ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

SISTEMA GENÉRICO DE ADMINISTRACIÓN DE EDIFICIOS BAJO PLATAFORMA WINDOWS

PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

WILSON HERNAN ARAUJO ZEAS
infringercat@hotmail.com

DIRECTORA: ING. LUZ MARINA VINTIMILLA MSc
luzmavj@hotmail.com

Quito, Agosto 2009

DECLARACIÓN

Yo, Wilson Hernan Araujo Zeas declaro bajo juramento que el trabajo aquí descrito es de mi completa autoría; que no ha sido previamente presentada para ningun grado o calificación profesional; y que he consultado las referencias bibliograficas que se incluyen en este documento.

A traves de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politecnica Nacional, según lo establecido por la ley de propiedad intelectual, por su reglamento y por la normatividad institucional vigente.

_____

Wilson Araujo Zeas

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Wilson Hernan Araujo Zeas bajo mi supervisión.

_____

Ing. Luz Marina Vintimilla MSc.
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a mis padres, familiares, compañeros y demas luces de vida que de una u otra forma me apoyaron en el desarrollo de este trabajo y durante el trascurso de mi carrera.

Especialmente agradezco a mi hermana Anita y mi novia Sofia que me brindaron todo su fuerza y energia a fin de lograr cumplir mis objetivos.

Finalmente me gustaria agradecer a mi tutora de tesis y a todos mis profesores de la Escuela Politecnica Nacional, quienes me transmitieron el conocimiento necesario para elaborar el presente trabajo.

Wilson Araujo Z.

DEDICATORIA

Dedico esta tesis a Dios, a los arcángeles Miguelo, Rafaelo, Chamuel, Uriel y Jofiel que me ayudaron y guiaron incondicionalmente durante la elaboración de mi tesis.

Por ustedes, que han permitido en esta vida, cumplir el primero de mis objetivos terrenales y a traves de éste encontrar el soporte necesario para cumplir la mision fundamental de mi vida.

Wilson Araujo Z.

CONTENIDO

DECLARACION…… i
CERTIFICACION…… …. ii
AGRADECIMIENTO…..……..... iii
DEDICATORIA…..……iv

RESUMEN

CAPITULO I PLANIFICACIÓN DEL PROYECTO

1.1 Antecedentes….……..... 1
1.2 Planteamiento del Problema…….…… 1
1.2.1 Estudio de Causas……2
1.2.2 Estudio de Escenarios…… 3
1.3 Formulación…….. 3
1.3.1 Sistematización……4
1.4 Objetivos…… 4
1.4.1 Objetivos Específicos…….....4
1.5 Justificación……... 5
1.6 Alcance…….. 5
1.7 Presupuesto…….. 6
1.8 Arquitectura de Desarrollo…….. 7
1.8.1 Software de Front End…… 7
1.8.2 Software de Back End……... 7

CAPITULO II MARCO TEÓRICO

2.1 Paradigma……. 8
2.2 Metodología ……...…….. 10
2.2.1 Metodologia OMT…… 11
2.2.2 Fases de la metodología OMT…… .. 13
2.2.2.1 Análisis…….....13
2.2.2.2 Diseño del Sistema…….. 14
2.2.2.3 Diseño de Objetos……..…… 14
2.2.2.4 Implementación……..……14
2.3 Modelos……..……14

2.3.1 Modelo Estático……15
2.3.2 Modelo Dinámico…….…. 16
2.3.3 Modelo Funcional……17
2.3.4 Lenguaje Unificado del Modelado..……18
2.3.4.2 Diagramas UML……... 18
2.3.4.3 Descripción de Actores……..... .19
2.3.4.4 Diagrama de Casos de Uso……..……..20
2.3.4.4.1 Tipos de Relaciones en los Casos de Uso……...21
2.3.4.5 Diagrama de Clases……..……21
2.3.4.5.1 Relaciones entre Clases……..24
2.3.4.5.1.1 Asociación...…….. 24
2.3.4.5.1.2 Agregación……25
2.3.4.5.1.3 Generalización……..26
2.3.4.5.2 Diccionario de Clases…… 27
2.3.4.6 Diagramas de Interacción...……..…… 28
2.3.4.6.1 Diagrama de Secuencia…….. 28
2.3.4.6.2 Diagrama de Colaboración…… 29
2.3.4.7 Diagrama de Actividades……30
2.3.5 Construcción……...…….. 31
2.3.5.1 Rational Rose 2000….....31
2.3.5.2 Microsoft SQL Server 2000……... 32
2.3.5.3 Microsoft Visual Basic 6.0……33
2.3.5.4 Cliente Servidor……...…… 35
2.3.6 Prueba Funcional…… 38
2.3.7 Mantenimiento……..……...39

CAPITULO III CONCLUSIONES Y RECOMENDACIONES

3.1 Conclusiones ……40
3.2 Recomendaciones……40
3.3 Glosario…….. 41
3.4 Bibliografía……42

ANEXOS

A. Manual Técnico…….…… 44
B. Manual de Usuario…….……... 104
C. Manual de Instalación……..... 126

RESUMEN

El Sistema Genérico de Administración de Edificios bajo plataforma Windows, se somete a los principios de la Ingeniería del Software; esto es, la investigación, el análisis, el diseño, el desarrollo, la construcción y las pruebas.

El presente trabajo surge como una necesidad imperante, que permitirá a los administradores de edificios, llevar un control automático de las actividades de registro de los valores correspondientes al registro de ingresos y gastos.

El proyecto obedece a un análisis Orientado a Objetos, por ser una metodología actual, que proporciona muchas facilidades y se adapta más al mundo real.

El capítulo uno, trata acerca de la problemática actual observada en la administración de edificos en el pais, las soluciones para mejorar la situación, como también los objetivos planteados y aspectos metodológicos del Sistema.

El capítulo dos, provee de una descripción más profunda y explicativa acerca de los aspectos teóricos como: El Paradigma Orientado a Objetos, la Metodología Orientada a Objetos y la plataforma que se usa en el desarrollo del Sistema.

El capítulo tres, plantea las conclusiones que se han podido apreciar en el transcurso del desarrollo del proyecto y las recomendaciones que sugieren lineamientos para mejorar las actividades que normalmente se realizan en la administración de edificos. Además proporciona la bibliografía utilizada para la consecución de este trabajo.

Finalmente, se anexa manuales tecnico, de usuario y de instalación a fin de proporcionar la debida información a los usuarios sobre el manejo del Sistema Generico de Administración de Edificios bajo plataforma Windows.

CAPITULO 1

PLANIFICACIÓN DEL PROYECTO

1.1. ANTECEDENTES

En la actualidad los dueños de edificios en el país controlan los ingresos y egresos de sus arrendatarios y copropietarios mediante registros a cargo de los guardias de seguridad o incluso del conserje del edificio. En edificios con un control mucho más riguroso, dicho control se lo hace mediante el trabajo conjunto de administradores, contadores, secretaria, conserje y guardias.

La responsabilidad de administrar un edificio implica recolectar de manera correcta los montos provenientes del cobro de alícuotas, cuotas extraordinarias y deudas anteriores que los propietarios mantengan con la administración. A más de controlar los ingresos de dinero, el administrador debe disponer de estos valores para el pago oportuno de los egresos generados por concepto de mantenimiento del edificio.

El mundo globalizado que nos rodea, ha impulsado a profesionales de todas las áreas a buscar apoyo informático para la ejecución de sus tareas diarias y, administradores de edificios no son la excepción ya que para mejorar su gestión, han optado por adquirir sistemas informáticos que mejoren los resultados en el cumplimiento de sus objetivos, y consecuentemente presten un mejor servicio a propietarios, copropietarios y arrendatarios.

1.2. PLANTEAMIENTO DEL PROBLEMA

La administración de edificios requiere de un verdadero compromiso, ya que actualmente el número de propietarios, copropietarios y arrendatarios registrados dentro de un condominio es bastante considerable. Esto provoca que el control manual o en fichas que se realiza actualmente en la mayoria de propiedades, se dificulte cada vez más, debido a la cantidad de información.

Al no tener un control exacto de los valores recaudados, el administrador del edificio debe esperar hasta fin de mes para obtener un balance económico de su gestión, perdiendo tiempo valioso para la toma de decisiones e invirtiendo horas de trabajo tedioso. Por otro lado, los copropietarios y arrendatarios no saben cuánto dinero han entregado a la administración para que sea invertido en mantenimiento o servicios.

Con base a estos antecedentes, se hace imperiosa la necesidad de un sistema que cubra estas expectativas, y así evitar la pérdida de información y la desconfianza de los arrendatarios y copropietarios, en cuanto a la gestión del administrador del edificio.

Un sistema automatizado beneficiará tanto a copropietarios como a los administradores, quienes podrán obtener reportes contables de su gestión en el momento que se requiera, evitará posibles errores manuales en la recopilación de información, así como errores de cálculo involuntarios en los valores manejados.

1.2.1. ESTUDIO DE CAUSAS

- Se registra el cobro de alícuotas de los departamentos manualmente.
- No se tiene una base de datos que facilite la recolección y tratamiento de la información.
- No existe un control de los copropietarios que adeudan valores a la administración.
- No se tiene un debido control de la fecha en que el administrador cancela los gastos del edificio.
- La necesidad de acceder cada vez a una mayor cantidad de información en un menor tiempo.
- Análisis erróneo de los datos cuando se solicita información de recaudaciones.
- Los datos de los departamentos, propietarios y arrendatarios se encuentran desorganizados.
- No existe un control en la seguridad de la información.
- No se lleva un control de las deudas anteriores de los propietarios.

1.2.2. ESTUDIO ESCENARIOS

Si los administradores de edificios continúan desempeñando su gestión de manera manual, en lo posterior se presentarían los siguientes escenarios:

- La información de propietarios y arrendatarios no estará disponible cuando la administración lo requiera.
- No se podrá atender en forma eficiente el cobro de alícuotas.
- No se controlará el cumplimiento del pago de deudas anteriores como requisito para el pago de las alícuotas mensuales.
- No se podrá tener un control adecuado de los egresos cancelados por la administración por conceptos de mantenimiento.
- No se podría entregar de manera oportuna reportes y balances para sustentar su trabajo.
- Probables errores en los cobros, y como consecuencia faltantes de caja.
- Pérdida de competitividad frente a otras administraciones.

Se concluye por lo tanto que administradores de edificios requieren automatizar la información referente a los procesos de recaudación de ingresos y cancelacion de egresos. Además llevar un correcto control de los comprobantes generados hacia propietarios, copropietarios o arrendatarios, transparentando de esta manera el manejo de los valores, logrando así que su gestión sea más eficiente.

1.3 FORMULACIÓN

¿Cómo utilizar la tecnología informática para dar soluciones a la administración de edificios?

1.3.1 SISTEMATIZACIÓN

- ¿Qué hacer para administrar los locales?
- ¿Qué hacer para administrar a propietarios?
- ¿Qué hacer para gestionar los ingresos de dinero?
- ¿Cómo regular egresos de dinero?
- ¿Qué información facilitar a los administradores para la toma de decisiones?
- ¿Cómo mantener un control histórico del pago de alícuotas en los edificios?
- ¿Cómo evitar la pérdida de información importante para los administradores?

1.4 OBJETIVOS

Desarrollar e implementar el Sistema Genérico para la Administración de Edificios bajo plataforma Windows, con aplicación visual segura y de fácil manejo, para solucionar problemas frecuentes en la gestión de edificios.

1.4.1 OBJETIVOS ESPECÍFICOS

- Diseñar, construir e implantar los siguientes módulos:
  - Administración de edificios, locales y arrendatarios.
  - Registro de ingresos y egresos.
  - Reportes, consultas y balances de la gestión del edificio.
  - Generación de cierre de año.
- Implementar una Base de Datos que se acople al sistema.

1.5 JUSTIFICACIÓN

Con la premisa que en la actualidad los administradores de edificios manejan gran cantidad de información con respecto a propietarios, copropietarios y arrendatarios, se puede afirmar que en el transcurso del tiempo el manejo de esta información se tornará mas complicada si se lo continua manejando de una manera manual; lo cual representará un impacto negativo sobre la gestión administrativa del edificio, por lo tanto el presente proyecto tiene una repercusión práctica sobre las actividades que se deben realizar, aportando de manera valiosa al manejo y conservación de esta importante información.

1.6 ALCANCE

Con la implantación de este sistema se solucionará la problemática contable identificada en la administración de edificios en el pais, específicamente se limita a la recaudación de alícuotas, deudas anteriores, cuotas extraordinarias y registro de gastos mensuales con sus respectivos reportes.

El sistema permitirá configurar los parámetros iniciales según las políticas de manejo y cobro establecidas por los administradores para cada edificio, optimizando los procesos relacionados a la emisión de comprobantes y definición de accesos para los usuarios.

Finalmente, con éste sistema se obtendrá balances que ayudarán a dueños, administradores, propietarios, copropietarios y arrendatarios a tomar decisiones correctas para el mejoramiento en la utilización de recursos comunales, brindando una mejor convivencia.

1.7 PRESUPUESTO

Hardware

Cantidad	Detalle	Costo Total
1	Computador core 2 duo 2.64gb en Ram	730 USD.
1	Impresora Matricial LX300+	220 USD.
TOTAL HARDWARE		**950 USD.**

Software

Cantidad	Detalle	Costo Total
1	Microsoft Visual Basic 6.0	400 USD.
1	Microsoft SQL Sever 2000	350 USD.
1	Rational Rose 2000	1100 USD.
1	Crystal Reports 8.5	550 USD.
TOTAL SOFTWARE		**2400 USD.**

Recursos Humanos

Cantidad	Detalle	Costo Total
1	Analista	5 USD/h. x 240 h. 1200 USD.
1	Programador	3 USD/h. x 300 h. 900 USD.
TOTAL RECURSOS HUMANOS		**2100 USD.**

Recursos de Oficina

Cantidad	Detalle	Costo Total
1	Hojas de Papel Bond	50 USD.
1	CDs	20 USD.
1	Cartuchos de Impresión	80 USD.
TOTAL		**150 USD.**

COSTO TOTAL	**5600 USD.**

1.8 ARQUITECTURA DE DESARROLLO

El sistema de administración de edificios fue desarrollado en una arquitectura cliente / servidor, es decir se definió la aplicación independiente del manejo de datos, con lo cual se optimiza el mantenimiento general del sistema.

En este sentido, se puede entender por cliente a la aplicación, quien solicita servicios a la base de datos, y se entiende por servidor a la base de datos que en estado activo, espera indefinidamente una solicitud de servicios por parte del cliente.

1.8.1 SOFTWARE DE FRONT END

Para la construcción de la interfase gráfica del cliente se utilizó Microsoft Visual Basic 6.0, ya que es una herramienta que permite crear sistemas de alto rendimiento, orientados a solucionar problemas cotidianos, mediante aplicaciones sencillas.

Proporciona además un completo conjunto de herramientas que faciltan la integración con cualquier base de datos, específicamente para el sistema de administracion de edificios, la interacción con SQL Server 2000 permitió generar fácilmente operaciones de creación, actualización y eliminación.

1.8.2 SOFTWARE DE BACK END

Microsoft SQL Server 2000, fue seleccionado como el motor de base de datos, debido al manejo intuitivo que requiere la herramienta y a la compatibilidad entre aplicaciones Microsoft.

Otra de las caracteristicas importantes de SQL 2000 es la consistencia y confiabilidad que ofrece en la gestión de datos, además de la capacidad para manejar grandes cantidades de registros y consultas simultáneas sin presentar inconvenientes.

CAPITULO 2

MARCO TEÓRICO

2.1 PARADIGMA[2]

Un paradigma es el esquema de control de avance de un proyecto de desarrollo de sistemas, hoy en día el Paradigma Orientado a Objetos, encierra una visión completa de la Ingeniería del Software; por lo que el mejor paradigma para la Ingeniería del Software Orientado a Objetos, es el modelo evolutivo de proceso.

Según Pressman, la Ingeniería de Software: Es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales. La Ingeniería del Software es una tecnología multicapa cuyo propósito es guiar hacia la construcción y mantenimiento de un producto de software que cumpla con los requisitos, los estándares de desarrollo y con factores de calidad (funcionalidad, confiabilidad, facilidad de mantenimiento, portabilidad, eficiencia).

Por tanto, el paradigma Orientado a Objetos usa análisis, diseño y desarrollo para producir software eficiente y económico, presentando una serie de ventajas como: la reutilización de componentes de software, que permite un desarrollo más rápido; el encapsulamiento, que esconde la implementación del objeto concediendo la privacidad de la información.

El paradigma de Ingeniería de Software maneja varios modelos evolutivos, entre ellos, el Modelo en Espiral, que según Pressman: "Es un modelo de proceso de software evolutivo que conjuga la naturaleza interativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial"[3].

---

[2] PRESSMAN, R; *Ingeniería de Software, un enfoque práctico*; Quinta Edición; 2000, pag 14

[3] PRESSMAN, R; *Ingeniería de Software, un enfoque práctico*; Quinta Edición; 2000, pag. 24

En el modelo espiral el software se desarrolla en una serie de versiones incrementales, por tanto entrega un producto operacional con cada incremento, siendo los primeros incrementos versiones incompletas del producto final, sin embargo permiten al usuario apreciar la funcionalidad y la plataforma para la evaluación.

El modelo espiral define *regiones de tareas* que no son más que los pasos que se requieren para una iteracción y éstos son:

- *Comunicación con el cliente*: Son las tareas que permiten la comunicación entre el desarrollador y el cliente. Se identifica el problema y las clases básicas.
- *Planificación:* Son las tareas que permiten definir recursos, tiempo, responsables y otra información inherente al proyecto.
- *Análisis de Riesgos:* Son las tareas requeridas que permiten evaluar riesgos técnicos y de gestión.
- *Ingeniería:* Son las tareas requeridas para construir representaciones de la aplicación.
- *Construcción y acción:* Son las tareas que permiten desarrollar, probar, implementar y proporcionar soporte al usuario (documentación y práctica).
- *Evaluación del cliente:* Permiten obtener la reacción del cliente ante la construcción e instalación del producto final.

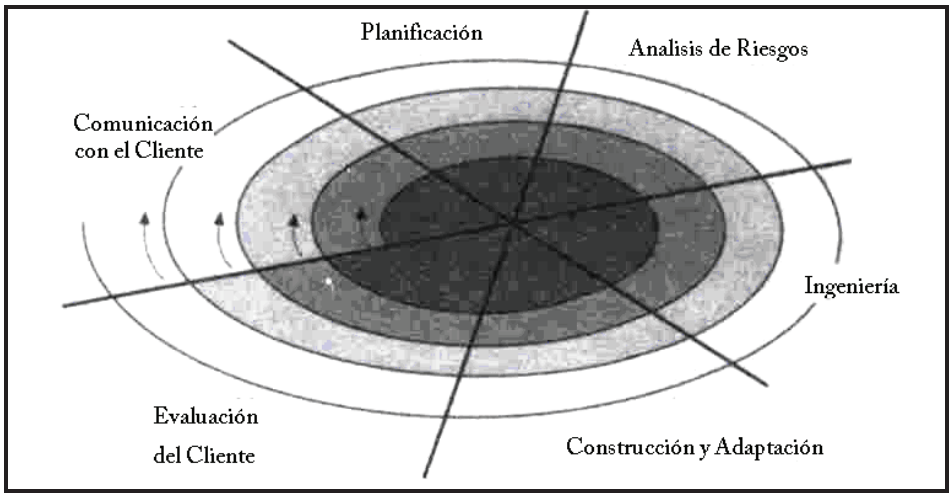


FIGURA 2.1 DIAGRAMA DEL MODELO ESPIRAL INCREMENTAL
FUENTE: PRESSMAN "*Ingeniería de Software*"

2.2 METODOLOGÍA

Para el desarrollo del presente sistema de Administración de Edificios utilizaré el paradigma de desarrollo de software espiral incremental, apoyado en la Metodología OMT (Object Modeling Technique), resumida en la siguiente base de desarrollo:

PARADIGMA ESPIRAL INCREMENTAL	METODOLOGÍA OMT	DIAGRAMAS UML	
Análisis	Modelo de Objetos (MO)	Descripción de Actores Diagrama de Casos de Uso Descripción de Casos de Uso Diagrama de Clases Diccionario de Clases	
Diseño	Modelo Dinámico (MD)	Diagrama de Interacción	Diagrama de Secuencia
			Diagrama de Colaboración
	Modelo Funcional (MF)	Diagrama de Actividades	
Construcción	Herramientas de Software	Rational Rose 2000 Microsoft SQL Server 2000 Microsoft Visual Basic 6.0 Crystal Reports 8.5 Plataforma Cliente/Servidor	
Pruebas	Prueba Funcional	Descripción de la Prueba Procedimiento de la Prueba	
Mantenimiento	La documentación del desarrollo y seguimiento de los modelos facilita el posterior mantenimiento.		

TABLA 2.1 METODOLOGÍA UTILIZADA
ELABORADO POR: EL AUTOR

2.2.1 METODOLOGÍA OMT

La metodología OMT (*Object Modeling Technique*), fue creada por James Rumbaugh[4] y Michael Blaha en 1991. La metodología consiste en construir un modelo de un dominio de aplicación añadiéndosele detalles de implementación durante el diseño de un sistema.

La característica fundamental de OMT es tener en cuenta todos los principios del paradigma de orientación a objetos y refuerza particularmente la necesidad de modelos consistentes de análisis y diseño del negocio como paso previo y fundamental al desarrollo de la programación y la definición de las bases de datos. OMT es una de las metodologías de análisis y diseño orientadas a objetos, más popular y eficiente utilizada en el mundo profesional. La metodología OMT ha servido de inspiración para nuevos métodos de desarrollo de proyectos de objetos.

Esta Técnica de Modelado de Objetos, se extiende desde el análisis hasta la implementación, pasando por el diseño, donde el modelo de análisis abstrae los aspectos esenciales del dominio de la aplicación, que constituyen el marco de trabajo del modelo de diseño; y el modelo de diseño se implementa en algún lenguaje de programación, base de datos o hardware.

La palabra *desarrollo* hace alusión a la parte inicial del ciclo vital del software: análisis, diseño e implementación. El *desarrollo orientado a objetos* se basa en abstracciones del mundo real, la esencia del desarrollo orientado a objetos es la identificación y organización de conceptos del dominio de aplicación, mas no de su representación final en un lenguaje de programación. En fin, el *desarrollo orientado a objetos* es fundamentalmente una forma de pensar y no una técnica de programación.

---

[4] RUMBAUGH, J; *Modelado y Diseño Orientado a Objetos. Metodología OMT*; Santa Fé, Bogota; Prentice Hall; 1999;pág 19-22

El enfoque *orientado a objetos*, organiza al software como una colección de objetos discretos que contienen tanto estructuras de datos como un comportamiento; es decir son entidades coherentes con identidad, estado y conducta. Existen temas que subyacen a la tecnología orientada a objetos, aunque no son exclusivos de los sistemas orientados a objetos, están apoyados por los mismos. Estos temas son: Abstracción, Encapsulación, Modularidad y Jerarquización.

- La *Abstracción* denota las características esenciales que distinguen a un objeto de otros tipos de objetos, definiendo precisas fronteras conceptuales, relativas al observador. Surge del reconocimiento de similaridades entre ciertos objetos, situaciones o procesos en el mundo real. Se concentra en éstas similaridades e ignora las diferencias.

- La *Encapsulación* es un concepto complementario al de abstracción, es la "*caja negra*" que permite ocultar la complejidad y protege a los componentes de otros, los que sólo conocerán lo que el componente les permita en su interfaz exterior. Se conoce también como ocultamiento o privacidad de la información.

- La *Modularidad* es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y vagamente conexos. En el diseño orientado a objetos, la modularización debe ceñirse a la estructura lógica elegida en el proceso de diseño. Al dividir un programa en componentes individualizados reduce de alguna manera su complejidad.

- La *Jerarquización* es una clasificación u ordenación de las abstracciones. Por jerarquía se denota el orden de relación que se produce entre abstracciones diferentes.

La tecnología orientada a objetos plantea una forma práctica y productiva de desarrollar software para diferentes aplicaciones independientemente del lenguaje final de implementación, acoplándose a todas las necesidades actuales y futuras de la ingeniería de software.

2.2.2 FASES DE LA METODOLOGÍA OMT

La Metodología OMT pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción , en el cual el análisis está enfocado en el mundo real para un nivel de diseño, también pone detalles particulares para modelado de recursos de la computadora.

Es una de las metodologías de análisis y diseño orientadas a objetos, más maduras y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), en consecuencia, sobrevivir con enorme vitalidad. Las fases que conforman a la metodología OMT son: analisis, diseño del sistema, diseño de objetos e Implementación; las cuales se decriben a continuación.

2.2.2.1 Análisis

El análisis empieza con la descripción del problema, donde el analista construye un modelo de dominio del problema, mostrando sus propiedades más importantes. El modelo de análisis es una abstracción resumida y precisa de lo *que* debe hacer el sistema deseado y no de la *forma* en que se hará.

Los objetos del modelo deben ser conceptos del dominio de aplicación y no conceptos de la implementación de la computadora, tales como estructuras de datos. Un buen modelo debe ser de fácil comprensión y deberá poder ser criticado por expertos de la aplicación que no sean programadores.

Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos, tales como estructuras de datos. Un buen modelo debe poder ser entendido y criticado por expertos en el dominio del problema que no tengan conocimientos informáticos.

2.2.2.2 Diseño del Sistema

En esta fase, el diseñador del sistema toma decisiones de alto nivel sobre la arquitectura global del proyecto, como también debe decidir las características de rendimiento que se deben optimar. El sistema se organiza en subsistemas basándose tanto en la estructura de análisis como en la arquitectura propuesta. Esta fase se caracteriza por el establecimiento de las decisiones más importantes.

2.2.2.3 Diseño de Objetos

El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis y añade detalles de implementación al modelo de acuerdo con la estrategia establecida en el diseño del sistema. El diseño de objetos enfoca las estructuras de datos y algoritmos del dominio de la computadora. OMT describe la forma en que el diseño es indistinto de los lenguajes de programación (orientados y no orientados a objetos).

2.2.2.4 Implementación

Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen por último a un lenguaje de programación específico.
La programación debe ser considerada como una parte pequeña y fundamentalmente mecánica del ciclo de desarrollo de sistemas. En esta fase de implementación es importante seguir los principios de la ingeniería de software, de tal forma que la correspondencia con el diseño sea directa y el sistema implementado continúe siguiendo los lineamientos que le permiten ser flexible y extensible.

2.3 MODELOS

En la Metodología OMT el uso de modelo es indispensable para lograr una correcta abstracción de los objetos dentro de un sistema, especialmente a nivel del analisis, por lo que se debe considerar cada estructura definida por los modelos: estatico, dinamico y funcional.

2.3.1 MODELO ESTÁTICO [5]

Llamado también *Modelo de Objetos*, este modelo describe la estructura de los objetos de un sistema, así como las relaciones con otros objetos, atributos y operaciones. Según Rumbaugh, el modelo de objetos proporciona el entorno esencial en el cual se puede situar el modelo dinámico y el modelo funcional. Los objetos son las unidades en las que se divide el mundo real. Este modelo debe ser refinado por medio de la iteración.

El modelo estático se representa gráficamente mediante diagramas, los mismos que contienen clases de objetos. Unos de los diagramas de este modelo son el de *clases* y el de *objetos*, aunque los diagramas de clases prevalecen más que los de objetos. Normalmente se puede construir un diagrama de clases y ocasionalmente uno de objetos para ilustrar las estructuras de datos más complejas.

El objetivo principal al construir un modelo de objetos es capturar aquellos conceptos más importantes del mundo real para una aplicación. Al modelar un problema de ingeniería, el modelo de objetos deberá contener términos familiares para la problemática que se desea resolver; así al modelar un problema comercial, los términos empleados deben ser del mundo de los negocios.

Este modelo permite representar gráficamente los objetos, las clases y sus relaciones mediante dos tipos de diagramas: los diagramas de clases y los diagramas de casos concretos (instancias). Los diagramas de clases describen las clases que componen el sistema y que permitirán la creación de casos concretos, los diagramas de casos concretos describen la manera en que los objetos del sistema se relacionan y los casos concretos que existen en el sistema de cada clase. En los diagramas que componen este modelo se pueden representar los siguientes elementos del sistema: objetos y clases, atributos, operaciones, y relaciones o asociaciones.

---

[5] RUMBAUGH, J; *Modelo y Diseño Orientado a Objetos. Metodología OMT*; Santa Fe, Bogota; Prentice Hall;1997; Pag 39

2.3.2 MODELO DINÁMICO [6]

Los aspectos del sistema que están relacionados con el tiempo y con los cambios constituyen *el modelo dinámico*, por contraste con el modelo estático o de objetos. La estructura del modelo dinámico se caracteriza por el envío de mensajes.

El *control* es aquella parte del sistema que describe las secuencias de operaciones que se producen en respuesta a estímulos externos, sin tener en cuenta lo que hagan las operaciones, aquello a lo que afectan, ni la forma en que sean implementadas.

Los conceptos más importantes del modelado dinámico son los *sucesos* y los *estados* que representan los valores de los objetos. Los sucesos marcan los cambios. Por tanto, el modelo dinámico describe aquellos aspectos del sistema que tratan de la temporización y secuencia de operaciones.

El modelo dinámico se representa gráficamente mediante diagramas de *estado* y de *sucesos*, y de éste último se puede generar el diagrama de Colaboración. Cada uno de estos diagramas muestra el estado y las secuencias de sucesos que son admisibles en un sistema. Las acciones de los diagramas de estado se corresponden con funciones procedentes del modelo funcional; los sucesos de un diagrama de estado pasan a ser operaciones que se aplican a objetos dentro del modelo de Objetos.

Los pasos que se siguen en el modelo dinámico son:

- Preparar escenarios de una interacción típica.
- Identificar eventos entre objetos preparando un trazado de eventos para cada escenario.
- Construir los diagramas de estados.
- Revisar los eventos entre los objetos para verificar su consistencia.

---

[6] RUMBAUGH, J; *Modelo y Diseño Orientado a Objetos. Metodología OMT*; Santa Fe, Bogota; Prentice Hall;1997; Pag 40

2.3.3 MODELO FUNCIONAL

El Modelo Funcional describe las transformaciones de datos del sistema. Modela las funciones, los valores de entrada y salida, e impone las restricciones pertinentes del sistema.

Rumbaugh dice que: "El modelo funcional captura lo que hace el sistema, independientemente de cuando se haga o de la forma en que se haga"[7].
Por tanto, describe las transformaciones (de función), de valores de datos que ocurren dentro del sistema.

En el modelo funcional, se suelen utilizar los tan discutidos Diagramas de Flujo de Datos Orientados a Objetos. Un diagrama de flujo de datos es un grafo cuyos nodos son procesos y cuyos arcos son flujos de datos, se muestra las dependencias entre los valores y el cálculo de valores de salida a partir de los de entrada y de funciones, sin considerar cuándo se ejecutan las funciones, ni siquiera si llegan a ejecutarse.

Las funciones se invocan como acciones en el modelo dinámico y se muestran como operaciones que afectan a objetos en el modelo de objetos.
Los conceptos tradicionales de computación tales como árboles de expresiones (sintácticos) y conceptos menos tradicionales como las hojas de cálculo son ejemplos de modelos funcionales.

El modelo funcional, muestra cómo se calculan los valores sin importar la secuencia, las decisiones o la estructura de los objetos. Los pasos para construir un modelo funcional son:

- Identificar valores de entrada y salida.
- Usar diagramas de flujo de datos para mostrar dependencias funcionales.
- Describir las funciones.
- Identificar restricciones

---

[7] RUMBAUGH, J; *Modelo y Diseño Orientado a Objetos. Metodología OMT*; Santa Fe, Bogota; Prentice Hall;1997; Pag 41

2.3.4 LENGUAJE UNIFICADO DEL MODELADO [8]

El UML (*Unified Modeling Languaje o Lenguaje Unificado de Modelado)* es un lenguaje gráfico para la especialización, visualización, construcción y documentación de piezas de información usadas o producidas durante el proceso de desarrollo de software. El UML provee un marco arquitectónico de diagramas para trabajar sobre análisis y diseño orientado a objetos, así como también el modelamiento de negocios y otros sistemas que no son software.

El UML es un lenguaje simbólico que sirve para expresar modelos orientados a objetos y no una metodología para desarrollarlos. Este lenguaje es el resultado de la convergencia de las mejores prácticas en la industria de la tecnología de software orientando a objetos, fue creado por tres de los más importantes metodólogos: Booch, Rumbaugh y Jacobson, quienes se agruparon en Rational Software para desarrollo.

El UML surge como la unión de la simbología utilizada por estos métodos, y aún más, puesto que incluye formas adicionales para manejar problemas que el modelado con estos métodos no abarca completamente. La mayoría de compañías están incorporando el UML como estándar en sus procesos y productos, que cubren disciplinas tales como modelamiento del negocio, requisitos de gerencia, análisis y diseño, programación y pruebas.

2.3.4.1 Diagramas UML

El UML puede describir cualquier tipo de sistemas en términos de diagramas orientados a objetos. Entre los diferentes tipos tenemos sistemas de información, sistemas de tiempo real, sistemas embebidos, sistemas distribuidos, software de sistemas, sistemas de negocios.

---

[8] AVILA, C; *Modelando con UML. Principios y Aplicaciones*; Perú; Primera edición; 2001.

Los diagramas se utilizan para dar diferentes perspectivas del problema según lo que nos interese representar de un determinado momento. Los diagramas que UML define son:

1) Diagramas de Casos de Uso
2) Diagramas de Clases
3) Diagramas de Secuencia
4) Diagramas de Colaboración
5) Diagramas de Actividades

2.3.4.2 Descripción de Actores

Los actores se representan mediante "*hombres de palo*", un actor representa un conjunto de roles que los usuarios desempeñan cuando interaccionan con los casos de uso. Un actor también es un clasificador y por lo tanto puede representarse como una clase.

Los actores pueden ser:

Principales: Personas que usan el sistema.
Secundarios: Personas que mantienen o administran el sistema.
Material Externo: Dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben se utilizados.
Otros Sistemas: Sistemas con los que el sistema interactúa.

Para encontrar los actores, se debe seguir los siguientes pasos:

- Identificar eficientemente los usuarios que intervienen en el sistema.
- Identificar los roles que realizan estos usuarios desde el punto de vista del sistema.
- Identificar otros sistemas con los cuales exista comunicación.

2.3.4.3 Diagrama de Casos de Uso

Un Diagrama de Casos de Uso representa lo *que* hace el sistema y *cómo* se relaciona con su entorno. Un diagrama de Casos de Uso representa los distintos requerimientos que hacen los usuarios al sistema, especificando las características de funcionalidad y comportamiento durante su interacción con los usuarios y otros sistemas.

A dichas funcionalidades se le conocen como Casos de Uso propiamente dichos, mientras que a los que provocan su ejecución se les conoce como Actores. Los casos de Usos y Actores interactúan produciendo Relaciones. Un *Caso de Uso* (Use Case) es una secuencia de acciones realizadas por el sistema que *producen un resultado observable y valioso para alguien* en particular.

Cabe mencionar que los diagramas de casos de uso se basan en la idea de que la mejor manera de comprender un sistema es mediante su descomposición funcional, independientemente de los objetos participantes. En este sentido los diagramas de casos de uso se acercan más al análisis estructurado y poco tienen que ver con entender cómo un conjunto de objetos interactúan.

Se puede deducir que los Casos de Uso son independientes del paradigma de construcción de software y por lo tanto el lenguaje de programación, pudiendo utilizarse como punto de partida para diseñar un sistema con un enfoque estructurado o un sistema con un enfoque orientado a los objetos.

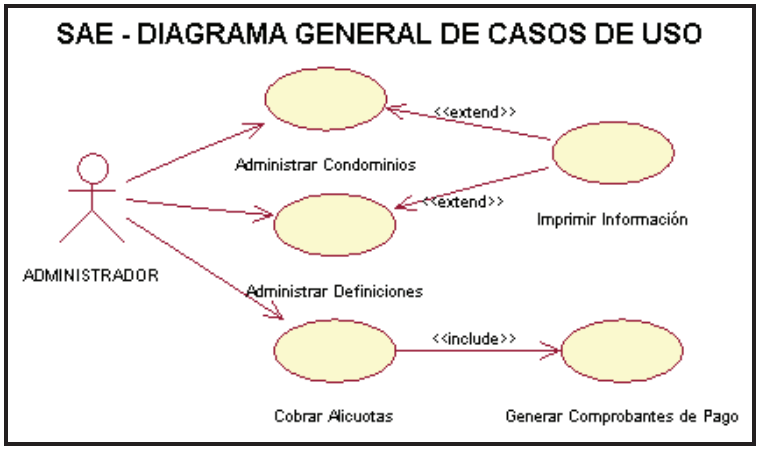


FIGURA 2.2 REPRESENTACIÓN GRÁFICA DE LOS CASOS DE USO
ELABORADO POR: EL AUTOR

2.3.4.4.1 Tipos de Relaciones en los Casos de Uso

Los tipos de relaciones que el UML define para los casos de uso son:

- **Asociación**: Esta relación representa la participación de un *Actor* con el *Caso de Uso*. Es la más general de las relaciones y la relación semántica más débil. Siempre parte de los actores y viaja en una sola dirección.
- **Inclusión** <<*include*>>: Se emplea cuando un caso de uso base incorpora *explícitamente* el comportamiento de otro caso de uso. El caso base *siempre utiliza* al caso de uso incluido. El caso de uso incluido *no puede ejecutarse* sin el caso de uso que lo incluye.
- **Extensión** <<*extend*>>: En esta relación el caso de uso base *extiende* el comportamiento del caso de uso destino. La relación extend une una *conducta normal* en un caso y la *conducta inusual* en el otro caso.
- **Herencia**: El Caso de Uso origen hereda la especificación del Caso de Uso destino.

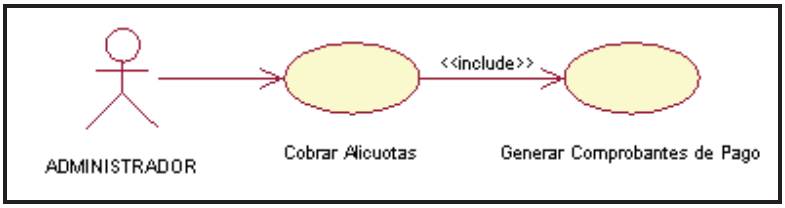


FIGURA 2.3 REPRESENTACIÓN DE LA RELACIÓN INCLUDE
ELABORADO POR: EL AUTOR

FIGURA 2.4 REPRESENTACIÓN DE LA RELACIÓN EXTEND
ELABORADO POR: EL AUTOR

2.3.4.5 Diagrama de Clases

Muestran un conjunto de clases (*grupos de objetos que tienen las mismas características y comportamiento*), así como sus relaciones. Estos diagramas son los más comunes en el modelado de los sistemas orientados a objetos y cubren la vista *estática* de un sistema.

Un diagrama de estructura estática muestra el conjunto de clases y objetos importantes, que conforman un sistema, junto con las relaciones existentes entre los mismos, pero no como actúan unos con otros, ni que mensajes se envían. Un diagrama de clases está compuesto por: Clases y Relaciones.

Clase, es la unidad básica que encapsula toda la información de un Objeto (una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.). En UML, una clase es representada por un rectángulo que posee tres divisiones:

FIGURA 2.5 REPRESENTACIÓN DE UNA CLASE
ELABORADO POR: EL AUTOR

Para la representación, se requiere la siguiente información en cada seccion:

- Parte Superior: Contiene el nombre de la Clase
- Parte Intermedia: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).
- Parte Inferior: Contiene los métodos u operaciones, los cuales son la forma cómo interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

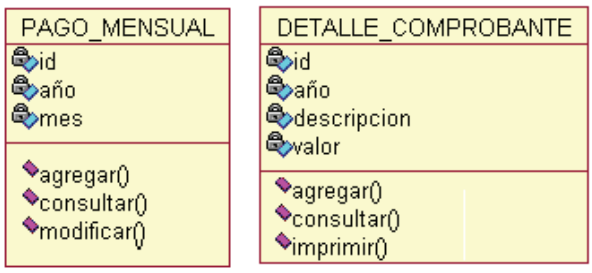


FIGURA 2.6 EJEMPLO DEL MODELADO DE CLASES
ELABORADO POR: EL AUTOR

Los atributos o características de una Clase pueden ser de tres tipos, los cuales definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

Public (+,)

Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.

Private (-,)

Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden accesar).

Protected (#,)

Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven.

Los Métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos también pueden tener las características de:

Public (+,)

Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.

Private (-,)

Indica que el método sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden accesar).

Protected (#,).

Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven.

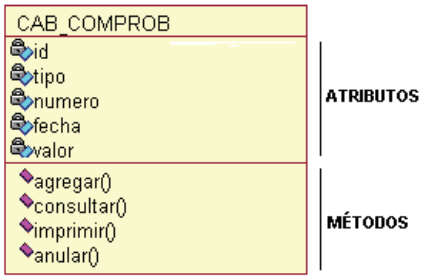


FIGURA 2.7 DESCRIPCIÓN DE ATRIBUTOS Y MÉTODOS DE LA CLASE CAB_COMPROB
ELABORADO POR: EL AUTOR

2.3.4.5.1 Relaciones entre Clases

Ya definido el concepto de Clase, es necesario explicar cómo se pueden interrelacionar dos o más clases (cada una con características y objetivos diferentes).

Sus formas de relación son:

- Asociación
- Agregación
- Generalización/Especialización

2.3.4.5.1.1 Asociación

La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y estas pueden ser:

- Uno y solo uno
- uno o muchos: 1..* (1..n)
- 0 o muchos: 0..* (0..n)
- 0 a uno: 0..1
- Desde M hasta N: M..N (enteros naturales)

En la figura siguiente se muesta una relación entre las clases Condominio y Gasto. Narrativamente la relación entre ambas clases se expresa de la siguiente manera:

” Un Condominio puede tener asociado muchos Gastos, en cambio un Gasto solo puede tener asociado un Condominio”.

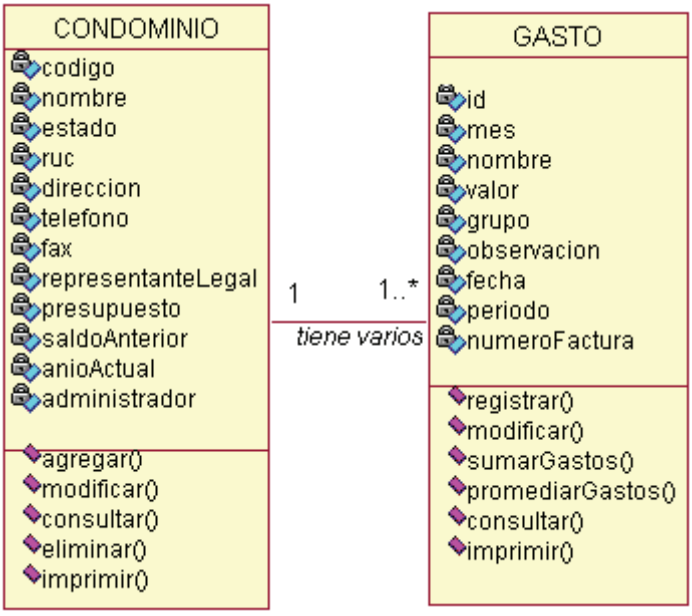


FIGURA 2.8 REPRESENTACION DE LA ASOCIACIÓN ENTRE CLASES
ELABORADO POR: EL AUTOR

2.3.4.5.1.2 Agregación

La agregación representa una relación *parte – de* entre objetos. A esta relación la caracteriza la precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

Por Valor: Es un tipo de relación estática, donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada “Composición” (el Objeto base se contruye a partir del objeto incluido, es decir, es "parte/todo"). La composición se representa con un rombo relleno.

Por Referencia: Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada Agregación (el objeto base utiliza al incluido para su funcionamiento). La agregación se representa por un rombo transparente.

En la figura 2.9 se ilustra la representación gráfica de una agregación, los rombos se colocan en el objeto que posee las referencias, cuando se destruye el objeto Cabecera_Comprobante, también se destruyen los objetos Detalle_Cuota_Extraordinaria y Detalle_Deuda_Anterior.

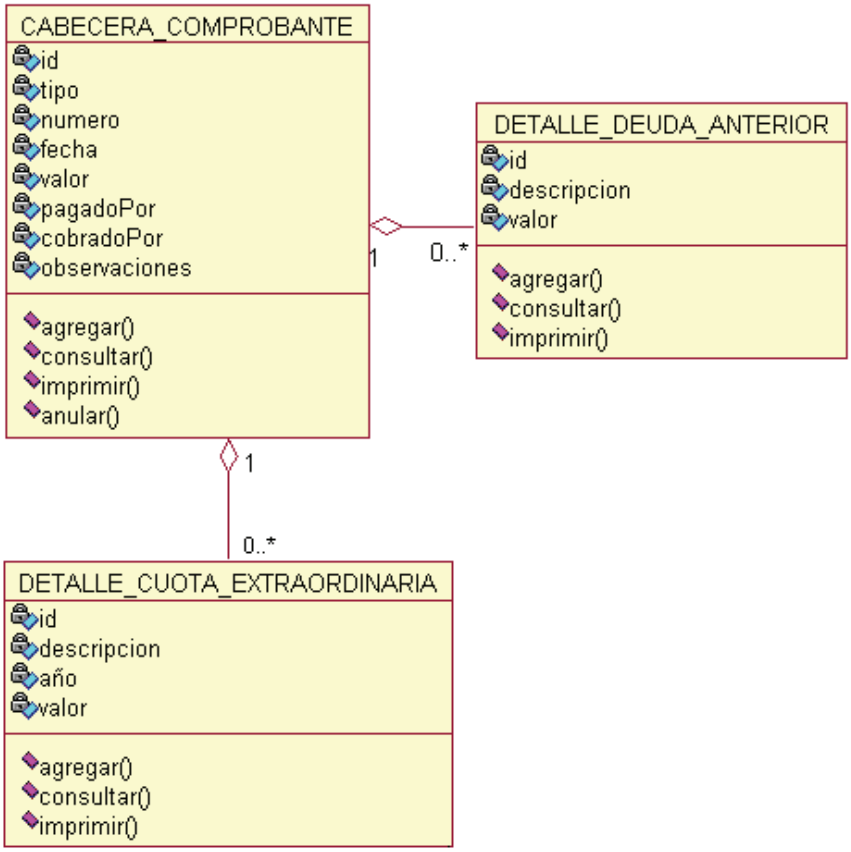


FIGURA 2.9 REPRESENTACIÓN GRÁFICA DE UNA AGREGACIÓN
ELABORADO POR: EL AUTOR

2.3.4.5.1.3 Generalización

Gestiona la complejidad mediante un ordenamiento de clases, emplea mecanismos de abstracción y/o especialización. La generalización consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general (Super Clase o clase padre). Indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por lo tanto la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase, como se muestra en la figura 2.10.

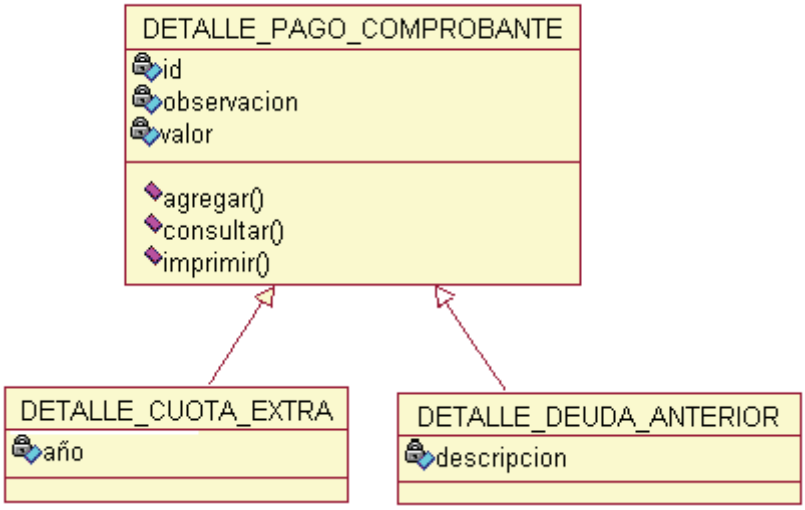


FIGURA 2.10 REPRESENTACIÓN GRÁFICA DE UNA GENERALIZACIÓN
ELABORADO POR: EL AUTOR

2.3.4.5.2 Diccionario de Clases

El diccionario de clases o diccionario de datos describe textualmente las clases identificadas durante el modelo del dominio del problema. Este diccionario sirve como un glosario de términos.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos mas importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

Se debe emplear el diccionario de datos tanto en la fase de análisis como en la de diseño. Mientras en la primera fase permite identificar la información a emplear; en la segunda, permite organizar la información que luego será almacenada en diferentes dispositivos de almacenamiento, sean estos primarios o secundarios.

Identifica los procesos donde se emplean los datos y los sitios dónde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

2.3.4.6 Diagramas de Interacción

Los diagramas de interacción representan un conjunto de mensajes que se intercambian entre un conjunto de objetos para lograr un objetivo.

Estos diagramas se utilizan para modelar los aspectos dinámicos de un sistema y para construir sistemas ejecutables por medio de ingeniería directa e inversa. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de caso de uso.

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en la representación de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

2.3.4.6.1 Diagrama de Secuencia

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Es útil para observar la vida de los objetos en el sistema, identificar llamadas o reconocer posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema.

Es imposible representar en un solo diagrama de secuencia, todas las secuencias posibles del sistema, por ello se escoge un escenario en concreto, el cual tiene punto de partida y muestra la secuencia de mensajes entre objetos.
En el diagrama de secuencia se muestra el orden de las llamadas en el sistema. El diagrama se forma con los objetos que forman parte de la secuencia, éstos se sitúan en la parte superior de la pantalla, de izquierda a derecha; esta ordenación horizontal de los objetos no tienen ningún significado.

De estos objetos sale una línea que indica su vida en el sistema, el tiempo transcurre de arriba hacia abajo. Cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua. Esta línea de vida simple, se convierte en una línea gruesa cuando representa que el objeto tiene el foco del sistema, es decir cuando él está activo.

Los diagramas de secuencia, formalmente diagramas de traza de eventos o de interacción de objetos, se utilizan con frecuencia para validar los casos de uso. Documentan el dise no desde el punto de vista de los casos de uso. Observando qué mensajes se envían a los objetos, componentes o casos de uso y viendo a grosso modo cuanto tiempo consume el método invocado, los diagramas de secuencia nos ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos. A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del diagrama de clases.

2.3.4.6.2 Diagrama de Colaboración.

Un Diagrama de Colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes. A diferencia de los diagramas de secuencia, los diagramas de colaboración incluyen las relaciones entre objetos pero no secuencias temporales.

Los Diagramas de Colaboración muestran interacciones organizadas alrededor de los roles y no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con *números de secuencia* tanto la secuencia de mensajes como los hilos concurrentes.

Un diagrama de colaboración es también un diagrama de clases que contiene roles de clasificador y roles de asociación en lugar de solo clasificadores y asociaciones. Los roles de clasificador y los de asociación describen la configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la colaboración.

Los Diagramas de colaboración, son útiles en la fase exploratoria para identificar objetos.Un flujo de mensajes, expresa el envío de un mensaje. Se representa con una flecha dirigida colocada cerca de un enlace. Esta flecha lleva el nombre del mensaje con sus respectivos parámetros entre paréntesis, en donde es opcional incluir el tipo de dato del parámetro, y puede haber un valor de retorno, al cual también se le indica el tipo.

Estos diagramas están indicados para mostrar una situación o flujo de programa específico y son considerados uno de los mejores diagramas para mostrar o explicar rápidamente un proceso dentro de la lógica del programa.

2.3.4.7 Diagrama de Actividades

Un Diagrama de Actividad muestra la realización de operaciones para conseguir un objetivo. Presenta una visión simplificada de lo que ocurre en un proceso, mostrando los pasos que se realizan, por eso se constituye en uno de los diagramas que modelan los aspectos dinámicos del sistema.

El Diagrama de Actividades es una especialización del Diagrama de Estado, organizado respecto de acciones y usado para especificar un método, un caso de uso o un proceso de negocio. Este diagrama que muestra el flujo de control entre actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad. Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes.

Gráficamente un Diagrama de Actividad es una colección de rectángulos redondeados, líneas gruesas, círculos rellenos, rombos, pentágonos, etc que representan actividades, acciones, barras de sincronización, indicadores de inicio y término, decisiones, carriles, flujo y estados de objetos, recepción y envío de señales, eventos diferidos.

2.3.5 CONSTRUCCION

2.3.5.1 Rational Rose 2000 [9, 10]

Esta herramienta de diseño, Rational Rose 2000, nace para representar modelos de sistemas especialmente intensivos en el uso de software. Es generada por consecuencia del Lenguaje Unificado del Modelado (UML), que es un lenguaje simbólico que ha tenido un gran éxito en el mercado, fue creado por Booch, Rumbaugh y Jacobson, grandes metódologos, quienes se reunieron en Rational Software, para crear una simbología unificada de sus métodos.

Dos características populares del Rational Rose 2000 son su capacidad de proporcionar el desarrollo iterativo y permitir que los diseñadores aprovechen el desarrollo iterativo (a veces llamado desarrollo evolutivo).

Rational Rose es una herramienta muy fácil de manejar con una gran facilidad y potencia en el diseño de todos los diagramas planteados en el UML, entre sus ventajas podemos listar:

- Posee una sección dedicada al ingreso de casos de uso, diagrama de clases, diagrama de secuencia y la generación automática de diagramas de colaboración, los cuales son fácilmente exportables a cualquier procesador de texto.

- Permite el manejo de actores en los diagramas de casos de uso, los cuales serán en lo posterior interpretados por el sistema como clases para la construccion de otros diagramas complementarios.

- Cada sistema dentro de Racional Rose recibe un nombre y está ligado a una ventana para un rápido acceso por parte del usuario.

---

[9] AVILA, C; *Modelando con UML. Principios y Aplicaciones*; Perú;Primera edición;2001

[10] http://www.di.uniovi.es/~cernuda/pfc/aoo.pdf

2.3.5.2 Microsoft SQL Server 2000 [11]

Microsoft SQL Server 2000 es un potente motor de bases de datos de alto rendimiento, capaz de soportar millones de registros por tabla, con un interface intuitivo y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable con el que podemos desarrollar cualquier aplicación que manipule componentes de SQL Server.

Microsoft SQL Server revoluciona el concepto de Base de Datos empresarial,. Ya que reúne en un sólo producto la potencia necesaria para cualquier aplicación empresarial crítica junto con unas herramientas de gestión que reducen al mínimo el coste de propiedad. Entre las características presentadas por la herramienta podemos listar las siguientes:

Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles, es decir a empresas centralizadas u oficinas distribuidas, replicando cientos de sites.

Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server. Posee los mejores registros de los benchmarks independientes (TCP) tanto en transacciones totales como en coste por transacción.

Gestión: Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft

Plataforma de desarrollo fácil y abierto: Integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server ofrece libertad de desarrollo y programación al usuario.

---

[11] http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml

2.3.5.3 Microsoft Visual Basic 6.0 [12, 13]

Visual Basic es un lenguaje de programación desarrollado para Microsoft por Alan Cooper. El lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma. Desde el 2001 Microsoft ha propuesto abandonar el desarrollo basado en la API Win32 y pasar a trabajar sobre un framework o marco común de librerías independiente de la version del sistema operativo, .NET Framework, a través de Visual Basic .NET (y otros lenguajes como C Sharp (C#) de fácil transición de código entre ellos) que presenta serias incompatibilidades con el código Visual Basic existente.

Visual Basic constituye un IDE (entorno de desarrollo integrado o en inglés Integrated Development Enviroment) que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

El compilador de Microsoft genera ejecutables que requieren una DLL para que funcionen, ya que complementa las funciones implementadas en el lenguaje. Además existen un gran número de bibliotecas (DLL) que facilitan el acceso a muchas funciones del sistema operativo y la integración con otras aplicaciones. Sin embargo esto sólo es una limitación en sistemas obsoletos, más se ha podido superarlos en Windows 2000 o Windows XP.

---

[12] http://es.wikipedia.org/wiki/Microsoft_Visual_Basic

[13] http://www.monografias.com/trabajos10/visual/visual.shtml

La ventana de propiedades contiene diferentes formas para utilizar el programa, cuales son: (Pointer) Apuntador o puntero, (Label) Etiqueta, (Frame) Marco, (CheckBox) Casilla de verificación, (ComboBox) Lista desplegable, (HScrollBar) Barra de desplazamiento horizontal, (Timer) Temporizador, (DirListBox) Lista de directorios, (Shape) Figura, (Image) Imagen, "OLE", (PictureBox) Caja de Imagen, (TextBox) Caja de texto, (CommandButton) Boton de pulsación, (OptionButton) Boton de opción, (ListBox) Lista, (VScrollBar) Barra de desplazamiento vertical, (DriveListBox) Lista de unidades de disco, (FileListBox) Lista de archivos, (Line) Linea y por último (Data) Datos.

Las ventajas mostradas por la herramienta, son las siguientes:

- Posee una curva de aprendizaje muy rápida.
- Integra el diseño e implementación de formularios de Windows.
- Permite usar con suma facilidad la plataforma de los sistemas Windows.
- El código en Visual Basic es fácilmente migrable a otros lenguajes.

Los inconvenientes detectados dentro de Visual Basic son los siguientes:

- No es multiplataforma, sin embargo se pueden usar emuladores e interpretes para correrlos en otras plataformas.
- Por defecto permite la programación sin declaración de variables, lo que puede ser sencillamente corregido escribiendo la frase Option Explicit.
- Sólo permite el uso de funciones de librerías dinámicas (DLL) stdcall.
- No maneja muy bien los apuntadores de memoria.
- No soporta tratamiento de procesos como parte del lenguaje.
- No incluye operadores de desplazamiento de bits como parte del lenguaje.
- No permite el manejo de memoria dinámica, punteros, etc. como parte del lenguaje.
- El tratamiento de mensajes de Windows es básico e indirecto.
- La gran gama de controles incorporados son, sin embargo en algunos casos, muy generales, lo que lleva a tener que reprogramar nuevos controles para una necesidad concreta de la aplicación.

2.3.5.4 Cliente Servidor [14]

Para lograr las características principales del manejo de base de datos, como el trabajo a distancia y compartir información, se debe trabajar con grupos de computadoras interconectadas en una red. La arquitectura *cliente/servidor* es un modelo de interacción común entre aplicaciones en una red, para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Este modelo es una comunicación basada en una serie de preguntas y respuestas, que asegura que si dos aplicaciones intentar comunicarse, una comienza la ejecución y espera indefinidamente que la otra le responda y luego continua con el proceso.

FIGURA 2.11 REPRESENTACIÓN DEL MODELO CLIENTE-SERVIDOR
ELABORADO POR: EL AUTOR

Los principales componentes del esquema cliente/servidor son los Clientes, los Servidores y la infraestructura de comunicaciones. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes. El Servidor es una computadora central, de gran capacidad, compartida por las otras computadoras de la red, llamadas Clientes.

FIGURA 2.12 FLUJO DE LA ARQUITECTURA CLIENTE-SERVIDOR
ELABORADO POR: EL AUTOR

---

[14] http://www.ucm.es/info/Psyap/Prieto/alum9798/intranet01/cliente.htm

El diseño de los servidores debe ser muy cuidadoso, debe incluir código para:

- Autenticación: verificar la identidad del cliente.
- Seguridad de datos: para que éstos no puedan ser accedidos inapropiadamente.
- Privacidad: garantizar que la información privada de un usuario, no sea accedida por alguien no autorizado.
- Protección: asegurar que las aplicaciones no monopolicen los recursos del sistema.
- Autorización: verificar si el cliente tiene acceso al servicio proporcionado por el servidor.

Las ventajas ofrecidas por ésta arquitectura son:

- Esta arquitectura da la posibilidad de conectar clientes y servidores independientemente de sus plataformas, pues las plataformas de software y hardware entre clientes y servidores son independientes.

- Proporciona una mayor consistencia a la información, porque posee un control centralizado de los elementos compartidos y porque facilita la construcción de interfaces gráficas interactivas, las cuales hacen que los datos se conviertan en información.

- Los costos de compra, arrendamiento y mantenimiento de grandes computadoras centrales son más altos que los que se realizan con la compra de componentes para crear una red de área local. Con frecuencia sucede que el costo de un sistema cliente-servidor completo es inferior al de la instalación de una computadora central.

- Permite la escalabilidad horizontal y vertical. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

- La arquitectura Cliente/Servidor constituye el ambiente ideal para facilitar el acceso a la información. El usuario no sólo puede tener un acceso transparente a toda la información que necesita, sino además está habilitado para procesarla como guste.

- Proporciona Flexibilidad en cuanto a que podemos elegir entre diseños de bases de datos, comunicación entre procesos, procesamiento de transacciones, generación de GUI (interfaces gráficas de usuarios).

- Permite distribuir físicamente los procesos y los datos en forma más eficiente pues establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.

- El cliente puede trabajar en el ambiente que más le convenga sin preocuparse de la conversión de interfaces y protocolos. Se concentra en el trabajo que debe realizar más que en la tecnología ya que la arquitectura lo realiza mediante procesos internos.

Las desventajas detectadas en ésta arquitectura son:

- Pérdida de flexibilidad, pues va a la par con la tecnología y está ofrece beneficios y sobre todo promueve el establecimiento de sistemas abiertos; por tanto, le otorga al modelo cliente/servidor un alto grado de complejidad, característica propia de los sistemas abiertos debido a la gran cantidad de componentes que no siempre tienen un compromiso firme o permanente entre sí.

- La gran cantidad de información que viaja al cliente congestiona demasiado el tráfico de red, lo que se traduce en bajo rendimiento para cualquier aplicación que utilice dicho medio.

2.3.6 PRUEBA FUNCIONAL [15]

La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. El objetivo de las pruebas es garantizar el correcto y preciso funcionamiento de un sistema, es decir que el producto de software no tenga errores de diseño y/o programación. Se sugieren realizar pruebas de unidad, de integración y de validación del sistema.

El objetivo de las pruebas funcionales es validar si el comportamiento observado en el software cumple o no con sus especificaciones. Las funciones son probadas ingresando las entradas y examinando las salidas. La estructura interna del programa raramente es considerada.

Principios básicos que guían las pruebas del software:

- A todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente.
- Las pruebas deben planificarse muchos antes de que empiecen.
- Las pruebas deberían empezar por lo pequeño y progresar hacia lo grande.
- No es posible realizar pruebas exhaustivas.
- Para ser más eficaces, las pruebas deberían ser realizadas por un equipo de trabajo independiente del desarrollo del sistema.

Para realizar pruebas funcionales, la especificación se analiza para derivar los casos de prueba. Técnicas como partición de equivalencia, análisis del valor límite, grafo causa-efecto y conjetura de errores son especialmente pertinentes para las pruebas funcionales. Se deben considerar condiciones inválidas e inesperadas de la entrada y tener en cuenta que la definición del resultado esperado es una parte vital de un caso de la prueba.

---

[15] PRESSMAN, R; *Ingeniería de Software, un enfoque práctico*; Quinta Edición; 2000, pag. 282

2.3.7 MANTENIMIENTO [16, 17]

El objetivo del mantenimiento es que el sistema se mantenga en constante funcionamiento, adicionalmente se considera que es la continuación del ciclo de vida del desarrollo de software mediante el paradigma espiral incremental, ya que luego de haber completado una primera versión de este, se deben considerar extensiones del sistema de acuerdo con las nuevas necesidades y requerimientos. De cierta manera, el mantenimiento significa seguir un nuevo ciclo de actividades de desarrollo, pero a partir de un sistema ya existente.

Una vez que la aplicación se encuentra en producción, es decir está funcionando para el cliente, cumpliendo con los objetivos y requisitos para los cuales fue creada; se emplea el mecanismo de mantenimiento para que asegure que los distintos subsistemas estén balanceados y que el sistema total se mantenga en equilibrio con su medio.

Se debe considerar que la mayor parte de sistemas de software elaborados, van a sufrir modificaciones a futuro; pues la etapa de mantenimiento supondrá pequeñas operaciones tanto de corrección como de mejora de la aplicación, sean estás fruto de nuevos requerimientos del usuario como causados por la evolución de la tecnología.

---

[16] WEITZENFELD, A; *Software Orientada a Objetos con UML, Java e Internet*; Tercera Edición; 2005, pag. 42

[17] http://www.manycomics.com/se/ciclo.htmx

CAPITULO 3

CONCLUSIONES Y RECOMENDACIONES

3.1 CONCLUSIONES

- Por medio del presente trabajo tuve una experiencia práctica en la cual se aplica de manera resumida los conocimientos adquiridos en la carrera de forma global.
- Para el análisis y desarrollo del sistema se aplicó la Metodología OMT, lo que permitió llevar un control de calidad, ayudándome a cumplir los objetivos planteados.
- Para el desarrollo del proyecto, fue importante la evaluación permanente del usuario a fin de generar versiones más completas de la aplicación.
- Después de mi experiencia de desarrollo utilizando Visual Basic 6.0, puedo recomendarla como herramienta de desarrollo debido a la facilidad de programación e interacción con un sin número de bases de datos.

3.2 RECOMENDACIONES

- Se recomienda cumplir con estándares para la documentación y desarrollo de un sistema de software, ya que de esta forma quedará abierto y flexible para el futuro cuando los métodos y la tecnología avancen.
- En un tiempo prudencial (un año), se debe evaluar la funcionalidad del sistema para verificar su pertinencia.
- Se recomienda realizar un buen plan de capacitación del sistema, el mismo que debería ser claramente definido y difundido dentro de los usuarios que van a manejarlo, con lo cual se logrará obtener un buen desempeño del sistema.
- Será recomendable que en el futuro se planifique elaborar una interfase que permita al sistema obtener información generada desde cualquier aplicación contable.

3.3 GLOSARIO

ABSTRACCIÓN: Características esenciales de una entidad que lo distinguen de los demás tipos de entidades. Una abstracción define un límite con respecto a la perspectiva del espectador.

ACTOR: Un conjunto coherente de las funciones que los usuarios desempeñan dentro de un caso de uso. Un actor tiene un papel para cada caso de uso con el que se comunica.

CLASE: Una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica. Una clase puede utilizar un conjunto de interfaces para especificar las colecciones de las operaciones que proporciona a su entorno.

ESCENARIO: Secuencia específica de acciones de un usuario que ilustra narrativamente un comportamiento.

HERENCIA: Indica que una subclase hereda los métodos y atributos especificados por una *Super Clase*, por lo tanto la *Subclase* además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase.

OMT (Object Modeling Techniques): Una de las metodologías de análisis y diseño orientadas a objetos, más maduras y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público.

SOFTWARE: Componente intangible en la informática, que generalmente se trata de una serie de instrucciones elaboradas por humanos en lenguajes de programación de alto nivel (código fuente) que luego son traducidas por un compilador a código máquina.

UML (Lenguaje Unificado de Modelado): Es un lenguaje gráfico que permite modelar los elementos que constituyen un Sistema de Software.

3.4 BIBLIOGRAFÍA

1. Yourdon, Edward. 1993, Análisis Estructurado Moderno. Toronto, Prentice Hall.
2. Kendall Kenneth E. 1997, Análisis y Diseño de Sistemas, México, México: Prentice Hall .
3. Pressman Rogers, Ingeniería del Software: Un Enfoque Práctico. Madrid, España: Mc Graw-Hill.
4. Weitzenfeld, Alfredo. 2005, *Software Orientada a Objetos con UML, Java e Internet*. México, México: Thomson.
5. http://www.osmosislatina.com/aplicaciones/bases_de_datos.htm, BASES DE DATOS.
6. http://rosalia.dc.fi.udc.es/pablo/Public_html/BASE_DATOS.html, BASES DE DATOS, CARACTERISTICAS.
7. http://bpccee20.ccee.ucm.es/guia/dir-asis/guiabases.htm, GUIAS DE BASES DE DATOS.
8. http://www.airtel.net/manuales/usuario/servicios/serv_7_cgi.htm, MANUALES DE BASES DE DATOS , SERVICIOS.
9. http://www.dbinternet.com.ar/imple.htm, IMPLEMENTACION DE BASES DE DATOS.
10. http://www.arsys.es/soporte/programacion/servbases.htm, PROGRAMACION.
11. http://www.microsoft.com/catalog/display.asp?site=729&subid=45&pg=2, MICROSOFT VISUAL BASIC 6.0.
12. Manual de Ayuda de Crystal Reports 8.5.

ANEXOS

Manual Técnico

CONTENIDO

INTRODUCCIÓN

1. ANALISIS

1.1 MODELO ESTÁTICO…….....48
1.1.1 Descripción de Actores.……48
1.1.2 Diagrama de Casos de Uso…… 49
1.1.3 Descripción General de Casos de Uso……. 50
1.1 4 Descripción Especifica de Casos de Uso…… 52
1.1.5 Diagrama de Clases……..75
1.1.6 Diccionario de Clases……... 76

2. DISEÑO

2.1 MODELO DINÁMICO…….....78
2.1.1 Diagramas de Colaboración…… 78
2.1.2 Diagramas de Secuencia……. 86

3. PRUEBAS

3.1.1 Pruebas Funcionales…….....93
3.1.2 Pruebas de Usuario…….....103

INTRODUCCIÓN

El presente manual esta orientado para usuarios avanzados o con conocimiento de Microsoft Visual Basic y Microsoft SQL Server 2000 , con el objetivo principal de describir técnicamente el programa a fin de facilitar futuros mantenimientos.

1. ANÁLISIS

Como primera etapa del diseño se construye el modelo del dominio, con la finalidad de mostrar las propiedades más importantes del Sistema de Administración de Edificios.

1.1 MODELO ESTÁTICO

El UML puede describir cualquier tipo de sistema en términos de diagramas orientados a objetos, razón por la cual se lo eligió para dar diferentes perspectivas del Sistema de Administración de Edificios.

1.1.1 DESCRIPCIÓN DE ACTORES

En la siguiente descripción se muestra a cada actor y el conjunto de roles que desempeñan cuando interaccionan el Sistema de Administración de Edificios.

Actores	Descripción
Administrador	Persona habilitada dentro del sistema para crear condominios y definir lo parámetros relativos al mismo, además de poder generar el cierre del año. Tiene acceso completo a las opciones de ingresos, egresos, reportes, consultas y balances.
Contador	Persona habilitada para registrar el cobro de alícuotas, cuotas extraordinarias y deudas anteriores. Registrar la información correspondiente a los egresos generados en el condominio. Tiene acceso a reportes, consultas y balances.
Secretaria	Persona que tiene acceso únicamente acceso a reportes, consultas y balances.

1.1.2 DIAGRAMA DE CASOS DE USO

En la Figura 1, se muestra los actores que interactúan en el sistema y los respectivos casos de uso con sus relaciones.

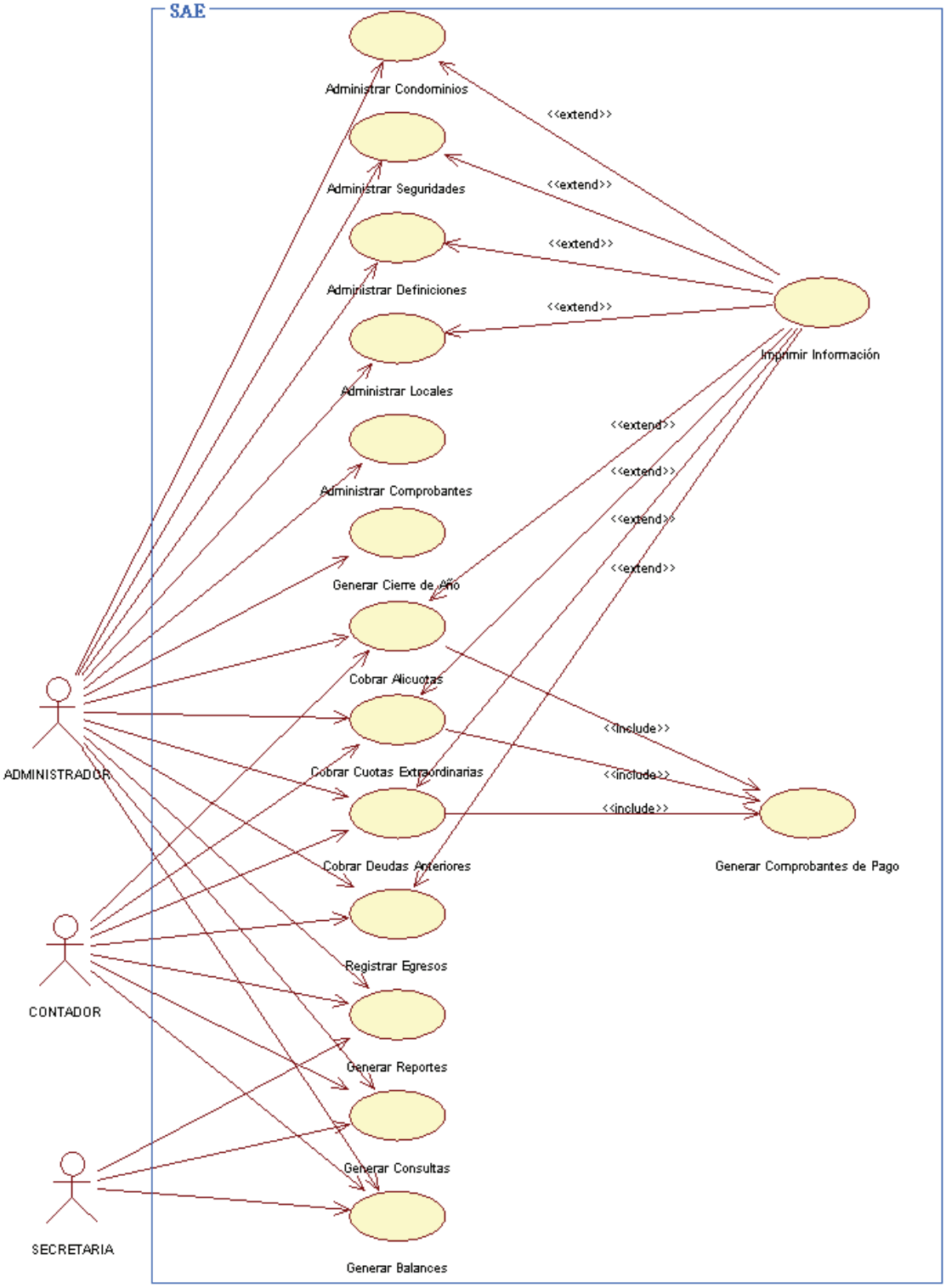


FIGURA 1: DIAGRAMA DE CASOS DE USO
ELABORADO POR: EL AUTOR

1.1.3 DESCRIPCIÓN GENERAL DE CASOS DE USO

Los Casos de Uso generales que se definieron para el diseño del Sistema de Administración de Edificios son los siguientes:

Caso de Uso	Actor(es)	Descripción
Administrar Condominios	Administrador	Permite al usuario la creación, edición, impresión y eliminación de condominios.
Administrar Seguridades	Administrador	Permite la asignación de perfiles a los usuarios según las actividades que desempeñan.
Administrar Definiciones	Administrador	Configura las opciones del sistema a fin de establecer el modo de trabajo.
Administrar Locales	Administrador	Permite al usuario la creación, edición, impresión y eliminación de locales.
Administrar Comprobantes	Administrador	Permite la reimpresión y anulación de comprobantes de pago.
Generar Cierre de Año	Administrador	Realiza todas las acciones necesarias para cerrar el año contable e iniciar el nuevo año.
Cobrar Alícuotas	Administrador Contador	Registra el pago por concepto de alícuotas mensuales que cada local cancela para el mantenimiento del condominio.
Cobrar Cuotas Extraordinarias	Administrador	Registra el pago de cuotas

	Contador	extraordinarias que cada local cancela a fin de cubrir gastos adicionales en la administración del condominio.
Cobrar Deudas Anteriores	Administrador Contador	Registra las obligaciones pendientes por concepto de alícuotas de años anteriores que no han sido canceladas.
Registrar Egresos	Administrador Contador	Permite el ingreso de la información y los valores de egresos pagados por concepto de mantenimiento del condominio.
Generar Reportes	Administrador Contador Secretaria	Contiene la funcionalidad asociada a la impresión de reportes según la información requerida por el usuario.
Generar Consultas	Administrador Contador Secretaria	Contiene la funcionalidad asociada a la impresión de consultas según los parámetros ingresados por el usuario.
Generar Balances	Administrador Contador Secretaria	Contiene la funcionalidad asociada a la impresión de balances de ingresos y egresos del periodo requerido por el usuario.

1.1.4 DESCRIPCIÓN ESPECÍFICA DE CASOS DE USO

Es necesario describir cada uno de los Casos de Uso a fin de comprender de mejor manera el diseño del Sistema de Administración de Edificios.

CU-01
Administrar Condominios
DESCRIPCIÓN:
Permite al usuario la creación, edición, impresión y eliminación de condominios.
ACTORES:
Administrador
PRECONDICIONES:
El usuario debe seleccionar al iniciar sesión el edificio en el cual va a trabajar.
POSCONDICIONES:
Condominio creado Condominio modificado Condominio eliminado
FLUJO BÁSICO:
Subflujo Básico 1: Crear Condominio 1. El usuario ingresa al menú Administración >> Condominios >> Crear Condominio 2. El sistema despliega los campos con la información a llenarse 3. El usuario ingresa la información y presiona el botón “Crear” 4. El sistema valida la información ingresada 5. Si la información ingresada es correcta, el sistema guarda el nuevo condominio en la Base de Datos 6. El sistema despliega un mensaje indicando que el registro se ha guardado con éxito

7. Si el usuario presiona el botón “Imprimir”, se realiza el caso de uso “Imprimir Información”
8. Fin del Caso de Uso

Subflujo Básico 2: Editar Condominio

1. El usuario ingresa al menú Administración>> Condominios >> Editar Condominio
2. El sistema despliega la información del condominio actual
3. El usuario modifica los datos del condominio y presiona el botón “Guardar”
4. El sistema valida la información ingresada
5. Si la información ingresada es correcta, el sistema actualiza el condominio en la Base de Datos
6. El sistema despliega un mensaje indicando que el registro se ha guardado con éxito
7. Si el usuario presiona el botón “Imprimir”, se realiza el caso de uso “Imprimir Información”
8. Fin del Caso de Uso

Subflujo Básico 3: Eliminar Condominio

1. El usuario ingresa al menú Administración>> Condominios >> Eliminar Condominio
2. El sistema despliega un mensaje de confirmación si desea eliminar el Condominio
3. Si el usuario selecciona la opción “Si”, el sistema elimina el condominio de la Base de Datos
4. El sistema despliega un mensaje indicando que el condominio ha sido eliminado y que el usuario debe salir del sistema e ingresar nuevamente para que los cambios se hagan efectivos
5. Fin del Caso de Uso

FLUJO ALTERNATIVO:
Flujo alternativo 1: Crear Condominio 5’ El usuario no ingresó la información obligatoria El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios Flujo alternativo 2: Editar Condominio 5’ El usuario no ingresó la información obligatoria El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios Flujo alternativo 3: Eliminar Condominio 3’ El usuario selecciona la opción “No” El sistema cancela la acción de eliminación del condominio.

CU-02
Administrar Seguridades
DESCRIPCIÓN:
Permite la asignación de perfiles a los usuarios según las actividades que desempeñan.
ACTORES:
Administrador
PRECONDICIONES:
El usuario debe estar registrado dentro del sistema con los accesos necesarios.
POSCONDICIONES:
Perfil asignado a usuario.
FLUJO BÁSICO:
Subflujo Básico 1: Crear Usuario

1. El usuario ingresa al menú Administración >> Seguridades >> Usuarios/ Accesos.
2. El sistema despliega los campos necesarios y carga los perfiles que pueden ser asignados.
3. El usuario ingresa la información, selecciona el perfil y presiona el botón guardar.
4. El sistema valida que el usuario haya seleccionado un perfil.
5. Si el usuario ha ingresado la información correctamente, el sistema guarda la información en la base de datos..
6. Si el usuario presiona el botón “Imprimir Todos”, se realiza el caso de uso “Imprimir Información de Usuarios”
7. Fin del Caso de Uso

Subflujo Básico 2: Editar Usuario

1. El usuario ingresa al menú Administración >> Seguridades >> Usuarios/ Accesos.
2. El sistema carga la información de los usuarios existentes en el sistema.
3. El usuario selecciona el registro del usuario que requiera modificar, y luego pulsa el botón “editar”.
4. El sistema habilita el botón guardar y deshabilita el resto de las opciones dentro del formulario.
5. El usuario modifica la información y presiona el botón guardar.
6. Si el usuario ha ingresado la información correctamente, el sistema guarda la información en la base de datos..
7. Si el usuario presiona el botón “Imprimir Todos”, se realiza el caso de uso “Imprimir Información de Usuarios”
8. Fin del Caso de Uso

Subflujo Básico 3: Eliminar Usuario

1. El usuario ingresa al menú Administración >> Seguridades >> Usuarios/ Accesos. 2. El sistema carga la información de los usuarios existentes en el sistema. 3. El usuario selecciona el registro del usuario que requiera eliminar, y luego pulsa el botón “eliminar”. 4. El sistema despliega el mensaje de confirmación. 5. Si el usuario presiona el botón “Si”, el sistema elimina el registro de la base de datos. 6. El sistema despliega un mensaje de información, indicando que el registro fue eliminado exitosamente. 7. Si el usuario presiona el botón “Imprimir Todos”, se realiza el caso de uso “Imprimir Información de Usuarios” 8. Fin del Caso de Uso
FLUJO ALTERNATIVO:
Flujo alternativo 1: Crear Usuario 4’ El usuario no seleccionó ningún perfil. El sistema despliega un mensaje de error indicando que debe asignarse un perfil. Flujo alternativo 2: Editar Usuario 3’ El usuario ingresó información incorrecta El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios Flujo alternativo 3: Eliminar Usuario 3’ El usuario selecciona la opción “No” El sistema cancela la acción de eliminación de usuario.

CU-03
Administrar Definiciones
DESCRIPCIÓN:
Configura las opciones del sistema a fin de establecer el modo de trabajo.

ACTORES:
Administrador
PRECONDICIONES:
El usuario debe seleccionar al iniciar sesión el edificio en el cual va a trabajar. El usuario debe iniciar sesión con el perfil adecuado.
POSCONDICIONES:
Secuencial definido. Cuota extraordinaria definida. Egreso definido. Deuda anterior definida. Tipo de Pago definido.
FLUJO BÁSICO:
Subflujo Básico 1: Definir Secuencial 1. El usuario ingresa al menú Administración >> Definiciones >> Definir Secuencial. 2. El sistema despliega el formulario donde se ingresará el número de comprobante inicial, es decir el número del primer comprobante de pago. 3. El usuario ingresa el número inicial y presiona el botón “Guardar”. 4. El sistema guarda la información ingresada por el usuario en la base de datos y despliega un mensaje de confirmación sobre la modificación del secuencial. . 5. Fin del Caso de Uso Subflujo Básico 2: Definir Cuota Extra 1. El usuario ingresa al menú Administración >> Definiciones >> Definir Cuota Extra 2. El sistema despliega el formulario donde se ingresará la información requerida para crear un nuevo ítem de cuota extra dentro del condominio. 3. El usuario ingresa la información necesaria y presiona el botón guardar.

4. El sistema valida que la información sea correcta.
5. Si la información es correcta el sistema registra la información en la base de datos.
6. Fin del Caso de Uso

Subflujo Básico 3: Definir Egresos

1. El usuario ingresa al menú Administración >> Definiciones >> Definir Egresos.
2. El sistema carga la información del tipo de egresos existentes en el sistema.
3. El usuario ingresa la información, selecciona el tipo de egreso y presiona el botón guardar.
4. El sistema valida que se haya seleccionado un tipo de egreso.
5. Si la información es correcta , el sistema guarda el nuevo egreso en la base de datos.
6. El sistema despliega un mensaje de confirmación de la acción realizada.
7. Fin del Caso de Uso

Subflujo Básico 4: Definir Deudas Anteriores

1. El usuario ingresa al menú Administración >> Definiciones >> Definir Deudas Ant.
2. El sistema carga en el formulario los locales existentes en el condominio actual, los meses y tres años anteriores al actual.
3. El usuario selecciona el local, el mes, el año e ingresa el valor para este registro, y presiona el botón guardar.
4. El sistema valida que se haya ingresado la información obligatoria.
5. Si la información esta completa, el sistema guarda la información en la base de datos.
6. El sistema despliega un mensaje de confirmación de la acción realizada.
7. Fin del Caso de Uso

Subflujo Básico 5: Definir Tipo Pago

1. El usuario ingresa al menú Administración >> Definiciones >> Definir Tipo Pago.
2. El sistema despliega el formulario donde se ingresará el nombre del tipo de egreso.
3. El usuario ingresa el nombre del tipo de pago y presiona el botón "Guardar".
4. El sistema guarda la información ingresada por el usuario en la base de datos y despliega un mensaje de confirmación sobre el ingreso del nuevo tipo de pago.
5. Fin del Caso de Uso

FLUJO ALTERNATIVO:

Flujo alternativo 1: Definir Secuencial

3' El usuario no ingreso el número de comprobante inicial.

El sistema despliega un mensaje de error indicando que debe ingresar un número inicial.

Flujo alternativo 2: Definir Cuota Extra.

3' El usuario ingresó información incorrecta.

El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios.

Flujo alternativo 3: Definir Egresos.

3' El usuario no ingresa toda la información necesaria.

El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios.

Flujo alternativo 4: Definir Deudas Anteriores.

3' El usuario no ingresa toda la información necesaria.

El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios.

Flujo alternativo 5: Definir Tipo de Pago.

3' El usuario no ingresa toda la información necesaria.

El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios.

CU-04
Administrar Locales
DESCRIPCIÓN:
Permite al usuario la creación, edición, impresión y eliminación de locales.
ACTORES:
Administrador
PRECONDICIONES:
El usuario debe seleccionar al iniciar sesión el edificio en el cual va a trabajar. El usuario debe iniciar sesión con el perfil adecuado.
POSCONDICIONES:
Local Creado. Local Modificado. Local Eliminado.
FLUJO BÁSICO:
Subflujo Básico 1: Crear Local 1. El usuario ingresa al menú: Locales >> Editar Locales. 2. El sistema despliega los campos con la información a llenarse y carga los datos relacionados a los presupuestos de años anteriores. 3. El usuario ingresa la información y presiona el botón “Guardar” 4. El sistema valida la información ingresada y calcula el valor de la alícuota. 5. Si la información ingresada es correcta, el sistema guarda el nuevo local en la Base de Datos 6. El sistema despliega un mensaje indicando que el registro se ha guardado con éxito 7. Si el usuario presiona el botón “Imprimir Info”, se realiza el caso de uso “Imprimir Información” 8. Fin del Caso de Uso

Subflujo Básico 2: Editar Local

1. El usuario ingresa al menú: Locales >> Editar Locales.
2. El sistema carga los locales existentes en el condominio actual.
3. El usuario selecciona el local que desea modificar, y presiona el botón "Editar".
4. El sistema valida habilita el botón "Guardar" y bloque el resto de botones dentro del formulario.
5. El usuario modifica la información y presiona el botón "Guardar".
6. El sistema valida la información ingresada.
7. Si la información ingresada es correcta, el sistema actualiza el local en la base de datos
8. El sistema despliega un mensaje indicando que el registro se ha guardado con éxito
9. Si el usuario presiona el botón "Imprimir Info", se realiza el caso de uso "Imprimir Información"
10. Fin del Caso de Uso

Subflujo Básico 3: Eliminar Local

1. El usuario ingresa al menú: Locales >> Editar Locales.
2. El sistema carga los locales existentes en el condominio actual.
3. El usuario selecciona el local que desea eliminar, y presiona el botón "Eliminar".
4. El sistema despliega un mensaje de confirmación si desea eliminar el Local.
5. Si el usuario selecciona la opción "Si", el sistema elimina el Local de la base de datos.
6. El sistema despliega un mensaje indicando que el local ha sido eliminado.

FLUJO ALTERNATIVO:
Flujo alternativo 1: Crear Local 3’ El usuario no ingresó la información obligatoria El sistema despliega un mensaje de error indicando que debe ingresar los campos obligatorios Flujo alternativo 2: Editar Local 5’ El usuario no ingresó la información obligatoria El sistema despliega un mensaje de error indicando que debe ingresar los datos con la información correcta. Flujo alternativo 3: Eliminar Local 5’ El usuario selecciona la opción “No” El sistema cancela la acción de eliminación del condominio

CU-05
Administrar Comprobantes
DESCRIPCIÓN:
Permite la reimpresión y anulación de comprobantes de pago.
ACTORES:
Administrador
PRECONDICIONES:
El usuario debe seleccionar al iniciar sesión el edificio en el cual va a trabajar. El usuario debe iniciar sesión con el perfil adecuado.
POSCONDICIONES:
Comprobante Reimpreso. Comprobante Anulado.
FLUJO BÁSICO:
Subflujo Básico 1: Reimpresión de Comprobantes.

1. El usuario ingresa al menú Administración >> Comprobantes >> Reimp. Comprobantes.
2. El sistema despliega el formulario donde se ingresará el número de comprobante que se requiere imprimir.
3. El usuario ingresa el número de comprobante y presiona el botón "Imprimir".
4. El sistema despliega la pantalla con el comprobante reimpreso.
5. Fin del Caso de Uso

Subflujo Básico 2: Anular Comprobantes.

1. El usuario ingresa al menú Administración >> Comprobantes >> Anular Comprob.
2. El sistema despliega el formulario donde se ingresará el número de comprobante que se requiere anular.
3. El usuario ingresa el número de comprobante y presiona el botón "Anular".
4. El sistema cambia el estado de dicho comprobante a "Anulado" y despliega el mensaje comprobante anulado exitosamente.
5. Fin del Caso de Uso

FLUJO ALTERNATIVO:

Flujo alternativo 1: Reimpresión de Comprobantes.

3' El usuario no ingresó ningún numero

El sistema despliega un mensaje de error indicando que debe ingresar un número de comprobante.

3'' El usuario no ingresó un número de comprobante existente

El sistema despliega un mensaje de error indicando que el número de comprobante que se ingresó no existe.

3''' El usuario ingresó un número de comprobante cuyo estado es anulado.

El sistema despliega un mensaje de error indicando que dicho comprobante se encuentra anulado.

Flujo alternativo 2: Anular Comprobantes.

3’ El usuario no ingresó ningún numero El sistema despliega un mensaje de error indicando que debe ingresar un número de comprobante. 3’’ El usuario no ingresó un número de comprobante existente El sistema despliega un mensaje de error indicando que el número de comprobante que se ingresó no existe. 3’’’ El usuario ingresó un número de comprobante cuyo estado es anulado. El sistema despliega un mensaje de error indicando que dicho comprobante se encuentra anulado.

CU-06
Generar Cierre de Año
DESCRIPCIÓN:
Realiza todas las acciones necesarias para cerrar el año contable e iniciar el nuevo año.
ACTORES:
Administrador
PRECONDICIONES:
El usuario debe seleccionar al iniciar sesión el edificio en el cual va a trabajar. El usuario debe iniciar sesión con el perfil adecuado.
POSCONDICIONES:
Cierre de Año Generado.
FLUJO BÁSICO:
1. El usuario ingresa al menú Administración >> Cierre de Año. 2. El sistema despliega el mensaje de confirmación de la acción. 3. Si el usuario presiona el botón “Si”, el sistema inicializa a cero todos los balances y egresos del sistema y avanza el registro del año. 4. El sistema despliega el detalle de información, indicando al usuario que la acción se realizó con éxito. Fin del Caso de Uso.

FLUJO ALTERNATIVO:
3’ El usuario selecciona la opción “No” El sistema cancela la acción de cierre de año.

CU-07
Cobrar Alícuotas
DESCRIPCIÓN:
Registra el pago por concepto de alícuotas mensuales que cada local cancela para el mantenimiento del condominio.
ACTORES:
Administrador Contador
PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado. El local debe estar creado. Debe estar definido el número de comprobante asignado. Debe estar definido el año en curso. Debe existir el valor de alícuota a pagar.
POSCONDICIONES:
Alícuota registrada
FLUJO BÁSICO:
1. El usuario ingresa al menú Ingresos >> Cobrar Alícuotas. 2. El sistema carga el número de comprobante que se utilizará y año actual. 3. El usuario elige el local. 4. El sistema despliega el valor de la alícuota mensual, la información de los meses pagados y las anotaciones referentes a ese local. 5. Si el usuario presiona el botón “Pagos Previos”, se realiza el caso de uso Consultar Pagos Previos.

6. El usuario marca el mes o los meses a pagar. 7. El sistema despliega el detalle de la información y el total a pagar. 8. El usuario ingresa opcionalmente la persona que realiza el pago, la persona que recibe el pago y la fecha de pago. 9. El usuario ingresa la información del pago. 10. El usuario selecciona la forma de pago e ingresa una observación y el valor. 11. El usuario modifica las anotaciones del local. 12. Si todos los campos obligatorios están ingresados, el sistema habilita el botón “guardar”. 13. El sistema valida que el total a pagar sea igual a la suma de las formas de pago. 14. Si la información es correcta, el sistema registra la nueva alícuota en la base de datos. 15. Se realiza el caso de uso Generar Comprobante de Pago. 16. Fin del caso de uso.
FLUJO ALTERNATIVO:
3’ El usuario no seleccionó un local El sistema bloquea las acciones hasta que se seleccione un local. 11’ El usuario no modifica las anotaciones del local El sistema conserva la misma información previamente guardada. 12’ El usuario no llena los campos necesarios. El sistema no habilita el botón guardar. 13’ El total a pagar no es igual a la las formas de pago. El sistema despliega un mensaje de error.

CU-08
Cobrar Cuotas Extraordinarias
DESCRIPCIÓN:
Registra el pago de cuotas extraordinarias que cada local cancela a fin de cubrir gastos adicionales en la administración del condominio.
ACTORES:
Administrador Contador
PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado. El local debe estar creado. Debe estar definido el número de comprobante asignado. Debe estar definido el año en curso. Debe existir el valor de cada cuota extraordinaria.
POSCONDICIONES:
Cuota extraordinaria Cobrada.
FLUJO BÁSICO:
1. El usuario ingresa al menú Ingresos >> Cobrar Cuotas Extr.. 2. El sistema carga los locales registrados en el condominio actual, y la fecha. 3. El usuario selecciona el local. 4. El sistema despliega las anotaciones referentes a ese local. 5. Si el usuario presiona el botón “Pagos Previos”, se realiza el caso de uso Consultar Pagos Previos. 6. El usuario selecciona la cuota que se pagará. 7. El sistema despliega el valor de la cuota y el comprobante que se utilizará. 8. El usuario ingresa opcionalmente la persona que realiza el pago, la persona que recibe el pago y la fecha de pago. 9. El usuario selecciona la forma de pago e ingresa una observación y el valor.

10. El usuario modifica las anotaciones del local. 11. Si todos los campos obligatorios están ingresados, el sistema habilita el botón “guardar”. 12. El sistema valida que el total a pagar sea igual al valor que se debe cancelar por la cuota extraordinaria seleccionada. 13. Si la información es correcta, el sistema registra el pago de la cuota extraordinaria en la base de datos. 14. Se realiza el caso de uso Generar Comprobante de Pago. 15. Fin del caso de uso.
FLUJO ALTERNATIVO:
3’ El usuario no seleccionó un local El sistema bloquea las acciones hasta que se seleccione un local. 10’ El usuario no modifica las anotaciones del local El sistema conserva la misma información previamente guardada. 11’ El usuario no llena los campos necesarios. El sistema no habilita el botón guardar. 13’ El total a pagar no es igual al valor de la cuota extraordinaria seleccionada. El sistema despliega un mensaje de error.

CU-09
Cobrar Deudas Anteriores
DESCRIPCIÓN:
Registra las obligaciones pendientes por concepto de alícuotas de años anteriores que no han sido canceladas.
ACTORES:
Administrador Contador

PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado. El local debe estar creado. Debe estar definido el número de comprobante asignado. Debe estar definido el año en curso. Debe existir los valores de deudas anteriores.
POSCONDICIONES:
Deuda Anterior Cobrada.
FLUJO BÁSICO:
1. El usuario ingresa al menú Ingresos >> Cobrar Deudas Ant. 2. El sistema carga los locales registrados en el condominio actual, y la fecha. 3. El usuario selecciona el local. 4. El sistema despliega las anotaciones referentes a ese local. 5. Si el usuario presiona el botón “Pagos Previos”, se realiza el caso de uso Consultar Pagos Previos. 6. El usuario selecciona la deuda que se pagará. 7. El sistema despliega el valor de la deuda anterior y el comprobante que se utilizará. 8. El usuario ingresa opcionalmente la persona que realiza el pago, la persona que recibe el pago y la fecha de pago. 9. El usuario selecciona la forma de pago e ingresa una observación y el valor. 10. El usuario modifica las anotaciones del local. 11. Si todos los campos obligatorios están ingresados, el sistema habilita el botón “guardar”. 12. El sistema valida que el total a pagar sea igual al valor que se debe cancelar por concepto de deudas anteriores. 13. Si la información es correcta, el sistema registra el pago de la deuda anterior en la base de datos. 14. Se realiza el caso de uso Generar Comprobante de Pago. Fin del caso de uso.

FLUJO ALTERNATIVO:
3’ El usuario no seleccionó un local El sistema bloquea las acciones hasta que se seleccione un local. 10’ El usuario no modifica las anotaciones del local El sistema conserva la misma información previamente guardada. 11’ El usuario no llena los campos necesarios. El sistema no habilita el botón guardar. 13’ El total a pagar no es igual al valor total de las deudas anteriores.. El sistema despliega un mensaje de error.

CU-10
Registrar Egresos
DESCRIPCIÓN:
Permite el ingreso de la información y los valores de egresos pagados por concepto de mantenimiento del condominio.
ACTORES:
Administrador Contador
PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado. El local debe estar creado. Deben estar definidos los egresos del condominio. Debe estar definido el año en curso. Debe estar definido el saldo del año anterior.
POSCONDICIONES:
Información de egresos registrada.

FLUJO BÁSICO:
1. El usuario ingresa al menú Egresos >> Registrar Egreso. 2. El sistema despliega el formulario donde se ingresará la información de los egresos y carga los meses del año. 3. El usuario selecciona el mes en el que se registrarán los egresos. 4. El sistema despliega los egresos definidos dentro del condominio y la información que se ha ingresado hasta el momento. 5. El usuario ingresa la información y los valores correspondientes a cada egreso. 6. El sistema calcula los totales y saldos del mes. 7. Si todos los campos obligatorios están ingresados, el sistema habilita el botón “guardar”. 8. El sistema valida que exista la fecha de cada valor que se ha ingresado. 9. Si los campos son correctos, el sistema registra la información dentro de la base de datos 10. Se realiza el caso de uso Generar Comprobante de Pago. 11. Fin del caso de uso.
FLUJO ALTERNATIVO:
3’ El usuario no seleccionó un mes El sistema bloquea las acciones hasta que se seleccione un mes. 10’ El usuario no modifica la información del formulario. El sistema conserva la misma información previamente guardada. 5’ El usuario no llena los campos necesarios. El sistema no habilita el botón guardar. 8’ El valor de un ítem de egreso no posee fecha. El sistema despliega un mensaje de error.

CU-11
Generar Reportes
DESCRIPCIÓN:
Contiene la funcionalidad asociada a la impresión de reportes según la información requerida por el usuario.
ACTORES:
Administrador Contador Secretaria
PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado.
POSCONDICIONES:
Reporte Generado.
FLUJO BÁSICO:
1. El usuario ingresa al menú Reportes. 2. El sistema despliega los reportes disponibles. 3. El usuario selecciona el reporte que desea visualizar. 4. El sistema genera el reporte y lo imprime. 5. Fin del caso de uso.
FLUJO ALTERNATIVO:
No existen flujos alternativos.

CU-12
Generar Consultas
DESCRIPCIÓN:
Contiene la funcionalidad asociada a la impresión de consultas según los parámetros ingresados por el usuario.
ACTORES:
Administrador Contador Secretaria
PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado.
POSCONDICIONES:
Consulta Generada
FLUJO BÁSICO:
1. El usuario ingresa al menú Consultas. 2. El sistema despliega las consultas disponibles. 3. El usuario selecciona la consulta que desea visualizar e ingresa los parámetros requeridos por el sistema. 4. El sistema genera la consulta y la imprime. 5. Fin del caso de uso.
FLUJO ALTERNATIVO:
No existen flujos alternativos.

CU-13
Generar Balances
DESCRIPCIÓN:
Contiene la funcionalidad asociada a la impresión de balances de ingresos y egresos del periodo requerido por el usuario.
ACTORES:
Administrador Contador Secretaria
PRECONDICIONES:
El usuario debe estar registrado en el sistema con el perfil adecuado. El condominio debe estar creado.
POSCONDICIONES:
Balance Generado
FLUJO BÁSICO:
1. El usuario ingresa al menú Balances. 2. El sistema despliega los balances disponibles. 3. El usuario selecciona el balance que desea visualizar. 4. El sistema genera el balance y lo imprime. 5. Fin del caso de uso.
FLUJO ALTERNATIVO:
No existen flujos alternativos.

1.1.5 DIAGRAMA DE CLASES

En el siguiente diagrama se muestra los grupos de objetos definidos dentro del Sistema de Administración de Edificios asociados en base a sus características y comportamientos similares.

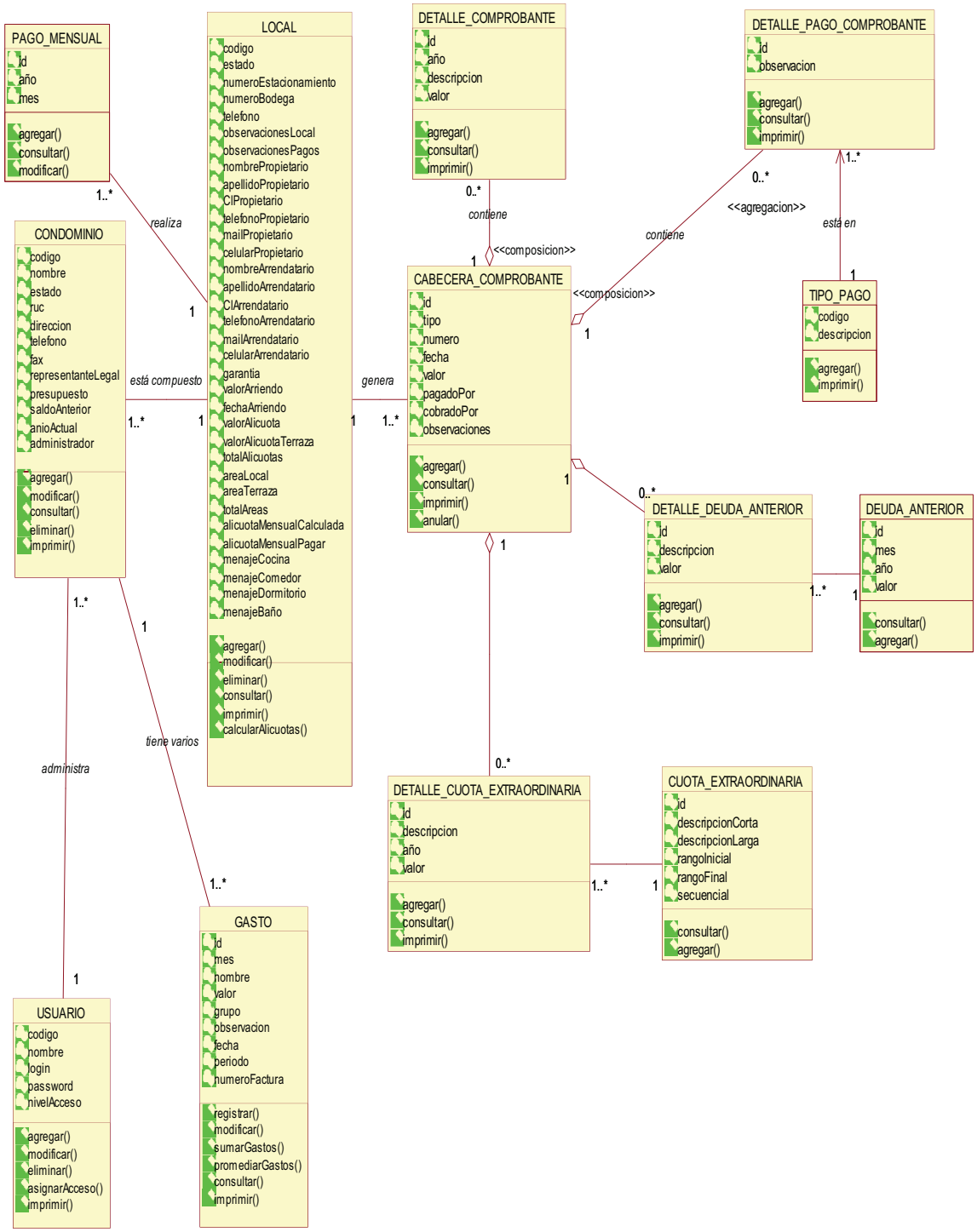


FIGURA 2: DIAGRAMA DE CLASES
ELABORADO POR: EL AUTOR

1.1.6 DICCIONARIO DE CLASES

En el Sistema de Administracion de Edificios se estableció las clases que intervienen en el diseño del mismo, es importante describir con terminos naturales lo que cada una de estas clases representa.

CLASE	DESCRIPCIÓN
CONDOMINIO	Conjunto de propiedades o inmuebles, que residen dentro de un mismo predio, cada propiedad tiene su dueño, mas existen gastos en común, como el mantenimiento de las áreas verdes, cañerías, y otros, que deberán ser cancelados por todos los propietarios.
LOCAL	Propiedad que compone un condominio y que es utilizada para describir de manera general a departamentos y oficinas.
USUARIO	Persona designada por los propietarios de un condominio para encargarse de la recaudación y manejo de los dineros comunales para el mantenimiento del condominio.
GASTO	Representa al valor monetario cancelado por concepto de mantenimiento, o servicios adicionales generados dentro del Condominio y que el pago del mismo compete a todos los propietarios.
PAGO MENSUAL	Representa la cancelación de una alícuota correspondiente a un mes.
CABECERA_COMPROBANTE	Representa la información principal de un Comprobante de Pago entregado como respaldo físico de la recolección de un valor monetario correspondiente al cobro de alícuotas.

CUOTA_EXTRAORDINARIA	Representa la información principal de un Comprobante de Pago entregado como respaldo físico de la recolección de un valor monetario correspondiente al cobro de cuotas extraordinarias.
DEUDA_ANTERIOR	Representa la información principal de un Comprobante de Pago entregado como respaldo físico de la recolección de un valor monetario correspondiente al cobro de deudas de períodos anteriores que aún no han sido canceladas.
DETALLE_COMPROBANTE	Representa los ítems que fueron cancelados mediante un Comprobante de Pago entregado como respaldo físico de la recolección de un valor monetario correspondiente al cobro de alícuotas.
DETALLE_CUOTA_ EXTRAORDINARIA	Representa los ítems que fueron cancelados mediante un Comprobante de Pago entregado como respaldo físico de la recolección de un valor monetario correspondiente al cobro de cuotas extraordinarias.
DETALLE_DEUDA_ANTERIOR	Representa los ítems que fueron cancelados mediante un Comprobante de Pago entregado como respaldo físico de la recolección de un valor monetario correspondiente al cobro de deudas de períodos anteriores que aún no han sido canceladas.
DETALLE_PAGO_ COMPROBANTE	Representa el listado de las formas de pago de cada Comprobante y su respectivo valor.
TIPO_PAGO	Alternativas de pago con las que se puede cancelar una alícuota, cuota extraordinaria o deuda anterior.

2. DISEÑO

En esta fase, de importantes decisiones se definió la arquitectura del Sistema de Administración de Edificios tomando en cuenta las necesidades de rendimiento y también la estructura de análisis propuesta.

2.1 MODELO DINÁMICO

Para representar los aspectos de control del Sistema de Adminsitarcion de Edificios se utilizó diagramas de colaboración y diagramas de secuencia.

2.1.1 DIAGRAMAS DE COLABORACIÓN

Fueron utilizados para representar la interacción entre objetos del Sistema de Administacion de Edificios.

CU-01: Administrar Condominios

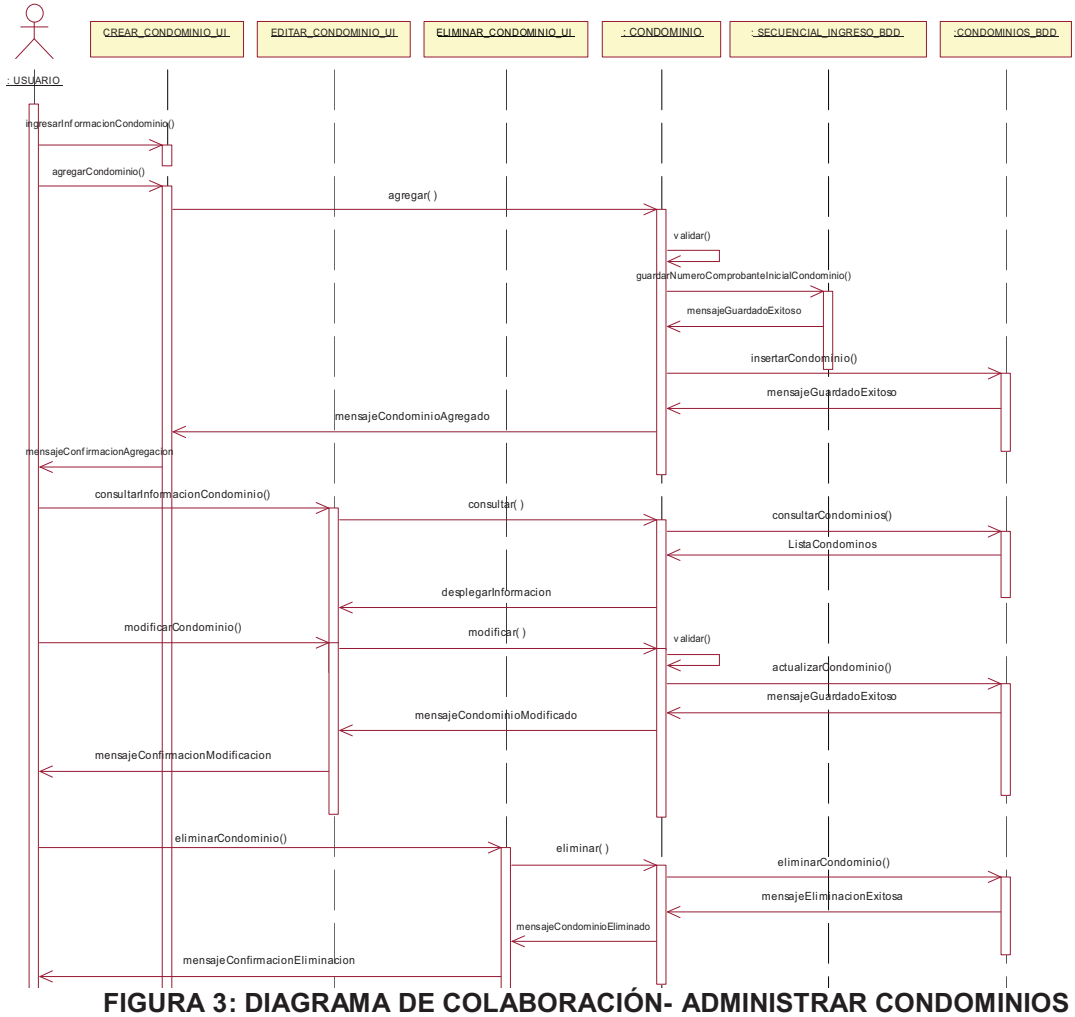


FIGURA 3: DIAGRAMA DE COLABORACIÓN- ADMINISTRAR CONDOMINIOS
ELABORADO POR: EL AUTOR

CU-02: Administrar Seguridades

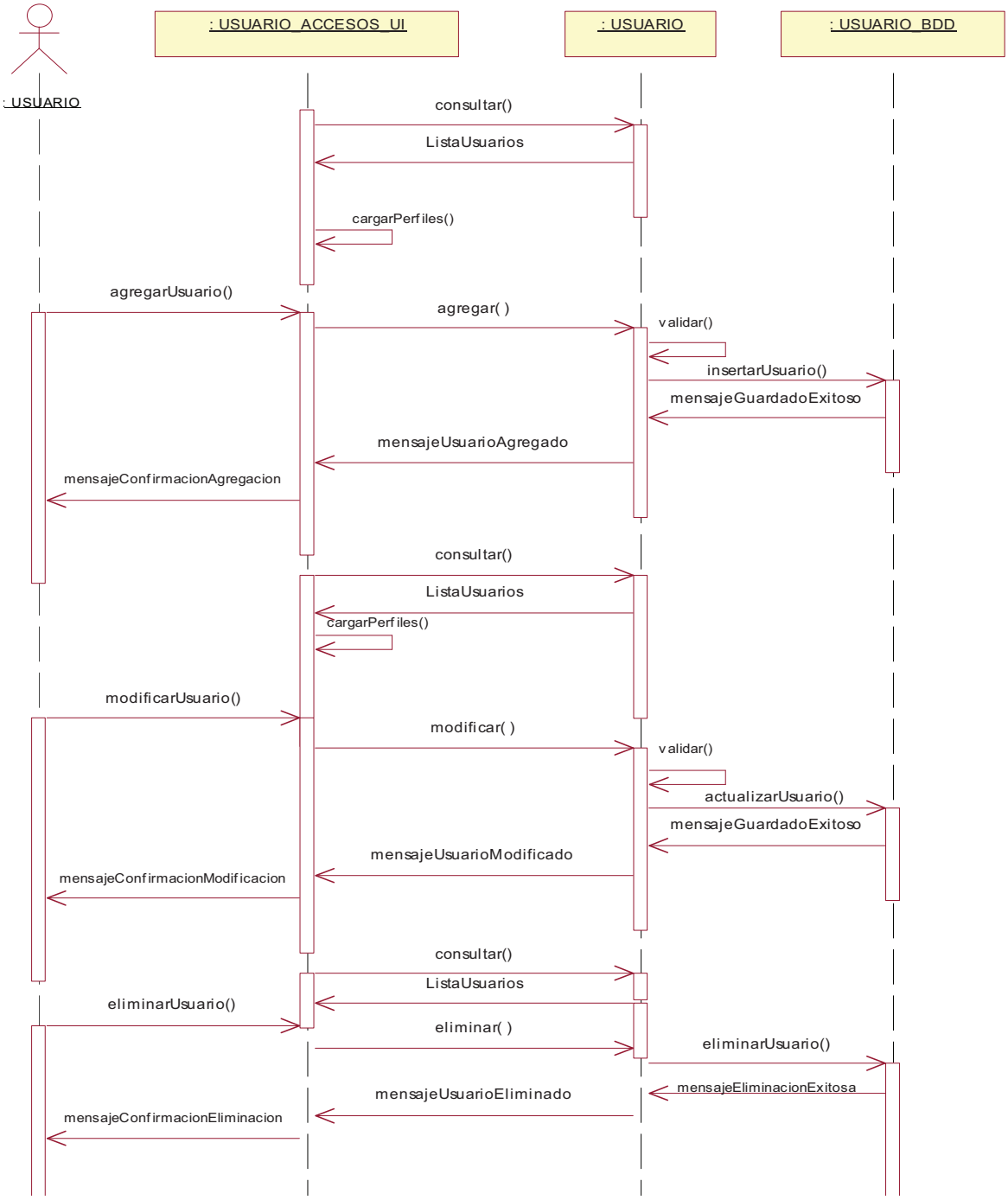


FIGURA 4: DIAGRAMA DE COLABORACIÓN- ADMINISTRAR SEGURIDADES
ELABORADO POR: EL AUTOR

CU-03: Administrar Definiciones

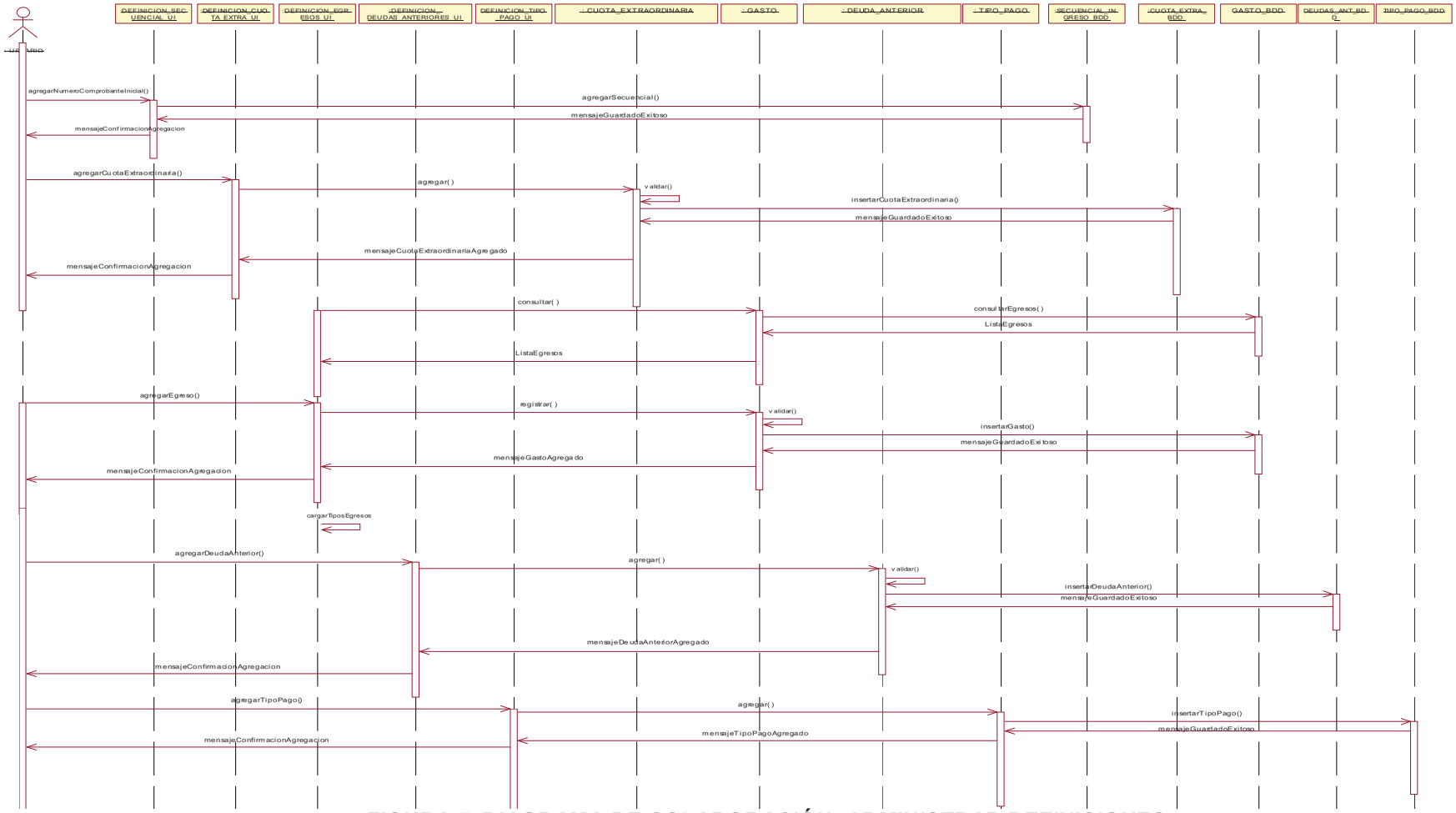


FIGURA 5: DIAGRAMA DE COLABORACIÓN- ADMINISTRAR DEFINICIONES
ELABORADO POR: EL AUTOR

CU-04: Administrar Locales

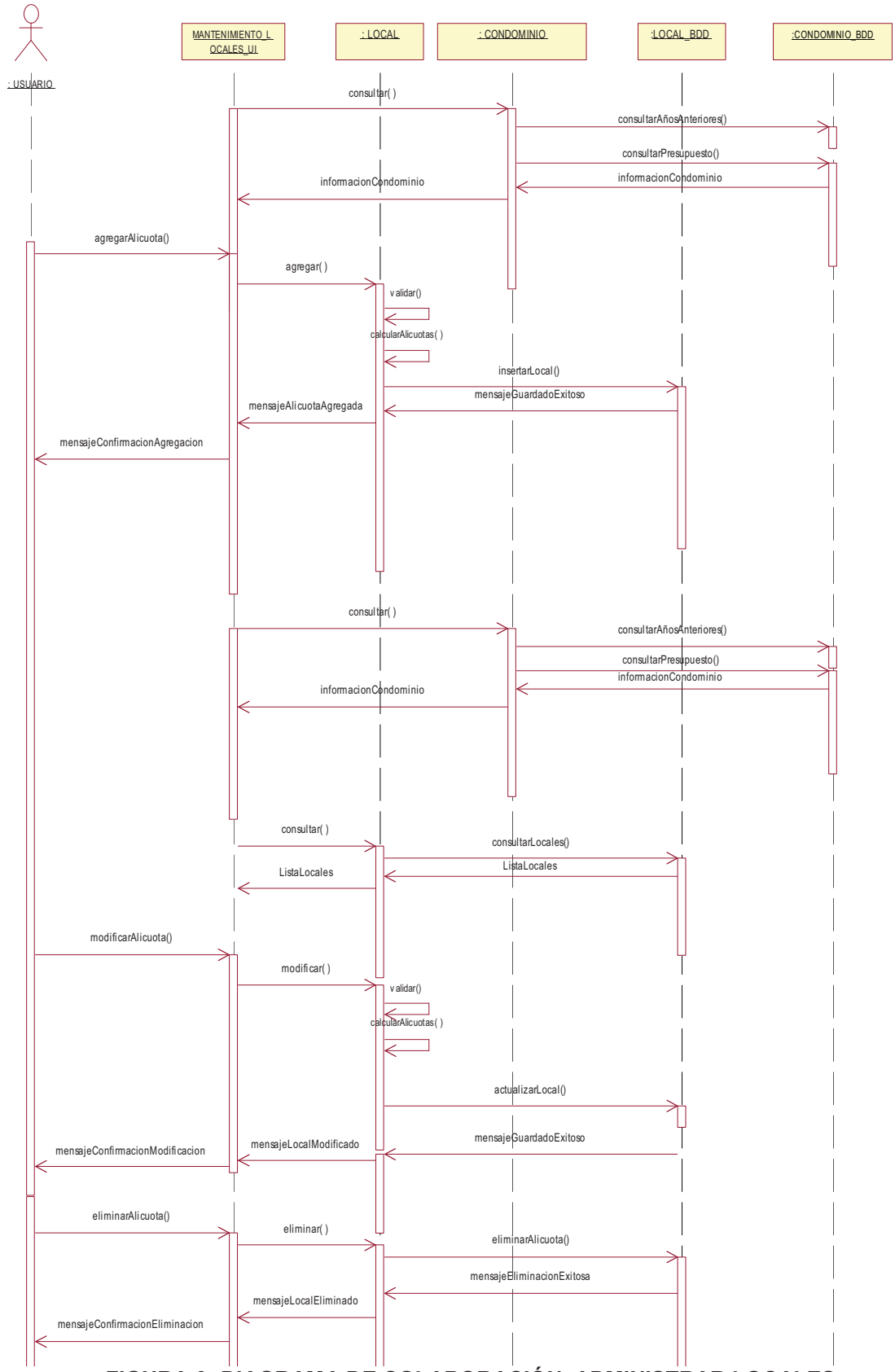


FIGURA 6: DIAGRAMA DE COLABORACIÓN- ADMINISTRAR LOCALES
ELABORADO POR: EL AUTOR

CU-05: Administrar Comprobantes

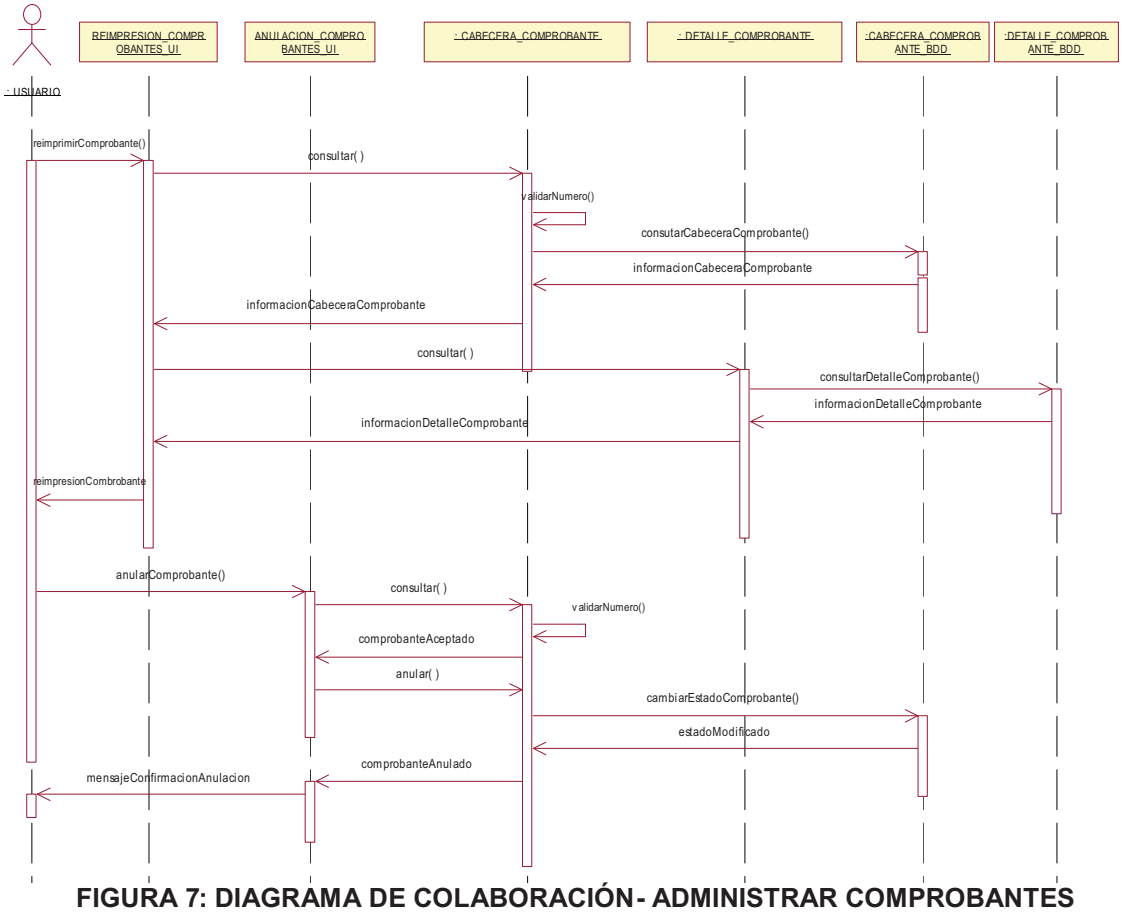


FIGURA 7: DIAGRAMA DE COLABORACIÓN- ADMINISTRAR COMPROBANTES
ELABORADO POR: EL AUTOR

CU-06: Generar Cierre de Año

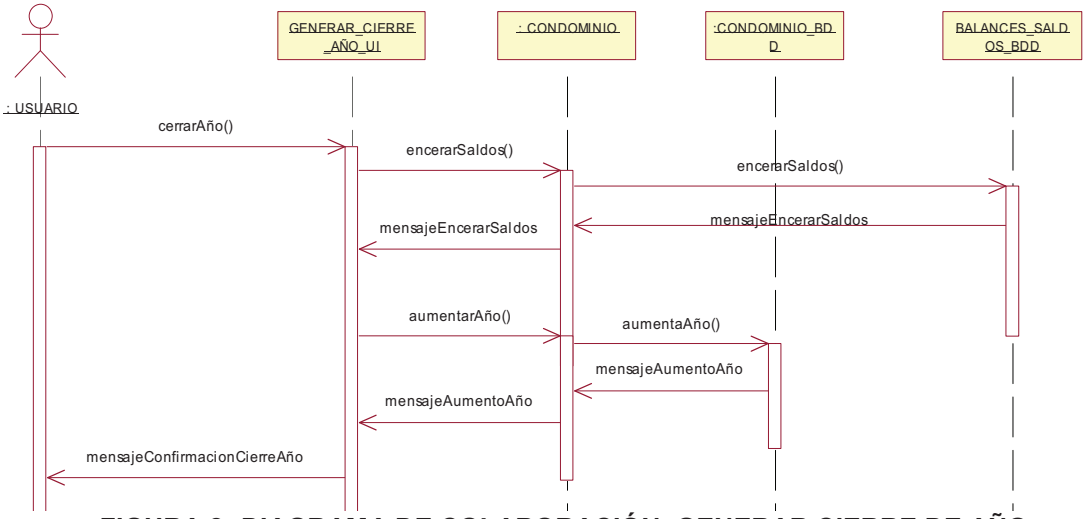


FIGURA 8: DIAGRAMA DE COLABORACIÓN- GENERAR CIERRE DE AÑO
ELABORADO POR: EL AUTOR

CU-07: Cobrar Alícuotas

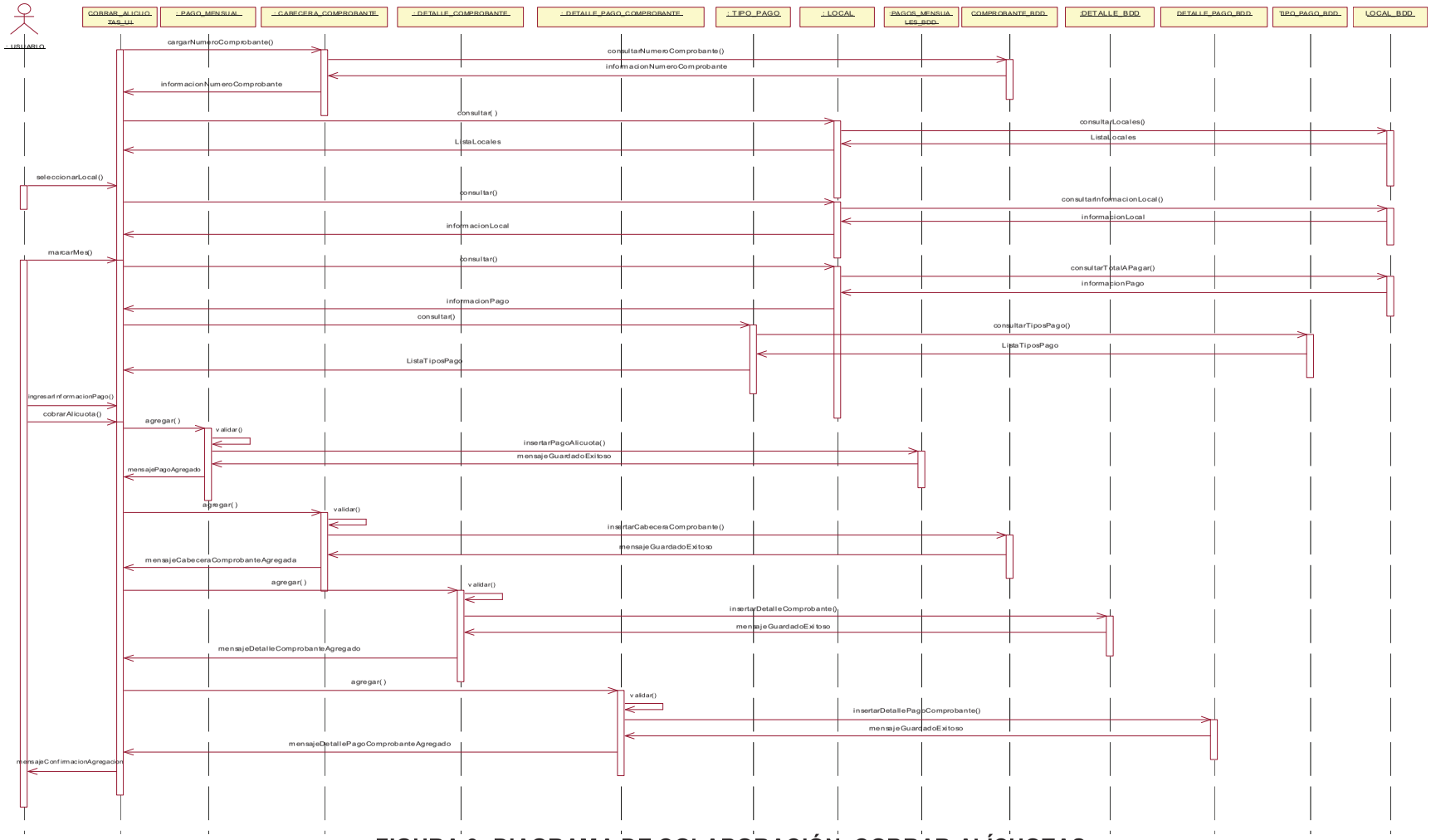


FIGURA 9: DIAGRAMA DE COLABORACIÓN- COBRAR ALÍCUOTAS
ELABORADO POR: EL AUTOR

CU-08: Cobrar Cuotas Extraordinarias

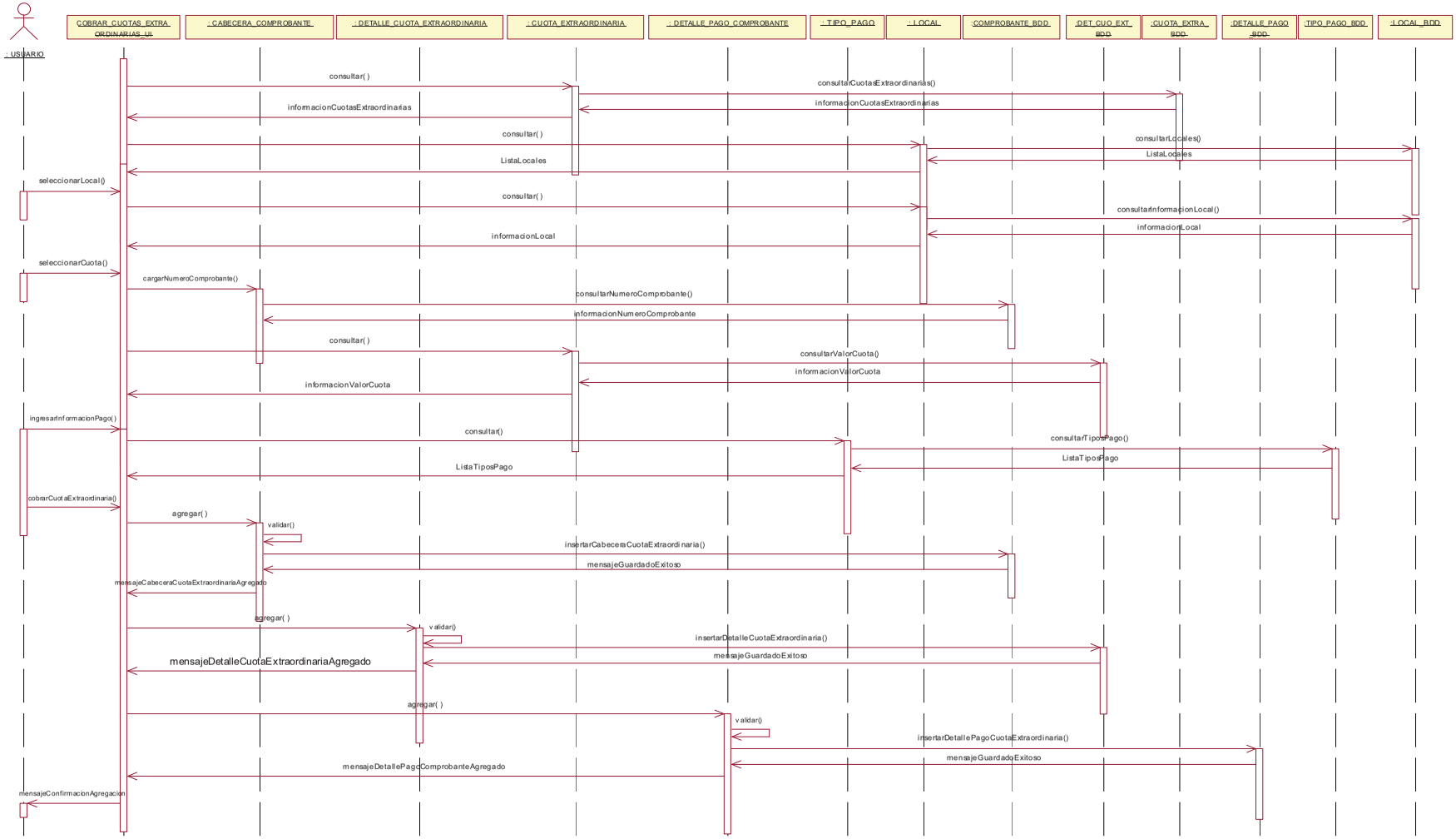


FIGURA 10: DIAGRAMA DE COLABORACIÓN - COBRAR CUOTAS EXTRAORDINARIAS
ELABORADO POR: EL AUTOR

CU-09: Cobrar Deudas Anteriores

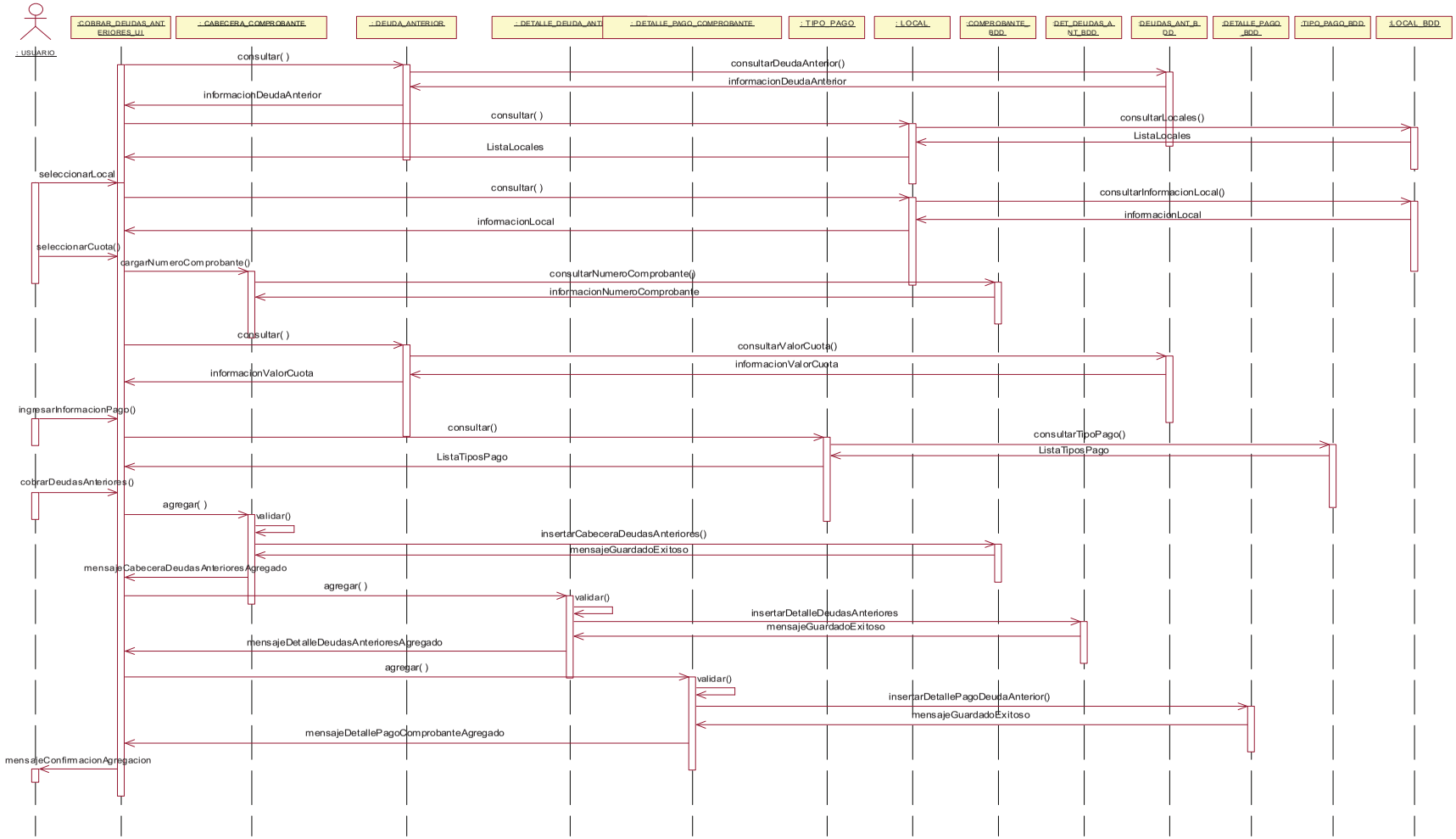


FIGURA 11: DIAGRAMA DE COLABORACIÓN- COBRAR DEUDAS ANTERIORES
ELABORADO POR: EL AUTOR

CU-10: Registrar Egresos

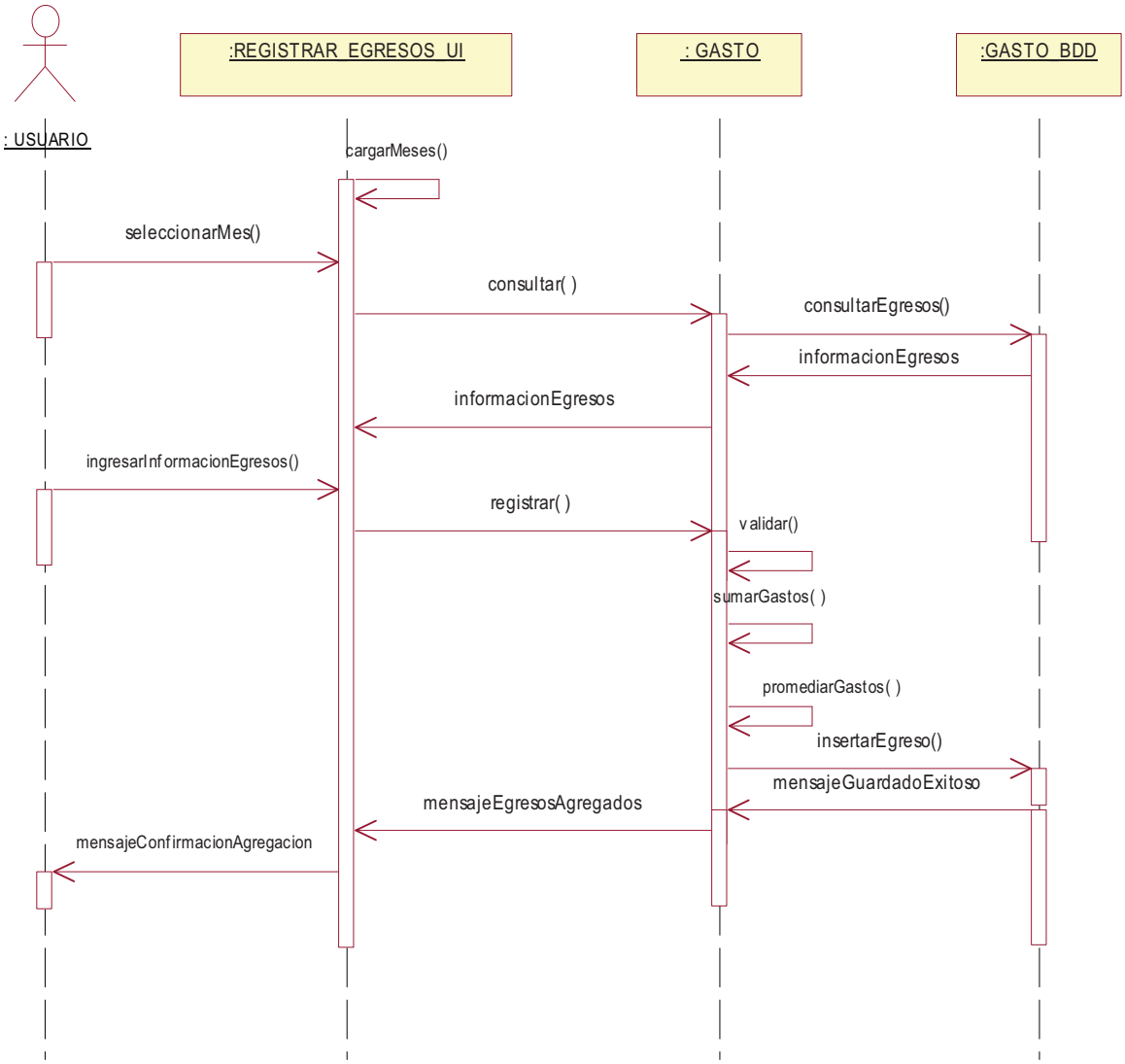


FIGURA 12: DIAGRAMA DE COLABORACIÓN- REGISTRAR EGRESOS
ELABORADO POR: EL AUTOR

2.1.2 DIAGRAMAS DE SECUENCIA

Utilizando los diagramas de secuencia que se muestra a continuación, se representa la interacción a través del tiempo de los objetos establecidos en el Sistema de Administración de Edificios.

CU-01: Administrar Condominios

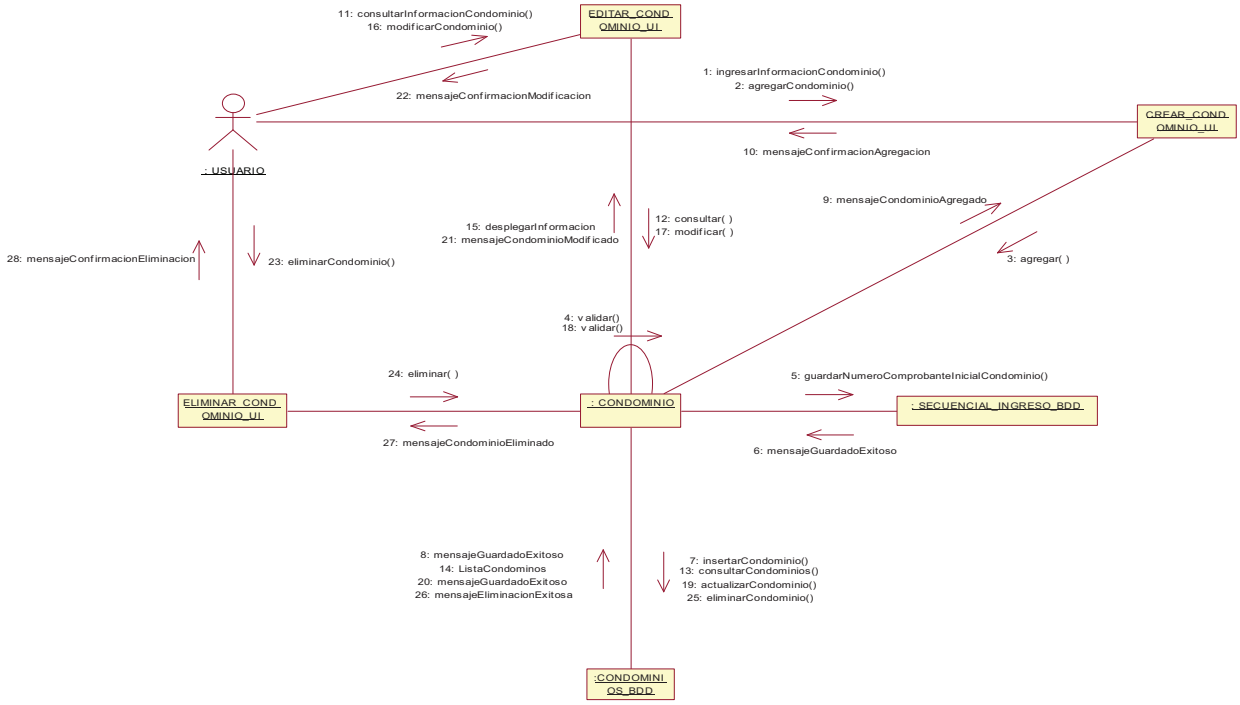


FIGURA 13: DIAGRAMA DE SECUENCIA- ADMINISTRAR CONDOMINIOS
ELABORADO POR: EL AUTOR

CU-02: Administrar Seguridades

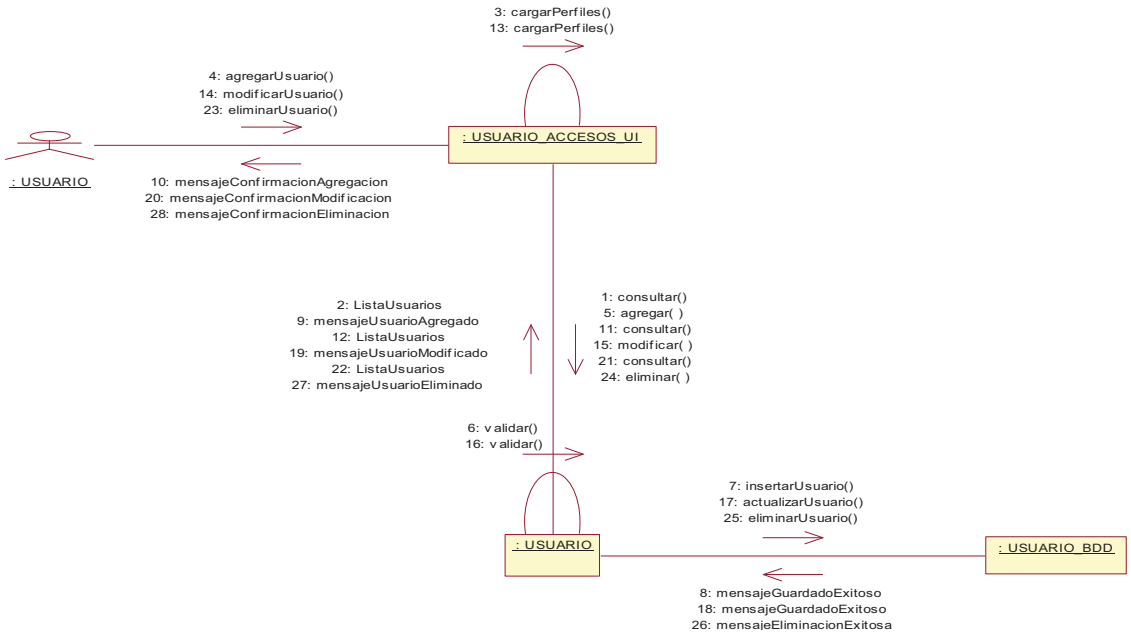


FIGURA 14: DIAGRAMA DE SECUENCIA- ADMINISTRAR SEGURIDADES
ELABORADO POR: EL AUTOR

CU-03: Administrar Definiciones

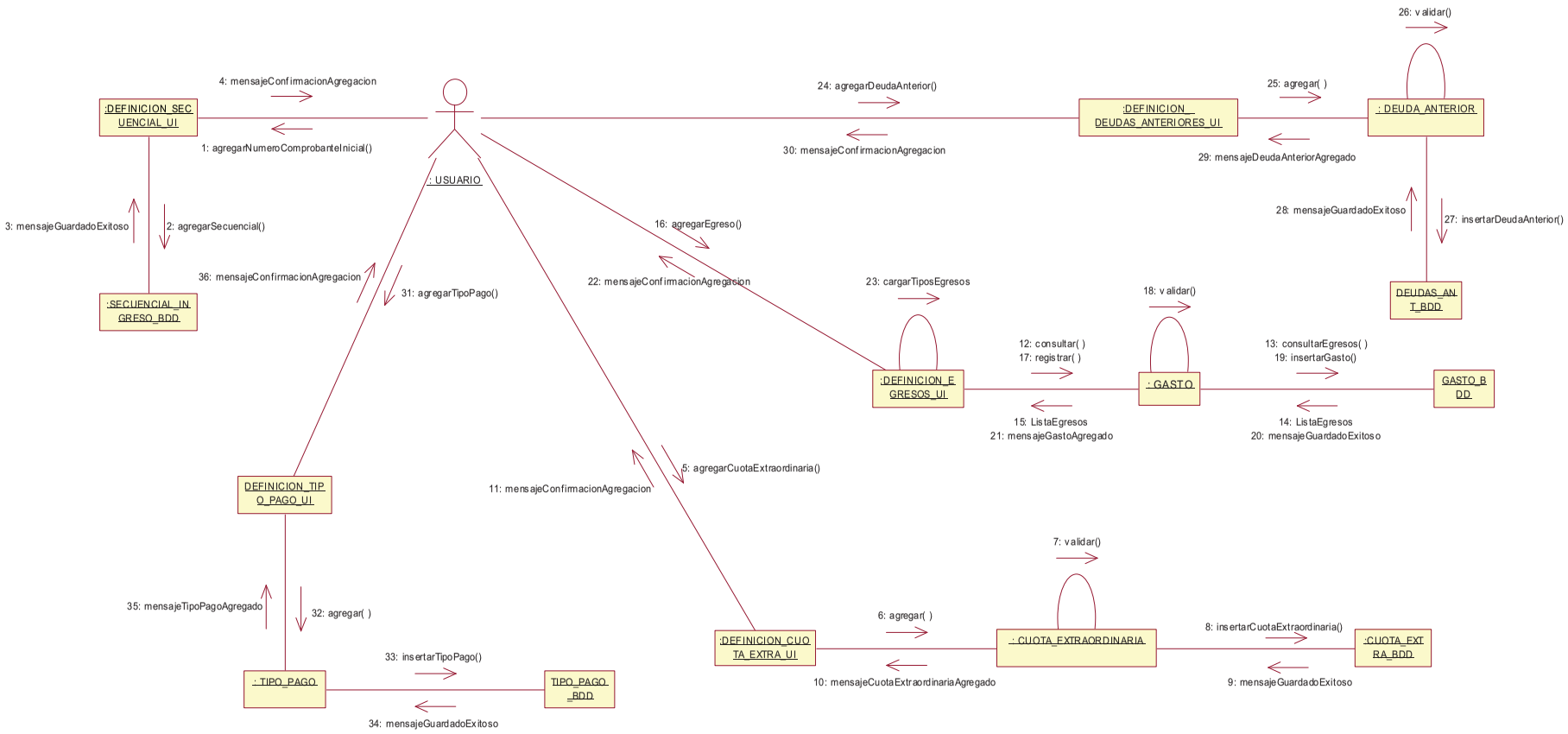


FIGURA 15: DIAGRAMA DE SECUENCIA- ADMINISTRAR DEFINICIONES
ELABORADO POR: EL AUTOR

CU-04: Administrar Locales

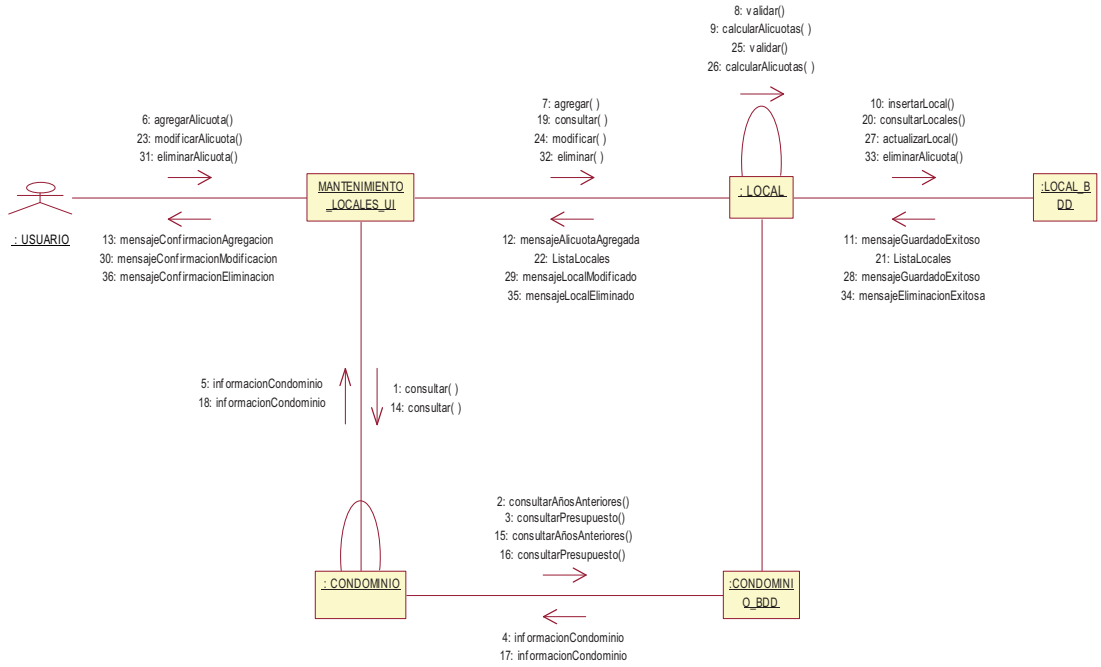


FIGURA 16: DIAGRAMA DE SECUENCIA- ADMINISTRAR LOCALES
ELABORADO POR: EL AUTOR

CU-05: Administrar Comprobantes

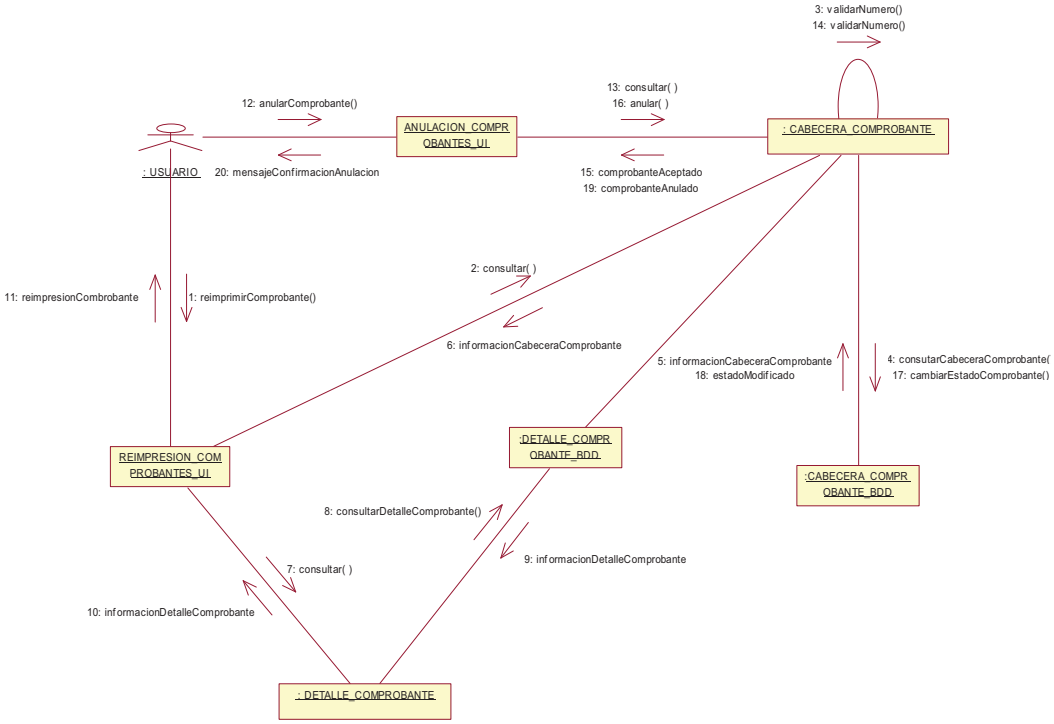


FIGURA 17: DIAGRAMA DE SECUENCIA- ADMINISTRAR COMPROBANTES
ELABORADO POR: EL AUTOR

CU-06: Generar Cierre de Año

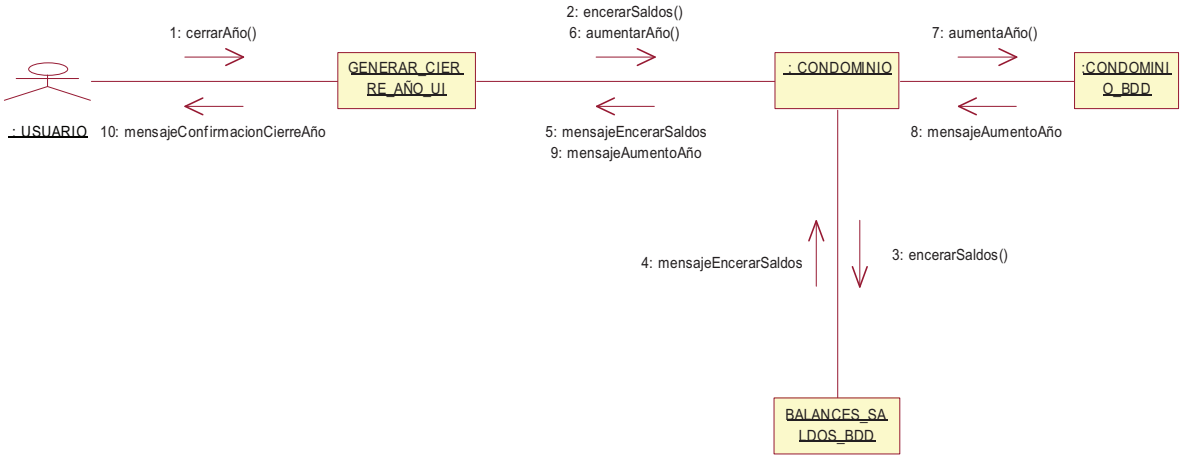


FIGURA 18: DIAGRAMA DE SECUENCIA- GENERAR CIERRE DE AÑO
ELABORADO POR: EL AUTOR

CU-07: Cobrar Alícuotas

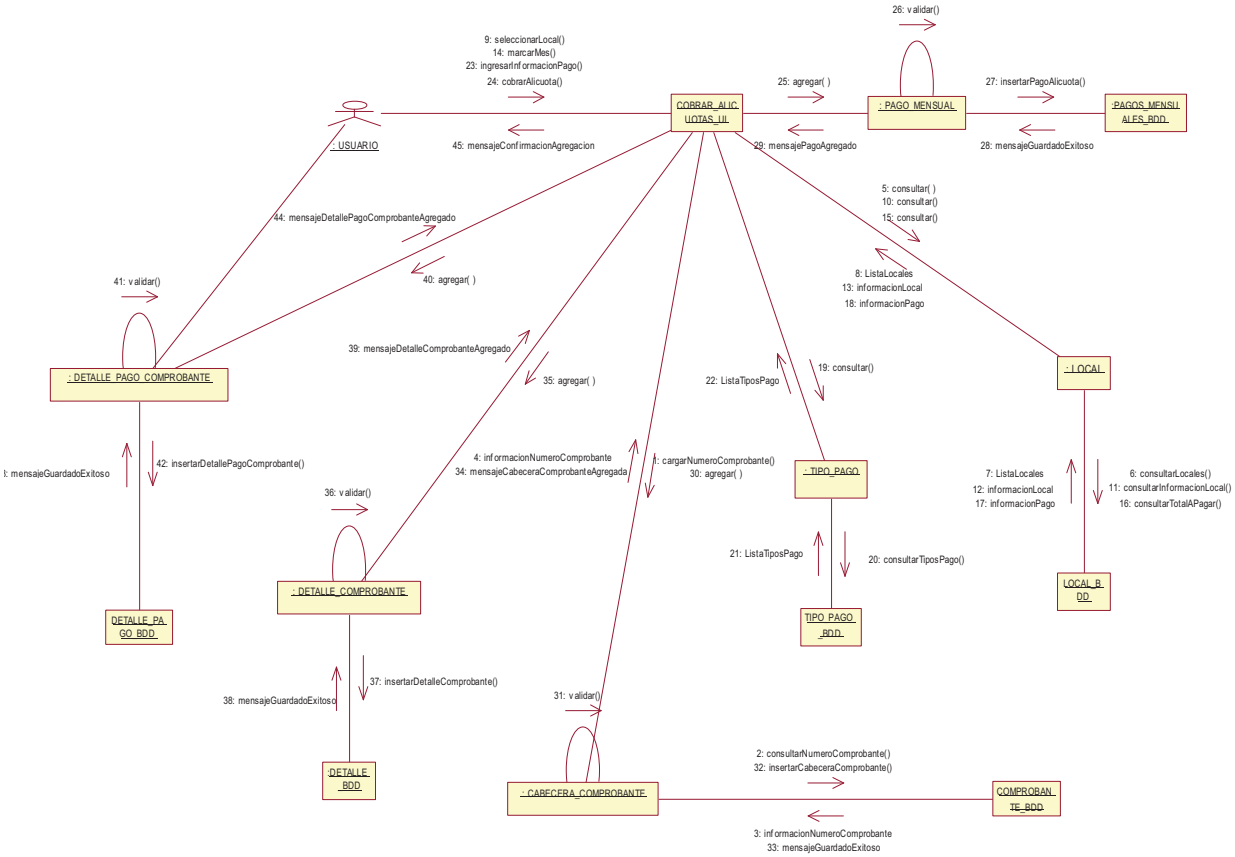


FIGURA 19: DIAGRAMA DE SECUENCIA- COBRAR ALÍCUOTAS
ELABORADO POR: EL AUTOR

CU-08: Cobrar Cuotas Extraordinarias

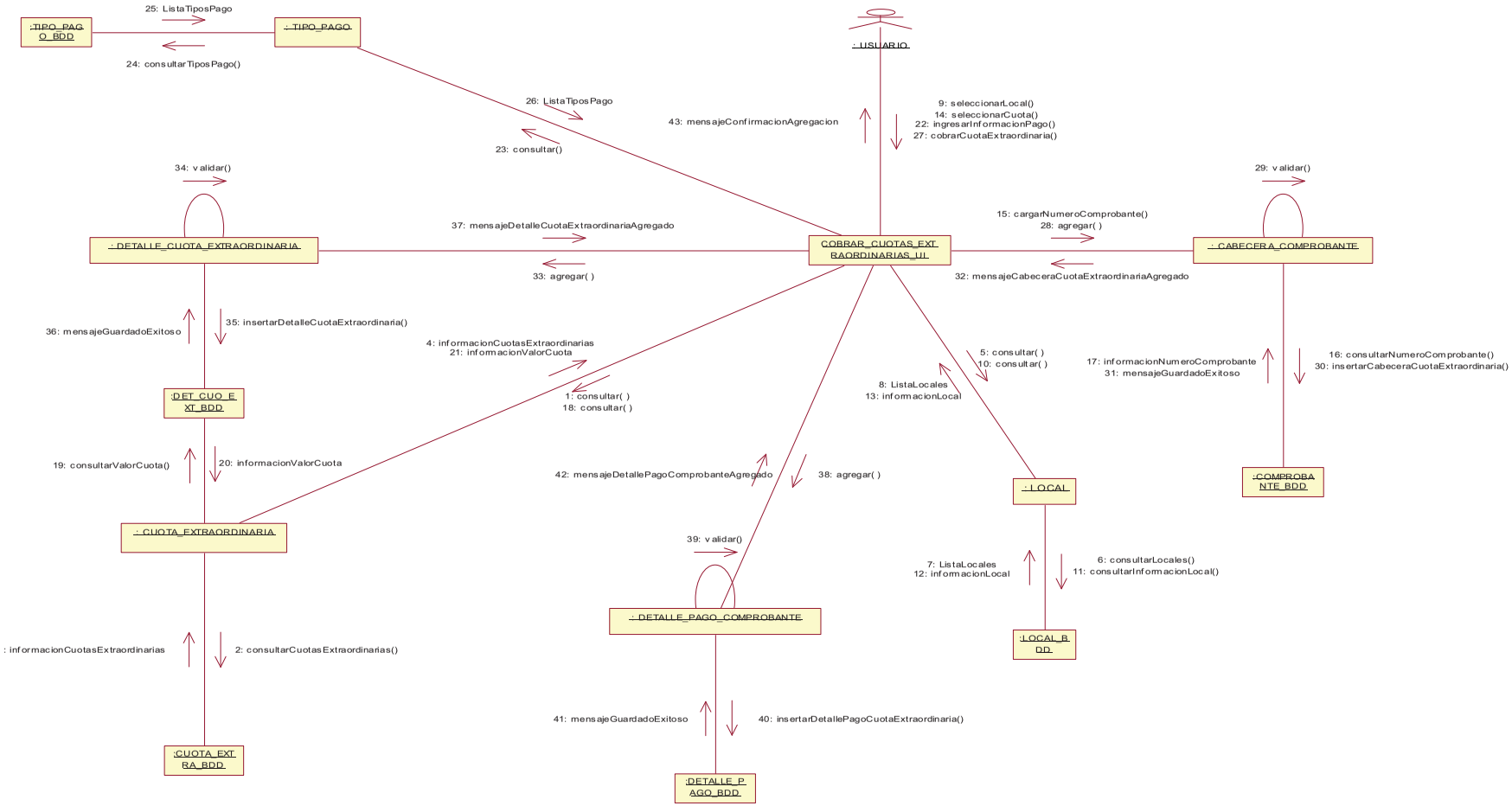


FIGURA 20: DIAGRAMA DE SECUENCIA- COBRAR CUOTAS EXTRAORDINARIAS
ELABORADO POR: EL AUTOR

CU-09: Cobrar Deudas Anteriores

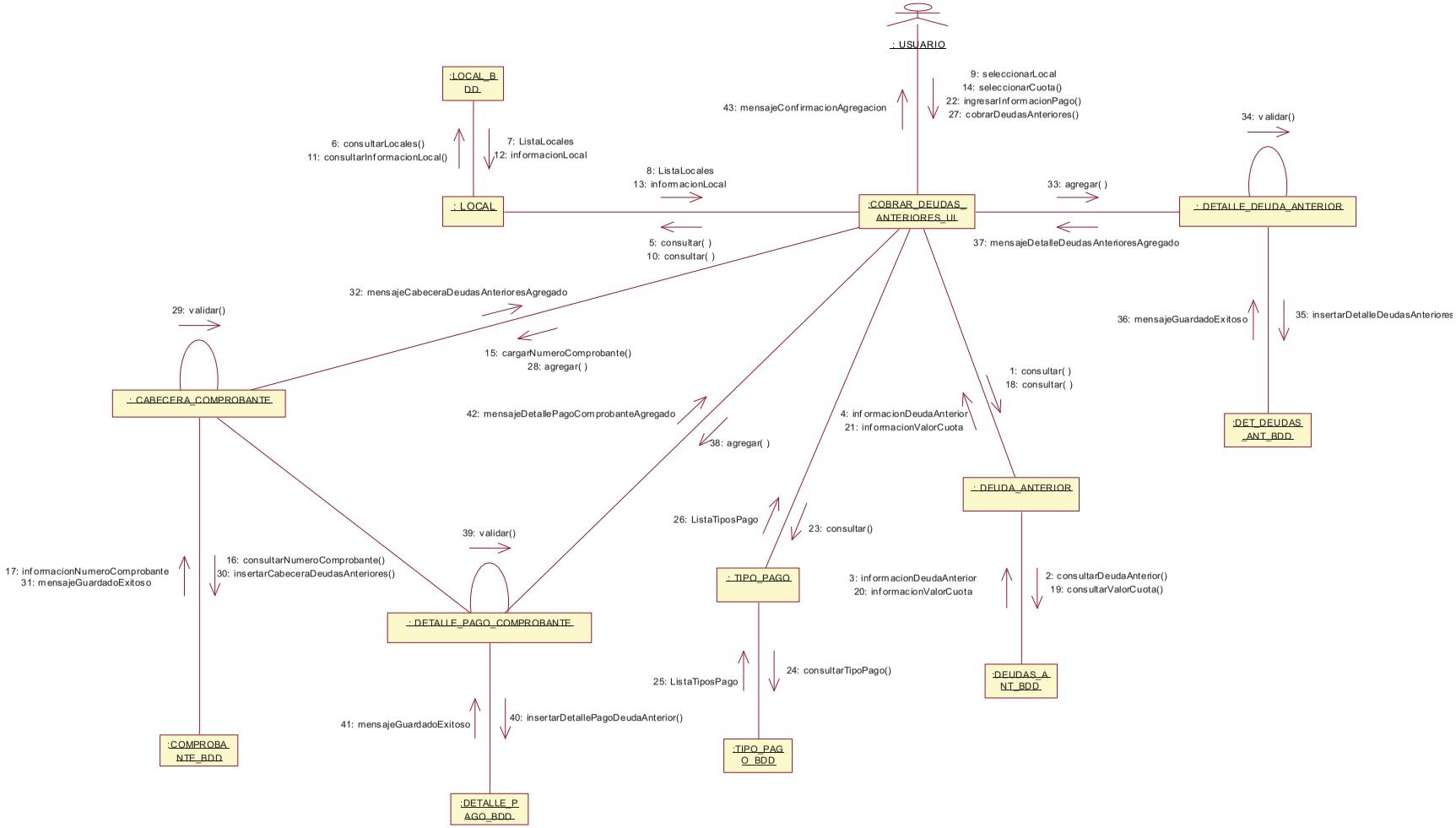


FIGURA 21: DIAGRAMA DE SECUENCIA- COBRAR DEUDAS ANTERIORES
ELABORADO POR: EL AUTOR

CU-10: Registrar Egresos

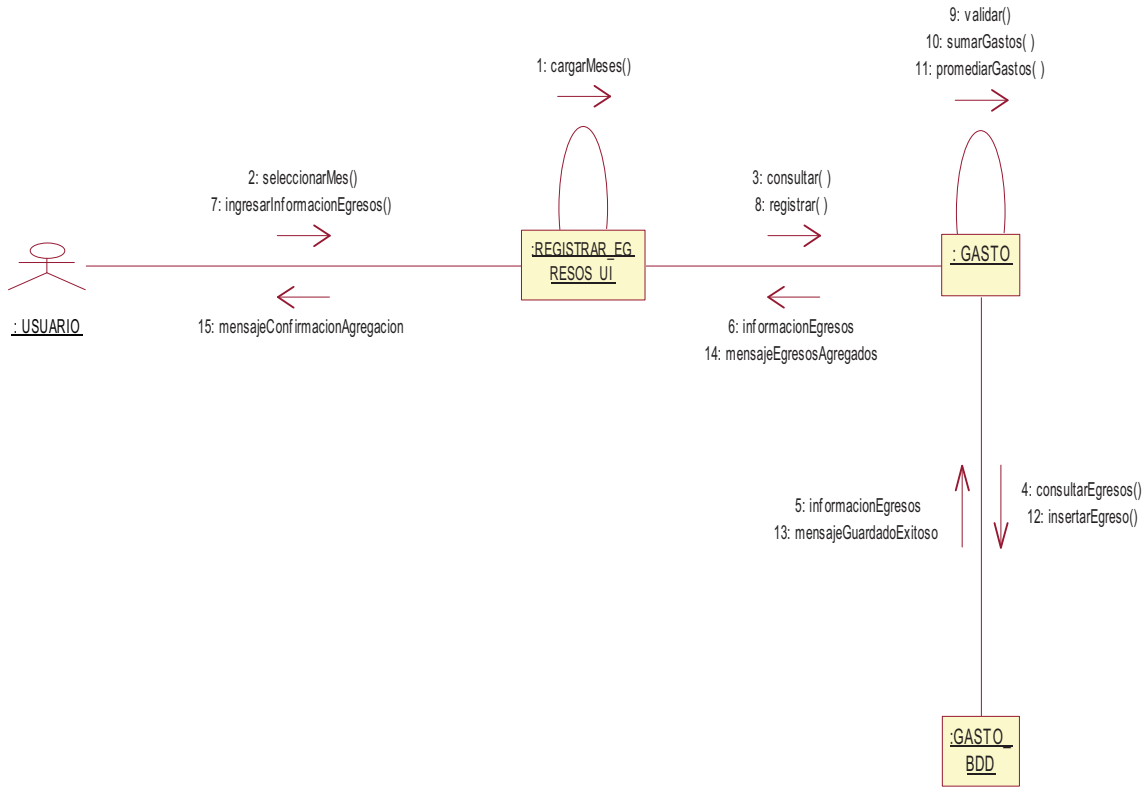


FIGURA 22: DIAGRAMA DE SECUENCIA- REGISTRAR EGRESOS
ELABORADO POR: EL AUTOR

3. PRUEBAS

Para realizar una correcta evaluación del desempeño, consistencia, caracteristicas y confiabilidad del Sistema de Administración de Edificios se efectuó dos tipos de pruebas: funcionales y de usuario.

3.1 PRUEBAS FUNCIONALES

Para realizar las pruebas funcionales del sistema se utilizó como base los casos de uso previamente definidos durante el diseño de la aplicación, de este modo se logra una efectiva retroalimentación de las funcionalidades. Este tipo de pruebas evalúa los datos de ingreso, las salidas esperadas y las salidas obtenidas.

CU-01: Administrar Condominios

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 10:30			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Administrar Condominios			
Descripción: Crear, editar, eliminar condominios.			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Ingresar Condominio	• Nombre • Ruc • Dirección • Teléfono • Fax • Repre. Legal • Administrador • Saldo año anterior • Presup.. Año Actual • Presup.. Alo Anterior • Presup. 2 Años Atras	**Salida 1:** Si se registra todos los datos ingresados y el sistema debe permitir el ingreso de los mismos en la tabla CONDOMINIOS. **Salida 2:** Si no se ingreso la información obligatoria: nombre del condominio, saldo año anterior y presupuesto del año actual el sistema debe notificar que debe ingresarse todos los campos.	**Salida 1:** El sistema registró todos los datos del condominio y bloquea los campos ingresados. **Salida 2:** El sistema desplegó mensajes indicando los campos obligatorios que no fueron ingresados.
Escenario 2. Editar Condominio	• Teléfono • Administrador	**Salida 1:** Si se modifica el teléfono y el nombre del administrador, el sistema debe permitir actualizar el registro del condominio con los nuevos datos	**Salida 1:** El sistema actualizó los campos de teléfono y administrador.
Escenario 3. Eliminar Condominio		**Salida 1:** Si se desea eliminar un registro el sistema lanza un mensaje de confirmación. Luego de seleccionar "si" se debe eliminar el registro.	**Salida 1.** El sistema eliminó el registro del condominio.

CU-02: Administrar Seguridades

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 10:45			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Administrar Seguridades			
Descripción: Crear, editar, eliminar usuarios			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Crear Usuario	• Usuario • Login • Password • Perfil	**Salida 1:** Si se registra todos los datos ingresados y el sistema debe permitir el ingreso de los mismos en la tabla USUARIOS. **Salida 2:** Si no se ingreso la información obligatoria: usuario y login; el sistema debe notificar que debe registrarse dichos campos.	**Salida 1:** El sistema registró todos los datos del usuario y bloquea los campos ingresados. **Salida 2:** El sistema desplegó mensajes indicando los campos obligatorios que no fueron ingresados.
Escenario 2. Editar Usuario	• Password • Perfil	**Salida 1:** Si se modifica el password y perfil, el sistema debe permitir actualizar el registro del usuario con los nuevos datos	**Salida 1:** El sistema actualizó los campos de password y perfil.
Escenario 3. Eliminar Usuario		**Salida 1:** Si se desea eliminar un registro el sistema lanza un mensaje de confirmación. Luego de seleccionar "si" se debe eliminar el registro.	**Salida 1.** El sistema eliminó el registro del usuario.

CU-03: Administrar Definiciones

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 11:00			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Administrar Definiciones			
Descripción: Configurar las opciones del sistema a fin de establecer el modo de trabajo.			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Definir Secuencial	• Numero Inicial.	**Salida 1:** Se registra el número inicial y el sistema guarda la información en la tabla SECUENCIAL INGRESO. **Salida 2:** Si no se ingreso el número inicial el sistema debe notificar que no existen datos mediante un mensaje.	**Salida 1:** El sistema registró todos los datos del secuencial y bloquea el campo ingresado. **Salida 2:** El sistema desplegó el mensaje: "Ingrese el numero de secuencial".
Escenario 2. Definir Cuota Extra	• Descripción • Razón • Valor • Comprobantes Necesarios	**Salida 1:** Se registra todos los datos y el sistema guarda la información en la tabla CUOTAEXTRA. **Salida 2:** Si no se ingreso la informacion el sistema notifica que no existen datos mediante un mensaje.	**Salida 1:** El sistema registró todos los datos de la cuota extra y bloquea los campos ingresados. **Salida 2:** El sistema desplegó el mensaje: "Ingrese el <<dato>>".
Escenario 3. Definir Egresos.	• Beneficiario • Grupo/Tipo	**Salida 1:** Se registra todos los datos y el sistema guarda la información en la tabla EGRESOS **Salida 2:** Si no se ingreso la información obligatoria, el sistema despliega mensajes informaivos.	**Salida 1:** El sistema registró todos los datos del nuevo egreso y bloquea los campos ingresados. **Salida 2:** El sistema desplegó el mensaje: "Ingrese el <<dato>>".
Escenario 2. Definir Deudas Anteriores	• Local • Descripción • Año • Valor	**Salida 1:** Se seleccina toda la información requerida y se especifica el valor; el sistema guarda la información en la tabla DEUDASANT. **Salida 2:** Si no se ingreso la informacion el sistema informa que no existen datos mediante un mensaje.	**Salida 1:** El sistema registró información sobre la deuda anterior y bloquea los campos ingresados. **Salida 2:** El sistema desplegó el mensaje: "Ingrese el <<dato>>".
Escenario 3. Definir Tipo Pago.	• Descripción	**Salida 1:** Se registra la descripción de la forma de pago y el sistema guarda la información en la tabla TIPOPAGO **Salida 2:** Si no se ingreso la informacion el sistema notifica que no existen datos mediante un mensaje.	**Salida 1:** El sistema registró todos los datos del tipo de pago y bloquea el campo descripción. **Salida 2:** El sistema desplegó el mensaje: "Ingrese la nueva forma de pago"

CU-04: Administrar Locales

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 11:15			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Administrar Locales			
Descripción: Crear, editar y eliminar Locales			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Ingresar Local	• Local • Estacionamiento 1 • Estacionamiento 2 • Estacionamiento 3 • Bodega 1 • Bodega 2 • Bodega 3 • EEQ • Teléfono • Observación • Apellidos Propietario • Nombres Propietario • CI Propietario • Teléfono Propietario • Mail Propietario • Apellidos Arrendatario • Nombres Arrendatario • CI Arrendatario • Teléfono Arrendatario • Mail Arrendatario • Valor Arrendamiento • Fecha Arrendamiento • Garantia • Alícuota Local • Alícuota Terreno • Alícuota Estac. 1 • Alícuota Estac. 2 • Alícuota Estac. 3 • Alícuota Bodega 1 • Alícuota Bodega 2 • Alícuota Bodega 3 • Área Local • Área Terreno • Área Estac. 1,2,3 • Área Bodega 1,2,3 • Alícuota a Pagar	**Salida 1:** Si se registra todos los datos ingresados y el sistema debe permitir el ingreso de los mismos en la tabla LOCAL. **Salida 2:** Si no se ingreso la información obligatoria: local, Alícuota a Pagar; el sistema debe notificar que debe ingresarse todos los campos.	**Salida 1:** El sistema registró todos los datos del local y bloquea los campos ingresados. **Salida 2:** El sistema desplegó mensajes indicando los campos obligatorios que no fueron ingresados.

Escenario 2. Editar Local	• Alícuota a Pagar • Apellidos Propietario • Nombres Propietario • CI Propietario • Teléfono Propietario	**Salida 1:** Si se modifica los datios, el sistema debe permitir actualizar el registro del local con los nuevos datos	**Salida 1:** El sistema actualizó los campos editados.
Escenario 3. Eliminar Local		**Salida 1:** Si se desea eliminar un registro el sistema lanza un mensaje de confirmación. Luego de seleccionar "si" se debe eliminar el registro.	**Salida 1.** El sistema eliminó el registro del local.

CU-05: Administrar Comprobantes

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 11:30			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Administrar Comprobantes			
Descripción: Reimprimir y anular comprobantes			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Reimpresión de Comprobantes	• Número de Comprobante	**Salida 1:** Se registra un número de comprobante; el sistema debe realizar la consulta en la tabla COMPROBANTES y debe generar el comprobante. **Salida 2:** Se registra un número de comprobante inexistente; el sistema debe desplegar un mensaje informando que el comprobante no existe. **Salida 3:** Si no se ingreso el número de comprobante el sistema debe notificar que no existe información ingresada.	**Salida 1:** El sistema consultó el número y generó nuevamente el comprobante. **Salida 2:** El sistema desplegó el mensaje: “Comprobante no existe”. **Salida 3:** El sistema desplegó el mensaje: “Ingrese el número de Comprobante”.
Escenario 2. Anulación de Comprobantes	• Número de Comprobante	**Salida 1:** Se registra un número de comprobante; el sistema debe realizar la consulta en la tabla COMPROBANTES y cambiar el estado del comprobante a anulado. **Salida 2:** Se registra un número de comprobante inexistente; el sistema debe desplegar un mensaje informando que el comprobante no existe. **Salida 3:** Si no se ingreso el número de comprobante el sistema debe notificar que no existe información ingresada.	**Salida 1:** El sistema consultó el número y cambió el estado del comprobante a anulado. **Salida 2:** El sistema desplegó el mensaje: “Comprobante no existe”. **Salida 3:** El sistema desplegó el mensaje: “Ingrese el número de Comprobante”.

U-06: Generar Cierre de Año

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 11:45			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Generar Cierre de Año			
Descripción: Realizar acciones necesarias para cerrar el año contable.			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Cerrar Año		**Salida 1:** El sistema debe lanzar un mensaje de confirmación. Luego de seleccionar "si" se debe realizar las acciones necesarias para cerrar el año.	**Salida 1:** El sistema realizó todas las acciones y desplegó un mensaje.

CU-07: Cobrar Alicuotas

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 12:00			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Cobrar Alicuotas			
Descripción: Registrar el pago de alícuotas por local			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Cobrar Alícuota	• Local • Mes a pagar • Anotaciones • Pagado por • Cobrado Por • Fecha • Tipo Pago • Observación del Pago • Valor	**Salida 1:** Se registran los datos y el sistema debe guardar la información del cobro de la alícuota en las tablas COMPROBANTE y DETALLE_PAGO **Salida 2:** Si no se ingreso la información obligatoria: local y Tipo de Pago; el sistema debe notificar que debe ingresarse todos los campos. **Salida 3:** Si el valor que debe pagar no coincide con el total pagado, el sistema debe desplegar un mensaje informando el error.	**Salida 1:** El sistema registró todos los datos del cobro de la alícuota y bloquea los campos ingresados. **Salida 2:** El sistema desplegó el mensaje: “Hubo un error al guardar los datos”. **Salida 3:** El sistema desplegó el mensaje: “Los valores no coinciden”.

CU-08: Cobrar Cuotas Extraordinarias

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 12:15			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Cobrar Cuotas Extraordinarias			
Descripción: Registrar el pago de cuotas extraordinarias			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Cobrar Cuotas Extraordinarias	• Local • Cuota Extraordinaria • Fecha • Pagado por • Cobrado Por • Tipo Pago • Observación de Pago • Valor • Anotaciones	**Salida 1:** Se registran los datos y el sistema debe guardar la información del cobro de la alícuota en las tablas COMPROBANTE y DETALLE CUOTAEXTRA **Salida 2:** Si no se ingreso la información obligatoria: local y Tipo de Pago; el sistema debe notificar que debe ingresarse todos los campos. **Salida 3:** Si el valor que debe pagar no coincide con el total pagado, el sistema debe desplegar un mensaje informando el error.	**Salida 1:** El sistema registró todos los datos del cobro de la alícuota y bloquea los campos ingresados. **Salida 2:** El sistema desplegó el mensaje: "Hubo un error al guardar los datos". **Salida 3:** El sistema desplegó el mensaje: "Los valores no coinciden".

CU-09: Cobrar Deudas Anteriores

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 12:30			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Cobrar Deudas Anteriores.			
Descripción: Registrar el pago de deudas anteriores			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Cobrar Deudas Anteriores	• Local • Deuda Anterior • Anotaciones • Pagado por • Cobrado Por • Fecha • Tipo Pago • Observación de Pago • Valor	**Salida 1:** Se registran los datos y el sistema debe guardar la información del cobro de la alícuota en las tablas COMPROBANTE y DETALLE DEUDASANT **Salida 2:** Si no se ingreso la información obligatoria: local y Tipo de Pago; el sistema debe notificar que debe ingresarse todos los campos. **Salida 3:** Si el valor que debe pagar no coincide con el total pagado, el sistema debe desplegar un mensaje informando el error.	**Salida 1:** El sistema registró todos los datos del cobro de la alícuota y bloquea los campos ingresados. **Salida 2:** El sistema desplegó el mensaje: "Hubo un error al guardar los datos". **Salida 3:** El sistema desplegó el mensaje: "Los valores no coinciden".

CU-10: Registrar Egresos

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 12:45			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Registrar Egresos			
Descripción: Ingresar información y valores de egresos del condominio			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Registrar Egresos	• Mes • Fecha • Número de Factura • Período • Valor • Observación	**Salida 1:** Se registra los datos para cada egreso y el sistema guarda la información en la tabla EGRESOS. **Salida 2:** Si no se el campo obligartorio fecha en cada uno de los egresos, el sistema debe notificar que la información no está completa.	**Salida 1:** El sistema registró todos los datos de los egresos. **Salida 2:** El sistema desplegó el mensaje: "No se guardó la información, revise los campos ingresados"

CU-11: Generar Reportes

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 13:00			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Generar Reportes			
Descripción: Generar reportes según la información requerida			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Generar Reporte		**Salida 1:** El sistema debe desplegar el reporte seleccionado por el usuario.	**Salida 1:** El sistema generó el reporte con la información requerida.

CU-12: Generar Consultas

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 13:15			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Generar Consultas			
Descripción: Generar consultas según los parámetros ingresados			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Generar Consulta	• Según la consulta	**Salida 1:** El sistema debe desplegar la consulta seleccionada según el parámetro ingresado. **Salida 2:** Si no se ingresa el parámetro, el sistema debe informa mediante un mensaje que no se han ingresado los datos necesarios para generar la consulta.	**Salida 1:** El sistema generó la consulta con la información requerida. **Salida 2:** El sistema desplegó un mensaje informando al usuario que no existen parámetros necesarios para generar la consulta.

CU-13: Generar Balances

Pruebas de Sistema			
Sistema de Administración de Edificios			
Fecha y Hora de Realización: 23 de Julio de 2009 / 13:30			
Responsable: Wilson Araujo			
Caso de Prueba para Caso de Uso: Generar Balances			
Descripción: Generar balances según el período establecido por el usuario			
Escenario	**Datos de Ingreso**	**Salidas Esperadas**	**Salidas Obtenidas**
Escenario 1. Generar Balance		**Salida 1:** El sistema debe desplegar el balance seleccionado por el usuairo.	**Salida 1:** El sistema generó el balance con la información requerida.

3.1 PRUEBAS DE USUARIO

Las pruebas de usuario se realizaron durante una semana y fueron dirigidas por el desarrollador y efectuadas por el potencial administrador del sistema. La información obtenida durante las pruebas de usuario se resume en la siguiente tabla:

Pruebas de Sistema		
Sistema de Administración de Edificios		
Fecha y Hora de Realización: 24 de Julio / 28 de Julio de 2009 08:00 – 10:00		
Responsable: Wilson Araujo		
CARACTERÍSTICA	**NIVEL DE SATISFACCIÓN**	**OBSERVACIÓN**
Ajuste de Requerimientos	Muy Alto	Los requerimientos han sido implementados completamente
Adaptación de Usuarios	Alto	Debido a que existen varios perfiles, el usuario tiene acceso a opciones personalizadas según sus funciones
Calidad del Vocabulario Utilizado	Alto	Los usuarios comprendieron la información que transmite el sistema.
Claridad de Datos a introducir	Alto	Los usuarios comprendieron de manera clara donde y que datos deben introducir.
Claridad de Mensajes de error	Alto	Los usuarios consideran que los mensajes de error son claros y oportunos
Facilidad de Navegación	Muy Alto	La navegación a través de las diferentes opciones del sistema es sencilla.
Presentación Entorno	Alto	Los usuarios consideran que pueden conocer claramente lo que deben haber y las opciones que tienen a su alcance
Facilidad de Uso	Muy Alto	Los usuarios consideran que es un sistema fácil de utilizar
Nivel de Preparación previa para uso del sistema	Medio	Tomando en cuenta que los usuarios estan al tanto de la información manejada por el sistema, se consideró que no se necesita una elevada preparación para la utilización del mismo.
Facilidad de Localización de la información	Muy Alto	La información es fácil de localizar en el momento que el usuario la requiera.

Manual de Usuario

CONTENIDO

1. INTRODUCCIÓN
1.1 ¿Qué es el SAE?…….……..... 106
1.2 Ingreso al Sistema…….……106

2. PANTALLA PRINCIPAL
2.1 Descripción del Menú……107
2.2 Descripción de Accesos Directos……109

3. MENÚ ARCHIVO
3.1 Cerrar Sesión……. 110
3.2 Salir……..110

4. MENÚ LOCALES
4.1 Editar Locales…… 111
4.2 Imprimir Información..……113

5. MENÚ INGRESOS
5.1 Cobrar Alicuotas…… 114
5.2 Cobrar Cuota Extraordinaria…… 115
5.3 Cobrar Deuda Anterior……..116

6. MENÚ EGRESOS
6.1 Registrar Egreso……... 117

7. MENÚ DE REPORTES, CONSULTAS Y BALANCES
7.1 Reportes……..... 118
7.2 Consultas…… 119
7.3 Balances……..120

8. MENÚ ADMINISTRACIÓN
8.1 Condominio……..... 121
8.2 Definiciones…….....121
8.3 Comprobantes……..... 123
8.4 Cierre de Año……..... 124
8.5 Seguridades……..... 125

1. INTRODUCCIÓN

El presente documento describe el manejo del Sistema de Administración de Edificios, detalla las funcionalidades que éste ofrece a los usuarios y principalmente explica el correcto manejo que se dará al sistema a fin de obtener excelentes resultados en el registro de la información.

1.1 ¿QUÉ ES EL SAE?

El Sistema de Administración de Edificios (SAE) ofrece a los administradores una completa herramienta para el manejo automático de la información referente a los procesos de recaudación de ingresos y cancelacion de egresos. Además permite llevar un correcto control de los comprobantes generados, transparentando de esta manera el manejo de los valores, logrando así que su gestión sea más eficiente.

1.2 INGRESO AL SISTEMA

Para el ingreso al sistema, el usuario dispone de las siguientes opciones:

1) Mediante al acceso directo ubicado en el escritorio, creado al momento de la instalacion.

FIGURA 1: ICONO DE ACCESO AL SISTEMA
ELABORADO POR: EL AUTOR

2) Ingresando a traves de la siguiente ubicación:

Inicio >> Todos los programas >> SAE >> SAE

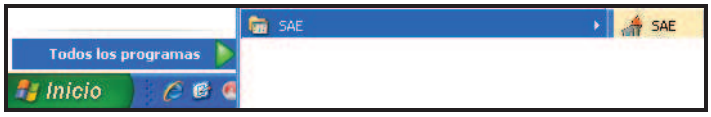


FIGURA 2: RUTA DE ACCESO AL SISTEMA
ELABORADO POR: EL AUTOR

Al ejecutar el programa, se desplegará la siguiente pantalla de bienvenida:

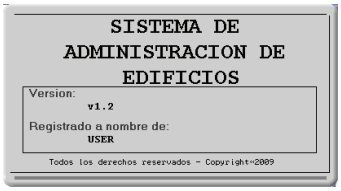


FIGURA 3: PANTALLA DE BIENVENIDA AL SISTEMA
ELABORADO POR: EL AUTOR

2. PANTALLA PRINCIPAL

Cada vez que el usuario ingresa al sistema, deberá identificarse en la siguiente pantalla para acceder a los módulos de la aplicación según su perfil:

FIGURA 4: LOGIN DEL SISTEMA
ELABORADO POR: EL AUTOR

Una vez ingresados los datos, hacer clic sobre el botón “Aceptar”.

El sistema despliega la pantalla principal, que contiene en la sección superior el menú de la aplicación y en la parte izquierda los accesos directos proporcionados al usuario para facilitar el acceso a las opciones mas utilizadas.

El usuario puede verificar el condominio en el que está trabajando en la barra de titulo que se encuentra en el extremo superior de la pantalla.

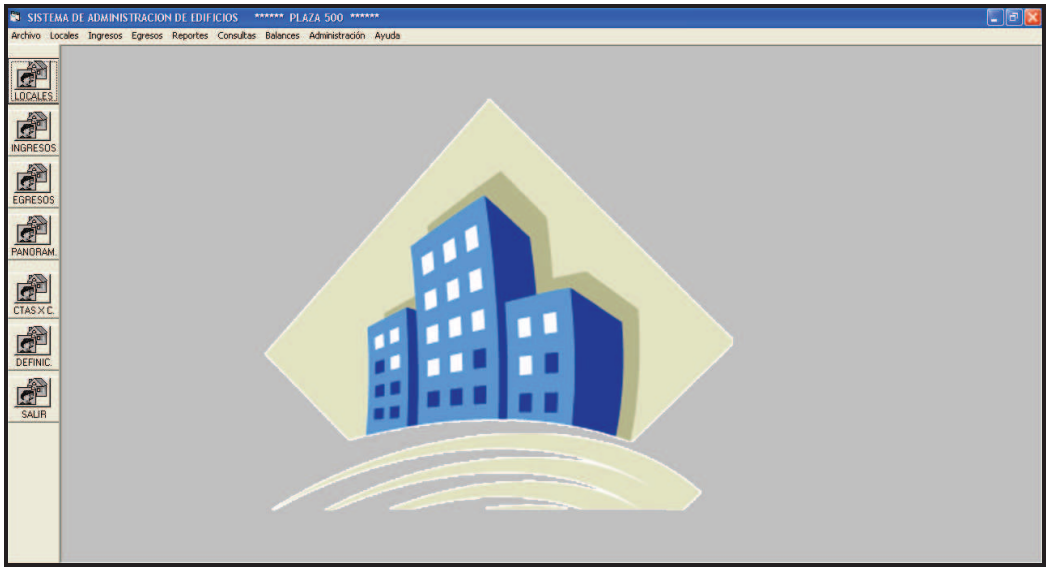


FIGURA 5: PANTALLA PRINCIPAL
ELABORADO POR: EL AUTOR

2.1 DESCRIPCIÓN DEL MENÚ

En la sección superior el sistema proporciona la barra de menú, misma que contiene las siguientes opciones:

MENU	OPCIONES	DESCRIPCIÓN
Archivo	Cerrar Sesión Salir	Proporciona las opciones de salida del sistema.
Locales	Editar Locales Imprimir Información	Permite administrar la información de locales registrados dentro del sistema.
Ingresos	Cobrar Alícuotas Cobrar Cuota Extraord. Cobrar Deuda Anterior	Permite seleccionar las opciones para el registro de ingresos.
Egresos	Registrar Egreso	Permite seleccionar las opciones para el registro de egresos por concepto de mantenimiento del condominio.
Reportes	Reporte Panoramico Reporte Ctas por Cob. Reporte de Morosos del Mes Reporte de Ingresos del Mes Reporte de Egresos del Mes Reporte de Alícuotas Prop. Reporte de Alícuotas Local	Proporciona los reportes disponibles para que el usuario disponga de la información resumida acerca las actividades realizadas.
Consultas	Estado de Cuenta Morosos por Mes Ingresos por Mes Egresos por Mes Comprobs. por Fechas	Proporciona la información requerida por el usuario en base a los parámetros ingresados.
Balances	Balance 1er. Semestre Balance 2do. Semestre	Proporciona al usuario un resumen de la actividad financiera semestral efectuada durante un semestre.
Administración	Condominio Definiciones Comprobantes Cierre de Año Seguridades	Permite al administrador del sistema definir las configuraciones necesarias para el correcto desempeño del sistema.
Ayuda	Acerca de	Proporciona la información de la versión del sistema.

2.2 DESCRIPCIÓN DE ACCESOS DIRECTOS

Para realizar las pruebas funcionales del sistema se utilizó como base los casos de uso previamente definidos durante el diseño de la aplicación, de este modo se logra una efectiva retroalimentación de las funcionalidades. Esta tipo de pruebas evalúa los datos de ingreso, las salidas esperadas y las salidas obtenidas.

ACCESO DIRECTO	DESCRIPCIÓN
	Permite administrar la información de locales registrados dentro del sistema.
	Proporciona al usuario las opciones necesarias para el registro de ingresos.
	Proporciona al usuario las opciones necesarias para el registro de egresos.
	Despliega el reporte con la información de las alícuotas cobradas, informando el numero de comprobante.
	Despliega el reporte con la información de las alícuotas cobradas, informando el numero de comprobante.
	Permite al administrador del sistema definir las configuraciones necesarias para un correcto desempeño.
	Permite al usuario salir correctamente de la aplicación.

3. MENÚ ARCHIVO

Proporciona al usuario las opciones necesarias para salir del sistema o cerrar la sesión iniciada e identificarse dentro de la aplicación con otro usuario, tambien permite seleccionar otro edificio en el cual se registrará información.

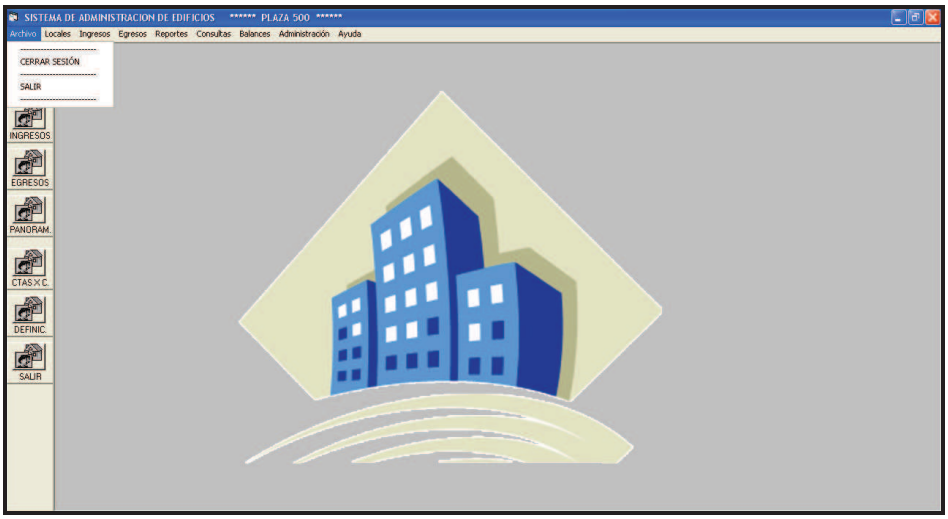


FIGURA 6: MENÚ ARCHIVO
ELABORADO POR: EL AUTOR

3.1 CERRAR SESIÓN

El usuario debe elegir esta opción cuando se requiera ingresar con el perfil de otro usuario o desee seleccionar otro edificio para desempeñar su trabajo.

FIGURA 7: CERRAR SESIÓN
ELABORADO POR: EL AUTOR

3.2 SALIR

Si el usuario selecciona la opcion salir, el sistema despliega un mensaje de confirmación, en el caso de responder afirmativamente la aplicación de cerrará, en el caso de contestar negativamente a la acción será cancelada y el usuario continuará trabajando.

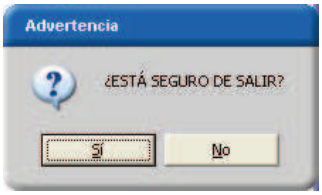


FIGURA 8: CONFIRMACIÓN DE SALIDA DEL SISTEMA
ELABORADO POR: EL AUTOR

4. MENÚ LOCALES

El módulo locales es indispensable para efectuar el registro de ingresos en el sistema, por lo cual la primera actividad que debe desarrollar el usuario es definir la información solicitada por la aplicación.

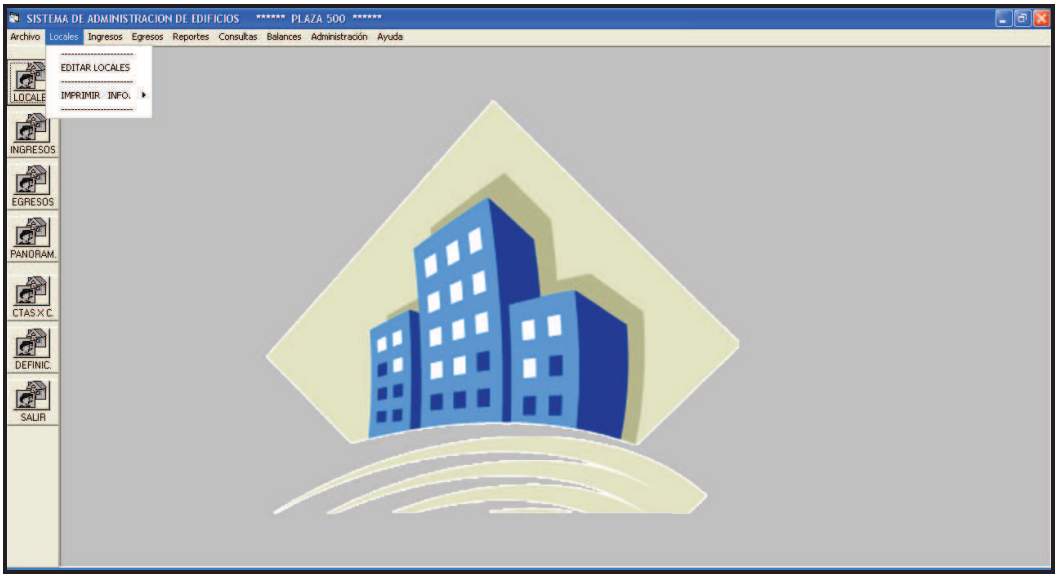


FIGURA 9: MENÚ LOCALES
ELABORADO POR: EL AUTOR

4.1 EDITAR LOCALES

Permite realizar las siguientes acciones:

Registrar un nuevo local.

Editar un local ingresado.

Eliminar un local.

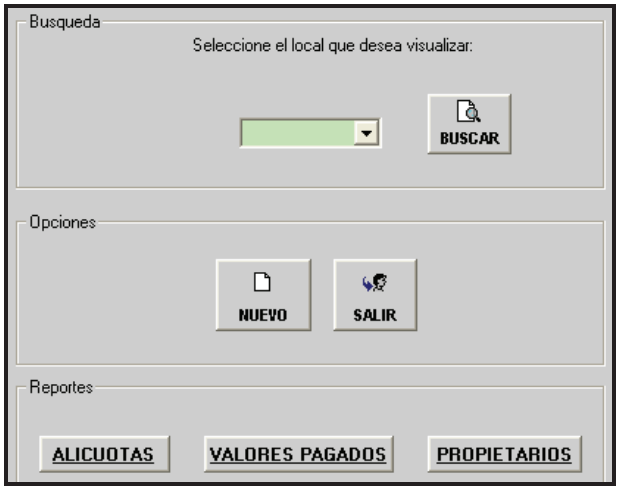


FIGURA 10: EDITAR LOCALES
ELABORADO POR: EL AUTOR

Para ingresar un local, el usuario debe pulsar el botón “Nuevo”, y llenar el formulario que se muestra a continuación:

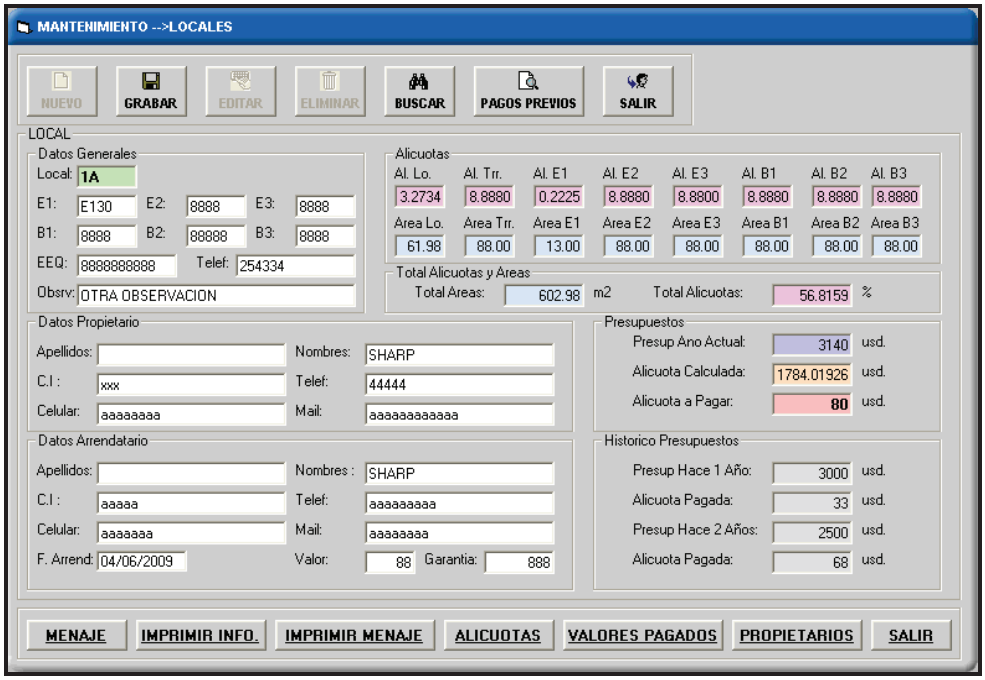


FIGURA 11: INFORMACIÓN DE LOCALES
ELABORADO POR: EL AUTOR

El usuario dispone de una seccion dentro del formulario para ingresar el menaje entregado en cada local, para esto deberá pulsar el botón menaje, y llenar la información solicitada.

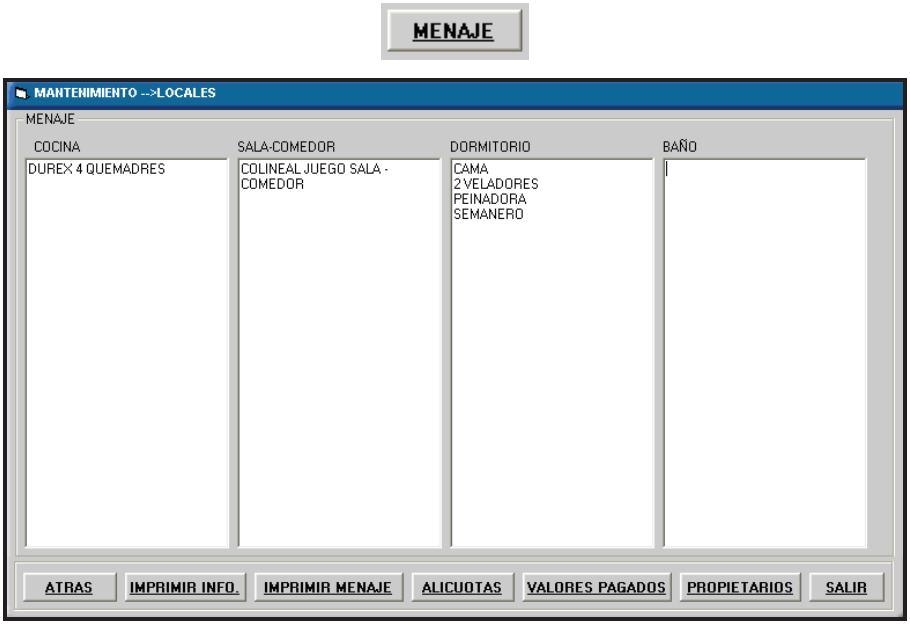
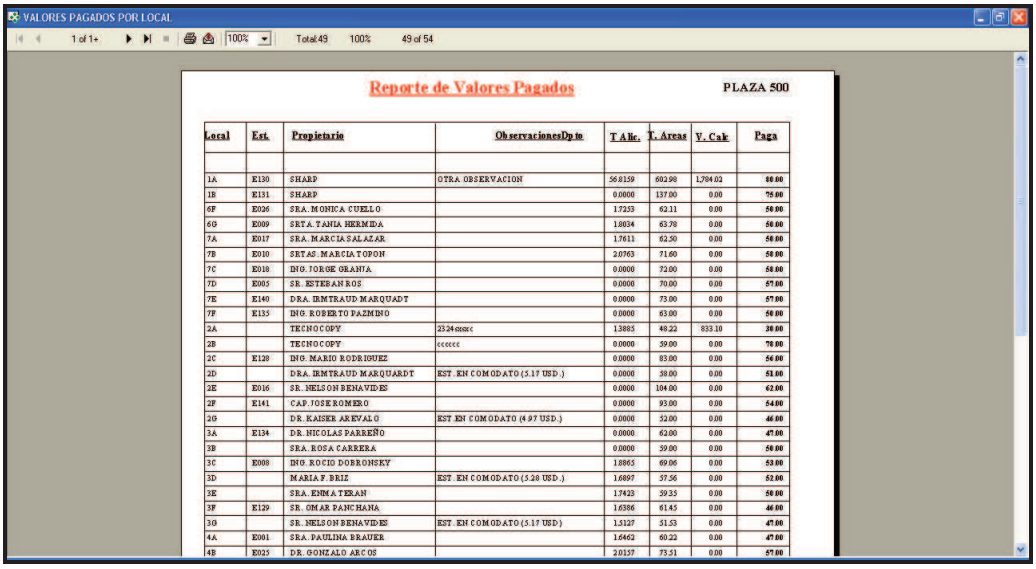


FIGURA 12: MENAJE DE LOCALES
ELABORADO POR: EL AUTOR

4.2 IMPRIMIR INFORMACIÓN

Permite al usuario visualizar un resumen de la información registrada para cada local, especialmente indica los valores por cocepto de alícuotas.



VALORES PAGADOS POR LOCAL

Reporte de Valores Pagados — PLAZA 500

Local	Est.	Propietario	ObservacionesDpto	T Alic.	T. Areas	V. Cak	Paga
1A	E130	SHARP	OTRA OBSERVACION	56.8159	602.98	1,794.02	80.00
1B	E131	SHARP		0.0000	137.00	0.00	75.00
6F	E026	SRA. MONICA CUELLO		1.7253	62.11	0.00	50.00
6G	E009	SRTA. TANIA HERMIDA		1.8034	63.78	0.00	50.00
7A	E017	SRA. MARCIA SALAZAR		1.7611	62.50	0.00	50.00
7B	E010	SRTAS. MARCIA TOPON		2.0763	71.60	0.00	58.00
7C	E018	ING. JORGE GRANJA		0.0000	72.00	0.00	58.00
7D	E005	SR. ESTEBAN ROS		0.0000	70.00	0.00	57.00
7E	E140	DRA. IRMTRAUD MARQUADT		0.0000	73.00	0.00	57.00
7F	E135	ING. ROBERTO PAZMINO		0.0000	63.00	0.00	50.00
2A		TECNOCOPY	23:24 cccc<	1.3885	48.22	833.10	30.00
2B		TECNOCOPY	<<<<<<	0.0000	59.00	0.00	70.00
2C	E128	ING. MARIO RODRIGUEZ		0.0000	83.00	0.00	56.00
2D		DRA. IRMTRAUD MARQUARDT	EST. EN COMODATO (5.17 USD.)	0.0000	59.00	0.00	51.00
2E	E016	SR. NELSON BENAVIDES		0.0000	104.00	0.00	62.00
2F	E141	CAP. JOSE ROMERO		0.0000	93.00	0.00	54.00
2G		DR. KAISER AREVALO	EST. EN COMODATO (4.97 USD.)	0.0000	52.00	0.00	46.00
3A	E134	DR. NICOLAS PARREÑO		0.0000	62.00	0.00	47.00
3B		SRA. ROSA CARRERA		0.0000	59.00	0.00	50.00
3C	E008	ING. ROCIO DOBRONSKY		1.8865	69.06	0.00	53.00
3D		MARIA F. BRIZ	EST. EN COMODATO (5.26 USD.)	1.6897	57.56	0.00	52.00
3E		SRA. ENMA TERAN		1.7423	59.35	0.00	50.00
3F	E129	SR. OMAR PANCHANA		1.6386	61.45	0.00	46.00
3G		SR. NELSON BENAVIDES	EST. EN COMODATO (5.17 USD.)	1.5127	51.53	0.00	47.00
4A	E001	SRA. PAULINA BRAUER		1.6462	60.22	0.00	47.00
4B	E025	DR. GONZALO ARCOS		2.0157	73.51	0.00	57.00

FIGURA 13: REPORTE DE VALORES PAGADOS
ELABORADO POR: EL AUTOR

5. MENÚ INGRESOS

Permite al usuario registrar los valores que el edificio recauda por concepto de: Alícuotas, Cuotas Extraordinarias y Deudas Anteriores.

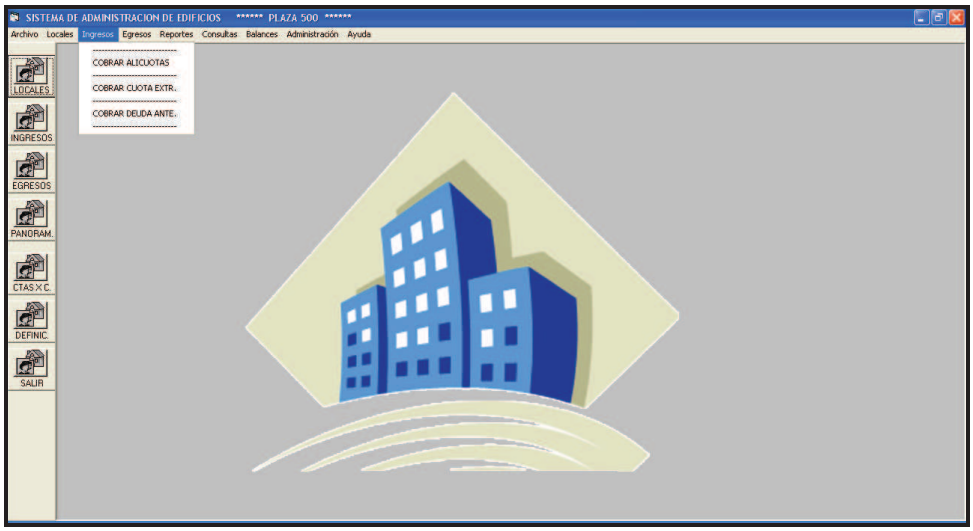


FIGURA 14: MENÚ INGRESOS
ELABORADO POR: EL AUTOR

5.1 COBRAR ALÍCUOTAS

Despliega el formulario mediante el cual el usuario del sistema registrará los cobros realizados mensualmente a propietarios o arrendatarios de locales por concepto de alícuotas.

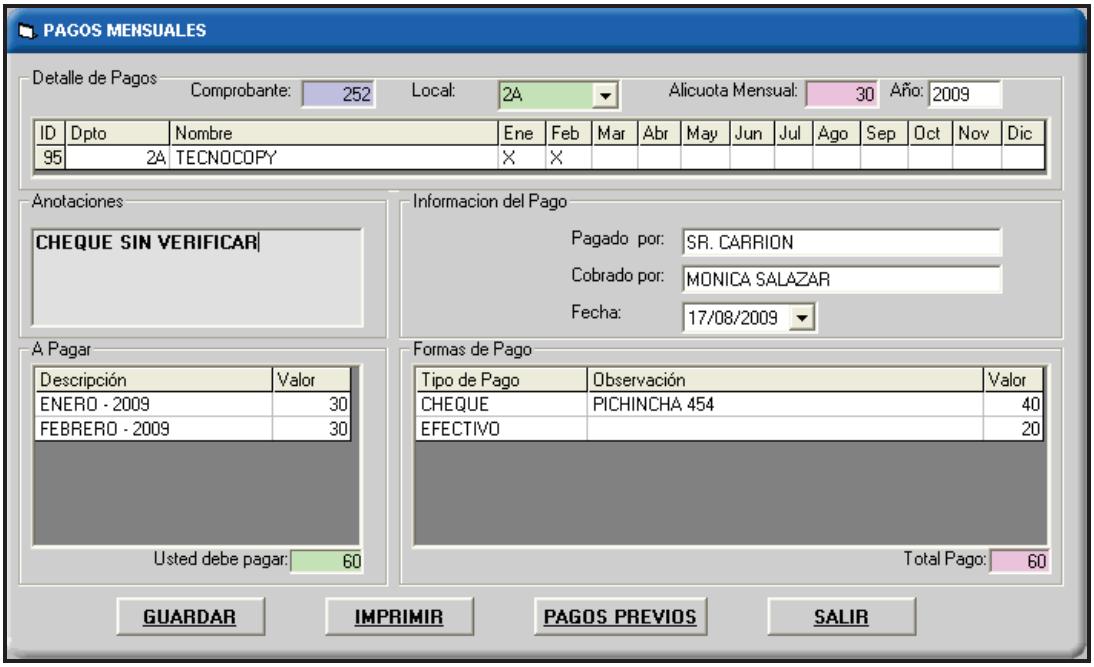


FIGURA 15: COBRO DE ALÍCUOTAS
ELABORADO POR: EL AUTOR

Luego de registrar la información, el usuario guarda el formulario y obtiene en la pantalla automáticamente el comprobante generado.

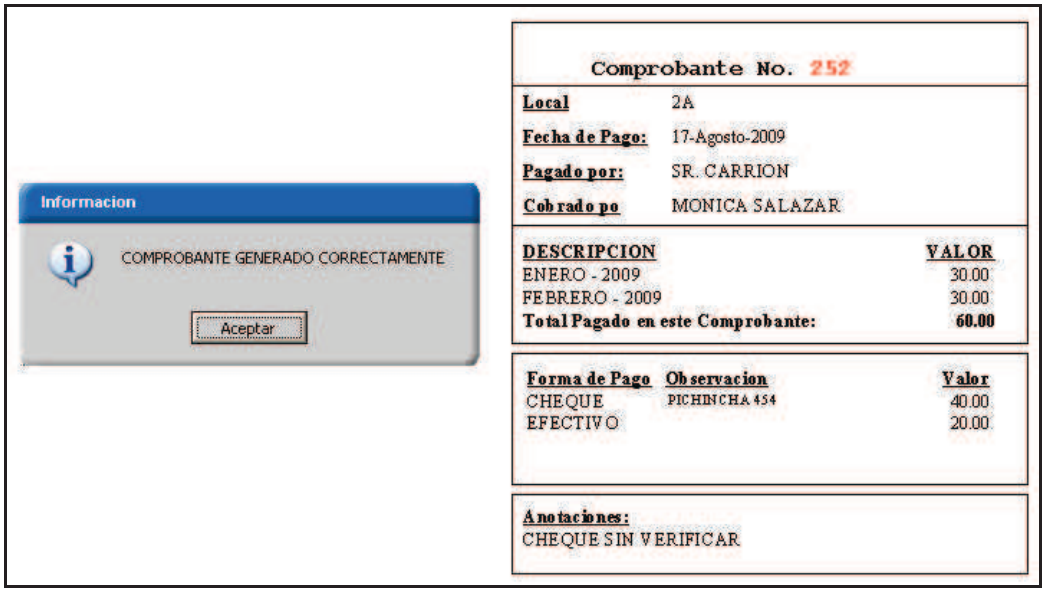


FIGURA 16: COMPROBANTE ALÍCUOTAS
ELABORADO POR: EL AUTOR

5.2 COBRAR CUOTA EXTRAORDINARIA

Este modulo es utilizado para cobrar cuotas extraordinarias que la administración del edificio establece con el objetivo de cubrir gastos que no fueron considerados dentro del presupuesto. Para efectuar un cobro, el administrador del sistema debe definir dicha cuota en el modulo "Administración", y posterior a este proceso realizar el cobro como se muestra a continuación:

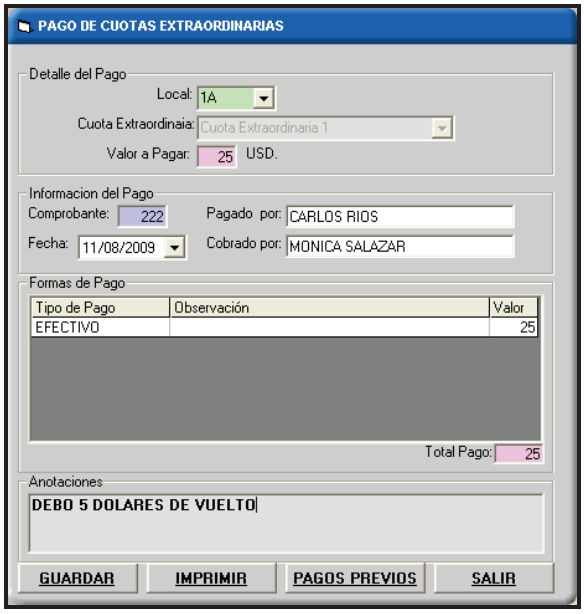


FIGURA 17: COBRO DE CUOTAS EXTRAORDINARIAS
ELABORADO POR: EL AUTOR

Luego de guardar la información ingresada, el sistema muestra automáticamente el comprobante generado.

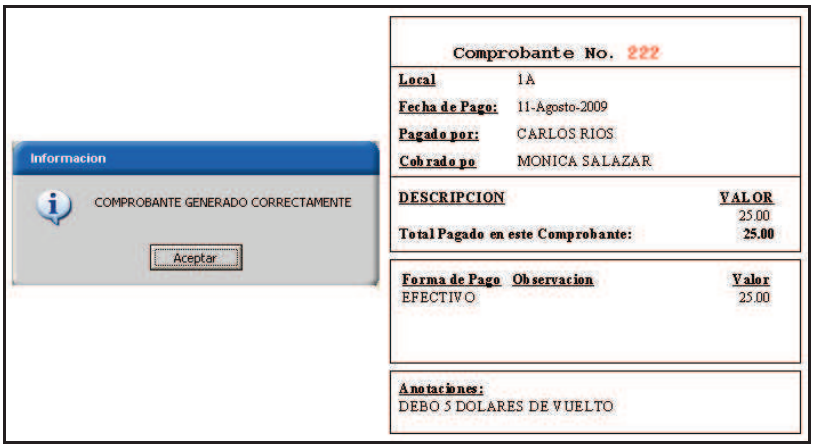


FIGURA 18: COMPROBANTE CUOTAS EXTRAORDINARIAS
ELABORADO POR: EL AUTOR

5.2 COBRAR DEUDA ANTERIOR

El usuario del sistema utilizará este modulo para cobrar a propietarios o arrendatarios deudas de años anteriores que no han sido canceladas a la administración oportunamente, los valores recaudados por este concepto serán incluidos a los ingresos del año vigente.

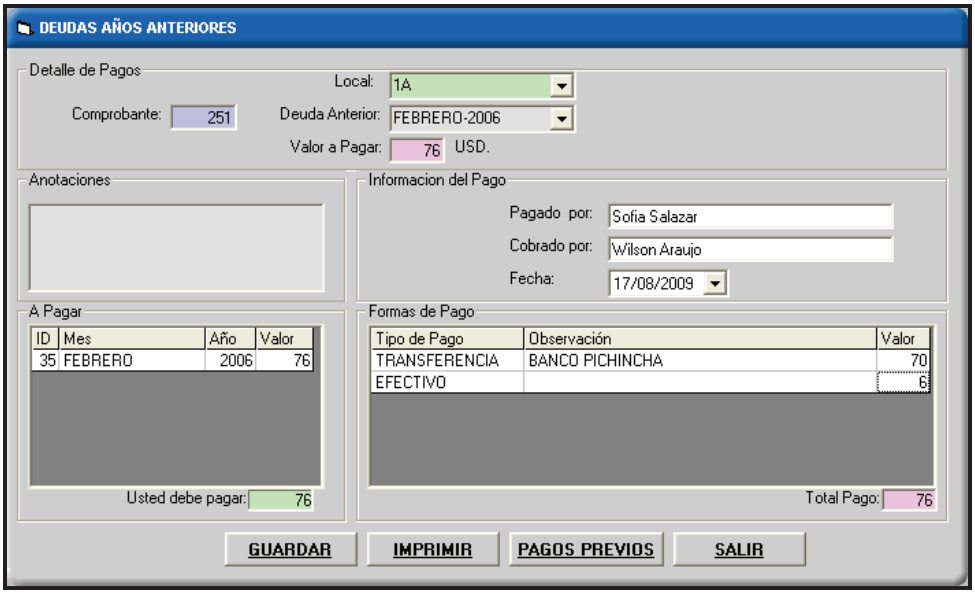


FIGURA 19: COBRO DE DEUDAS ANTERIORES
ELABORADO POR: EL AUTOR

Posterior al ingreso de la información el usuario guarda el formulario y se genera el comprobante respectivo.

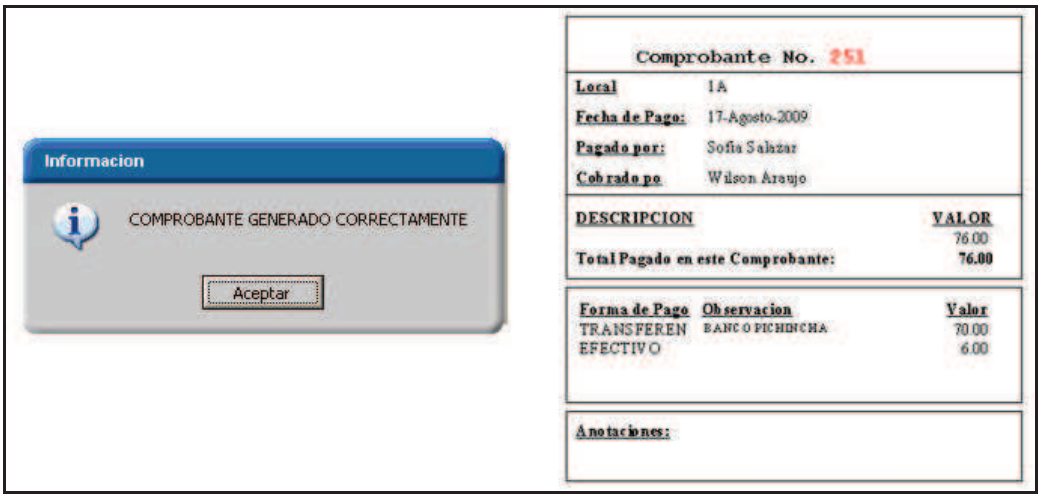


FIGURA 20: COMPROBANTE CUOTAS EXTRAORDINARIAS
ELABORADO POR: EL AUTOR

6. MENÚ EGRESOS

Permite al usuario registrar cada uno de los gastos que el administrador realiza a fin de cubrir gastos administrativos y de mantenimiento del edificio.

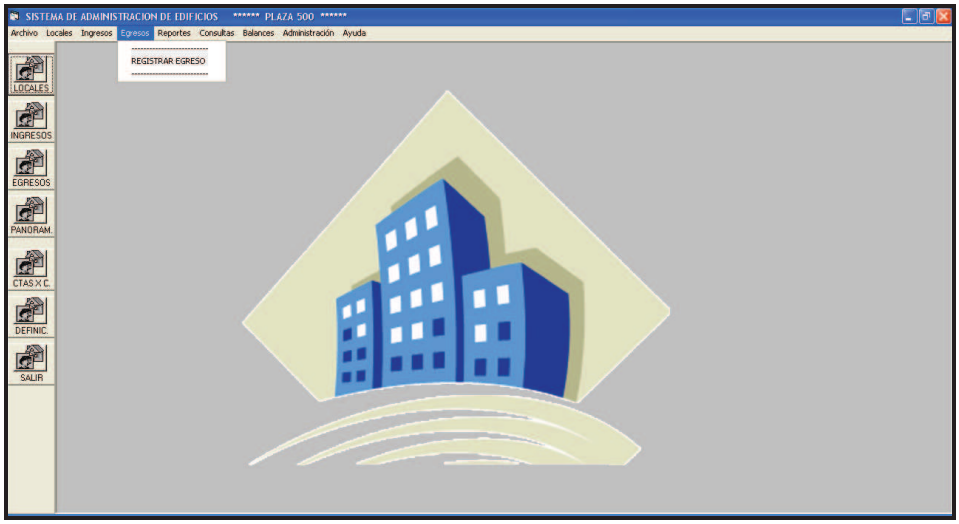


FIGURA 21: MENÚ EGRESOS
ELABORADO POR: EL AUTOR

6.1 REGISTRAR EGRESO

Para el registro de egresos, se debe previamente definir los beneficiarios o proveedores a quienes la administración cancela valores por sus servicios; en el presente modulo la información se registra como se muestra a continuación:

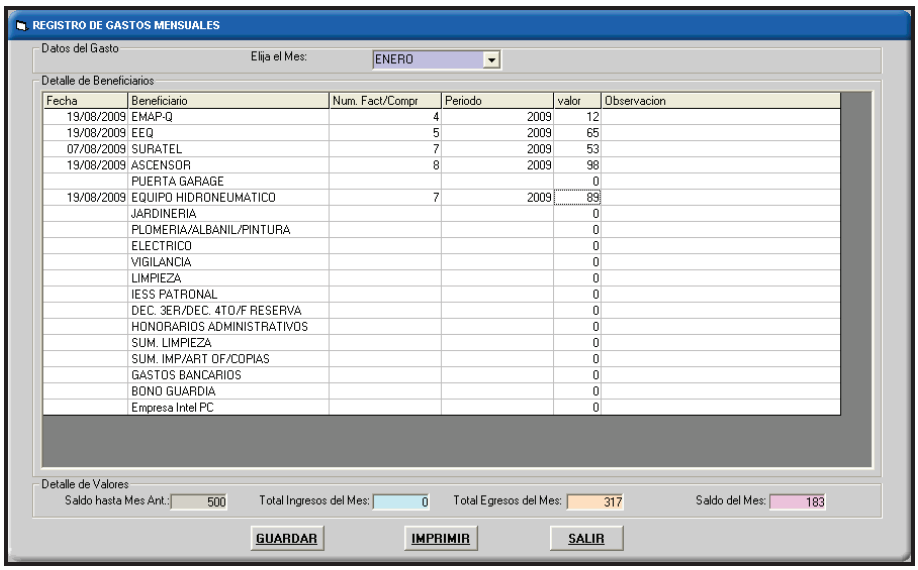


FIGURA 22: REGISTRO DE EGRESOS
ELABORADO POR: EL AUTOR

7. MENÚ DE REPORTES, CONSULTAS Y BALANCES

En esta sección se describe tres módulos disponibles dentro del sistema con el objetivo de ofrecer al usuario documentos de resumen de toda la actividad economica registrada en los diferentes modulos de la aplicación.

7.1 REPORTES

Las opciones disponibles dentro del modulo reportes se describen a continuación:

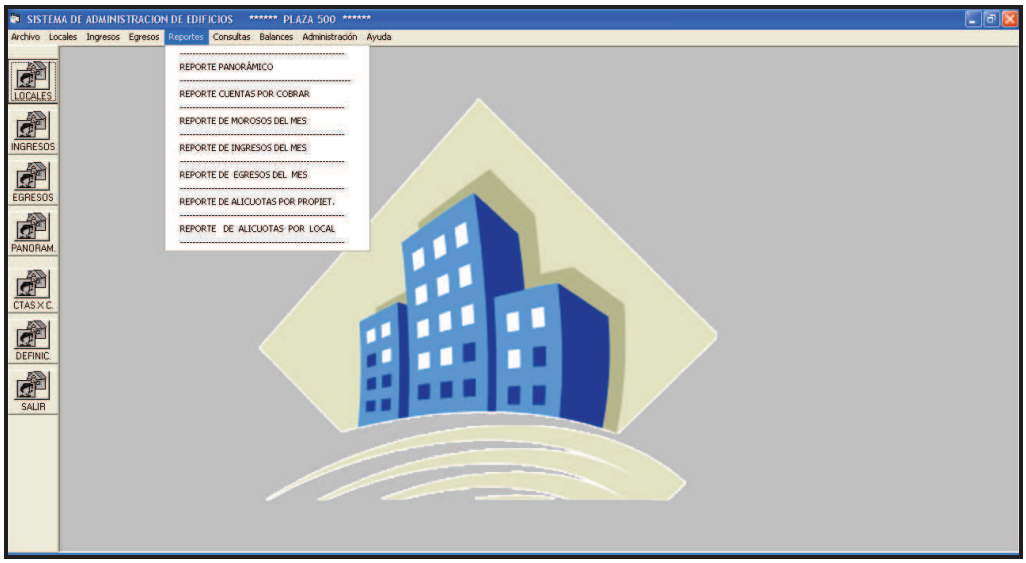


FIGURA 23: REPORTES DEL SISTEMA
ELABORADO POR: EL AUTOR

REPORTE	DESCRIPCIÓN
Reporte Panorámico	Muestra un resumen del cobro de alícuotas realizado en funcion del comprobante entregado.
Reporte Ctas por Cob.	Muestra una síntesis de los valores que no han sido cancelados a la administración.
Reporte de Morosos del Mes	Muestra la información de los locales que no han cancelado sus obligaciones en el mes actual.
Reporte de Ingresos del Mes	Muestra la información de todos los comprobantes que han sido generados en el mes actual.
Reporte de Egresos del Mes	Muestra la información de los valores entregados por la administración a los proveedores durante el mes actual.
Reporte de Alícuotas Prop.	Muestra los montos que deben ser cancelados por cada local en funcion del nombre del propietario.
Reporte de Alícuotas Local	Muestra los montos que deben ser cancelados por cada local en funcion del código del local.

7.2 CONSULTAS

Las consultas disponibles en el sistema y los datos requeridos se describen a dentro del módulo consulta se describen a continuación:

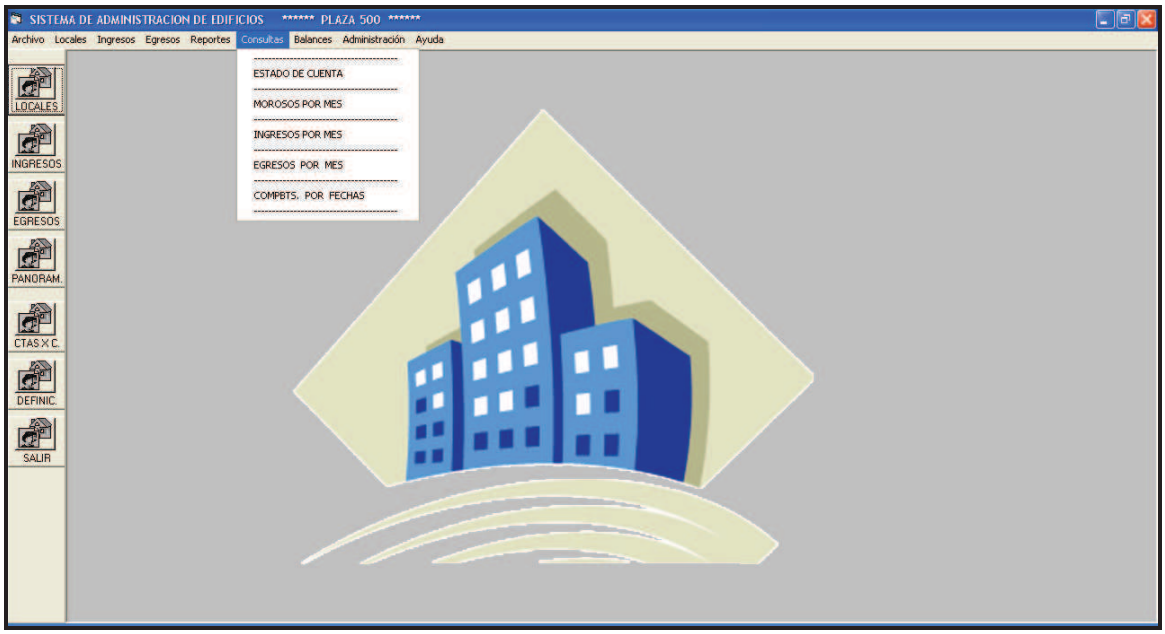


FIGURA 24: CONSULTAS DEL SISTEMA
ELABORADO POR: EL AUTOR

CONSULTA	DATOS NECESARIOS	DESCRIPCIÓN
Estado de Cuenta	Local Rango de Fechas	Indica los valores pagados por un local en un periodo determinado de tiempo.
Morosos por Mes	Mes Consultado	Indica los locales que no han pagado el monto establecido como alícuota durante el mes consultado
Ingresos por Mes	Mes Consultado	Indica los locales que han cancelado sus alícuotas en funcion de los comprobantes durante el mes consultado.
Egresos por Mes	Mes Consultado	Indica los valores pagados por la administración a sus proveedores durante el mes consultado.
Comprobs. por Fechas	Rango de Fechas	Indica los comprobantes generados durante un periodo de tiempo.

7.3 BALANCES

El Sistema de Administración de Edicios ofrece al usuario dos balances que resumen y respaldan la gestión del administrador durante periodos semestrales. Al igual que un sistema contable, el balance del primer semestre resume las actividades realizadas durante los seis primeros meses del año, y el balance del segundo semestre las actividades efectuadas en los ultimos seis meses del año.

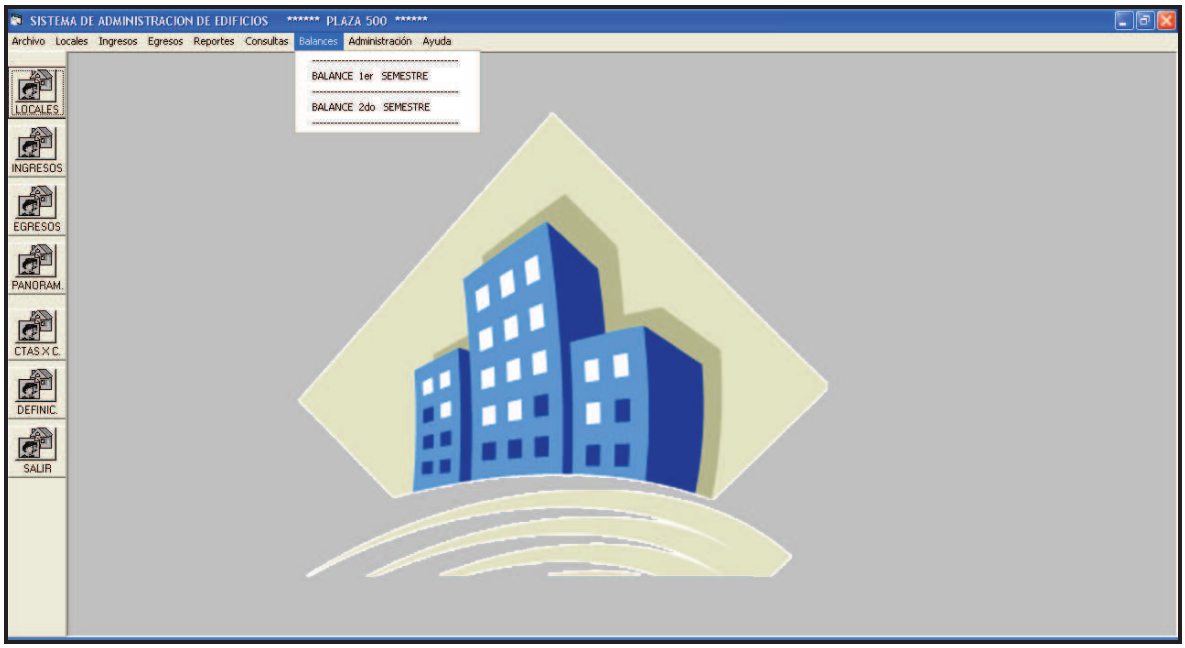


FIGURA 25: BALANCES DEL SISTEMA
ELABORADO POR: EL AUTOR

BALANCE	DESCRIPCIÓN
Balance 1er. Semestre	Resume los ingresos y egresos registrados en un edificio desde el mes de enero hasta el mes de junio.
Balance 2do. Semestre	Resume los ingresos y egresos registrados en un edificio desde el mes de julio hasta el mes de diciembre.

8. MENÚ ADMINISTRACIÓN

En este modulo se agrupan las principales definiciones que el usuario debe especificar para garantizar un optimo desempeño del sistema. Unicamente los usuarios con perfil de “Administrador” pueden acceder a modificar las opciones.

8.1 CONDOMINIO

En este módulo el sistema permite incluir un nuevo condominio, editar la información previamente ingresada o finalmente eliminar un registro.

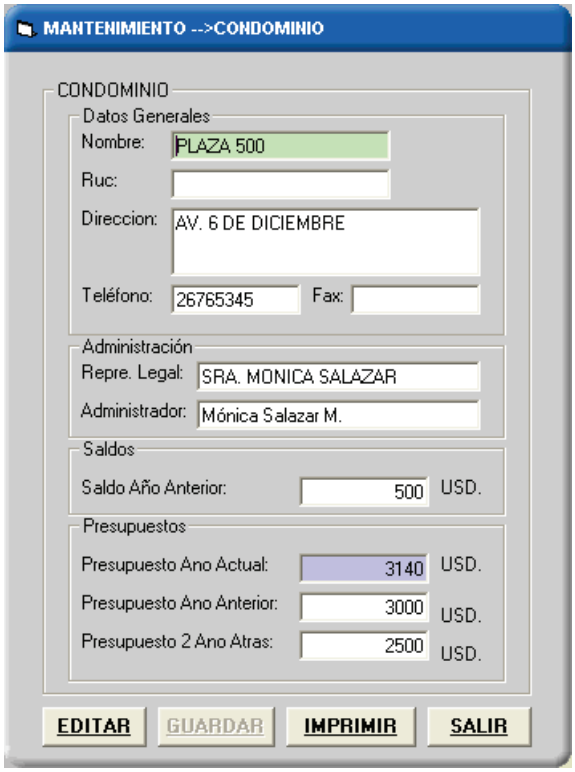


FIGURA 26: CREACIÓN DE NUEVO CONDOMINIO
ELABORADO POR: EL AUTOR

8.2 DEFINICIONES

Las definiciones proporcionadas al usuario para indicar los parámetros que deben ser editados, se listan a continuación:

<u>Definir Secuencial</u>

Especifica el número inicial que será utilizado en el registro e impresión de comprobantes de ingreso.

FIGURA 27: DEFINICIÓN DE SECUENCIAL
ELABORADO POR: EL AUTOR

Definir Cuota Extraordinaria

Una cuota extraordinaria es utilizada para cubrir un gasto del edificio que no fue considerado en el presupuesto del año actual, la informacion que debe ser ingresada es: datos de la cuota, valor a pagar y número de comprobantes que serán utilizados para el cobro.

FIGURA 28: DEFINICIÓN DE CUOTA EXTRAORDINARIA
ELABORADO POR: EL AUTOR

Definir Egresos

El usuario debe definir los beneficiarios o proveedores de servicios a los cuales la administración cancela mensualmente valores, cada registro debe ser ubicado dentro de un grupo a fin de organizarlos dentro de los reportes.

FIGURA 29: DEFINICIÓN DE EGRESOS
ELABORADO POR: EL AUTOR

Definir Deudas Anteriores

Permite al administrador especificar dentro del sistema deudas que no fueron canceladas oportunamente para que sean cobradas en el presente periodo.

FIGURA 30: DEFINICIÓN DE DEUDAS ANTERIORES
ELABORADO POR: EL AUTOR

Definir Tipo Pago

En este modulo el administrador del sistema debe indicar las formas de pago que estarán habilitadas para el cobro de alicuotas, cuotas extraordinarias y deudas anteriores.

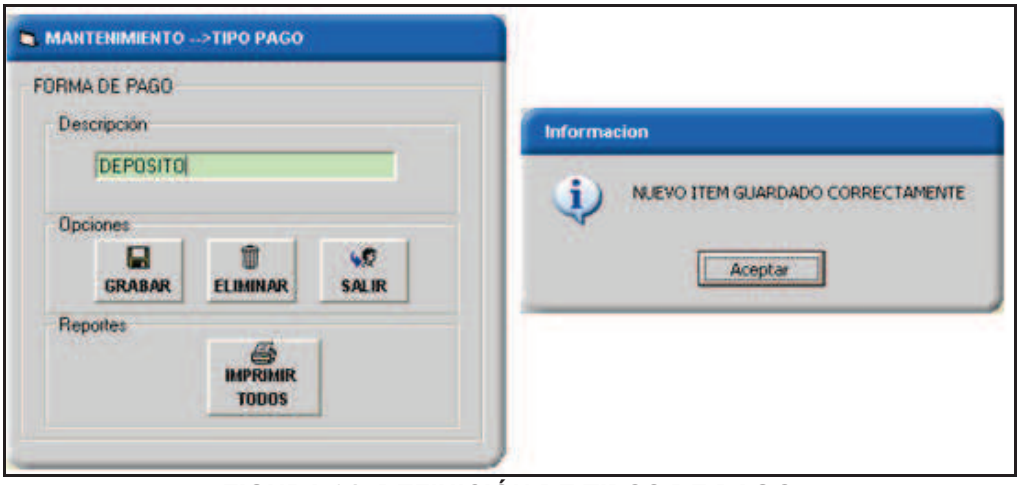


FIGURA 31: DEFINICIÓN DE TIPOS DE PAGO
ELABORADO POR: EL AUTOR

8.3 COMPROBANTES

Permite al usuario realizar las siguientes acciones: reimpresión de comprobantes y anulación de comprobantes.

Reimpresión de Comprobantes

Genera nuevamente un comprobante efectuado durante el proceso de cobro.

FIGURA 32: REIMPRESIÓN DE COMPROBANTES
ELABORADO POR: EL AUTOR

Anulación de Comprobantes

Cambia el estado de un comprobante a fin de no incluirlo dentro de los balances, ni tomarlo en cuenta al momento de calcular los ingresos de un periodo.

FIGURA 33: ANULACIÓN DE COMPROBANTES
ELABORADO POR: EL AUTOR

8.4 CIERRE DE AÑO

El proceso de cierre de año ofrece al usuario la posibilidad de acumular las deudas que no fueron canceladas durante el periodo actual y reiniciar los modulos del sistema a fin de comenzar un nuevo periodo de actividades.

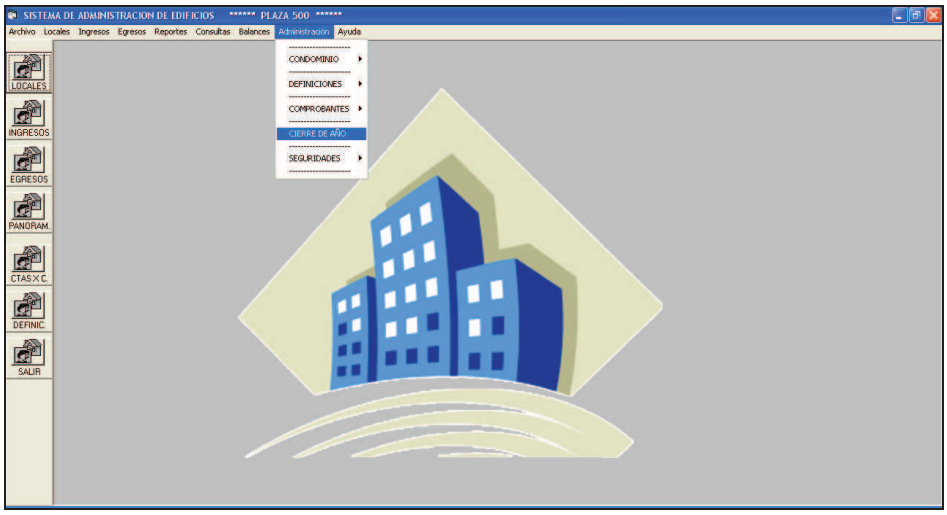


FIGURA 34: CIERRE DE AÑO
ELABORADO POR: EL AUTOR

El usuario debe confirmar la ejecución de este proceso respondiendo afirmativamente a la pregunta expuesta en el recuadro informativo.

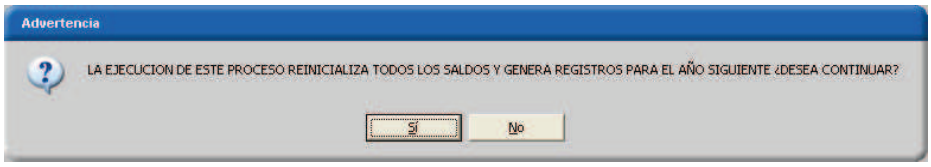


FIGURA 35: CONFIRMACIÓN DE CIERRE DE AÑO
ELABORADO POR: EL AUTOR

8.4 SEGURIDADES

Las seguridades establecidas dentro del sistema evita que la información de un condominio sea alterada o incluso eliminada, por ello se han definido tras perfiles: administración, cobranzas y consultas. En el presente modulo se puede realizar las siguientes acciones con la información de un usuario: incluir un nuevo registro, editar la información y eliminarlo del sistema.

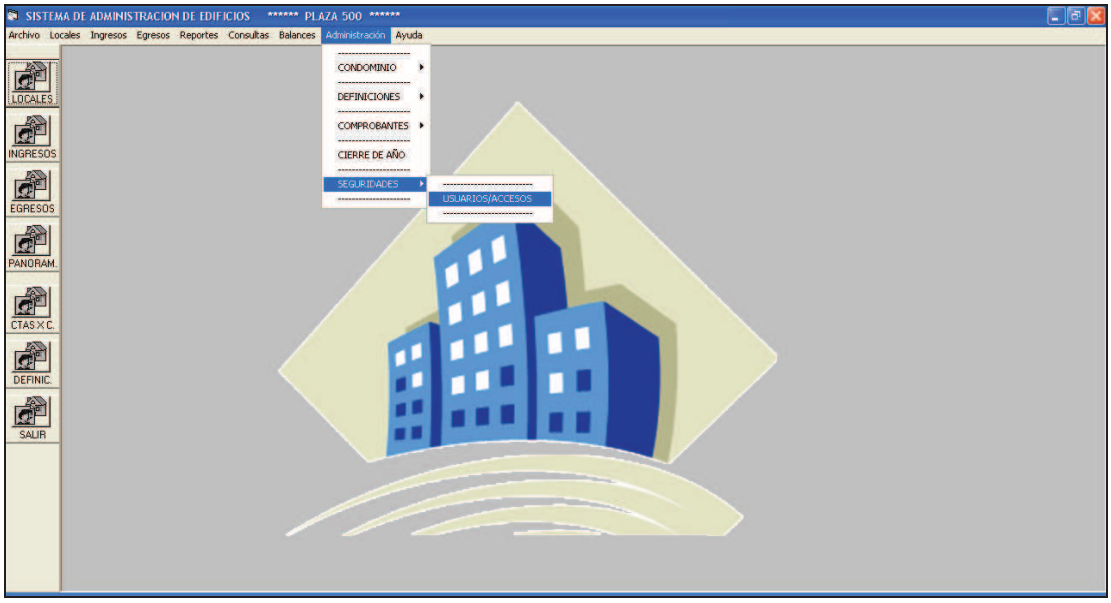


FIGURA 36: SEGURIDADES DEL SISTEMA
ELABORADO POR: EL AUTOR

En la figura 37, se muestra el ingreso de un nuevo usuario y la asignación de su respectivo perfil.

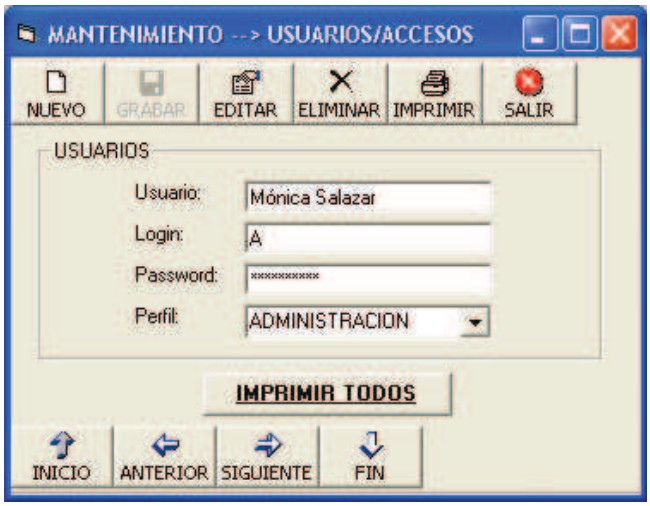


FIGURA 37: MANTENIMIENTO DE USUARIOS
ELABORADO POR: EL AUTOR

Manual de Instalación

CONTENIDO

1. REQUERIMIENTOS
1.1 Hardware……..... 128
1.2 Software…… 128

2. INSTALACIÓN
2.1 Procedimiento…… 128

3. ACCESOS DIRECTOS
3.1 Inicio de Windows…… 133
3.2 Escritorio…… 133

1. REQUERIMIENTOS

La instalación del Sistema de Administración de Edificios implica que el usuario debe disponer de recursos tanto fisicos como logicos para que el sistema obtenga un óptimo funcionamiento.

1.1 HARDWARE

El sistema debe ser instalado en un computador que posea las siguientes caracteristicas de hardware:

- Procesador Intel Pentium 4 o superior.
- Memoria RAM 512 Mb. o superior.
- Resolucion de Pantalla minima de 800 x 600.

1.2 SOFTWARE

La plataforma establecida como óptima para el funcionamiento del Sistema de Administración de Edificios requiere el siguiente software:

- Windows XP o superior
- SQL Server 2000 o superior (Preinstalado)

2. INSTALACIÓN

Para facilitar al usuario el proceso de instalación, se ha empaquetado el sistema de Administración de Edificios en un instalador muy sencillo de comprender. A continuación se describe el procedimiento que el usuario debe seguir para lograr una completa instalacion de la aplicación.

2.1 PROCEDIMIENTO

La instalación del sistema inicia localizando el archivo “Setup” incluido en el CD que contine el instalador de la aplicación.

FIGURA 1: ICONO DE INSTALACIÓN DEL SISTEMA
ELABORADO POR: EL AUTOR

Realizar doble clic sobre el icono “Setup” y leer el primer recuadro, y pulsar el botón “Siguiente”.

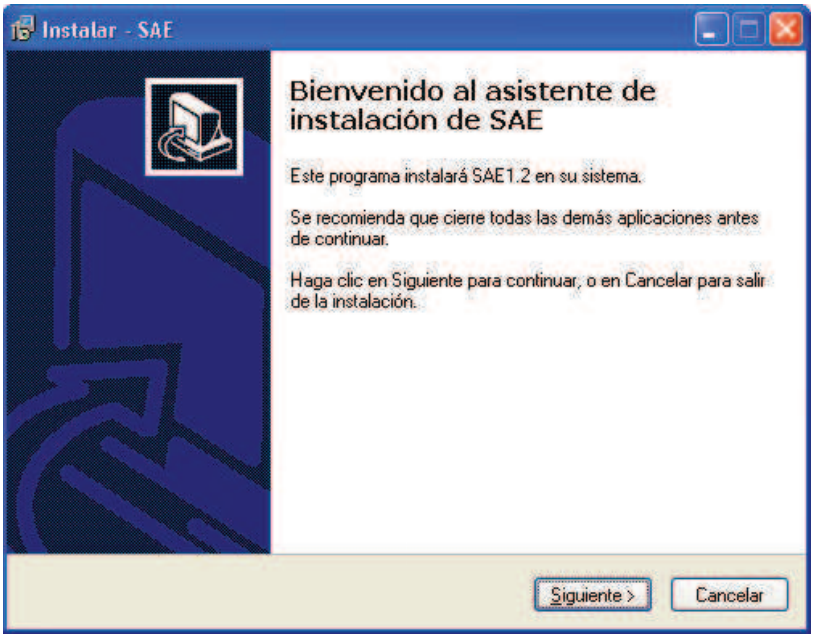


FIGURA 2: PANTALLA DE BIENVENIDA A LA INSTALACIÓN
ELABORADO POR: EL AUTOR

En el recuadro siguiente, el usuario debe especificar la ubicación donde se copiarán los archivos del sistema, luego pulsar el botón “Siguiente”

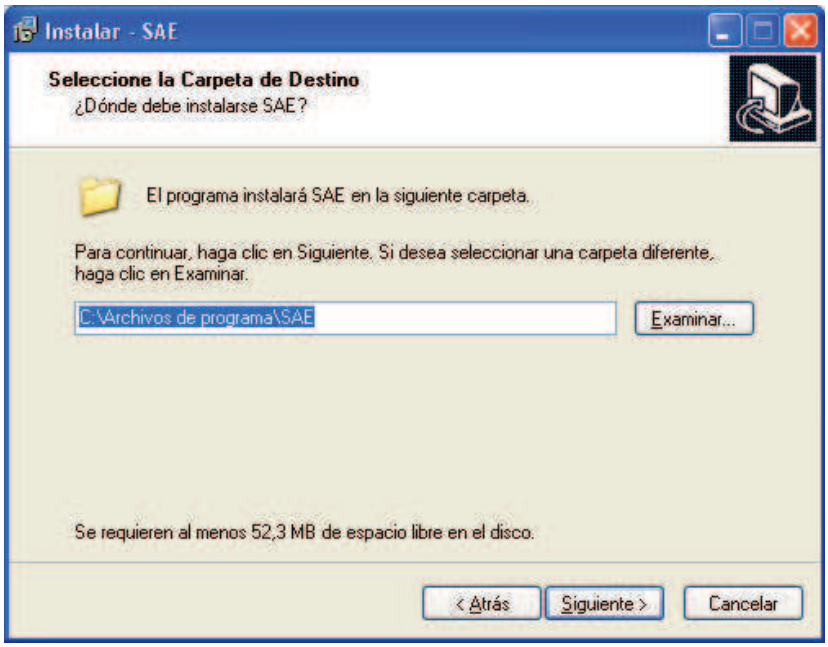


FIGURA 3: SELECCIÓN DE LA CARPETA DE DESTINO
ELABORADO POR: EL AUTOR

Se obtiene un nuevo recuadro en el que se debe especificar el nombre de la carpeta, que en la que se almacenarán los ejecutables, luego pulsar “Siguiente”.

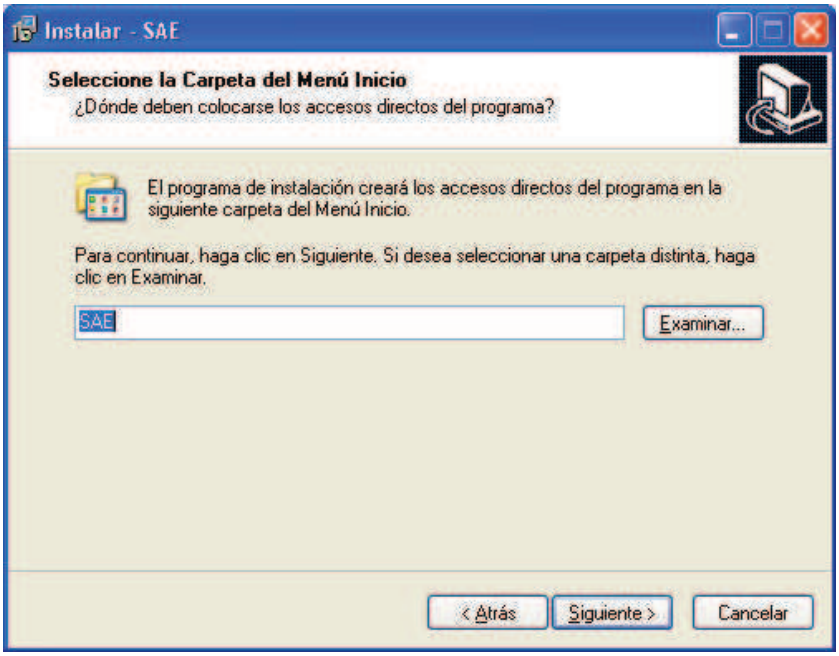


FIGURA 4: SELECCIÓN DE LA CARPETA DEL MENÚ INICIO
ELABORADO POR: EL AUTOR

Marcar mediante el recuadro que se muestra a continuación, crear un icono en el escritorio, luego pulsar “Siguiente”.

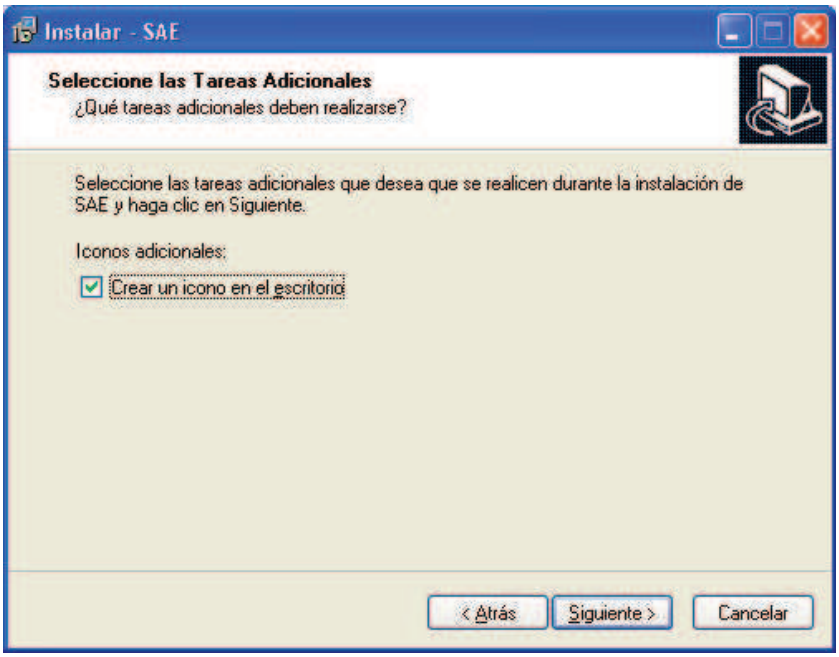


FIGURA 5: SELECCIÓN DE TAREAS ADICIONALES
ELABORADO POR: EL AUTOR

La siguiente pantalla muestra la confirmación de los datos ingresados para la instalación, luego de lo cual se debe presionar el botón “Instalar”.

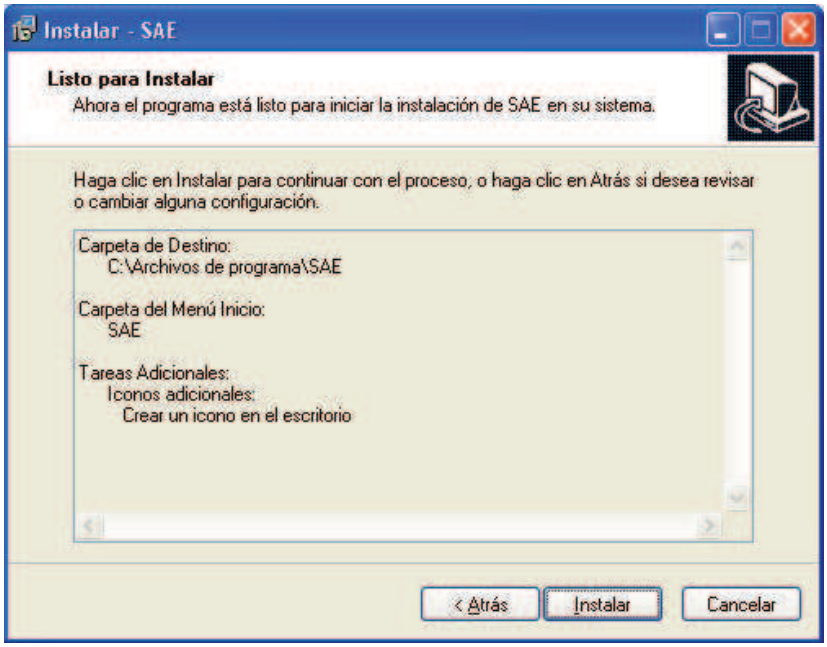


FIGURA 6: INICIO DEL PROCESO DE INSTALACIÓN
ELABORADO POR: EL AUTOR

El usuario podrá observer el progreso de la instalació mediante el siguiente recuadro.

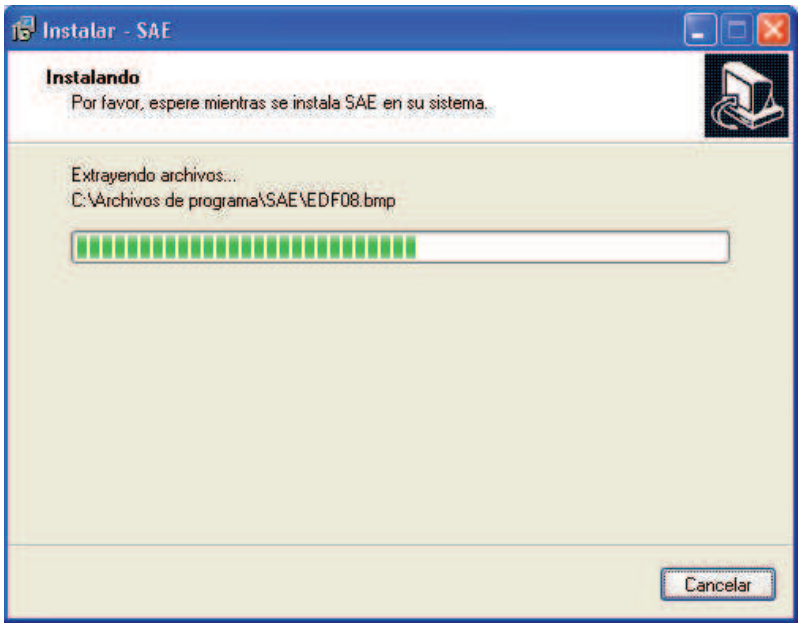


FIGURA 7: EJECUCIÓN DE LA INSTALACIÓN DEL SISTEMA
ELABORADO POR: EL AUTOR

Para finalizar la instalación el usuario debe marcar afirmativamente el cuadro “Ejecutar SAE”, y pulsar el botón “Finalizar”.

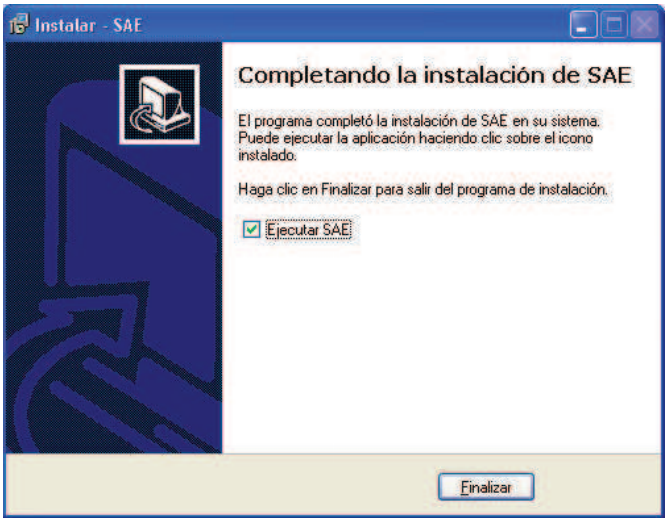


FIGURA 8: FINALIZACIÓN DE LA INSTALACIÓN
ELABORADO POR: EL AUTOR

El instalador del programa muestra un mensaje informativo al usuario, que indica la manera en que se podrá desinstalar la aplicación de ser necesario quitarla del computador.

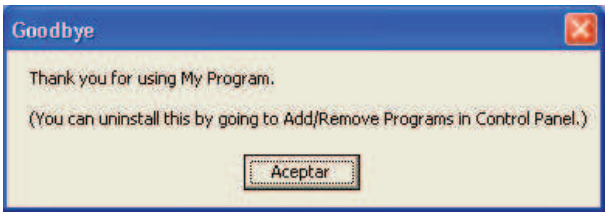


FIGURA 9: MENSAJE DE AGRADECIMIENTO
ELABORADO POR: EL AUTOR

Automáticamente el sistema arranca y muestra al usuario la pantalla de bienvenida.

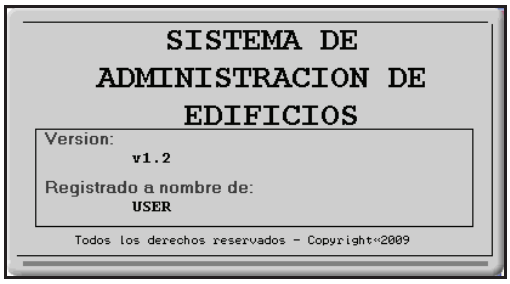


FIGURA 10: PANTALLA DE BIENVENIDA DEL SISTEMA
ELABORADO POR: EL AUTOR

3. ACCESOS DIRECTOS

Una vez que el sistema ha sido instalado en el computador, el usuario dispone de varios accesos directos que pueden ser utilizados para ingresar a la aplicación.

3.1 INICIO DE WINDOWS

La ruta más utilizada para el ingreso al sistema es la siguiente:

Inicio >> Todos los programas >> SAE >> SAE

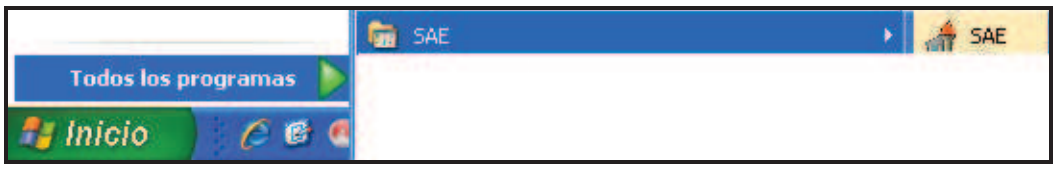


FIGURA 11: RUTA DE ACCESO AL SISTEMA
ELABORADO POR: EL AUTOR

3.2 ESCRITORIO

Luego de instalar el sistema, el usuario puede encontrar en el escritorio de su computador el icono de acceso directo que le ofrece una rápida alternativa de ingreso al Sistema de Admistración de Edificios, su imagen se muestra a continuación.

FIGURA 12: ICONO DE ACCESO CREADO EN EL ESCRITORIO
ELABORADO POR: EL AUTOR