

SCI



SISTEMA DE CONTROL INMOBILIARIO

NATALY BERMUDEZ CASTIBLANCO

**CORPORACION UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERIA
PROGRAMA DE TECNOLOGIA EN INFORMATICA
BOGOTA. ENERO DE 2008**

SCI

SISTEMA DE CONTROL INMOBILIARIO

TRABAJO DE GRADO

NATALY BERMUDEZ CASTIBLANCO

Asesor de Proyecto

CARLOS ARMANDO LOPEZ SOLANO

CORPORACION UNIVERSITARIA MINUTO DE DIOS

FACULTAD DE INGENIERIA

PROGRAMA DE TECNOLOGIA EN INFORMATICA

BOGOTA. ENERO DE 2008

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Este trabajo va dedicado a mis padres Marlen y Jairo y a mi hermano Jhon Jairo ya que gracias a ellos ha sido posible alcanzar esta meta, a mis amigos incondicionales de carrera Alejandro y Armando por todo su apoyo y por supuesto a mi Universidad Minuto de Dios y todos los docentes de tecnología en informática.

Nataly Bermúdez Castiblanco

AGRADECIMIENTOS

Quiero expresar mis agradecimientos a mis compañeros de carrera quienes fueron de gran apoyo en la realización de este proyecto, a los docentes de el programa de tecnología en informática que de alguna manera fueron participes de este proyecto y a todas aquellas personas como familiares y amigos quienes me brindaron su apoyo para sacar adelante en este objetivo.

TABLA DE CONTENIDO

	Introducción	12
1.1	Tema	13
1.2	Título del proyecto	14
1.3	Planteamiento del problema	14
1.4	Justificación	16
1.5	Objetivos	17
1.5.1	Objetivo general	17
1.5.2	Objetivos específicos	17
1.6	Metodología de investigación	17
1.7	Linea de investigación	19
2.	Marco referencial	21
2.1	Estado del arte	29
2.2	Fundamentos teóricos	32
2.2.1	Lenguajes de programación	32
2.2.2	Técnicas de programación	33
2.2.3	Plataforma de desarrollo	37
2.2.4	Diagramas UML	38
2.2.5	Entorno gráfico	38

2.3	Referencia Organizacional	39
2.3.1	Misión	40
2.3.2	Visión	41
2.3.3	Estructura Organizacional	41
3.	Ingeniería del proyecto	42
3.1	Metodología y modelo de desarrollo	43
3.1.1	Ingeniería del proceso de negocio	43
3.1.2	Identificación de requisitos	44
3.1.3	Modelado del sistema	46
3.1.4	Validación de requisitos	46
4.	Análisis	48
4.1	Descripción detallada de cada proceso	48
4.1.1	Invitado	48
4.1.2	Propietarios	49
4.1.3	Arrendatario	49
4.1.4	Ventas	50
4.1.5	Cartera	52
4.1.6	Jurídica	52
4.1.7	Administrador	53
5.	Diseño	56
5.1	Diccionario de datos	56

5.1.2	Modelo entidad relación	61
5.2	Diagramas UML	62
5.2.1	Diagrama de casos de uso	62
5.2.2	Diagrama de clases	63
6.	Desarrollo	64
6.1	Especificaciones técnicas	64
6.1.1	Software	64
6.1.2	Hardware	65
6.2	Estructura del programa	65
6.3	Estructura del código fuente	65
7.	Pruebas	67
7.1	Pruebas de Análisis y Diseño	67
7.2	Pruebas de Código	67
7.3	Pruebas de Sistema	68
8.	Cronograma	70
9.	Glosario	71
10.	Conclusiones	79
11.	Sugerencias y recomendaciones	80
12.	Bibliografía	81
13.	Manuales.	82

13.1 Manual de Sistema.	82
--------------------------------	-----------

13.2 Manual de Usuario.	83
--------------------------------	-----------

INDICES

INDICE DE TABLAS

1. Tabla No 1 Contratos	56
2. Tabla No 2 Inmuebles	56
3. Tabla No 3 Movimientos	58
4. Tabla No 4 Permisos	58
5. Tabla No 5 Roles	59
6. Tabla No 6 tiposInmueble	59
7. Tabla No 7 Usuarios	59
8. Tabla No 8 Zonas	60
9. Tabla No 9 Cronograma	70

INDICE DE DIAGRAMAS

1. Diagrama 1. Modelo tradicional para aplicaciones Web comparado con el modelo AJAX.	28
2. Diagrama 2. Patrón modelo vista controlador.	34

3. Diagrama 3. Modelo Entidad Relación.	61
4. Diagrama 4. Diagrama de casos de uso.	62
5. Diagrama 5. Diagrama de Clases.	63
6. Diagrama 6. Estructura Organizacional.	41
7. Diagrama 7. Propietarios.	49
8. Diagrama 8. Arrendatario.	49
9. Diagrama 9. Vendedor.	50
10. Diagrama 10. Gerente Cartera.	52
11. Diagrama 11. Abogado.	52
12. Diagrama 12. Administrador.	53

INDICE FIGURAS

Figura No 1. Acceso a SCI	84
Figura No 2. Modulo Invitado	85
Figura No 3. Modulo Invitado, Botón info adicional	86
Figura No 4. Modulo Propietarios, Botón Mis propiedades	87
Figura No 5. Modulo Propietarios,Botón Registro	88
Figura No 6. Modulo Arrendatarios,Botón Contratos	89
Figura No 7. Modulo Ventas, Botón Propiedades	90
Figura No 8. Modulo Ventas,Botón Asesores	91

Figura No 9. Modulo Jurídica,Botón Propiedades	92
Figura No 10.Modulo Jurídica,Botón Contratos	93
Figura No 11. Modulo Admin,Botón Temas	94
Figura No 12. Modulo Admin,Botón Servidor Base de datos	95
Figura No 13. Modulo Admin,Botón Usuarios	96
Figura No 14. Modulo Admin, Registro de usuario	97
Figura No 15. Modulo Admin,Validación datos	98
Figura No 16. Modulo Admin,Botón compañía	99
Figura No 17. Modulo Admin,Botón Roles	100
Figura No 18. Modulo Admin, Editar rol de usuario	101
Figura No 19. Modulo Admin, Eliminar rol	102
Figura No 20. Modulo Admin, Botón Zonas	103
Figura No 21. Modulo Admin, Botón Tipos.	104

INTRODUCCIÓN

SCI es un proyecto por medio del cual se desarrollará una aplicación que tiene como objetivo sistematizar cada uno de los procesos necesarios que permitan la administración y el control de inmuebles y servicios inmobiliarios, ofreciendo a los usuarios la posibilidad vender, comprar o alquilar sus inmuebles.

El sistema de control inmobiliario comprenderá las siguientes áreas:

Gestión de usuarios: Control de registro de usuarios, Administración de usuarios.

Gestión de inmuebles: Venta de Inmuebles, Compra de Inmuebles, Alquiler de Inmuebles,

Valoración de Inmuebles, Administración de Inmuebles.

SCI será una aplicación orientada a la Web capaz de administrar inmuebles de carácter, comercial (locales comerciales) o residencial(casas, apartamentos), permitiendo organizar la información de cada uno de los clientes mediante el uso de bases de datos relacionales, todo esto con el fin de encontrar rápidamente posibles compradores/arrendadores que se ajusten a las especificaciones y accedan a las propiedades de forma responsable y cumplida además de crear una herramienta la cual brinde organización en la información que se genera por los procesos llevados a cabo en el sistema de control inmobiliario y evitando producir gran cantidad de formatos y registros impresos en papel ya que esto torna una dificultad en general a

futuro consultas que se requieran o administrar usuarios , inmuebles o control en los pagos de arrendamiento

También tendrá la capacidad de brindar información acerca de cada uno de los diferentes inmuebles permitiendo a los usuarios la publicación de éstos de manera ágil, clara y precisa, para la consulta de miles de potenciales clientes.

En el momento de acceder a servicios inmobiliarios la gente quiere encontrar propiedades que den solución a sus necesidades y por lo general buscan los métodos mas eficientes para cumplir estos fines, SCI será un software que dará respuesta a esta situación prestando un servicio ágil y personalizado.

1.1 TEMA

BASES DE DATOS RELACIONALES

Para llevar a cabo el desarrollo del sistema de control inmobiliario se hará referencia a las bases de datos relacionales, las cuales hablan acerca de los datos visibles al usuario. Estos datos están organizados por tablas de valores en las cuales operan las tareas de la base de datos.

Se puede decir que una relación logra parecerse a un archivo, una tupla a un registro, y un atributo a un campo, pero una relación no debe considerarse como "solo un archivo", sino mas bien como un archivo disciplinado, dando como resultado de esta una reducción de las estructuras de datos con las que el usuario interactúa, lo cual a su vez simplifica los operadores requeridos para manejar esas estructuras.

Teniendo en cuenta lo anterior se pueden mencionar distintas características de los archivos relacionales como las siguientes:

- Cada "archivo" contiene solo un tipo de registros
- Los campos no tienen un orden específico, de izquierda a derecha
- Los registros no tienen un orden específico, de arriba hacia abajo
- Cada campo tiene un solo valor

Los registros poseen un campo identificador único (o combinación de campos) llamada clave primaria.

De esta forma, todos los datos en una base de datos relacional se representan de una y solo una manera, a saber, por su valor explícito (esta se denomina en ocasiones principio básico del modelo relacional'). En particular, las conexiones lógicas dentro de una relación y entre las relaciones se representan mediante esos valores; no existen ligas o apuntadores visibles para el usuario, ni ordenamientos visibles para el usuario, ni grupos repetitivos visibles para el usuario, etc."

1.2 TITULO DEL PROYECTO

"**SCI**", Sistema de Control Inmobiliario, Se escogió este nombre para el proyecto porque describe de una forma clara y precisa las principales características del mismo.

1.3 PLANTEAMIENTO DEL PROBLEMA

Día a día las soluciones de software están mas presentes en cada uno de los sectores de la sociedad proporcionando sencillez, agilidad y competencia en el

momento de realizar todo tipo de tareas, esta situación hace que la tecnología desempeñe un papel importante en el diario vivir de las personas, de la misma forma se generan nuevas necesidades con la búsqueda de la sistematización de procesos que faciliten cada actividad realizada por el hombre y mas aun cuando se trata de asuntos laborales.

Sin embargo en la mayoría de las pequeñas y medianas empresas existen insatisfacciones en el manejo de información, pues no cuentan con el software especializado que les permita gestionar y organizar cada uno de los datos de manera eficiente, esto genera cierta desorganización a medida que cada registro va creciendo y a su vez se torna difícil el acceso a la información cuando se requiere de varias consultas. Este tipo de inconvenientes se ve claramente reflejado en el servicio que las empresas ofrecen a sus usuarios.

A partir de esta realidad y teniendo en cuenta que el objeto de estudio de este proyecto se centra en el sector inmobiliario el cual maneja una gran cantidad de información de suprema importancia, se establece la necesidad de proporcionar alternativas de solución que se involucren con la tecnología y obtengan el mayor provecho de ella. De la misma forma para el sector inmobiliario es vital que sus clientes dispongan de la capacidad y la facilidad de acceder a los diferentes servicios que aquí se prestan. Por lo cual surge la siguiente pregunta:

¿Qué importancia tiene la creación de nuevas herramientas de software que faciliten la sistematización, gestión y administración de la información en los procesos inmobiliarios?

1.4 JUSTIFICACIÓN

Actualmente las empresas requieren de mecanismos que les permitan almacenar información, generar reportes, realizar consultas, administrar clientes y muchas funciones mas, sin embargo en la mayoría de los casos es común encontrar que los mecanismos utilizados se alejan de la tecnología, tornando extenso y difícil cada uno de los procesos anteriormente mencionados. Ejemplo claro de esto son los archivadores físicos con contenidos innumerables de formatos y registros impresos en papel, a lo cual se le suma el espacio necesario para su alojamiento.

La gestión de una empresa inmobiliaria debe contar con las herramientas necesarias que permitirán realizar un seguimiento exhaustivo sobre cada factor que interviene en su negocio inmobiliario, esto con el fin de realizar satisfactoriamente una óptima labor y administración del negocio, tanto para uso interno en el manejo de la información, como para uso externo a través de la Internet permitiendo informar a los visitantes. Este tipo de herramientas se traduce en soluciones de software inmobiliario, el cual debe ser caracterizado por su fácil manejo para que cualquier persona con conocimientos básicos en informática pueda gestionar un negocio inmobiliario sin necesidad de perder mucho tiempo aprendiendo un sistema inmobiliario y sus funciones, permitiendo el fácil desempeño laboral tanto para las venta como para el alquiler de inmuebles. De la misma forma debe contar con la capacidad de manejar gran cantidad de información acerca de clientes, inmuebles y cada una de sus características exigiendo almacenamiento seguro que permita acceder a la información de manera confiable.

Por estos motivos existe la necesidad de crear una herramienta que brinde facilidad de manejo de información, generación de reportes, flexibilidad a la hora de trabajar y brindar el mejor de los servicios a cada uno de los clientes.

1.5 OBJETIVOS

1.5.1 Objetivo General.

Desarrollar una aplicación que brinde a usuarios como propietarios, vendedores(Asesores), arrendatarios y demás personas interesadas en el negocio del sector inmobiliario, los servicios de intermediación del mismo de una manera personalizada soportado en la tecnología orientada a la web, permitiendo el almacenamiento de información de inmuebles para su uso en la oferta y control de cartera.

1.5.2 Objetivos Específicos.

- Proporcionar un acceso seguro y en tiempo real a la información almacenada de los inmuebles y de sus propietarios y arrendatarios.
- Permitir un movimiento ágil y eficaz de la documentación disminuyendo el riesgo de pérdidas de la información referente a los inmuebles.
- Gestionar información con la que se puede evaluar la situación de los inmuebles ofertados y en arriendo.

1.6 METODOLOGÍA DE INVESTIGACIÓN

El objetivo de la metodología de investigación es adquirir conocimientos y efectuar la elección del método adecuado que permita desarrollar un proyecto del sistema de control inmobiliario con éxito, por tanto es importante tener un desarrollo de software controlado, donde estén definidos los pasos a seguir y sobre todo la utilidad de los mismos.

El primer paso siempre que se desea realizar una investigación o proyecto es generar una elección de tema en este caso sobre el sector inmobiliario definiéndolo con claridad y precisión allí es donde se encuentra formalmente la idea, donde se define en si lo que se va a realizar.

Para llevar a cabo este proceso hay que tener en cuenta los siguientes interrogantes:

Este es un problema realmente importante?, tendrá alguna utilidad inmediata el resultado de este proyecto?

La formulación del problema de la misma forma implica una delimitación del campo de investigación lo cual establece de la misma manera dentro de que parámetros y límites se desarrollará el sistema de control inmobiliario, y de ser así las probabilidades de no perder el trabajo realizado por decirlo de esa manera tienden a extenderse.

Los criterios que se deben tener en cuenta para el planteamiento adecuado son los siguientes:

El problema debe expresar una relación entre dos o más variables, el problema debe estar formulado claramente y como pregunta, y el planteamiento implica la posibilidad de prueba empírica es decir debe poder observarse en la realidad.

La metodología de investigación implementada en este proyecto será descriptiva - experimental, pues es ágil y esta orientada tanto al proceso como al resultado, busca definir claramente un objetivo y los puntos fuertes o débiles del proyecto. Este tipo de investigación genera datos de primera mano para realizar después un

análisis general, con lo cual se podrá hacer una descripción de todos los requerimientos que se desean cubrir así como de todos los procesos que se realizarán para llegar al cumplimiento del objetivo de este proyecto.

De acuerdo con lo anterior el primer paso será siempre la recolección de información primaria que pueda servir como base de análisis y para esto la información se recopilará mediante técnicas como la observación en investigaciones de campo, se estudiará la asociación o relación entre el contexto estructural del área inmobiliaria y sus usuarios o visitantes, además de la interacción que existe entre estos.

La experimentación consiste en analizar uno o varios elementos del entorno y su mercado (precio, servicio, calidad, publicidad) con el fin de generar datos acerca de reacciones y comportamiento del usuario o población involucrada. Con lo cual se busca identificar el impacto de SCI sobre la conducta del mercado, determinando si es positivo o negativo.

1.7 LINEA DE INVESTIGACIÓN

La tecnología en informática ocupa un papel importante y cada día más imprescindible en sectores tan variados como: comercio, industria, gobierno, medicina, educación, entretenimiento y la sociedad en general. Los desarrolladores de software contribuyen, mediante la participación directa o enseñanza, al análisis, especificación, diseño, desarrollo, certificación, mantenimiento y pruebas de sistemas de software.

Dentro de Uniminuto la facultad de ingeniería esta suscrita a la tercera (3) Línea de investigación (Innovaciones tecnológicas y cambio social), a su vez el departamento

de informática proporciona para cada programa sublíneas de investigación. En el programa de Tecnología en Informática y Telecomunicaciones existen las siguientes sublíneas de investigación:

Sublínea 1 - Desarrollo de Software

Sublínea 2 - Sistemas de Información

Sublínea 3 - Redes de Computadores

Sublínea 4 - Plataformas

SCI adoptará la primera sublínea de Investigación “Desarrollo de Software”. Se ha elegido esta sublínea de investigación debido al enfoque del programa de Tecnología en Informática y Telecomunicaciones, de acuerdo a su orientación y énfasis hacia el desarrollo de aplicaciones en diferentes lenguajes de programación.

2. MARCO REFERENCIAL

Un sistema de información es el conjunto de recursos que permiten recoger, gestionar, controlar y difundir la información de toda una empresa u organización en este caso hablamos del sistema de control inmobiliario.

Desde los años setenta, los sistemas de bases de datos han ido reemplazando a los sistemas de ficheros en los sistemas de información de las empresas. Al mismo tiempo, se ha ido reconociendo la gran importancia que tienen los datos que éstas manejan, convirtiéndose en uno de sus recursos más importantes. Esto ha hecho que muchas entidades tengan departamentos que se encarguen de gestionar toda su información, que estará almacenada en una base de datos. Es allí donde aparecen los papeles de administrador de datos y administrador de la base de datos quienes son los encargados de supervisar y controlar todas las actividades relacionadas con los datos de la empresa y con el ciclo de vida de las aplicaciones de bases de datos.

Un sistema de información está formado por los siguientes componentes:

- La base de datos.
- El SGBD.
- Los programas de aplicación.
- Los dispositivos físicos (ordenadores, dispositivos de almacenamiento, etc.).
- El personal que utiliza y que desarrolla el sistema.

La base de datos es un componente fundamental de un sistema de información. El ciclo de vida de un sistema de información está ligado al ciclo de vida del sistema de base de datos sobre el que se apoya. Al ciclo de vida de los sistemas de información también se le denomina ciclo de vida de desarrollo de software. Las etapas típicas del ciclo de vida de desarrollo del software son: planificación, recolección y análisis de los requisitos, diseño, creación de prototipos, implementación, prueba, conversión y mantenimiento.

Este ciclo de vida hace énfasis en la identificación de las funciones que realiza la empresa y en el desarrollo de las aplicaciones que lleven a cabo estas funciones. Se dice que el ciclo de vida de desarrollo del software sigue un enfoque orientado a funciones, ya que los sistemas se ven desde el punto de vista de las funciones que llevan a cabo. Por esta razón, el análisis estructurado hace énfasis en los diagramas de flujo de datos, siguiendo el movimiento de los datos a través de una secuencia de transformaciones, y refinando éstas a través de una serie de niveles. Lo mismo ocurre en el diseño estructurado, que ve a un sistema como una función que se descompone sucesivamente en niveles o subfunciones.

Concentrándose en las funciones se infravaloran los datos y, en especial, la estructura de los datos que son manipulados por las funciones. El resultado es que estos sistemas tienen valor durante poco tiempo en relación con las necesidades de los usuarios a largo plazo. Esto sucede debido a que al poco tiempo de haber instalado un sistema, las funciones implementadas son en realidad un subconjunto de las funciones que los usuarios realmente desean. Casi inmediatamente, los usuarios descubren una gran variedad de servicios adicionales que quisieran incorporar al sistema. Estas necesidades causan problemas a los sistemas

obtenidos con un diseño orientado a funciones, puesto que este diseño puede requerir una revisión importante para acomodar las funciones adicionales.

En contraste, el enfoque orientado a datos se centra en el análisis de los datos utilizados por las funciones. Esto tiene dos ventajas. La primera es que los datos son una parte considerablemente más estable que las funciones. La segunda ventaja es que la propia estructura de un esquema de base de datos requiere de un análisis sofisticado de los datos y de sus relaciones. Una vez que se haya construido un esquema para la base de datos que sea lógico, podrían diseñarse tantas funciones como fuera necesario para sacar provecho del mismo. Sin embargo, sin un esquema tal, la base de datos sólo podría ser útil para una única aplicación. Por lo tanto, el enfoque orientado a funciones puede ser bueno para el desarrollo a corto plazo, pero pierde su valor real a largo plazo. Usando un enfoque orientado a datos, los datos pasan a ser las bases sobre las cuales se puede construir una gran variedad de funciones diferentes.

Las bases de datos relacionales son las que han tenido mayor uso comercial; cada una de las tablas de las que se habla anteriormente lleva un nombre de una relación y contiene filas y columnas y nuevas tablas pueden ser constituidas a partir de ellas mediante registros y campos y la manera de poder acceder a cualquiera de ellos es por medio de un campo llamado llave el cual se elige después de analizar cada uno de los campos del registro.

Para llevar a cabo una óptima aplicación del sistema de control inmobiliario hablando ya de otras herramientas a trabajar se utilizará lo que se denomina una unión entre varias tecnologías las cuales darán grandes resultados, se habla de AJAX hoy en día una herramienta de gran uso en las aplicaciones Web.

Gracias a esto se puede realizar un sitio completo sin la necesidad de recargar la página y solo cargando las partes que se requirieren. La tecnología usada es el XMLHTTPREQUEST, lo que se conoce recientemente con el nombre de AJAX. A continuación podremos ver primero como cargar un contenido, luego como interactuar con php y enviar pequeñas partes de información y para terminar como enviar información mas grande.

CREANDO EL OBJETO XMLHTTPREQUEST:

Realizar una función genérica es la mejor idea para este modelo. A base de try y catch se puede realizar una función cross-browser

```
function nuevoAjax() {
    var xmlhttp=false;
    try {
        xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            xmlhttp = new
ActiveXObject("Microsoft.XMLHTTP");
        } catch (E) {
            xmlhttp = false;
        }
    }

    if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
        xmlhttp = new XMLHttpRequest();
    }
    return xmlhttp;
}
```

Primer ejemplo; cargar datos externos.

La función de cargar contenido se definirá de la siguiente manera:

```

function cargarContenido(){
    var contenedor;
    contenedor = document.getElementById('contenedor');
    ajax=nuevoAjax();
    ajax.open("GET", "ejemploajax1.html",true);
    ajax.onreadystatechange=function() {
        if (ajax.readyState==4) {
            contenedor.innerHTML = ajax.responseText
        }
    }
    ajax.send(null)
}
window.onload= function(){cargarContenido()}

```

Debemos tener en cuenta varias cosas en este punto por ejemplo, primero obtenemos un elemento que tenga como id 'Contenedor', después se crea un nuevo objeto ajax , luego se carga el archivo html por el método GET y lo importante aquí es qué hacer cuando se cargue, esto es onreadystatechange. Readystate 4 significa que ya acabó de cargarlo Finalmente el método send es realizar la acción y llamar la función en el window.onload

Segundo ejemplo: Envío de datos por el método GET

Lo siguiente es enviar pequeñas cantidades de datos, por ejemplo de un input text con límite a 255 caracteres, esto lo haremos por el método GET. Utilizaremos también la función crearAjax y modificaremos un poco cargarContenido.

La idea es meter un formulario, saber los valores de los input que incluye el formulario y enviarla a php. El código modificado de cargarContenido sería:

```

function cargarContenido(){
    var t1, t2, contenedor;
    contenedor = document.getElementById('contenedor');

```

```

t1 = document.getElementById('texto1').value;
t2 = document.getElementById('texto2').value;
ajax=nuevoAjax();
ajax.open("GET", "ejemploajax2.php?t1="+t1+"&t2="+t2,true);
ajax.onreadystatechange=function() {
    if (ajax.readyState==4) {
        contenedor.innerHTML = ajax.responseText
    }
}
ajax.send(null)
}

```

El archivo php sólo contiene este código:

```

<?
    echo "Datos en GET: <pre>";
    print_r($_GET);
    echo "</pre>";

    echo "Datos en POST: <pre>";
    print_r($_POST);
    echo "</pre>";
?>

```

Tercer ejemplo: Envío de datos por el método POST

Por último y para terminar, enviar datos por el método post. Habrá que cambiar algunas cosas en la función cargarContenido:

Debemos agregar una línea adicional: setRequestHeader especifica qué tipo de datos llegarán al servidor. Asimismo especificamos el método en el método open y ahora sí utilizaremos parámetros para send.

La función quedaría de la siguiente forma:

```

function cargarContenido(){
    var t1, t2, contenedor;
    contenedor = document.getElementById('contenedor');
}

```

```
t1 = document.getElementById('texto1').value;
t2 = document.getElementById('texto2').value;
ajax=nuevoAjax();
ajax.open("POST", "ejemploajax2.php", true);
ajax.onreadystatechange=function() {
    if (ajax.readyState==4) {
        contenedor.innerHTML = ajax.responseText
    }
}
ajax.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
ajax.send("t1="+t1+"&t2="+t2)
}
```

NOTAS:

- Por algún extraño motivo, el objeto que es usado una vez ya no se puede usar otra vez. Por eso en la función cargarContenido siempre se llama a la función para crear un nuevo objeto ajax. Si no entienden de qué hablo, creen el objeto fuera de la función y verán cómo sólo funciona una vez
- Enviar y recibir datos con acentos/símbolos extraños es un verdadero problema. Básicamente, en php es la utilización de utf8_decode cuando se reciben y de utf8_encode cuando se envían.

¹Teniendo en cuenta lo anterior la herramienta AJAX incorpora:

1. presentación basada e estándares usando XHTML y CSS;
2. exhibición e interacción dinámicas usando el Document Object Model ;
3. Intercambio y manipulación de datos usandoXML and XSLT;
4. Recuperación de datos asincrónica usando XMLHttpRequest yJavaScript poniendo todo junto.

1 <http://www.cristalab.com/tutoriales/165/introducción-a-ajax-con-php-y-formularios>

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz tienen un requerimiento HTTP al servidor web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una página HTML al cliente.

Este es un modelo adaptado del uso original de la Web como un medio hipertextual, lo que hace a la Web buena para el hipertexto, no la hace necesariamente buena para las aplicaciones de software.



Diagrama 1: El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der.).

Para tener una breve idea de lo que es AJAX se pueden ver las cosas que se han construido con esta unión (AJAX) como lo es Google maps, Gmail, Outlook Web Access entre otras aplicaciones ya conocidas.

AJAX es acrónimo para Asynchronous JavaScript + XML (lo que representa un cambio fundamental en que es posible en la Web) lo cual significa en resumen cargar y renderizar una página luego mantenerse en esa página mientras rutinas y scripts van al servidor en busca de los datos que se usan para actualizar la página en background solo mostrando u ocultando partes de la página .

La clase Ajax de Weblinx(MApache) permite la creación de código javascript para la utilización del xmlhttpobject. Adicionalmente cuenta con tres métodos(GETMethodSend(), POSTMethodSend() y tableInfoPOSTMethodSend()) standar que permiten el uso de scripts php en la ejecución de procesos sin la necesidad de cambiar de dirección url (evitando así el refresco de las páginas)

2.1 ESTADO DEL ARTE

En la actualidad existen diversos tipos de software que ofrecen al usuario las facilidades y diversas formas para llevar a cabo un proceso.

En el caso de sistemas para inmobiliarias podemos ver un ejemplo en un software llamado fincasoft el cual gestiona inmuebles de manera virtual de forma ágil y cómoda, se puede acceder vía Web a una zona restringida donde el administrador puede añadir, modificar y eliminar diferentes inmuebles que quiera ofrecer. Este software es totalmente personalizable donde se manejan aspectos como colores, logotipos, normativas gráficas, entre otras.

Fincasoft es un sistema sencillo de manejar como se indica anteriormente pero se enfoca únicamente en las agencias inmobiliarias que quieran aprovechar esta herramientas basadas en el mercado vía Internet sin brindar la oportunidad a otros tipos de usuarios realizar tareas o procesos como lo son ofertas de inmuebles y modificaciones de las mismas y búsqueda según su interés, sus recursos en conclusión a este software únicamente pueden acceder aquellas personas las cuales tengan algún vínculo con dichas entidades de inmuebles para poder realizar la gestión.

Existe en el mundo del Internet otro tipo de software llamado Second life el cual es un tipo de mundo virtual al cual se accede por medio de una instalación de un programa en el su equipo como por ejemplo una instalación de un juego.

Acompañado a este sistema viene Anshe Chung el cual es el responsable de realizar las negociaciones de compra de parcelas a bajo precio y realizar construcción en ellas de hoteles, apartamentos, casas etc. y luego vender o arrendar.

En Second el espacio que se maneja se divide en dos partes en las cuales puede haber tierra o una parcela o mar, allí se realizaran compras para generar edificaciones por un valor determinado o valor mensual o hacer algún tipo de reserva de la parcela o mar donde en un futuro habrá tierra.

Para ello también se llevar a cabo subastas por medio de la pagina Web teniendo un monto base con respecto a las características del lugar.

Este software hace énfasis en la especulación de suelo y no se manejan otro tipo de información general del sector inmobiliario y no dispone de módulos a los cuales los

usuarios puedan acceder para generar sus ofertas o realizar administración de los mismos.

www.Infocomercial.com es un lugar donde se pueden generar todo tipo de negocios entre ellos la administración de portales inmobiliarios el cual trae múltiples beneficios para el usuario como incorporación de propiedades al sitio Web, panel de control mediante el cual podrá realizar la administración del sistema de manera rápida.

Este sistema se adapta a las necesidades de cualquier empresa, del mercado y de esta misma forma pueden definir los tipos de inmuebles que comercializan y sus diversas características, precio entre otras.

Allí se pueden incluir imágenes de los inmuebles que se van a comercializar planos los cuales están disponibles para realizar distintas vistas.

Así mismo es de fácil instalación ya que es un sitio Web el cual no requiere de complejos programas únicamente se necesita estar conectado a Internet para poder acceder al servicio.

Se puede apreciar que este sistema tiene muchas ventajas pero de la misma forma que se indicaba del software anterior únicamente es para empresas las cuales puedan realizar sus ofertas al mercado y estos serian sus únicos usuarios además tiene un limite de ingreso de registros por un valor especifico a cancelar mensualmente.

2.2 FUNDAMENTOS TEÓRICOS

2.2.1 Lenguajes de Programación.

PHP:

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. El nombre es el acrónimo recursivo de "PHP: Hypertext Preprocessor" (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, GNU/LINUX, Windows y Mac OS X.

PHP es hoy el lenguaje más utilizado en el mundo gracias a su baja curva de aprendizaje y a que en sus últimas versiones se hace énfasis en el desarrollo del paradigma de la programación orientada a objeto

El sistema de administración para base de datos que se utilizara para el desarrollado del sistema de control inmobiliario es MySQL (rdbms) ya que provee una solución robusta a los usuarios con poderosas herramientas multi-usuario, soluciones de base de datos SQL (structured Query Language) multi-threaded. Es rápido, robusto y fácil de utilizar.

2.2.2 Técnicas de Programación.

Programación Orientada a Objetos:

SCI será desarrollado bajo el estilo de programación orientado a objetos, debido a que día a día estas técnicas toman mas fuerza y sus metodologías se antepone a los modelos clásicos de programación. La P.O.O presenta las siguientes ventajas:

Las tecnologías de objetos llevan a reutilizar, y la reutilización (de componente de software) lleva a un desarrollo de software más rápido y a programas de mejor calidad.

El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente poco acoplada. Esto lleva a menores efectos colaterales cuando se deben hacer cambios y provoca menos frustración en el ingeniero del software y en el cliente. Además, los sistemas orientados a objetos son más fáciles de adaptar y más fácilmente escalables (por ejemplo: pueden crearse grandes sistemas ensamblando subsistemas reutilizables).

Patrón Modelo Vista Controlador (MVC):

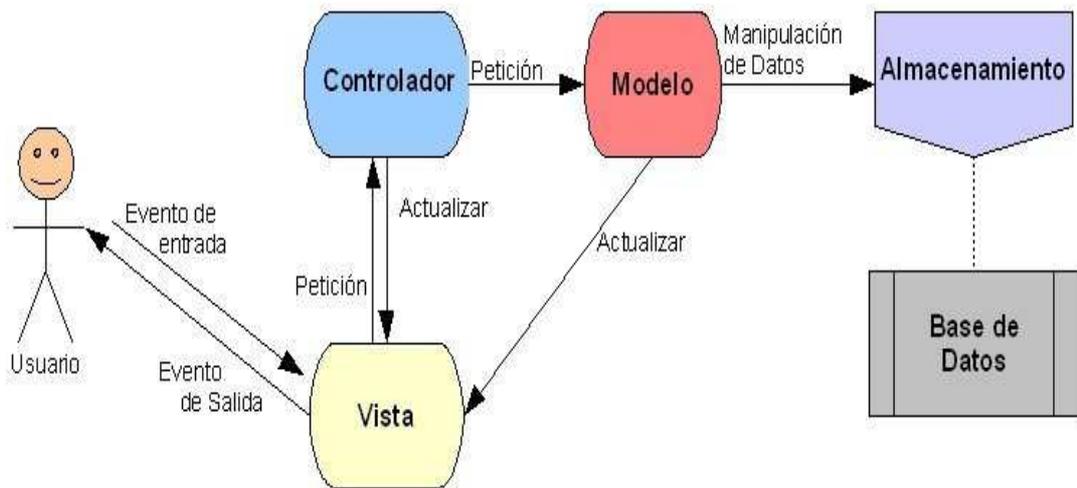


Diagrama 2 – Modelo, Vista, Controlador

La codificación de SCI será distribuida bajo el patrón Modelo-Vista-Controlador (MVC) que usualmente es implementado en aplicaciones orientadas a la web. Este patrón ofrece grandes ventajas para el programador a la hora de desarrollar una aplicación, una de ellas y seguramente la mas importante es que, el MVC permite la reutilización de código fuente de una manera mucho mas fácil debido a que lo separa en varias partes, esto nos da la posibilidad de manipular, modificar o incluir el código necesario de una forma independiente, sin comprometer el resto del programa.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. , donde la vista es la página HTML, el control es el código que provee de datos dinámicos a la página, y el modelo contiene clases

representativas de la aplicación (como el mensaje de un foro, un miembro registrado, etc.).

- Modelo: Ésta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos; por ejemplo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.
- Vista: Éste presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.
- Controlador: Éste responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.
- Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente (como puede ser una base de datos) para almacenar los datos. MVC no menciona específicamente esta capa de acceso a datos.

SCI será desarrollado bajo un estilo de programación desarrollado en la comunidad de software libre Arca-Csl se trata de MApache

Mapache esta compuesto por un Paquete de clases que integran código HTML y JavaScript necesario para la construcción de cualquier interfaz gráfica compuesta por tablas, botones, imágenes formularios entre otros.

Esto permitirá que el código de la aplicación sea limpio y más ordenado, es decir que no es necesario mezclar Php con Html o JavaScript, el Html o JavaScript tan solo se incluye en el código como una librería.

La aplicación creada bajo lenguajes como PHP, Python o EMBPERL deben ser puestas en un directorio especial que es tomado por APACHE como su raíz, luego dependiendo de los mods que disponga el servidor se mostrará en un navegador sin iconos la ejecución de la misma.

Los "mods" de APACHE (servidor web de distribución libre y de código abierto para plataformas unix el cual genera fiabilidad y estabilidad) permiten interpretar diversos lenguajes, así que pueden haber aplicaciones desarrolladas con varios lenguajes. Si se desea trabajar con aplicaciones Stand Alone se puede llamar la aplicación mediante la dirección localhost y si se está en una red solo se requiere llamarla mediante la dirección IP del servidor que la contiene. Y si se desea tenerla en la Internet solo se requiere tener un servicio de hosting con soporte APACHE PHP y/o EMBPERL (Hoy día son muy comunes) en donde pondríamos nuestra aplicación. Así que una aplicación podría tener estas tres modalidades sin tener que modificarle absolutamente nada, a eso es lo que se llama MApache.

Adicionalmente MApache presenta un grupo de clase que buscan mejorar la organización interna de los scripts dando énfasis en el desarrollo de aplicaciones con el paradigma de programación orientada a objetos.

En estos momentos la versión más reciente es la MApache V 0,0,2 la cual incluye clases para el manejo de consultas a bases de datos, creación de gráficos mediante la librería GD, paquetes de sistemas de edición y grupos de validaciones de formularios. Prácticamente todo el código de etiquetas html está configurado dentro de las clases.

2.2.3 Plataforma de Desarrollo.

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. 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, además tiene amplia aceptación en la red, 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 (estadísticas históricas y de uso diario proporcionadas por Netcraft).

Apache no solo tiene la capacidad de comportarse como servidor web, sino que también puede comportarse como maquina virtual para aplicaciones stand alone, además esta diseñado para la inclusión de módulos externos entre los cuales se encuentran: mod_perl (Páginas dinámicas en Perl), mod_php (Páginas dinámicas en PHP), mod_python (Páginas dinámicas en Python), mod_ruby (Páginas dinámicas en Ruby). Esta plataforma permitirá que orientador digital tenga versatilidad y pueda ofrecer grandes soluciones a los usuarios.

Para el sistema operativo Windows se trabajara con Appserv o Xampplite, siendo estas herramientas OpenSource, que facilitan la instalación de Apache, MySQL y PHP en una sola herramienta, esta característica facilita la tarea al usuario ya que se configuran las aplicaciones de forma automática. Como extra incorpora phpMyAdmin para el manejo de MySQL

2.2.4 Diagramas UML

UML es una herramienta de diseño mediante la cual se realizan diagramas que permiten crear aplicaciones, dichos diagramas se denominan diagramas UML y se usan especialmente para aplicaciones creadas bajo la programación orientada a objetos

Existen distintos tipos de diagramas uml los cuales se utilizan según la finalidad de la aplicación y su análisis por ejemplo si se desea analizar la entrada y el uso de la aplicación podemos usar el diagrama denominado diagrama de casos de uso, en este se muestran las opciones de acceso a distintas partes de la aplicación.

2.2.5 Entorno Gráfico

Una interfaz es el elemento que hace posible la comunicación entre dos entes. La interfaz se necesita cuando los dos entes no pueden comunicarse directamente por que se expresan en lenguajes diferentes o mediante señales de especificaciones diferentes. Las interfaces gráficas de usuario son aquellas que incluyen cosas como menús, ventanas, teclado, ratón, los beeps, y algunos, otros sonidos que la computadora hace, en general, todos aquellos canales por los cuales se permite la comunicación.

Gracias a la configuración de PHP, es posible el diseño de aplicaciones con interfaces gráficas utilizando la librería GTK+ o el lenguaje de etiquetas **html** junto con **Javascript**, esto facilita mucho las cosas a la hora de crear aplicaciones ya que el lenguaje de etiquetas html es de muy fácil manejo y **PHP** se acondiciona muy bien. Se usara html para la creación de entornos gráficos complementado con hojas

de estilo en cascada (*Cascading Style Sheets*, CSS), las ventajas de utilizar CSS son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto, con lo que aumenta considerablemente la accesibilidad.
- Una página puede disponer de diferentes hojas de estilo.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

2.3 REFERENCIA ORGANIZACIONAL

Anteriormente, las organizaciones se administraban desde el punto de vista de la permanencia, en donde la regla era la estabilidad en que ocurrirán los eventos y la excepción, los cambios que producían. Sin embargo, en la actualidad, se han presentado transformaciones tan radicales, en donde el cambio es la regla y la excepción es la estabilidad.

Cabe destacar que esta era se ha caracterizado por las paradojas y la complejidad de contradicciones, entendidas y atendidas con una visión más amplia y adecuada, porque las organizaciones presentan modelos y propuestas que deben ser revisados a profundidad, para obtener nuevas respuestas, tomando en cuenta el

medio donde se desenvuelve que se ha tornado más complejo e incierto, es decir, más interrelacionado, diverso y en continuo cambio.

Los elementos fundamentales del cambio son el factor humano y el aprendizaje Organizacional que actualmente se implementan como estrategias del desarrollo organizacional y la búsqueda de ventajas competitivas las cuales se han denominado "paradigma de la complejidad", resaltando la disposición para aprender y aprehender valores, actitudes y experiencias, en donde se hace fundamental la aplicación de las ciencias cognitivas demostrando que los sistemas de pensamiento son los llamados a determinar la acción y las capacidades de cambio.

En ese sentido, las nuevas tecnologías se orientan hacia el ejercicio de los principios más elementales de la disciplina, la flexibilización de las formas , métodos y organismos, es decir, la aplicación del diseño organizacional como herramienta para atender la complejidad, proporcionando innovación en el entorno que exige mayor velocidad de respuesta, capacidad de renovación rápida de productos, procesos, habilidades y competencias que conllevan a un aprendizaje más dispuesto, de allí que el conocimiento se convierte en el recurso estratégico y el aprendizaje, en la llave del proceso que garantiza una ventaja competitiva sostenida.

2.3.1 MISION:

La misión gerencial de C.I. Promover es construir una empresa líder, creciente y rentable en la prestación de servicios al sector publico y privado a nivel nacional en el campo de Proyectos de Inversión, Inmobiliario, jurídico, contable y de la Ingeniería y Arquitectura.

Nuestro deber primordial consiste en fortalecer la responsabilidad social, fomentar

servicios de alta calidad con respaldo de mercadeo y tecnología que satisfagan a nuestros clientes y poder ganar el respeto y lealtad de los empleados y accionistas.

2.3.2 VISION:

C.I. Promover será promotor de proyectos en el mercado inmobiliario con actitud emprendedora, perseverante y de adaptabilidad al cambio, brindando alternativas y soluciones estratégicas al sector de la construcción y los servicios encaminadas al mejoramiento de la calidad de vida de las familias colombianas.

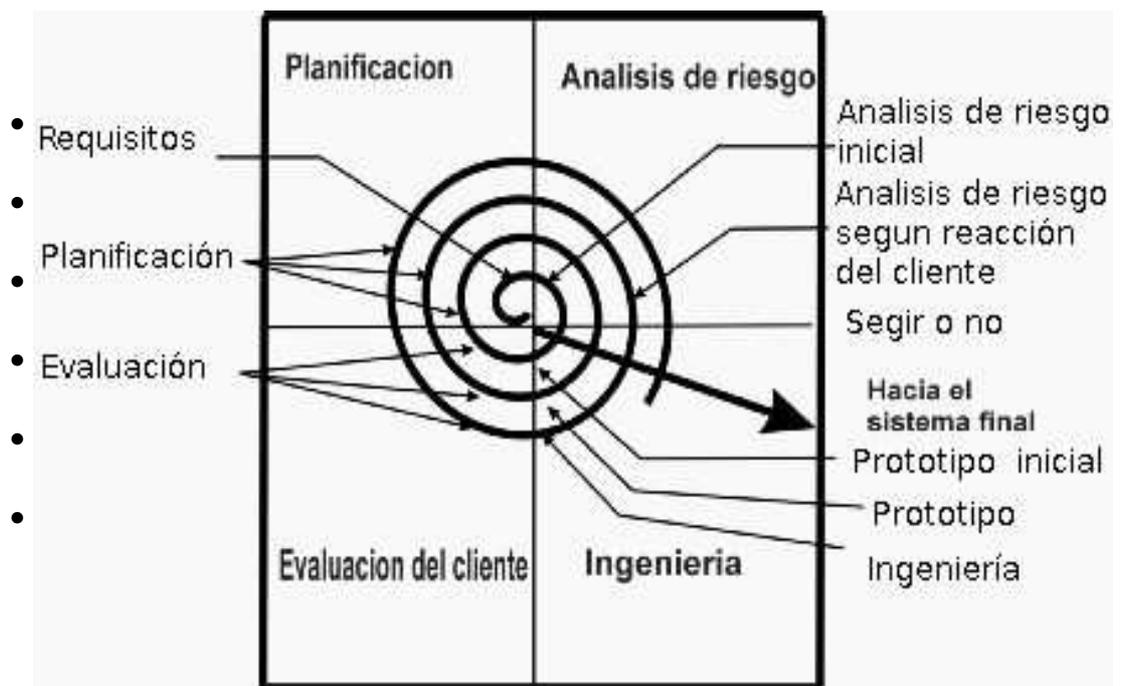
2.3.3 ESTRUCTURA ORGANIZACIONAL:



Diagrama 6 . Estructura Organizacional.

3. INGENIERÍA DEL PROYECTO

SCI será desarrollado utilizando el ciclo de vida espiral con el fin de cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo el análisis de riesgo. El modelo representado mediante la espiral, define cuatro actividades principales:



- Planificación: determinación de objetivos, alternativas y restricciones

- Análisis de riesgo: análisis de alternativas e identificación y resolución de riesgos.
- Ingeniería: desarrollo del producto del "siguiente nivel",
- Evaluación del cliente: Valorización de los resultados de la ingeniería.

²Este modelo permitirá a los desarrolladores poder reaccionar a los riesgos en cada nivel evolutivo. También utiliza la creación de prototipos como un mecanismo de reducción de riesgo. .

3.1 METODOLOGÍA Y MODELO DE DESARROLLO

3.1.1 Ingeniería del Proceso de Negocio.

El objetivo de la ingeniería de proceso de negocio es definir arquitecturas que permitan a las empresas emplear la información eficazmente.

Cuando se habla de una visión general de las necesidades de tecnología de información de una compañía, existen pequeñas incertidumbres que son planteadas a ingeniería de sistemas. La ingeniería de proceso de negocio es un acercamiento para crear un plan general para implementar la arquitectura de computación.

Se deben analizar y diseñar tres arquitecturas diferentes dentro del contexto de objetivos y metas de Negocio:

³La arquitectura de datos proporciona una estructura para las necesidades de Información de un negocio o de una función de negocio. Los ladrillos de la

2 <http://sistemas.itlp.edu.mx/tutoriales/analisis/24.htm>

3 <http://www.monografias.com/trabajos5>

arquitectura son los objetos de datos que emplea la empresa. Un objeto de datos contiene un conjunto de atributos que definen aspectos, cualidades, características o descriptor de los datos que han sido descritos.

La arquitectura de aplicación comprende aquellos elementos de un sistema que transforman objetos dentro de la arquitectura de datos por algún propósito del negocio. En el contexto de este libro, consideramos normalmente que la arquitectura de aplicación es el sistema de programas (software) que realiza esta transformación. Sin embargo, en un contexto más amplio, la arquitectura de aplicación podría incorporar el papel de las personas (por ejemplo, cliente servidor) que ha sido diseñado para implementar estas tecnologías.

La infraestructura tecnológica proporciona el fundamento de las arquitecturas de datos y de aplicaciones. La infraestructura comprende el hardware y el software empleados para dar soporte a las aplicaciones y datos. Esto incluye computadoras y redes de computadora, enlaces de telecomunicaciones, tecnologías de almacenamiento y la arquitectura (por ejemplo, cliente servidor) diseñada para implementar estas tecnologías.

3.1.2 Identificación de Requisitos

Problemas de alcance. El límite del sistema está mal definido o los detalles técnicos innecesarios, que han sido aportados por los clientes/usuarios, pueden confundir más que clarificar los objetivos del sistema.

Problemas de comprensión. Los clientes/usuarios no están completamente seguros de lo que necesitan, tienen una pobre comprensión de las capacidades y limitaciones

de su entorno de computación, no existe un total entendimiento del dominio del problema, existen dificultades para comunicar las necesidades al ingeniero del sistema, la omisión de información por considerar que es «obvia», especificación de requisitos que están en conflicto con las necesidades de otros clientes/usuarios, o especificar requisitos ambiguos o poco estables.

Problemas de volatilidad. Los requisitos cambian con el tiempo.

Para ayudar a solucionar estos problemas, profesionales en el área de sistemas e informática deben aproximarse de una manera organizada a través de reuniones para definir requisitos. Se sugieren un conjunto de actuaciones para la obtención de requisitos, que están descritos en las tareas siguientes:

- Valorar el impacto en el negocio y la viabilidad técnica del sistema propuesto.
- Identificar las personas que ayudarán a especificar requisitos y contrastar su papel en la organización;
- Definir el entorno técnico (arquitectura de computación, sistema operativo, necesidades de telecomunicaciones) en el sistema o producto a desarrollar e integrar;
- Identificar “restricciones de dominio” (características específicas del entorno de negocio en el dominio de la aplicación) que limiten la funcionalidad y rendimientos del sistema o producto a construir;
- Definir uno o más métodos de obtención de requisitos (entrevistas, grupos de trabajo, equipos de discusión);

- Solicitar la participación de muchas personas para que los requisitos se definan desde diferentes puntos de vista, asegurarse de identificar lo fundamental de cada requisito registrado;
- Identificar requisitos ambiguos como candidatos para el prototipado, y crear escenarios de uso para ayudar a los clientes/usuarios a identificar mejor los requisitos fundamentales.

3.1.3 Modelado del Sistema

Con el modelo será relativamente fácil asegurar la eficiencia del trabajo, la estética visual. Es importante evaluar los componentes del sistema y sus relaciones entre sí; determinar cómo están reflejados los requisitos, y valorar como se ha concebido la estética en el sistema.

Todos los sistemas pueden modelarse como una transformación de la información empleando una arquitectura del tipo entrada proceso salida.

Mediante la representación de entrada, proceso, salida, tratamiento de la interfaz de usuario y de auto comprobación, un ingeniero de sistemas puede crear un modelo de componentes de sistema que establezca el fundamento para análisis de requisitos posteriores y etapas de diseño en cada una de las disciplinas de ingeniería.

3.1.4 Validación de Requisitos

El resultado del trabajo realizado es una consecuencia de la ingeniería de requisitos (especificación del sistema e información relacionada) y es evaluada su calidad en

la fase de validación. La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto.

4. ANÁLISIS

4.1 DESCRIPCIÓN DETALLADA DE CADA PROCESO

En una inmobiliaria los procesos son realizados por los siguientes actores:

Propietarios, Compradores, vendedores, Arrendatarios, arrendador, asesores comerciales (personas encargadas de administrar el sistema).

El sistema de control inmobiliario constará de los siguientes módulos:

- **Invitado.**
- **Propietarios.**
- **Arrendatario.**
- **Ventas.**
- **Cartera**
- **Jurídica**
- **Administrador.**

4.1.1 Invitado:

Un cliente propietario tiene la opción de solicitar la información sobre inmuebles teniendo en cuenta que tipo de usuario es, por ejemplo si es un arrendatario puede solicitar información sobre su inmueble arrendado, puede confirmar cual es el dueño de dicho inmueble y al igual verificar o solicitar informe de pago de arriendo.

Ya si es un dueño de algún inmueble puede solicitar información sobre sus inmuebles tal información puede ser arrendatario, el vendedor que fue asignado para la gestión del inmueble y informe de los pagos realizados de dichos inmuebles

4.1.2 Propietarios:

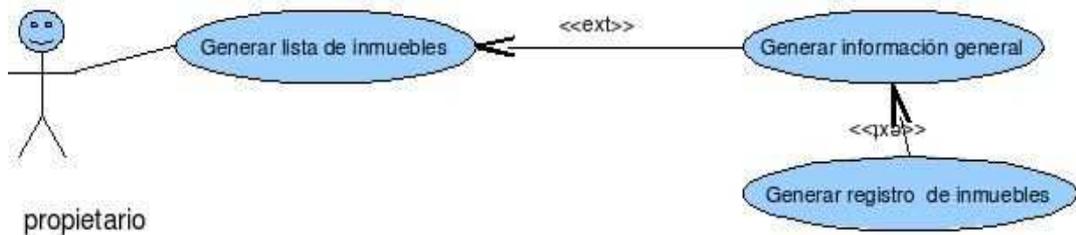


Diagrama 7. Propietario

Un propietario puede consultar todas las propiedades que se encuentren registradas en el sistema y de igual forma las que tiene registradas el mismo específicamente.

Un propietario tendrá la opción de ingresar un nuevo inmueble y editar los que ya tengan registrados anteriormente.

4.1.3 Arrendatario:

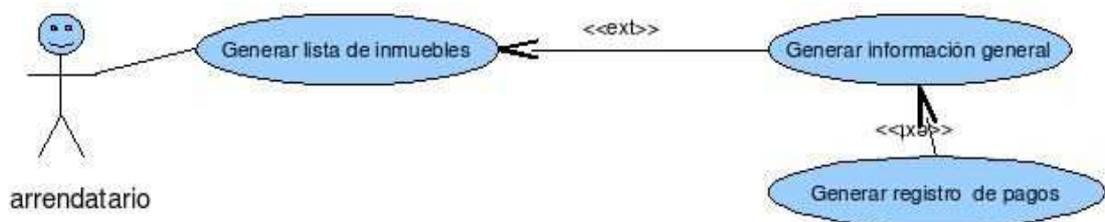


Diagrama 8. Arrendatario

Un cliente ingresa al sitio Web y tiene opción de realizar consultas sobre los inmuebles que en el momento están disponibles para la oferta junto con las fotografías de los mismos acompañados de una breve descripción del inmueble.

Si el cliente / arrendatario esta interesado en alguno de los inmuebles consultados tendría la posibilidad de acceder a la información completa y/o características del inmueble elegido.

A continuación el arrendatario solicitará una cita para ver el inmueble y corroborar las condiciones del mismo.

Después de realizado este proceso el arrendatario tomará la decisión de adquirir el inmueble a través de la inmobiliaria mediante un contrato establecido teniendo en la inmobiliaria es la intermediaria entre el arrendador y el arrendatario.

El contrato de arrendamiento traerá especificado todos los datos tanto del arrendatario, del arrendador y del inmueble: Fecha del contrato, ciudad, identificaciones, ubicación del inmueble y de la misma manera traerá cláusulas que se deben cumplir por parte de las personas involucradas.

4.1.4 Ventas:

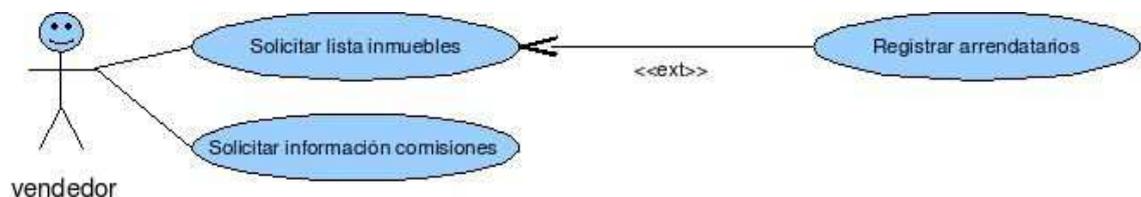


Diagrama 9.Vendedor

Un cliente propietario puede ofrecer su inmueble(s) a través de una inmobiliaria siempre y cuando acepte las condiciones del contrato que es impuesto por la inmobiliaria donde se indica todo lo referente a asuntos legales y el costo que tiene la promoción y venta del inmueble(s).

La inmobiliaria realizará previamente un estudio del inmueble(s) en el cual se definirá el avalúo de la propiedad y las condiciones de promoción y venta, de esta manera la inmobiliaria se reserva el derecho de aceptar o no la oferta del cliente propietario.

Si el propietario acepta dichas condiciones y el avalúo de su(s) inmueble(s), puede registrar su(s) propiedad(es) en Internet a través del sitio Web o en las oficinas a través de los asesores comerciales, de ser vendido el inmueble(s) la inmobiliaria obtendrá un porcentaje de la venta establecido en el contrato.

Compra de inmuebles:

Un cliente ingresa al sitio Web y tiene opción de realizar consultas sobre los inmuebles que en el momento están disponibles para la oferta junto con las fotografías de los mismos acompañados de una breve descripción del inmueble.

Si el cliente/comprador esta interesado en alguno de los inmuebles consultados tendría la posibilidad de acceder a la información completa y/o características del inmueble elegido.

A continuación el comprador solicitará una cita para ver el inmueble y corroborar las condiciones del mismo.

Después de realizado este proceso el comprador tomara la decisión de adquirir el inmueble a través de la inmobiliaria mediante un contrato establecido teniendo en cuenta que la inmobiliaria es la intermediaria entre el comprador y el vendedor.

4.1.5 Cartera:

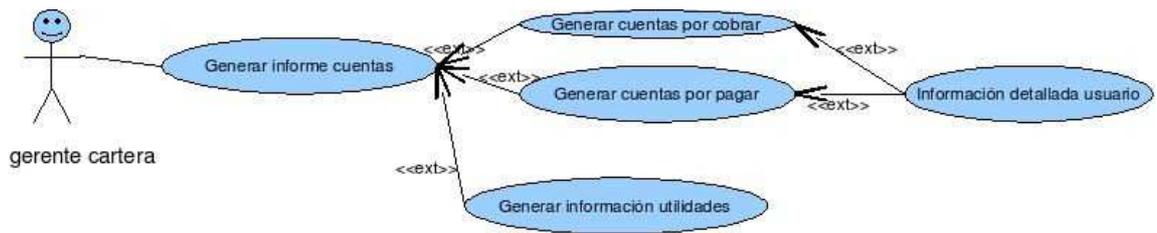


Diagrama 10. Gerente cartera

En este el gerente de cartera tendrá la opción de solicitar información acerca de las cuentas por cobrar, cuentas por pagar y las utilidades de todos los procesos correspondientes a los inmuebles, de la misma forma el gerente de cartera generar los registros de los pagos que realicen los arrendatarios

4.1.6 Jurídica:



Diagrama 11. Abogado

La parte jurídica del sistema de control inmobiliario es la encargada de realizar o generar la confirmación de los registros de los arrendatarios y de la misma manera el registro de los inmuebles ya que antes de que aparezcan en el sistema dichos inmuebles se deben someter a verificación y no tengan ningún inconveniente legal para generar la gestión en “SCI” ya sea compra venta entre otros.

4.1.7 Administrador:

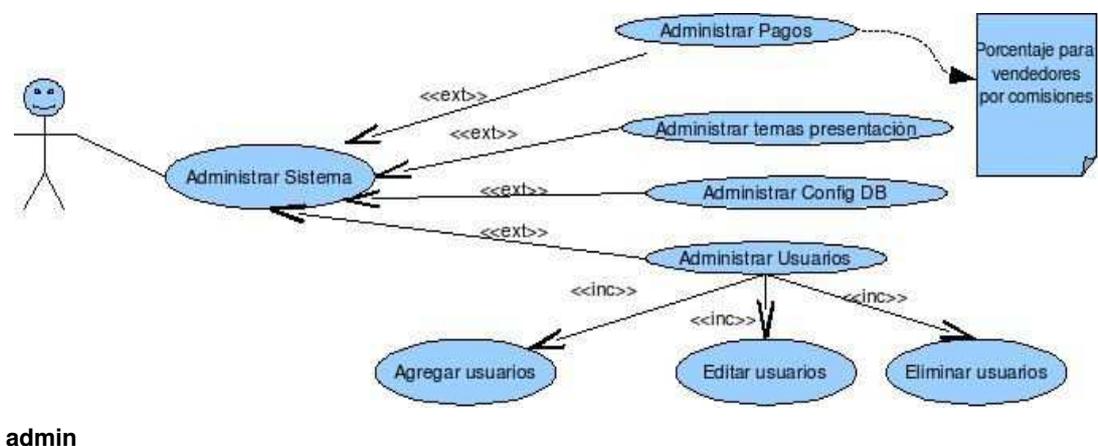


Diagrama 12. Administrador sistema

El cliente/propietario solicita el servicio de administración de uno o varios de sus inmuebles, entonces la inmobiliaria enviará varios peritos (personas especializadas en valoraciones de inmuebles) que permitan hacer un diagnóstico de lo requerido para la administración de cada inmueble.

Luego la inmobiliaria define la forma y los elementos que permitan administrar cada inmueble para así determinar el costo mensual de administración.

Definiendo el costo, la inmobiliaria y el cliente/propietario realizan un contrato que indica los deberes y responsabilidades de cada uno de ellos así como el costo del proceso de administrar cada inmueble.

Después de que el cliente/propietario acepta los términos del contrato, la inmobiliaria registra el o los inmuebles a administrar indicando todo lo requerido para hacerlo. Teniendo en cuenta el personal necesario y los insumos requeridos para cumplir con lo pactado en el contrato, la inmobiliaria fijará las fechas de los pagos mensuales correspondientes a la administración del predio

La inmobiliaria deberá hacer una revisión periódica a los inmuebles administrados a fin de determinar las necesidades que surjan a través del tiempo.

El cliente /propietario podrá dar sus comentarios, sugerencias o quejas a la inmobiliaria vía web o a través de la oficina de la misma.

Al cliente/propietario se le generará un recibo de pago en el que se indicará la información correspondiente al servicio de administración mensual.

El administrador del sistema podrá solicitar informes acerca de todos los procesos realizados en el sistema de control inmobiliario dichos procesos son: solicitar informes de ventas por cada corte, solicitar informes de cartera, solicitar la lista del personal de SCI.

Análisis de las características del inmueble:

Consiste en analizar las características propias de la edificación teniendo en cuenta además las propiedades ambientales y de calidad de su entorno.

En este paso, se determinará el mejor uso del inmueble, en función del planeamiento urbanístico y de la oferta y demanda del mercado, se contemplarán las posibilidades de reestructuración (por ejemplo, conversión de vivienda en local de negocio) o de rehabilitación, teniendo en cuenta las características propias de la edificación existente y la oferta de inmuebles semejantes.

El cliente/propietario solicitará el servicio de avalúo a la inmobiliaria, la cual envía uno de sus expertos a revisar el estado del inmueble, llenando un formulario que permite definir el estado de la vivienda así como su costo comercial.

Este procedimiento tiene un costo que es definido por la inmobiliaria, el cual debe ser pagado por el cliente/propietario para la obtención del resultado del avalúo.

5. DISEÑO

5.1 Diccionario de datos

Tabla 1. Contratos

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>id</u>	int(11)	No		
idPropietario	int(11)	No		
idArrendatario	int(11)	No		
idInmueble	int(11)	No		
idAsesor	int(11)	No		
fechaInicio	date	No		
fechaFinal	date	No		
diasPagoMes	int(11)	No		
aprobadojuridico	tinyint(1)	No		

Tabla 2. Inmuebles

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>id</u>	int(11)	No		
identificación	tinytext	No		
idTipo	int(11)	No		
dirección	tinytext	No		
ciudad	tinytext	No		
país	tinytext	No		
teléfono	tinytext	No		
idPropietario	int(11)	No		
idArrendatario	int(11)	No		
montoArriendo	int(11)	No		
disponible	tinytext	No		
aprobadoJuridico	tinytext	No		
reconstrucción	float	No		
frente	float	No		
fondo	float	No		
nPisos	int(11)	No		
nBanos	int(11)	No		
parqueadero	tinytext	No		

jardín	tinytext	No		
nCocinas	int(11)	No		
precio	int(11)	No		
descripción	text	No		
imagesPath	tinytext	No		
idZona	int(11)	No		

Tabla 3. Movimientos

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>id</u>	int(11)	No		
idInmueble	int(11)	No		
fecha	date	No		
monto	int(11)	No		
meses	int(11)	No		

Tabla 4. Permisos

Campo	Tipo	Nulo	Predeterminado	Comentarios
--------------	-------------	-------------	-----------------------	--------------------

<u>id</u>	int(11)	No		
idRol	int(11)	No		
sección	tinytext	No		

Tabla 5. Roles

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>id</u>	int(11)	No		
nombre	tinytext	No		
descripción	tinytext	No		

Tabla 6. TiposInmueble

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>id</u>	int(11)	No		
nombre	tinytext	No		

Tabla 7. Usuarios

Campo	Tipo	Nulo	Predeterminado	Comentarios
--------------	-------------	-------------	-----------------------	--------------------

<u>id</u>	int(11)	No		
idRol	int(11)	No		
nombre	tinytext	No		
apellido	tinytext	No		
nick	tinytext	No		
password	tinytext	No		
teléfono	tinytext	No		
dirección	tinytext	No		
ciudad	tinytext	No		
país	tinytext	No		
email	tinytext	No		

Tabla 8. Zonas

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>id</u>	int(11)	No		
nombre	tinytext	No		

5.1.2 MODELO ENTIDAD RELACION

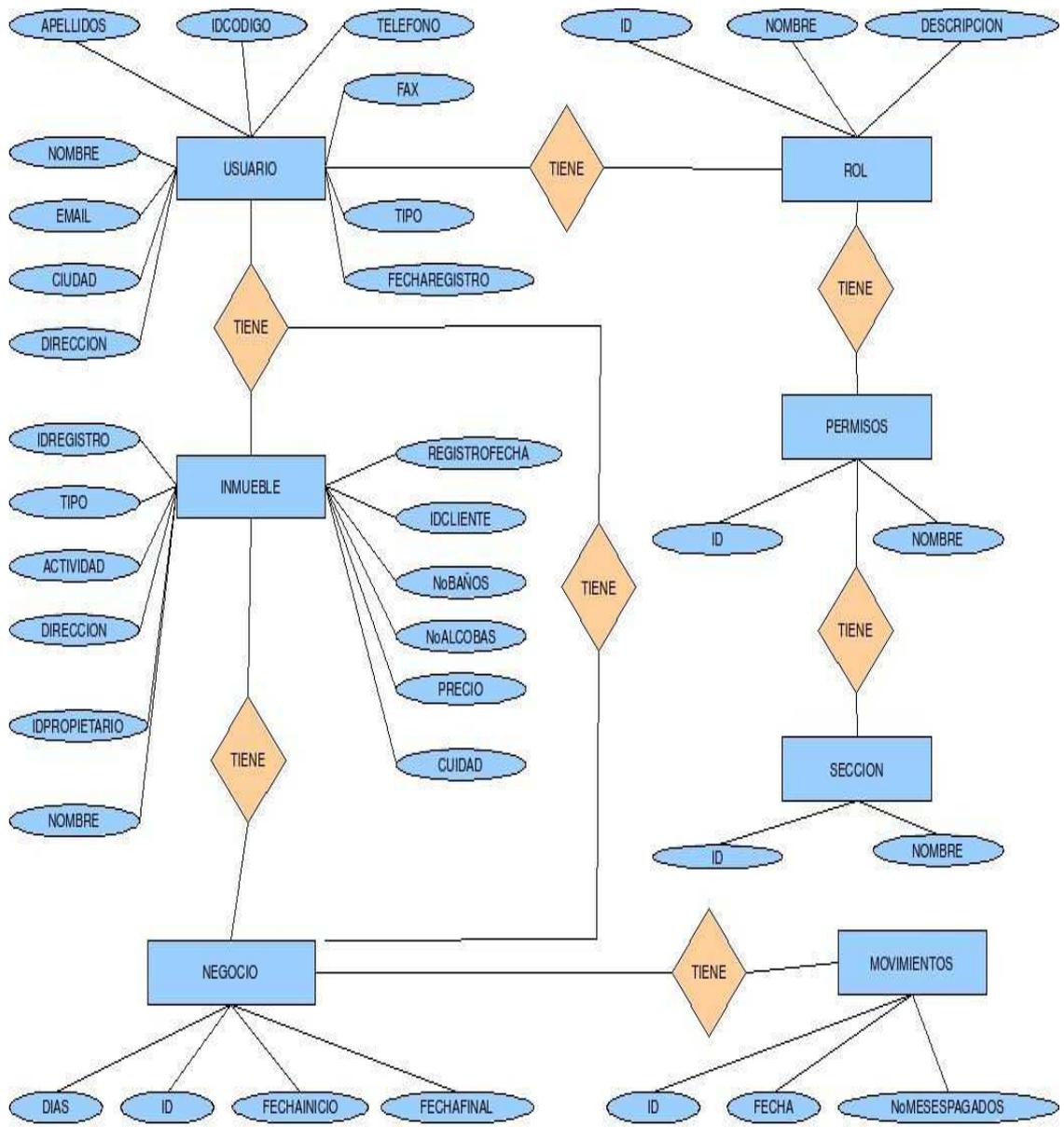
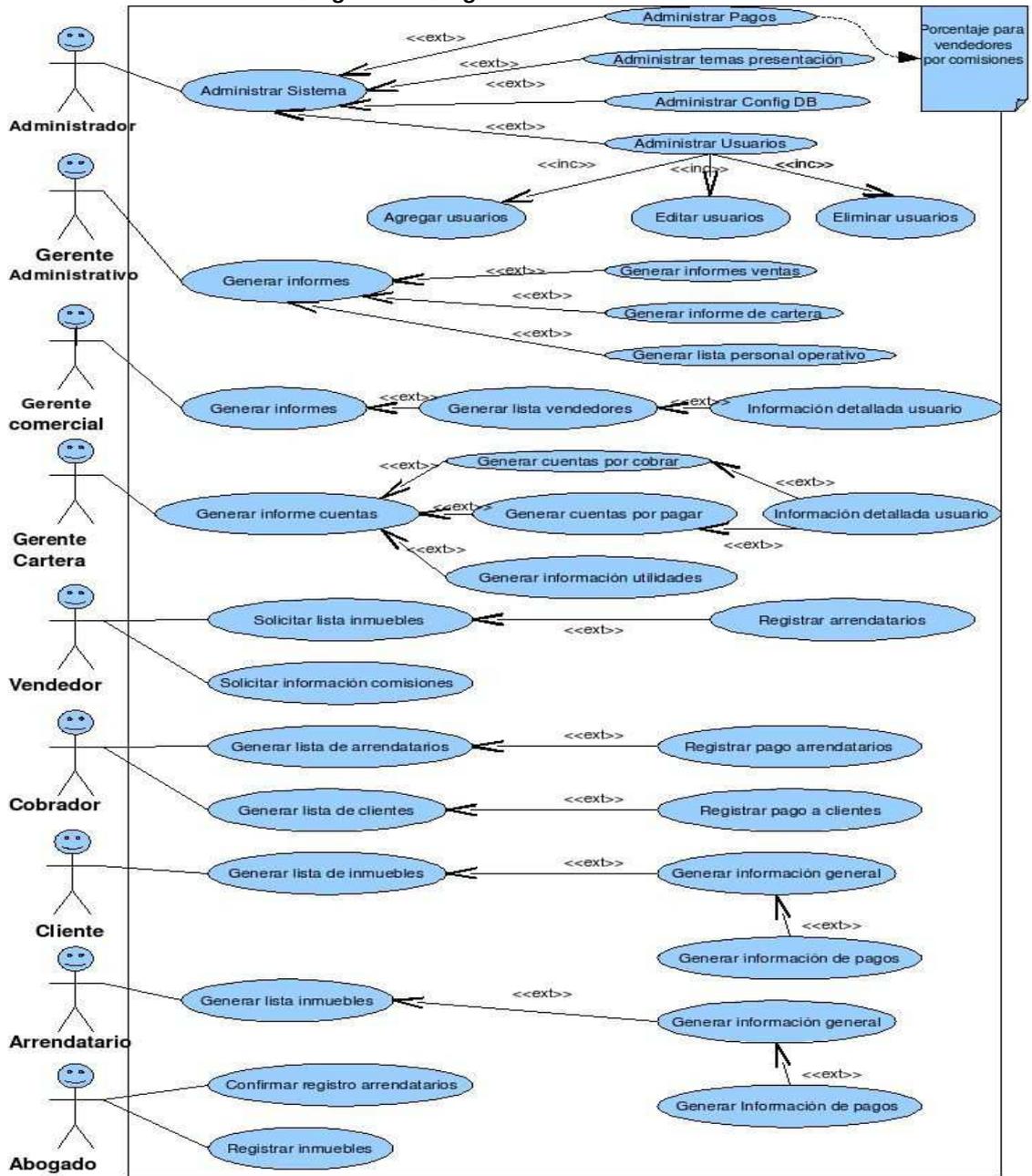


Diagrama 3. Diagrama Entidad relación

5.2 DIAGRAMAS UML

5.2.1 Diagrama de casos de Uso.

Diagrama 4. Diagrama de Casos de Uso.



5.2.2 Diagrama de Clases.

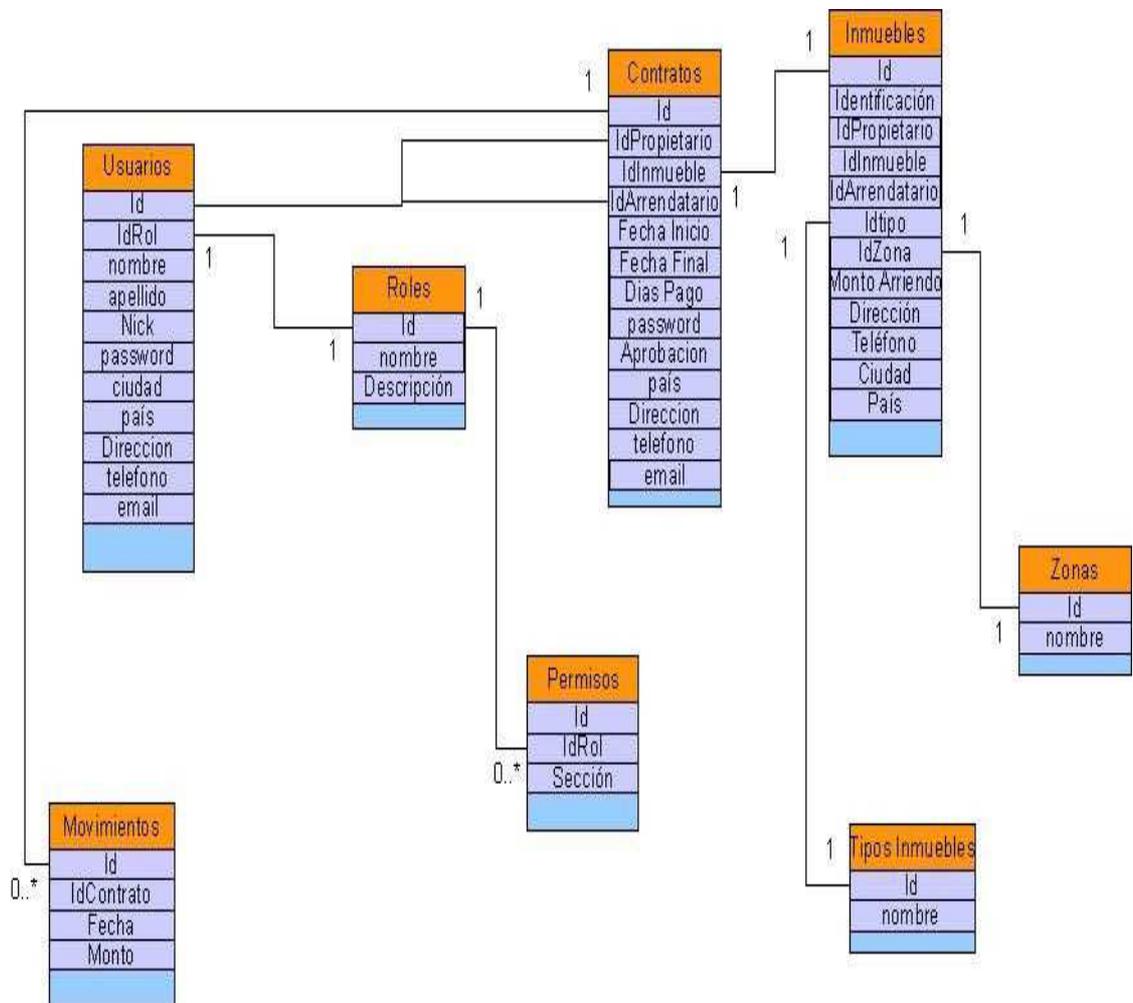


Diagrama 5 .Diagrama de Clases.

6. DESARROLLO

6.1 ESPECIFICACIONES TÉCNICAS

6.1.1 Software

El sistema de control inmobiliario es una aplicación multiplataforma y tiene la capacidad de interactuar con cualquier sistema operativo.

El software requerido para que el Sistema de control inmobiliario funcione correctamente se describirá a continuación tanto en el cliente como en el servidor ya que la aplicación fue desarrollada bajo el estilo de programación Mapache.

Se requiere utilizar un navegador libre como Mozilla Firefox 2.0 o versiones superiores ya que los navegadores como Internet Explorer (navegador privativo) no tiene compatibilidad con el estilo MApache y no cumplen todos los estándares Web.

Es necesario que el sistema operativo en el cual se instale el sistema de control inmobiliario cuente con las versiones actuales o superiores de las herramientas requeridas como lo son:

Apache, Mysql y PHP.

El paquete de clases de MApache debe quedar en el mismo directorio donde quede instalado el sistema de control inmobiliario.

Cabe recordar que MApache es un estilo de programación el cual permite o genera una mejor organización y estructuración del código fuente y evita tratar con vario lenguajes de programación en un mismo script.

6.1.2 HARDWARE

Debido a que el sistema de control inmobiliario podrá ser instalado en cualquier sistema operativo no es necesario realizar un estudio del hardware y puede utilizarse en cualquier computador existente sin ser robusto o especializado mas sin embargo se recomiendan las siguientes características para un mejor trabajo de la aplicación:

Procesador: Amd Duron

Memoria Ram: 256Mgb o superior

Disco Duro: 40 Gbs o superior.

6.2 ESTRUCTURA DEL PROGRAMA

Para realizar la aplicación SCI (Sistema de control inmobiliario) se utiliza programación orientada a objetos para que sea posible trabajar de manera fácil futuras versiones de la aplicación, de la misma forma se PHP, lenguaje de programación orientado a la Web

6.3 ESTRUCTURA DEL CÓDIGO FUENTE

El código fuente como se había mencionado anteriormente esta basado en el estilo de programación MApache y en el se encontrara únicamente código PHP sin estar este embebido en etiquetas HTML y de esta manera esta estructura facilita la lectura del código y las respectivas correcciones del mismo.

7. PRUEBAS

7.1 PRUEBAS DE ANÁLISIS Y DISEÑO

- Revisión de modelos: Un tipo de inspección, que puede ser desde un revisión técnica formal hasta un recorrido informal , realizado por personas diferentes a las que estuvieron directamente involucradas en el desarrollo del modelo.
- Revisión de prototipos: Es un proceso mediante el cual los usuarios trabajan a través de una colección de casos de uso, utilizando un prototipo como si fuera el sistema real. El objetivo principal es probar si el diseño del prototipo satisface las necesidades de esos usuarios.
- Demostrar con el código: La mejor forma de determinar si un modelo realmente refleja lo que se necesita, o lo que se debe construir, es construyendo software basado en el modelo para mostrar que el modelo esta bien
- Pruebas de escenarios de uso: Una técnica de prueba en la cual una o mas personas validan un modelo siguiendo la lógica de los escenarios de uso.

7.2 PRUEBAS DE CÓDIGO

- Prueba de Caja-Negra: La prueba verifica que el ítem que se esta probando, cuando se dan las entradas apropiadas produce los resultados esperados.
- Prueba de Valores-Frontera: Es la prueba de situaciones extremas o inusuales que el ítem debe ser capaz de manejar.
- Prueba de Integración de Clases: Es el acto de asegurar que las clases, y sus instancias, conforman un software que cumple con el comportamiento definido.
- Revisión de Código: Una forma de revisión técnica en la que a partir del ejecutable se revisa en el código fuente.

- Prueba de Cubrimiento: Es el acto de asegurar que toda línea de código es ejercita al menos una vez.
- Prueba de Regresión de Herencia: Es el acto de ejecutar casos de prueba de las súper clases, tanto de forma directa como indirecta, en una subclase específica.
- Prueba de Método: Consiste en realizar pruebas para verificar que un método (función miembro) funciona tal como esta definido.
- Prueba de Caminos: Es el acto de asegurar que todos los caminos lógicos en el código se ejercitan al menos una vez. Prueba de Caja-Blanca: Consiste en realizar pruebas para verificar que líneas específicas de código funcionan tal como esta definido. También se le conoce como prueba de caja-transparente.

7.3 PRUEBAS DE SISTEMA

- Prueba de instalación: Consiste en verificar que el sistema funcione adecuadamente cuando se instale en cualquier ordenador.
- Prueba de Stress: El acto de asegurar que el sistema funciona como se espera bajo grandes volúmenes de transacciones, usuarios, carga y demás.

8. CRONOGRAMA

CRONOGRAMA		
TAREAS	ACTIVIDAD	FECHA
PLANEACION	Identificación de necesidades	02/01/07
	Viabilidad técnica y económica	08/01/07
	Sitio Web	11/01/07
ANALISIS	Estado del arte	18/01/07
	Recopilación de requerimientos	22/01/07
	Documentación	04/02/07
	Análisis Base de Datos	20/02/07
	Análisis Módulos	05/03/07
DISEÑO	Diseño Modelo Entidad Relación	27/03/07
	Diseño Diagramas UML	04/04/07
	Adaptación lenguaje de programación	19/04/07
	Diseño de interfaces	26/04/07
CODIFICACION	Construcción base de datos	04/05/07
	Construcción modulo administración	16/05/07
	Construcción modulo propietarios	19/06/07
	Construcción modulo ventas	21/07/07
	Construcción modulo jurídica	23/08/07
	Construcción modulo arrendatarios	21/09/07
	Construcción modulo cartera	12/10/07
	Construcción modulo invitado	15/10/07
PRUEBAS	Pruebas	31/10/07
	Manuales	02/11/07
	Implementación	13/11/07

Tabla 9. Cronograma

9. GLOSARIO

Apache:

Es un software servidor de HTTP de código abierto para plataformas UNIX (BSD, GNU/Linux, etc.). Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA.

Archivo:

Datos estructurados que pueden recuperarse fácilmente y usarse en una aplicación determinada. Se utiliza como sinónimo de fichero. El archivo no contiene elementos de la aplicación que lo crea, sólo los datos o información con los que trabaja el usuario.

Conjunto de bytes almacenados como una entidad individual. Todos los datos en disco se almacenan como un archivo con un nombre de archivo asignado que es único dentro del directorio que se reside.

Para el computador, un archivo no es más que una serie de bytes. La estructura de archivo es conocida para el software que lo maneja. Por ejemplo, los archivos de bases de datos están compuestos por una serie de registros. Los archivos de procesamiento de texto, también llamados documentos, contienen un flujo continuo de texto.

Bases de datos:

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Bases de datos relacionales:

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.

Estructuras de datos:

En programación una **estructura de datos** es una forma de organizar un conjunto de datos elementales (un dato elemental es la mínima información que se tiene en el sistema) con el objetivo de facilitar la manipulación de estos datos como un todo o individualmente.

Una **estructura de datos** define la organización e interrelacionamiento de estos, y un conjunto de operaciones que se pueden realizar sobre él.

GNU/LINUX:

Sistema operativo con una amplia difusión, una plataforma para el desarrollo de aplicaciones aceptada mundialmente por usuarios, instituciones y gobiernos para la cual hay disponibles multitud de lenguajes, bibliotecas y herramientas que, en su mayor parte, siguen la filosofía de código de uso libre.

Hardware:

Componente físico tecnológico, que trabaja o interactúa de algún modo con la computadora. No sólo incluye elementos internos como el disco duro, CD-ROM, disquetera, sino que también hace referencia al cableado, circuitos, gabinete, etc. E incluso hace referencia a elementos externos como la impresora, el mouse, el teclado, el monitor y demás

Hosting:

(Alojamiento Web, webhosting, alojamiento de una página Web). Servicio que ofrecen algunas compañías (los webhost) en Internet que consiste en ceder un espacio en sus servidores para subir (alojar, hostear) un sitio Web para que pueda ser accedido en todo momento de forma online.

Html:

Acrónimo inglés de **HyperText Markup Language** (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

Internet:

Es una combinación de hardware (ordenadores interconectados por vía telefónica o digital) y software (protocolos y lenguajes que hacen que todo funcione). Es una infraestructura de redes a escala mundial (grandes redes principales (tales como **MILNET**, **NSFNET**, y **CREN**), y redes más pequeñas que conectan con ellas) que conecta a la vez a todos los tipos de ordenadores

Java Script:

Lenguaje interpretado orientado a las páginas Web, con una sintaxis semejante a la del lenguaje Java. fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores de Internet comerciales. Se utiliza en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente.

Localhost:

Es un nombre reservado que tiene todo ordenador, router o dispositivo que disponga de una tarjeta de red ethernet para referirse a sí mismo. El nombre localhost es traducido como la dirección IP de loopback 127.0.0.1 en IPv4, o como: **1** en IPv6.

Login:

Es una clave que se le asigna al usuario con el fin de que pueda utilizar todos los recursos de una computadora, identifica de igual manera al usuario dentro de Internet junto con la dirección electrónica de la computadora a utilizar.

Mysql:

Sistema de administración para bases de datos relacionales (rdbms) que provee una solución robusta a los usuarios con poderosas herramientas multi-usuario, soluciones de base de datos SQL (structured Query Language) multi-threaded. Es rápido, robusto y fácil de utilizar.

Mozilla Firefox:

Es un navegador de internet, con interfaz gráfica de usuario desarrollado por la Corporación Mozilla y un gran número de voluntarios externos 1 Firefox, oficialmente abreviado como **Fx o fx 2** y comúnmente como **FF**, comenzó como un derivado del Mozilla Application Suite, que terminó por reemplazarlo como el producto bandera del proyecto Mozilla, bajo la dirección de la Fundación Mozilla.

OpenSource:

Es el término con el que se conoce al software distribuido y desarrollado libremente. Fue a utilizado por primera vez en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (*free software*).

Oracle:

Una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer, o JDeveloper. Su CEO actual es Larry Ellison.

Programación orientada a objetos:

Es un paradigma de programación que define los programas en términos de “clases de objetos”, objetos que son entidades que combinan *estado* (es decir, datos), *comportamiento* (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar

tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Password:

Es un código utilizado para acceder o ingresar a un espacio o sistema restringido. El password puede tener caracteres alfanuméricos y símbolos y nunca va a ser visible en el momento de digitar con el fin que el único en conocerla sea el usuario.

Postgres:

Es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.

Programación PHP:

Es un lenguaje de script (o de guiones), diseñado para, entre otras cosas, aumentar, incrementar el dinamismo de las páginas Web. Originalmente se trataba de un conjunto de macros concebidas para ayudar en el mantenimiento de páginas Web. Desde entonces, sus características han ido creciendo hasta convertirse en un lenguaje de programación completo, capaz de manejar entornos que integran grandes bases de datos.

Python:

Es un lenguaje de programación creado por Guido van Rossum en el año 1990. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. La última versión estable del lenguaje es actualmente (septiembre de 2006) la 2.5.

Python es considerado como la "oposición leal" a Perl, lenguaje con el cual mantiene una rivalidad amistosa. Los usuarios de Python consideran a éste mucho más limpio y elegante para programar.

Software:

Se denomina software, programática, equipamiento lógico o soporte lógico a todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Stand Alone:

Es un término libremente definido usado para clasificar programas de computadora. Intenta dibujar una cierta distinción entre los programas invocados por un cierto acontecimiento de la computadora y esos invocada por otros programas. No obstante esta distinción no está parada hasta escrutinio fuera del contexto de ciertos sistemas encajados, puesto que la computadora tiene generalmente funcionar un cierto programa con el cual prepare (y a veces inicialmente los procesos) el programa "independiente" para comenzar. Por lo tanto todos los programas son lanzados o preparados/procesados por otros programas (con la excepción única del cargador de elástico de bota), y ningún código no se puede realmente decir a independiente.

UNIX:

Es un sistema operativo multitarea y multiusuario, lo cual significa que puede ejecutar varios programas simultáneamente, y que puede gestionar a varios usuarios simultáneamente. Se desarrolló en los laboratorios Bell (por Kernighan & Thompson), y aunque al principio se diseñó para el PDP-11, una máquina de Digital, ahora se ejecuta en gran cantidad de plataformas con muchos tipos de microprocesadores diferentes, haciéndolo un sistema multiplataforma, y provocando por tanto que un programa en código máquina ejecutable en una plataforma en UNIX no tenga por qué ser ejecutable en otra.

Web:

Es una fuente de información adaptada para la World Wide Web y accesible mediante un navegador de Internet. Ésta información se presenta generalmente en formato HTML y puede contener hiperenlaces a otras páginas Web, constituyendo la *red* enlazada de la World Wide Web.

XML:

Sigla en inglés de **eXtensible Markup Language** (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

10. CONCLUSIONES

- Se genero la creación de una aplicación la cual brinda a todos sus usuarios una interacción con la misma de manera ágil, segura y de alto nivel para realizar cualquier tipo de procedimiento del sector inmobiliario.
- Se facilita el manejo de la información proporcionando una organización en la documentación, generando una reducción en los costos de almacenamiento y aumentado la privacidad de los datos de la inmobiliaria.
- Se garantiza la opción de que en posteriores versiones se puedan añadir nuevas y mejores funcionalidades ya que es una aplicación que cumple con los modelos para realizar dicho proceso.
- La implementación de nuevas herramientas facilitan y garantizan un orden en la elaboración de la aplicación para futuras modificaciones al igual que la gestión en los procesos que se lleven a cabo en el sector inmobiliario tales como arrendamientos y control de pagos.

11. SUGERENCIAS Y RECOMENDACIONES

- Se recomienda utilizar Mozilla Firefox , el sistema de control inmobiliario tiene la capacidad de soportar cualquier navegador libre sin que surja ningún tipo de inconveniente. para poder realizar las descargas de las ultimas versiones de dichos navegadores se puede ingresar por el siguiente enlace:

<http://www.mozilla.org/products/firefox>

- Se recomienda utilizar el sistema operativo GNU/Linux ya que este es un sistema operativo seguro y es distribuido libremente lo que significa que puede ser usado, modificado y redistribuido.
- A pesar de que se recomienda utilizar el sistema operativo GNU/Linux es probable que la aplicación pueda ser cargada e otros servidores pero para estos se recomienda el uso de Apache Server en su instalación.

12. BIBLIOGRAFÍA

LERMA, Héctor Daniel. Metodología de la Investigación. Propuesta, Anteproyecto y Proyecto. Colombina: 1999. 135 p.

PRESSMAN, Roger S. Ingeniería del Software. Un enfoque práctico. 5A Edición. Madrid: McGraw-Hill, 2002. 589p.

ANAYA, Luke Welling, Laura Thompson. Desarrollo Web con PHP y MySQL. Madrid: Artes Gráficas Guemo, S.L., 2003. 912 p.

Wikipedía enciclopedia libre (online)

<http://es.wikipedia.org/wiki/Wikipedia>

www.angelfire.com/tomaustin/Met/metinacap.htm

academia.uniminuto.edu

www.infocomercial.com

13. MANUALES

13.1 MANUAL DEL SISTEMA

El sistema de control inmobiliario puede ser instalada de las siguientes maneras:

1. Instalación en un servidor en internet .
2. Instalación en un servidor a modo localhost.
3. Instalación en un servidor portable.

El software externo requerido para generar un buen funcionamiento de la aplicación debe ser como primera medida el el servidor Web denominado Apache el cual nos permite acceder a dicha aplicación por medio de un navegador, MySQL donde se permite realizar la conexión con la base de datos utilizada en el sistema, Mapache que es el estilo de programación utilizado por el sistema de control inmobiliario, un navegador libre el cual soporte todos los estándares establecidos por la W3C (World Wide Web (consorcio internacional)).

Se debe tener en cuenta que en el directorio donde se instale el sistema de control inmobiliario se debe ubicar el paquete de clases de MApache como por ejemplo en WWW o htdocs

13.2 MANUAL DEL USUARIO

El sistema de control inmobiliario es una aplicación la cual tiene como objetivo sistematizar cada uno de los procedimientos que se llevan a cabo en la inmobiliaria tales como la administración y el control de los inmuebles y servicios de dicha inmobiliaria.

SCI (sistema de control inmobiliario) comprenderá las siguientes áreas:

- Gestión de Usuario
- Gestión de Inmuebles

De esta manera todos los procesos realizados en el sistema se llevaran a cabo de una manera segura, eficaz y ágil.

la aplicación de sistema de control inmobiliario siendo orientada a la Web se deberá ingresar a ella mediante un navegador el cual debe cumplir con todos los estándares de la W3C como por ejemplo Mozilla Firefox, Konqueror entre otros.

Para poder acceder a la aplicación se debe dirigir a la siguiente dirección:

<http://localhost/projects/sci2>

A continuación SCI se activara y como primera medida aparecerá una ventana de acceso en la cual se solicita su nick y password

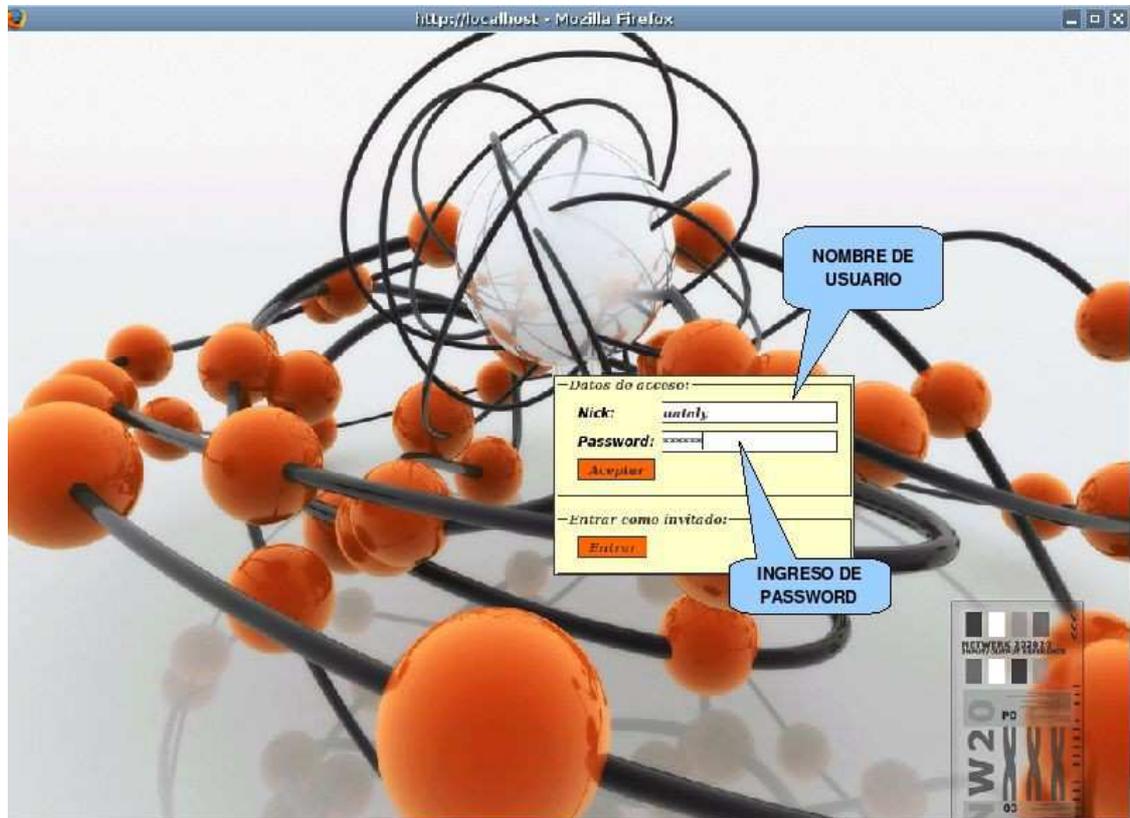


Figura No1. Acceso A SCI

A continuación aparecerán en pantalla todos los módulos de los cuales se distribuye el sistema de control inmobiliario pero esto depende del tipo de usuario o el rol que anteriormente se halla asignado, en este caso se ingresa al sistema el administrador quien tendrá manipulación total del sistema.

Los módulos mencionados anteriormente son los siguientes:

Invitado: Este modulo tiene como fin generar la opción a un usuario que ingrese al sistema poder verificar los inmuebles que en ese momento estén disponibles y de la misma forma poder realizar una búsqueda específica según sus requerimientos. se podrá también confirmar información adicional haciendo click en el botón



En este mismo modulo el usuario que ingrese como invitado tendrá la opción de registrarse para mas adelante poder realizar distintos procedimientos en la aplicación.

http://localhost - Mozilla Firefox

Invitado Propietarios Arrendatarios Ventas Cartera Jurídica Admin

Props.

-Buscar Inmueble:

Digite identificación: por:

Elija Tipo: Elija Zona:

Elija rango de precios: Elija rango de montos de arriendo:

Id	Identificación	Precio	Monto_arriendo_mensual	Zona	Tipo	
2	12a	\$200000000	\$200000	2	1	
11	12b	\$200000000	\$200000	2	1	
5	12c	\$200000000	\$200000	2	1	

cerrar SCI Ayuda Usuario

Terminado

Figura No2. Modulo Invitado SCI

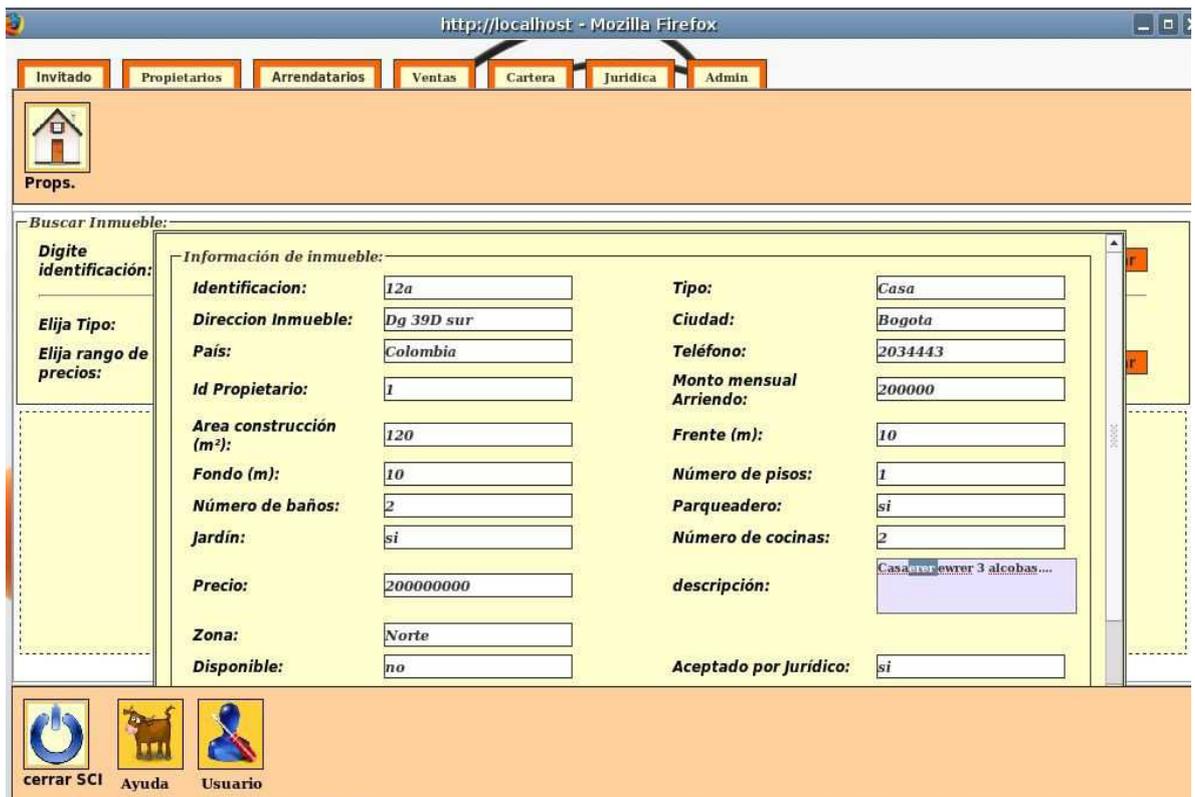


Figura No 3. Modulo Invitado SCI, Botón información adicional

Propietarios: Además de tener las mismas opciones del invitado como por ejemplo verificar los inmuebles que se encuentran disponibles en este modulo de propietario el mismo puede confirmar todos sus inmuebles registrados en el sistema y verificar toda la información de dichos inmuebles, podrá mediante el botón mis propiedades confirmar sus propiedades registradas  con sus respectivas opciones o botones para ver información adicional,  eliminar el inmueble y editarlo



http://localhost - Mozilla Firefox

Invitación Propietarios Arrendatarios Ventas Cartera Jurídica Admin

Props. MyProps. Register

Buscar Inmueble:

Digite Identificación: por: identificación

Elija Tipo: Casa Elija Zona: Norte

Elija rango de precios: Cualquiera Elija rango de montos de arriendo: Cualquiera

Id	Identificación	Precio	Monto arriendo mensual	Zona	Tipo	
2	12a	\$200000000	\$200000	2	1	  
11	12b	\$200000000	\$200000	2	1	  

cerrar SCI Ayuda Usuario

BOTONES PARA INFO ADICIONAL, ELIMINAR Y EDITAR

Figura No 4. Modulo Propietarios SCI, Botón Mis Propiedades

En este modulo el propietario también  podrá ingresar un nuevo inmueble de cualquier tipo mediante el botón registro  y verificar información adicional del mismo con el Botón correspondiente 

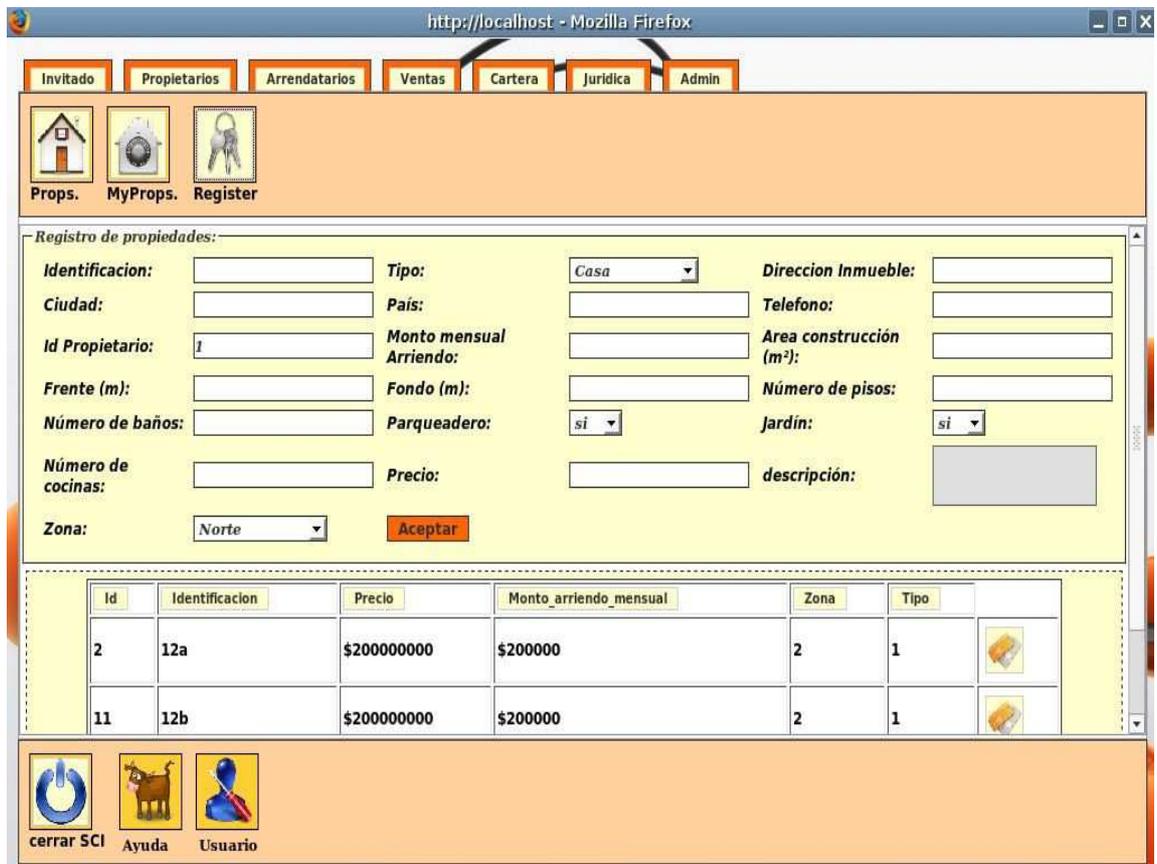


Figura No 5. Modulo Propietarios SCI, Botón registro

Arrendatarios: En el modulo de arrendatarios aparece un botón llamado contratos en el cual podremos verificar por el numero de contratos todos los inmuebles que fueron aprobados o no por el departamento jurídico y de la misma manera podemos ver la información detallada de dicho inmueble como por ejemplo la fecha final e inicial del contrato, los días que se debe realizar el pago, el asesor que genero el contrato entre otras. en conclusión podemos verificar los inmuebles “candidatos” para ser publicados en el sistema para su futura consulta.

Se presenta también la opción de ver mas información , eliminar y editar el inmueble haciendo click en estos botones respectivamente

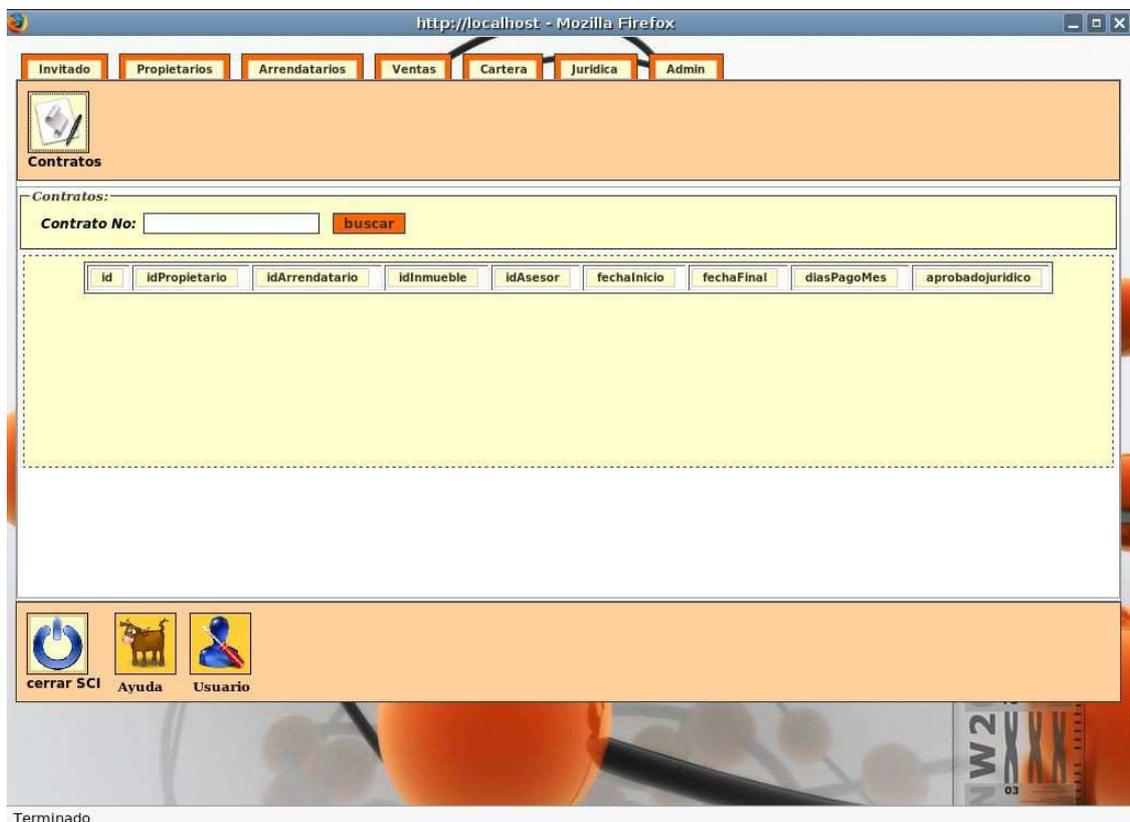


Figura No 6. Modulo Arrendatarios SCI, botón contratos

Ventas: En este modulo podremos verificar todas las propiedades o inmuebles que se encuentran registradas en el sistema cada una con su respectiva información detallada, en el botón propiedades.

En este mismo modulo cada uno de los asesores podrá confirmar el numero de contratos que le fueron aceptados en un mes determinado con su información respectiva y si fue aprobado por el departamento jurídico sera publicado en sistema para la gestión .

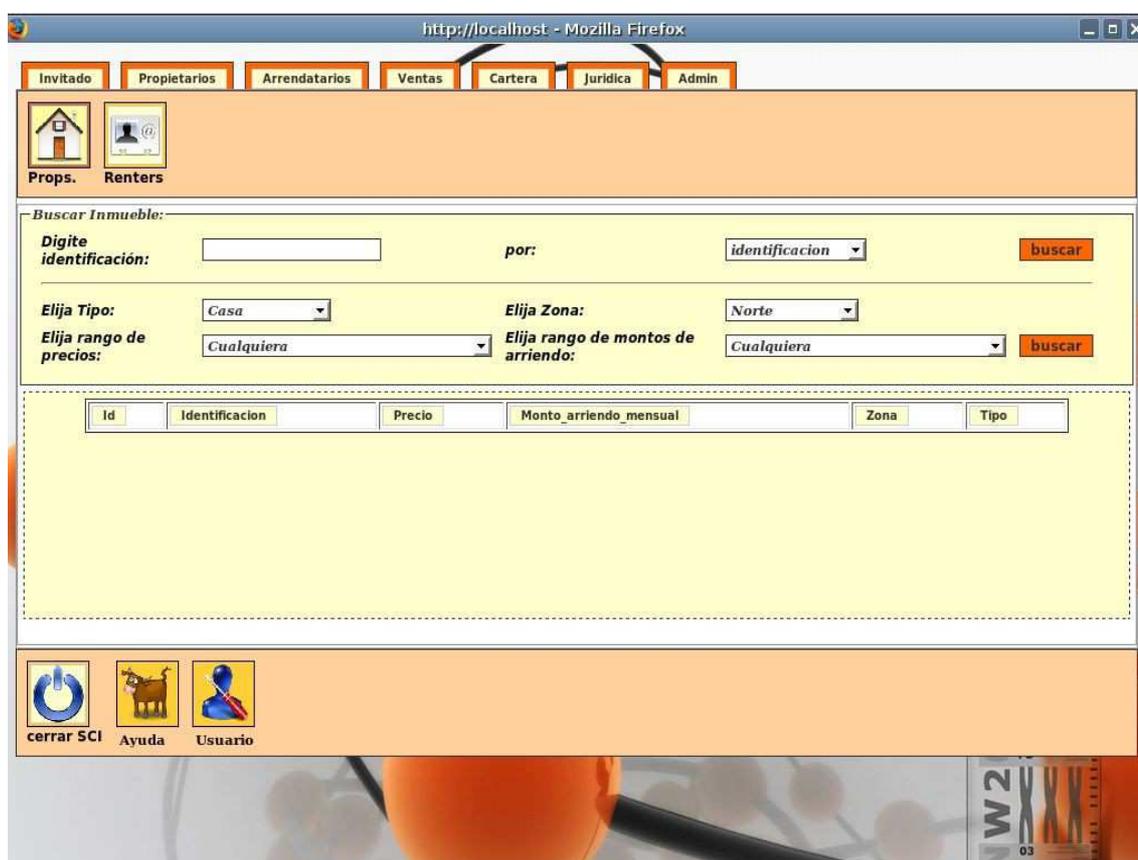


Figura No 7. Modulo Ventas SCI, botón propiedades



Figura No 8. Modulo Ventas SCI, botón asesores

Cartera: En el modulo de cartera se puede verificar con el numero de contrato la información de los mismos, en este modulo de podrán actualizar, modificar y eliminar.

Jurídica: En este modulo también tendrá la opción de confirmar todos los inmuebles que se encuentran registrados en el sistema los cuales se puede ver mas información acerca del mismo, eliminarlo y editarlo haciendo click en los respectivos botones.



http://localhost - Mozilla Firefox

Invitado Propietarios Arrendatarios Ventas Cartera **Jurídica** Admin

Props. Contratos

Buscar Inmueble:

Digite identificación: por: identificación

Elija Tipo: Casa Elija Zona: Norte

Elija rango de precios: Cualquiera Elija rango de montos de arriendo: Cualquiera

Id	Identificación	Precio	Monto_arriendo_mensual	Zona	Tipo	
2	12a	\$200000000	\$200000	2	1	  
11	12b	\$200000000	\$200000	2	1	  
5	12c	\$200000000	\$200000	2	1	  

cerrar SCI Ayuda Usuario

Figura No 9. Modulo Jurídica SCI, botón propiedades

Puede también en el botón de contratos puede confirmar todos los contratos que se encuentran registrados en la aplicación



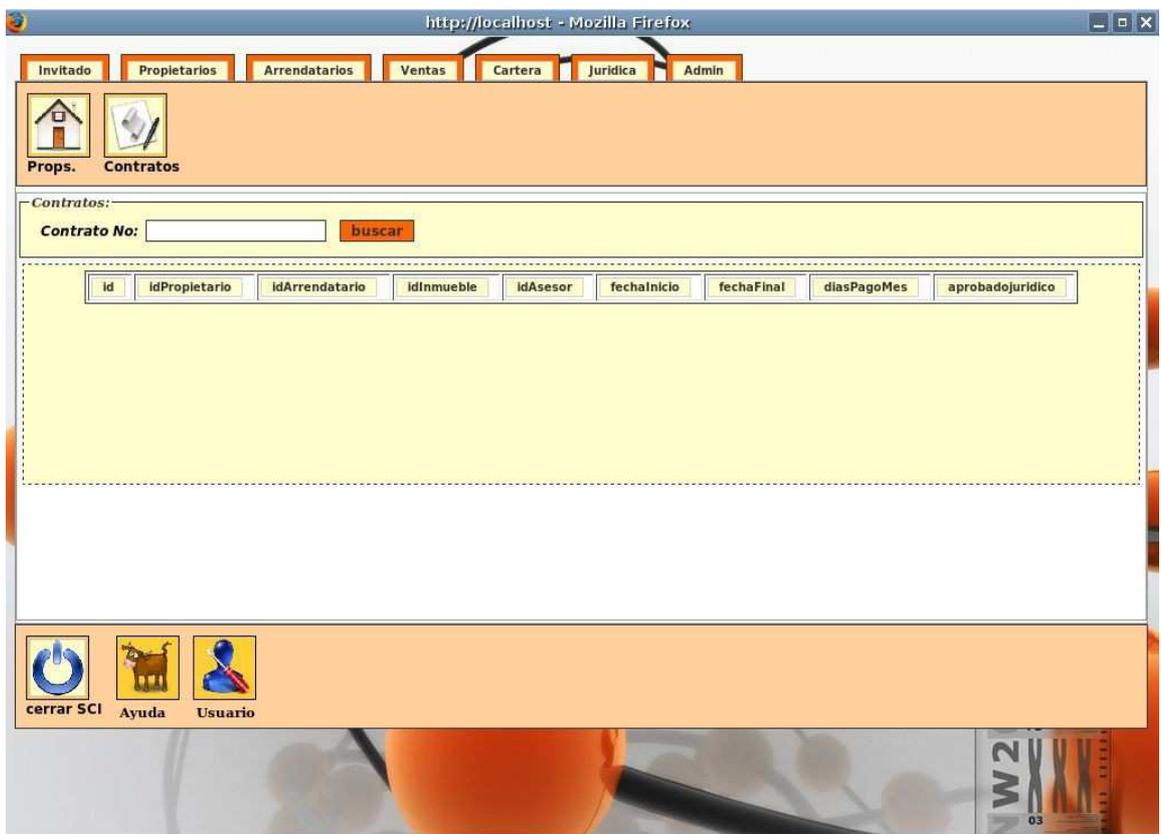


Figura No 10. Modulo Jurídica SCI, botón contratos

Admin : Este modulo tiene el control de todas las secciones del sistema de control inmobiliario como veremos a continuación en este modulo se encuentra como primera medida el botón de temas mediante esta opción se podrá cambiar el aspecto de la aplicación como sus colores, estilo, distribución entre otras.



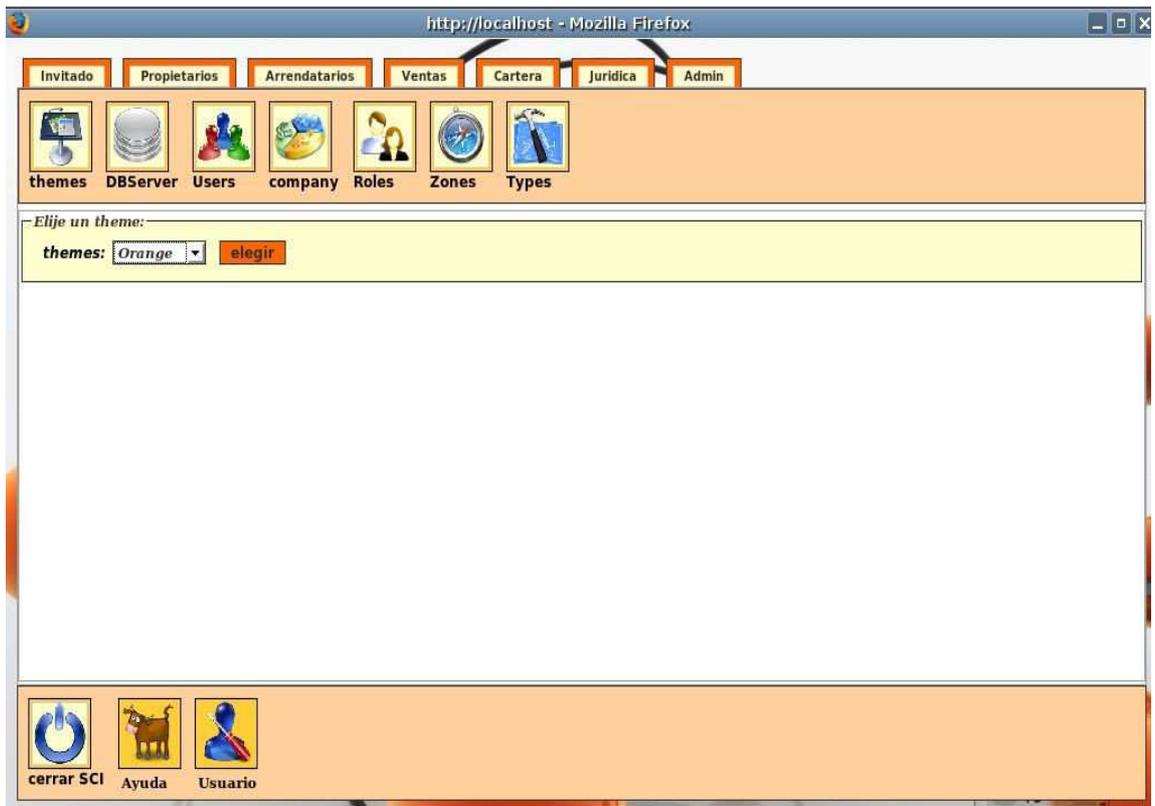


Figura No 11. Modulo Admin SCI, botón temas

Con el botón DBServer  tendrá la opción de configurar la ubicación de la base de datos que utiliza la aplicación

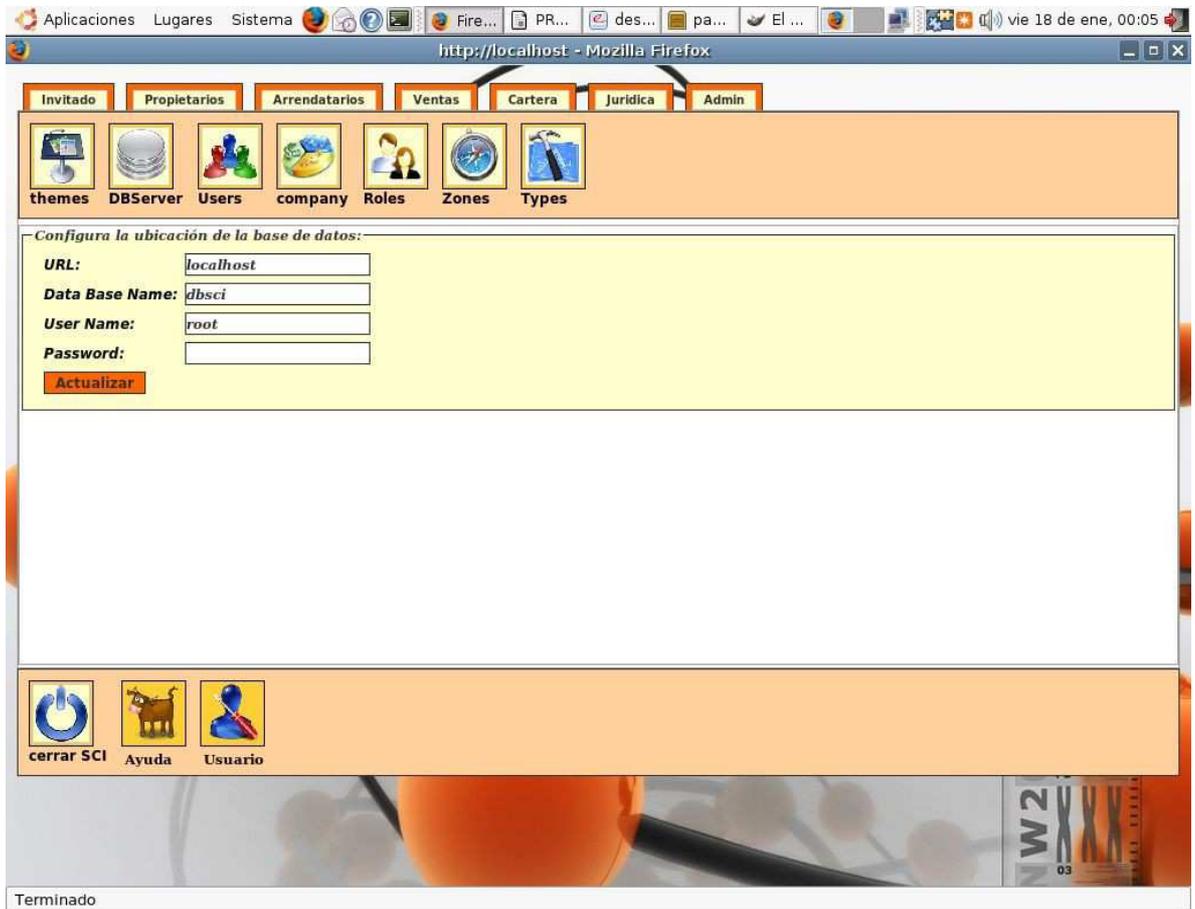


Figura No 12. Modulo Admin SCI, botón Servidor base de datos

Mediante el botón usuarios  tendrá la opción de registrar nuevos usuarios y de igual manera generar una  búsqueda de los mismos

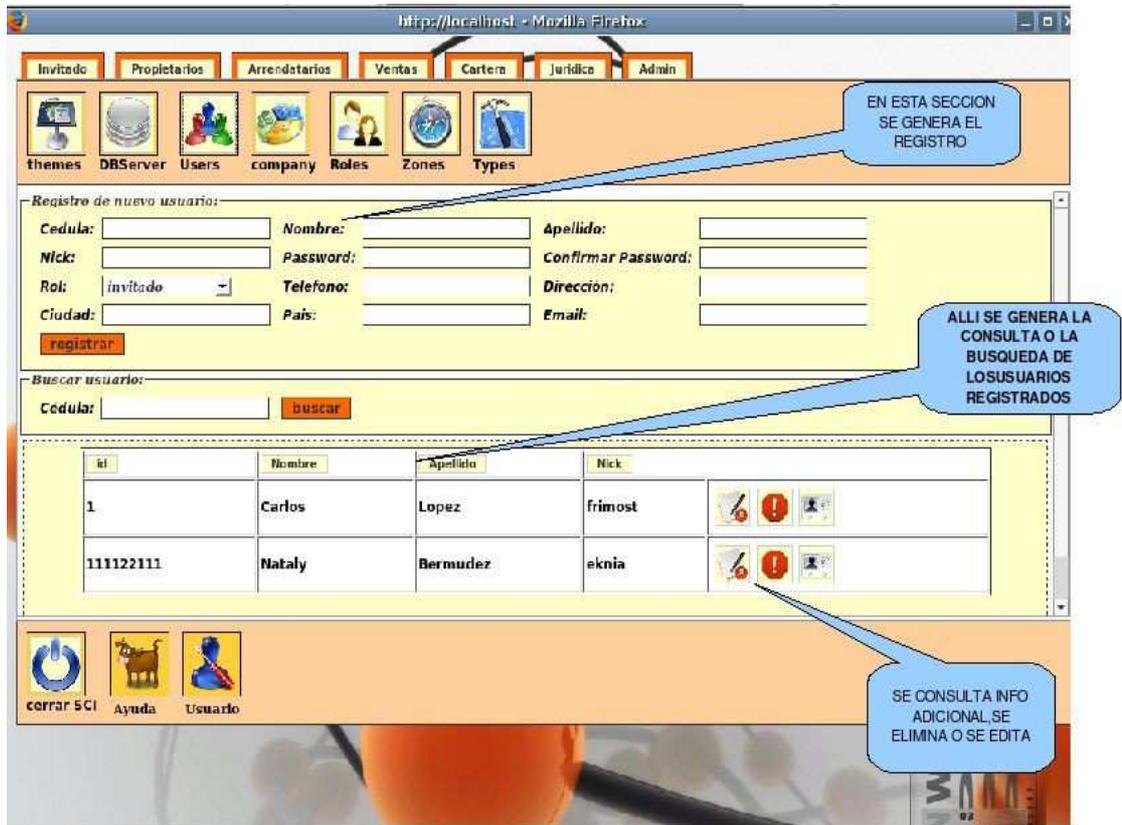


Figura No 13. Modulo Admin SCI, botón usuarios

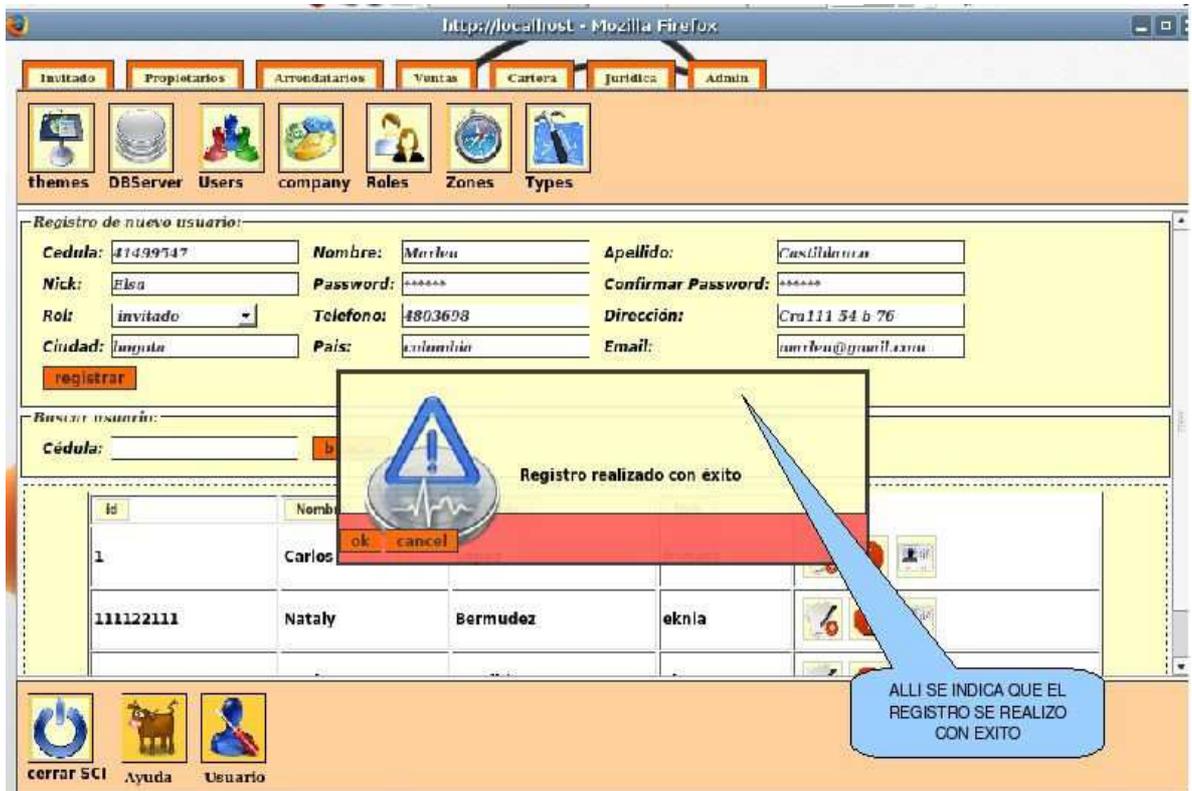


Figura No 14. Modulo Admin SCI, éxito en el registro

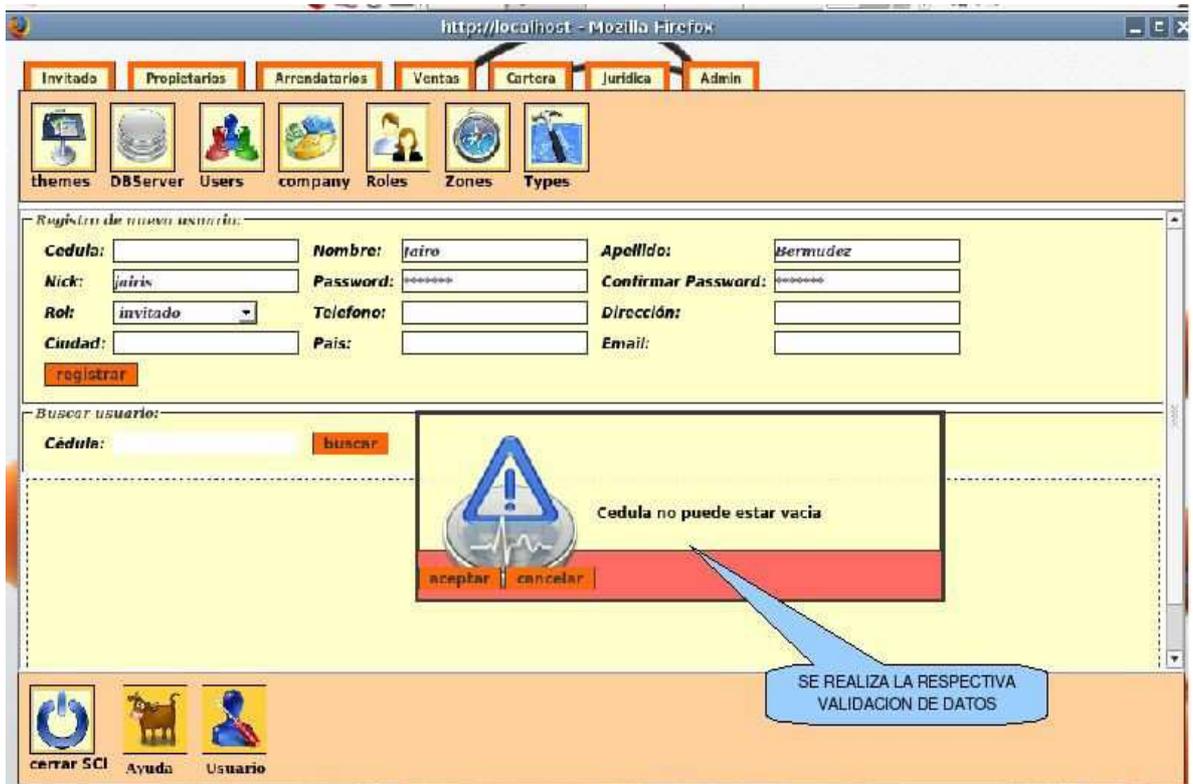


Figura No 15. Modulo Admin SCI, validación de datos

El el botón de compañía todos los datos de la empresa o inmobiliaria por ejemplo nombre, Nit, país , dirección y demás datos básicos.



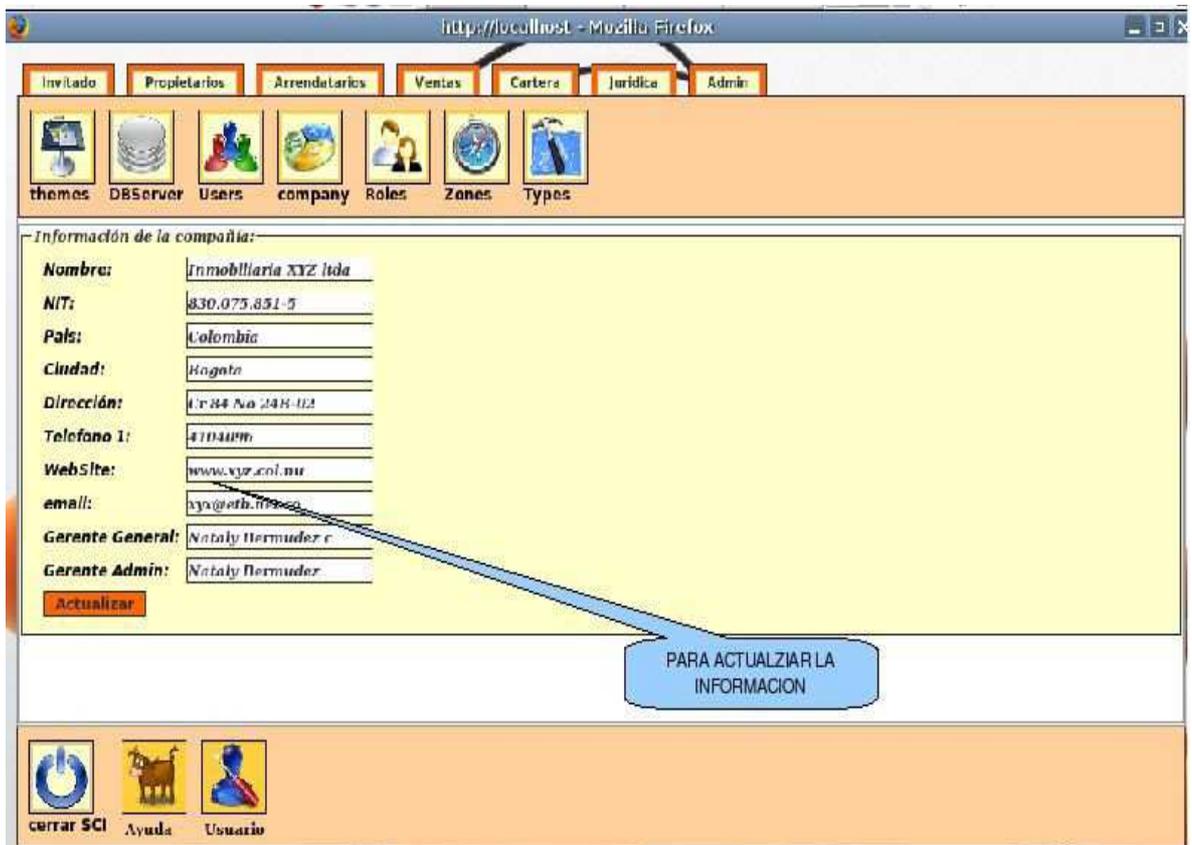


Figura No 16. Modulo Admin SCI, botón compañía



En el botón de roles puede crear como primera un rol de usuario como por ejemplo invitado, administrador etc, y realizar una breve descripción del mismo.

EN ESTA SECCION SE PUEDE CREAR UN NUEVO ROL DE USUARIO Y SU RESPECTIVA DESCRIPCION

SE PUEDE VERIFICAR LOS USUARIOS SU DESCRIPCION Y VERIFICAR INFO ADICIONAL, ELIMINAR O EDITAR

Id	Nombre	Descripcion	
6	invitado	Invitado	  
4	Administrador	Administrador del sistema SCI	  

Figura No 17. Modulo Admin SCI, botón Roles

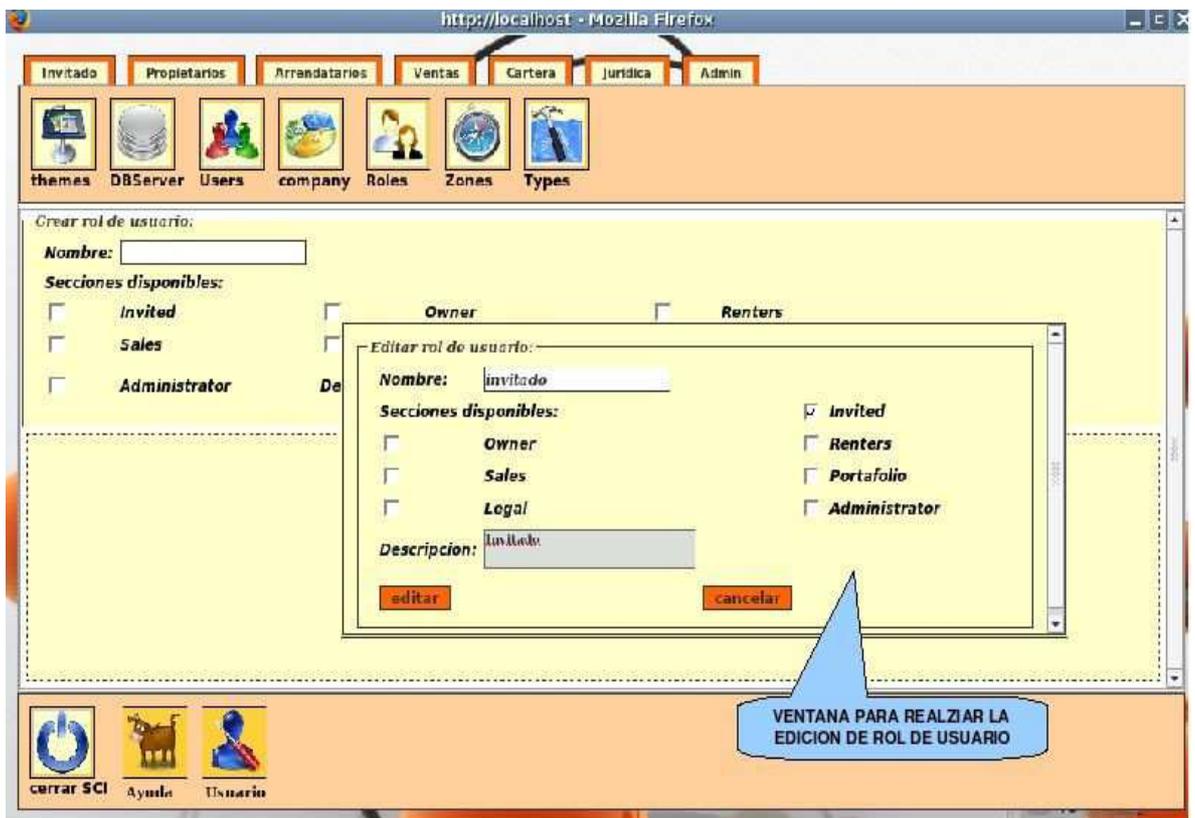


Figura No 18. Modulo Admin SCI, Editar rol de usuario

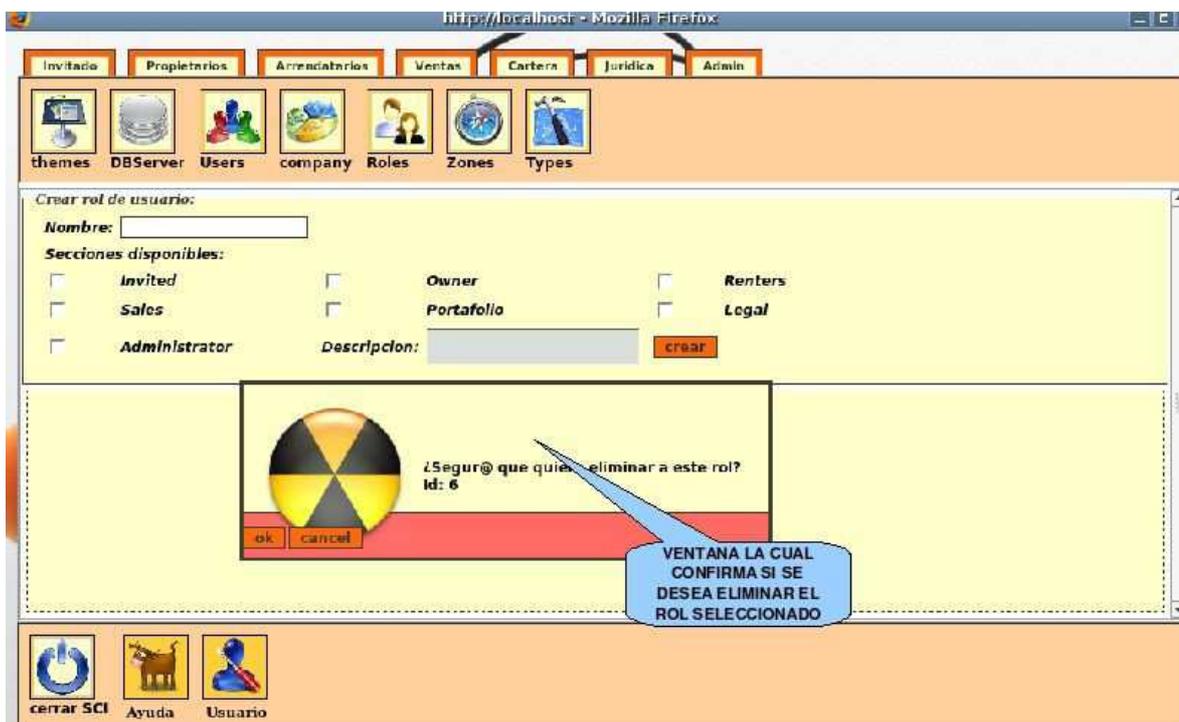


Figura No 19. Modulo Admin SCI, Eliminar rol



En la sección de zonas tendrá la opción de realizar la creación de una nueva zona esto significa donde se encuentran ubicados los inmuebles según los puntos cardinales.

Puede verificar los ya existentes y eliminarlos si se desea.

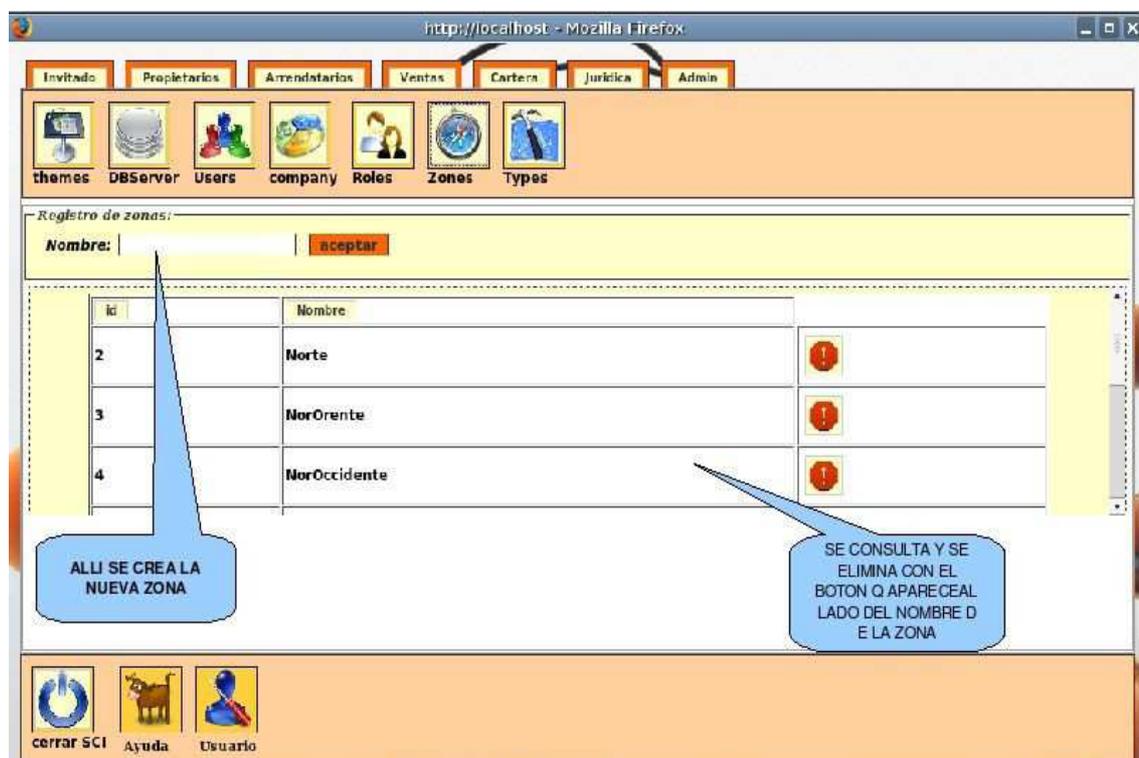


Figura No 20. Modulo Admin SCI, botón Zonas

En el botón de tipos podemos ver como en la imagen anterior los tipos de inmuebles registrados en el sistema y de la misma manera tiene la opción de crear un nuevo tipo de inmueble.

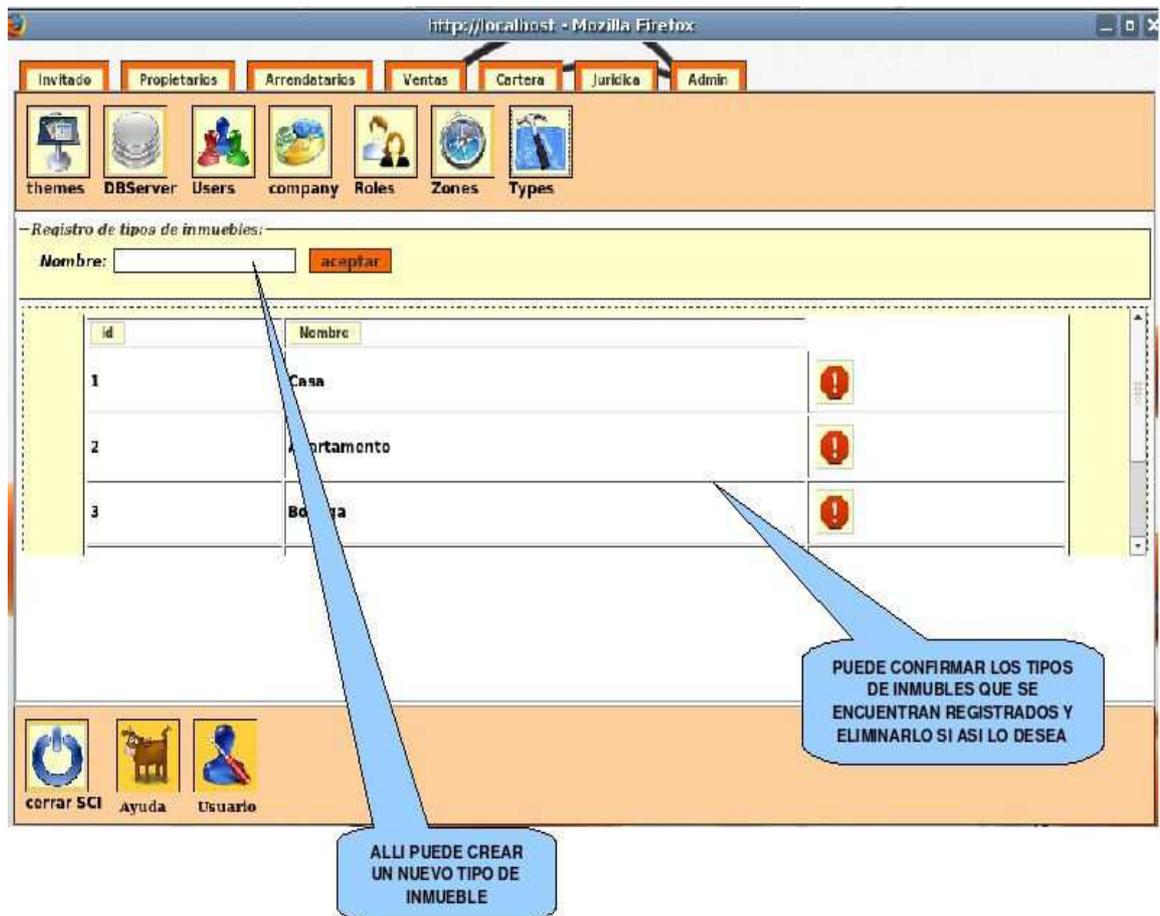
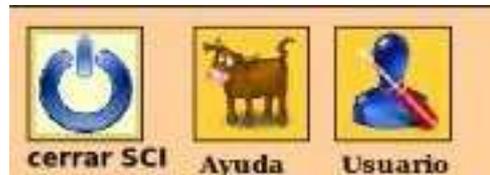


Figura No 21. Modulo Admin SCI, botón tipos



Cada una de las secciones de la aplicación trae en la parte inferior la ayuda, la cual presenta el manual de operaciones desde la aplicación , la opción de salida la cual permite la salida del usuario de manera limpia y la opción de usuario la cual le permite al usuario actual generar una consulta de sus datos y modificación de los mismos teniendo en cuenta que los datos como el nick y la identificación no pueden ser modificados .

Adicionalmente todas las opciones para cada usuario se presentan de acuerdo a su rol o su perfil en la barra superior izquierda donde aparece cada sección.