

PRÁCTICA EMPRESARIAL EN GRUPO ANT & CIA. LTDA.

**MIGRACIÓN DEL SISTEMA ADMINISTRATIVO INTELIGENTE TIMAX
A LA NUEVA PLATAFORMA .NET**

ELVER OMAR GALLO LANCHEROS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2009

PRÁCTICA EMPRESARIAL EN GRUPO ANT & CIA. LTDA.

**MIGRACIÓN DEL SISTEMA ADMINISTRATIVO INTELIGENTE TIMAX
A LA NUEVA PLATAFORMA .NET**

ELVER OMAR GALLO LANCHEROS

**Trabajo de grado realizado como cumplimiento de los requisitos para
optar por el título de Ingeniero de Sistemas**

MSc. JORGE HERRERA CASTILLO

**Director Práctica Empresarial
Magister en Informática**

Ing. FREDY RAMIRO SÁNCHEZ SIERRA

**Tutor Práctica Empresarial
Jefe de Sistemas**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2009

DEDICATORIA

A Dios que siempre ha sido mi compañero incondicional durante toda mi vida, gracias a ti Señor por permitirme disfrutar de este éxito que con tanto esfuerzo lo he conseguido.

A ti Yadira, mi Bebe, que apareciste en el momento más importante de mi vida, Gracias por brindarme ese gran amor que es el motor de mi vida, por su comprensión, perseverancia y paciencia conmigo, cada mañana doy gracias a la vida por haberme regalado una mujer tan maravillosa, gracias por los momentos tan hermosos que juntos hemos vivido y por esta familia que apenas comienza.

A mi madre Flor María, a quien le debo lo que soy, por ese gran apoyo que tuve durante todos estos años que estuviste a mi lado, por sus consejos y cariño tan especial que tuviste conmigo.

A mi padre Ceferino, en donde tú estés, este título va dedicado a ti, por su ejemplo de trabajo y disciplina que forjaste en mi, que Dios te tenga en su gloria.

A mis hermanos Freddy y Maye, los extraño mucho, y gracias por todos los momentos tan maravillosos que juntos compartimos, siempre los llevaré en mi corazón.

AGRADECIMIENTOS

Agradezco al profesor Jorge Herrera Castillo, director del proyecto por apoyarme con sus sabias ideas en el desarrollo del proyecto.

A la empresa GRUPO A.N.T. & Cía. Ltda., en especial a los ingenieros Fredy Ramiro Sánchez Sierra, Martín Eduardo Suárez Arias y Luis Alberto Suárez Cordero, por brindarme la confianza para realizar este trabajo y permitirme representar a su organización, gracias por su apoyo sobre todo en los momentos difíciles y por su esfuerzo para lograr la realización de esta práctica.

A Beto, por su valiosa colaboración en esta etapa del proyecto, a mis entrañables amigos Pacho, Pedro, Wilson, Roque, Cesar, amigos que me acompañaron durante diferentes etapas de mi vida, y me brindaron su apoyo incondicional y la energía para seguir adelante, esto va dedicado a ustedes.

A todas esas personas, familiares, amigos y compañeros que de alguna u otra forma intervinieron para lograr esta meta, a todos ellos un infinito agradecimiento.

CONTENIDO

	Pág.
INTRODUCCIÓN	1
1 DESCRIPCIÓN DE LA PRÁCTICA	2
1.1. DESCRIPCIÓN DE LA EMPRESA	2
1.1.1. Nombre de la Empresa	2
1.1.2. Quiénes somos	2
1.1.3. Misión de la Empresa	2
1.1.4. Visión de la Empresa	3
1.1.5. Valores Corporativos	3
1.1.6. Estructura Organizacional	4
1.2. DESCRIPCIÓN DEL PROYECTO	5
1.2.1. Planteamiento del problema	5
1.2.2. Objetivos	6
1.2.2.1. Objetivo General	6
1.2.2.2. Objetivos Específicos	6
1.2.3. Justificación	7
1.2.4. Impacto	9
1.2.5. Viabilidad	10
2 MARCO TEÓRICO	11
2.1. PLATAFORMA .NET	11
2.1.1. Orígenes	11
2.1.2. .NET Framework	16
2.1.2.1. Arquitectura del .NET Framework	17
2.1.2.2. Funcionamiento del .NET Framework	19
2.1.3. Herramientas de Desarrollo	20
2.1.3.1. Visual Studio .NET	20
2.1.3.2. Visual Basic .NET	21
2.2. SISTEMA DE BASE DE DATOS	22
2.2.1. Definición de una Base de Datos	22
2.2.2. Niveles de una Base de Datos	23

2.2.3. Propiedades de una Base de Datos	24
2.2.4. Definición de Sistema de Administración de Base de Datos (DBMS)	26
2.2.5. Administrador de Base de Datos (DBA)	28
2.3. POSTGRESQL	28
2.3.1. Antecedentes	28
2.3.2. Arquitectura	29
2.3.3. Características	30
3. ANÁLISIS DEL SISTEMA	34
3.1. METODOLOGÍA	34
4. DESARROLLO DE LA PRÁCTICA	37
4.1. COMPRENDER EL CONTEXTO DEL SISTEMA	37
4.1.1. Descripción de la Aplicación	38
4.2. OBJETIVOS DEL SISTEMA	40
4.3. REQUISITOS DEL SISTEMA	42
4.4. REQUISITOS FUNCIONALES DEL SISTEMA	47
4.4.1. Diagramas de Casos de Uso	47
4.4.2. Definición de Actores del Sistema	57
4.4.3. Casos de Uso del Sistema	58
4.4.4. Modelo de Datos	76
4.4.5. Desarrollo de la Aplicación	77
4.4.5.1. Actividades Desarrolladas	77
4.4.5.1.1. Primer Prototipo	78
4.4.5.1.2. Segundo Prototipo	80
4.4.5.1.3. Tercer Prototipo	81
5. CONCLUSIONES	83
6 RECOMENDACIONES	85
BIBLIOGRAFÍA	86

LISTADO DE FIGURAS

	Pág.
Figura 1. Estructura Organizacional de Grupo A.N.T.	4
Figura 2. Arquitectura del .NET Framework	18
Figura 3. Modelo de ejecución del .NET Framework	19
Figura 4. Imagen simplificada de un Sistema de Base de Datos	22
Figura 5. Imagen de la Arquitectura de PostgreSQL	30
Figura 6. Modelo Evolutivo Incremental	35
Figura 7. Diagrama de Casos de Uso de los Actores del Sistema	47
Figura 8. Diagrama de Subsistemas del Sistema Administrativo Inteligente TIMAX	48
Figura 9. Diagrama de Casos de Uso Configuración de la Aplicación	49
Figura 10. Diagrama de Casos de Uso Gestión de Usuarios	49
Figura 11. Diagrama de Casos de Uso Módulo de Directorio	49
Figura 12. Diagrama de Casos de Uso Módulo de Recursos Humanos	50
Figura 13. Diagrama de Casos de Uso Módulo de Facturación	50
Figura 14. Diagrama de Casos de Uso Módulo de Inventarios	51
Figura 15. Diagrama de Casos de Uso Módulo Cuentas por Cobrar	51
Figura 16. Diagrama de Casos de Uso Módulo de Contabilidad	52
Figura 17. Diagrama de Casos de Uso Módulo Cuentas por Pagar	52
Figura 18. Diagrama de Casos de Uso Módulo de Tesorería	53
Figura 19. Diagrama de Casos de Uso Módulo de Presupuesto	53
Figura 20. Diagrama de Casos de Uso Módulo de Activos Fijos	54
Figura 21. Diagrama de Casos de Uso Módulo Centro de Costos	54
Figura 22. Diagrama de Casos de Uso Módulo de Impuesto Predial	55
Figura 23. Diagrama de Casos de Uso Módulo de Industria y Comercio	55
Figura 24. Diagrama de Casos de Uso de Servicios Públicos	56
Figura 25. Diagrama de Casos de Uso Módulo Buscar	56
Figura 25. Diagrama de Casos de Uso Usuario registrado	56

LISTADO DE TABLAS

	Pág.
Tabla 1. Objetivo del Sistema 01	40
Tabla 2. Objetivo del Sistema 02	40
Tabla 3. Objetivo del Sistema 03	40
Tabla 4. Objetivo del Sistema 04	40
Tabla 5. Objetivo del Sistema 05	40
Tabla 6. Objetivo del Sistema 06	41
Tabla 7. Objetivo del Sistema 07	41
Tabla 8. Objetivo del Sistema 08	41
Tabla 9. Objetivo del Sistema 09	41
Tabla 10. Objetivo del Sistema 10	41
Tabla 11. Requisito del Sistema 01	42
Tabla 12. Requisito del Sistema 02	42
Tabla 13. Requisito del Sistema 03	42
Tabla 14. Requisito del Sistema 04	43
Tabla 15. Requisito del Sistema 05	44
Tabla 16. Requisito del Sistema 06	44
Tabla 17. Requisito del Sistema 07	44
Tabla 18. Requisito del Sistema 08	45
Tabla 19. Requisito del Sistema 09	45
Tabla 20. Requisito del Sistema 10	46
Tabla 21. Requisito del Sistema 11	46
Tabla 22. Definición de Actores del Sistema	57
Tabla 23. Caso de Uso – Creación de la Nueva Empresa	58
Tabla 24. Caso de Uso – Edición de Tablas	58
Tabla 25. Caso de Uso – Parametrización	59
Tabla 26. Caso de Uso – Inicializar Consecutivos	59
Tabla 27. Caso de Uso – Configurar Apariencia	60
Tabla 28. Caso de Uso – Agregar Usuario	60
Tabla 29. Caso de Uso – Editar Usuario	61
Tabla 30. Caso de Uso – Desactivar Usuario	62

Tabla 31. Caso de Uso – Establecer Permisos	63
Tabla 32. Caso de Uso – Crear Ficha	64
Tabla 33. Caso de Uso – Editar Ficha	64
Tabla 34. Caso de Uso – Desactivar Ficha	65
Tabla 35. Caso de Uso – Listar Directorio	65
Tabla 36. Caso de Uso – Asignar Códigos Contables	66
Tabla 37. Caso de Uso – Editar Plan Único de Cuentas (PUC)	67
Tabla 38. Caso de Uso – Realizar Asientos Contables	68
Tabla 39. Caso de Uso – Editar Clientes	69
Tabla 40. Caso de Uso – Editar Proveedores	70
Tabla 41. Caso de Uso – Editar Artículo	71
Tabla 42. Caso de Uso – Editar Activo	72
Tabla 43. Caso de Uso – Editar Grupos	73
Tabla 44. Caso de Uso – Editar Predios	74
Tabla 45. Caso de Uso – Editar Establecimientos	75

LISTADO DE ANEXOS

	Pág.
ANEXO A. DIAGRAMA ENTIDAD – RELACIÓN DE LA BASE DE DATOS	88
ANEXO B. DICCIONARIO DE DATOS DE LA BASE DE DATOS	96

RESUMEN

TITULO: MIGRACIÓN DEL SISTEMA ADMINISTRATIVO INTELIGENTE TIMAX A LA NUEVA PLATAFORMA .NET.

AUTOR: ELVER OMAR GALLO LANCHEROS

PALABRAS CLAVE

TIMAX, Evolutivo Incremental, Postgres, .Net Framework, Visual Studio .NET, UML

DESCRIPCIÓN

La empresa GRUPO A.N.T. & CIA. LTDA., concentra su actividad en el desarrollo de herramientas informáticas de administración y manejo contable para apoyar la gestión de las pequeñas y grandes empresas a nivel nacional mediante el Sistema Administrativo Inteligente TIMAX.

Para establecer el plan de trabajo, y la planificación del proyecto, se adoptó la metodología de ciclo de vida Evolutivo Incremental, el cual permitió alcanzar los requisitos definidos y los resultados esperados; su selección se basó en las condiciones iniciales así como el entorno en que se desarrollarían las diferentes actividades del proyecto.

Así mismo, la aplicación fue diseñada para funcionar en una red interna, con una arquitectura cliente-servidor; y para su desarrollo se utilizó como herramientas, Visual Studio 2005 siendo el lenguaje de programación el Visual Basic .NET y como gestor de Base de Datos el software libre Postgres, aprovechando las funcionalidades y ventajas que cada una de estas proporcionaba.

Durante el desarrollo de la práctica empresarial se aplicaron todos los conocimientos adquiridos durante la formación académica como son: el levantamiento de requisitos, el modelado de los casos de uso a través de los diagramas de UML, el diseño de interfaces, la normalización de tablas, y la aplicación de los estándares de programación, que conllevaron a facilitar el buen desempeño en las diferentes etapas del desarrollo de la Aplicación y a cumplir con el cronograma de actividades.

*Trabajo de Grado

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Universidad Industrial de Santander.

Director. Jorge Herrera Castillo

Tutor. Ingeniero Fredy Ramiro Sánchez Sierra

ABSTRACT

TITLE: MIGRACIÓN DEL SISTEMA ADMINISTRATIVO INTELIGENTE TIMAX A LA NUEVA PLATAFORMA .NET.

AUTHOR: ELVER OMAR GALLO LANCHEROS

KEY WORDS

TIMAX, Evolutivo Incremental, Postgres, .Net Framework, Visual Studio .NET, UML

DESCRIPTION

The company GRUPO A.N.T. & CIA. LTDA., concentrated its activities on developing tools for the administration and management accounting to support the management of small and large businesses nationwide through the Intelligent Management System Timax.

To set the work plan, and planning the project was adopted methodology Incremental Evolutionary life cycle, which helped achieve the defined requirements and expected results, their selection was based on the initial conditions and the environment in which would develop the various project activities.

Also, the application was designed to run on an internal network with a client-server architecture, in addition to its development tools are used as the Visual Studio 2005 is the programming language Visual Basic. NET and as manager of Base Postgres open source data, taking advantage of the features and benefits that each of these proportions.

During the development of business practice applied all the knowledge acquired during academic training, such as: lifting requirements, modeling the use cases using UML diagrams, interface design, normalization of tables, and implementation of programming standards, which led to provide good performance at different stages of implementation and compliance with the schedule of activities.

*Graduation Work

**Physical – Mechanical Engineering's Faculty
System Engineering and Computer Science School. UIS.

Director: Jorge Herrera Castillo.

Tutor: Ingeniero Fredy Ramiro Sánchez Sierra.

INTRODUCCIÓN

La modalidad de proyecto de grado de práctica empresarial, es una excelente oportunidad para el estudiante, de poner en práctica en una empresa todos los conocimientos adquiridos durante su carrera universitaria, en donde se medirá no sólo sus cualidades intelectuales, sino también sus habilidades para reaccionar ante las diferentes situaciones que se le presenten, la facilidad de integrarse y trabajar conjuntamente en equipo, las relaciones con los clientes, entre otras muchas experiencias que le permitirán al estudiante adquirir un concepto claro sobre la vida empresarial.

En los últimos años la tecnología informática ha venido creciendo vertiginosamente, y se hace necesario para cualquier empresa desarrolladora de Software estar al día con las últimas tendencias tecnológicas y así lograr mantenerse en un mercado que cada vez se hace más competitivo; para tal fin la empresa GRUPO A.N.T. & CIA. LTDA., se encuentra en un proceso de adaptación de nuevas herramientas como Visual Studio .NET 2005 que le permitirán desarrollar entornos Cliente-Servidor, al mismo tiempo, está cambiando su manera de desarrollar aplicaciones con la nueva filosofía de la Programación Orientada a Objetos, y por último está migrando sus Bases de Datos a un motor más robusto y confiable como lo es Postgres; de la misma manera, está capacitando al personal para obtener la estandarización en los procesos y metodologías para el uso de estas nuevas herramientas.

El presente documento realiza la recopilación teórica y práctica de los eventos y actividades realizadas durante los seis meses de desarrollo de la práctica empresarial, en donde se documentan todas las experiencias adquiridas por el estudiante practicante, así como las conclusiones, y recomendaciones. Se espera que este proyecto sirva para el progreso y mejora de los productos que esta empresa posee y así contribuir con el desarrollo tecnológico de nuestra región.

1. DESCRIPCIÓN DE LA PRÁCTICA EMPRESARIAL

En el presente capítulo se describen los aspectos generales de la organización en la que se desarrolló la práctica empresarial, tales como la razón social, la reseña histórica y campo de acción, además se presenta una descripción de los objetivos, alcance del proyecto y a su vez el impacto y su viabilidad.

1.1. DESCRIPCIÓN DE LA EMPRESA

1.1.1. Nombre de la Empresa

Razón Social:	Grupo A.N.T. & Cía. Ltda.
Nit:	804.005.980-2
Tipo de Organización:	Sociedad Limitada.
Fecha de fundación:	
Domicilio:	Carrera 36 No.
Ciudad:	Bucaramanga.
Representante Legal:	Ing. Martin Eduardo Suárez Arias.

1.1.2. Quiénes somos

GRUPO A.N.T. & CIA. Ltda., es una empresa santandereana con más de 19 años de experiencia en el desarrollo de herramientas informáticas de administración y manejo contable.

1.1.3. Misión de la Empresa

Desarrollar y comercializar herramientas informáticas de administración y manejo contable mediante gestión de mercadeo, suministrar asistencia técnica oportuna y eficaz para apoyar la gestión administrativa de cualquier tipo de empresa, ya sea comercializadora, de servicios, cooperativa, hospitalaria, manufacturera o gubernamental.

Nuestras aplicaciones son de fácil manejo y total integración entre sus módulos (Facturación, Inventarios, Producción, Cuentas por Cobrar, Cuentas por Pagar, Tesorería, Contabilidad, Activos Fijos, Presupuesto, Centro de Costos, Cargos y Gastos Diferidos), lo que permite al usuario manipular el sistema con sólo conocer los fundamentos contables y administrativos en los que se basa el programa, sin involucrarse con códigos contables ni conceptos tributarios.

1.1.4. Visión de la Empresa

Ser la organización líder a nivel nacional, diseñando y desarrollando buenos productos informáticos y aplicando las mejores tecnologías en el desarrollo de software, con un personal altamente competitivo y actualizado para ofrecer una mayor calidad en el producto y el servicio a los clientes en las áreas administrativas y contables de una empresa.

Consolidarnos, posicionarnos y ser reconocidos por nuestros clientes actuales y potenciales como una empresa con un alto nivel técnico para el soporte administrativo y contable, al mismo tiempo ser una empresa renovadora y creciente orientada a los requerimientos del cliente para satisfacer sus necesidades.

1.1.5. Valores Corporativos

La empresa Grupo A.N.T. & Cía. Ltda. tiene como base fundamental los siguientes valores para asegurar que todos los ejecutivos, empleados y clientes, se comprometan con la excelencia de nuestra actividad empresarial:

Trabajar en equipo, para lograr los objetivos que se establecen, dentro de un ambiente de cordialidad, confianza y respeto mutuo.

El compromiso de los clientes, para actuar con iniciativa y buena disponibilidad y así ayudarlos a lograr sus objetivos empresariales.

La toma de decisiones acertadas para lograr nuevas ventajas empresariales y abrir oportunidades de negocios que permiten demarcar el liderazgo empresarial.

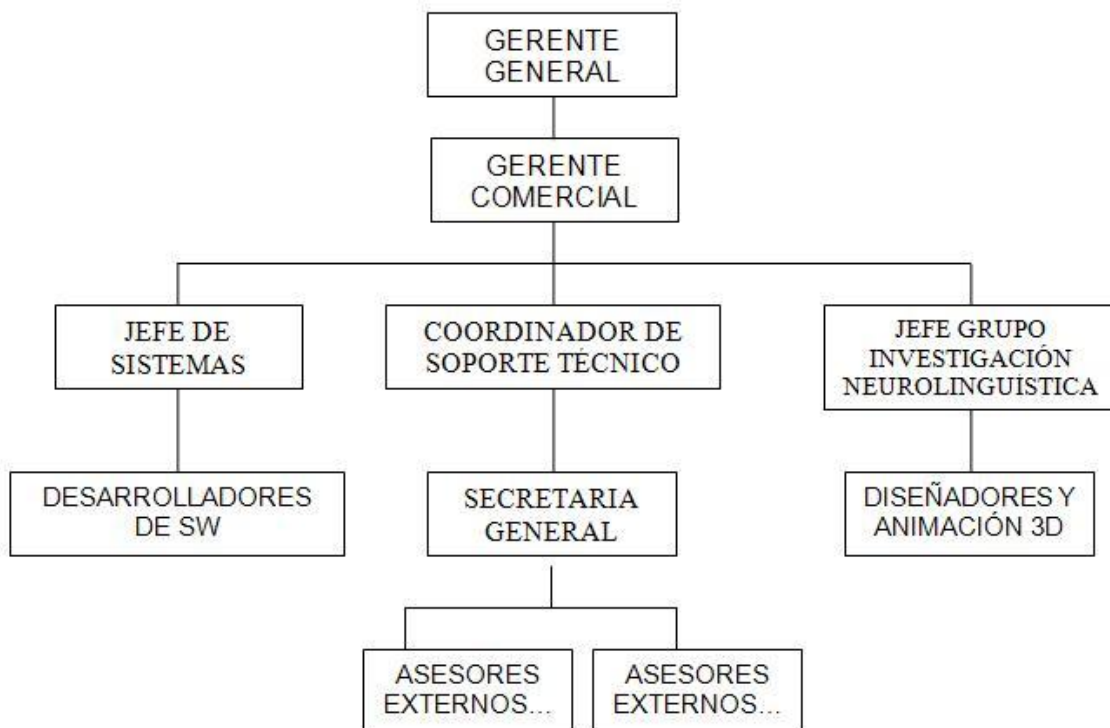
El servicio, debe ser de manera eficaz y con la más alta calidad que sean consistentes con las necesidades y preferencias de los clientes.

La innovación, es aceptar la variedad y la ventaja tecnológica que nos ofrecen hoy en día las nuevas herramientas de desarrollo, para interpretarlas y obtener el mejor provecho en los nuevos retos tecnológicos.

La responsabilidad, actuar de forma coherente con el desarrollo empresarial y la sociedad, para de esta forma enmarcar la operación en una estructura logística que beneficie a todos los actores que hacen parte de nuestro entorno.

1.1.6. Estructura Organizacional

Figura 1. Estructura Organizacional de Grupo A.N.T.



Fuente: El Autor

1.2. DESCRIPCIÓN DEL PROYECTO

1.2.1. Planteamiento del problema

La empresa GRUPO A.N.T. & CIA. LTDA., concentra su actividad en el desarrollo de herramientas informáticas de administración y manejo contable para apoyar la gestión de las pequeñas y grandes empresas a nivel nacional mediante Sistema Inteligente TIMAX.

Este sistema de Información TIMAX, se inició desde 1.986 en versión D.O.S. desarrollado en lenguaje Clíper en donde la información se guardaba en tablas libres DBASE I; a comienzos del año 2005 se lanzó al mercado la versión WINDOWS, en el cual el código fuente se migró al lenguaje XBASE (el Visual de Clíper), y la información se siguió almacenando en tablas libres pero ahora con DBASE IV.

Debido al creciente volumen de información de algunas empresas en donde el TIMAX se encuentra instalado, las tablas han bajado su rendimiento por causa de la redundancia de datos debido a que no existe alguna relación entre ellas, de la misma manera en el caso de ocurrir un fallo de corriente en el momento de alguna transacción, la información se pierde o en el peor de los casos, las tablas involucradas en la transacción se dañan.

En la migración del código fuente del lenguaje Clíper al lenguaje XBASE, se mantuvo el mismo tipo de programación procedimental en el que fue desarrollado inicialmente y algunas falencias de programación que se presentaban en la versión anterior no se detectaron y pasaron a la nueva versión, en donde se han venido corrigiendo a medida que se han encontrado los errores.

Teniendo en cuenta los problemas presentados en las versiones anteriores, en cuanto al manejo de la información se hace necesario desarrollar una Base de

Datos Relacional para corregir la redundancia de información y así mejorar la velocidad en las transacciones y la confiabilidad en los datos.

1.2.2. Objetivos

1.2.2.1. Objetivo General

Migrar el código fuente de los módulos del sistema administrativo inteligente TIMAX hechos en lenguaje XBASE y Visual Fox a la nueva plataforma .NET con lenguaje Visual Basic .NET y migración de la Base de Datos de DBASE a Postgres.

1.2.2.2. Objetivos Específicos

- Migrar los Sistemas de Facturación de: Impuesto Predial, Industria y Comercio, Acueducto, Alcantarillado y Aseo desarrollados en Visual Fox Pro 8 para la nueva plataforma de .NET con el lenguaje de programación Visual Basic .NET.
- Migrar del Sistema Administrativo Inteligente TIMAX desarrollado en XBASE para la nueva plataforma .NET con el lenguaje de programación Visual Basic .NET.
- Desarrollar un nuevo Sistema Relacional de Base de Datos (RDBMS) en el motor de Base de Datos Postgres para integrar los anteriores módulos puesto que se encuentran separados.
- Permitir la introducción y almacenamiento de la información de las fichas correspondientes a cada uno de los módulos en donde se necesita, por ejemplo fichas del directorio en el TIMAX, fichas de predios en Impuesto Predial, fichas de establecimientos en Industria y Comercio, fichas de usuarios y medidores en Acueducto y Alcantarillado.

- Realizar búsquedas de manera dinámica es decir, por usuarios, o por códigos contables, o por facturas, o por comprobantes de contabilidad, o por predios, o por establecimientos.
- Permitir la introducción de los avalúos de los predios emitida por Corporación Agustín Codazzi en el módulo de Impuesto Predial, también los ingresos, retenciones y deducciones de los establecimientos en el módulo de Industria y Comercio, y de igual manera la toma de lecturas de consumo de agua en el módulo de Acueducto y Alcantarillado.
- Generar los procesos de liquidación y facturación en los módulos de Impuesto Predial, Industria y Comercio y en Acueducto y Alcantarillado; así como también crear la interfaz para recibir el pago de las facturas por parte de los usuarios; todos estos movimientos deben estar integrados al módulo de contabilidad, presupuesto y tesorería en el TIMAX.
- Generar toda clase de reportes necesarios en cada módulo como por ejemplo, listado de usuarios, predios, establecimientos, facturas, comprobantes de ingreso, comprobantes de egreso, informe de cartera por edades, saldos, informe de recaudos, auditorías, consumo, novedades, lecturas, deudores morosos, etc. y estos reportes sean generados también en formato Excel.

1.2.3. Justificación

En los últimos años la tecnología informática ha venido creciendo vertiginosamente, y es necesario para cualquier empresa estar al día con las últimas tendencias tecnológicas para así lograr mantenerse en un mercado que a su vez es muy competitivo.

Para tal fin la empresa GRUPO A.N.T. & CIA. LTDA. Se encuentra en un proceso de adaptación de nuevas herramientas como VISUAL STUDIO .NET 2005 que le permitirán desarrollar aplicaciones orientadas a ambiente Windows y por lo tanto necesita personal capacitado que le pueda brindar apoyo en la estandarización en los procesos y metodologías para el uso de estos nuevos programas.

De la misma manera debido a los problemas planteados anteriormente con respecto a la redundancia de los datos, la seguridad y confiabilidad en la información, se hace necesario aplicar la metodología de un Sistema Relacional de Base de Datos, que nos garantiza los siguientes beneficios:

- Integridad en la información, esto es asegurar que los datos de una base de datos estén correctos mediante la implementación de las restricciones de integridad también conocidas como reglas del negocio.
- Reducir la redundancia, al aplicar este mecanismo ayuda a controlar la inconsistencia de la información y el desperdicio de espacio de almacenamiento.
- Manejo de transacciones, una transacción es una unidad de trabajo lógica que comprende varias operaciones a la base de datos, una base de datos Relacional garantiza que exista esta atomicidad en las transacciones.
- Seguridad de la información, asegurar que el único medio de acceso a la base de datos sea a través de los canales adecuados y por lo tanto definir las reglas o restricciones de seguridad.

Por tal motivo la empresa Grupo A.N.T. & Cía. Ltda. optó por vincular a un estudiante en la modalidad de práctica empresarial para que cumpliera con esta labor de adaptación hacia las nuevas tecnologías de información.

1.2.4. Impacto

Técnico

Debido a los problemas anteriormente mencionados, y la disposición de la empresa en buscar nuevas y mejores alternativas para seguir avanzando en el mercado, se hace la presentación de este proyecto, en el cual se busca que a través de la aplicación de nuevas tecnologías, proporcionarle a la empresa GRUPO A.N.T & CIA. LTDA. un avance en su proceso de evolución hacia las nuevas tendencias de desarrollo de software.

Económico

Debido a la existencia de un importante número de usuarios del Sistema Inteligente TIMAX que aún se encuentran trabajando con la versión de D.O.S. y no están interesados en actualizarse a la versión de Windows, basada en XBASE y Visual Fox, esta nueva versión con un Sistema Relacional de Base de Datos en Postgres en donde se mejorará notablemente el rendimiento en las transacciones y los novedosos diseños en la nueva plataforma .NET logrará que muchos usuarios contemplen la posibilidad de cambio.

La empresa deberá invertir en las licencias necesarias para el desarrollo del sistema en la nueva plataforma .NET y en cuanto al motor de Base de Datos no acarreará costo alguno puesto que pertenece a las tecnologías de Software libre.

Social

Varios de los objetivos del proyecto se enfocan en mejorar la calidad del software que desarrolla y comercializa la empresa GRUPO A.N.T & CIA. LTDA., así como también su confiabilidad en el manejo de la información.

1.2.5. Viabilidad

Técnica

La empresa GRUPO A.N.T. & CIA. LTDA. en su larga trayectoria, posee el equipo de hardware y software necesario para el desarrollo e implantación de un Sistema de Información como el propuesto y cuenta también con profesionales calificados y de experiencia en el manejo de herramientas de desarrollo de software.

Económica

La empresa no tendrá que invertir en la compra de equipos especializados para el desarrollo del proyecto, puesto que cuenta con computadores actualizados. Igualmente no tendrá que involucrarse en rigurosos pagos a personal nuevo para el desarrollo porque ella cuenta con su propio grupo de programadores; el costo del proyecto tendrá un incremento por la compra de licencias de las herramientas para el desarrollo del software, este costo se puso a consideración de la empresa y después de un estudio por parte de la junta directiva fué aprobado.

Social

El desarrollo de este proyecto coincide con la misma dirección en que la empresa GRUPO A.N.T. & CIA. LTDA. está avanzando, puesto que la tendencia es a modernizar y actualizar continuamente sus productos, sin perder la forma y los mecanismos para realizar sus procesos, que son los puntos claves que distinguen al Sistema Inteligente TIMAX en su facilidad de uso y manejo en todos sus módulos.

2. MARCO TEÓRICO

2.1. PLATAFORMA .NET

Para conocer el funcionamiento de la plataforma .NET es necesario conocer su filosofía, sus bases y orígenes, también es necesario conocer su motor, el Framework .NET, el cual es el pilar de las herramientas de desarrollo.

2.1.1. Orígenes

La iniciativa de .NET surgió como la convergencia de varios objetivos distintos pero estrechamente unidos, el más importante de los cuales es la necesidad de superar las limitaciones del modelo de programación COM y de encontrar un paradigma común de programación para las aplicaciones Cliente – Servidor y las de Internet.

A pesar del gran avance que ha experimentado la industria del software en los últimos años en términos de herramientas de programación disponibles, tales como lenguajes de desarrollo rápido de aplicaciones (RAD), depuradores y generadores de código; escribir aplicaciones robustas para el ámbito empresarial sigue siendo un trabajo complicado, debido a los siguientes factores:

- **Las inconsistencias de Windows**, la plataforma de Windows ha evolucionado caóticamente en los últimos años. Hemos tenido, por ejemplo, hasta tres modelos de programación distintos para producir aplicaciones para el manejo de gráficos (GDI, DirectX y OpenGL), siendo cada modelo completamente distinto a los demás. También se han producido diferentes modelos de programación para acceder a las bases de datos (DAO, RDO, ADO), siendo cada uno de ellos incompatibles con los demás.

Además el propio Windows se presentaba en diferentes formas (Windows 95, 98, Me, NT, 2000, CE, XP y Vista). No todas las versiones son compatibles con todas las funciones, y el resultado neto es que los diseñadores ignoraban, con frecuencia, determinadas funciones de Windows con la intención de que sus aplicaciones pudieran ser utilizadas por la mayoría de los usuarios.

La seguridad es otra área de preocupación que afecta en gran medida a los diseñadores de las plataformas Windows, en el sentido que deben optar por diferentes niveles de seguridad.

- **Defectos COM**, el Modelo de Objetos de Componentes (COM), ha demostrado ser lo suficientemente versátil como para convertirse en la infraestructura de numerosas aplicaciones distribuidas y empresariales, pero al mismo tiempo, ha demostrado ser excesivamente complejo y peor aún, las aplicaciones COM son inherentemente frágiles porque dependen fuertemente de la información almacenada en el Registro del sistema, que tiene una merecida reputación de poder ser corrompido con facilidad.

Las aplicaciones COM padecen otro serio problema, el control de versiones: cada versión más moderna de un componente COM sobrescribe las versiones más antiguas, por lo que, por ejemplo no podrá tener varias versiones de ADO en el mismo equipo. En la práctica aplicaciones que utilizaban versiones anteriores pueden dejar de repente de funcionar debido a las pequeñas incompatibilidades entre estas versiones.

- **Interoperabilidad entre lenguajes**, la plataforma Windows dificulta enormemente la integración de piezas de código escritas en diferentes lenguajes, porque cada lenguaje cuenta con sus propios convenios de llamada y limitaciones.

Como integrar diferentes lenguajes resulta bastante difícil, muchos diseñadores de software prefieren utilizar un único lenguaje, perdiéndose así las ventajas particulares que cada lenguaje puede ofrecer. Por ejemplo, Visual Basic resulta bastante más productivo para aplicaciones empresariales y de base de datos y, apesada de todo, muchos diseñadores prefieren seguir utilizando C++ debido a su mayor potencia y flexibilidad. Si la integración de lenguajes fuera más sencilla, se podrían desarrollar las distintas partes de una aplicación de gran tamaño utilizando el lenguaje que fuera más apropiado para cada una de ellas y de esta forma el desarrollo de software sería más barato en términos de tiempo y dinero.

Debido a estos problemas que presentan las herramientas de desarrollo de software actuales, la solución que ofrece Microsoft .NET es la siguiente:

- **Potenciar las plataformas Windows**, Microsoft .NET ofrece una visión orientada a objeto del sistema operativo Windows e incluye cientos de clases que encapsulan los objetos más importantes del núcleo de Windows. Por ejemplo, una parte de .NET denominada GDI+ contiene todos los objetos que muestran gráficos en una ventana. Dependiendo de la versión específica de Windows, estos objetos pueden utilizar funciones GDI, DirectX o incluso OpenGL, pero desde el punto de vista del diseñador lo importante es que sólo existe un modelo de programación.

La seguridad está construida directamente sobre .NET por lo que no tendrá que preocuparse demasiado del modelo de seguridad del sistema operativo. La seguridad de .NET es superior a la seguridad de Windows y el administrador puede, entre otras muchas cosas, garantizar o revocar el derecho a acceder al Registro o al sistema de archivos de determinadas aplicaciones. Este modelo de seguridad es independiente de la versión específica de Windows sobre la que se ejecute la aplicación.

Las aplicaciones y componentes .NET son también inherentemente más seguras que los componentes COM y las aplicaciones Windows “al viejo estilo”. Por ejemplo, los componentes .NET se verifican de forma automática para garantizar que su código no ha sido alterado. También podrá firmar una aplicación .NET con una firma digital y utilizar una tecnología como Authenticode para permitir que los usuarios sepan quién es el autor de la aplicación.

- **.NET como mejora de COM**, se puede pensar en .NET como la siguiente generación de COM, ya que estos dos modelos de programación le permiten resolver más o menos los mismos problemas. Sin embargo entre bastidores, son completamente distintos y .NET es superior a COM en muchos aspectos.

En primer lugar, una aplicación .NET puede constar de uno o más Ensamblados. Cada ensamblado suele ser un archivo ejecutable .EXE o DLL. Los ensamblados también pueden incluir otros archivos, tales como DLL adicionales, o .HTML, .gif o cualquier otro archivo no ejecutable. Los ensamblados son una unidad de control de versiones y de implementación en el sentido de que todos los archivos contenidos en el ensamblado tienen el mismo número de versión y no podrá instalar un ensamblado cuyos archivos estén marcados con diferentes números de versiones. Normalmente, una aplicación utiliza varios ensamblados externos, incluyendo aquellos que pertenecen al propio .NET Framework.

.NET no utiliza el Registro para almacenar cualquier información sobre ensamblados (aunque sigue utilizando el Registro para almacenar información sobre el propio .NET Framework). En general toda la información relacionada con una aplicación y con los componentes que utiliza se encuentra almacenada en los archivos de configuración contenidos en el directorio principal de la aplicación. Los archivos de

configuración utilizan la sintaxis XML para almacenar datos de forma jerárquica y, por tanto, son más flexibles que los viejos archivos .INI.

La forma en que .NET resuelve el problema de versiones es el siguiente: existen dos tipos de componentes .NET, los privados y los compartidos. Los componentes privados se encuentran almacenados en el directorio principal de la aplicación (o en uno de sus subdirectorios) y no son visibles para otras aplicaciones. De esta forma, cada aplicación utiliza sus propios componentes privados y las distintas aplicaciones pueden utilizar diferentes versiones de estos componentes porque se encuentran almacenados en diferentes directorios.

Por otro lado, los componentes compartidos son visibles a todas las aplicaciones .NET. Se almacenan normalmente en un almacén central denominado Caché de ensamblados global (GAC) localizado en el directorio C:\Windows\Assembly. Cada componente contenido en el GAC se encuentra almacenado en un directorio independiente y en el mismo equipo pueden coexistir diferentes versiones del mismo componente compartido. Cada aplicación compilada con determinada versión y de un determinado componente continuará funcionando correctamente incluso si el usuario instala una versión más moderna (o más antigua) del mismo componente.

- **Todos los lenguajes de .NET son iguales**, .NET desplaza la mayoría de su funcionalidad desde el lenguaje de programación al propio .NET Framework. Por ejemplo, .NET Framework dispone de clases para abrir, leer y escribir archivos de texto, de la misma manera, también tiene clases para crear ventanas y controles, por lo que no existe ningún motivo para que estas funciones estén incluidas en los propios lenguajes de programación. Entonces todos los lenguajes de programación .NET pueden utilizar estas clases y, por tanto, todos los lenguajes tendrán la misma capacidad de crear aplicaciones con una muy buena interfaz.

Microsoft ha incluido varios lenguajes .NET, entre ellos están Visual Basic .NET, C#,C++, J#, entre otros.

2.1.2. .NET Framework

El .NET Framework (traducido como “Marco de Trabajo”) es el componente fundamental de la plataforma Microsoft .NET, necesario tanto para poder desarrollar aplicaciones como para poder ejecutarlas luego en entornos de prueba o producción.

El .NET Framework tiene tres variantes principales, todas descargables gratuitamente desde Internet

- **.NET Framework Redistributable Package:** este es el mínimo componente de la plataforma .NET que se necesita para poder ejecutar aplicaciones. Normalmente ésta es la variante que se instala en los entornos productivos, una vez que el desarrollo y las pruebas de la aplicación han finalizado.

Está compuesto por:

- El entorno de ejecución de la plataforma .NET
- Las bibliotecas de funcionalidad reutilizable
- **.NET Framework SDK:** esta versión contiene herramientas de desarrollo de línea de comandos (compiladores, depuradores, entre otros), documentación de referencia, ejemplos y manuales para desarrolladores de aplicaciones. Normalmente ésta variante se instala en los entornos de desarrollo de aplicaciones, y es más útil a los programadores que a los usuarios finales. Para poder instalar la versión SDK (Software Development Kit) es necesario instalar previamente el Redistributable Package.
- **.NET Compact Framework:** esta es una versión reducida del .NET Framework Redistributable, especialmente pensada para ser instalada en dispositivos móviles como Pocket PC y SmartPhones.

El .NET Framework puede ser instalado en cualquier sistema operativo de la familia Windows superior a Windows 98.

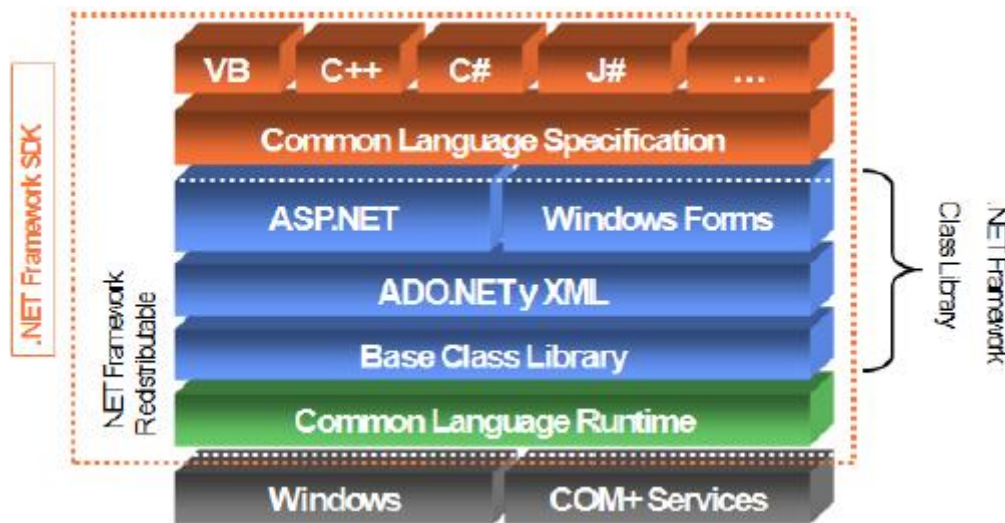
2.1.2.1. Arquitectura del .NET Framework

En la figura No.2 se pueden apreciar las distintas partes que componen al .NET Framework, incluidas el entorno de ejecución de aplicaciones (CLR, en verde), el conjunto de bibliotecas de funcionalidad reutilizable (.NET Framework Class Library, en azul) y los compiladores y herramientas de desarrollo para los lenguajes .NET (en rojo). Todos estos componentes se ubican por encima de la familia de sistemas operativos Windows.

Dentro del conjunto de la .NET Framework Class Library se distinguen 4 sub-componentes principales:

- La Base Class Library (BCL - Biblioteca de Clases Base), que contiene la funcionalidad más comúnmente utilizada para el desarrollo de todo tipo de aplicaciones. Algunos ejemplos de la funcionalidad provista por la BCL son el manejo de colecciones, cadenas de texto, entrada/salida, threading o hilos, operaciones matemáticas y dibujos 2D.
- ADO.NET, que contiene un conjunto de clases que permiten interactuar con bases de datos relacionales y documentos XML como repositorios de información persistente.
- ASP.NET, que constituye la tecnología dentro del .NET Framework para construir aplicaciones con interfaz de usuario Web (es decir, aplicaciones cuya lógica se encuentra centralizada en uno o varios servidores y que los clientes pueden acceder usando un browser o navegador mediante una serie de protocolos y estándares como HTTP y HTML).
- Windows Forms (o simplemente WinForms), que constituye la tecnología dentro del .NET Framework que permite crear aplicaciones con interfaz de usuario basada en formularios y ventanas Windows de funcionalidad rica y que se ejecutan directamente en los clientes.

Figura 2. Arquitectura del .NET Framework



Fuente: Material Desarrollador Cinco Estrellas. Microsoft

Common Language Runtime (CLR): actúa como un motor de ejecución de aplicaciones y componentes manejados.

Tiene las siguientes características:

- Compilación Just-In-Time (JIT), se encarga de compilar las aplicaciones .NET a código de máquina nativo para el sistema operativo y la plataforma de hardware en la que se está ejecutando.
- Gestión automática de memoria (Garbage Collector).
- Gestión de errores consistente (Excepciones).
- Ejecución basada en componentes (Assemblies), todas las aplicaciones .NET son empaquetadas en componentes reutilizables denominados genéricamente Assemblies.
- Gestión de Seguridad.
- Multithreading, provee un entorno de ejecución multi-hilos por sobre las capacidades del Sistema Operativo

Base Class Library (BCL): es la Biblioteca de Clases del .NET Framework, que provee cientos de tipos básicos (clases e interfaces principalmente) orientados a objetos, extensibles mediante herencia.

Los tipos están organizados en jerarquías lógicas de nombres, denominados NAMESPACEs. Los tipos son independientes del lenguaje de desarrollo.

Common Language Specification (CLS): Uno de los objetivos de diseño de la plataforma .NET fue el ser independiente del lenguaje de programación elegido para el desarrollo de aplicaciones. Para lograr esto se creó la Especificación de Lenguaje Común (o CLS, por sus siglas en inglés), que define y estandariza un subconjunto de todas las características soportadas por el CLR y que son necesarias en la mayoría de las aplicaciones. Todos los componentes desarrollados y compilados de acuerdo con la especificación CLS pueden interactuar entre sí, independientemente del lenguaje de programación de alto nivel en el que fueron escritos, permitiendo la interoperabilidad entre lenguajes de programación.

2.1.2.2. Funcionamiento del .NET Framework

Figura 3. Modelo de ejecución del .NET Framework



Fuente: Material Desarrollador Cinco Estrellas. Microsoft

Cuando se crea una aplicación Windows en algún lenguaje compatible con la plataforma .NET, se puede utilizar cualquiera de los servicios que la biblioteca de clases de .NET provee. Cuando se compila una aplicación, se crea un código intermedio llamado MSIL. Este código es independiente de la plataforma de hardware. Una vez compilado, el ejecutor de lenguaje común administra la ejecución de la aplicación.

Uno de los subsistemas del Common Language Runtime se llama compilación JIT, que transforma el código intermedio MSIL en código de máquina en el sistema donde la aplicación se va a ejecutar. Esta compilación a lenguaje de máquina se hace en el momento de ejecución del código. La aplicación sin embargo puede interactuar con otras aplicaciones .NET y servicios independientemente del lenguaje en que fueron desarrollados.

2.1.3. Herramientas de Desarrollo

2.1.3.1. Visual Studio .NET

Visual Studio es la herramienta de desarrollo por excelencia de la plataforma .NET, siendo una herramienta única que permite desarrollar cualquier tipo de aplicación (Web, Windows, de Consola, para dispositivos Móviles, para Microsoft Office, de Bases de Datos y más) en cualquiera de los lenguajes .NET provistos por Microsoft (C#, VB.NET, C++.NET y J#).

Visual Studio .NET es el entorno RAD (“Rapid Application Development” o Desarrollo Rápido de Aplicaciones), que permite utilizar .NET Framework para desarrollar las aplicaciones de una forma rápida y visual, incluyendo características de desarrollo como el IntelliSense. Visual Studio .NET utiliza .NET Framework.

Todas las aplicaciones que se realicen con .NET, serán desarrollos orientados a objetos. Este cambio es especialmente problemático para los desarrolladores que estaban acostumbrados a trabajar con Visual Basic 6 sin tener

conocimientos sobre la orientación a objetos. La problemática llega porque será necesario cambiar el esquema de trabajo que se llevaba a cabo cuando se trabajaba con Visual Basic.

Dentro del marco o infraestructura de desarrollo .NET (".NET Framework") conviven diferentes lenguajes de desarrollo; sin embargo, todos comparten una serie de características que son idénticas para cada uno de ellos. La más importante es que comparten el mismo entorno de trabajo, .NET Framework contiene todo lo necesario para programar, compilar y ejecutar las aplicaciones.

2.1.3.2. Visual Basic .NET

Es la tecnología que permite hacer aplicaciones Cliente – Servidor basado en interfaces de usuario gráficas GUI. Visual Basic .NET proporciona las siguientes características:

- Visual Basic .NET es ahora un lenguaje Orientado a Objetos, con cualidades similares a C++, conservando su especialidad: la manipulación y creación de componentes.
- Podemos crear clases, derivándolas de otras así estén escritas en Visual Basic .NET o en otros lenguajes .NET como el C#. Dispone, por tanto, de una característica largo tiempo reclamada por una parte de los usuarios de Visual Basic: la Herencia.
- También es un lenguaje que cuenta con la capacidad de sobrecargar métodos de tal forma que pueden existir múltiples versiones de un mismo método, cada uno tomando una lista de parámetros distinta.
- Se ha añadido un control estructurado de excepciones moderno, dando paso a la superación del arcaico On Error GoTo. Existen operadores aritméticos y lógicos abreviados, inicialización de variables durante la declaración, mayor consistencia de tipos de datos, entre otros.

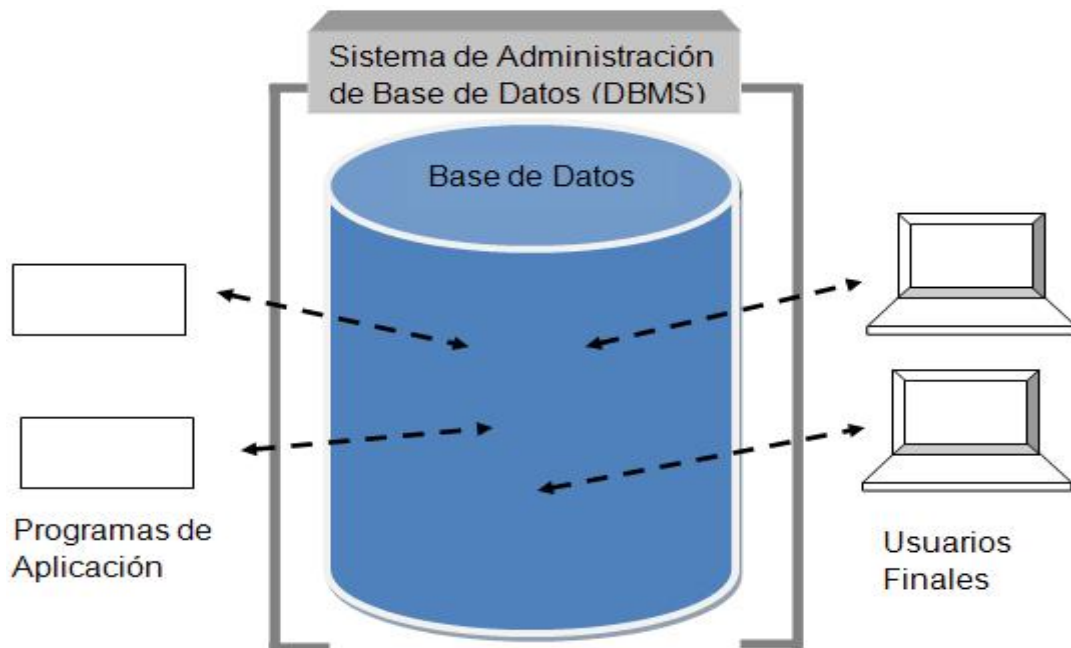
- Cualquier tipo de aplicación escrita en Visual Basic .NET puede utilizar libremente varios hilos de procesos concurrentes.
- No sólo se puede crear aplicaciones para Windows y componentes para ellas, también se puede utilizar Visual Basic .NET para producir aplicaciones de Consola, servicios de Windows, interfaces Web, componentes de Servidor ASP .NET y Servicios Web.

2.2. SISTEMA DE BASE DE DATOS

2.2.1. Definición de una Base de Datos

Una Base de datos es un conjunto de archivos almacenados de forma **integrada** y **compartida**, cuya finalidad es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones; este manejo implica el uso de un software administrador DBMS (Sistema de Administración de Base de Datos).

Figura 4. Imagen simplificada de un Sistema de Base de Datos



Fuente: Introducción a los Sistemas de Bases de Datos

2.2.2. Niveles de una Base de Datos

Nivel Interno

Está constituido por el conjunto de bloques de disco, con sus diferentes registros y sus respectivas direcciones, apuntadores, contadores y datos.

El nivel interno es el interactúa directamente con el sistema operativo (el servidor del disco) y a la tarjeta controladora del disco, por lo cual es un esquema físico binario, escondido en la intimidad del hardware.

Nivel Conceptual

Corresponde a la percepción de datos común a todos los usuarios de una organización. Por lo tanto se ve como un conjunto universal de entidades vinculadas entre sí, y cada una con sus propios atributos.

La apariencia de este nivel salta a la vista pero es generalmente amorfo y sin existencia física.

Nivel Externo

Corresponde a la percepción de datos que individualmente o en grupo tienen los usuarios de una misma área funcional de una organización.

Es una vista o subconjunto del conjunto universal, que describe únicamente la parte de los datos de interés para cada usuario y que, tampoco tiene existencia física.

Cada vista puede omitir uno o más registros, atributos o relaciones del esquema conceptual, o también cambiar su orden.

2.2.3. Propiedades de una Base de Datos

Independencia de los Datos

La independencia de los datos es un objetivo principal de los sistemas de Base de Datos, se define como la inmunidad de las aplicaciones a cambios en la representación física y en la técnica de acceso; lo que implica que las aplicaciones involucradas no dependan de ninguna representación física o técnica de acceso en particular.

Es la capacidad de modificar los esquemas interno o conceptual sin causar cambios en los programas (que usan esquemas externos).

La independencia entre lo externo y lo interno se llama "Independencia física" y, entre lo externo y lo conceptual "Independencia lógica". La independencia lógica sirve para el mantenimiento pero fomenta la separación entre la estructura y la funcionalidad organizacional.

Integridad

La integridad es el hecho de asegurar que los datos de una base de datos estén correctos. El control centralizado de la base de datos puede ayudar a evitar inconsistencia en la información, permitiendo que el administrador de la base de datos implemente las restricciones de integridad o reglas del negocio.

Los datos, apuntadores, direcciones, enlaces, índices y contadores, deben mantenerse siempre incorruptibles.

La corrupción puede aparecer por fallas del hardware, defectos del software, actualizaciones incompletas o inválidas, o la violación a las reglas de integridad referencial.

Seguridad

Los datos deben estar protegidos contra accesos no autorizados, y además reservados en diferentes rangos de permisividad para accesos autorizados.

Al tener la completa jurisdicción sobre la base de datos, el DBA puede asegurar que le único medio de acceso a la base de datos sea a través de los canales adecuados y por lo tanto puede definir las reglas o restricciones de seguridad que serán verificadas siempre que intente acceder a los datos. Es posible establecer diferentes restricciones para cada tipo de acceso (recuperación, inserción, eliminación y actualización) para cada parte de la información de la base de datos.

Control de Redundancia

Es la propiedad por medio de la cual, la repetición de datos es mínima y es controlable.

El proceso de normalización busca precisamente reducir al máximo la redundancia de datos, aunque nunca al 100%

Consistencia

Es la propiedad que impide las contradicciones entre datos.

Si la redundancia estuviera descontrolada, un mismo dato repetido en lugares diferentes podría tener valores diferentes, y por tanto inconsistentes. Resulta claro que una base de datos en un estado inconsistente es capaz de proporcionar a sus usuarios información incorrecta o contradictoria.

Control de Concurrencia

Es la capacidad para ejercer un riguroso control en la ejecución simultánea de transacciones con el fin de proteger la consistencia de las actualizaciones y consultas a la Base de datos.

Auditoría

Es el examen de la Base de datos y su entorno, a fin de comprobar que se ajusta a lo establecido. Este examen no es solo a posteriori sino que también es a priori

Respaldo y Recuperación

Es la facilidad para obtener copias de la Base de datos (backups de respaldo) que permitan recuperar la Base de datos ante cualquier falla.

Rutas de Acceso

Es la facilidad para obtener diferentes rutas de acceso, por medio de claves primarias y secundarias, pudiendo obtener respuesta a diferentes consultas aplicando diversos criterios de búsqueda.

2.2.4. Definición de Sistema de Administración de Base de Datos (DBMS)

Un DBMS es el software que maneja todo acceso a la base de datos y tiene las siguientes funciones:

- **Definición de datos:** es la creación objetos en la base de datos a partir de sentencias. En otras palabras el DBMS incluye entre sus componentes un compilador DDL (Lenguaje de Definición de Datos).

- **Manipulación de Datos:** es el manejo de peticiones para recuperar, actualizar o eliminar datos existentes en la base de datos o agregar nuevos datos a ésta. En otras palabras el DBMS incluye un componente compilador DML (Lenguaje de Manejo de Datos).
- **Optimización y ejecución:** las peticiones DML son procesadas por el componente optimizador, cuya finalidad es determinar una forma eficiente de implementar la petición.
- **Seguridad e Integridad:** su función es vigilar las peticiones del usuario y rechazar todo intento de violar las restricciones de seguridad e integridad definidas por el DBA.
- **Recuperación de datos y concurrencia:** el componente del DBMS denominado administrador de transacciones o monitor de procesamiento de transacciones (monitor PT) establece ciertos controles de recuperación y concurrencia.
- **Diccionario de Datos:** El diccionario de datos es el conjunto de datos que describen la BD, y contiene:
 - La descripción de todos los esquemas (externo, conceptual e interno).
 - La descripción de todos los campos (o atributos), registros (o tuplas) y referencias cruzadas entre los registros de varios archivos (o tablas, varrels o entidades).
 - Los códigos de autorización y seguridad de los datos y sus redefiniciones con las que puedan ser referidos con nombres diferentes en programas diferentes.

Para llevar a cabo su función de forma óptima, un DBMS se apoya en el Administrador de Base de Datos (DBA).

2.2.5. Administrador de Base de Datos (DBA)

El Administrador de Datos (DA) es la persona que toma las decisiones de estrategia y política con respecto a los datos de la empresa y el DBA es la persona que proporciona el apoyo técnico necesario para implementar dichas decisiones. Por lo tanto, el DBA es el responsable del control general del sistema al nivel técnico. Algunas de las tareas del DBA son las siguientes:

- Definir el Esquema Conceptual.
- Definir el Esquema Interno.
- Establecer un Enlace con los Usuarios.
- Definir las restricciones de Seguridad y de Integridad.
- Definir la políticas de vaciado y recarga.

2.3. POSTGRESQL

2.3.1. Antecedentes

- La implementación del DBMS POSTGRES empezó en 1996 en la Universidad de Berkeley en California por el profesor Michael Stonebraker.
- PostgreSQL es código abierto descendiente del original Ingres de Berkeley.
- La creciente comunidad de usuarios demandó gran cantidad de esfuerzo en soporte y mantenimiento, que debería ser invertido en Investigación. Por lo cual Berkeley terminó oficialmente el proyecto en 1993.
- En 1994, Andrew Yu y Jolly Chen le adicionaron un intérprete SQL. Postgres95 fue lanzado en la Web para que encontrara su propio camino. Fue escrito en C ANSI y se le realizaron algunas mejoras (funciones agregadas, group by, libpg, psql, libpgtcl, objetos grandes).

- En 1996, Se cambio el nombre a PostgreSQL y se empezó a numerar desde la versión 6.0.

2.3.2. Arquitectura

Las bases de la arquitectura del sistema PostgreSQL, entendiendo como las partes de Postgres, usa un modelo cliente/servidor conocido como "proceso por usuario".

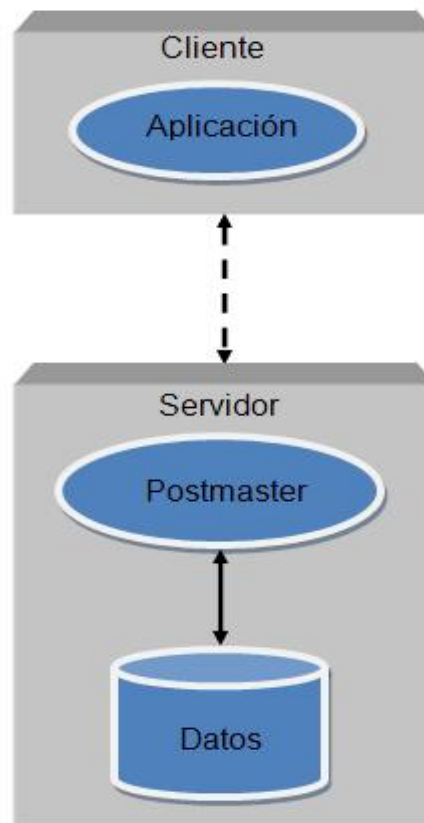
Una sesión PostgreSQL consiste en los siguientes procesos cooperativos:

- **Un proceso servidor:** Maneja los archivos de base de datos, acepta conexiones a la base de datos y permite acciones sobre estos archivos. El proceso servidor es llamado **Postmaster**.
- **Las aplicaciones de usuario final (frontEnd):** permiten realizar operaciones sobre la base de datos. Pueden ser de diversa naturaleza: Una herramienta de texto o consola, una aplicación gráfica, o una interfaz Web.

Un único Postmaster controla una colección de bases de datos dadas en un único host. Debido a esto una colección de bases de datos se suele llamar una instalación o un sitio.

Las aplicaciones frontEnd que quieren acceder a una determinada base de datos hacen llamadas a la librería y ésta envía peticiones de usuario a través de la red al Postmaster (en donde se establece una conexión), el cual en respuesta inicia un nuevo proceso en el servidor (backend).

Figura 5. Imagen de la Arquitectura de PostgreSQL



Fuente: PostgreSQL Introduction and Concepts

2.3.3. Características

Las principales mejoras en PostgreSQL incluyen:

- Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.
- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).

- Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena, literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20 - 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

Aislamiento Transaccional

El estándar ANSI/ISO SQL define cuatro niveles de aislamiento transaccional en función de tres hechos que deben ser tenidos en cuenta entre transacciones concurrentes. Estos hechos no deseados son:

- **Lecturas "sucias"**, una transacción lee datos escritos por una transacción no esperada, no confirmada (committed).
- **lecturas no repetibles**, una transacción vuelve a leer datos que previamente había leído y encuentra que han sido modificados por una transacción confirmada.
- **lectura "fantasma"**, una transacción vuelve a ejecutar una consulta, devolviendo un conjunto de filas que satisfacen una condición de búsqueda y encuentra que otras filas que satisfacen la condición han sido insertadas por otra transacción confirmada.

Bloqueos

Postgres ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere Postgres automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

Todos los modos de bloqueo (excepto para AccessShareLock) adquiridos en una transacción se mantienen hasta la duración de la transacción.

Existen dos tipos de bloqueos:

- Bloqueos a nivel de tabla.
- Bloqueos a nivel de registro.

Índices

Internamente Postgres almacena los datos en archivos del sistema operativo; cada tabla tiene su propio archivo.

Un índice es un archivo separado ordenado por una o más columnas, contiene apuntadores al archivo de datos permitiendo rápido acceso a la tabla.

Cluster

El comando cluster reordena físicamente una tabla para que coincida con el orden del índice. Este comando es de gran valor cuando el desempeño es crítico y cuando existen muchos valores duplicados.

La tabla es realmente copiada a una tabla temporal en el orden indicado, luego renombra la nueva con el nombre de la original. Por esta razón, todos los permisos sobre la tabla se pierden.

Vacuum

Cuando Postgres actualiza un registro, mantiene una copia del original y agrega uno nuevo. El registro original es marcado como borrado. De manera similar los registros eliminados con marcados como borrados.

El comando VACUUM remueve los registros marcados como borrados del archivo. Se debe correr VACCUM con cierta frecuencia sobre tablas que tienen gran movimiento.

Puede correr el comando sobre una tabla o sobre toda la Base de datos.

3. ANÁLISIS DEL SISTEMA

3.1. METODOLOGÍA

El ciclo de vida del software es un modelo con el cual se representan el proceso y las actividades con que se genera un producto software, además los requisitos frecuentemente cambian conforme al desarrollo de la aplicación.

Se reconoce que el software, al igual que todos los sistemas complejos, evoluciona con el tiempo.

Los requisitos de gestión y desarrollo de los productos a menudo cambian conforme avanza el proyecto, haciendo que el camino que lleva al producto final no sea real; las estrictas fechas de entrega hacen que sea imposible finalizar un producto completo, por lo que se debe introducir una versión limitada para cumplir con la presión competitiva; las versiones hacen comprender perfectamente el conjunto de requisitos centrales o del sistema, pero todavía se tienen que definir los detalles de extensiones del producto o sistema.

En estas y en otras situaciones similares, los ingenieros del software necesitan un modelo de proceso que se ha diseñado explícitamente para acomodarse a un producto que evolucione con el tiempo.

Se comprende que existe un conjunto de requisitos globales que dan una perspectiva general del resultado deseado por el Cliente, sin embargo, existen a su vez detalles de estos requisitos globales que se conocerán a profundidad en el desarrollo del ciclo de vida del software; en consecuencia a esta situación se necesita un modelo que se ajuste fácilmente a los cambios en los detalles de los requisitos globales, en otras palabras, un modelo que evolucione el software de acuerdo a los cambios planteados.

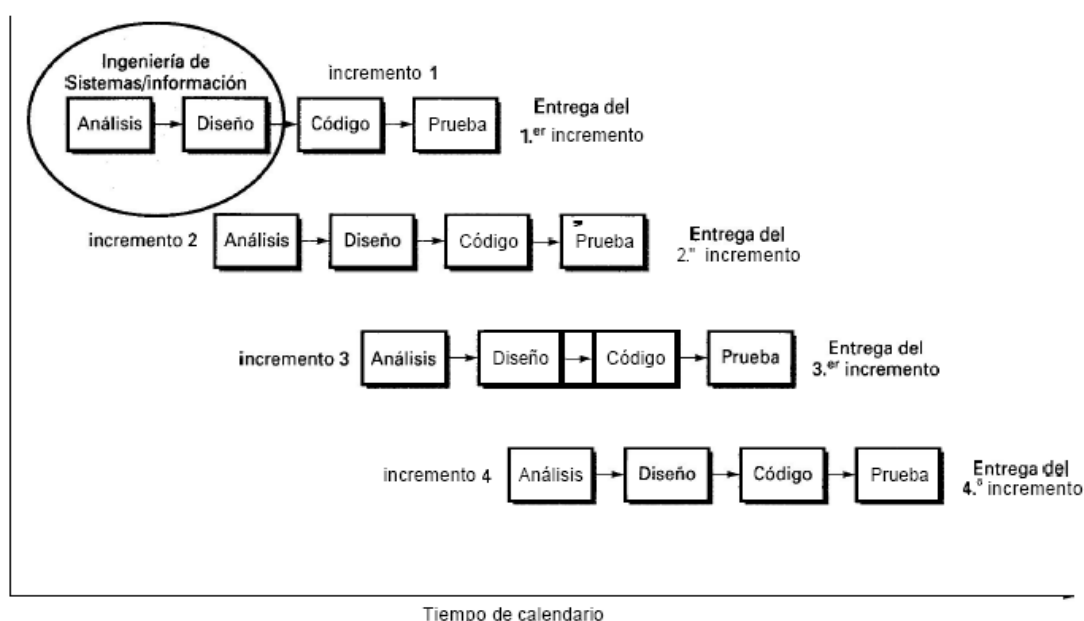
De acuerdo a esta necesidad, la Ingeniería del Software propone el Modelo Evolutivo Incremental que combina elementos del modelo lineal secuencial aplicados repetidas veces, con el objetivo de construir prototipos funcionales de forma iterativa, cada una de estas iteraciones o incrementos se realizan de forma escalonada proporcionando un mejoramiento del software.

El Modelo Evolutivo Incremental se centra en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones «incompletas» del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación.

El Modelo Evolutivo Incremental considera las siguientes partes como se muestran en la Figura 6:

- Análisis
- Diseño
- Codificación
- Prueba

Figura 6. Modelo Evolutivo Incremental



Fuente: Ingeniería del Software un enfoque práctico

Análisis

Esta etapa es indispensable para el desarrollo posterior del software, es la etapa donde se deben recolectar y comprender perfectamente los requisitos del mismo para disminuir el riesgo de malas interpretaciones a la hora de realizar el Diseño.

Diseño

En la etapa de diseño se realiza el proceso de traducción de los requisitos en una representación del software, esta debe estar debidamente documentada para ser usada en la fase de Codificación de tal forma que se obtenga un software con la calidad requerida.

Codificación

Durante la etapa de codificación se realiza la transformación del Diseño en un lenguaje de programación, esta codificación debe estar debidamente documentada para su posterior mantenimiento.

Prueba

Luego de haber generado el código correspondiente se procede a la realización de las pruebas, el uso de las pruebas nos asegura que las entradas definidas para el software produzcan correctamente los resultados que se requieren, posteriormente se inicia una nueva etapa comenzando desde el análisis.

Como consecuencia de la evolución en las tecnologías y los lenguajes de programación de Microsoft, se considera que la aplicación a desarrollar deberá evolucionar en el tiempo a la par de los adelantos tecnológicos para brindar el soporte requerido por la Comunidad por tanto se hace necesaria la aplicación del Modelo Evolutivo Incremental.

4. DESARROLLO DE LA PRÁCTICA

En el presente capítulo se plantea el contexto y las actividades desarrolladas durante los seis meses de duración de la práctica empresarial, que permitieron cumplir con los objetivos propuestos, y describe el seguimiento detallado del trabajo realizado durante el desarrollo del proyecto, basado en las fases planteadas por la metodología escogida.

Cada fase que compone la metodología del modelo evolutivo incremental será descrita en detalle a través del capítulo mostrando inicialmente el análisis y la especificación de los requerimientos identificados, a partir de los cuales se elabora un diseño del prototipo, en donde se establecen los diferentes módulos que apoyan los procesos identificados al igual que su correspondiente estructura de datos. Por último se presenta la implementación de los prototipos, mostrando los avances para cada una de las entregas definidas en el diseño.

4.1. COMPRENDER EL CONTEXTO DEL SISTEMA

Para comprender el contexto del sistema recibí una inducción sobre el esquema de trabajo desarrollado en la empresa GRUPO A.N.T & Cía. Ltda. y los productos software que actualmente se encuentran bajo su mantenimiento y desarrollo.

Al mismo tiempo recibí una capacitación sobre el manejo de los Módulos de Activos Fijos, Cargos Diferidos, Servicios Públicos, Impuesto Predial e Industria y Comercio, así como también el conocimiento del código fuente (desarrollado en Visual Fox Pro 8.0) de los módulos anteriormente mencionados para su posterior mantenimiento correctivo en el caso que se necesite hacer cualquier modificación.

De igual manera recibí una capacitación sobre la forma de trabajo y el manejo general del Sistema Administrativo Inteligente TIMAX, así como también el

conocimiento de cada una de las tablas que comprenden los diferentes módulos del Sistema.

4.1.1. Descripción de la Aplicación

El Sistema Administrativo Inteligente TIMAX, es un sistema de información el cual está enfocado al manejo y administración de todos los departamentos de una empresa tales como: contabilidad, facturación, recursos humanos, cartera, inventarios, centros de costos y otros; también este sistema de información es flexible al tipo de actividad de la empresa bien sea de tipo comercial, cooperativo, oficial, de servicios públicos o prestadora de servicios. El Sistema Administrativo Inteligente TIMAX, mantiene un esquema general en cada uno de sus módulos de la siguiente manera:

- **El menú “carita”:** o menú de configuración, en este menú es donde se parametriza y se configuran las tablas que van a ser utilizadas en el módulo correspondiente.
- **El menú Fichas:** es donde se maneja la información de los usuarios y los empleados (en el caso de los módulos de Directorio, Recursos humanos, Servicios públicos), los clientes (en el caso del módulo de Cuentas por cobrar), los proveedores (en el caso del módulo Cuentas por pagar), el plan de cuentas contable (en el caso del módulo de Contabilidad), los predios (en el caso del módulo de Impuesto predial), los establecimientos comerciales (en el caso del módulo de Industria y comercio), los artículos y activos de la empresa (en el caso de los módulos de Inventarios, Activos Fijos y Cargos diferidos).
- **El menú de Movimientos:** es donde se manejan todas las transacciones dependiendo del tipo de módulo en que se esté trabajando, por ejemplo: facturación individual, facturación masiva, compras, ventas, pagos, anulaciones de deuda, prescripciones de

deuda, novedades, toma de preguntas, quejas y reclamos (PQR), entre otras.

- **El menú de Informes:** este menú comprende una serie de informes que también están sujetos a los módulos en que esté trabajando el usuario; entre ellos se destacan las facturas, los comprobantes, los reportes de movimientos mensuales de la empresa como informes de recaudos, informes de gastos, balances de contabilidad, informes para la Contraloría General de la Nación, para la Superintendencia de Servicios Públicos, para la DIAN, y los informes particulares por empresa.
- **El menú de Procesos:** es donde se realizan los reversos de los movimientos como la facturación, los pagos o abonos, las novedades, las financiaciones, entre otros.

De igual manera en este menú se manejan los mantenimientos a las tablas en donde se almacena la información, como el reindexar las tablas, generación de índices, o la reorganización de la información de la empresa, esto es porque las tablas que son de tipo DBASE IV, se dañan o desactualizan sus índices debido a problemas de corriente o fallos en el sistema operativo, o en otros casos por problemas de manejo del usuario, como es la incorrecta asignación de consecutivos o incorrecta asignación de los precios de ventas, entre otros.

4.2. OBJETIVOS DEL SISTEMA

Tabla 1. Objetivo del Sistema 01

OBJ-01	Proporcionar una interfaz para la Creación de la Contabilidad de una Empresa desde Cero.
Descripción	El sistema permitirá que cualquier Usuario esté en capacidad de crear la contabilidad de una empresa con sólo tener los fundamentos básicos contables.

Tabla 2. Objetivo del Sistema 02

OBJ-02	Permitir actualizar la Información General de una Empresa.
Descripción	El sistema deberá permitir editar y cambiar la información contable de la empresa, siempre y cuando ésta no interfiera con las transacciones ya realizadas, esto sólo es posible a través de los usuarios autorizados.

Tabla 3. Objetivo del Sistema 03

OBJ-03	Proporcionar la interfaz para administración de Usuarios.
Descripción	La aplicación debe garantizar el manejo de contraseñas y permisos de los usuarios con el fin de certificar la seguridad dentro de cada módulo del sistema.

Tabla 4. Objetivo del Sistema 04

OBJ-04	Construir el mecanismo para el manejo del Directorio de la Empresa.
Descripción	El sistema debe proporcionar la interfaz para navegar, y actualizar la información del directorio de la empresa en cada uno de sus módulos.

Tabla 5. Objetivo del Sistema 05

OBJ-05	Ofrecer seguridad y confiabilidad en el manejo de la información.
Descripción	El sistema de información deberá contar en una base de datos robusta y basada en el concepto Entidad-Relación para proporcionar la integridad, y la seguridad en los datos, evitando así la redundancia.

Tabla 6. Objetivo del Sistema 06

OBJ-06	Permitir la personalización de los temas visuales para toda la aplicación.
Descripción	El sistema debe permitir configurar plantillas de diseño o seleccionar los colores que el usuario desee para toda la aplicación.

Tabla 7. Objetivo del Sistema 07

OBJ-07	Mantener los Usuarios informados sobre los plazos para los pagos y la entrega de los informes al Estado.
Descripción	El sistema deberá informar mediante ventanas los plazos que la empresa tiene para entregar los informes y/o pagos que debe realizar, por ejemplo, pago del IVA, informe de Retención en la Fuente, Medios Magnéticos, entre otros.

Tabla 8. Objetivo del Sistema 08

OBJ-08	Permitir la búsqueda por medio de términos en cualquier módulo de la aplicación.
Descripción	El sistema debe proporcionar una interfaz para permitir la búsqueda por cualquier término dependiendo el módulo en que se encuentre trabajando actualmente.

Tabla 9. Objetivo del Sistema 09

OBJ-09	Proporcionar la interfaz adecuada para el ingreso y edición de las fichas en cada uno de los módulos.
Descripción	El sistema debe proporcionar la Interfaz Gráfica del Usuario (GUI) para permitir el ingreso, modificación y borrado de las fichas en cada uno de los módulos que lo componen.

Tabla 10. Objetivo del Sistema 10

OBJ-10	Proporcionar la interfaz para la creación y restauración de copias de seguridad de la Base de Datos de la Empresa.
Descripción	El sistema debe proporcionar la Interfaz para la creación y restauración de copias de seguridad de la base de datos de la empresa a usuarios autorizados para realizar este proceso.

4.3. REQUISITOS DEL SISTEMA

Tabla 11. Requisito del Sistema 01

IRQ-01	Información de la Empresa
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-1 Proporcionar una interfaz para la creación de la contabilidad de una Empresa desde cero. • OBJ-2 Permitir actualizar la información general de una Empresa.
Requisitos Asociados:	Creación de la Empresa
Descripción:	El sistema deberá almacenar la información sobre el tipo de actividad de la empresa y con base a ésta configuración generar el Plan de Unico Cuentas -PUC- que se ajuste a su necesidad.
Datos Específicos:	Nombre de la Empresa. Nombre corto de la Base de Datos. Fecha de Inicio del Año Fiscal. Fecha Final del año Fiscal. Tipo de Actividad de la Empresa.

Tabla 12. Requisito del Sistema 02

IRQ-02	Información de Usuarios.
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-3 Proporcionar la interfaz para administración de Usuarios.
Requisitos Asociados:	Creación de los Usuarios.
Descripción:	El sistema almacenará la información de los Usuarios registrados, así como también permitirá su actualización
Datos Específicos:	Identificador del Usuario. Contraseña del Usuario. Foto del Usuario. Estado del Usuario. Permisos a los Módulos.

Tabla 13. Requisito del Sistema 03

IRQ-03	Información de los Permisos de Usuarios.
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-3 Permitir actualizar la información general de una Empresa.
Requisitos Asociados:	Inicio de la aplicación y autenticación del Usuario.
Descripción:	El sistema solicitará el Nombre del Usuario y su Contraseña en donde validará y habilitará los módulos correspondientes.
Datos Específicos:	Identificador del Usuario. Contraseña del Usuario. Foto del Usuario. Habilitar Módulos.

Tabla 14. Requisito del Sistema 04

IRQ-04	Información del Directorio de la Empresa.
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-4 Construir el mecanismo para el manejo del Directorio de la Empresa.
Requisitos Asociados:	<ul style="list-style-type: none"> • Creación de las Fichas de Clientes y Proveedores de la Empresa. • Modificación de los Datos de Clientes y Proveedores. • Calcular automáticamente el dígito de verificación con base al tipo de documento y su número. • Activar el nombre de la razón social si el tipo de documento es Nit o Nit de Extranjería. • Validar el tipo de régimen con base al tipo de documento seleccionado. • Validar si la ficha creada es autoretenedor y agente retenedor dependiendo del tipo de régimen. • Validar que los campos: Identificador de Usuario, Tipo de documento, Número de documento, Apellidos y Nombres, Razón social, Dirección, Tipo de régimen, Agente retenedor y Autoretenedor estén diligenciados correctamente.
Descripción:	El sistema permitirá la creación, modificación y validación de las fichas de los Clientes y Proveedores, así como también permitirá la navegación a través del Directorio.
Datos Específicos:	Identificador del Usuario. Tipo de Documento. Número de Documento. Dígito de Verificación. Persona Jurídica. Activo. Primer y Segundo Apellido. Primer y Segundo Nombre. Razón Social. Observaciones. Fecha de Ingreso. Referencia. Dirección. Barrio. Ciudad (Departamento). Área. Zona. Correo Electrónico. Página Web. Teléfono. Celular. Fax. Tipo de Régimen. Agente Retenedor. Autoretenedor. Exportador. Retefuente. Reteica. Tipo de actividad para Industria y Comercio. Tipo de Empresa. Aplica para Remisiones. Aplica para Ventas. Aplica para Compras. Imágenes. Campos Auxiliares.

Tabla 15. Requisito del Sistema 05

IRQ-05	Permitir el manejo de plantillas para aplicar temas visuales en la aplicación.
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-06 Permitir la personalización de los temas visuales para toda la aplicación.
Requisitos Asociados:	Establecer el nombre de la plantilla del tema visual.
Descripción:	El sistema le permitirá al usuario elegir entre los diferentes temas visuales para aplicarlos en todas las ventanas de la aplicación.
Datos Específicos:	Nombre del tema visual.

Tabla 16. Requisito del Sistema 06

IRQ-06	Permitir el manejo de colores para aplicarlos en el Sistema.
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-06 Permitir la personalización de los temas visuales para toda la aplicación.
Requisitos Asociados:	Establecer el color a aplicar.
Descripción:	El sistema le permitirá al usuario elegir un color entre los disponibles, para usarlo en toda la aplicación.
Datos Específicos:	Nombre del color.

Tabla 17. Requisito del Sistema 07

IRQ-07	Proporcionar información sobre las fechas de pagos pendientes que tiene la empresa.
Objetivos Asociados:	OBJ-07 Mantener los Usuarios informados sobre los plazos para los pagos y la entrega de los informes al Estado.
Requisitos Asociados:	Inicio de la Aplicación.
Descripción:	El sistema le informará al usuario en el inicio de la aplicación los pagos que tiene pendiente con el Estado como el caso del pago del IVA, el pago de la Retención en la Fuente y demás obligaciones Tributarias.
Datos Específicos:	Nombre de la Empresa. Login del Usuario. Contraseña del Usuario.

Tabla 18. Requisito del Sistema 08

IRQ-08	Proporcionar información de las fechas de plazo para la entrega de informes de la empresa.
Objetivos Asociados:	OBJ-07 Mantener los Usuarios informados sobre los plazos para los pagos y la entrega de los informes al Estado.
Requisitos Asociados:	Inicio de la Aplicación.
Descripción:	El sistema le informará al usuario en el inicio de la aplicación las fechas de plazo para la presentación de informes al Estado como el caso del informe a la DIAN, el informe de los Medios Magnéticos, y demás informes.
Datos Específicos:	Nombre de la Empresa. Login del Usuario. Contraseña del Usuario.

Tabla 19. Requisito del Sistema 09

IRQ-09	Permitir la búsqueda y consulta por medio de palabras claves.
Objetivos Asociados:	OBJ-08 Permitir la búsqueda por medio de términos en cualquier módulo de la aplicación.
Requisitos Asociados:	
Descripción:	El sistema le permitirá al usuario buscar y consultar por medio de palabras claves o por el campo seleccionado, dependiendo el módulo en donde se encuentre trabajando.
Datos Específicos:	Cadena de búsqueda. Palabras claves. Campo o Columna seleccionada.

Tabla 20. Requisito del Sistema 10

IRQ-10	Creación y edición de Fichas en cada uno de los módulos.
Objetivos Asociados:	OBJ-09 Proporcionar la interfaz adecuada para el ingreso y edición de las fichas en cada uno de los módulos.
Requisitos Asociados:	<ul style="list-style-type: none"> • Creación del PUC (Contabilidad, Tesorería, Presupuesto), creación de Usuarios (Servicios Públicos), creación de Predios (Impuesto Predial), creación de Establecimientos (Industria y Comercio), creación de Artículos y Activos (Inventarios y Activos Fijos), creación de Empleados (Nómina). • Modificación de los Datos anteriormente mencionados. • Validación de los datos relevantes en cada una de las fichas.
Descripción:	El sistema almacenará toda la información correspondiente a la ficha en el módulo en que se encuentre trabajando y permitirá su actualización.
Datos Específicos:	<p>Dependiendo del módulo en que se esté trabajando existen los siguientes datos:</p> <p>Código contable Identificador del usuario Identificador del predio Identificador del establecimiento Código del artículo o activo Identificador del empleado Tipo de identificación Número de identificación Nombres Apellidos Dirección Teléfono Bodega Centro de costos Avalúo Estrato Uso Área Actividad comercial Fecha de registro Usuario del sistema que ingresó el registro</p>

Tabla 21. Requisito del Sistema 11

IRQ-11	Creación y restauración de copias de seguridad de la Base de Datos de la Empresa a usuarios autorizados.
Objetivos Asociados:	OBJ-10 Proporcionar la interfaz para la creación y restauración de copias de seguridad de la Base de Datos de la Empresa.
Requisitos Asociados:	Usuarios autorizados.
Descripción:	El sistema proporcionará una interfaz gráfica para que el usuario autorizado pueda escoger el directorio en donde va a guardar o a restaurar la Copia de Seguridad de la Base de Datos de la Empresa.
Datos Específicos:	<p>Nombre del usuario. Clave del usuario.</p>

4.4. REQUISITOS FUNCIONALES DEL SISTEMA

4.4.1. Diagramas de Casos de Uso

Figura 7. Diagrama de Casos de Uso de los Actores del Sistema

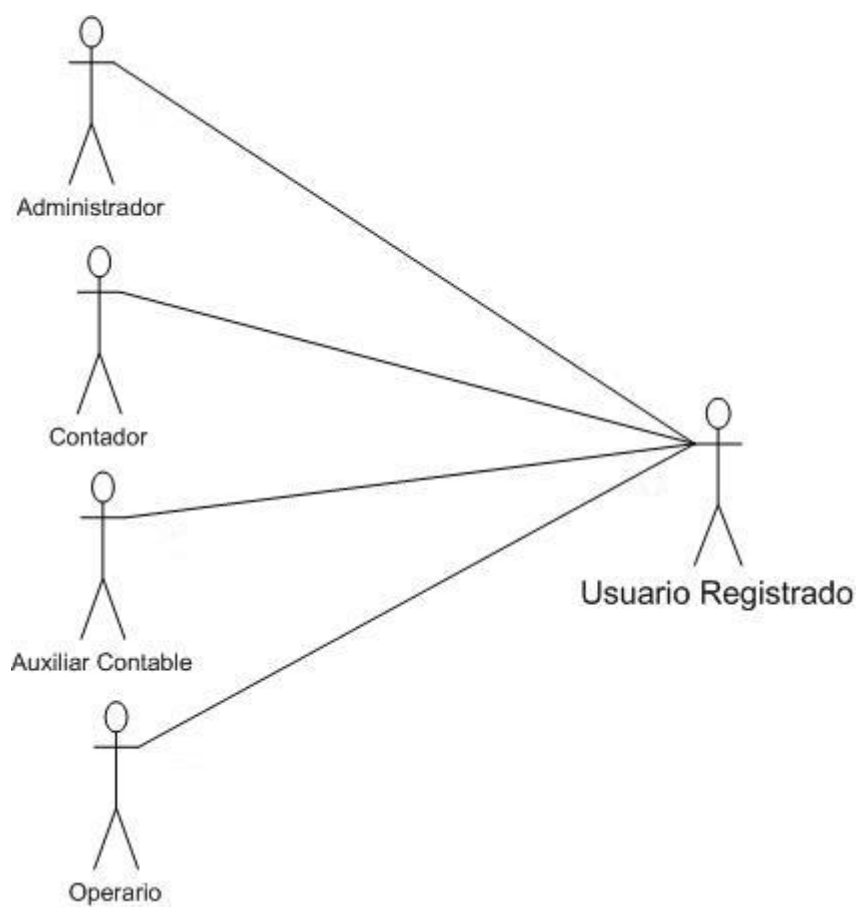


Figura 8. Diagrama de Subsistemas del Sistema Administrativo Inteligente TIMAX



Figura 9. Diagrama de Casos de Uso Configuración de la Aplicación

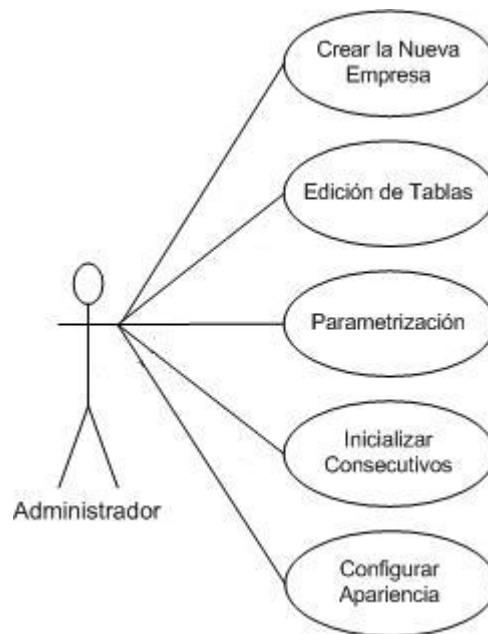


Figura 10. Diagrama de Casos de Uso Gestión de Usuarios

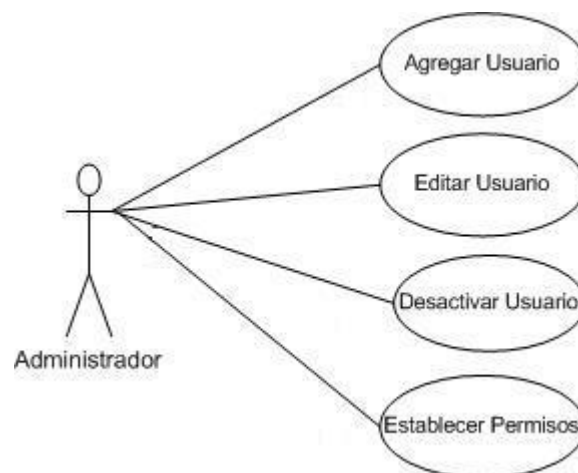


Figura 11. Diagrama de Casos de Uso Módulo de Directorio

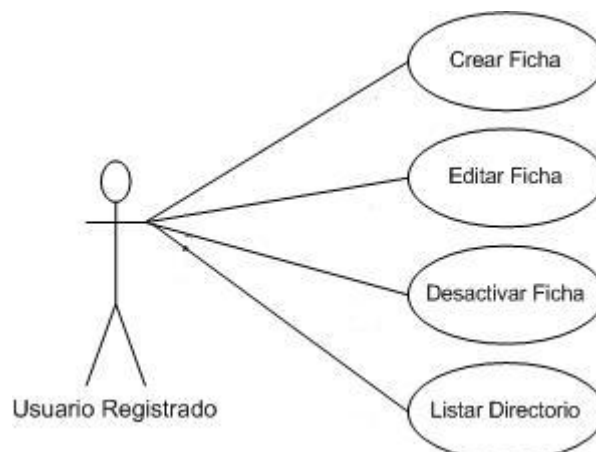


Figura 12. Diagrama de Casos de Uso Módulo de Recursos Humanos

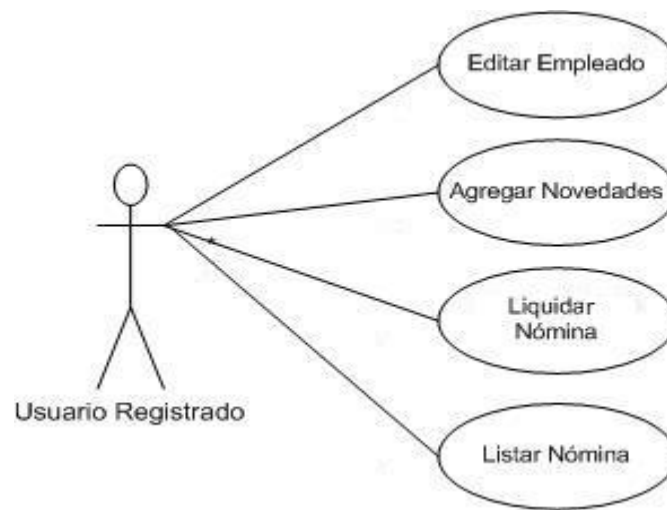


Figura 13. Diagrama de Casos de Uso Módulo de Facturación

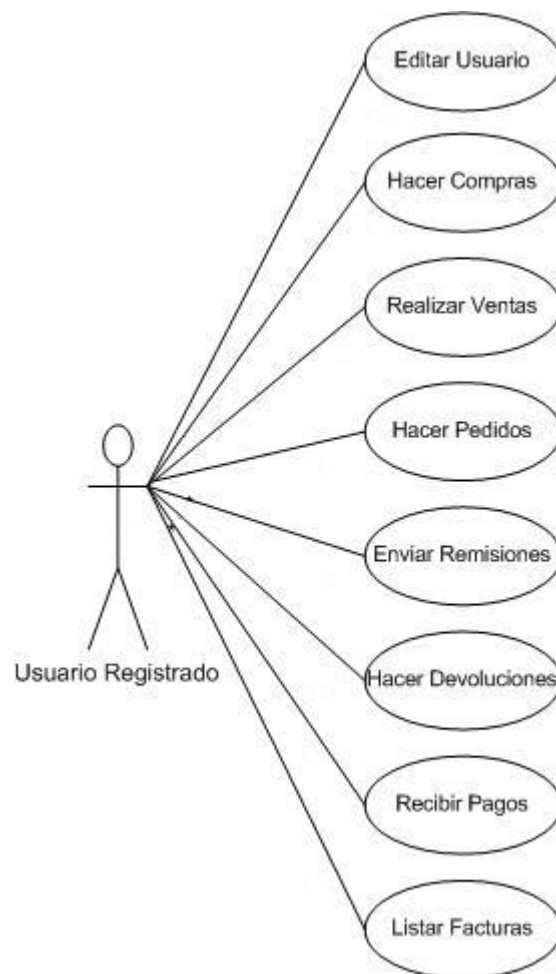


Figura 14. Diagrama de Casos de Uso Módulo de Inventarios

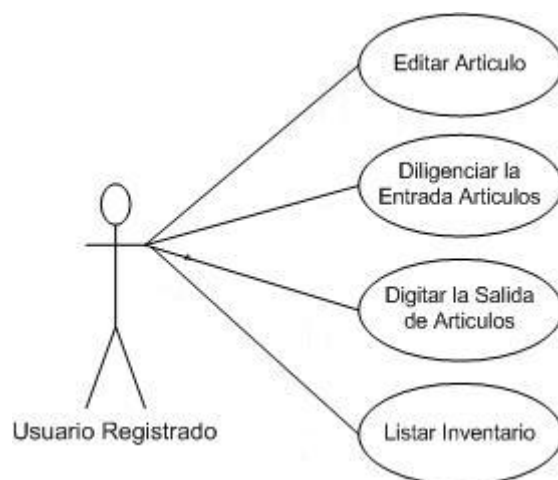


Figura 15. Diagrama de Casos de Uso Módulo Cuentas por Cobrar



Figura 16. Diagrama de Casos de Uso Módulo de Contabilidad

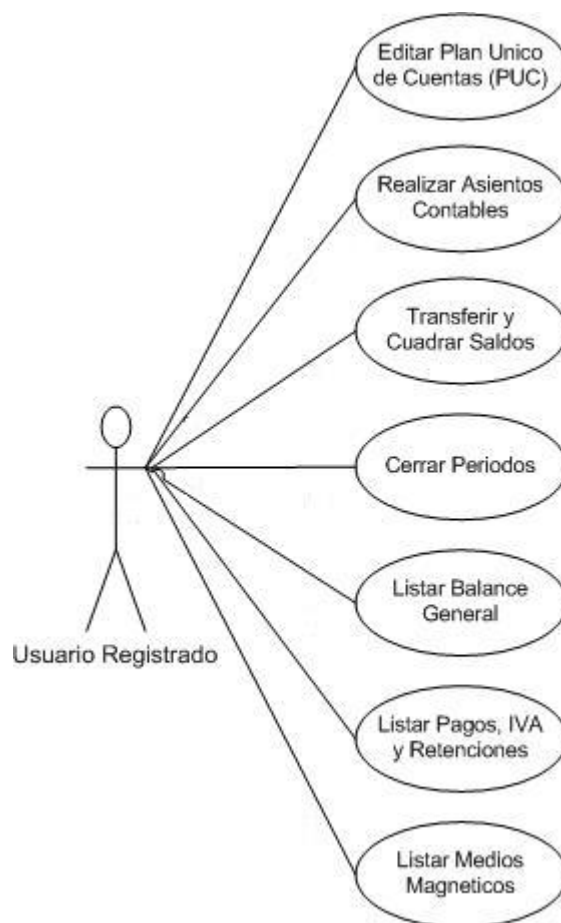


Figura 17. Diagrama de Casos de Uso Módulo Cuentas por Pagar

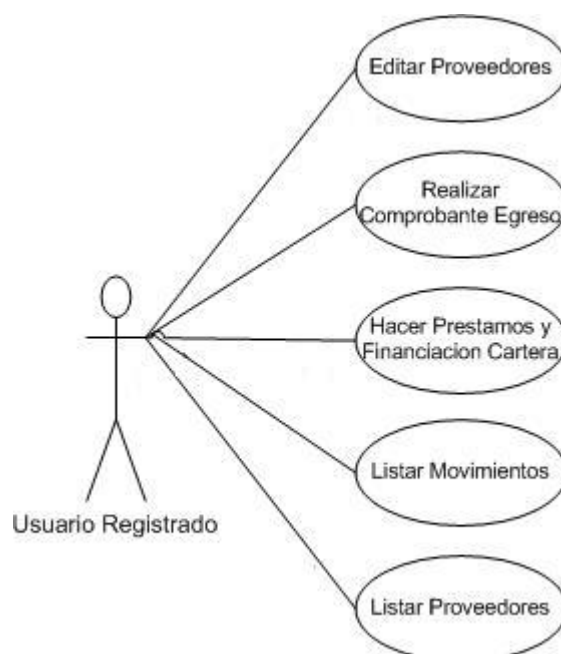


Figura 18. Diagrama de Casos de Uso Módulo de Tesorería



Figura 19. Diagrama de Casos de Uso Módulo de Presupuesto

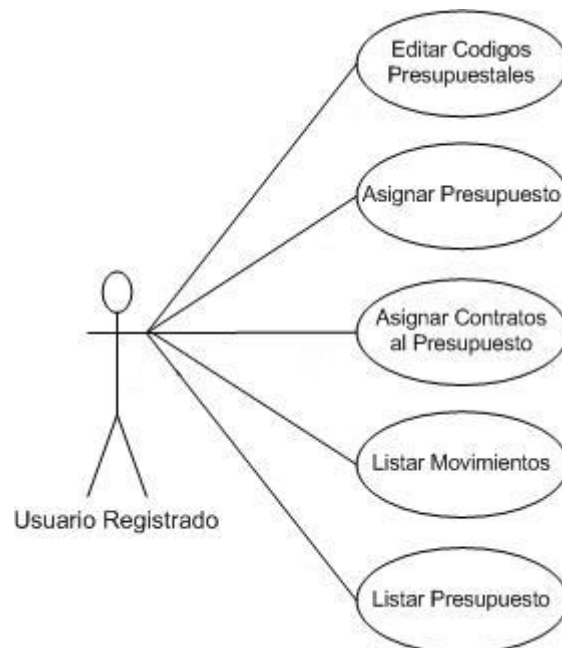


Figura 20. Diagrama de Casos de Uso Módulo de Activos Fijos

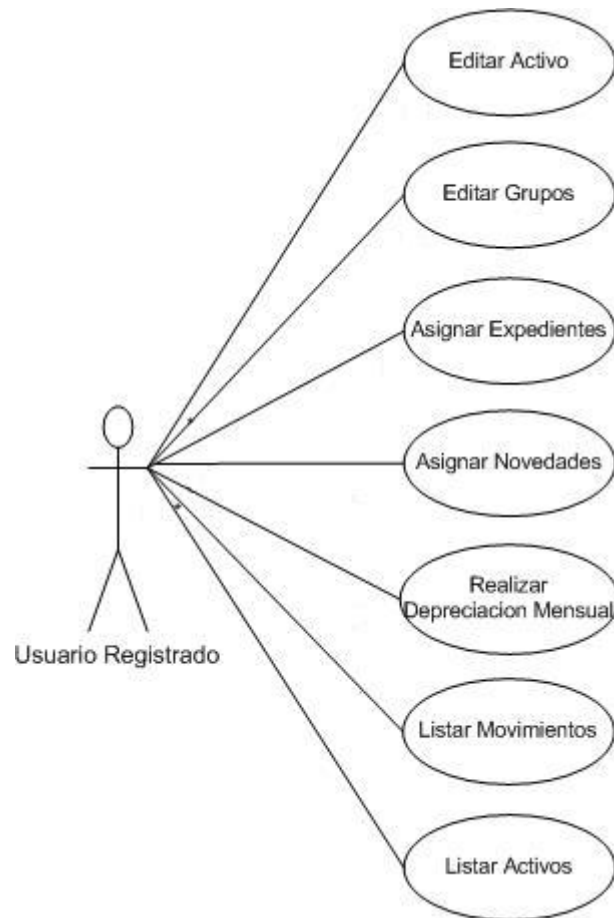


Figura 21. Diagrama de Casos de Uso Módulo Centro de Costos

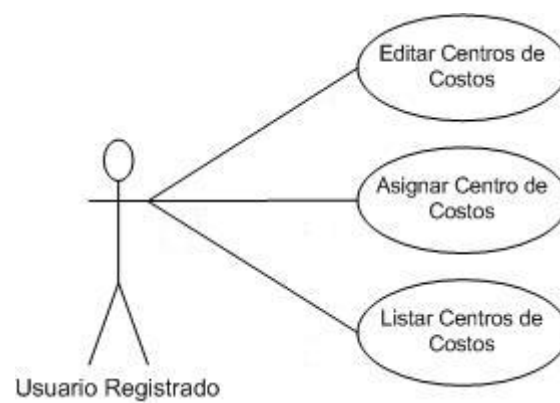


Figura 22. Diagrama de Casos de Uso Módulo de Impuesto Predial

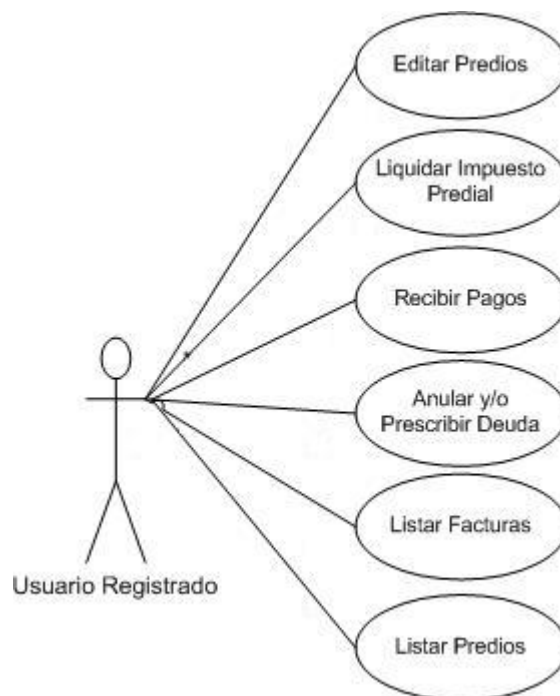


Figura 23. Diagrama de Casos de Uso Módulo de Industria y Comercio

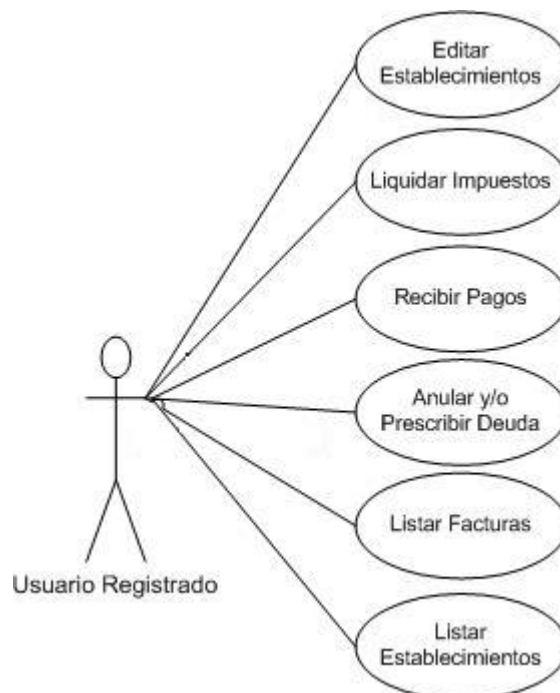


Figura 24. Diagrama de Casos de Uso de Servicios Públicos

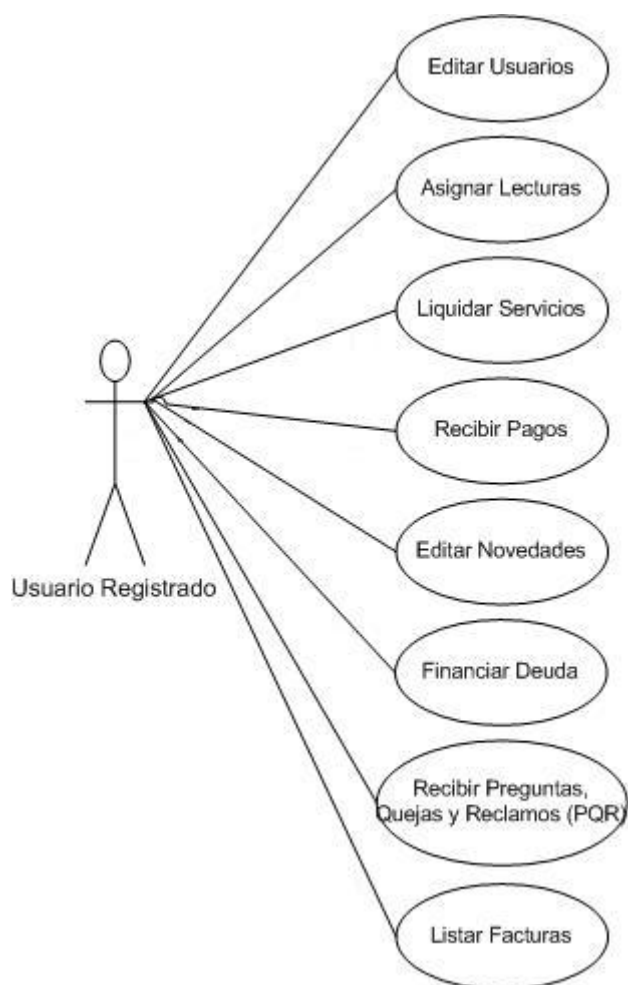


Figura 25. Diagrama de Casos de Uso Módulo Buscar

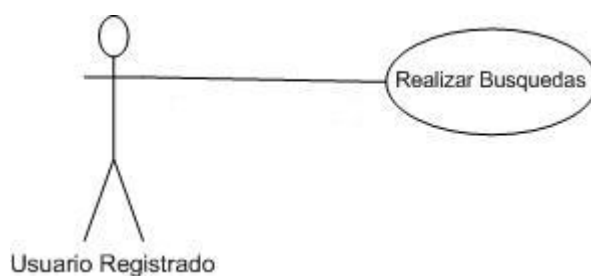
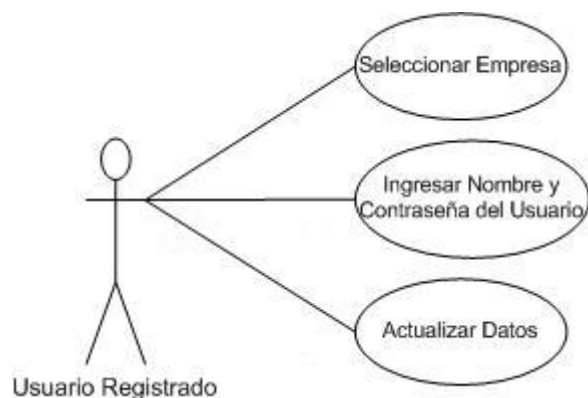


Figura 26. Diagrama de Casos de Uso Usuario Registrado



4.4.2. Definición de Actores del Sistema

Tabla 22. Definición de Actores del Sistema

ACTORES	DESCRIPCIÓN	FUNCIONES QUE REALIZA
Operador	Es el usuario que ejecuta funciones secundarias.	Sus funciones se destacan: hacer ventas, imprimir informes, hacer pedidos, recibir devoluciones, facturación, entre otras.
Auxiliar Contable	Es el Usuario que se encarga de llevar a cabo la contabilidad de la empresa.	Tiene autorización para manejar los comprobantes de contabilidad, generar informes de Balances Contables, Presupuestales, de IVA, de Retención en la Fuente.
Contador	En la mayoría de los casos este Usuario tiene los mismos permisos del Administrador.	Tiene autorización para Editar el Plan Único de Cuentas (PUC), para reversar movimientos, anulaciones de deuda, y tiene también permisos sobre todos los módulos de la Aplicación.
Administrador	Es el usuario que tiene acceso a todos los módulos de la Aplicación.	Funciones Exclusivas del administrador: <ul style="list-style-type: none"> • Creación de la Contabilidad de la Empresa. • Administrar Cuentas de Usuarios. • Administración y Auditoría en todos los módulos de la Aplicación. • Reversos de Movimientos como: facturas, pagos, financiaciones. • Cierre de Periodos y de Año Fiscal. • Visualizar reportes

4.4.3. Casos de Uso del Sistema

Configuración de la Aplicación

Tabla 23. Caso de Uso – Creación de la Nueva Empresa

Caso de Uso	Crear la Nueva Empresa
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-01 • OBJ-02
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-01
Proceso:	Este caso de uso inicia cuando el Administrador ingresa por primera vez a la Aplicación y crea la contabilidad de la nueva empresa. La Aplicación se comportará como se describe a continuación:
Precondición:	El Actor Administrador debe ingresar la información inicial de la Empresa que va a llevar la contabilidad.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario Administrador debe digitar el Nombre o Razón Social de la Empresa. 2. También debe digitar un nombre corto para la Base de Datos. 3. Debe colocar la fecha de inicio y finalización en que va a llevar la contabilidad de la empresa. La aplicación por defecto colocará el 01 de Enero del año en curso hasta el 31 de Diciembre del mismo año. 4. El Usuario Administrador debe seleccionar el tipo de Actividad de la Empresa. 5. El Usuario dará clic en Cerrar para finalizar.
Postcondición:	La Aplicación generará la nueva Base de Datos e introducirá la información establecida por el usuario de este caso de uso.
Excepción:	

Tabla 24. Caso de Uso – Edición de Tablas

Caso de Uso	Edición de Tablas
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-02
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-01
Proceso:	Este caso de uso inicia cuando el Administrador crea las tablas de grupos, zonas, prefijos, integración con contabilidad, integración con Centro de Costos, Bodegas, lista de precios, intereses, descuentos, entre otras. La aplicación se comportará como se describe a continuación:
Precondición:	El Actor Administrador deberá establecer los tipos de precios que se manejan, lo mismo hará con la tasa de interés, los porcentajes para cada tipo de IVA y las reglas para aplicar los descuentos.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario debe seleccionar la tabla que va a editar. 2. Editar y actualizar los valores de cada una de las tablas. 3. Establecer el nombre de los formatos que fueron desarrollados particularmente. 4. El Usuario dará clic en Cerrar para finalizar.
Postcondición:	La Aplicación guardará la información introducida por el usuario de este caso de uso.
Excepción:	

Tabla 25. Caso de Uso – Parametrización

Caso de Uso	Parametrización
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-02
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-01
Proceso:	Este caso de uso inicia cuando el Administrador configura la parametrización general de la Aplicación. Esta se comportará como se describe a continuación:
Precondición:	El Actor Administrador debe parametrizar y establecer las reglas de juego que van a regir las transacciones que se realizarán en la aplicación.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario debe seleccionar el tipo de redondeo que se llevará a cabo en todas las transacciones. 2. Debe establecer los porcentajes para los diferentes tipos de IVA que se aplican. 3. Debe parametrizar el porcentaje de Retención en la Fuente, sobretasas, e impuestos. 4. El Usuario debe marcar los códigos contables que servirán para la generación de informes, también debe parametrizar los prefijos que se utilizarán luego en las facturas, notas contables y comprobantes de ingreso. 5. El Usuario dará clic en Cerrar para finalizar.
Postcondición:	La Aplicación guardará toda la información seleccionada e introducida dependiendo del caso por el usuario de este caso de uso.
Excepción:	

Tabla 26. Caso de Uso – Inicializar Consecutivos

Caso de Uso	Inicializar Consecutivos
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-02
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-01
Proceso:	Este caso de uso inicia cuando el Administrador establece los rangos de consecutivos para las facturas, notas de contabilidad y los comprobantes de ingreso. La aplicación se comportará como se describe a continuación:
Precondición:	El Actor Administrador debe previamente haber creado la tabla de prefijos en la sección de edición de tablas.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario establece el prefijo para cada módulo. 2. Establece el rango de consecutivos que regirán a lo largo de las diferentes transacciones que se realicen. 3. El Usuario dará clic en Cerrar para finalizar.
Postcondición:	La aplicación guardará la información introducida por el usuario de este caso de uso y en cada una de las transacciones asignará su consecutivo correspondiente.
Excepción:	

Tabla 27. Caso de Uso – Configurar Apariencia

Caso de Uso		Configurar Apariencia
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 	
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-06 	
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-05 • IRQ-06 	
Descripción:	Este caso de uso inicia cuando el Actor ingresa a la Aplicación. El Sistema se comportará como se describe a continuación:	
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.	
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Carita" y hace clic en "Configurar Apariencia". 2. El Sistema le mostrará los diferentes estilos de ventanas. 3. El Usuario hace clic al Estilo que desea aplicar, y el Sistema procede a aplicar el Estilo. 4. El Sistema también le mostrará los diferentes colores que puede escoger. 5. El Usuario hace clic al Color que desea aplicar, y el Sistema procede a aplicar el Color. 	
Postcondición:	La aplicación establece el Estilo y Color elegido para todo el Sistema.	
Excepción:		

Gestión de Usuarios

Tabla 28. Caso de Uso – Agregar Usuario

Caso de Uso		Agregar Usuario
Actor(es):	<ul style="list-style-type: none"> • Administrador 	
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-03 	
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-02 	
Descripción:	Este caso de uso inicia cuando el Administrador ingresa al Módulo de Administración de Usuarios. El Sistema se comportará como se describe a continuación:	
Precondición:	El Actor Administrador debe haber ingresado su nombre de usuario y contraseña.	
Secuencia Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el menú de "Carita" y hace clic en "Claves de Acceso". 2. El Sistema mostrará el formulario de "Administración de Usuarios". 3. El Usuario agregará la información del Nuevo Usuario y dará clic en Guardar Usuario. 4. El Sistema validará la información suministrada por el Usuario. 5. El Sistema guardará la información e informará "El usuario ha sido creado". 6. El usuario dará clic en Cerrar para finalizar. 	
Postcondición:	Se agregará un nuevo usuario a la lista de usuarios del Sistema.	
Excepción:	<ol style="list-style-type: none"> 3. En caso de que el nombre de usuario ya exista se le notificará al Usuario que intente con otro nombre. 	

Tabla 29. Caso de Uso – Editar Usuario

Caso de Uso	Editar Usuario
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-03
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-02
Descripción:	Este caso de uso inicia cuando el Administrador ingresa al Módulo de Administración de Usuarios. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor Administrador debe haber ingresado su nombre de usuario y contraseña, y debe existir la cuenta de Usuario que va a ser modificada.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el menú "Carita" y hace clic en "Claves de Acceso". 2. El Sistema mostrará el formulario "Administración de Usuarios". 3. El Usuario Administrador mediante la ventana de búsqueda seleccionará al Usuario y dará clic en "Modificar Usuario" para habilitar los campos que puede modificar. 4. El Usuario cambiará la información necesaria y dará clic en "Guardar Usuario". 5. El Sistema validará la información suministrada por el Usuario Administrador. 6. El Sistema guardará la información e informará "El usuario ha sido modificado". 7. El usuario dará clic en "Cerrar" para finalizar.
Postcondición:	La aplicación actualizará los datos del usuario.
Excepción:	

Tabla 30. Caso de Uso –Desactivar Usuario

Caso de Uso	
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-03
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-02
Descripción:	Este caso de uso inicia cuando el Administrador ingresa al Módulo de Administración de Usuarios. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor Administrador debe haber ingresado su nombre de usuario y contraseña, y debe existir la cuenta de Usuario que va a ser desactivada.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el menú "Carita" y hace clic en "Claves de Acceso". 2. El Sistema mostrará el formulario "Administración de Usuarios". 3. El Usuario Administrador mediante la ventana de búsqueda seleccionará al Usuario y quitará el chequeo en "Usuario Activo". 4. El Usuario dará clic en "Guardar Usuario". 5. El Sistema validará nuevamente la información suministrada por el Usuario Administrador. 6. El Sistema guardará la información e informará "El usuario ha sido Desactivado". 7. El usuario dará clic en Cerrar para finalizar.
Postcondición:	Se marcará como desactivado del Sistema el Usuario elegido por el administrador, esto es porque ningún usuario puede ser borrado de la Base de Datos.
Excepción:	

Tabla 31. Caso de Uso – Establecer Permisos

Caso de Uso	Establecer Permisos
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-03
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-02
Descripción:	Este caso de uso inicia cuando el Administrador ingresa al Módulo de Administración de Usuarios. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor Administrador debe haber ingresado su nombre de usuario y contraseña.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el menú "Carita" y hace clic en "Claves de Acceso". 1. El Sistema mostrará el formulario "Administración de Usuarios". 2. El Usuario Administrador mediante la ventana de búsqueda seleccionará al Usuario (si existe en la Base de Datos) o de lo contrario creará uno nuevo. 3. En el marco de "Selección de Permisos" el Usuario tendrá la opción de seleccionarle o deshabilitarle todos los permisos a los diferentes Módulos del Sistema de forma automática, al usuario que está editando, o también puede seleccionarle los permisos de forma manual en cada uno de los Módulos del Sistema. 4. El Sistema validará nuevamente la información suministrada por el Usuario Administrador. 5. El Sistema guardará la información e informará "El usuario ha sido Guardado". 6. El usuario dará clic en Cerrar para finalizar.
Postcondición:	Se establecerán los permisos en cada uno de los Módulos del Sistema a los cuales tendrá derecho el Usuario elegido por el administrador.
Excepción:	

Módulo de Directorio

Tabla 32. Caso de Uso – Crear Ficha

Caso de Uso	Crear Ficha
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-04
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-04
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Directorio. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú de "Gestión" y hace clic en "Directorio". 2. El Sistema mostrará el formulario de "Directorio". 3. El Usuario agregará la información del nuevo Cliente, Proveedor o Empleado y dará clic en Guardar. 4. El Sistema validará la información suministrada por el Usuario (campos obligatorios). 5. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 6. El Usuario dará clic en Cerrar o pulsará escape para finalizar.
Postcondición:	Se agregará una nueva Ficha al Directorio del Sistema.
Excepción:	<ol style="list-style-type: none"> 4. En caso de que el código (o cuenta) de la nueva Ficha ya exista se le notificará al Usuario que intente con otro código (o cuenta).

Tabla 33. Caso de Uso – Editar Ficha

Caso de Uso	Editar Ficha
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-04
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-04
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Directorio. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña, y debe existir la cuenta de la Ficha que va a ser modificada.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Directorio". 2. El Sistema mostrará el formulario de "Directorio". 3. El Usuario mediante la ventana de búsqueda seleccionará la Ficha que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede modificar. 4. El Usuario cambiará la información necesaria y dará clic en "Guardar" 5. El Sistema validará la información suministrada por el Usuario. 6. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 7. El Usuario dará clic en Cerrar o pulsará escape para finalizar.
Postcondición:	La aplicación actualizará los datos de la Ficha.
Excepción:	

Tabla 34. Caso de Uso – Desactivar Ficha

Caso de Uso		Desactivar Ficha
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 	
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-04 	
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-04 	
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Directorio. El Sistema se comportará como se describe a continuación:	
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña, y debe existir la cuenta de la Ficha que va a ser desactivada.	
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Carita" y hace clic en "Directorio". 2. El Sistema mostrará el formulario "Directorio". 3. El Usuario mediante la ventana de búsqueda seleccionará la Ficha y quitará el chequeo en "Activo". 4. El Usuario dará clic en "Guardar". 5. El Sistema validará nuevamente la información suministrada por el Usuario. 6. El Sistema guardará la modificación e informará "Registro Almacenado Satisfactoriamente". 7. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 	
Postcondición:	Se marcará como desactivado del Sistema, la Ficha elegida por el Usuario, esto es porque ninguna Ficha puede ser borrada de la Base de Datos.	
Excepción:		

Tabla 35. Caso de Uso – Listar Directorio

Caso de Uso		Listar Directorio
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 	
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-04 	
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-04 	
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Directorio. El Sistema se comportará como se describe a continuación:	
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.	
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Carita" y hace clic en "Directorio". 2. El Sistema mostrará el formulario "Directorio". 3. En el menú de Informes del Módulo de Directorio, se encuentra una interfaz con una serie de filtros como por ejemplo: por código de usuario, por nombres, por fechas de creación, por ubicación, entre otros. 4. Una vez aplicado el filtro se visualiza el informe y permite las opciones de exportarlo a Excel e imprimirlo. 5. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 	
Postcondición:		
Excepción:		

Módulo de Tesorería

Tabla 36. Caso de Uso – Asignar Códigos Contables

Caso de Uso		Asignar Códigos Contables	
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 		
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09 		
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10 		
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Tesorería. El Sistema se comportará como se describe a continuación:		
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.		
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Tesorería". 2. En el Módulo de Tesorería escoge el menú de Fichas y hace clic en "Cuentas Cajas/Bancos" y el Sistema mostrará el formulario de las Fichas de los Códigos Contables asignados al Módulo de Tesorería. 3. El Usuario primero deberá escoger si va a generar una cuenta de Caja o de Bancos. 4. Mediante la ventana de búsqueda seleccionará el Código Contable que va a ser asignado al Módulo de Tesorería. 5. Tendrá la opción de cambiar el Código Contable seleccionado mediante el botón de "Modificar". 6. Igualmente, mediante el botón de "Eliminar" podrá borrar el Código Contable del Módulo de Tesorería. 7. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 		
Postcondición:	La aplicación actualizará los datos de la Ficha del Módulo de Tesorería.		
Excepción:	<ol style="list-style-type: none"> 5. El Código Contable asignado al Módulo de Tesorería sólo podrá ser modificado o borrado cuando no tenga movimientos contables. 		

Módulo de Contabilidad

Tabla 37. Caso de Uso – Editar Plan Único de Cuentas (PUC)

Caso de Uso	Editar Plan Unico de Cuentas (PUC)
Actor(es):	<ul style="list-style-type: none"> Contador
Objetivos Asociados:	<ul style="list-style-type: none"> OBJ-09
Requisitos Asociados:	<ul style="list-style-type: none"> IRQ-10
Descripción:	Este caso de uso inicia cuando el Actor Contador ingresa al Módulo de Contabilidad. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor Contador debe haber ingresado su nombre de usuario y contraseña.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Contabilidad". 2. En el Módulo de Contabilidad escoge el menú de Fichas y hace clic en "Plan de Cuentas" y el Sistema mostrará el formulario con el Plan de Cuentas de la Empresa, estos Códigos Contables fueron cargados automáticamente cuando fue creada la contabilidad de la Empresa. 3. Mediante los botones de Edición (Nuevo, Modificar, Guardar, Eliminar), el Usuario puede editar los Códigos Contables. 4. Con los botones de Navegación (Primero, Anterior, Siguiente y Ultimo), el Usuario puede recorrer el Plan de Cuentas. 5. El Usuario dará clic en Cerrar o pulsará escape para finalizar.
Postcondición:	La aplicación actualizará los datos de la Ficha del Módulo de Contabilidad.
Excepción:	<ol style="list-style-type: none"> 3. El Código Contable asignado al Módulo de Contabilidad, sólo podrá ser modificado o borrado si NO tiene movimientos contables. 3. Solamente será válido el Código Contable que esté compuesto por números y puntos. 3. Si el Código Contable ya existe, se le notificará al Usuario que intente con otro Código.

Tabla 38. Caso de Uso – Realizar Asientos Contables

Caso de Uso Realizar Asientos Contables	
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Contabilidad. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Contabilidad". 2. En el Módulo de Contabilidad escoge el menú de Movimientos y hace clic en "Comprobantes de Contabilidad" y el Sistema mostrará el formulario de Comprobantes Contables. 3. El Usuario llenará o seleccionará (dependiendo el caso) los campos correspondientes al encabezado y cuerpo (o detalle) del comprobante contable como lo son: tipo de comprobante, fecha, Código Contable, Cuenta del Directorio, tipo de documento, número de documento, detalle y valor. 4. Mediante los botones de Edición (Nuevo, Modificar, Guardar, Eliminar, Buscar), el Usuario puede editar los Comprobantes Contables. 5. Una vez terminado la edición del Comprobante Contable, el Usuario al dar clic en "Guardar", el Sistema le asigna el Prefijo y su correspondiente Consecutivo. 6. El Usuario dará clic en Cerrar o pulsará escape para finalizar.
Postcondición:	La aplicación guardará la información del Comprobante Contable correspondiente al Módulo de Contabilidad.
Excepción:	<ol style="list-style-type: none"> 3. En el detalle del Comprobante Contable NO pueden existir dos registros totalmente idénticos.

Módulo de Cuentas por Cobrar

Tabla 39. Caso de Uso – Editar Clientes

Caso de Uso	Editar Clientes
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Cuentas por Cobrar. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Cuentas por Cobrar". 2. En el Módulo de Cuentas por Cobrar escoge el menú de Fichas y hace clic en "Directorio Cobrar" y el Sistema mostrará el formulario con el Directorio de este Módulo. 3. El Usuario agregará la información del nuevo Cliente y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará la Ficha del Cliente que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar". 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada una de las Fichas de los Clientes. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. De la misma manera, con los botones de Navegación (Primero, Anterior, Siguiente y Ultimo), el Usuario puede navegar por todo el Directorio del Módulo de Cuentas por Cobrar. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar.
Postcondición:	La aplicación guardará los datos de la Ficha del Directorio de Cuentas por Cobrar.
Excepción:	<ol style="list-style-type: none"> 3. Si el Código de la Ficha ya existe, se le notificará al Usuario que intente con otro Código.

Módulo de Cuentas por Pagar

Tabla 40. Caso de Uso – Editar Proveedores

Caso de Uso		Editar Proveedores	
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 		
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09 		
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10 		
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Cuentas por Pagar. El Sistema se comportará como se describe a continuación:		
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.		
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Cuentas por Pagar". 2. En el Módulo de Cuentas por Pagar escoge el menú de Fichas y hace clic en "Directorio Pagar" y el Sistema mostrará el formulario con el Directorio de este Módulo. 3. El Usuario agregará la información del nuevo Proveedor y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará la Ficha del Proveedor que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar" 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada una de las Fichas de los Proveedores. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. Mediante los botones de Navegación (Primero, Anterior, Siguiente y Ultimo), el Usuario puede navegar por todo el Directorio del Módulo de Cuentas por Pagar. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 		
Postcondición:	La aplicación guardará los datos de la Ficha del Directorio de Cuentas por Pagar.		
Excepción:	<ol style="list-style-type: none"> 3. Si el Código de la Ficha ya existe, se le notificará al Usuario que intente con otro Código. 		

Módulo de Inventarios

Tabla 41. Caso de Uso – Editar Artículos

Caso de Uso		Editar Artículos
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 	
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09 	
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10 	
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Inventarios. El Sistema se comportará como se describe a continuación:	
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.	
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Inventarios". 2. En el Módulo de Inventarios escoge el menú de Fichas y hace clic en "Artículos Inventarios" y el Sistema mostrará el formulario de Artículos. 3. El Usuario agregará la información del nuevo Artículo y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará la Ficha del Artículo que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar" 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada una de las Fichas de los Artículos. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. Mediante los botones de Navegación (Primero, Anterior, Siguiete y Ultimo), el Usuario puede navegar por todos los Artículos del Módulo de Inventarios. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 	
Postcondición:	La aplicación guardará los datos de la Ficha de Artículos de Inventarios.	
Excepción:	<ol style="list-style-type: none"> 3. Si el Código del Artículo ya existe, se le notificará al Usuario que intente con otro Código. 	

Módulo de Activos Fijos

Tabla 42. Caso de Uso – Editar Activo

Caso de Uso		Editar Activos	
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 		
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09 		
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10 		
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Activos Fijos. El Sistema se comportará como se describe a continuación:		
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.		
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Activos Fijos". 2. En el Módulo de Activos Fijos escoge el menú de Fichas y hace clic en "Directorio de Activos" y el Sistema mostrará el formulario de los Activos. 3. El Usuario agregará la información del nuevo Activo y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará la Ficha del Activo que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar" 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada una de las Fichas de los Activos. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. Mediante los botones de Navegación (Primero, Anterior, Siguiete y Ultimo), el Usuario puede navegar por todas las Fichas del Módulo de Activos Fijos. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 		
Postcondición:	La aplicación guardará los datos de la Ficha del Módulo de Activos Fijos.		
Excepción:	<ol style="list-style-type: none"> 3. Si el Código del Activo ya existe, se le notificará al Usuario que intente con otro Código. 		

Tabla 43. Caso de Uso – Editar Grupos

Caso de Uso		Editar Grupos
Actor(es):	<ul style="list-style-type: none"> • Administrador • Contador • Usuario Autorizado 	
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09 	
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10 	
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Activos Fijos. El Sistema se comportará como se describe a continuación:	
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.	
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Activos Fijos". 2. En el Módulo de Activos Fijos escoge el menú de Fichas y hace clic en "Directorio de Grupos" y el Sistema mostrará el formulario de los Grupos. 3. El Usuario agregará la información del nuevo Grupo y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará el Grupo que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar". 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada uno de los Grupos de Activos. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. Mediante los botones de Navegación (Primero, Anterior, Siguiete y Ultimo), el Usuario puede navegar por todos los Grupos del Módulo de Activos Fijos. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 	
Postcondición:	La aplicación guardará los datos del Grupo del Módulo de Activos Fijos.	
Excepción:	<ol style="list-style-type: none"> 3. Si el Código del Grupo ya existe, se le notificará al Usuario que intente con otro Código. 	

Módulo de Impuesto Predial

Tabla 44. Caso de Uso – Editar Predios

Caso de Uso	Editar Predios
Actor(es):	<ul style="list-style-type: none"> • Administrador • Usuario Autorizado
Objetivos Asociados:	<ul style="list-style-type: none"> • OBJ-09
Requisitos Asociados:	<ul style="list-style-type: none"> • IRQ-10
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Impuesto Predial. El Sistema se comportará como se describe a continuación:
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Impuesto Predial". 2. En el Módulo de Impuesto Predial escoge el menú de Fichas y hace clic en "Predios" y el Sistema mostrará el formulario de las Fichas de los Predios. 3. El Usuario agregará la información del nuevo Predio y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará la Ficha del Predio que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar". 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada una de las Fichas de los Predios. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. Mediante los botones de Navegación (Primero, Anterior, Siguiente y Ultimo), el Usuario puede navegar por todas las Fichas del Módulo de Impuesto Predial. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar.
Postcondición:	La aplicación actualizará los datos de la Ficha del Módulo de Impuesto Predial.
Excepción:	<ol style="list-style-type: none"> 3. Si el Código del Predio ya existe, se le notificará al Usuario que intente con otro Código.

Módulo de Industria y Comercio

Tabla 45. Caso de Uso – Editar Establecimientos

Caso de Uso		Editar Establecimientos	
Actor(es):		<ul style="list-style-type: none"> • Administrador • Usuario Autorizado 	
Objetivos Asociados:		<ul style="list-style-type: none"> • OBJ-09 	
Requisitos Asociados:		<ul style="list-style-type: none"> • IRQ-10 	
Descripción:	Este caso de uso inicia cuando el Actor ingresa al Módulo de Industria y Comercio. El Sistema se comportará como se describe a continuación:		
Precondición:	El Actor debe haber ingresado su nombre de usuario y contraseña.		
Secuencia Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona el menú "Gestión" y hace clic en "Industria y Comercio". 2. En el Módulo de Industria y Comercio escoge el menú de Fichas y hace clic en "Establecimientos" y el Sistema mostrará el formulario de las Fichas de los Establecimientos. 3. El Usuario agregará la información del nuevo Establecimiento y dará clic en "Guardar". 4. El Usuario mediante la ventana de búsqueda seleccionará la Ficha del Establecimiento que desea editar y dará clic en "Modificar", en donde el Sistema le habilitará los campos que puede actualizar. 5. El Usuario cambiará la información necesaria y dará clic en "Guardar" 6. Mediante los botones de Edición (Nuevo, Modificar, Guardar y Buscar), el Usuario puede editar cada una de las Fichas de los Establecimientos. 7. El Sistema validará la información suministrada por el Usuario. 8. El Sistema guardará los datos e informará "Registro Almacenado Satisfactoriamente". 9. Mediante los botones de Navegación (Primero, Anterior, Siguiente y Ultimo), el Usuario puede navegar por todas las Fichas del Módulo de Industria y Comercio. 10. El Usuario dará clic en Cerrar o pulsará escape para finalizar. 		
Postcondición:	La aplicación actualizará los datos de la Ficha del Módulo de Industria y Comercio.		
Excepción:	<ol style="list-style-type: none"> 3. Si el Código del Establecimiento ya existe, se le notificará al Usuario que intente con otro Código. 		

4.4.4. Modelo de Datos

La Base de Datos se realizó a partir del concepto de la arquitectura Cliente/Servidor de dos niveles físicos. Este modelo está orientado a la conexión de terminales con servidores conectados a una red.

El caso escogido para desarrollar el Sistema, es el más común de este tipo de arquitecturas, en el cual la lógica de presentación y la lógica de negocio se establecen en la máquina cliente y la lógica de datos en el servidor.

El proceso de Diseño:

Para desarrollar el diseño de la Base de Datos en cada uno de los prototipos, se trabajó de la siguiente manera:

- Recopilación y organización de toda la información necesaria a través de la inducción del Sistema Administrativo Inteligente TIMAX por parte del Ingeniero Tutor.
- Análisis de la información recolectada. Se tomaron las tablas que involucraban las transacciones y se normalizaron hasta la tercera forma normal en los casos que se podía; a las nuevas tablas se les asignaron nombres más específicos para entender su función.
- Elección de las llaves primarias. Se identificaron el o los campos que representarían el identificador de cada tabla, siguiendo las reglas de integridad.
- Inspección de cada tabla. Se identificó la relación de los datos de una tabla con las demás, se crearon campos nuevos para establecer relaciones con otras tablas y al mismo tiempo se establecieron las llaves foráneas con sus correspondientes restricciones.

- Análisis de los diseños de cada prototipo. Se estudió cada diseño de la Base de Datos perteneciente a cada prototipo buscando detectar errores para su posterior corrección.

En el Anexo A se presentan los Diagramas de Entidad – Relación de las tablas que se implementaron en cada uno de los módulos del tercer prototipo.

En el Anexo B se presenta el Diccionario de Datos de las tablas anteriormente mencionadas.

4.4.5. Desarrollo de la Aplicación

La metodología aplicada para el desarrollo de la herramienta fue el Modelo Evolutivo Incremental, con el cual se completó el desarrollo de dos prototipos y un se inició un tercero que aún se encuentra en curso.

4.4.5.1 Actividades Desarrolladas

Una vez comprendido el contexto del Sistema Administrativo Inteligente TIMAX, esto es, la descripción de un modelo de negocio general de las empresas para el cual va dirigido esta herramienta administrativa, se levantaron las actas de requerimientos para los diferentes prototipos que se implementaron durante el desarrollo de la práctica.

También se generaron los diferentes diagramas de UML en cada uno de los prototipos como lo son los actores del sistema, los diagramas de Clases y los diagramas de Casos de Uso.

Se diseñaron los diferentes diagramas de Entidad – Relación de cada uno de los prototipos y se implementó la nueva Base de Datos con sus tablas aplicando las políticas del Sistema Relacional de Base de Datos a partir del diseño propuesto en conjunto con los demás compañeros de desarrollo del

proyecto el cual fué debidamente aprobado por el Jefe de Sistemas de la Empresa.

La nueva Base de Datos Relacional se creó a partir de la Normalización de las tablas universales existentes, en donde se cambian sus nombres para hacer más explícita su función, pero conservando el nombre de los campos que conforman dichas tablas, para no crear conflicto con la información en el momento de la migración a la nueva Base de Datos, lo cual podría implicar un importante retraso en el proceso de conversión.

Se generaron las Interfaces gráficas y los menús de acceso a los diferentes módulos del Sistema en cada uno de los prototipos entregados durante el desarrollo la aplicación.

Una vez entregados los prototipos, el Jefe de Sistemas se encargaba de hacer las pruebas de rigor, como lo son las pruebas de rendimiento de la aplicación, pruebas de optimización de funciones, detección de errores en la interfaz, pruebas de acceso a datos, pruebas de comportamiento entre otras; una vez realizada las pruebas y se detectaban los errores se hacían las correcciones necesarias y se entregaban nuevamente los prototipos para posteriores pruebas.

A continuación se presentan de una forma detallada todas las actividades que se realizaron en cada uno de los prototipos:

4.4.5.1.1. Primer Prototipo

El primer prototipo se inicia con el Módulo de Entrada de la aplicación, esto incluye primero la inducción y comprensión del manejo del Sistema en esta primera fase, el conocimiento de las tablas que involucran cada uno de los procesos de la entrada al Sistema y por último la generación de las Interfaces gráficas que se visualizarán en esta fase.

Antes de empezar el primer prototipo se hizo la instalación de las herramientas de desarrollo, como el Visual Studio 2005 en cada una de las terminales, también se acondicionó un equipo Servidor para el código fuente que incluía el

Sistema Operativo Windows Server 2003 y el Team Foundation Server de Visual Studio, este último es el encargado de manejar y controlar el código fuente, así como también, tiene control sobre las tareas asignadas, el avance del proyecto y las pruebas de carga masiva.

En otro equipo se instaló el Sistema Operativo Linux Fedora 6, en donde se habilitaron los servicios de Postgres, para luego utilizarlo como Servidor de Base de Datos.

Al mismo tiempo se investigó y se instaló un proveedor de datos ODBC para hacer el puente entre la Aplicación y la Base de Datos; este proveedor va a estar incluido en el paquete de instalación del Sistema.

Una vez instaladas todas las herramientas de desarrollo se realizaron las siguientes actividades:

- Como resultado del estudio de las tablas que involucran la entrada al Sistema, se creó una Base de Datos “modelo” llamada **antdbnet**, en donde se incluyeron cada una de las tablas que se fueran generando durante el desarrollo del proyecto. Las nuevas tablas generadas a partir de la Normalización en esta primera fase son: Empresas, Usuarios, SMenu, UsuMenu, Líneas, Marcas, Abreviaturas, LstItems, Prefijos, Tipos, entre otras.
- Se implementó una Clase en donde contiene todas las funciones para la conexión y transacciones a la Base de Datos; funciones para la validación de campos y funciones para implementación de formularios.
- Se desarrolló el módulo principal en donde se establecen todas las variables globales del Sistema, y por donde se inicializa la aplicación.
- Se generaron los formularios correspondientes a la Entrada del Sistema, como lo son: el formulario principal, el formulario de “splash” o

presentación, el de seleccionar empresa, el de autenticación del usuario, el de cambio de servidor, el de Administración de Cuentas de Usuarios y el de Prefijos.

4.4.5.1.2. Segundo Prototipo

Se implementaron las Fichas de cada uno de los Módulos más relevantes del Sistema, estos módulos son los de mayor trabajo en la mayoría de empresas en donde se encuentra instalado el Sistema y corresponden a los módulos de: Directorio, Contabilidad, Cuentas por Cobrar, Cuentas por Pagar, Inventarios, Tesorería, Activos Fijos, a excepción del Impuesto Predial que es exclusivo de las Empresas del sector oficial.

En esta segunda fase del desarrollo del proyecto se hizo lo siguiente:

- Se Normalizaron las tablas que comprenden las Fichas de los Módulos de Directorio, de Clientes, de Proveedores, de Tesorería, Fichas de Inventarios, de Activos Fijos e Impuesto Predial.
- Se crearon los diferentes formularios para guardar la información de las Fichas de cada uno de los Módulos, esto incluye también, la implementación de funciones para la validación de datos en el instante en que se digiten y la definición de los campos que requieran dicha validación.
- El Jefe de Sistemas hizo pruebas con datos reales para comprobar el buen funcionamiento del sistema, revisó todas las funciones para verificar que se estuvieran cumpliendo con los requisitos propuestos, he hizo pruebas para verificar que el rendimiento sea eficiente con un gran volumen de datos.
- Después de las pruebas se detectaron algunas falencias tanto de rendimiento como de funcionalidad, que se rectificaron, así como también algunas sugerencias que fueron implementadas.

4.4.5.1.3. Tercer Prototipo

Esta tercera fase se centra en los movimientos que presentan cada uno de los módulos, en esta fase se implementaron más funciones para las transacciones a la Base de Datos en la Clase General.

Las actividades desarrolladas fueron las siguientes:

- Primero se inicia con los movimientos de Tesorería, que consiste en la creación y edición de los comprobantes de ingreso; en el transcurso de este desarrollo se implementaron algunas tablas correspondiente a la entrada del sistema que se necesitaron para continuar esta tercera fase; después se continuó con el desarrollo de esta interfaz dejando pendiente algunos requisitos que deben implementarse primero en otros módulos, para luego utilizarlos en este formulario.
- Después se continuó con el desarrollo de los movimientos de Contabilidad, que comprenden los comprobantes contables de tipo ingreso, egreso y nota contable. De la misma manera que el anterior módulo, también se modificaron tablas que pertenecen a la entrada del Sistema y se normalizaron las tablas involucradas en las transacciones de los comprobantes contables. Cabe resaltar que esta interfaz no quedó totalmente terminada, puesto que algunos requisitos deben implementarse en otros módulos para después poderlos utilizar en esta interfaz, por ejemplo: en el caso que el comprobante maneje un artículo, se necesita que esté implementado en la entrada de artículos esto corresponde a los movimientos del módulo de inventarios.
- Por decisión de los coordinadores del proyecto se decide suspender la implementación de los movimientos en los módulos, para crear el proyecto que contiene el generador del instalador de la aplicación. Esto es para que cuando se haga el proceso de pruebas de versiones se genere siempre un instalador completo con la versión del Sistema.

Se debe destacar que la aplicación no fue terminada en su totalidad, pero se cumplió la mayoría del cronograma establecido para el tiempo de práctica, el cual es muy corto para la magnitud de este proyecto, sin embargo actualmente se sigue trabajando en desarrollo del mismo.

También se desarrollaron otras actividades adicionales como el mantenimiento correctivo y soporte técnico de los sistemas de Acueducto y Alcantarillado, Impuesto Predial e Industria y Comercio que actualmente se encuentran instalados en diferentes empresas, y aunque no hacen parte del proyecto, sí contribuyeron con un retraso del desarrollo del mismo.

Además se debe resaltar el retraso que tuvo el proyecto por lapso de un mes, porque la empresa me envió a las alcaldías en donde tiene instalado el Módulo de Impuesto Predial e Industria y Comercio, para dar el correspondiente soporte técnico y montaje de la liquidación de los predios y su respectiva facturación.

5. CONCLUSIONES

- El desarrollo de proyectos de grado en un medio real, brinda la oportunidad a los estudiantes de poner en práctica los conocimientos alcanzados durante la formación académica, obtener conocimiento a través de la investigación de temas propios para el desarrollo del proyecto, así como adquirir experiencia de las situaciones presentadas en el transcurso de su realización.
- La utilización de metodologías avanzadas de desarrollo de software, así como el uso de herramientas y lenguajes de diseños unificados (UML), facilitan las labores en las diferentes etapas de realización de un proyecto y reduce el tiempo de ejecución, esto implica una disminución en los costos y cumplimiento de los cronogramas de actividades.
- La tecnología .NET es una excelente herramienta para la construcción de aplicaciones software, porque que facilita las labores de programación, se acopla a las metodologías de desarrollo de aplicaciones actuales, y el soporte que se encuentra en internet tanto en la página de MSDN de Microsoft como en los foros ayuda a la solución de problemas.
- El motor de base de datos de Postgres, es una aplicación excelente para el manejo de los datos, porque facilitan su uso y garantizan la seguridad tanto en las transacciones como en la información que almacena.
- La aplicación de estándares de programación mejora el mantenimiento del software durante todas las etapas del desarrollo del mismo y se presenta como una opción para dar calidad al código de un producto software.

- Encargarse de actividades que implican el contacto con clientes de la empresa, desarrolla en el estudiante, habilidades en relaciones públicas, muy importantes dentro del medio laboral que nos ayudan a la formación tanto personal como profesional.

6. RECOMENDACIONES

- Es importante que se continúe con la terminación del prototipo actual y el desarrollo de los demás prototipos, para tener un excelente producto software que satisfaga todas las necesidades de los clientes y contribuir al progreso de la empresa.
- Desarrollar una versión Web o algunos de sus módulos en versión Web como por ejemplo, el módulo de Impuesto Predial, para el caso en que alguna alcaldía desee tener toda la información correspondiente al valor del impuesto predial de los predios de su municipio en alguna página Web, para que los usuarios puedan consultarlos; este módulo Web debe permitir interactuar en línea con la información residente en el servidor de Base de Datos dentro de la aplicación Cliente – Servidor.
- Para un mejor avance en el desarrollo del proyecto es conveniente tener un mayor número de programadores en esta fase, teniendo en cuenta la cantidad de tareas que existen actualmente para cada uno de los módulos y por la complejidad del Sistema Administrativo Inteligente TIMAX.
- Dedicar un grupo de programadores exclusivamente para el desarrollo del proyecto y otro para el soporte técnico de los productos que actualmente están desarrollados, y con esto poder ofrecer un mejor servicio a los clientes y evitar retrasos en el desarrollo del proyecto.

BIBLIOGRAFÍA

LIBROS

PRESSMAN, Roger S., Ingeniería del Software un enfoque práctico, Edición No. 5, España, Mc Graw Hill, 2002.

BALENA, Francesco, Programación Avanzada con Microsoft Visual Basic .NET, Mc Graw Hill Interamericana, 2003.

DATE, C. J., Introducción a los Sistemas de Bases de Datos, Séptima Edición, Prentice Hall, 2001.

BLANCO, Luis Miguel, Programación en Visual Basic .NET, Grupo EIDOS, 2002.

PROYECTOS DE GRADO

Sistema Software para Manejar el Área Académica del Colegio Andrés Páez Sotomayor.

Desarrollo de un Sistema de Información Extranet para la Empresa Quirón Software Versión 1.0.

TECNOLOGÍAS Y HERRAMIENTAS

MOMJIAN, Bruce. PostgreSQL Introduction and Concepts, Addison-Wesley, USA 2001.

LOCKHART, Thomas. The PostgreSQL Programmer's Guide. The PostgreSQL Global Development Group, USA 1998.

LOCKHART, Thomas. The PostgreSQL Developer's Guide. The PostgreSQL Global Development Group, USA 1998.

SITIOS EN INTERNET

<http://microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/default.asp>

<http://elguille.info/NET/cursoVB.NET/indice.html>

<http://microsoft.com/spanish/msdn/vs2005/default.aspx>

<http://www.postgresql.org/docs/admin>

<http://www.postgresql.org/docs/programmer>

<http://www.postgresql.org/docs/tutorial>

<http://www.postgresql.org/docs/user>

ANEXO A. DIAGRAMA ENTIDAD-RELACION DE LA BASE DE DATOS

Figura 27. Diagrama Entidad – Relación de la Entrada del Sistema

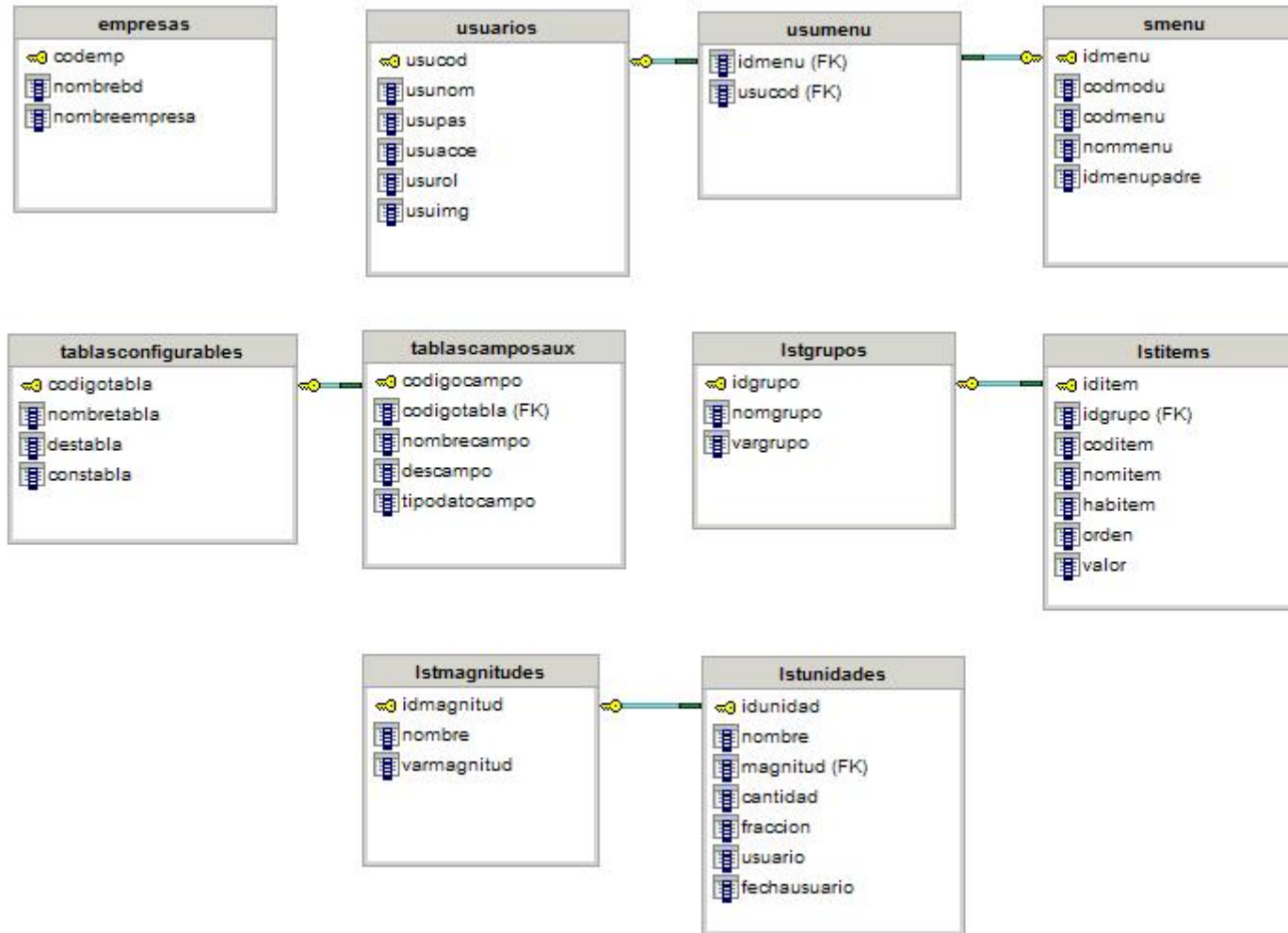


Figura 28. Diagrama Entidad – Relación Módulo Usuarios

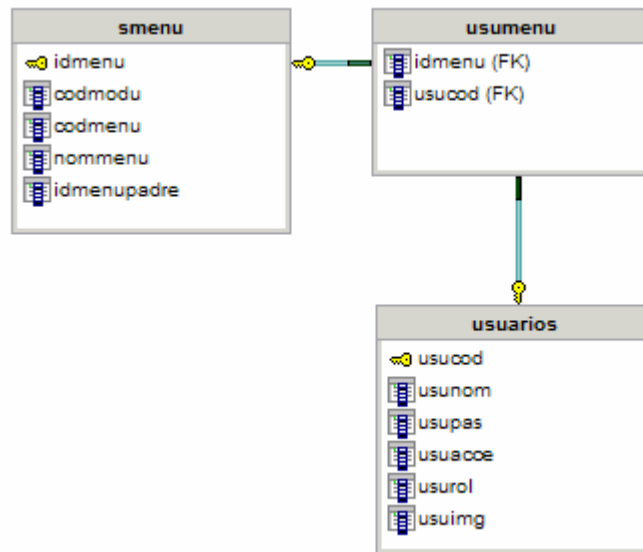


Figura 29. Diagrama Entidad – Relación Módulo Directorio

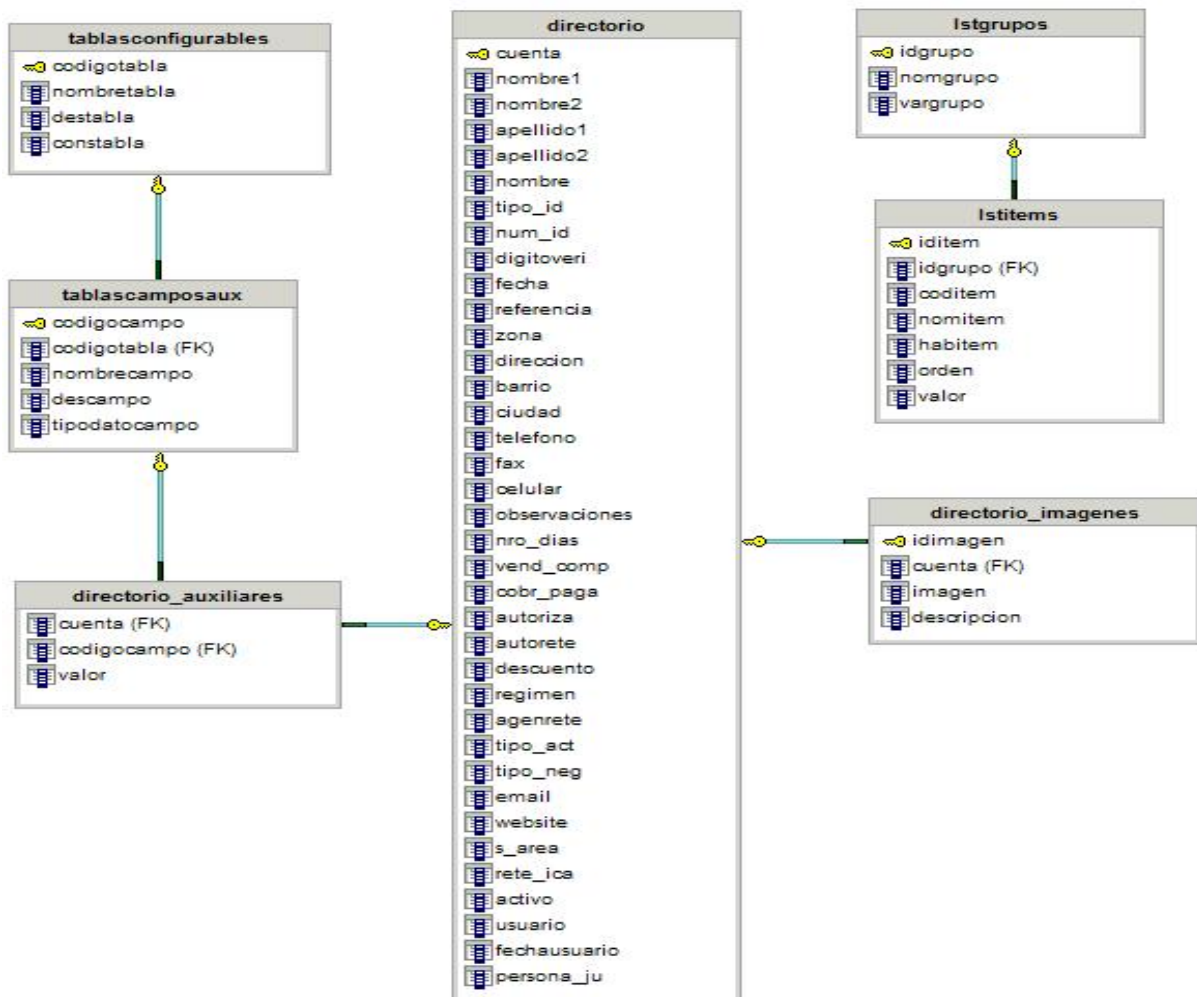


Figura 30. Diagrama Entidad – Relación Módulo Contabilidad

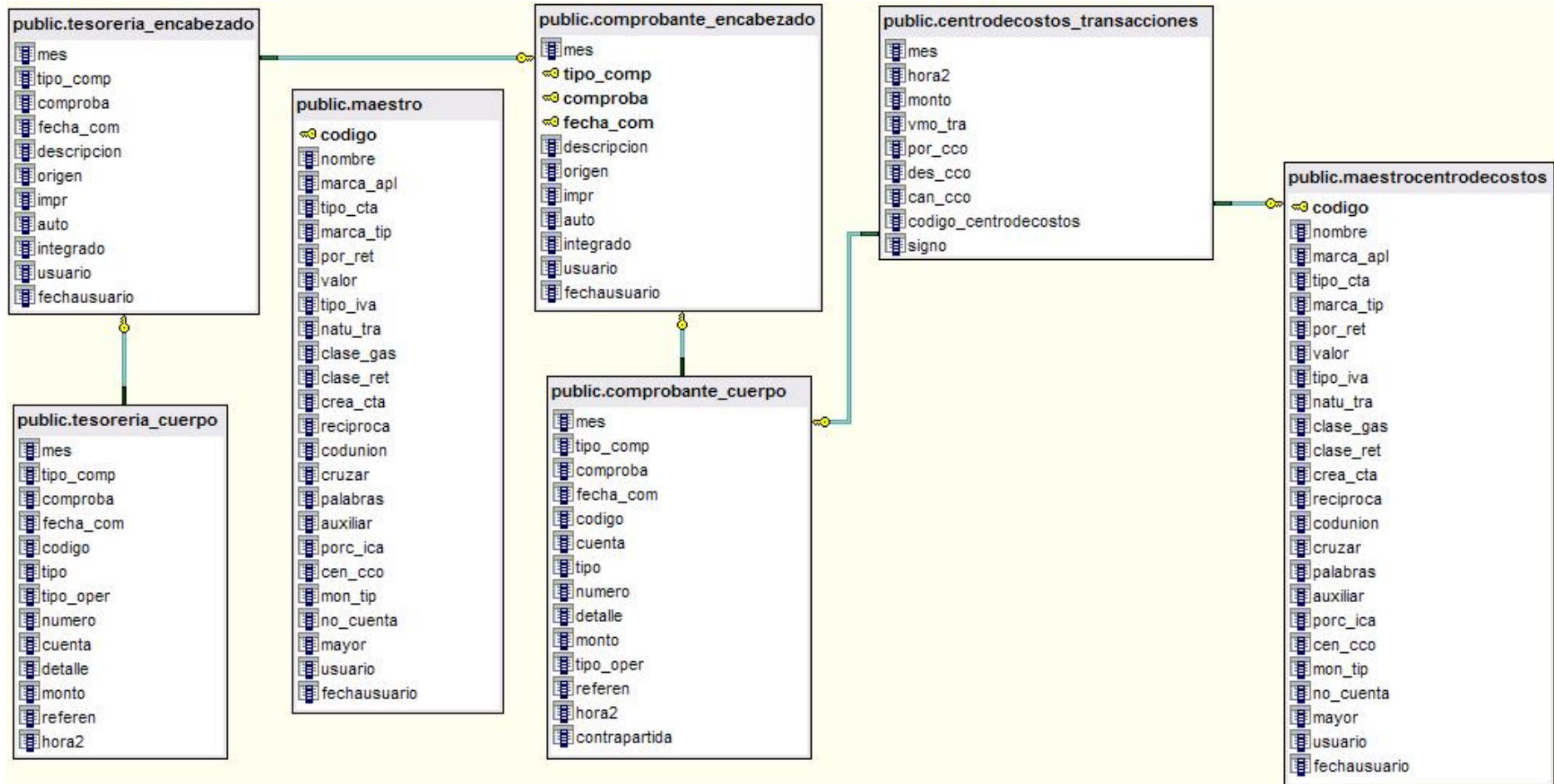


Figura 31. Diagrama Entidad – Relación Módulo Cuentas por Cobrar (Fichas)

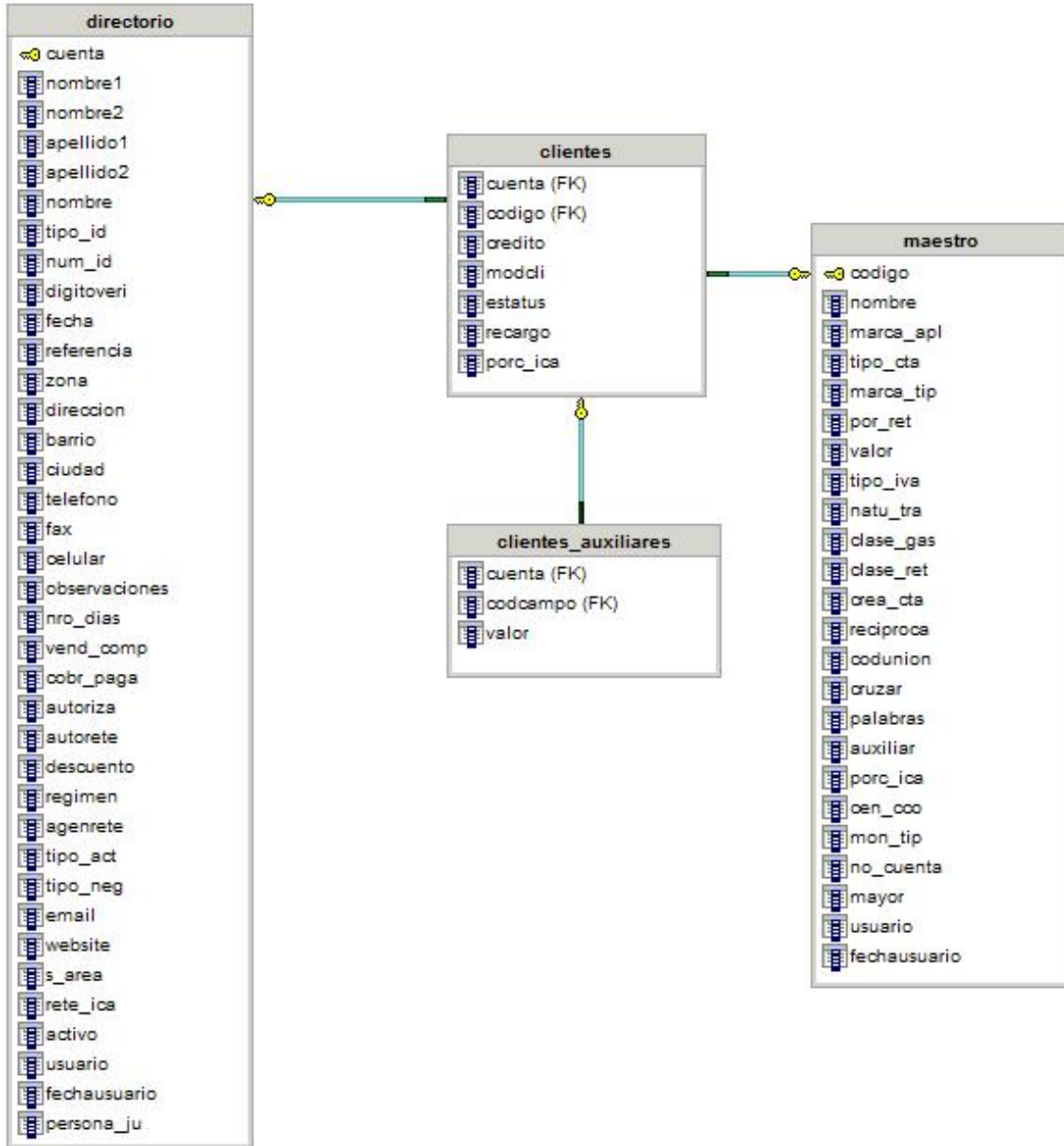


Figura 32. Diagrama Entidad – Relación Módulo Cuentas por Pagar (Fichas)

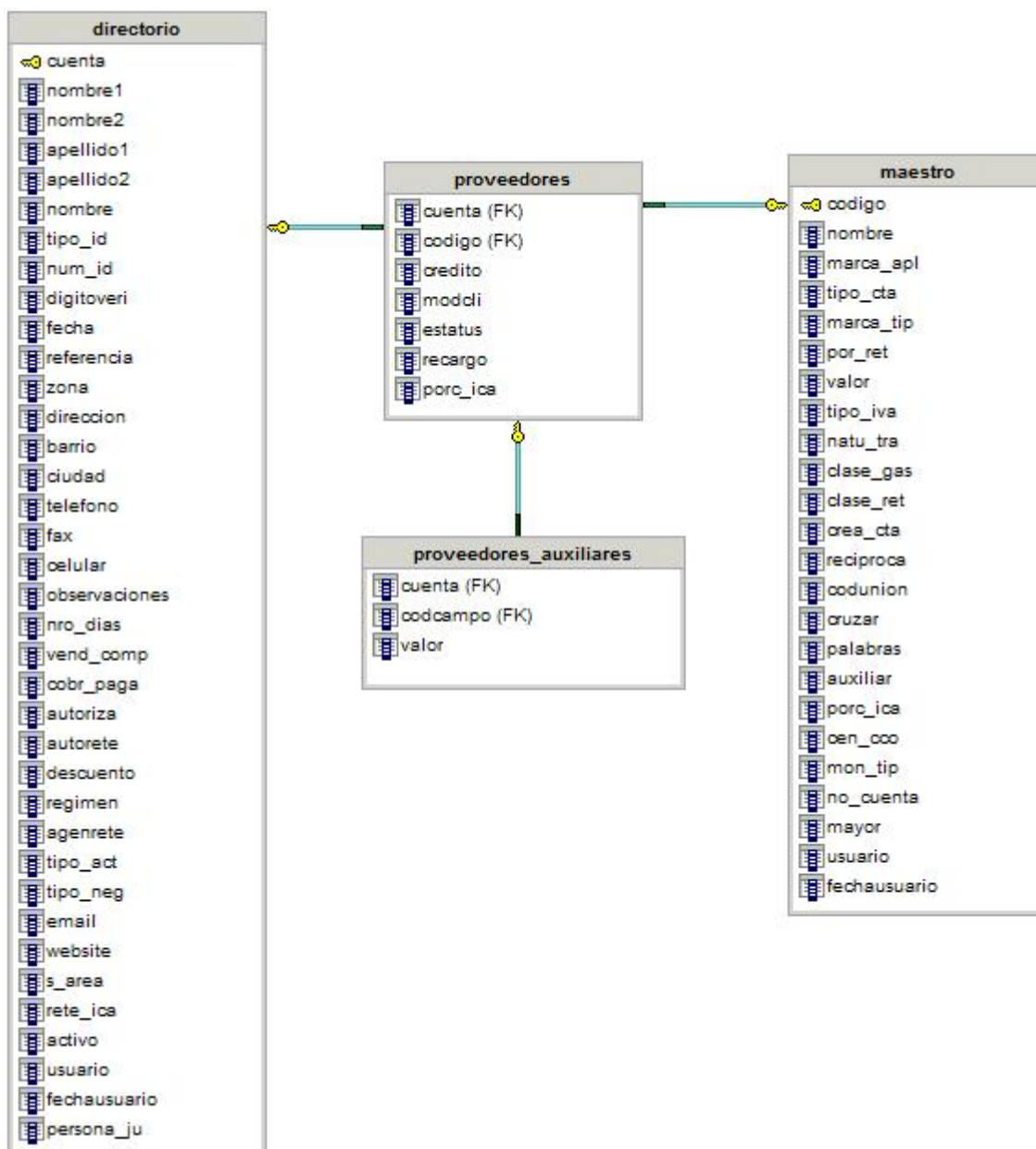


Figura 32. Diagrama Entidad – Relación Módulo Inventarios (Fichas)

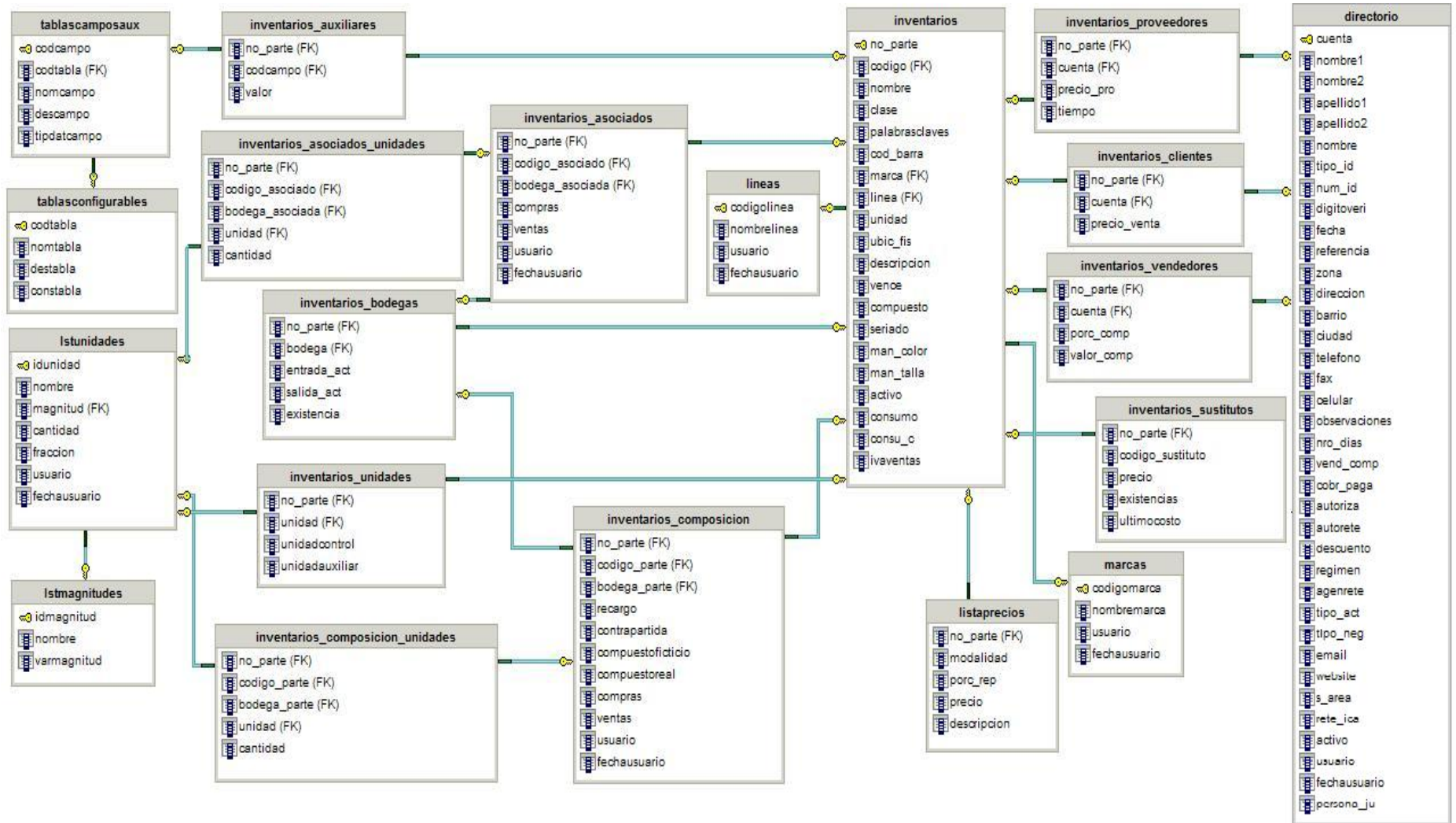


Figura 33. Diagrama Entidad – Relación Módulo Activos Fijos (Fichas)

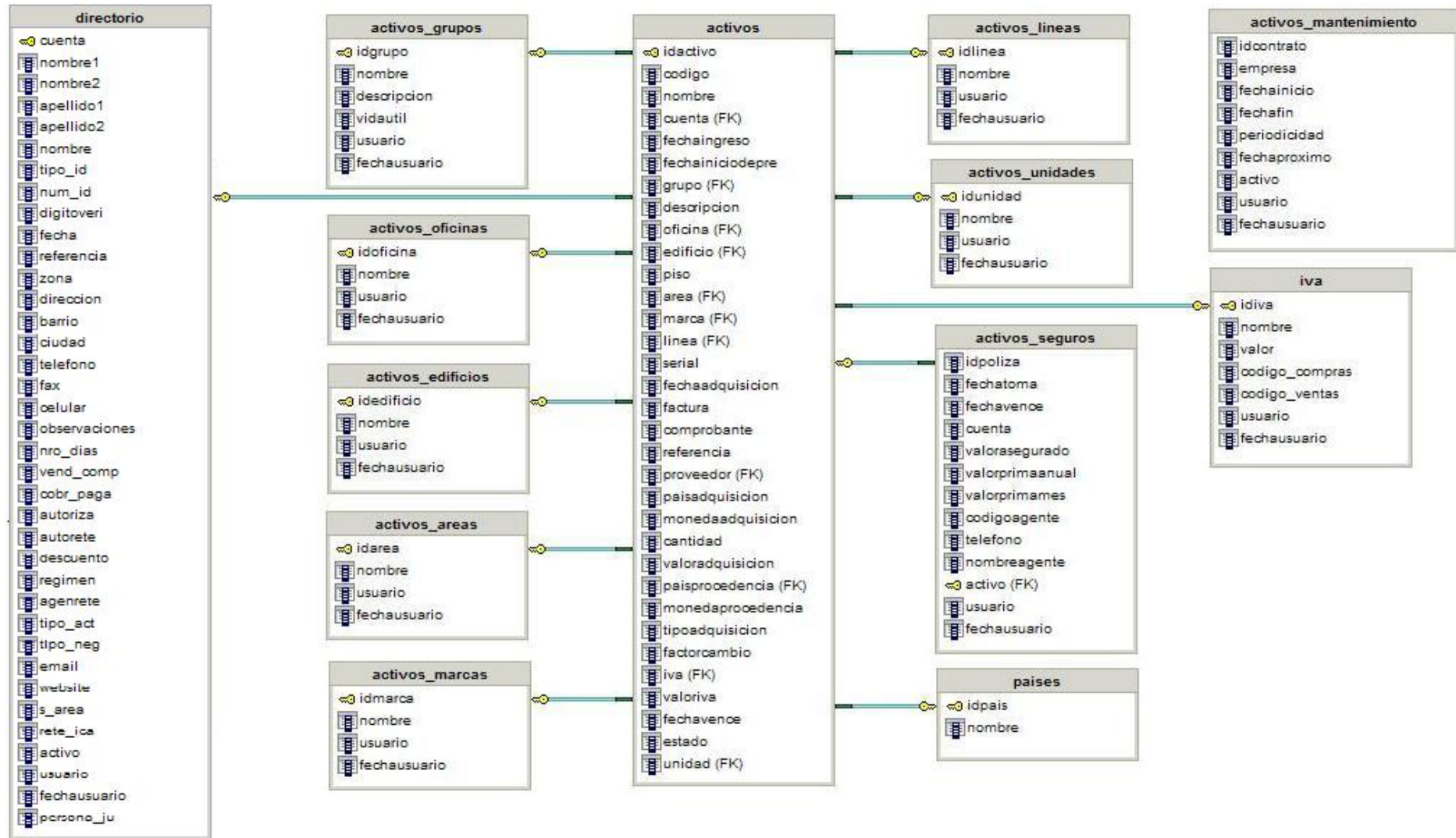


Figura 31. Diagrama Entidad – Relación Módulo Impuesto Predial (Fichas)

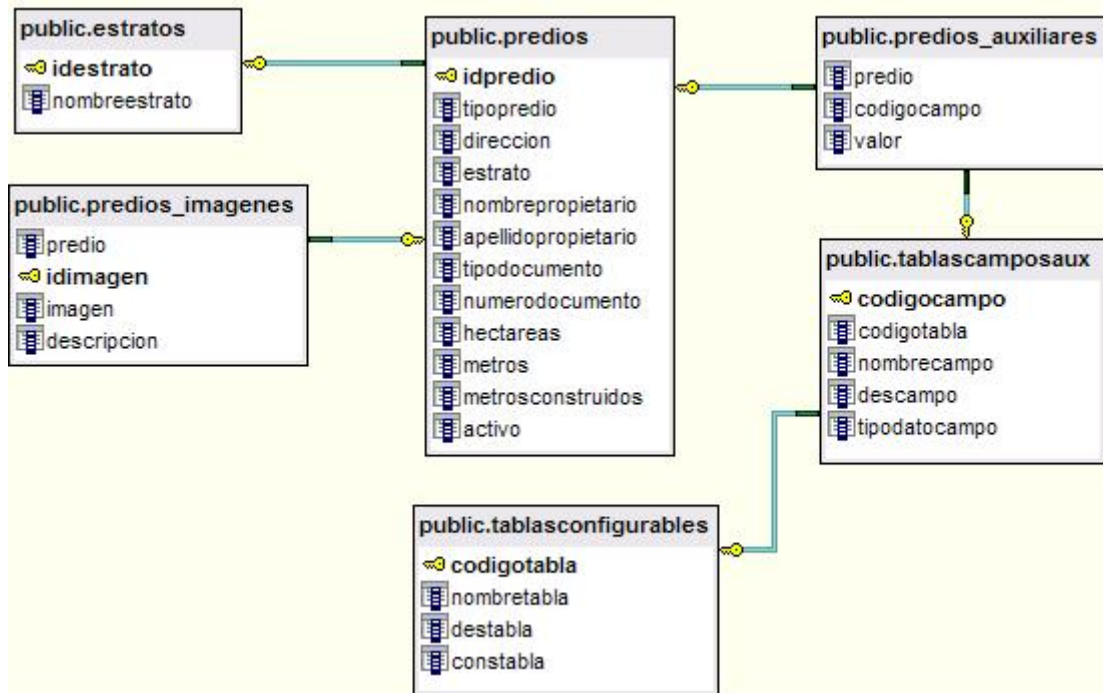
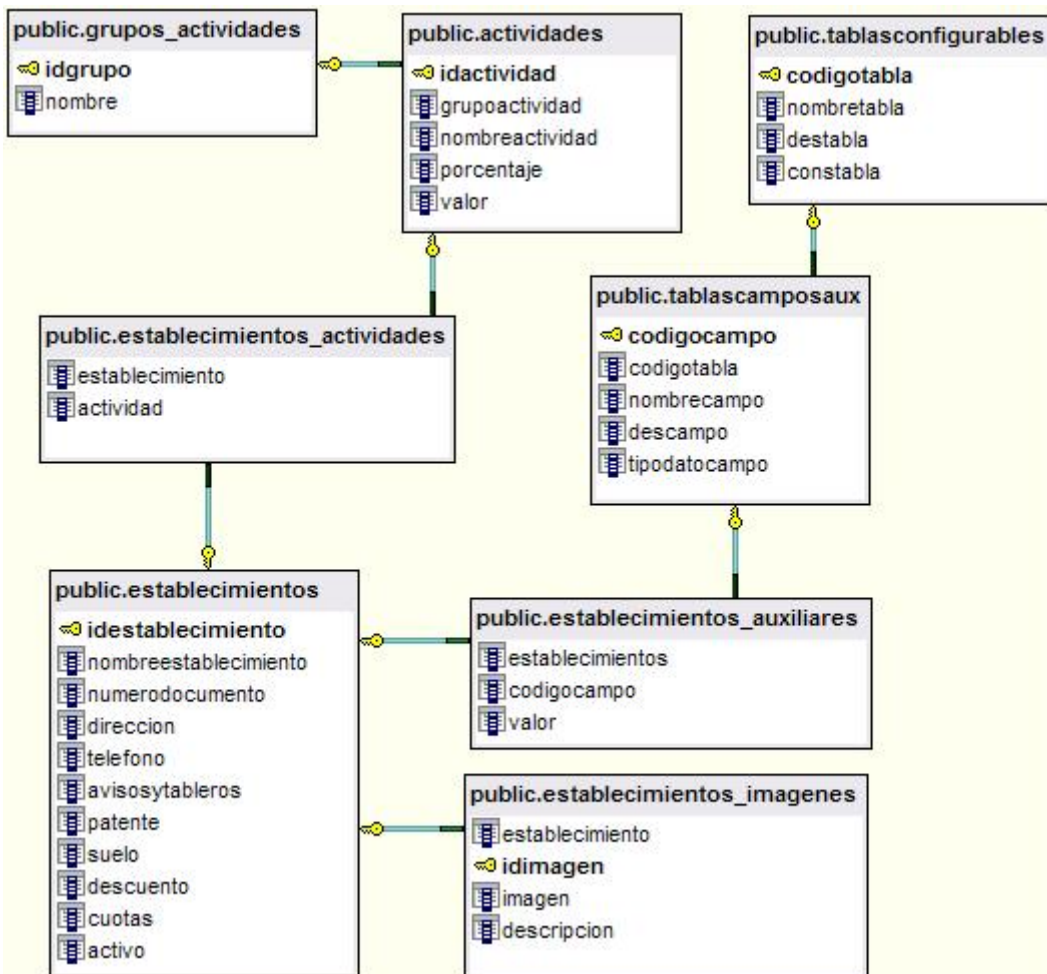


Figura 32. Diagrama Entidad – Relación Módulo Industria y Comercio (Fichas)



ANEXO B. DICCIONARIO DE DATOS DE LA BASE DE DATOS

Tabla 46. Diccionario de Datos – Empresas

NOMBRE	TIPO	DESCRIPCIÓN
codemp	integer	Código de la Empresa
nombred	varchar(8)	Nombre corto de la Base de Datos
nombre empresa	varchar	Nombre largo de la Empresa

Tabla 47. Diccionario de Datos – Usuarios

NOMBRE	TIPO	DESCRIPCIÓN
usucod	varchar(20)	Código del Usuario
usunom	varchar(250)	Nombre del Usuario
usupas	varchar	Clave del Usuario
usuacce	boolean	Define si el Usuario está activo
usurol	boolean	Perfil del Usuario
usuimg	bytea	Foto del Usuario

Tabla 48. Diccionario de Datos – UsuParam

NOMBRE	TIPO	DESCRIPCIÓN
idmenu	integer	Código del Menú
usucod	varchar(20)	Código del Usuario

Tabla 49. Diccionario de Datos – SMenu

NOMBRE	TIPO	DESCRIPCIÓN
idmenu	integer	Código consecutivo del registro del menú
codmodu	varchar(15)	Módulo al que pertenece el menú
codmedu	varchar(50)	Código del menú programable
nommenu	varchar(150)	Nombre del menú
idmenupadre	Integer	Menú padre al que pertenece el menú

Tabla 50. Diccionario de Datos – Consecutivos

NOMBRE	TIPO	DESCRIPCIÓN
mes	varchar(2)	Mes para los consecutivos
nc_comproba	varchar(12)	Consecutivo para las notas contables
in_comproba	varchar(12)	Consecutivo para los ingresos
eg_comproba	varchar(12)	Consecutivos para los egresos
en_numero	varchar(12)	Consecutivos para las entradas

Tabla 51. Diccionario de Datos – TablasConfigurables

NOMBRE	TIPO	DESCRIPCIÓN
codigotabla	integer	Código de Identificación de la tabla configurable
nombretabla	varchar(12)	Nombre de la tabla
destabla	varchar(12)	Descripción detallada de la tabla
constabla	varchar(12)	Valor Constante de la tabla

Tabla 52. Diccionario de Datos – TablasCamposAux

NOMBRE	TIPO	DESCRIPCIÓN
codigocampo	Integer	Código identificador del campo auxiliar
codigotabla	Integer	Código de la tabla configurable
nombrecampo	varchar(25)	Nombre del campo auxiliar
descampo	varchar(200)	Descripción detallada del campo auxiliar
tipodatocampo	varchar(200)	Tipo de dato que se almacena en este campo auxiliar

Tabla 53. Diccionario de Datos – LstGrupos

NOMBRE	TIPO	DESCRIPCIÓN
idgrupo	integer	Id del grupo de listas
nomgrupo	varchar(150)	Nombre del grupo de listas
vargrupo	varchar(20)	Variable de grupo

Tabla 54. Diccionario de Datos – LstItems

NOMBRE	TIPO	DESCRIPCIÓN
iditem	integer	Identificador del ítem de la lista
idgrupo	integer	Grupo al que pertenece el ítem
coditem	varchar(30)	Código del ítem
nomitem	varchar(150)	Nombre del ítem de la lista
habitem	boolean	Define si el ítem está habilitado
orden	integer	Establece el orden de prioridad del ítem
valor	double	(Opcional) Almacena el valor del ítem

Tabla 55. Diccionario de Datos – IVA

NOMBRE	TIPO	DESCRIPCIÓN
idiva	varchar(1)	Código del tipo de IVA
nombre	varchar(15)	Nombre del tipo de IVA
valor	double	Valor del porcentaje del tipo de IVA
codigo_compras	varchar(40)	Código contable para compras
codigo_ventas	varchar(40)	Código contable para ventas
usuario	varchar(20)	Usuario del Sistema responsable de modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 56. Diccionario de Datos – LstMagnitudes

NOMBRE	TIPO	DESCRIPCIÓN
Idmagnitud	varchar(4)	Código Identificador de la Magnitud
nombre	varchar(15)	Descripción de la Magnitud
varmagnitud	varchar(15)	Variable Global para la unidad de la magnitud
eg_comproba	varchar(12)	Consecutivos para los egresos
en_numero	varchar(12)	Consecutivos para las entradas

Tabla 57. Diccionario de Datos – LstUnidades

NOMBRE	TIPO	DESCRIPCIÓN
idunidad	varchar(3)	Código Identificador de la Unidad de Medida
nombre	varchar(30)	Descripción de la Unidad de Medida
magnitud	varchar(4)	Identificador del grupo al cual pertenece
cantidad	double	Cantidad de piezas que conforman la Unidad
fraccion	varchar(1)	Cantidad que ocupa en la Unidad
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 58. Diccionario de Datos – Abreviaturas

NOMBRE	TIPO	DESCRIPCIÓN
abreviatura	varchar(5)	Código identificador de la abreviatura
descripcion	varchar(250)	Descripción detallada de la abreviatura

Tabla 59. Diccionario de Datos – Tallas

NOMBRE	TIPO	DESCRIPCIÓN
idtalla	varchar(5)	Código Identificador de la Talla
nombre	varchar(30)	Descripción o nombre de la Talla
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 60. Diccionario de Datos – ListaPrecios

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código Identificador del Artículo, Servicio, Gasto
modalidad	varchar(1)	Modalidad de Precio (A, B, C, D...)
porc_rep	double	Porcentaje o rentabilidad del Precio
precio	double	Valor de la modalidad del Precio
descripcion	varchar(100)	Descripción de la modalidad del Precio

Tabla 61. Diccionario de Datos – Directorio

NOMBRE	TIPO	DESCRIPCION
cuenta	varchar(15)	Código Identificador de la Ficha del Directorio
nombre1	varchar(20)	Primer nombre de la Ficha
nombre2	varchar(20)	Segundo nombre de la Ficha
apellido1	varchar(20)	Primer apellido de la Ficha
apellido2	varchar(20)	Segundo apellido de la Ficha
nombre	varchar(80)	Nombre completo de la Ficha
tipo_id	varchar(2)	Tipo de identificación (CC,CE,NI,NE)
numero_id	varchar(25)	Número del documento de identificación
digitoveri	varchar(1)	Dígito de verificación del número de identificación
fecha	date	Fecha del registro de la Ficha
referencia	varchar(30)	Código para vincular la Ficha a inventarios
zona	varchar(10)	Código de la Zona
direccion	varchar(200)	Dirección de la Ficha
barrio	varchar(100)	Barrio de la Ficha
ciudad	varchar(5)	Código de la ciudad de la Ficha
telefono	varchar(20)	Teléfono(s) de la Ficha
fax	varchar(20)	Fax(es) de la Ficha
celular	varchar(12)	Número de celular de la Ficha
observaciones	varchar(100)	Anotaciones adicionales de la Ficha
nro_dias	varchar(3)	Días de crédito que tiene la Ficha
regimen	varchar(1)	Indica el Tipo de Régimen de la Ficha
vend_comp	varchar(15)	Código de la Ficha de Cuentas por Cobrar
cobr_paga	varchar(15)	Código de la Ficha de Cuentas por Pagar
autoriza	varchar(15)	Código de la Ficha que le autoriza crédito
descuento	double	Porcentaje del descuento al que tiene la Ficha
autorete	boolean	Define si la Ficha es autoretenedor
agenrete	boolean	Establece si la Ficha es Agente Retenedor
tipo_act	varchar(3)	Indica el Tipo de Actividad de la Ficha
tipo_neg	varchar(5)	Define el Tipo de Negocio de la Ficha
email	varchar(100)	Correo Electrónico
website	varchar(100)	Página Web
s_area	varchar(10)	Area donde se Ubica a la Ficha
persona_ju	boolean	Indica si la Ficha es Persona jurídica
rete_ica	varchar(1)	Define si el Reteica es un valor mínimo o no
activo	boolean	Indica si la Ficha esta Activa o Inactiva
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 62. Diccionario de Datos – Directorio_Auxiliares

NOMBRE	TIPO	DESCRIPCIÓN
cuenta	varchar(15)	Código de la Ficha del Directorio
codigocampo	integer	Código del Campo Auxiliar
valor	varchar(200)	Valor del Campo Auxiliar

Tabla 63. Diccionario de Datos – Directorio_Imagenes

NOMBRE	TIPO	DESCRIPCIÓN
idimagen	integer	Código Identificador de la cada Imagen de la Ficha del Directorio
cuenta	varchar(15)	Código de la Ficha del Directorio
imagen	bytea	Imagen de la Ficha del Directorio
descripción	varchar(200)	Descripción detallada de la imagen de la Ficha

Tabla 64. Diccionario de Datos – Tesoreria_Encabezado

NOMBRE	TIPO	DESCRIPCIÓN
mes	varchar(2)	Mes en que se realiza la Transacción
tipo_comp	varchar(2)	Tipo de Comprobante IN: Ingreso, EG: Egreso, NC: Nota Contable
comproba	varchar(12)	Número del Comprobante
fecha_com	date	Fecha de Emisión del Comprobante
descripcion	text	Descripción detallada del Comprobante
origen	varchar(30)	Nombre del Modulo que Genera el Comprobante
impr	boolean	Valor que Indica si se imprimió el Comprobante
auto	boolean	
integrado	boolean	Indica si fue integrado el Comprobante a Contabilidad
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 65. Diccionario de Datos – Tesoreria_Cuerpo

NOMBRE	TIPO	DESCRIPCIÓN
mes	varchar(2)	Mes en que se realiza la Transacción
tipo_comp	varchar(2)	Tipo de Comprobante IN: Ingreso, EG: Egreso, NC: Nota Contable
comproba	varchar(12)	Número del Comprobante
fecha_com	date	Fecha de Emisión del Comprobante
codigo	varchar(40)	Código Contable de Caja o Bancos
origen	varchar(30)	Nombre del Modulo que Genera el Comprobante
tipo	varchar(2)	Tipo de Operación EC: Entrada Caja, SC: Salida Caja, DP: Deposito
tipo_oper	varchar(2)	Forma de Pago CH: Cheque, FE: Efectivo, TJ: Tarjeta
numero	varchar(12)	Número del Documento de Pago
Cuenta	varchar(15)	Código de la Ficha del Directorio que realizo la Transacción
detalle	text	Detalle del Documento de Pago
monto	double	Monto parcial del Documento de Pago
hora2	integer	Código identificador que enlaza con Centro de Costos

Tabla 66. Diccionario de Datos – Maestro

NOMBRE	TIPO	DESCRIPCIÓN
codigo	varchar(100)	Código Contable. Identificador del Plan Unico de Cuentas (PUC)
nombre	varchar(250)	Nombre del Código Contable
marca_apl	varchar(3)	Para marcar el Código Contable para los Formularios
tipo_cta	varchar(1)	Marcación que indica que el Código Contable es una cuenta para Ajustes
marca_tip	varchar(3)	Define el Tipo de Retención
por_ret	double	Establece el porcentaje de Retención que se aplica
valor	double	Si la cuenta está enlazada a Inventarios, indica el valor del Artículo
tipo_iva	varchar(5)	Establece el tipo de IVA que se le aplica al Código Contable
natu_tra	varchar(5)	naturaleza de las transacciones que aplica
clase_gas	varchar(5)	Indica la Clase de gasto
clase_ret	varchar(5)	Define la clase de retención
crea_cta	varchar(1)	Indica la Base imponible
reciproca	varchar(1)	indica se es una Código contable para contrapartidas
codunion	varchar(9)	Se utiliza para enlazarse con el Directorio en el caso que la Ficha sea de Régimen Gubernamental
cruzar	varchar(1)	Por investigar
palabras	varchar(250)	Descripción larga del Código Contable
auxiliar	varchar(250)	(Opcional), este campo se utiliza cuando existen códigos auxiliares
porc_ica	double	Porcentaje del Reteica aplicado a este Código Contable
cen_cco	boolean	Indica si el Código se enlaza con el módulo de Centro de Costos
mon_tip	varchar(1)	Indica el tipo de moneda L:Local, T:Transiccion
no_cuenta	varchar(50)	Contiene el número de la Cuenta Bancaria en el caso que el Código pertenezca a Bancos
mayor	boolean	Anotaciones adicionales de la Ficha
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro

Tabla 67. Diccionario de Datos – Comprobante_Encabezado

NOMBRE	TIPO	DESCRIPCIÓN
mes	varchar(2)	Mes en que se realiza la Transacción
tipo_comp	varchar(2)	Tipo de Comprobante IN: Ingreso, EG: Egreso, NC: Nota Contable
comproba	varchar(12)	Número del Comprobante
fecha_com	date	Fecha de Emisión del Comprobante
descripcion	text	Descripción detallada del Comprobante
origen	varchar(30)	Nombre del Módulo que Genera el Comprobante
impr	boolean	Valor que Indica si se imprimió el Comprobante
auto	boolean	Si la integración es automática
integrado	boolean	Indica si fue integrado el Comprobante a Contabilidad
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 68. Diccionario de Datos – Comprobante_Cuerpo

NOMBRE	TIPO	DESCRIPCIÓN
mes	varchar(2)	Mes en que se realiza la Transacción
tipo_comp	varchar(2)	Tipo de Comprobante IN: Ingreso, EG: Egreso, NC: Nota Contable
comproba	varchar(12)	Número del Comprobante
fecha_com	date	Fecha de Emisión del Comprobante
codigo	varchar(40)	Código Contable de la Transacción
origen	varchar(30)	Nombre del Modulo que Genera el Comprobante
tipo	varchar(2)	Tipo del Documento EC: Entrada Caja, SC: Salida Caja, DP: Deposito
tipo_oper	varchar(2)	Tipo de Operación del registro
numero	varchar(12)	Número del Documento
cuenta	varchar(15)	Código de la Ficha del Directorio que realizo la Transacción
detalle	text	Detalle del Documento
monto	double	Monto parcial del Documento
hora2	integer	Código identificador que enlaza con Centro de Costos
contrapartida	boolean	Código de contrapartida para los Códigos Contables

Tabla 69. Diccionario de Datos – CentrodeCostos_Transacciones

NOMBRE	TIPO	DESCRIPCIÓN
mes	varchar(2)	Mes en que se realiza la Transacción
hora2	integer	Llave foránea que se enlaza con la tabla comprobante_cuerpo
monto	double	Valor que se le asigna al Centro de Costos
vmo_tra	double	Contiene el mismo valor del Monto
por_cco	double	Porcentaje asignado al Centro de Costos
des_cco	double	Valor de descuento que se le asigna al Centro de Costos
can_cco	double	Cantidad de Unidades que conforman el Centro de Costo
código_centrodecostos	varchar(40)	Código para el Centro de Costos
signo	integer	Define el signo del monto (+1, -1)

Tabla 70. Diccionario de Datos – Clientes

NOMBRE	TIPO	DESCRIPCIÓN
cuenta	varchar(15)	Llave foránea a la tabla Directorio
codigo	varchar(40)	Código Contable que enlaza a la tabla Maestro
crédito	double	Límite de Crédito que tiene aprobado el Cliente
mod_cli	varchar(1)	Tipo de modalidad de precio pactado con el Cliente
estatus	varchar(2)	Estado en que se encuentra el Cliente A:Activo, D:Deuda
recargo	varchar(1)	Tipo de IVA que se le aplica al Cliente
porc_ica	double	Porcentaje de Reteica

Tabla 71. Diccionario de Datos – Clientes_Auxiliares

NOMBRE	TIPO	DESCRIPCIÓN
cuenta	varchar(15)	Código de la Ficha del Cliente
codigocampo	integer	Código del Campo Auxiliar
valor	varchar(200)	Valor del Campo Auxiliar

Tabla 72. Diccionario de Datos – Proveedores

NOMBRE	TIPO	DESCRIPCIÓN
cuenta	varchar(15)	Llave foránea a la tabla Directorio
codigo	varchar(40)	Código Contable que enlaza a la tabla Maestro
crédito	double	Límite de Crédito pactado con el Proveedor
mod_cli	varchar(1)	Tipo de modalidad de precio pactado con el Proveedor
estatus	varchar(2)	Estado en que se encuentra el Proveedor A:Activo, D:Deuda
recargo	varchar(1)	Tipo de IVA que aplica el Proveedor
porc_ica	double	Porcentaje de Reteica

Tabla 73. Diccionario de Datos – Proveedores_Auxiliares

NOMBRE	TIPO	DESCRIPCIÓN
cuenta	varchar(15)	Código de la Ficha del Proveedor
codigocampo	integer	Código del Campo Auxiliar
valor	varchar(200)	Valor del Campo Auxiliar

Tabla 74. Diccionario de Datos – Inventarios_Auxiliares

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
codigocampo	integer	Código del Campo Auxiliar
valor	varchar(200)	Valor del Campo Auxiliar

Tabla 75. Diccionario de Datos – Inventarios_Asociados

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha principal de Inventarios
codigo_asociado	varchar(30)	Código del artículo asociado al no_parte
bodega_asociada	varchar(15)	Código de la Bodega al cual pertenece
compras	boolean	Si utiliza compras
ventas	boolean	Si utiliza ventas
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 76. Diccionario de Datos – Inventarios_Asociados_Unidades

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
codigo_asociado	varchar(30)	Código de la Ficha de Inventarios asociado al no_parte
bodega_asociado	varchar(15)	Código de la bodega donde pertenece el codigo_asociado
unidad	varchar(3)	Código de la Unidad de Costeo o de Control
cantidad	double	Cantidad de Unidades

Tabla 77. Diccionario de Datos – Inventarios_Bodegas

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
bodega	varchar(15)	Código de la bodega donde pertenece la Ficha de Inventarios
entrada_act	double	Cantidad Fichas que ingresa a la Bodega
salida_act	double	Cantidad Fichas que salen de la Bodega
existencia	double	Cantidad de Fichas que hay disponibles en la Bodega

Tabla 78. Diccionario de Datos – Inventarios_Unidades

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
unidad	varchar(3)	Código de la Unidad que se le asigna
unidadcontrol	boolean	Si es Unidad de Control
unidadauxiliar	boolean	Si la Unidad es Auxiliar

**Tabla 79. Diccionario de Datos –
Inventarios_Composicion**

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios que es compuesto
codigo_parte	varchar(30)	Código de la Ficha de Inventarios que compone al no_parte
bodega_parte	varchar(15)	Código de la Bodega donde pertenece el codigo_parte
recargo	double	Si no maneja unidades, almacena el valor del recargo
contrapartida	varchar(40)	Código contable de la Contrapartida
compuestoficticio	boolean	Si es compuesto ficticio
compuestoreal	boolean	Si es compuesto real
compras	varchar(1)	Si se utiliza en compras
ventas	varchar(1)	Si se utiliza en ventas
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

**Tabla 80. Diccionario de Datos –
Inventarios_Composicion_Unidades**

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
codigo_parte	varchar(30)	Código de la Ficha de Inventarios que compone al no_parte
bodega_parte	varchar(15)	Código de la bodega donde pertenece es codigo_parte
unidad	varchar(3)	Código de la Unidad de Costeo o de Control
cantidad	double	Cantidad de Unidades

Tabla 81. Diccionario de Datos – Líneas

NOMBRE	TIPO	DESCRIPCIÓN
codigolinea	varchar(10)	Código Identificador de la Línea
nombrelinea	varchar(100)	Descripción detallada de la Línea
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 82. Diccionario de Datos – Inventarios_Proveedores

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
cuenta	varchar(15)	Código de la Ficha del Directorio
precio_pro	double	Precio de la Ficha que ofrece el Proveedor
tiempo	integer	Tiempo (días) de entrega por parte del Proveedor

Tabla 83. Diccionario de Datos – Inventarios_Clientes

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
cuenta	varchar(15)	Código de la Ficha del Directorio
precio_venta	double	Precio que se le ofrece al Cliente

Tabla 84. Diccionario de Datos – Inventarios_Vendedores

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha de Inventarios
cuenta	varchar(15)	Código de la Ficha del Directorio
porc_comp	double	Porcentaje de Comisión asignado al Vendedor
valor_com	double	Valor de Comisión asignado al Vendedor

Tabla 85. Diccionario de Datos – Inventarios_Sustitutos

NOMBRE	TIPO	DESCRIPCIÓN
no_parte	varchar(30)	Código de la Ficha principal de Inventarios
codigo_sustituto	varchar(30)	Código de la Ficha de Inventarios que sustituye al no_parte
precio	double	Precio de la Ficha sustituto
existencias	integer	Cantidad de Fichas sustitutos que reemplazan al no_parte
ultimocosto	double	último costo de la Ficha sustituto

Tabla 86. Diccionario de Datos – Inventarios_Marcas

NOMBRE	TIPO	DESCRIPCIÓN
codigomarca	varchar(10)	Código Identificador de la Marca
nombremarca	varchar(100)	Descripción detallada de la Marca
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 87. Diccionario de Datos – Activos

NOMBRE	TIPO	DESCRIPCIÓN
idactivo	varchar(30)	Código Identificador del Activo
codigo	varchar(40)	Código contable que se enlaza con el Activo
nombre	varchar(60)	Nombre de la Ficha del Activo
cuenta	varchar(15)	Código de la Ficha del Directorio responsable del Activo
fechaingreso	date	Fecha de ingreso del Activo al Sistema
fechainiciodepre	date	Fecha de inicio de la depreciación del Activo
grupo	varchar(10)	Código del grupo al cual pertenece el Activo
descripcion	text	Descripción detallada del Activo
oficina	varchar(10)	Código de la Oficina de la ubicación el Activo
edificio	varchar(10)	Código del Edificio de la ubicación el Activo
piso	varchar(10)	Código del Piso de la ubicación el Activo
marca	varchar(10)	Código de la Marca del Activo
línea	varchar(10)	Código de la Línea del Activo
serial	varchar(10)	Número de Serie del Activo
fechaadquisicion	date	Fecha en que se adquirió (Compra/Donación) del Activo
factura	varchar(15)	Número de la Factura de la Compra o Entrada
comprobante	varchar(12)	Número del comprobante de egreso con el cual se canceló el Activo
referencia	varchar(20)	Número de referencia del Activo
proveedor	varchar(15)	Código de la Ficha de Proveedores
paisadquisicion	varchar(5)	País en donde se recibe el Activo
monedaaquisicion	varchar(10)	Código de la moneda del país en donde se adquirió el Activo
cantidad	integer	Cantidad de Activos comprados o recibidos por donación
factorcambio	double	Factor de cambio de la moneda de procedencia
iva	varchar(1)	Código de la Ficha del IVA
valoriva	double	Valor del IVA en la compra o donativo del Activo
fechavence	date	Fecha de vencimiento del Activo
asegurable	boolean	Indica si el Activo tiene una Póliza de Seguro
estado	varchar(23)	Estado del Activo: NUEVO, USUADO, MAL ESTADO, RECONSTRUIDO.
unidad	varchar(10)	Código de la Unidad de medida del Activo
vidautil	integer	Vida Útil del Activo
condición	varchar(20)	Condición del Activo: ACTIVO, DEPRECIADO, FUERA DE SERVICIO, INACTIVO, VENDIDO

Tabla 88. Diccionario de Datos – Activos_Grupos

NOMBRE	TIPO	DESCRIPCIÓN
idgrupo	varchar(10)	Código de Identificación del Grupo
nombre	varchar(60)	Nombre del Grupo
descripción	text	Descripción detallada del Grupo
vidautil	integer	Vida Util del Grupo
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 89. Diccionario de Datos – Activos_Oficinas

NOMBRE	TIPO	DESCRIPCIÓN
idoficina	varchar(10)	Código de Identificación de la Oficina
nombre	varchar(60)	Nombre de la Oficina
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 90. Diccionario de Datos – Activos_Edificios

NOMBRE	TIPO	DESCRIPCIÓN
idedificio	varchar(10)	Código de Identificación del Edificio
nombre	varchar(60)	Nombre del Edificio
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 91. Diccionario de Datos – Activos_Areas

NOMBRE	TIPO	DESCRIPCIÓN
idarea	varchar(10)	Código de Identificación del Área
nombre	varchar(60)	Nombre del Área
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 92. Diccionario de Datos – Activos_Marcas

NOMBRE	TIPO	DESCRIPCIÓN
idmarca	varchar(10)	Código de Identificación de las Marcas
nombre	varchar(60)	Nombre de la Marca
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 93. Diccionario de Datos – Activos_Lineas

NOMBRE	TIPO	DESCRIPCIÓN
idlinea	varchar(10)	Código de Identificación de la Líneas
nombre	varchar(60)	Nombre de la Línea
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 94. Diccionario de Datos – Activos_Unidades

NOMBRE	TIPO	DESCRIPCIÓN
idunidad	varchar(10)	Código de Identificación de las Unidades
nombre	varchar(60)	Nombre de la Unidad
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 95. Diccionario de Datos – Activos_Seguros

NOMBRE	TIPO	DESCRIPCIÓN
idpoliza	integer	Número de Identificación de la Póliza
fechatoma	date	Fecha de toma de la Póliza
fechavence	date	Fecha de vencimiento de la Póliza
cuenta	varchar(15)	Código de la Ficha del Directorio que representa la Empresa que vendió la Póliza
valorasegurado	double	Valor asegurado por la Póliza
valorprimaanual	double	Valor de la prima anual de la Póliza
valorprimames	double	Valor de la prima mensual de la Póliza
codigoagente	varchar(10)	Código del agente perteneciente a la Aseguradora
nombreagente	varchar(60)	Nombre del agente perteneciente a la Aseguradora
teléfono	integer	Teléfono del agente
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 96. Diccionario de Datos – Activos_Mantenimiento

NOMBRE	TIPO	DESCRIPCIÓN
idcontrato	integer	Número del contrato del mantenimiento
empresa	varchar(60)	Nombre de la empresa encargada del contrato
fechainicio	date	Fecha de inicio del mantenimiento
fechafin	date	Fecha de finalización del contrato de mantenimiento
periodicidad	varchar(10)	periodicidad en que se realiza el mantenimiento
fachaproximo	date	Fecha en que se realizará el próximo mantenimiento
usuario	varchar(20)	Usuario del Sistema responsable de ingresar/modificar el registro
fechausuario	varchar(19)	Fecha/ Hora en que el Usuario modificó el registro

Tabla 97. Diccionario de Datos – Países

NOMBRE	TIPO	DESCRIPCIÓN
idpais	varchar(5)	Código de Identificación del país
nombre	varchar(60)	Nombre del País

Tabla 98. Diccionario de Datos – Predios

NOMBRE	TIPO	DESCRIPCIÓN
idpredio	varchar(15)	Código de Identificación del Predio
tipopredio	varchar(3)	Define el tipo de predio RUR:Rural o URB:Urbano
direccion	varchar(30)	Dirección del Predio
estrato	integer	Válido para los Predios Urbanos, define la estratificación del Predio
nombrepropietario	varchar(30)	Nombres del Propietario del Predio
apellidopropietario	varchar(30)	Apellidos del Propietario del Predio
tipodocumento	varchar(2)	Define el tipo de documento: CC: Cedula de Ciudadanía, CE: Cedula de Extranjería,
numerodocumento	varchar(12)	Número de documento de Identificación del Propietario del Predio
hectáreas	integer	Área del Predio en Hectáreas
metros	integer	Área del Predio en metros cuadrados
metrosconstruidos	integer	Área construida del Predio en metros cuadrados
activo	boolean	Marcación que permite generar factura al predio en caso de que este en estado 'True'

Tabla 99. Diccionario de Datos – Predios_Imagenes

NOMBRE	TIPO	DESCRIPCIÓN
predio	varchar(15)	Código Identificador del Predio
idimagen	integer	Código consecutivo de la Imagen
imagen	bytea	Imagen de la Ficha del Predio
descripción	varchar(200)	Descripción detallada de la imagen del Predio

Tabla 100. Diccionario de Datos – Predios_Auxiliares

NOMBRE	TIPO	DESCRIPCIÓN
predio	varchar(15)	Código Identificador del Predio
codigocampo	integer	Código del campo auxiliar
valor	varchar(100)	Valor dependiendo del tipo de dato definido del campo auxiliar

Tabla 101. Diccionario de Datos – Estratos

NOMBRE	TIPO	DESCRIPCIÓN
idestrato	integer	Código Identificador del Estrato
nombreestrato	varchar(10)	Nombre del Estrato

Tabla 102. Diccionario de Datos – Establecimientos

NOMBRE	TIPO	DESCRIPCIÓN
idestablecimiento	varchar(10)	Código de Identificación del Establecimiento
nombreestablecimiento	varchar(100)	Nombre del establecimiento
numerodocumento	varchar(15)	Número de Identificación o Nit del Establecimiento
direccion	varchar(100)	Dirección del Establecimiento
telefono	varchar(8)	Teléfono del Establecimiento
avisosytableros	boolean	Define si el Establecimiento tiene avisos y tableros
patente	boolean	Define si el Establecimiento debe pagar por Patente Nocturna
suelo	boolean	Define si el Establecimiento debe pagar por Uso del Suelo
descuento	boolean	Define si el Establecimiento tiene descuentos especiales
cuotas	integer	Número de Cuotas en que pagará el Impuesto
activo	boolean	Marcación que permite generar factura al Establecimiento en caso de que este en estado 'True'

Tabla 103. Diccionario de Datos – Establecimientos_Actividades

NOMBRE	TIPO	DESCRIPCIÓN
establecimiento	varchar(10)	Código del Establecimiento
actividad	varchar(5)	Código de la Actividad

Tabla 104. Diccionario de Datos – Establecimientos_Auxiliares

NOMBRE	TIPO	DESCRIPCIÓN
establecimiento	varchar(15)	Código Identificador del Establecimiento
codigocampo	integer	Código del campo auxiliar
valor	varchar(100)	Valor dependiendo del tipo de dato definido del campo auxiliar

Tabla 105. Diccionario de Datos – Establecimientos_Imagenes

NOMBRE	TIPO	DESCRIPCIÓN
establecimiento	varchar(15)	Código Identificador del Establecimiento
idimagen	integer	Código consecutivo de la Imagen
imagen	bytea	Imagen de la Ficha del Establecimiento
descripcion	varchar(200)	Descripción detallada de la imagen del Establecimiento

Tabla 106. Diccionario de Datos – Actividades

NOMBRE	TIPO	DESCRIPCIÓN
idactividad	varchar(5)	Código Identificador de las Actividades Comerciales
grupoactividad	varchar(3)	Código del grupo al cual pertenece la Actividad
nombreactividad	varchar(60)	Nombre de la Actividad
porcentaje	double	Porcentaje que se aplica
valor	double	(Opcional) valor que se aplica al impuesto

Tabla 107. Diccionario de Datos – Grupos_Actividades

NOMBRE	TIPO	DESCRIPCIÓN
idgrupo	varchar(3)	Código Identificador del Grupo de Actividad
nombre	varchar(30)	Nombre del Grupo de Actividad