

UNIVERSIDAD DE CIENFUEGOS

"CARLOS RAFAEL RODRÍGUEZ"

Facultad de Informática

# Sistema de Pago de Impuesto Online *Para el Ayuntamiento de* Georgetown Guyana.



Trabajo de Diploma para optar por el título de Ingeniería en  
Informática

Autor: Adunni Christian

Tutores: MSc. Kadir Hector Ortiz

MSc. Oscar Alejo Machado

CIENFUEGOS, 2008

### **Declaración de Autoría**

Declaro que el presente trabajo de diploma fue realizado en la Universidad de Cienfuegos Carlos Rafael Rodríguez como parte de la culminación de los estudios de la especialidad de Ingeniería Informática autorizando a que el mismo sea utilizado por la institución y el Ayuntamiento de la ciudad de Georgetown para los fines que estimen conveniente, ya sea parcial o totalmente.

---

Firma Autor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

---

Firma Tutor

---

Firma Tutor

---

Vicedecano

---

Firma ICT

## ***Agradezco:***

- *Al más alto señor, ELIJAH, por su misericordia durante los años, pues sin él nada es posible.*
- *A todo mi familia, especialmente mi esposa, mis hijas y mi madre por su amor paciencia y fe durante este largo viaje.*
- *A Wayne y Eva por creer en mi cuando yo no creía en mi mismo.*
- *A mi soldado Oluki por todo su ayuda y amistad.*
- *A mis tutores, MSc. Kadir Héctor y MSc. Oscar Alejo, especialmente a Kadir por su paciencia, su preocupación, y el tiempo que me dedicó.*
- *A la Revolución Cubana, y el ayuntamiento de la ciudad de Georgetown les agradezco mucho por darme esa oportunidad.*
- *A todos, muchísimas gracias.*

## **Resumen**

El presente trabajo está orientado a desarrollar un sistema informático que permita la gestión de la información en el ayuntamiento de la Ciudad de Georgetown Guyana y el pago de los impuestos de propiedad online en el departamento de tesorería de dicho ayuntamiento. Con este sistema se pretende brindar a los propietarios de la ciudad la posibilidad de pagar su impuesto de propiedad a través del Internet y que dentro del ayuntamiento se pueda manipular información mediante una Intranet

El resultado de la aplicación propuesta representa un significativo aporte al proceso de pago de impuestos para el departamento de tesorería, aumentando el alcance de su gestión por su presencia en Internet y ofrece una vía directa de intercambio entre los propietarios y el departamento de tesorería, optimizando así todo el proceso, con lo que se contribuye significativamente a lograr los objetivos de desarrollo.

Para el desarrollo del sistema se utilizó la metodología RUP, por su amplia aceptación y facilidad de uso. Como herramienta de desarrollo se utilizó el Adobe Dreamweaver CS3, el cual incluye facilidades de desarrollo para aplicaciones Web, incluyendo generación en código PHP, Java Script, HTML; lenguajes utilizados en la programación del software. Para la gestión de los datos se utilizó como Sistema de Gestión de Base de Datos: MySQL.

## Índice

1.1	Introducción.....	5
1.2	Principales conceptos asociados al dominio del problema. ....	5
1.3	Descripción general del objeto de estudio.....	12
1.4	Flujo actual de los procesos y análisis crítico de la ejecución de estos. ....	13
1.5	Descripción de los sistemas existentes.....	13
1.6	Paradigmas de la programación.....	13
1.7	Descripción de las Tecnologías a utilizar .....	15
1.8	Lenguaje y Metodologías para la modelación. ....	25
1.9	Conclusiones.....	27
Capítulo 2 – Modelo del negocio .....		29
2.1	Introducción.....	29
2.2	Descripción de los procesos del negocio .....	29
2.3	Reglas del negocio a considerar .....	31
2.4	Modelo de casos de uso del negocio .....	32
2.4.1	Actores del negocio .....	32
2.4.2	Diagramas de casos de uso del negocio .....	33
2.4.6	Modelo de objetos del negocio.....	41
Capítulo 3: Construcción de la Solución Propuesta .....		42
3.1	Introducción.....	42
3.2	Requerimientos funcionales .....	42
3.3	Requerimientos no funcionales .....	43
3.4	Modelo de casos de uso del sistema .....	47
3.4.1	Actores del sistema.....	47

3.4.2 Caso de Uso del Sistema .....	48
3.4.3 Descripción de los casos de uso del sistema .....	50
3.5 Principios de Diseño del Sistema .....	62
3.6 Estándares en la Interfaz de la Aplicación .....	62
3.7 Tratamiento de Excepciones .....	63
Modelo de Clases Web .....	63
3.8 Diseño de la base de datos .....	65
3.8.1 Diagrama del modelo lógico de datos .....	65
3.8.2 Diagrama del modelo físico de datos .....	65
3.9 Diagrama de Implementación .....	65
3.10 Conclusiones .....	65
Capitulo IV Estudio de Factibilidad .....	67
4.1 Introducción .....	67
4.2 Planificación por puntos de función .....	67
4.3 Conclusiones .....	78
Conclusiones .....	80
Recomendaciones .....	81
Referencias Bibliográficas .....	82
Bibliografía .....	87
Anexo 1 Diagramas de clase web .....	88
ANEXO II PROTOTIPOS .....	102
Anexo III Diagramas de la base de Datos y de Implementación .....	117
1.10 Anexo 3.3 Diagrama de Implementación .....	119

## Índice de tablas

Tabla 1. Comparación entre SGBD.....	25
Tabla 2. Descripción de los actores del negocio .....	33
Tabla 3: Trabajadores del Ngocio .....	34
Tabla 4: Descripción Caso de uso Pagar impuesto.....	36
Tabla 5: Descripción Caso de uso Emitir carta de aviso .....	37
Tabla 6. Descripción de los actores del sistema .....	48
Tabla 7Descripción del caso de uso Autenticarse.....	50
Tabla8 Descripción del caso de uso Recuperar la contraseña.....	51
Tabla 9: Descripción del caso de uso Cambiar Contraseña.....	51
Tabla 10: Descripción del caso de uso Gestionar Noticias.....	52
Tabla 11: Descripción del caso de uso Gestionar Eventos. ....	53
Tabla 12: Descripción del caso de uso Gestionar Usuarios. ....	55
Tabla 13: Descripción del caso de uso Gestionar Departamentos.....	55
Tabla 14: Descripción del caso de uso Gestionar Propiedad.....	57
Tabla 15: Descripción del caso de uso Gestionar Propietarios. ....	58
Tabla 16: Descripción del caso de uso Visualizar estado de la Cuenta. ....	58
Tabla 17: Descripción del caso de uso Pagar la cuenta de la propiedad. ....	59
Tabla 18: Descripción del caso de uso Visualizar lista de empleados .....	60
Tabla 19. Descripción del caso de uso de sistema Visualizar Noticias .....	60
Tabla 20: Descripción del caso de uso de sistema Visualizar Eventos .....	61
Tabla 21: Descripción del caso de uso de sistema Generar Reportes.....	64
Tabla 22 Planificación: Entradas Externas .....	70
Tabla 23. Planificación: Salidas Externas.....	70
Tabla 24. Planificación: Peticiones .....	71

Tabla 25 Planificación: Ficheros Internos.....	72
Tabla 26 Planificación: Puntos de función.....	73
Tabla 27 Planificación: Puntos de función.....	73
Tabla 28 Planificación: Puntos de función.....	75
Tabla 29 Costos totales.....	78

## Índice de Figuras

Figura 1. Esquema de representación del PHP .....	21
Figura 1. Diagramas de casos de uso del negocio.....	33
Figura 2. Diagramas de actividades del caso de uso Pagar Impuesto .....	39
Figura 3. Diagramas de actividades del caso de uso Emitir la carta de aviso .....	40
Figura 3. Diagramas de clases del modelo de objetos del negocio.....	41
Figura 4: Diagrama de caso de Uso del Sistema. ....	49

### INTRODUCCIÓN

El vertiginoso desarrollo Científico-Técnico alcanzado en el mundo a partir de las últimas décadas del siglo XX, ha llevado al hombre a lograr tal desarrollo que ya sería imposible prescindir de la informática.

Las actuales perspectivas de este desarrollo llevan a brindar la información oportuna y rápida, lo que le imprime valor y utilidad. Contar con aplicaciones desarrolladas en computadoras posibilita un mejor y más fácil acceso a la gestión de la información. Asimismo permite tomar la decisión más acertada en cada momento, clave del éxito que utiliza además de los conocimientos y las aptitudes del hombre, la tecnología informática actual.

El departamento de tesorería del ayuntamiento de la ciudad de Georgetown en Guyana cuenta con un sistema para el pago de propiedades, donde los propietarios de la ciudad están obligados a llegar a la oficina del ayuntamiento para pagar sus impuestos.

Debido a que muchos de ellos viven en el extranjero resulta difícil a la hora de saldar el pago de la propiedad a tiempo. Si esta información debe manejarse en un departamento de tesorería con grandes requerimientos, dotado de una red local y avanzados equipos de cómputo la solución puede ser el diseño de una intranet.

Los avances gigantescos de la informática en los últimos años, fundamentalmente dirigido a la seguridad de las comunicaciones, las nuevas concepciones desde el punto de vista de programación y el aumento de las necesidades de los usuarios, motivaron a iniciar un proyecto de intranet.

#### **Situación problemática**

En el departamento de tesorería de la ciudad de Georgetown tienen lugar varios procesos importantes, uno de ellos es la gestión de trabajadores y clientes del lugar que actualmente es realizada de manera manual, usando herramientas que ayudan en el

## Introducción

procesos como son documentos en Microsoft Word y tablas en Excel, por la gran cantidad de clientes se tiene almacenado volúmenes numerosos de información en formato duro, que carecen de seguridad, cuando hay necesidad de realizar una búsqueda se vuelve una tarea engorrosa. Todo esto provoca que la información esté duplicada y para realizar cualquier acción con dicha información es necesario una pérdida de tiempo el que aumentará con el paso de los años.

Otro proceso, y el más importante del lugar es el pago de los impuestos de propiedad, actualmente esta acción solo se puede realizar con la presencia del propietario en el departamento, muchos de los habitantes de la ciudad que a su vez son propietarios trabajan fuera del país y en la mayoría de las ocasiones no se encuentran en la fecha que deben pagar los impuestos, aplicándose un recargo que aumenta al pasar los días, para el ayuntamiento también repercute negativamente porque por la falta de ingresos se encuentra en dificultades financieras anualmente.

De la situación problemática antes descrita el **problema a resolver** es: La gestión de la información del departamento de tesorería de la ciudad de Georgetown es realizada de manera manual, y el pago de los impuestos de propiedad requiere de la asistencia personalizada del propietario.

### Idea a defender

Con la creación de un sistema informático que permita la gestión de la información del departamento de tesorería de la ciudad de Georgetown y el pago de los impuestos de propiedad online, se eleva la productividad en cuanto a la obtención e intercambio de información referente al proceso de pago de impuesto de propiedades.

A partir de esta situación problemática se define el **objetivo general** del presente trabajo:

Analizar el flujo de información referente al proceso de pago de impuesto de propiedades del departamento de tesorería, con lo cual se debe diseñar e implementar un sistema informático para la gestión de la información del

departamento de tesorería de la Ciudad de Georgetown y el pago de los impuestos de propiedad online.

De este objetivo general se desprenden los siguientes **objetivos específicos**:

- Realizar el estudio del flujo de información del proceso de pago de impuesto online de propiedades.
- Diseñar un modelo de datos que abarque el proceso de pago de impuesto online.
- Diseñar y establecer niveles de acceso a la información.
- Implementar el proceso de pago de impuesto online referido a los propietarios.
- Diseñar un sistema multiplataforma en cuanto a Sistema Operativo y Gestor de Base de datos se refiere.

Para cumplir estos objetivos se realizarán las siguientes tareas:

- Entrevistas a empleados y directivos del departamento de tesorería.
- Investigación y recopilación de recomendaciones en los temas de accesibilidad, autenticación y personalización de la información del proceso de pagos de impuestos de propiedades.
- Investigación y comprobación en el funcionamiento del mecanismo de autenticación a través de Internet.
- Análisis del diseño de la base de datos existente.
- Diseño y creación de la interfaz gráfica de la aplicación.
- Implementación de la aplicación.

La estructura del documento es la siguiente:

## **Introducción**

Capítulo I “Fundamentación Teórica”: Donde se hace un análisis crítico de las bibliografías utilizadas. Se describen las metodologías, softwares y lenguajes de programación además de hacer alusión al por qué de su utilización.

Capítulo II: Este capítulo se centra fundamentalmente en analizar el modelo de negocio, así como la descripción de dicho proceso utilizando los artefactos de UML.

Capítulo III “Construcción de la Solución Propuesta”: Se basa en los diferentes principios aplicados en el sistema, el tratamiento de ellos y el modelo de clases Web así como el diseño de trabajo de la base de datos con los diagramas de modelo lógico y físico de los datos y su implementación.

Capítulo IV: Estudio de Factibilidad. Se hace un análisis de las variables tiempo, costo y esfuerzo a través de su estimación mediante la variante ‘Análisis de Puntos de Casos de Uso’, se enuncian además los beneficios tangibles e intangibles que aporta el sistema.

### Capítulo I: Fundamentación Teórica

#### **1.1 Introducción**

En este capítulo se abordarán aspectos teóricos del tema que se va a analizar, exponiendo los principales conceptos asociados al dominio del mismo. Se describe el contexto donde se enmarca, las características y dificultades que lo acompañan, así como las metodologías y tecnologías a utilizar.

#### **1.2 Principales conceptos asociados al dominio del problema.**

El comercio electrónico ha sido una práctica realizada desde hace más de una década, siendo ejemplos, en su vertiente de gran consumo, las ventas de grandes almacenes mediante videotexto, los catálogos electrónicos y las tele tiendas dentro del comercio entre empresas deben señalarse las ventas que utilizan los protocolos EDI (Intercambio Electrónico de Datos) y la distribución de servicios turísticos y de transporte (SMD: Sistemas Mundiales de Distribución) entre compañías aéreas y agencias de viajes. Estos precursores tenían -y tienen- muchas restricciones de uso, como la necesidad de establecer un preacuerdos entre empresas, exigir detalladas normalizaciones de los sistemas informáticos y de gestión, usar costosas redes especiales ó servir solo para intercambios dentro de sectores específicos.

Por ello, y a pesar de estos amplios antecedentes, el término “Comercio electrónico” sólo ha sido asumido más recientemente, con la generalización del comercio a través de Internet. Este medio tiene como principales ventajas la resolución de muchas rigideces anteriores de las relaciones entre empresas, aportar una red de comunicaciones de uso general y más asequible a todo tipo de empresas, medios de pagos más flexibles y variados y usar procedimientos normalizados de seguridad.

Básicamente, el comercio electrónico puede definirse como la transacción en sí a través de medios electrónicos, entre los que se encuentra Internet. Una definición más

## Capítulo I: Fundamentación Teórica

abarcadora muestra el comercio electrónico como el aspecto que la empresa con presencia y actividad en Internet muestra hacia el exterior [1]

### **Ayuntamiento**

El ayuntamiento o corporación municipal es el órgano de administración de un municipio. Suele estar formado por un alcalde o presidente municipal que ostenta la presidencia de la administración local y del pleno y los concejales o ediles que reunidos en pleno ejercen la potestad normativa a nivel local.

Generalmente, el Ayuntamiento es el órgano administrativo menor y más cercano al ciudadano, aunque los municipios grandes suelen subdividirse administrativamente en barrios, distritos, delegaciones o pedanías (estas últimas de carácter más rural) [2].

Un impuesto sobre la propiedad es obligado por ser dueño de la misma. Existen tres tipos de de propiedad: tierra, mejoras a tierras (cosas inmobiliarias edificios) y propiedad personal (cosas mobiliarias). Los impuestos sobre la propiedad también se denominan impuestos inmobiliarios. Estos impuestos se pagan a la autoridad fiscal local o a la municipalidad. El monto que usted paga generalmente se puede deducir de sus impuestos federales sobre la renta. Los impuestos sobre la propiedad a menudo se gravan como un porcentaje del valor fiscal de su vivienda. Por ejemplo, si usted paga el 0.5% del valor fiscal en impuestos sobre la propiedad, una vivienda cuyo valor fiscal asciende a \$250,000 tendría un impuesto sobre la propiedad de \$1,250 [3].

En el caso del ayuntamiento de Georgetown el impuesto sobre la propiedad es calculado como un porcentaje del valor de la propiedad. Este porcentaje es aplicado en dependencia del uso de la propiedad.

### **Tarjetas de Crédito**

## Capítulo I: Fundamentación Teórica

Una tarjeta de crédito es una tarjeta de plástico con una banda magnética, a veces un microchip, y un número en relieve que sirve para hacer compras y pagarlas en fechas posteriores. Por su capacidad de realizar pagos se les llama también dinero plástico.

Entre las más conocidas del mercado están: Visa, American Express, MasterCard y Diners Club, entre otras.

Los usuarios tienen límites con respecto a la cantidad que pueden cargar, pero no se les requiere que paguen la cantidad total cada mes. En lugar de esto, el saldo (o "revolvente") acumula interés, y sólo se debe hacer un pago mínimo. Se cobran intereses sobre el saldo pendiente.

La mayor ventaja es la flexibilidad que le da al usuario, quien puede pagar sus saldos por completo cada mes o puede pagar en parte. La tarjeta establece el pago mínimo y determina los cargos de financiamiento para el saldo pendiente. Las tarjetas de crédito también se pueden usar en los cajeros automáticos o en un banco para servirse de un adelanto de efectivo aunque, a diferencia de las tarjetas de débito, se cobra un interés.

Un pago con tarjeta de crédito es un pago con dinero (dinero crediticio) y no es creado por los bancos centrales sino por los bancos privados al dar créditos. Por tanto, el hacer efectivo un cobro con tarjeta de crédito depende de la solvencia de la entidad emisora de la tarjeta.

Cuando se compra con una tarjeta de crédito, normalmente la entidad financiera carga el importe en cuenta a principios del mes siguiente sin intereses. También hay otra modalidad de pago en la que se paga una parte de las compras cada mes, pero, generalmente, tiene la desventaja de que se cobran intereses. En ambos casos, se suele cobrar una cuota anual.

Cuando se paga con tarjeta en el comercio, el cobrador suele pedir una identificación (identificación personal, permiso de conducir, etc.) y exige la firma del recibo para acreditar que se es propietario de la tarjeta. Existen algunas excepciones donde no se solicita firmar el recibo, a éste sistema se le denomina "autorizado sin firma" y suele

## **Capítulo I: Fundamentación Teórica**

utilizarse en comercios con grandes aglomeraciones de gente, como lo son cines, restaurantes de comida rápida y otros lugares similares.

En caso de uso fraudulento no hay más que anular el cargo y el banco debe demostrar que la compra ha sido hecha por el propietario.

Las compras con tarjeta de crédito pueden tener diversos seguros sobre el saldo financiado **[4]**.

### **Intranet**

Intranet es la aplicación de los estándares Internet dentro de un ámbito corporativo para mejorar la productividad, reducir costos y mantener los sistemas de información existentes. Es una forma de poner al alcance de los trabajadores todo el potencial informativo de una entidad, para resolver problemas, mejorar los procesos, construir nuevos recursos o mejorar los ya existentes, divulgar información de manera rápida y convertir a estos trabajadores en miembros activos de una red corporativa, o sea da al usuario la información que éste necesita para su trabajo. La Intranet pretende que cada cual tenga la información necesaria en el momento oportuno sin que tenga que recurrir a terceros para conseguirla. Una Intranet es una copia de Internet dentro de una entidad **[5]**.

### **Razones para implantar una Intranet**

Una Intranet permite a cualquier organización compartir eficazmente una gran cantidad de recursos de información que fluye de forma rápida y transparente desde las fuentes de origen hasta los puestos de trabajo de los empleados, esto con el mínimo de costo, tiempo y esfuerzo.

Teniendo en cuenta que los cambios organizativos son cada vez más frecuentes en el mundo actual, cada vez más dinámico y competitivo es preciso que la comunicación y el flujo de información sea lo más exacto y puntual posible. Surge entonces el concepto

## Capítulo I: Fundamentación Teórica

de Intranet como una respuesta a estos problemas, siendo una solución tecnológica que: permite la entrega de información bajo demanda y en el momento en que se requiere; garantiza la precisión y actualización de la información; asegura que la información se almacena en una única fuente lógica; y permite que la información sea mantenida por los propios departamentos o personas que la generaron.

Con el uso de una intranet se reducen de manera considerable los costos de distribución de la información, quedando obsoletos los mecanismos tradicionales, entiéndase correo ordinario, documentos, actas, normativas, proyectos, agendas comunes, etc., que pueden encontrarse centralizados en páginas web internas y ser compartidas por todos los interesados, que las consultarán en el momento más oportuno, siempre y cuando tengan acceso a la Intranet.

### Características de una Intranet

Las aplicaciones desarrolladas bajo el concepto de intranet poseen características propias, que la diferencian de otros productos comerciales, algunas de estas se listan a continuación [5].

- Rápida implantación.
- Escalable (se puede diseñar en función de las necesidades).
- Fácil navegación.
- Accesible a través de la mayoría de las plataformas informáticas del mercado.
- Puede integrar entornos distribuidos.
- Se puede añadir a fuentes de información propietarias (bases de datos, documentos realizados con procesador de texto, bases de datos de grupomática (groupware).
- Es extensible para aplicaciones con sonido, vídeo, interactivas, etc.

## Capítulo I: Fundamentación Teórica

- Comunicación Interactiva en línea.
- Correo Electrónico Inteligente
- Consultas a Bases de Datos
- Políticas de Administración de Documentos
- Motores de Búsqueda
- Integración a las redes locales
- Rápida visualización en línea de cualquier documento.
- Actualización, seguridad y privacidad
- Servicios Internet, FTP, WWW, Mail, Telnet, etc.
- Transacciones de información rápidas y seguras
- Integración con aplicaciones comerciales
- Seguridad en una intranet

En el desarrollo de este documento, la seguridad se refiere a las políticas, acciones, y sistemas necesarios para proteger la integridad de la información de los sistemas de comunicación. Varios niveles de seguridad son necesarios para cumplir estos objetivos. Generalmente, los requisitos de seguridad de la información están agrupados en las siguientes categorías:

***Integridad de la información:*** La garantía de que los datos no han sido alterados ni interceptados.

***Confidencialidad:*** La garantía de que sólo las personas a las que van dirigidas los datos acceden a éstos.

***Autenticación:*** La garantía de que el usuario o el grupo de trabajo que pide acceder a otro usuario, o grupo de trabajo, recurso o servicio es realmente ese usuario o grupo de

## Capítulo I: Fundamentación Teórica

trabajo. Además, la garantía de que la información descrita y asociada con el autor, o administrador, de un objeto digital no sea desconocida.

**Verificación:** Comprobar que los mecanismos de seguridad son sólidos, potentes y que están correctamente implementados.

**Disponibilidad:** Garantizar que los recursos estén disponibles cuando se necesiten.

**Valor estratégico y táctico:** Internet e Intranet presentan una serie de oportunidades muy interesantes para alcanzar una rica variedad de propósitos de comunicación de carácter estratégico y táctico. Si 1995 ha sido considerado el “año de Internet”, desde 1996 se configura la era Intranet. Miles de empresas de todo el mundo se han dado cuenta de que las redes internas pueden ser de gran utilidad., y esto supone mayor ventaja competitiva para la compañía, elevar la moral del trabajador y obtener más información de los clientes y proveedores

**Seguridad de la información:** La confidencialidad e integridad de la información ha de basarse en un estricto control de los accesos. Esto es posible mediante privilegios del sistema y de objetos. Los privilegios pueden ser encapsulados en roles.

Diversos controles de acceso contribuyen a establecer el privilegio mínimo, es decir, que el usuario tenga únicamente aquel privilegio que necesita para hacer su trabajo. La integridad de los datos se garantiza a través de mecanismos de consistencia de datos: para ejecutar una transacción hay que confirmar una serie de datos.

**Seguridad en los servidores:** Las empresas operan en Internet y almacenan datos de sus clientes en una base de datos que reside detrás de un cortafuego (*firewall*) dentro de una Intranet. El reto consiste en poder acceder a la información situada en la Intranet, protegida por el cortafuego, salvaguardando al mismo tiempo la confidencialidad e integridad de los datos.

Si la seguridad en la base de datos es un requisito previo y necesario para la seguridad en Internet, asegurar los datos frente a miradas acechantes mientras viajan a través de la red es también muy importante. Es posible cifrar mediante técnicas criptográficas la comunicación entre navegadores y servidores Web en Internet, o en una Intranet,

## Capítulo I: Fundamentación Teórica

utilizando el nivel SSL 3.0 (Secure Sockets Layer 3.0) y el sistema de encriptación SET (Secure Electronic Transaction).

En un sistema, la seguridad es un punto débil, es el corazón y la principal causa de fallos y pérdidas de la información. Para garantizar la seguridad en una Intranet se deben observar dos aspectos fundamentales: dónde y cómo están almacenados los datos de nuestra organización; para así garantizar hacia donde se van a mover y en que momento.

Las conexiones son otro punto esencial en la seguridad. Los datos viajan por usuarios particulares, en computadoras particulares, en determinados momentos y formas variables, por lo que una combinación usuario/ computadora/ momento/ aplicación puede ser un gran hueco suficientemente largo para comprometer la seguridad del sistema. Como solución a esto se pueden establecer entonces diferentes horarios para las salvadas y restauraciones de la información, o para las actualizaciones automáticas, que garanticen la no convergencia del personal con estos procesos.

### ***1.3 Descripción general del objeto de estudio***

El ayuntamiento de la ciudad de Georgetown Guyana es una organización no gubernamental que toma su mandato del capítulo 28:01 de las leyes de Guyana. Es responsable por el control y mantenimiento de la ciudad [4].

Es controlado por 30 oficiales elegidos por elecciones y varios empleados profesionales. Es dividido en seis departamentos que son:

TOWN CLERK'S (Administrador de la Ciudad)

CITY TREASURER'S (Tesorero de la Ciudad)

CITY ENGINEER'S (Ingeniero de la Ciudad)

CITY CONSTABULARY (Policía de la Ciudad)

PUBLIC HEALTH (Salud Pública)

MARKETS (Mercados)

### ***1.4 Flujo actual de los procesos y análisis crítico de la ejecución de estos.***

Uno de los procesos fundamentales que tienen lugar en el ayuntamiento de Georgetown es el cobro de impuestos a ciudadanos. Actualmente para poder pagar los impuestos es necesario asistir presencialmente, y después de estar en el ayuntamiento todo el proceso se realiza de forma manual, incurriendo en gran pérdida de tiempo, tanto para el personal del ayuntamiento como para los clientes. Además si los propietarios no se presentan a pagar sus impuestos en tiempo y fecha se acumula un interés. Para el ayuntamiento es un problema porque por la falta de ingreso se encuentra en dificultades financieras anualmente.

### ***1.5 Descripción de los sistemas existentes***

Después de haber realizado un análisis bibliográfico sobre los sistemas existentes relacionados con el pago de impuestos online se encontraron varios sistemas principalmente en Estados Unidos en la ciudad de Kansas y en el estado de Chicago (ciudad de Bushnell), ambos sistemas permiten pagar los impuestos de propiedades a través de la web, pero por las características de la estructura gubernamental no se adecuan a las especificaciones de Guyana, donde no existe ningún sistema que haga lo antes mencionado.

### ***1.6 Paradigmas de la programación***

#### **Programación por capas**

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y

## Capítulo I: Fundamentación Teórica

en caso de algún cambio, sólo se modifica el nivel requerido sin tener que revisar todo el código. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la Interfaz de Programación de Aplicaciones (Application Programming Interface - API por sus siglas en Inglés) que existe entre los niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas, teniendo en cuenta las ventajas que esta ofrece. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables es decir que pueden ampliarse con facilidad en caso de que las necesidades aumenten. El diseño más usado en la actualidad es el de tres capas **[5]**.

1. Capa de presentación
2. Capa de lógica de negocio
3. Capa de acceso a datos

1.- Capa de presentación: Esta capa provee la interfaz de usuario (IU). Aquí es donde la aplicación presenta información a los usuarios y acepta entradas o visualiza respuestas. Idealmente, la IU no desarrolla ningún procesamiento de negocios o reglas de validación de negocios. Por el contrario, la IU debería relegar sobre la capa de negocios para manipular estos asuntos. Esto es importante, especialmente hoy en día, debido a que es muy común para una aplicación tener múltiples IU.

2.- Capa de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio - LN) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos el almacenamiento o la recuperación de los mismos.

3.- Capa de datos: La capa de acceso a datos (AD) está formada por uno o más Sistemas Gestores de Bases de Datos (SGBD) que se encargan del almacenamiento y recuperación de los datos que necesita la capa de negocio.

Todas estas capas pueden residir en un mismo equipo. Lo más usual es que haya una multitud de computadoras donde resida la capa de presentación, es decir los clientes de la arquitectura cliente/servidor, y una computadora central que ejerce la función de servidor en dicha arquitectura. Las capas de negocio y de datos pueden residir en un mismo servidor y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o mas servidores. Así mismo, si el tamaño o la complejidad de la base de datos aumentara, se puede separar en varios servidores de datos, los cuales recibirán las peticiones del servidor donde reside la capa de negocio. Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en una o más computadoras que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de ordenadores sobre los cuales corre la base de datos. En este caso se está haciendo referencia a Sistemas de Aplicaciones Distribuidas.

En una arquitectura de tres niveles, los términos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas de forma física.

### ***1.7 Descripción de las Tecnologías a utilizar***

#### **Capa de presentación (IU)**

Para el desarrollo en esta capa existen muchas herramientas por ejemplo: Adobe Dreamweaver, Borlan C, C++, Microsoft FrontPage, etc. Pero para desarrollo de una

## Capítulo I: Fundamentación Teórica

interfaz de aplicaciones web es conveniente el uso de herramientas web por lo se realizo un análisis de Adobe Dreamweaver por sus características y de fácil utilización

Adobe Dreamweaver (Dw) Es una aplicación en forma de estudio (Basada por supuesto en la forma de estudio de Adobe Flash®) pero con más parecido a un taller destinado para la edición WYSIWYG de páginas web, creado inicialmente por Macromedia (actualmente es propiedad de Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Expression Web. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas

Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer, y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes [6].

### Lenguaje de desarrollo

#### HTML

HTML, no es un lenguaje de programación, es un lenguaje de especificación de contenidos para un tipo específico de documentos. Es decir, mediante HTML podemos especificar, usando un conjunto de etiquetas o tags, cómo se representa la información en un navegador o browser. Se centra en la representación en la pantalla de la información [7].

El HTML es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII. Las marcas más utilizadas suelen

## Capítulo I: Fundamentación Teórica

describirse por textos descriptivos encerrados entre signos de "menor" (<) y "mayor" (>), siendo lo más usual que exista una marca de principio y otra de final [7].

Este lenguaje nos permite aglutinar textos, sonidos e imágenes y combinarlos a nuestro gusto. Además, y es aquí donde reside su ventaja, HTML nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto o hipervínculos.

### **Hojas de estilo en cascada (CSS)**

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Por ejemplo, el elemento de HTML <H1> indica que un bloque de texto es un encabezamiento y que es más importante que un bloque etiquetado como <H2>. Versiones más antiguas de HTML permitían atributos extra dentro de la etiqueta abierta para darle formato (como el color o el tamaño de fuente). No obstante, cada etiqueta <H1> debía disponer de la información si se deseaba un diseño consistente para una página, y además, una persona que lea esa página con un navegador pierde totalmente el control sobre la visualización del texto [8].

### **JavaScript**

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas

## Capítulo I: Fundamentación Teórica

web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML [9].

Y **Adobe Photoshop 7.0** como herramienta para el diseño y creación grafica

Es una aplicación informática de edición y retoque de imágenes bitmap, jpeg, gif, etc, elaborada por la compañía de software Adobe inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.

A medida que ha ido evolucionando el software ha incluido diversas mejoras fundamentales, como la incorporación de un espacio de trabajo multicapa, inclusión de elementos vectoriales, gestión avanzada de color (ICM / ICC), tratamiento extensivo de tipografías, control y retoque de color, efectos creativos, posibilidad de incorporar plugins de terceras compañías, exportación para Web entre otros.

Photoshop se ha convertido, casi desde sus comienzos, en el estándar mundial en retoque fotográfico, pero también se usa extensivamente en multitud de disciplinas del campo del diseño y fotografía, como diseño Web, composición de imágenes bitmap, estilismo digital, fotocomposición, edición y grafismos de vídeo y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales [10].

Aunque el propósito principal es la edición fotográfica, este también puede ser usado para crear imágenes, efectos, gráficos y más con muy buena calidad [10].

### **Capa de lógica de negocio** (LN)

Es necesario en esta capa seleccionar un lenguaje de programación y un servidor de aplicaciones web. Teniendo en cuenta que existen varios lenguajes de programación web del lado del servidor (Server Side Scripts) como ASP, JSP, PHP entre otros, se analizaron las características fundamentales que permitieran seleccionar cual utilizar y en el caso de los servidores web se analizaron las características fundamentales de dos servidores: Internet Information Server (IIS) y Apache.

### **ASP**

Páginas Activas en el Servidor (Active Server Pages), es una tecnología creada por Microsoft, destinada a la creación de sitios Web. No se trata de un lenguaje de programación en sí mismo, sino de un marco sobre el cual construir aplicaciones basadas en Internet.

Las páginas ASP comienzan a ejecutarse cuando un usuario solicita un archivo .asp al servidor Web a través del explorador. El servidor Web llama a ASP, que lee el archivo solicitado, ejecuta las secuencias de comandos que encuentre y envía los resultados al explorador del cliente.

Puesto que las secuencias de comandos se ejecutan en el servidor y no en el cliente, es el servidor el que hace todo el trabajo necesario para generar las páginas que se envían al explorador. Las secuencias de comandos quedan ocultas a los usuarios, estos sólo reciben el resultado de la ejecución en formato HTML, reconocible por cualquier navegador.

ASP añade otra alternativa en sus posibles opciones para el desarrollo de las funcionalidades del lado del servidor. ASP le permite combinar HTML y código Script en el servidor para crear páginas Web dinámicas y altamente interactivas. [12].

### **JSP**

## Capítulo I: Fundamentación Teórica

JSP es un acrónimo de Java Server Pages, (Páginas de Servidor Java). Es una tecnología orientada a crear páginas web con programación en Java. Con JSP Capítulo I: Fundamentación Teórica 19 pueden crearse aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

### PHP

PHP (Profesional Home Pages) es un lenguaje de programación el cual se ejecuta en los servidores Web y que permite crear contenido dinámico en las páginas HTML, con un lenguaje propietario derivado del Perl.

PHP fue creado por Rasmus Lerdorf a finales de 1994, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó, Personal Home Page Tools [13].

Al principio, PHP sólo estaba compuesto por algunas macros que facilitaban el trabajo a la hora de crear una página Web. Hacia mediados de 1995 se creó el analizador sintáctico que se llamó PHP/F1 Versión 2 y sólo reconocía el texto HTML y algunas directivas de MySQL. A partir de este momento, la contribución al código fue pública. El crecimiento de PHP desde entonces ha sido exponencial y han surgido nuevas versiones continuamente hasta la actual PHP 5.

PHP dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones para Internet.

Es multiplataforma, funciona tanto para Unix, como para Windows de forma que el código que se haya creado para uno de ellos no tiene por qué modificarse al pasar al otro.

## Capítulo I: Fundamentación Teórica

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones, entre otras. La sintaxis que utiliza, la toma de otros lenguajes muy extendidos como C y Perl.



Figura 1. Esquema de representación del PHP [13]

El funcionamiento del PHP se puede describir a través de los pasos siguientes:

- Escribir en las páginas HTML pero con el código PHP dentro.
- Guardar la página en el servidor Web.
- Un navegador solicita una página al servidor.
- El servidor interpreta el código PHP.
- El servidor envía el resultado del conjunto de código HTML y el resultado del código PHP que también es HTML.

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes al usuario, el código PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

## Capítulo I: Fundamentación Teórica

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de Macromedia, a JSP/Java de Sun Microsystems, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE comercial llamado Zend Optimizer.

Es un lenguaje multiplataforma.

Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.

- Lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones) disponibles de forma gratuita en Internet.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia e incluida.

- No requiere definición de tipos de variables ni manejo detallado de bajo nivel [14].

### IIS

Internet Information Services (IIS) engloba una serie de herramientas administrativas que permite controlar sitios Web, FTP, SMTP (correo saliente) y Servicio de noticias (NNTP). Dispone también del soporte para crear páginas dinámicas (ASP), tecnología para el desarrollo de aplicaciones para Internet ampliamente extendida. IIS es un servidor de distribución gratuita, pero tiene como condición que corre solamente sobre plataforma Windows. Al igual que Apache posee mensajes de error configurables, y a diferencia del mismo si posee una interfaz gráfica que ayuda en la configuración del sitio en su totalidad.

### Servidor de Aplicaciones Web (Apache)

El servidor Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, carece de una interfaz gráfica que ayude en su configuración. En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache [15].

### Capa de acceso a datos

#### **Sistemas Gestores de Base de Datos**

Una Base de Datos (BD) es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo. **[16]**

Un **Sistema Gestor de Base de Datos** (SGBD) es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez. **[16]**

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

En el mercado existen un sinnúmero de gestores de base de datos, todos con sus características que lo hacen una opción a escoger, pero la elección, la mayoría de las veces, depende del cliente y no del desarrollador.

A continuación mostramos una tabla (Tabla 1) comparativa de algunos de los SGBD en cuanto a si son Multiplataforma o no, Gratis o no y en cuanto a la disponibilidad del código fuente.

	<b>Multiplataforma</b>	<b>Gratis</b>	<b>Código Fuente</b>
<b>MSSQL</b>	No	No	No

<b>MSDE</b>	No	Si	No
<b>MySQL</b>	Sí	Si	Sí
<b>Postgres</b>	Sí	Si	Sí
<b>Firebird</b>	Sí	Si	Si
<b>Interbase</b>	Si	No	Sí
<b>SyBase</b>	Si	No	No
<b>INFORMIX</b>	Sí	No	No
<b>Oracle</b>	Sí	no	No

Tabla 1. Comparación entre SGBD.

### **1.8 Lenguaje y Metodologías para la modelación.**

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar [17].

UML no puede compararse con la programación estructurada, pues UML significa (Lengua de Modelación Unificada), no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la orientación a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

### **Proceso Unificado de Desarrollo (RUP)**

El Proceso Unificado de Desarrollo (RUP, por su denominación en inglés Rational Unified Process), fue creado por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Esta metodología concibió desde sus inicios el uso de UML como lenguaje de modelado. Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software **[18]**.

A continuación se muestran estas prácticas.

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modelación del software visualmente
- Verifica la calidad del software.
- • Controla los cambios.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

### **1.9 Conclusiones**

Debido a la gran variedad de plataformas en cuanto a Sistema Operativo (SO) y en cuanto a soporte para datos (SGBD) se pretende con el presente trabajo elaborar un proyecto que permita la multiplataforma en ambos casos. Por lo tanto después de un exhaustivo análisis de las tecnologías antes expuestas llegamos a las siguientes conclusiones preeliminarias:

La solución que propone este proyecto se basa en la construcción de una INTRANET centrándose fundamentalmente el desarrollo de la primera versión del módulo de pago de impuesto para el departamento de tesorería

Se propone utilizar la programación orientada a objetos por los beneficios que está brinda, por lo que se considera RUP (Proceso Unificado de Desarrollo) la metodología más apropiada para el desarrollo del proyecto y UML como el lenguaje de modelación necesario en este caso.

## **Capítulo I: Fundamentación Teórica**

Debe estar basada en el modelo de programación de tres capas, que permite el desarrollo independiente del proyecto en cuanto a: Diseño de Interfaz, Lógica del Negocio y Acceso a Datos.

En el caso del servidor de aplicaciones, la opción del Servidor Apache es la idónea, por ser este servidor multiplataforma, gratuito y el más usado en Internet.

Una vez escogido Apache como servidor de aplicación, se selecciona PHP como el lenguaje para desarrollar los módulos necesarios, pues es este quien mejor funciona con el servidor escogido además de que no requiere de pago por su uso.

A nivel de la capa de datos, las opciones son varias y cualquiera de ellas resulta beneficiosa, por lo que se decide trabajar en función de lograr un sistema multiplataforma, aunque para el desarrollo concreto del proyecto se escoge MySQL.

Por último la elección de la interfaz de usuario, para la cual se puede utilizar cualquier explorador web que soporte HTML 4.0, debido a que el peso fundamental de la aplicación está del lado del servidor.

## Capítulo 2 – Modelo del negocio

### ***2.1 Introducción***

El presente trabajo está relacionado con el modelo del negocio de una sección dentro de un departamento del ayuntamiento de la ciudad de Georgetown Guyana. Para alcanzar su misión el ayuntamiento cuenta con seis (6) departamentos y una variedad grande de procesos. El foco de esta investigación es la sección de colección de impuestos que forma parte del departamento de finanzas. La sección tiene dos procesos fundamentales, la colección de impuestos de propiedad y la generación de la carta de la notificación de demanda. En el modelo del negocio se describen los actores, casos de usos y trabajadores, así como la relación entre ellos visualizada en el diagrama de caso de uso del negocio y el modelo de objetos, utilizando para su modelado el Lenguaje UML.

### ***2.2 Descripción de los procesos del negocio***

El modelado del negocio es una técnica que permite comprender los procesos de negocio de la organización. Está compuesto por dos tipos de modelos de UML, modelos de casos de uso y modelos de objetos [19].

A continuación se describe la actividad que se realiza en el departamento del ayuntamiento de la ciudad de Georgetown Guyana que constituye el proceso del negocio. Se describen además, las mejoras que se proponen al negocio actual indicando como se solucionarían las limitantes que originaron la situación problemática.

Entendiendo como proceso de negocio a un grupo de tareas relacionadas de manera lógica que se llevan a cabo en determinada secuencia y producen o manipulan una

colección de datos, empleando recursos del departamento; para dar resultados que apoyan sus objetivos; se identificaron los siguientes procesos del negocio: **[19]**

- El proceso de la colección de Impuestos.
- El proceso de generación de la carta de la notificación de demanda.

### **El proceso de la colección de Impuestos:**

Llega un propietario o un emisario de él y le dice al cajero que desea pagar el impuesto de su propiedad. El cajero busca la su información en el sistema existente y le dice al propietario los datos de la cuenta. Después que el propietario abona el debito de su cuenta, el cajero actualiza los datos en el sistema y genera un recibo reflejando la transacción realizada entregándole el mismo al propietario.

### **Situación problemática**

Esta metodología representa un gran consumo de tiempo y esfuerzo tanto para los empleados que operan el sistema existente como para los propietarios que llegan a pagar sus impuestos.

### **Mejoras Propuestas**

La información referente a los propietarios estará disponible vía Internet en un portal específicamente diseñado al efecto, de manera que desde cualquier lugar del mundo y en todo momento se podrá obtener información detallada y actualizada servicios del ayuntamiento, pudiendo pagar así su propiedad.

### **El proceso de generación de la carta de la notificación de demanda:**

Después de cada cuarto del año se genera una lista de los propietarios deudores de la ciudad. Los propietarios que se encuentran en la lista reciben una carta de aviso donde se les informa que tiene deuda y que debe contactar con la oficina para rectificar la situación.

Además esta lista es enviada al jefe de departamento para ayudar en el análisis del estado financiero del ayuntamiento.

### ***2.3 Reglas del negocio a considerar***

- Solo el supervisor puede cambiar el monto total que se debe pagar.
- Cualquiera persona puede pagar el impuesto para una propiedad dada.
- El cajero tiene un acceso limitado al programa que gestiona el pago.
- El ayuntamiento tiene derecho a confiscar la propiedad de un propietario deudor, y venderla para recuperar dicha deuda.
- Cada propiedad tiene un valor evaluado que le asigna el gobierno del país. El 40% de este valor es el monto total debido para el año completo. En el caso que la propiedad es utilizado para negocio, el monto debido es el 250% del valor evaluado. Si la propiedad no tiene ningún edificio la cifra debida será el 500% del valor evaluado.
- El año es dividido en 4 cuartos, teniendo 3 meses cada cuarto. El monto total debido por el año es dividido entre 4 y este valor debe ser pagado al principio del cuarto. El propietario tiene un período de gracia de un mes antes de que se empieza a acumular el interés, salvo el primer cuarto que tiene un período de gracia de 2 meses. O sea el impuesto debe ser pagable en cuatro instalaciones iguales y será debido el primer día de febrero o el día que recibe la carta del

aviso, el que sea el más tarde, el primer día de abril, el primer día de julio y el primer día de octubre.

- El interés es del tipo acumulado y es calculado diario.
- Un contribuyente puede pagar su impuesto antes de que sea debido.
- El ayuntamiento pueda aplicar un descuento no más que el 10% del impuesto debido si un propietario lo paga completamente antes de que sea debida la primera instalación.

### **2.4 Modelo de casos de uso del negocio**

El modelo de Casos de Uso del Negocio describe los procesos de negocio de una entidad en términos de casos de uso y actores del negocio en correspondencia con los procesos del negocio y los clientes, respectivamente. Este modelo presenta un sistema desde la perspectiva de su uso y esquematiza como proporciona valor a sus usuarios, permitiendo a los modeladores comprender mejor que valor proporciona el negocio a sus actores. [19]

#### **2.4.1 Actores del negocio**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos al negocio; con los que este interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [20]

<b>Actor del negocio</b>	<b>Justificación</b>
Propietario	Cualquier persona que llega al ayuntamiento en aras de pagar el impuesto de su propiedad.

Supervisor de Impuesto	Encargado de preparar y emitir aviso de demanda a los propietarios deudores.
------------------------	--

Tabla 2. Descripción de los actores del negocio

2.4.2 Diagramas de casos de uso del negocio

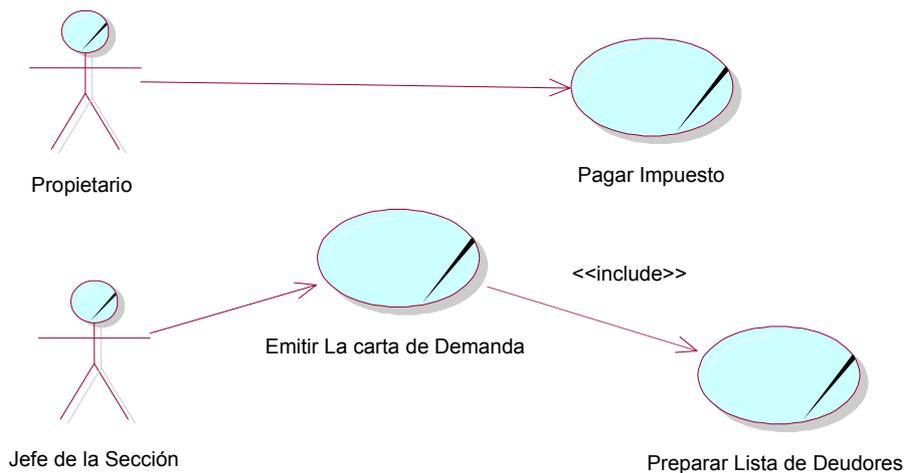


Figura 1. Diagramas de casos de uso del negocio

2.4.3 Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. [21]

Trabajador del negocio	Justificación
Cajero	Encargado de atender a clientes que llegan a al ayuntamiento para hacer el pago de su propiedad,

## Capítulo II: Modelo del Negocio

	consulta el sistema para saber la cifra a pagar por el propietario.
Jefe de la Sección	Se encarga de la organización y el tratamiento de las cuentas de los propietarios. Además es el encargado de la preparación e emisión del aviso de demanda.
Colector de Impuesto	Ayuda con la emisión del aviso de demanda y la entrega a los propietarios deudores para que el impuesto pueda ser pagado en una manera oportuna.
El Sistema	Tiene toda la información referente a los propietarios y la propiedad correspondiente, gestiona el pago del impuesto y genera recibo después de ser actualizado por el cajero.

**Tabla 3: Trabajadores del Negocio**

### 2.4.4 Descripción de los casos de uso del negocio

<b>Nombre del Caso de Uso</b>	Pagar impuesto
<b>Actores</b>	El propietario (inicia)
<b>Propósito</b>	Pagar Impuesto de la Propiedad
<b>Resumen</b>	El caso de uso es iniciado cuando el propietario pide al cajero la información relacionada con la propiedad a pagar. Después de recibir dicha información el propietario paga al cajero, este pago puede ser a través de cheque o en efectivo, el propietario obtiene un recibo de pago, culminando

## Capítulo II: Modelo del Negocio

	así el caso de uso.
<b>Casos de Usos Asociados : Ninguno</b>	
<b>Curso Normal de los eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del proceso de negocio</b>
1. El cliente se presenta en la oficina del ayuntamiento para pagar su impuesto.	1.1 El Cajero le pide al propietario los datos de la propiedad a pagar.
2. El cliente le entrega la información de la propiedad.	2.1 El cajero consulta al sistema para buscar la información solicitada por el cliente.  2.2 El cajero le informa al cliente la cantidad que debe pagar sobre su de su propiedad.
3. El cliente le entrega al cajero la cantidad que debe. Si no tiene dudas con la cantidad solicitada por el ayuntamiento.	3.1 Actualiza el sistema con el nuevo pago y genera un recibo.  3.2 El cajero entrega el recibo al cliente.
<b>Curso Alternativo de los eventos</b>	
Acción 2.1	El cajero no encuentra la propiedad en el sistema y por lo tanto el cliente no puede realizar el pago.
Acción 3	El cliente tiene dudas con la cantidad que debe pagar. El cajero debe llamar al jefe de la sección para aclarar las dudas del cliente. Después regresa a la acción 3 si todas las dudas han sido aclaradas. Si no, el propietario no realiza el pago.

<b>Prioridad</b>	
<b>Mejoras</b>	El proceso tiene la restricción de que un propietario debe presentarse a la oficina del ayuntamiento para realizar el pago. La propuesta es poner el servicio disponible en Internet.
<b>Otros</b>	

**Tabla 4: Descripción Caso de uso Pagar impuesto**

<b>Nombre del Caso de Uso</b>	Emitir carta de aviso	
<b>Actores</b>	Jefe de la sección (inicia)	
<b>Propósito</b>	Emitir La Carta de Aviso	
<b>Resumen</b>	El caso de uso es iniciado cuando el jefe de la sección ordena al cajero genera la lista de deudores, el cajero consulta al sistema para generar dicha lista, la cual se le entrega al jefe de la sección. Utilizando la misma el jefe de la sección prepara la carta de aviso para cada propietario deudor y se le entrega al colector de impuesto para llevar a los diferentes propietarios culminando así el caso de uso.	
<b>Casos de Usos Asociados : Ninguno</b>		
<b>Curso Normal de los eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del proceso de negocio</b>	
1. El jefe la sección ordene al cajero para generar la lista de deudores.	1.1 El Cajero consulta al sistema para generar la lista.	
2. El jefe de la sección prepara la carta	2.1 El colector de impuesto lleva la carta de	

de aviso y se le entrega al colector de impuesto para llevar a los propietarios.	aviso a los propietarios.
<b>Curso Alternativo de los eventos</b>	
Acción 1.1	No existen deudores, por tal motivo la carta de aviso no puede ser generada.
<b>Prioridad</b>	
<b>Mejoras</b>	La propuesta es mandar la carta al correo de cada usuario al principio del periodo de pago si tiene deudas
<b>Otros</b>	

Tabla 5: Descripción Caso de uso Emitir carta de aviso

#### 2.4.5 Diagramas de actividades del negocio

El diagrama de actividad es un gráfico que contiene los estados en que puede hallarse la actividad a analizar. Cada estado de la actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En resumen describe un proceso que explora el orden de las actividades que logran los objetivos del negocio [22].

# Capitulo II: Modelo del Negocio

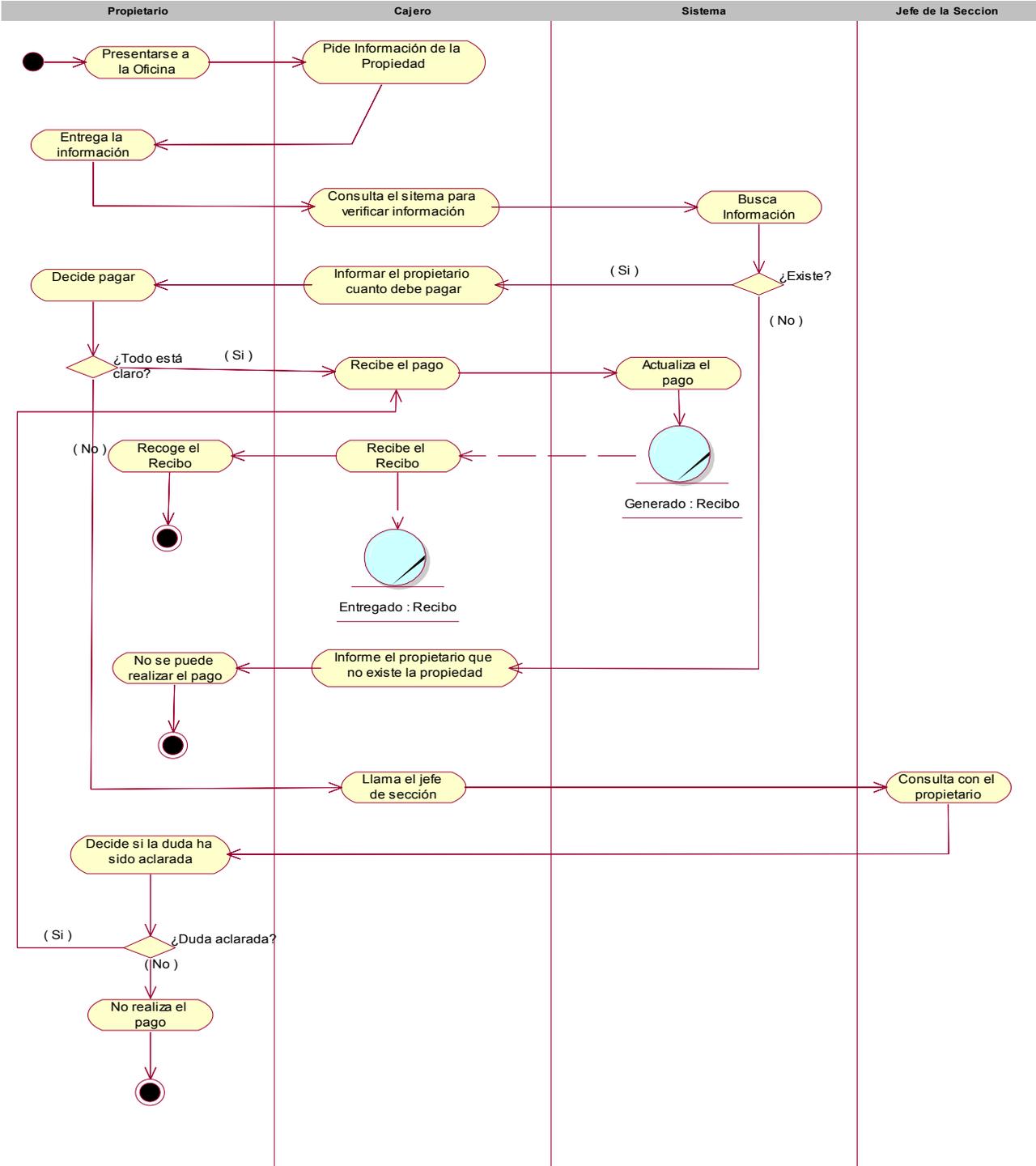


Figura 2. Diagramas de actividades del caso de uso Pagar Impuesto

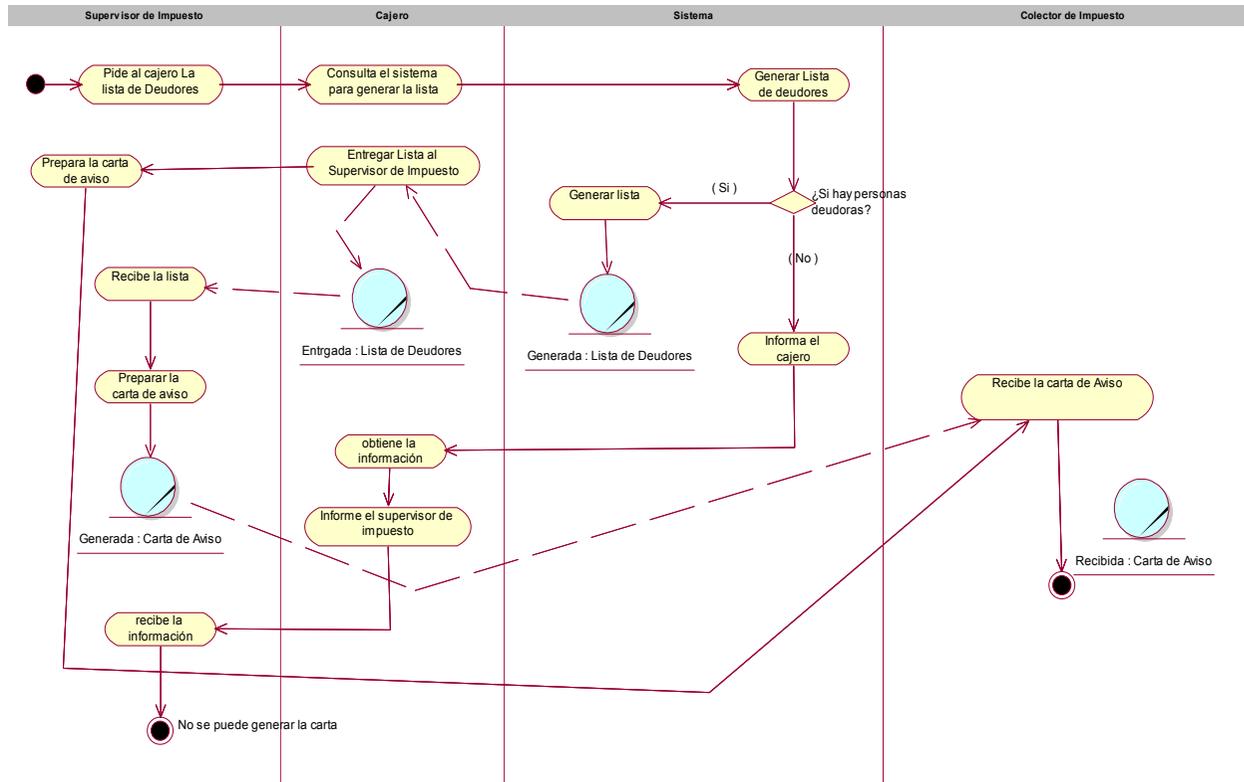


Figura 3. Diagramas de actividades del caso de uso Emitir la carta de aviso

### 2.4.6 Modelo de objetos del negocio

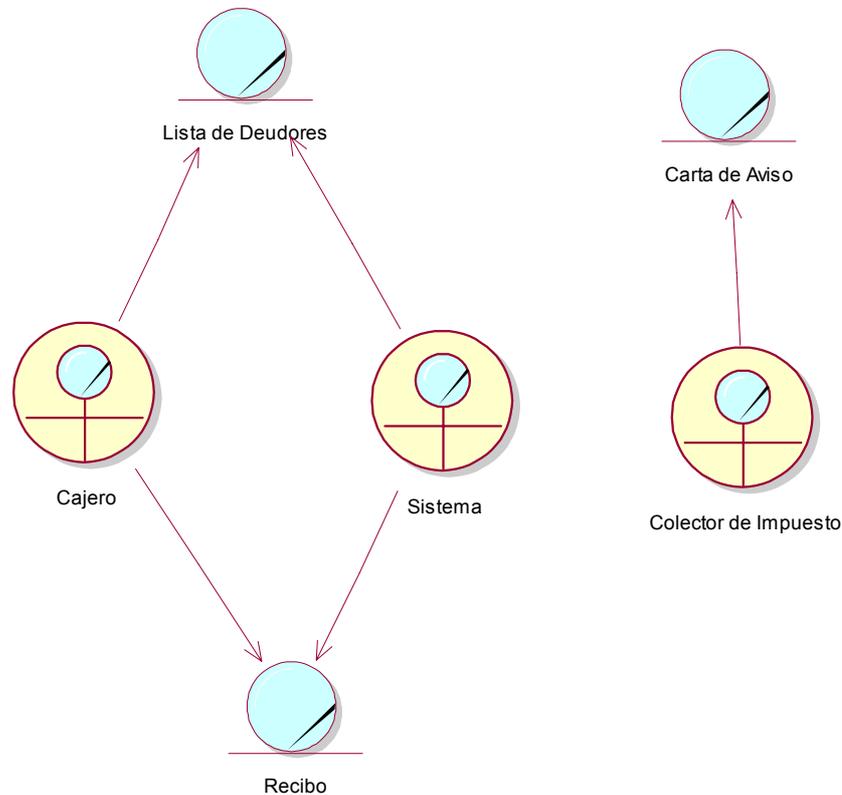


Figura 3. Diagramas de clases del modelo de objetos del negocio

## 2.5 Conclusiones

En este capítulo se describió el proceso del negocio de un departamento del ayuntamiento de la Ciudad de Georgetown Guyana., las reglas del negocio y se identificaron los actores y trabajadores del mismo, además se elaboraron los modelos de caso de uso, lográndose una mejor comprensión del negocio, sus limitaciones y las mejoras propuestas para el perfeccionamiento de su gestión.

## Capítulo 3: Construcción de la Solución Propuesta

### 3.1 Introducción

En este capítulo se hace una descripción detallada de los elementos necesarios para realizar un análisis de los requisitos funcionales del sistema.

### 3.2 Requerimientos funcionales

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. **[23]** Todo esto basándose en las necesidades de los usuarios y clientes.

Los requerimientos funcionales del sistema propuestos son los siguientes:

1. Autenticarse
2. Insertar propiedad
3. Modificar propiedad
4. Eliminar propiedad
5. Crear permiso al usuario
6. Quitar permiso de un usuario
7. Insertar usuario
8. Eliminar usuario
9. Modificar usuario
10. Habilitar usuario
11. Inhabilitar usuarios
12. Insertar propietario
13. Eliminar propietario

## Capítulo III: Construcción de la Solución Propuesta

14. Modificar propietario
15. Cambiar contraseña
16. Recuperar contraseña
17. Insertar noticia.
18. Eliminar noticia.
19. Modificar noticia
20. Insertar eventos.
21. Modificar eventos.
22. Eliminar eventos.
23. Modificar información de los departamentos
24. Insertar información de los departamentos.
25. Eliminar información de los departamentos.
26. Visualizar lista de empleados.
27. Visualizar estado de la cuenta.
28. Pagar la cuenta de la propiedad.
29. Visualizar eventos.
30. Visualizar noticia.
31. Generar reportes de deudores.
32. Generar reportes de transacciones del día.
33. Generar reportes de transacciones de un período dado.
34. Generar reportes de transacciones de los pagos hasta la fecha.

### ***3.3Requerimientos no funcionales***

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en

## Capítulo III: Construcción de la Solución Propuesta

cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema. [24]

Para el sistema propuesto se han definido los siguientes requisitos no funcionales:

### **Apariencia o interfaz externa**

El sistema debe tener una interfaz sencilla, amigable, muy legible y simple de usar, el producto debe ser autoritario e interactivo para que los usuarios se sientan confiados.

Se debe informar al usuario dónde está y qué puede hacer desde allí, al proporcionar señales de navegación que conduzcan al usuario hasta el contenido que desea y evitarle navegar a través de muchas áreas para ello. El contenido debe ser mostrado de manera comprensible para el usuario.

### **Usabilidad**

Los usuarios podrán interactuar con el sistema, aún teniendo pocos conocimientos del manejo de la computadora y del ambiente Web. Para su manejo se documentarán a través de la ayuda.

### **Rendimiento**

El sistema deberá ser rápido ante las solicitudes de los usuarios y en el procesamiento de la información, eso depende en gran medida del aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor y de la velocidad de las consultas a la base de datos. En el cliente se realizará la validación de los datos y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran.

## Capítulo III: Construcción de la Solución Propuesta

### **Requerimientos de Soporte**

Se documentará la aplicación con una ayuda dirigida a los diferentes tipos de usuarios para garantizar el soporte de la herramienta. Se realizará mantenimiento al sistema a fin de aumentar las funcionalidades del mismo a través de versiones posteriores. Los servicios de instalación y mantenimiento del sistema será responsabilidad del administrador del sistema en la entidad que sea utilizado.

### **Requerimientos de Seguridad**

La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización que puedan garantizar el cumplimiento de esto: cuenta, contraseña y nivel de acceso, de manera que cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios, garantizando así la confidencialidad.

Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas. Se guardará encriptada esta información en la base de datos utilizando para ello MD5 como algoritmo de encriptación.

Se harán validaciones de la información tanto en el cliente como en el servidor.

Se crearán usuarios con diferentes niveles de acceso al sistema. Se limitarán los permisos de los usuarios que ejecutan sentencias o consultas SQL. Se utilizarán, además, procedimientos almacenados dado que el modo en que se pasan los parámetros, evita el uso de inyección de código SQL.

## Capítulo III: Construcción de la Solución Propuesta

Estas medidas no afectarán el rendimiento de la aplicación.

### **Requisitos de Portabilidad.**

La plataforma seleccionada para desarrollar la aplicación fue Windows, pero puede ser ejecutada desde otras plataformas como Linux, que soporten el lenguaje PHP y MySQL

### **Requerimientos Políticos**

La aplicación debe cumplir con lineamientos, políticos y/o regulaciones de la entidad que utilice el sistema.

### **Requerimientos de Confiabilidad**

El sistema debe ser tolerante ante los fallos y las operaciones a realizar deben ser transaccionales.

### **Ayuda y documentación en línea**

Tendrá una documentación básica que comprenda los aspectos generales a tener en cuenta.

Dispondrá de una ayuda bien detallada sobre las principales opciones del sistema. Además.

### **Requerimientos de Software**

La aplicación debe poderse ejecutar en entornos *Windows, Linux, etc.* (Multiplataforma), para su ejecución del lado del servidor necesita MySQL como sistema gestor de base de datos y Apache como servidor Web, del lado del cliente cualquiera de los exploradores Web existentes en el mercado.

### Requerimientos de Hardware

Se requiere de una computadora como servidor de base de datos con los requerimientos de hardware que necesita MySQL. Las terminales clientes solo requerirán de una computadora conectada a la red, para poder ejecutar los navegadores de Web al menos deben cumplir los requisitos mínimos (que requiera el navegador en cuestión)

### **3.4 Modelo de casos de uso del sistema**

“El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.” [25]

“Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones.”

“El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario.”

#### **3.4.1 Actores del sistema**

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interactúan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que hemos identificado los actores del sistema, tenemos identificado el entorno externo del sistema [26].

<b>Nombre del actor</b>	<b>Descripción</b>
Empleado	Persona que puede registrarse en el sistema. Podrá

### Capítulo III: Construcción de la Solución Propuesta

	visualizar, lista de empleados, revisar noticias y eventos que van a estar publicadas en la intranet.
Administrador Del Sitio	El encargado de administrar el sistema (Super-Administrador). Gestiona noticias, eventos, usuarios del sistema y la información de los departamentos.
Administrador de Finanzas	Persona que gestiona la información de los pagos de propiedades de los clientes y genera los reportes del sistema.
Propietario	Es la persona que chequea el estado de su cuenta y paga su propiedad.

**Tabla 6: Descripción de los actores del sistema**

#### 3.4.2 Caso de Uso del Sistema

Cada forma en que los actores usan el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. [27].

### Capitulo III: Construcción de la Solución Propuesta

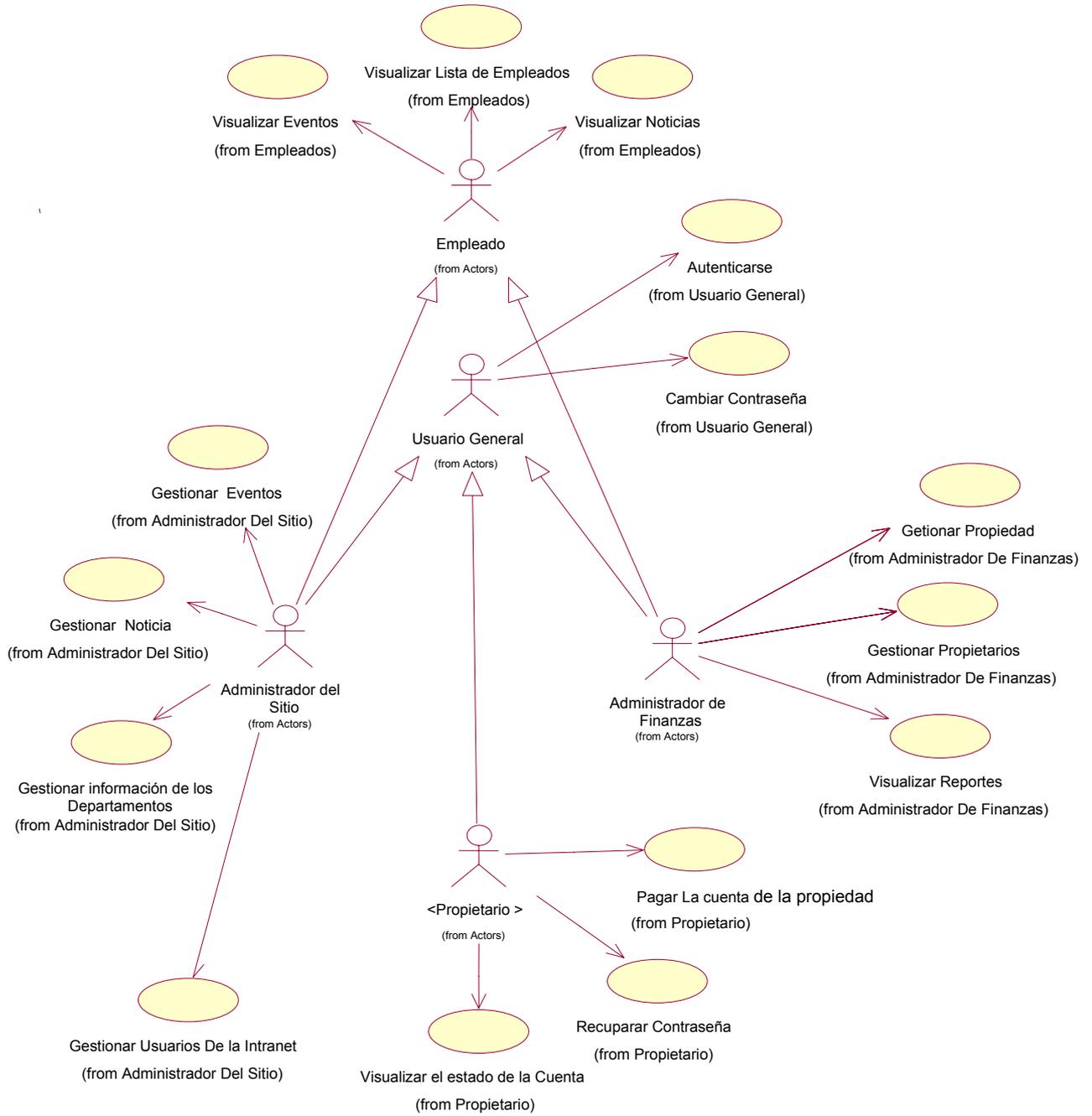


Figura 4: Diagrama de caso de Uso del Sistema.

## Capítulo III: Construcción de la Solución Propuesta

### 3.4.3 Descripción de los casos de uso del sistema

<b>Caso de uso</b>	Autenticarse
<b>Actores</b>	Usuario General (inicia)
<b>Propósito</b>	Iniciar sesión en el sistema.
<b>Resumen</b>	<p>El caso de uso se inicia cuando el usuario desea acceder a la información o servicios que se brindan en el sistema. Debe introducir su nombre de usuario y contraseña. El sistema verifica en la base de datos si los datos introducidos son correctos y pasa a mostrar su perfil, culminando así el caso de uso.</p>
<b>Referencias</b>	R1
<b>Precondiciones</b>	Para poder iniciar sesión debe existir el nombre de usuario y que su contraseña sea válida para el sistema.
<b>Post-condiciones</b>	El usuario puede acceder a toda la información a la que tiene permisos.
<b>Prototipo</b>	Ver Anexo 2.1.1, 2.1.2, 2.1.3.

**Tabla 7: Descripción del caso de uso Autenticarse**

Caso de uso	Recuperar la contraseña
Actores	Propietario (inicia)
Propósito	Permitir que el propietario recupere su contraseña si fue olvidada.
<b>Resumen</b>	<p>El caso de uso se inicia cuando un propietario desea recuperar su contraseña olvidada.</p>

### Capítulo III: Construcción de la Solución Propuesta

El sistema pide al usuario la respuesta de una pregunta secreta. Si la respuesta es correcta el propietario puede introducir una nueva contraseña.	
<b>Referencias</b>	R16
<b>Precondiciones</b>	Debe existir información almacenada del propietario para poder recuperar la contraseña
<b>Post-condiciones</b>	La contraseña queda cambiada en la base de datos
<b>Prototipo</b>	Ver anexo 2.2

**Tabla8: Descripción del caso de uso Recuperar la contraseña**

<b>Caso de uso</b>	Cambiar Contraseña
<b>Actores</b>	Usuario General (inicia)
<b>Propósito</b>	Permitir que la persona registrada cambie su contraseña.
<b>Resumen</b>	
El caso de uso se inicia cuando un usuario registrado desea cambiar su contraseña, el sistema le da la posibilidad de poner su cuenta, la contraseña antigua y la nueva, si los datos son correctos la contraseña queda cambiada, de lo contrario se muestra un mensaje de error, finalizando así el caso de uso.	
<b>Referencias</b>	R15
<b>Precondiciones</b>	Debe existir información almacenada de este usuario.
<b>Post-condiciones</b>	Queda modificada la contraseña del usuario.
<b>Prototipo</b>	Ver anexo 2.3

**Tabla 9: Descripción del caso de uso Cambiar Contraseña.**

### Capítulo III: Construcción de la Solución Propuesta

<b>Caso de uso</b>	Gestionar Noticias
<b>Actores</b>	Administrador del sitio (inicia)
<b>Propósito</b>	Permite mantener las noticias actualizada.
<b>Resumen</b>	<p>El caso de uso se inicia cuando el administrador del sitio actualiza, inserta o elimina noticias. En el caso que se actualiza o elimina se hace una búsqueda de la noticia, se selecciona y se ejecuta la acción. El caso de uso termina con la actualización de las Noticias.</p>
<b>Referencias</b>	R17, R18, R19
<b>Precondiciones</b>	Si se va a actualizar o eliminar noticias, debe estar en la base de datos.
<b>Post-condiciones</b>	<p>Si acción: insertar, se insertan noticias.</p> <p>Si acción: actualizar, se actualizan las noticias</p> <p>Si acción: eliminar, se eliminan las noticias</p>
<b>Prototipo</b>	Ver Anexo 2.4.1, 2.4.2, 2.4.3.

**Tabla 10: Descripción del caso de uso Gestionar Noticias.**

<b>Caso de uso</b>	Gestionar Eventos
<b>Actores</b>	Administrador del sitio (inicia)
<b>Propósito</b>	Mantener los Eventos de los departamentos

### Capítulo III: Construcción de la Solución Propuesta

	actualizados.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sitio actualiza, inserta o elimina Eventos. En el caso que se actualiza o elimina se hace una búsqueda del evento, se selecciona y se ejecuta la acción. El caso de uso termina con la actualización o inserción del evento.
<b>Referencias</b>	R20, R21, R22
<b>Precondiciones</b>	Si se va a actualizar o eliminar eventos, debe estar en la base de datos.
<b>Post-condiciones</b>	Si acción: insertar, se insertan eventos. Si acción: actualizar, se actualizan insertan eventos. Si acción: eliminar, se eliminan insertan eventos.
<b>Prototipo</b>	Ver Anexo 2.5.1, 2.5.2, 2.5.3

**Tabla 11: Descripción del caso de uso Gestionar Eventos.**

<b>Caso de uso</b>	Gestionar Usuarios
<b>Actores</b>	Administrador del sitio (inicia)
<b>Propósito</b>	Mantener los usuarios del sistema actualizado, inserta, eliminar, modifica, habilitar/inhabilitar, crear y quitar permisos a los mismos.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sitio modifica, inserta, eliminar,

### Capítulo III: Construcción de la Solución Propuesta

<p>habilitar/inhabilitar, crea y quitar permisos a los usuario que interactúan con el sistema. En el caso de modificar, habilitar/inhabilitar, eliminar, crear y quitar permisos, el administrador debe hacer la búsqueda del usuario luego de encontrar la información seleccionarla y ejecutar la acción, en caso de insertar se introducen los datos necesarios. El caso de uso termina cuando el administrador ejecuta la acción seleccionada.</p>	
<b>Referencias</b>	R5, R6, R7, R8, R9, R10, R11
<b>Precondiciones</b>	<p>Si se va a modifica, eliminar, habilitar/inhabilitar, crea y quitar permisos de los usuarios, deben estar en la base de datos.</p> <p>Si se va a insertar debe verificarse que el usuario no este en la base de datos.</p>
<b>Post-condiciones</b>	<p>Si acción: insertar, queda registrada la cuenta de usuario.</p> <p>Si acción: actualizar, quedan modificado los datos del usuario.</p> <p>Si acción: eliminar, queda eliminado el usuario.</p> <p>Si acción: habilitar, queda activada la cuenta del usuario.</p> <p>Si acción: inhabilitar, queda desactivada la cuenta del usuario.</p> <p>Si acción: crear permiso, queda asignado permiso al usuario.</p> <p>Si acción: quitar permiso, se quitan el permiso asignado al usuario.</p>

### Capítulo III: Construcción de la Solución Propuesta

<b>Prototipo</b>	Ver anexo 2.6.1, 2.6.2, 2.6.3
------------------	-------------------------------

**Tabla 12: Descripción del caso de uso Gestionar Usuarios.**

<b>Caso de uso</b>	Gestionar Información Departamentos
<b>Actores</b>	Administrador del sitio (inicia)
<b>Propósito</b>	Permite mantener actualizada la información de los departamentos.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sitio actualiza, inserta o elimina información de los departamentos. En el caso que se actualiza o elimina se hace una búsqueda de la misma, se selecciona y se ejecuta la acción
<b>Referencias</b>	R23, R24, R25
<b>Precondiciones</b>	Si se va a actualizar o eliminar departamentos, debe estar en la base de datos.  Si se va a insertar debe verificarse que el departamento no este en la base de datos
<b>Post-condiciones</b>	Se actualiza los departamentos.  Si acción: insertar, se inserta Departamento.  Si acción: actualizar, se actualiza un Departamento.  Si acción: eliminar, se elimina un Departamento.
<b>Prototipo</b>	Ver anexo 2.7.1, 2.7.2, 2.7.3

**Tabla 13: Descripción del caso de uso Gestionar Departamentos.**

### Capítulo III: Construcción de la Solución Propuesta

<b>Caso de uso</b>	Gestionar Propiedad
<b>Actores</b>	Administrador de Finanzas (inicia)
<b>Propósito</b>	Mantener las propiedades del sistema actualizadas, inserta, eliminar, modifica, a las mismas.
<b>Resumen</b>	<p>El caso de uso se inicia cuando el administrador de finanzas modifica, inserta, eliminar, a las propiedades. En el caso de modificar, eliminar, el administrador debe hacer la búsqueda de la propiedad luego de encontrar la información seleccionarla y ejecutar la acción, en caso de insertar se introduce los datos necesarios.</p>
<b>Referencias</b>	R2, R3, R4
<b>Precondiciones</b>	<p>El administrador debe estar autenticado y debe existir dicha propiedad, si quiere actualizar o eliminarla.</p> <p>Si se va a insertar debe verificarse que la propiedad no este en la base de datos.</p>
<b>Post-condiciones</b>	<p>Si acción: insertar, queda insertada la propiedad en la base de datos</p> <p>Si acción: actualizar, quedan modificado los datos de la propiedad.</p> <p>Si acción: eliminar, queda eliminada la propiedad.</p>
<b>Prototipo</b>	Ver Anexo 2.8.1, 2.8.2, 2.8.3

### Capítulo III: Construcción de la Solución Propuesta

Tabla 14: Descripción del caso de uso Gestionar Propiedad.

<b>Caso de uso</b>	Gestionar Propietarios
<b>Actores</b>	Administrador de Finanzas (inicia)
<b>Propósito</b>	Mantener los propietarios del sistema actualizado, inserta, eliminar, modifica, a los mismos.
<b>Resumen</b>	<p>El caso de uso se inicia cuando el administrador de finanzas modifica, inserta, eliminar, a un propietario. En el caso de modificar, eliminar, el administrador debe hacer la búsqueda del propietario luego de encontrar la información seleccionarla y ejecutar la acción, en caso de insertar se introduce los datos necesarios. El caso de uso termina cuando el administrador ejecuta la acción seleccionada.</p>
<b>Referencias</b>	R12, R13, R14
<b>Precondiciones</b>	<p>El administrador debe estar autenticado y que existe un propietario si quiere actualizar o eliminarlo.</p> <p>Si se va a insertar debe verificarse que el propietario no este en la base de datos.</p>
<b>Post-condiciones</b>	<p>Se actualizan los Propietarios.</p> <p>Si acción: insertar, se inserta un Propietario.</p> <p>Si acción: actualizar, se actualiza un Propietario.</p> <p>Si acción: eliminar, se elimina un Propietario.</p>

### Capítulo III: Construcción de la Solución Propuesta

<b>Prototipo</b>	Ver anexo 2.9.1, 2.9.2, 2.9.3
------------------	-------------------------------

**Tabla 15: Descripción del caso de uso Gestionar Propietarios.**

<b>Caso de uso</b>	Visualizar estado de la Cuenta
<b>Actores</b>	Propietario (inicia)
<b>Propósito</b>	Permitir que la persona registrada visualice la información de su cuenta.
<b>Resumen</b>	<p>El caso de uso se inicia cuando un propietario registrado desea visualizar su información. El propietario debe pasar al sistema el número de la propiedad, que debe saber como propietario. El sistema muestra todos los datos necesarios para realizar el pago como el total anual a pagar, el interés, y el total para el periodo a pagar, finalizando así el caso de uso.</p>
<b>Referencias</b>	R27
<b>Precondiciones</b>	Debe existir información almacenada de esta propiedad en la base de datos para poder visualizarlo.
<b>Post-condiciones</b>	
<b>Prototipo</b>	Ver anexo 2.10

**Tabla 16: Descripción del caso de uso Visualizar estado de la Cuenta.**

<b>Caso de uso</b>	Pagar la cuenta de la propiedad
<b>Actores</b>	Propietario (inicia)
<b>Propósito</b>	Permitir que el propietario pague la cuenta de su

### Capítulo III: Construcción de la Solución Propuesta

	impuesto
<b>Resumen</b>	
<p>El caso de uso se inicia cuando un propietario desea realizar el pago de su impuesto. El propietario debe pasar al sistema el número de la propiedad, que debe saber como propietario. El sistema muestra todos los datos necesarios para realizar el pago como el total anual a pagar, el interés, y el total para el periodo a pagar, finalizando así el caso de uso.</p>	
<b>Referencias</b>	R28
<b>Precondiciones</b>	Debe existir información almacenada de esta propiedad.
<b>Post-condiciones</b>	La cuenta queda actualizada si el pago fue realizado
<b>Prototipo</b>	Ver anexo 2.11

**Tabla 17: Descripción del caso de uso Pagar la cuenta de la propiedad.**

<b>Caso de uso</b>	Visualizar lista de empleados
<b>Actores</b>	Administrador del sitio, Administrador de finanzas, Empleado (inicia)
<b>Propósito</b>	Permite visualizar los empleados del ayuntamiento.
<b>Resumen</b>	
<p>El caso de uso se inicia cuando el Administrador de finanzas, el Administrador del sitio o un empleado desean visualizar los empleados con datos de interés sobre el mismo.</p>	
<b>Referencias</b>	R26,
<b>Precondiciones</b>	Que exista trabajadores en la base de datos.

### Capítulo III: Construcción de la Solución Propuesta

<b>Post-condiciones</b>	
<b>Prototipo</b>	Ver anexo 2.13

**Tabla 18: Descripción del caso de uso Visualizar lista de empleados**

<b>Caso de uso</b>	Visualizar Noticias
<b>Actores</b>	Administrador del sitio, Administrador de finanzas, Empleado
<b>Propósito</b>	Permite visualizar las noticias publicadas.
<b>Resumen</b>	El caso de uso se inicia cuando el Administrador de finanzas, el Administrador del sitio o un empleado desean visualizar las noticias publicadas en la intranet del ayuntamiento.
<b>Referencias</b>	R30
<b>Precondiciones</b>	Que exista Eventos en la base de datos.
<b>Post-condiciones</b>	
<b>Prototipo</b>	Ver Anexo 2.14

**Tabla 19. Descripción del caso de uso de sistema Visualizar Noticias**

<b>Caso de uso</b>	Visualizar Eventos
<b>Actores</b>	Administrador del sitio, Administrador de finanzas, Empleado (inicia)

### Capítulo III: Construcción de la Solución Propuesta

<b>Propósito</b>	Permite visualizar las eventos
<b>Resumen</b>	
El caso de uso se inicia cuando el Administrador de finanzas, el Administrador del sitio o un empleado desean visualizar eventos publicados en la intranet del ayuntamiento .	
<b>Referencias</b>	R29
<b>Precondiciones</b>	Que exista Eventos en la base de datos.
<b>Post-condiciones</b>	
<b>Prototipo</b>	Ver Anexo 2.15

**Tabla 20: Descripción del caso de uso de sistema Visualizar Eventos**

<b>Caso de uso</b>	Generar Reportes
<b>Actores</b>	Administrador de Finanzas (inicia)
<b>Propósito</b>	Permite que el administrador de finanzas visualizar los reportes de los pagos.
<b>Resumen</b>	
El caso de uso se inicia cuando el administrador de finanzas desea acceder a la información sobre los propietarios deudores, los pagos para el día, los pagos de un Período dado, los pagos desde el principio del año hasta la fecha actual.	
<b>Referencias</b>	R31, R32, R33, R34, R35
<b>Precondiciones</b>	Debe estar registrado el administrador de finanzas para ver cualquiera de los reportes.

### Capítulo III: Construcción de la Solución Propuesta

<b>Post-condiciones</b>	El administrador de finanzas puede visualizar cualquier de los reportes.
<b>Prototipo</b>	Ver anexo 2.12

Tabla 21: Descripción del caso de uso de sistema Generar Reportes

#### ***3.5 Principios de Diseño del Sistema***

El diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física [28].

#### ***3.6 Estándares en la Interfaz de la Aplicación***

La primera impresión del usuario cuando visita una aplicación web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta confortable, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc., que a continuación se detallan. En el sistema, el diseño de la interfaz está basado en páginas Web, se utilizan las tonalidades suaves y refrescantes. El vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios.

Se utilizan imágenes identificativas como vínculos para la navegación dentro del sitio web. La letra utilizada en todo el sistema es verdana (12, 16) lográndose un diseño estándar en todo el sitio. Los mensajes de error son pequeños.

### ***3.7 Tratamiento de Excepciones***

El diseño de la interfaz ha estado dirigido a evitar errores, teniendo en cuenta paralelamente la creación de interfaces útiles y amigables. Se ha buscado simplificar la validación de los datos garantizando una validación intrínseca de los mismos, procurando facilitar la corrección de errores lógicos tanto en la introducción de la información como en cualquier otro momento del tratamiento de la misma.

La técnica para el manejo de los errores en el sistema se concebirá de manera que cuando ocurra un error se genere una excepción; es decir, la ejecución normal se detenga y se transfiera el control a la zona de tratamiento de excepciones. Las excepciones internas se generan automáticamente por el sistema.

Los mensajes de error que emita el sistema ya sea de la base de datos o de la aplicación cliente se captarán y se traducirán a un lenguaje comprensible para el usuario.

Los formularios manejan los datos en memoria y solo se actualiza en la base de datos cuando se indique salvarlos.

### ***Modelo de Clases Web***

Un diagrama de clases Web representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. [29] Al tratar de utilizar el diagrama de clases tradicionales para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

### Capítulo III: Construcción de la Solución Propuesta

<b>Caso de Uso</b>	<b>Diagrama de Clases Web</b>
Autenticarse	Anexo 1- Diagrama 1.1
Recuperar la contraseña	Anexo 1- Diagrama 1.2
Cambiar Contraseña	Anexo 1- Diagrama 1.3
Gestionar Noticias	Anexo 1- Diagrama 1.4
Gestionar Eventos	Anexo 1- Diagrama 1.5
Gestionar Usuarios	Anexo 1- Diagrama 1.6
Gestionar Información Departamentos	Anexo 1- Diagrama 1.7
Gestionar Propiedad	Anexo 1- Diagrama 1.8
Gestionar Propietarios	Anexo 1- Diagrama 1.9
Visualizar estado de la Cuenta	Anexo 1- Diagrama 1.10
Pagar la cuenta de la propiedad	Anexo 1- Diagrama 1.11
Visualizar lista de empleados	Anexo 1- Diagrama 1.12
Visualizar Noticias	Anexo 1- Diagrama 1.13
Visualizar Eventos	Anexo 1- Diagrama 1.14
Generar Reportes	Anexo 1- Diagrama 1.15

**Tabla 22: Tabla diagramas de clases de diseño.**

### ***3.8 Diseño de la base de datos.***

En este epígrafe se muestra el diseño de la base de datos del sistema propuesto a través de los diagramas de clases persistente y el esquema de la base de datos generados a partir de este, con el modelo de datos.

#### **3.8.1 Diagrama del modelo lógico de datos.**

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo [30]. Ver Anexo 3.1.

#### **3.8.2 Diagrama del modelo físico de datos.**

El modelo de datos que muestra la estructura física de las tablas de la base de datos, obtenido a partir del diagrama de clases persistentes [31], Ver Anexo 3.2.

### ***3.9 Diagrama de Implementación***

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros [32]. Ver Anexo 3.3

### ***3.10 Conclusiones***

En este capítulo han sido descritos los casos de uso del sistema, así como los actores que están asociados con ellos, utilizando el formato de alto nivel. Se hace mención de las precondiciones y poscondiciones del caso de uso.

### **Capitulo III: Construcción de la Solución Propuesta**

También, en el presente capítulo se muestran los diagramas de clases de la aplicación realizada, se describe el diseño de la base de datos y el diagrama de implementación, además, los principios de diseño seguidos, específicamente, los temas de estándares de la interfaz, concepción del tratamiento de excepciones. Se mostraron también los resultados de las etapas de diseño e implementación del sistema propuesto.

## Capítulo IV Estudio de Factibilidad

### **4.1 Introducción**

Se hace un análisis de las variables tiempo, costo y esfuerzo a través de su estimación mediante la variante 'Análisis de Puntos de Función', se enuncian además los beneficios tangibles e intangibles que aporta el sistema.

Los estudios de factibilidad son investigaciones altamente enfocadas en un proyecto de diseño propuesto. Estos son diseñados para producir información crítica sobre la viabilidad de negocios concernientes al proyecto, mientras que solamente se compromete una fracción de los recursos que serán necesarios para realizar el esfuerzo de un diseño completo de un solo golpe [33]. Para el análisis y cálculo de los costos se ha utilizado el modelo COCOMO II.

### **4.2 Planificación por puntos de función**

La técnica de puntos de función [34] fue introducida por Albrecht [Albrecht, 1979] y su propósito es medir el software cualificando la funcionalidad que proporciona externamente, basándose en el diseño lógico del sistema.

De acuerdo al Análisis de Puntos de Función, tanto las Transacciones (Entradas Externas, Salidas Externas, Consultas Externas) como los Archivos (Archivos Lógicos Internos, Archivos de Interfaz Externos) deben ser clasificados con una complejidad Baja, Media o Alta. Para calcular la cantidad de instrucciones fuentes hay que tener en cuenta la conversión al PHP, MySQL Y Java Script, herramientas seleccionadas para implementar este software, es de 44, 40 y 58 puntos respectivamente.

### **Entradas Externas:**

Se definen como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro. El Actor del Caso de Uso provee datos al sistema, los cuales pueden tratarse de información para agregar, modificar o eliminar de un Archivo Lógico Interno, o bien información de control o del negocio [34].

### **Salidas Externas:**

Se definen como un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (esto es, datos que se calculan a partir de otros datos) cruzan la frontera del sistema desde adentro hacia afuera. Adicionalmente, las Salidas Externas pueden actualizar un Archivo Lógico Interno. Los datos crean reportes o archivos que se envían hacia el Actor del Caso de Uso (que puede ser un humano u otro sistema). Estos reportes y archivos se crean desde uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos [34].

### **Ficheros lógicos Internos:**

Grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de las Entradas Externas. [IFPUG, 1999]

### **Ficheros de Interfaz Externos:**

## Capítulo IV: Estudio de Factibilidad

Grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es un Archivo Lógico Interno de otra aplicación [34].

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio Alto)
Insertar propiedad	1	6	Bajo
Modificar propiedad	1	6	Bajo
Eliminar propiedad	1	6	Bajo
Insertar usuario	2	15	Medio
Eliminar usuario	2	15	Medio
Modificar usuario	2	15	Medio
Insertar propietario	2	15	Medio
Eliminar propietario	2	15	Medio
Modificar propietario	2	15	Medio
Insertar noticia.	2	7	Medio
Eliminar noticia.	2	7	Medio
Modificar noticia	2	7	Medio
Insertar eventos.	2	7	Medio
Modificar eventos.	2	7	Medio
Eliminar eventos.	2	7	Medio
Modificar información de los departamentos	1	3	Bajo
Insertar información de los departamentos.	1	3	Bajo

## Capítulo IV: Estudio de Factibilidad

Eliminar información de los departamentos.	1	3	Bajo
--	---	---	------

**Tabla 22 Planificación: Entradas Externas**

<b>Nombre de la salida externa</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Clasificación (Bajo, Medio Alto)</b>
Visualizar lista de empleados.	2	6	Bajo
Visualizar estado de la cuenta.	2	7	Medio
Visualizar eventos.	2	4	Bajo
Visualizar noticia.	2	4	Bajo
Generar reportes de deudores.	2	6	Bajo
Generar reportes de transacciones del día.	2	6	Bajo
Generar reportes de transacciones de un período dado.	2	6	Bajo
Generar reportes de transacciones de los pagos hasta la fecha.	2	6	Bajo

**Tabla 23. Planificación: Salidas Externas**

## Capítulo IV: Estudio de Factibilidad

<b>Nombre de la Petición</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Clasificación (Bajo, Medio y Alto)</b>
Autenticarse	1	2	Bajo
Cambiar contraseña	1	2	Bajo
Recuperar contraseña	2	4	Bajo
Pagar la cuenta de la propiedad	2	7	Medio

**Tabla 24. Planificación: Peticiones**

<b>Nombre del fichero interno</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Clasificación (Bajo, Medio y Alto)</b>
departamento	1	3	Bajo
evento	1	5	Bajo
noticia	1	5	Bajo
operator	1	7	Bajo
Pago	1	4	Bajo
Persona	1	8	Bajo
Pertenece	1	5	Bajo
Pertenece_a	1	5	Bajo

## Capítulo IV: Estudio de Factibilidad

propiedad	1	6	Bajo
Tipo_propiedad	1	3	Bajo
Question	1	3	Bajo
Trabajador	1	3	Bajo

**Tabla 25 Planificación: Ficheros Internos**

<b>Elementos</b>	<b>Bajos</b>	<b>X Peso</b>	<b>Medios</b>	<b>X Peso</b>	<b>Altos</b>	<b>X Peso</b>	<b>Subtotal de puntos de función</b>
Ficheros lógicos internos	12	7	13	10	0	15	84
Ficheros de interfaces externas	0	5	0	7	0	10	0
Entradas externas	6	3	12	4	0	6	57
Salidas externas	7	4	1	5	0	7	33
Peticiones	12	3	0	4	0	16	36

## Capítulo IV: Estudio de Factibilidad

<b>Total</b>							<b>210</b>
--------------	--	--	--	--	--	--	------------

**Tabla 26 Planificación: Puntos de función**

Características	Valor		
Puntos de función desajustados	210		
Lenguaje	mysql	php	java script
Instrucciones fuentes por puntos de función	40	44	58
Por ciento de la aplicación en cuanto requerimientos funcionales	35%	55%	10%
Instrucciones fuentes	3024	5082	1218
<b>Total de instrucciones fuente</b>	<b>9324</b>		

**Tabla 27 Planificación: Puntos de función**

Cálculo de:	Valor	Justificación
RCPX	1,00	BD moderada, no se requiere de amplia documentación. La aplicación Web tiene una

## Capítulo IV: Estudio de Factibilidad

		moderada complejidad. (Nominal)
RUSE	1,00	Se implementa código reutilizable para el aprovechamiento de este en toda la aplicación.(Nominal)
PDIF	1,00	El sistema no posee grandes limitaciones en cuanto al tiempo de ejecución ya que podrá estar trabajando por varias horas. EL Software no presenta restricción de memoria impuesta. La plataforma de la aplicación tiene una gran persistencia. (Nominal)
PERS	0,81	Hay poco movimiento del personal. (Alto)
PREX	0,88	El equipo tiene buen dominio y posee conocimiento del lenguaje de programación. (Alto)
FCIL	0,85	La aplicación se desarrolla en un solo lugar, para esto se utilizaron distintas herramientas de programación como: Zend Studio y Macromedia Dreamweaver, así como la herramienta Rational Rose para la documentación, empleando como notación UML. (Alto)
SCED	1,00	La planificación se hace con moderada frecuencia. (Nominal)
PREC	3,70	El equipo de desarrollo posee un conocimiento considerable de los objetivos del producto, no

## Capítulo IV: Estudio de Factibilidad

		tiene gran experiencia en la realización de aplicaciones de este tipo. (Nominal)
FLEX	3,0	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa. (Nominal)
TEAM	1,10	El equipo que va a desarrollar altamente cooperativo. (Muy Alto)
RESL	4,20	Teniendo en cuenta la alta experiencia que existe en el país acerca de este tipo de estudios existen algunos factores de riesgo. (Nominal)
PMAT	5,80	Nivel I Alto porque se encuentra en su primera etapa un poco avanzada. (Bajo)

**Tabla 28 Planificación: Puntos de función**

Miles de instrucciones fuentes (MF): 9,324.

### Multiplicador de esfuerzos

7

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED$$

i=1

7

$$EM = \prod_{i=1}^7 E_{mi} = 1,00 * 1,00 * 1,00 * 0,81 * 0,88 * 0,85 * 1,00$$

$$EM = 0,60588 \approx 0,60$$

i=1

### Factores de escala

5

$$SF = \sum_{i=1}^5 SFi = \text{PREC} + \text{FLEX} + \text{RESL} + \text{TEAM} + \text{PMAT}$$

$$SF = \sum_{i=1}^5 SFi = 3,70 + 3,0 + 4,20 + 1,10 + 5,80 = 17,8$$

### Valores de los coeficientes

$$\mathbf{A} = 2,94; \mathbf{B} = 0,91; \mathbf{C} = 3,67; \mathbf{D} = 0,24$$

$$\mathbf{E} = \mathbf{B} + 0,01 * \mathbf{SF}$$

$$E = 0,91 + 0,01 * SF$$

$$E = 0,91 + 0,01 * 17,8$$

$$\mathbf{E} = \mathbf{1,088}$$

$$\mathbf{F} = \mathbf{D} + 0,2 * (\mathbf{E} - \mathbf{B})$$

$$F = 0,24 + 0,2 * (1,088 - 0,91)$$

$$\mathbf{F} = \mathbf{0,2756}$$

### Esfuerzo

$$\mathbf{PM} = \mathbf{A} * (\mathbf{MF})^{\mathbf{E}} * \mathbf{EM}$$

$$\mathbf{PM} = 2,94 * (9,324)^{1,088} * 0,60588$$

$$PM = 20,21$$

**Cálculo del tiempo de desarrollo**

$$TDEV = C * PM^F$$

$$TDEV = 3.67 * (20.21)^{0.2756}$$

$$TDEV = 8,40 \approx 9 \text{ meses}$$

**Cálculo de la cantidad de hombres**

$$CH = PM / TDEV$$

$$CH = 20,21 / 8,40$$

$$CH = 2,40 \approx 3 \text{ Hombres}$$

**Costo**

$$CHM = 3 * \text{Salario Promedio}$$

$$CHM = 3 * 250$$

$$CHM = \$750 / \text{Mes}$$

$$\text{Costo} = CHM * PM$$

$$\text{Costo} = 750 * 20,21$$

$$\text{Costo} = \$15 157,5$$

## Capítulo IV: Estudio de Factibilidad

Los costos en los que se incurriría de desarrollarse el sistema serían:

<b>Cálculo de:</b>	<b>Valor</b>
Esfuerzo(PM)	23,45
Tiempo de desarrollo	9 meses
Cantidad de hombres	3
Costo	\$15 157,5
Salario medio	\$250
RCPX	1,00
RUSE	1,00
PDIF	1,00
PREX	0,88
FCIL	0,85
SCED	1,00

**Tabla 29 Costos totales**

### **4.3 Conclusiones**

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

## Capítulo IV: Estudio de Factibilidad

Como resultado final se obtuvo un costo total del proyecto de \$15 157,5 a desarrollar por tres personas en un tiempo de 9 meses de trabajo.

### Conclusiones

La gestión de la información en el ayuntamiento de la Ciudad de Georgetown Guyana un lleva consigo varios procesos, que de forma manual resultan engorrosos y lentos, por lo que la realización de un sistema informático basado en el modelo de una Intranet resultaría beneficioso y agilizaría de forma extraordinaria el manejo de los datos y el pago de los impuestos de propiedad en el departamento de tesorería de dicho ayuntamiento.

Después de haber realizado un estudio de los flujos del proceso de pago de impuesto de propiedades se presenta una propuesta de sistema que basa su funcionamiento en los estándares de la W3C para la web. Esta propuesta permite a los usuarios de la red producir y consumir información referente al proceso de pago de impuesto de propiedades.

- Se diseñó un modelo de datos para el proceso de pago de impuesto online.
- Se establecieron niveles de acceso a la información para los roles de los usuarios en el sistema.
- Se diseñó un sistema multiplataforma en cuanto a Sistema Operativo y Gestor de Base de datos.
- Se implementó un sistema para la gestión del proceso de pago de impuesto online referido a los propietarios de la Ciudad de Georgetown

## Recomendaciones

A pesar de que los objetivos trazados con la realización de este trabajo fueron cumplidos, el desarrollador del mismo sugiere tomar esta propuesta solo como la primera fase de un proyecto mucho más ambicioso y continuar esta investigación. Se recomienda entonces:

- Continuar con el estudio de los flujos de trabajos y del ayuntamiento de la Ciudad de Georgetown para abarcar todos los aspectos referidos a este, así como profundizar en los aspectos tratados con mayor detalle.
- Extender su uso a otras ciudades del país ajustando el sistema a las particularidades de cada ciudad.
- Conservar la idea de que el sistema debe ser desarrollado sobre la base de la multiplataforma lo que permitirá adaptarse a las nuevas tecnologías de la informática moderna.

## **Referencias Bibliográficas**

[1] Boar, Bernard H. Aligning Information Technology with Business Strategies. -

EEUU:Editorial John Wiley & Sons Inc, 1994.-p. 132

[2] Ayuntamiento. Microsoft® Encarta® 2007 [DVD]. Microsoft Corporation, 2006.

[3] Impuesto sobre la Propiedad. Tomado De:

<http://170.224.70.221/bofaespanol/common/glossary.fcs?qstrTerm=PropertyTax>  
10/06/2008

[4] Tarjetas de crédito. Tomado De:

<http://www.zonacredito.com/> 12/06/2008.

[5] Intranet. Tomado De:

<http://www.internet.com/manuales.htm> , (20/01/06).

[6] *Adobe Dreamweaver*. Tomado De:

<http://www.adobe-dreamweaver.softonic.com/>13/05/2008.

[7] HTML. Tomado De:

[http://es.wikipedia.org/wiki/Código\\_HTML](http://es.wikipedia.org/wiki/Código_HTML) 11/05/2008.

[8] CSS Hojas de estilo en cascada(CSS). Tomado De:

<http://www.es.wikipedia.org/wiki/CSS> 11/05/2008

[9] Java Script. Tomado De:

<http://es.wikipedia.org/wiki/JavaScript> 12/05/2008

[10] Photoshop. Tomado De:

<http://www.extropia.com/tutorials/photoshop/toc.html> 15/01/08

[11] Tecnologías Web ASP. Tomado de:

<http://es.selfhtml.org/introduccion/tecnologias/java.htm> 31/02/06

[12] Tecnologías Web ASP. Tomado de:

<http://es.selfhtml.org/introduccion/tecnologias/java.htm> 31/02/06

[13] Converse, T. *PHP 5 and MySQL Bible*.—EEUU: Editorial Wisley Publishing inc.,  
2004.

[14] PHP. Tomado De:

<http://www.webestilo.com/php/> 06/03/07.

## Bibliografía

[15] Apache Http Server. Tomado De:

[http://www.es.wikipedia.org/wiki/Apache\\_http\\_server](http://www.es.wikipedia.org/wiki/Apache_http_server) 13/05/2008.

[16] Matos, Rosa María. *Introducción al trabajo con Base de Datos*-Asignatura de

Sistemas de Gestión de Base de Datos. --Ciudad Habana:Editorial CUJAE, 2004.--  
p.10.

[17] Lenguaje de Modelación Unificado (UML). Tomado De:

[http://www.es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://www.es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado) 12/05/2008.

[18] Proceso Unificado de Desarrollo (RUP). Tomado De:

[http://www.es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Rational](http://www.es.wikipedia.org/wiki/Proceso_Unificado_de_Rational) 12/05/2008.

[19] Jacobson, I. *El Proceso Unificado de Desarrollo de software*. –EEUU:Editorial

Addison-Wesley.–p. 115.

[20] Idem, p.116.

[21] Idem, p.117

[22] Idem, p.121.

[23] Idem, p.127

[24]Especificaciones de Requerimientos. Tomado De:

<http://mitecnologico.com/Main/EspecificacionesDeRequerimientos>, 7/05/06.

## Bibliografía

[25] Jacobson, I. El Proceso Unificado de Desarrollo de software. –EEUU: Editorial

Addison-Wesley, 2000. –p.127.

[26] Idem. p.127.

[27] Idem. p. 115.

[28] Idem. p. 123

[29] Idem. p. 116

[30] Idem. p. 133

[31] Idem. p. 127

[32] Idem. p.134

[33] Estudio de Factibilidad Tomado De:

[http://www.tejasre.com/Spanish/Feasibility\\_Studies.cfm](http://www.tejasre.com/Spanish/Feasibility_Studies.cfm) 06/06/2008

## **Bibliografia**

- [34] [COC 00] COCOMO II. Model Definition Manual. Center for Software Engineering at the University of Southern California, 2000.

## Bibliografía

Pressman, Roger. Ingeniería de software, Un enfoque práctico. Roger Pressman. – España: Editorial McGraw –Hill, 2002. – 395p.

Rumbaugh, J. El Lenguaje Unificado de Modelado. Manual de referencia. J Rumbaugh, I Jacobson y G Booch. –EEUU: Editorial Addison-Wesley, 2000. – 312p.

Conallen, Jim. Modeling Web Application Architectures with UML. Tomado De: [http://www.rational.com/media/uml/resources/documentation/27662\\_webapps.pdf](http://www.rational.com/media/uml/resources/documentation/27662_webapps.pdf) 10/05/2008.

Jorge A., Torres. Tutorial de PHP. Tomado De: [http://www.elguruprogramador.com.ar/tutoriales/tutorial\\_php.asp](http://www.elguruprogramador.com.ar/tutoriales/tutorial_php.asp) 03/ 05/ 2008.

Jacobson, Ivar. El Proceso Unificado de Desarrollo de software. I Jacobson, G Booch, J Rumbaugh. –EEUU: Editorial Addison-Wesley, 2000. –438p.  
84

\_\_\_\_\_. El Lenguaje Unificado de Modelado. I Jacobson, G Booch, J Rumbaugh. --EEUU: Editorial Addison-Wesley, 2000. –526p.

## Anexo 1 Diagramas de clase web

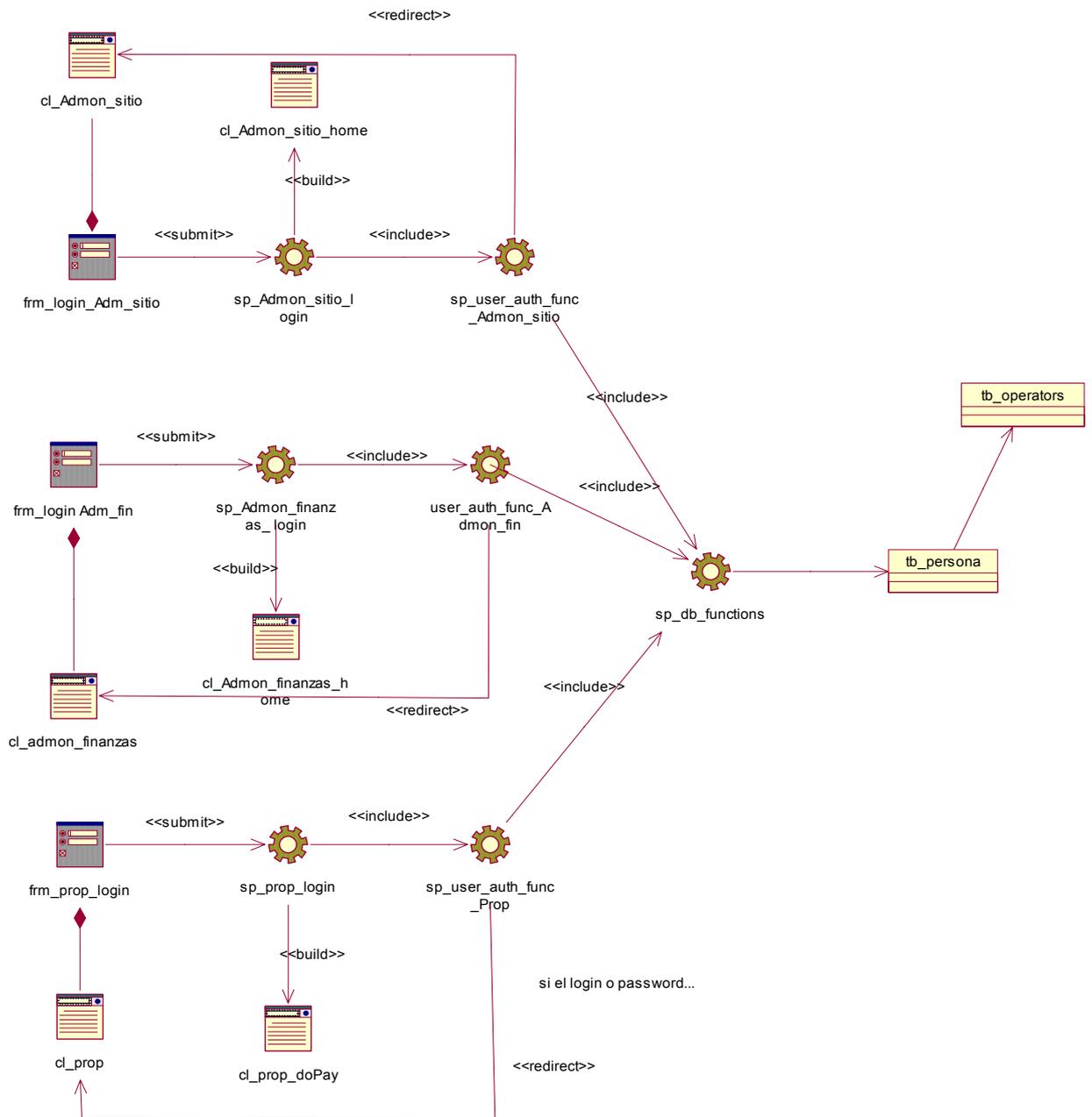


Figura 1.1: Diagrama de clases Web Autenticarse.

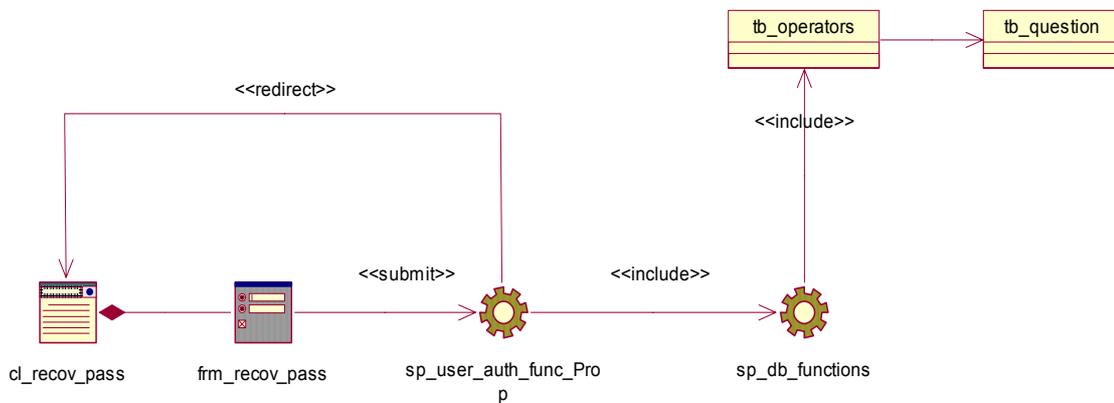


Figura 1.2: Diagrama de clases Web Recuperar la contraseña

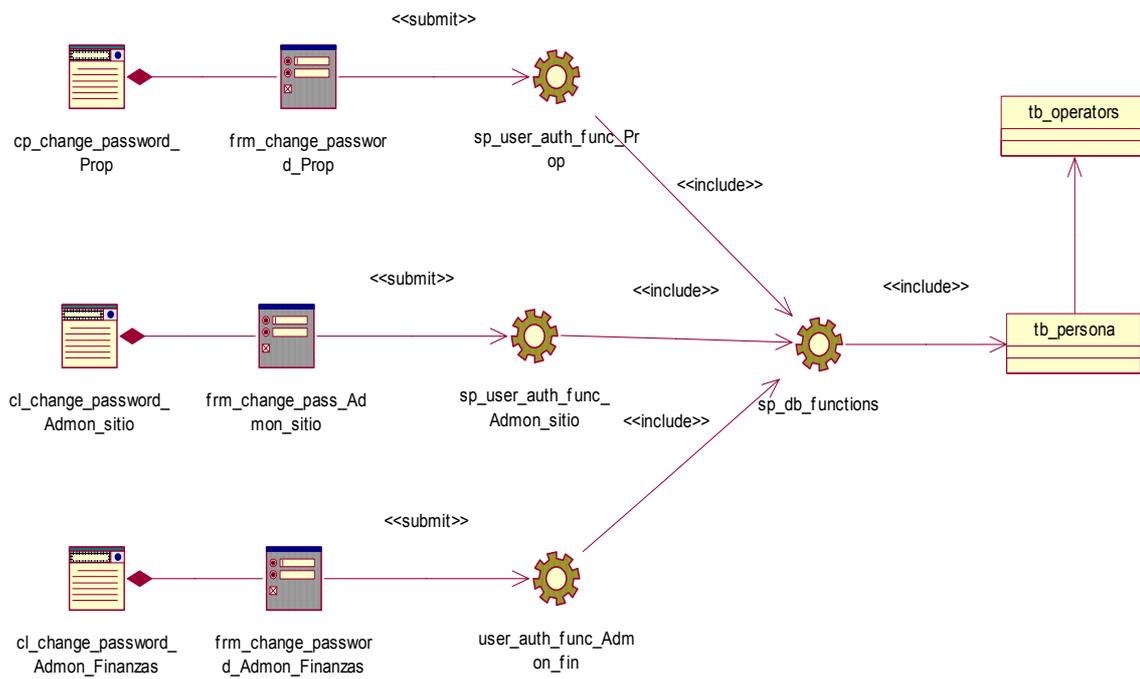


Figura 1.3: Diagrama de clases Web Cambiar Contraseña

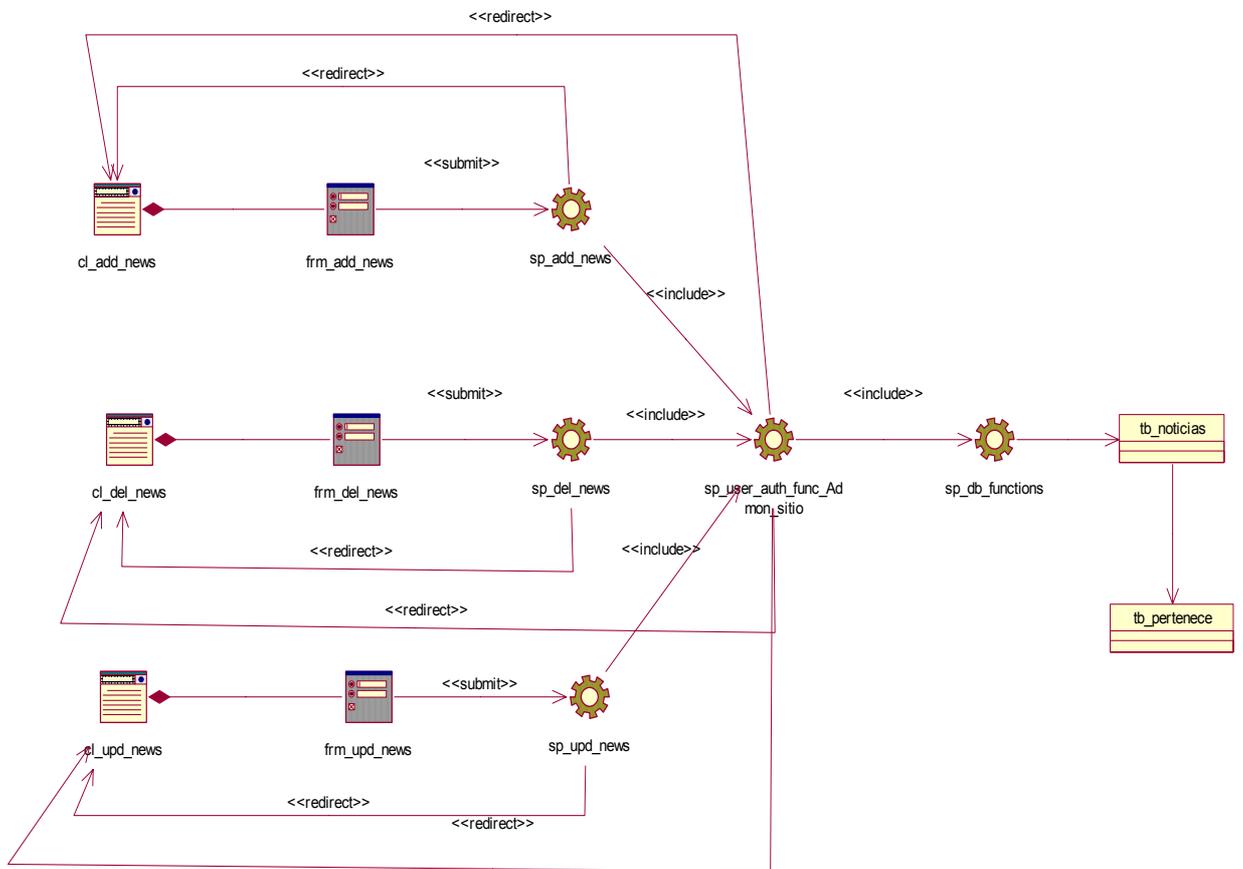


Figura 1.4: Diagrama de clases Web Gestionar Noticias

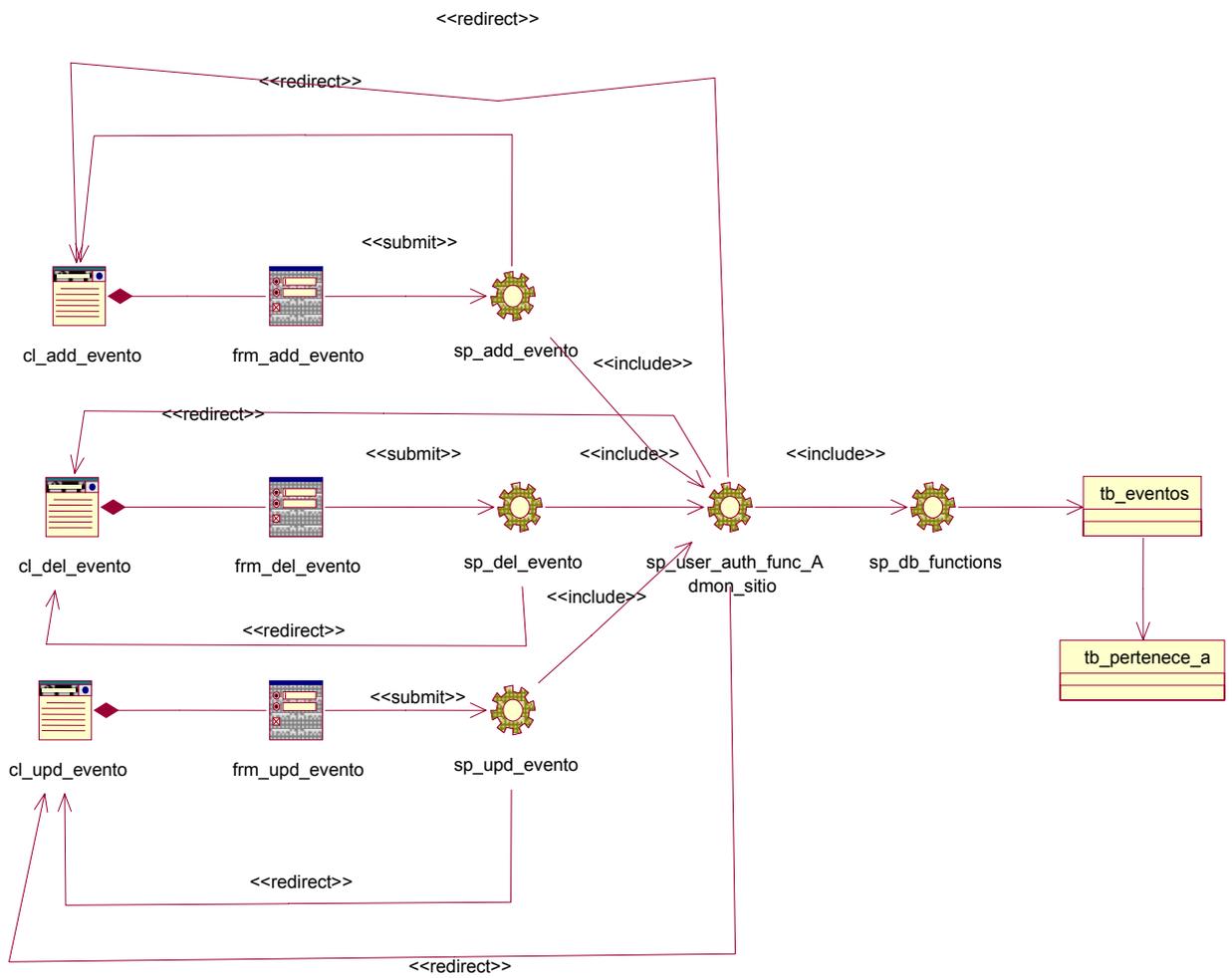


Figura 1.5: Diagrama de clases Web Gestionar Eventos

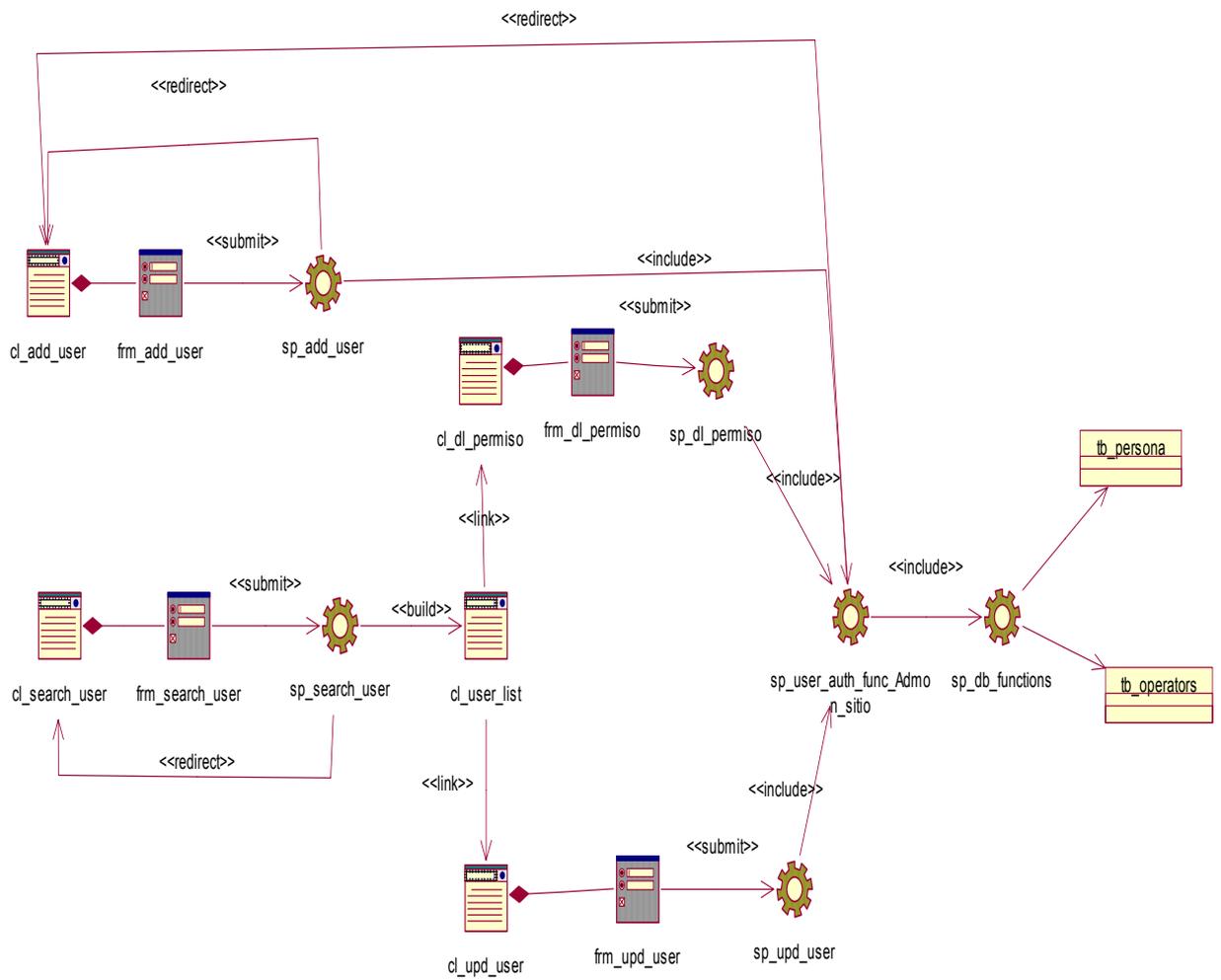


Figura 1.6: Diagrama de clases Web Gestionar Usuarios

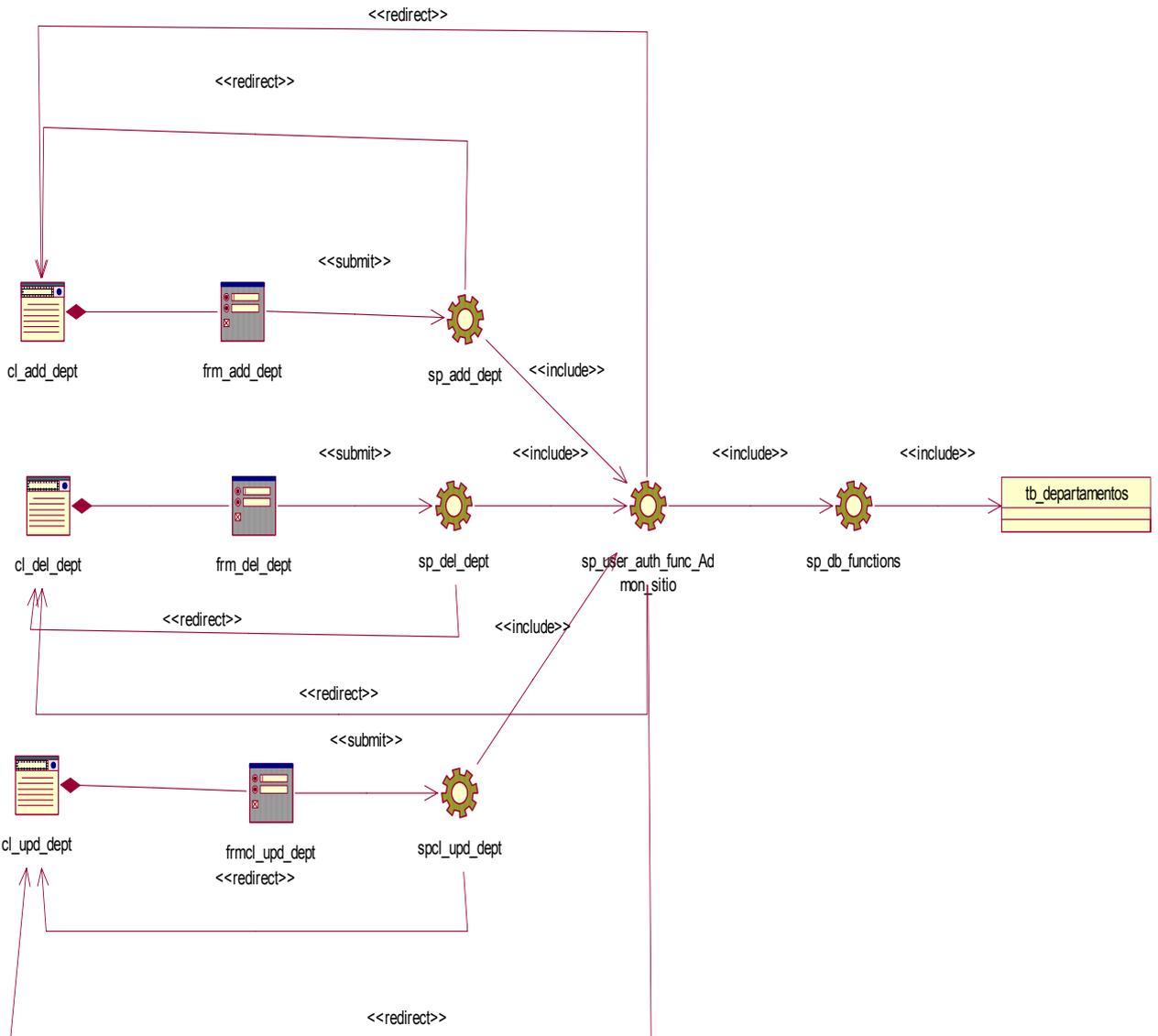


Figura 1.7: Diagrama de clases Web Gestionar Información Departamentos



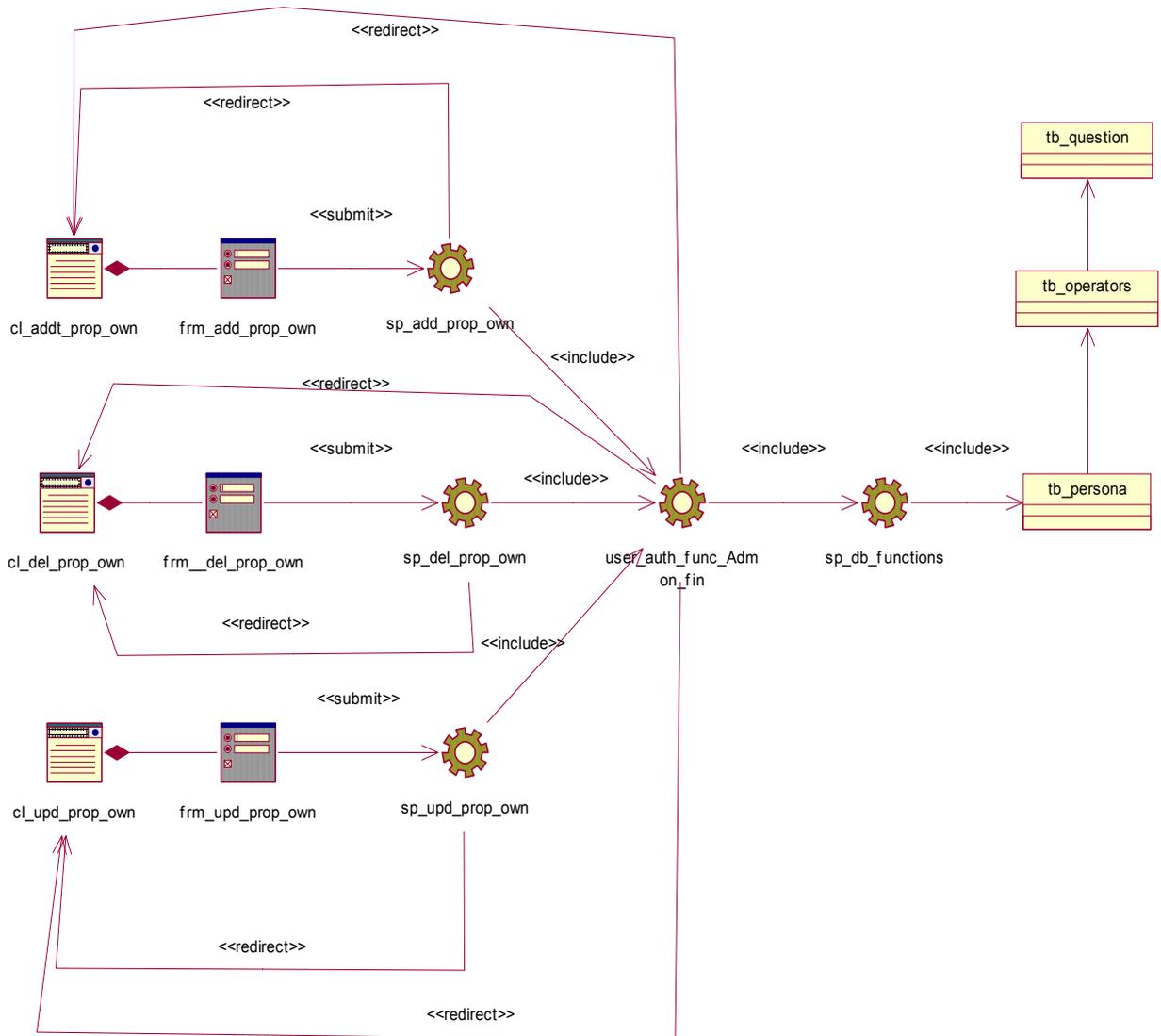


Figura 1.9: Diagrama de clases Web Gestionar Proprietarios

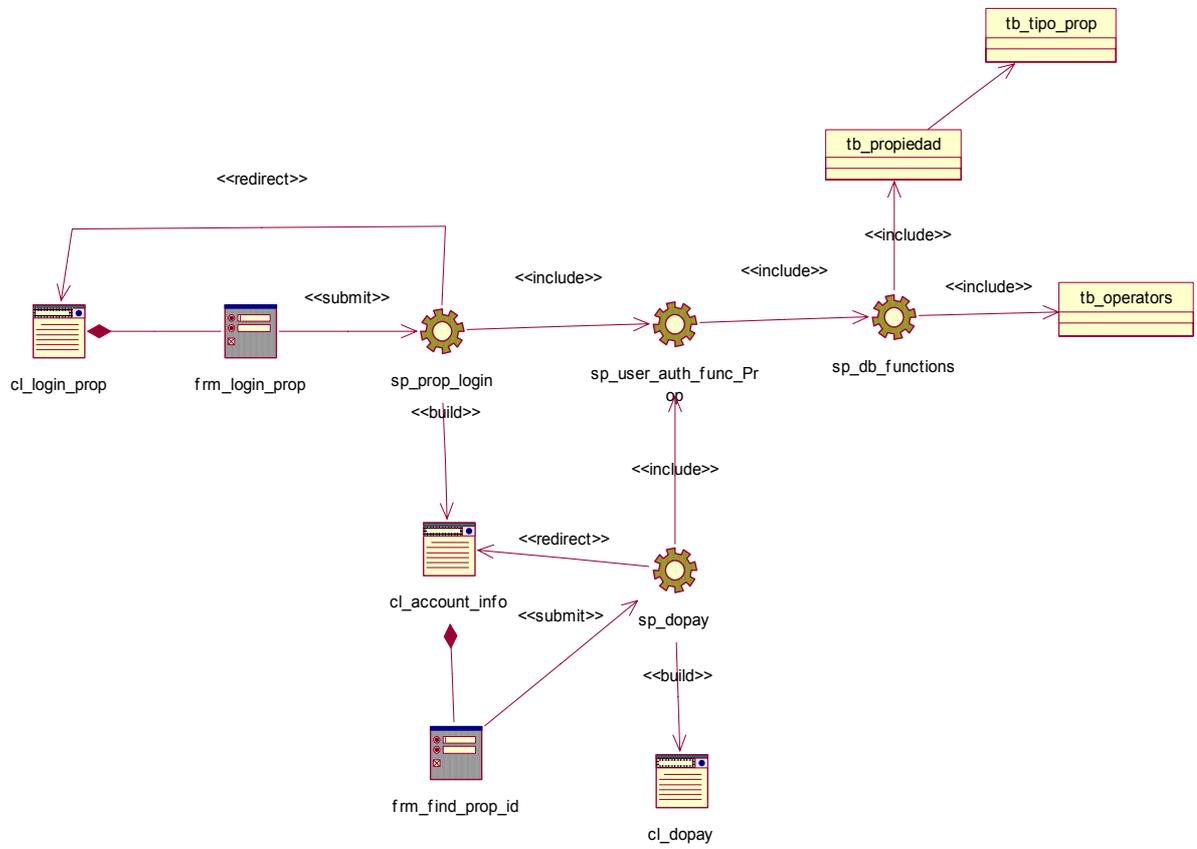


Figura 1.10: Diagrama de clases Web Visualizar estado de la Cuenta

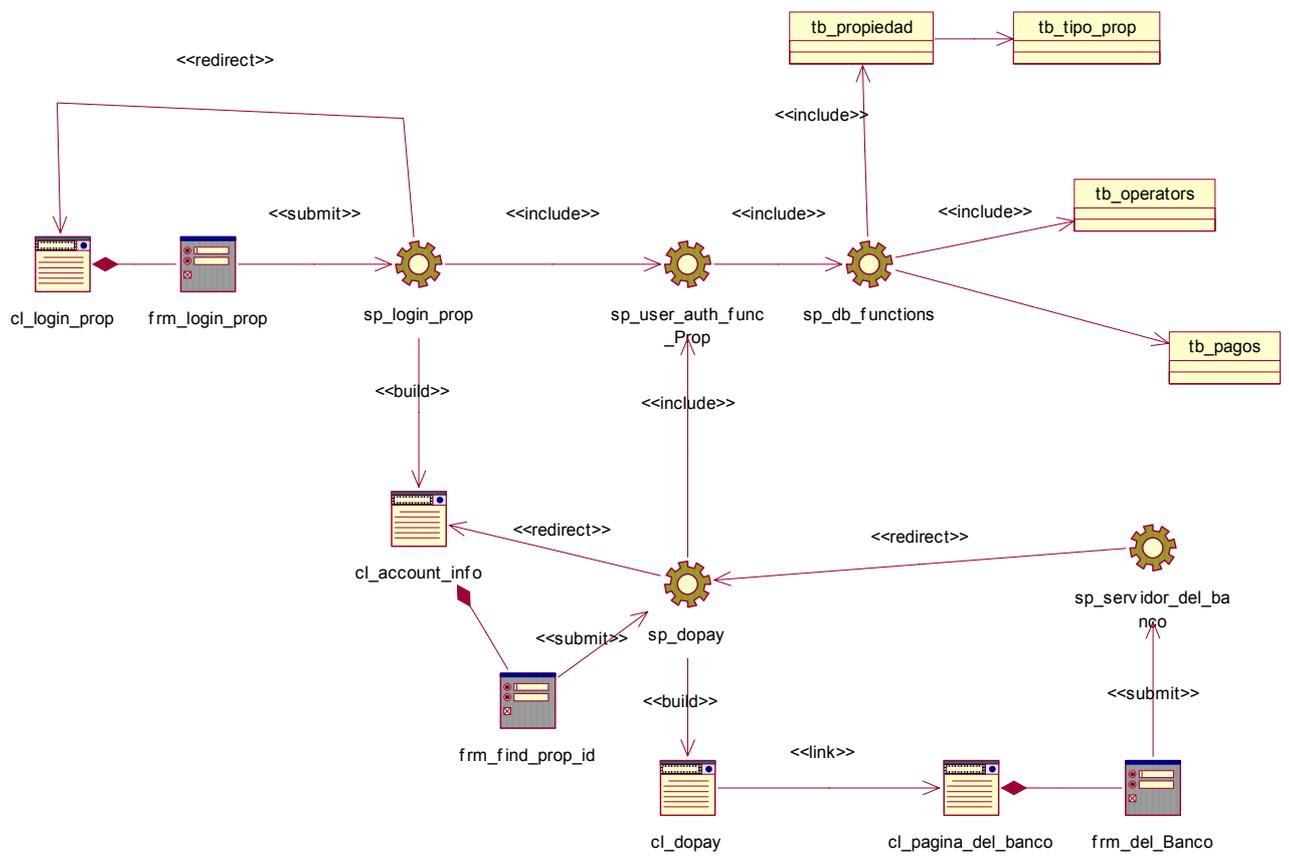


Figura 1.11: Diagrama de clases Web Pagar la cuenta de la propiedad

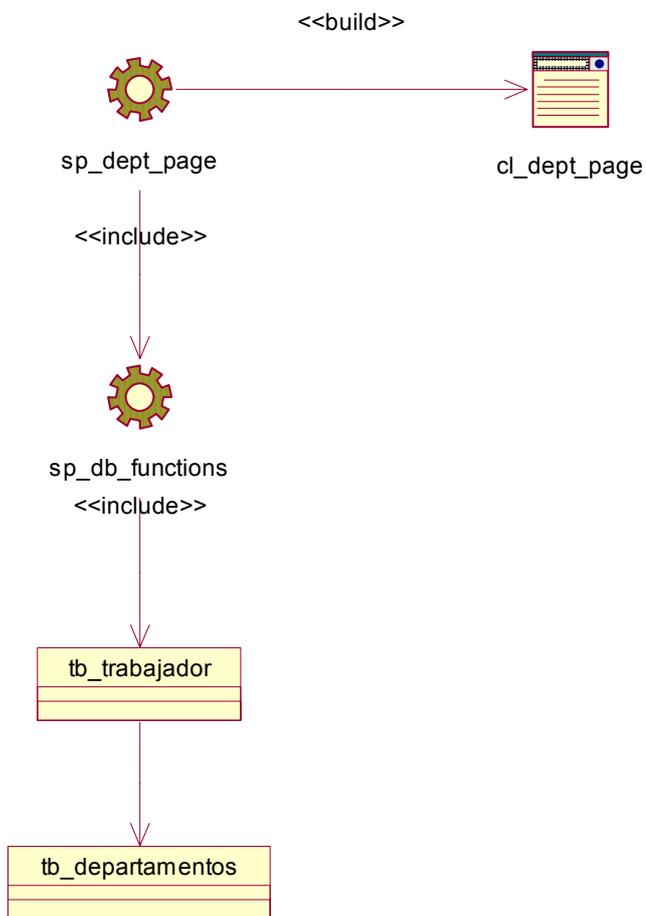


Figura 1.12: Diagrama de clases Web Visualizar lista de empleados

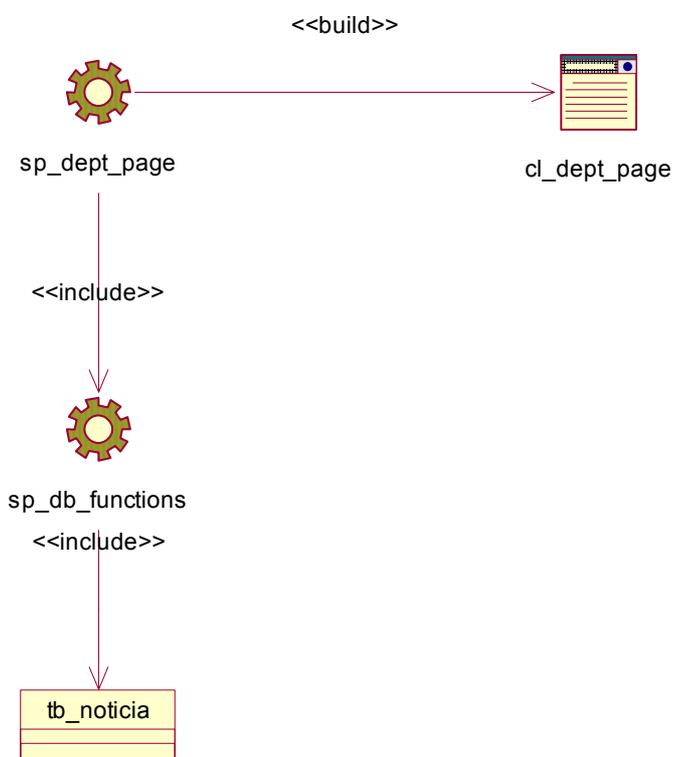


Figura 1.13: Diagrama de clases Web Visualizar Noticias

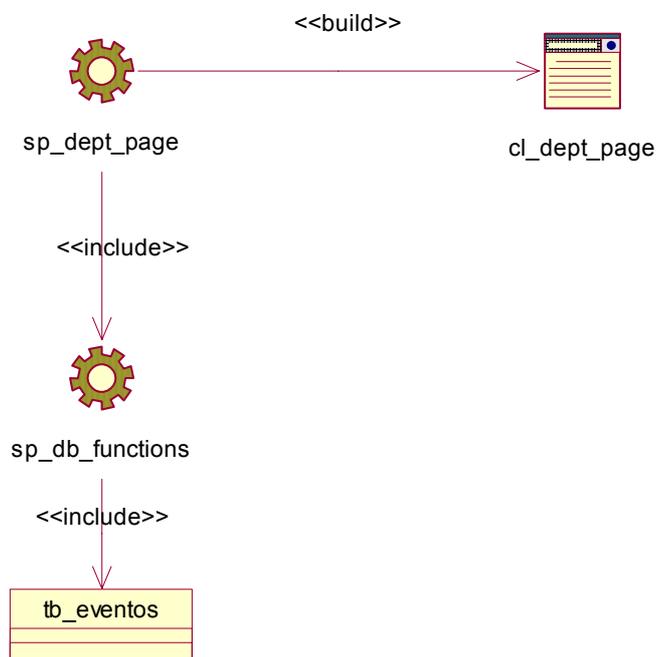


Figura 1.14: Diagrama de clases Web Visualizar Eventos

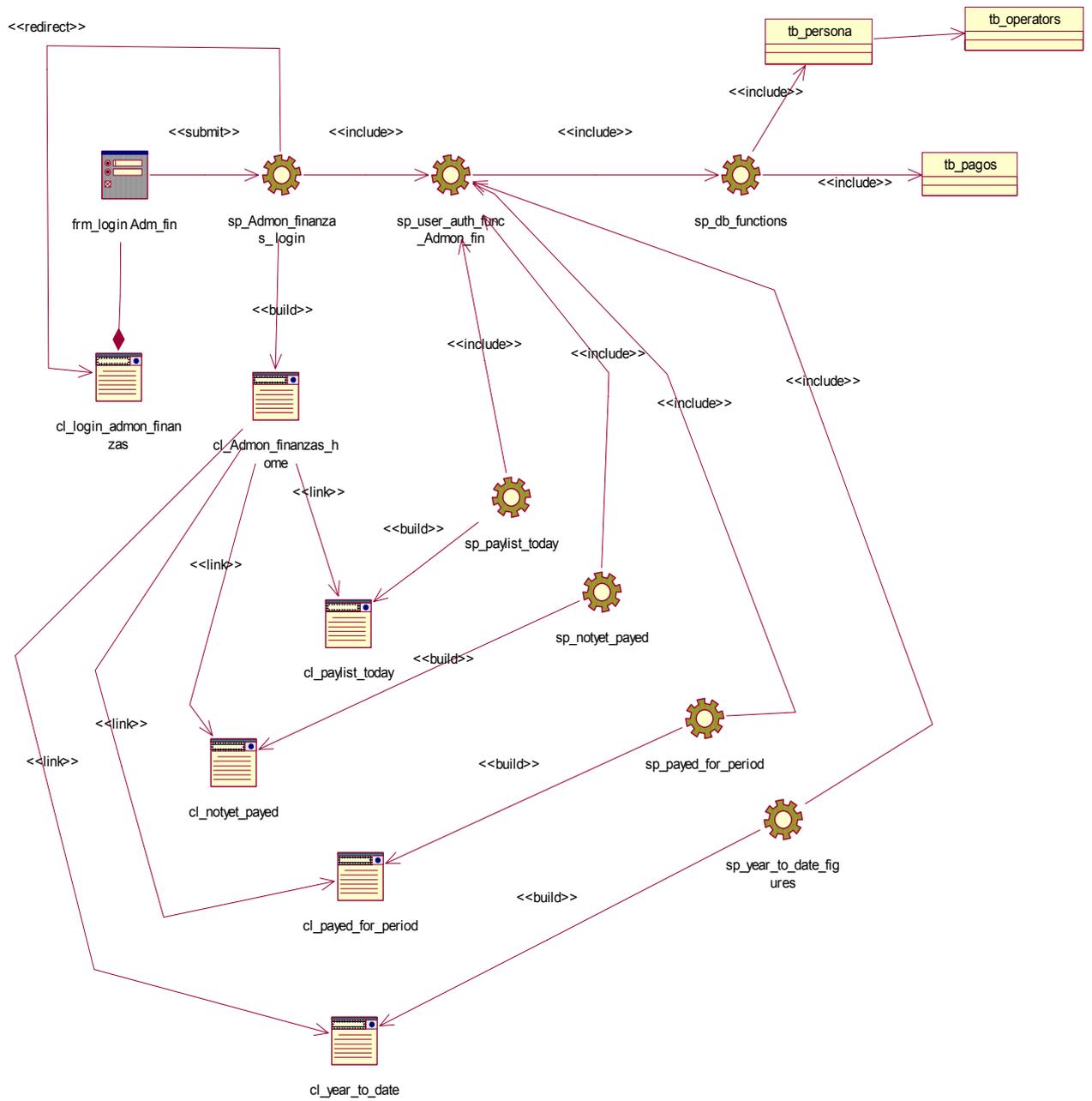
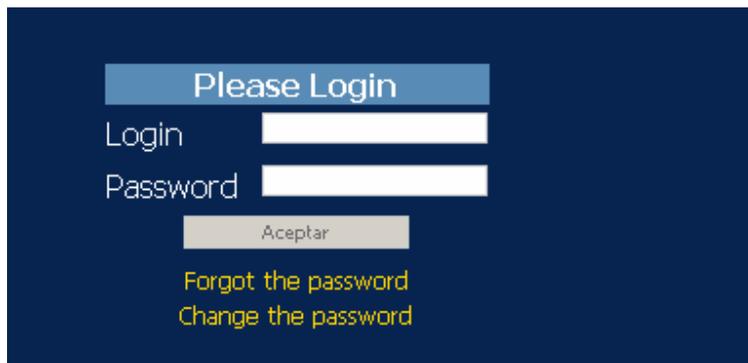


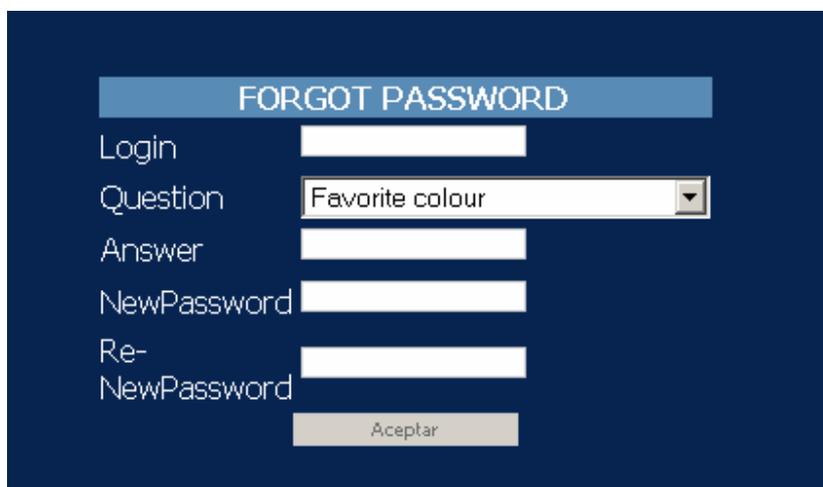
Figura 1.15: Diagrama de clases Web Generar Reportes

## ANEXO II PROTOTIPOS



A dark blue rectangular form with a light blue header bar containing the text "Please Login". Below the header, there are two white input fields: "Login" and "Password". Below the "Password" field is a grey button labeled "Aceptar". At the bottom of the form, there are two yellow links: "Forgot the password" and "Change the password".

Prototipo 2.1. Autenticarse



A dark blue rectangular form with a light blue header bar containing the text "FORGOT PASSWORD". Below the header, there are five white input fields: "Login", "Question" (with a dropdown menu showing "Favorite colour"), "Answer", "NewPassword", and "Re-NewPassword". Below the "Re-NewPassword" field is a grey button labeled "Aceptar".

Prototipo 2.2 Recuperar Contraseña

**CHANGE PASSWORD**

Login

Password

New Password

Re-New Password

Remember to say **NO** to any auto complete options that is offered by the browser.  
Your password is private and should **NOT** be shared with anyone.

Prototipo 2.3 Cambiar Contraseña

**Add News**

**Header**

**Department**

**Body**

Prototipo 2.4.1 Gestionar Noticias (Add News)

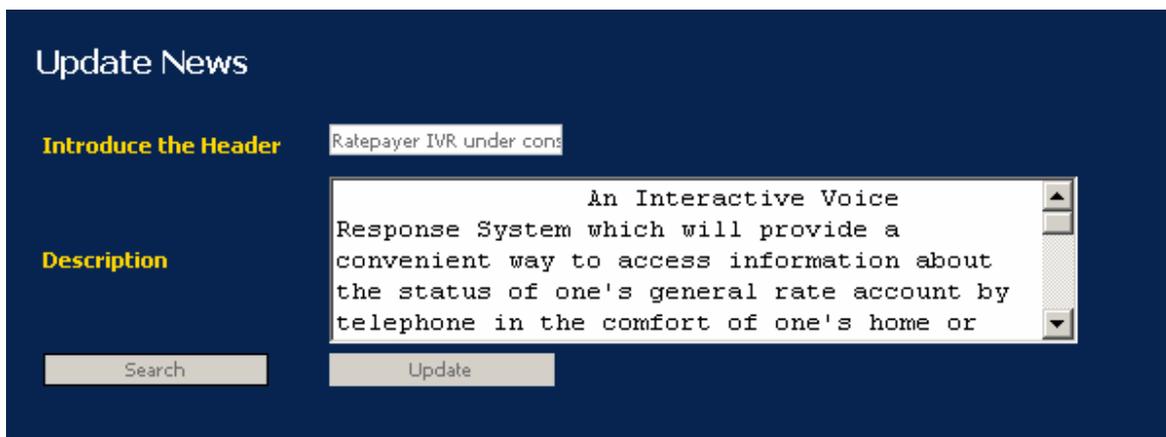


Delete News

Select Department Name CITY ENGINEERS

Search

Prototipo 2.4.2 Gestionar Noticias (Delete News)



Update News

Introduce the Header Ratepayer IVR under cons

Description

An Interactive Voice Response System which will provide a convenient way to access information about the status of one's general rate account by telephone in the comfort of one's home or

Search Update

Prototipo 2.4.3 Gestionar Noticias (Update News)

### Add Events

**Header**

**Department** CITY ENGINEERS ▾

**Body**

Save

Prototipo 2.5.1 Gestionar Eventos (Añadir Eventos)

### Delete Events

**Select Department Name** CITY ENGINEERS ▾

Search

No	Events	Delete
1	salary	<input type="checkbox"/>

Delete

Prototipo 2.5.2 Gestionar Eventos (Eliminar Eventos)

### Update Events

**Introduce the Header**

**Description**

Prototipo 2.5.1 Gestionar Eventos (Actualizar Eventos)

### Search User

**Search**

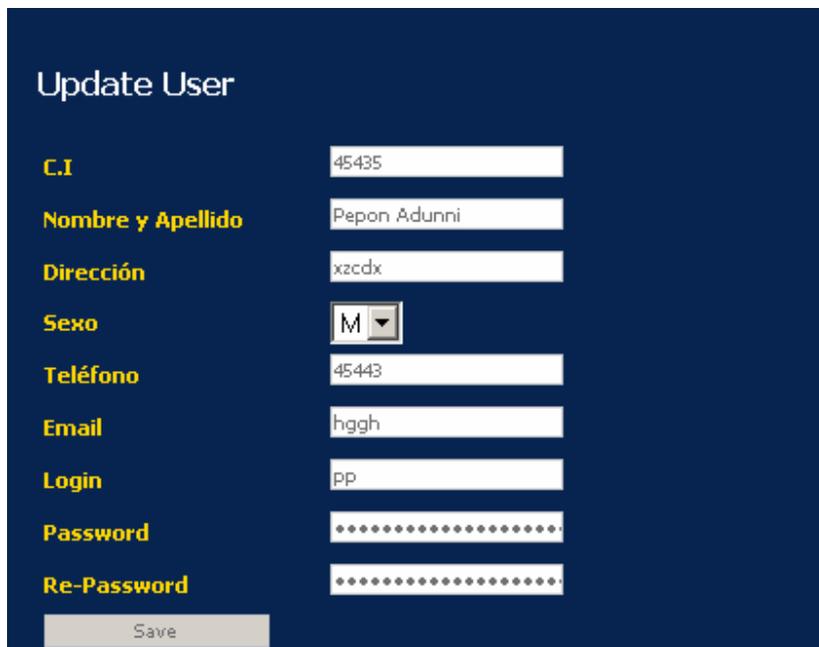
Nombre	Email	Permisos	Activar
Pepon Adunni	hggh		SI
pp	cxvfxc		SI
kadir	fgdfgddffd		SI
jozee	jozee@234		SI
lar ric	haha		SI
dwan ed	jhdchq		SI
sam sammy	sam@sam		SI
dishon gren	dish@buil5		SI
q	q		SI
sheld small	shel@where		SI
gramps tawia	gramps@ss		SI

Prototipo 2.6.1 Gestionar Usuarios (Permisos)

The image shows a dark blue form titled "Add Employee". It contains the following fields and controls:

- C.I.**: A white text input field.
- Nombre y Apellido**: A white text input field.
- Dirección**: A white text input field.
- Sexo**: A dropdown menu with "F" selected.
- Teléfono**: A white text input field.
- Email**: A white text input field.
- Departamento**: A dropdown menu with "CITY ENGINEERS" selected.
- Ocupacion**: A white text input field.
- Login**: A white text input field.
- Password**: A white text input field.
- Re-Password**: A white text input field.
- Save**: A grey button at the bottom left.

Prototipo 2.6.2 Gestionar Usuarios (Añadir Usuarios)



The image shows a 'Update User' form on a dark blue background. The form contains the following fields and values:

Field	Value
C.I	45435
Nombre y Apellido	Pepon Adunni
Dirección	xzcdx
Sexo	M
Teléfono	45443
Email	hggh
Login	pp
Password	.....
Re-Password	.....

A 'Save' button is located at the bottom left of the form.

Prototipo 2.6.3 Gestionar Usuarios (Actualizar Usuarios)



The image shows an 'Add Department' form on a dark blue background. The form contains the following fields:

- Department Name:** A text input field.
- Descripcion:** A large text area with a vertical scrollbar on the right side.

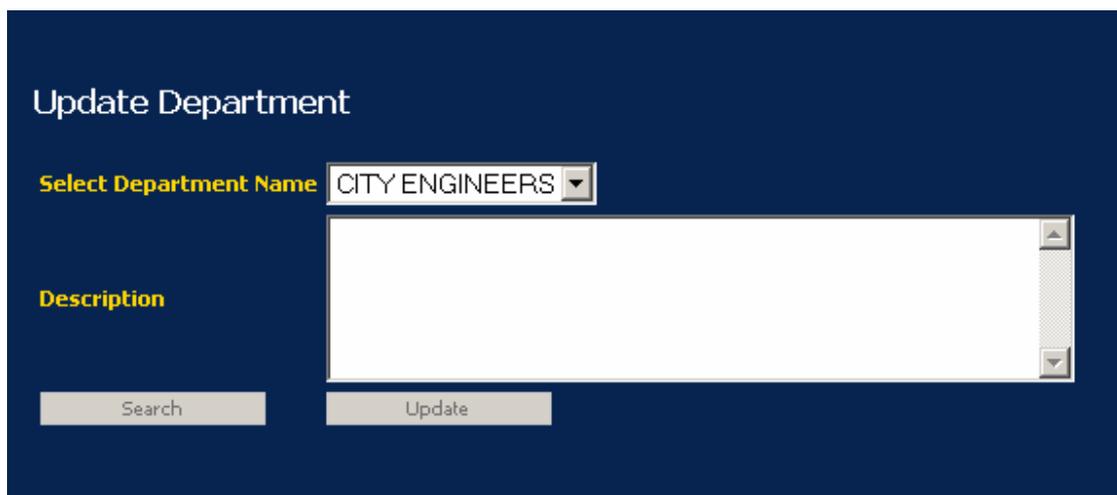
A 'Save' button is located at the bottom left of the form.

Prototipo 2.7.1 Gestionar Departamentos (Añadir Departamentos)



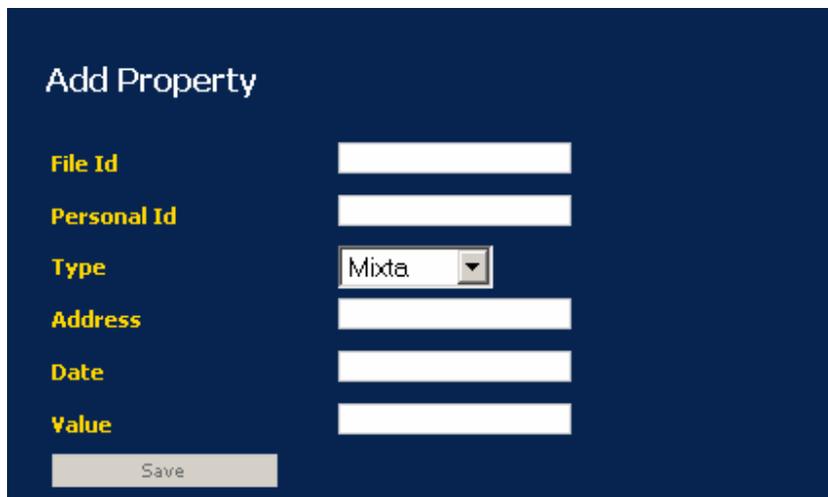
The image shows a dark blue rectangular panel with the title "Delete Department" in white text at the top left. Below the title, the text "Select Department Name" is written in yellow, followed by a white dropdown menu containing the text "CITY ENGINEERS" and a downward-pointing arrow. Below the dropdown menu is a light gray rectangular button with the word "Delete" centered in black text.

Prototipo 2.7.2 Gestionar Departamentos (Eliminar Departamentos)



The image shows a dark blue rectangular panel with the title "Update Department" in white text at the top left. Below the title, the text "Select Department Name" is written in yellow, followed by a white dropdown menu containing the text "CITY ENGINEERS" and a downward-pointing arrow. Below the dropdown menu is a large white rectangular text area with a vertical scrollbar on the right side. To the left of the text area, the text "Description" is written in yellow. Below the text area are two light gray rectangular buttons: "Search" on the left and "Update" on the right, both with black text.

Prototipo 2.7.3 Gestionar Departamentos (Actualizar Departamentos)



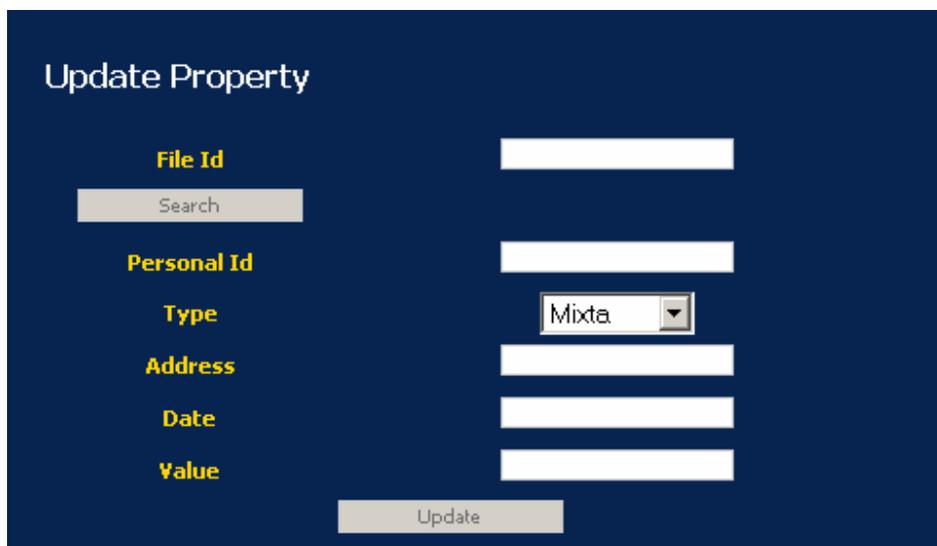
The image shows a dark blue rectangular form titled "Add Property". On the left side, there are six labels in yellow: "File Id", "Personal Id", "Type", "Address", "Date", and "Value". To the right of each label is a white input field. The "Type" field is a dropdown menu with "Mixta" selected. At the bottom left of the form is a grey button labeled "Save".

Prototipo 2.8.1 Gestionar Propiedad (Añadir Propiedad)



The image shows a dark blue rectangular form titled "Delete Property". On the left side, there is a yellow label "File Id". To its right is a white input field. At the bottom left of the form is a grey button labeled "Delete".

Prototipo 2.8.2 Gestionar Propiedad (Eliminar Propiedad)



The 'Update Property' form is set against a dark blue background. It features the title 'Update Property' at the top left. Below the title, there are six input fields arranged vertically, each with a label to its left: 'File Id', 'Personal Id', 'Type', 'Address', 'Date', and 'Value'. The 'Type' field is a dropdown menu with 'Mixta' selected. A 'Search' button is positioned to the left of the 'File Id' field. At the bottom center, there is an 'Update' button.

Prototipo 2.8.1 Gestionar Propiedad (Actualizar Propiedad)



The 'Add Property Owner' form is set against a dark blue background. It features the title 'Add Property Owner' at the top left. Below the title, there are seven input fields arranged vertically, each with a label to its left: 'Identity Number', 'Name(s)', 'Last Name', 'Address', 'Sex', 'Telephone Number', and 'Email'. The 'Sex' field is a dropdown menu with 'Masculino' selected. A 'Save' button is located at the bottom left of the form.

Prototipo 2.9.1 Gestionar Propietario (Añadir Propietario)



The image shows a dark blue rectangular form titled "Delete Property Owner" in white text. Below the title, the label "Identity Number" is written in yellow. To its right is a white rectangular input field. Below the input field is a light gray button with the word "Delete" in dark gray text.

Prototipo 2.9.2 Gestionar Propietario (Eliminar Propietario)



The image shows a dark blue rectangular form titled "Update Property Owner" in white text. Below the title, the label "Identity Number" is written in yellow, followed by a white input field and a light gray "Search" button. Below this, the labels "Name(s)", "Last Name", "Address", "Sex", "Telephone Number", and "Email" are listed in yellow, each followed by a white input field. The "Sex" field is a dropdown menu with "Masculino" selected. At the bottom of the form is a light gray "Update" button.

Prototipo 2.9.3 Gestionar Propietario (Actualizar Propietario)

CITY OF GEORGETOWN **ONLINE TAX**

Enter Property Number **To View Your Account**

Property Number

Search

Account Information

File ID	Annual Total	Total Amount Paid	Subtotal To be Paid	Interes	Quarterly Demand
01	22500	0	7500	804.04109589	8304.04109589

Prototipo 2.10 Visualizar estado de la Cuenta

Enter Your Info

Card Number

ID

Expiry Date

Amount

Pay

Prototipo 2.11 Pagar la cuenta de la propiedad

### Payed until Today 2008-06-23

No	File Id	Property Owner	Date	Amount
1	00112	George Bank	05-13-2008	\$37292
2	02113	Tamy Sing	05-13-2008	\$ 6323
3	01231	Shirley Nawan	06-09-2008	\$ 7642
4	01245	Terry Vimytu	06-11-2008	\$ 4532
5	00225	Tory Payne	06-12-2008	\$25674
6	02434	Comilia Dask	06-20-2008	\$ 9689
7	00025	Bammy Naughton	06-21-2008	\$24922
8	00234	Gregory Sills	06-21-2008	\$42022
Year to Date Total				\$158096

### Prototipo 2.12 Visualizar Reportes

#### Town Clerk's Employees

No	Name	Sex	Email
1	Jarred Palmer	M	<a href="mailto:Jpalmer1@yahoo.com">Jpalmer1@yahoo.com</a>
2	Syril Panter	M	<a href="mailto:sypam@mail.com">sypam@mail.com</a>
3	Denise Farmer	F	<a href="mailto:denisef@hotmail.com">denisef@hotmail.com</a>
4	Hamia Damel	F	<a href="mailto:hdam@yahoo.com">hdam@yahoo.com</a>
5	Felton Mack	M	<a href="mailto:Feltonmack_1@gmail.com">Feltonmack_1@gmail.com</a>
6	Carrie Norbert	F	<a href="mailto:Cnorbert2001@hotmail.com">Cnorbert2001@hotmail.com</a>
7	Sylman Barn	M	<a href="mailto:sylmanb@yahoo.com">sylmanb@yahoo.com</a>
8	Vermona Nalman	F	<a href="mailto:vnalman@gmail.com">vnalman@gmail.com</a>

### Prototipo 2.13 Visualizar empleados

## Flash News!

### Public Health Automated Technology Programme underway

The computerized management information systems for the Georgetown Municipality's Public Health Department and the Department of Solid Waste Management known as the Public Health Automated Technology (PHAT) programme is underway and is progressing smoothly and according to plan. The software applications for the Day Care Services, Maternal and Child Welfare and the Abattoir/ Food Hygiene sections, along with the one for the Solid Waste Department have been completed. These computer software applications which have all been developed Softlogic Solutions, the software development company contracted by the council under the administration of the Division of Information Technology, have been completed and installed on the network file server of the City Council. The main features of the Day Care Management System caters for the processing of all information relevant to the children who are registered at the various centers and the fees collected for the use of these centers, For the Maternal & Child Welfare Management system, the core element deals with the processing of all information pertinent to registered women and children of the centers, including the monitoring of treatment provided and fees collected. With regards to the Abattoir/Food Hygiene Management System, focus was placed on the issuance of licenses and fees collected for the use of the abattoir's facilities and for the licenses issued to food establishments and food handlers. The management system for the Department of Solid Waste Management, caters for processing and management of data that is pertinent to all garbage being hauled and disposed of within the city. The PHAT programme will create an environment where public health related information access is simplistic, integrated, reliable, secure, remotely available and manageable. When fully implemented all the applications of the various sections of the Public Health Department will be accessible through a single workstation that is easy to use with the common user interfaces. Phase two of PHAT which is the creation of the databases is currently underway. Already those for the Solid Waste and Day Care systems have been completed.

### Prototipo 2.14 Visualizar Noticias

#### Ratepayer IVR under consideration

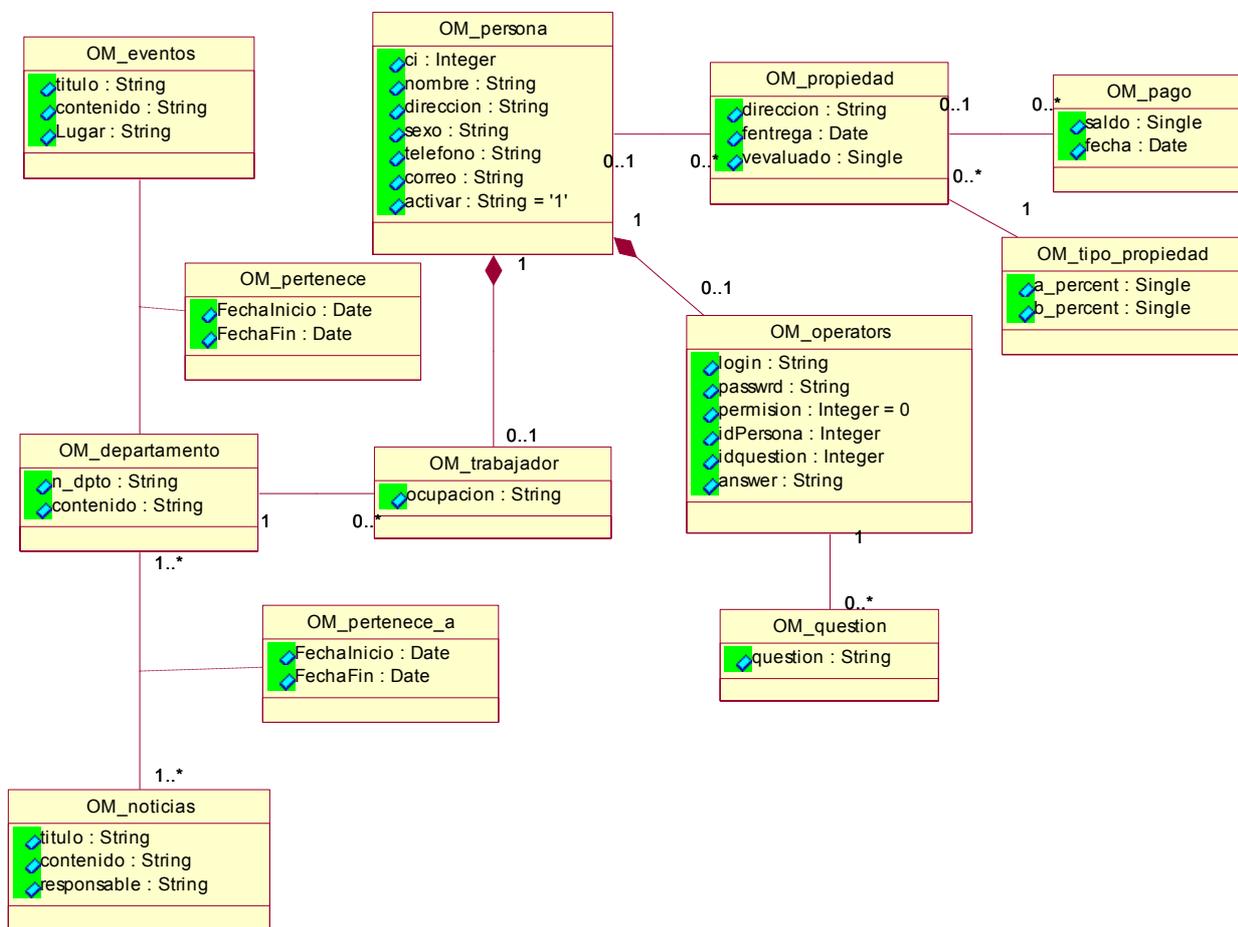
An Interactive Voice Response System which will provide a convenient way to access information about the status of one's general rate account by telephone in the comfort of one's home or place of work, on a 24 hour basis, is currently being examined by the Division of Information Technology. The benefits of such a system include allowing ratepayers to better manage their accounts, by being able to check their account balances on a regular basis and will eliminate the need to call or visit the Tax Office at City Hall to obtain billing information. The confidentiality of the property owner shall be maintained, as an access code, which will be peculiar to each property will always be required, to obtain this information. This service is also expected to accommodate property owners who reside overseas.

### Prototipo 2.15 Visualizar Eventos

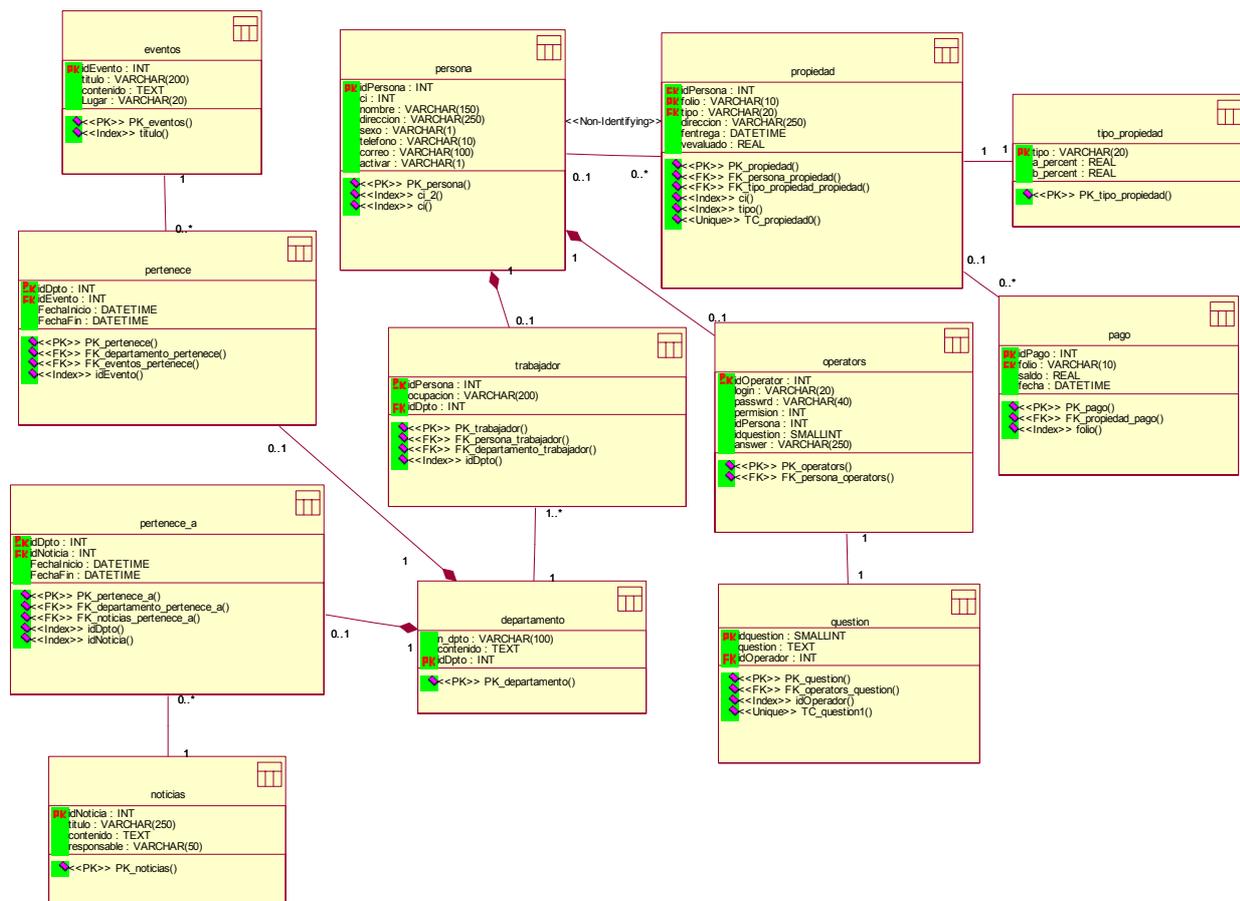


## Anexo III Diagramas de la base de Datos y de Implementación

### Anexo 3.1 Diagrama del modelo lógico de datos



### Anexo 3.2 Diagrama de Clases del modelo físico de datos.



1.10 Anexo 3.3 Diagrama de Implementación

