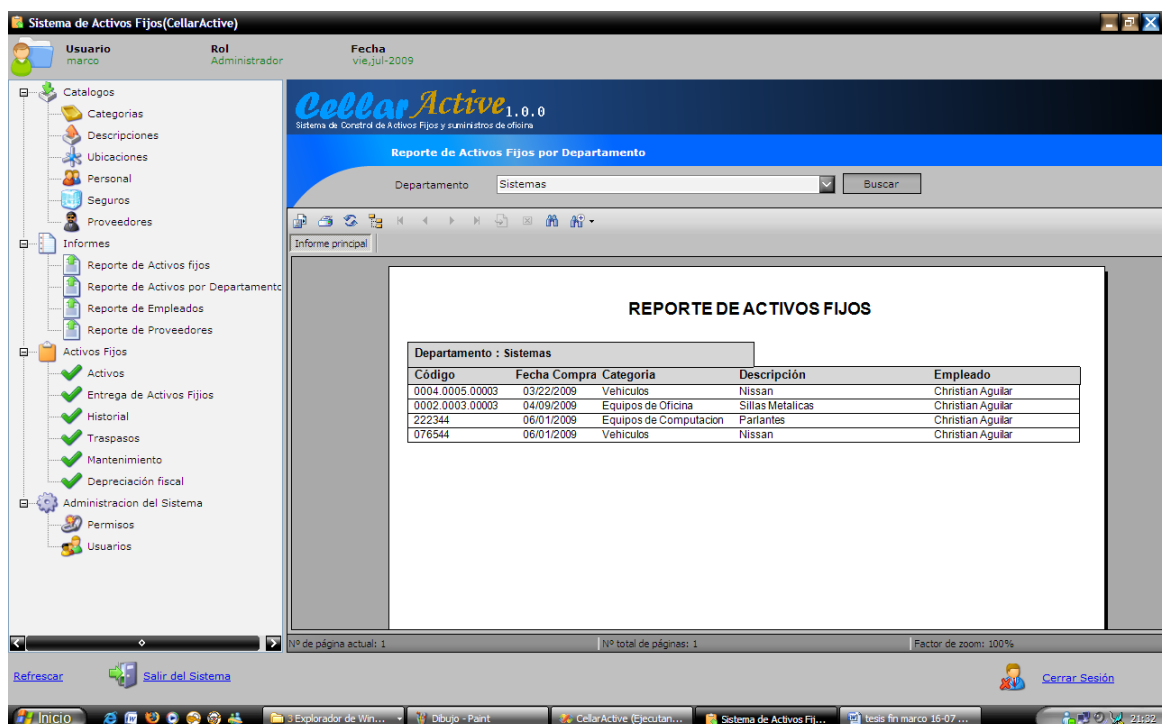
Departamento : Contabilidad

Código	Fecha Compra	Categoría	Descripción
0003.0001.00005	04/04/2009 0:00:00	Equipos de Computacion	Toshiba Satellite
0003.0001.00006	04/04/2009 0:00:00	Equipos de Computacion	Toshiba Satellite
0001.0004.00001	22/03/2009 0:00:00	Equipos de Telecomunicacion	Router Cisco
0003.0001.00002	22/03/2009 0:00:00	Equipos de Computacion	Parlantes
0004.0005.00002	22/03/2009 0:00:00	Vehiculos	Nissan

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Refrescar Salir del Sistema Cerrar Sesión

6.2. Informe de activos por departamento



Sistema de Activos Fijos(CellarActive)

Usuario: marcos Rol: Administrador Fecha: vie,jul-2009

Cellar Active 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Reporte de Activos Fijos por Departamento

Departamento: Sistemas Buscar

Informe principal

REPORTE DE ACTIVOS FIJOS

Departamento : Sistemas

Código	Fecha Compra	Categoría	Descripción	Empleado
0004.0005.00003	03/22/2009	Vehiculos	Nissan	Christian Aguilar
0002.0003.00003	04/09/2009	Equipos de Oficina	Sillas Metalicas	Christian Aguilar
222344	06/01/2009	Equipos de Computacion	Parlantes	Christian Aguilar
076544	06/01/2009	Vehiculos	Nissan	Christian Aguilar

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Refrescar Salir del Sistema Cerrar Sesión

6.3. Reporte de empleados

Sistema de Activos Fijos(CellarActive)

Usuario: marco | Rol: Administrador | Fecha: vie,jul-2009

Reporte de de Empleados

REPORTE DE FUNCIONARIOS

Nombres	Dirección	Profesión	Cargo	Ext	Departamento
Christian Aguilar	Las Casas	Ing Sistemas	Desarrollador	002	Sistemas
Christian López	Las casas	Ing. Civil	Asesor	185	Bodega
Damián	Suárez	Contador	Contador	189	Contabilidad
Fernando Cruz	Solanda	Ing Sistemas	Ing Sistemas		Sistemas
Pablo Romero	Pio XII	Lic. Ciencias Exactas	Profesor	123	Matriz
Wilson	Estacio	Arquitecto	Jefe	178	Matriz

Nº de página actual: 1 | Nº total de páginas: 1 | Factor de zoom: 100%

Refrescar | Salir del Sistema | Cerrar Sesión

6.4. Reporte de Proveedores

Sistema de Activos Fijos(CellarActive)

Usuario: marco | Rol: Administrador | Fecha: vie,jul-2009

Reporte de de Proveedores

REPORTE DE PROVEEDORES

Nombre	Apellido	Cédula	Empresa	Teléfono	Teléfono(2)	Dirección	Celular
Christian Rodrigo	Poveda Aguilar	1111	PYNOL II	00009	9090	Brasilia 5	99888
Josue Aninal	Aguilar Poveda	099	ADAMSYS	2403678	098878	Brasilia	098655
Jose	Rey	09087685949	Telconet	3868697		La Floresta	09857694
Luis Almeida	Almeida Donoso	098887	KIA	24343	9887	8765	098766544
Espinoza	Espinoza	000	000	00	00	00	0
Omar	Guamaní	1718394858	IDC	2885997		La Vicentina	085677849

Nº de página actual: 1 | Nº total de páginas: 1 | Factor de zoom: 100%

Refrescar | Salir del Sistema | Cerrar Sesión

7. Activos: Permite el manejo de los activos fijos

Registro de Activos Fijos

Busqueda:

Categorias: ---Seleccione una opción

Descripcion: ---Seleccione una opción

Proveedor: ---Seleccione una opción

Buscar

Codigo	Descripcion	Categoria	Fecha Compra	Proveedor	Documento	Valor...
0004.0005.00...	Nissan	Vehiculos	2008-12-19	Christian Rodrigo		\$0.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-03-22	Luis Almeida	1200	\$200.00
0004.0005.00...	Nissan	Vehiculos	2009-03-22	Christian Rodrigo		\$0.00
0003.0001.00...	Parlantes	Equipos de Computacion	2009-03-22	Espinoza	1234	\$1200...
0004.0005.00...	Nissan	Vehiculos	2009-03-22	Josue Aninal	12000	\$1200...
0001.0004.00...	Router Cisco	Equipos de Telecomunicaciones	2009-03-22	Christian Rodrigo	1200	\$1200...
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-03-28	Luis Almeida	324	\$43.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	49	\$2000...
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	3w2	\$1200...
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	343	\$32.00
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-04-04	Christian Rodrigo	343	\$12.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	888	\$0.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	S	\$22.00
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-04-09	Christian Rodrigo	1000	\$1000...
0004.0005.00...	Nissan	Vehiculos	2009-04-09	Christian Rodrigo		\$0.00
0004.0005.00...	Nissan	Vehiculos	2009-04-09	Christian Rodrigo	6666	\$77.00
0004.0005.00...	Nissan	Vehiculos	2009-04-09	Christian Rodrigo	1000	\$1200...
222344	Parlantes	Equipos de Computacion	2009-06-01	Espinoza	mobile	\$0.00
076544	Toshiba Satellite	Equipos de Computacion	2009-06-01	Espinoza	mobile	\$0.00

7.1. Nuevo: Permite el registro de un nuevo activo fijo

Registro del Activo

Registro de Activos Fijos

Categoria: 0002-Equipos de Oficina

Descripcion: 0003-Sillas Metalicas

Proveedor: Christian Rodrigo

Fecha Compra: 30/07/2009 Valor Compra: 200 Cantidad: 1

Documento/Factura: 4858

Características:

Tipo: Mueble

Guardar Cancelar

7.2. Actualizar: permite modificar los datos de un activo fijo

Registro del Activo

Registro de Activos Fijos

Categoría: 0004-Vehiculos

Descripción: 0005-Nissan

Proveedor: Espinoza

Fecha Compra: 01/06/2009 Valor Compra: 0.000 Cantidad: 1

Documento/Factura: mobile

Características:

Placa: pnh-7865

Tipo de Vehiculo: Automovil

Color: Negro

Numero Chasis: 573489klr4

Guardar Cancelar

8. Entrega de Activos Fijos: Permite el manejo de las entregas de los activos fijos

Sistema de Activos Fijos(CellarActive)

Usuario: marco Rol: Administrador Fecha: jue,jul-2009

CellarActive 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Entrega de Activos Fijos

Solicitud de Entrega de Activos Fijos

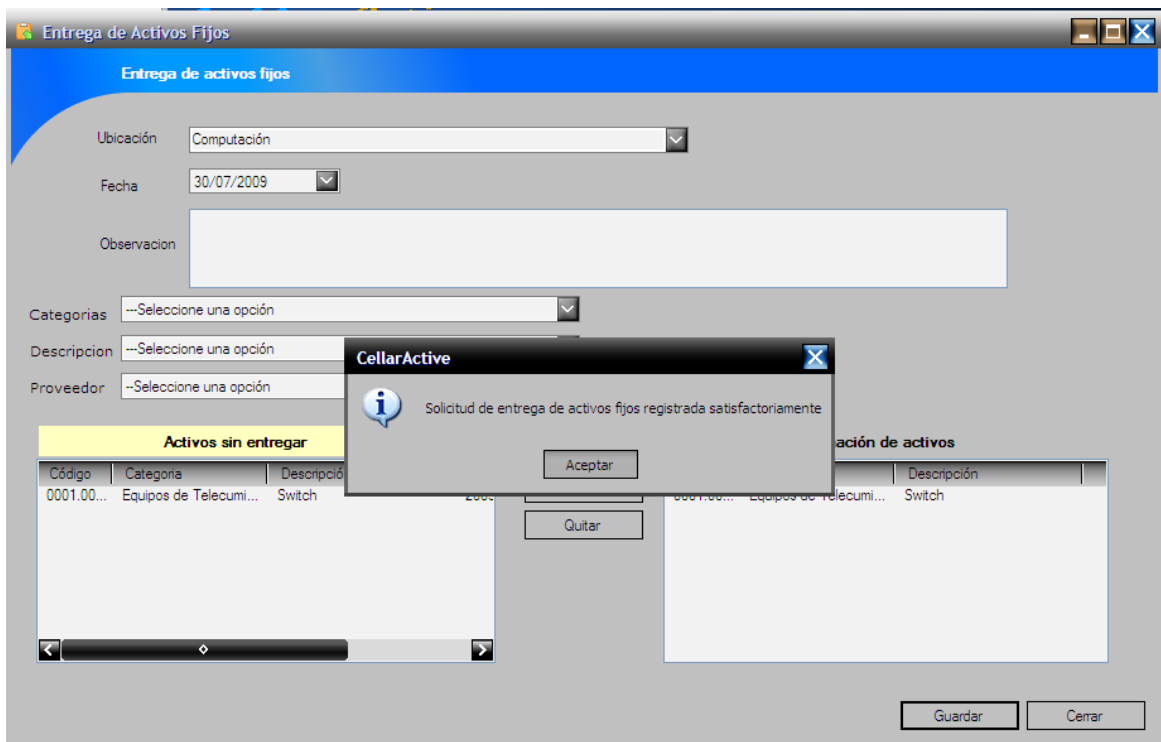
Nuevo Aprobar Solicitud Eliminar Solicitud Imprimir

Ubicacion: --Selección una opción-- Buscar

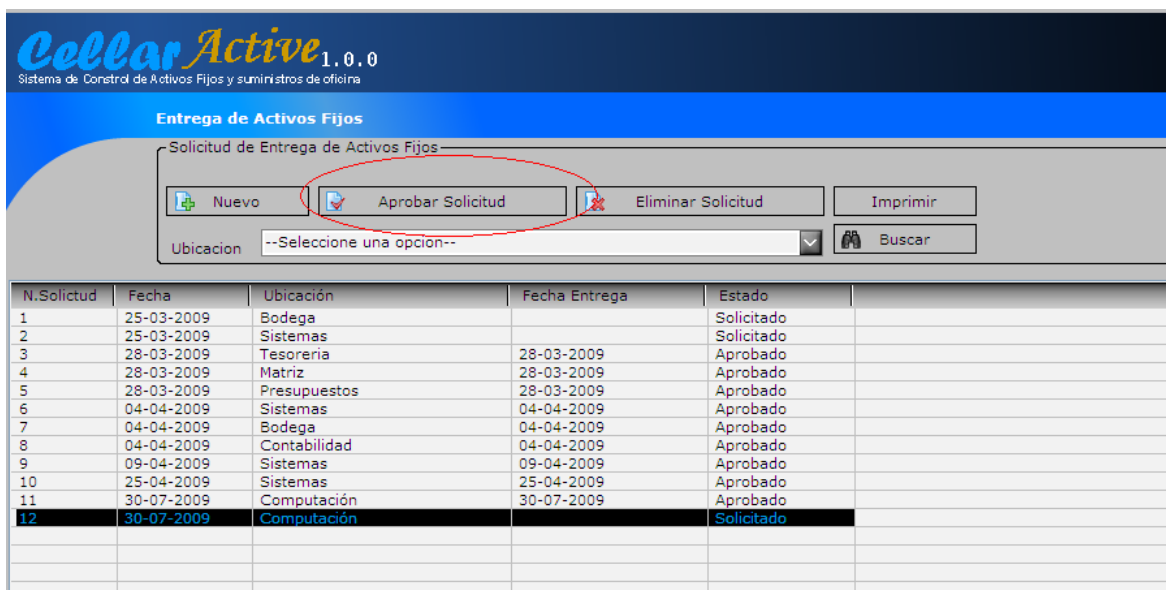
N.Solicitud	Fecha	Ubicación	Fecha Entrega	Estado
1	25-03-2009	Bodega		Solicitado
2	25-03-2009	Sistemas		Solicitado
3	28-03-2009	Tesoreria	28-03-2009	Aprobado
4	28-03-2009	Matriz	28-03-2009	Aprobado
5	28-03-2009	Presupuestos	28-03-2009	Aprobado
6	04-04-2009	Sistemas	04-04-2009	Aprobado
7	04-04-2009	Bodega	04-04-2009	Aprobado
8	04-04-2009	Contabilidad	04-04-2009	Aprobado
9	09-04-2009	Sistemas	09-04-2009	Aprobado
10	25-04-2009	Sistemas	25-04-2009	Aprobado

Refrescar Salir del Sistema Cerrar Sesión

8.1. Nuevo: Genera una solicitud para ser aprobada y entregar el activo fijo a un departamento



8.2. Aprobar solicitud: Permite dar el visto bueno a un departamento para que el activo fijo sea entregado



8.3. Eliminar solicitud: Permite negar una solicitud a un departamento

CellarActive 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Entrega de Activos Fijos

Solicitud de Entrega de Activos Fijos

Nuevo Aprobar Solicitud **Eliminar Solicitud** Imprimir

Ubicacion --Seleccione una opcion-- Buscar

N.Solicitud	Fecha	Ubicación	Fecha Entrega	Estado
1	25-03-2009	Bodega		Solicitado
2	25-03-2009	Sistemas		Solicitado
3	28-03-2009	Tesorería	28-03-2009	Aprobado
4	28-03-2009	Matriz	28-03-2009	Aprobado
5	28-03-2009	Presupuestos	28-03-2009	Aprobado
6	04-04-2009	Sistemas	04-04-2009	Aprobado
7	04-04-2009	Bodega	04-04-2009	Aprobado
8	04-04-2009	Contabilidad	04-04-2009	Aprobado
9	09-04-2009	Sistemas	09-04-2009	Aprobado
10	25-04-2009	Sistemas	25-04-2009	Aprobado
11	30-07-2009	Computación	30-07-2009	Aprobado
12	30-07-2009	Computación	30-07-2009	Eliminado

9. Historial: Permite visualizar un detalle de los procesos que se realizan en el manejo de un activo fijo.

Sistema de Activos Fijos(CellarActive)

Usuario marco Rol Administrador Fecha vie,jul-2009

CellarActive 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Historial de Activos Fijos

Categorias --Equipos de Telecomunicaciones--
Descripcion ---Seleccione una opción---
Proveedor ---Seleccione una opción--- Buscar

Codigo	Descripcion	Categoria	Fecha Compra	Proveedor
0001.0004.00001	Router Cisco	Equipos ...	2009-03-22	Christian Rodrig...
0001.0008.00001	Switch	Equipos ...	2009-07-30	Omar

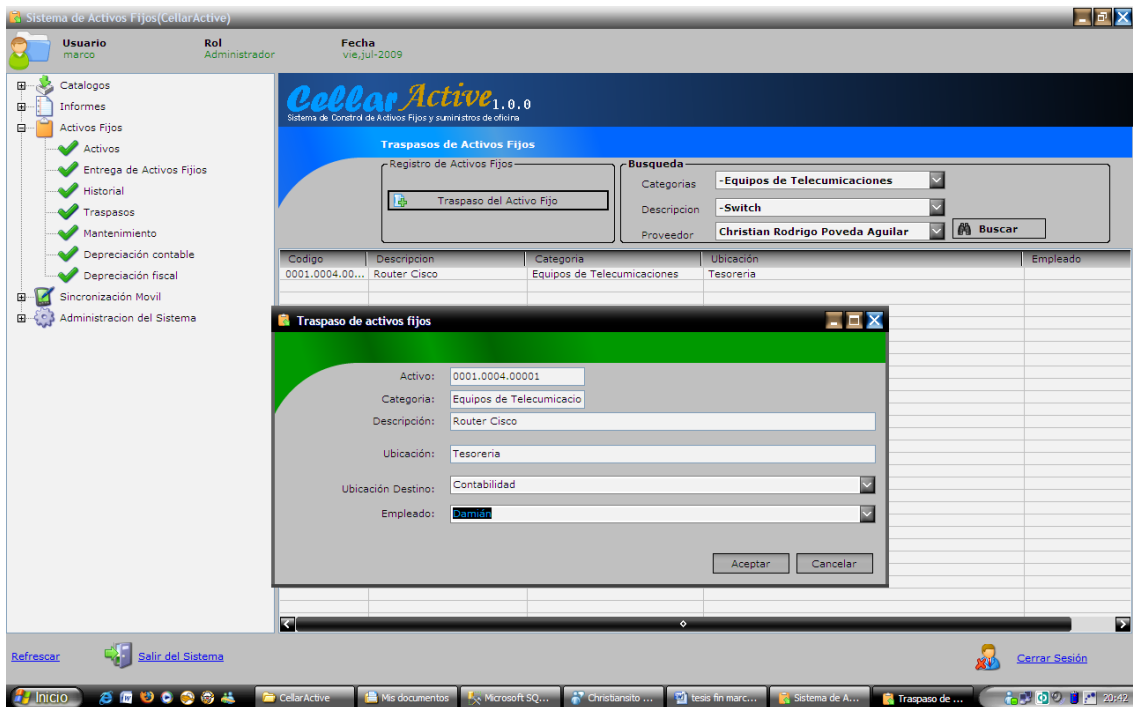
Historial Reporte

Generar reporte

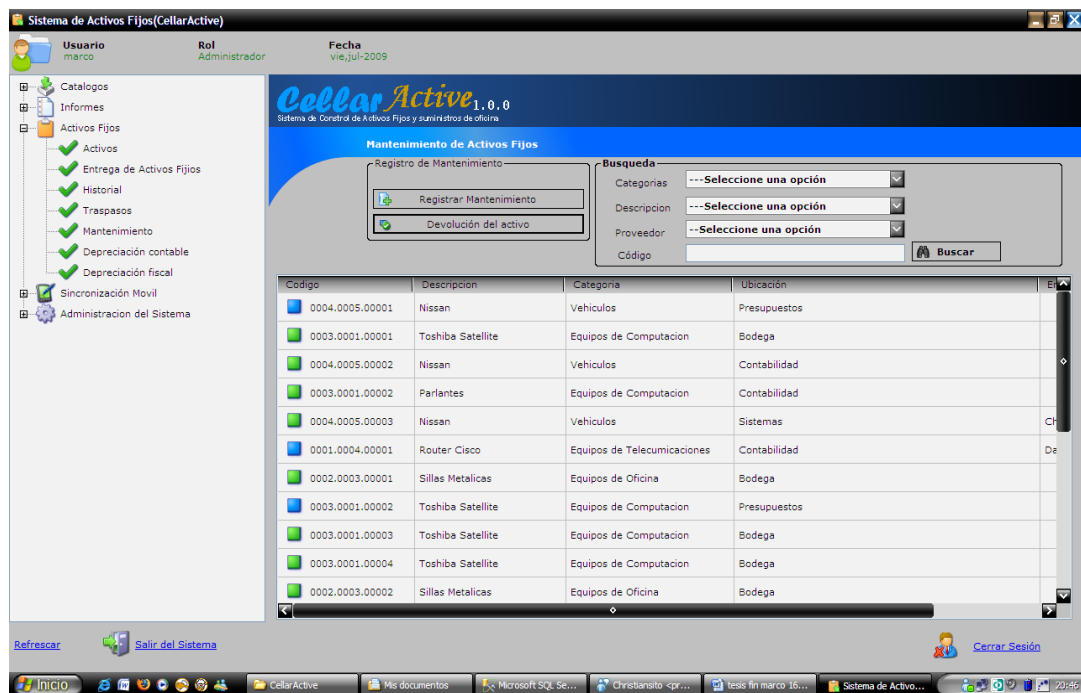
Usuario	Tipo	Fecha	Observacion
chris	Ingreso	22-03-2...	Carga Inicial
chris	Eliminacion de...	28-03-2...	Eliminacion de solicitud de A...
chris	SolicitudDeEnt...	28-03-2...	Solicitud N.4
chris	Entrega a Dep...	28-03-2...	Entrega de Activo al departa...
chris	Mantenimiento	18-04-2...	Activo en Mantenimiento ghr...

Refrescar Salir del Sistema Cerrar Sesión

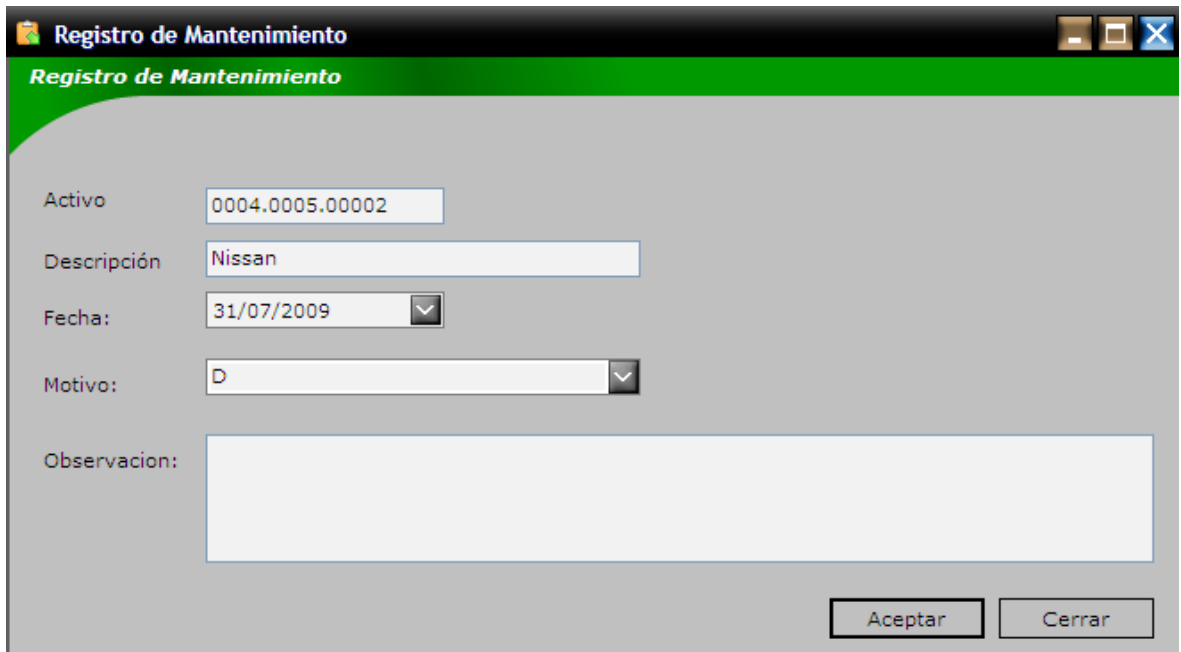
10. Traspasos: Permite realizar un cambio de ubicación del activo fijo hacia otro departamento



11. Mantenimiento: Permite registrar que activos fijos se están reparando y los que han sido devueltos luego de su arreglo.



11.1. Registrar Mantenimiento: Registra el ingreso de un activo fijo para una reparación

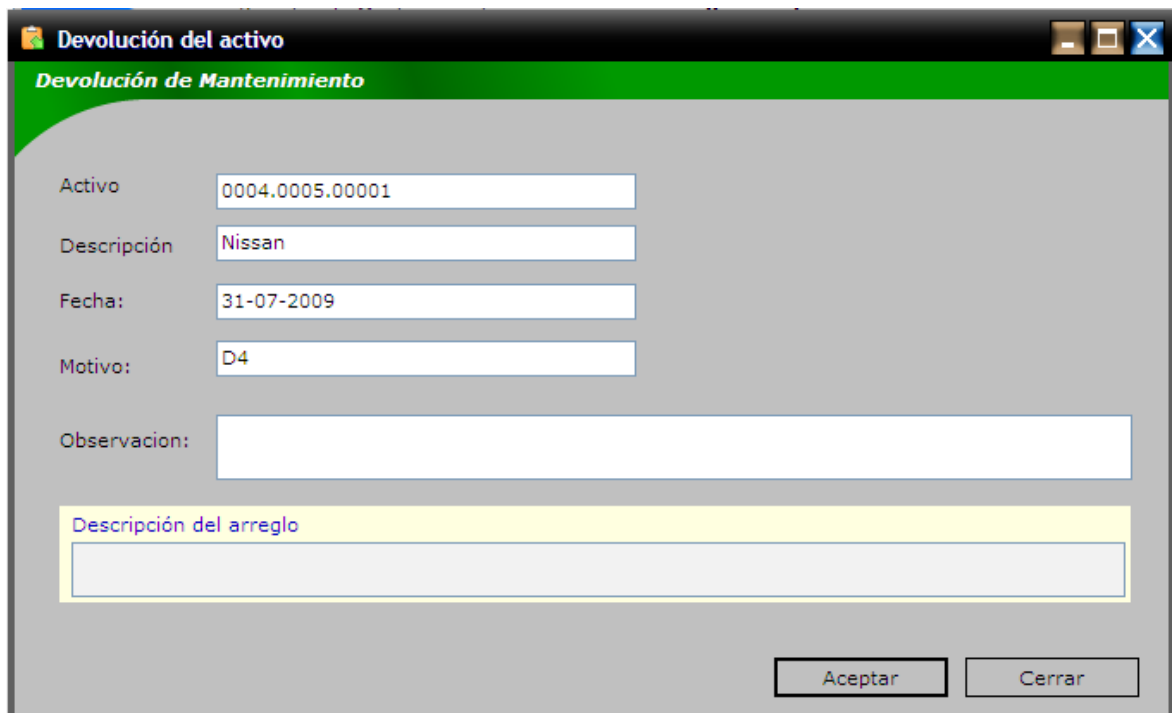


The screenshot shows a software window titled "Registro de Mantenimiento" with a green header bar. The form contains the following fields:

- Activo: 0004.0005.00002
- Descripción: Nissan
- Fecha: 31/07/2009 (dropdown menu)
- Motivo: D (dropdown menu)
- Observación: (empty text area)

Buttons at the bottom right: "Aceptar" and "Cerrar".

11.2. Devolución del activo: Indica que el activo fijo ha sido reparado y entregado



The screenshot shows a software window titled "Devolución del activo" with a green header bar. The form contains the following fields:

- Activo: 0004.0005.00001
- Descripción: Nissan
- Fecha: 31-07-2009
- Motivo: D4
- Observación: (empty text area)
- Descripción del arreglo: (empty text area with a yellow highlight)

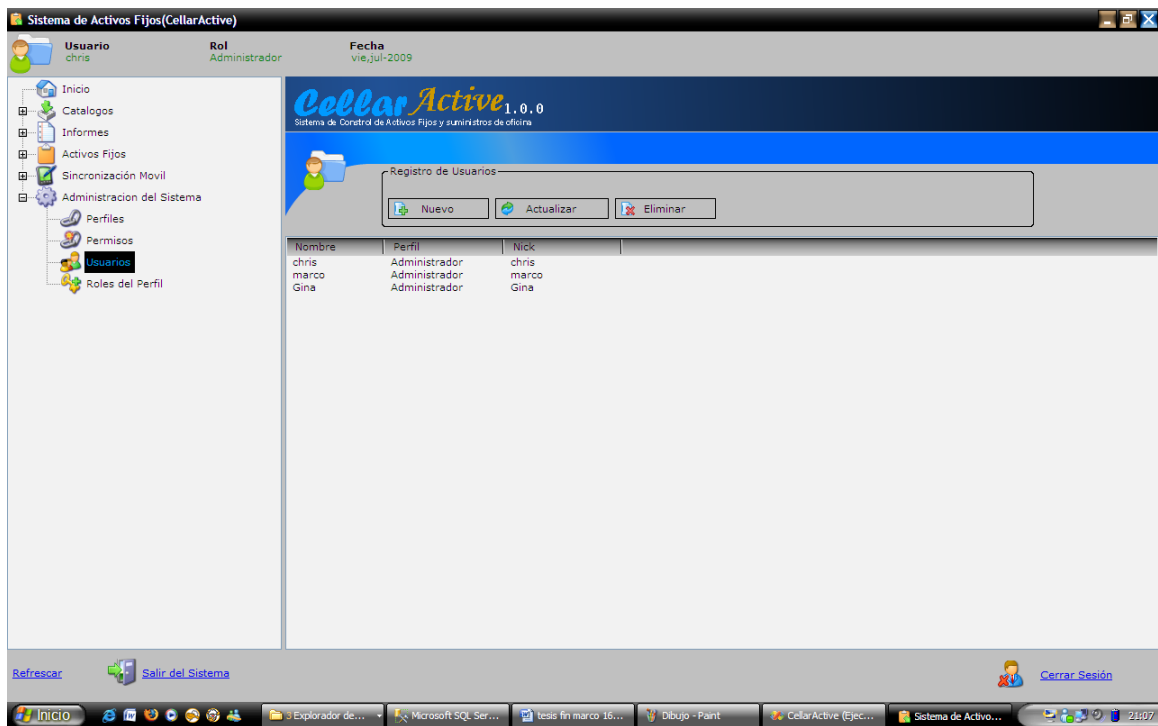
Buttons at the bottom right: "Aceptar" and "Cerrar".

12. Depreciación: Permite visualizar el valor residual y la depreciación que ha sufrido un activo fijo

Codigo	Descripcion	Categoria	Fecha Co...	Proveedor	Valor residual	Depreciación	V.Co...
0004.0005.00...	Nissan	Vehiculos	2008-12-19	Christian Rodrigo	\$0.00	\$0.00	\$0.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-03-22	Luis Almeida	\$66.57	\$44.44	\$200.00
0004.0005.00...	Nissan	Vehiculos	2009-03-22	Christian Rodrigo	\$0.00	\$0.00	\$0.00
0003.0001.00...	Parlantes	Equipos de Computacion	2009-03-22	Espinosa	\$4000.00	\$2666.67	\$1200...
0004.0005.00...	Nissan	Vehiculos	2009-03-22	Josee Animal	\$120.00	\$108.00	\$1200...
0001.0004.00...	Router Cisco	Equipos de Telecomunicaciones	2009-03-22	Christian Rodrigo	\$400.00	\$266.67	\$1200...
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-03-28	Luis Almeida	\$8.60	\$6.88	\$43.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	\$666.67	\$444.44	\$2000...
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	\$4000.00	\$2666.67	\$1200...
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	\$10.67	\$7.11	\$32.00
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-04-04	Christian Rodrigo	\$2.40	\$1.92	\$12.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	\$0.00	\$0.00	\$0.00
0003.0001.00...	Toshiba Satellite	Equipos de Computacion	2009-04-04	Christian Rodrigo	\$7.33	\$4.99	\$22.00
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-04-09	Christian Rodrigo	\$200.00	\$160.00	\$1000...
0004.0005.00...	Nissan	Vehiculos	2009-04-09	Christian Rodrigo	\$0.00	\$0.00	\$0.00
0004.0005.00...	Nissan	Vehiculos	2009-04-09	Christian Rodrigo	\$7.70	\$6.93	\$77.00
0004.0005.00...	Nissan	Vehiculos	2009-04-09	Christian Rodrigo	\$1200.00	\$1080.00	\$1200...
22344	Parlantes	Equipos de Computacion	2009-06-01	Espinosa	\$0.00	\$0.00	\$0.00
076544	Nissan	Vehiculos	2009-06-01	Espinosa	\$0.00	\$0.00	\$0.00
0002.0003.00...	Sillas Metalicas	Equipos de Oficina	2009-07-30	Christian Rodrigo	\$40.00	\$32.00	\$200.00
0001.0008.00...	Switch	Equipos de Telecomunicaciones	2009-07-30	Omar	\$10.00	\$6.67	\$30.00

13. Permisos: Permite dar diferentes tipos de acceso a cada usuario que maneje el sistema CellarActive

14. Usuarios: Permite ingresar, modificar y eliminar usuarios que manejen el sistema CellarActive.



Sistema de Activos Fijos(CellarActive)

Usuario: chris Rol: Administrador Fecha: vie, jul 2, 2009

CellarActive 1.0.0
Sistema de Control de Activos Fijos y suministros de oficina

Registro de Usuarios

Nuevo Actualizar Eliminar

Nombre	Perfil	Nick
chris	Administrador	chris
marco	Administrador	marco
Gina	Administrador	Gina

Refrescar Salir del Sistema Cerrar Sesión

Inicio Explorador de... Microsoft SQL Ser... tesis fin marco 16... Dibujo - Paint Cellar Active (Ejec... Sistema de Activo... 21:07

CAPITULO 5.

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Después de haber finalizado el presente trabajo se pudo concluir que para la institución que es objeto de estudio, ha sido de gran relevancia la creación y utilización del sistema informático para el manejo y control de información de activos fijos mediante tecnología móvil.

Se ilustraron los métodos empleados para validar el sistema, se debe señalar que se tiene un enfoque hacia el usuario, por lo que su manejabilidad jugó un papel importante en las modificaciones y mejoras que se llevaron a cabo.

El sistema fue evaluado por los usuarios alrededor de 4 semanas, la interacción de ingresos y entregas de los activos a los diferentes departamentos del establecimiento no generaron errores graves, en general los problemas que se enfrentaron fue al familiarizarse en la interacción sistema (fijo o móvil) – usuario.

Finalmente con este programa se pudo lograr los siguientes aspectos que se mencionan a continuación:

- Se realizó un análisis para el desarrollo del sistema mediante la recolección de las necesidades de la institución y las perspectivas que tiene el cliente en base al sistema.
- Para la presentación del Sistema de Control y Manejo de Activos fijos se realizó una interfaz amigable, con las funcionalidades requeridas tales como la comunicación Wireless, fácil acceso y confiabilidad de los datos.
- Se crearon procedimientos, que permiten un apropiado control de la información, rigiéndose a las normas y estándares establecidos por la Institución.
- Se priorizó la seguridad, integridad, respaldo, disponibilidad y confiabilidad de los datos, para brindar un mejor servicio mediante la implementación de

políticas y procedimientos que se ajusten adecuadamente a los requerimientos de los usuarios.

- Se estableció e identificó los requisitos necesarios para el procesamiento de datos, para la elaboración del sistema CELLARACTIVE al aplicar los métodos de observación e interactuando con los actores dentro del proceso del manejo y control de Activos Fijos.
- Se estandarizó los diagramas y la documentación permitiendo realizar un adecuado trabajo de ingeniería, utilizando para ello el análisis y diseño OMT, para la aplicación de los diagramas UML, se utilizó la herramienta CASE Rational Rose.
- El sistema CELLARACTIVE permitió documentar el inventario de activo fijo para registrarlo por tipo de bien y año de operación (transferido, adquirido).

5.2 RECOMENDACIONES

Este programa puede ser de uso general para otras organizaciones siempre y cuando se realicen las adaptaciones necesarias de la organización en la cual se implementará.

- Es necesario, la realización de cursos de capacitación para las personas que van a interactuar con el sistema, realizando una adaptación a la nueva forma de trabajo, que facilitará y optimizará la ejecución de los procesos de manejo y control de los activos fijos.
- También se recomienda a los usuarios, de que sus claves son únicas e intransferibles, que mantengan su confidencialidad, para evitar inconvenientes internos de responsabilidad, sobre el uso adecuado del sistema.
- Los sistemas con tecnologías móviles deben proporcionar diagnósticos de innovación que conlleven a nuevas alternativas para mejorar procesos y servicios.

- Se recomienda que el Colegio Particular Prado tenga en mente siempre una visión a futuro mediante la implementación del sistema aquí presentado y desarrollado, utilizando estrategias de innovación tecnológica, para de este modo obtener a largo plazo los resultados deseados, en vez de pensar en soluciones momentáneas.

REFERENCIAS BIBLIOGRAFICAS

- **Chong, Esteban;** Contabilidad Intermedia, Tomo I, Estados financieros y cuentas del activo, Universidad del Pacífico, 1992.
- **Liberty, Jesse & Hurwitz, Dan;** Programación Con Asp.Net 2.0, Anaya O Reilly.
- **Korth, F. & Silberschatz, Abraham;** Fundamentos de bases de Datos, Henry, McGraw-Hill.
- **Gitman Pearson, Lawrence J.;** Principios Administración Financiera, 3ra Edición, 2003.
- **Madura, Jeff;** Tesis Administración Financiera, Universidad San Francisco de Quito, 2001.
- **Elizondo, Arturo;** Tesis Procesos Contable, Universidad San Francisco de Quito, 2000.
- **Guevara Albornoz, Javier Mauricio & Olmedo Baldeón, Verónica Alexandra;** Metodología, recodificación y control de activos fijos de una empresa de servicios petroleros, PUCE 2002.
- **Scott, Building;** XML Web Services for the Microsoft .NET Platform, Microsoft Press 2002.
- **Conallen, Jim Building;** Web applications with UML, Addison Wesley 2002.
- **Oiris, Sofía;** Gestión de Proyectos de Software Curso, Octubre 2002.
- **Wesley, Adisson;** Software Engineering, 6º Edición.
- **Schmuller, Joseph;** Aprendiendo Uml en 24 horas, Prentice Hall.
- **Alarcón, Raúl;** Diseño orientado a objetos con UML, Grupo EIDOS Madrid España
- <http://www.sri.gov.ec>

ANEXOS

ANEXO 1

- **Administrador Financiero:** el administrador Financiero del Colegio Particular Prado es la Sra. Rosmery Galvez.
- **Asistente Administrativa:** la Sra. Cecilia Coba encargada de contestar el teléfono y la encargada de llevar apuntes en las reuniones que realizan en todo el departamento, como también de recoger los informes de cada departamento lo que han hecho durante el mes.
- **Colectora:** la Sra. Elena Rubio encargada de recibir por parte de las empresas ganadoras de las licitaciones uniformes, libros y útiles escolares.
- **Contadora:** la Sra. Lucia Arellano se encarga de controlar los asientos contables de la institución
- **Transporte:** el Sr. Carlos Castro encargado de matricular el vehículo de la institución y elabora un cronograma de actividades para todo el año. Además se encarga de darles indicaciones generales a los señores que hacen los recorridos a los alumnos.
- **Área de Servicio:** el Sr. Javier Valencia delegado para hacer una limpieza general en toda la institución cada día para conservar una buena imagen. Y al final del año se encarga de pintar lo interno y lo externo de la institución.