



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

OPTIMIZACIÓN DE LOS PROCESOS DE ACTIVOS FIJOS MEDIANTE UNA
APLICACIÓN INFORMÁTICA WEB PARA LA EMPRESA ABS COMPUTER S.
A EN LA CIUDAD DE QUITO.

Proyecto de investigación previo a la obtención del título de

Tecnólogo en Análisis de Sistemas

Autor: Cauja Pilataxi Guillermo Patricio

Tutor: Ing. Marco Obando

Quito, Abril 2015

DECLARACION DE APROBACION DE TUTOR Y LECTOR

En mi calidad de tutor del trabajo sobre el tema: OPTIMIZACIÓN DE LOS PROCESOS DE ACTIVOS FIJOS MEDIANTE UNA APLICACIÓN INFORMÁTICA WEB PARA LA EMPRESA ABS COMPUTERS S. A EN LA CIUDAD DE QUITO, presentado por el ciudadano: Cauja Pilataxi Guillermo Patricio, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del tribunal de Grado, que el honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Abril del 2015

Ing. Marco Obando

TUTOR

Ing. Patricia Garzón

LECTOR

Declaratoria

Declaro que la investigación es absolutamente original, autentica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Cauja Pilataxi Guillermo Patricio

C.C. 1720735933

Contrato de Cesión sobre Derechos Propiedad Intelectual

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante CAUJA PILATAXI GUILLERMO PATRICIO, por sus propios y personales derechos, a quien en lo posterior se le denominará el "CEDENTE"; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el "CESIONARIO". Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado "OPTIMIZACIÓN DE LOS PROCESOS DE ACTIVOS FIJOS MEDIANTE UNA APLICACIÓN INFORMÁTICA WEB PARA LA EMPRESA ABS COMPUTER S.A EN LA CIUDAD DE QUITO", el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del

software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial

y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvencción, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los 6 días del mes de Abril del dos mil quince.

f) _____

f) _____

C.C. N° 1720735933

Instituto Superior Tecnológico Cordillera

CEDENTE

CESIONARIO

Agradecimiento

Gracias a Dios por darme fuerzas y salud para cada día superarme y lograr culminar una más de mis metas en mi vida.

A mis Padres y hermanos que con apoyo incondicional y sabiduría que me dieron en todo este transcurso hicieron de mí una persona luchadora y optimista para así lograr ser un apoyo a la sociedad.

A mi tutor de tesis y profesor Ing. Marco Obando, quien ha sabido guiarme con su ética profesional y sabiduría para la culminación de este proyecto.

Dedicatoria

A mis padres y hermanos que con sus consejos y paciencia
supieron guiarme y hacer de mí una persona de bien.

A mis amigos y profesores con quien compartí grandes
momentos de alegría y a veces de amargura

Índice General

Titulo	Página
Declaratoria de Aprobación de lector y Tutor.....	ii
Declaratoria.....	iii
Cesion de Derechos.....	iv
Agradecimiento.....	vii
Dedicatoria.....	viii
Indice General.....	ix
Indice de Figuras.....	xii
Indice de Tablas.....	xiv
Indice de Anexos.....	xvi
Resumen Ejecutivo.....	xvii
Abstract.....	xix
Capítulo I: Antecedentes.....	1
1.01 Contexto.....	1
1.02 Justificacion.....	2
1.03 Definico del Problema Central.....	3
Capítulo II: Analisis de Involucrados.....	5
2.01 Requerimientos.....	5
2.01.01 Descripcion Actual del Sistema.....	5
2.01.02 Vision y Alcance	5
2.01.02.01 Vision	5
2.01.02.02 Alcance	5
2.01.03 Entrevistas.....	6
2.01.04 Matriz de Requerimientos.....	6
2.01.05 Descripcion Detallada.....	7
2.02 Mapeo de Involucrados.....	18

2.03 Matriz de involucrados.....	19
Capítulo III: Problemas y Objetivos.....	20
3.01 Árbol de Problemas.....	20
3.02 Árbol de Objetivos.....	21
3.03 Diagrama de Caso de Uso.....	22
3.04 Casos de Uso de Realización.....	23
3.05 Diagrama de Secuencia General de Sistema.....	29
3.06 Especificación de Casos de uso.....	32
Capítulo IV: Analisis de Alternativas.....	35
4.01 Matriz de Analisis de Alternativas.....	35
4.02 Matriz de Impacto de Objetivo.....	36
4.03 Estándares para el Diseño de Clases.....	37
4.04 Diagrama de Clases.....	38
4.05 Modelo Lógico - Físico.....	39
4.06 Diagrama de Componentes.....	40
4.07 Diagrama de Estrategia.....	41
4.08 Matriz de Marco Lógico.....	42
4.09 Vistas Arquitectónicas.....	43
4.09.01 Vista Lógica.....	43
4.09.02 Vista Física.....	43
4.09.03 Vista de Desarrollo.....	44
4.09.04 Vista de Procesos.....	44
Capitulo V: Propuesta.....	45
5.01 Especificacion de Estandares de Programación.....	45
5.02 Diseño de Interfaces de Usuario.....	46
5.03 Especificación Pruebas de Unidad	56
5.04 Especificación Pruebas de Aceptación.....	65

5.05 Especificación Pruebas de Carga.....	68
5.06 Configuración del Ambiente Mínimo/Ideal.....	69
Capítulo VI: Aspectos Administrativos.....	70
6.01 Recursos.....	70
6.02 Presupuesto.....	71
6.03 Cronograma.....	72
Capítulo VII: Conclusiones y Recomendaciones.....	73
7.01 Conclusiones.....	73
7.02 Recomendaciones.....	73

Índice de Figuras

Fig 1 Mapeo de Involucrados.....	18
Fig 2 Arbol de Problemas.....	20
Fig 3 Arbol de Objetivos.....	21
Fig 4 Caso de uso General	22
Fig 5 UC Registro,Modificar.....	23
Fig 6 UC Asignacion Custodios.....	24
Fig 7 UC Generar Etiquetas.....	25
Fig 8 UC Autorizacion de Activos.....	26
Fig 9 UC Depreciacion Activos.....	27
Fig 10 UC Reporte de Activos.....	28
Fig 11 D.S Registro de Activos.....	29
Fig 12 D.S Etiqueta Activo Fijo.....	29
Fig 13 D.S Activo Depreciado.....	30
Fig 14 D.S Asignacion Custodio.....	30
Fig 15 D.S Salida Activo	31
Fig 16 D.S Reporte Activo.....	31
Fig 17 Diagrama de Clases.....	38
Fig 18 Modelo Logico.....	39
Fig 19 Modelo Fisico.....	40
Fig 20 Diagrama de Componentes.....	40
Fig 21 Diagrama de Estrategias.....	41
Fig 22 Vista Logica.....	43
Fig 23 Vista Fisica.....	43
Fig 24 Vista de Desarrollo.....	44
Fig 25 Vista de Proceso.....	44
Fig 26 Interfaz de Inicio... ..	46

Fig 27 Interfaz Ingreso Sistema.....	47
Fig 28 Ventana Registro Usuarios.....	48
Fig 29 Ventana registro de Activos.....	49
Fig 30 Ventana Registro Tipo de Activo.....	50
Fig 31 Ventana Ubicación de los Activos.....	51
Fig 32 Ventana de Registro de empleados.....	52
Fig 33 Ventana Depreciacion de Activo.....	53
Fig 34 Ventana Salida de Activos.....	54
Fig 35 Ventana Consulta de Activos.....	55
Fig 36 Cronograma.....	72

Índice de Tablas

Tabla 1 Análisis de fuerzas T.....	3
Tabla 2 Entrevistas.....	6
Tabla 3 RF001.....	7
Tabla 4 RF002.....	8
Tabla 5 RF003.....	9
Tabla 6 RF004.....	10
Tabla 7 RF005.....	11
Tabla 8 RF006.....	12
Tabla 9 RF007.....	13
Tabla 10 NRF001.....	14
Tabla 11 NRF002.....	15
Tabla 12 NRF003.....	16
Tabla 13 NRF004.....	17
Tabla 14 Matriz de Involucrados.....	19
Tabla 15 C.R Registrar, Modifica, Eliminar Activos Fijos.....	23
Tabla 16 C.R Asignación Custodios.....	24
Tabla 17 C.R Generar Etiqueta.....	25
Tabla 18 C.R Autorización Activos Fijos.....	26
Tabla 19 C.R Depreciación de Activos.....	27
Tabla 20 C.R Reporte Activos	28
Tabla 21 E.C.U Registro, Modificación de Activos Fijos.....	32
Tabla 22 E.C.U Etiqueta de Activos Fijos	32
Tabla 23 E.C.U Activo Depreciado.....	33
Tabla 24 E.C.U Asignación Custodio.....	33
Tabla 25 E.C.U Salida e ingresos de activos.....	34
Tabla 26 E.C.U Reporte de Activos Fijos.....	34

Tabla 27 Matriz de Análisis de Alternativas.....	35
Tabla 28 Matriz de Impacto de Objetivo.....	36
Tabla 29 Estándares para Diseño de clases.....	37
Tabla 30 Matriz de Marco Lógico.....	42
Tabla 31 Controles de Programación.....	45
Tabla 32 Variables de datos.....	46
Tabla 33 Pruebas de unidad Validación Usuario.....	56
Tabla 34 Registro Usuario.....	57
Tabla 35 Registro de Activos.....	58
Tabla 36 Registro Tipo de Activo.....	59
Tabla 37 Ubicación de Activos.....	60
Tabla 38 Vida Útil.....	61
Tabla 39 Depreciación de Activos.....	62
Tabla 40 Salida de Activos.....	63
Tabla 41 Consulta de activos.....	64
Tabla 42 Prueba Aceptación Usuarios.....	65
Tabla 43 Pruebas de Aceptación Registro de Activo.....	66
Tabla 44 Pruebas de Aceptación Baja de Activos.....	67
Tabla 45 Prueba de Carga de Usuarios.....	68
Tabla 46 Requerimientos de la aplicación.....	69
Tabla 47 Recursos.....	70
Tabla 48 Presupuesto.....	71

Índice de Anexos

A.01Requerimientos.....	76
A.02 Manual de Instalación.....	78
A.03 Manual de Usuario.....	101
A.04 Manual Técnico.....	111

Resumen Ejecutivo

Capítulo I: Antecedentes

Describe la manera que se manejan los procesos de la empresa ABS COMPUTER S.A Ubicada en el Norte de Quito, en la Avenida Riobamba N55-100 la cual tiene como actividad el brindar servicios de soluciones de mantenimientos a empresas, los activos que se usan para la labor diaria de los técnicos no se controlan adecuadamente por lo que la empresa esta propensa a grandes pérdidas económicas.

Capítulo II: Análisis de Involucrados

Se analiza, reconoce y recopila información de quienes se encuentran inmersos directa e indirectamente en el proceso para la elaboración de la aplicación informática donde se encuentran las fuerzas impulsadoras y fuerzas bloqueadoras del negocio.

Capítulo III: Problemas y Objetivos

Describe los problemas que posee empresa ABS COMPUTER S.A donde se detalla soluciones y objetivos para solucionar tales inconvenientes.

La solución propuesta se detalla en diferentes casos de uso quienes muestran paso a paso los procesos que dieron para la elaboración del sistema.

Capítulo IV: Análisis de Alternativas

Son las alternativas iniciales del diseño del proyecto enfocado en lo realizable donde se toma en cuenta los objetivos y alcance del sistema para tener una idea clara de los resultados a obtener.

Capítulo V: Propuesta

Para la implementación del sistema en el diseño y desarrollo se utiliza como lenguaje de programación Microsoft Visual Studio, a su vez de motor de base de datos SQL Server. En la arquitectura del sistema se usó el programa Rational Rouse.

Capítulo VI: Aspectos Administrativos

Para la elaboración del proyecto se hicieron cálculos de costos y para saber si era factible la elaboración del sistema informático para la empresa ABS COMPUTER S.A

Capítulo VII: Conclusiones y Recomendaciones

El funcionamiento del sistema se encuentra acorde al nivel de perfil de cada usuario para su manejo óptimo y automatizado de los activos fijos, el cual cumple con los requerimientos establecidos.

Para un correcto funcionamiento del sistema los usuarios deberán revisar los manuales de usuario.

Abstract

Chapter I: Background

Describe the way the business processes ABS COMPUTER SA Located in northern Quito, Riobamba N55-100 Avenue in which the activity is to provide maintenance solutions services businesses are managed , assets that are used for daily work of the technicians are not adequately controlled by what the company is prone to large economic losses.

Chapter II: Analysis Involved

It analyzes, recognizes and collects information from those who are directly and indirectly involved in the process for developing the software application where the driving forces blockading forces and business.

Chapter III: Challenges and Objectives

Describe the problems that has now ABS COMPUTER SA where solutions and aims to solve such problems is detailed.

The proposed solution is detailed in different use cases who show step by step process that led to the development of the system.

Chapter IV: Analysis of Alternatives

They are the initial project design alternatives focused on the achievable which takes into account the objectives and scope of the system to have a clear idea of the results you get.

Chapter V: Proposal

To implement the system design and development is used as the programming language Microsoft Visual Studio, in turn Database Engine SQL Server database. In the system architecture Rational Rouse program was used.

Chapter VI: Administrative Issues

For the preparation of project cost estimates were made and whether the development of computer system for the company ABS COMPUTER SA was feasible

Chapter VII: Conclusions and Recommendations

The operating system is commensurate to the level of each user profile for optimal and automated management of fixed assets, which complies with the requirements. For proper operation of the system, users should review the user manuals.

Capítulo I: Antecedentes

1.01 Contexto

La nueva era de la tecnología en la que nos encontramos inmersos a diario, se ha convertido en una de nuestras primordiales herramientas de trabajo por sus grandes beneficios en la hora de procesar grandes cantidades de información con gran rapidez y precisión.

La tecnología va evolucionando a grandes pasos en los últimos años por lo cual da mayor confiabilidad automatizar los procesos manuales y así lograr llevar un control seguro de los procesos que se realizan a diario en una empresa o industria.

Las empresas en la actualidad no cuentan con mecanismos de trabajo que les permitan llevar de manera eficaz y controlada los activos fijos de tal, es así que necesitan implantar aplicaciones informáticas que controlen cada uno de los activos de la empresa.

Los activos fijos son partes esenciales de una empresa ya que son los recursos que dispone la empresa y sirven para el trabajo cotidiano del personal y para brindar un buen servicio al cliente.

La empresa ABS COMPUTER S.A es una empresa que ofrece servicio técnico a varias empresas por lo cual poseen bienes en materiales y toda la información de ello se registra en hojas de papel y archivos de Excel existiendo confusiones y mal conteo de los activos.

La empresa se encuentra en un periodo de modernización por lo que el uso de papel, esferos gráficos, y más implementos para el trabajo habitual no será necesario para el día laboral de cada uno de los trabajadores de la empresa.

1.02 Justificación

La presente investigación permitirá resolver los problemas de conteo de activos fijos de la empresa y asignar a cada empleado activos que se crean necesarios para labor diaria de cada uno de ellos.

Además de la empresa se beneficiara el personal encargado de asignar custodios para cada uno de los activos fijos, y brindara un control estricto y ordenado de tales.

Automatizara el ámbito laboral el cual se encarga de todo el control como el ingreso y egreso de activos a la empresa y dar de baja a activos depreciados o destruidos que la empresa ya no les de uso.

El principal beneficio para la solución es la seguridad, rapidez y exactitud que hoy en día la empresa carecía en los procesos manuales, así también el personal que labora en la empresa podrá saber cuáles son sus herramientas de trabajo asignados a cada uno de ellos.

La empresa cuenta con módulos factibles para la realización del proyecto por lo cual aplicarla será fácil y sencillo.

Para la realización de este sistema informático la empresa nos orientara y facilitara los listados de todos los activos fijos que poseen en tal.

El sistema contara con una base de datos para la optimalización de los procesos y seguridad de los datos, impresión de reportes confiables y con estos brindara una mejor atención a los clientes.

1.03 Definición del Problema Central

En la presente matriz se describe las situaciones: empeorada, actual y mejorada mediante la cual describiremos los extremos a evitar, la problemática en el presente y lo que buscamos al hacer nuestra investigación. Además describe las fuerzas mediante la cual son calificadas con la intensidad y el potencial de campo en la escala del 1 al 5.

Tabla 1:

Análisis de fuerzas T

ANÁLISIS DE FUERZAS T					
Situación Empeorada	Situación Actual				Situación Mejorada
Perdida de activos fijos de la empresa	Perdida de materiales-herramientas				Facilitar una herramienta para el control de activos
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
El uso de la información es de manera manual	2	5	5	3	Falta de recursos económicos
Mejorar el ambiente de trabajo	1	5	4	2	Falta de distribución de personal
Facilitar la asignación de bienes a custodios	3	4	4	3	Inadecuado manejo de información
Facilitar la información de los bienes pertenecientes a los empleados.	1	4	4	5	Inadecuado registro de información de los activos fijos
Ahorrar tiempo en la búsqueda de bienes de la empresa	1	4	4	1	Tener la información en una base de datos
Perdida de activos fijos de la empresa	2	5	4	5	Mala información de activo

Nota: Significado de las siglas I = Intensidad - PC = Potencial de Cambio.

Los niveles de porcentaje se miden de la siguiente manera

1 = Bajo 2 = Medio Bajo 3 = Medio 4 = Medio Alto 5 = Alto

Análisis

La presente matriz de fuerzas T nos muestra el estado en el que se encuentra la empresa su situación de fuerzas por lo que nos da a entender la infraestructura de ella.

Se conoce el trabajo diario y así se podrá aportar una aplicación acorde a la necesidad que requiere la empresa y sus empleados.

La matriz está separada en rango de 1 a 5 ,siendo el rango 1 el cual nos da a notar las deficiencias más baja en la que se encuentra el proceso de inventario y necesita una mejora inmediata en otro caso el rango 5 nos da la plenitud del proceso y su mantenimiento adecuado .

Capítulo II: Análisis de Involucrados

2.01 Requerimientos

2.01.01 Descripción del Sistema Actual

La empresa ABS COMPUTER S.A lleva un control de los activos fijos lo cual se controla depreciaciones y salidas de los activos de forma manual en hojas de papel las que se encuentran almacenadas en carpetas en las oficinas de la empresa, a la hora de comprobar activos existentes en la empresa ocurren errores de organización de cada uno de los activos lo que dificulta saber los estados de cada uno de ellos y los activos que se dieron de baja por mal estado de tales.

2.01.02 Visión y Alcance

2.01.02.01 Visión.

Desarrollar una aplicación informática vía web que permita a la empresa ABS COMPUTER S.A llevar los registros, asignaciones, depreciaciones y salidas de cada uno de los activos fijos de la empresa de forma ordenada y sencilla, los custodios de los activos podrán estar informados de los bienes que se le fueron asignados y el estado que se encuentran para así evitar pérdidas de cada uno de los bienes.

2.01.02.02 Alcance.

Mediante esta aplicación se permitirá el fácil acceso a información que el usuario pueda requerir como los bienes a cargo de cada uno y la depreciación de tal bien.

Además llevar un registro controlado de los estados de los bienes por ubicaciones.

2.01.03 Entrevista

Tabla 2:

Entrevistas

Identificador: 001

Preguntas	Objetivos	Análisis Posterior
¿Cuál es el problema que se presenta en la empresa?	Conocer los problemas de la empresa y dar solución a tal.	<ol style="list-style-type: none"> 1. Automatizar el proceso de registro de los activos fijos. 2.- El sistema debe permitir ingresar, consultar y modificar los movimientos de ingreso y de egreso de los activos fijos en la empresa. 3.- El sistema deberá corregir errores de activos mal registrados por el personal a cargo de tal tarea. 4.- El sistema dará a conocer la ubicación y es estado de los activos fijos. 5- La aplicación deberá ejecutarse o ser compatible con cualquier navegador 6.- El sistema debe contar con una base de datos confiable y segura de vulnerar.

2.01.04 Matriz de requerimientos

Permite la identificación de requerimientos tanto funcionales como no funcionales que se generan en la empresa.

Ver Anexo A.01

2.01.05 Descripción detallada

Detalla los requerimientos identificados anteriormente en la descripción.

Tabla 3:

Identificador RF001

Registros de activos Fijos en base de datos	Estado	En revisión
Creado por:	Guillermo Cauja	Actualizado por: Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización: 2014-12-16
Identificador	RF 001	
Tipo de Requerimiento	Critico	Tipo de Requerimiento: Funcional
Datos de Entrada	Descripción del Activo Código del Activo Fijo Serie del Activo Fijo Fecha de compra del Activo Valor de compra del Activo Estado del Activo	
Descripción	Registrar en la base de datos los activos fijos de la empresa y su descripción.	
Datos de Salida	Etiqueta con serie del Activo	
Resultados Esperados	Tener una base de datos de los Activos para registrar su información y tener un control adecuado.	
Origen	Gerente	
Dirigido a	Administrador	
	Contador	
	Custodio	
	Secretaria	
Prioridad	5	
Requerimientos Asociados	Ninguno	
Especificación		
Precondiciones	Tener información de todos los activos fijo de la empresa.	
Pos condiciones	La serie del activo fijo debe ser única para poder tener una constancia de tal.	
Criterios de Aceptación	Ayuda a la verificación de los activos fijos y su estado original.	

Tabla 4:

Identificador RF002

Software realiza registros de activos con la información obtenida de la base de datos		Estado	En revisión
Creado por:	Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización	2014-12-16
Identificador	RF 002		
Tipo de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Búsqueda información del Activo Detalle del Activo		
Descripción	Sistema óptimo para ingresar nuevos registros de activos fijos que ingresen a la empresa.		
Datos de Salida	Información detallada de los Activos		
Resultados Esperados	Dar mayor información de los activos fijos y los estados que se encuentran.		
Origen	Gerente Administrador		
Dirigido a	Contador Custodio Secretaria		
Prioridad	4		
Requerimientos Asociados	Ninguno		
Especificación			
Precondiciones	El Activo debe estar en funcionamiento		
Pos condiciones	Ninguna		
Criterios de Aceptación	Brindará información del estado del activo fijo		

Tabla 5:

Identificador RF003

Búsqueda de activos fijos	Estado	En revisión
Creado por: Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación 2014-12-16	Fecha de actualización	2014-12-16
Identificador	RF 003	
Tipo de Requerimiento	Critico	Tipo de Requerimiento Funcional
Datos de Entrada	Código del Activo Fijo Custodio del Activo Fijo Ubicación del Activo Fijo	
Descripción	Con el código del activo fijo se sabrá toda la información del activo.	
Datos de Salida	Información precisa de los activos fijos	
Resultados Esperados	Eficacia en la búsqueda de activos fijos	
Origen	Gerente	
	Administrador	
	Contador	
Dirigido a	Custodio	
	Secretaria	
Prioridad	4	
Requerimientos Asociados	Ninguno	
Especificación		
Precondiciones	El código de cada activo fijo debe ser el correcto	
Pos condiciones	Ninguna	
Criterios de Aceptación	Búsqueda eficiente de los activos fijos.	

Tabla 6:

Identificador RF004

Dar de baja activos fijos deteriorados de la empresa	Estado	En revisión	
Creado por:	Guillermo Cauja	Actualizado por: Guillermo Cauja	
Fecha de creación	2014-12-16	Fecha de actualización: 2014-12-16	
Identificador	RF 004		
Tipo de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Código del activo fijo que va a ser dado de baja		
Descripción	Se remite una orden de salida al custodio a cargo del activo.		
Datos de Salida	Autorización para dar de baja al activo fijo		
Resultados Esperados	Evitar equivocaciones al momento de dar de baja a un activo fijo de la empresa		
Origen	Gerente		
	Administrador		
	Contador		
Dirigido a	Custodio		
	Secretaria		
Prioridad	5		
Requerimientos Asociados	Ninguno		
Especificación			
Precondiciones	Para dar de baja a un activo fijo se debe verificar el código del activo fijo registrado en la base de datos.		
Pos condiciones	Todo activo fijo dado de baja debe quedar registrado en la base de datos.		
Criterios de Aceptación	Brindará información de los estados que se encontraba el activo y el custodio que estuvo a cargo.		

Tabla 7:

Identificador RF005

Seguridades para usuarios		Estado	En revisión
Creado por:	Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización	2014-12-16
Identificador	RF 005		
Tipo de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Usuario Contraseña		
Descripción	Ingresos de usuarios y contraseñas para mostrar información únicamente de cada usuario.		
Datos de Salida	Se mostrara información únicamente por cada usuario.		
Resultados Esperados	Se mostrara información validada para el perfil de cada usuario.		
Origen	Gerente		
	Administrador		
	Contador		
Dirigido a	Custodio		
	Secretaria		
Prioridad	5		
Requerimientos Asociados	Ninguno		
Especificación			
Precondiciones	Los usuarios del sistema podrán ingresar con un nombre de usuario y contraseña que estará almacenado en la base de datos del sistema.		
Pos condiciones	Los nombres de usuarios y contraseñas son únicos para cada uno de los usuarios.		
Criterios de Aceptación	Establece información a mostrar a cada uno de los usuarios según su competencia.		

Tabla 8:

Identificador RF006

Depreciación de los Activos Fijos	Estado	En revisión
Creado por: Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización
		2014-12-16
Identificador	RF 006	
Tipo de Requerimiento	Critico	Tipo de Requerimiento Funcional
Datos de Entrada	Código del Activo	
	Descripción estado del Activo	
Descripción	Se comprobara el estado del activo fijo para dar solución a tal o dar de baja al activo.	
Datos de Salida	Se verificara el estado del activo fijo para ver su depreciación.	
Resultados Esperados	Estar al tanto de los mantenimientos que se le dieron a los activos y saber cuánto tiempo tendrá de vida útil.	
Origen	Gerente	
	Administrador	
Dirigido a	Contador	
	Custodio	
	Secretaria	
Prioridad	5	
Requerimientos Asociados	Ninguno	
Especificación		
Precondiciones	La depreciación del activo es acorde a la vida útil de tal.	
Pos condiciones	Ninguna	
Criterios de Aceptación	La empresa no tendrá mucha perdida por activos depreciados.	

Tabla 9:

Identificador RF007

Almacenamiento y registro de movimientos de los Activo Fijos en la empresa		Estado	En revisión
Creado por:	Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización	2014-12-16

Identificador	RF 007
Tipo de Requerimiento	Critico
Tipo de Requerimiento	Funcional
Datos de Entrada	Ingreso de los activos fijos Asignación de activos a custodios Estado del Activo Fijo
Descripción	Información de los activos de la empresa
Datos de Salida	Ubicación de los activos y custodios a cargo
Resultados Esperados	Mejorar la información a mostrar de los estados, custodios a cargo y ubicación de los activos de la empresa.
Origen	Gerente
Dirigido a	Administrador Contador Custodio Secretaria
Prioridad	5
Requerimientos Asociados	Ninguno
Especificación	
Precondiciones	Los activos nuevos deber ser almacenadas en una base de datos donde se encuentre la información antigua.
Pos condiciones	Ninguna
Criterios de Aceptación	Información de los movimientos de los activos en la empresa

Tabla 10:

Identificador NRF001

Aplicación amigable y compatible con diferentes navegadores en internet		Estado	En revisión
Creado por:	Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización	2014-12-16

Identificador	NRF 001
Tipo de Requerimiento	Critico
Tipo de Requerimiento	Funcional
Datos de Entrada	Internet Dirección web
Descripción	Aplicación compatible con distintos exploradores de Internet.
Datos de Salida	Visualización del sistema vía internet
Resultados Esperados	El sistema funcionara en cualquier tipo de navegador
Origen	Gerente Administrador
Dirigido a	Contador Custodio Secretaria
Prioridad	4
Requerimientos Asociados	Ninguno
Especificación	
Precondiciones	La aplicación será desarrollada en lenguajes de programación compatibles con los navegadores.
Pos condiciones	Sistema compatible con todos los navegadores
Criterios de Aceptación	No tenga dificultad el usuario al usar la aplicación.

Tabla 11:

Identificador NRF002

Visualización de datos en diferentes ventanas	Estado	En revisión
Creado por:	Guillermo Cauja	Actualizado por: Guillermo Cauja
Fecha de creación	2014-12-16	Fecha de actualización 2014-12-16
Identificador	NRF 002	
Tipo de Requerimiento	Critico	Tipo de Requerimiento Funcional
Datos de Entrada	Datos de los activos fijos.	
Descripción	Aplicación desarrollada para visualización sencilla.	
Datos de Salida	Datos que el usuario lo solicite	
Resultados Esperados	Mejorar la visualización de la aplicación.	
Origen	Gerente	
	Administrador	
Dirigido a	Contador	
	Custodio	
	Secretaria	
Prioridad	5	
Requerimientos Asociados	Ninguno	
Especificación		
Precondiciones	Tener instaladas las herramientas	
Pos condiciones	Ninguna	
Criterios de Aceptación	Mayor facilidad al imprimir etiquetas con la información.	

Tabla 12:

Identificador NRF003

Inicio sencillo para el usuario.	Estado	En revisión
Creado por: Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación 2014-12-16	Fecha de actualización	2014-12-16
Identificador	NRF 003	
Tipo de Requerimiento	Critico	Tipo de Requerimiento Funcional
Datos de Entrada	Ingreso de los datos de los activos Ingreso de información del usuario	
Descripción	La consulta de información se sencilla de entender.	
Datos de Salida	Acceso al sistema Información de los Activos	
Resultados Esperados	Los usuarios podrán hacer todo lo que el sistema lo permita sin ninguna restricción.	
Origen	Gerente	
	Administrador	
Dirigido a	Contador	
	Custodio	
	Secretaria	
Prioridad	4	
Requerimientos Asociados	Ninguno	
Especificación		
Precondiciones	El internet es fundamental para su funcionamiento.	
Pos condiciones	Ninguna	
Criterios de Aceptación	Sistema seguro para navegar.	

Tabla 13:

Identificador NRF004

Base de datos sin datos erróneos.	Estado	En revisión
Creado por: Guillermo Cauja	Actualizado por:	Guillermo Cauja
Fecha de creación 2014-12-16	Fecha de actualización	2014-12-16
Identificador	NRF 004	
Tipo de Requerimiento	Critico	Tipo de Requerimiento Funcional
Datos de Entrada	Ingreso de datos al sistema	
Descripción	Que no exista información duplicada en la base de datos.	
Datos de Salida	Acceso al sistema Información de los Activos	
Resultados Esperados	Cada activo tiene información única con códigos.	
Origen	Gerente	
	Administrador	
	Contador	
Dirigido a	Custodio	
	Secretaria	
Prioridad	4	
Requerimientos Asociados	Ninguno	
Especificación		
Precondiciones	El sistemas estará correctamente validado para evitar la duplicidad de información	
Pos condiciones	El administrador solo deberá ingresa información real de los Activos	
Criterios de Aceptación	El sistemas no permitirá duplicidad de activos	

2.02 Mapeo de Involucrados

Permite identificar a las personas involucradas y a las organizaciones interesadas en el proyecto y a las personas que tienen poder sobre la utilización de la aplicación.



Fig 1. Mapeo de Involucrados: Muestra los elementos involucrados en el desarrollo del sistema.

2.03 Matriz de Involucrados

Tabla 14:

Matriz de Involucrados

Actores Involucrados	Interés sobre el problema central	Problemas Percibidos	Recursos, Mandatos y Capacidades	Interés sobre el proyecto	Conflictos Potenciales
Dueño de la empresa	Mejorar la calidad de atención a los clientes basándose en herramientas tecnológicas que faciliten la labor diaria.	Control inadecuado de los activos de la empresa	Recursos Económicos y Humanos	Efectivizar la labor diaria de los empleados teniendo en cuenta los recursos que utilizan se encuentran funcionando en la empresa.	Falta de una aplicación tecnológica que controle los activos.
Gerente General	Prevenir la pérdida de activos y dar mantenimiento a los activos dañados.	Grandes pérdidas económicas por pérdidas de activos en la empresa.	Recursos Económicos y Humanos	Mejorar el control de los activos de la empresa.	Falta de una aplicación tecnológica que controle los activos.
Bodega	Facilidad de manejo de la información detallada de los Activos	Desconocer el tiempo de vida útil de un activo fijo.	Recursos Económicos y Humanos	Facilidad de consultada de datos en el sistema	Errores de codificación de los activos fijos
Contador	Beneficiar al control de datos de los activos y depreciaciones de la empresa.	Mejorar la fluidez en el proceso de información de estados de los activos.	Recursos Humanos	El sistema beneficiara a los procesos de consulta y registro de los activos.	Mal registro de los datos de los Activos.
Área informática	Conocer los activos a que custodios y en qué estado le fueron entregado a cada uno.	Pérdida de los activos sin saber que custodio estaba a cargo de tal.	Recursos Humanos	Tener el control de todos los activos fijos de la empresa.	No poseer una herramienta virtual que almacene de forma segura la información de la empresa.
Desarrollador	Tener el control de todos los activos de la empresa y sus características.	Desconocimiento de la depreciación que puede tener un activo.	Recursos Humanos	Facilidad de información y búsqueda de los Activos	Perdida del registro de los Activos.

Capítulo III: Problemas y Objetivos

3.01 Árbol de Problemas

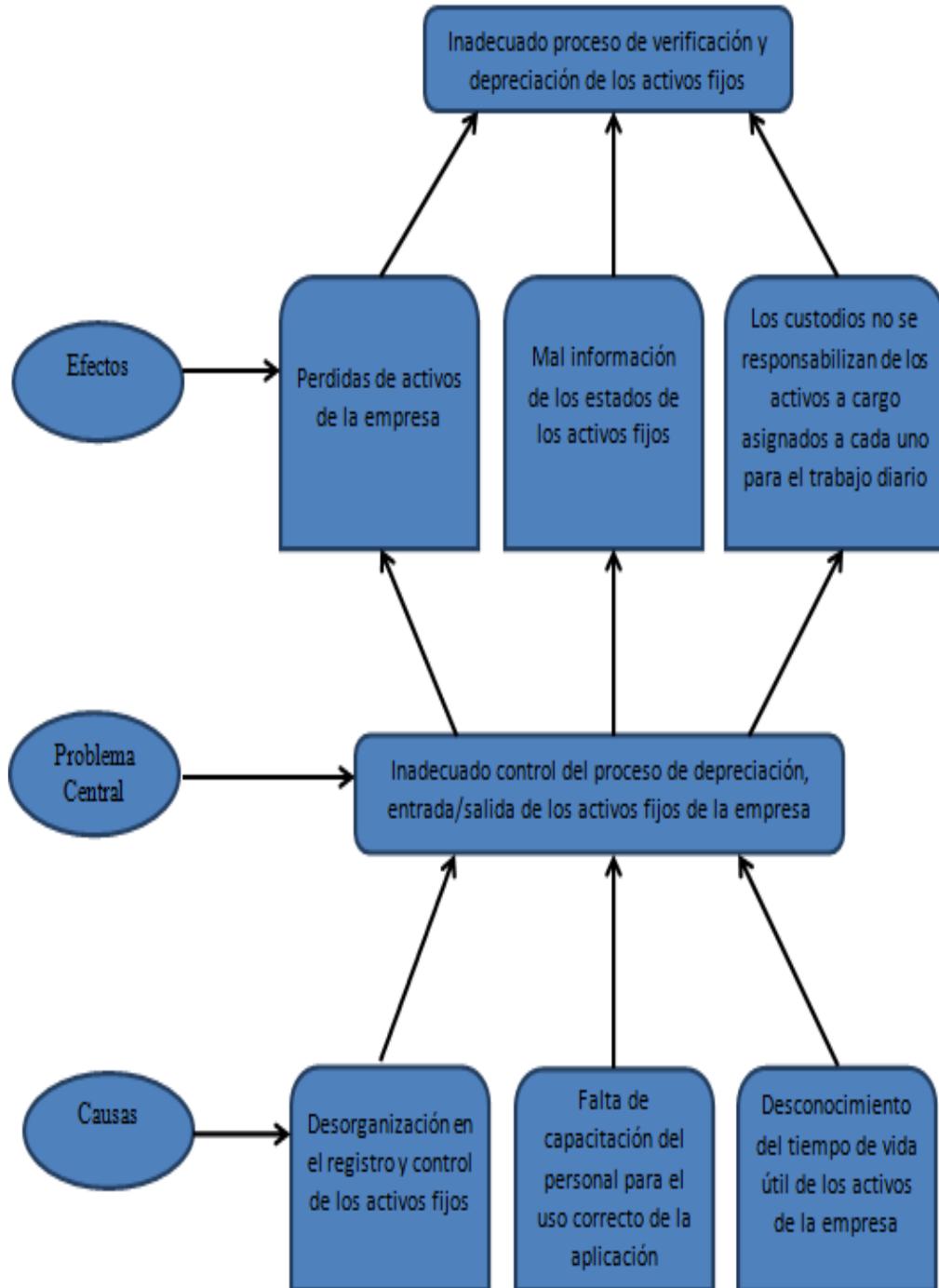


Fig 2. Árbol de problemas: Verifica he identifica los problemas que la aplicación pueda ayudar resolver.

3.02 Árbol de Objetivos

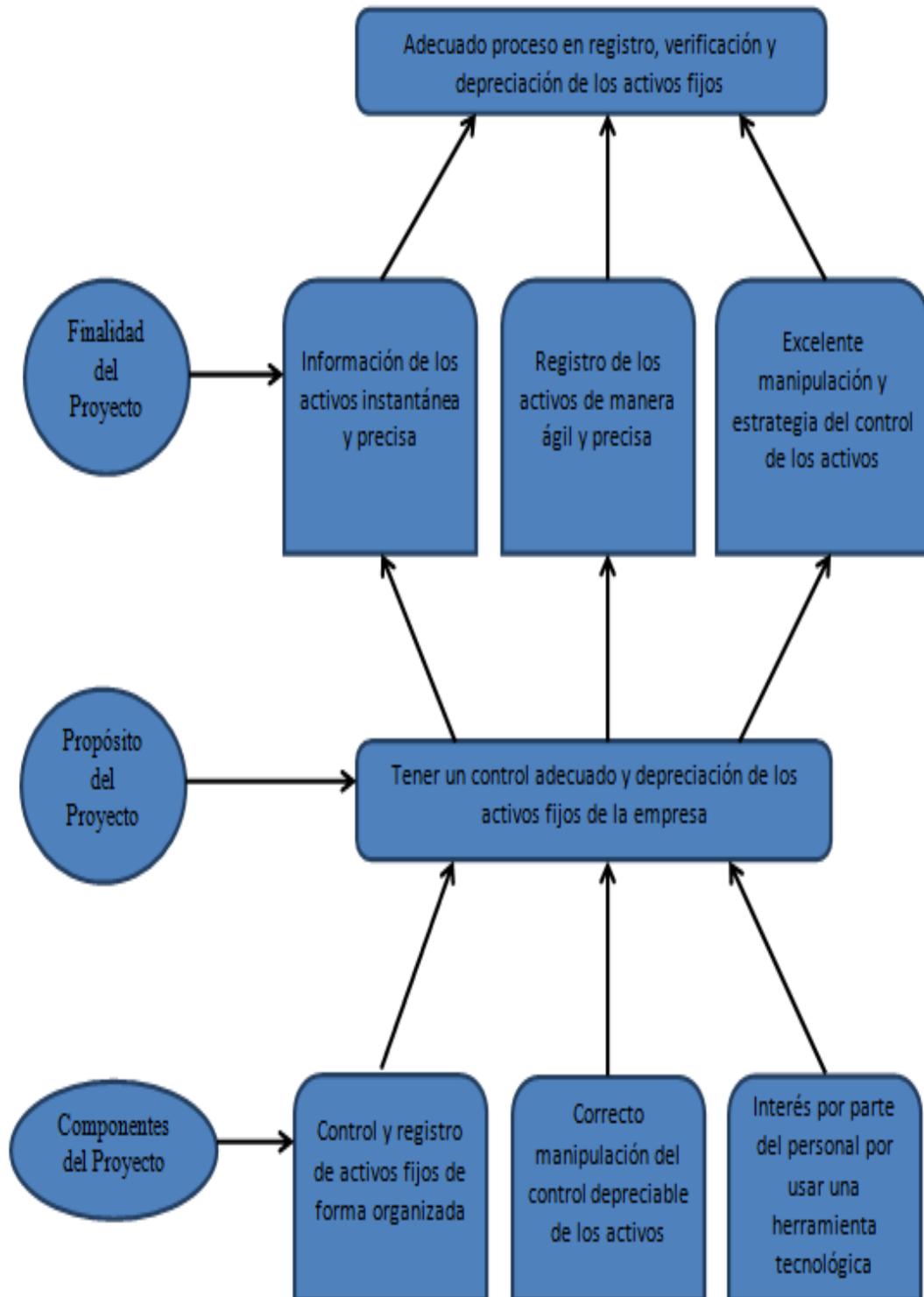


Fig 3. Árbol de objetivos: Verifica he identifica los beneficios que el proyecto pueda proporcionar a los usuarios.

3.03 Casos de Uso

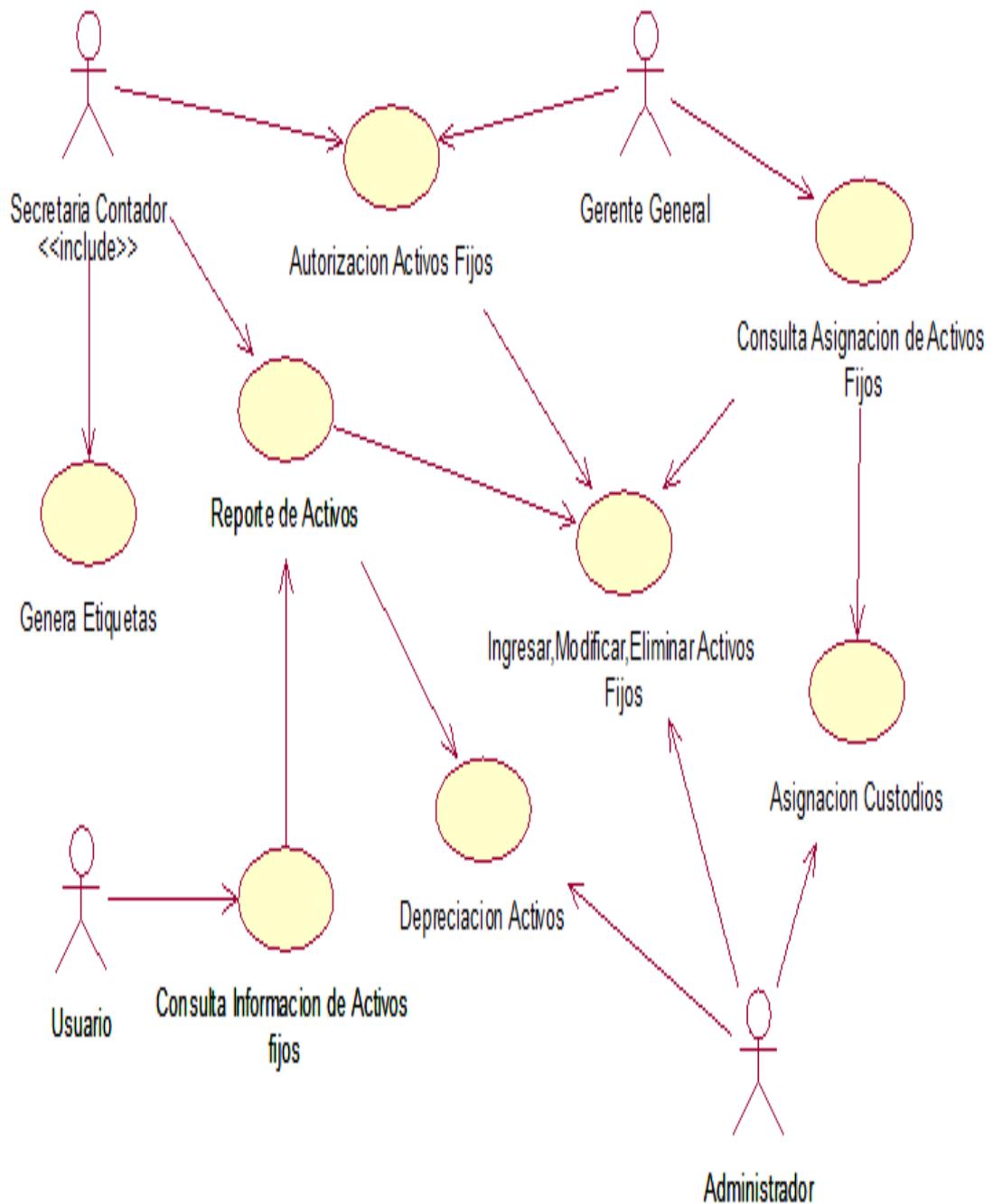


Fig 4. Caso de Uso General: Representa los procesos que realizan los Actores en la aplicación para el correcto funcionamiento de tal.

3.04 Casos de Uso de Realización

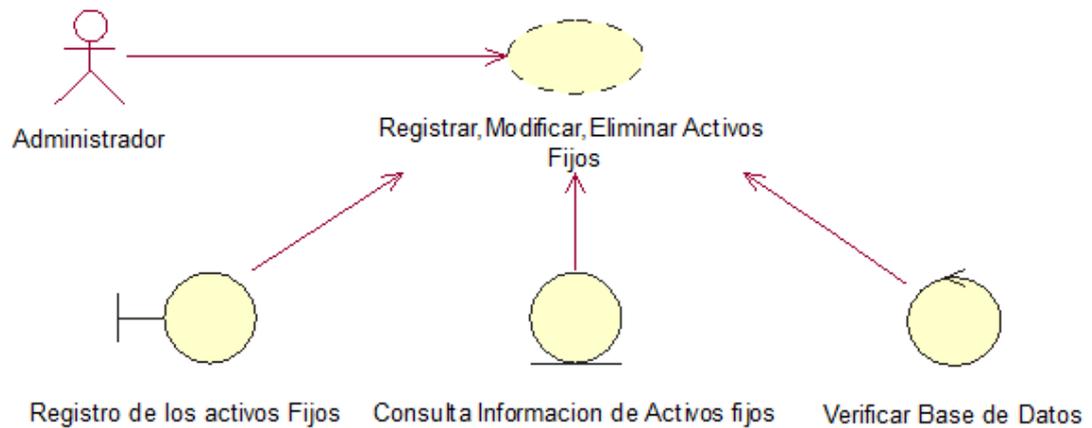


Fig 5.Caso de Uso Registrar, Modificar: Este caso de uso representa los registros, modificaciones y eliminación de los activos fijos.

Tabla 15:

Caso Realización Registrar, Modificar Activos Fijos.

Nombre	Registrar, Modifica, Eliminar Activos Fijos
Identificador	C.U.1
Responsabilidades	Registro de Activos
Tipo	Sistema
Referencia Casos de Uso	Consulta de Información de Activos fijos
Referencia Requisitos	Ingreso de información de los Activos fijos
PRECONDICIONES	
De instancia:	
1. El administrador ingresa con su usuario y contraseña.	
2. Ingresar la información del Activo Fijo.	
3. Comparar si la información el código del activo fijo no se repita.	
4. Registrar el Activo fijo.	
De relación:	
Ninguno	
POSCONDICION	
De instancia:	
1. El activo se verificara en la base de datos.	
2. El activo será asignado a un custodio.	
De relación:	
1. El activo será relacionado con la asignación a un custodio	
SALIDAS DE PANTALLA	
1. Registro guardado correctamente en el sistema.	
2. Editar, Eliminar activos fijos.	

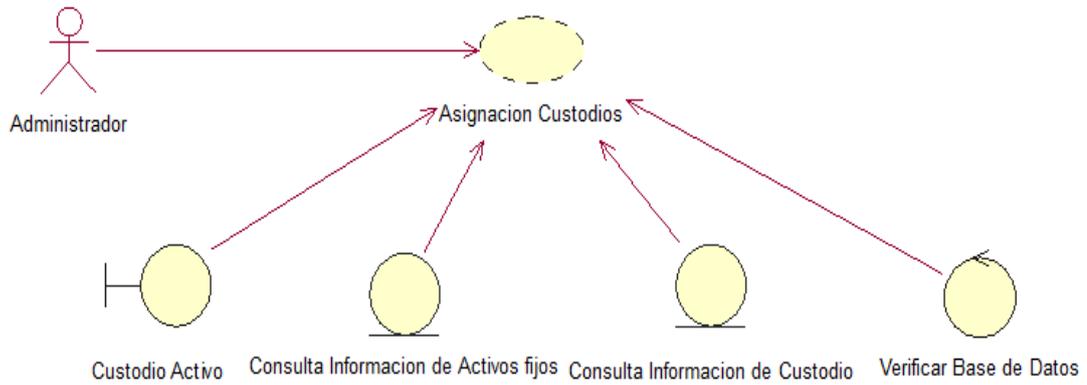


Fig 6. Caso de Uso Asignación Custodio; Este caso de uso representa la asignación de los activos fijos a los custodios a cargo.

Tabla 16:

Caso Realización Asignación Custodios

Nombre	Asignación Custodios
Identificador	C.U.2
Responsabilidades	Asignar Activos
Tipo	Sistema
Referencia Casos de Uso	Organización de los Activos
Referencia Requisitos	El activo debe estar registrado en el sistema
PRECONDICIONES	
De instancia:	
1. Verificar si el activo se encuentra correctamente registrado.	
De relación:	
1. La información registrada debe estar correcta.	
POSCONDICION	
De instancia:	
1. Guardar la asignación del Activo	
De relación:	
Ninguna	
SALIDAS DE PANTALLA	
1. El activo será asignado al custodio que usted ha seleccionado.	
2. Activo asignado correctamente.	

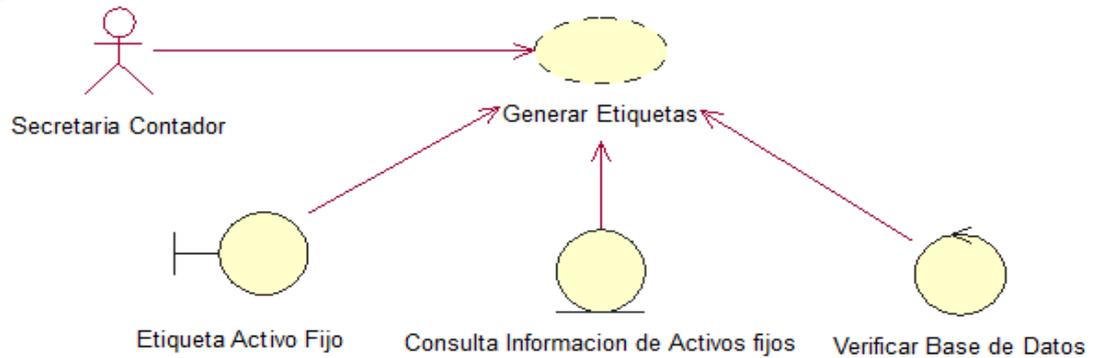


Fig 7. Caso de uso Generar Etiqueta: Representa los procesos que el sistema toma para generar e imprimir una etiqueta.

Tabla 17:

Caso Realización Generar Etiqueta

Nombre	Generar Etiqueta
Identificador	C.U.3
Responsabilidades	Etiquetar Activos
Tipo	Sistema
Referencia Casos de Uso	Organización de los Activos
Referencia Requisitos	Las etiquetas tienen un código único para cada activo
PRECONDICIONES	
De instancia:	
<ol style="list-style-type: none"> 1. La secretaria ingresara al sistema con su cuenta y contraseña. 2. Se llenara los campos necesarios de un activo. 3. El código se generara automáticamente 4. Se guarda la información. 5. Se imprimirá las etiquetas. 	
De relación:	
Ninguno	
POSCONDICION	
De instancia:	
<ol style="list-style-type: none"> 1. La etiqueta se imprime 2. Poner la etiqueta en cada activo fijo 	
De relación:	
<ol style="list-style-type: none"> 1. La etiqueta posee un código único para cada activo fijo 	
SALIDAS DE PANTALLA	
<ol style="list-style-type: none"> 1. Código generado correctamente 2. Imprimir etiqueta 	

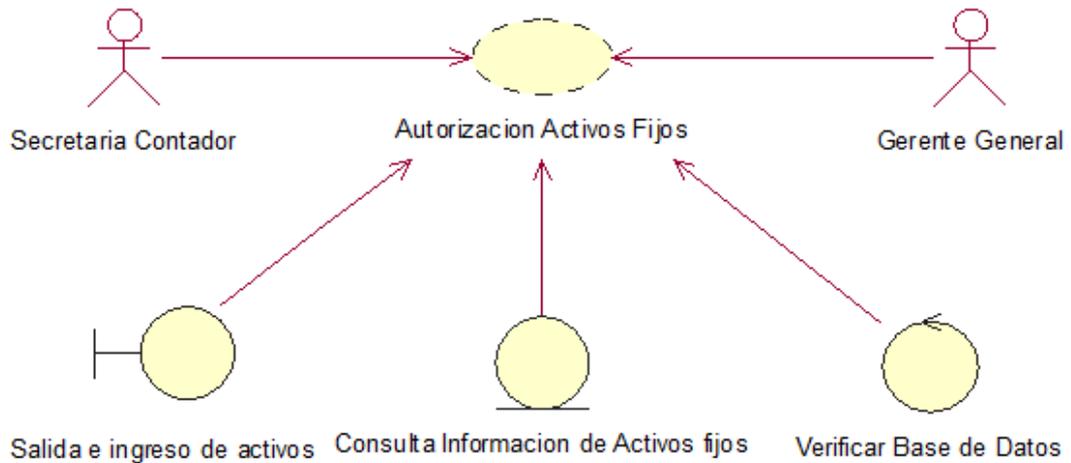


Fig 8. Caso de Uso Autorización Activos Fijos: Son los procesos que realiza el sistema para autorizar la salida de activos fijos deteriorados.

Tabla 18:

Caso Realización Autorización Activos Fijos

Nombre	Autorización de Salida e ingreso de activos fijos
Identificador	C.U.4
Responsabilidades	Autorización salida e ingreso de activos fijos en la empresa
Tipo	Sistema
Referencia Casos de Uso	Seguridad de los Activos
Referencia Requisitos	El activo debe estar registrado en la base de datos del sistema
PRECONDICIONES	
De instancia:	
1. Se debe verificar si el activo se encuentra registrado en la base de datos de la empresa	
2. Ver el estado que se encuentra el Activo Fijo	
De relación:	
1. El estado del Activo fijo debe estar con la información correcta.	
POSCONDICION	
De instancia:	
1. Se guarda e imprime la autorización del Activo Fijo.	
De relación:	
Ninguna	
SALIDAS DE PANTALLA	
1. Ingresar código del activo fijo	
2. Desea generar la autorización.	
3. La autorización ha sido guardada en el sistema	
4. Impresión de la autorización correcta	

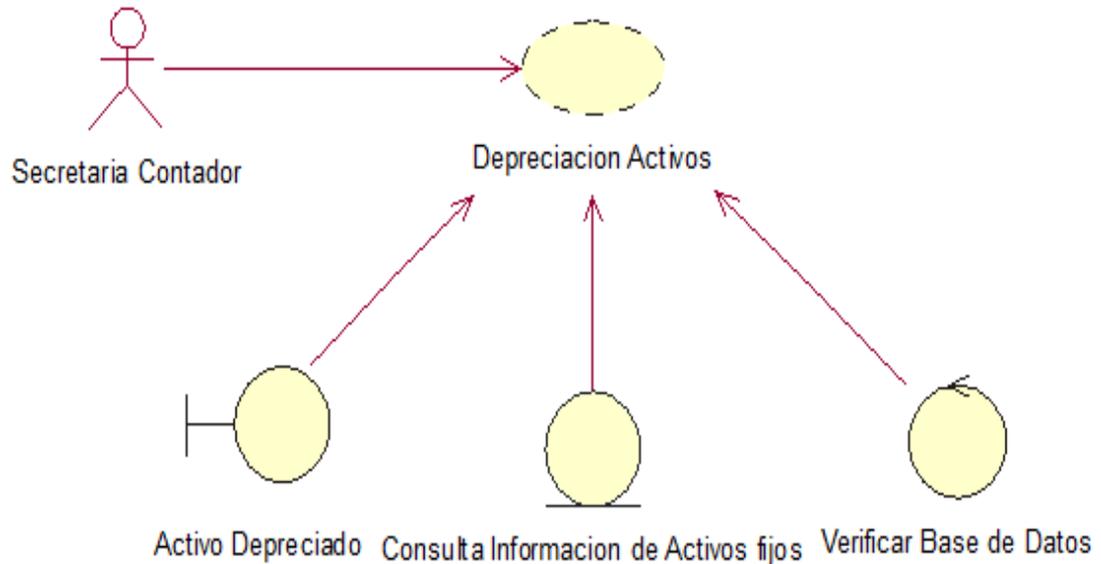


Fig 9. Caso de Uso Depreciación de Activos: Son los procesos que realiza el sistema para depreciar los activos.

Tabla 19:

Caso Realización Depreciación de Activos

Nombre	Depreciación de Activos
Identificador	C.U.5
Responsabilidades	Verificar la vida útil de los activos
Tipo	Sistema
Referencia Casos de Uso	Control de Activos
Referencia Requisitos	El activo debe estar registrado en el sistema
PRECONDICIONES	
De instancia:	
1. Constatar que el activo se encuentre registrado en la base de datos de la empresa	
2. Registrar vida útil del activo	
3. Consultar depreciación que tiene el activo por año	
De relación:	
Ninguna	
POSCONDICION	
De instancia:	
1.Registrar los estados del activo	
2. Llevar un registro de los activos fijos dados de baja	
De relación:	
1. Verificar el estado que se encuentra el activo	
SALIDAS DE PANTALLA	
1.El activo fijo será dado de baja	
2. Éxito el activo fijo se ha dado de baja	

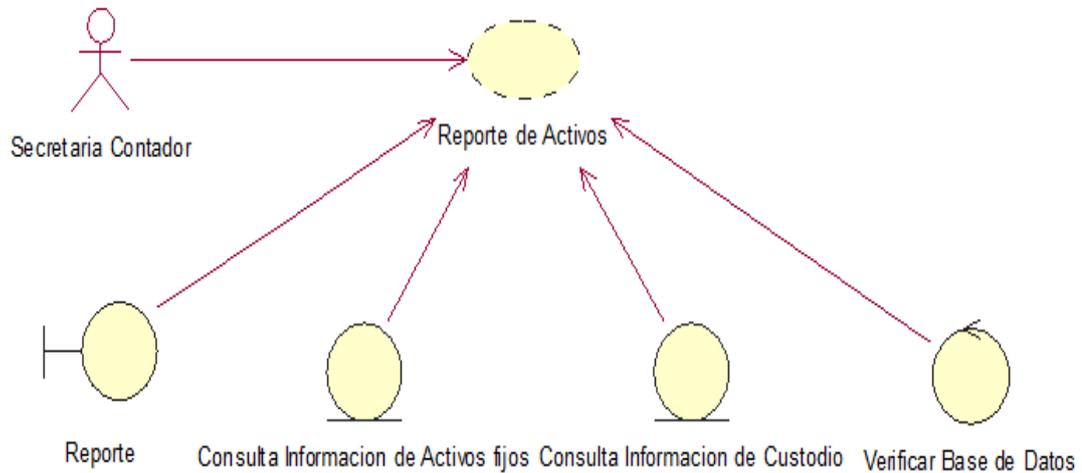


Fig 10. Caso de Uso Reporte de Activos: Son los procesos que realiza el sistema para generar reportes.

Tabla 20:

Caso Realización Reporte Activos

Nombre	Reporte activos
Identificador	C.U.6
Responsabilidades	Informar al usuario la información que solicite
Tipo	Sistema
Referencia Casos de Uso	Información de los activos
Referencia Requisitos	Los activos deben estar registrados en la base de datos
PRECONDICIONES	
De instancia:	
1. El activo debe estar registrado en la base de datos	
2. Consultar la información del activo por el código del activo	
De relación:	
Ninguna	
POSCONDICION	
De instancia:	
1.Los activos deben tener información válida para el reporte de cada uno de los activos	
2. Imprimir reporte de los activos.	
De relación:	
1. Verificar en qué estado se encuentra el activo para realizar alguna autorización	
SALIDAS DE PANTALLA	
1.Desea verificar el activo fijo	
2. Desea imprimir el reporte	

3.05 Diagramas de Secuencia del Sistema

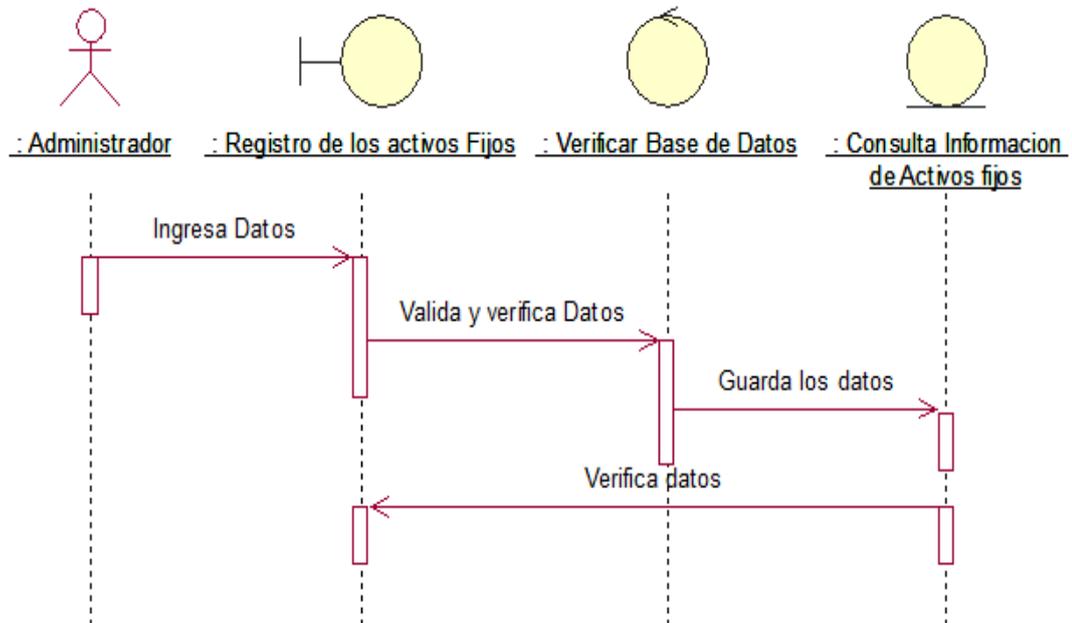


Fig 11. Diagrama de Secuencia Registro de Activos Fijos: Muestra el proceso que el Administrador debe realizar para registrar un activo.

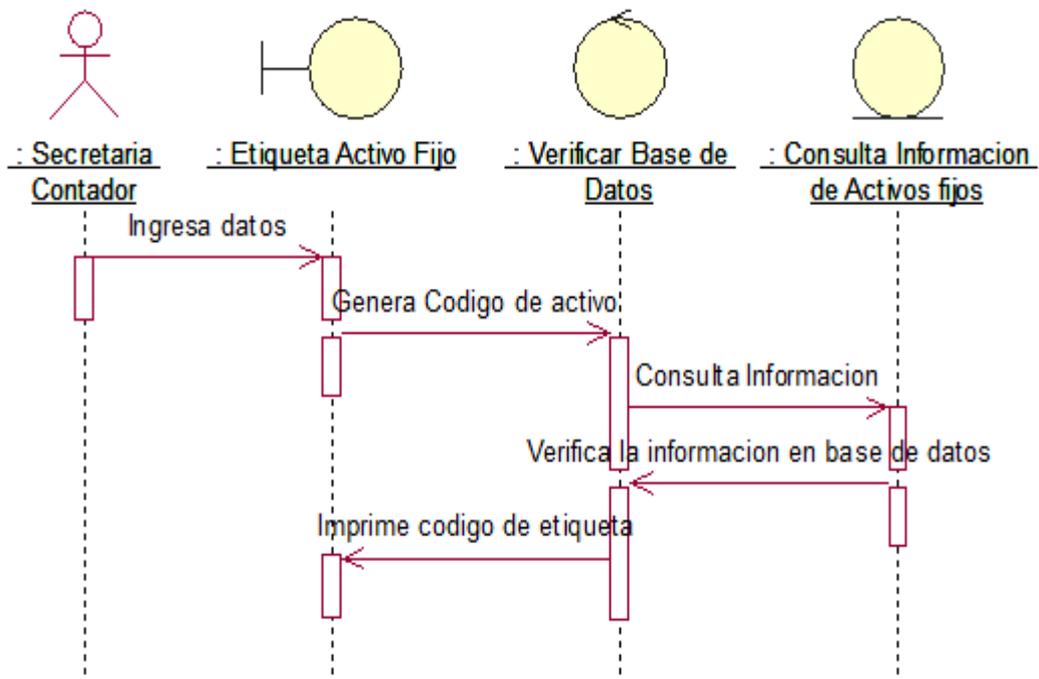


Fig 12. Diagrama de Secuencia Etiqueta de Activo Fijo: Muestra el proceso que la secretaria debe realizar para generar una etiqueta de un activo.

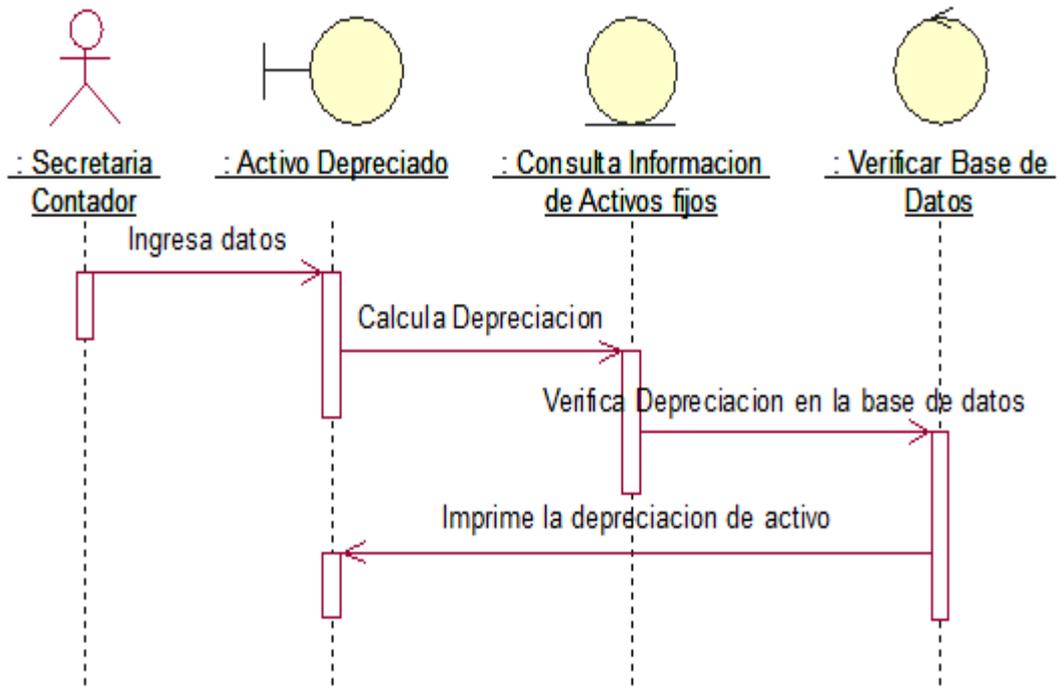


Fig 13. Diagrama de Secuencia Activo Depreciado: Muestra el proceso que la secretaria y contador debe realizar para generar la depreciación de los activos.

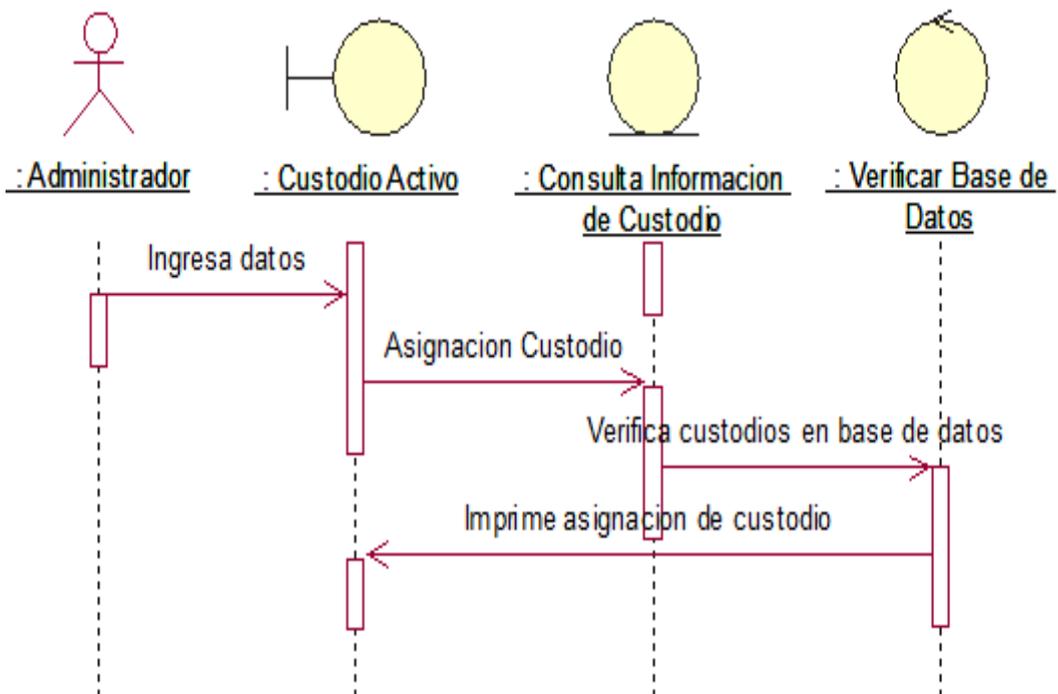


Fig 14. Diagrama de Secuencia Asignación de Custodio de Activo: Muestra el proceso que el administrador debe realizar para asignar un custodio.

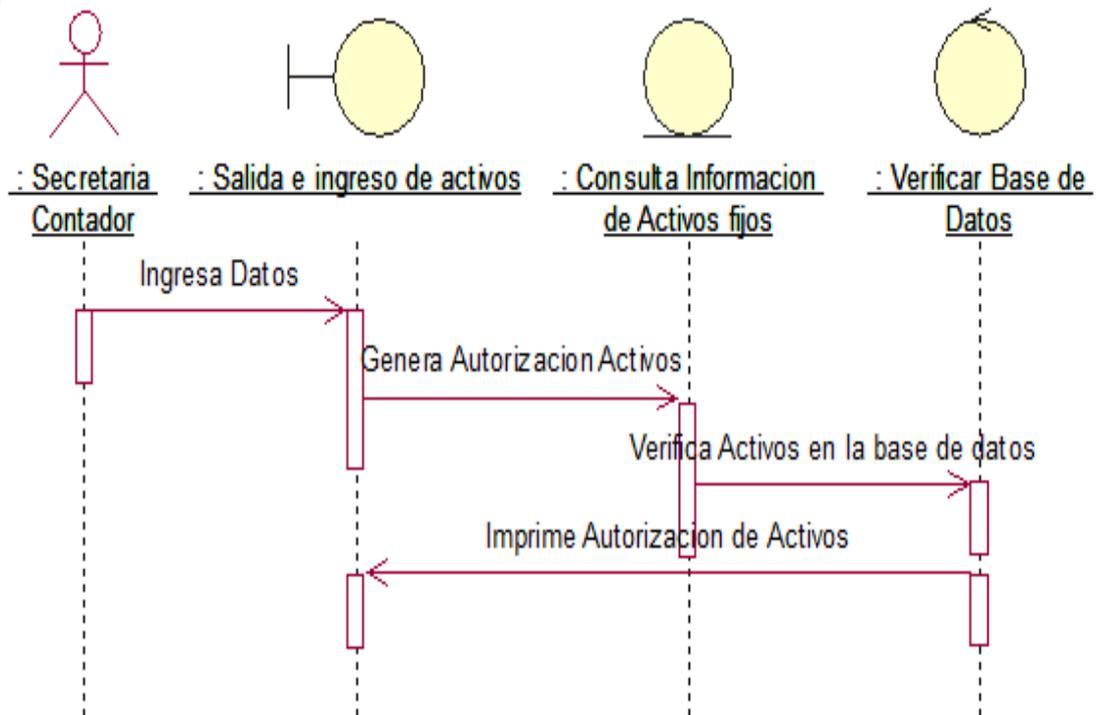


Fig 15. Diagrama de Secuencia Salida e ingreso de Activo: Muestra el proceso que el contador debe realizar para asignar la salida de activos fijos de la empresa.

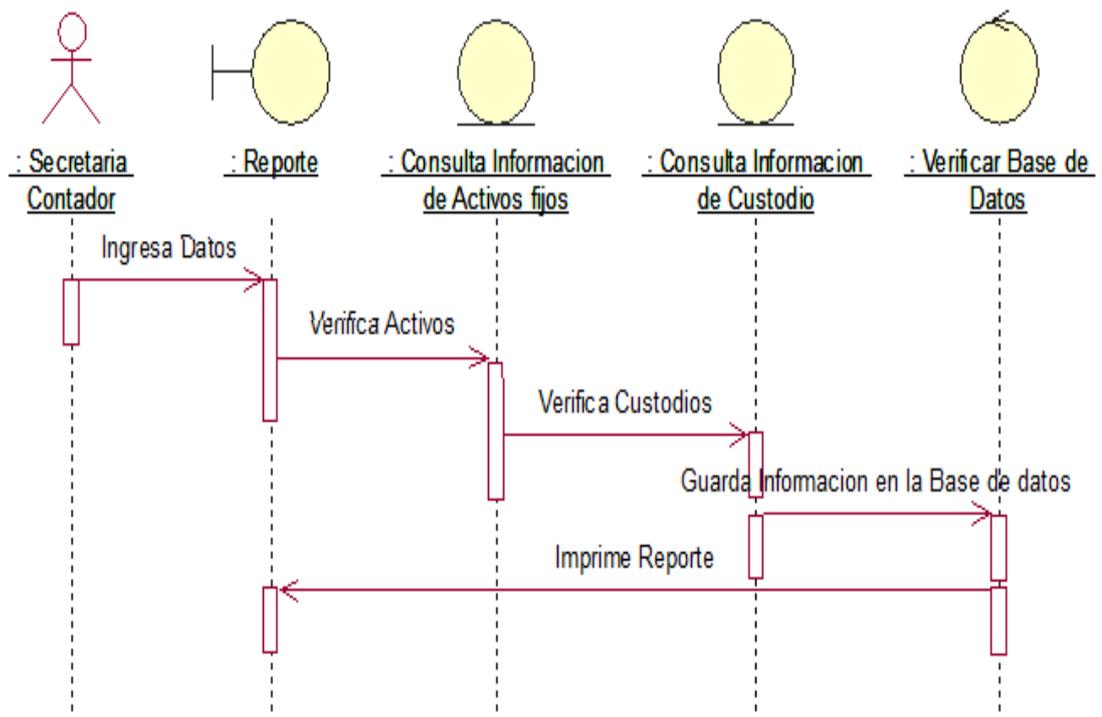


Fig 16. Diagrama de Secuencia Reporte de Activo: Muestra el proceso que la secretaria debe realizar para generar un reporte.

3.06. Especificación de Casos de Uso

Tabla 21:

Especificación Casos Uso: Registro, Modificación de Activo.

Caso de uso	Registro de activos Fijos
Identificador	C.U.001
Curso típico de Eventos	
Usuario	Sistema
El administrador del sistema debe llenar los registros de los activos fijos con información de cada uno de ellos	<p>Las operaciones de casos de uso se realizan siempre y cuando el usuario se encuentre operando en el sistema.</p> <p>Se ingresa al menú de Activo</p> <p>Se llena información que el sistema lo requiera para el almacenamiento en la base de datos.</p> <p>Se registra vida útil del activo</p>
Cursos Alternativos	
1. El usuario puede Ingresar, Modificar y Actualizar.	

Nota: Significado de las siglas CU; CU=Caso de Uso.

Tabla 22:

Especificación Casos Uso: Etiqueta de Activos Fijos

Caso de uso	Etiqueta de Activos Fijos
Identificador	C.U.002
Curso típico de Eventos	
Usuario	Sistema
El administrador debe ingresar al sistema y generar el código del Activo Fijo.	<p>Las operaciones de casos de uso se realizan siempre y cuando el usuario se encuentre operando en el sistema.</p> <p>Se ingresa al menú de Etiquetas de activos</p> <p>Se llena información que el sistema lo requiera para generar etiqueta</p> <p>Se guardara información en la base de datos</p> <p>Se imprimirá etiqueta del activo fijo.</p>
Cursos Alternativos	
1. El usuario puede Ingresar, Modificar y Actualizar.	

Nota: Significado de las siglas CU; CU=Caso de Uso.

Tabla 23:

Especificación Casos Uso: Activo Depreciado

Caso de uso	Activo Depreciado
Identificador	C.U.003
Curso típico de Eventos	
Usuario	Sistema
El administrador debe ingresar al sistema y generar la depreciación del activo.	<p>Las operaciones de casos de uso se realizan siempre y cuando el usuario se encuentre operando en el sistema.</p> <p>Se ingresa al menú de Depreciaciones de Activo.</p> <p>Se llena la información que el sistema lo requiera para generar depreciación.</p> <p>Se guardara información en la base de datos</p> <p>Se imprimirá depreciación del activo Fijo.</p>
Cursos Alternativos	
1. El usuario puede Ingresar, Modificar y Actualizar.	

Nota: Significado de las siglas CU; CU=Caso de Uso.

Tabla 24:

Especificación Casos Uso: Asignación Custodio

Caso de uso	Asignación Custodio
Identificador	C.U.004
Curso típico de Eventos	
Usuario	Sistema
El administrador debe ingresar al sistema y asignar un custodio por activo fijo.	<p>Las operaciones de casos de uso se realizan siempre y cuando el usuario se encuentre operando en el sistema.</p> <p>Se ingresa al menú de Asignación de custodio</p> <p>Se llena la información que el sistema lo requiera para asignar un custodio.</p> <p>Se imprimirá asignación del custodio.</p> <p>Se guardara información en la base de datos</p>
Cursos Alternativos	
1. El usuario puede Ingresar, Modificar y Actualizar.	

Nota: Significado de las siglas CU; CU=Caso de Uso.

Tabla 25:

Especificación Casos Uso: Salida e ingresos de activos

Caso de uso	Salida e ingreso de activos
Identificador	C.U.005
Curso típico de Eventos	
Usuario	Sistema
El administrador debe ingresar al sistema y generar autorización de activos fijos.	<p>Las operaciones de casos de uso se realizan siempre y cuando el usuario se encuentre operando en el sistema.</p> <p>Se ingresa al menú de Autorización de activos fijos.</p> <p>Se llena información que el sistema lo requiera para generar la autorización del activo.</p> <p>Se guardará información en la base de datos</p> <p>Se imprimirá autorización de entrada o salida del activo fijo de la empresa.</p>
Cursos Alternativos	
1. El usuario puede Ingresar, Modificar y Actualizar.	

Nota: Significado de las siglas CU; CU=Caso de Uso.

Tabla 26:

Especificación Casos Uso: Reporte de Activos Fijos

Caso de uso	Reporte de Activos Fijos
Identificador	C.U.006
Curso típico de Eventos	
Usuario	Sistema
El administrador debe ingresar al sistema y generar reportes de los activos fijos	<p>Las operaciones de casos de uso se realizan siempre y cuando el usuario se encuentre operando en el sistema.</p> <p>Se ingresa al menú Generar reportes</p> <p>Se llena información que el sistema lo requiera para generar el reporte</p> <p>Se guardara información en la base de datos</p> <p>Se imprimirá los reportes si el usuario lo desea.</p>
Cursos Alternativos	
1. El usuario puede Ingresar, Modificar y Actualizar.	

Nota: Significado de las siglas CU; CU=Caso de Uso.

Capítulo IV: Análisis de Alternativas

4.01 Matriz de Análisis de Alternativas

Tabla 27:

Matriz de Análisis de Alternativas

	Factibilidad de Lograrse	Impacto en Género	Impacto Ambiental	Relevancia	Sostenibilidad	Total
Control y registro de activos fijos	Brindará mayor efectividad que el valor que costara	Ninguno el género es indistinto	Ayuda a mejorar y proteger el medio ambiente.	Cumple las expectativas de los usuarios.	Los usuarios interactúan con la aplicación	Baja
Registro de activo con código de barras de forma ágil y precisa	Es la mejor opción para el contador y la secretaria	Ninguno el género es indistinto	Mejora el entorno social	Beneficiará al servicio de bodega y auditorias	Los usuarios pueden utilizar el sistema	Alta
Personal capacitado para emplear y manejar la aplicación	Apoyo total por parte de las autoridades de la empresa	Ninguno el género es indistinto	Ayuda a la educación por el medio ambiente	Es una prioridad para la gerencia	Se puede conseguir apoyo a futuro	Alta
Infraestructura tecnológica utilizada eficientemente	Infraestructura útil y adecuada	Ninguno el género es indistinto	El uso de recursos es muy baja	Se utilizará la infraestructura nueva	Fortalece la infraestructura eficientemente.	Alta
	Puntos 20	Puntos 0	Puntos 16	Puntos 20	Puntos 18	Alta 74
Disminución en tiempo de entregas						

En este análisis se toma en cuenta los objetivos a cumplir para culminar con éxito el proyecto, ayuda a la toma de decisiones correctas en el momento de identificar las posibles alternativas que se obtendrá.

4.02 Matriz de Impacto de Objetivos

Tabla 28:

Matriz de Impacto de Objetivos

Objetivos	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categorías
Control y registro de activos fijos	5	3	2	2	0	12	Media
Infraestructura tecnológica utilizada eficientemente	4	4	4	4	0	16	Alta
Personal capacitado para emplear y manejar la aplicación	5	5	4	4	0	18	Alta
Total	14	12	10	10	0	46	

4.03. Estándares para el Diseño de Clases

Tabla 29:

Estándares para Diseño de clases

Objeto	Gráfico	Concepto
Clase		<p>Es la unidad donde se almacena toda la información de un Objeto.</p> <p>Las clases implementan una o más interfaces.</p>
Atributos		<p>Los registros que contiene una clase por ejemplo ID_Activo, etc.</p>
Operación		<p>Son las operaciones que realiza la clase</p>
Relación de uno a muchos	<p>uno o muchos: 1..* (1..n) 0 o muchos: 0..* (0..n)</p>	<p>Sirve para relacionar diferentes clases de una clase a muchas o de muchas clases a una.</p>
Dependencia o Instanciación (uso)		<p>Son las herencias que contienen las clases.</p>

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de

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

Fuente: web

<http://www.monografias.com/trabajos34/ingenieria-software/ingenieria-software.shtml>

4.04. Diagrama de clases

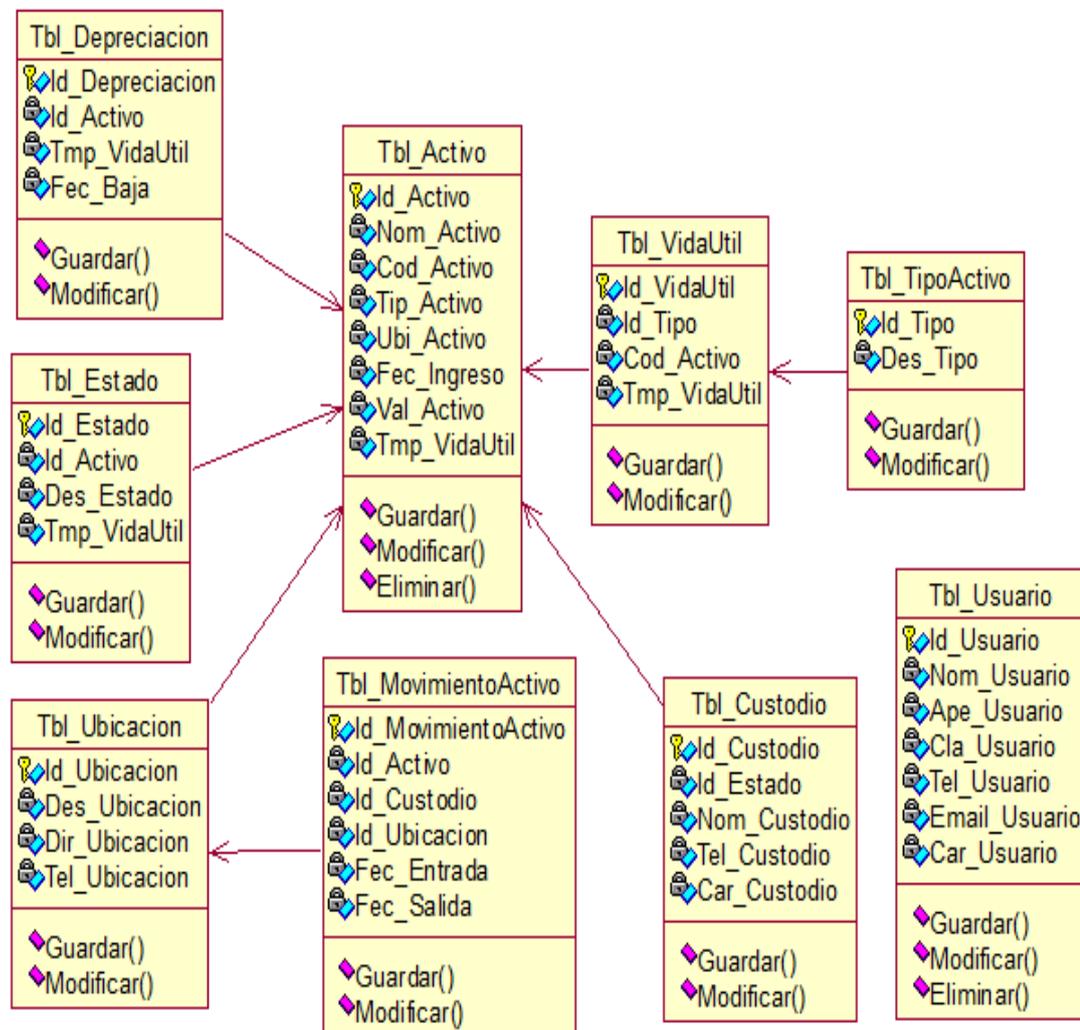


Fig 17. Diagrama de Clases: Esta Figura muestra el modelad de la solución realizado en Rational Rouse

4.05. Modelo Lógico – Físico

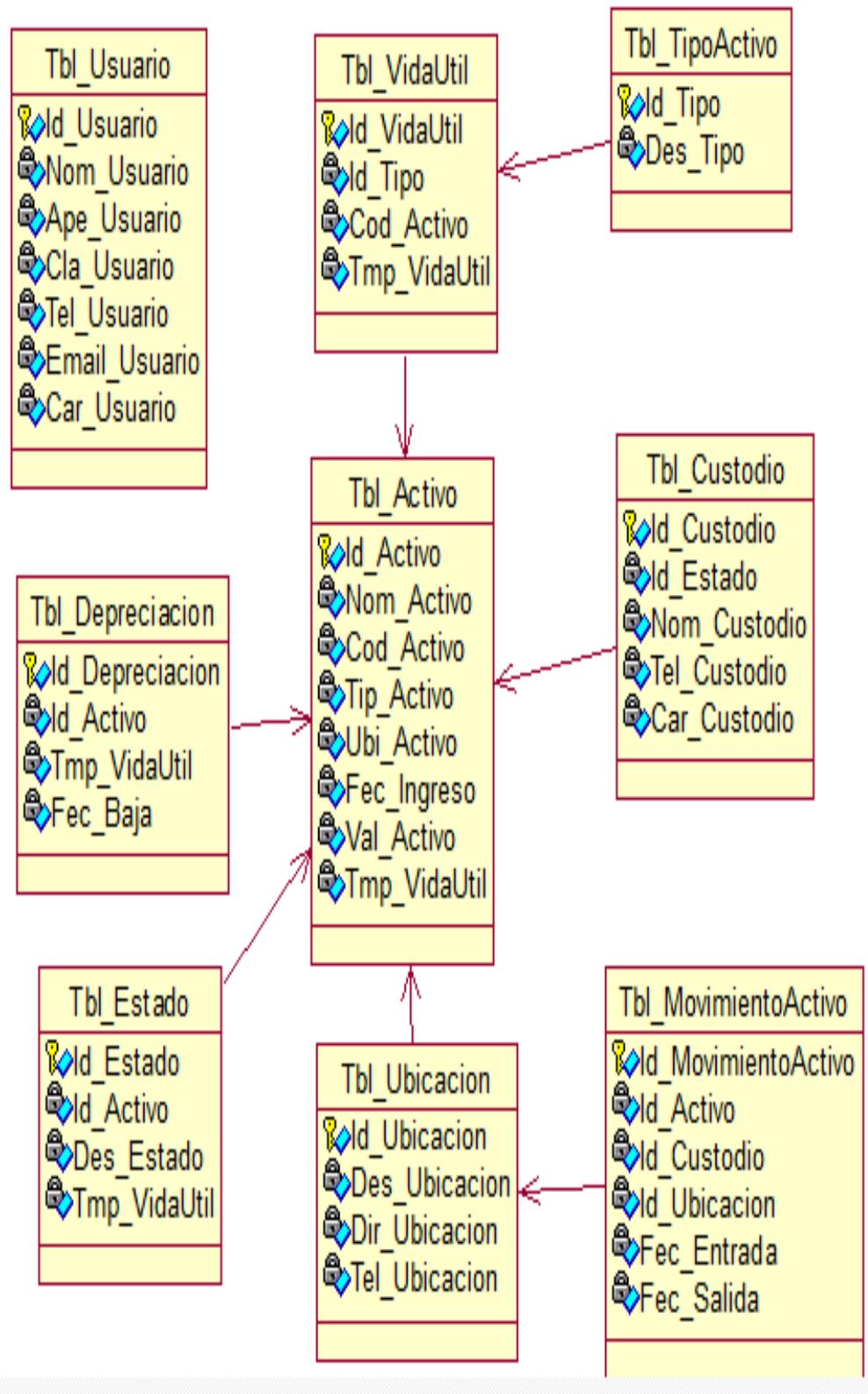


Fig 18. Modelo Lógico: Muestra la infraestructura de la base de datos

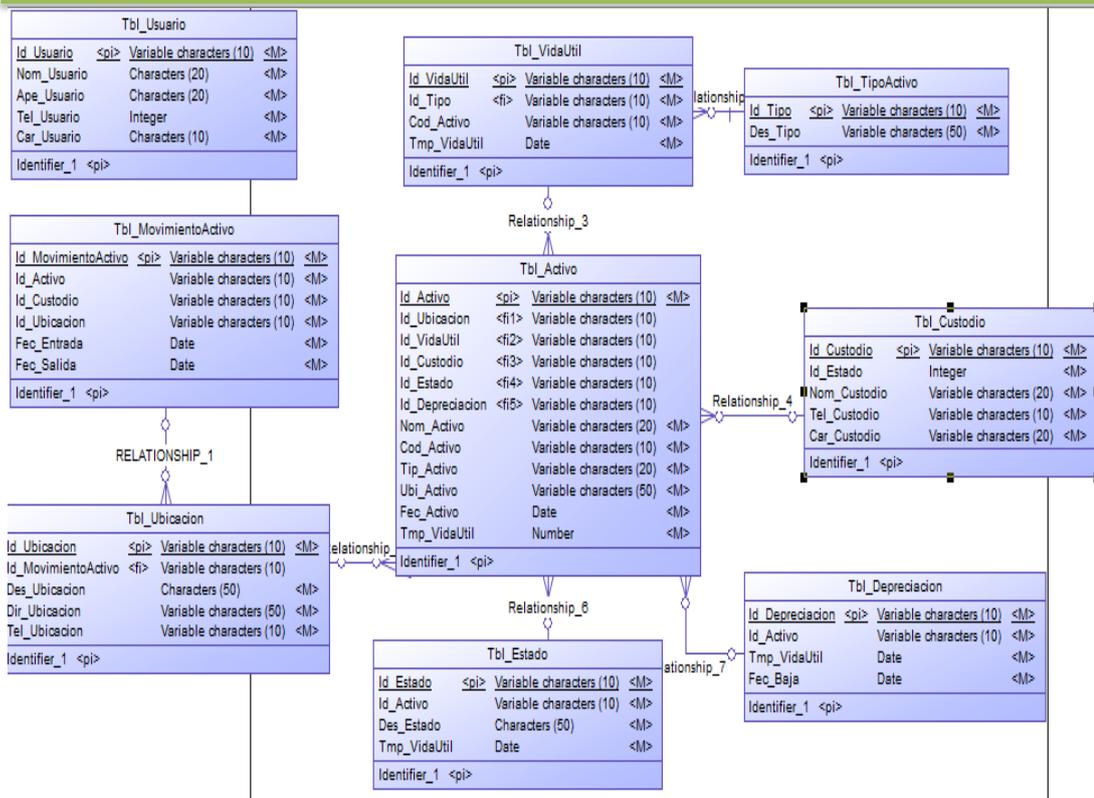


Fig 19. Modelo Físico: Es la base de datos del sistema

4.06. Diagrama de Componentes

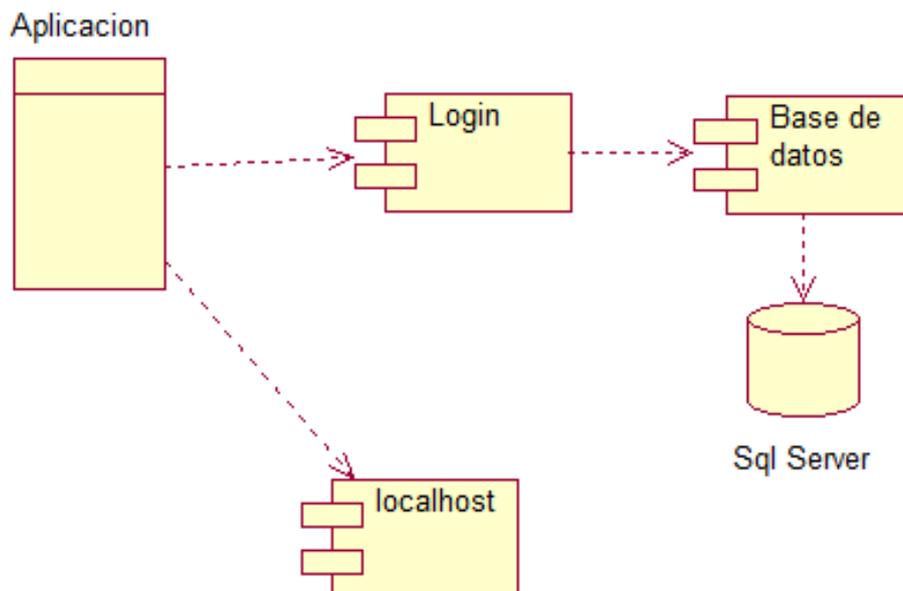


Fig 20. Diagrama de Componentes: Esta Figura muestra los componentes de la aplicación.

4.07 Diagramas de Estrategias

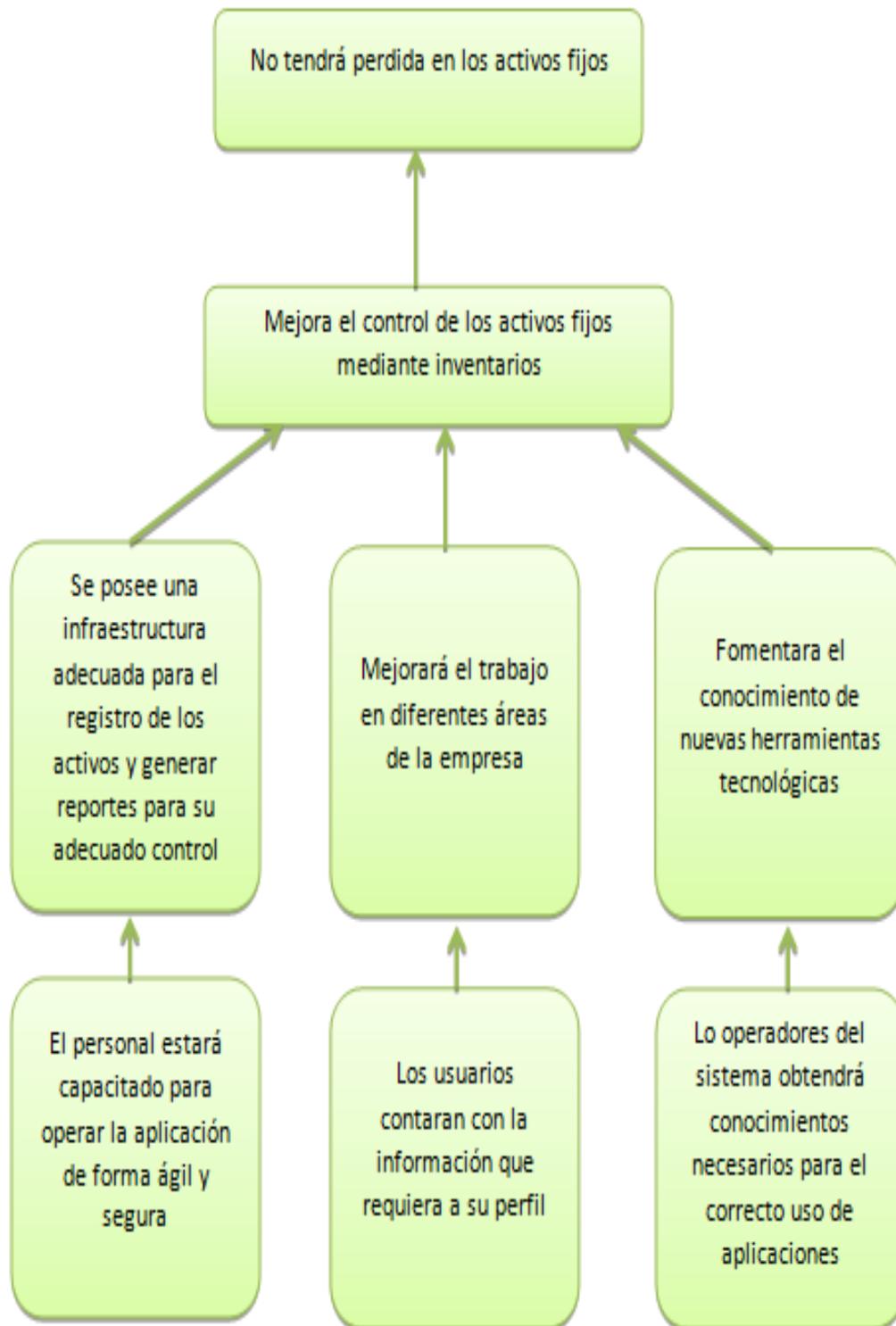


Fig 21. Diagrama de Estrategias: Esta Figura muestra los componentes para alcanzar y llegar a la finalidad del proyecto

4.08 Matriz de Marco Lógico

Facilita los procesos a realizar con la aplicación como el diseño la ejecución y evaluación del sistema, permite medir el nivel de satisfacción y logro alcanzado.

Tabla 30:

Matriz de Marco Lógico

Finalidad	Indicadores	Medios de verificación	Supuestos
Reducción del tiempo utilizado para el inventario de activos fijos.	Se controlara mediante tiempo registrado en el sistema.	Sera controlado en reportes digitales e impresos.	Mayor control de los activos fijos.
Propósito			
Aumento de seguridad en el control de activos.	Se verá reflejado en el menor gasto en activos.	Se podrá imprimir un reporte de las depreciaciones de los activos.	Organización y agilidad en los procesos de control.
Componentes			
El personal será capacitado para el buen uso de la aplicación en los registros y en la generación de reportes.	Mejorar la eficiencia en la empresa mediante el manejo de herramientas tecnológicas.	Satisfacción de los usuarios por el perfecto funcionamiento de la aplicación.	Se fomentara nuevas estrategias de trabajo mediante herramientas tecnológicas.
Actividades			
Gracias a esta aplicación los usuarios tendrán acceso a información por su perfil	Mayor facilidad en el manejo de información de los activos	El usuario se siente identificado con la aplicación	Las estrategias para inculcar nuevas tecnologías crecen

4.09 Vistas arquitectónicas

4.09.01 Vista lógica

Da acogida a los requerimientos funcionales lo que el sistema debe lograr brinda a los usuarios.

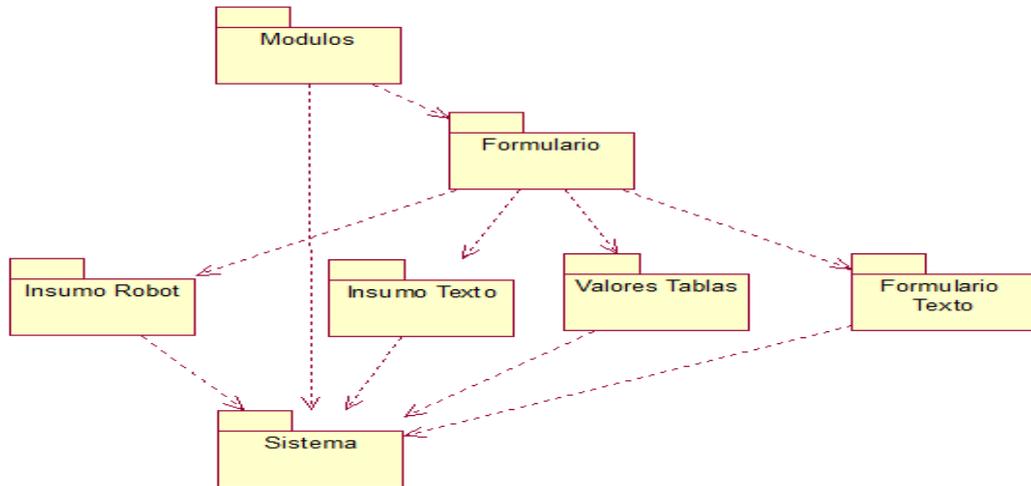


Fig 22. Vista Lógica: Describen los componentes que se aplican en el lenguaje modelador.

4.09.02. Vista física

Son los requisitos no funcionales del sistema como la disponibilidad, confiabilidad y escalabilidad.

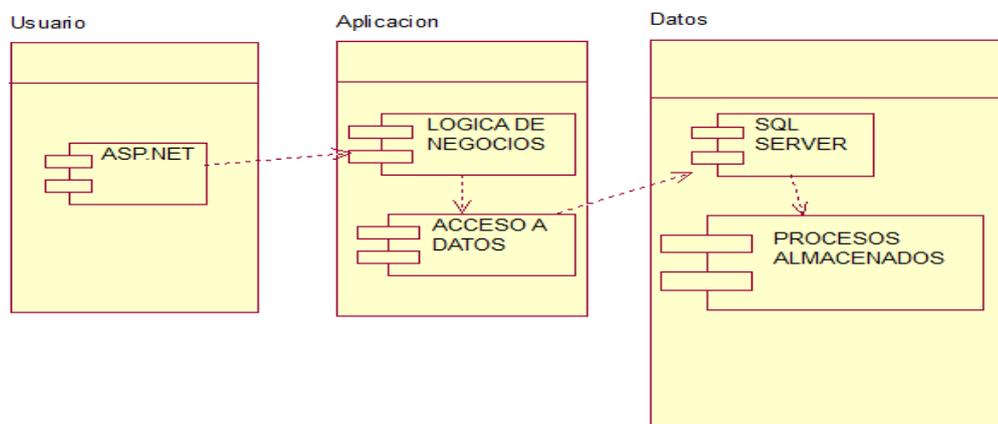


Fig 23. Vista Física: Modelado de vistas desde el usuario a los datos almacenados.

4.09.03 Vista de desarrollo

Son los módulos que el programador acopla para establecer un ambiente ideal de desarrollo por capas.

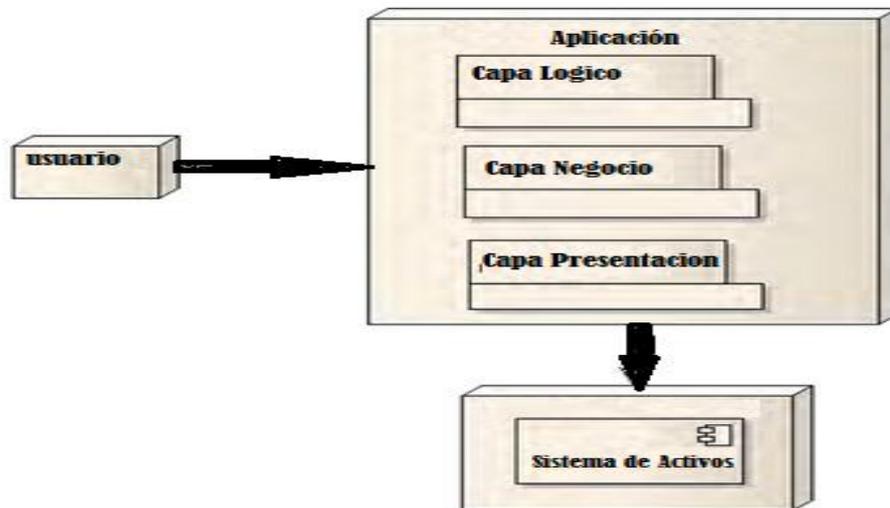


Fig 24. Vista de Desarrollo: Muestra los módulos de desarrollo de la aplicación.

4.09.04 Vista de procesos

Detalla los procesos que realizan cada uno de los usuarios con las restricciones que tiene cada uno de ellos como el administrador y el usuario.

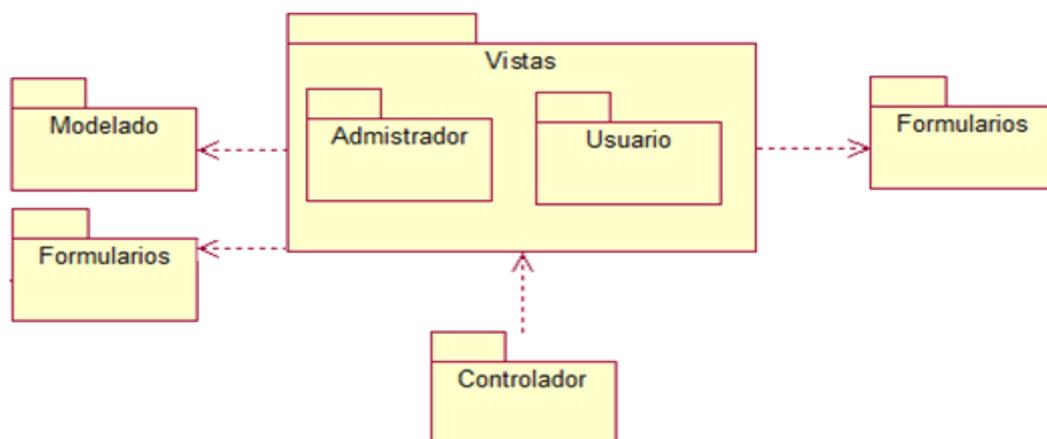


Fig 25. Vista de Procesos: Muestra los procesos de usuario y administrador

Capítulo V: Propuesta

5.01 Especificación de los estándares de Programación.

En la programación de aplicaciones tecnológicas es necesario identificar y reconocer de inmediato los controles, tipos de variables que se utilizan para su desarrollo.

Tabla 31:

Controles de Programación

Control	Imagen	Prefijo	Ejemplo
Label		Lbl	Lbl_nombre
TextBox		Txt	Txt_custodio
CommandButton		Btn	Btn_generar
OptionButton		Opt	Opt_usuario
CheckBox		Chk	Chk_estado
ComboBox		Cmb	Cmb_ingreso
ListBox		Lst	Lst_usuario
Image		Img	Img_logo

Tabla 32:

Variables de datos

Tipo	Memoria	Equivalencias	Observaciones
Int	2-4 Bytes	Entero	Controladores, bucles, etc.
Long	4-8 Bytes	Entero	Características similar al int pero admite valores más amplios.
Single	4 Bytes	Real	Hasta 6 decimales, admite enteros.
Double	8 Bytes	Real	Hasta 14 decimales, admite enteros.
Boolean	2 Bytes	Booleano	Valor falso predeterminado y asociado con un 0
String	10 más 1 cada carácter	Alfanumérica	Texto de longitud variable
Variant	Variable	No tiene	Admite números enteros, decimales o variables

5.02 Diseño de interfaces de usuario

ADMINISTRADOR



Fig 26. Interfaz de Inicio: Esta Figura muestra la ventana de inicio del sistema

A.- Nombre de la empresa

B.- Botón Inicio

C.- Botón Ingreso al sistema

D.-Logos de la empresa

The image shows a login form titled "LOGIN". It contains two input fields: "USUARIO" and "CONTRASEÑA". Below these fields is a green button labeled "Ingresar". At the bottom left, there is a button labeled "Inicio". Red callout boxes with letters A, B, C, and D point to the "USUARIO" field, the "CONTRASEÑA" field, the "Ingresar" button, and the "Inicio" button, respectively.

Fig 27. Interfaz Ingreso Sistema: Esta Figura muestra el usuario y la contraseña para ingresar al sistema.

A.- Cuadro Usuario

B.- Cuadro Contraseña

C.- Botón Ingresar

ABS Computer S.A. Ecuador



Usuario: ppilataxi (A)
 Nombre Usuario: Patricio (B)
 Apellido Usuario: Pilataxi (C)
 Perfil: Administrador (D)
 Contraseña: (E)
 Confirmar Contraseña: (F)

(G)

Código	Perfil	Nombre	Apellido	Contraseña	Editar (H)
admin	Administrador	admin	admin		Editar
ccauja	Administrador	Guillo	Cauja		Editar

Soluciones Informáticas Integrales

Fig 28. Ventana Registro de Usuarios: Esta ventana muestra los campos de texto a llenar para el registro de usuario.

A.- Cuadro de Texto Usuario

B.- Cuadro de Texto Nombre Usuario

C.- Cuadro de Texto Apellido Usuario

D.- Cuadro de Texto Perfil Usuario

E.- Cuadro de Texto Contraseña Usuario

F.- Cuadro de Texto Confirmación de Contraseña

G.- Botón Ingresar Usuario

H.- Opción Edición Usuario

ABS Computer S.A. Ecuador



Ubicacion	Matriz	A
Area	Sistemas	B
Tipo Activo	Telenos	C
Nombre	Samsung	D
Modelo	S5	E
Serial	mx23-4	F
Valor	700	G
Fecha de Ingreso	3/30/2015	H
<input type="button" value="Ingresar"/>		I

Codigo	Tipo Activo	Nombre	Area	J
ce0001	Contrucciones y Edificaciones	Oficina Central	Contabilidad	
te0001	Telenos	sony	Contabilidad	

Soluciones Informáticas Integrales

Fig 29. Ventana Registro de Activos: Esta ventana muestra los campos de texto a llenar para el registro de activos..

A.- Cuadro de Texto Ubicación del Activo

B.- Cuadro de Texto Área

C.- Cuadro de Texto Tipo de Activo

D.- Cuadro de Texto Nombre de Activo

E.- Cuadro de Texto Modelo de Activo

F.- Cuadro de Texto Serial del Activo

G.- Cuadro de Texto Valor del Activo

H.- Cuadro de Texto Fecha de Ingreso

I.- Botón Ingresar Activos

J.- Opción Editar Activo

ABS Computer S.A. Ecuador



Nombre **A**

Identificador **B**

Tiempo Util **C**

D

Codigo	Nombre	Identificador	Tiempo Util	Editar
1	Contrucciones y Edificaciones	ce	10	Editar E
2	Maquinaria y Equipo	me	10	Editar
3	Equipos de Computo	ec	4	Editar
4	Telenos	te	5	Editar

Soluciones Informáticas Intearales

Fig 30. Ventana Registro Tipo de Activo: Esta ventana muestra los campos de texto a llenar para el registro de tipo de activo.

A.-Cuadro de Texto Nombre Tipo de Activo

B.- Cuadro de Texto Identificador Tipo de Activo

C.- Cuadro de Texto Tiempo Útil

D.- Botón Ingresar Tipo de Activo

E.- Opción Edición Tipo de Activo

ABS Computer S.A. Ecuador



Nombre **A**

Dirección **B**

Telefono **C**

Ubicacion Principal **D**

E

ID	Nombre	Direccion	Telefono	Principal	Editar
1	Matriz	Parque Ingles	3103169	True	Editar F
2	Sucursal Real Audiencia	Los Pinos	3465789	False	Editar
3	Sucursal El Recreo	Alonso de Angulo	3456767	False	Editar
4	Matriz Uno	Los alpes	234534	False	Editar

Soluciones Informáticas Integrales

Fig 31. Ventana de Ubicación de los Activos: Esta ventana muestra los campos de texto a llenar para el registro de Ubicación de los activos.

A.- Cuadro de Texto Nombre Ubicación

B.- Cuadro de Texto Dirección Ubicación

C.- Cuadro de Texto Teléfono de la Ubicación

D.- Opción de Ubicación Principal

E.- Botón Ingresar Ubicación

F.- Opción Editar Ubicación

ABS Computer S.A. Ecuador



Area: Contabilidad (A)

Cedula: (B)

Nombre: (C)

Apellido: (D)

Fecha de Ingreso: (E)

Responsable de Area: (F)

Ingresar (G)

ID	Area	Nombre	Apellido	Fecha Ingreso	Principal	Editar
1	Contabilidad	Juan	Muñoz		True	Editar (H)
2	Sistemas	Carlos	Espinola		True	Editar
3	Bodega	Ricardo	Montalvo		True	Editar
4	Administracion	Laura	Montoya		True	Editar
5	Sistemas	Ricardo	Mendez	2015-03-30 00:00:00	False	Editar

Soluciones Informáticas Integrales

Fig 32. Ventana de Registro Empleados: Esta ventana muestra los campos de texto a llenar para el registro de los empleados.

A.- Cuadro de Texto Área Empleado

B.- Cuadro de Texto Cedula Empleado

C.- Cuadro de Texto Nombre Empleado

D.- Cuadro de Texto Apellido Empleado

E.- Cuadro de Texto Fecha de Ingreso

F.- Cuadro de Texto Responsable de Área

G.- Botón Ingresar Empleado

H.- Opción Editar Empleado

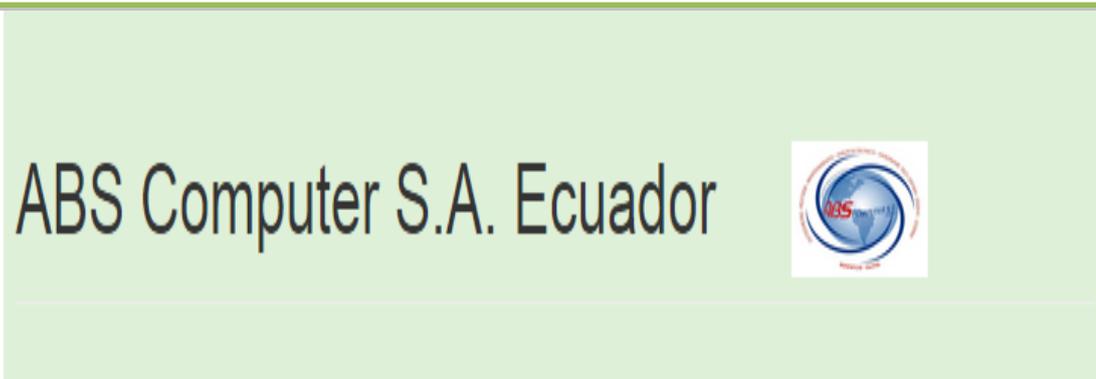
A screenshot of a web application window titled "Depreciación de Activo". The window contains several input fields and buttons. At the top left, there is a label "Tipo de Activo" above a text input field containing "Telenos", marked with a red callout 'A'. Below it is a label "Activo" above a text input field containing "te0001-sony", marked with a red callout 'B'. At the bottom left, there are two buttons: "Consultar" (marked with a red callout 'C') and "Nueva Consulta" (marked with a red callout 'D'). At the bottom center, there is a text label "Soluciones Informaticas Integrales".

Fig 33. Ventana Depreciación de Activo: Esta ventana muestra los campos de texto a seleccionar para el cálculo de depreciación de los activos.

- A. Cuadro de Texto Tipo de Activo
- B. Cuadro de Texto Activo
- C. Botón Consultar Depreciación de Activo
- D. Botón Nueva Consulta

ABS Computer S.A. Ecuador



Tipo de Activo	Telenos	A
Detalle Activo	te0001-sony	B
Código Activo	009875	C
Custodio	Ricardo Mendez	D
Nombre Autoriza	Juan Muñoz	E
	<input type="button" value="Registrar"/>	F

Soluciones Informáticas Integrales

Fig 34. Ventana Salida de Activos: Esta ventana muestra los campos de texto a seleccionar para el registro de salida del activo.

A.- Cuadro de Texto Tipo de Activo

B.- Cuadro de Texto Detalle de Activo

C.- Cuadro de Texto Código del Activo

D.- Cuadro de Texto Custodio del Activo

E.- Cuadro de Texto Nombre Autoriza

F.- Botón Registrar la Baja de Activo

ABS Computer S.A. Ecuador



Código Activo
Empleado
Ubicación
Tipo de Activo

A

-ESCOJA UN EMPLEADO- ▾ **B**

-ESCOJA UNA UBICACION- ▾ **C**

-ESCOJA UN TIPO DE ACTIVO- ▾ **D**

E

Soluciones Informáticas Integrales

Fig 35. Ventana Consulta de Activos: Esta ventana muestra los campos de texto a seleccionar para la consulta de activos.

A.- Cuadro de Texto Código Activo

B.- Cuadro de Texto Empleado

C.- Cuadro de Texto Ubicación

D.- Cuadro de Texto Tipo de Activo

E.- Botón Consultar el Activo

5.03 Especificación pruebas de unidad

Estas pruebas facilitan el entendimiento del sistema por lo que el código es más fácil de entender.

Tabla 33

Pruebas de unidad: Validación Usuario

Identificador de la Prueba	PDU001
Método a Probar	Validación del Usuario
Objetivo de la Prueba	Dar constancia que la validación de usuario funcione de manera correcta con sus condiciones, usuario y contraseña
Datos de Entrada	<ol style="list-style-type: none"> 1.- Usuario 2.- Contraseña
Resultado Esperado	<ol style="list-style-type: none"> 1.- Usuario este registrado 2.- Contraseña correcta 3.- Perfil del usuario correcto
Comentario	<ol style="list-style-type: none"> 1.- Si se ingresa erróneamente la contraseña o el usuario el sistema se bloqueara al usuario hasta que el administrador de tal lo habilite de nuevo.

Tabla 34

Pruebas de unidad: Registro Usuario

Identificador de la Prueba	PDU002
Método a Probar	El registro de Usuarios en la aplicación
Objetivo de la Prueba	Comprobar que el método de registro de usuarios sea factible
Datos de Entrada	
<ul style="list-style-type: none"> 1.- Identificación Usuario 2.- Nombre 3.- Apellido 4.- Perfil 5.- Contraseña 6.- Verificación Contraseña 7.- Botón Ingresar 	
Resultado Esperado	
<p>1.- En todos los campos se ingresan solo letras excepto en la contraseña ya que su contenido puede ser números y letras el cuadro de texto es encriptado por motivos de seguridad.</p>	
Comentario	
<ul style="list-style-type: none"> 1.- Los usuarios tienen perfil asignado por el administrador 	

Tabla 35

Prueba de Unidad: Registro de Activos

Identificador de la Prueba	PDU003
Método a Probar	El registro de Activos en la aplicación
Objetivo de la Prueba	Comprobar que el método de registro de activos sea factible
Datos de Entrada	
<ul style="list-style-type: none"> 1.-Ubicación 2.- Tipo 3.- Nombre 4.- Modelo 5.- Serial 6.- Valor 7.- Estado 	
Resultado Esperado	
<ul style="list-style-type: none"> 1.-Ubicación se conecta con el nombre de la empresa registrado en Ubicación 2.- Tipo se conecta con el tipo de activo registrado en Tipo de Activos 3.- Nombre campo validado 4.- Modelo campo validado 5.- Serial campo validado 6.- Valor campo validado 7.- Estado se conecta con los estados de la empresa 	
Comentario	
1.- El registro de activos van asociados con otros campos del sistema	

Tabla 36

Prueba de Unidad: Registro Tipo de Activo

Identificador de la Prueba	PDU004
Método a Probar	El registro de Tipo de Activo en la aplicación
Objetivo de la Prueba	Comprobar que el método de registro tipo de activos sea factible
Datos de Entrada	<p>1.-Nombre</p> <p>2.- Identificador</p>
Resultado Esperado	<p>1.-Nombre campo validado</p> <p>2.- Identificador campo validado</p>
Comentario	<p>1.- El identificador del activo debe ir asociado al activo que se registra.</p>

Tabla 37

Prueba de Unidad: Ubicación de Activos

Identificador de la Prueba	PDU005
Método a Probar	El registro de Ubicación de Activos en la aplicación
Objetivo de la Prueba	Comprobar que el método de Ubicación de activos sea factible
Datos de Entrada	<ol style="list-style-type: none">1.- Nombre2.- Dirección3.- Teléfono4.- Ubicación
Resultado Esperado	<ol style="list-style-type: none">1.- Nombre campo validado2.- Dirección campo validado3.- Teléfono campo validado4.- Ubicación a selección de campo
Comentario	<ol style="list-style-type: none">1.- Ninguno

Tabla 38

Prueba de Unidad: Vida Útil

Identificador de la Prueba	PDU006
Método a Probar	El registro de la Vida Útil de Activos en la aplicación
Objetivo de la Prueba	Comprobar que el método de Vida Útil de activos sea factible
Datos de Entrada	<p>1.- Tipo de Activo</p> <p>2.- Tiempo Útil</p>
Resultado Esperado	<p>1.- Tipo de Activo es conectado con la Tabla Vida Útil</p> <p>2.- Tiempo Útil es el cálculo que realiza el sistema para calcular el tiempo de vida útil</p>
Comentario	<p>1.- Ninguno</p>

Tabla 39

Prueba de Unidad: Depreciación de Activos

Identificador de la Prueba	PDU007
Método a Probar	El registro de la Depreciación de Activos en la aplicación
Objetivo de la Prueba	Comprobar que el método de Depreciación de activos sea factible
Datos de Entrada	<p>1.- Tipo de Activo</p> <p>2.- Activo</p>
Resultado Esperado	<p>1.- Tipo de Activo se conecta con los registros obtenidos en Tipo de Activos</p> <p>2.- Activo se conecta con los registros obtenidos en Registro de Activos</p>
Comentario	1.- Los datos ingresado no deben ser erróneos caso contrario no mostrara la depreciación.

Tabla 40

Prueba de Unidad: Salida de Activos

Identificador de la Prueba	PDU008
Método a Probar	El registro de la Salida de Activos en la aplicación
Objetivo de la Prueba	Comprobar que el método de Salida de Activos sea factible
Datos de Entrada	<ul style="list-style-type: none"> 1.- Tipo de Activo 2.- Detalle 3.- Código 4.- Modelo 5.- Serie 6.- Custodio 7.- Nombre Autoriza
Resultado Esperado	<ul style="list-style-type: none"> 1.- Tipo de Activo se conecta con los registros obtenidos en Tipo de Activos 2.- Detalle selecciona datos ingresados 3.- Código ingresa código de activo 4.- Modelo campo validado 5.- Serie campo validado 6.- Custodio selecciona datos ingresados 7.- Nombre Autoriza selecciona datos ingresados
Comentario	<ul style="list-style-type: none"> 1.- Genera reportes de salida

Tabla 41

Prueba de Unidad: Consulta de activos

Identificador de la Prueba	PDU009
Método a Probar	El registro de la Consulta de Activos en la aplicación
Objetivo de la Prueba	Comprobar que el método de Consulta de Activos sea factible
Datos de Entrada	<ul style="list-style-type: none"> 1.- Serial Activo 2.- Custodio 3.-Ubicación 4.-Tipo de Activo
Resultado Esperado	<ul style="list-style-type: none"> 1.- Serial Activo se ingresa la serie de activo 2.- Custodio selecciona datos ingresados 3.-Ubicación selecciona datos ingresados 4.-Tipo de Activo selecciona datos ingresados
Comentario	<ul style="list-style-type: none"> 1.- Genera reportes de los activos

5.04 Especificación de pruebas de aceptación

Tabla 42

Prueba Aceptación: Usuarios

Identificador de la Prueba	EPA001
Caso de uso	Ninguno
Tipo de usuario	Administrador
Objetivo de la Prueba	Usuarios operando el sistema correctamente
Secuencia de Eventos	<ol style="list-style-type: none"> 1.- Registrar Datos del Usuario 2.- Asignar un tipo de usuario 3.- Asignar un usuario y contraseña
Resultado Esperado	<ol style="list-style-type: none"> 1.- Usuarios registrados exitosamente 2.- El sistema cumplirá las funciones de acuerdo al perfil de usuario
Comentario	El sistema cumple con los requerimientos de la prueba
Estado:	Aceptado

Tabla 43

Pruebas de Aceptación: Registro de Activo

Identificador de la Prueba	EPA002
Caso de uso	UC001
Tipo de usuario	Administrador
Objetivo de la Prueba	Registrar datos de los activos fijos en el sistema
Secuencia de Eventos	1.- Registrar Activos 2.- Generar reportes de los Activos
Resultado Esperado	1.- El sistema cumplirá las funciones de acuerdo al perfil del tipo de usuario
Comentario	El sistema Funciona correctamente cumpliendo con las expectativas del requerimiento
Estado:	Aceptado

Tabla 44

Pruebas de Aceptación: Baja de Activos

Identificador de la Prueba	EPA003
Caso de uso	UC002
Tipo de usuario	Administrador
Objetivo de la Prueba	Registrar datos de un activo depreciado y dado de baja
Secuencia de Eventos	<ol style="list-style-type: none">1.- Generar depreciación2.- Generar reporte3.- Autorización dar de baja
Resultado Esperado	<ol style="list-style-type: none">1.- El sistema cumplirá las funciones de acuerdo al perfil del tipo de usuario
Comentario	El sistema Funciona correctamente cumpliendo con las expectativas del requerimiento
Estado:	Aceptado

5.05 Especificación Pruebas de Carga

Todo sistema informático debe pasar una serie de pruebas para observar posibles falencias que puedan corromper el buen funcionamiento de la aplicación y su rendimiento.

Por ello se realizó la prueba de caja negra que consiste en revisar el sistema con sus respectivas líneas de código

Tabla 45

Prueba de Carga: Usuarios

Identificador de la Prueba	EPC001
Tipo de Prueba	Usuarios
Objetivo	Colapsar el sistema con exceso de usuarios en línea
Descripción	Ingresar al sistema varios usuarios y probar el rendimiento
Resultados Esperados	El sistema funcione correctamente sin ningún error y su rendimiento sea optimo
Comentarios	La prueba cumplió con los resultados esperados

Nota: Significado de las siglas EPC; **EPC**=Especificación Pruebas de Carga

5.06 Configuración del Ambiente mínima/ideal

Las aplicaciones como otros comandos del software necesitan interactuar con un hardware acorde a lo que requiere el sistema para un óptimo funcionamiento.

Tabla 46

Requerimientos de la aplicación

Requerimientos mínimos de Hardware	
Procesador	Inter Core 2 Dúo
Disco Duro	150 Gb
Memoria RAM	2 Gb
Requerimientos Recomendados de Hardware	
Procesador	Inter Core i3
Disco Duro	500Gb
Memoria RAM	4 Gb
Equipo del Cliente	
Procesador	Inter Core 2 Dúo
Disco Duro	150 Gb
Memoria RAM	2 Gb
Requerimiento del Software	
SQL Server 2010, 2012	Base de Datos
Windows 7, 8	Sistema Operativo
Visual Studio 2010,2012	Compilador
Equipo del Cliente	
Windows 7, 8	Sistema Operativo
Conexión a internet	Red

Capítulo VI: Aspectos Administrativos

6.01 Recursos

Tabla 47

Recursos

Recursos Humanos			
Humano	Actividad	Nombre	Responsabilidad
Tutor	Director del Proyecto	Ing. Marco Obando	Guía e instruye
Gerente	Autoriza la elaboración del sistema	Ing. Bolívar Alvares	Toma de decisiones
Secretaria, Contador	Proporciona Información		Ejecutar correctamente el control y registro de Activos
Usuario	Consultor		Verificar los activos a cargo
Recursos Económicos			
Descripción	Cantidad	Valor Unitario	Valor Total
Computadora	1	750,00	750,00
Impresiones Color B/N	600	0,25	150,00
Impresora laser	1	299,99	299,99
Empastado	1	7,00	7,00
Anillados	3	3,00	9,00
Internet	6	22,00	132,00
Servicios Básicos	5	15,00	75,00
Tutorías	1	200,00	200,00
Seminario	1	522,00	522,00
Transporte	70	0,25	17,50
Alimentación	100	2,00	200,00
Capacitación	1	100,00	100,00
Valor Total			2461,50

6.02 Presupuesto

Es un reporte detallado en el todos los gastos que se invirtieron para la elaboración del proyecto por parte del alumno y la empresa auspiciante.

Tabla 48:

Presupuesto

Presupuesto			
Detalle	Valor Unitario	Cantidad	Valor Total
Computadora	1	750,00	750,00
Impresiones Color B/N	600	0,25	150,00
Impresora laser	1	299,99	299,99
Empastado	1	7,00	7,00
Anillados	3	3,00	9,00
Internet	6	22,00	132,00
Servicios Básicos	5	15,00	75,00
Tutorías	1	200,00	200,00
Seminario	1	522,00	522,00
Transporte	70	0,25	17,50
Alimentación	100	2,00	200,00
Capacitación	1	100,00	100,00
Valor Total			2461,50

6.03 Cronograma

Es la representación del tiempo invertido para la elaboración del proyecto de titulación.

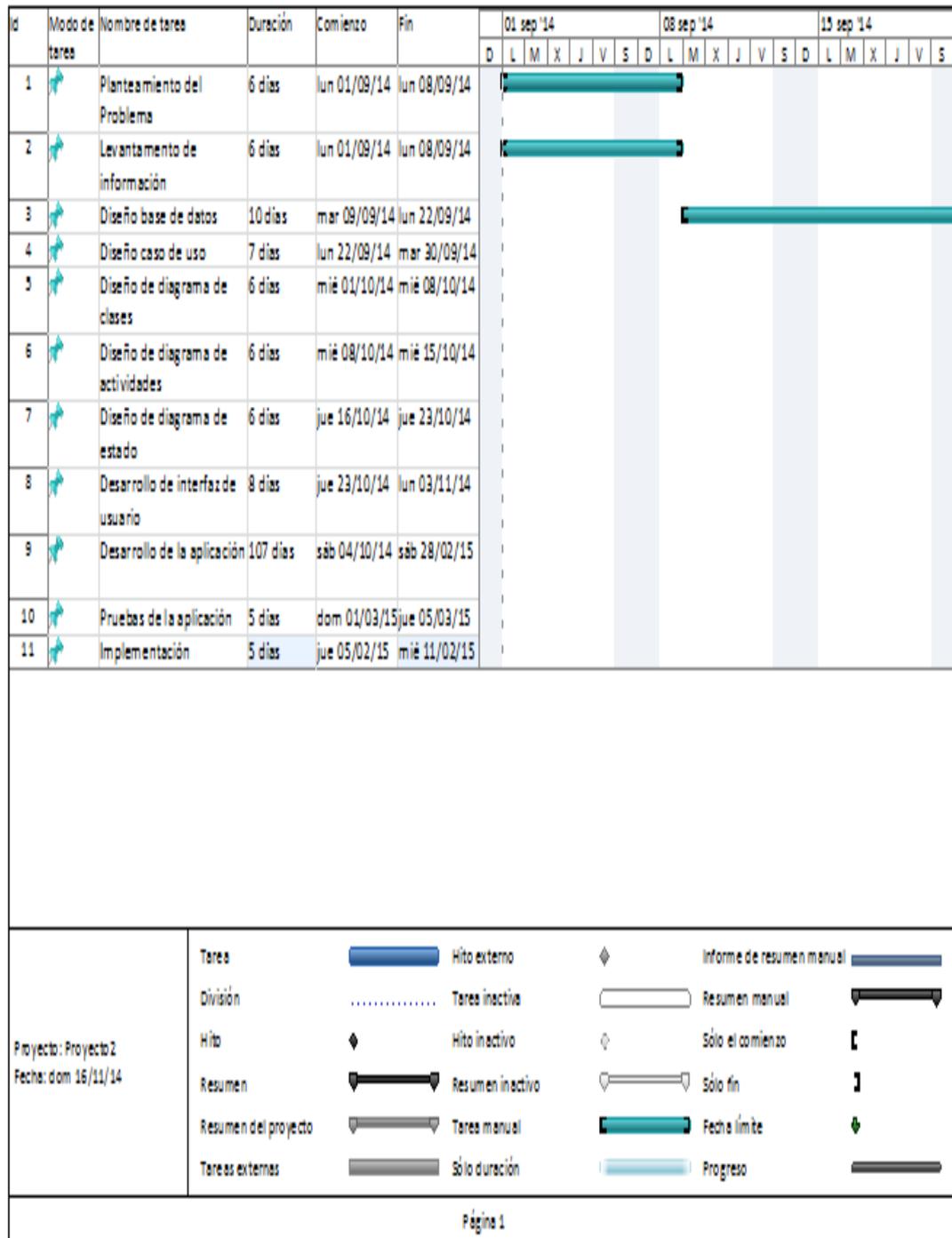


Fig 36. Cronograma: Muestra el tiempo de elaboración del proyecto

Capítulo VII: Conclusiones y Recomendaciones

7.01 Conclusiones

Con la presente investigación de este proyecto se ve que es factible el diseño e implementación de una aplicación tecnológica para el control de los activos fijos en la empresa ABS COMPUTER S.A por lo que optimizara recursos gracias a esta.

Ayudará a mantener la información sistematizada gracias a la base de datos que se implementó y así lograr una búsqueda más ágil y precisa de los activos fijos.

La aplicación contara con un interfaz gráfico de fácil manejo lo que facilitara que el usuario comprenda de mejor manera su funcionamiento.

El sistema generara reportes de los activos existentes ya sea con su respectivo código o por los custodios asignados, en la empresa y fuera de ella demás de ello se generara sus respectivas depreciaciones.

Los desarrolladores de software deben aplicar una metodología que permita el seguimiento para desarrollar programas que satisfagan a los usuarios y a la empresa y así mejorar los procesos internos de la empresa ABS COMPUTER S.A

7.02 Recomendaciones

Para el correcto funcionamiento del sistema los usuarios deberán estar debidamente capacitados y contar con su respectivo manual de usuario.

Si el sistema presenta falencias en su respectivo funcionamiento el usuario deberá reportar al administrador y al personal del área informática para dar solución a tal error.

Los mantenimientos del sistema lo debe realizar solo el administrador debido a que la aplicación posee información privada de la empresa y por seguridad y para el

correcto funcionamiento de la aplicación se cuenta con perfiles tales como usuario (consultor), contador, secretaria (reportes) y administrador.

ANEXOS

A.01

*Requerimientos***MATRIZ DE REQUERIMIENTOS****REQUERIMIENTOS FUNCIONALES**

Identificado	Descripción	Fuente	Prioridad	Tipo	Estados	Usuarios Involucrados
RF001	Registros de activos Fijos en base de datos	Gerente	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria
RF002	Software realiza regístritos de activos con la información obtenida de la base de datos	Gerente	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria
RF003	Búsqueda de activos fijos	Gerente	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria
RF004	Dar de baja activos fijos deteriorados de la empresa	Gerente Contador	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria
RF005	Seguridades para usuarios	Gerente	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria
RF006	Depreciación de los Activos Fijos	Gerente	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria
RF007	Almacenamiento y registro de movimientos de los Activo Fijos en la empresa	Gerente	Alta	Sistema	En revisión	Administrador Contador Custodio Secretaria

REQUERIMIENTOS NO FUNCIONALES

NRF 001	Aplicación amigable y compatible con diferentes navegadores en	Gerente	Media	Usuario	En revisión	Administrador Contador Custodio Secretaria
----------------	--	---------	-------	---------	-------------	---

internet						
NRF 002	Visualización de datos en diferentes ventanas	Gerente	Media	Usuario	En revisión	Administrador Contador Custodio Secretaria
NRF 003	Inicio sencillo para el usuario.	Gerente	Media	Usuario	En revisión	Administrador Contador Custodio Secretaria
NRF 004	Base de datos sin información errónea.	Gerente	Media	Usuario	En revisión	Administrador Contador Custodio Secretaria

A.02

MANUAL DE INSTALACIÓN



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

OPTIMIZACIÓN DE LOS PROCESOS DE ACTIVOS FIJOS MEDIANTE UNA
APLICACIÓN INFORMÁTICA WEB PARA LA EMPRESA ABS COMPUTER S. A EN
LA CIUDAD DE QUITO.

Proyecto de investigación previo a la obtención del título de

Tecnólogo en Análisis de Sistemas

MANUAL DE INSTALACIÓN

Autor: Cauja Pilataxi Guillermo Patricio

Tutor: Ing. Marco Obando

Quito, Abril 2015

Figuras Manual de Instalación

Visual Studio 2010

Fig 1 Instalador Microsoft Visual Studio.....	82
Fig 2 Ventana Instalar Microsoft Visual Studio.....	82
Fig 3 Ventana Asistente Instalación Microsoft Visual Studio.....	83
Fig 4 Ventana Aceptar los términos de licencia Microsoft Visual Studio.....	83
Fig 5 Ventana Ruta de la instalación Microsoft Visual Studio.....	84
Fig 6 Ventana instalador de complementos Microsoft Visual Studio.....	84
Fig 7 Ventana instalador de complementos Microsoft Visual Studio.	85
Fig 8 Ventana icono Microsoft Visual Studio.....	85
Fig 9 Ventana ejecutar como administrador Microsoft Visual Studio.....	86
Fig 10 Elegir configuración de entorno predeterminada.....	86
Fig 11 Ventana principal Microsoft Visual Studio.....	87

INSTALACION SQL SERVER 2008 R2

Fig 12 Icono Instalador SQL SERVER 2008.....	88
Fig 13 Centro de Instalación SQL SERVER 2008.....	88
Fig 14 Nueva instalación.....	89
Fig 15 Setup Support Rules	89
Fig 16 Clave del Producto	90
Fig 17 Términos de Licencia	90
Fig 18 Archivos Auxiliares.....	91
Fig 19 Reglas Auxiliares.....	91
Fig 20 Rol de Instalación.....	92
Fig 21 Selección de Características.....	92
Fig 22 Reglas de Instalación.....	93

Fig 23 Configuración de Instancia.....	93
Fig 24 Requisitos de Espacio en Disco	94
Fig 25 Configurador del Servidor.....	94
Fig 26 Configurador del Servidor.....	95
Fig 27 Configurador del Servidor NT Authority.....	95
Fig 28 Configuración de Análisis Services.....	96
Fig 29 Configuración de Reporting Services.....	96
Fig 30 Informes de errores.....	97
Fig 31 Reglas de Configuración de Instalación.....	97
Fig 32 Listo para Instalar.....	98
Fig 33 Progreso de la Instalación	98
Fig 34 Operación Completada.....	99
Fig 35 Icono SQL SERVER MANAGEMENT.....	99
Fig 36 Conectar SQL SERVER 2008.....	100
Fig 37 Ventana Principal SQL SERVER 2008.....	100

Instalación de Visual Studio 2010

1.- En el icono de Setup dar clic derecho, elegir la opción Ejecutar como Administrador

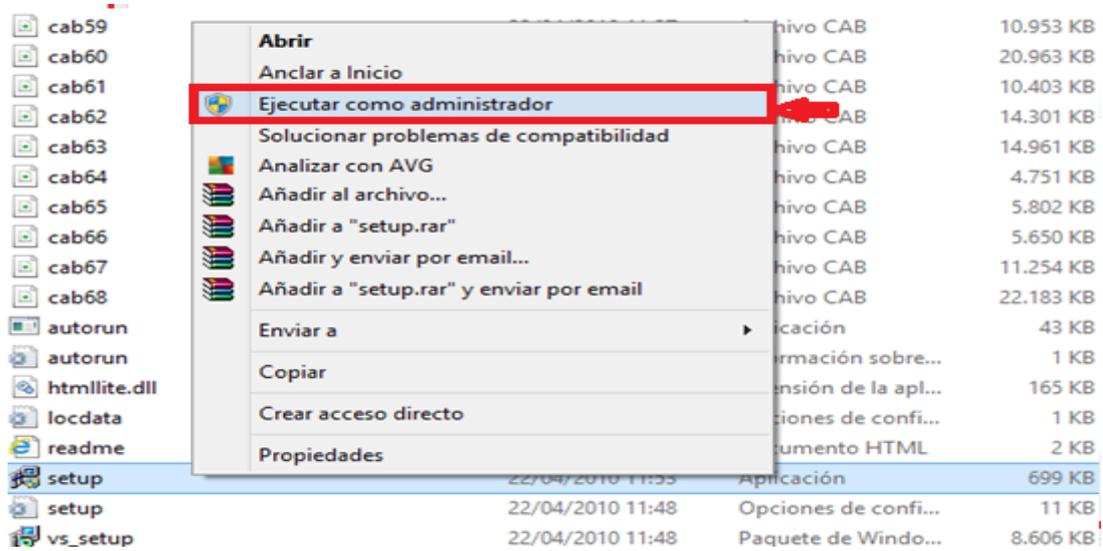


Fig1 Instalador Microsoft Visual Studio

2.- Aparecerá una ventana en la cual debe elegir la opción INSTALAR MICROSOFT VISUAL STUDIO 2010.

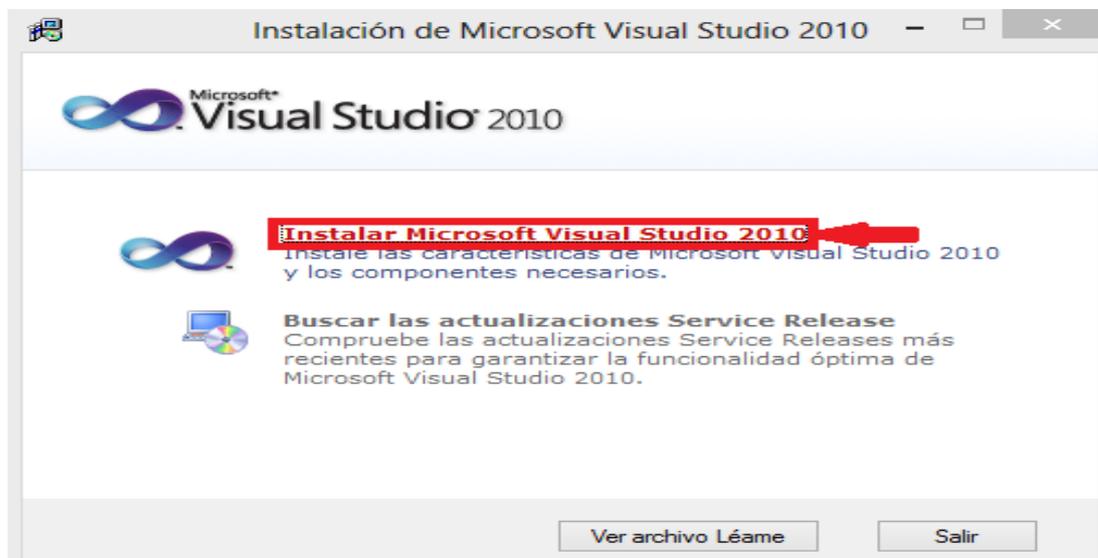


Fig 2 Ventana Instalar Microsoft Visual Studio

3.- En esta ventana debe elegir la opción SIGUIENTE, puede quitar el visto que aparece al lado izquierdo en el cajón ya que es opcional.



Fig 3 Ventana Asistente Instalación Microsoft Visual Studio

4.- En esta ventana puede leer los términos de la licencia de Microsoft e imprimir si lo desea

5.- Dar clic en la casilla HE LEIDO LOS TERMINOS DE LA LICENCIA Y LO ACEPTO

6.-Dar clic en SIGUIENTE

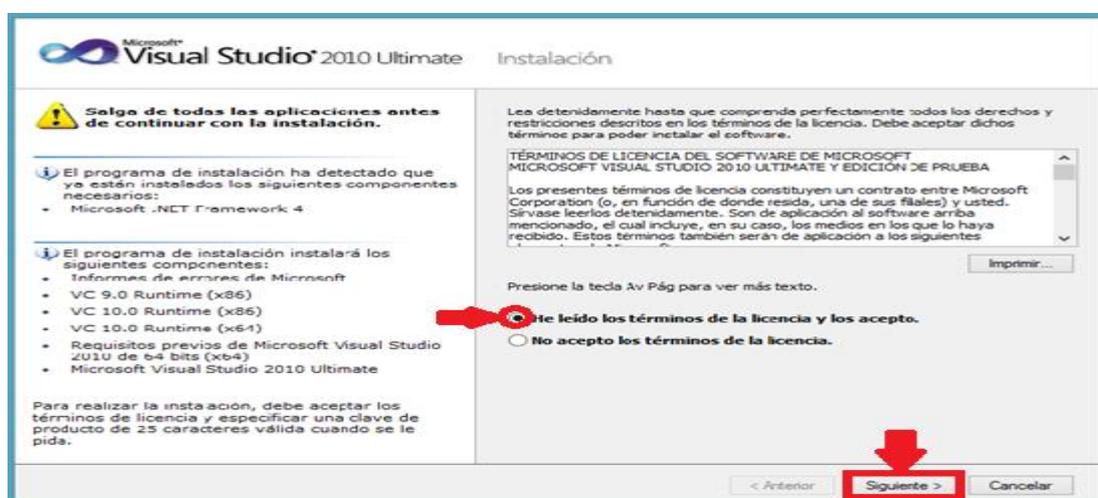


Fig 4 Ventana Aceptar los términos de licencia Microsoft Visual Studio

7.- En el lado izquierdo de esta pantalla aparecerá la opción de instalación en la cual puede elegir que complementos desea instalar por esta ocasión seleccionaremos TODOS.

8.- Al lado derecho de la pantalla aparece la ruta de instalación del sistema.

9.- Elegir el botón INSTALAR.

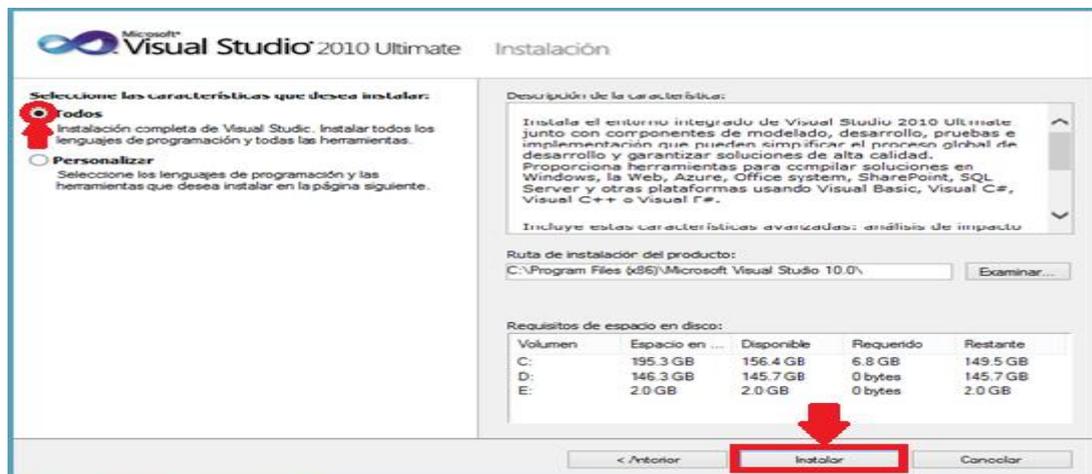


Fig 5 Ventana Ruta de la instalación Microsoft Visual Studio

10.- En esta ventana aparecerán los complementos que se instalarán en su ordenador dar clic en SIGUIENTE.

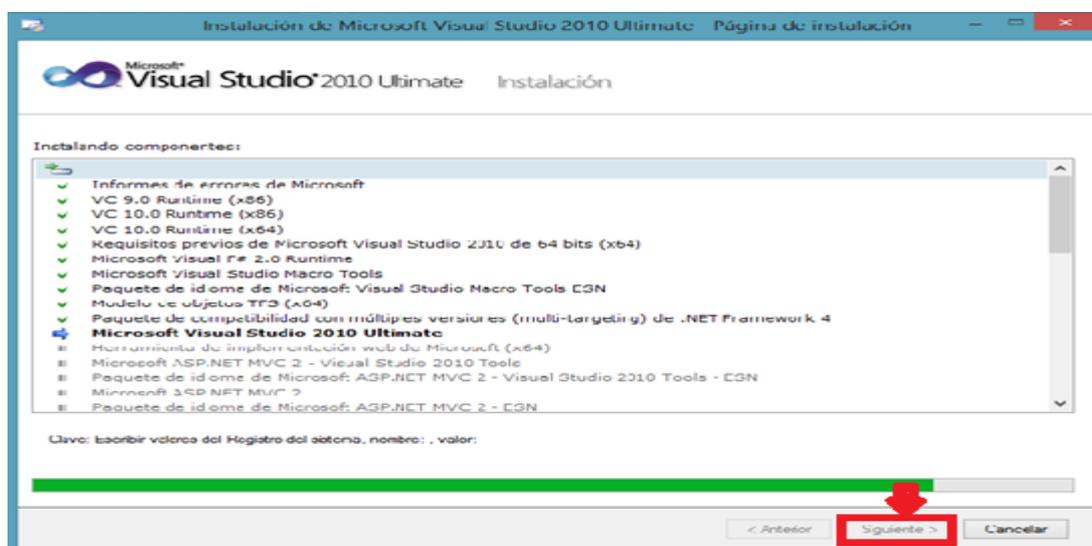


Fig 6 Ventana instalador de complementos Microsoft Visual Studio

11.- Aparecerá una ventana en la cual mostrara el mensaje que el sistema se instaló con satisfacción dar clic en FINALIZAR.



Fig 7 Ventana instalador de complementos Microsoft Visual Studio

12.- Dirigirse al icono MICROSOFT VISUAL STUDIO 2010



Fig 8 Ventana icono Microsoft Visual Studio

13.- Dar clic derecho en el icono de MICROSOFT VISUAL STUDIO 2010 y elegir la opción EJECUTAR COMO ADMINISTRADOR.

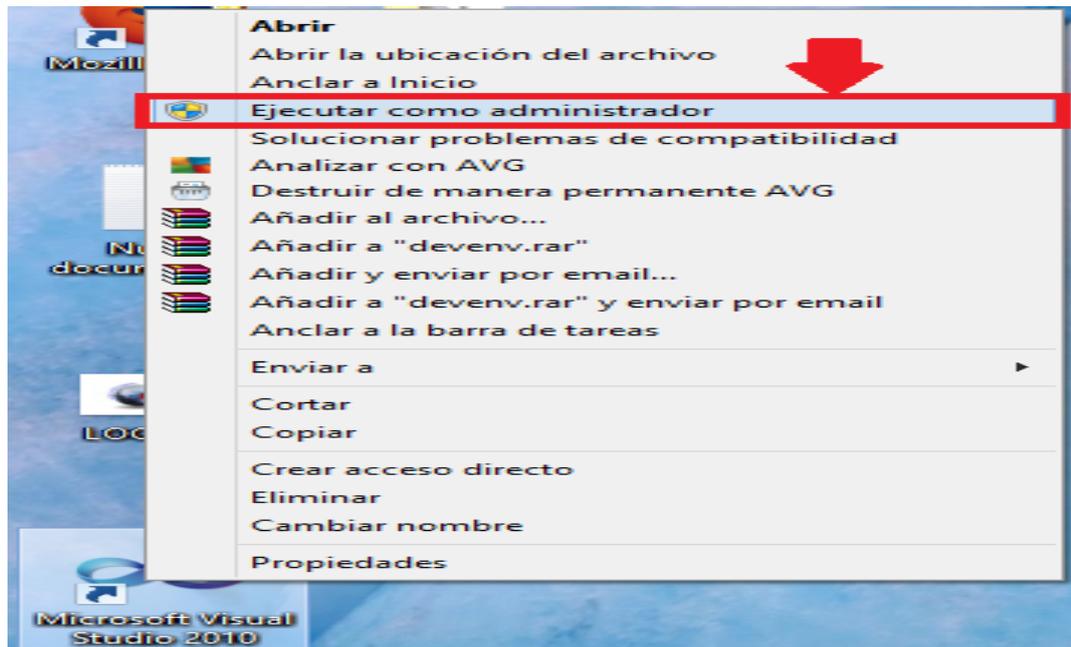


Fig 9 Ventana ejecutar como administrador Microsoft Visual Studio

14.- Deberá elegir la opción INICIAR VISUAL STUDIO

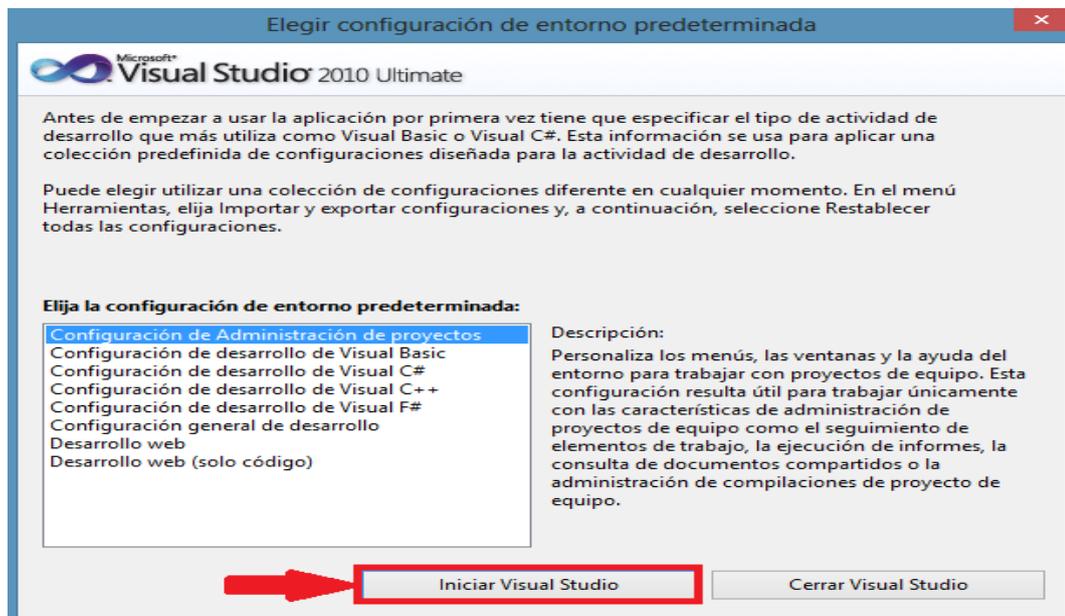


Fig 10 Elegir configuración de entorno predeterminada

15.- Aparecerá la ventana para el desarrollo de la aplicación.

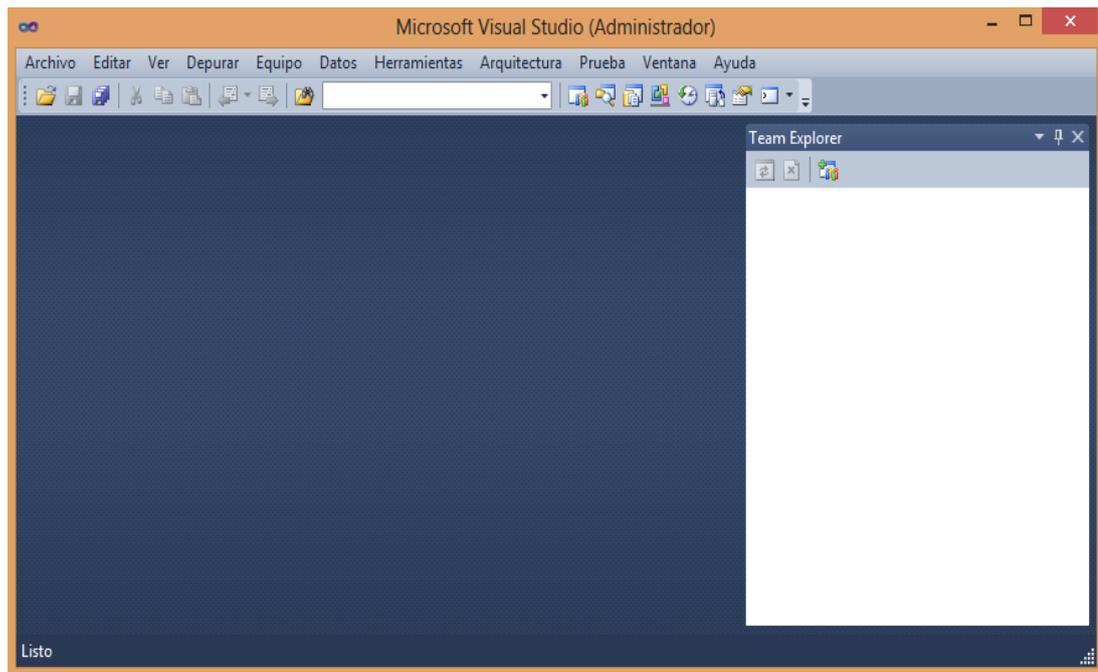


Fig 11 Ventana principal Microsoft Visual Studio

INSTALACION SQL SERVER 2008 R2

1.- Dar clic derecho y elegir la opción INSTALAR O EJECUTAR EL PROGRAMA DESDE LOS MEDIOS.

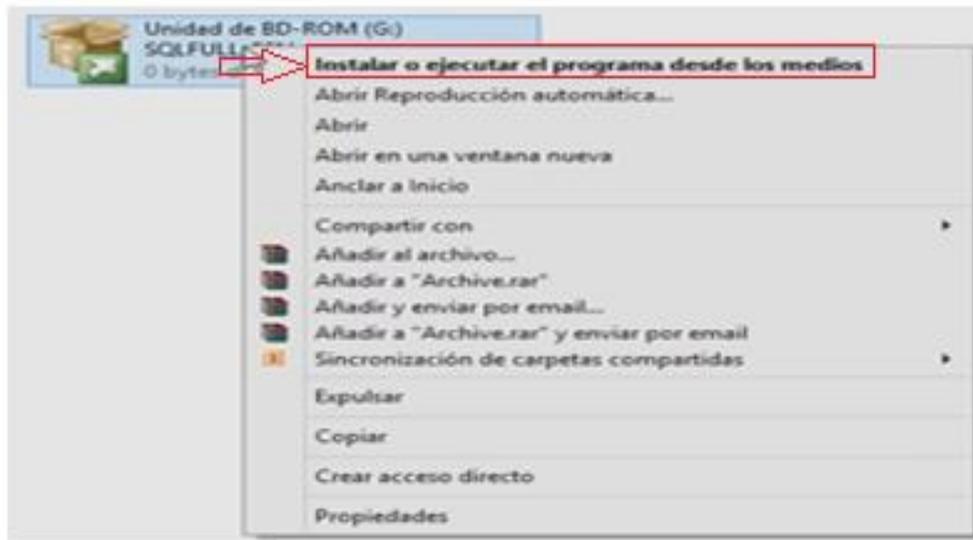


Fig 12 Icono Instalador SQL SERVER 2008

2.- En la parte izquierda de esta pantalla elegir la opción INSTALACION



Fig 13 Centro de Instalación SQL SERVER 2008

3.- Se desplegará una nueva pantalla en la cual se debe dirigirse al lado derecho y dar clic en opción NUEVA INSTALACION O AGREGAR CARACTERISTICAS A UNA INSTALACION EXISTENTE.



Fig 14 Nueva instalación

4.- Se desplegará una nueva ventana en la que debe comprobar que su ordenador haya pasado todas las pruebas y dar clic en la opción ACEPTAR

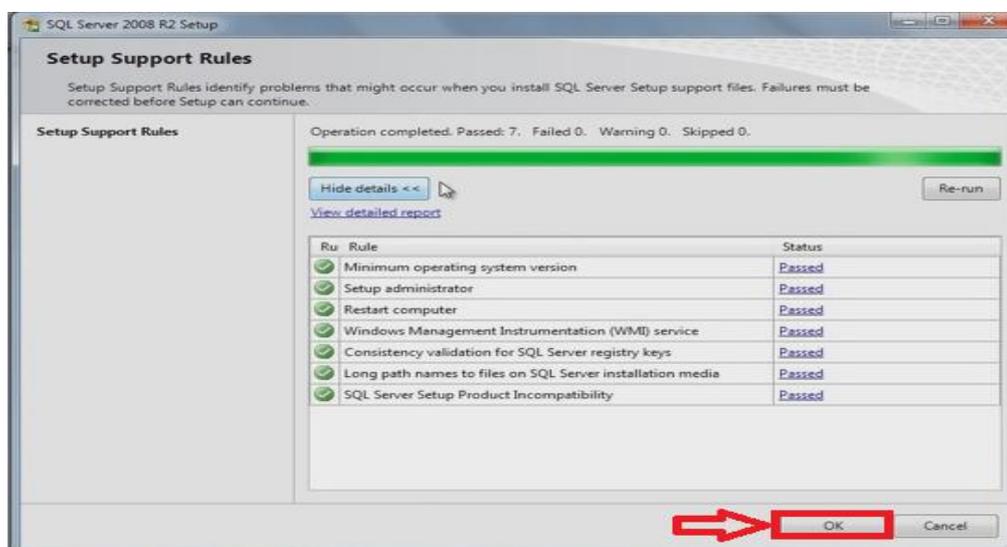


Fig 15 Setup Support Rules

5.- En esta ventana aparecerá el tipo de producto que va a instalar y debe escribir la clave de producto.

6.- Dar clic en Siguiente

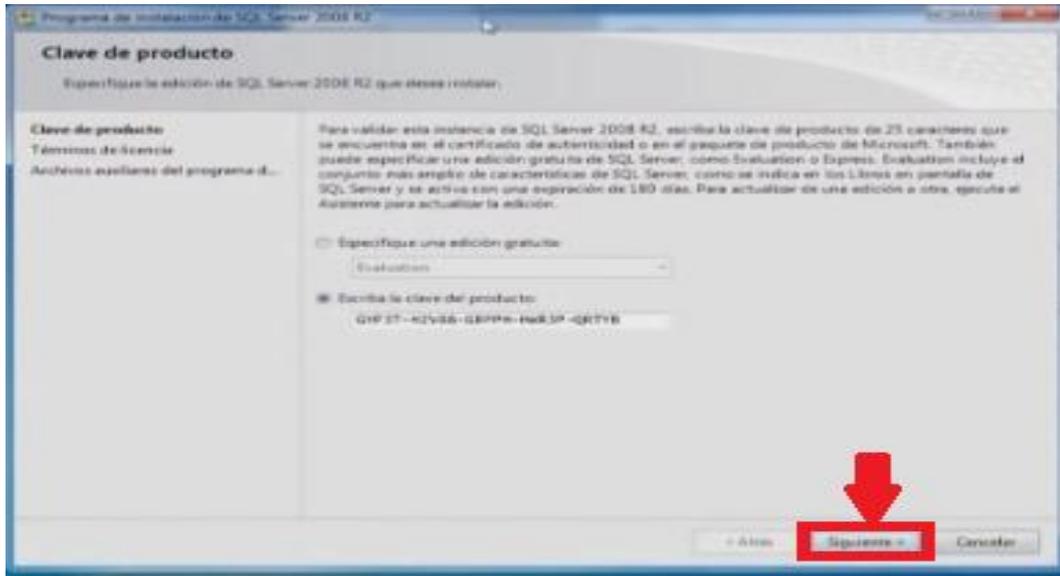


Fig 16 Clave del Producto

7.- Aparecerá una nueva ventana en la que debe Aceptar los términos de licencia y dar clic en Siguiente.



Fig 17 Términos de Licencia

8.- En esta ventana se instalarán los archivos de soporte dar clic en Instalar.

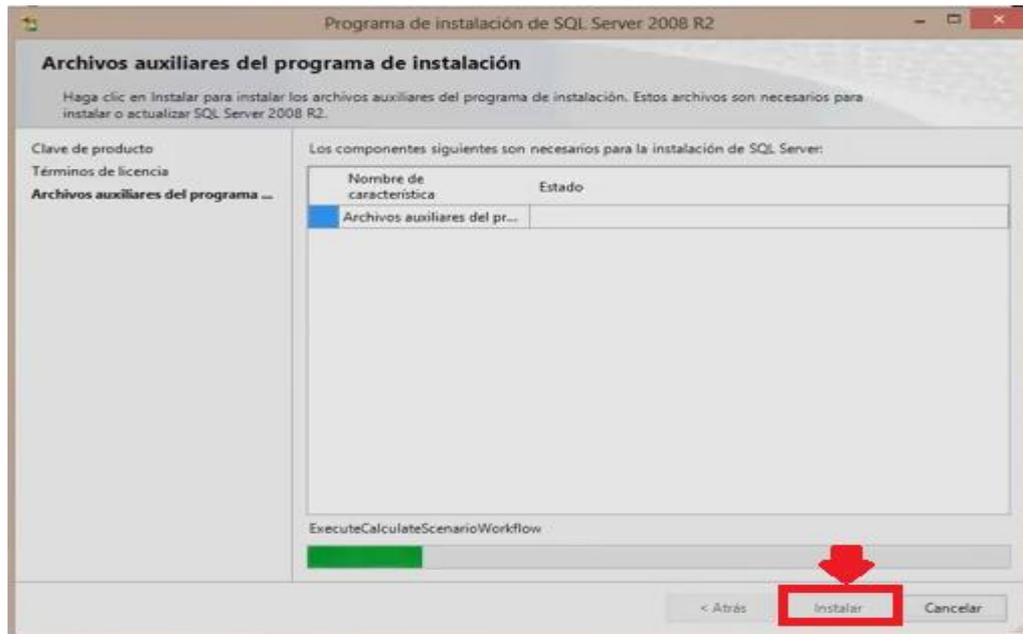


Fig 18 Archivos Auxiliares

9.- Verificar que todos los archivos de soporte hayan pasado las pruebas de soporte y dar clic en Siguiente.

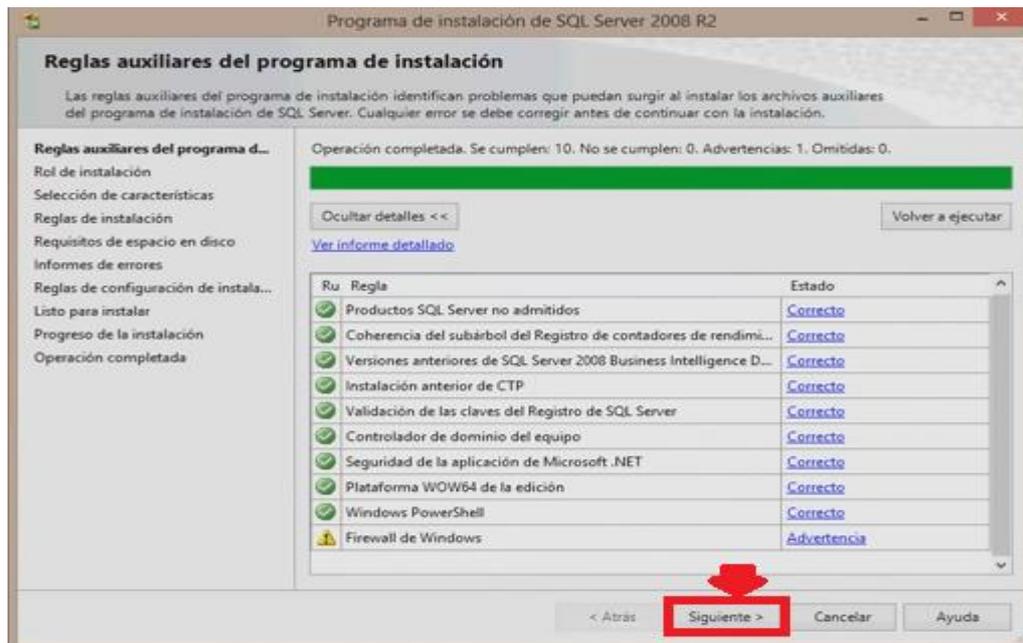


Fig 19 Reglas Auxiliares

10.- Seleccionar la opción SQL Server Instalación de características de SQL Server y dar clic en Siguiente.

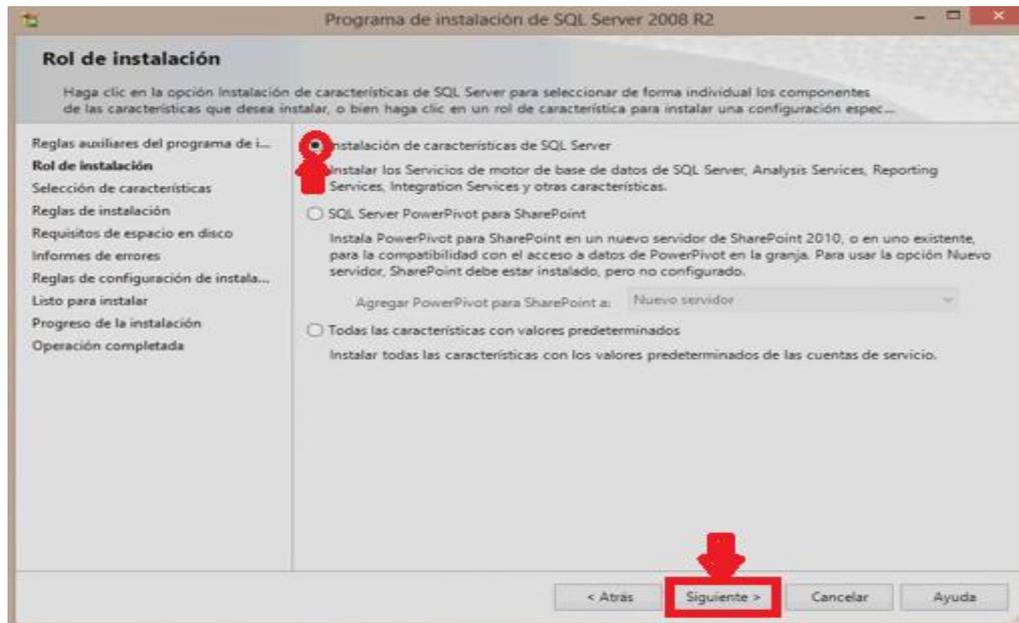


Fig 20 Rol de Instalación

11.- Escoger las características que desee instalar y dar clic en NEXT.

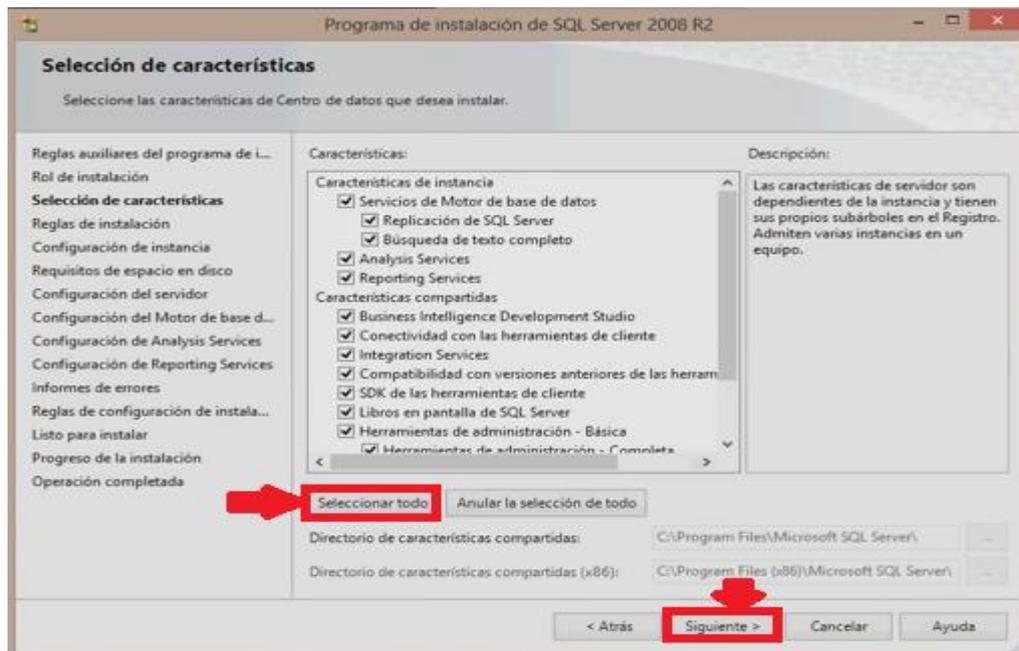


Fig 21 Selección de Características

12.- Muestra en esta pantalla que paso las pruebas y dar clic en Siguiente.

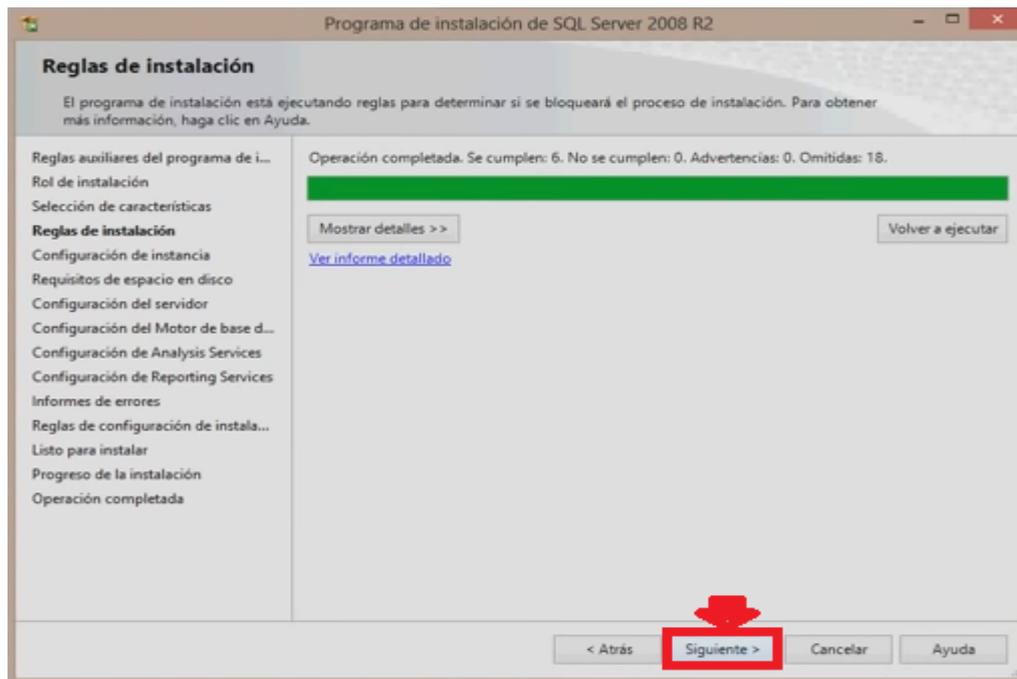


Fig 22 Reglas de Instalación

13.- Escoger la opción Instancia Predeterminada y dar clic en Siguiente.

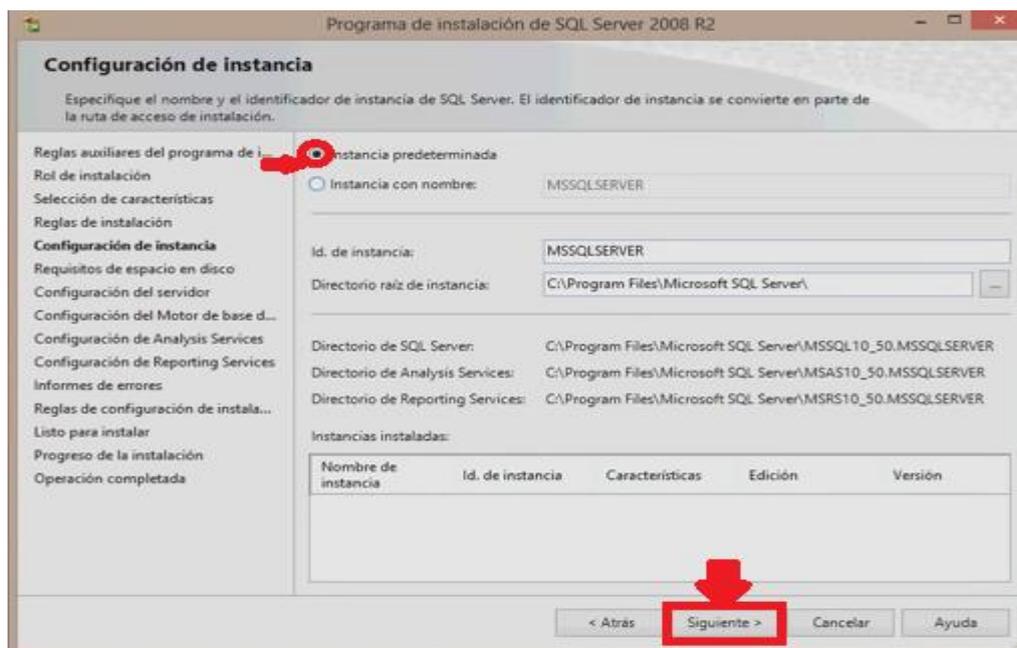


Fig 23 Configuración de Instancia

14.- Muestra el espacio de requerimiento del disco dar clic en Siguiente.

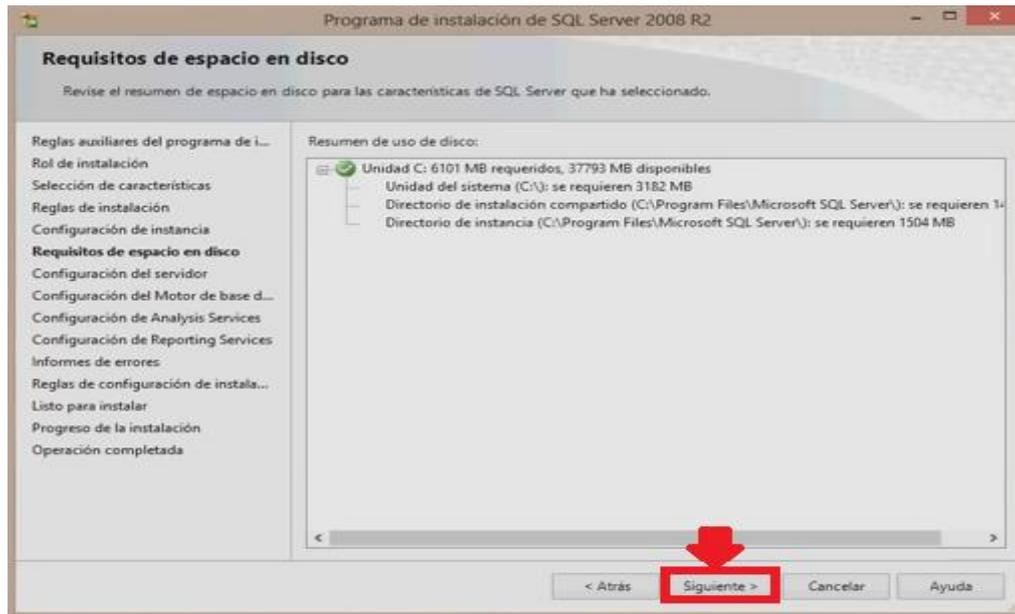


Fig 24 Requisitos de Espacio en Disco

15.- En esta ventana se debe elegir la opción USAR LA MISMA CUENTA PARA TODOS LOS SERVICIOS DE SQL SERVER.

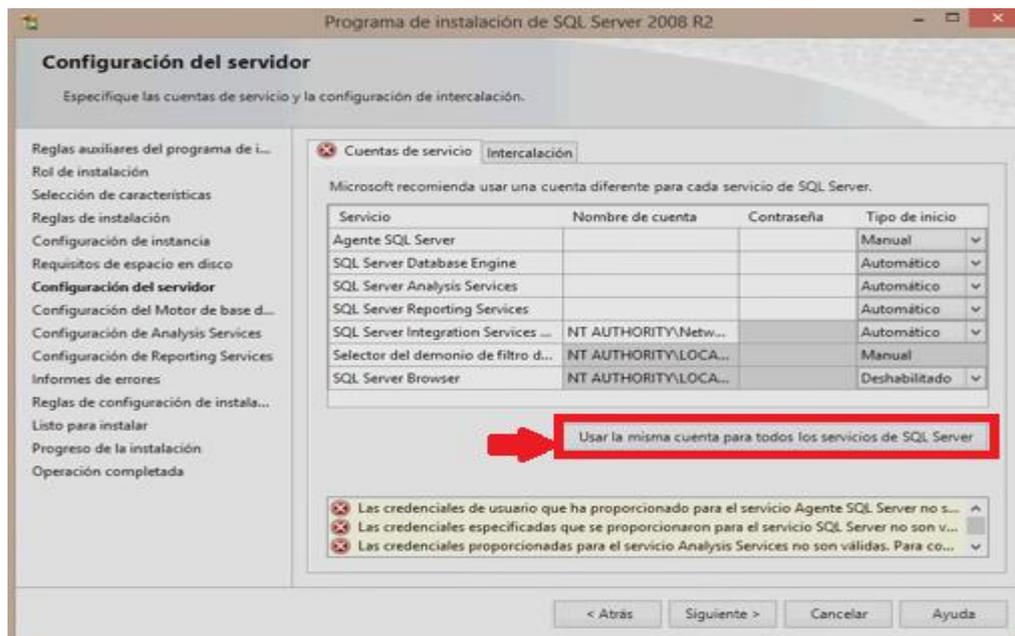


Fig 25 Configurador del Servidor

16.-Elegir la opción NT AUTHORITY SYSTEM y ACEPTAR.

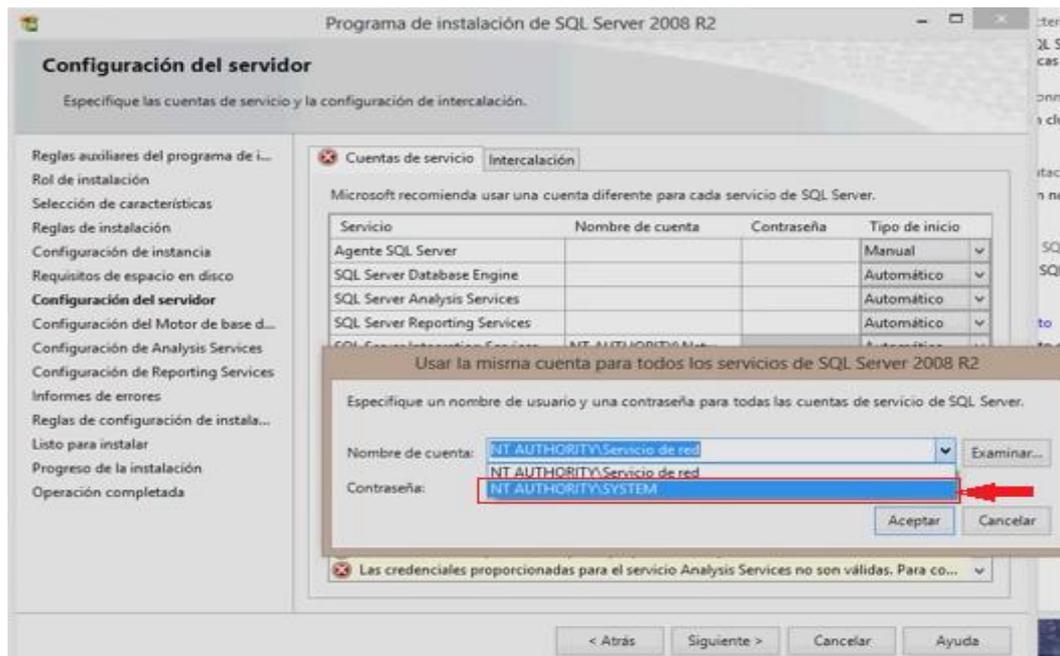


Fig 26 Configurador del Servidor

17.- Dar clic en siguiente

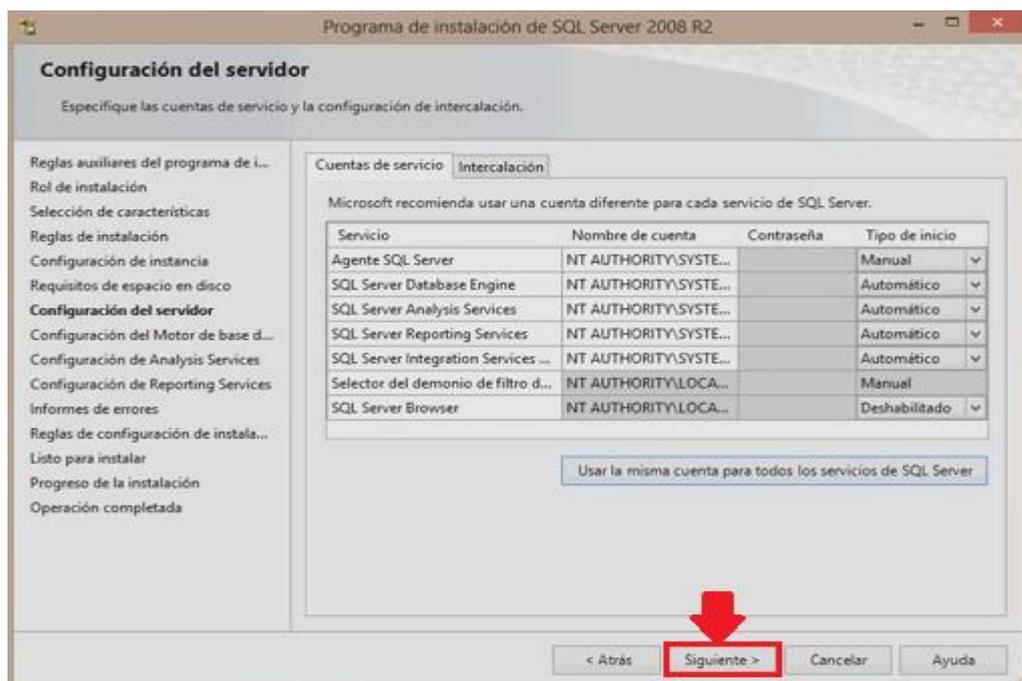


Fig 27 Configurador del Servidor NT Authority

18.- Elegir MODO AUTENTIFICACION DE WINDOWS y dar clic en SIGUIENTE.

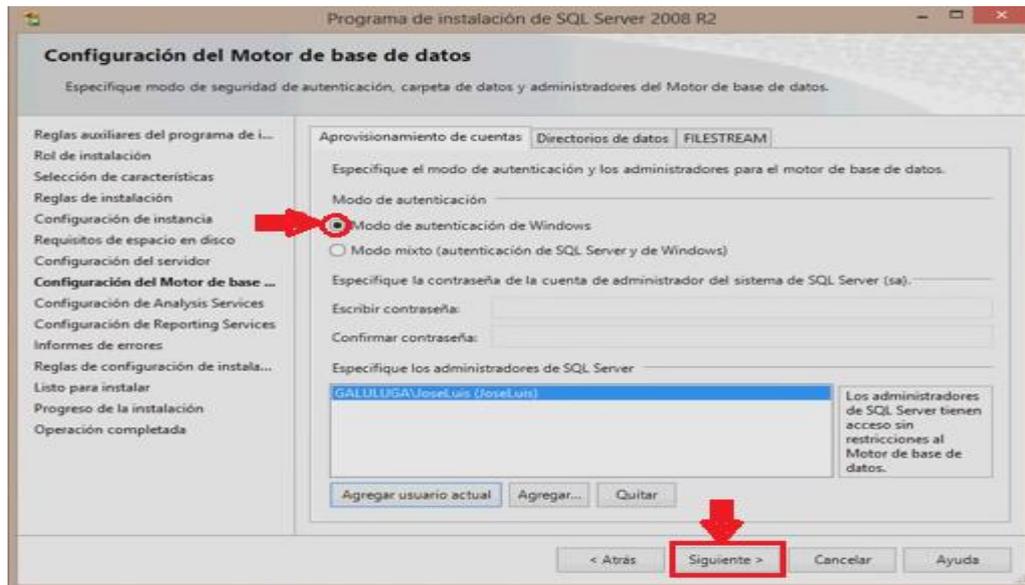


Fig 28 Configuración del Motor de Base de Datos

19.- Dar clic en AGREGAR USUARIO ACTUAL y SIGUIENTE.

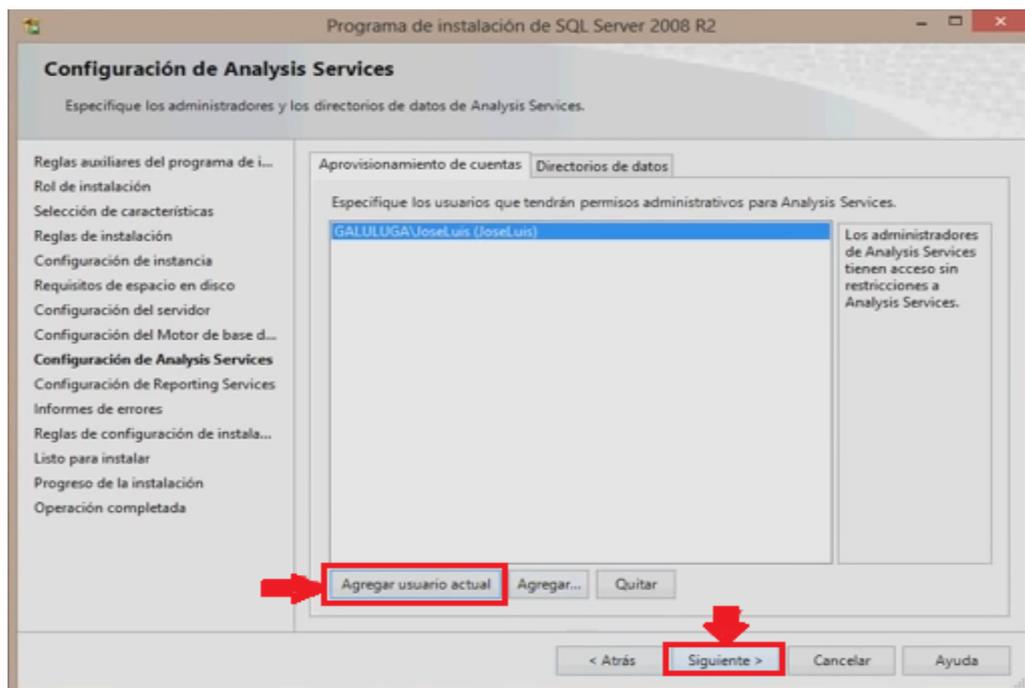


Fig 30 Configuración de Análisis Services

20.- Elegir la opción **INSTALAR LA CONFIGICION PREDETERMINADA DEL MODO NATIVO Y SIGUIENTE.**



Fig 31 Configuración de Reporting Services

21.- Dar clic en el botón **SIGUIENTE.**

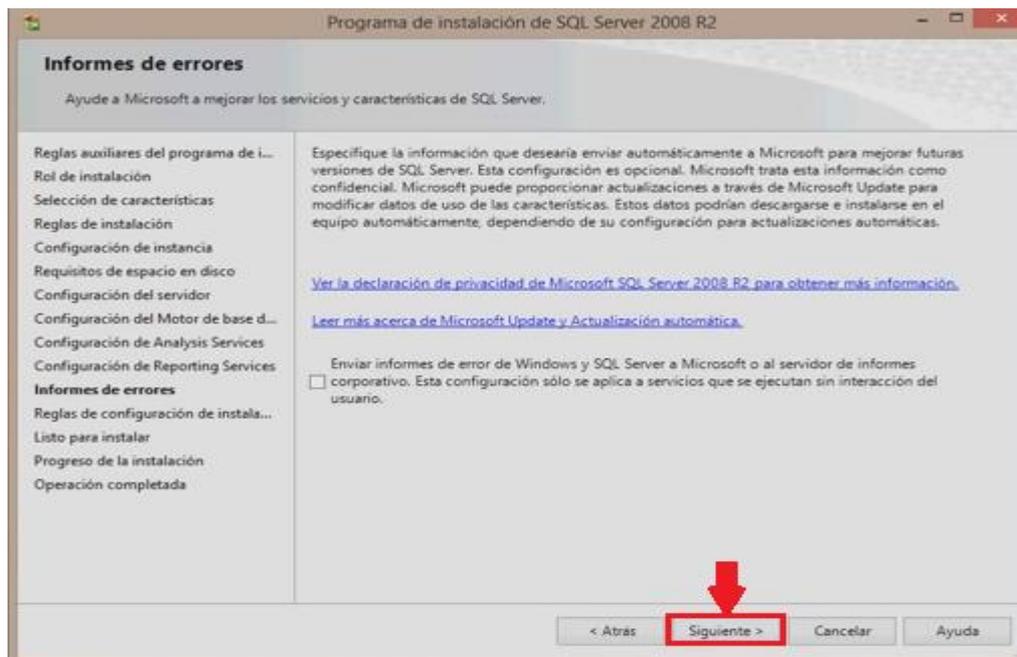


Fig 32 Informes de errores

22.- Dar clic en siguiente

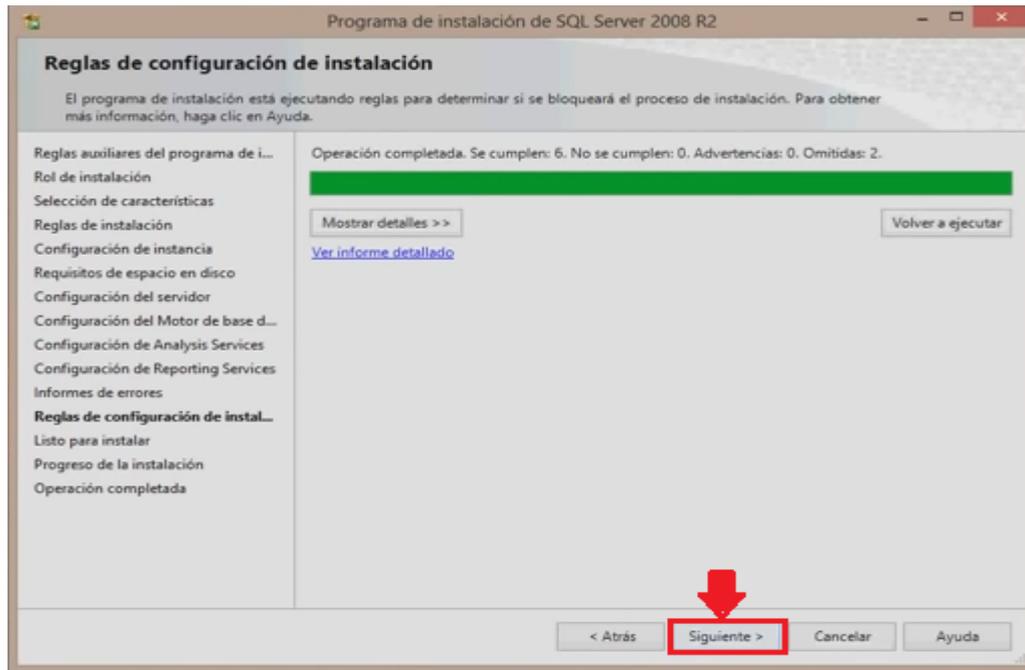


Fig 33 Reglas de ConFgición de Instalación

23.- Dar clic en INSTALAR

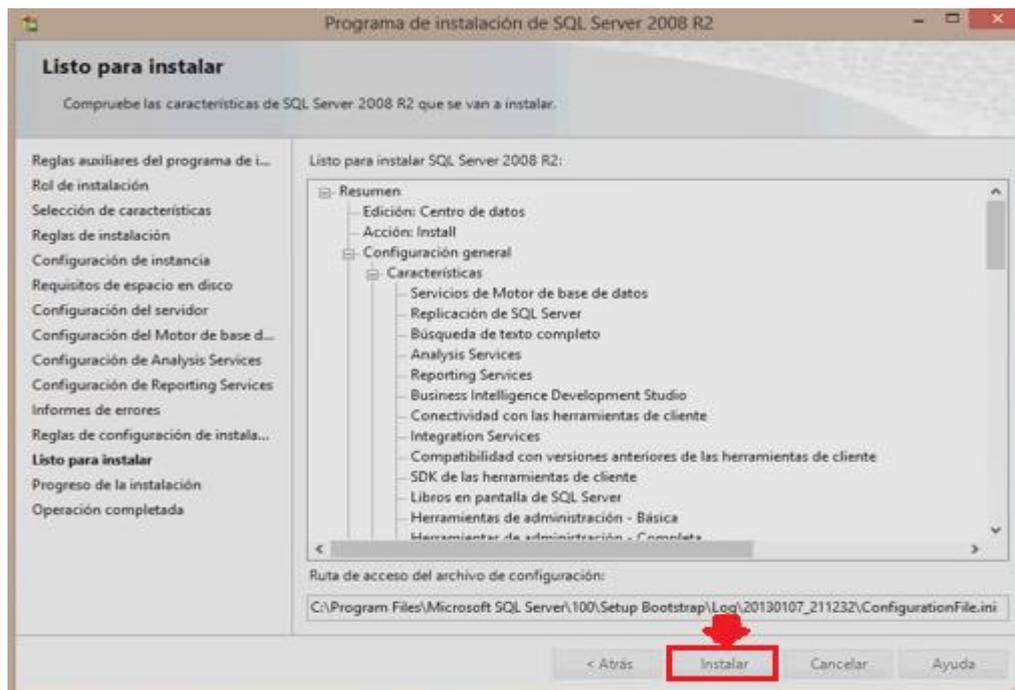


Fig 34 Listo para Instalar

24.- En esta ventana se presentara el progreso de la instalación.

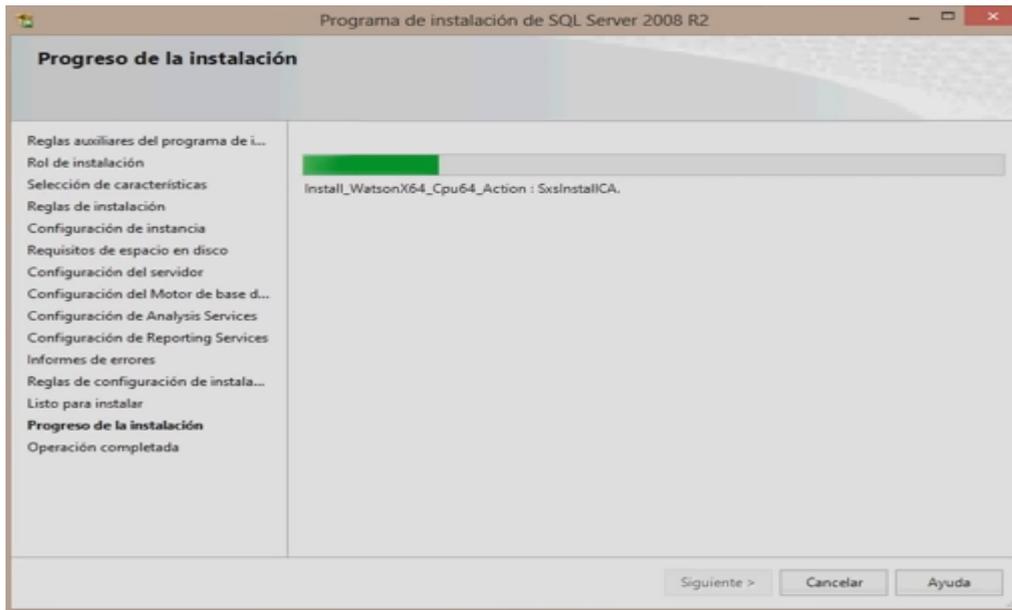


Fig 35 Progreso de la Instalación

25.- En esta ventana se muestra que la instalación ha terminado dar clic en CERRAR.

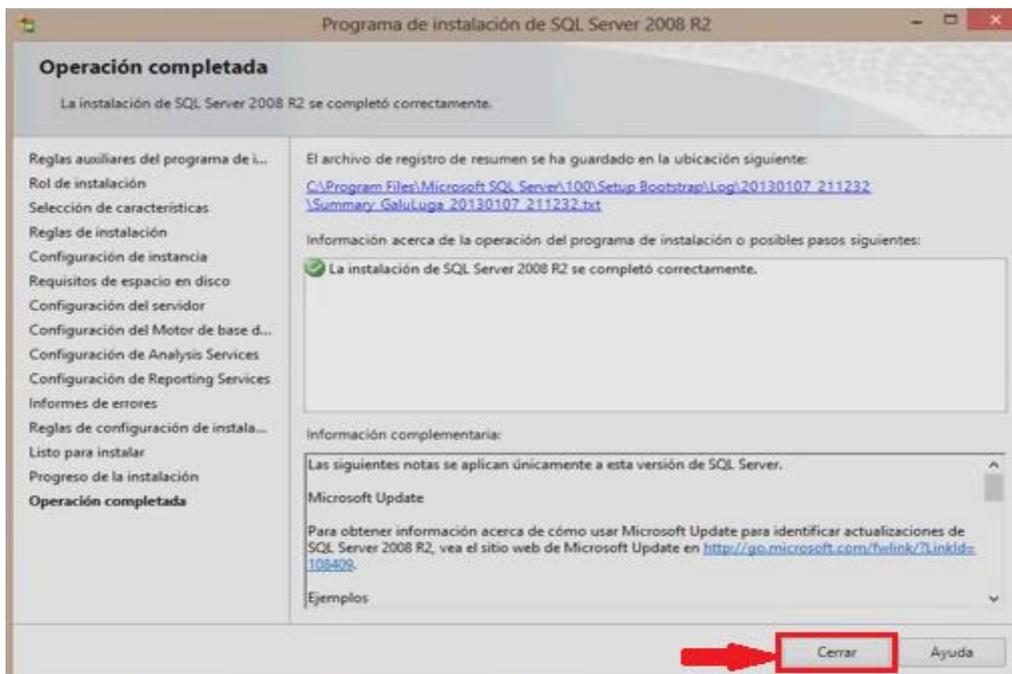


Fig 36 Operación Completada

26.- Dar clic en el icono de SQL SERVER MANAGEMENT.



Fig 37 Icono SQL SERVER MANAGEMENT

27.- En esta ventana en el NOMBRE DEL SERVIDOR se debe poner (local) y dar clic en el botón CONECTAR.

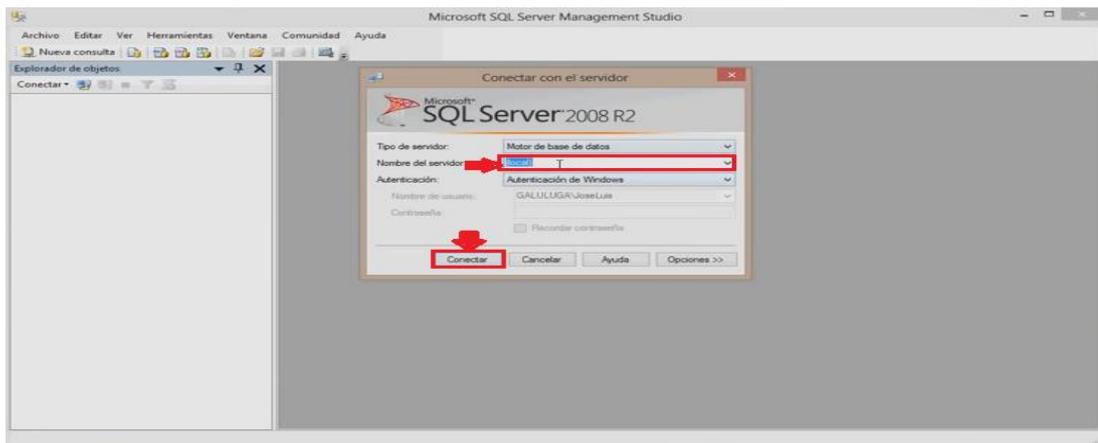


Fig 38 Conectar SQL SERVER 2008

28.- SQL SERVER está listo para utilizar.

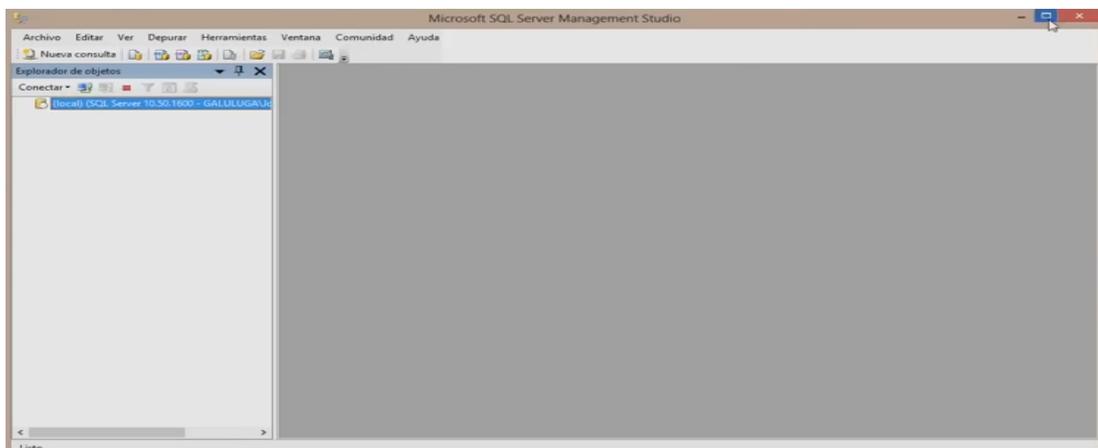


Fig 39 Ventana Principal SQL SERVER 2008

A.03

MANUAL DE USUARIO



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

OPTIMIZACIÓN DE LOS PROCESOS DE ACTIVOS FIJOS MEDIANTE UNA
APLICACIÓN INFORMÁTICA WEB PARA LA EMPRESA ABS COMPUTER S.
A EN LA CIUDAD DE QUITO

Proyecto de investigación previo a la obtención del título de

Tecnólogo en Análisis de Sistemas

MANUAL DE USUARIO

Autor: Cauja Pilataxi Guillermo Patricio

Tutor: Ing. Marco Obando

Quito, Marzo 2015

Figuras Manual de Usuario

Fig 1 Ventana de Inicio.....	104
Fig 2 Login.....	104
Fig 3 Validación ontraseña	105
Fig 4 Ventana Principal del Sistema.....	105
Fig 5 Registro Usuario	106
Fig 6 Tipo de Activo	106
Fig 7 Ubicación Activo	107
Fig 8 Activo.....	107
Fig 9 Tiempo Útil.....	108
Fig 10 Custodio	108
Fig 11 Estado.....	109
Fig 12 Reporte de Activos	109
Fig 13 Reporte Depreciación	110
Fig 14 Salida de Activos	110

Administrador

1.- En esta pantalla nos muestra los logos de la empresa, en la parte superior se encuentra el Botón de Ingreso al Sistema.



Fig 1 Ventana de Inicio

2.- En esta pantalla muestra dos cajas de texto en el cual se debe poner el usuario y la contraseña válidos al perfil de cada uno, luego presionar el botón ingresar.

The image shows a login form titled 'LOGIN'. It contains two text input fields: 'USUARIO' and 'CONTRASEÑA'. Below the input fields is a large green button labeled 'Ingresar'.

Fig 2 Login

3.- El usuario y la contraseña deben ser válidos o en caso de solo ingresar el usuario y no la contraseña por varias ocasiones le saldrá un mensaje indicando que excedido el número de intentos lo que procederá a bloquear al usuario.

Entrada al Sistema

Usuario

Contraseña

Ha excedido el número de intentos permitidos

Fig 3 Validación Contraseña

4.- El administrador contara con acceso al sistema por completo y a todos los menús tal como lo muestra esta pantalla.

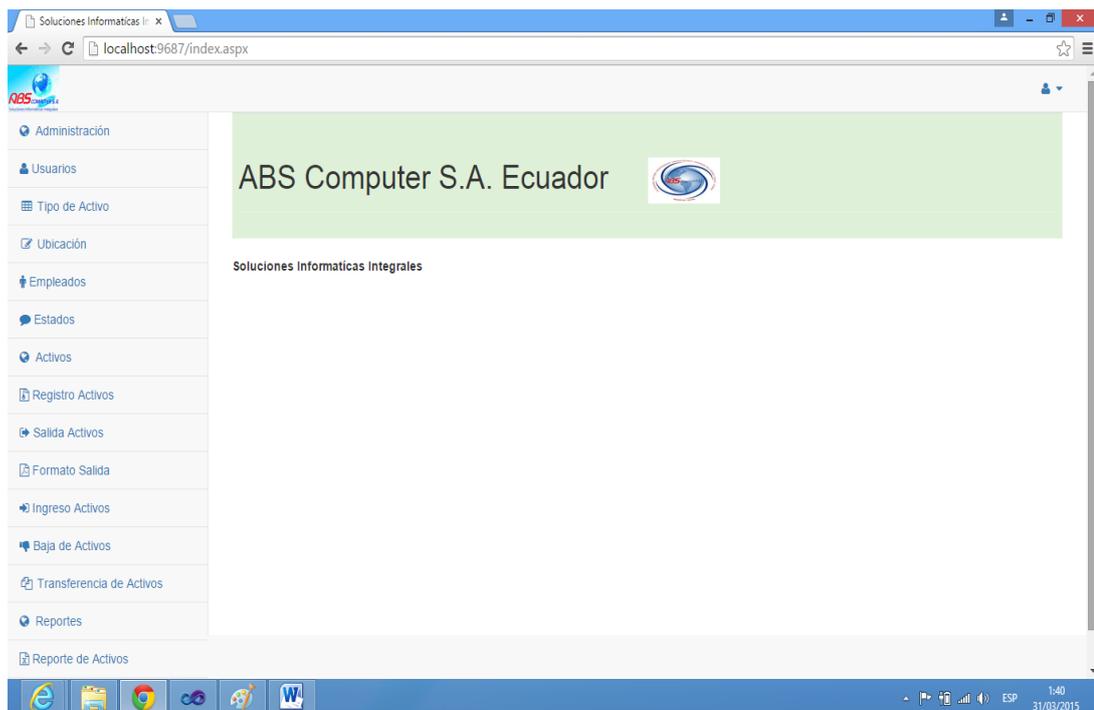


Fig 4 Ventana Principal del Sistema

5.- Si escoge la opción usuario del menú nos aparecerá esta pantalla en la cual deberá llenar con el usuario, nombre, apellido, perfil, contraseña del usuario a registrar acompañando el proceso con el botón Ingresar. En la parte inferior aparece un cuadro con los usuarios ya registrados da opción a editar la información de los usuarios.



Fig 5 Registro Usuario

6.-Si escoge la opción Tipo de Activo aparecerán dos cuadros de texto en el que debe llenar el nombre el nombre, identificador, tiempo de vida útil y dar clic en el botón ingresar. Podrá modificar Tipos de activos ya ingresado con la opción editar que aparece en la parte inferior derecha de la pantalla.

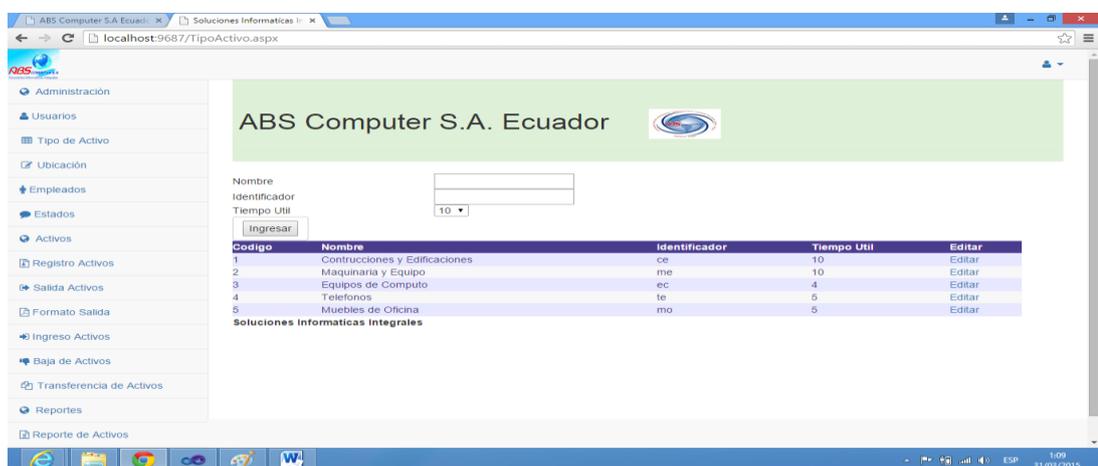


Fig 6 Tipo de Activo

7.- Si escoge la opción Ubicación aparecerán cuatro cuadros de texto en la cual debe llenar Nombre Dirección, Teléfono y si es la ubicación principal donde se encuentra el activo luego dar clic en ingresar.

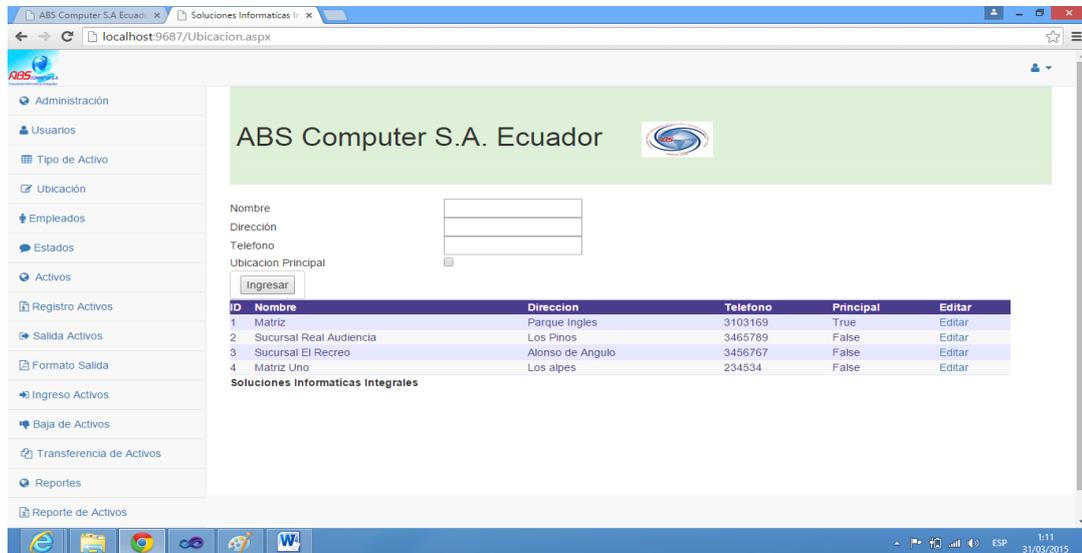


Fig 7 Ubicación Activo

8.- Si escoge la opción Activo aparecerán ocho cuadros de texto en la cual debe llenar Ubicación, Área, Tipo de Activo, Nombre, Modelo, Serial, Valor y Fecha de Ingreso luego dar clic en ingresar.



Fig 8 Activo

9.- Si escoge la opción Custodio aparecerán dos cuadros de texto en la cual debe llenar Nombre de la persona jefe de área y luego el asignara a un custodio.

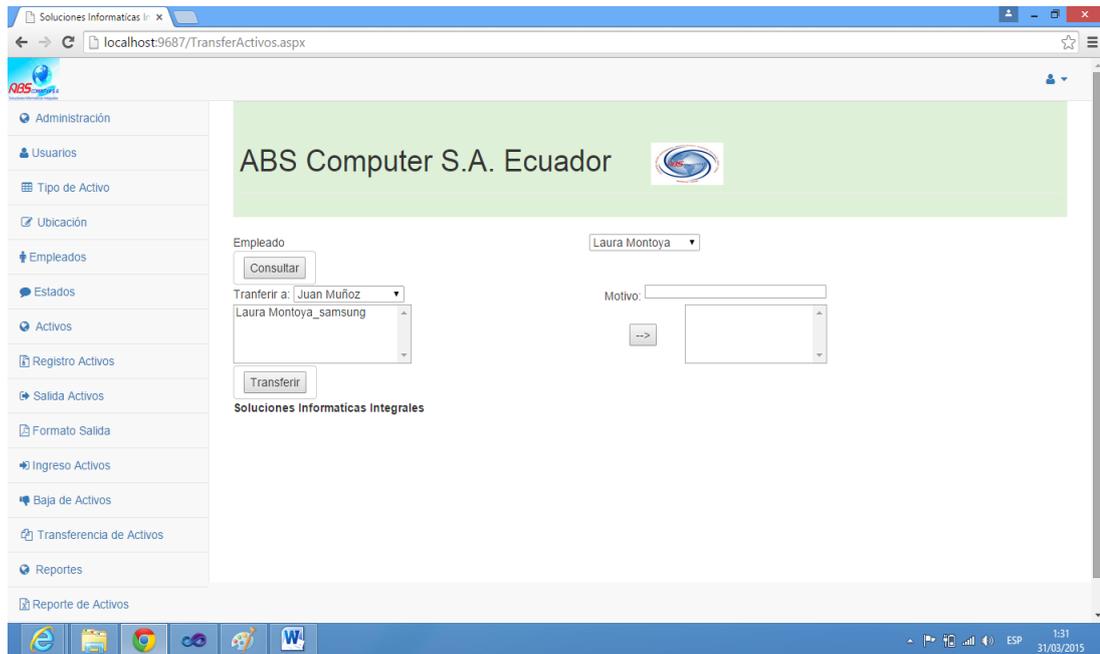


Fig 9 Custodio

10.- Si escoge la opción Estado aparecerá un cuadro de texto en la cual debe llenar los estados de un activo.

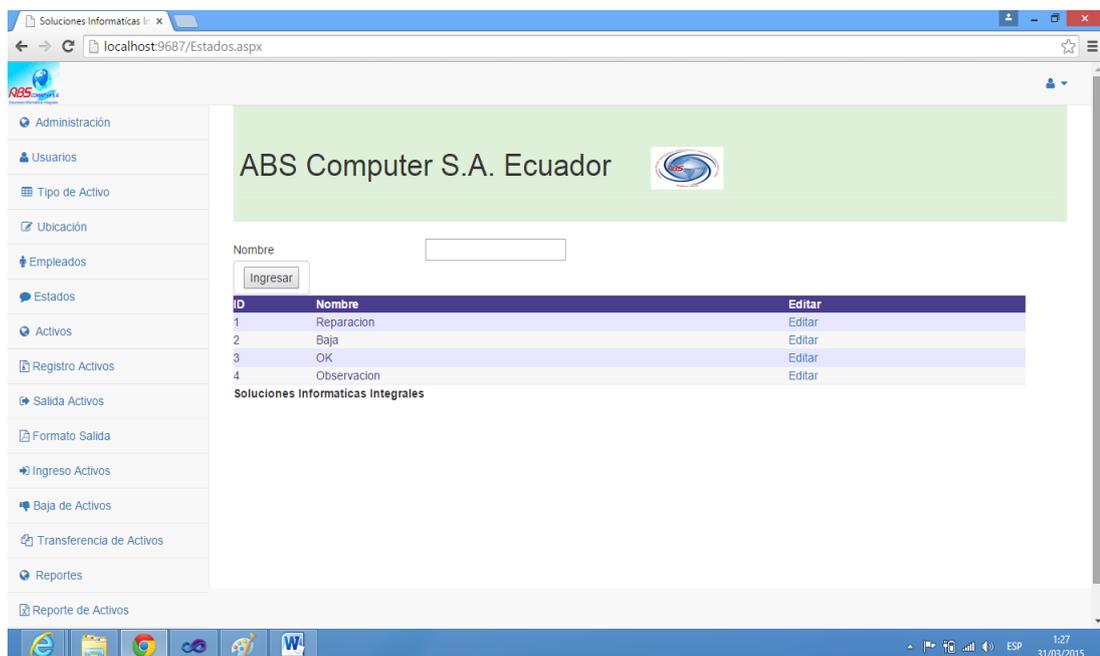


Fig 10 Estado

11.- Si escoge la opción Reporte de Activos se generara la consulta de los Activos al presionar el botón Consultar se procede a la consulta de los Activos.

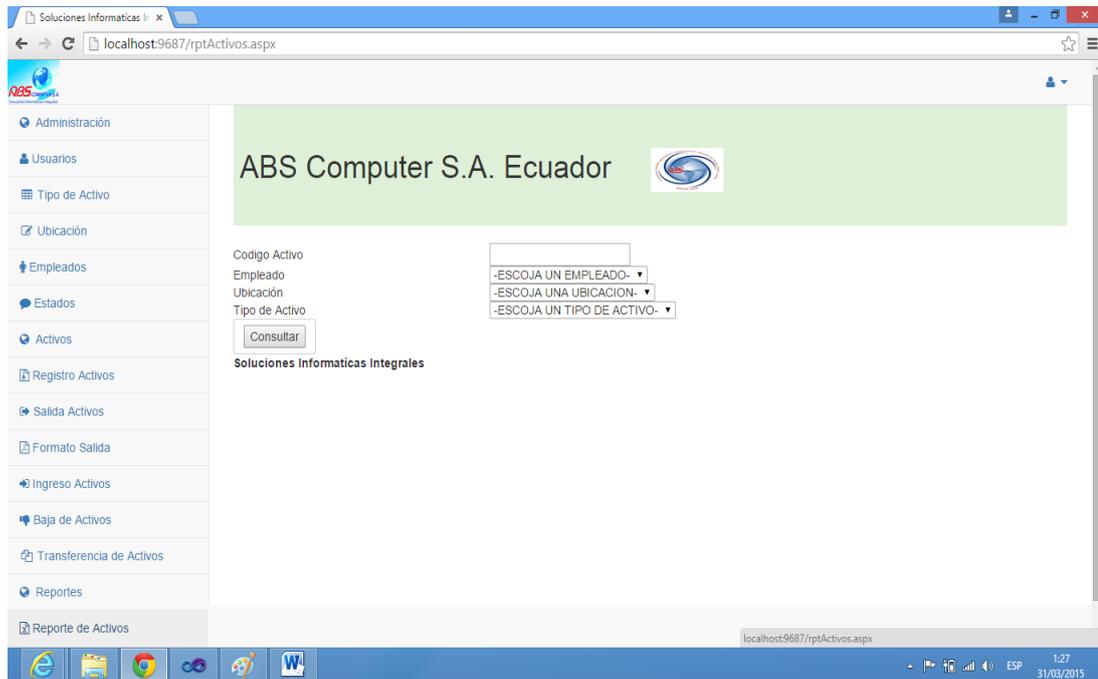


Fig 11 Reporte de Activos

12.- Si escoge la opción Reporte Depreciación se generara la consulta de la depreciación de los Activos al presionar el botón Consultar.

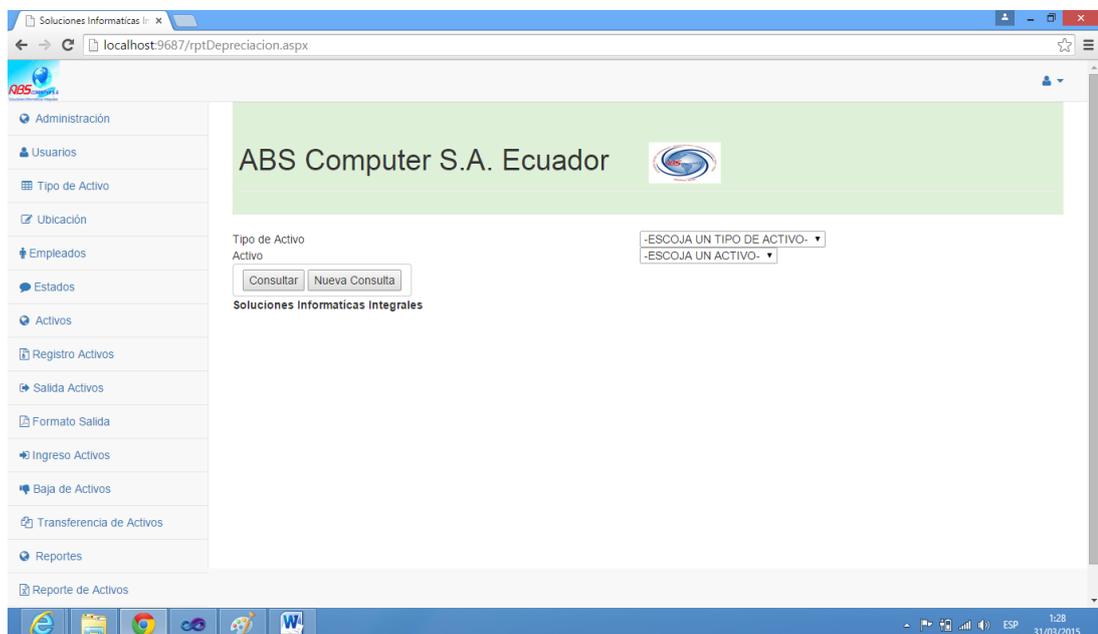


Fig 12 Reporte Depreciación

13.- Si escoge la opción Salida de Activo se registrara la salida de los activos dados de baja presionando el botón Registrar. Se deberán llenar los siguientes campos de texto Tipo de Activo, Detalle, Código Activo, Modelo Activo, Serie Activo, Custodio y Nombre Autorizador.

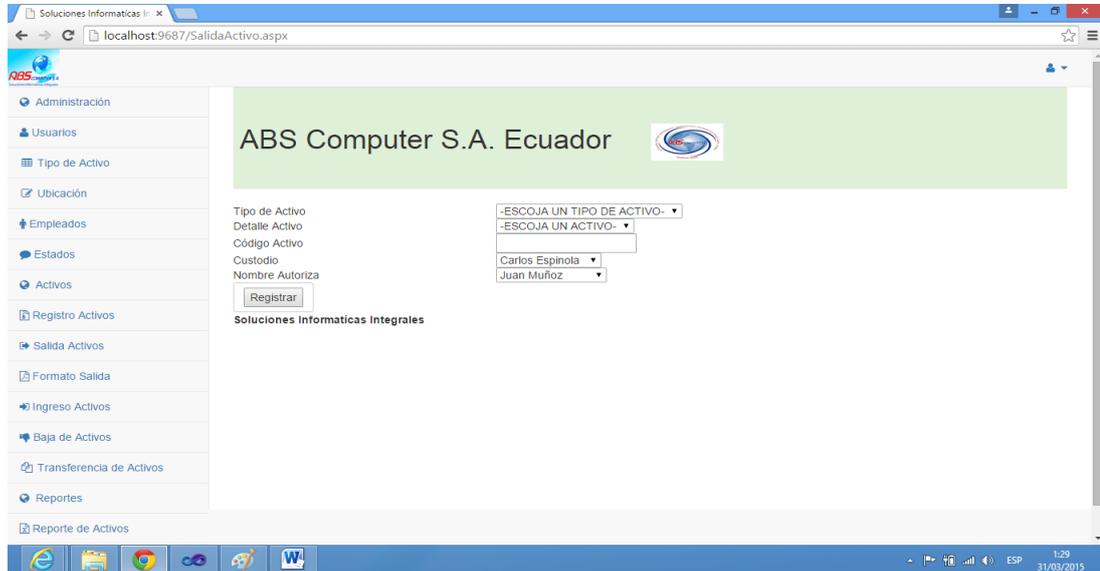


Fig 13 Salida de Activos

A.04

MANUAL

TÉCNICO



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

OPTIMIZACIÓN DE LOS PROCESOS DE ACTIVOS FIJOS MEDIANTE UNA
APLICACIÓN INFORMÁTICA WEB PARA LA EMPRESA ABS COMPUTER S.
A EN LA CIUDAD DE QUITO

Proyecto de investigación previo a la obtención del título de
Tecnólogo en Análisis de Sistemas

MANUAL TÉCNICO

Autor: Cauja Pilataxi Guillermo Patricio

Tutor: Ing. Marco Obando

Quito, Abril 2015

SCRIPT BASE DE DATOS

```
GO
ALTER DATABASE [DB_ACTIVOS] SET COMPATIBILITY_LEVEL = 110
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [DB_ACTIVOS].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [DB_ACTIVOS] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET ANSI_NULLS OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET ANSI_PADDING OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET ARITHABORT OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [DB_ACTIVOS] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [DB_ACTIVOS] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [DB_ACTIVOS] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET DISABLE_BROKER
GO
ALTER DATABASE [DB_ACTIVOS] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [DB_ACTIVOS] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET RECOVERY FULL
GO
ALTER DATABASE [DB_ACTIVOS] SET MULTI_USER
```

```
GO
ALTER DATABASE [DB_ACTIVOS] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [DB_ACTIVOS] SET DB_CHAINING OFF
GO
ALTER DATABASE [DB_ACTIVOS] SET FILESTREAM( NON_TRANSACTED_ACCESS =
OFF )
GO
ALTER DATABASE [DB_ACTIVOS] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
EXEC sys.sp_db_vardecimal_storage_format N'DB_ACTIVOS', N'ON'
GO
USE [DB_ACTIVOS]
GO
/***** Object: StoredProcedure [dbo].[BajaActivo]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BajaActivo]
```

```
@activo int,
```

```
@motivo varchar(MAX)
```

```
AS
```

```
INSERT INTO [TB_BAJAS]
```

```
    ([ID_ACTIVOS]
```

```
    ,[FECHA_BAJA]
```

```
    ,[MOTIVO])
```

```
VALUES
```

```
    (@activo,
```

```
    getdate(),
```

```
    @motivo)
```

```
UPDATE TB_ACTIVOS
```

```
SET ID_ESTADO = 3
```

```
WHERE ID_ACTIVOS = @activo
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[BloqueaUsuario] Script Date: 16/03/2015 10:30:24
*****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[BloqueaUsuario]
```

```
@usuario varchar(12)
```

```
AS
```

```
declare @intentos int
```

```
select @intentos = ISNULL(INTENTOS,0)
from TB_USUARIOS
where ID_USUARIO = @usuario
```

```
select @intentos
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[ConsultaActivo] Script Date: 16/03/2015 10:30:24
*****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[ConsultaActivo]
```

```
@codigo varchar(20),
```

```
@empleado int,
```

```
@ubicacion int,
```

```
@tipo int
```

```
AS
```

```
declare @sql varchar(max)
```

```
set @sql = 'SELECT
a.[ID_ACTIVO],[CODIGO_ACTIVO],NOMBRE_TIPO,NOMBRE_UBICACION,[NOMBRE_EM
PLEADO] + " " + [APELLIDO_EMPLEADO]
EMPLEADO,[NOMBRE_ACTIVO],[MODELO],[SERIAL], '
+ '[VALOR],a.[FECHA_INGRESO],NOM_ESTADO '
```

```
+ 'FROM [TB_ACTIVOS] a '  
  
+ 'LEFT JOIN [TR_MOVIMIENTO_ACTIVIVO] b on a.ID_ACTIVO = b.ID_ACTIVO AND  
a.ID_TIPO = b.ID_TIPO '  
  
+ 'INNER JOIN TB_UBICACION c on a.ID_UBICACION = c.ID_UBICACION '  
  
+ 'INNER JOIN TB_TIPO_ACTIVIVO d on a.ID_TIPO = d.ID_TIPO '  
  
+ 'INNER JOIN TB_ESTADO e on a.ID_ESTADO = e.ID_ESTADO '  
  
+ 'LEFT JOIN TR_ASIGNA_ACTIVIVO f on a.ID_ACTIVO = f.ID_ACTIVO '  
  
+ 'LEFT JOIN TB_EMPLEADO g on f.ID_EMPLEADO = g.ID_EMPLEADO '
```

```
if @codigo != ''
```

```
begin
```

```
set @sql = @sql + 'WHERE [CODIGO_ACTIVIVO] = '' + @codigo + '' '
```

```
if @ubicacion != 0
```

```
set @sql = @sql + ' AND a.ID_UBICACION = ' + cast(@ubicacion as varchar)
```

```
if @empleado != 0
```

```
set @sql = @sql + ' AND f.[ID_EMPLEADO] = ' + cast(@empleado as varchar)
```

```
if @tipo != 0
```

```
set @sql = @sql + ' AND a.ID_TIPO = ' + cast(@tipo as varchar)
```

```
end
```

```
else
```

```
begin
```

```
if @ubicacion != 0

begin

    set @sql = @sql + 'WHERE a.ID_UBICACION = ' + cast(@ubicacion as varchar)

        if @empleado != 0

            set @sql = @sql + ' AND f.[ID_EMPLEADO] = ' + cast(@empleado as varchar)

                if @tipo != 0

                    set @sql = @sql + ' AND a.ID_TIPO = ' + cast(@tipo as varchar)

end

else

begin

    if @empleado != 0

        begin

            set @sql = @sql + 'WHERE f.[ID_EMPLEADO] = ' + cast(@empleado as varchar)

                if @tipo != 0

                    set @sql = @sql + ' AND a.ID_TIPO = ' + cast(@tipo as varchar)

end

else

begin

    if @tipo != 0
```

```
set @sql = @sql + 'WHERE a.ID_TIPO = ' + cast(@tipo as varchar)

end

end

end

set @sql = @sql + 'GROUP BY
a.[ID_ACTIVO],[CODIGO_ACTIVO],NOMBRE_TIPO,NOMBRE_UBICACION,[NOMBRE_EM
PLEADO],[APELLIDO_EMPLEADO],[NOMBRE_ACTIVO],[MODELO],[SERIAL], '

+ '[VALOR],a.[FECHA_INGRESO],NOM_ESTADO'

exec (@sql)

GO
/***** Object: StoredProcedure [dbo].[ExisteUsuario]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[ExisteUsuario]

@usuario varchar(12)

AS

if exists(select 1
          from TB_USUARIOS
          where ID_USUARIO = @usuario)

select 1

else

select 0

GO
/***** Object: StoredProcedure [dbo].[Ingresa_Activo]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Ingresa_Activo]

@ubicacion int,

@tipo int,
```

@nombre varchar(100),

@modelo varchar(10),

@serial varchar(25),

@valor float,

@fecha datetime,

@area int

AS

declare @codigo varchar(20)

declare @ident varchar(5)

declare @secuencial varchar(5)

declare @empleado int

declare @activo int

```
select @empleado = ID_EMPLEADO
```

```
from [TB_EMPLEADO]
```

```
where ID_AREA = @area
```

```
and PRINCIPAL = 1
```

```
select @secuencial = isnull(cast(SUBSTRING(max(CODIGO_ACTIVO),3  
,len(max(CODIGO_ACTIVO))) as int),0) + 1 ---  
isnull(cast(SUBSTRING(max(CODIGO_ACTIVO),charindex('-',max(CODIGO_ACTIVO)) + 1  
,len(max(CODIGO_ACTIVO))) as int),0) + 1
```

```
from TB_ACTIVOS
```

```
where ID_TIPO = @tipo
```

```
select @ident = IDENTIFICADOR
```

```
from TB_TIPO_ACTIVO
```

```
where ID_TIPO = @tipo
```

```
set @codigo = @ident + replicate('0',4-len(@secuencial)) + LTRIM(RTRIM(STR(@secuencial)))
```

```
INSERT INTO [TB_ACTIVOS]
```

```
    ([ID_UBICACION]
```

```
    ,[ID_TIPO]
```

```
    ,[CODIGO_ACTIVIVO]
```

```
    ,[NOMBRE_ACTIVIVO]
```

```
    ,[FECHA_INGRESO]
```

```
    ,[MODELO]
```

```
    ,[SERIAL]
```

```
    ,[VALOR]
```

```
    ,[ID_ESTADO]
```

,[ID_AREA])

VALUES

(@ubicacion

,@tipo

,@codigo

,@nombre

,@fecha

,@modelo

,@serial

,@valor

,1

,@area)

set @activo = @@IDENTITY

INSERT INTO [TR_ACTIVO_ESTADO]

([ID_ACTIVO]

,[ID_ESTADO])

VALUES

(@activo

,1)

INSERT INTO [TR_ASIGNA_ACTIVOS]

([ID_ACTIVOS]

,[ID_EMPLEADO]

,[FECHA_ASIGNA])

VALUES

(@activo

,@empleado

,GETDATE())

GO

/***** Object: StoredProcedure [dbo].[IngresoActivo] Script Date: 16/03/2015 10:30:24 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[IngresoActivo]

@tipoactivo int,

@activo int,

@estado int,

@observacion varchar(max)

AS

```
UPDATE [TR_MOVIMIENTO_ACTIVADO]
SET [FEC_ENTREGA] = GETDATE(),
[ID_ESTADO] = @estado,
[OBSERVACION] = @observacion
WHERE [ID_TIPO] = @tipoactivo
AND [ID_ACTIVADO] = @activo
AND [FEC_ENTREGA] IS NULL
GO
/***** Object: StoredProcedure [dbo].[InsertaCustodio] Script Date: 16/03/2015 10:30:24
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InsertaCustodio]
```

```
@nombre varchar(100),
```

```
@estado int
```

AS

```
INSERT INTO [TB_CUSTODIO]
([NOMBRE_CUSTODIO]
,[FEC_INGRESO]
,[ID_CARGO])
VALUES
(@nombre,getdate(),@estado)
```

```
GO
/***** Object: StoredProcedure [dbo].[InsertPerfil] Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InsertPerfil]
```

```
(
    @NOMBRE_PERFIL varchar(50)
)
```

AS

```
SET NOCOUNT OFF;
INSERT INTO [TB_PERFIL] ([NOMBRE_PERFIL]) VALUES (@NOMBRE_PERFIL)
GO
```

```
/***** Object: StoredProcedure [dbo].[InsertTipoActivo] Script Date: 16/03/2015 10:30:24
*****/
```

```
SET ANSI_NULLS ON
```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InsertTipoActivo]
(
    @NOMBRE_TIPO varchar(50),
    @IDENTIFICADOR varchar(5)
)
AS
    SET NOCOUNT OFF;
INSERT INTO [TB_TIPO_ACTIVOS] ([NOMBRE_TIPO], [IDENTIFICADOR]) VALUES
(@NOMBRE_TIPO, @IDENTIFICADOR)
GO
/***** Object: StoredProcedure [dbo].[InsertUbicacion]  Script Date: 16/03/2015 10:30:24
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InsertUbicacion]
(
    @NOMBRE_UBICACION varchar(100),
    @DIRECCION_UBICACION varchar(100),
    @TELEFONO_UBICACION varchar(15),
    @PRINCIPAL bit
)
AS
    SET NOCOUNT OFF;
INSERT INTO [dbo].[TB_UBICACION] ([NOMBRE_UBICACION],
[DIRECCION_UBICACION], [TELEFONO_UBICACION], [PRINCIPAL]) VALUES
(@NOMBRE_UBICACION, @DIRECCION_UBICACION, @TELEFONO_UBICACION,
@PRINCIPAL)
GO
/***** Object: StoredProcedure [dbo].[InsertUser]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InsertUser]
(
    @ID_USUARIO varchar(12),
    @ID_PERFIL int,
    @NOMBRE_USUARIO varchar(50),
    @APELLIDO_USUARIO varchar(50),
    @CLAVE_USUARIO varchar(10)
)
AS
    SET NOCOUNT OFF;
INSERT INTO [dbo].[TB_USUARIOS] ([ID_USUARIO], [ID_PERFIL], [NOMBRE_USUARIO],
[APELLIDO_USUARIO], [CLAVE_USUARIO]) VALUES (@ID_USUARIO, @ID_PERFIL,
@NOMBRE_USUARIO, @APELLIDO_USUARIO, @CLAVE_USUARIO)
GO
/***** Object: StoredProcedure [dbo].[LogIngresoUsuario]  Script Date: 16/03/2015 10:30:24
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[LogIngresoUsuario]

```

```
@usuario varchar(15)
```

```
AS
```

```
INSERT INTO [TB_LOG_INGRESO]
  ([ID_USUARIO],[FECHA_INGRESO])
VALUES
  (@usuario,GETDATE())
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[LogTransaccionUsuario]  Script Date: 16/03/2015
10:30:24 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE [dbo].[LogTransaccionUsuario]
```

```
@usuario varchar(15),
@operacion varchar(30),
@campos varchar(max)
```

```
AS
```

```
INSERT INTO [TB_LOG_AUDITORIA]
  ([ID_USUARIO]
  ,[OPERACION]
  ,[CAMPOS]
  ,[FECHA])
VALUES
  (@usuario
  ,@operacion
  ,@campos
  ,GETDATE())
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[rptDepreciacion]  Script Date: 16/03/2015 10:30:24
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE [dbo].[rptDepreciacion]
```

```
@tipoactivo int,
```

```
@activo int
```

```
AS
```

```
declare @tiempo_util int
```

```
declare @depreciacion int

declare @nombreactivo varchar(100)

declare @nombretipo varchar(50)

declare @contador int

declare @valor float

declare @dpanual float

declare @acum float

declare @tbl table(anio int,concepto varchar(100),dpanual float,dpacum float,vactivo float)

declare @fechaingreso datetime

set @acum = 0

set @contador = 1

select @tiempo_util = TIEMPO_UTIL, @depreciacion = DEPRECIACION
from TB_TIPO_ACTIVO
where ID_TIPO = @tipoactivo

select @nombreactivo = ISNULL(NOMBRE_ACTIVO,'ACTIVO'),@valor =
VALOR,@fechaingreso = DATEADD(YY,1,FECHA_INGRESO)
from TB_ACTIVOS
where ID_ACTIVO = @activo

set @dpanual = @valor / cast(@tiempo_util as float) --- (@valor*cast(@depreciacion as float))/100

while @contador <= @tiempo_util

begin
```

```
set @valor = @valor - @dpanual
```

```
set @acum = @acum + @dpanual
```

```
insert into @tbl values(@contador,'Depreciación ' + @nombreactivo + ' ' +  
convert(varchar(10),@fechaingreso,120),@dpanual,@acum,@valor)
```

```
set @fechaingreso = DATEADD(YY,1,@fechaingreso)
```

```
set @contador = @contador + 1
```

```
end
```

```
select anio,concepto,dpacum,dpanual,vactivo =  
case when vactivo < 1 then 0  
else vactivo end  
from @tbl
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[SalidaActivo] Script Date: 16/03/2015 10:30:24 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE [dbo].[SalidaActivo]
```

```
@tipoactivo int,
```

```
@activo int,
```

```
@codigo varchar(10),
```

```
@empleado int,
```

```
@supervisor int
```

```
AS
```

```
if @codigo != "
```

```
select @activo = ID_ACTIVOS  
from TB_ACTIVOS  
where CODIGO_ACTIVOS = @codigo
```

```
if not exists(select 1 from [TR_MOVIMIENTO_ACTIVOS] where ID_ACTIVOS = @activo and  
FEC_SALIDA is null)
```

```
INSERT INTO [TR_MOVIMIENTO_ACTIVOS]
```

```
    (ID_TIPO)
```

```
    ,[ID_ACTIVOS]
```

```
    ,[ID_EMPLEADO]
```

```
        ,[ID_SUPERVISOR]
    ,[FEC_SALIDA]
    )
VALUES
    (@tipoactivo
    ,@activo
    ,@empleado
    ,@supervisor
    ,getdate())

GO
/***** Object: StoredProcedure [dbo].[SugiereTiempoUtil]  Script Date: 16/03/2015 10:30:24
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[SugiereTiempoUtil]

@tipo int

AS

declare @tiempo int

select @tiempo = [TIEMPO_UTIL]
from [TB_TIEMPO_UTIL]
where [ID_TIPO] = @tipo

if @@ROWCOUNT = 0
set @tiempo = 10

select @tiempo as [TIEMPO_UTIL]

GO
/***** Object: StoredProcedure [dbo].[TransfiereActivo]  Script Date: 16/03/2015 10:30:24
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[TransfiereActivo]

@asigna int,
@empleado int,
@newempleado int,
@motivo varchar(100)

AS

declare @fechaant datetime
declare @activo int

select @fechaant = [FECHA_ASIGNA],@activo = [ID_ACTIVIVO]
from [TR_ASIGNA_ACTIVIVO]
```

where [ID_ASIGNACION] = @asigna

```
IF NOT EXISTS(SELECT 1 FROM [TR_ASIGNA_ACTIVO] WHERE ID_ACTIVO = @activo
AND ID_EMPLEADO = @newempleado)
BEGIN
```

```
UPDATE [TR_ASIGNA_ACTIVO]
SET [ID_EMPLEADO] = @newempleado
,[ID_EMP_ANTERIOR] = @empleado
,[FECHA_ASIGNA] = getdate()
,[FECHA_REGRESA] = @fechaant
,[MOTIVO] = @motivo
WHERE [ID_ASIGNACION] = @asigna
```

```
INSERT INTO [TR_ASIGNA_ACTIVO]
([ID_ACTIVO]
,[ID_EMPLEADO]
,[FECHA_ASIGNA])
VALUES
(@activo
,@newempleado
,GETDATE())
```

END

GO

/****** Object: StoredProcedure [dbo].[Valida_Usuario] Script Date: 16/03/2015 10:30:24

*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Valida_Usuario]

@usuario varchar(12),

@pwd varchar(10)

AS

declare @valido int

```
SELECT @valido = 1
FROM [TB_USUARIOS]
WHERE [ID_USUARIO] = @usuario
AND [CLAVE_USUARIO] = @pwd
```

if @@ROWCOUNT > 0

set @valido = 1

else

set @valido = 0

SELECT @valido

GO

/****** Object: StoredProcedure [dbo].[ValidaUsuario] Script Date: 16/03/2015 10:30:24 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ValidaUsuario]

@usuario varchar(12),

```
@pwd varchar(20)
```

```
AS
```

```

    if exists(select 1
              from TB_USUARIOS
              where ID_USUARIO = @usuario
              and CLAVE_USUARIO = @pwd)

```

```
begin
```

```

select '1',ID_PERFIL
from TB_USUARIOS
where ID_USUARIO = @usuario
      and CLAVE_USUARIO = @pwd

```

```
end
```

```
else
```

```
begin
```

```

update TB_USUARIOS
set INTENTOS = ISNULL(INTENTOS,0) + 1
where ID_USUARIO = @usuario

```

```
select '0',0
```

```
end
```

```
GO
```

```
/****** Object: Table [dbo].[TB_ACTIVOS] Script Date: 16/03/2015 10:30:24 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```

CREATE TABLE [dbo].[TB_ACTIVOS](
    [ID_ACTIVIVO] [int] IDENTITY(1,1) NOT NULL,
    [ID_UBICACION] [int] NULL,
    [ID_TIPO] [int] NULL,
    [ID_AREA] [int] NULL,
    [CODIGO_ACTIVIVO] [varchar](20) NULL,
    [NOMBRE_ACTIVIVO] [varchar](100) NULL,
    [FECHA_INGRESO] [datetime] NULL,
    [MODELO] [varchar](10) NULL,
    [SERIAL] [varchar](25) NULL,
    [VALOR] [float] NULL,
    [ID_ESTADO] [int] NULL,

```

```
CONSTRAINT [PK_TB_ACTIVOS] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [ID_ACTIVIVO] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
```

```
GO
```

```
/****** Object: Table [dbo].[TB_AREA] Script Date: 16/03/2015 10:30:24 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_AREA](
    [ID_AEREA] [int] IDENTITY(1,1) NOT NULL,
    [NOMBRE_AREA] [varchar](50) NULL,
    CONSTRAINT [PK_TB_AREA] PRIMARY KEY CLUSTERED
(
    [ID_AEREA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_BAJAS]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_BAJAS](
    [ID_BAJA] [int] IDENTITY(1,1) NOT NULL,
    [ID_ACTIVO] [int] NULL,
    [FECHA_BAJA] [datetime] NULL,
    [MOTIVO] [varchar](max) NULL,
    CONSTRAINT [PK_TB_DEPRECIACION] PRIMARY KEY CLUSTERED
(
    [ID_BAJA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_CARGO]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_CARGO](
    [ID_CARGO] [int] IDENTITY(1,1) NOT NULL,
    [NOMBRE_CARGO] [varchar](50) NULL,
    CONSTRAINT [PK_TB_CARGO] PRIMARY KEY CLUSTERED
(
    [ID_CARGO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_CUSTODIO]  Script Date: 16/03/2015 10:30:24 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_CUSTODIO](
    [ID_CUSTODIO] [int] IDENTITY(1,1) NOT NULL,
    [NOMBRE_CUSTODIO] [varchar](100) NULL,
    [FEC_INGRESO] [datetime] NULL,
    [FEC_SALIDA] [datetime] NULL,
    [ID_CARGO] [int] NULL,
    CONSTRAINT [PK_TB_CUSTODIO] PRIMARY KEY CLUSTERED
(
    [ID_CUSTODIO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_EMPLEADO]   Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_EMPLEADO](
    [ID_EMPLEADO] [int] IDENTITY(1,1) NOT NULL,
    [ID_AREA] [int] NULL,
    [NOMBRE_EMPLEADO] [varchar](50) NULL,
    [APELLIDO_EMPLEADO] [varchar](50) NULL,
    [CEDULA] [numeric](18, 0) NULL,
    [FECHA_INGRESO] [datetime] NULL,
    [FECHA_SALIDA] [datetime] NULL,
    [PRINCIPAL] [bit] NULL,
    CONSTRAINT [PK_TB_EMPLEADO] PRIMARY KEY CLUSTERED
(
    [ID_EMPLEADO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_ESTADO]   Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_ESTADO](
    [ID_ESTADO] [int] IDENTITY(1,1) NOT NULL,
    [NOM_ESTADO] [varchar](30) NULL,
    CONSTRAINT [PK_TB_ESTADO] PRIMARY KEY CLUSTERED
(

```

```

        [ID_ESTADO] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_LOG_AUDITORIA]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_LOG_AUDITORIA](
    [ID_TRANSACCION] [int] IDENTITY(1,1) NOT NULL,
    [ID_USUARIO] [varchar](15) NULL,
    [OPERACION] [varchar](30) NULL,
    [CAMPOS] [varchar](max) NULL,
    [FECHA] [datetime] NULL,
    CONSTRAINT [PK_TB_LOG_AUDITORIA] PRIMARY KEY CLUSTERED
    (
        [ID_TRANSACCION] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_LOG_INGRESO]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_LOG_INGRESO](
    [ID_LOG] [int] IDENTITY(1,1) NOT NULL,
    [ID_USUARIO] [varchar](15) NULL,
    [FECHA_INGRESO] [datetime] NULL,
    CONSTRAINT [PK_TB_LOG_INGRESO] PRIMARY KEY CLUSTERED
    (
        [ID_LOG] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_PERFIL]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_PERFIL](

```

```

[ID_PERFIL] [int] IDENTITY(1,1) NOT NULL,
[NOMBRE_PERFIL] [varchar](50) NULL,
CONSTRAINT [PK_TB_PERFIL] PRIMARY KEY CLUSTERED
(
    [ID_PERFIL] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_TIEMPO_UTIL]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TB_TIEMPO_UTIL](
    [ID_TIEMPO] [int] IDENTITY(1,1) NOT NULL,
    [ID_TIPO] [int] NULL,
    [ID_ACTIVO] [int] NULL,
    [TIEMPO_UTIL] [int] NULL,
    [DEPRECIACION] [int] NULL,
    CONSTRAINT [PK_TB_TIEMPO_UTIL] PRIMARY KEY CLUSTERED
(
    [ID_TIEMPO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[TB_TIPO_ACTIVO]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_TIPO_ACTIVO](
    [ID_TIPO] [int] IDENTITY(1,1) NOT NULL,
    [NOMBRE_TIPO] [varchar](50) NULL,
    [IDENTIFICADOR] [varchar](5) NULL,
    [TIEMPO_UTIL] [int] NULL,
    [DEPRECIACION] [int] NULL,
    CONSTRAINT [PK_TB_TIPO_ACTIVO] PRIMARY KEY CLUSTERED
(
    [ID_TIPO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_UBICACION]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON

```

```

GO
CREATE TABLE [dbo].[TB_UBICACION](
    [ID_UBICACION] [int] IDENTITY(1,1) NOT NULL,
    [NOMBRE_UBICACION] [varchar](100) NULL,
    [DIRECCION_UBICACION] [varchar](100) NULL,
    [TELEFONO_UBICACION] [varchar](15) NOT NULL,
    [PRINCIPAL] [bit] NULL,
    CONSTRAINT [PK_TB_UBICACION] PRIMARY KEY CLUSTERED
(
    [ID_UBICACION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TB_USUARIOS]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TB_USUARIOS](
    [ID_USUARIO] [varchar](12) NOT NULL,
    [ID_PERFIL] [int] NULL,
    [NOMBRE_USUARIO] [varchar](50) NULL,
    [APELLIDO_USUARIO] [varchar](50) NULL,
    [CLAVE_USUARIO] [varchar](10) NULL,
    [INTENTOS] [int] NULL,
    CONSTRAINT [PK_TB_USUARIOS] PRIMARY KEY CLUSTERED
(
    [ID_USUARIO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TR_ACTIVIVO_ESTADO]  Script Date: 16/03/2015 10:30:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TR_ACTIVIVO_ESTADO](
    [ID_ACTIVIVO_STD] [int] IDENTITY(1,1) NOT NULL,
    [ID_ACTIVIVO] [int] NULL,
    [ID_ESTADO] [int] NULL,
    [FEC_BAJA] [datetime] NULL,
    CONSTRAINT [PK_TR_ACTIVIVO_ESTADO] PRIMARY KEY CLUSTERED
(
    [ID_ACTIVIVO_STD] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[TR_ASIGNA_ACTIVIVO]  Script Date: 16/03/2015 10:30:24 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TR_ASIGNA_ACTIVOS](
    [ID_ASIGNACION] [int] IDENTITY(1,1) NOT NULL,
    [ID_ACTIVOS] [int] NULL,
    [ID_EMPLEADO] [int] NULL,
    [ID_EMP_ANTERIOR] [int] NULL,
    [FECHA_ASIGNA] [datetime] NULL,
    [FECHA_REGRESA] [datetime] NULL,
    [MOTIVO] [varchar](100) NULL,
    CONSTRAINT [PK_TR_ASIGNA_ACTIVOS] PRIMARY KEY CLUSTERED
(
    [ID_ASIGNACION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TR_MOVIMIENTO_ACTIVOS]   Script Date: 16/03/2015 10:30:24
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TR_MOVIMIENTO_ACTIVOS](
    [ID_TIPO] [int] NULL,
    [ID_ACTIVOS] [int] NULL,
    [ID_EMPLEADO] [int] NULL,
    [ID_SUPERVISOR] [int] NULL,
    [FEC_SALIDA] [datetime] NULL,
    [FEC_ENTREGA] [datetime] NULL,
    [ID_ESTADO] [int] NULL,
    [OBSERVACION] [varchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[TB_ACTIVOS] WITH CHECK ADD CONSTRAINT
[FK_TB_ACTIVOS_TB_AREA] FOREIGN KEY([ID_AREA])
REFERENCES [dbo].[TB_AREA] ([ID_AEREA])
GO
ALTER TABLE [dbo].[TB_ACTIVOS] CHECK CONSTRAINT [FK_TB_ACTIVOS_TB_AREA]
GO
ALTER TABLE [dbo].[TB_ACTIVOS] WITH CHECK ADD CONSTRAINT
[FK_TB_ACTIVOS_TB_TIPO_ACTIVOS] FOREIGN KEY([ID_TIPO])
REFERENCES [dbo].[TB_TIPO_ACTIVOS] ([ID_TIPO])
GO
ALTER TABLE [dbo].[TB_ACTIVOS] CHECK CONSTRAINT
[FK_TB_ACTIVOS_TB_TIPO_ACTIVOS]
GO

```

```

ALTER TABLE [dbo].[TB_ACTIVOS] WITH CHECK ADD CONSTRAINT
[FK_TB_ACTIVOS_TB_UBICACION] FOREIGN KEY([ID_UBICACION])
REFERENCES [dbo].[TB_UBICACION] ([ID_UBICACION])
GO
ALTER TABLE [dbo].[TB_ACTIVOS] CHECK CONSTRAINT
[FK_TB_ACTIVOS_TB_UBICACION]
GO
ALTER TABLE [dbo].[TB_BAJAS] WITH CHECK ADD CONSTRAINT
[FK_TB_DEPRECIACION_TB_ACTIVOS] FOREIGN KEY([ID_ACTIVOS])
REFERENCES [dbo].[TB_ACTIVOS] ([ID_ACTIVOS])
GO
ALTER TABLE [dbo].[TB_BAJAS] CHECK CONSTRAINT
[FK_TB_DEPRECIACION_TB_ACTIVOS]
GO
ALTER TABLE [dbo].[TB_CUSTODIO] WITH CHECK ADD CONSTRAINT
[FK_TB_CUSTODIO_TB_CARGO] FOREIGN KEY([ID_CARGO])
REFERENCES [dbo].[TB_CARGO] ([ID_CARGO])
GO
ALTER TABLE [dbo].[TB_CUSTODIO] CHECK CONSTRAINT
[FK_TB_CUSTODIO_TB_CARGO]
GO
ALTER TABLE [dbo].[TB_EMPLEADO] WITH CHECK ADD CONSTRAINT
[FK_TB_EMPLEADO_TB_AREA] FOREIGN KEY([ID_AREA])
REFERENCES [dbo].[TB_AREA] ([ID_AEREA])
GO
ALTER TABLE [dbo].[TB_EMPLEADO] CHECK CONSTRAINT
[FK_TB_EMPLEADO_TB_AREA]
GO
ALTER TABLE [dbo].[TB_USUARIOS] WITH CHECK ADD CONSTRAINT
[FK_TB_USUARIOS_TB_PERFIL] FOREIGN KEY([ID_PERFIL])
REFERENCES [dbo].[TB_PERFIL] ([ID_PERFIL])
GO
ALTER TABLE [dbo].[TB_USUARIOS] CHECK CONSTRAINT
[FK_TB_USUARIOS_TB_PERFIL]
GO
ALTER TABLE [dbo].[TR_ACTIVO_ESTADO] WITH CHECK ADD CONSTRAINT
[FK_TR_ACTIVO_ESTADO_TB_ACTIVOS] FOREIGN KEY([ID_ACTIVOS])
REFERENCES [dbo].[TB_ACTIVOS] ([ID_ACTIVOS])
GO
ALTER TABLE [dbo].[TR_ACTIVO_ESTADO] CHECK CONSTRAINT
[FK_TR_ACTIVO_ESTADO_TB_ACTIVOS]
GO
ALTER TABLE [dbo].[TR_ACTIVO_ESTADO] WITH CHECK ADD CONSTRAINT
[FK_TR_ACTIVO_ESTADO_TB_ESTADO] FOREIGN KEY([ID_ESTADO])
REFERENCES [dbo].[TB_ESTADO] ([ID_ESTADO])
GO
ALTER TABLE [dbo].[TR_ACTIVO_ESTADO] CHECK CONSTRAINT
[FK_TR_ACTIVO_ESTADO_TB_ESTADO]
GO
ALTER TABLE [dbo].[TR_MOVIMIENTO_ACTIVOS] WITH CHECK ADD CONSTRAINT
[FK_TR_MOVIMIENTO_ACTIVOS_TB_ACTIVOS] FOREIGN KEY([ID_ACTIVOS])
REFERENCES [dbo].[TB_ACTIVOS] ([ID_ACTIVOS])
GO
ALTER TABLE [dbo].[TR_MOVIMIENTO_ACTIVOS] CHECK CONSTRAINT
[FK_TR_MOVIMIENTO_ACTIVOS_TB_ACTIVOS]
GO
ALTER TABLE [dbo].[TR_MOVIMIENTO_ACTIVOS] WITH CHECK ADD CONSTRAINT
[FK_TR_MOVIMIENTO_ACTIVOS_TB_EMPLEADO] FOREIGN KEY([ID_EMPLEADO])
REFERENCES [dbo].[TB_EMPLEADO] ([ID_EMPLEADO])
GO

```



```

        </div>
    </nav>
</section><div class="ic">ABS Computer S.A. Ecuador</div>
</header>
<section id="content">
    <div class="top">
        <div class="container">
            <div class="clearfix">
                <section id="gallery">
                    <div class="pics">
                        
                        
                        
                    </div>
                    <a href="#" id="prev"></a>
                    <a href="#" id="next"></a>
                </section>
                <section id="intro">
                    </section>
            </div>
        </div>
    </div>
<div class="middle">
    <div class="container">
        <div class="wrapper">
            <div class="grid9">
                <h2>Soluciones Informat&iacute;cas
Integrales</h2>
                <p>
                    </p>
                </div>
            </div>
        </div>
    </div>
<div class="bottom">
    <div class="container">
        <div class="wrapper">
            <div class="grid3">
                <div id="datepicker"></div>
                <script type="text/javascript">

```

```

$(document).ready(function () {
    $('.pics').cycle({
        fx: 'toss',
        next: '#next',
        prev: '#prev'
    });

    $('#datepicker').datepicker({
        inline: true
    });
});
</script>
</body>
</html>

```

WEB CONFIG

```

<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <httpHandlers>
      <add verb="*" path="Barcode.axd" type="Limilabs.Web.BarcodeHttpHandler"/>
    </httpHandlers>
    <compilation debug="true" targetFramework="4.0">
      <assemblies>
        <add assembly="System.Design, Version=4.0.0.0, Culture=neutral,
        PublicKeyToken=B03F5F7F11D50A3A"/>
        <add assembly="System.Web.Extensions.Design, Version=4.0.0.0, Culture=neutral,
        PublicKeyToken=31BF3856AD364E35"/>
        <add assembly="System.Windows.Forms, Version=4.0.0.0, Culture=neutral,
        PublicKeyToken=B77A5C561934E089"/>
        <add assembly="Barcode, Version=2.0.0.20, Culture=neutral,
        PublicKeyToken=6DC438AB78A525B3"/>
        <add assembly="System.Security, Version=4.0.0.0, Culture=neutral,
        PublicKeyToken=B03F5F7F11D50A3A"/>
      </assemblies>
    </compilation>
  </system.web>
  <connectionStrings>
    <add name="Activos" connectionString="Data Source=.;Initial Catalog=db_activos;Integrated
    Security=True"/>
  </connectionStrings>
</configuration>

```

NEGOCIO

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Data.Common;
using System.Linq;

```

```
using System.Text;

namespace Activos.Conexion
{
    public class clsConexion
    {

        private string cadena = ConfigurationManager.ConnectionStrings["Activos"].ConnectionString;

        public SqlConnection con;
        private SqlCommandBuilder cmb;
        public DataSet ds = new DataSet();
        public SqlDataAdapter da;
        public SqlCommand comando;
        public SqlDataReader dr;
        public SqlTransaction tr = null;

        private void conectar()
        {
            con = new SqlConnection(cadena);

        }

        public clsConexion()
        {
            conectar();
        }

        public void consultar(string sql, string tabla)
        {
            try
            {
                ds.Tables.Clear();
                da = new SqlDataAdapter(sql, con);
                cmb = new SqlCommandBuilder(da);
                da.Fill(ds, tabla);
            }
            catch
            {
                throw;
            }
        }

        public bool eliminar(string tabla, string condicion)
        {
            try
            {
                con.Open();
                string sql = "delete from " + tabla + " where " + condicion;
                comando = new SqlCommand(sql, con);
                int i = comando.ExecuteNonQuery();
                con.Close();

                if (i > 0)
                    return true;
            }
        }
    }
}
```

```

        else
            return false;
    }
    catch
    {
        throw;
    }
}

public bool actualizar(string tabla, string campos, string condicion)
{
    try
    {
        con.Open();
        string sql = "update " + tabla + " set " + campos + " where " + condicion;
        comando = new SqlCommand(sql, con);
        int i = comando.ExecuteNonQuery();
        con.Close();

        if (i > 0)
            return true;
        else
            return false;
    }
    catch
    {
        throw;
    }
}

public DataTable consulta(string tabla)
{
    try
    {
        string sql = "select * from " + tabla;
        da = new SqlDataAdapter(sql, con);
        DataSet ds = new DataSet();
        da.Fill(ds, tabla);
        DataTable dt = new DataTable();
        dt = ds.Tables[tabla];
        return dt;
    }
    catch
    {
        throw;
    }
}

public DataTable consultasql(string sql)
{
    try
    {
        con.Open();
        comando = new SqlCommand(sql, con);

        dr = comando.ExecuteReader(CommandBehavior.CloseConnection);
        DataTable dt = new DataTable();

        dt.Load(dr);
    }
}

```

```

        return dt;
    }
    catch
    {
        throw;
    }
}

public bool insertar(string sql)
{
    try
    {
        con.Open();
        comando = new SqlCommand(sql, con);
        int i = comando.ExecuteNonQuery();
        con.Close();

        if (i > 0)
            return true;
        else
            return false;
    }
    catch
    {
        throw;
    }
}

}

}

```

NEGOCIO

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using Activos.Conexion;

namespace Activos.Negocio
{
    public class clsNegocio
    {

        clsConexion CN = new clsConexion();

        public DataTable ListadoPerfiles()
        {

            try
            {
                CN.consultar("SELECT * FROM TB_PERFIL", "TB_PERFIL");
                return CN.ds.Tables["TB_PERFIL"];
            }
            catch

```

```

    {
        throw;
    }

}

public bool InsertaPerfil(string valor)
{
    try
    {
        string sql = "INSERT INTO [TB_PERFIL] ([NOMBRE_PERFIL]) VALUES ('" + valor +
        ")";

        return CN.insertar(sql);

    }
    catch
    {
        throw;
    }
}

public DataTable ListadoUsuarios()
{
    try
    {
        string sql = "SELECT a.ID_USUARIO, a.ID_PERFIL, b.NOMBRE_PERFIL,
a.NOMBRE_USUARIO, a.APELLIDO_USUARIO,CLAVE_USUARIO "
        + "FROM TB_USUARIOS AS a INNER JOIN "
        + "TB_PERFIL AS b ON a.ID_PERFIL = b.ID_PERFIL";

        return CN.consultasql(sql);

    }
    catch
    {
        throw;
    }
}

public DataTable ListadoUbicacion()
{
    try
    {
        string sql = "SELECT
[ID_UBICACION],[NOMBRE_UBICACION],[DIRECCION_UBICACION],[TELEFONO_UBICA
CION],[PRINCIPAL] "
        + "FROM [TB_UBICACION]";

        return CN.consultasql(sql);

    }
    catch
    {
        throw;
    }
}

```

```

public DataTable ListadoCustodio()
{
    try
    {
        string sql = "SELECT
[ID_CUSTODIO],[NOMBRE_CUSTODIO],[FEC_INGRESO],[FEC_SALIDA],a.ID_CARGO,[NO
MBRE_CARGO]"
        + "FROM [TB_CUSTODIO] a "
        + "INNER JOIN [TB_CARGO] b on a.[ID_CARGO] = b.[ID_CARGO]";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoCargos()
{
    try
    {
        string sql = "SELECT [ID_CARGO],[NOMBRE_CARGO] "
        + "FROM [TB_CARGO]";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoEstados()
{
    try
    {
        string sql = "SELECT [ID_ESTADO],[NOM_ESTADO] "
        + "FROM [TB_ESTADO]";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoTipoActivo()
{
    try
    {
        string sql = "SELECT
[ID_TIPO],[NOMBRE_TIPO],[IDENTIFICADOR],[TIEMPO_UTIL],[DEPRECIACION] "
        + "FROM [TB_TIPO_ACTIVOS]";

        return CN.consultasql(sql);
    }
}

```

```

    }
    catch
    {
        throw;
    }
}

public DataTable ListadoAreas()
{
    try
    {
        string sql = "SELECT [ID_AEREA],[NOMBRE_AREA] "
            + "FROM [TB_AREA]";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable DetalleActivo(int activo)
{
    try
    {
        string sql = "SELECT a.ID_UBICACION,a.ID_TIPO,CODIGO_ACTIVIVO,
NOMBRE_ACTIVIVO, FECHA_INGRESO, SERIAL, MODELO,VALOR,a.ID_AREA "
            + "FROM TB_ACTIVOS a "
            + "INNER JOIN TB_UBICACION b on a.ID_UBICACION = b.ID_UBICACION "
            + "INNER JOIN TB_TIPO_ACTIVIVO c on a.ID_TIPO = c.ID_TIPO "
            + "INNER JOIN TB_ESTADO d on a.ID_ESTADO = d.ID_ESTADO "
            + "INNER JOIN TB_AREA e on a.ID_AREA = e.ID_AEREA "
            + "WHERE ID_ACTIVIVO =" + activo;

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoActivos()
{
    try
    {
        string sql = "SELECT ID_ACTIVIVO,
a.ID_UBICACION,NOMBRE_UBICACION,a.ID_TIPO, NOMBRE_TIPO, CODIGO_ACTIVIVO,
NOMBRE_ACTIVIVO, FECHA_INGRESO, SERIAL,
MODELO,VALOR,a.ID_ESTADO,NOM_ESTADO,a.ID_AREA,NOMBRE_AREA "
            + "FROM TB_ACTIVOS a "
            + "INNER JOIN TB_UBICACION b on a.ID_UBICACION = b.ID_UBICACION "
            + "INNER JOIN TB_TIPO_ACTIVIVO c on a.ID_TIPO = c.ID_TIPO "
            + "INNER JOIN TB_ESTADO d on a.ID_ESTADO = d.ID_ESTADO "
            + "INNER JOIN TB_AREA e on a.ID_AREA = e.ID_AEREA";

        return CN.consultasql(sql);
    }
}

```

```

        catch
        {
            throw;
        }
    }

    public DataTable ListadoActivo()
    {
        try
        {
            string sql = "SELECT ID_ACTIVADO, CODIGO_ACTIVADO + '_' + NOMBRE_ACTIVADO
NOMBRE_ACTIVADO "
                + "FROM TB_ACTIVOS";

            return CN.consultasql(sql);
        }
        catch
        {
            throw;
        }
    }

    public DataTable ListadoActivo(int tipo)
    {
        try
        {
            string sql = "SELECT ID_ACTIVADO, CODIGO_ACTIVADO + '_' + NOMBRE_ACTIVADO
ACTIVO "
                + "FROM TB_ACTIVOS "
                + "WHERE ID_TIPO =" + tipo;

            return CN.consultasql(sql);
        }
        catch
        {
            throw;
        }
    }

    public DataTable ListadoTiempoUtil()
    {
        try
        {
            string sql = "SELECT [ID_TIEMPO],a.ID_TIPO
,b.NOMBRE_TIPO,a.ID_ACTIVADO,NOMBRE_ACTIVADO, "
                + "[TIEMPO_UTIL],[DEPRECIACION] "
                + "FROM [TB_TIEMPO_UTIL] a "
                + "INNER JOIN [TB_TIPO_ACTIVADO] b on a.ID_TIPO = b.ID_TIPO "
                + "INNER JOIN TB_ACTIVOS c on a.ID_ACTIVADO = c.ID_ACTIVADO and
a.ID_TIPO = c.ID_TIPO";

            return CN.consultasql(sql);
        }
        catch
        {
            throw;
        }
    }

```

```

}

public DataTable ListadoActivoVigente()
{
    try
    {
        string sql = "SELECT [ID_ACTIVO] "
            + ",[CODIGO_ACTIVO] + '-' + [NOMBRE_ACTIVO] as ACTIVO "
            + "FROM [TB_ACTIVOS]"
            + "WHERE ID_ESTADO = 1";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoActivoVigente(int tipo)
{
    try
    {
        string sql = "SELECT [ID_ACTIVO] "
            + ",[CODIGO_ACTIVO] + '-' + [NOMBRE_ACTIVO] as ACTIVO "
            + "FROM [TB_ACTIVOS]"
            + "WHERE ID_ESTADO = 1"
            + "AND ID_TIPO = " + tipo;

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoActivoSalida(int tipo)
{
    try
    {
        string sql = "SELECT [ID_ACTIVO] "
            + ",[CODIGO_ACTIVO] + '-' + [NOMBRE_ACTIVO] as ACTIVO "
            + "FROM [TB_ACTIVOS] "
            + "WHERE [ID_ACTIVO] NOT IN (SELECT [ID_ACTIVO] "
            + "FROM [TR_MOVIMIENTO_ACTIVO]) "
            + "AND ID_ESTADO = 1 "
            + "AND ID_TIPO = " + tipo
            + " GROUP BY [ID_ACTIVO] ,[CODIGO_ACTIVO],[NOMBRE_ACTIVO] "
            + "UNION "
            + "SELECT a.[ID_ACTIVO] "
            + ",[CODIGO_ACTIVO] + '-' + [NOMBRE_ACTIVO] as ACTIVO "
            + "FROM [TB_ACTIVOS] a "
            + "INNER JOIN [TR_MOVIMIENTO_ACTIVO] b on a.ID_ACTIVO =
b.ID_ACTIVO and a.ID_TIPO = b.ID_TIPO "
            + "WHERE a.ID_ESTADO = 2 "

```

```

+ "AND FEC_ENTREGA IS NULL "
+ "AND a.ID_TIPO = " + tipo
+ " AND FEC_SALIDA IN (SELECT MAX(FEC_SALIDA) "
+ "FROM [TR_MOVIMIENTO_ACTIVADO] X "
+ "WHERE b.ID_ACTIVADO = X.ID_ACTIVADO) "
+ "GROUP BY a.[ID_ACTIVADO],[CODIGO_ACTIVADO],[NOMBRE_ACTIVADO]";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoActivoIngreso(int tipo)
{
    try
    {
        string sql = "SELECT a.[ID_ACTIVADO] "
            + ",[CODIGO_ACTIVADO] + '-' + [NOMBRE_ACTIVADO] as ACTIVO "
            + "FROM [TB_ACTIVOS] a "
            + "INNER JOIN [TR_MOVIMIENTO_ACTIVADO] b on a.ID_ACTIVADO =
b.ID_ACTIVADO and a.ID_TIPO = b.ID_TIPO "
            + "WHERE a.ID_ESTADO = 1 "
            + "AND FEC_ENTREGA IS NULL "
            + "AND a.ID_TIPO = " + tipo
            + " GROUP BY a.[ID_ACTIVADO],[CODIGO_ACTIVADO],[NOMBRE_ACTIVADO]";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoEmpleados()
{
    try
    {
        string sql = "SELECT [ID_EMPLEADO] "
            + ",[ID_AREA],[NOMBRE_AREA] "
            + ",[NOMBRE_EMPLEADO],[APELLIDO_EMPLEADO] "
            + ",convert(varchar(20),[FECHA_INGRESO],120)
FECHA_INGRESO,[PRINCIPAL] "
            + "FROM [TB_EMPLEADO] a "
            + "INNER JOIN [TB_AREA] b on a.ID_AREA = b.ID_AEREA";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

```

```

public DataTable ListadoSupervisores()
{
    try
    {
        string sql = "SELECT [ID_EMPLEADO] "
            + ",[NOMBRE_EMPLEADO] + ' ' + [APELLIDO_EMPLEADO] EMPLEADO "
            + "FROM [TB_EMPLEADO] "
            + "WHERE [PRINCIPAL] = 1";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoPupilos()
{
    try
    {
        string sql = "SELECT [ID_EMPLEADO] "
            + ",[NOMBRE_EMPLEADO] + ' ' + [APELLIDO_EMPLEADO] EMPLEADO "
            + "FROM [TB_EMPLEADO] "
            + "WHERE [PRINCIPAL] = 0";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoSalida()
{
    try
    {
        string sql = "SELECT NOMBRE_TIPO as 'Tipo Activo' "
            + ",[CODIGO_ACTIVOS] + '-' + [NOMBRE_ACTIVOS] as 'Nombre Activo' "
            + ",d.NOMBRE_EMPLEADO + ' ' + d.APELLIDO_EMPLEADO as 'Custodio' "
            + ",e.NOMBRE_EMPLEADO + ' ' + e.APELLIDO_EMPLEADO as 'Autoriza' "
            + ",a.[FEC_SALIDA] as 'Fecha Salida' "
            + "FROM [TR_MOVIMIENTO_ACTIVOS] a "
            + "INNER JOIN [TB_ACTIVOS] b on a.ID_ACTIVOS = b.ID_ACTIVOS "
            + "INNER JOIN [TB_TIPO_ACTIVOS] c on a.ID_TIPO = c.ID_TIPO "
            + "INNER JOIN [TB_EMPLEADO] d on a.ID_EMPLEADO = d.ID_EMPLEADO "
            + "INNER JOIN [TB_EMPLEADO] e on a.ID_SUPERVISOR = "
            + "e.ID_EMPLEADO "
            + "WHERE FEC_ENTREGA IS NULL";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

```

```

public DataTable ListadoEntrada()
{
    try
    {
        string sql = "SELECT NOMBRE_TIPO as 'Tipo Activo' "
            + ",NOMBRE_ACTIVADO as 'Nombre Activo' "
            + ",d.NOMBRE_EMPLEADO + ' ' + d.APELLIDO_EMPLEADO as 'Custodio' "
            + ",a.[FEC_SALIDA] as 'Fecha Salida' "
            + ",a.[FEC_ENTREGA] as 'Fecha Ingreso' "
            + ",[OBSERVACION] as 'Observacion',[NOM_ESTADO] as 'Estado' "
            + "FROM [TR_MOVIMIENTO_ACTIVADO] a "
            + "INNER JOIN [TB_ACTIVOS] b on a.ID_ACTIVADO = b.ID_ACTIVADO "
            + "INNER JOIN [TB_TIPO_ACTIVADO] c on a.ID_TIPO = c.ID_TIPO "
            + "INNER JOIN [TB_EMPLEADO] d on a.ID_EMPLEADO = d.ID_EMPLEADO
"
            + "INNER JOIN [TB_ESTADO] f on a.ID_ESTADO = f.ID_ESTADO";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoBajas()
{
    try
    {
        string sql = "SELECT b.CODIGO_ACTIVADO + '-' + b.NOMBRE_ACTIVADO
NOMBRE_ACTIVADO,c.NOMBRE_TIPO,MOTIVO,FECHA_BAJA "
            + "FROM TB_BAJAS a "
            + "INNER JOIN TB_ACTIVOS b on a.ID_ACTIVADO = b.ID_ACTIVADO "
            + "INNER JOIN TB_TIPO_ACTIVADO c on b.ID_TIPO = c.ID_TIPO "
            + "ORDER BY FECHA_BAJA";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ListadoEmpleado()
{
    try
    {
        string sql = "SELECT [ID_EMPLEADO],[NOMBRE_EMPLEADO] + ' ' +
[APELLIDO_EMPLEADO] EMPLEADO "
            + "FROM [TB_EMPLEADO] ";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

```

```

}

public DataTable ListadoEmpleado(int empleado)
{
    try
    {
        string sql = "SELECT [ID_EMPLEADO],[NOMBRE_EMPLEADO] + ' ' +
[APELLIDO_EMPLEADO] EMPLEADO "
        + "FROM [TB_EMPLEADO] "
        + "WHERE [ID_EMPLEADO] !=" + empleado;

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable AsignacionActivo(int empleado)
{
    try
    {
        string sql = "SELECT cast([ID_ASIGNACION] as varchar) + '_' +
cast(a.[ID_EMPLEADO] as varchar) CODIGO "
        + ", [NOMBRE_EMPLEADO] + ' ' + [APELLIDO_EMPLEADO] + ' ' +
NOMBRE_ACTIVAS ASIGNACION "
        + "FROM [TR_ASIGNA_ACTIVAS] a "
        + "INNER JOIN TB_ACTIVAS b on a.[ID_ACTIVAS] = b.[ID_ACTIVAS] "
        + "INNER JOIN TB_EMPLEADO c on a.[ID_EMPLEADO] = c.[ID_EMPLEADO]
"
        + "WHERE a.[ID_EMPLEADO] = " + empleado
        + " AND [ID_EMP_ANTERIOR] IS NULL";

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ReporteDepreciacion(int tipoactivo, int activo)
{
    try
    {
        string sql = "exec rptDepreciacion " + tipoactivo + "," + activo;

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public DataTable ReporteActivos(string activo, int ubicacion, int empleado, int tipo)
{
    try

```

```

    {
        string sql = "exec ConsultaActivo " + activo + "," + empleado + "," + ubicacion + "," +
tipo;

        return CN.consultasql(sql);
    }
    catch
    {
        throw;
    }
}

public string NombrePerfil(int idperfil)
{
    try
    {
        string sql = "select nombre_perfil "
            + "from tb_perfil "
            + "where id_perfil =" + idperfil;

        DataTable dtNP = CN.consultasql(sql);

        return dtNP.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}

public string NombreUbicacion(int idubicacion)
{
    try
    {
        string sql = "select [NOMBRE_UBICACION] "
            + "from [TB_UBICACION] "
            + "where [ID_UBICACION] =" + idubicacion;

        DataTable dtUB = CN.consultasql(sql);

        return dtUB.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}

public string NombreArea(int idarea)
{
    try
    {
        string sql = "SELECT [NOMBRE_AREA] "
            + "FROM [TB_AREA] "
            + "WHERE [ID_AEREA] =" + idarea;

        DataTable dtAr = CN.consultasql(sql);
        return dtAr.Rows[0][0].ToString();
    }
    catch

```

```
{
    throw;
}
}

public string NombreTipoActivo(int idtipo)
{
    try
    {
        string sql = "select [NOMBRE_TIPO] "
            + "from [TB_TIPO_ACTIVO] "
            + "where [ID_TIPO] =" + idtipo;

        DataTable dtTI = CN.consultasql(sql);

        return dtTI.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}

public string NombreTipoActivo(string nombre)
{
    try
    {
        string sql = "select [NOMBRE_TIPO] "
            + "from [TB_TIPO_ACTIVO] "
            + "where [NOMBRE_TIPO] =" + nombre + """;

        DataTable dtTI = CN.consultasql(sql);

        return dtTI.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}

public string NombreIdenTipoActivo(string ident)
{
    try
    {
        string sql = "select [IDENTIFICADOR]"
            + "from [TB_TIPO_ACTIVO] "
            + "where [IDENTIFICADOR] =" + ident + """;

        DataTable dtTI = CN.consultasql(sql);

        return dtTI.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}
```

```
public bool ExisteSerial(string serial)
{
    try
    {
        string sql = "select [SERIAL] "
            + "from [TB_ACTIVOS] "
            + "where [SERIAL] =" + serial + """;

        DataTable dtTI = CN.consultasql(sql);

        if (dtTI.Rows.Count > 0)
            return true;
        else
            return false;
    }
    catch
    {
        throw;
    }
}

public bool ExisteEmpleado(string cedula)
{
    try
    {
        string sql = "SELECT 1 "
            + "FROM [TB_EMPLEADO] "
            + "WHERE [CEDULA] = " + cedula;

        DataTable dtE = CN.consultasql(sql);

        if(dtE.Rows.Count > 0)
            return true;
        else
            return false;
    }
    catch
    {
        throw;
    }
}

public string NombreCodigoActivo(int codigo)
{
    try
    {
        string sql = "select [NOMBRE_ACTIVOS],[CODIGO_ACTIVOS] "
            + "from [TB_ACTIVOS] "
            + "where [ID_ACTIVOS] =" + codigo;

        DataTable dtTI = CN.consultasql(sql);

        return dtTI.Rows[0][0].ToString() + "_" + dtTI.Rows[0][1].ToString();
    }
    catch
    {

```

```
        throw;
    }
}

public string NombreCargo(int idcargo)
{
    try
    {
        string sql = "select [NOMBRE_CARGO] "
            + "from [TB_CARGO] "
            + "where [ID_CARGO] =" + idcargo;

        DataTable dtES = CN.consultasql(sql);

        return dtES.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}

public string NombreEstado(int idestado)
{
    try
    {
        string sql = "select [NOM_ESTADO] "
            + "from [TB_ESTADO] "
            + "where [ID_ESTADO] =" + idestado;

        DataTable dtES = CN.consultasql(sql);

        return dtES.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}

public string NombreActivo(int idactivo)
{
    try
    {
        string sql = "select [ID_ACTIVO],[CODIGO_ACTIVO] + '_' + [NOMBRE_ACTIVO]
ACTIVO "
            + "from [TB_ACTIVOS] "
            + "where [ID_ACTIVO] =" + idactivo;

        DataTable dtES = CN.consultasql(sql);

        return dtES.Rows[0][0].ToString();
    }
    catch
    {
        throw;
    }
}
```

```
public int TiempoUtil(int tipo)
{
    try
    {
        string sql = "exec SugiereTiempoUtil " + tipo;

        DataTable dtT = CN.consultasql(sql);

        return Convert.ToInt32(dtT.Rows[0][0]);
    }
    catch
    {
        throw;
    }
}

public string ValidaUsuario(string user, string pwd)
{
    try
    {
        string sql = "exec ValidaUsuario '" + user + "'," + pwd + """;

        DataTable tbValida = CN.consultasql(sql);

        string valida = "0";
        string perfil = "0";

        if (tbValida.Rows.Count > 0)
        {
            valida = tbValida.Rows[0][0].ToString();
            perfil = tbValida.Rows[0][1].ToString();

        }

        return valida + "_" + perfil;
    }
    catch
    {
        throw;
    }
}

public int ExisteUsuario(string user)
{
    try
    {
        string sql = "exec ExisteUsuario '" + user + """;
        int existe = 0;

        DataTable tbExiste = CN.consultasql(sql);

        if (tbExiste.Rows.Count > 0)
```

```

        existe = Convert.ToInt32(tbExiste.Rows[0][0]);

        return existe;
    }
    catch
    {
        throw;
    }
}

public int BloqueaUsuario(string user)
{
    try
    {
        string sql = "exec BloqueaUsuario " + user + "";
        int intentos = 0;

        DataTable tbExiste = CN.consultasql(sql);

        if (tbExiste.Rows.Count > 0)
            intentos = Convert.ToInt32(tbExiste.Rows[0][0]);

        return intentos;
    }
    catch
    {
        throw;
    }
}

public bool InsertaUsuario(string sql)
{
    try
    {
        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool InsertaUbicacion(string nombre, string direccion, string telefono, int principal)
{
    try
    {
        string sql = "INSERT [TB_UBICACION] "
            + "[NOMBRE_UBICACION] "
            + ", [DIRECCION_UBICACION] "
            + ", [TELEFONO_UBICACION] "
            + ", [PRINCIPAL] "
            + "VALUES "
            + "(" + nombre + " "
            + ", " + direccion + " "
            + ", " + telefono + " "
            + ", " + principal + ")";
    }
}

```

```
        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool InsertaEstado(string nombre)
{
    try
    {
        string sql = "INSERT INTO [TB_ESTADO]([NOM_ESTADO]) "
            + "VALUES('" + nombre + "')";

        return CN.insertar(sql);

    }
    catch
    {
        throw;
    }
}

public bool InsertaTiempoUtil(int tipo,int activo,int tiempo,int depreciacion)
{
    try
    {
        string sql = "INSERT INTO [TB_TIEMPO_UTIL] "
            + "([ID_TIPO],[ID_ACTIVO],[TIEMPO_UTIL],[DEPRECIACION]) "
            + "VALUES(" + tipo + "," + activo + "," + tiempo + "," + depreciacion + ")";

        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool InsertarCustodio(string nombre,int estado )
{
    try
    {
        string sql = "exec InsertaCustodio '" + nombre + "'," + estado;

        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool TransfiereActivo(int codigo, int empleado, int newempleado,string motivo)
{
```

```

    try
    {
        string sql = "exec TransfiereActivo " + codigo + "," + empleado + "," + newempleado + "," +
+ motivo + """;
        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool SalidaActivo(int tipo, int activo, string codigo, int empleado, int supervisor)
{
    try
    {
        string sql = "SalidaActivo " + tipo + "," + activo + "," + codigo + "," + empleado + "," +
supervisor;

        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool IngresoActivo(int tipo, int activo, int estado, string observacion)
{
    try
    {
        string sql = "IngresoActivo " + tipo + "," + activo + "," + estado + "," + observacion + """;

        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

public bool InsertarTipoActivo(string nombre, string identificador, int tiempo, int depreciacion)
{
    try
    {
        string sql = "INSERT INTO
[dbo].[TB_TIPO_ACTIVOS]([NOMBRE_TIPO],[IDENTIFICADOR],[TIEMPO_UTIL],[DEPRECIACION]) "
+ "VALUES('" + nombre + "','" + identificador + "','" + tiempo + "','" + depreciacion
+ ")";

        return CN.insertar(sql);
    }
    catch
    {
        throw;
    }
}

```

```

public bool InsertarEmpleado(int area,long cedula,string nombre,string apellido,string fecha,int
principal)
{
    try
    {
        string sql = "INSERT INTO [TB_EMPLEADO] "
        +
        "([ID_AREA],[CEDULA],[NOMBRE_EMPLEADO],[APELLIDO_EMPLEADO],[FECHA_INGRE
SO],[PRINCIPAL]) "
        + "VALUES( " + area + "," + cedula + "," + nombre + "," + apellido + "," + fecha + ","
+ principal + ")";

        return CN.insertar(sql);

    }
    catch
    {
        throw;
    }
}

public bool InsertaActivos(int ubicacion,int tipo,string nombre,string modelo,string serial,double
valor,string fecha,int area)
{
    try
    {
        string sql = "exec Ingresa_Activo " + ubicacion + "," + tipo + "," + nombre + "," + modelo
+ "," + serial + "," + valor + "," + fecha + "," + area;

        return CN.insertar(sql);

    }
    catch
    {
        throw;
    }
}

public bool BajaActivos(int activo,string motivo)
{
    try
    {
        string sql = "exec BajaActivo " + activo + "," + motivo + """;

        return CN.insertar(sql);

    }
    catch
    {
        throw;
    }
}

public bool ActualizaUsuario(string usuario, int perfil, string nombre, string apellido, string pwd)
{
    try
    {
        string tabla = "TB_USUARIOS";

```

```

        string campos = "ID_PERFIL =" + perfil + ",NOMBRE_USUARIO=" + nombre +
        ",APELLIDO_USUARIO=" + apellido + ",CLAVE_USUARIO=" + pwd + """;
        string condicion = "ID_USUARIO=" + usuario + """;

        return CN.actualizar(tabla, campos, condicion);

    }
    catch
    {
        throw;
    }
}

public bool ActualizarTipoActivo(int codigo, string nombre, string identificador, int
depreciacion)
{
    try
    {
        string tabla = "TB_TIPO_ACTIVO";
        string campos = "NOMBRE_TIPO=" + nombre + ",IDENTIFICADOR=" + identificador
+ """; //,DEPRECIACION=" + depreciacion;
        string condicion = "ID_TIPO=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {
        throw;
    }
}

public bool ActualizaEmpleado(int codigo,int area, string nombre, string apellido, string fecha,
int principal)
{
    try
    {
        string tabla = "TB_EMPLEADO";
        string campos = "ID_AREA=" + area + ",NOMBRE_EMPLEADO=" + nombre +
        ",APELLIDO_EMPLEADO=" + apellido + ",FECHA_INGRESO=" + fecha + ",PRINCIPAL=" +
principal;
        string condicion = "ID_EMPLEADO=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {
        throw;
    }
}

public bool ActualizarUbicacion(int codigo, string nombre, string direccion, string fono, int
principal)
{
    try
    {
        string tabla = "TB_UBICACION";

```

```

        string campos = "NOMBRE_UBICACION=" + nombre +
        ",DIRECCION_UBICACION=" + direccion + ",TELEFONO_UBICACION=" + fono +
        ",PRINCIPAL=" + principal;
        string condicion = "ID_UBICACION=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {
        throw;
    }
}

public bool ActualizaActivo(int codigo, int ubicacion, int tipo, string nombre, string modelo,
string serial, double valor, int area)
{
    try
    {
        string tabla = "TB_ACTIVOS";
        string campos = "ID_UBICACION=" + ubicacion + ",ID_TIPO=" + tipo +
        ",NOMBRE_ACTIVO=" + nombre + ",MODELO=" + modelo + ", SERIAL=" + serial +
        ",VALOR=" + valor + ",ID_AREA=" + area;
        string condicion = "ID_ACTIVO=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {
        throw;
    }
}

public bool ActualizaCustodio(int codigo, string nombre, int estado)
{
    try
    {
        string tabla = "TB_CUSTODIO";
        string campos = "NOMBRE_CUSTODIO=" + nombre + ",ID_CARGO=" + estado;
        string condicion = "ID_CUSTODIO=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {
        throw;
    }
}

public bool ActualizaEstado(int codigo, string nombre)
{
    try
    {
        string tabla = "TB_ESTADO";
        string campos = "NOM_ESTADO=" + nombre + """;
        string condicion = "ID_ESTADO=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {

```

```

        throw;
    }
}

public bool ActualizaTiempoUtil(int codigo, int tipo,int activo, int tiempo,int depreciacion)
{
    try
    {
        string tabla = "TB_TIEMPO_UTIL";
        string campos = "DEPRECIACION=" + depreciacion;
        string condicion = "ID_TIEMPO=" + codigo;

        return CN.actualizar(tabla, campos, condicion);
    }
    catch
    {
        throw;
    }
}

public bool ValidaCedula(string cedula)
{
    string[] digitos = new string[10];
    bool valido = true;
    int suma = 0;
    int digito = 0;
    int residuo = 0;

    if (cedula.Length != 10)
        valido = false;
    else
    {
        if (Convert.ToInt32(cedula.Substring(0, 2)) < 1 || Convert.ToInt32(cedula.Substring(0, 2)) >
22)
            valido = false;
        else
        {
            for (int i = 0; i < 10; i++)
            {
                if (i < 9) /
                {
                    if (i % 2 == 0)
                    {
                        if (Convert.ToInt32(cedula.Substring(i, 1)) * 2 >= 10)
                        {
                            suma += Convert.ToInt32((Convert.ToInt32(cedula.Substring(i, 1)) *
2).ToString().Substring(0, 1)) + Convert.ToInt32((Convert.ToInt32(cedula.Substring(i, 1)) *
2).ToString().Substring(1, 1));
                        }
                        else
                            suma += Convert.ToInt32(cedula.Substring(i, 1)) * 2;
                    }
                    else
                        suma += Convert.ToInt32(cedula.Substring(i, 1));
                }
                digitos[i] = cedula.Substring(i, 1);
            }

            residuo = suma % 10;

            if (residuo != 0)

```

```

        digito = 10 - residuo;

        if (digito.ToString() != digitos[9]) /
            valido = false;
    }
}

return valido;
}
}
}
}

```

USUARIO

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb.Administracion
{
    public partial class Usuarios : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                {
                    DataTable dtPerfil = nego.ListadoPerfiles();

                    ddlPerfil.DataSource = dtPerfil;
                    ddlPerfil.DataValueField = "ID_PERFIL";
                    ddlPerfil.DataTextField = "NOMBRE_PERFIL";
                    ddlPerfil.DataBind();

                    Fill_Grid();
                }
            }
            catch (Exception ex)
            {
                Response.Redirect("Error.aspx?error=" + ex.Message);
            }
        }

        private void Fill_Grid()
        {
            try
            {

```

```

        DataTable dt = nego.ListadoUsuarios();

        grUsers.DataSource = dt;
        grUsers.DataBind();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }

}

protected void bt_Add_Click(object sender, EventArgs e)
{
    try
    {
        string sql = "INSERT INTO [dbo].[TB_USUARIOS] ([ID_USUARIO], [ID_PERFIL],
[NOMBRE_USUARIO], [APELLIDO_USUARIO], [CLAVE_USUARIO]) "
+ "VALUES (" + txtUsuario.Text + "," + ddlPerfil.SelectedValue + "," + txtNombre.Text
+ "," + txtApellido.Text + "," + txtPwd.Text + ")";

        nego.InsertaUsuario(sql);

        Fill_Grid();
    }
    catch (Exception ex)
    {
        string[] msg = ex.Message.Split('.');
        Response.Redirect("Error.aspx?error=" + msg[0] + " " + msg[1]);
    }
}

protected void grUsers_RowDataBound(object sender, GridViewRowEventArgs e)
{
    try
    {

        DataTable dtPerfil = nego.ListadoPerfiles();

        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            Label lblPerfil = (Label)e.Row.FindControl("lblPerfil");

            if (lblPerfil != null)
            {
                int perfil = Convert.ToInt32(lblPerfil.Text);
                lblPerfil.Text = (string)nego.NombrePerfil(perfil);
            }

            DropDownList cmbPerfil = (DropDownList)e.Row.FindControl("cmbPerfil");
            if (cmbPerfil != null)
            {
                cmbPerfil.DataSource = dtPerfil;
                cmbPerfil.DataTextField = "NOMBRE_PERFIL";
            }
        }
    }
}

```

```

        cmbPerfil.DataValueField = "ID_PERFIL";
        cmbPerfil.DataBind();
        cmbPerfil.SelectedValue = grUsers.DataKeys[e.Row.RowIndex].Values[1].ToString();
    }
}
}
catch (Exception ex)
{
    Response.Redirect("Error.aspx?error=" + ex.Message);
}
}

protected void grUsers_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
{
    try
    {
        grUsers.EditIndex = -1;
        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grUsers_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    try
    {
        Label lblId = (Label)grUsers.Rows[e.RowIndex].FindControl("lblId");
        TextBox txtName = (TextBox)grUsers.Rows[e.RowIndex].FindControl("txtName");
        TextBox txtLastName =
        (TextBox)grUsers.Rows[e.RowIndex].FindControl("txtLastName");
        DropDownList cmbPerfil =
        (DropDownList)grUsers.Rows[e.RowIndex].FindControl("cmbPerfil");
        TextBox txtPass = (TextBox)grUsers.Rows[e.RowIndex].FindControl("txtPass");

        nego.ActualizaUsuario(lblId.Text, Convert.ToInt32(cmbPerfil.SelectedValue),
        txtName.Text, txtLastName.Text, txtPass.Text);

        grUsers.EditIndex = -1;
        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grUsers_RowEditing(object sender, GridViewEditEventArgs e)
{
    try
    {
        grUsers.EditIndex = e.NewEditIndex;

        Fill_Grid();
    }
}

```

```

        catch (Exception ex)
        {
            Response.Redirect("Error.aspx?error=" + ex.Message);
        }
    }
}
}

```

UBICACIÓN

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb.Administracion
{
    public partial class Ubicacion : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                    Fill_Grid();
            }
            catch (Exception ex)
            {
                Response.Redirect("Error.aspx?error=" + ex.Message);
            }
        }

        protected void bt_Add_Click(object sender, EventArgs e)
        {
            try
            {
                string nombre = txtNombre.Text;
                string direc = txtDireccion.Text;
                string fono = txtFono.Text;
                int prefe;

                if (chkPrincipal.Checked)
                    prefe = 1;
                else
                    prefe = 0;

                nego.InsertaUbicacion(nombre, direc, fono, prefe);

                Fill_Grid();
            }
            catch (Exception ex)

```

```

    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

private void Fill_Grid()
{
    try
    {
        DataTable dt = nego.ListadoUbicacion();

        grUbicacion.DataSource = dt;
        grUbicacion.DataBind();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grUbicacion_RowCancelingEdit(object sender, GridViewCancelEventArgs e)
{
    try
    {
        grUbicacion.EditIndex = -1;
        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grUbicacion_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    try
    {
        int main = 0;
        Label lblId = (Label)grUbicacion.Rows[e.RowIndex].FindControl("lblId");
        TextBox txtName = (TextBox)grUbicacion.Rows[e.RowIndex].FindControl("txtName");
        TextBox txtAdress = (TextBox)grUbicacion.Rows[e.RowIndex].FindControl("txtAdress");
        TextBox txtPhone = (TextBox)grUbicacion.Rows[e.RowIndex].FindControl("txtPhone");
        CheckBox chkMain =
(CheckBox)grUbicacion.Rows[e.RowIndex].FindControl("chkMain");

        if (chkMain.Checked) main = 1;

        nego.ActualizarUbicacion(Convert.ToInt32(lblId.Text), txtName.Text, txtAdress.Text,
txtPhone.Text, main);

        grUbicacion.EditIndex = -1;

        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

```

```

    }

    protected void grUbicacion_RowEditing(object sender, GridViewEditEventArgs e)
    {
        try
        {
            grUbicacion.EditIndex = e.NewEditIndex;

            Fill_Grid();
        }
        catch (Exception ex)
        {
            Response.Redirect("Error.aspx?error=" + ex.Message);
        }
    }
}
}

```

TRANSFERIR ACTIVO

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb
{
    public partial class TransferActivos : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                {
                    DataTable dtE = nego.ListadoEmpleado();

                    ddlEmpleado.DataSource = dtE;
                    ddlEmpleado.DataTextField = "EMPLEADO";
                    ddlEmpleado.DataValueField = "ID_EMPLEADO";
                    ddlEmpleado.DataBind();
                }
            }
            catch (Exception ex)
            {
                Response.Redirect("Error.aspx?error=" + ex.Message);
            }
        }

        protected void bt_Search_Click(object sender, EventArgs e)
        {
            try
            {

```

```

int empleado = Convert.ToInt32(ddlEmpleado.SelectedValue);

DataTable dtE = nego.ListadoEmpleado(empleado);

ddlNewEmpleado.DataSource = dtE;
ddlNewEmpleado.DataTextField = "EMPLEADO";
ddlNewEmpleado.DataValueField = "ID_EMPLEADO";
ddlNewEmpleado.DataBind();

DataTable dtAsg = nego.AsignacionActivo(empleado);

if (dtAsg.Rows.Count > 0)
{
    ListBox1.DataSource = dtAsg;
    ListBox1.DataValueField = "CODIGO";
    ListBox1.DataTextField = "ASIGNACION";
    ListBox1.DataBind();

    ListBox1.Visible = true;
    ListBox2.Visible = true;
    Button1.Visible = true;
    //Button2.Visible = true;
    ddlNewEmpleado.Visible = true;
    bt_Tran.Visible = true;
}

}

catch (Exception ex)
{
    Response.Redirect("Error.aspx?error=" + ex.Message);
}

}

protected void Button1_Click(object sender, EventArgs e)
{
    foreach (ListItem item in ListBox1.Items)
    {
        if (item.Selected)
        {
            string codigo = item.Value;
            string valor = item.Text;
            string coempleado = ddlNewEmpleado.SelectedValue;
            string nomempleado = ddlNewEmpleado.SelectedItem.ToString();
            string motivo = txtMotivo.Text;
            string[] codigos = codigo.Split('_');
            string[] valores = valor.Split('_');

            string newcode = codigos[0] + "_" + codigos[1] + "_" + coempleado + "_" + motivo;
            string newvalor = nomempleado + "_" + valores[1] + "_" + motivo;

            ListBox2.Items.Insert(0, new ListItem(newvalor, newcode));
        }
    }
}

```



```

clsNegocio nego = new clsNegocio();

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        if (!IsPostBack)
        {
            ddlAnio.SelectedValue = "10";
            Fill_Grid();
        }
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void bt_Add_Click(object sender, EventArgs e)
{
    try
    {
        string nombre = nego.NombreTipoActivo(txtNombre.Text);
        string ident = nego.NombreIdenTipoActivo(txtIdent.Text);

        if (nombre != "")
        {
            if (ident != "")
            {
                nego.InsertarTipoActivo(txtNombre.Text, txtIdent.Text,
Convert.ToInt32(ddlAnio.SelectedValue), Convert.ToInt32(txtDeprecia.Text));

                Fill_Grid();
                lblMsg.Text = "";
                lblMsg.Visible = false;
            }
            else
            {
                lblMsg.Text = "Ya existe un tipo de activo con este identificador";
                lblMsg.Visible = true;
            }
        }
        else
        {
            lblMsg.Text = "Ya se ha ingresado este tipo de activo";
            lblMsg.Visible = true;
        }
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

private void Fill_Grid()
{
    try
    {
        DataTable dt = nego.ListadoTipoActivo();
    }
}

```

```

        grTipo.DataSource = dt;
        grTipo.DataBind();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grTipo_RowDataBound(object sender, GridViewRowEventArgs e)
{

}

protected void grTipo_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
{
    try
    {
        grTipo.EditIndex = -1;
        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grTipo_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    try
    {

        Label lblId = (Label)grTipo.Rows[e.RowIndex].FindControl("lblId");
        TextBox txtName = (TextBox)grTipo.Rows[e.RowIndex].FindControl("txtName");

        TextBox txtIdenti = (TextBox)grTipo.Rows[e.RowIndex].FindControl("txtIdenti");

        //TextBox txtDepr = (TextBox)grTipo.Rows[e.RowIndex].FindControl("txtDepr");

        //nego.ActualizarTipoActivo(Convert.ToInt32(lblId.Text), txtName.Text,
txtIdenti.Text, Convert.ToInt32(txtDepr.Text));
        nego.ActualizarTipoActivo(Convert.ToInt32(lblId.Text), txtName.Text, txtIdenti.Text, 0);

        grTipo.EditIndex = -1;

        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grTipo_RowEditing(object sender, GridViewEditEventArgs e)
{
    try
    {

```



```

    }
}

protected void bt_search_Click(object sender, EventArgs e)
{
    try
    {
        DataTable dt = nego.ReporteDepreciacion(Convert.ToInt32(ddlTipo.Selected.Value),
        Convert.ToInt32(ddlActivo.Selected.Value));

        if (dt.Rows.Count > 0)
        {
            grDeprecia.DataSource = dt;
            grDeprecia.DataBind();
            grDeprecia.Visible = true;
            lblMsg.Visible = false;
            bt_Excel.Visible = true;
        }
        else
        {
            grDeprecia.Visible = false;
            lblMsg.Text = "Aun no se ha ingresado la depreciación de este activo";
            lblMsg.Visible = true;
            bt_Excel.Visible = false;
        }
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void bt_new_Click(object sender, EventArgs e)
{
    Response.Redirect("rptDepreciacion.aspx");
}

protected void bt_Excel_Click(object sender, ImageClickEventArgs e)
{
    Response.Clear();
    Response.Buffer = true;
    Response.AddHeader("content-disposition", "attachment;filename=GridViewExport.xls");
    Response.Charset = "";
    Response.ContentType = "application/vnd.ms-excel";

    using (StringWriter sw = new StringWriter())
    {
        HtmlTextWriter hw = new HtmlTextWriter(sw);

        //grDeprecia.AllowPaging = false;

        grDeprecia.HeaderRow.BackColor = Color.White;

        foreach (TableCell cell in grDeprecia.HeaderRow.Cells)
        {
            cell.BackColor = grDeprecia.HeaderStyle.BackColor;
        }
    }
}

```

```

foreach (GridViewRow row in grDeprecia.Rows)
{
    row.BackColor = Color.White;
    foreach (TableCell cell in row.Cells)
    {
        if (row.RowIndex % 2 == 0)
        {
            cell.BackColor = grDeprecia.AlternatingRowStyle.BackColor;
        }
        else
        {
            cell.BackColor = grDeprecia.RowStyle.BackColor;
        }
        cell.CssClass = "phone";
    }
}

//grDeprecia.RenderControl(hw);

d.RenderControl(hw);

string style = @"<style> .textmode { mso-number-format:\@; } </style>";
Response.Write(style);
Response.Output.Write(sw.ToString());
Response.Flush();
Response.End();

}
}

public override void VerifyRenderingInServerForm(Control control)
{
    /* Verifies that the control is rendered */
}

}
}

```

PERFILES

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb.Administración
{
    public partial class Perfiles : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {

```

```
        Fill_Grid();

    }

    private void Fill_Grid()
    {

        DataTable dtPerfil = nego.ListadoPerfiles();

        grPerfil.DataSource = dtPerfil;
        grPerfil.DataBind();

    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        bool inserta = nego.InsertaPerfil(txtPerfil.Text);

    }

}
}
```

INGRESO ACTIVO

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb
{
    public partial class IngresoActivo : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                {

                    DataTable dtT = nego.ListadoTipoActivo();

                    ddlTipo.DataSource = dtT;
                    ddlTipo.DataTextField = "NOMBRE_TIPO";
                    ddlTipo.DataValueField = "ID_TIPO";
                    ddlTipo.DataBind();
                }
            }
        }
    }
}
```

```

        DataTable dtA = nego.ListadoActivoVigente();

        ddlActivo.DataSource = dtA;
        ddlActivo.DataTextField = "ACTIVO";
        ddlActivo.DataValueField = "ID_ACTIVO";
        ddlActivo.DataBind();

        DataTable dtE = nego.ListadoEstados();

        ddlEstado.DataSource = dtE;
        ddlEstado.DataTextField = "NOM_ESTADO";
        ddlEstado.DataValueField = "ID_ESTADO";
        ddlEstado.DataBind();

        FillGrid();
    }
}
catch (Exception ex)
{
    Response.Redirect("Error.aspx?error=" + ex.Message);
}
}

protected void btregistrar_Click(object sender, EventArgs e)
{
    try
    {
        nego.IngresoActivo(Convert.ToInt32(ddlTipo.SelectedValue),
Convert.ToInt32(ddlActivo.SelectedValue), Convert.ToInt32(ddlEstado.SelectedValue),
txtObservacion.Text);

        FillGrid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

private void FillGrid()
{
    try
    {
        DataTable dt = nego.ListadoEntrada();

        grIngresa.DataSource = dt;
        grIngresa.DataBind();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}
}
}
}

```

EMPLEADOS

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb
{
    public partial class Empleados : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();
        protected void Page_Load(object sender, EventArgs e)
        {
            txtIngresa.Attributes.Add("readonly", "readonly");

            if (!IsPostBack)
            {
                DataTable dt = nego.ListadoAreas();

                ddlArea.DataSource = dt;
                ddlArea.DataTextField = "NOMBRE_AREA";
                ddlArea.DataValueField = "ID_AEREA";
                ddlArea.DataBind();

                FillGrid();
            }
        }

        private void FillGrid()
        {
            DataTable dt = nego.ListadoEmpleados();

            grEmpleado.DataSource = dt;
            grEmpleado.DataBind();
        }

        protected void bt_add_Click(object sender, EventArgs e)
        {
            int principal;

            if (chkPrincipal.Checked)
                principal = 1;
            else
                principal = 0;

            bool valido = nego.ValidaCedula(txtCedula.Text);

            if (valido)
```

```

{
    if (!nego.ExisteEmpleado(txtCedula.Text))
    {
        nego.InsertarEmpleado(Convert.ToInt32(ddlArea.SelectedValue),
Convert.ToInt64(txtCedula.Text), txtNombre.Text, txtApellido.Text, txtIngresa.Text, principal);

        FillGrid();
    }
    else
    {
        lblMsg.Text = "Empleado ya existe";
        lblMsg.Visible = true;
    }
}
else
{
    lblMsg.Text = "Cedula Invalida";
    lblMsg.Visible = true;
}
}

protected void grEmpleado_RowDataBound(object sender, GridViewRowEventArgs e)
{
    try
    {
        DataTable dtA = nego.ListadoAreas();

        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            Label lblArea = (Label)e.Row.FindControl("lblArea");

            if (lblArea != null)
            {
                int area = Convert.ToInt32(lblArea.Text);
                lblArea.Text = (string)nego.NombreArea(area);
            }

            DropDownList cmbArea = (DropDownList)e.Row.FindControl("cmdArea");
            if (cmbArea != null)
            {
                cmbArea.DataSource = dtA;
                cmbArea.DataTextField = "NOMBRE_AREA";
                cmbArea.DataValueField = "ID_AREA";
                cmbArea.DataBind();
                cmbArea.SelectedValue =
grEmpleado.DataKeys[e.Row.RowIndex].Values[4].ToString();
            }
        }
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grEmpleado_RowCancelingEdit(object sender, GridViewCancelEventArgs e)

```

```

    {
        grEmpleado.EditIndex = -1;
        FillGrid();
    }

    protected void grEmpleado_RowEditing(object sender, GridViewEditEventArgs e)
    {
        grEmpleado.EditIndex = e.NewEditIndex;
        FillGrid();
    }

    protected void grEmpleado_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        int main = 0;

        Label lblId = (Label)grEmpleado.Rows[e.RowIndex].FindControl("lblId");
        DropDownList cmbArea =
        (DropDownList)grEmpleado.Rows[e.RowIndex].FindControl("cmbArea");
        TextBox txtName = (TextBox)grEmpleado.Rows[e.RowIndex].FindControl("txtName");
        TextBox txtLast = (TextBox)grEmpleado.Rows[e.RowIndex].FindControl("txtLast");
        TextBox txtDate = (TextBox)grEmpleado.Rows[e.RowIndex].FindControl("txtDate");
        CheckBox chkMain = (CheckBox)grEmpleado.Rows[e.RowIndex].FindControl("chkMain");

        if (chkMain.Checked) main = 1;

        nego.ActualizaEmpleado(Convert.ToInt32(lblId.Text),
        Convert.ToInt32(ddlArea.SelectedValue), txtName.Text, txtLast.Text, txtDate.Text, main);
        grEmpleado.EditIndex = -1;
        FillGrid();
    }
}
}
}

```

BAJA DE ACTIVOS

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb
{
    public partial class BajaActivos : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                {
                    Fill_Grid();
                }
            }
        }
    }
}

```

```

        catch
        {
            throw;
        }
    }

    protected void bt_Baja_Click(object sender, EventArgs e)
    {
        int activo = Convert.ToInt32(ddlActivo.SelectedValue);

        nego.BajaActivos(activo,txtMotivo.Text);

        Fill_Grid();
    }

    private void Fill_Grid()
    {
        try
        {
            DataTable dt = nego.ListadoBajas();
            grBajas.DataSource = dt;
            grBajas.DataBind();
        }
        catch
        {
            throw;
        }
    }
}

```

ESTADOS

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Activos.Negocio;

namespace ActivosWeb.Administracion
{
    public partial class Estados : System.Web.UI.Page
    {
        clsNegocio nego = new clsNegocio();

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                    Fill_Grid();
            }
            catch (Exception ex)
            {
                Response.Redirect("Error.aspx?error=" + ex.Message);
            }
        }
    }
}

```

```

}

protected void bt_add_Click(object sender, EventArgs e)
{
    try
    {
        nego.InsertaEstado(txtNombre.Text);

        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

private void Fill_Grid()
{
    try
    {
        DataTable dt = nego.ListadoEstados();

        grEstado.DataSource = dt;
        grEstado.DataBind();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grEstado_RowCancelingEdit(object sender, GridViewCancelEventArgs e)
{
    try
    {
        grEstado.EditIndex = -1;
        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}

protected void grEstado_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    try
    {
        Label lblId = (Label)grEstado.Rows[e.RowIndex].FindControl("lblId");
        TextBox txtName = (TextBox)grEstado.Rows[e.RowIndex].FindControl("txtName");

        nego.ActualizaEstado(Convert.ToInt32(lblId.Text), txtName.Text);

        grEstado.EditIndex = -1;
        Fill_Grid();
    }
    catch (Exception ex)

```

```
{
    Response.Redirect("Error.aspx?error=" + ex.Message);
}

protected void grEstado_RowEditing(object sender, GridViewEditEventArgs e)
{
    try
    {
        grEstado.EditIndex = e.NewEditIndex;

        Fill_Grid();
    }
    catch (Exception ex)
    {
        Response.Redirect("Error.aspx?error=" + ex.Message);
    }
}
}
```

Bibliografía

<http://www.monografias.com/trabajos14/implantacion-datos/implantacion-datos.shtml>

http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:modelo4_1.pdf

<http://es.slideshare.net/cliceduca/pruebas-de-software-2420588>

<http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>

<http://www.monografias.com/trabajos14/implantacion-datos/implantacion-datos.shtml>

www.oocities.org/.../Documento_de_Estandares_de_Programacion_Final.

https://www.google.com.ec/.../diagrama_archivos_insumo.

<https://www.google.com.ec/.../documentacion-arquitectura>.

<http://aprenderaprogramar.com/.../curso-aprender-a-programar-visual-basic-desde-cero>