



# **Universidad del Azuay**

## **Facultad de Ciencias de la Administración**

### **Escuela de Ingeniería de Sistemas**

*Sistema de Gestión Inmobiliaria en Ambiente Web, utilizando  
Dispositivos Móviles, aplicado a la empresa INMOCAS*

**Trabajo de graduación previo a la obtención del título de  
Ingeniero de Sistemas**

**Autoras:**

**Claudia Pamela Castro Guerrero**

**Janina Patricia Zúñiga Solórzano**

**Director: Ing. Lenin Erazo Garzón**

**Cuenca, Ecuador**

**2010**

***Dedicatoria:***

Dedico la culminación de esta monografía a mis queridos padres por su apoyo y amor incondicional, además de ser mi inspiración primordial para el cumplimiento de cada una de mis metas.

Claudia Castro Guerrero

A mis padres por su ejemplo de trabajo y superación y por su apoyo incondicional. A mi hermana que ha estado siempre conmigo y su sonrisa ha iluminado muchos días de mi vida. A mi familia que me hace sonreír siempre. A mis amigas y amigos con los que hemos pasado hermosos momentos estos 5 años.

Janina Zúñiga Solórzano

### ***Agradecimientos:***

Agradecemos a Dios por darnos vida, salud y fortaleza para enfrentar todos los retos que se nos han ido presentando en este trayecto.

A nuestra familia, y de manera especial a nuestros padres por ser la fuerza que impulsa nuestra alas para seguir adelante, por su apoyo incondicional y por vivir con nosotros los buenos y malos momentos como si fueran suyos.

A nuestros amigos con los que compartimos momentos de estudio y amistad, por su apoyo y por hacernos sonreír en los malos momentos.

A nuestros maestros que nos brindaron la mejor formación humana y profesional, y que supieron con paciencia y cariño darnos lo mejor de su conocimiento, de manera especial al Ing. Lenin Erazo Garzón, quien nos guió desde el inicio de este proyecto.

## INDICE DE CONTENIDOS

Dedicatorias.....	ii
Agradecimientos .....	iii
Índice de Contenidos.....	iv
Resumen.....	vi
Abstract .....	vii
Introducción .....	1
Capítulo 1: Recolección y Levantamiento de la Información.....	2
1.1 Especificación de Requisitos de Software – ERS .....	2
1.1.1 Introducción.....	2
1.1.1.1 Propósito.....	2
1.1.1.2 Ámbito del Sistema.....	2
1.1.1.3 Definiciones.....	4
1.1.1.4 Visión General del Producto.....	4
1.1.2 Descripción General .....	5
1.1.2.1 Perspectiva del Producto.....	5
1.1.2.1.1 Indicar si es un producto independiente.....	5
1.1.2.1.2 Interfaces de sistema .....	5
1.1.2.1.3 Interfaces software .....	5
1.1.2.1.3.1 Descripción del software.....	5
1.1.2.1.3.2 Propósito de la interfaz .....	6
1.1.2.1.4 Operaciones de backup y recuperación.....	6
1.1.3 Requisitos Específicos.....	8
1.1.3.1 Requisitos Funcionales .....	8
1.1.3.2 Descripción de Actores.....	10
1.1.3.3 Descripción de casos de uso .....	10
1.1.3.4 Requerimientos de interfaces externos .....	20
1.1.3.4.1 Interfaces de software .....	20
1.1.3.4.2 Interfaces de usuario .....	21
1.1.3.5 Requisitos de Rendimiento.....	21
1.1.3.6 Restricciones de Diseño.....	21
1.1.3.7 Atributos del Sistema.....	22
Capítulo 2: Análisis y Diseño .....	23
2.1 Diagramación UML .....	23
2.1.1 Diagrama de Clases .....	25
2.1.2 Diagrama de Despliegue.....	26
2.1.3 Diagramas de Estado .....	26
2.1.4 Diagramas de Secuencia.....	43
2.2 Análisis y definición de los procesos .....	78
2.2.1 Diagrama de Contexto.....	78
2.2.2 Diagramas de Flujo de Datos .....	79
2.2.2.1 Diagramas de Nivel Superior.....	79
2.2.2.2 Diagramas de Expansión .....	82
2.3 Creación del Modelo Entidad-Relación .....	91

2.4 Normalización del Modelo .....	93
2.5 Definición de tablas e índices del esquema de base de datos.....	96
2.6 Diccionario de Datos .....	98
2.7 Definición del modelo de arquitectura .....	106
2.8 Definición de los niveles de seguridad y privilegios.....	107
2.9 Diseño de Navegación.....	108
2.10 Creación de la interfaz y las plantillas del Portal Web .....	109
2.10.1 Bosquejo General de la Interfaz .....	109
2.10.2 Diseño de Plantilla.....	111
2.11 Definición de la tecnología de integración con la base de datos.....	113
Capítulo 3: Codificación .....	116
3.1 Referencia sobre las Herramientas utilizadas para la Codificación ...	116
3.2 Estándares.....	117
3.3 Procesos Relevantes de la Codificación.....	118
3.3.1 Lineamientos Generales para los Mantenimientos.....	118
3.3.2 Ubicación Geográfica en el Mapa .....	120
3.3.3 Características Configurables .....	125
3.3.4 Aplicativo Móvil .....	131
3.3.5 Crear Galerías Fotográficas para cada propiedad.....	134
3.3.6 Búsquedas Especializadas .....	136
3.3.7 Reportes Gerenciales .....	137
3.3.8 Envío del Email .....	143
3.3.9 AdRotator .....	144
Capítulo 4: Instalación .....	146
4.1 Creación del instalador para el dispositivo móvil .....	146
4.2 Instalación de la Aplicación en el dispositivo móvil.....	150
4.3 Pruebas del Sistema.....	154
4.3.1 Prueba de Contenido.....	154
4.3.2 Prueba de la Interfaz del Usuario .....	155
4.3.3 Prueba de Componentes o Unidades .....	155
4.3.4 Pruebas de Navegación.....	156
4.3.5 Pruebas de Integración .....	156
Conclusiones y Recomendaciones .....	158
Bibliografía .....	160

## **RESUMEN**

La presente monografía trata sobre la automatización de la Gestión de una Inmobiliaria mediante un Sistema en Ambiente Web, además de la utilización de Dispositivos Móviles para recolectar datos con los que luego se alimentará al Sistema.

El sistema abarca principalmente estos aspectos: la publicidad y promoción vía web, la recolección de las características de los inmuebles en un dispositivo móvil, la configuración dinámica de las características que puede tener un inmueble, y la ubicación geográfica de la propiedad en Google Maps.

Estos aspectos esperan cubrir las necesidades del usuario del sitio web, del administrador del mismo y de los vendedores de la Inmobiliaria.

## **ABSTRACT**

This monograph deals with the automation of real state through a system in a web environment as well as the use of mobile devices to collect data, which will later feed the system.

The system principally includes these aspects: publicity and promotion through the Web, the dynamic configuration of the characteristics of the properties, the collection of the characteristics on a mobile device under the care of the sellers and the geographical location of the property on Google Maps.

The aspects are hoped to cover the needs of the Website user, the administrator and the sellers of the property.

## INTRODUCCIÓN

La presente monografía tiene como finalidad la creación de un Sistema para la Gestión de una Inmobiliaria, donde se abarcarán principalmente tres aspectos, la publicidad y promoción vía web, la portabilidad de los datos para ser utilizados en un dispositivo móvil y la configuración dinámica de las características que puede tener un inmueble.

En lo referente a la publicidad y promoción de los bienes inmuebles, este sistema busca la difusión de las propiedades mediante elementos multimedia como galerías de fotos y localización geográfica en un mapa; además de contar con búsquedas personalizadas que faciliten al usuario encontrar el inmueble que se ajuste a sus necesidades. La publicidad y promoción tiene como finalidad que el cliente se interese en las propiedades y posteriormente efectúe una reservación para que le sea mostrada. La publicidad se realizará mediante el correo electrónico del cliente, a donde se le enviarán las ofertas de las propiedades, según sus solicitudes.

Al respecto de la portabilidad, este sistema permitirá a los agentes vendedores de la inmobiliaria, la fácil recolección en un dispositivo móvil de las diferentes características de cada una de las propiedades a ser vendidas para su posterior descarga en la Base de Datos de la empresa. En cuanto a las características configurables dinámicamente, este sistema permitirá agregar características a los diferentes tipos de propiedades mediante una interfaz de usuario web sin que el administrador necesite desarrollar tareas adicionales de programación.

Este sitio beneficiará tanto al cliente como a los empleados de la inmobiliaria en su trabajo operativo. Este tema se ha elegido, porque hemos visto importante la automatización de tareas que se hacen manualmente y que ocupan tiempo valioso del personal de la empresa, el mismo que podrían dedicar a tareas de mayor importancia. La parte móvil del proyecto es para aprovechar las tecnologías actuales en beneficio de la empresa.



# CAPITULO I

## RECOLECCIÓN Y LEVANTAMIENTO DE LA INFORMACIÓN

### 1.1 Especificación de Requisitos de Software – ERS

#### 1.1.1 INTRODUCCIÓN

Este documento es una Especificación de Requisitos de Software para un Sistema de Gestión Inmobiliaria en Ambiente Web, Utilizando Dispositivos Móviles Aplicado a la Empresa INMOCAS. Todo su contenido será elaborado con la colaboración del personal de la Empresa conjuntamente con las desarrolladoras del sistema.

##### 1.1.1.1 Propósito

El propósito de la Especificación de Requisitos es definir de forma clara la funcionalidad del sistema que se va a desarrollar.

Este documento constituye la base para el posterior desarrollo de este proyecto.

##### 1.1.1.2 Ámbito del sistema

El nombre del Sistema es: Sistema de Gestión Inmobiliaria en Ambiente Web, Utilizando Dispositivos Móviles Aplicado a la Empresa INMOCAS.

Este Sistema tendrá las siguientes funcionalidades:

- Galerías fotográficas para cada propiedad.
- Búsquedas especializadas para los clientes.
- Reportes gerenciales
- Un mapa donde se pueda ver la ubicación geográfica del inmueble.
- Recepción de solicitudes por parte de los clientes sobre propiedades con características específicas que ellos estén buscando. En caso de ser

encontrada una oferta que se adapte a sus especificaciones, se enviará al correo electrónico del cliente una notificación para que ingrese al portal de la empresa y corrobore si el inmueble ofertado cumple con sus expectativas.

- Reservas para solicitar un avalúo o para que el cliente conozca una determinada propiedad.
- Aplicativo en un dispositivo móvil que permita a los vendedores recolectar la información acerca de las propiedades que el cliente desea vender. Esta información luego puede ser descargada en la Base de Datos de la Empresa.

El objetivo del proyecto es construir un sistema que permita a los clientes y al público en general, obtener información detallada acerca de las propiedades que oferta la inmobiliaria mediante recursos multimedia como galerías fotográficas y un mapa donde se puede ubicar al inmueble. Adicionalmente, el cliente podrá realizar búsquedas especializadas según las características del inmueble que está buscando. En caso de tener interés, el portal Web permite que el cliente realice una reservación para conocer el inmueble. Los clientes que deseen que se les avalúe sus propiedades pueden realizar una cita para este fin. Los clientes que deseen suscribirse para recibir notificaciones vía correo electrónico acerca de las ofertas de su interés contarán con una opción para registrarse.

Las metas de este proyecto son:

- Que el cliente obtenga información clara, precisa y detallada acerca de cada propiedad utilizando recursos multimedia.
- La promoción de las propiedades ofertadas mediante correo electrónico y las búsquedas personalizadas.
- Facilitar las reservas para avalúos o para conocer las propiedades.

### 1.1.1.3 Definiciones

Dispositivos móviles	También conocidos como computadora de mano, Palmtop. Son aparatos pequeños, con ciertas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.
Visual Basic.Net	Es un lenguaje de programación orientada a objetos. Es un producto de Microsoft que pertenece al entorno integrado de Microsoft Visual Studio. Se considera una evolución de Visual Basic 6.0.
Crystal Reports	Es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una base de datos. Microsoft Visual Studio, incluyen una versión de Crystal Reports como una herramienta de propósito general de informes y reportes.
Smart Device	Es una herramienta de Microsoft Visual Studio para el desarrollo de aplicativos que se pueden instalar en dispositivos móviles y otras plataformas basadas en .NET. Cuenta con un emulador para probar la aplicación.
JavaScript	Es un lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.
MySql	MYSQL es un sistema de gestión de bases de datos relacional (SGBD) multiusuario, multiplataforma, multihilo y de código abierto. Pertenece a la compañía sueca MySQL AB, a la que le pertenece casi todos los derechos del código fuente.

Tabla 1.1 Definiciones

### 1.1.1.4 Visión General del Producto

Este documento ha sido organizado en secciones. La sección Introdutoria proporciona una visión general de la ERS. La sección de Descripción General permite conocer la perspectiva, funciones, características, restricción y requisitos de

este sistema sin detalle. Al final, se encuentra la sección de Requisitos Específicos donde se detallan los requisitos que debe satisfacer este sistema.

## **1.1.2 DESCRIPCIÓN GENERAL**

En esta sección se presenta una descripción de alto nivel del sistema que incluye la funcionalidad del sistema, la información que se utilizará, y otros factores que afectan al desarrollo de este proyecto.

### **1.1.2.1 Perspectiva del Producto**

Se espera que este sistema facilite a los clientes la obtención de información y permita una mayor promoción de los bienes inmuebles que oferta la inmobiliaria.

#### **1.1.2.1.1 Indicar si es un producto independiente o parte de un sistema mayor**

Este es un sistema independiente que no parte de un sistema mayor.

#### **1.1.2.1.2 Interfaces de sistema**

Las interfaces de este sistema serán realizadas en Visual Basic.Net utilizando Web Forms, para que se puedan visualizar en un browser. En lo referente al aplicativo para dispositivos móviles las interfaces se realizarán en Smart Device.

#### **1.1.2.1.3 Interfaces software**

##### **1.1.2.1.3.1 Descripción del producto software utilizado**

Los productos software que se van utilizar para el desarrollo de este proyecto son:

- Programación de Web Forms: Visual Basic.Net y JavaScript
- Lenguaje para consultas: SQL
- Gestor de Base de Datos: MySQL
- Gestor de Reportes: Crystal Reports

#### **1.1.2.1.3.2 Propósito de la interfaz**

El propósito de la interfaz es facilitar la búsqueda y entrada de datos por parte del usuario. La interfaz debe ser intuitiva, fácil de manejar y amigable. Adicionalmente, los reportes gerenciales que necesite el usuario deben tener un formato estándar.

#### **1.1.2.1.4 Operaciones de backup y recuperación**

Los datos recolectados se van a almacenar en el Gestor de Base de Datos MySQL.

### **Funciones del Producto**

Este producto proporcionará a los visitantes y a los clientes de INMOCAS, la oportunidad de conocer de manera más detallada los inmuebles que se encuentran en oferta; adicionalmente se ofrece la reserva de citas para avalúos y para conocer las propiedades de su interés. Este software permitirá a la empresa ampliar su mercado potencial y promocionarse a un mayor número de personas ya que actualmente la Internet es un medio de comunicación que llega a todo el mundo.

El mecanismo de este producto será el siguiente: el usuario ingresa al portal Web donde encontrará diferentes opciones para elegir: propiedades ofertadas, búsquedas especializadas, reservas, avalúos y registrarse como usuario.

En la sección de propiedades ofertadas se presentará al cliente todos los inmuebles que están a la venta. Al escoger una propiedad específica se presentará como texto todas las características que posee, además de una galería fotográfica para visualizarlas de forma más real. Cada propiedad tendrá un vínculo hacia un mapa donde se podrá ver su ubicación geográfica. Si el cliente tiene interés en conocer personalmente el inmueble, en esta sección habrá un vínculo que llevará directamente a la sección de reservas. En caso de ser la primera vez que la persona hace una reserva, primero deberá registrarse como usuario.

En la sección de búsquedas especializadas los usuarios tendrán la opción de buscar una propiedad según las características que desean con la finalidad de determinar si

alguno de los inmuebles ofertados cumple con lo solicitado. Se presentarán los inmuebles que cumplan con todas las características especificadas en la búsqueda. Las búsquedas se podrán realizar por la ubicación, es decir, por provincia, ciudad, parroquia y sector; adicionalmente se permitirá una búsqueda por tipo de inmueble, por precio y por metros cuadrados.

En la sección de reservas para ventas, el cliente hará una cita para conocer personalmente una propiedad de su interés. El primer paso para realizar una reserva es estar registrado como usuario para conocer todos los datos personales del cliente. En segundo lugar el usuario deberá elegir un fecha y hora tentativa, y a continuación el sistema le asignará un agente vendedor, pero esto está sujeto a cambio debido a que el tiempo de duración de una cita es totalmente subjetivo, es decir, depende de muchos factores tales como la distancia a la que se encuentra el inmueble, el temperamento del cliente, factores externos y ambientales, entre otras cosas, por lo cual se realizará una confirmación de la cita vía telefónica, informándole al cliente la hora y fecha exactos, además del nombre del vendedor que lo acompañará.

La sección de avalúos es para acceder a un servicio especial donde el agente vendedor visitará una propiedad para evaluarla y en un futuro pueda ser ofertada por la Inmobiliaria. Este servicio tiene un precio. Para realizar una cita para avalúos el mecanismo a seguir será igual al de las reservas, la diferencia es que el cliente deberá pagar por este servicio.

Registrarse como usuario es una sección que le ayudará a la empresa a tener una base de datos de clientes potenciales a quienes enviarles la información vía correo electrónico sobre nuevas ofertas y promociones en respuesta a sus peticiones. A las personas que deseen registrarse como usuarios se les hará llenar un formulario con sus datos personales (obligatorios y opcionales). Los usuarios registrados, como ya se indicó anteriormente, podrán realizar reservas, avalúos, peticiones y recibir información promocional.

## **Características de los usuarios**

Esta aplicación va dirigida a los clientes establecidos, al mercado potencial y al personal de la empresa INMOCAS. El público en general también podrá tener acceso a la información.

## **Suposiciones y Dependencias**

Se asume que los requisitos descritos en este documento no tendrán variaciones grandes luego de haber sido aprobados por el gerente de la empresa.

## **Requisitos Futuros**

A medida que avance el sistema puede necesitarse afinar los criterios de búsqueda.

### **1.1.3 REQUISITOS ESPECÍFICOS**

Esta sección contiene todos los requisitos del sistema a nivel de detalle para diseñar un proyecto que satisfaga al cliente y que permita el diseño de las pruebas que ratifiquen que el sistema cumple con las necesidades.

#### **1.1.3.1 Requisitos Funcionales**

Los requisitos funcionales se agruparán en casos de uso, que darán una perspectiva del sistema desde el punto de vista de los actores, y dará una visión general del alcance del sistema.



Figura 1.1: Diagrama de Casos de Uso del Sistema de Gestión Inmobiliaria



### 1.1.3.2 Descripción de Actores

**Visitantes:** navegan por el sitio Web, pueden registrarse como usuarios, eligen propiedades, ven las características de los inmuebles, realizan búsquedas y, consultan la ubicación geográfica en el mapa.

**Clientes:** navegan por el sitio Web, eligen propiedades, ven las características de los inmuebles, realizan búsquedas, consultan la ubicación geográfica en el mapa, realizan reservas, solicitan avalúos, solicitan inmuebles y modifican sus datos.

**Agentes Vendedores:** Recogen características de las propiedades en los dispositivos móviles y, descargan la información recolectada en la base de datos de la empresa.

**Administrador:** Modifica las características de las propiedades, sube fotografías, crea reportes gerenciales, envía correos electrónicos, realiza reservas a nombre de los clientes que se comuniquen vía telefónica o que visiten las instalaciones de la Empresa y administra toda la información de los propietarios, clientes, vendedores y tipos de inmuebles.

### 1.1.3.3 Descripción de casos de uso

Prioridad de automatización

Manual	La funcionalidad se realiza manualmente y no se registra inherentemente en el sistema.
Opcional	Se especificará explícitamente por parte de la persona responsable si se automatizará las funcionalidades, bajo la supervisión del responsable del control de aceptación de los ERS.
Necesario	Estas funcionalidades podrían ser implementadas de diferentes formas
Obligatorio	Estas funcionalidades serán automatizadas 100%

<b>Caso de uso 1</b>	Registrarse como Usuario
<b>Actor:</b>	Visitante
<b>Descripción:</b>	El visitante se registra con sus datos personales para acceder a los servicios de la inmobiliaria.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.1.1 El sistema permitirá registrar los datos del visitante como: Cédula, nombres, apellidos, dirección, teléfono, celular, correo electrónico, nombre de usuario y contraseña.	
R.1.2 El sistema envía un correo electrónico de bienvenida al nuevo usuario registrado	

<b>Caso de uso 2</b>	Consultar Propiedades
<b>Actor:</b>	Cliente y Visitante
<b>Descripción:</b>	El cliente o el visitante consultan una propiedad específica para conocer su detalle.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.2.1 El sistema deberá listar todas las propiedades disponibles según el tipo de inmueble que el cliente escoja.	
R.2.2 El sistema debe mostrar los datos para ubicar el inmueble como: provincia, ciudad, parroquia, sector y dirección. Se debe mostrar el precio y una foto.	
R.2.3 El sistema para cada inmueble mostrado tendrá tres enlaces: detallar las características del inmueble, reservaciones para conocer la propiedad personalmente y consultar la ubicación geográfica.	

<b>Caso de uso 3</b>	Consultar Galería Fotográfica
<b>Actor:</b>	Cliente y Visitante
<b>Descripción:</b>	El cliente o el visitante observan las fotos de la propiedad.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.3.1 El sistema mostrará en miniatura todas las fotografías asociadas al inmueble que el visitante o el cliente escoge ver.</p> <p>R.3.2 El sistema mostrará en tamaño grande la fotografía específica que el cliente o el visitante escoja.</p> <p>R.3.3 El sistema permitirá que se recorra la galería con botones para ir a: la siguiente, la anterior, la primera y la última foto.</p> <p>R.3.4 El sistema aceptará como máximo 10 fotografías por cada inmueble.</p>	

<b>Caso de uso 4</b>	Ver Características
<b>Actor:</b>	Cliente y Visitante
<b>Descripción:</b>	El cliente o el visitante leen las características físicas que se detallan de cada propiedad
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.4.1 El sistema debe listar todas las características que posee el inmueble que el cliente escogió detallar.</p>	

<b>Caso de uso 5</b>	Consultar Ubicación Geográfica
<b>Actor:</b>	Cliente y Visitante
<b>Descripción:</b>	El cliente o el visitante pueden consultar en un mapa la ubicación exacta del inmueble
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.5.1 El sistema obtendrá las coordenadas de donde se encuentra el inmueble.</p> <p>R.5.2 El sistema mostrará en un mapa de Google Maps la ubicación exacta del inmueble que el cliente escogió localizar.</p>	

<b>Caso de uso 6</b>	Realizar búsquedas
<b>Actor:</b>	Visitante y Cliente
<b>Descripción:</b>	El cliente o el visitante pueden realizar búsquedas personalizadas para encontrar si una propiedad satisface sus requerimientos.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.6.1 El sistema permitirá realizar búsquedas de inmuebles por: provincia, ciudad, parroquia, sector, precio mínimo, precio máximo, metros mínimos, metros máximos y por tipo de inmueble.</p> <p>R.6.2 El sistema permitirá buscar según uno o varios de los criterios mencionados, a excepción del tipo de inmueble que será obligatorio.</p>	

<b>Caso de uso 7</b>	Realizar Peticiones
<b>Actor:</b>	Cliente o Administrador
<b>Descripción:</b>	El cliente puede realizar peticiones personalizadas para solicitar que el administrador le envíe un correo electrónico cuando encuentre una propiedad que satisfaga sus requisitos.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.7.1 El sistema aceptará las características del inmueble que el cliente está buscando como: tipo de inmueble, precio mínimo, precio máximo, metros totales de terreno, provincia, ciudad, parroquia, sector y otros datos adicionales.</p> <p>R.7.2 El sistema permitirá al administrador modificar una petición.</p> <p>R.7.3 El sistema permitirá al administrador eliminar una petición.</p> <p>R.7.4 El sistema permitirá al administrador consultar las peticiones según un criterio de búsqueda específico.</p>	

<b>Caso de uso 8</b>	Realizar reservas
<b>Actor:</b>	Cliente
<b>Descripción:</b>	El cliente realiza una cita para conocer una propiedad específica
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.8.1 El sistema receptorá la fecha y hora en la que el cliente desea conocer un inmueble específico.</p> <p>R.8.2 El sistema manejará los siguientes estados en las reservas: realizada, confirmada y cancelada.</p> <p>R.8.3 El sistema permitirá al administrador modificar la reservación.</p> <p>R.8.4 El sistema permitirá al administrador eliminar la reservación.</p> <p>R.8.5 El sistema permitirá que el administrador asigne un vendedor a la reservación cuando ésta sea confirmada.</p> <p>R.8.6 El sistema permitirá al administrador consultar las reservas según un criterio de búsqueda específico.</p>	

<b>Caso de uso 9</b>	Solicitar avalúos
<b>Actor:</b>	Cliente
<b>Descripción:</b>	El cliente hace una cita para se le visite y se avale su propiedad
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.9.1 El sistema receptorá la fecha y hora en la que el cliente desea que se le haga un avalúo al inmueble.</p> <p>R.9.2 El sistema manejará los siguientes estados en la solicitud de avalúo: realizada, confirmada y cancelada.</p> <p>R.9.3 El sistema permitirá al administrador modificar los datos de la solicitud de avalúo.</p> <p>R.9.4 El sistema permitirá al administrador eliminar la solicitud de avalúo.</p> <p>R.9.5 El sistema permitirá que el administrador asigne un vendedor a la solicitud de avalúo cuando ésta sea confirmada.</p> <p>R.9.6 El sistema permitirá al administrador consultar las solicitudes según un criterio de búsqueda específico.</p>	

<b>Caso de uso 10</b>	Mantenimiento de Clientes
<b>Actor:</b>	Cliente o Administrador
<b>Descripción:</b>	El Administrador o el Cliente pueden modificar o eliminar sus datos personales.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.10.1 El sistema almacenará la información de nuevos clientes tales como: Cédula, nombre, dirección, teléfono, celular, email, usuario y contraseña.</p> <p>R.10.2 El sistema permitirá modificar los datos de los clientes.</p> <p>R.10.3 El sistema permitirá al administrador eliminar a un cliente.</p> <p>R.10.3 El sistema permitirá al administrador consultar un cliente de acuerdo a un criterio de búsqueda específico.</p> <p>R.10.4 El administrador o el cliente puede eliminar la cuenta</p>	

<b>Caso de uso 11</b>	Administrar Propiedades
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El administrador puede ingresar, eliminar o modificar los datos de los inmuebles.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.11.1 El sistema permitirá ingresar nuevas propiedades</p> <p>R.11.2 El sistema mostrará las características según el tipo de inmueble.</p> <p>R.11.3 El sistema almacenará dinámicamente las características de cada inmueble.</p> <p>R.11.3 El sistema permitirá modificar los datos del inmueble.</p> <p>R.11.4 El sistema permitirá eliminar el inmueble.</p> <p>R.11.5 El sistema permitirá consultar los inmuebles según un criterio de búsqueda específico.</p>	

<b>Caso de uso 12</b>	Crear Reportes Gerenciales
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El Administrador realiza reportes gerenciales en un rango de fechas
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.12.1 El sistema permitirá generar un reporte del vendedor que más ventas ha realizado en un rango de fechas específicas.</p> <p>R.12.2 El sistema permitirá generar un reporte del tipo de inmueble más vendido dentro de un rango de fechas.</p> <p>R.12.3 El sistema permitirá generar un reporte de las ventas realizadas dentro de un rango de fechas.</p>	

<b>Caso de uso 13</b>	Crear Reportes de Ventas
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El Administrador crea reportes de las ventas realizadas en un rango de fechas
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.13.1 El sistema permitirá generar un reporte de las ventas realizadas dentro de un rango de fechas.</p> <p>R.13.2 El sistema presentará un conteo total del número de ventas realizadas dentro de un rango de fechas</p> <p>R.13.3 El sistema mostrará en el reporte el número de venta, la fecha, el inmueble, el propietario del inmueble y el precio al que fue vendido.</p>	

<b>Caso de uso 14</b>	Crear un Reporte con el Tipo de Inmueble más Vendido
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El Administrador realiza un reporte del tipo de inmueble más vendido en un rango de fechas
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.14.1 El sistema permitirá generar un reporte del tipo de inmueble más vendido dentro de un rango de fechas.</p> <p>R.14.2 El sistema presentará un gráfico representativo de los resultados con los porcentajes respectivos de acuerdo al tipo de bien.</p> <p>R.14.3 El sistema mostrará en el reporte el tipo de bien vendido, el inmueble, el precio final y la fecha en la que se realizó la venta.</p> <p>R.14.4 El sistema calculará el total de inmuebles vendidos dentro de rango de fechas</p>	

<b>Caso de uso 15</b>	Crear un Reporte del Vendedor más exitoso
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El Administrador realiza un reporte del vendedor más exitoso en un rango de fechas
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.15.1 El sistema permitirá generar un reporte del vendedor que más ventas ha realizado en un rango de fechas específicas.</p> <p>R.15.2 El sistema presentará un gráfico representativo de los resultados con los porcentajes respectivos de acuerdo a las ventas de cada vendedor.</p> <p>R.15.3 El sistema mostrará el conteo de ventas y la suma de los precios de los inmuebles vendidos por cada empleado.</p> <p>R.15.4 El sistema mostrará el conteo de ventas total y la suma de los precios de los inmuebles vendidos.</p>	



<b>Caso de uso 16</b>	Enviar Correo Electrónico
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El administrador envía al clientes un correo electrónico si ha encontrado una propiedad según lo solicitado
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.16.1 El sistema permitirá al administrador enviar un email a un cliente como respuesta a alguna de las peticiones que este último haya realizado.	
R.16.2 El sistema enviará en el email la fecha y el tipo de inmueble de la petición a la que se está dando respuesta, también el código del inmueble que el cliente tiene que buscar en el portal web para poder visualizar el inmueble sugerido.	

<b>Caso de uso 17</b>	Administrar Ventas
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El Administrador ingresa las ventas de cada propiedad
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.17.1 El sistema permitirá ingresar la fecha de la venta, el inmueble vendido, el vendedor que intervino en la venta, el precio final del inmueble y el cliente que realizó la compra.	
R.17.2 El sistema permitirá modificar la venta.	
R.17.3 El sistema permitirá eliminar la venta.	
R.17.4 El sistema permitirá consultar las ventas de acuerdo a un criterio específico.	

<b>Caso de uso 18</b>	Crear Archivos XML
<b>Actor:</b>	Administrador
<b>Descripción:</b>	El Administrador escoge los XML que va importar y poner en el pocket para su funcionamiento
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.18.1 El sistema creará archivos XML con los datos de los propietarios, las características y los tipos de inmuebles para su uso en el dispositivo móvil.	

<b>Caso de uso 19</b>	Mantenimiento Vendedor
<b>Actor:</b>	Vendedor o Administrador
<b>Descripción:</b>	El Administrador o el Vendedor pueden gestionar sus datos personales.
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.19.1 El sistema permitirá al administrador ingresar un nuevo vendedor.</p> <p>R.19.2 El sistema permitirá modificar los datos de un vendedor.</p> <p>R.19.3 El sistema permitirá al administrador eliminar un vendedor.</p> <p>R.19.4 El sistema permitirá consultar los datos de un vendedor según un criterio específico.</p>	

<b>Caso de uso 20</b>	Recoger Características
<b>Actor:</b>	Vendedor
<b>Descripción:</b>	El vendedor recoge las características físicas de la propiedad o inmueble que desea ser vendido
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
<p>R.20.1 El sistema permitirá ingresar los datos del inmueble en el aplicativo móvil.</p> <p>R.20.2 El sistema mostrará las características según el tipo de inmueble que se quiera ingresar.</p> <p>R.20.3 El sistema permitirá grabar en un archivo XML la información de todos los inmuebles que se han ingresado.</p>	

<b>Caso de uso 21</b>	Descargar características
<b>Actor:</b>	Vendedor
<b>Descripción:</b>	El vendedor descarga en la Base de Datos de la empresa la información recolectada de las propiedades
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.21.1 El sistema podrá ingresar inmuebles a partir de los datos contenidos en archivos XML.	
R.2.1.2 El sistema podrá ingresar nuevos propietarios a partir de los datos contenidos en archivos XML.	
R.21.3 El sistema validará los datos de los archivos XML antes de almacenarlos.	

<b>Caso de uso 22</b>	Consultar Horarios
<b>Actor:</b>	Vendedor
<b>Descripción:</b>	El Vendedor consulta los horarios en los que tiene las citas con los clientes
<b>Prioridad:</b>	Obligatorio
<b>Requisitos Asociados</b>	
R.22.1 El sistema permitirá al vendedor consultar los horarios, inmuebles y clientes a los que tendrá que atender en un día específico.	
R.22.2 El sistema mostrará un conteo del número de citas que el vendedor tenga en un día	

### 1.1.3.4 Requerimientos de interfaces externos

#### 1.1.3.4.1 Interfaces de software

Las máquinas de los usuarios deben tener un browser para visualizar el portal. Los navegadores compatibles son Firefox e Internet Explorer a partir de la versión 6.

Los dispositivos móviles deben tener instalado la versión móvil del sistema para recolectar la información y Windows Mobile.

#### **1.1.3.4.2 Interfaces de usuario**

La interfaz de este sistema es orientada a la Web y a dispositivos móviles en el caso de los agentes vendedores.

La interfaz para ambos casos debe ser:

- Clara e intuitiva.
- Equilibrada, esto quiere decir que no sea minimalista pero tampoco tenga sobrepoblación de enlaces y elementos.
- Capacidad de respuesta, esto significa que la interfaz no debe sentirse pesada
- Consistente
- Estética
- Eficiente

Estas interfaces deben diseñarse para funcionar en Internet Explorer 6.0 o versiones superiores u otros navegadores como Mozilla, que son los más comúnmente usados. Una parte del sistema va dirigido para dispositivos móviles, por lo que algunas interfaces deben diseñarse para funcionar correctamente en dispositivos móviles.

#### **1.1.3.5 Requisitos de Rendimiento**

Este sistema debe soportar el acceso de varios usuarios simultáneamente desde sus propias terminales mediante un Portal Web.

#### **1.1.3.6 Restricciones de Diseño**

Los estándares acerca de colores, íconos y nombres que se usarán en las aplicaciones deben hacerse en consenso por todos los miembros del equipo de desarrollo.

### **1.1.3.7 Atributos del Sistema**

A este sistema se ingresa a través de un Portal Web, pero para acceder a los servicios especiales del sistema se tendrán que contar con una cuenta de usuario y una contraseña.

## CAPITULO 2

### ANÁLISIS Y DISEÑO

#### 2.1 Diagramación UML

Se ha elegido utilizar UML 2.0 (Lenguaje Unificado de Modelo) porque nos ayuda a especificar, visualizar y documentar modelos de un sistema, incluyendo su estructura y diseño.

UML 2.0 posee 13 tipos de diagramas divididos en 2 categorías: diagramas de estructura estática y diagramas de comportamiento. Los diagramas de interacción son una subclase de los diagramas de comportamiento.

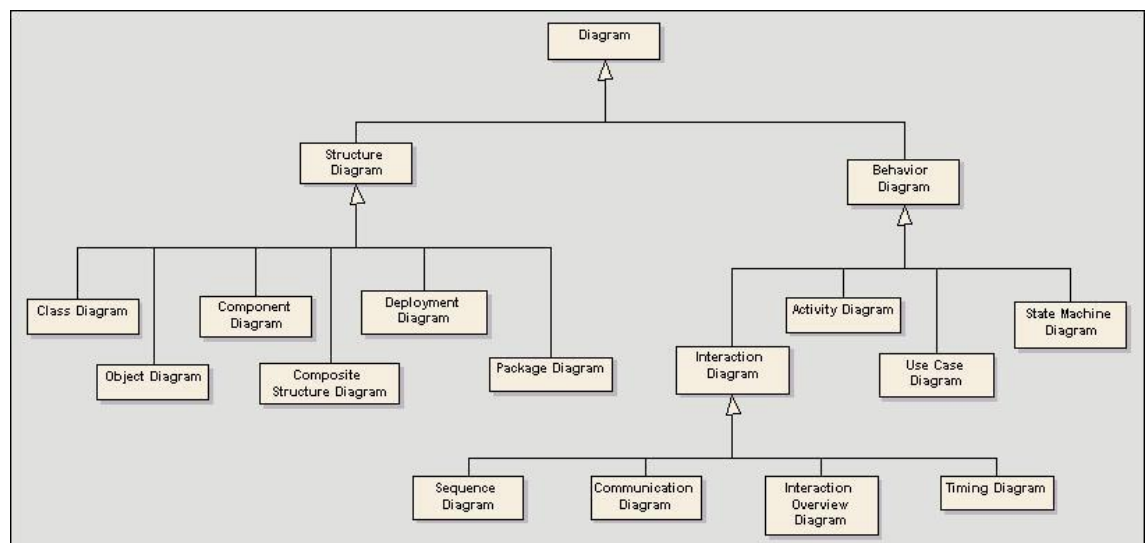


Figura 2.1: Clasificación de los Diagramas UML

Los diagramas que se han utilizado para modelar el Sistema de Gestión Inmobiliaria son:

## Diagramas de Estructura Estática

- **Diagrama de Clases:** Capturan las relaciones estáticas del software, que atributos tienen las clases y cuáles son sus referencias hacia otras clases. Este diagrama proporciona la estructura física de un sistema.
- **Diagrama de Despliegue:** Este diagrama muestra un mapa de los componentes software de un sistema y el hardware en el que se van a ejecutar para modelar la arquitectura en tiempo de ejecución de un sistema. Al software se le llama artefactos y al hardware que lo contiene nodos. El hardware que se usa para conectar hardware y software (ratón, teclado) se les llama rutas de comunicación.

## Diagramas de Comportamiento

- **Diagramas de Casos de Uso:** Sirven para capturar la funcionalidad del sistema y sus requerimientos, es decir, que tiene que hacer el sistema. En este diagrama se involucra los elementos funcionales (casos de uso) y los actores (personas que interactuarán con el sistema). Cada caso de uso tiene su respectiva descripción textual.
- **Diagramas de Estado:** Capturan el comportamiento de un sistema. Se usan para modelar el comportamiento de una clase, un caso de uso o la aplicación entera. Muestra los estados por los que pasa el sistema, como producto de las comunicaciones con las entidades externas, es decir luego de que se desencadena un evento producido por un usuario (ej., de evento: iniciar sesión).
- **Diagramas de Interacción:** Estos diagramas describen la comunicación entre objetos, estas comunicaciones y sus respuestas son lo que llamamos interacciones.
  - **Diagramas de Secuencia:** Estos diagramas describen una secuencia específica según los eventos van ocurriendo. Se enfocan en los

mensajes que intercambian los objetos, junto con sus respuestas a lo largo de las Líneas de Vida.

Para la realización de estos Diagramas utilizamos Microsoft Visio 2003, en la Sección Software en Diagrama de Modelo UML.

### 2.1.1 Diagrama de Clases

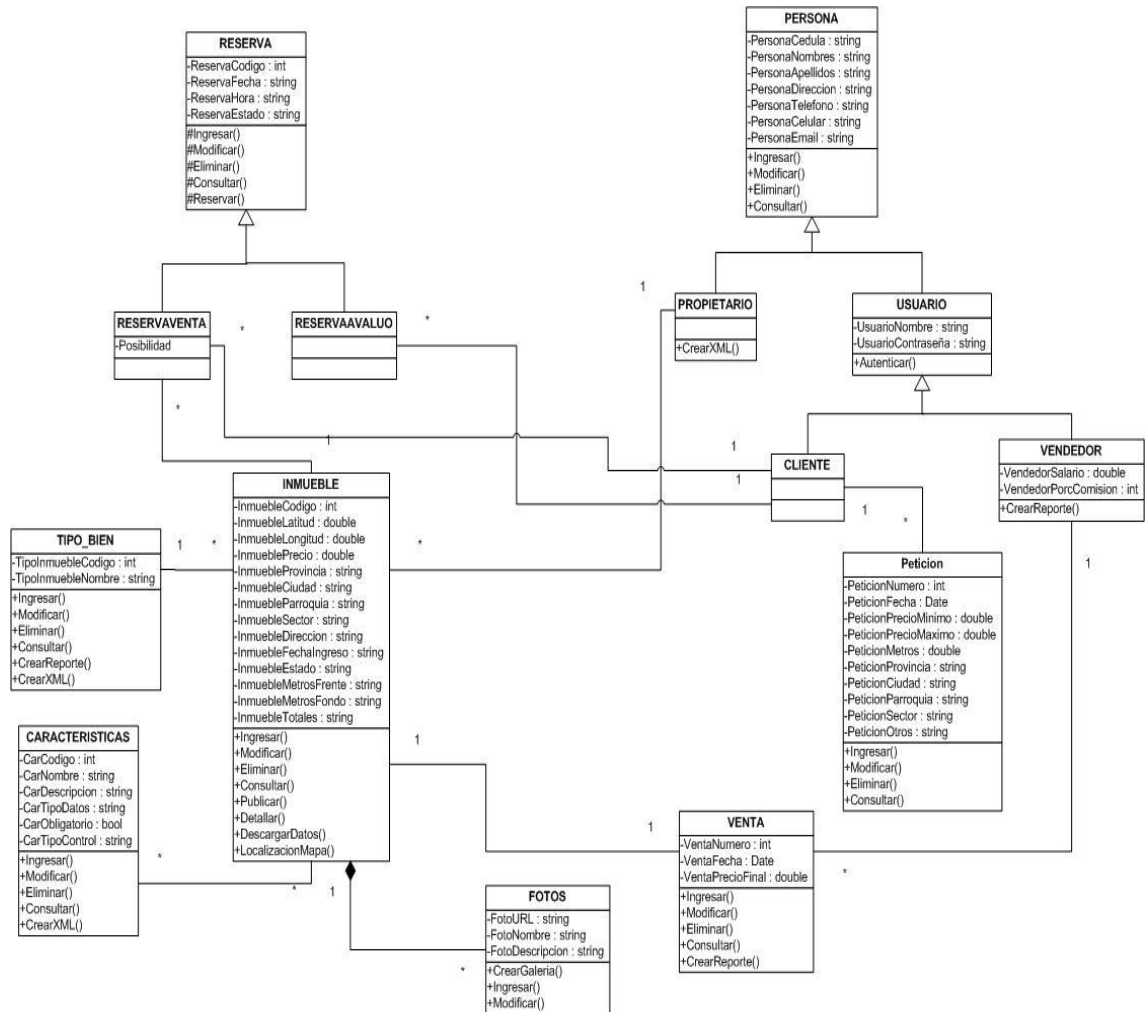


Figura 2.2: Diagrama de Clases del Sistema de Gestión Inmobiliaria



## 2.1.2 Diagrama de Despliegue

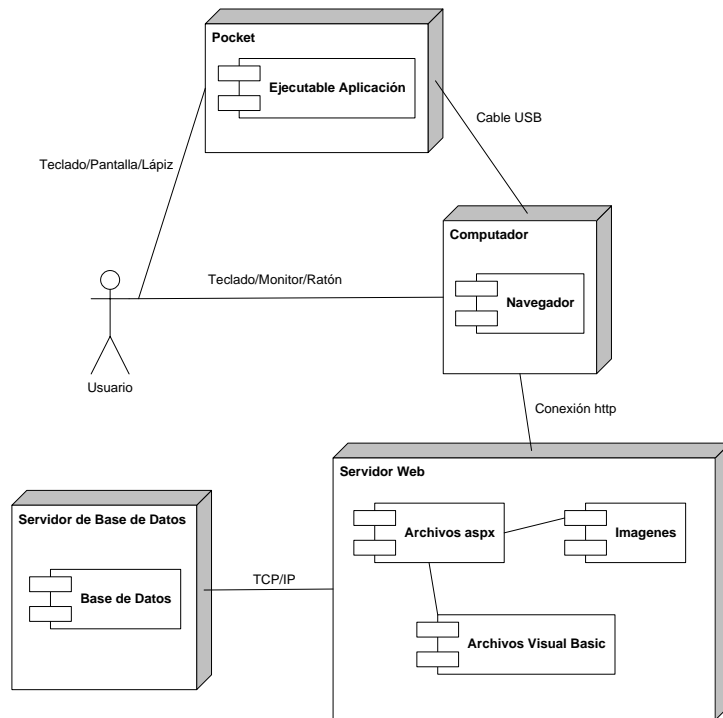


Figura 2.3: Diagrama de Despliegue del Sistema de Gestión Inmobiliaria

## 2.1.3 Diagramas de Estado

Se realizó un diagrama de Estado para cada caso de uso.

### Caso de Uso 1: Registrarse como Usuario

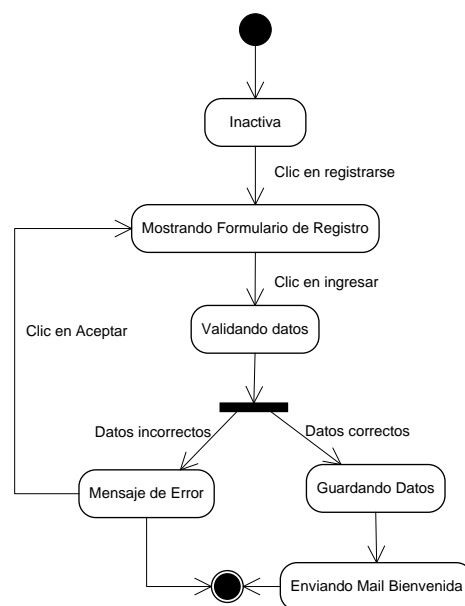


Figura 2.4: Diagrama de Estado del Caso de Uso 1

## Caso 2: Consultar Propiedades

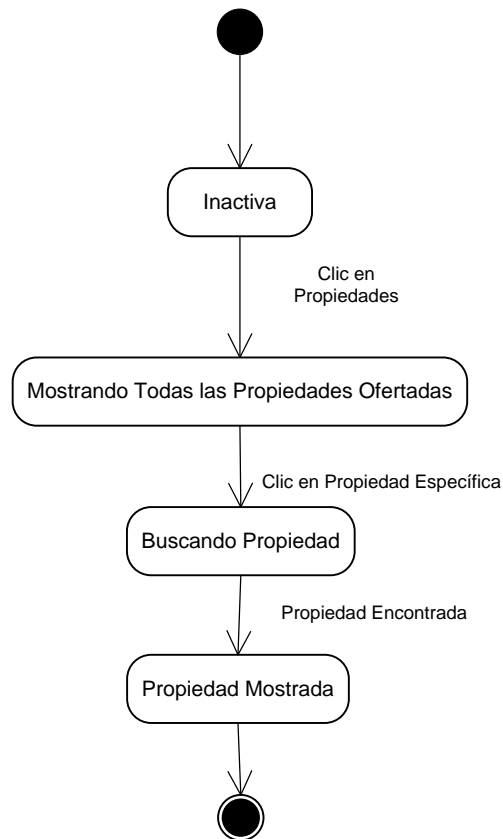


Figura 2.5: Diagrama de Estado del Caso de Uso 2

## Caso de Uso 3: Consultar Galería Fotográfica

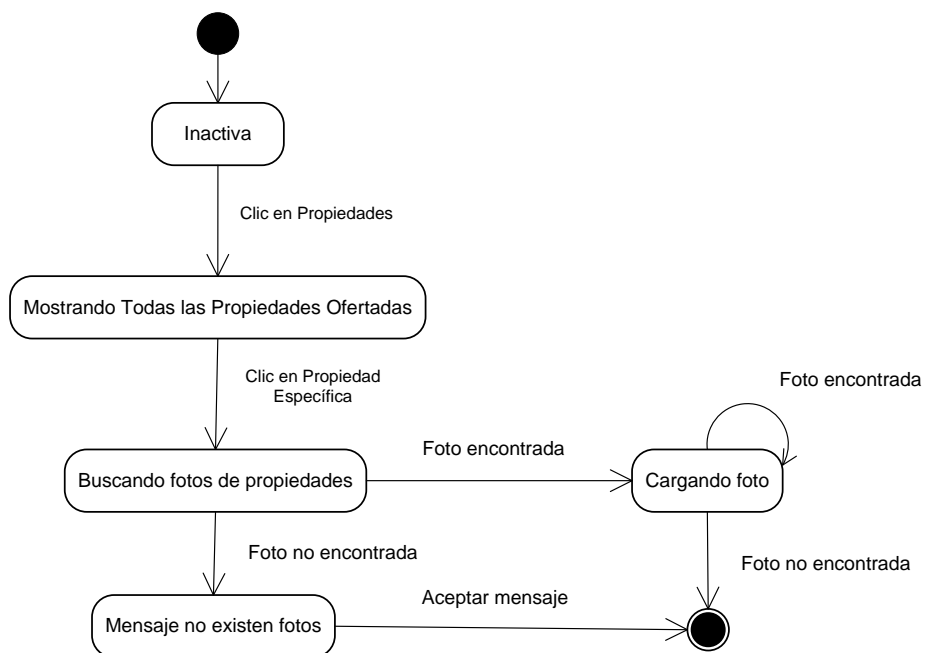


Figura 2.6: Diagrama de Estado del Caso de Uso 3

#### Caso 4: Ver Características

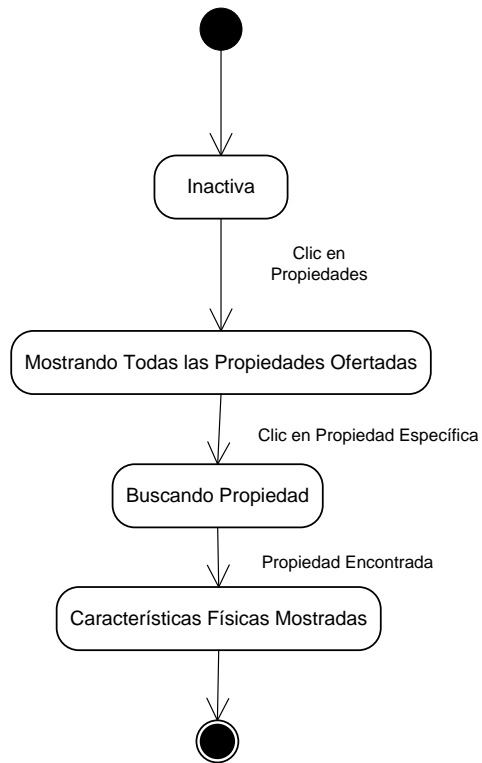


Figura 2.7: Diagrama de Estado del Caso de Uso 4

#### Caso 5: Consultar Ubicación Geográfica

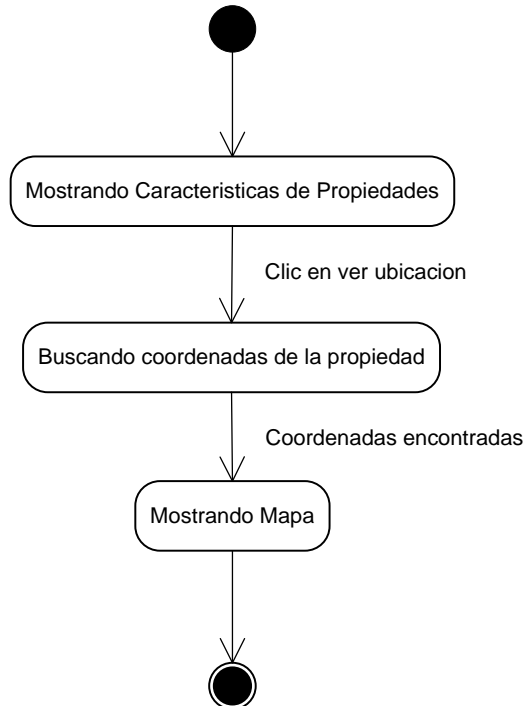


Figura 2.8: Diagrama de Estado del Caso de Uso 5

## Caso 6: Realizar Búsquedas

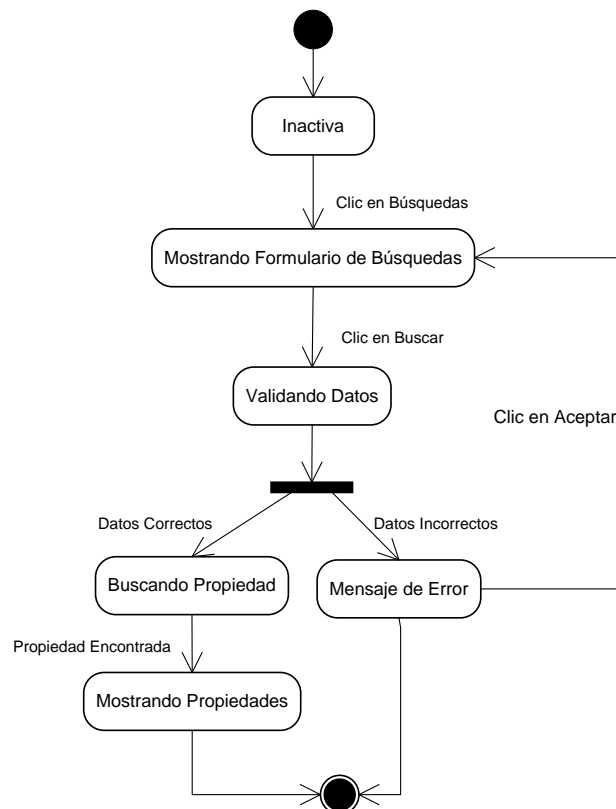


Figura 2.9: Diagrama de Estado del Caso de Uso 6

## Caso 7: Realizar Peticiones

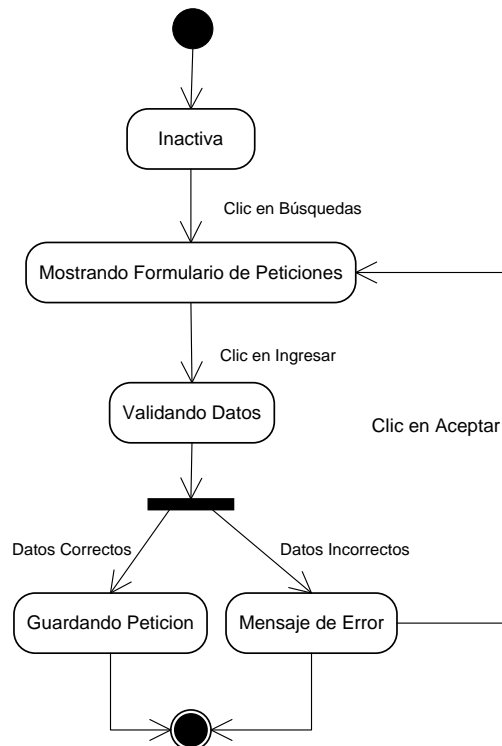


Figura 2.10: Diagrama de Estado del Caso de Uso 7

## Caso de Uso 8: Realizar Reservas

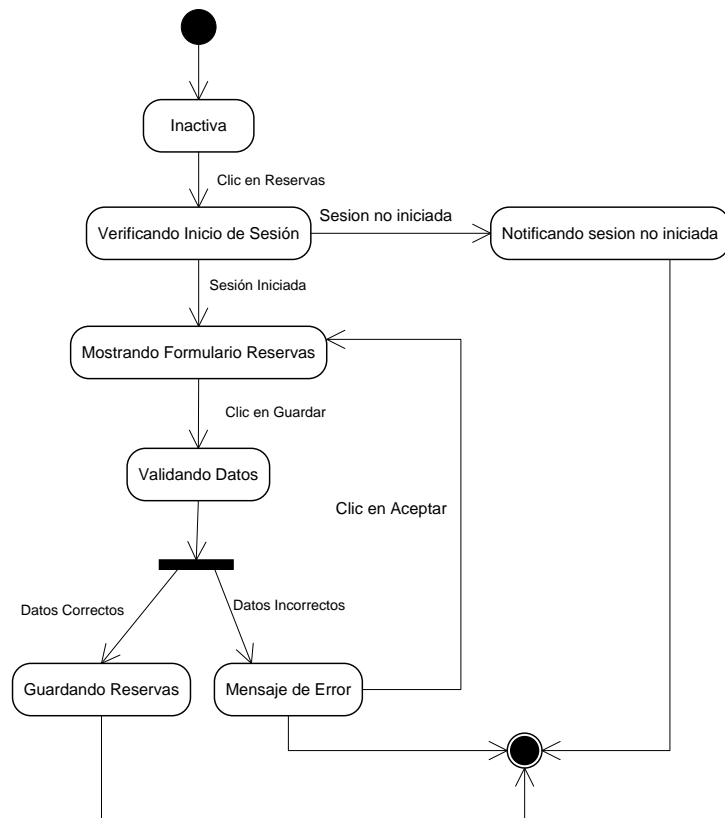


Figura 2.11: Diagrama de Estado del Caso de Uso 8

## Caso de Uso 9: Solicita Avalúos

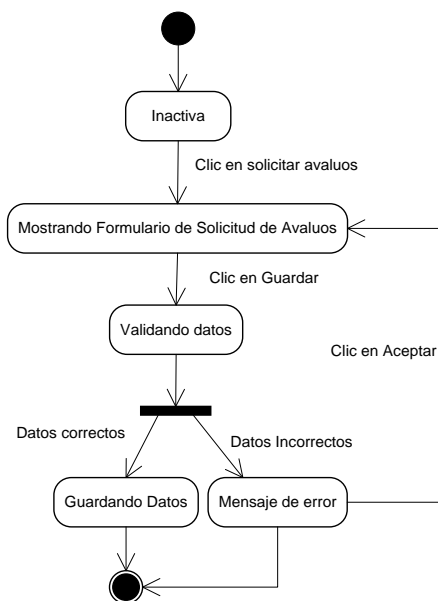


Figura 2.12: Diagrama de Estado del Caso de Uso 9

**Caso de Uso 10: Mantenimiento de Clientes**

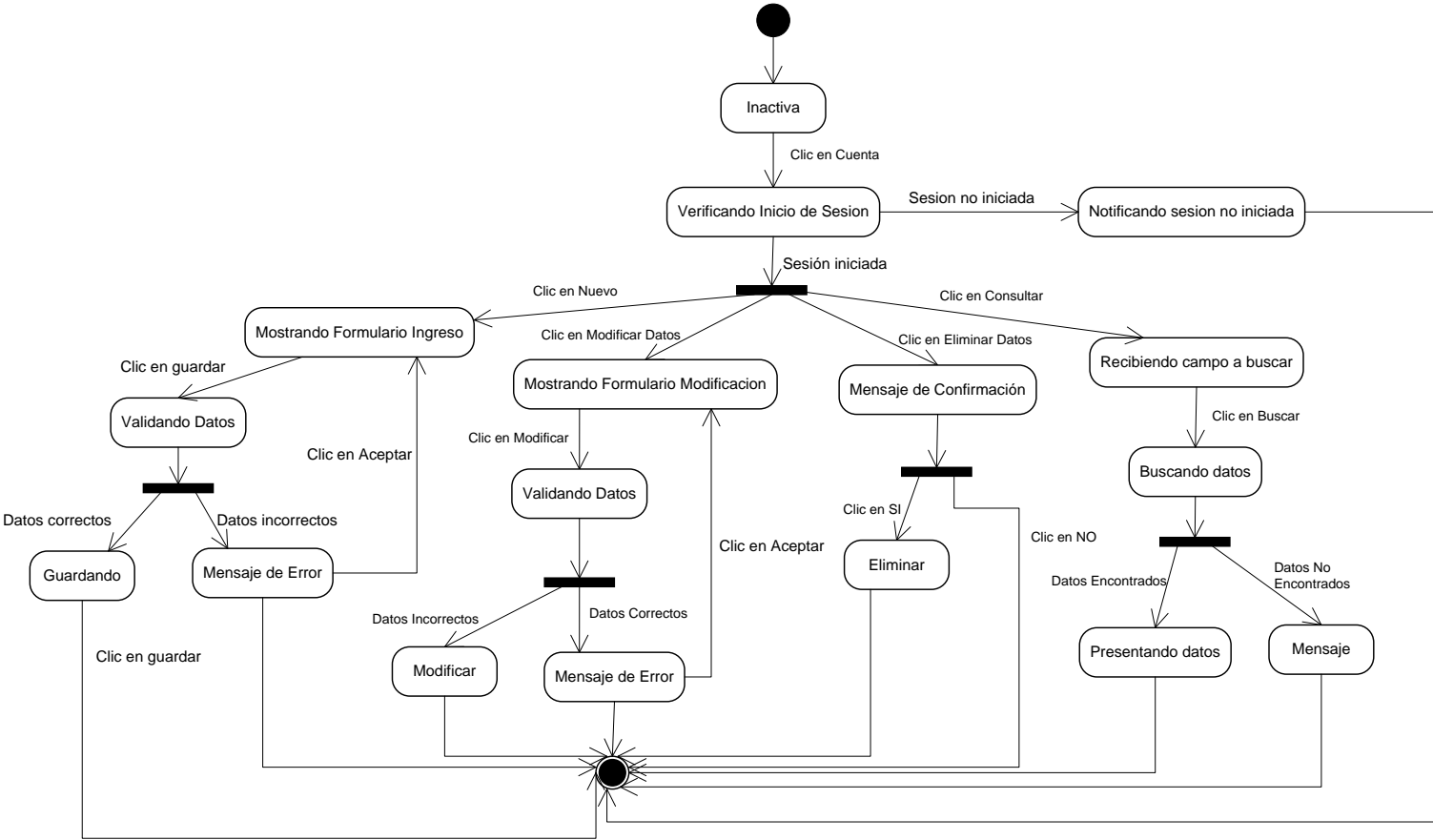


Figura 2.13: Diagrama de Estado del Caso de Uso 10

# Caso de Uso 11: Administrar Propiedades

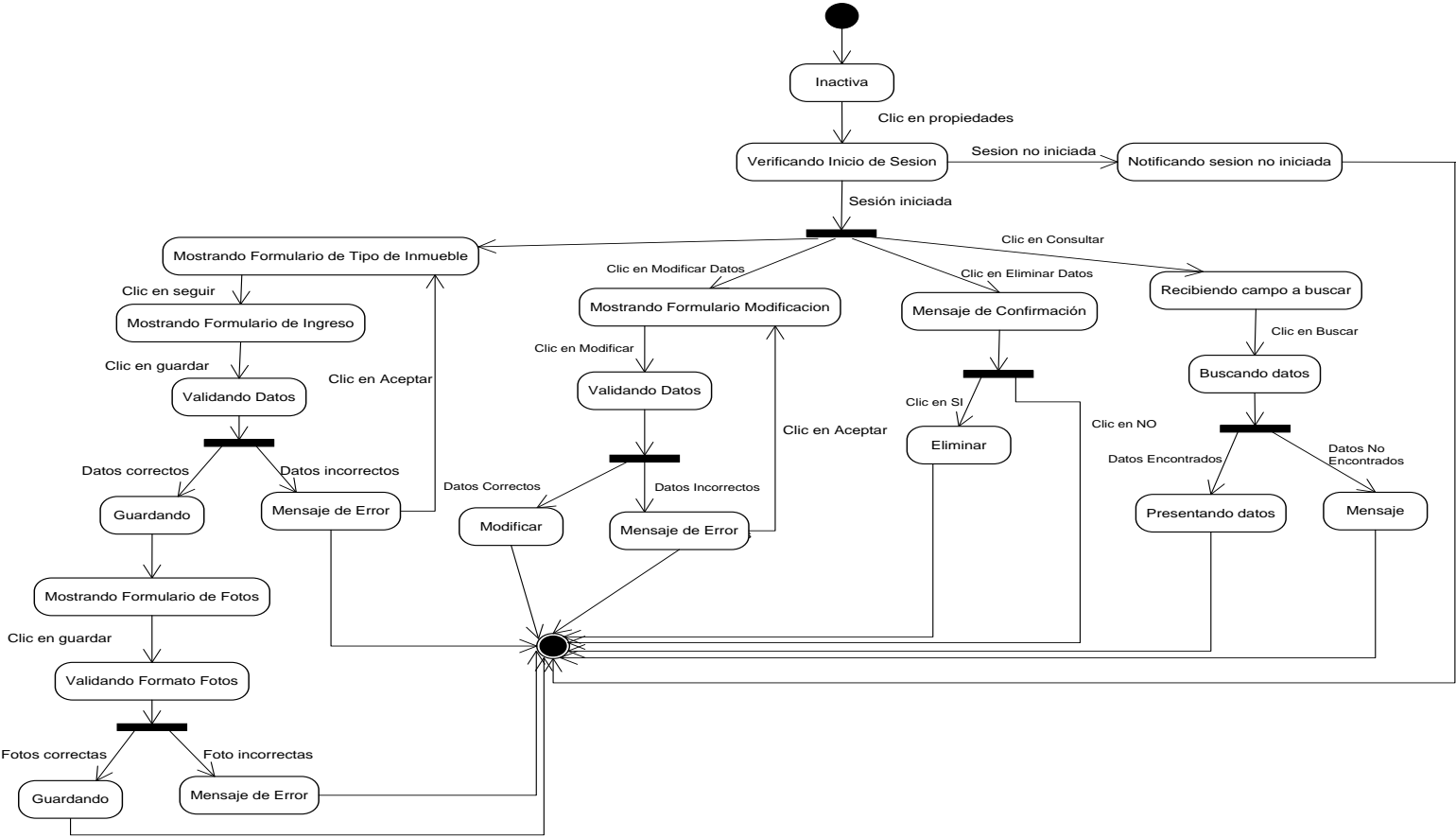


Figura 2.14: Diagrama de Estado del Caso de Uso 11

## Caso de Uso 12: Crear Reportes Gerenciales

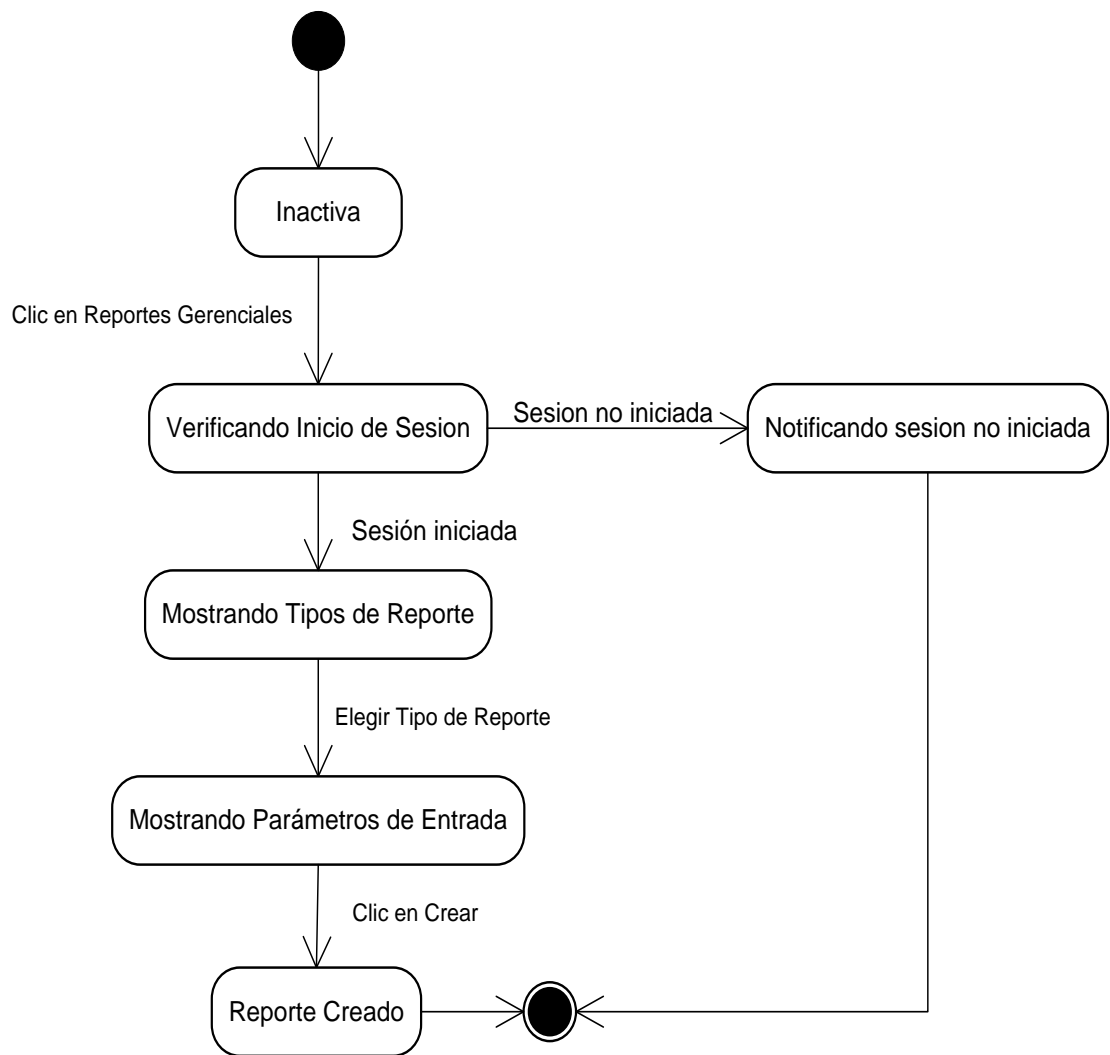


Figura 2.15: Diagrama de Estado del Caso de Uso 12



### Caso de Uso 13: Crear Reportes de Ventas

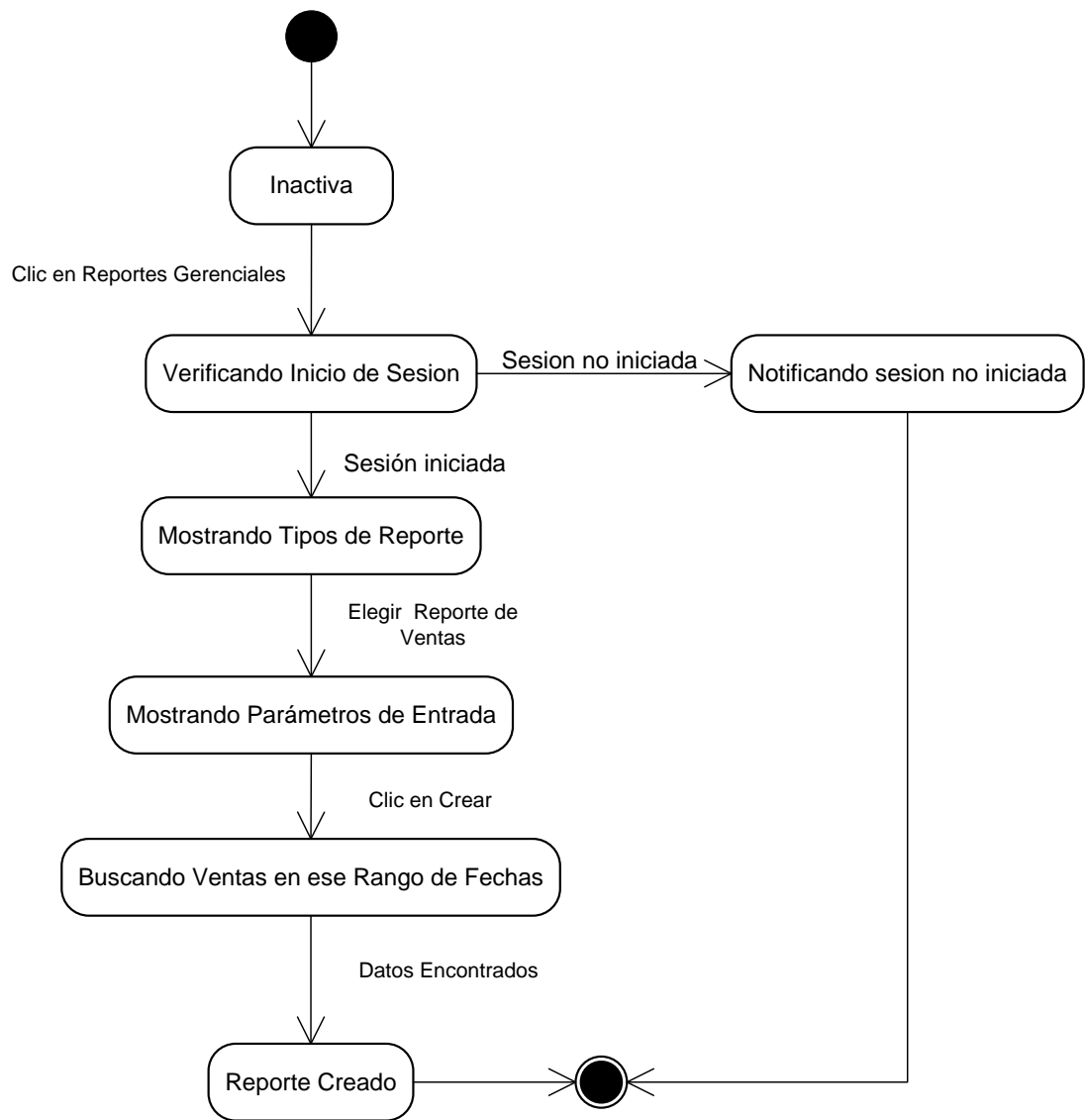


Figura 2.16: Diagrama de Estado del Caso de Uso 13

## Caso de Uso 14: Crear un Reporte con el tipo de Inmueble más Vendido

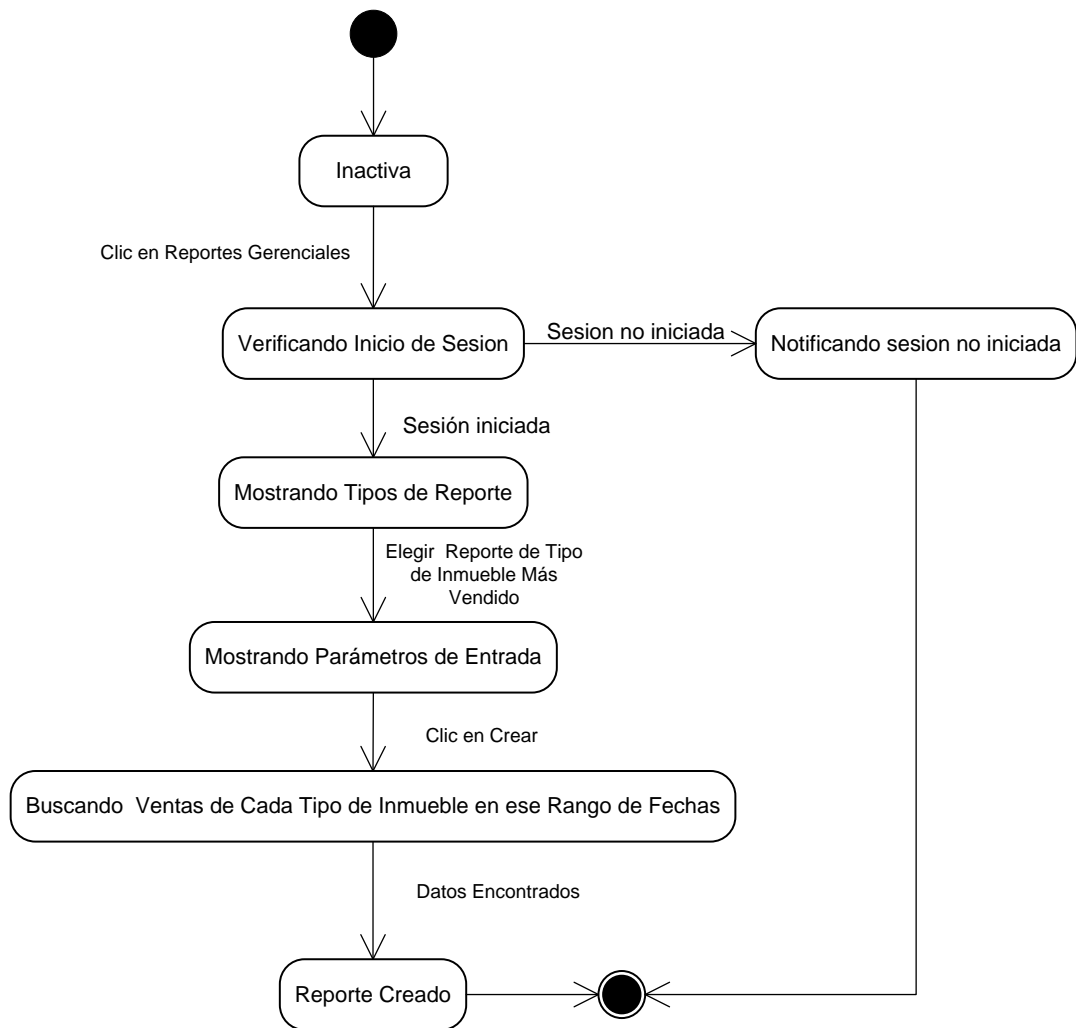


Figura 2.17: Diagrama de Estado del Caso de Uso 14

## Caso de Uso 15: Crear un Reporte del Vendedor más Exitoso

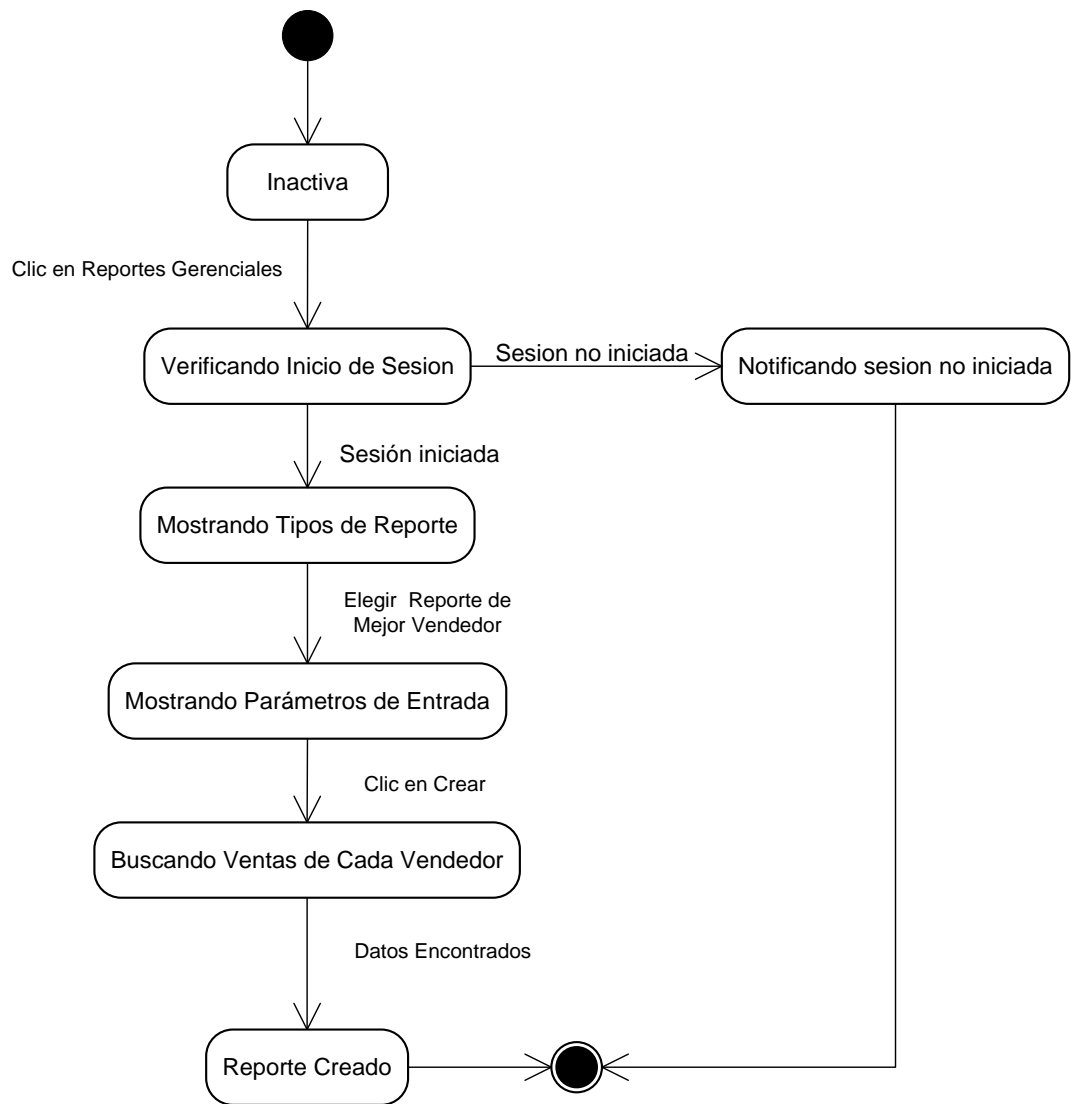


Figura 2.18: Diagrama de Estado del Caso de Uso 15

## Caso de Uso 16: Enviar Correos Electrónicos

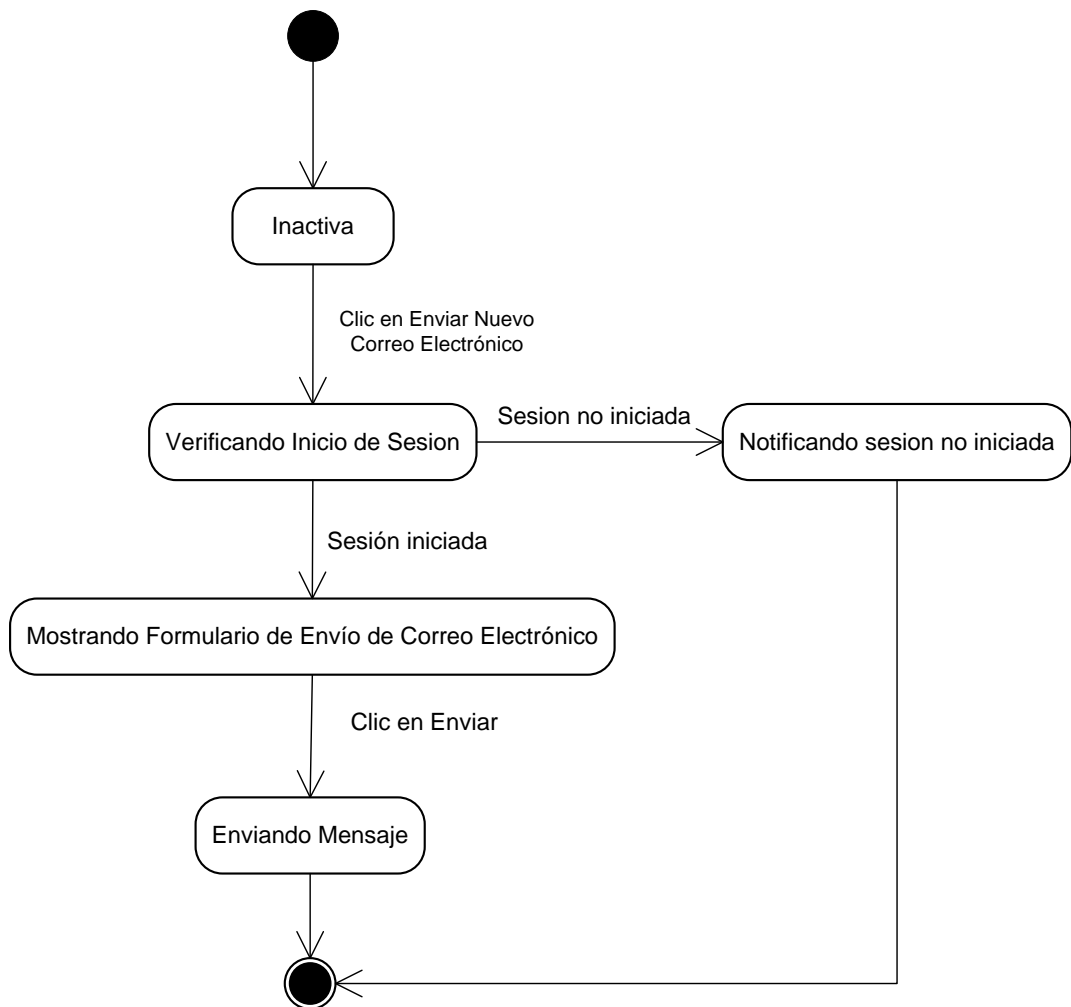


Figura 2.19: Diagrama de Estado del Caso de Uso 16

**Caso de Uso 17: Administrar Ventas**

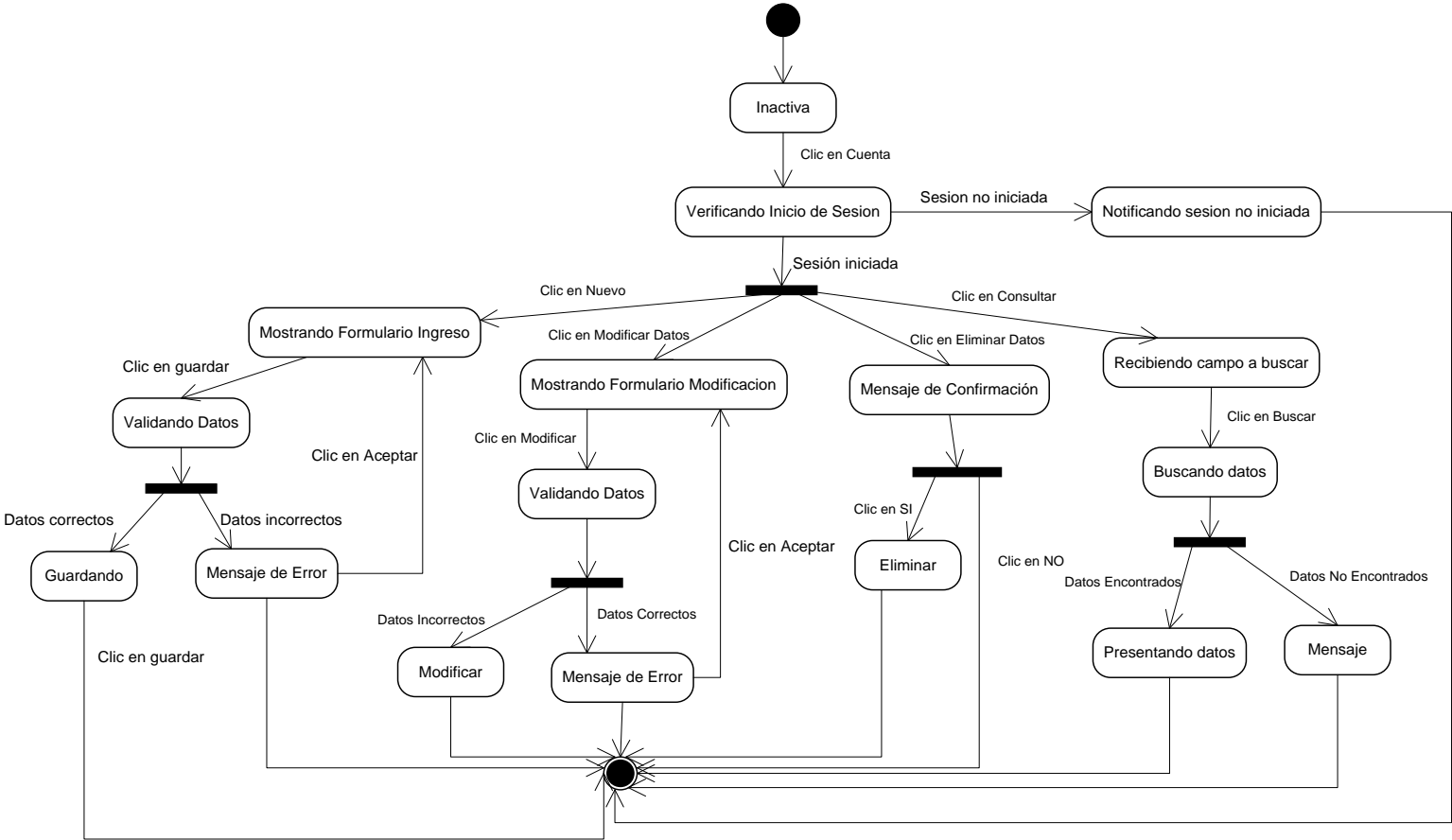


Figura 2.20: Diagrama de Estado del Caso de Uso 17

## Caso de Uso 18: Crear Archivos XML

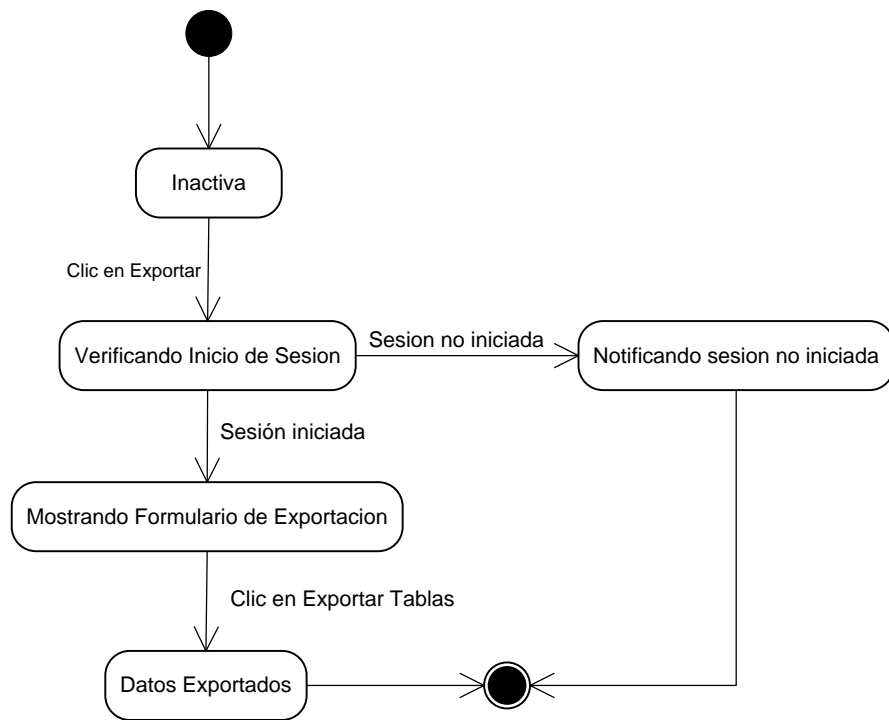


Figura 2.21: Diagrama de Estado del Caso de Uso 18

**Caso de Uso 19: Mantenimiento de Vendedores**

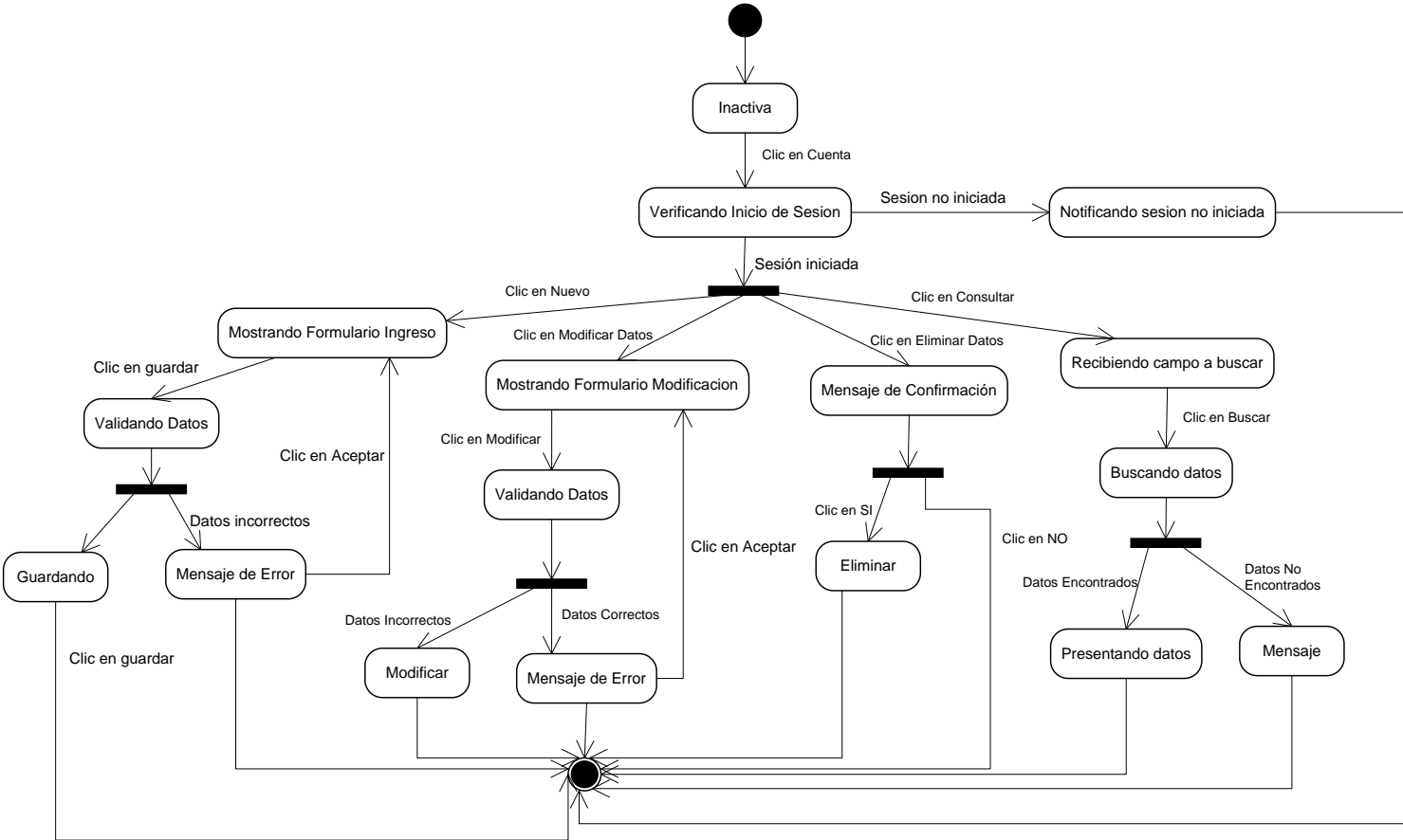


Figura 2.22: Diagrama de Estado del Caso de Uso 19

## Caso de Uso 20: Recoger Características

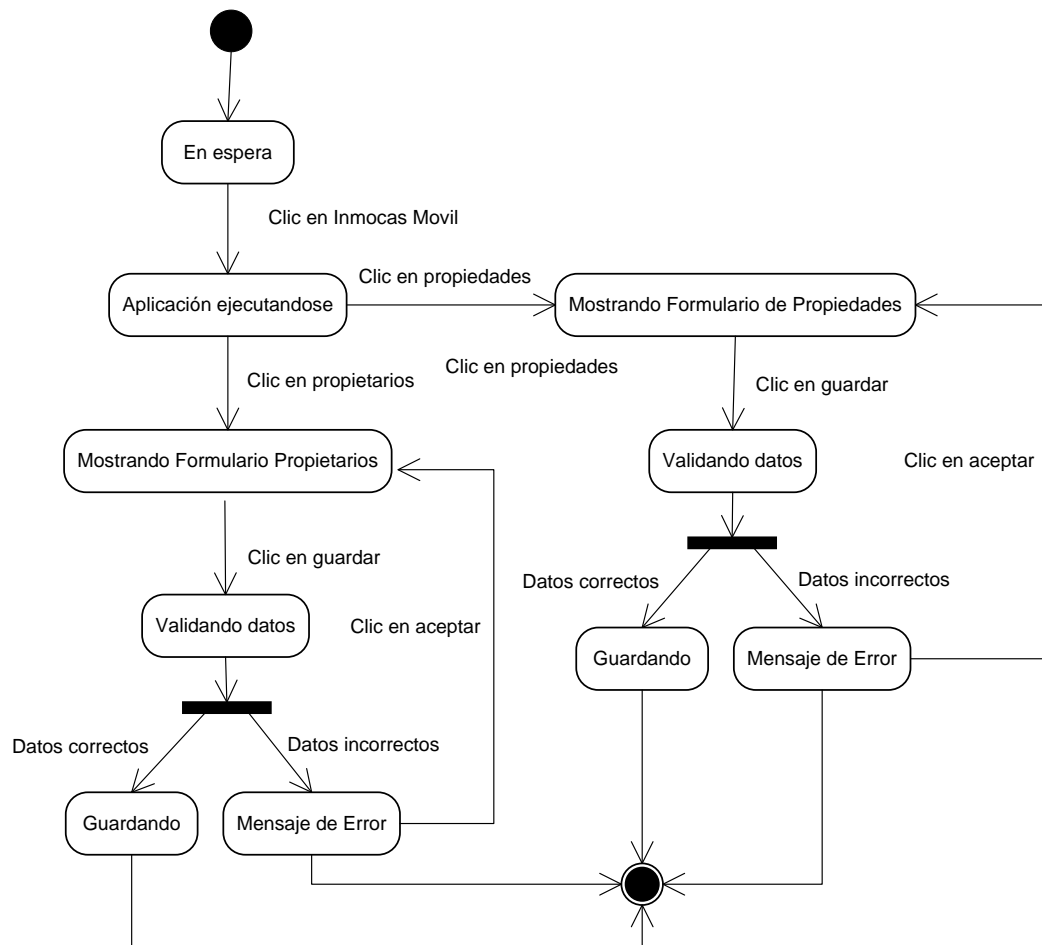


Figura 2.23: Diagrama de Estado del Caso de Uso 20



## Caso de Uso 21: Descargar Características

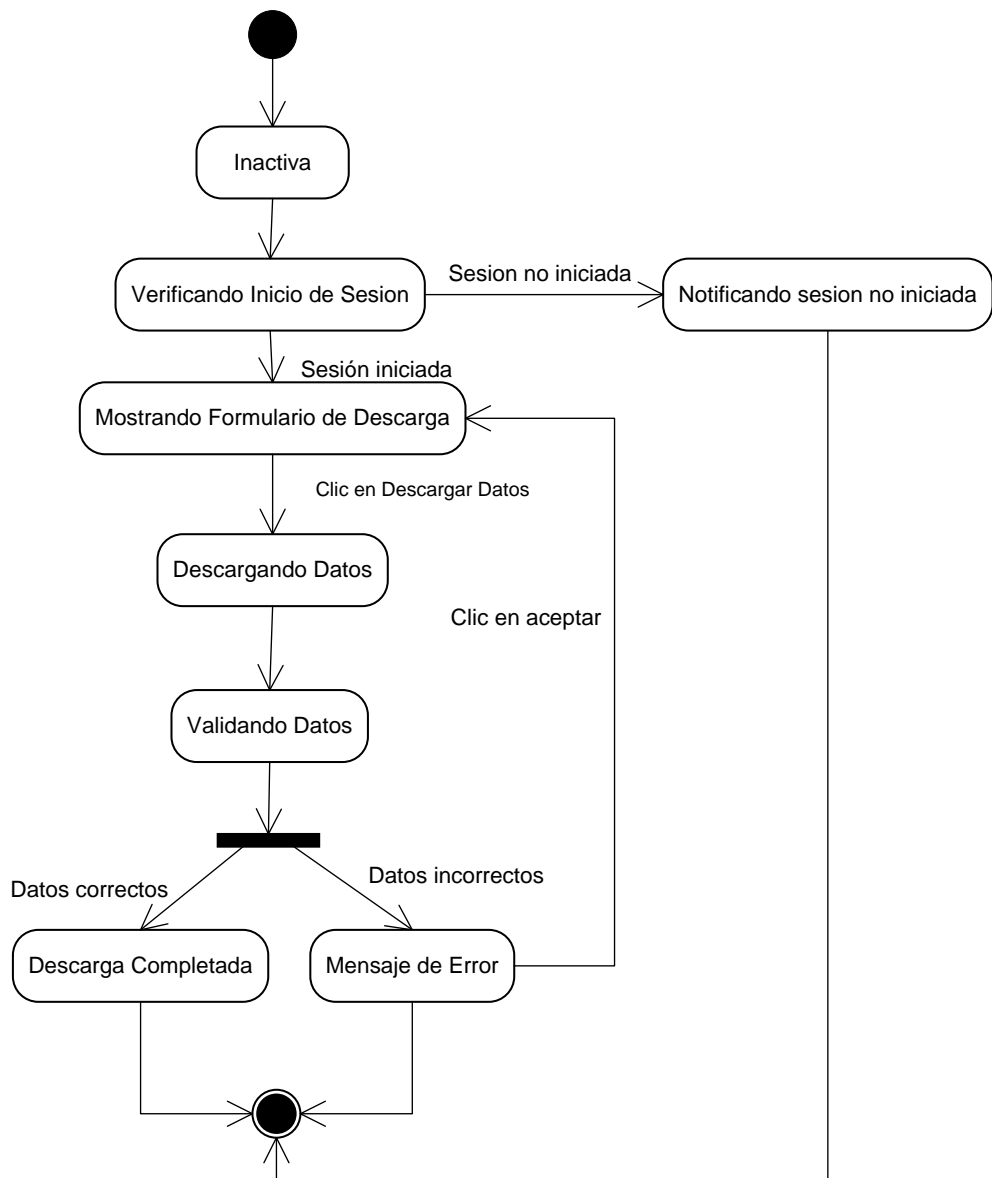


Figura 2.24: Diagrama de Estado del Caso de Uso 21

## Caso de Uso 22: Consultar Horarios

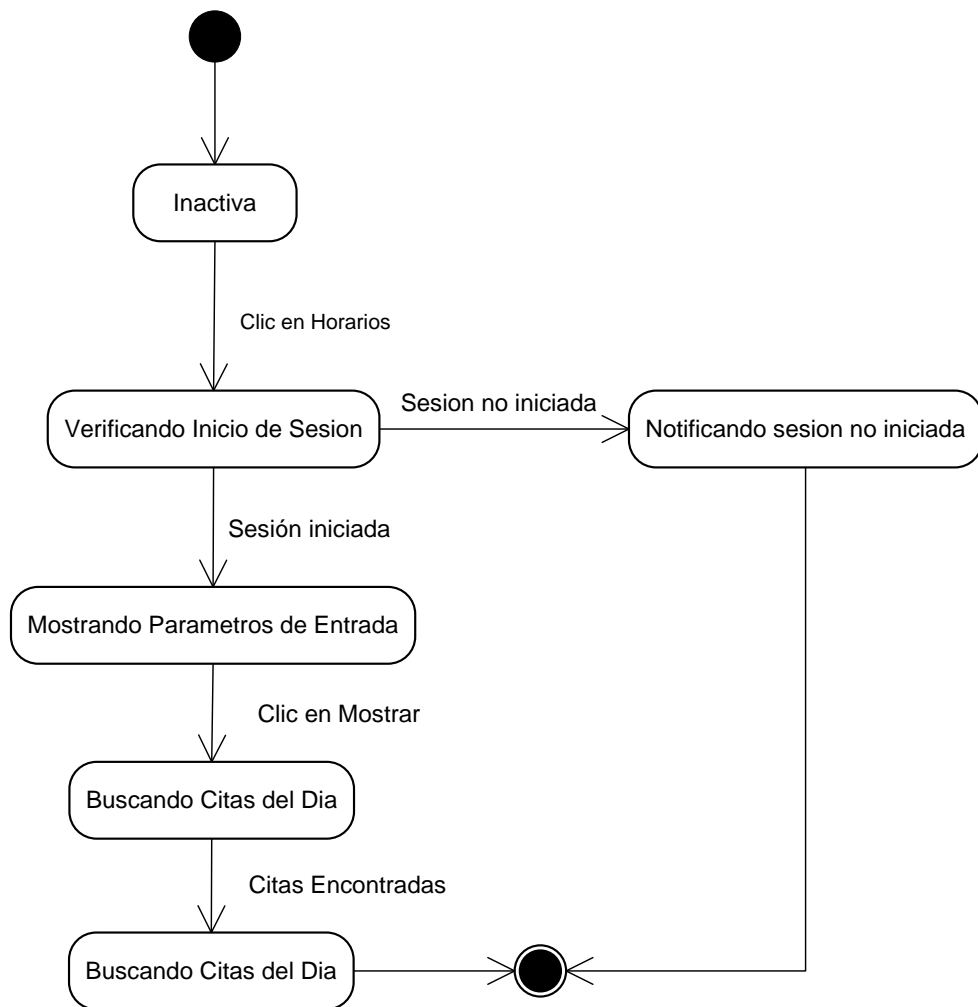


Figura 2.25: Diagrama de Estado del Caso de Uso 22

### 2.1.4 Diagramas de Secuencia

Se realizó un diagrama de Secuencia para cada caso de uso.

## Caso de Uso 1: Registrarse como Usuario

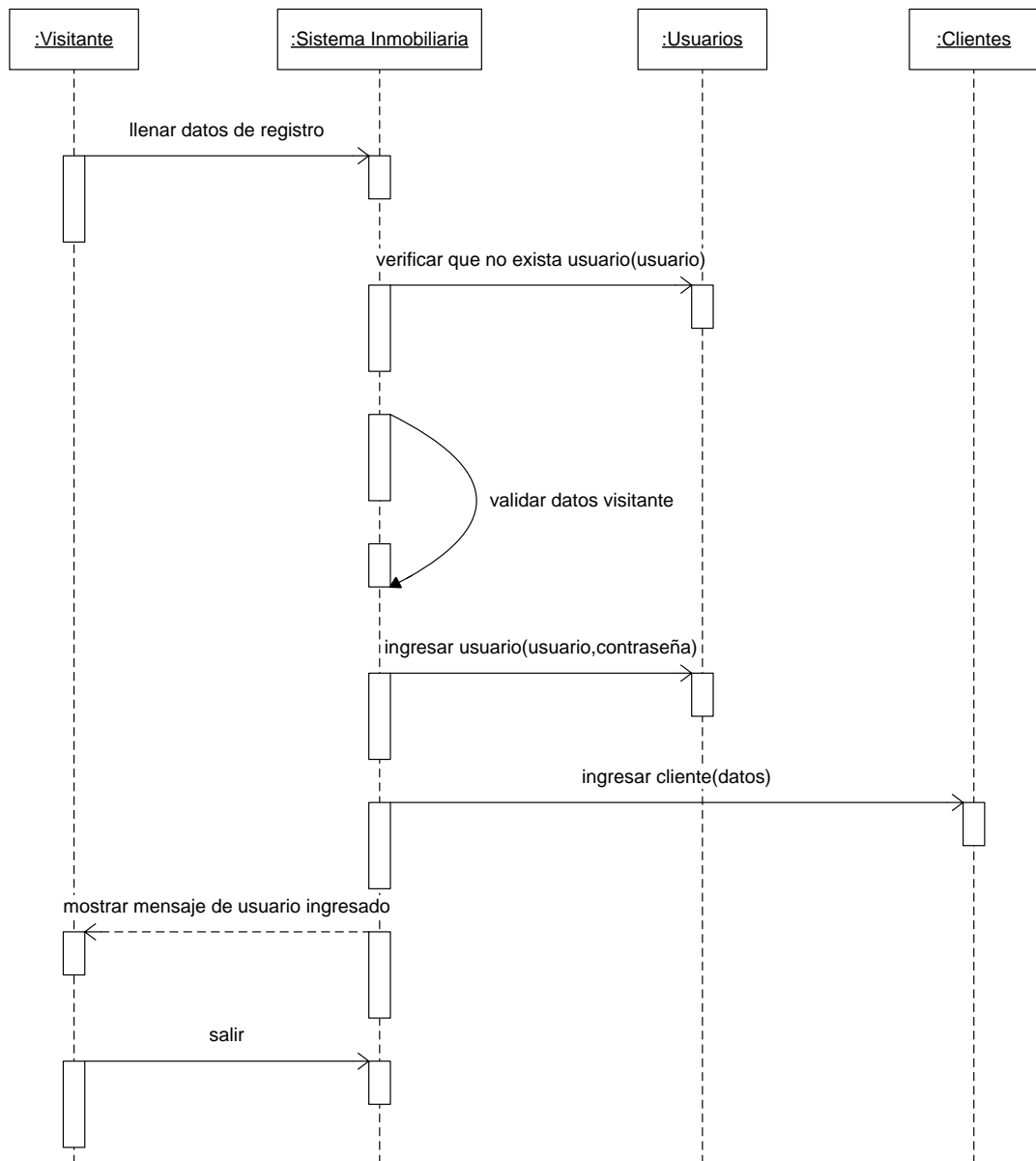


Figura 2.26: Diagrama de Estado del Caso de Uso 1

## Caso 2: Consultar Propiedades

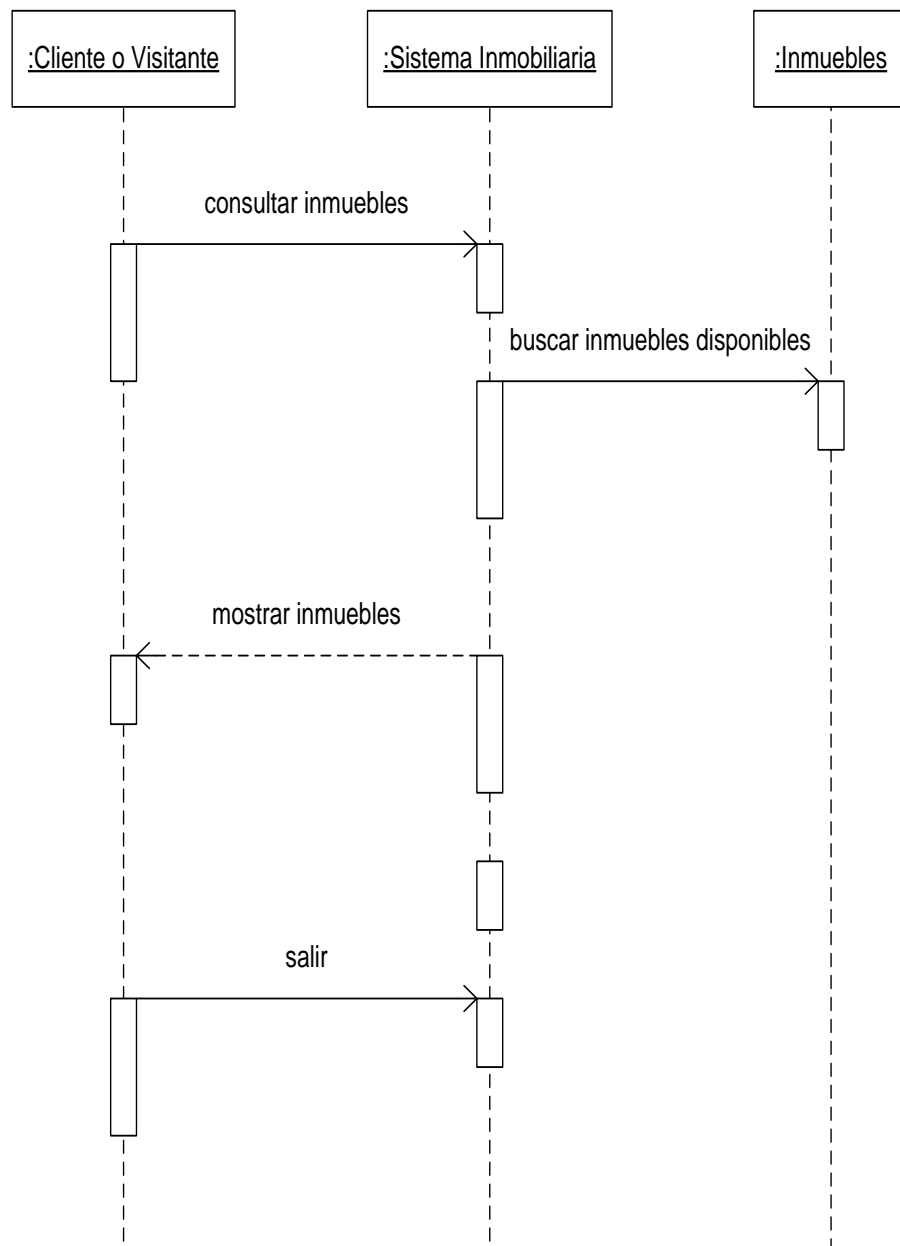


Figura 2.27: Diagrama de Estado del Caso de Uso 2

### Caso de Uso 3: Consultar Galería Fotográfica

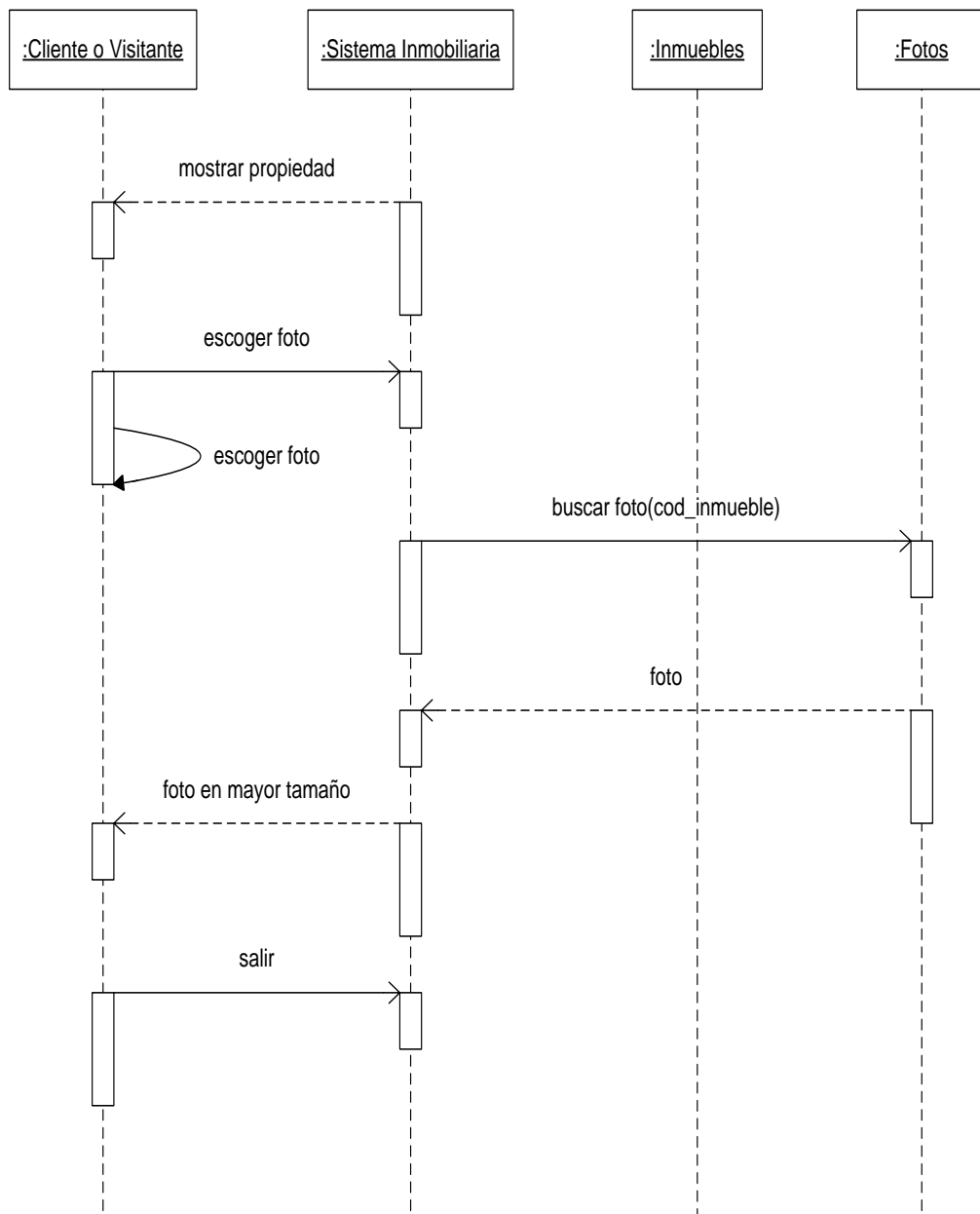


Figura 2.28: Diagrama de Estado del Caso de Uso 3

#### Caso 4: Ver Características

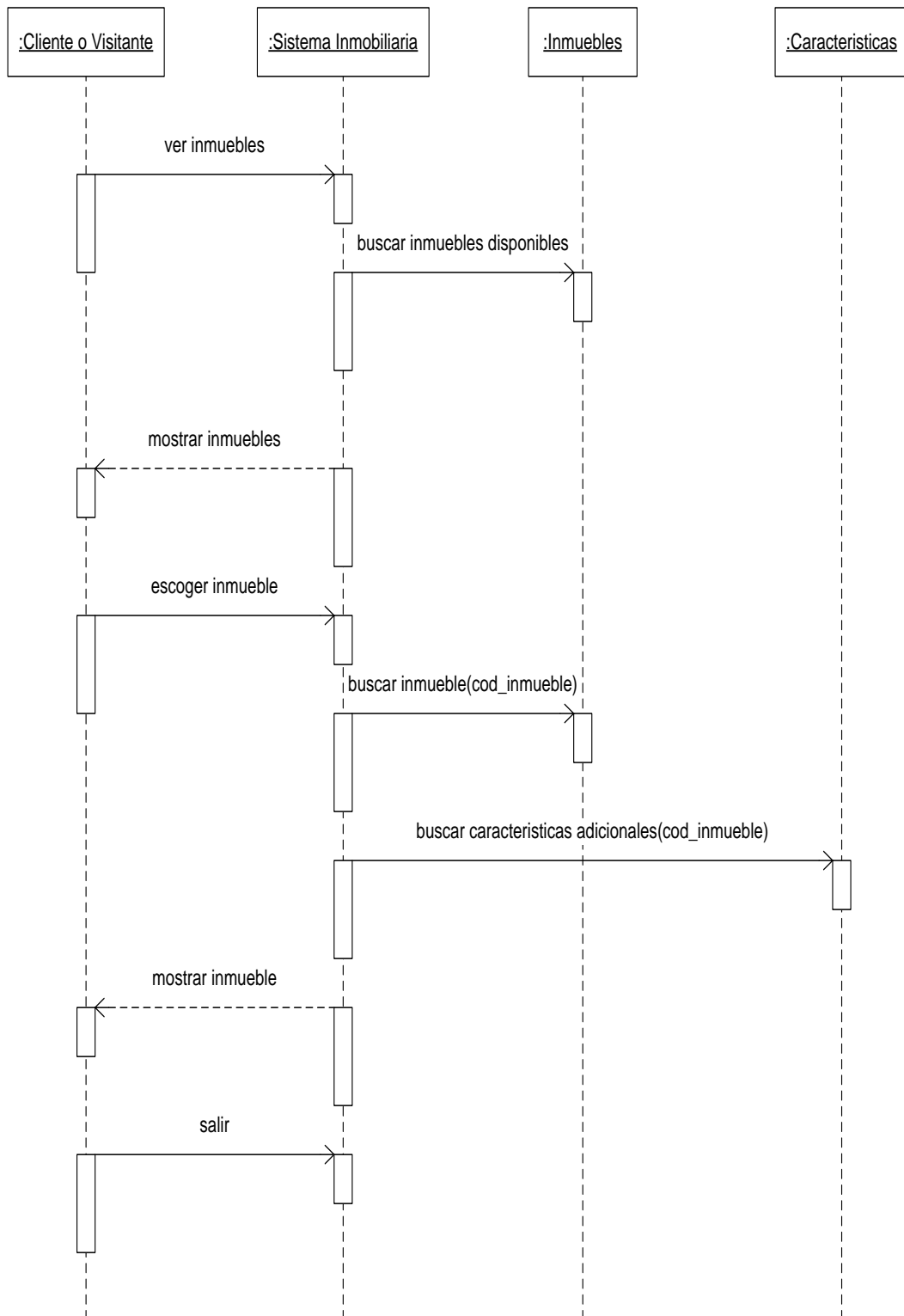


Figura 2.29: Diagrama de Estado del Caso de Uso 4

## Caso 5: Consultar Ubicación Geográfica

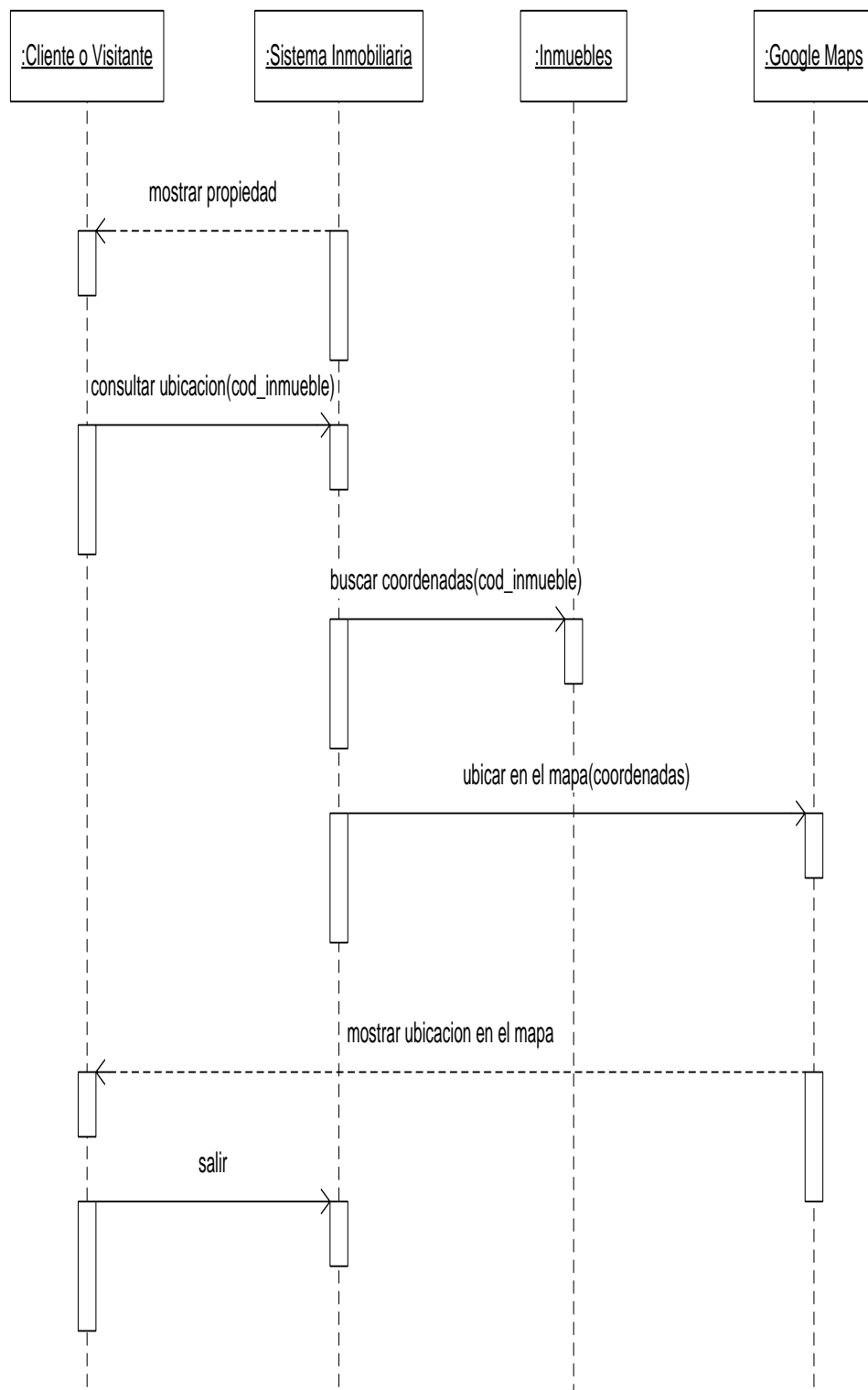


Figura 2.30: Diagrama de Estado del Caso de Uso 5

## Caso 6: Realizar Búsquedas

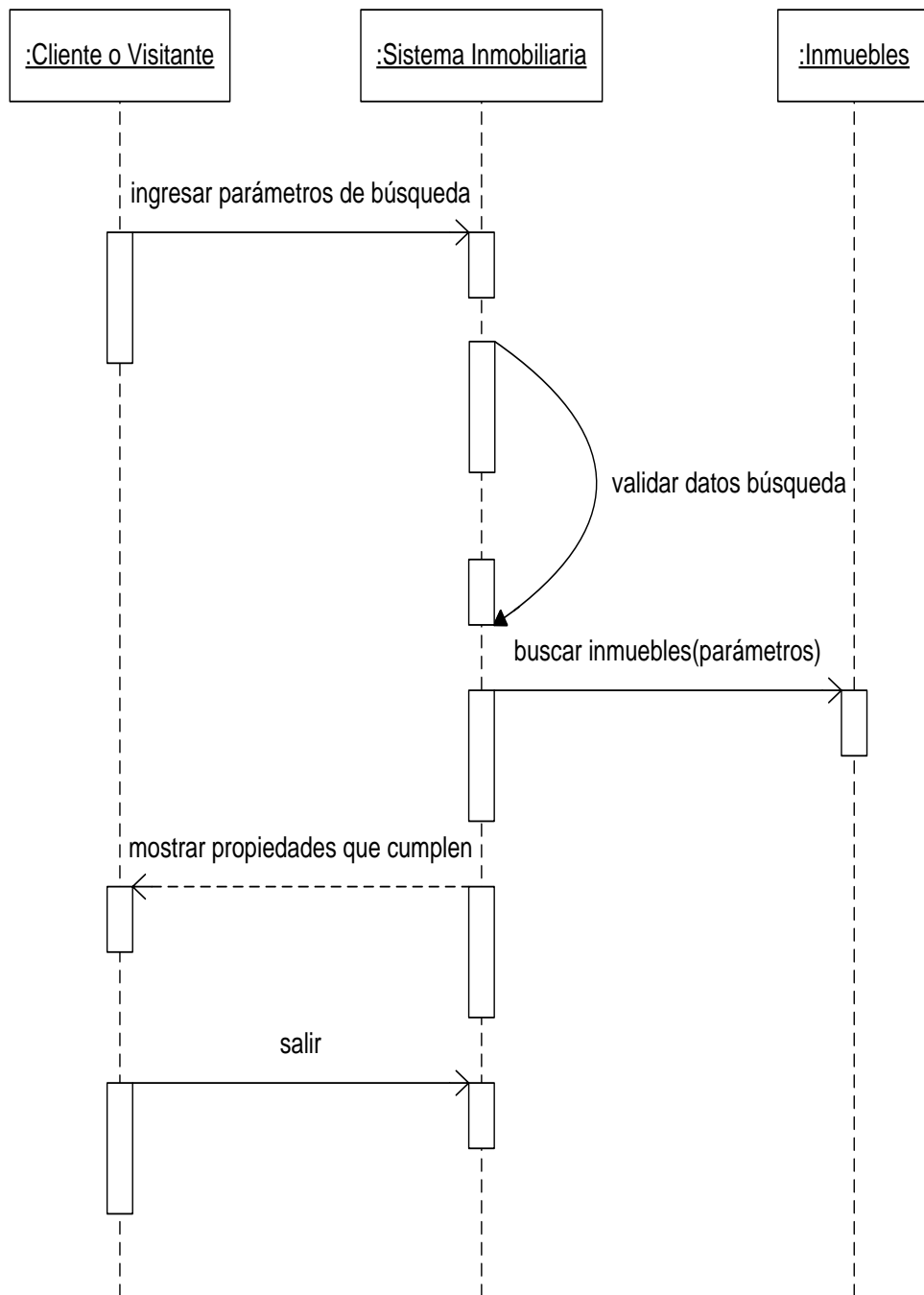


Figura 2.31: Diagrama de Estado del Caso de Uso 6



## Caso 7: Realizar Peticiones

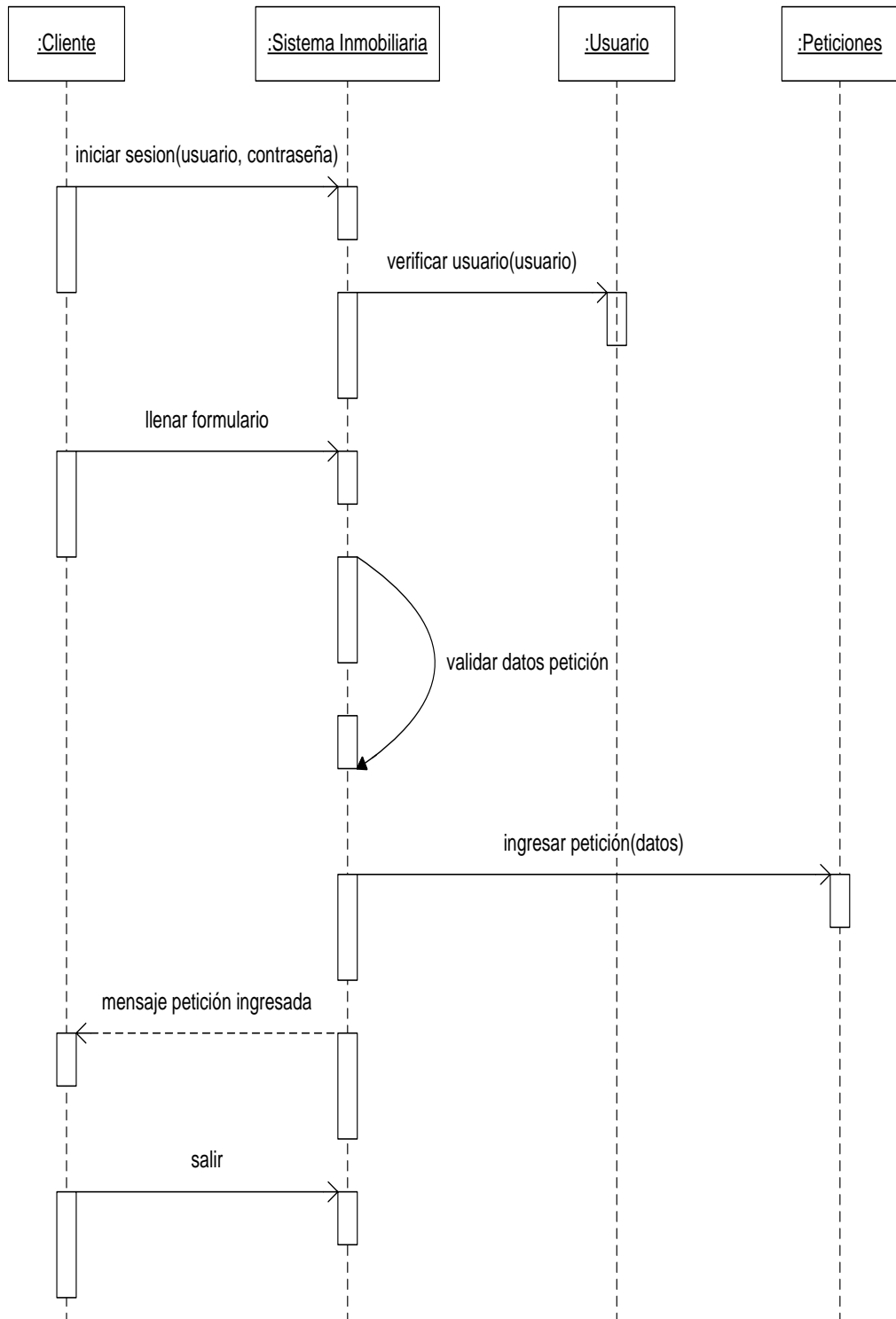


Figura 2.32: Diagrama de Estado del Caso de Uso 7

## Caso de Uso 8: Realizar Reservas

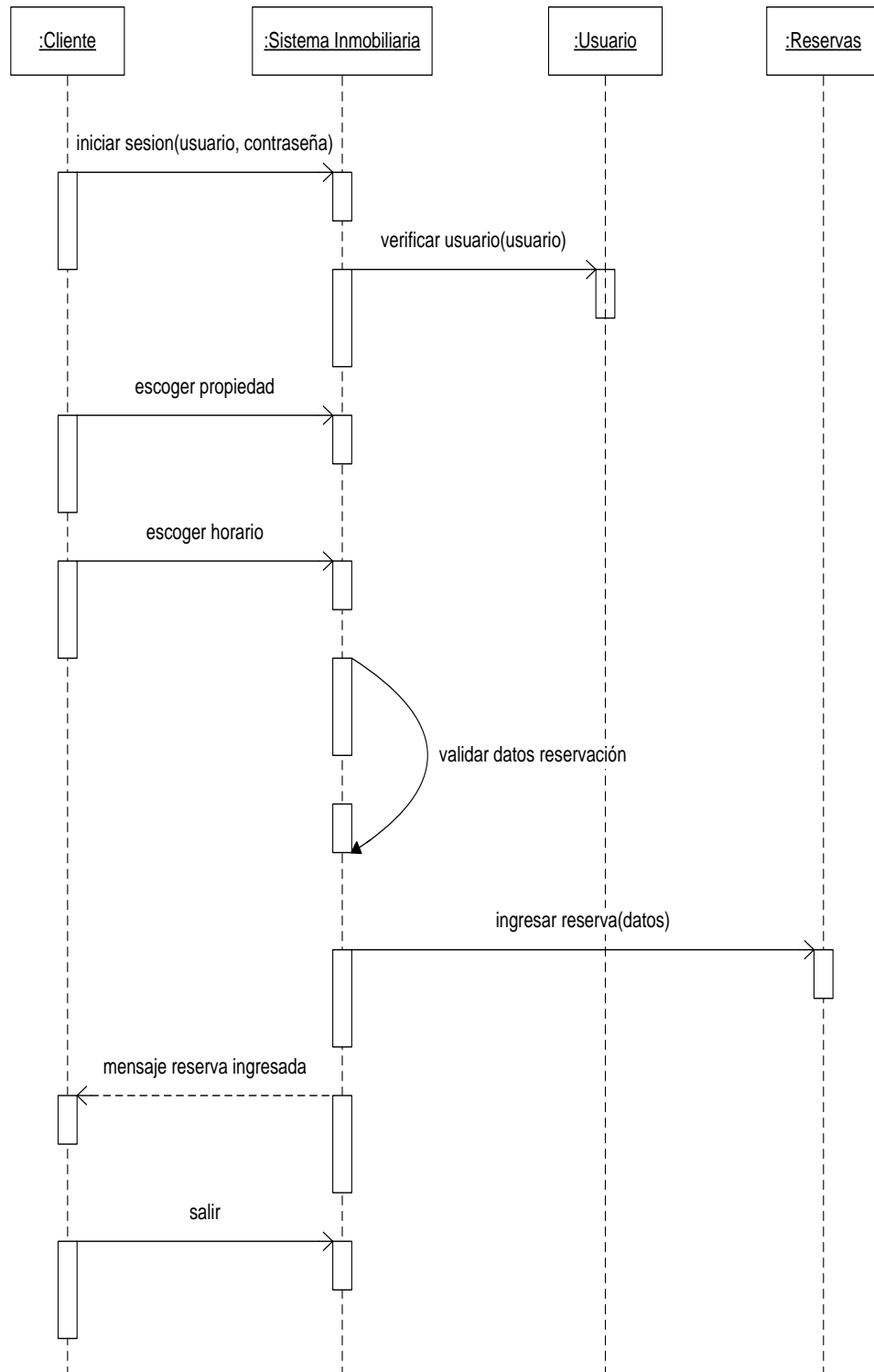


Figura 2.33: Diagrama de Estado del Caso de Uso 8

## Caso de Uso 9: Solicita Avalúos

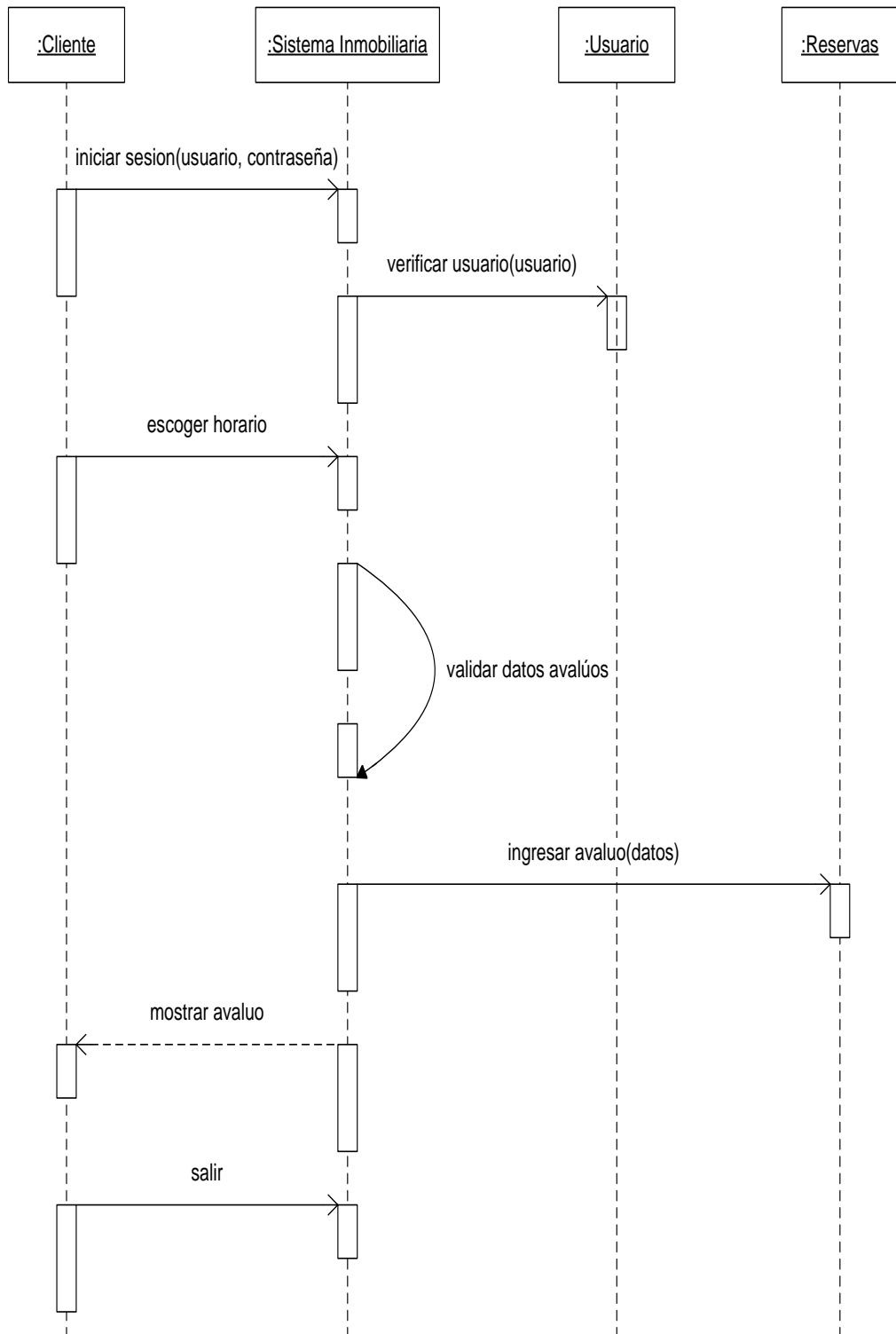


Figura 2.34: Diagrama de Estado del Caso de Uso 9

## Caso de Uso 10: Mantenimiento de Clientes

Para este Caso de Uso se precisó hacer 4 diagramas de Secuencia: Ingresar, Modificar, Eliminar y Consultar.

### Ingresar Cliente

En este diagrama de secuencia tenemos como actor al Administrador y no al Cliente porque en el Caso de Uso 1: Registrarse como Usuario tenemos al Cliente como Actor.

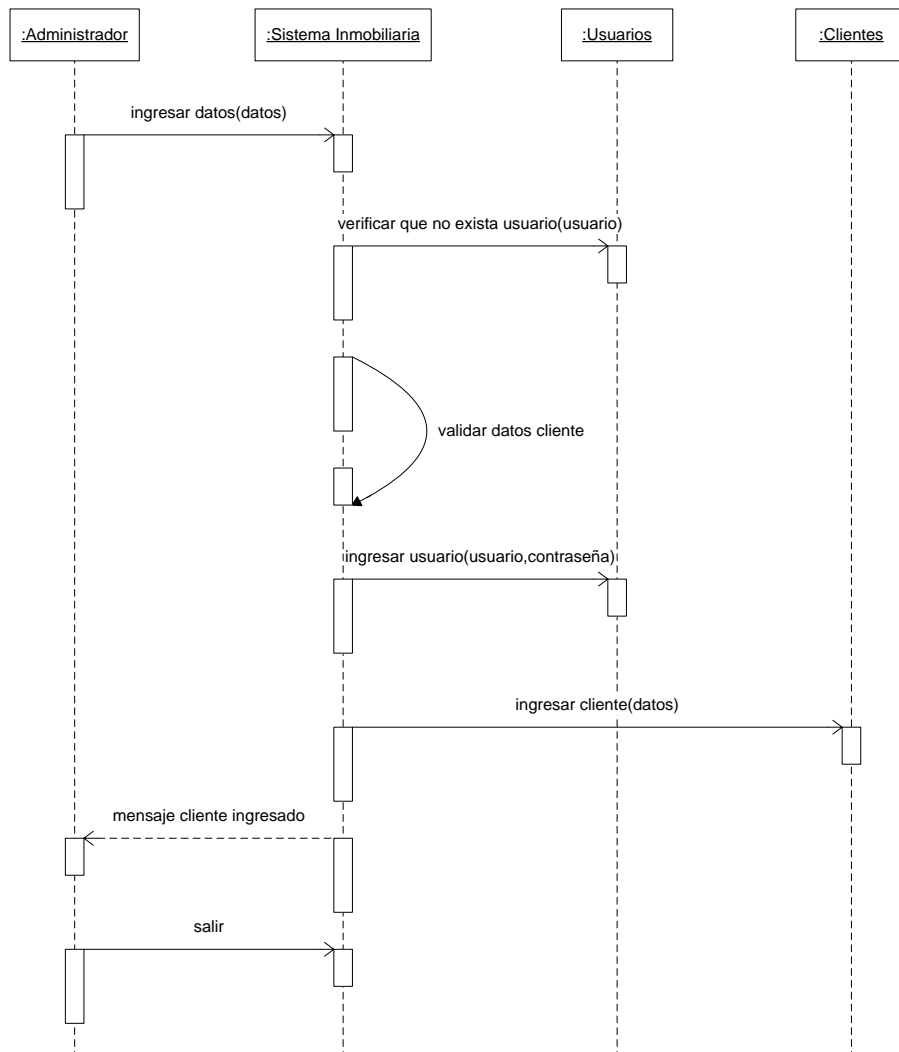


Figura 2.35: Diagrama de Estado del Caso de Uso 10 Ingreso

## Modificar Cliente

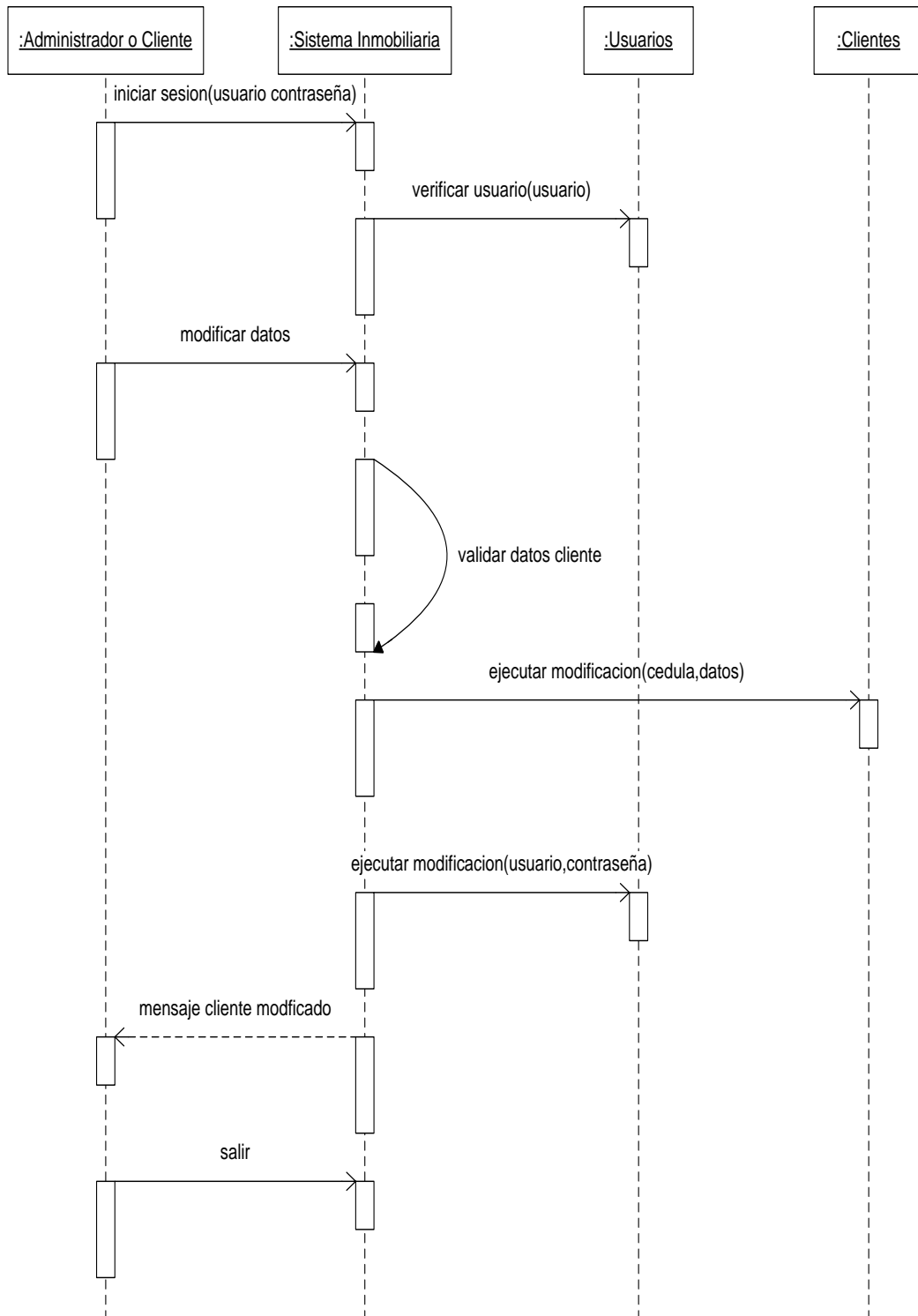


Figura 2.36: Diagrama de Estado del Caso de Uso 10 Modificar

## Eliminar Cliente

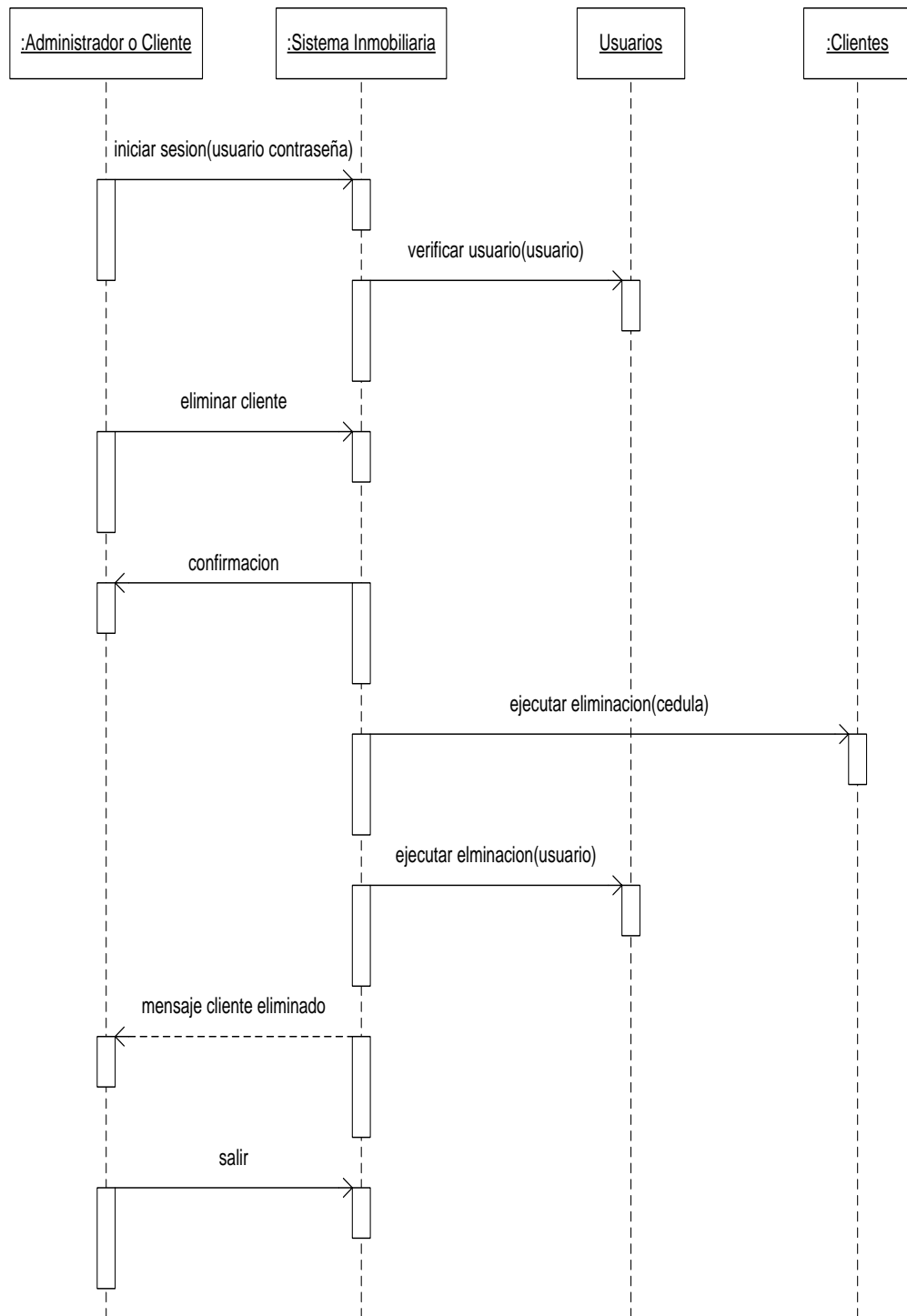


Figura 2.37: Diagrama de Estado del Caso de Uso 10 Eliminar

## Consultar Cliente

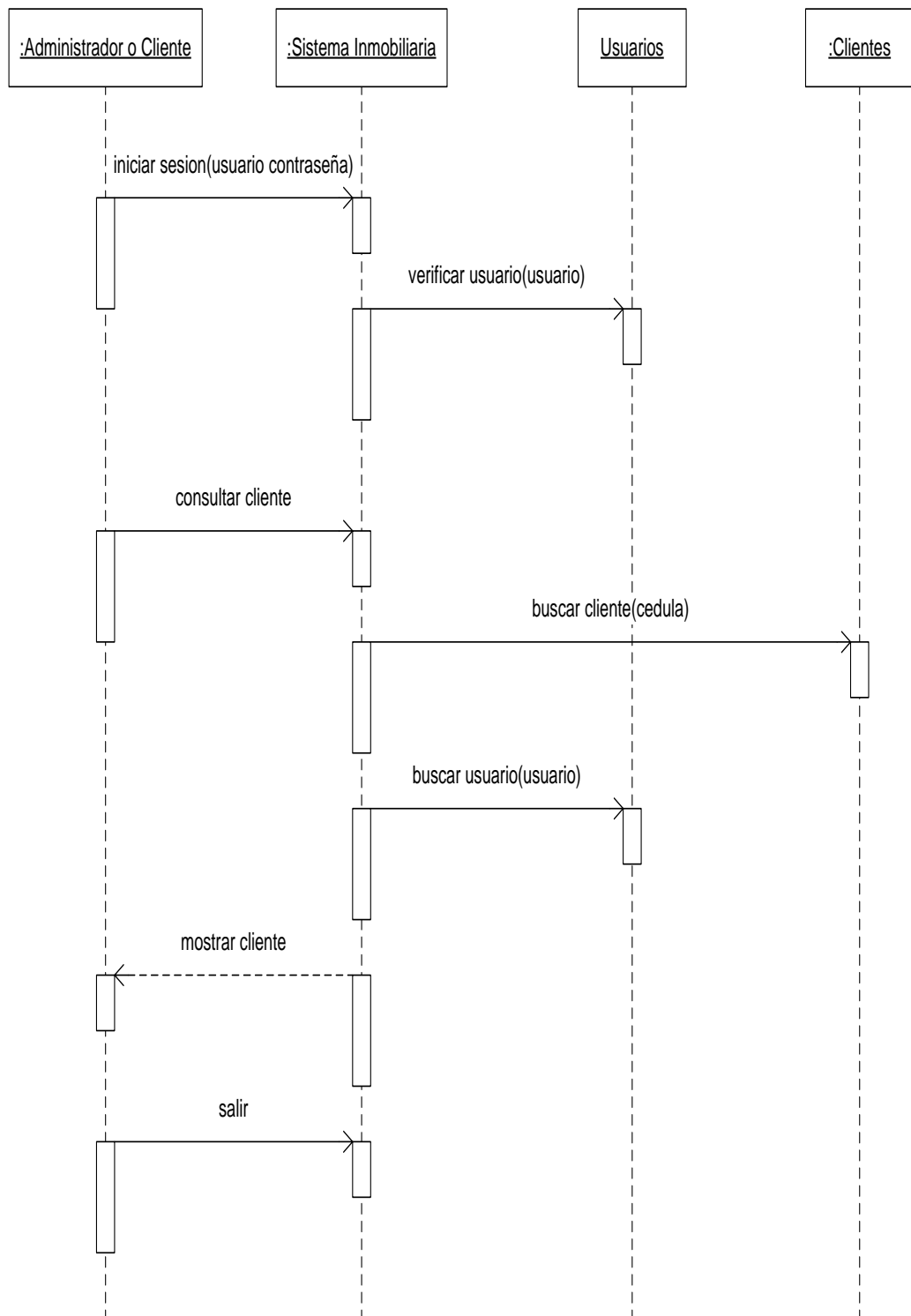


Figura 2.38: Diagrama de Estado del Caso de Uso 10 Consultar

## Caso de Uso 11: Administrar Propiedades

### Ingresar

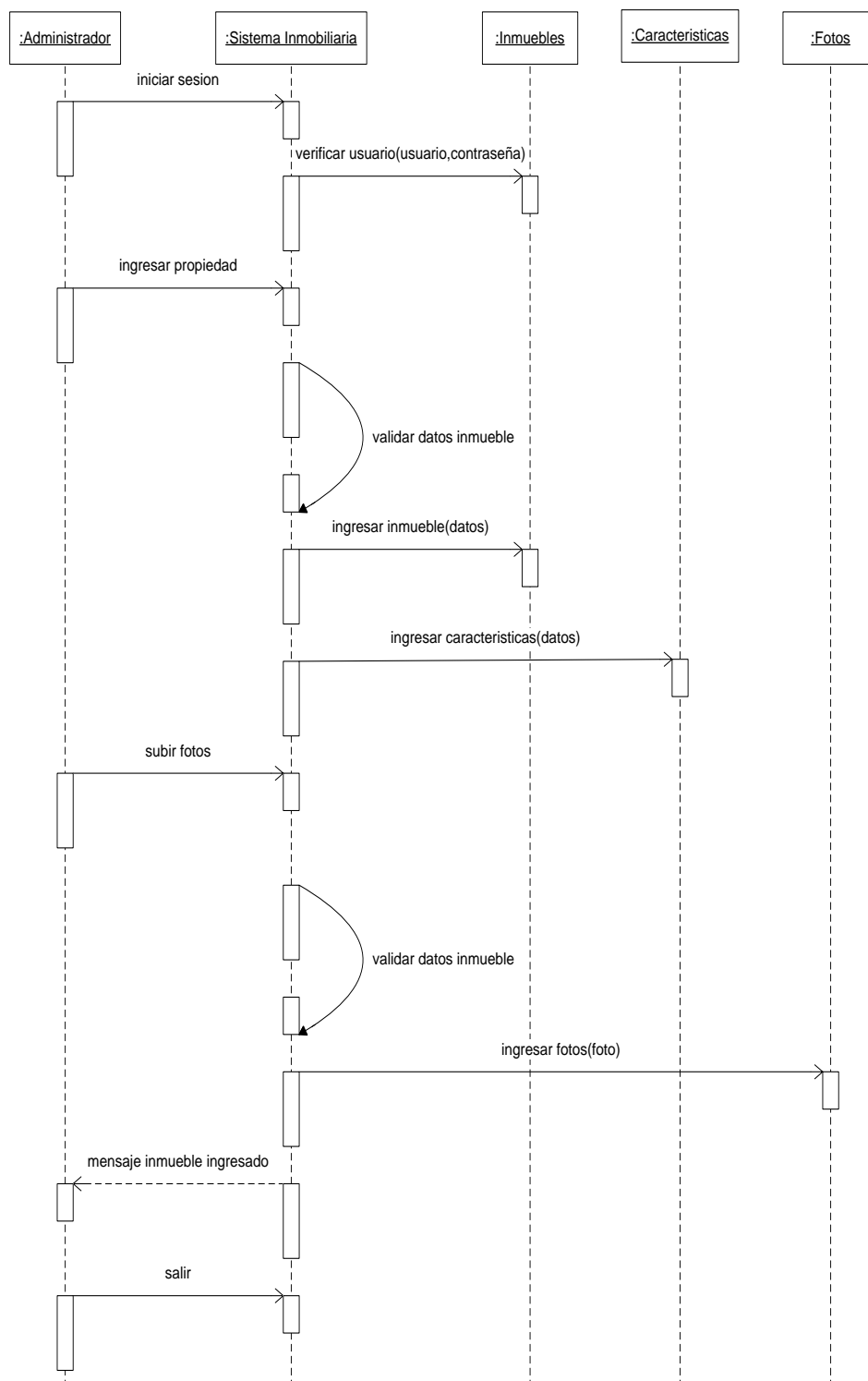


Figura 2.39: Diagrama de Estado del Caso de Uso 11 Ingresar



## Modificar

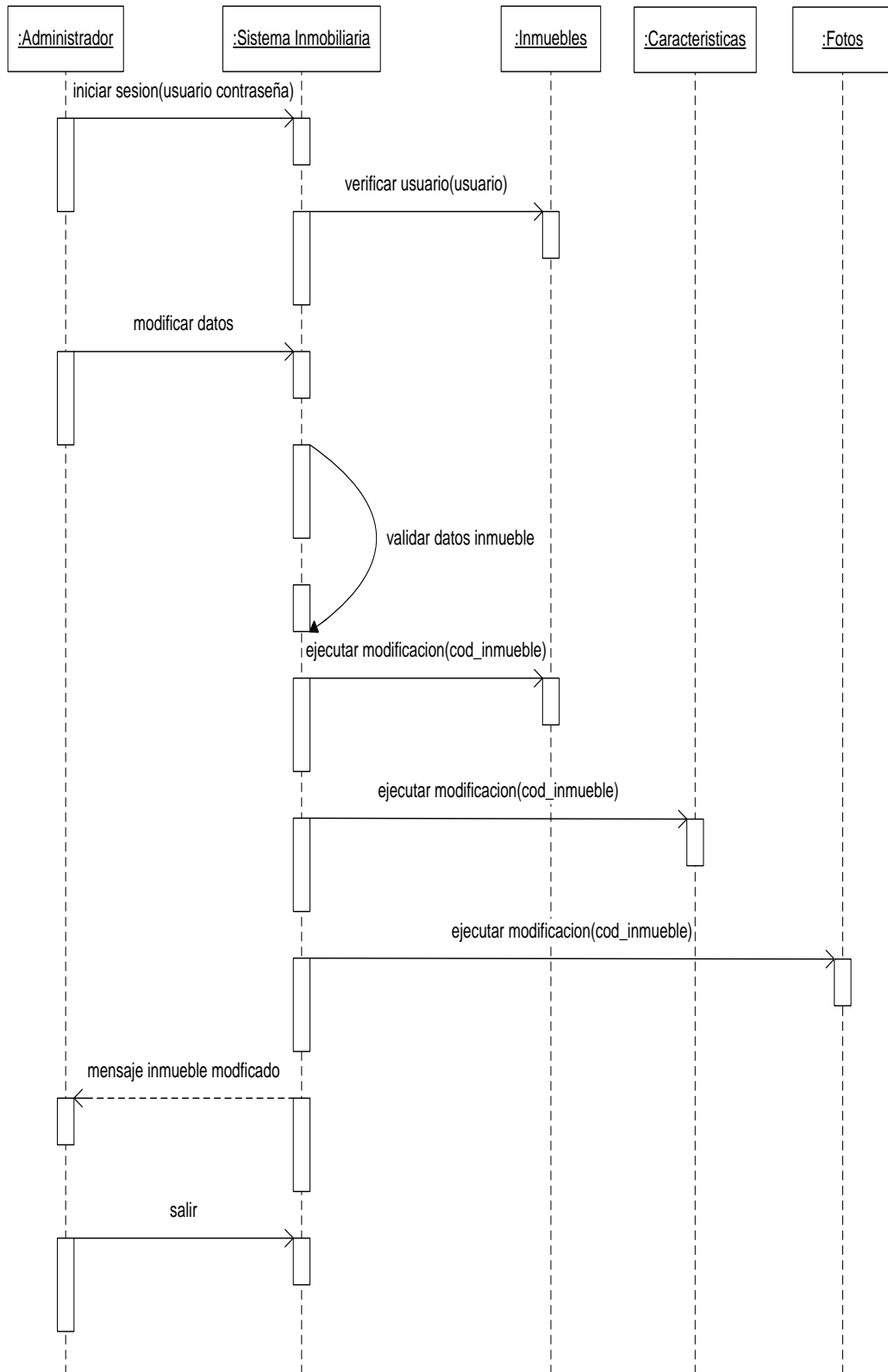


Figura 2.40: Diagrama de Estado del Caso de Uso 11 Modificar

## Eliminar

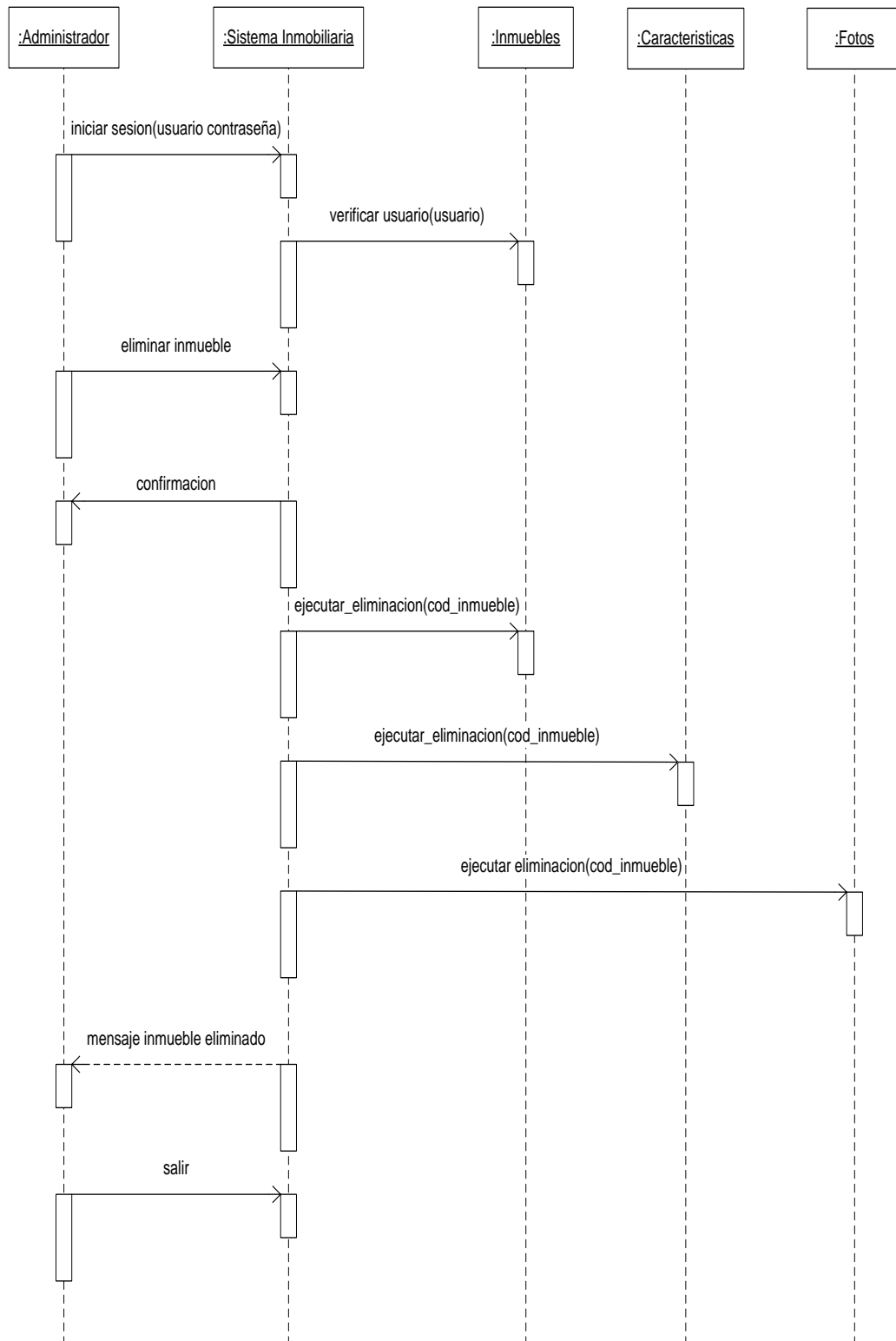


Figura 2.41: Diagrama de Estado del Caso de Uso 11 Eliminar

## Consultar

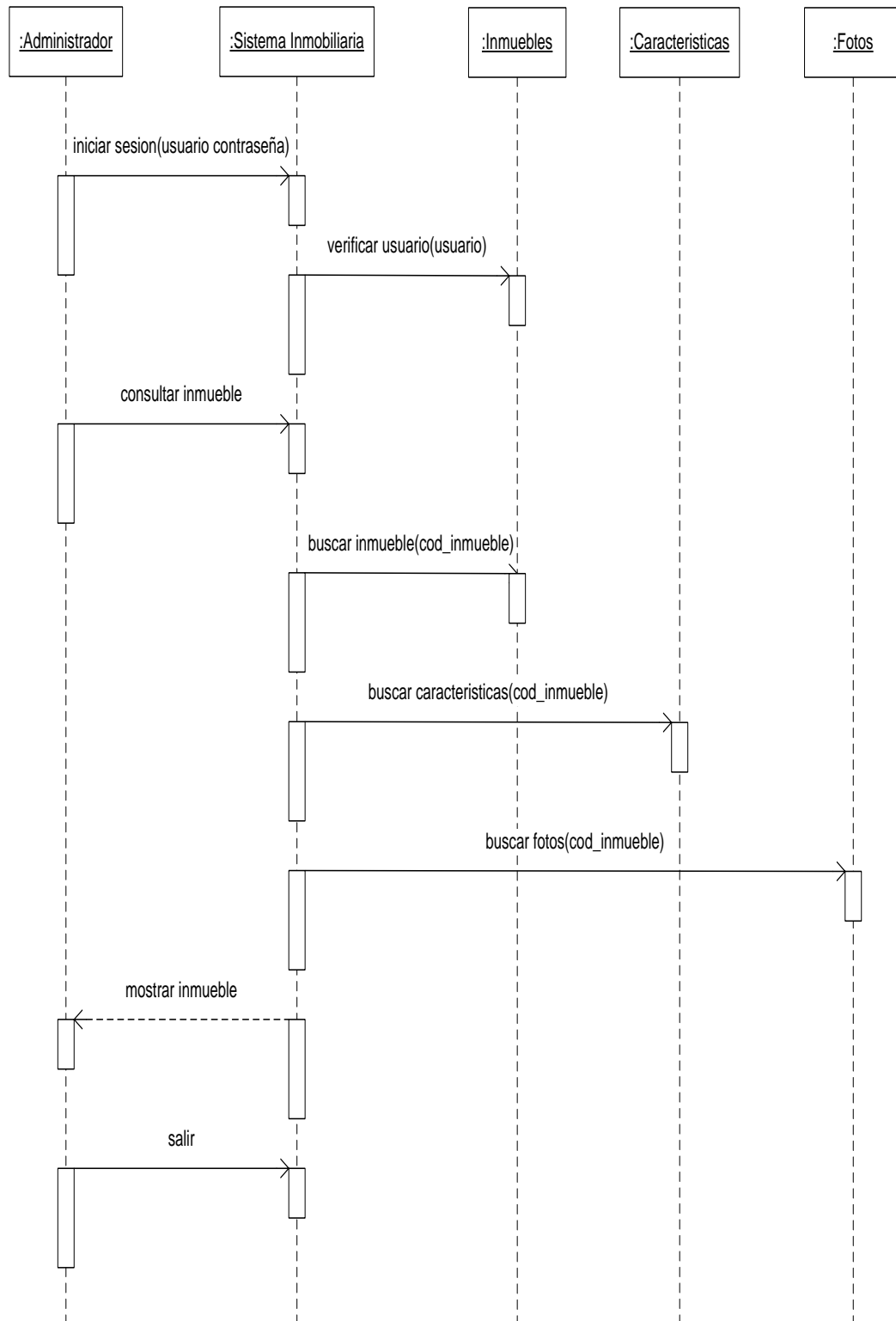


Figura 2.42: Diagrama de Estado del Caso de Uso 11 Consultar

## Caso de Uso 12: Crear Reportes Gerenciales

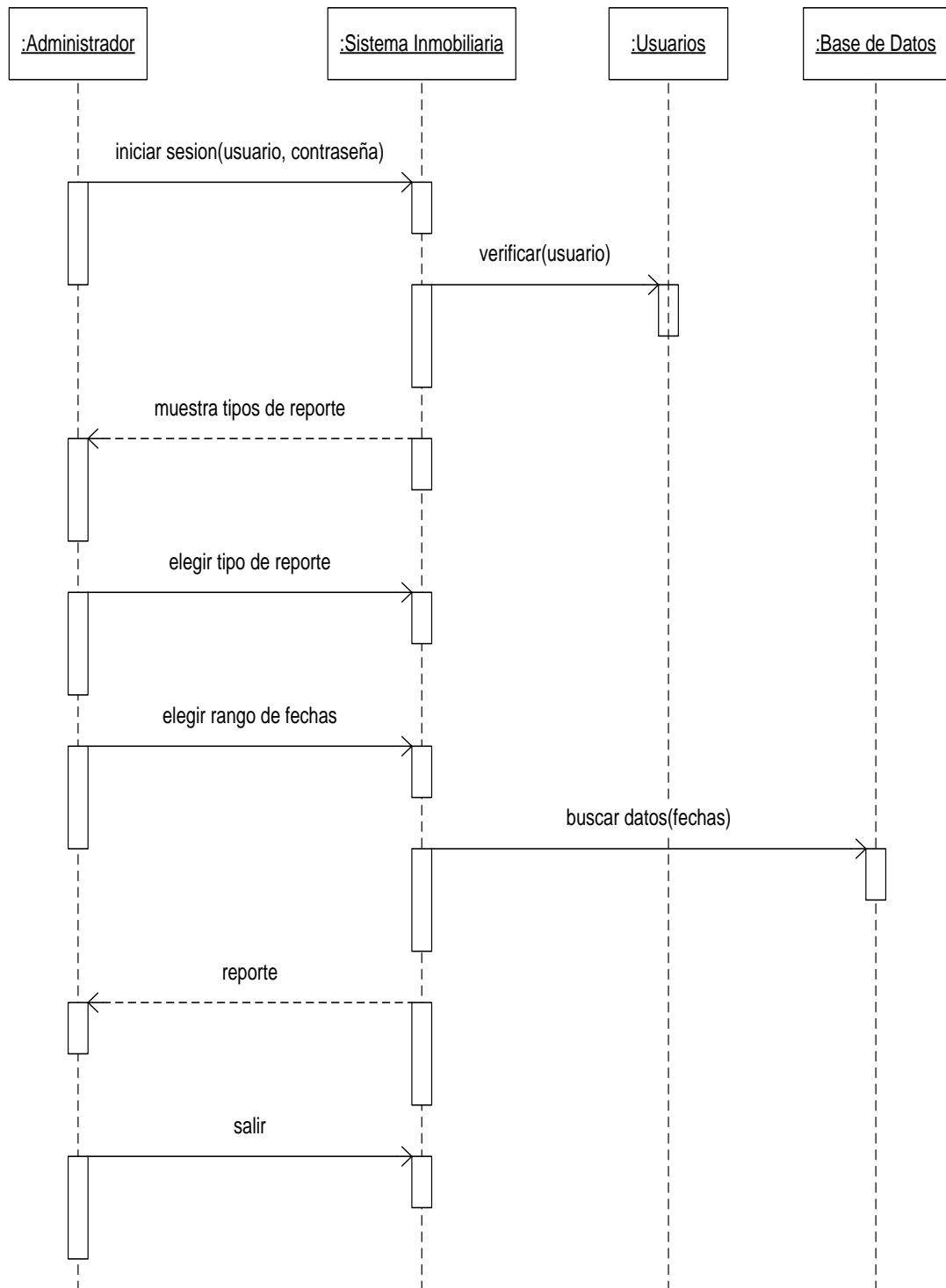


Figura 2.43: Diagrama de Estado del Caso de Uso 12

### Caso de Uso 13: Crear Reportes de Ventas

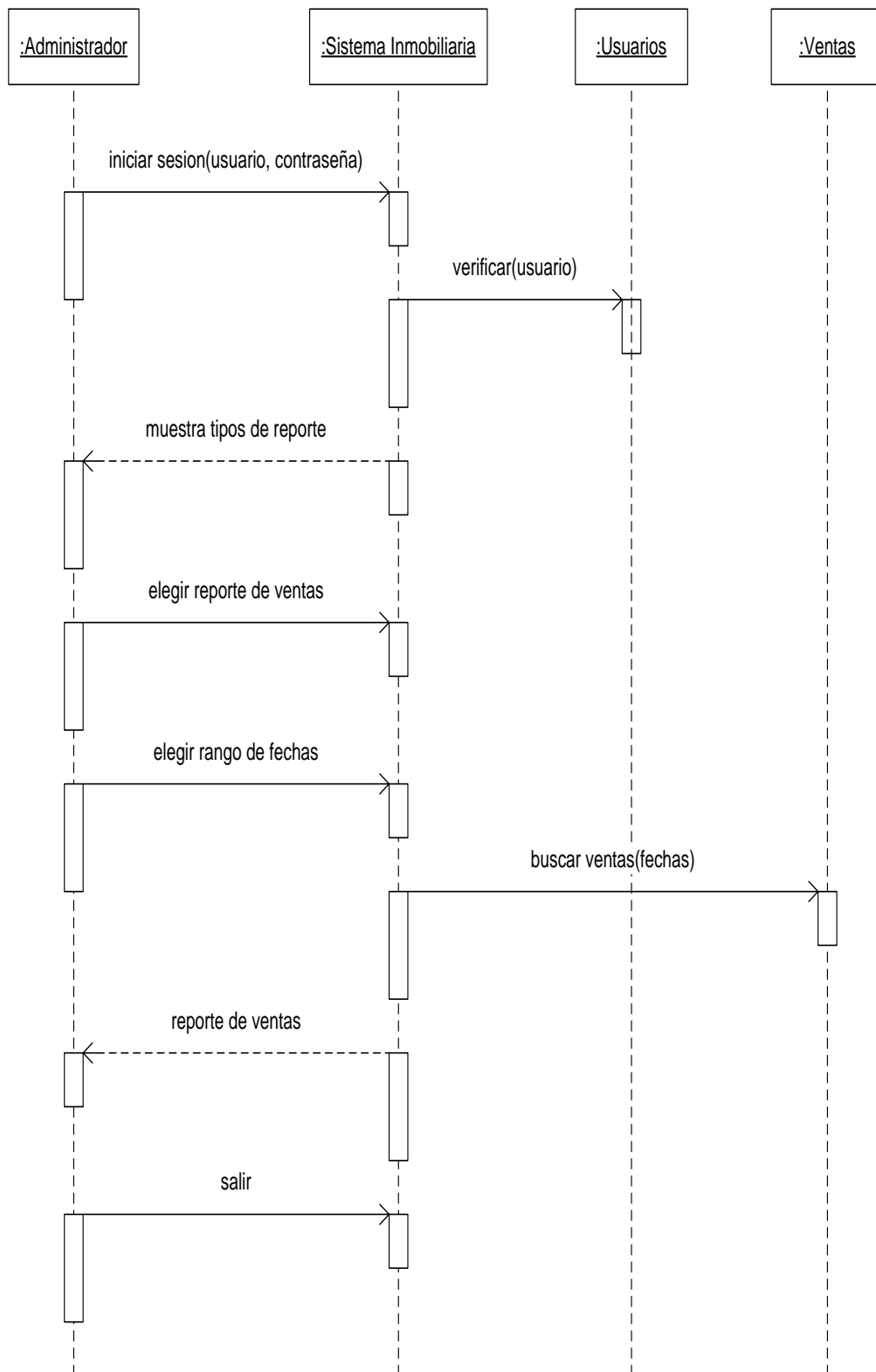


Figura 2.44: Diagrama de Estado del Caso de Uso 13

### Caso de Uso 14: Crear un Reporte con el tipo de Inmueble más Vendido

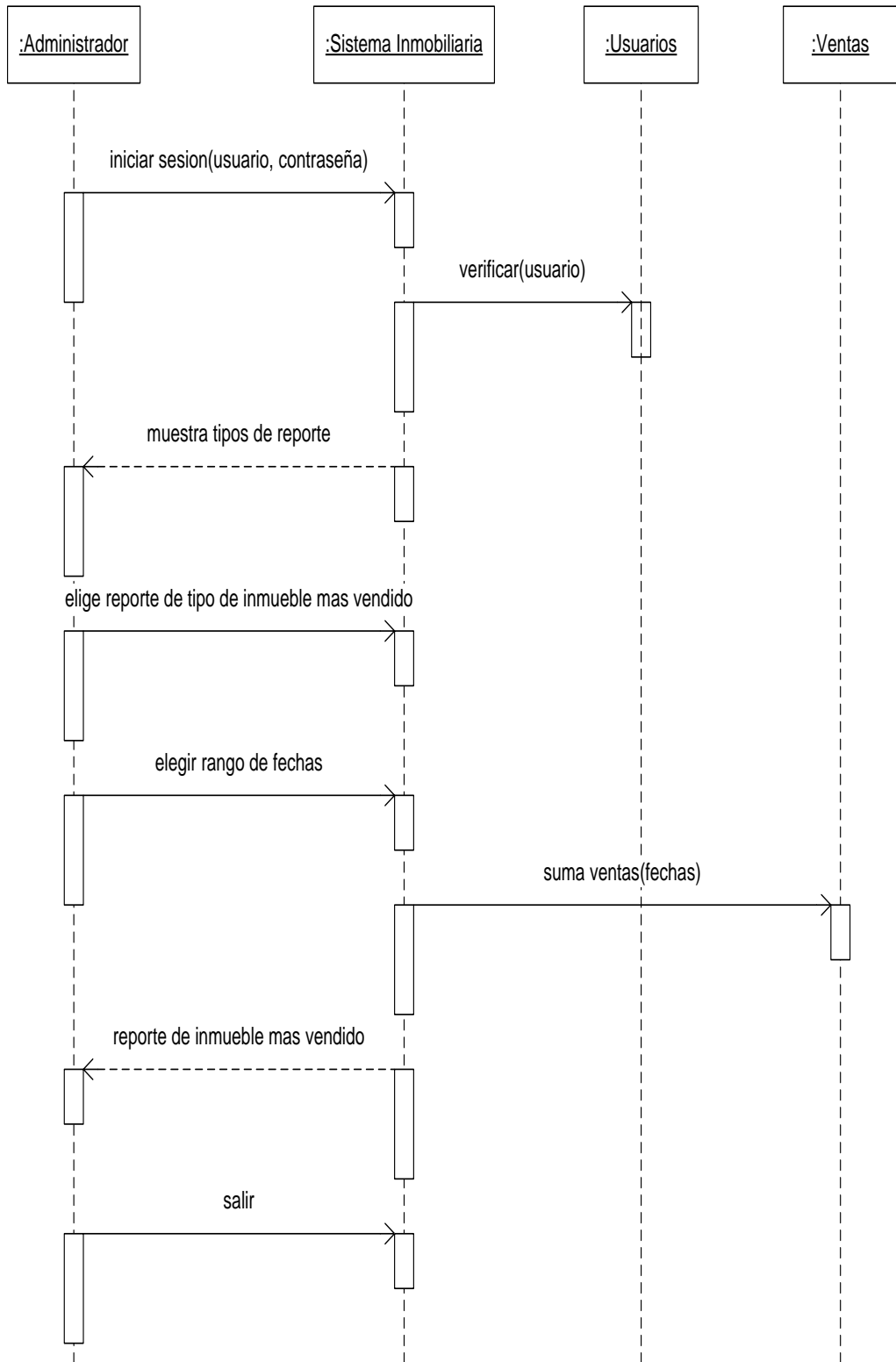


Figura 2.45: Diagrama de Estado del Caso de Uso 14

## Caso de Uso 15: Crear un Reporte del Vendedor más Exitoso

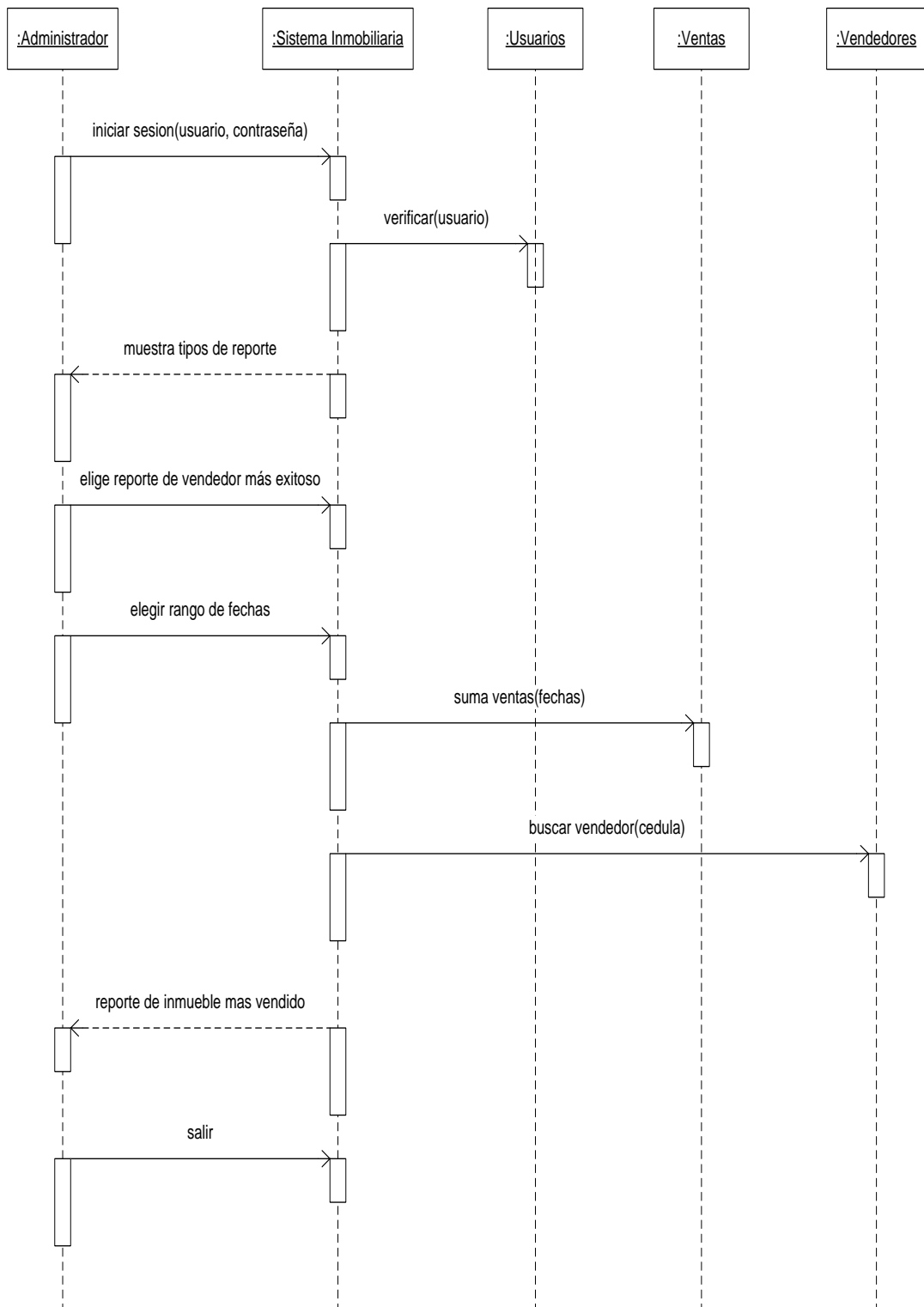


Figura 2.46: Diagrama de Estado del Caso de Uso 15

## Caso de Uso 16: Enviar Correo Electrónico

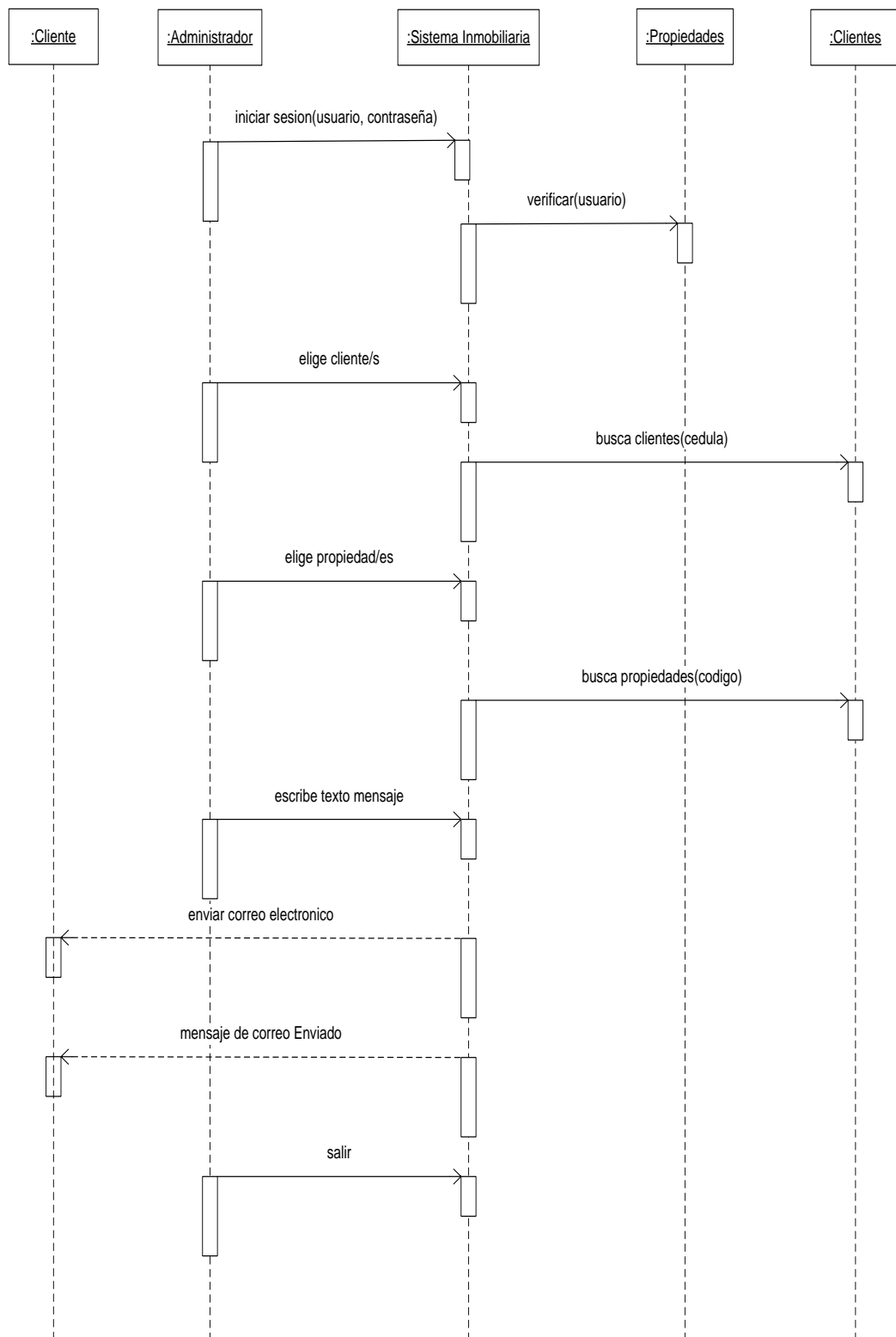


Figura 2.47: Diagrama de Estado del Caso de Uso 16



## Caso de Uso 17: Administrar Ventas

### Ingresar

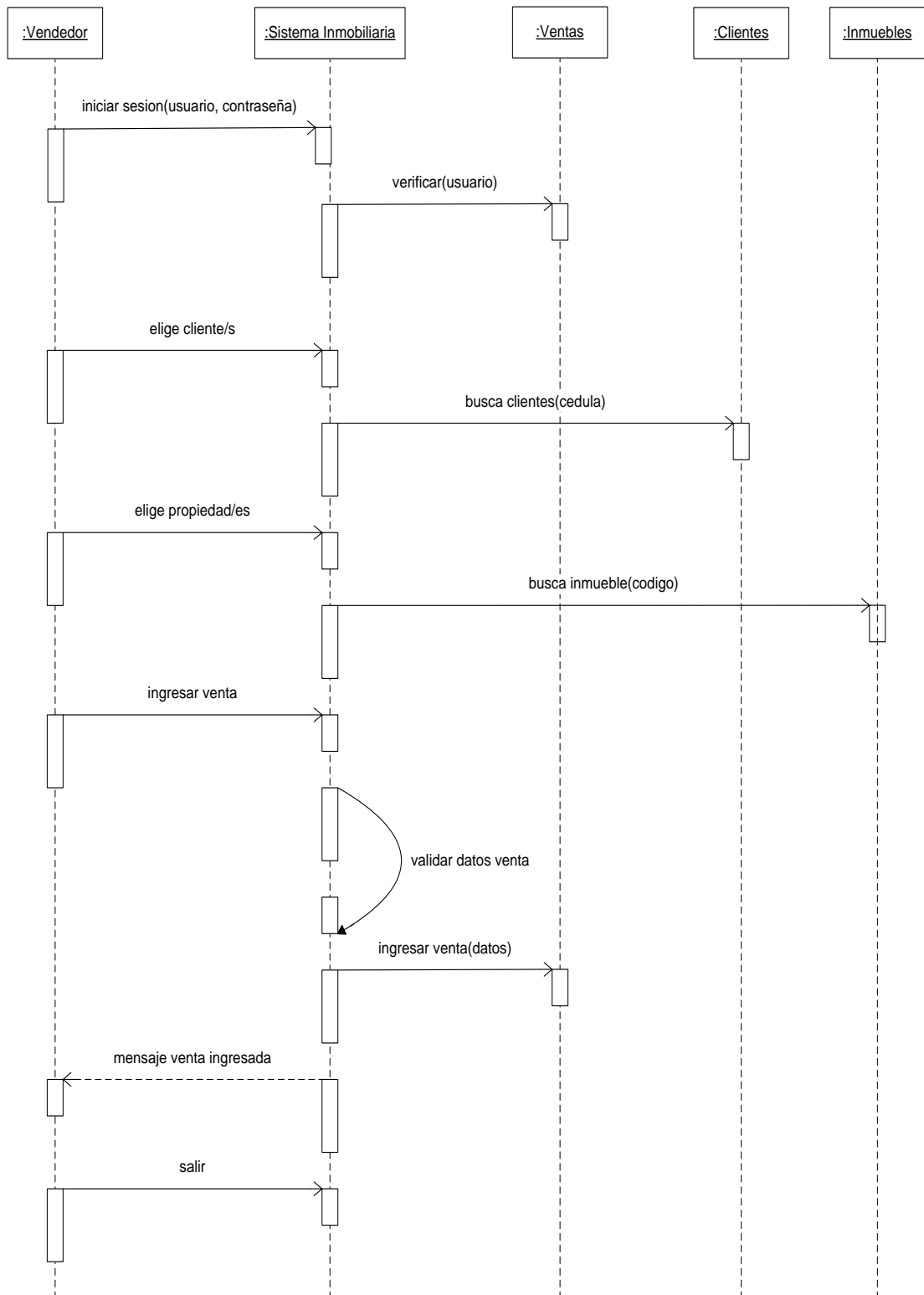


Figura 2.48: Diagrama de Estado del Caso de Uso 17 Ingresar

## Modificar

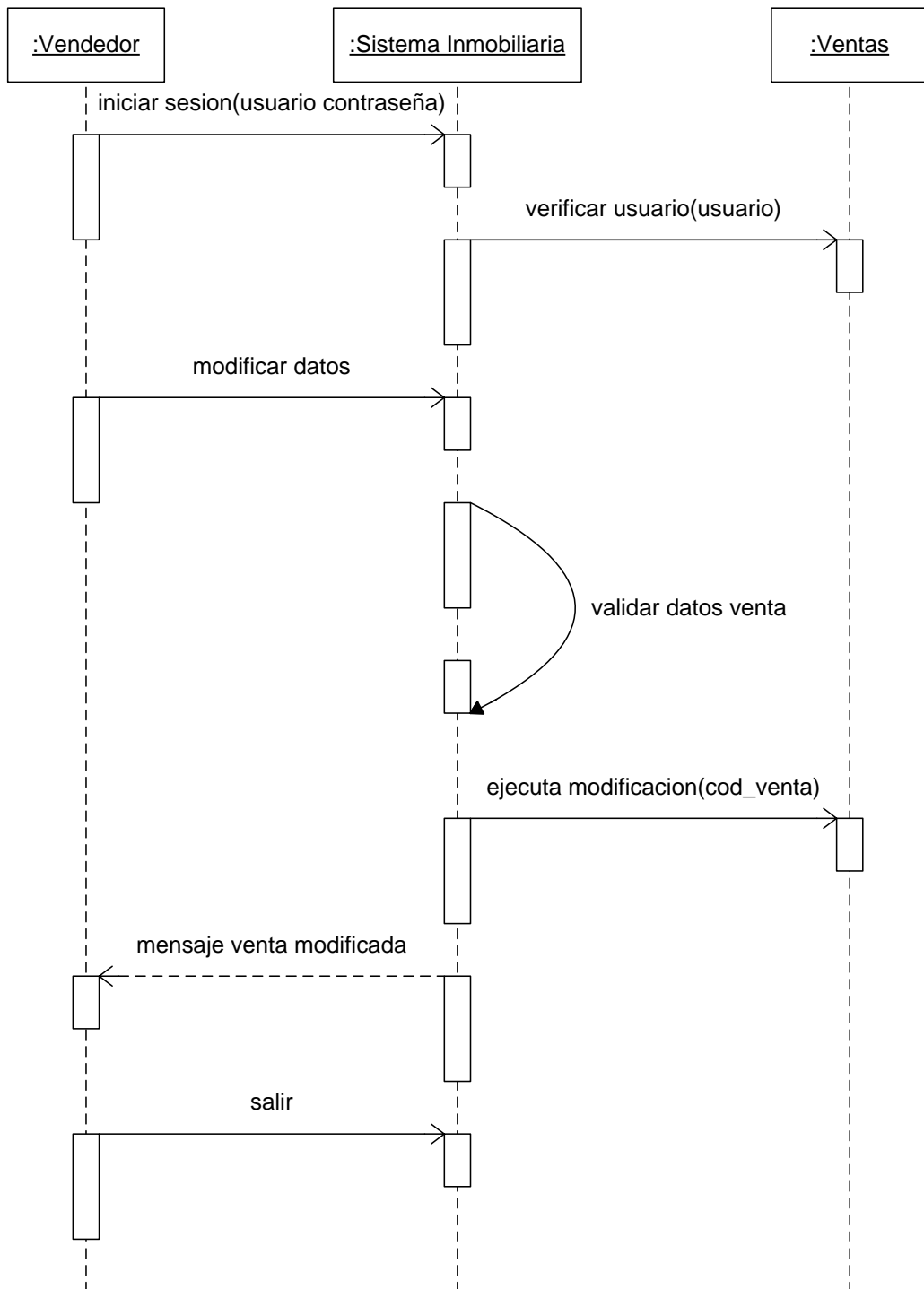


Figura 2.49: Diagrama de Estado del Caso de Uso 17 Modificar

## Eliminar

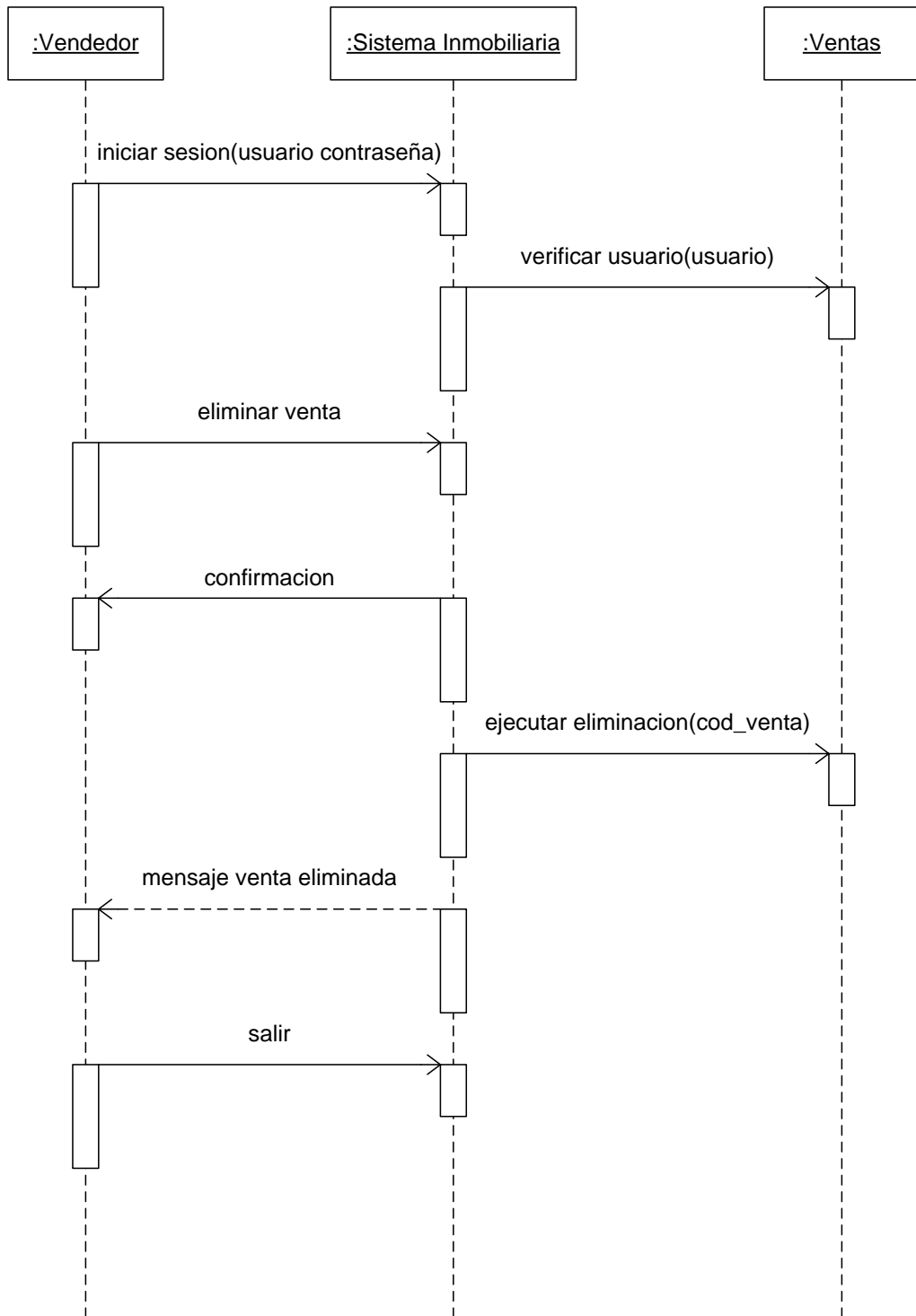


Figura 2.50: Diagrama de Estado del Caso de Uso 17 Eliminar

## Consultar

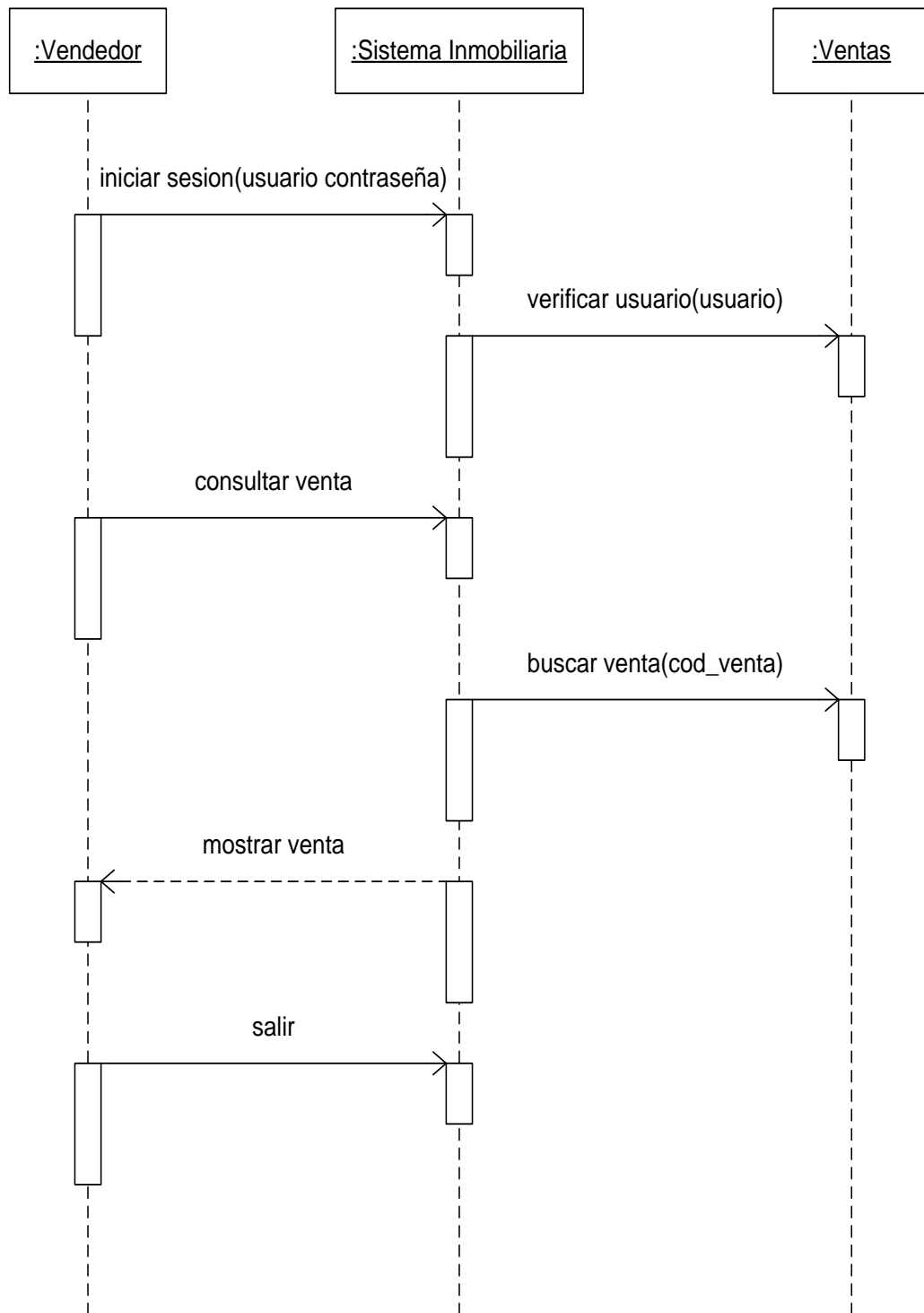


Figura 2.51: Diagrama de Estado del Caso de Uso 17 Consultar

### Caso de Uso 18: Crear Archivos XML

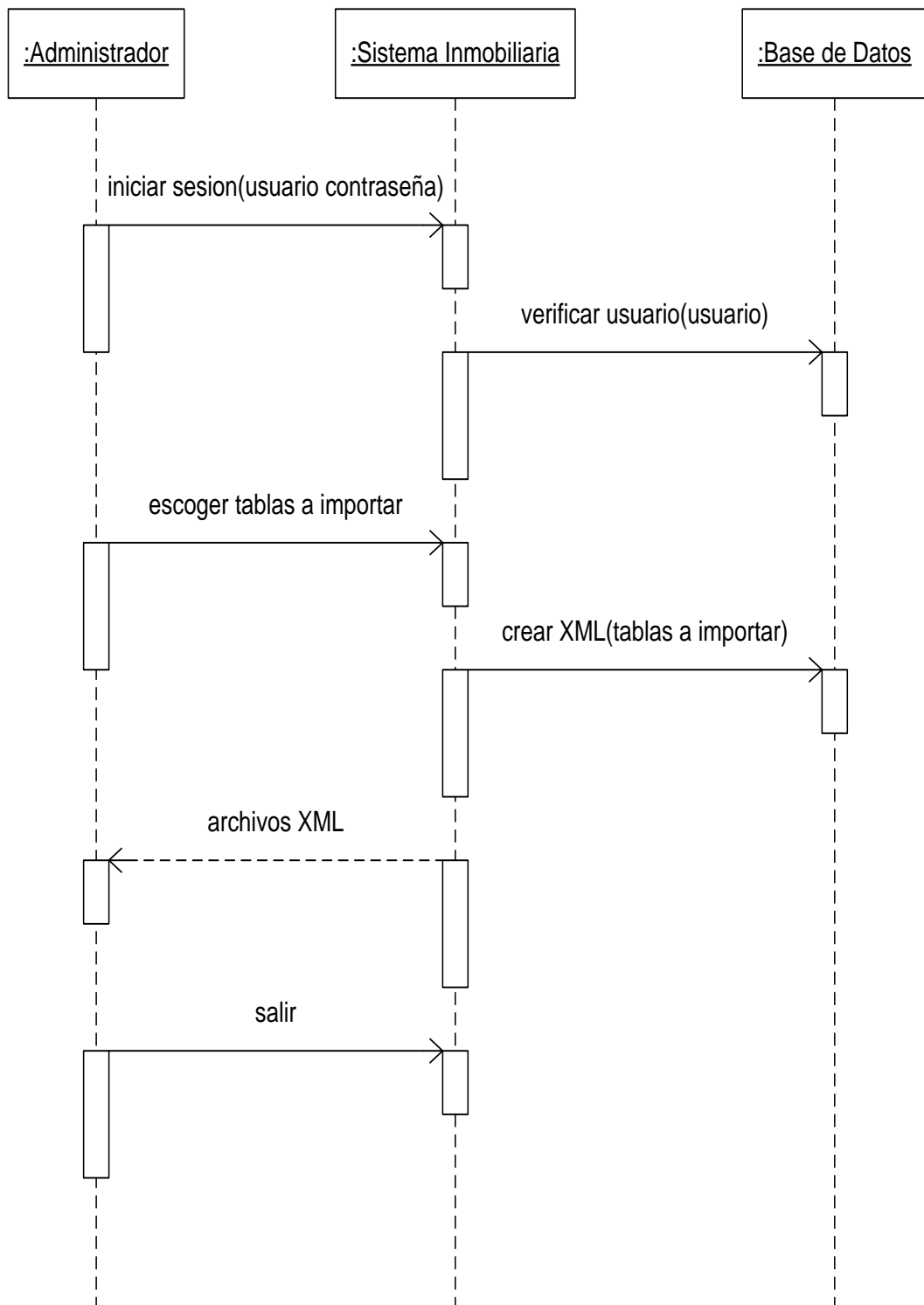


Figura 2.52: Diagrama de Estado del Caso de Uso 18

## Caso de Uso 19: Mantenimiento de Vendedores

### Ingresar

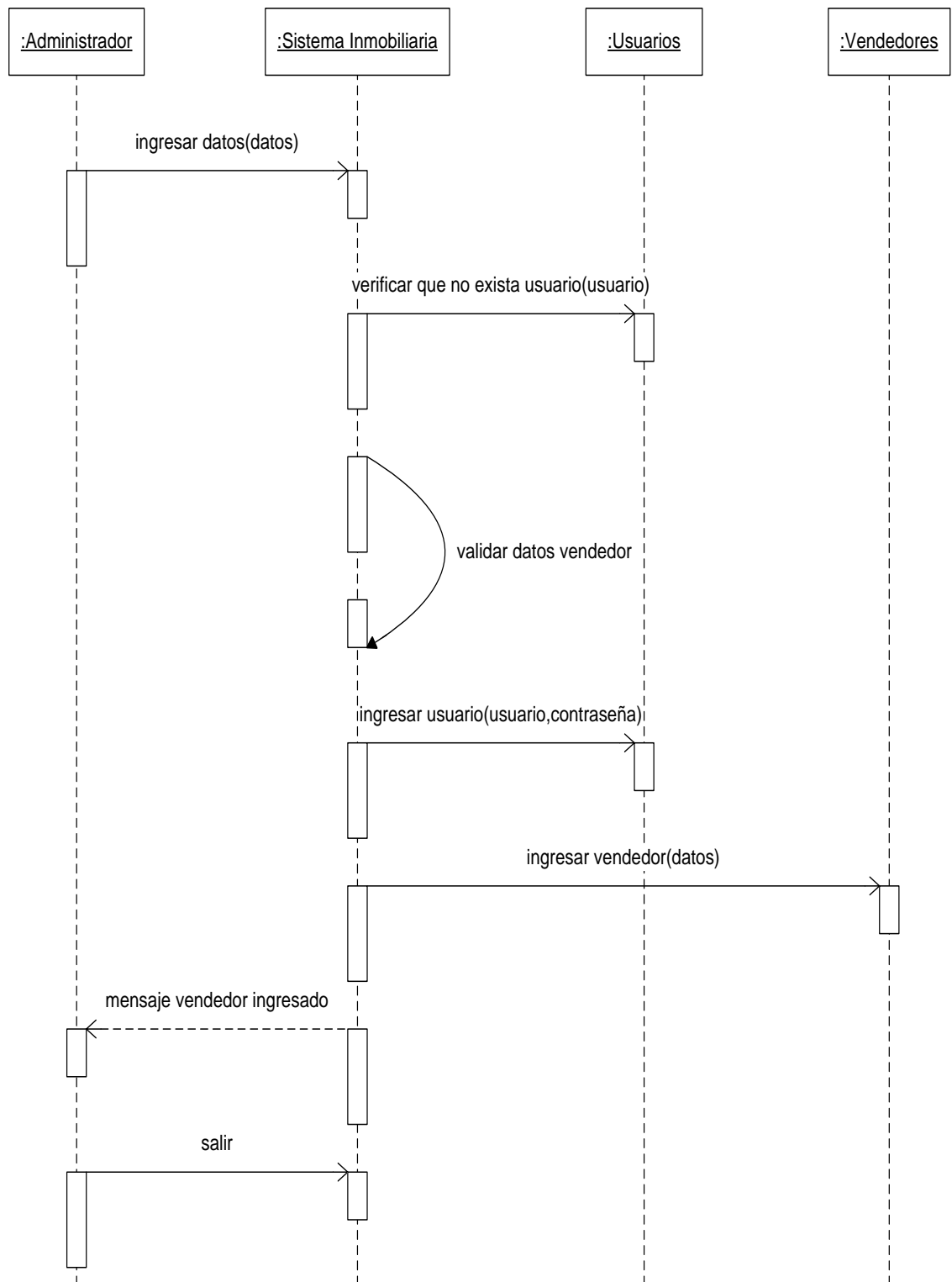


Figura 2.53: Diagrama de Estado del Caso de Uso 19 Ingresar

## Modificar

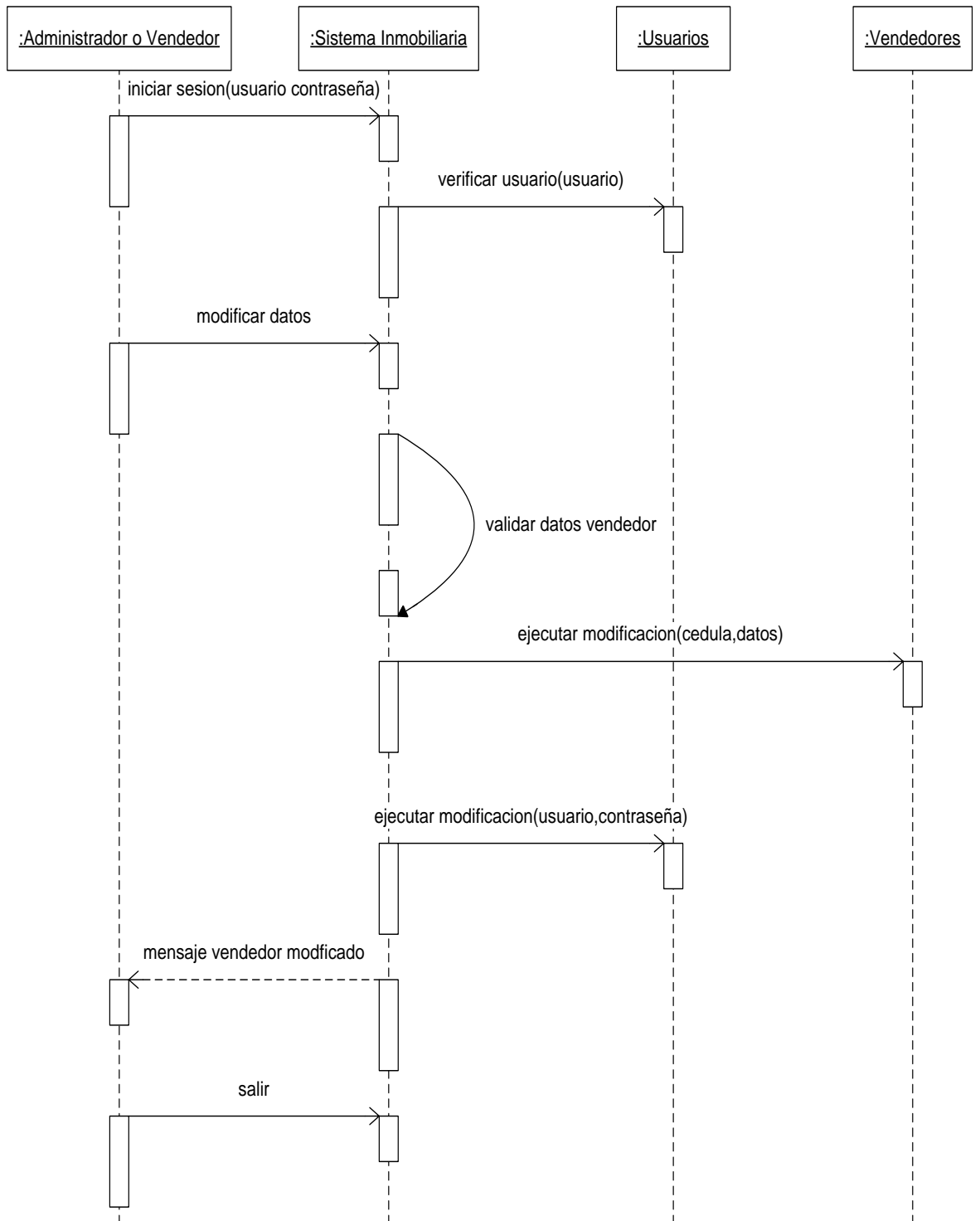


Figura 2.54: Diagrama de Estado del Caso de Uso 19 Modificar

## Eliminar

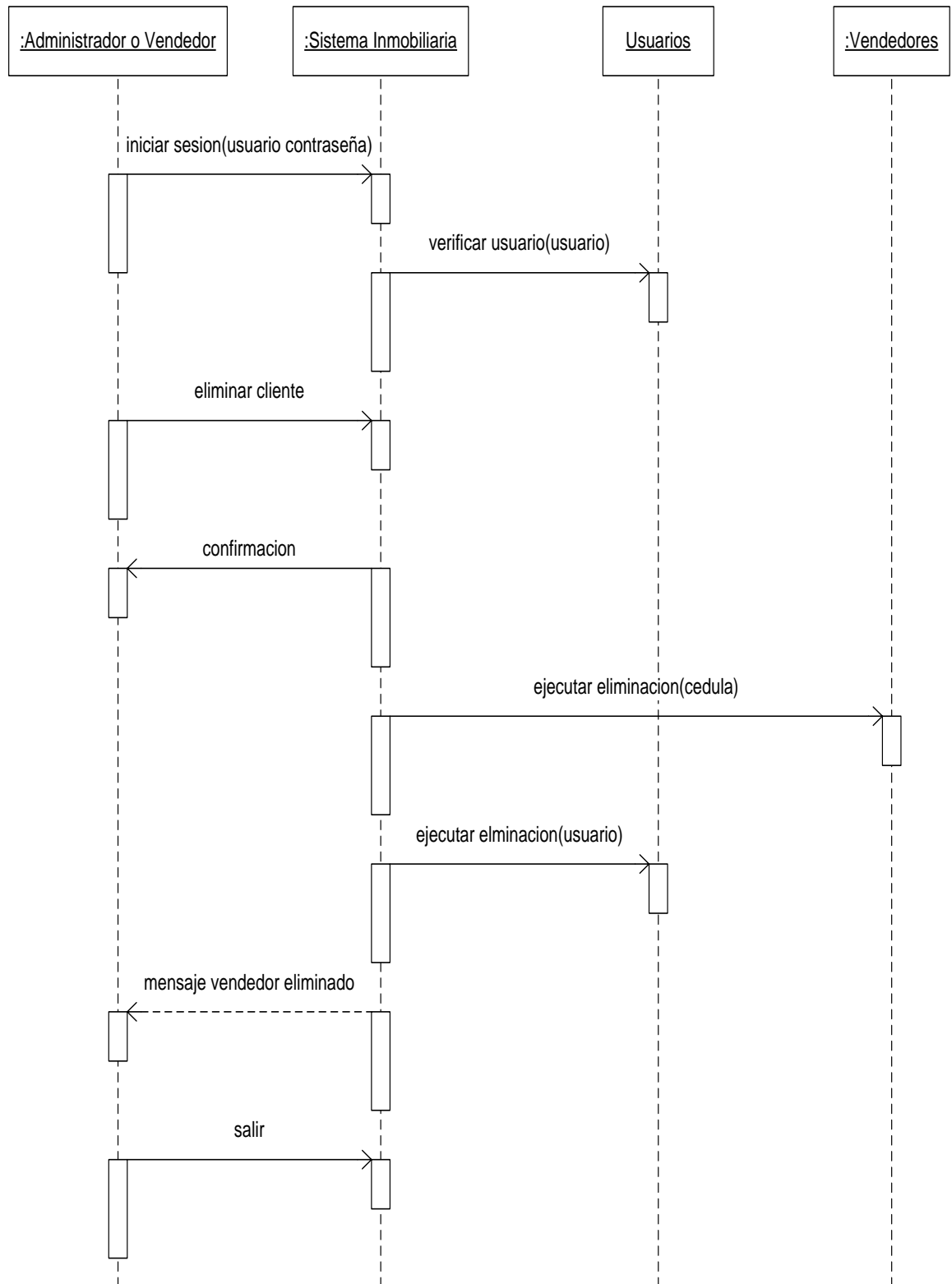


Figura 2.55: Diagrama de Estado del Caso de Uso 19 Eliminar



## Consultar

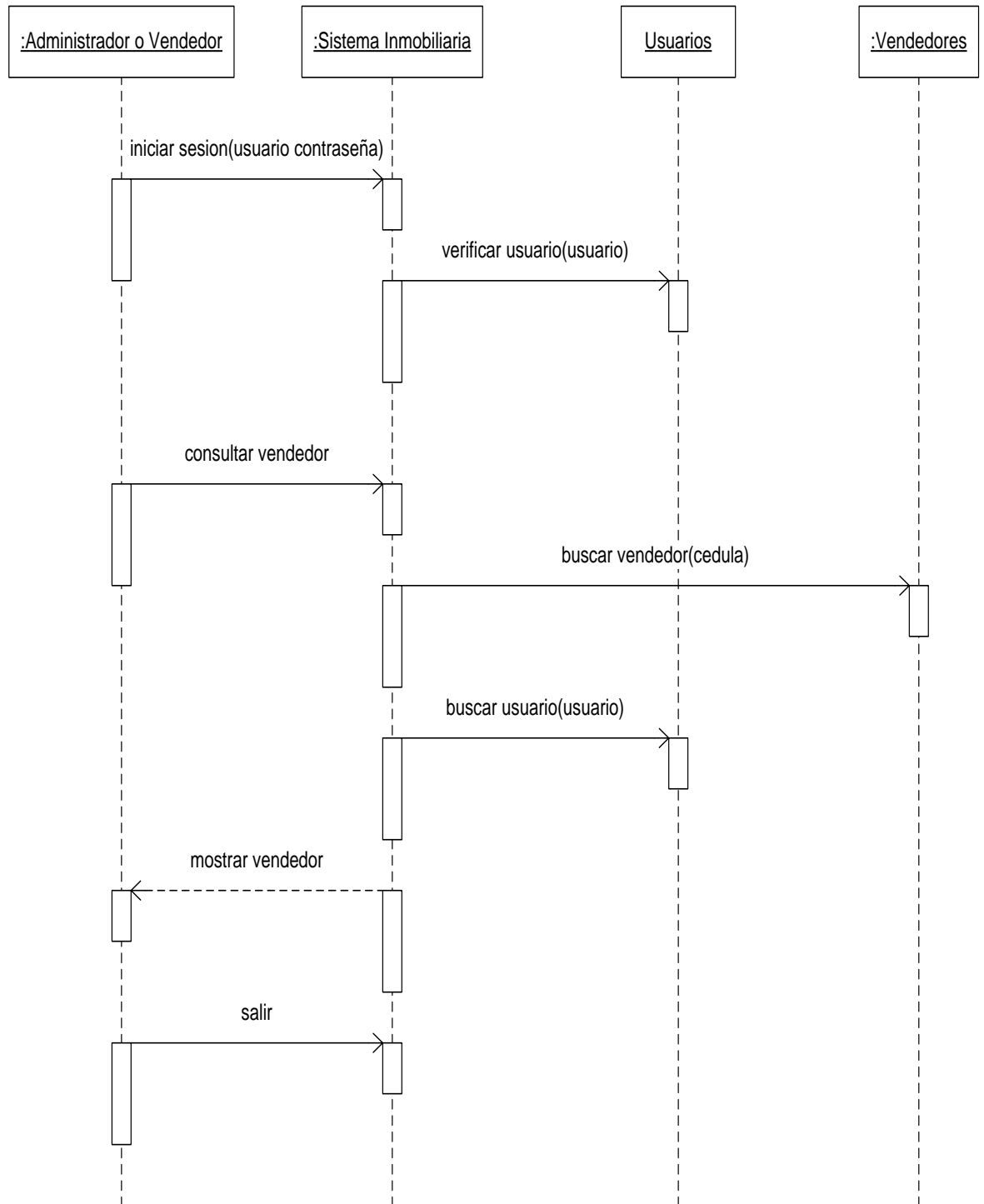


Figura 2.56: Diagrama de Estado del Caso de Uso 19 Consultar

## Caso de Uso 20: Recoger Características

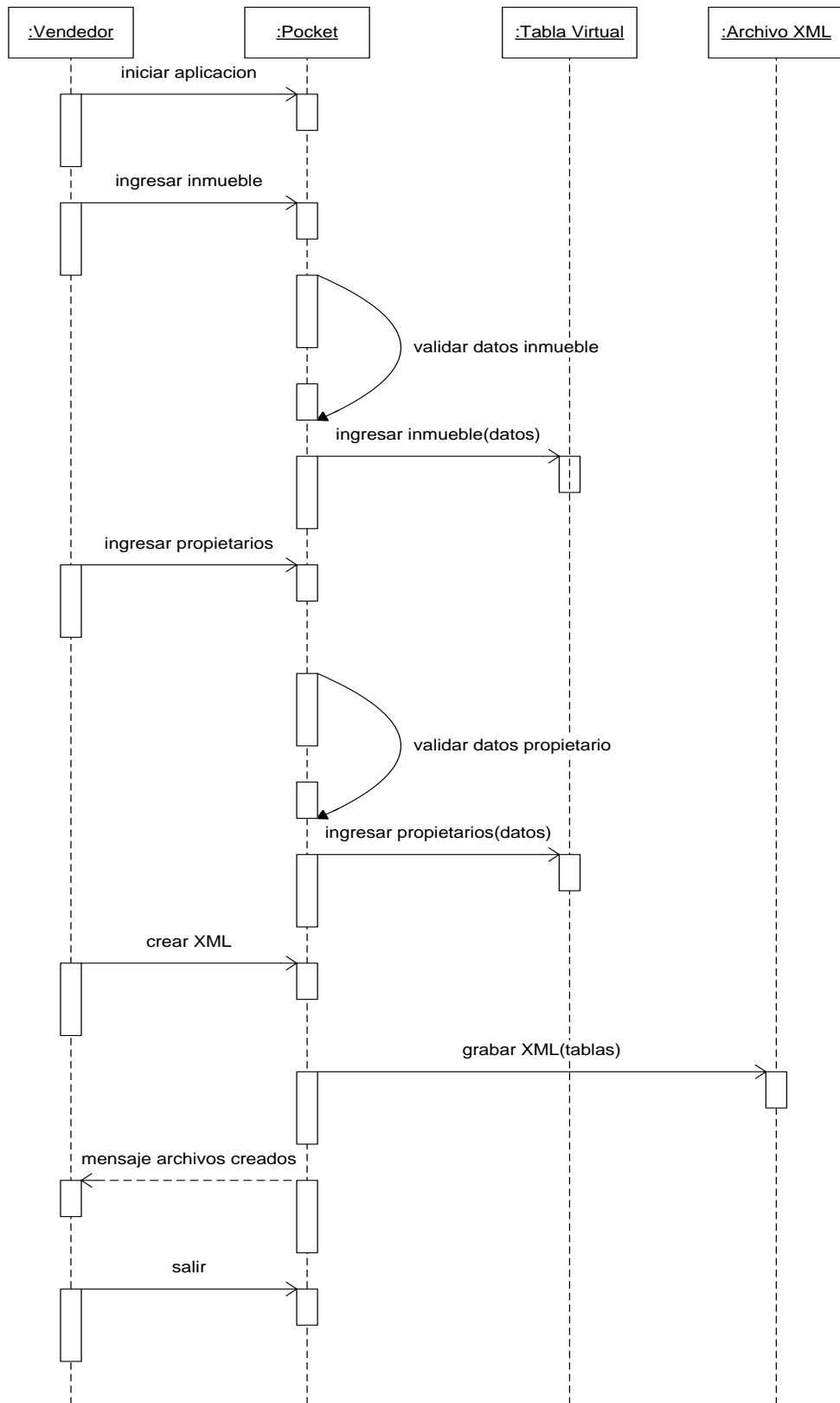


Figura 2.57: Diagrama de Estado del Caso de Uso 20

## Caso de Uso 21: Descargar Características

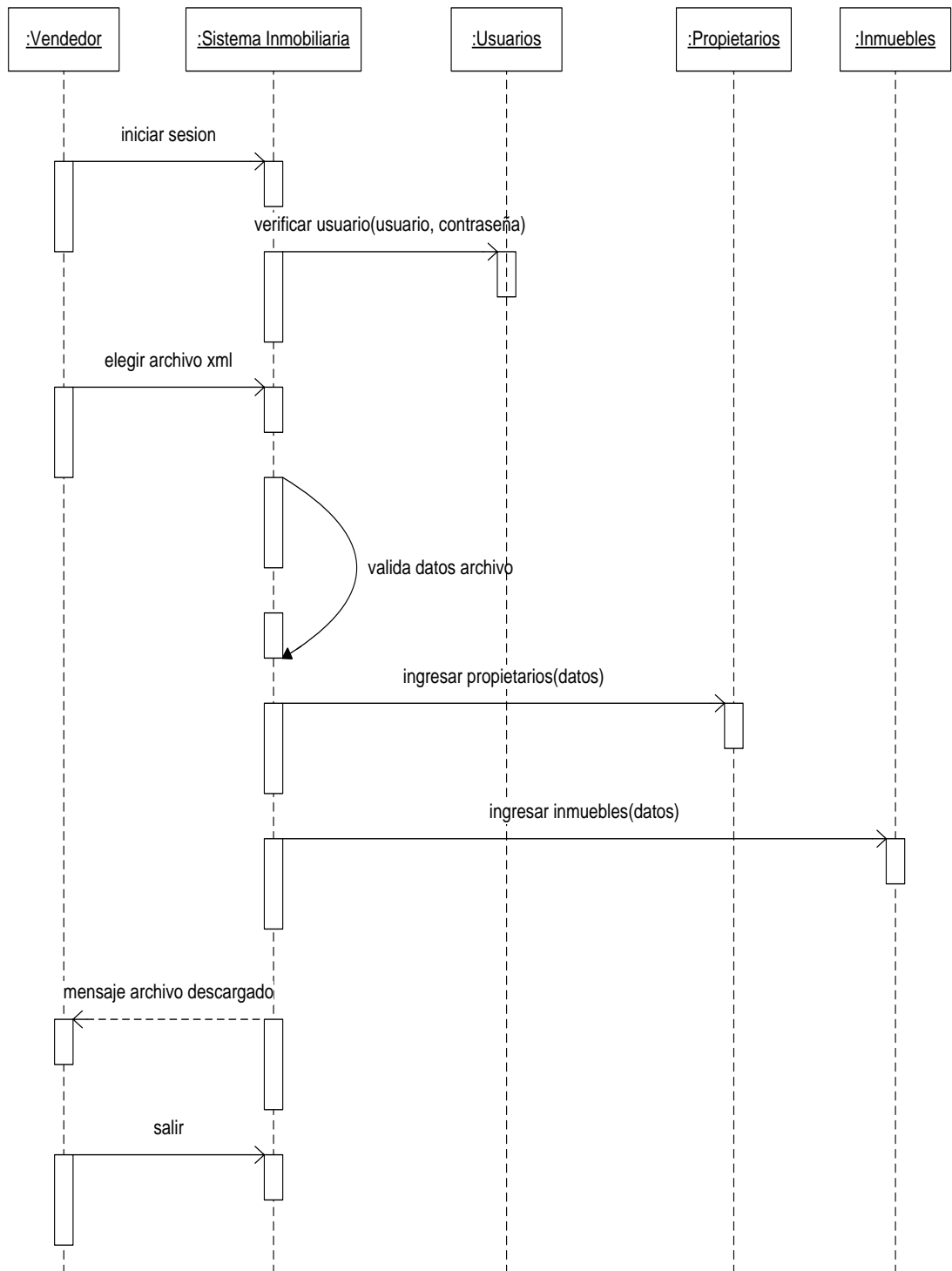


Figura 2.58: Diagrama de Estado del Caso de Uso 21

## Caso de Uso 22: Consultar Horarios

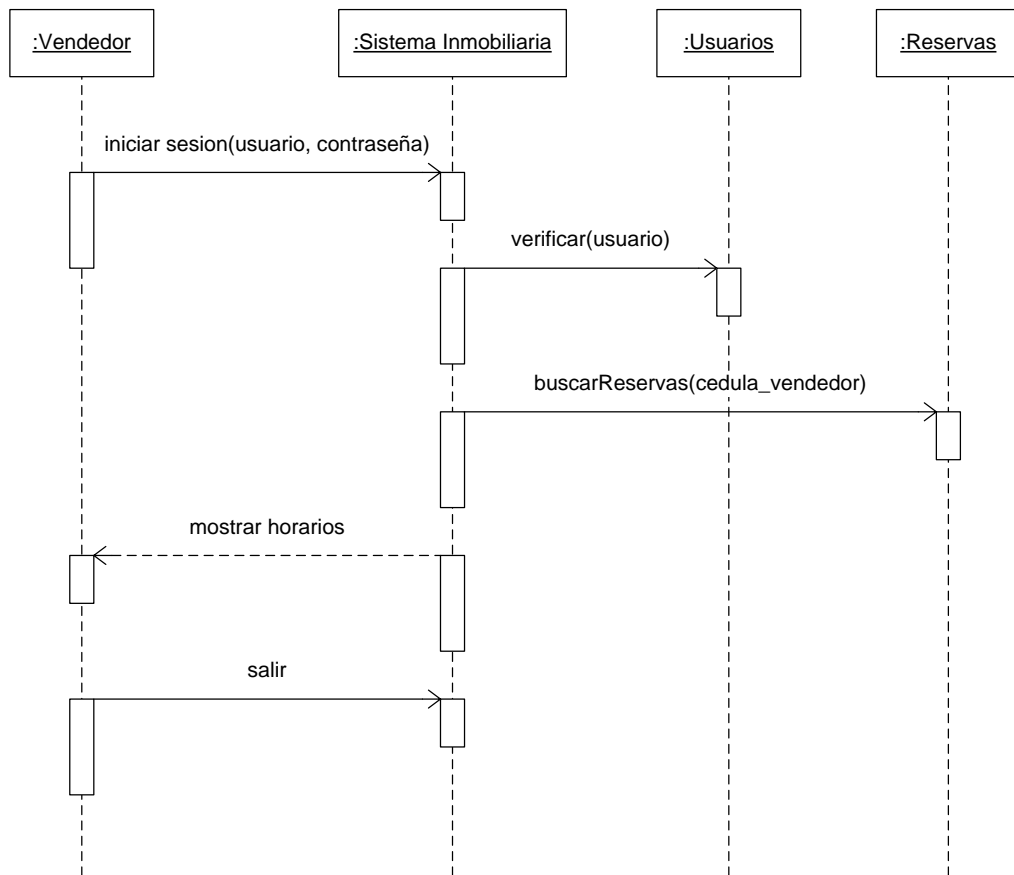


Figura 2.59: Diagrama de Estado del Caso de Uso 22

## **2.1 Análisis y definición de los procesos que formarán parte del Sistema de Gestión para una Inmobiliaria**

El análisis y definición de los procesos del Sistema de Gestión Inmobiliaria, se harán mediante diagramas de flujo de datos utilizando la Notación de Yourdon. Estos diagramas nos darán una visión general de las principales funciones y transformaciones del sistema. Se hizo para profundizar el análisis del sistema.

Los diagramas de flujo permiten identificar los flujos de entrada y los flujos de salida y sus respectivos orígenes, es decir las entidades que envían flujos de entrada al Sistema y las entidades a las que van dirigidas los flujos de salida. Es importante recalcar que se obvian los flujos de datos de los errores.

### **2.1.1 Diagrama de Contexto**

El Diagrama de Contexto o también llamado Diagrama de Nivel 0 delimita el sistema y su entorno. Ha sido realizado para mostrar la interacción entre las entidades externas (Visitante, Clientes, Vendedores y Administrador) y el Sistema de Gestión Inmobiliaria.

En este diagrama podemos ver al Sistema como un solo macroproceso, es decir se definirá a que aspectos principales debe responder la aplicación. Se debe considerar que en este diagrama se especificará que debe realizar el Sistema, no como debe hacerlo.

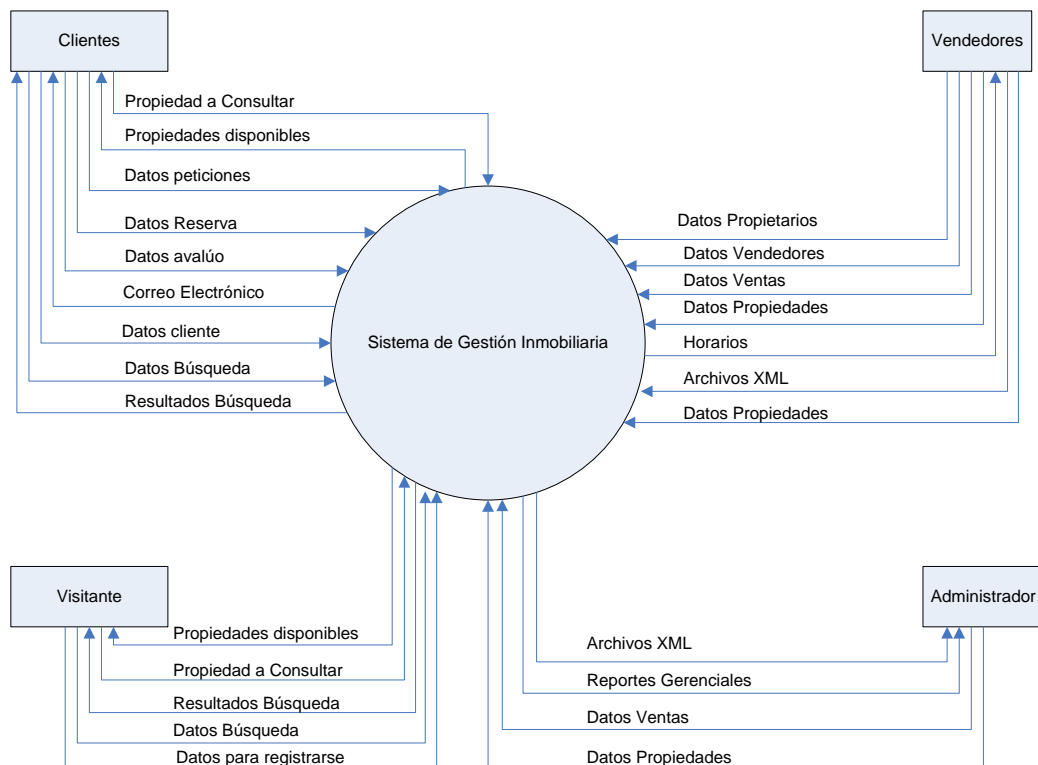


Figura 2.60: Diagrama de Contexto del Sistema de Gestión Inmobiliaria

## 2.1.2 Diagramas de Flujo de Datos

Los Diagramas de Flujo de datos se realizan en base a los flujos de información especificados en el Diagrama de Contexto. Para el Sistema de Gestión Inmobiliaria se ha visto conveniente diagramar a Nivel Superior y a Nivel de Expansión, también llamado Nivel de Detalle.

Al Nivel Superior también se lo conoce como Diagrama de Nivel 1.

Al Nivel de Detalle se lo conoce como Diagramas de Nivel 2.

### 2.1.2.1 Diagramas de Nivel Superior

El Diagrama de Flujo de Datos de Nivel Superior, es una explosión del Diagrama de Contexto. Explosión quiere decir que se va a descomponer al Sistema en procesos, los cuales tendrán nombre propio y numeración para identificarlos posteriormente.

El Diagrama de Nivel 1 se ha realizado en base al Diagrama de Contexto y al Modelo de Casos de Uso, el mismo que indica que debe hacer el Sistema. En este diagrama se especificará los flujos de entrada y salida para cada proceso, las entidades externas que alimentan y reciben información del sistema, y los almacenes de datos donde se guardará la información que va a ser procesada.

Para evitar flujos de información redundantes y un diagrama excesivamente grande e ilegible cabe recalcar que el Administrador puede realizar todos los procesos que realizan el vendedor, el cliente y el visitante.

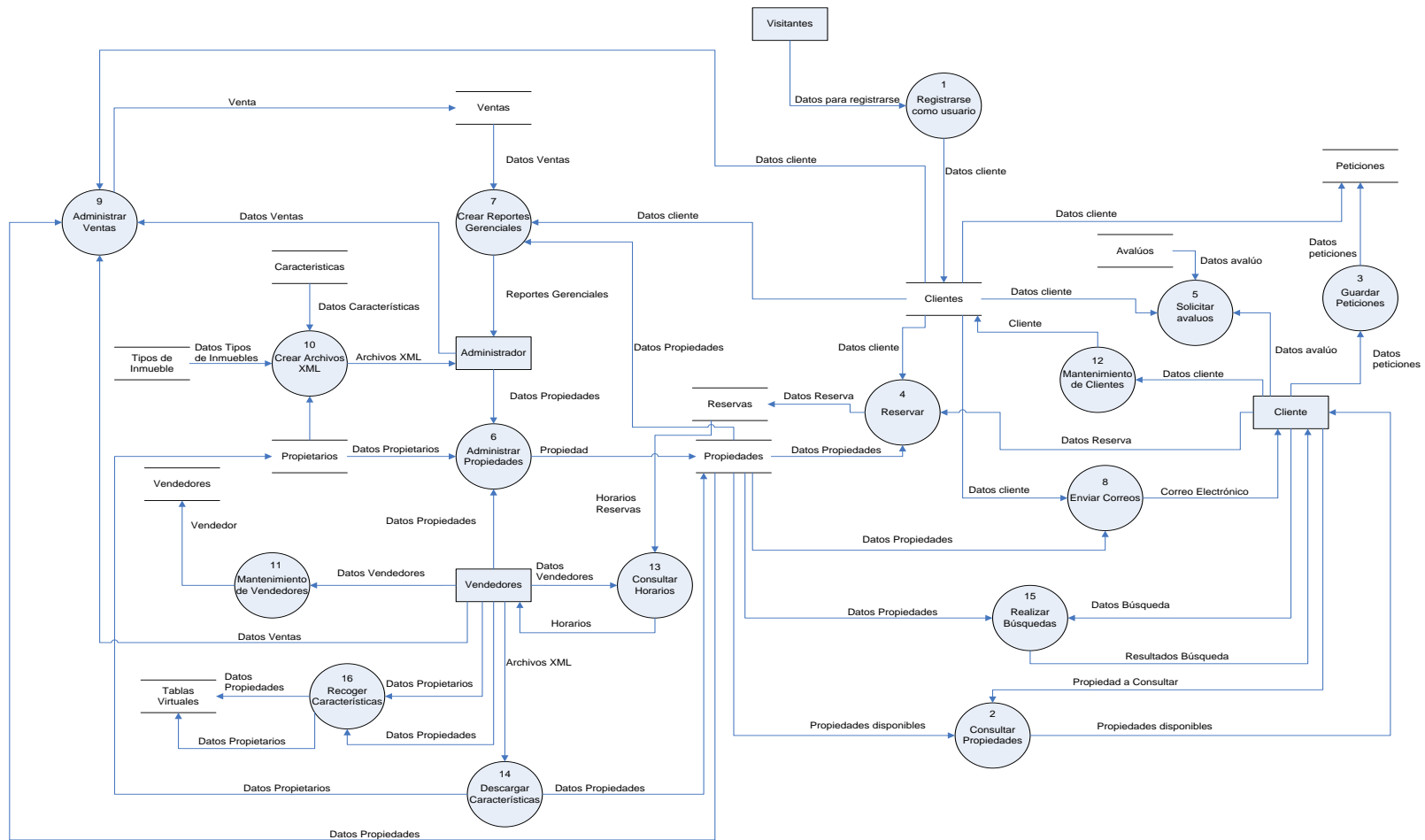


Figura 2.61: Diagrama de Nivel Superior del Sistema de Gestión Inmobiliaria



### 2.1.2.2 Diagramas de Expansión

Los Diagramas de Expansión se utilizan para refinar los procesos en un DFD más detallado. A partir del Diagrama de Nivel 1 se ha refinado cada uno de los 16 procesos que hemos señalado como principales, a excepción de los Procesos 3 (Guardar Peticiones), 10 (Crear Archivos XML) y 13 (Consultar Horarios) que no poseen subprocesos, por lo cual no requieren refinamiento.

Se ha refinado hasta el Nivel 2, ya que en este Nivel hemos encontrado los procesos primitivos, es decir que ya no se pueden descomponer a mayor detalle.

#### Proceso 1: Registrarse como Usuario

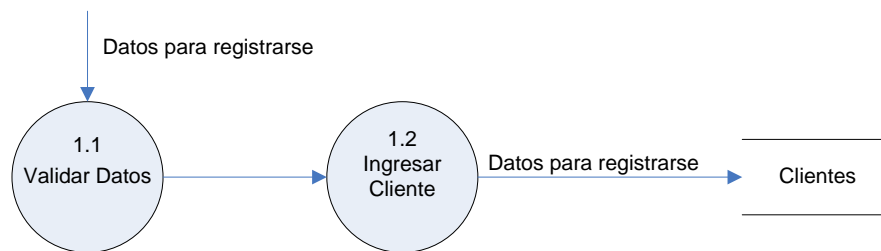


Figura 2.62: Diagrama a Nivel de Detalle del Proceso 1

#### Proceso 2: Consultar Propiedades

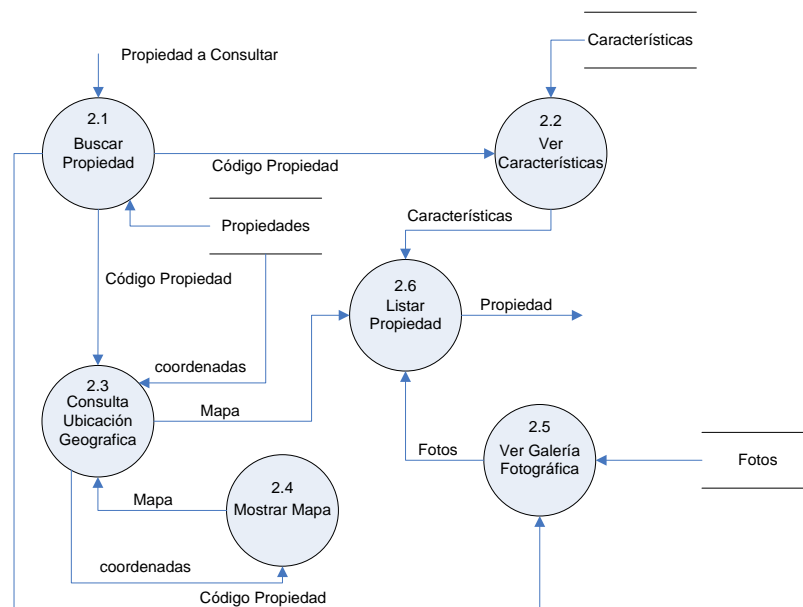


Figura 2.63: Diagrama a Nivel de Detalle del Proceso 2

### Proceso 3: Guardar Peticiones. No Requiere Refinamiento

### Proceso 4: Reservar

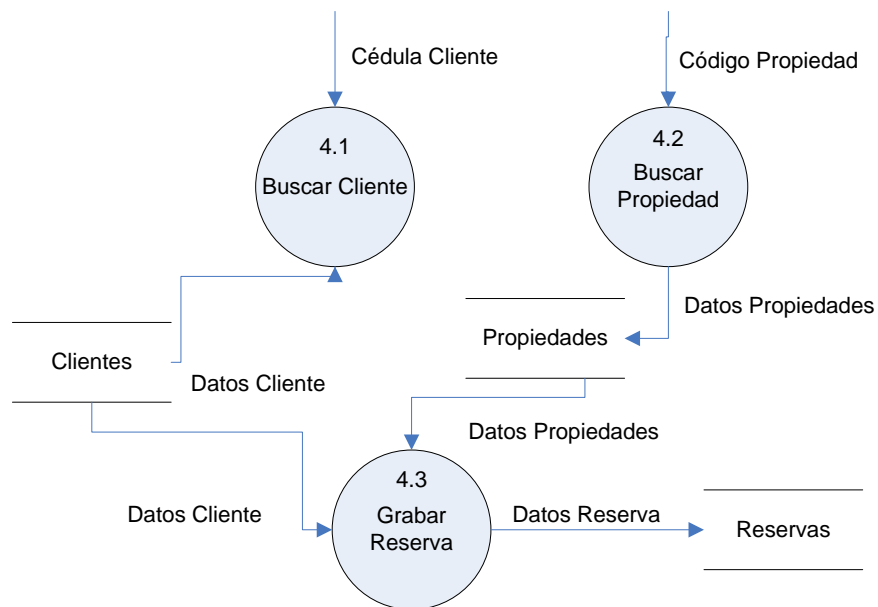


Figura 2.64: Diagrama a Nivel de Detalle del Proceso 4

### Proceso 5: Solicitar Avalúo

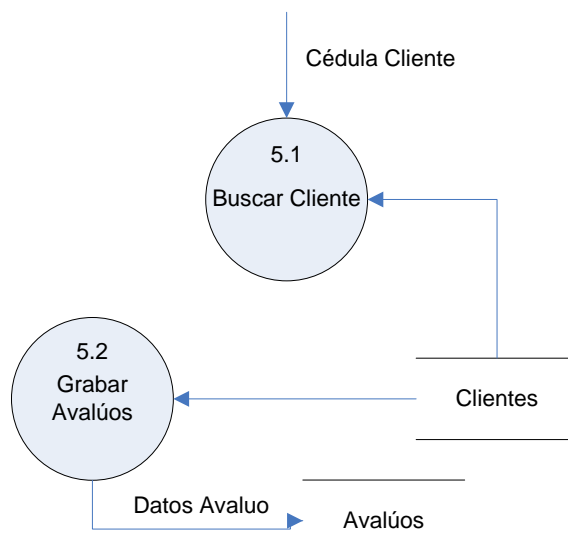


Figura 2.65: Diagrama a Nivel de Detalle del Proceso 5

**Proceso 6: Administrar Propiedades**

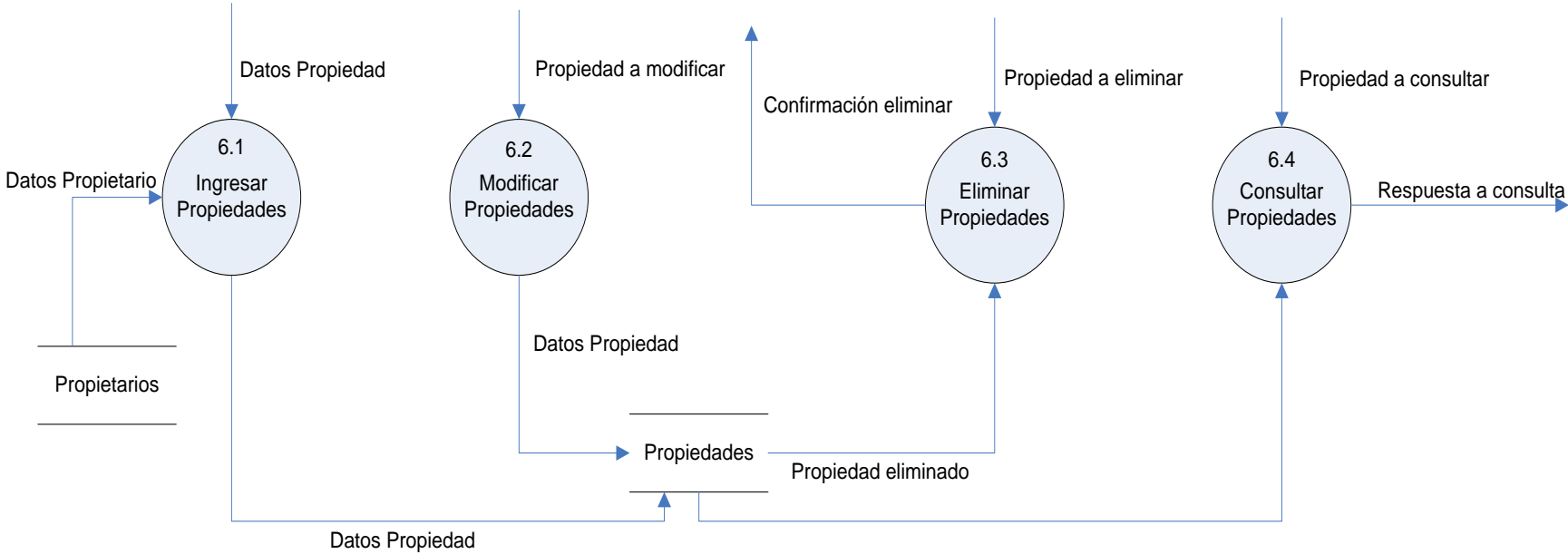


Figura 2.66: Diagrama a Nivel de Detalle del Proceso 6

**Proceso 7: Crear Reportes Gerenciales**

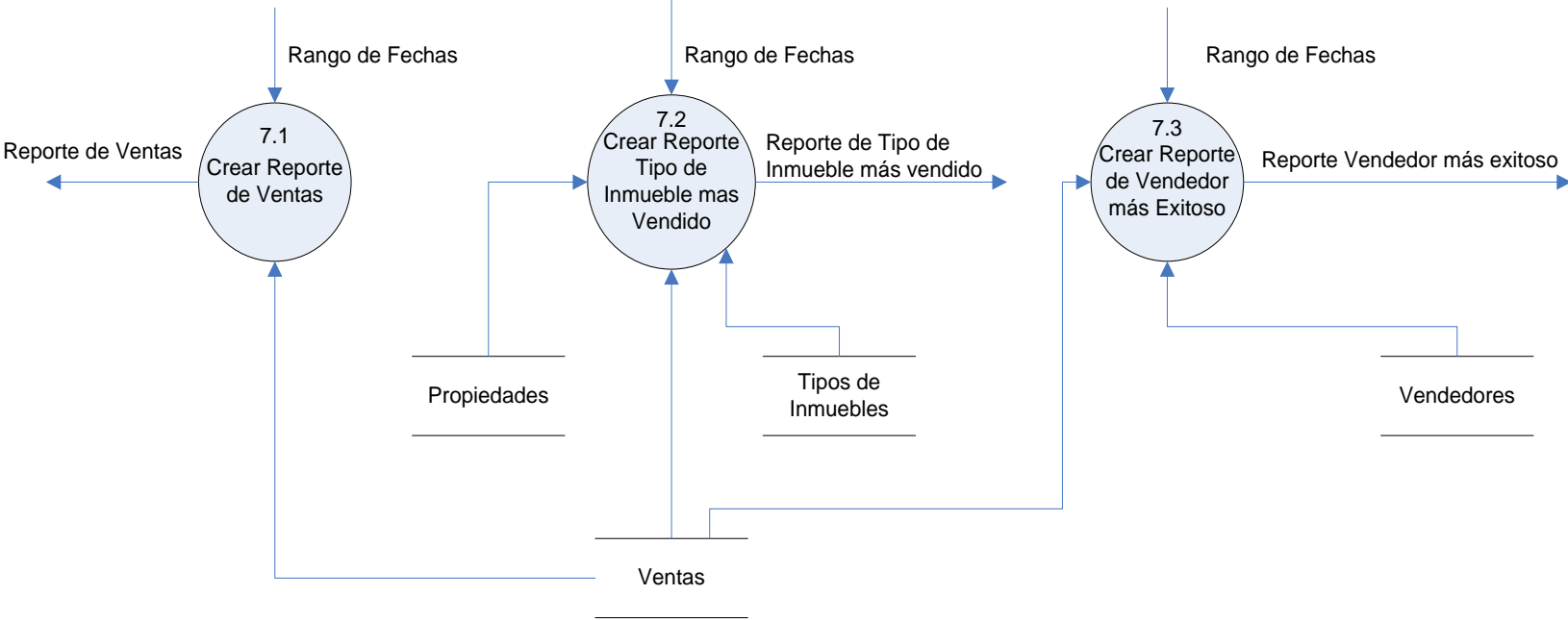


Figura 2.67: Diagrama a Nivel de Detalle del Proceso 7

**Proceso 8: Enviar Correos Electrónicos**

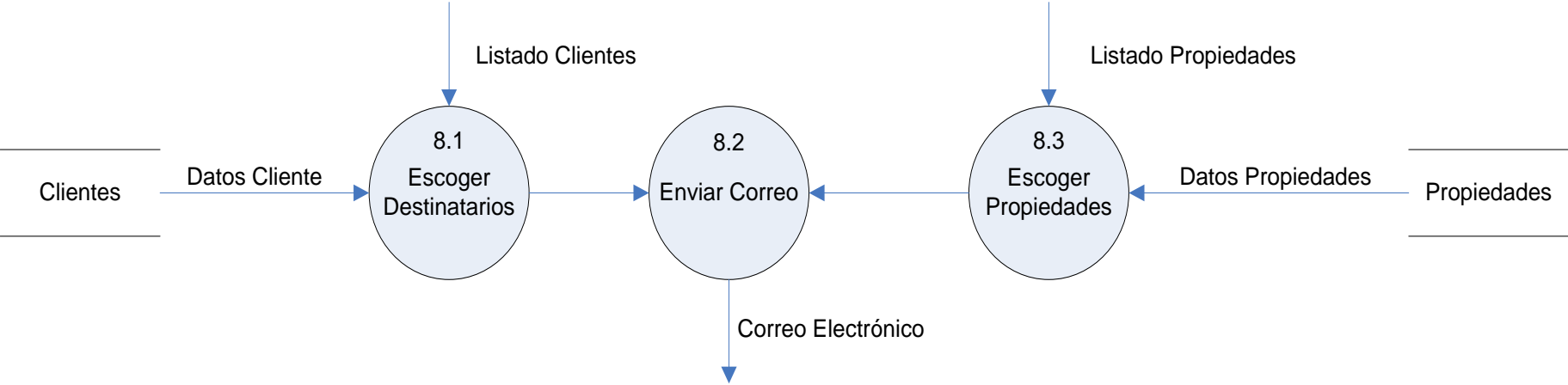


Figura 2.68: Diagrama a Nivel de Detalle del Proceso 8

**Proceso 9: Administrar Ventas**

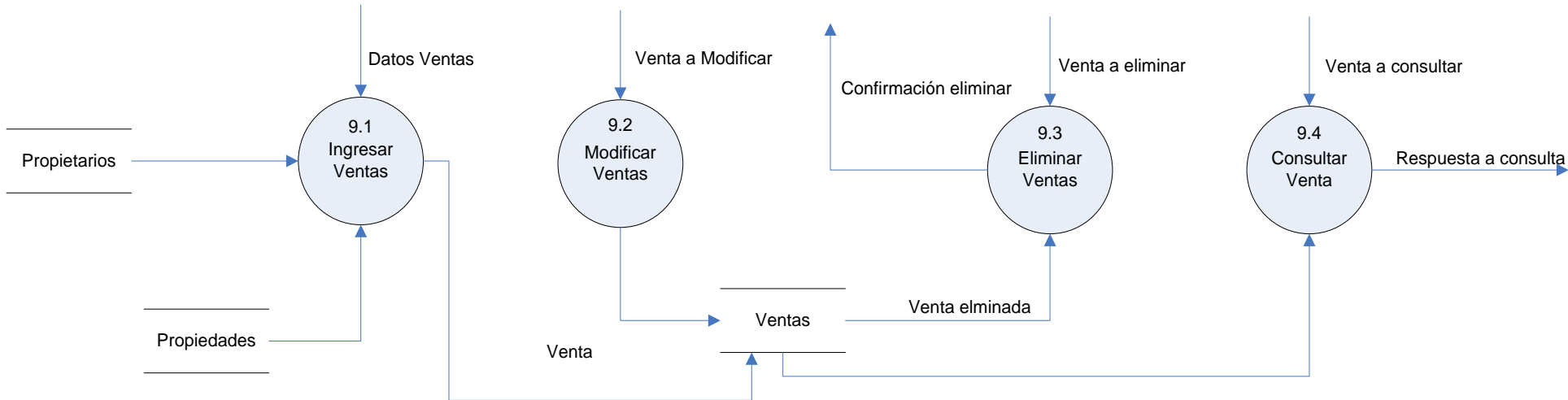


Figura 2.69: Diagrama a Nivel de Detalle del Proceso 9

**Proceso 10: Crear Archivos XML. No Requiere Refinamiento**

**Proceso 11: Mantenimiento de Clientes**

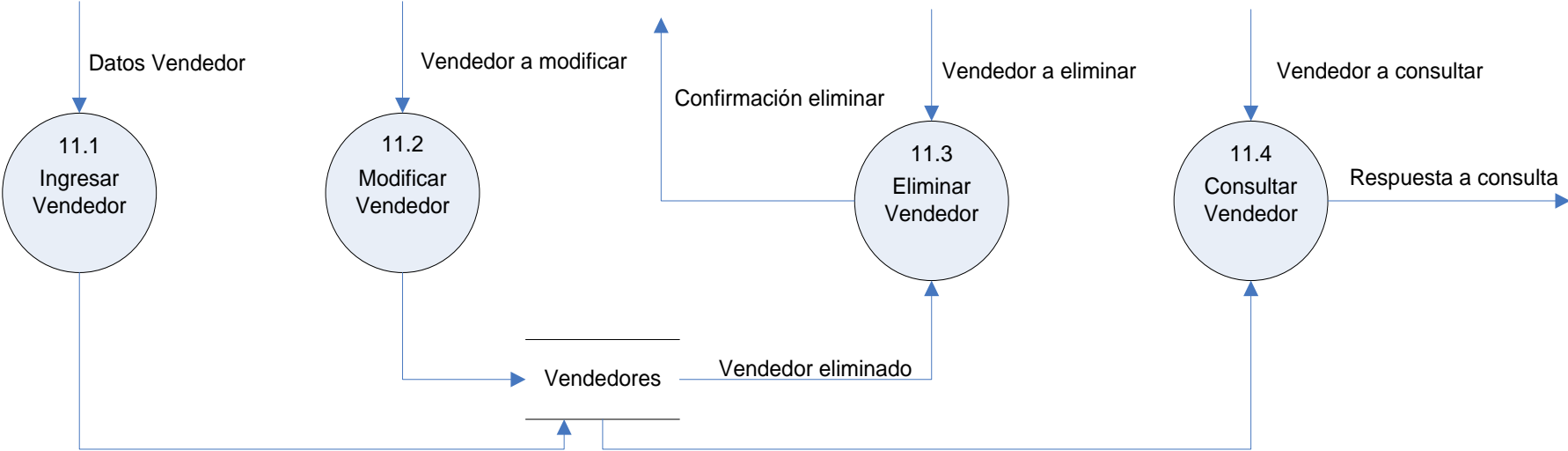


Figura 2.70: Diagrama a Nivel de Detalle del Proceso 11

**Proceso 12: Mantenimiento de Clientes**

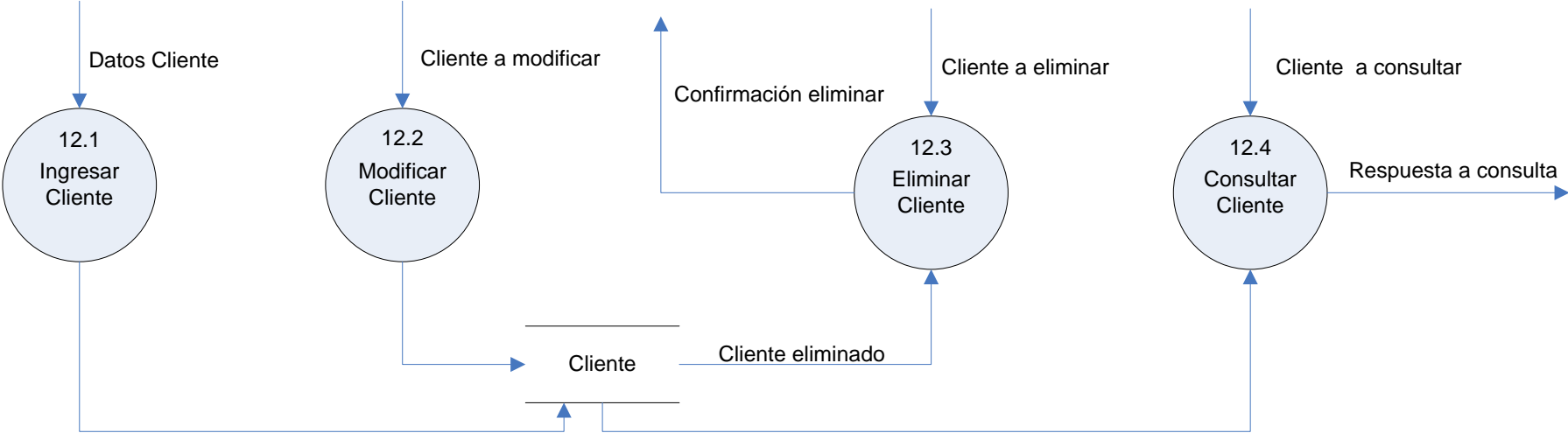


Figura 2.71: Diagrama a Nivel de Detalle del Proceso 12

**Proceso 13: Consultar Horarios. No Requiere Refinamiento.**



## Proceso 14: Descargar Características

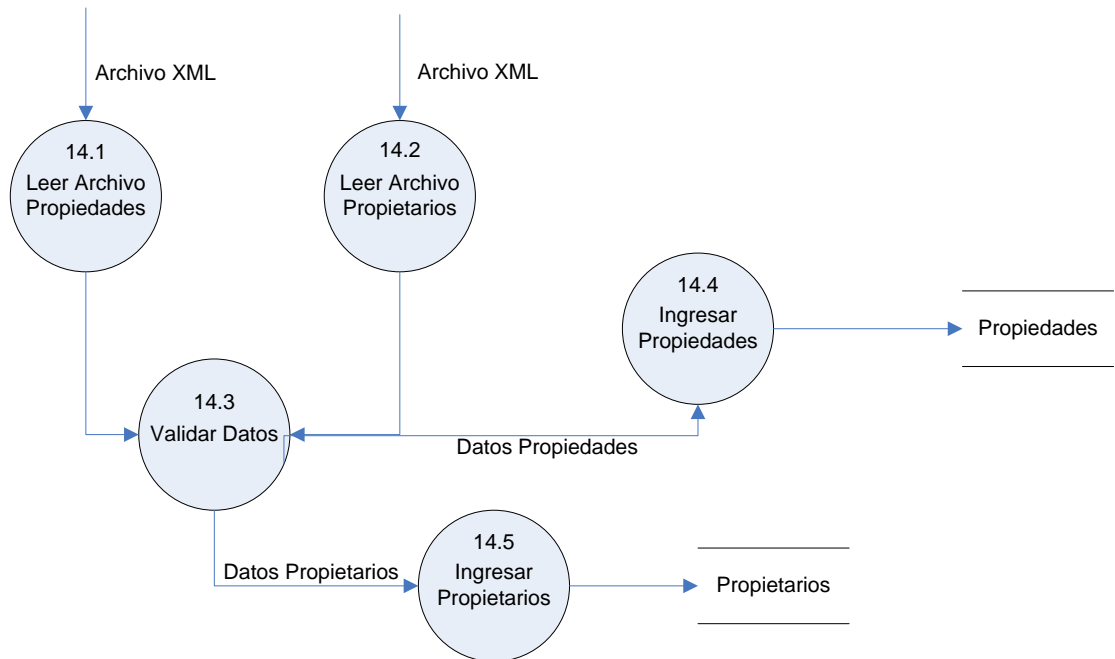


Figura 2.72: Diagrama a Nivel de Detalle del Proceso 14

## Proceso 15: Realizar Búsquedas

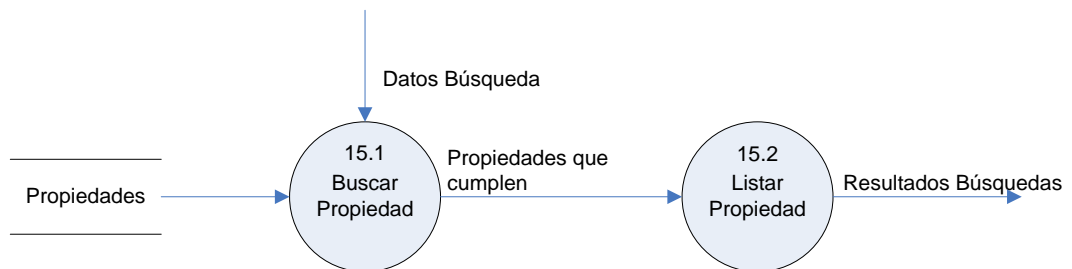


Figura 2.73: Diagrama a Nivel de Detalle del Proceso 15

## Proceso 16: Recoger Características

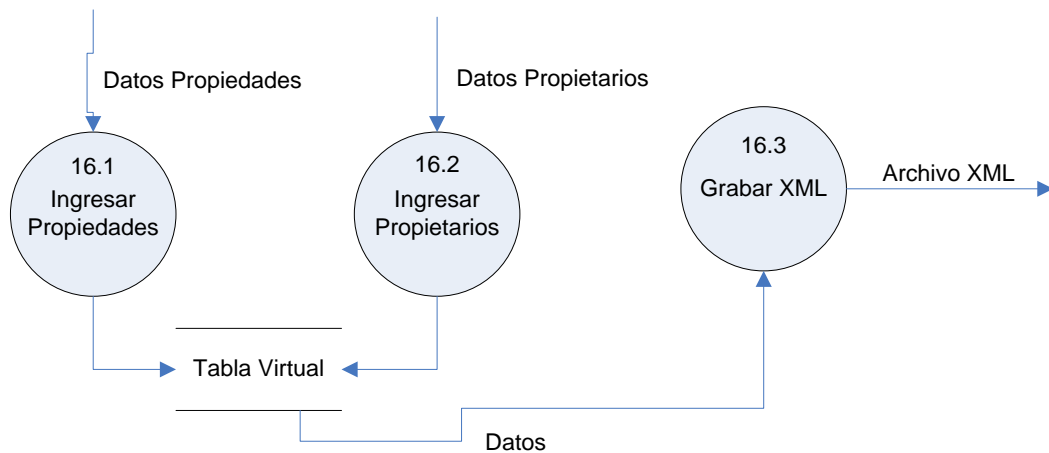


Figura 2.74: Diagrama a Nivel de Detalle del Proceso 16

### 2.2 Creación del Modelo Entidad-Relación

El Modelo Entidad – Relación (MER) es un método gráfico utilizado ampliamente para diseñar Bases de Datos, basándose en objetos del mundo real y las asociaciones existentes entre esos objetos. A los objetos se les denomina entidades, las cuales tienen propiedades que describen sus características. A estas propiedades se les llama atributos. A las asociaciones entre objetos se les conoce como relaciones.

Para realizar el MER del Sistema de Gestión Inmobiliaria, utilizamos una herramienta que permite la construcción de modelos entidad relación y realiza la normalización del mismo automáticamente. Este software se llama Power Designer versión 12.5, fue creado por la empresa Sybase y es muy útil al momento de realizar este tipo de diagramas.

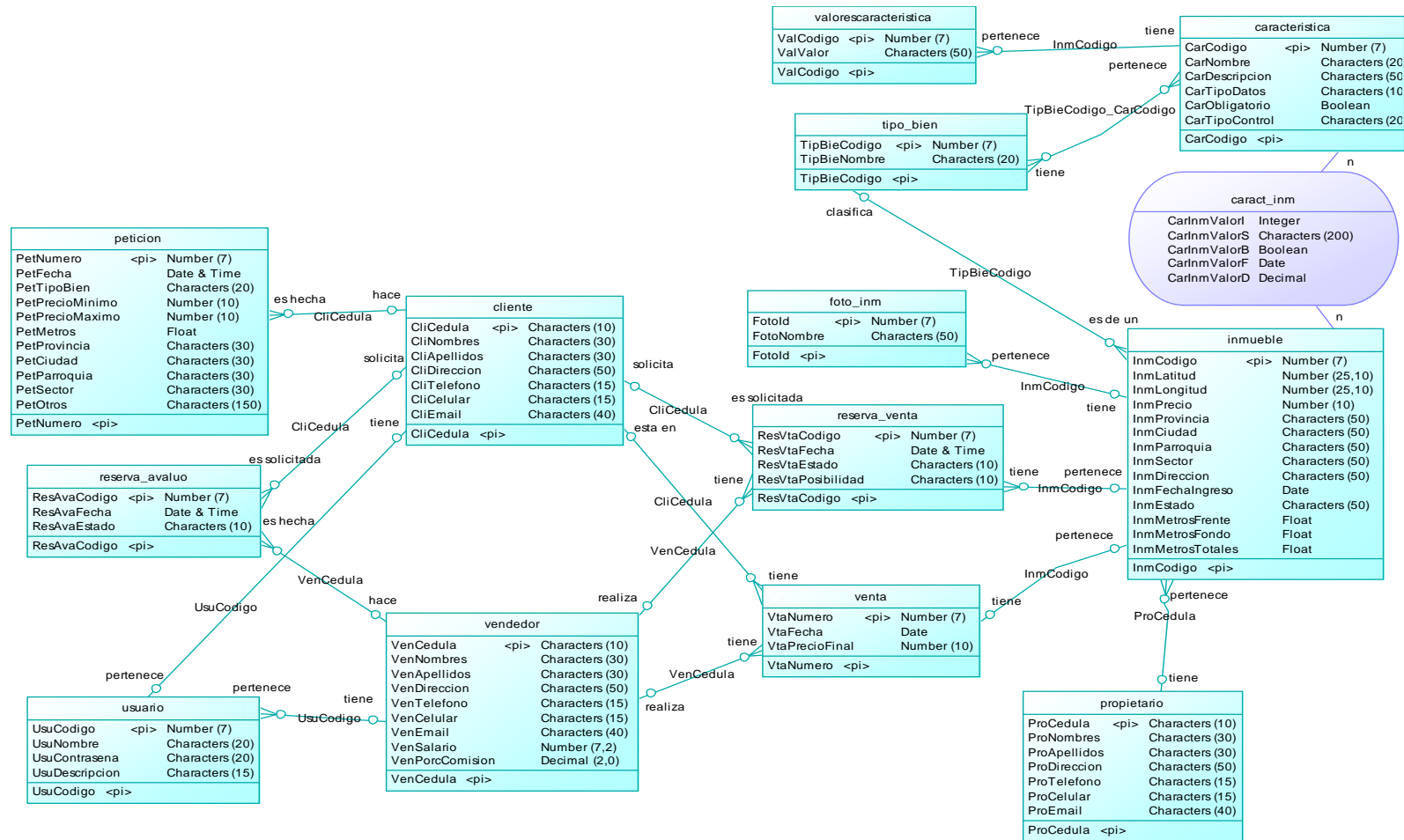


Figura 2.75: Diagrama Entidad - Relación del Sistema de Gestión Inmobiliaria

## 2.3 Normalización del Modelo

La Normalización del MER es un conjunto de reglas que sirven para desarrollar un esquema de Base de Datos, donde sea fácil la manipulación de los mismos, exista integridad referencial y se minimiza las posibilidades de que existan errores al momento de actualizar o eliminar los datos.

De la misma manera, se utilizó el software Power Designer para normalizar el modelo. Se ha elegido hacerlo en Tercera Forma Normal para evitar la redundancia de los datos. A continuación, mediante pantallas se va mostrar el sencillo proceso para normalizar el modelo.

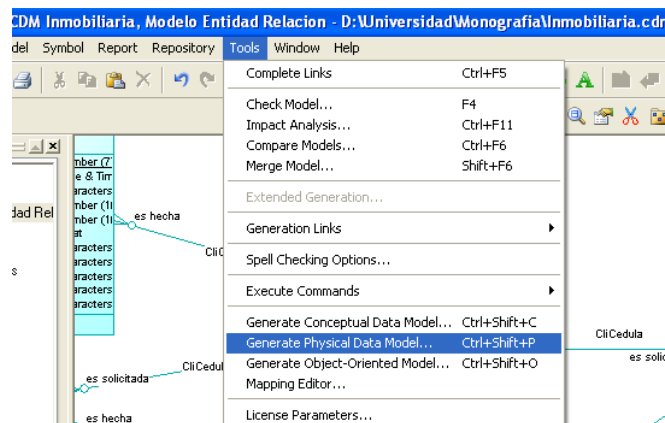


Figura 2.76: Paso 1 para normalizar un MER realizado en Power Designer

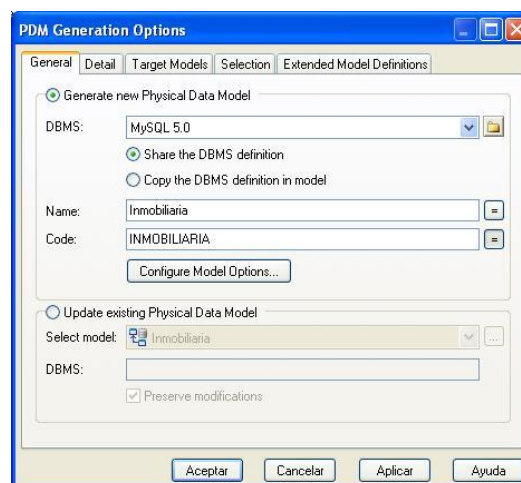


Figura 2.77: Paso 2 para normalizar un MER realizado en Power Designer

Lo que obtenemos, es el modelo normalizado donde como podemos ver se crean las tablas producto de las relaciones N - N. La punta de la flecha se dirige hacia la tabla de donde proviene la llave foránea.

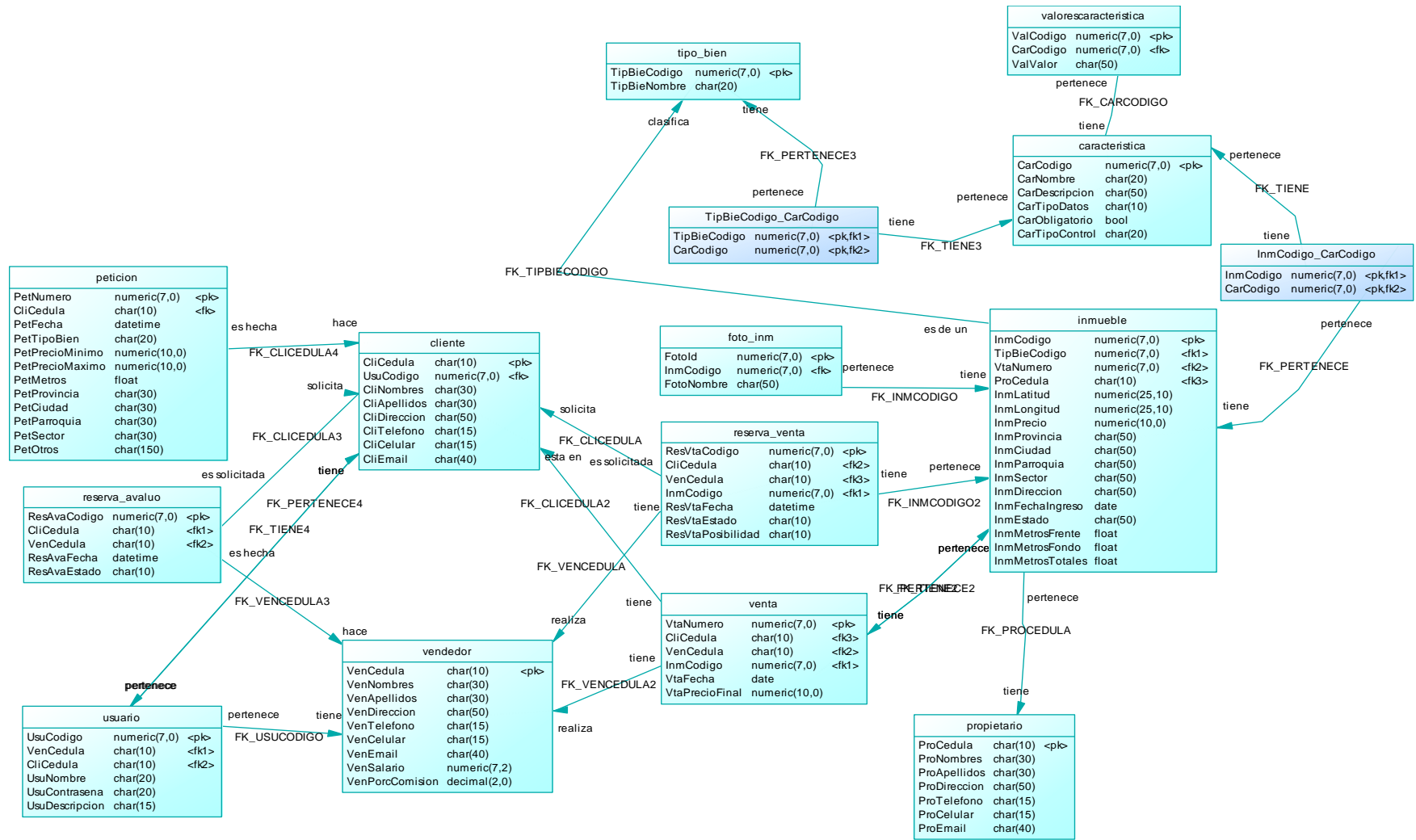


Figura 2.78: Modelo Normalizado

## 2.4 Definición de tablas e índices del esquema de base de datos

Una vez obtenido el Modelo Normalizado se procedió a la creación de la Base de Datos en MySQL 5.0. MySQL, que es un sistema de gestión de bases de datos relacionales. Este gestor de Base de Datos es Open Source distribuido por la empresa MySQL AB. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL.

Con el fin de una mejor administración del esquema de base de datos se necesita una interfaz gráfica de usuario. Se ha elegido para este propósito la herramienta SQLyog Enterprise que es un potente administrador de base de datos que proporciona una interfaz intuitiva y amigable para la interacción con la BD.

Las ventajas que se vieron en este software fueron:

- Interfaz Amigable e Intuitiva
- Facilidad de Instalación
- Fácil exportación e importación de la Base de Datos

La Base de Datos fue generada con PowerDesigner, y el script se ejecutó en SQLyog obteniendo el esquema de base de datos que se muestra a continuación:

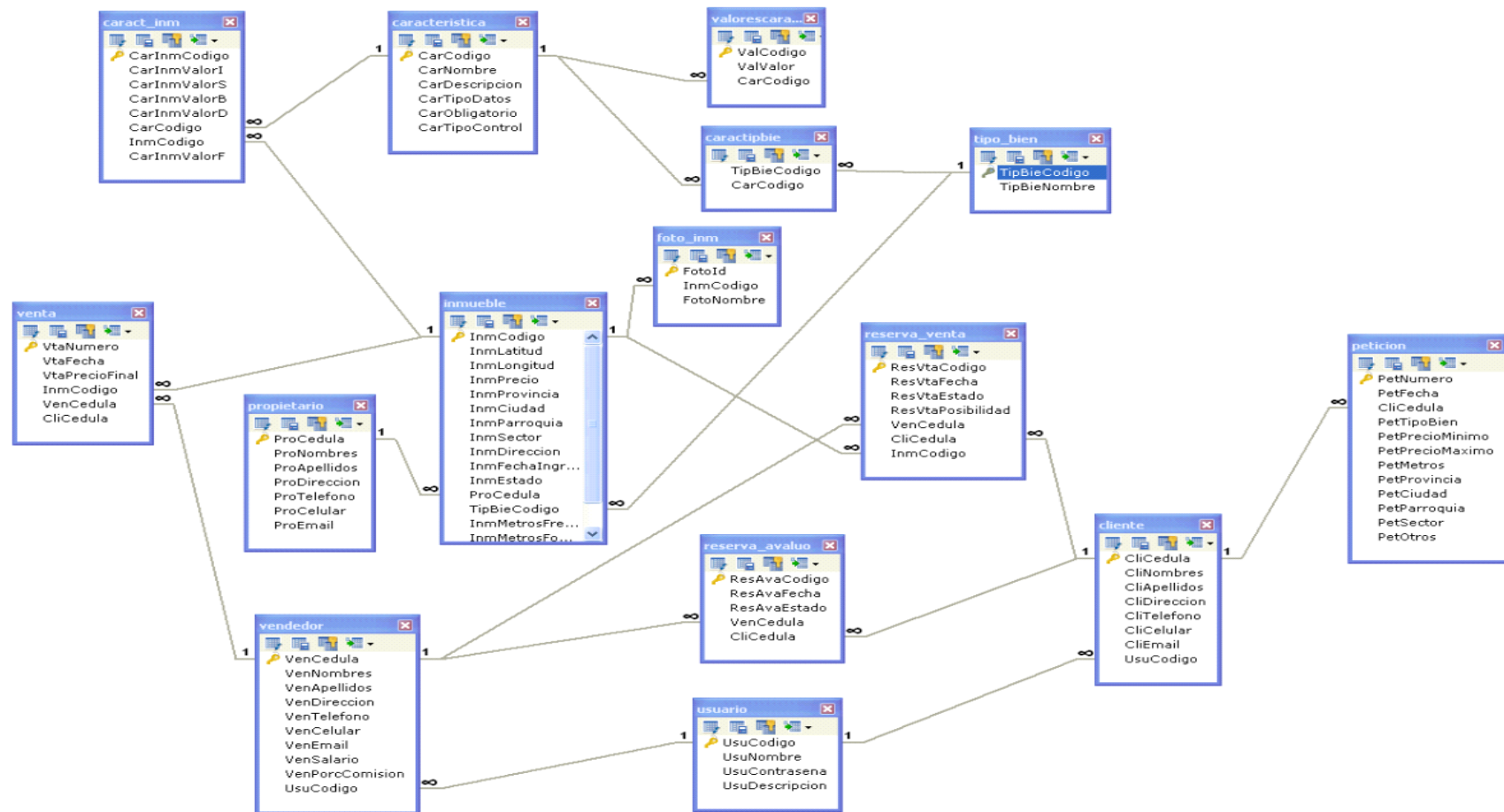


Figura 2.79: Esquema del Modelo de Base de Datos en SQLyog



En la figura de arriba se puede observar cuales son las tablas, llaves primarias y llaves foráneas del esquema. Las tablas fueron creadas con tipo InnoDB, este tipo de tabla en MySQL permite aplicar la integridad referencial.

Se ha visto la necesidad de definir índices únicos en las siguientes tablas:

TABLA	CAMPO INDICE
caracteristica	CarNombre
tipo_bien	TipBieNombre
Usuario	UsuNombre

Tabla 2.1: Índices Únicos

## 2.5 Diccionario de Datos

El Diccionario de Datos sirve para que los desarrolladores sepan la utilidad de cada tabla y sus respectivos campos de la Base de Datos.

SQLyog nos permite agregar una descripción breve de cada campo en la estructura de la tabla.

### Tabla: característica

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
CarCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código de la Característica. Llave Primaria
CarNombre	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombre de la Característica
CarDescripcion	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Descripción respecto a lo que contiene la característica
CarTipoDatos	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tipo de Datos que puede tener la característica
CarObligatorio	tinyint	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Si la Característica se aplicará a todos los tipos de Inmueble
CarTipoControl	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tipo de Control en el que se presentará la característica en pantalla

### Tabla: cliente

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
CliCedula	varchar	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cédula del Cliente. Llave Primaria
CliNombres	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombres del Cliente
CliApellidos	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Apellidos del Cliente
CliDireccion	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección del Domicilio del Cliente
CliTelefono	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Teléfono fijo del Cliente
CliCelular	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Celular del Cliente
CliEmail	varchar	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección de Correo Electrónico del Cliente
UsuCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Codigo de usuario. Llave Foránea

**Tabla: inmueble**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
InmCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código del Inmueble. Llave Primaria
InmLatitud	double	25,10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Latitud del Inmueble para ubicar en el mapa
InmLongitud	double	25,10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Longitud del Inmueble para ubicar en el mapa
InmPrecio	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Precio del Inmueble
InmProvincia	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Provincia donde está ubicado el Inmueble
InmCiudad	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Ciudad donde está ubicado el Inmueble
InmParroquia	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Parroquia donde está ubicado el Inmueble
InmSector	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sector referencia donde está ubicado el Inmueble
InmDireccion	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección exacta donde está ubicado el Inmueble.
InmFechaIngreso	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fecha desde la que el Inmueble es parte de las ofertas
InmEstado	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Estado en el que se encuentra el Inmueble para su promoción
ProCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cedula del Propietario del Inmueble
TipBieCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Clasificación del Inmueble
InmMetrosFrente	double			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Metros de Frente que posee el Inmueble
InmMetrosFondo	double			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Metros de Fondo que posee el Inmueble
InmMetrosTotales	double			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Metros Totales de Terreno que posee el Inmueble

**Tabla: petición**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
PetNumero	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Número de la Petición
PetFecha	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fecha en la que el cliente realizó la petición
CliCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cedula del Cliente que realizó la petición
PetTipoBien	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tipo de Bien que solicitó el Cliente en la petición
PetPrecioMinimo	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Precio mínimo que el cliente puede pagar
PetPrecioMaximo	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Precio máximo que el cliente puede pagar
PetMetros	double			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Número de metros que el cliente espera que tenga el inmueble
PetProvincia	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Provincia donde el cliente quiere que esté ubicado el inmueble
PetCiudad	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Ciudad donde el cliente quiere que esté ubicado el inmueble
PetParroquia	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Parroquia donde el cliente quiere que esté ubicado el inmueble
PetSector	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sector donde el cliente quiere que esté ubicado el inmueble
PetOtros	varchar	150		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Características Adicionales que el cliente especifica en su petición

**Tabla: propietario**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
ProCedula	varchar	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cédula del Propietario del Inmueble
ProNombres	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombres del Propietario del Inmueble
ProApellidos	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Apellidos del Propietario del Inmueble
ProDireccion	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección Exacta del Propietario del Inmueble
ProTelefono	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Teléfono fijo del Propietario del Inmueble
ProCelular	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Celular del Propietario del Inmueble
ProEmail	varchar	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección de Correo Electrónico del Propietario del Inmueble

**Tabla: reserva\_avaluo**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
ResAvaCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código del Avalúo
ResAvaFecha	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fecha en la que el cliente desea se le realice el avalúo
ResAvaEstado	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Estado de la Reserva para Avalúo. Realizada, Confirmada, Cancelada
VenCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cedula del Vendedor que realizó el avalúo
CliCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cedula del Cliente que solicitó el avalúo

**Tabla: reserva\_venta**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
ResVtaCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código de la Reservación
ResVtaFecha	datetime			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fecha en la que el cliente desea conocer el inmueble
ResVtaEstado	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Estado de la Reservación. Realizada, Corfirmada, Cancelada
ResVtaPosibilidad	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Posibilidad de venta del inmueble luego de que el cliente ha conocido
VenCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cedula del Vendedor que fue a la cita
CliCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cedula del Cliente que solicitó la reservación de la cita
InmCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Codigo del Inmueble que se solicitó conocer en la reservación

**Tabla: tipo\_bien**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
TipBieCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código del Tipo de Inmueble
TipBieNombre	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombre del Tipo de Inmueble

### Tabla: usuario

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
UsuCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código de Usuario
UsuNombre	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombre de Usuario especificado por el cliente, administrador o vendedor
UsuContrasena	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Contraseña especificada por el cliente, administrador o vendedor
UsuDescripcion	varchar	15		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tipo de Usuario. Administrador, Vendedor, Cliente.

### Tabla: valorecaracteristica

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
ValCodigo	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código del Valor de la Característica
ValValor	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Valor que puede tomar la característica si posee varias opciones
CarCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Código de la Característica

**Tabla: vendedor**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
VenCedula	varchar	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cédula del Vendedor
VenNombres	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombres del Vendedor
VenApellidos	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Apellidos del Vendedor
VenDireccion	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección del domicilio del Vendedor
VenTelefono	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Teléfono del Vendedor
VenCelular	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Celular del Vendedor
VenEmail	varchar	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dirección de correo electrónico del Vendedor
VenSalario	double	7,2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Salario fijo del Vendedor
VenPorcComision	decimal	2,0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Porcentaje de comisión del Vendedor
UsuCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Codigo de usuario del Vendedor

**Tabla: venta**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
VtaNumero	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Número de la Venta
VtaFecha	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fecha en la que se realizó la venta
VtaPrecioFinal	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Precio final en el que se vendió el Inmueble
InmCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Código del Inmueble
VenCedula	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cédula del Vendedor que realizó la venta
CliCedula	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cédula del Cliente que compró el Inmueble

**Tabla: caract\_inm**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
CarInmCodigo	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Valor para guardar cada característica del inmueble.
CarInmValorI	int	14		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Si la Característica es de Tipo Integer se guardará en este campo
CarInmValorS	varchar	200		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Si la Característica es de Tipo String se guardará en este campo
CarInmValorB	tinyint	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Si la Característica es de Tipo Boolean se guardará en este campo
CarInmValorD	double	14,2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Si la Característica es de Tipo Double se guardará en este campo
CarCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Código de la Característica
InmCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Código del Inmueble. Llave Foránea
CarInmValorF	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Si la Característica es de Tipo Fecha se guardará en este campo

**Tabla: caractipbie**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
TipBieCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Codigo del Tipo de Bien. Llave Foránea
CarCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Codigo de la Característica. Llave Foránea

**Tabla: foto\_inm**

Field Name	Datatype	Len	Defa	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
FotoId	int	7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identificador de la Foto. Llave Primaria
InmCodigo	int	7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Código del Inmueble al que hace referencia la fotografía
FotoNombre	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombre de la Fotografía



## 2.6 Definición del modelo de arquitectura

En esta sección vamos a describir la infraestructura que va a tener la aplicación. La Arquitectura de la mayoría de aplicaciones web se basa en el concepto de multicapa ya que este tipo de sistemas se distribuye en 3 niveles:

- **Nivel de Presentación:** en este nivel actúa el navegador como medio para presentar la interfaz de usuario.
- **Nivel de negocio:** en este nivel se realizan las operaciones necesarias para atender las peticiones del usuario.
- **Nivel de administración de datos:** este nivel hace referencia a la base de datos que suministra y almacena la información para el nivel de negocio

El Sistema de Gestión Inmobiliaria va a ser diseñado según esta lógica multicapa, por lo que se ha visto conveniente elegir la Arquitectura MVC (Modelo - Vista - Controlador). Esta arquitectura tiene 3 dimensiones principales, para encajar con el concepto de múltiples capas.

- La dimensión de Modelo que se corresponde con el Nivel de Administración de datos, es decir con la gestión de la información.
- La Vista es equivalente al Nivel de Presentación, donde se la aplicación va a interactuar con el usuario.
- El Controlador se refiere al nivel de negocio, que va a manejar el comportamiento del Sistema.

La Vista son las interfaces que se presentan al usuario, pueden existir varios tipos de vistas (pdf, web, presentaciones en flash) para un mismo modelo.

El Controlador realiza el procesamiento de la información para darle a conocer a través de las vistas. Se refiere al código fuente.

El Modelo contiene los datos de la aplicación.

Con estos conceptos podemos concluir que en nuestro Sistema la arquitectura MVC sería estructurada con los siguientes elementos:

**Vista:**

- Páginas ASPX
- Crystal Reports que se pueden exportar a formato PDF
- Formularios Smart Device para la aplicación móvil.

**Controlador:** Código fuente en Visual Basic.Net tanto para Web Forms como para Smart Device.

**Modelo:** Base de datos en MySQL administrada mediante SQLyog.

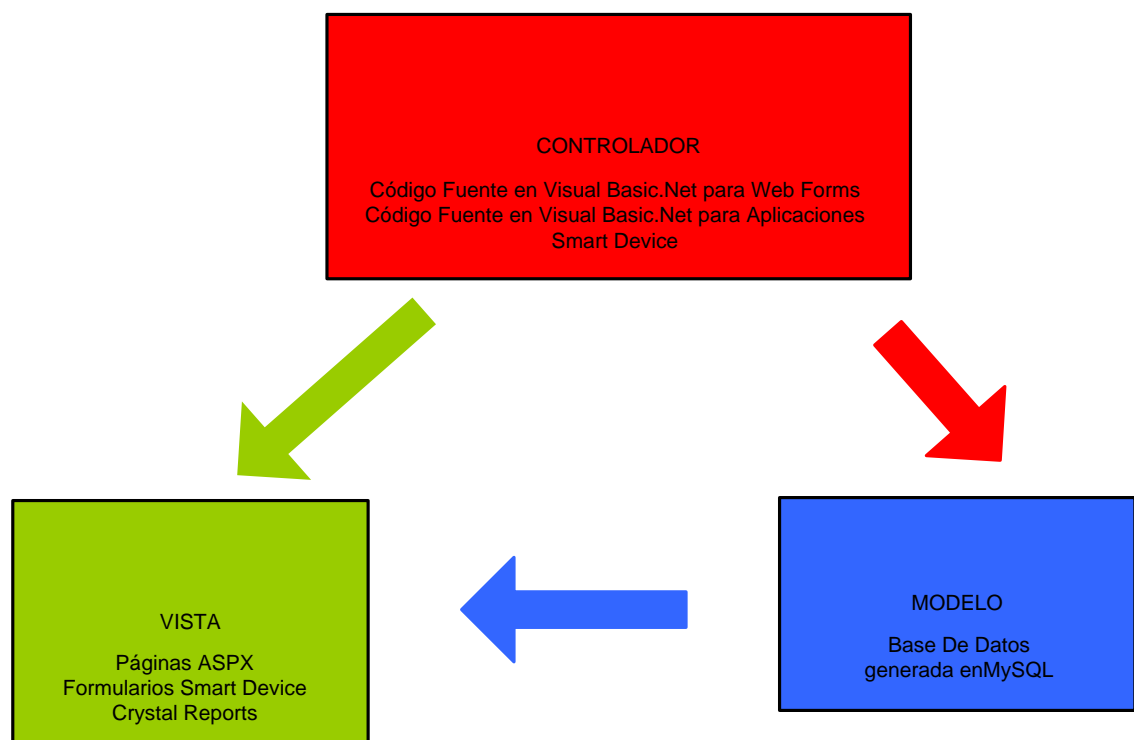


Figura 2.80: Arquitectura del Sistema de Gestión Inmobiliaria

## 2.7 Definición de los niveles de seguridad, privilegios y políticas del sistema

El sistema cuenta con 4 tipos de usuarios:

- Administrador
- Vendedor
- Cliente
- Visitante

El Administrador posee todos los privilegios y permisos necesarios para efectuar todas las transacciones dentro del sistema.

El Vendedor tiene acceso para cambiar sus datos personales, descargar archivos XML y consultar los horarios en los cuales tiene que ir a mostrar propiedades o realizar avalúos.

El Visitante puede observar las propiedades, revisar sus características, ubicarlas en el mapa y realizar búsquedas.

El Cliente puede realizar lo mismo que el visitante, además puede cambiar sus datos personales, efectuar reservas tanto para conocimiento de propiedades como para realización de avalúos de sus inmuebles y realizar solicitudes de inmuebles con características específicas.

El manejo de usuarios se hará de la siguiente manera: Al ingresar en el Sitio Web, el usuario será un visitante, si inicia sesión se verificará que tipo de usuario es (cliente, vendedor y administrador) y según esto se le habilitarán diferentes opciones de navegación.

Para mejorar la promoción de las propiedades es de suma importancia que el administrador ingrese como mínimo una foto por cada inmueble.

## **2.8 Diseño de Navegación**

El Diseño de Navegación se refiere a los vínculos que son parte de la estructura hipertexto de la Aplicación, como se van a organizar los contenidos en el Sitio Web de modo que sean fáciles e intuitivos de acceder por parte de los usuarios.

Existen 4 estructuras de contenido

- Estructura Lineal
- Estructura en Retícula
- Estructura Jerárquica

- Estructura en Red

Para el diseño del Sistema de Gestión Inmobiliaria hemos elegido la Estructura en Red, también conocida como WebPura donde los componentes arquitectónicos (páginas web) pueden pasar el control a otros componentes mediante enlaces (hipertextos). Con esta arquitectura se puede ir desde cualquier página hacia cualquier página que conforme el Sitio Web.

Se quiere lograr la estructura en red con un menú que siempre esté visible y permita flexibilidad al momento de navegar por la página.

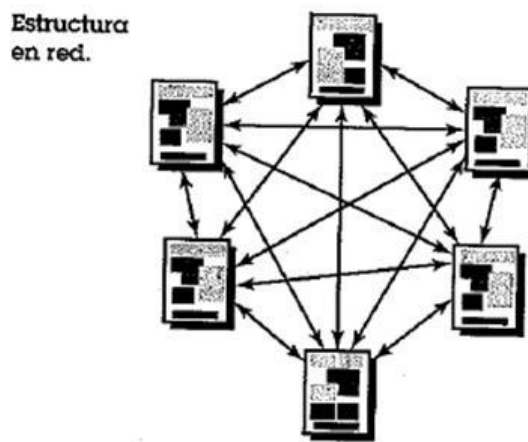


Figura 2.81: Dibujo de una Arquitectura de Contenido en Red

## 2.9 Creación de la interfaz y las plantillas a ser utilizadas en el Portal Web

### 2.9.1 Bosquejo General de la Interfaz

La interfaz del sistema se dividirá en 3 secciones principales como son la del Encabezado, el Menú y el Contenido, pero adicionalmente posee un Pie de Página y una sección para Iniciar Sesión, mostrar la Última Oferta y conocer la Ubicación General de la Empresa.

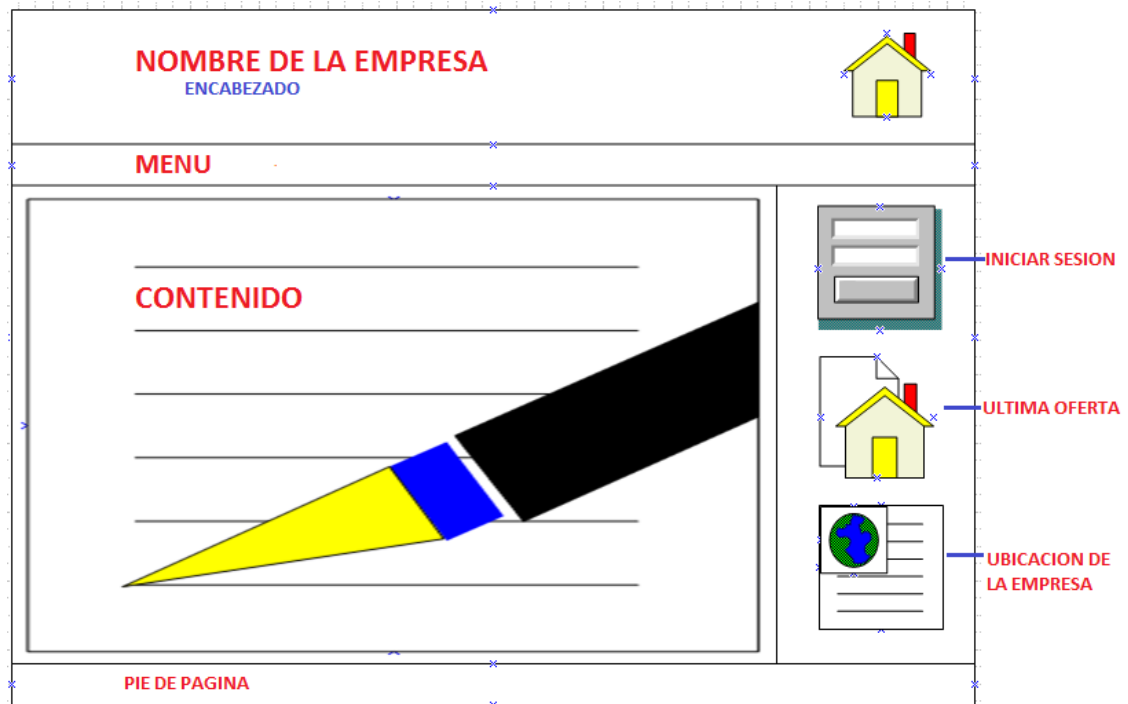


Figura 2.82: Bosquejo General de la Interfaz

Se utilizarán 3 Master Pages dependiendo del tipo de usuario que acceda al sistema, en donde la diferencia en sí son las opciones que existen en el menú. Los siguientes gráficos ilustran los links que proporcionará el Menú para cada usuario.

MENU DEL VISITANTE O CLIENTE				
Inicio	Ver Propiedades	Servicios Online	Búsquedas	Cuenta
	Quintas	Reservas para conocer Propiedad	Especializadas	Registrarse
	Departamentos	Reservas para solicitar Avalúos	Por Código	Modificar Datos
	Edificios	Solicitudes de Propiedades		
	Fincas			
	Bodegas			
	Locales Comerciales			
	Oficina			
	Terreno			
	Villa			
	Casa			

Figura 2.83: Menú del Visitante o Cliente

MENU DEL VENDEDOR		
Modificar Datos	Consultar Horarios	Móvil
		Importar
		Descargar

Figura 2.84: Menú del Vendedor

MENU DEL ADMINISTRADOR						
Inicio	Propiedades	Usuarios	Servicios Online	Móvil	Reportes	Ventas
	Tipo de Inmuebles	Vendedores	Reservas de Ventas	Importar	Ventas	
	Inmuebles	Clientes	Reservas de Avalúos	Descargar	Inmueble más vendido	
	Características		Solicitudes		Vendedor más exitoso	
	Propietarios					

Figura 2.85: Menú del Administrador

### 2.9.2 Diseño de Plantilla

Para el diseño de la Plantilla de las Master Pages se ha utilizado el software Artisteer, el cual es un generador de plantillas web que no requiere ningún conocimiento técnico, es muy fácil de utilizar y además de páginas HTML normales puede crear plantillas para Wordpress, Drupal, Joomla o aplicaciones web.

La plantilla en general quedaría de la siguiente manera:



Figura 2.86: Plantilla de la Aplicación Web

Para el mantenimiento de los datos que el administrador manejará dentro del sistema, existe una estructura de interfaz general, en la cual se muestran todos los datos en una grilla y una vez escogido uno de los registros se puede acceder a la modificación y eliminación de los mismos. También es posible a través de esta interfaz ingresar datos o buscarlos por un determinado filtro.

**INMOCAS**  
Encuentra El Inmueble De Tus Sueños

Inicio Propiedades Búsquedas Especializadas Servicios on Line Registrarse

**MANTENIMIENTO DE CLIENTES**

Criterio de Búsqueda: Cedula Valor:

	Cedula	Nombres	Apellidos	Direccion	Telefono	Celular	Email	Usuario	N
<a href="#">Seleccionar</a>	0	Roberto Carlos	Bravo Ramon	El Molino	4074151	098080591	robbrav@hotmail.com	32	
<a href="#">Seleccionar</a>	0100826775	Beatriz	Guerrero	General Torres	2823563	095324567	bety@hotmail.com	14	
<a href="#">Seleccionar</a>	0104321860	Claudia Pamela	Castro Guerrero	Gran Colombia 15-24	2830534	087136582	claupame_castroxjg@hotmail.com	30	cla
<a href="#">Seleccionar</a>	0104321868	Pamela	Castro	Gran	28384855	087136582		5	p
<a href="#">Seleccionar</a>	0104449061	Janina Patricia	Zuniga Solorzano	Calle del Mirto	2859173	084344422	janny1500@hotmail.com	36	
<a href="#">Seleccionar</a>	12	Cristian	Coronel	oris	234	234	j@hotmail.com	15	
<a href="#">Seleccionar</a>	6	Zoila Luisa	Fernandez Luo	Turuhuaioho 12-42	2878789			33	

1 2

**Usuarios Registrados**  
Iniciar sesión

**ULTIMAS OFERTAS**

**NUESTRA EMPRESA**  
Estamos en Gran Colombia 15-24 y Coronel Talbot Cuenca, Ecuador  
Email: [inmocas@hotmail.com](mailto:inmocas@hotmail.com)  
Telefono: (593) 072-830534

Contactenos | FAQ | Terminos y Servicios | Politicas de Privacidad  
Copyright © 2010 — Derechos Reservados

Figura 2.87: Formato para los Mantenimientos

## 2.10 Definición de la tecnología a ser utilizada para la integración del Sitio Web con la base de datos

La tecnología que se va a utilizar para enlazar la Base de Datos en MySQL con Microsoft Visual Studio 2008 es MySQL Connector/ODBC versión 5.1.6.

MySQL Connector/ODBC es una familia de controladores creados por MySQL para poder acceder a sus bases de datos usando el estándar ODBC (Open Database Connectivity - Conectividad de Base de Datos Abierta), que fue desarrollado por Microsoft, empresa creadora de Visual Studio 2008. Este estándar fue creado para que se pueda acceder desde cualquier aplicación a los datos sin importar el Gestor de Base de Datos que se esté usando. ODBC es el intermediario entre el DBMS y la aplicación, esto quiero decir que la aplicación utiliza los comandos ODBC para acceder a los datos y el Gestor debe responder a las consultas satisfactoriamente.



Para poder utilizar MySQL Connector/ODBC v.5.1.6 se debe realizar el siguiente proceso, luego del mismo podremos utilizar durante la codificación, los comandos ODBC para acceder a los datos almacenados en la BD Inmobiliaria. Entramos en Panel de Control -> Herramientas Administrativas -> Orígenes de Datos (ODBC) -> Pestaña DNS del Sistema -> Pulsamos Agregar y se debe desplegar la siguiente pantalla:

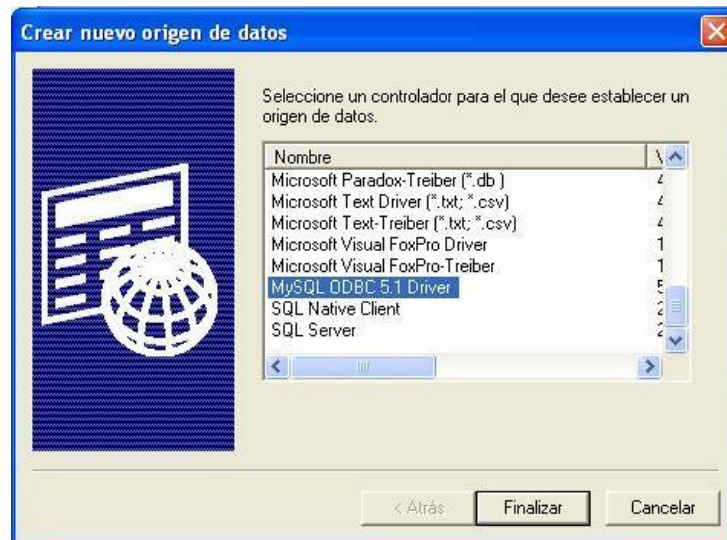


Figura 2.88: Paso 1 para habilitar MySQL Connector/ODBC v.5.1.6

Escoger la opción señalada con azul y a continuación pulsar Finalizar y se mostrará la siguiente pantalla:

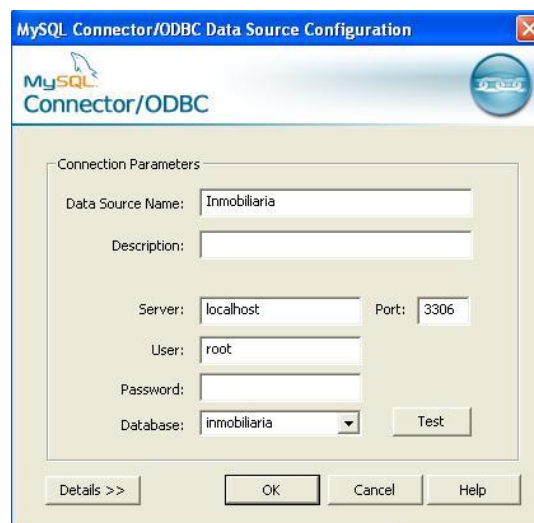


Figura 2.89: Paso 2 para habilitar MySQL Connector/ODBC v.5.1.6

Se mostrará esta pantalla con el origen de datos ya agregado. Este sería el proceso para que la aplicación reconozca los comandos ODBC y se puede utilizar el lenguaje SQL para realizar consultas a la Base de Datos.



Figura 2.90: Pantalla que nos muestra el origen de datos ya agregado

Adicionalmente, cuando realizamos la conexión a la Base de Datos desde Microsoft Visual Studio se debe utilizar MySQL Connector/NET, que es una herramienta específicamente para la integración de Lenguajes .NET con una Base de Datos MySQL.

Esta herramienta permitirá que el Sistema de Gestión Inmobiliaria, que es una aplicación .NET tenga una conexión segura y de alto rendimiento con la Base de Datos Inmobiliaria que esta generada en MySQL.

MySQL Connector/NET v.6.2.2 es un administrador de ADO.NET escrito en C#.

## CAPITULO 3

### CODIFICACIÓN

#### 3.1 Referencia sobre las Herramientas utilizadas para la Codificación

Microsoft Visual Studio 2008 es un conjunto de herramientas integradas que ofrece crear soluciones en los siguientes lenguajes de programación:

- Visual C#
- Visual C++
- Visual Basic

Se pueden crear aplicaciones Web ASP.NET, Servicios Web XML, Aplicaciones de Escritorio y aplicaciones móviles. Todos los lenguajes utilizan el mismo entorno de desarrollo integrado. También ofrece complementos para Office 2007 y 2003. Se pueden crear reportes e informes por medio de Crystal Reports o Aplicación de Informes y además soporta Microsoft.Net Framework 2.0, 3.0 y 3.5.

Una de las principales ventajas de Visual Studio es que permite a los desarrolladores construir aplicaciones con funciones integradas de diseño y programación, permitiendo reducir el número de líneas de programación ingresadas por el desarrollador, además de la fácil integración con otros lenguajes de programación.

De todos los lenguajes de Visual Studio hemos escogido Visual Basic para desarrollar nuestro proyecto debido a que es una evolución de Visual Basic 6.0, que ya era conocido por nosotras anteriormente, por lo que es más fácil la familiarización con su sintaxis. Al ser un lenguaje que soporta Web y está orientado a objetos nos brinda la oportunidad de desarrollar una aplicación que va de acuerdo con las necesidades que actualmente expresan los clientes.

Visual Studio proporciona la compatibilidad para el desarrollo de aplicaciones que se pueden ejecutar en dispositivos inteligentes tanto con Visual C# como con Visual

Basic, en nuestro caso escogimos la última opción conocida como Proyecto Smart Device cuya plataforma de destino es Windows Mobile 5.0 Pocket PC SDK compatible con Net Framework 3.5, para poder implementar la parte móvil de nuestro proyecto que consiste en la recolección de información de las propiedades por parte del Vendedor y su posterior descarga en la Base de Datos en la aplicación Web.

Visual Studio ofrece la posibilidad de emular un dispositivo móvil para la realización de las pruebas.

Para el cumplimiento de los siguientes objetivos: envío de emails a los clientes y la publicación de la ubicación de las propiedades en Google Maps dentro de la aplicación se necesita la importación de la librería System.Net.Mail en el primer caso y la biblioteca de enlace dinámico Gmaps.dll en el segundo caso.

### 3.2 Estándares

Para mayor facilidad de mantenimiento de la aplicación se ha decidido estandarizar los nombres de los controles:

CONTROL	ESTANDAR
TextBox	Txt & complemento
DropDownList	Cbo & complemento
Button	Btn & complemento
Calendar	Cld & complemento
Controles de Validacion	Validator & complemento
Radio Button	Radio & complemento
FileUpload	Upload & complemento

Tabla 3.1 Estándares para controles

Todos los complementos tienen que ser nombres nemotécnicos que hagan referencia a la función que cumplirá el control dentro del formulario.

### 3.3 Procesos Relevantes de la Codificación

#### 3.3.1 Lineamientos Generales para los Mantenimientos

La forma de realizar los mantenimientos tiene un estándar común. En cada mantenimiento la primera pantalla que se va desplegar es la siguiente:



Figura 3.1: Mantenimiento de Clientes Pantalla Inicial

En la imagen podemos observar un listado y las siguientes opciones:

- **Buscar:** se busca según el criterio de búsqueda que elija el usuario. Los criterios de búsqueda se presentan en un combo y en un texto se ingresa el valor a buscar. En algunos casos la consulta SQL se realiza con el comando LIKE.
- **Nuevo y Modificar:** Estas opciones enlazan hacia otra pantalla donde según el tipo de operación (Registro Nuevo o Modificar Registro) se manejará el formulario. Si el tipo de operación es Registro Nuevo el formulario se presentará con los controles en blanco para que el usuario los llene y guarde el registro. Si el tipo de operación es Modificar Registro primero se debe seleccionar un registro del listado y los datos de este se mostrarán en los controles y el usuario podrá modificar los valores como desee.
- **Eliminar:** para eliminar, el usuario debe previamente seleccionar un registro del listado. Cuando pulse el botón de eliminar se enviará un mensaje de confirmación que en caso de ser aceptado hará que el sistema elimine el registro.
- **Refrescar:** mostrará el listado original sin los filtros de las búsquedas.

Para validar los formularios se han utilizado los controles de validación:

- **RequiredFieldValidator** para campos que se quieren sean obligatorios.
- **RangeValidator** para determinar un rango de entrada para el campo.
- **RegularExpressionValidator** donde se ha utilizado expresiones regulares para campos que puedan admitir solo letras, solo números o se trate de un correo electrónico.

The image shows a web form titled "INGRESO DE PROPIETARIOS" with a home icon. The form contains the following fields and labels:

Cédula	<input type="text"/>	Obligatorio
Nombres	<input type="text"/>	Obligatorio
Apellidos	<input type="text"/>	Obligatorio
Dirección	<input type="text"/>	Obligatorio
Teléfono	<input type="text"/>	Obligatorio
Celular	<input type="text"/>	
Email	<input type="text"/>	Obligatorio

Below the form, there is a list of validation messages:

- > La Cédula es Obligatoria
- > El Nombre es Obligatorio
- > El Apellido es Obligatorio
- > La Dirección es Obligatoria
- > El Teléfono es Obligatorio
- > El Mail es Obligatorio

Figura 3.2: Formulario de Ingreso y Modificación

INGRESO DE PROPIETARIOS

Cédula: xxxx Solo admite numeros

Nombres: 999 Solo admite letras

Apellidos: 999 Solo admite letras

Dirección:

Teléfono: xxx Solo admite numeros

Celular: xxx Solo admite numeros

Email: xxx Mail Inválido

Figura 3.3: Formulario de Ingreso y Modificación Validando Datos

### 3.3.2 Ubicación Geográfica en el Mapa

La librería System.Net.Mail ya es parte de Visual Studio, pero para Google Maps es necesario descargarse de la Web el archivo Gmaps.dll y luego seguir el siguiente proceso:

1. En el cuadro de herramientas dar un clic derecho, Agregar Ficha y le ponemos el nombre de Google.

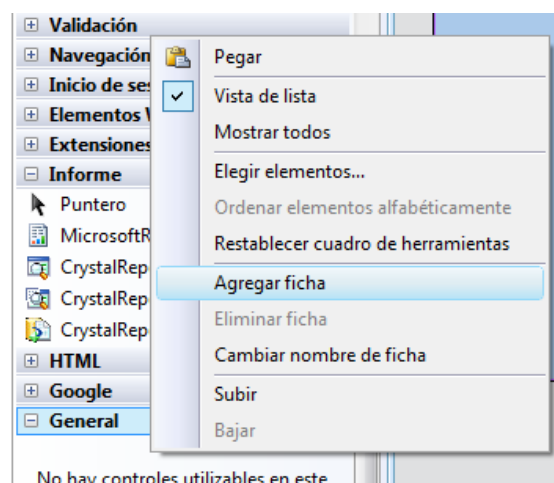


Figura 3.4: Paso 1 para agregar el Componente de Google Maps

2. Dentro de la nueva ficha creada dar clic derecho y escoger elegir elementos.

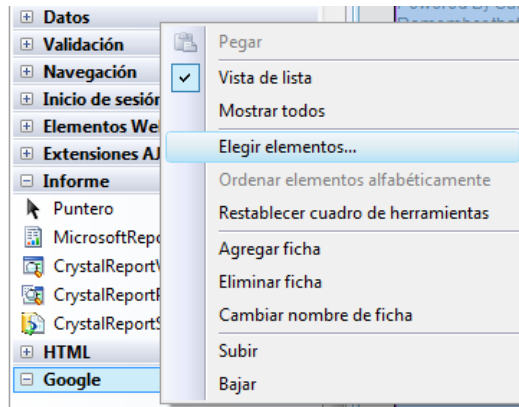


Figura 3.5: Paso 2 para agregar el Componente de Google Maps

3. Pulsamos el botón examinar, buscamos el archivo Gmaps.dll y presionamos Abrir.

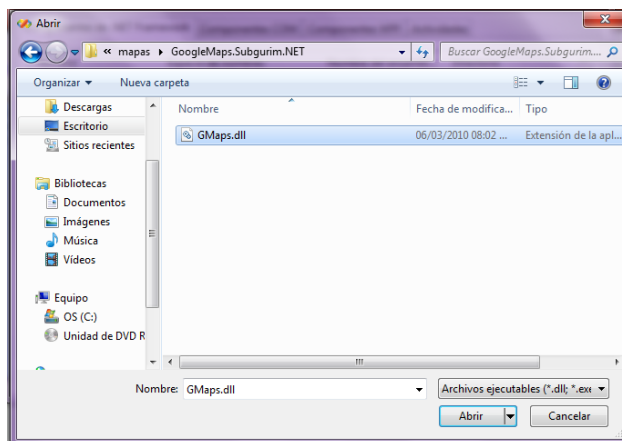


Figura 3.6: Paso 3 para agregar el Componente de Google Maps

4. Como podemos ver se agregaron los controles para la manipulación de los mapas con Google Maps.

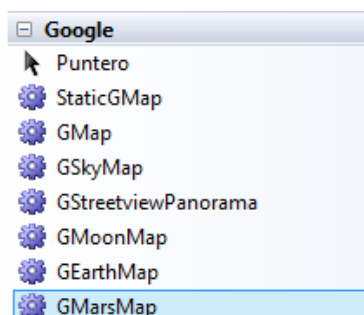


Figura 3.7: Paso 4 para agregar el Componente de Google Maps



5. Debemos asegurarnos de tener una conexión a internet y luego arrastrar al formulario un control GMap al cual llamaremos GMapPrueba.

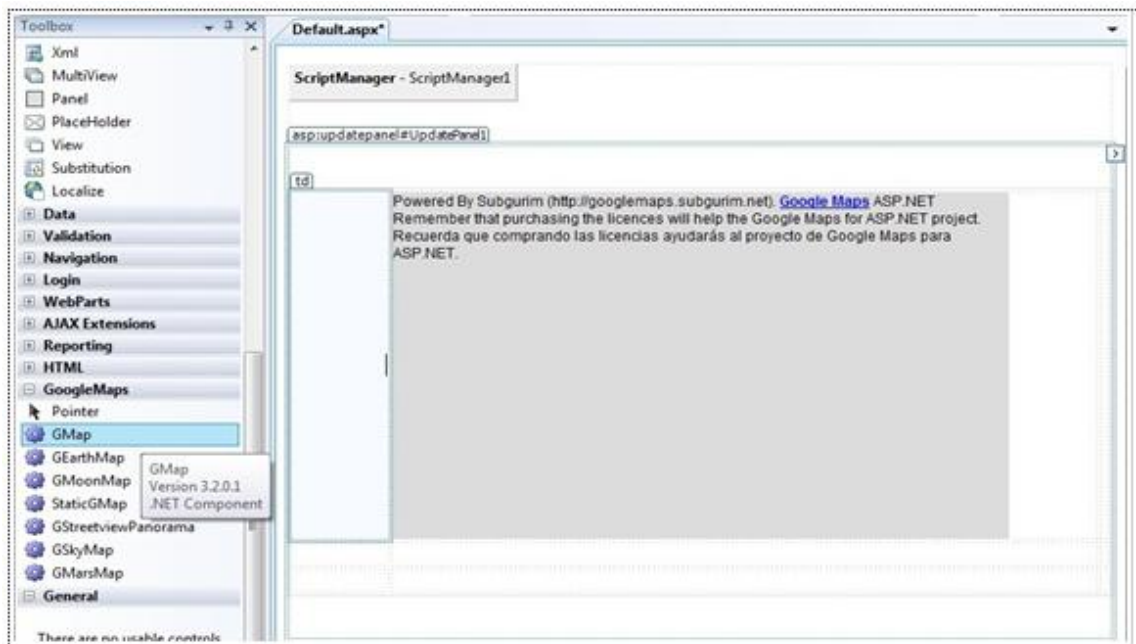


Figura 3.8: Paso 5 para agregar el Componente de Google Maps

6. Corremos el formulario pero nos da un error porque no tenemos una Key de Google Maps.

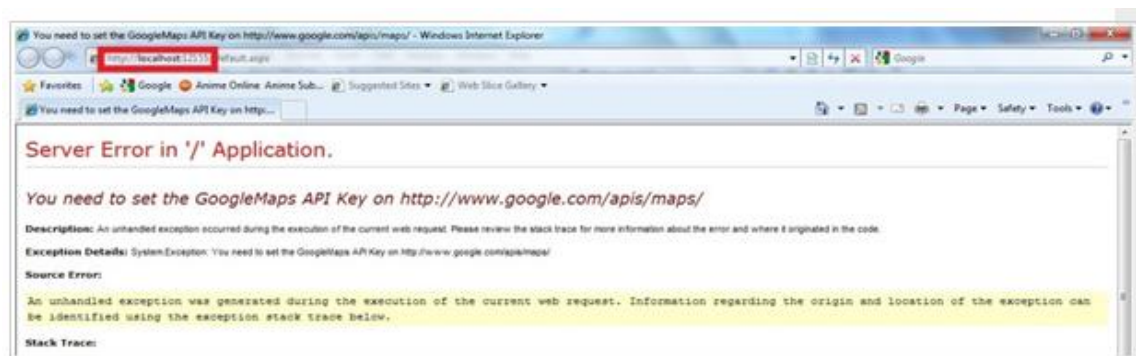


Figura 3.9: Paso 6 para agregar el Componente de Google Maps

7. Ingresamos a <http://code.google.com/intl/es/apis/maps/signup.html>, en la parte inferior hacemos clic en el check para aceptar los términos y en el recuadro de la dirección de URL colocar <http://localhost:12155>, por último presionamos el botón Generar Clave de API.



Figura 3.10: Paso 7 para agregar el Componente de Google Maps

8. Copiamos la clave que sale en el primer recuadro

## Google Maps API - API Key Signup



Figura 3.11: Paso 8 para agregar el Componente de Google Maps

9. Abrimos el web.config del proyecto actual y en la parte de appSettings y colocar la clave obtenida

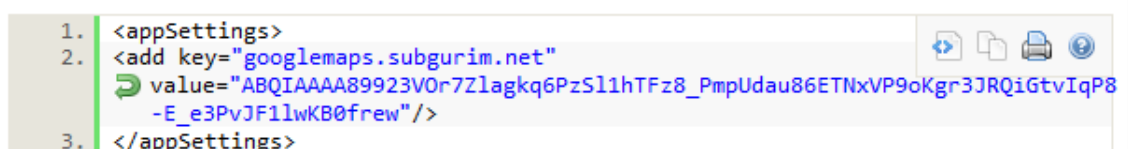


Figura 3.12: Paso 9 para agregar el Componente de Google Maps

10. Al correr el proyecto se mostrará por defecto España

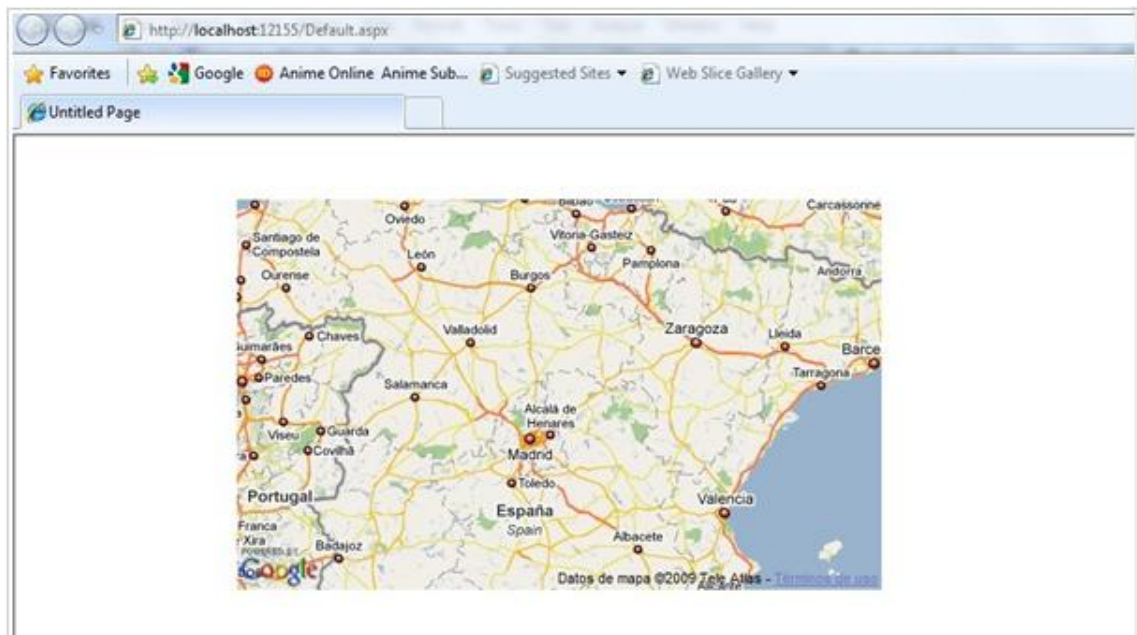


Figura 3.13: Mapa de España en Google Maps

11. Para cambiar la ubicación a una coordenada específica se deben importar la siguientes librerías:

```
Imports System
Imports System.Collections
Imports System.Configuration
Imports System.Data
Imports System.Linq
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.HtmlControls
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Xml.Linq
Imports Subgurim.Controles
```

12. Dentro de la clase del formulario actual poner el siguiente código:

```
Inherits System.Web.UI.Page

Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
```

```

    If IsPostBack Then
    Else
        Mostrar()
    End If
End Sub

Public Sub Mostrar()
    Dim Latitud, Longitud As Double
    Latitud = Request.QueryString.Item("Lat")
    Longitud = Request.QueryString.Item("Long")
    GMapPrueba.Height = 500
    GMapPrueba.Width = 500
    GMapPrueba.setCenter(New GLatLng(Latitud, Longitud), 17)
    Dim punto = New GLatLng(Latitud, Longitud)
    Dim marca = New GMarker(punto)
    GMapPrueba.enableHookMouseWheelToZoom = True
    GMapPrueba.enableGKeyboardHandler = True
    GMapPrueba.enableGoogleBar = True
    GMapPrueba.enableScrollWheelZoom = True
    GMapPrueba.EnableViewState = True
    GMapPrueba.addGMarker(marca)
    GMapPrueba.addKeyDragZoom(New KeyDragZoom)
End Sub

```

En donde, las variables Latitud y Longitud son pasadas como parámetros de Javascript. El número 17 del comando setCenter indica el zoom a mostrar y la marca muestra una imagen en el punto exacto de acuerdo a las variables de Latitud y Longitud. El resto de comandos añaden características adicionales que pueden ser suprimidas sin afectar al resultado final.

### 3.3.3 Características Configurables

Nuestro proyecto se enfoca en la creación de características configurables, las mismas que nos sirven para mostrar en pantalla solo las características según el tipo de inmueble, ya que no todos los tipos de inmuebles que existen en el mercado poseen las mismas características. El funcionamiento de las características configurables se da de esta manera:

Al ingresar una característica el Administrador debe especificar si la misma es obligatoria para todos los tipos de inmueble; en caso de no ser así debe indicar a que tipos de inmueble pertenece la característica, el tipo de dato que se va a manejar y el tipo de control. Si la característica tiene varias opciones para escoger, el Administrador tiene que ingresar cuáles serán.

Ej., ingresamos la característica Tipo de Terreno, se debe indicar que tipo de terrenos hay (Residencial, Comercial, Macrolotes), etc. La característica Sala puede tomar dos valores SI o NO.

Se están manejando 4 tipos de datos: Integer, Double, Boolean y String, y 4 tipos de controles Texto, Lista Desplegable, Radio Botones, Casillas de Verificación. Cuando el Administrador ingresa un inmueble, en primera instancia se le va a presentar la opción para escoger el tipo de inmueble y según su elección se le desplegará en pantalla todas las características que ese tipo de inmueble puede tener.

Cada característica está asociada a un tipo de control, el mismo que se escogerá según las necesidades de cada característica, es decir, si la característica tiene varias opciones y solo se puede escoger una, se utilizará una lista desplegable o radio botones. Si está permitido seleccionar varias opciones se utilizan casillas de verificación. En caso de que el usuario deba digitar el valor que va a tomar la característica se mostrará un texto.

Al tener 4 tipos de datos y 4 tipos de controles, se obtienen 16 combinaciones, estas combinaciones servirán para que al momento de guardar en la BD, se almacene según el tipo de dato de la característica. La Tabla que manejará esto es `caract_inm` fruto de la relación N - N entre inmuebles y características cuyos campos son:

`CarInmCodigo`

`CarInmValorI` (características de tipo entero)

`CarInmValorD` (características de tipo doble)

`CarInmValorB` (características de tipo booleano)

`CarInmValorS` (características de tipo cadena)

`InmCodigo`

`CarCodigo`

Se ha estructurado la tabla de esta manera, para que el Administrador tenga flexibilidad al ingresar las características según vayan surgiendo.

Es importante recalcar, que así como se manejan características dinámicas, el inmueble también tiene características estáticas, datos que obligatoriamente todo inmueble debe tener, como son: latitud, longitud, precio, provincia, ciudad, parroquia, sector, dirección, fecha de ingreso, cédula del propietario, estado, metros de frente, metros de fondo y metros totales del terreno.

```
Protected Sub btnGuardar0_Click(ByVal sender As Object, ByVal e As System.Web.UI.ImageClickEventArgs) Handles btnGuardar0.Click
```

```
    Dim tbl_inmueble As New DataSetInmobiliariaTableAdapters.inmuebleTableAdapter
    Dim tbl_caracinm As New DataSetInmobiliariaTableAdapters.caract_inmTableAdapter
    Dim codcarc, codinm As Integer
    Dim ingreso As New DataSet
    Dim tipo_dato, tipo_control As String
    Dim i, j As Integer
    Dim fecha As Date
    Dim valor_booleano As Byte
    Dim inmueble As New DataSet

    conexion.conectar()
    tbl_inmueble.Insert(Me.TxtLatitud.Text, Me.TxtLongitud.Text, Me.TxtPrecio.Text, Me.CboProvincia.SelectedValue, Me.TxtCiudad.Text, Me.TxtParroquia.Text, Me.TxtSector.Text, Me.TxtDireccion.Text, Me.CalFechaIngreso.SelectedDate, Me.CboEstado.SelectedValue, CboPropietario.SelectedValue, tipoInmueble)

    conexion.conectar()
    ingreso.Clear()
    Comando = New Odbc.OdbcCommand("SELECT max(InmCodigo) cod from inmueble", conexion.conexionBD)
    Adaptador = New Odbc.OdbcDataAdapter(Comando)
    Adaptador.Fill(ingreso, "inmueble") 'database
    conexion.conexionBD.Close()
    codinm = ingreso.Tables(0).Rows(0).Item("cod")
```

```

conexion.conectar()
auxiliar.Clear()
Comando = New Odbc.OdbcCommand("SELECT
DISTINCT(CarCodigo),CarTipoControl FROM
auxiliar_ingreso_caracteristicas", conexion.conexionBD)
Adaptador = New Odbc.OdbcDataAdapter(Comando)
Adaptador.Fill(auxiliar, "auxiliar_ingreso_caracteristicas")
'database
conexion.conexionBD.Close()

For i = 0 To auxiliar.Tables(0).Rows.Count - 1

    codcarc = auxiliar.Tables(0).Rows(i).Item("CarCodigo")
    conexion.conectar()
    'ingreso.Clear()
    Comando = New Odbc.OdbcCommand("SELECT * FROM
caracteristica WHERE CarCodigo=" & codcarc, conexion.conexionBD)
    Adaptador = New Odbc.OdbcDataAdapter(Comando)
    Adaptador.Fill(ingreso, "caracteristica")
    conexion.conexionBD.Close()
    tipo_dato =
ingreso.Tables(1).Rows(i).Item("CarTipoDatos")
    tipo_control =
auxiliar.Tables(0).Rows(i).Item("CarTipoControl")

    Select Case tipo_control
        Case "DropDownList"
            cbo = tab.FindControl(codcarc)
            If tipo_dato = "Double" Then
                tbl_caracinm.Insert(0, "", 0,
cbo.SelectedValue, codcarc, codinm, fecha)
            ElseIf tipo_dato = "String" Then
                tbl_caracinm.Insert(0, cbo.SelectedValue, 0,
0, codcarc, codinm, fecha)
            ElseIf tipo_dato = "Boolean" Then
                tbl_caracinm.Insert(0, "",
cbo.SelectedValue, 0, codcarc, codinm, fecha)
            ElseIf tipo_dato = "Integer" Then
                tbl_caracinm.Insert(cbo.SelectedValue, "",
0, 0, codcarc, codinm, fecha)

```

```

        End If
    Case "CheckBoxList"
        chk = tab.FindControl(codcarc)
        If tipo_dato = "Double" Then
            For j = 0 To chk.Items.Count - 1
                If chk.Items(j).Selected = True Then
                    tbl_caracinm.Insert(0, "", 0,
chk.Items(j).Value, codcarc, codinm, fecha)
                End If
            Next
        ElseIf tipo_dato = "String" Then
            For j = 0 To chk.Items.Count - 1
                If chk.Items(j).Selected = True Then
                    tbl_caracinm.Insert(0,
chk.Items(j).Value, 0, 0, codcarc, codinm, fecha)
                End If
            Next
        ElseIf tipo_dato = "Boolean" Then
            For j = 0 To chk.Items.Count - 1
                If chk.Items(j).Selected = True Then
                    tbl_caracinm.Insert(0, "",
chk.Items(j).Value, 0, codcarc, codinm, fecha)
                End If
            Next
        ElseIf tipo_dato = "Integer" Then
            For j = 0 To chk.Items.Count - 1
                If chk.Items(j).Selected = True Then
tbl_caracinm.Insert(chk.Items(j).Value, "", 0, 0, codcarc, codinm,
fecha)
                End If
            Next
        End If
    Next
End If
Case "RadioButtonList"
    rad = tab.FindControl(codcarc)
    If tipo_dato = "Double" Then
        tbl_caracinm.Insert(0, "", 0,
rad.SelectedValue, codcarc, codinm, fecha)
    ElseIf tipo_dato = "String" Then
        tbl_caracinm.Insert(0, rad.SelectedValue, 0,
0, codcarc, codinm, fecha)
    End If
End Case
End Sub

```



```

ElseIf tipo_dato = "Boolean" Then
    If rad.SelectedValue = "Si" Then
        valor_booleano = 1
    Else
        valor_booleano = 0
    End If
    tbl_caracinm.Insert(0, "", valor_booleano,
0, codcarc, codinm, fecha)
ElseIf tipo_dato = "Integer" Then
    tbl_caracinm.Insert(rad.SelectedValue, "",
0, 0, codcarc, codinm, fecha)
End If
Case "TextBox"
    txt = tab.FindControl(codcarc)
    Dim d = Me.txt.Text
    If tipo_dato = "Double" Then
        tbl_caracinm.Insert(0, "", 0,
CDBl(txt.Text), codcarc, codinm, fecha)
    ElseIf tipo_dato = "String" Then
        tbl_caracinm.Insert(0, txt.Text, 0, 0,
codcarc, codinm, fecha)
    ElseIf tipo_dato = "Boolean" Then
        tbl_caracinm.Insert(0, "", txt.Text, 0,
codcarc, codinm, fecha)
    ElseIf tipo_dato = "Integer" Then
        tbl_caracinm.Insert(CInt(txt.Text), "", 0,
0, codcarc, codinm, fecha)
    End If
End Select
Next i

MsgBox("La Propiedad ha sido ingresada exitosamente",
MsgBoxStyle.Information, "PROPIEDADES")
conexion.conexionBD.Close()

conexion.conectar()
inmueble.Clear()
Comando = New Odbc.OdbcCommand("SELECT max(InmCodigo) cod
from inmueble", conexion.conexionBD)
Adaptador = New Odbc.OdbcDataAdapter(Comando)
Adaptador.Fill(inmueble, "inmueble") 'database

```

```
conexion.conexionBD.Close()
codinm = inmueble.Tables(0).Rows(0).Item("cod")
Response.Write("<script>
location.href='AñadirFotografias.aspx?codinm=" & codinm & "';
</script>")
End Sub
```

### 3.3.4 Aplicativo Móvil

Los mismos principios que se aplican para almacenar las características de un inmueble en la parte Web, se utilizan en la codificación para el Dispositivo Móvil, salvo con algunas variaciones que se detallan a continuación:

En el dispositivo móvil los datos para la creación de los controles dinámicos se van a tomar desde archivos XML, que previamente se cargaran en el dispositivo una vez hayan sido importados desde la aplicación. Estos Archivos XML contendrán los datos de las tablas de la BD.

El proceso de ingresar inmuebles se lo hará en una tabla virtual y al final de la jornada el vendedor podrá grabar todos los datos recolectados en un archivo XML, que se deberá descargar en la BD a través de la aplicación.

Un punto interesante de comparación entre Web Forms y Formularios Smart Device, es que mientras el evento Load en Web Forms se carga siempre que la página es refrescada, en Formularios Smart Device se carga solo al ejecutar la aplicación. Por este motivo se hizo un proceso para borrar los controles dinámicos y permitir que el Vendedor pueda ingresar más de una propiedad.

En el aplicativo para móviles se cuenta con la opción de ingresar los datos del propietario, en caso de que no esté registrado en la Base de Datos del Sistema de Gestión Inmobiliaria.

Para depurar y probar la aplicación móvil contamos con el Administrador de emuladores de dispositivos, donde escogemos la opción que se ajuste a la plataforma de destino, especificada cuando creamos el proyecto Smart Device.

A continuación mostramos los pasos para implementar ejecutar el proyecto en el emulador de dispositivos Menú->Herramientas -> Administrador de emuladores de dispositivos:

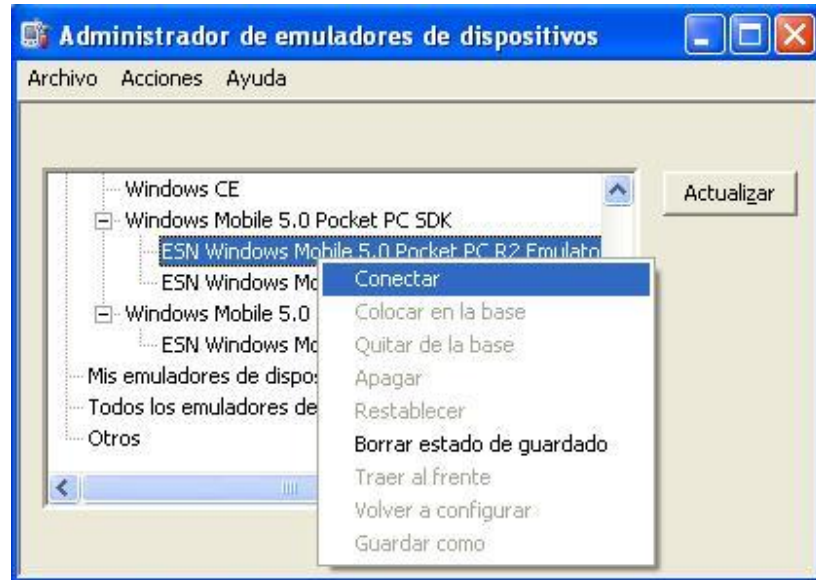


Figura 3.14: Como Conectar el Emulador de Dispositivos

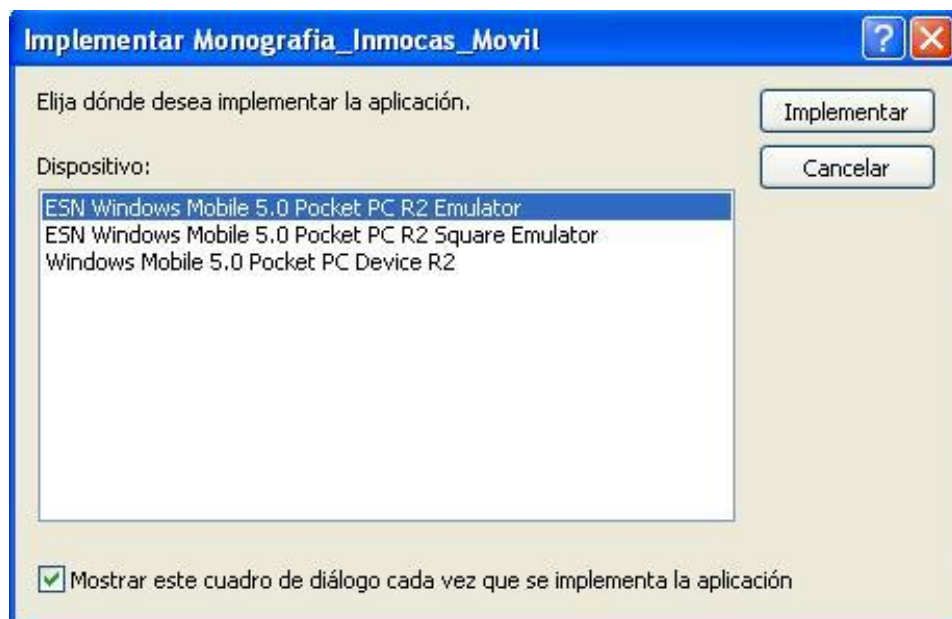


Figura 3.15: Elección del Emulador



Figura 3.16: Pantalla Principal del Aplicativo Móvil

Como podemos ver en el menú contamos con las dos opciones indicadas anteriormente, para ingresar Inmuebles y Propietarios. En la Aplicación Web contamos con un formulario para crear los archivos XML que se utilizarán en el Dispositivo Móvil y otro formulario para Descargar los datos recolectados en el pocket.

IMPORTAR BASE DE DATOS PARA POCKET

Elija las tablas que desea importar

- Tipo de Inmueble
- Características
- Propietarios

Figura 3.17: Formulario para importar archivos XML en la Aplicación Web

DESCARGAR INFORMACION DEL POCKET

Propietarios  Examinar...

Inmuebles  Examinar...

Figura 3.18: Formulario para descargar Archivos XML en la Aplicación Web

### 3.3.5 Crear Galerías Fotográficas para cada propiedad

Este punto es importante pues al cliente le interesa ver las fotos de la propiedad para corroborar que la información descrita con palabras encaja con las imágenes.

El Administrador podrá ingresar como mínimo 1 fotografía de cada inmueble y como máximo 10. A estas imágenes antes de ser guardadas se les cambia el nombre, que estará conformado por el código del inmueble, el número de foto y la extensión. En la Base de Datos se guarda el nombre de la foto y el inmueble al que pertenece, pero la imágenes en si están guardadas en una carpeta.

En el menú principal de la página tenemos la clasificación de los tipos de inmueble, cuando el usuario escoja un tipo específico, el sistema le dirigirá a una página donde se mostrará un listado de todas las ofertas disponibles para ese tipo de inmueble. Ej., El usuario eligió Villas y en la página siguiente se le desplegarán todas las villas disponibles para la venta.

Para este propósito se investigó y utilizó el control DataList que nos permite realizar un listado solo con las propiedades que deseamos mostrar. Se realiza una consulta SQL para obtener los inmuebles que el usuario solicitó ver, los resultados de esta consulta se guardan en un DataSet. Los datos contenidos en el DataSet se pasan a un DataTable que se enlaza al DataList y los datos se muestran en pantalla en forma de listado.

Al control DataList se le puede dar un formato personalizado, el mismo que se utiliza con todos los elementos de la lista. Para esto el DataList cuenta con un editor de

plantillas para el encabezado, pie de de ítem y el contenido del ítem (ItemTemplate). Se ha dado el siguiente formato a la plantilla ItemTemplate:



Figura 3.19: Plantilla para mostrar las Propiedades Ofertadas

El link Ver Mas Info, nos mostrará los detalles de la propiedad incluida la galería fotográfica. La Galería Fotográfica se realizó con una imagen para visualizar la fotografía que el usuario elija en mayor tamaño, y con ImageButton para las fotografías en miniatura.

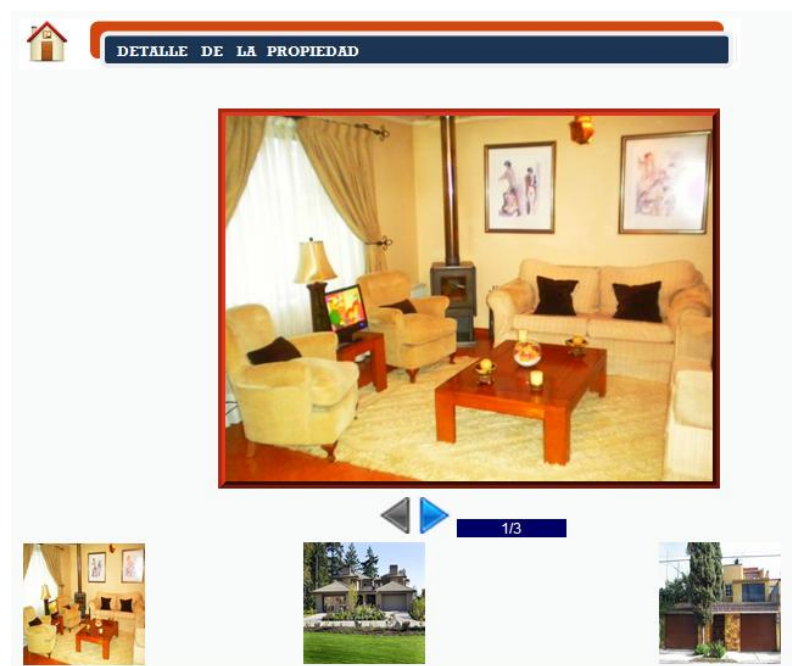
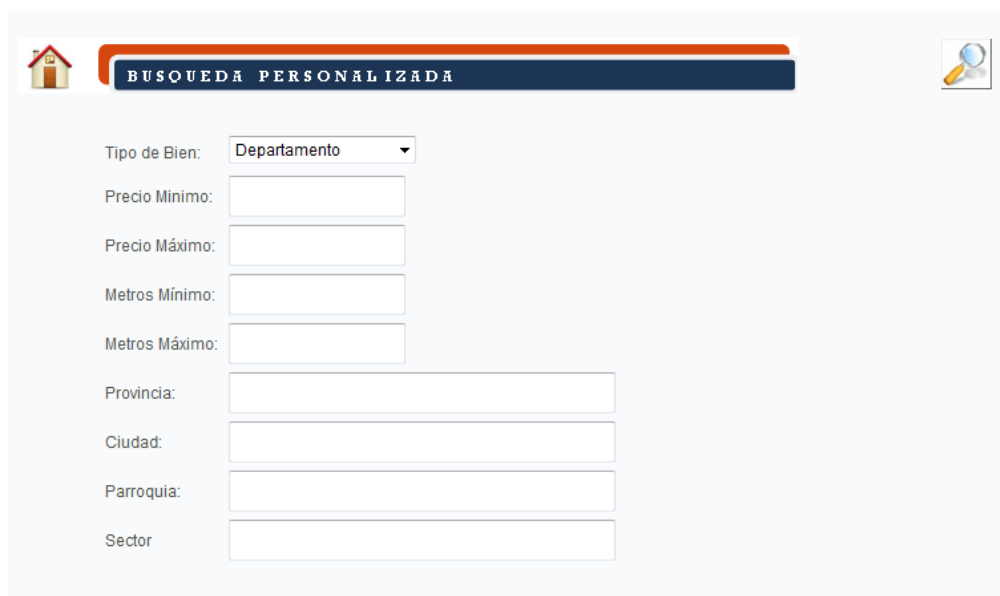


Figura 3.20: Galería Fotográfica

### 3.3.6 Búsquedas Especializadas

Con la finalidad de facilitarle al cliente la ubicación de una propiedad con determinadas características se ha implementado la búsqueda especializada, para la cual se presentan 9 filtros diferentes de búsqueda:



El formulario de búsqueda personalizada, titulado "BUSQUEDA PERSONALIZADA", contiene los siguientes campos:

- Tipo de Bien:
- Precio Mínimo:
- Precio Máximo:
- Metros Mínimo:
- Metros Máximo:
- Provincia:
- Ciudad:
- Parroquia:
- Sector:

Figura 3.21: Formulario para Búsqueda Personalizada

El usuario puede ingresar cualquiera de estos valores, siendo el Tipo de Inmueble el único valor obligatorio.

Cuando el usuario no ingresa alguno de los campos tipo carácter, esto no influye en la Consulta que se debe implementar para la búsqueda debido al uso del comando "LIKE", pero cuando alguno de los valores tipo Numérico no contiene datos se debe crear una nueva consulta para que no se genere un error, es decir, al tener 4 campos de tipo Numérico se deben crear todas las combinaciones necesarias para poder generar la búsqueda específica. En el siguiente cuadro se especifica de manera más clara las combinaciones que se crean al ingresar o no cualquiera de los 4 valores.

<b>Ingresa Metros Mínimo</b>	<b>Ingresa Metros Máximo</b>	<b>Ingresa Precio Mínimo</b>	<b>Ingresa Precio Máximo</b>
SI	SI	SI	SI
SI	SI	SI	NO
SI	SI	NO	SI
SI	SI	NO	NO
SI	NO	SI	SI
SI	NO	SI	NO
SI	NO	NO	SI
SI	NO	NO	NO
NO	SI	SI	SI
NO	SI	SI	NO
NO	SI	NO	SI
NO	SI	NO	NO
NO	NO	SI	SI
NO	NO	SI	NO
NO	NO	NO	SI
NO	NO	NO	NO

Tabla 3.2: Combinaciones para la Búsqueda Personalizada

El resultado total de la búsqueda se muestra en un DataList con la foto principal y los datos más importantes de cada propiedad, además de accesos directos para ver todas las características del inmueble, realizar una reserva o ubicarla en el mapa.

Para mostrar las propiedades en el DataList se debe seguir un proceso en el que primeramente se llena un dataset consultando en la tabla de propiedades y características de acuerdo a los valores de los filtros ingresados, luego dentro de un bucle que va recorriendo cada inmueble encontrado se llena otro dataset con la foto principal del inmueble actual, dentro del mismo bucle se pasan los datos del inmueble actual junto con su foto principal a un datatable, el cual se enlaza con un DataView y este último con el DataList que muestra el resultado final.



La plantilla del DataList se puede modificar al ingresar al Editor de Plantillas de este control, en él se colocan los links, fotos y campos que se desean visualizar por medio del comando `<%#DataBinder.Eval(Container.DataItem, "InmPrecio")%>` en donde InmPrecio en este caso, es uno de los campos que contiene el datatable enlazado con el DataView.

### 3.3.7 Reportes Gerenciales

El Sistema presenta tres tipos de Reportes Gerenciales que le permitirá al personal administrativo de la empresa conocer resultados clave acerca de la operación de la misma. Los reportes generados son:

- El Vendedor que más ventas realiza
- El Tipo de Inmueble más vendido
- Ventas Realizadas

Todos los reportes reciben como parámetros de entrada una Fecha de Inicio y una de Fin con el objetivo de evaluar los datos en un período de tiempo determinado. La lógica para la creación de cada uno de ellos es la misma y conlleva los siguientes pasos:

1. En el Explorador de Soluciones dar clic derecho en el Proyecto y a continuación en Agregar nuevo elemento.
2. Escoger Crystal Report y ponerle un nombre (Crystal Report.rpt)
3. Usar el Asistente de Informes Estándar
4. En Orígenes de Datos disponibles escoger ODBC (creada anteriormente con el ODBC connector en las Herramientas Administrativas de Windows)

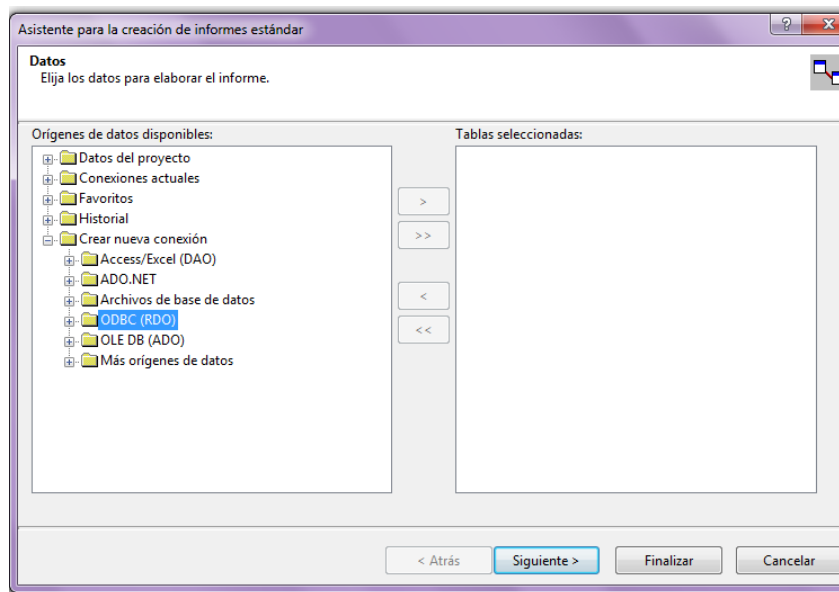


Figura 3.22: Escoger el Origen de Datos

5. Desplegar el origen de datos ODBC y escoger el DSN creado cuando se agregó el Origen de datos en las herramientas administrativas.
6. Escoger la o las tablas necesarias para crear el reporte y presionar el botón Siguiete.
7. Escoger los campos exactos que se desean visualizar en el reporte.
8. En el caso de los reportes el Vendedor Exitoso o el Tipo de Bien más Vendido es necesario agrupar los datos por el nombre del Vendedor y por el nombre del Tipo de Bien respectivamente.

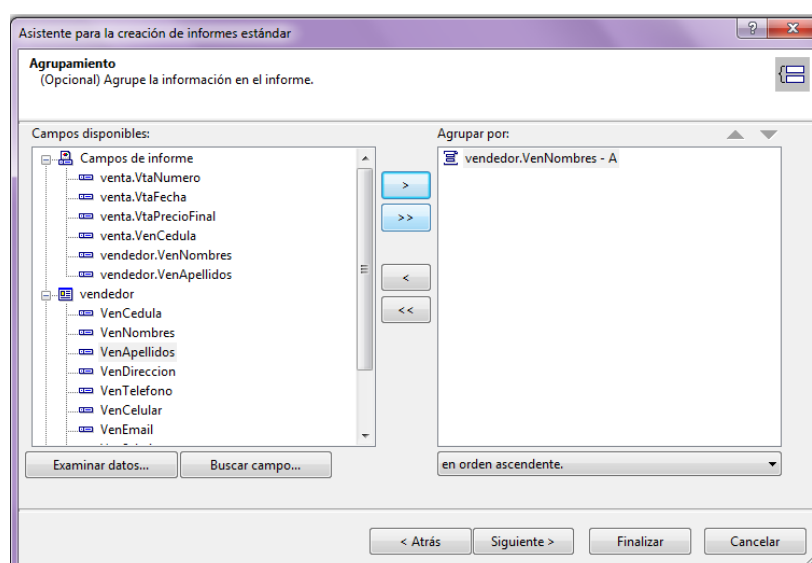


Figura 3.23: Agrupar la información del Informe

9. Para sumar el número de ventas por Empleado y por Tipo de Bien, en la sección de Campos resumidos, se debe escoger la clave de la tabla de la que se requiere llevar la cuenta y en la parte inferior escoger en lugar de la opción de Suma que viene por omisión, la opción de Recuento. Presionar el botón Siguiente.

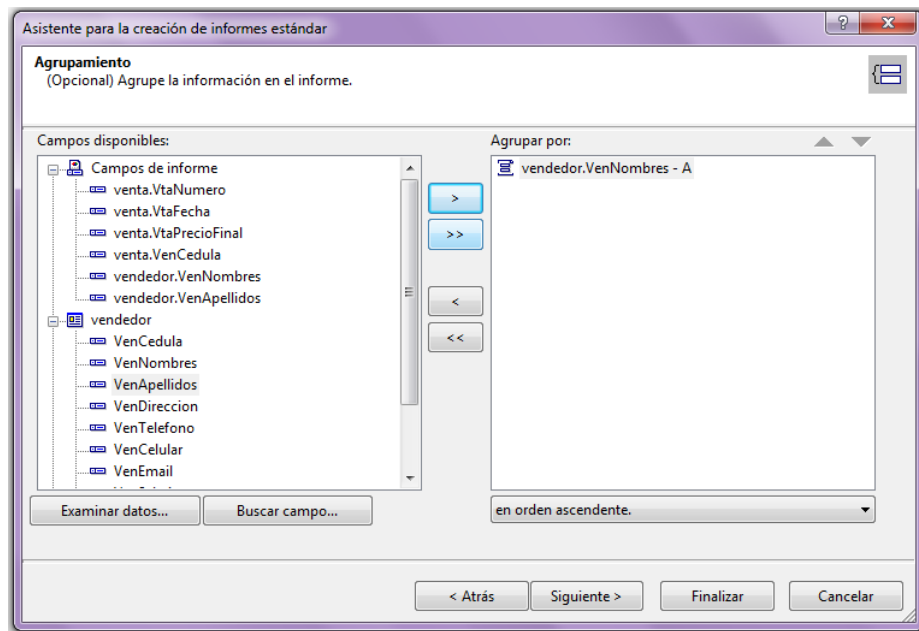


Figura 3.24: Escoger la llave correcta

10. Si se desea mostrar un diagrama ilustrativo de los resultados se hace clic en el tipo de diagrama que se quiera y en la parte inferior escribir el nombre con el que se mostrará.

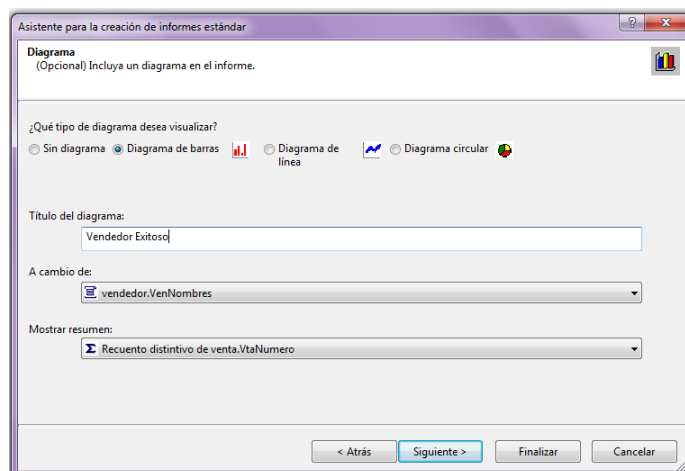


Figura 3.25: Agregar un Diagrama al Informe

11. Crear un nuevo Web Form y en él agregar un control Crystal Report Source ubicado en el Cuadro de Herramientas en la sección de Informe.

12. Enlazar el Crystal Report Source con el Crystal Report.rpt creado anteriormente yendo a configurar el origen del informe y escogiendo el Crystal Report.

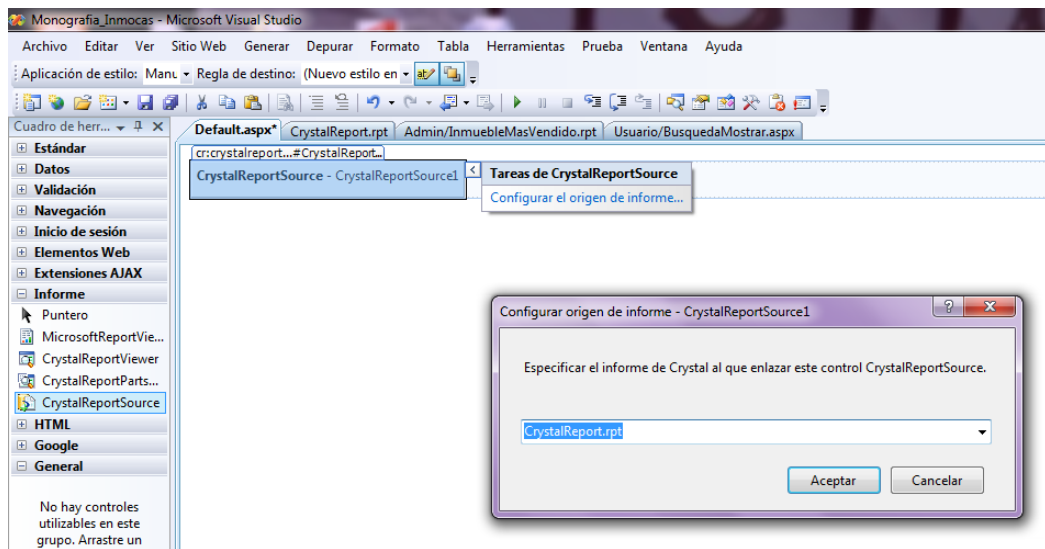


Figura 3.26: Enlazar Crystal Report Source con Crystal Report.rpt

13. Agregar un Crystal Report Viewer al Web Form y enlazarlo de la misma manera que el punto anterior con el Crystal Report Source.

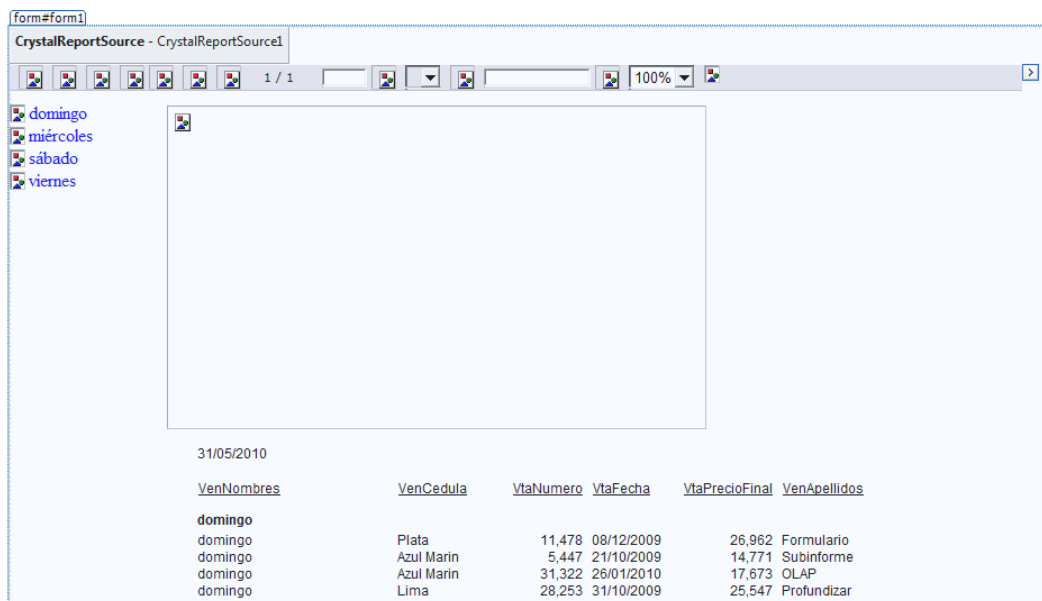


Figura 3.27: Agregar un Crystal Report Viewer a un Web Form

14. Para poder recibir los parámetros de entrada hay que ubicarse en el Crystal Report.rpt y se abrirá el explorador de campos en la parte izquierda de la pantalla.
15. Dar clic derecho en Campos de Parámetro y luego en Nuevo.

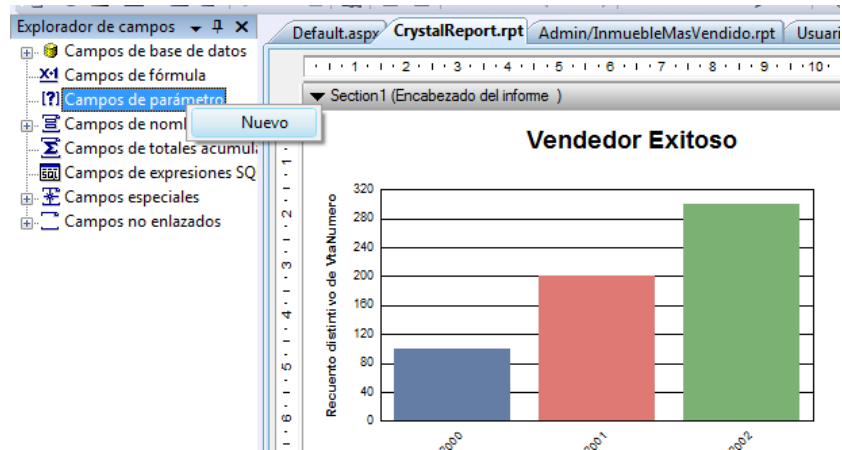



Figura 3.28: Agregar Parámetros

16. Poner Date como tipo de dato de los parámetros.
17. Ir a la Barra de Herramientas de Crystal Report y dar clic en el botón  que es el Asistente de Selección.
18. Escoger el campo que será validado en los parámetros, en este caso sería la Fecha de la Venta y presionar Aceptar.
19. Para especificar que la fecha de la Venta esté entre los parámetros de Fecha Inicial y Fecha Final ingresados por el Usuario en el cuadro presentado escoger la opción “es entre” y en los dos cuadros que presenta esta opción poner cada una de las variables ingresadas como parámetros.

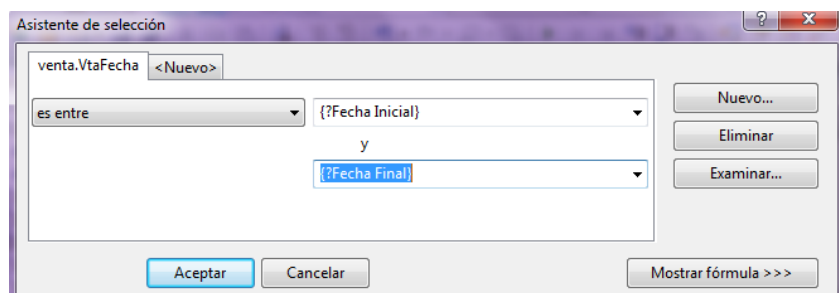


Figura 3.29: Especificar Parámetros

### 3.3.8 Envío del Email

Después de que un visitante de la página web se registra como cliente, el sistema le envía un mail dándole la bienvenida e informándole cuales fueron los datos ingresados. También como respuesta a las peticiones se le puede enviar un mail al cliente notificándole que se ha encontrado una propiedad con las características requeridas.

Para este proceso primeramente es importante crearse una cuenta en GMAIL con la finalidad de utilizar su servidor de correos.

El cuerpo del mail fue creado previamente en Dreamweaver 8 y para adaptarlo a Visual Studio se debe crear una nueva instancia de la clase StringBuilder, la misma que puede albergar una cadena de caracteres modificable.

```
Dim html As New StringBuilder()
```

Cada una de las líneas de código obtenidas en la creación en Dreamweaver se agrega a la variable html por medio del comando Append

```
html.Append("<html xmlns='http://www.w3.org/1999/xhtml'>")  
html.Append("<head>")
```

Al importar la librería System.Net.Mail se puede declarar una variable de tipo mensaje de correo electrónico con la línea

```
Dim email As New System.Net.Mail.MailMessage
```

La línea email.To.Add("raulcardoso@hotmail.com") especifica que se mandará el correo a la dirección descrita en el parámetro.

Para especificar quien envía el mail se utiliza email.From = New MailAddress("inmocasbienesraices@gmail.com", "INMOCAS Bienes Raíces") en donde el primer parámetro es el email en gmail creado como primer punto y el segundo parámetro es el título.

La incorporación del cuerpo al mensaje se hace por medio de la línea:

```
email.Body = html.ToString
```

Donde html es la variable que instancia la clase StringBuilder creada anteriormente.

Las líneas:

```
email.Subject = "Bienvenido a INMOCAS"
```

```
email.IsBodyHtml = True
```

Indican el asunto del mail y que el cuerpo del mensaje de correo está en HTML.

El protocolo utilizado para el envío del mail es SMTP, por lo cual es necesario crear una variable de este tipo mediante:

```
Dim smtp As New System.Net.Mail.SmtpClient("smtp.gmail.com", 587)
```

En donde el primer parámetro corresponde al servidor smtp de gmail y el segundo al puerto utilizado.

Para la autenticación del remitente se utiliza

```
smtp.Credentials = New
```

```
System.Net.NetworkCredential("inmocasbienesraices@gmail.com", "raulcastro")
```

En donde el segundo parámetro es la contraseña del email.

Por último la línea smtp.Send(email) es la que envía finalmente el mail.

### 3.3.9 AdRotator

Para mejorar la interfaz de la Master Page se le agregó un control AddRotator, con el fin de presentar una secuencia de imágenes de inmuebles en tiempo de ejecución. Este control se basa en un archivo XML que permite presentar las imágenes, para ello se utilizan las etiquetas `<ImageUrl>imagenes/2.jpg</ImageUrl>` que especifican las URLs de cada una de las imágenes que se desean visualizar.

En la Master Page se debe colocar un control UpdatePanel que se encuentra en el Cuadro de Herramientas en la sección de Extensiones de AJAX, y dentro de este

colocar un Timer, un ScriptManager y el AdRotator. La propiedad Interval del Timer determinará cada cuantos milisegundos se cambiará de imagen. La propiedad que relaciona el archivo XML con el AdRotator es la AdvertisementFile, al hacer clic en ella se puede seleccionar el archivo XML correspondiente.



## CAPITULO 4

### PRUEBAS Y PUESTA EN OPERACIÓN DEL SISTEMA

#### 4.1 Creación del instalador del Sistema de Gestión para una Inmobiliaria en el dispositivo móvil

Uno de los objetivos de nuestro proyecto es crear e implementar un aplicativo para un dispositivo móvil. En este punto vamos a ver como se empaquetó la solución para su instalación en el dispositivo. El primer paso es añadir un Proyecto Cab a la Solución Smart Device.

Nos dirigimos a Archivo -> Nuevo Proyecto y se nos desplegará la siguiente pantalla:

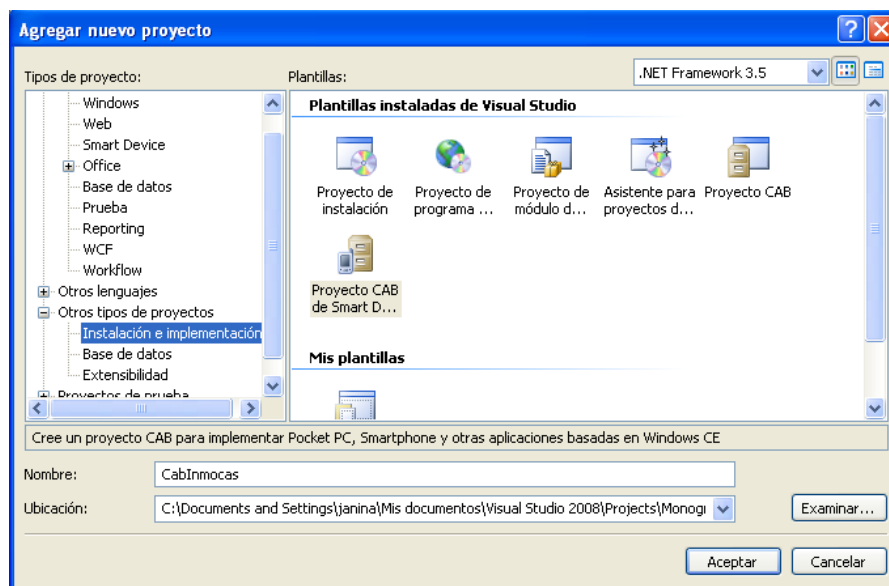


Figura 4.1: Elección de un Proyecto CAB de Smart Device

Elegimos Otros Tipos de proyectos -> Instalación e implementación y seleccionamos Proyecto CAB de Smart Device. Para cambiar el nombre del Instalador, en el Explorador de Soluciones damos clic derecho en el CabInmocas y escogemos la opción Propiedades, donde se mostrará esta pantalla:

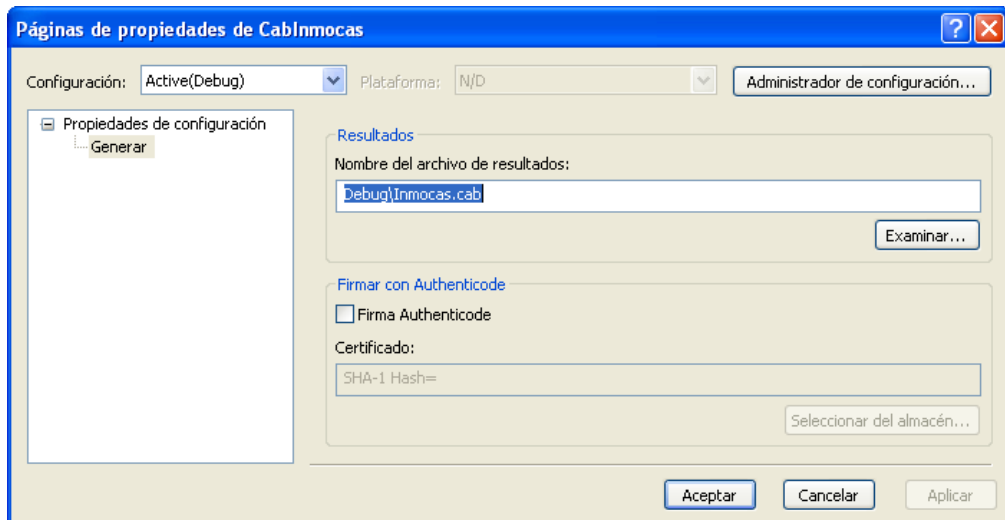


Figura 4.2: Establece el nombre del Instalador

El tercer pasó es ir a la pestaña Sistema de Archivos -> Carpeta de la aplicación para elegir los archivos que formarán parte de la solución que se va a empaquetar.

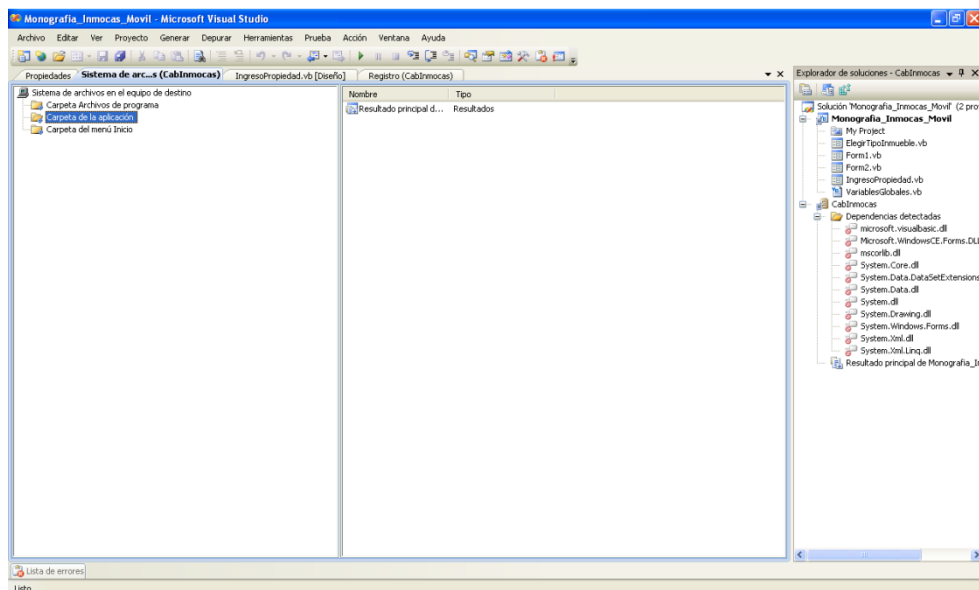


Figura 4.3: Carpeta de la Aplicación

Una vez que señalemos la Carpeta de la Aplicación vamos al menú Acción->Agregar->Resultados del Proyecto:

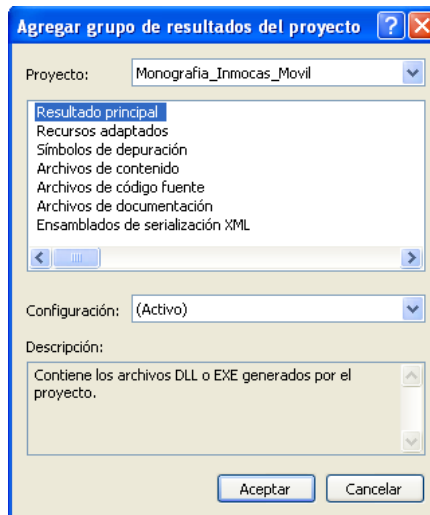


Figura 4.4: Elección de la Solución a empaquetar

Luego de haber elegido Resultado principal, vamos a crear el Acceso Directo. Para esto señalamos al Resultado principal y vamos al menú Acción -> Crear Acceso Directo como nos muestra la imagen:

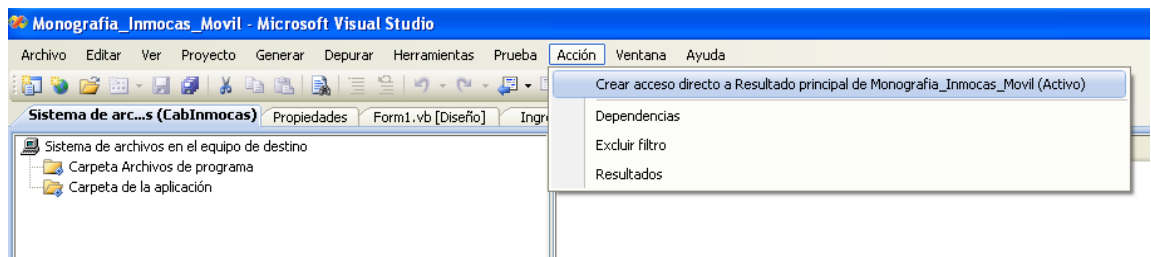


Figura 4.5: Crear Acceso Directo para el Aplicativo Móvil

A continuación se crea una entrada de Registro. Vamos al menú Ver->Editor->Registro, y creamos una clave en la ruta que nos muestra la imagen:

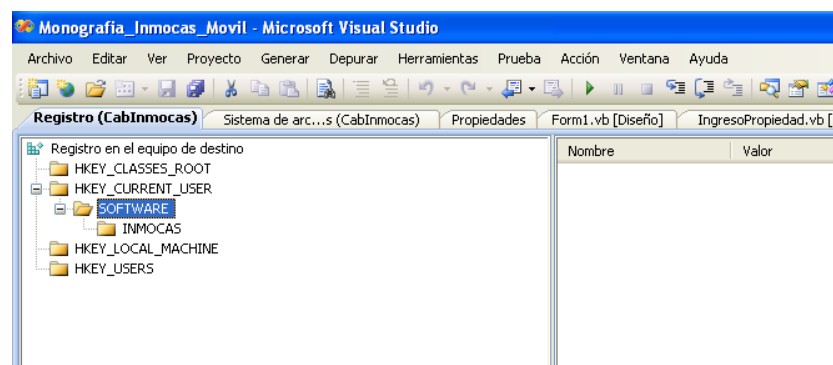


Figura 4.6: Crear una Entrada de Registro

A continuación, vamos a generar la solución:

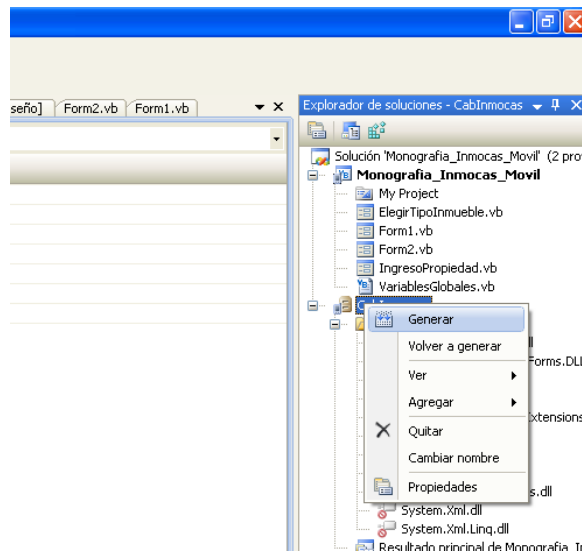


Figura 4.7: Generar Solución

Luego de Generar vamos al menú Archivo->Guardar Todo. Ahora veamos donde se guardó el instalador:

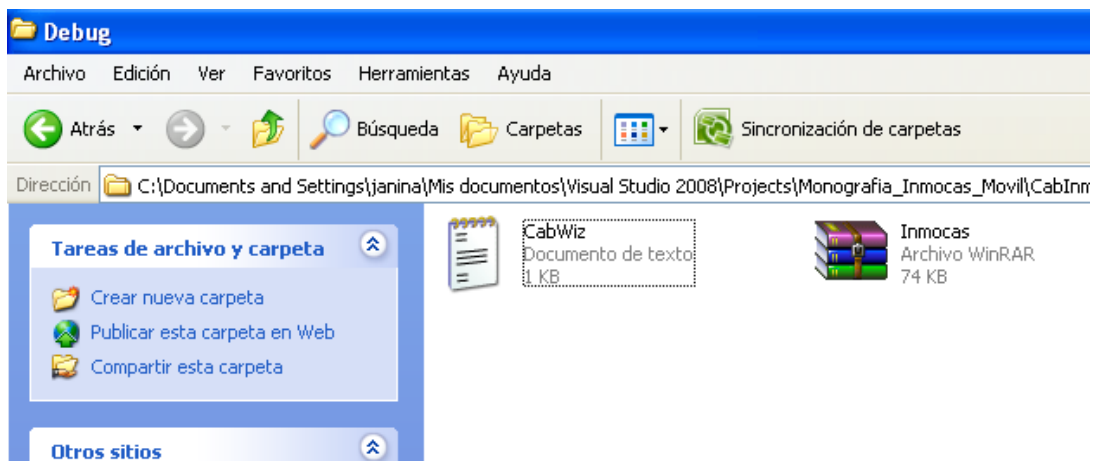


Figura 4.8: Instalador

El archivo .zip es en donde se encuentra el instalador.

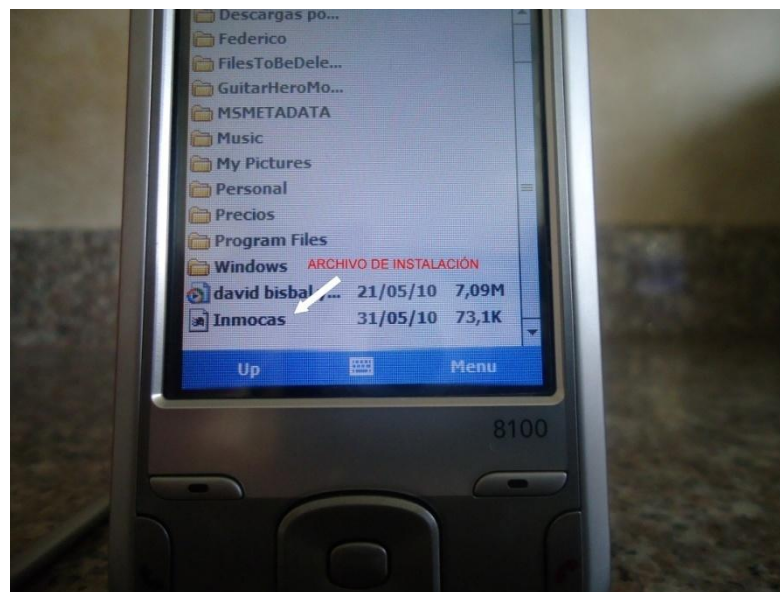
## 4.2 Instalación de la Aplicación en el dispositivo móvil

Ahora que hemos obtenido el instalador, vamos a realizar la instalación en el Pocket. Primero vamos a copiar el instalador en el dispositivo móvil.



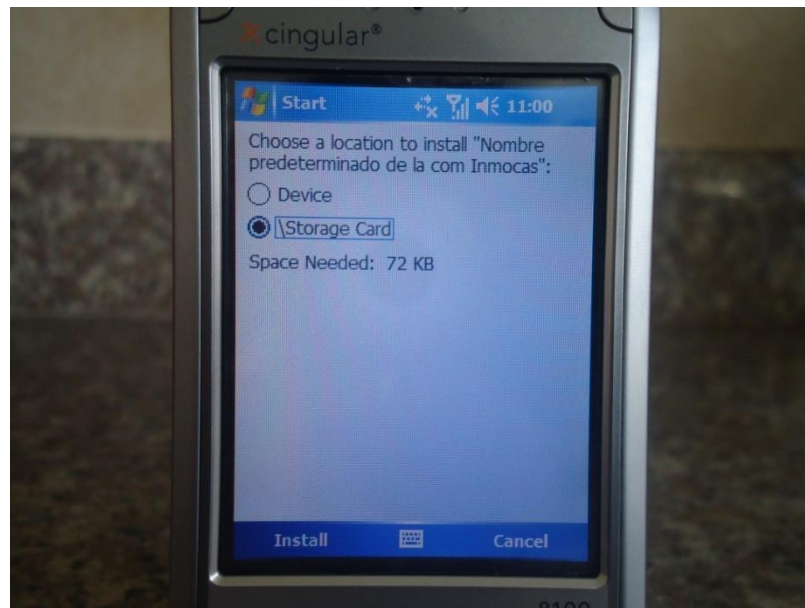
Fotografía 4.1: Pocket donde se instalará el Aplicativo

En la imagen podemos observar el archivo de instalación:



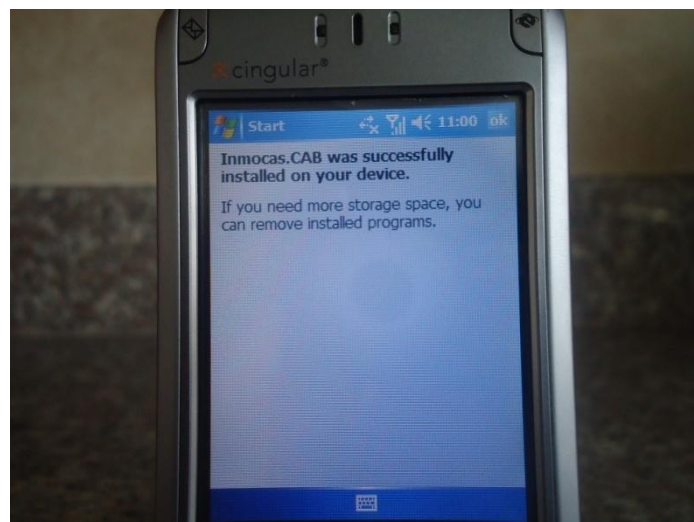
Fotografía 4.2: Archivo de Instalación

Damos doble clic con el lápiz en el instalador y se nos presenta la siguiente imagen:



Fotografía 4.3: Escoger el lugar de Instalación

En este caso hemos elegido instalar el aplicativo en la Tarjeta de Memoria, damos clic en Aceptar y se instala el aplicativo. Si se ha instalado correctamente nos va a salir el siguiente mensaje:



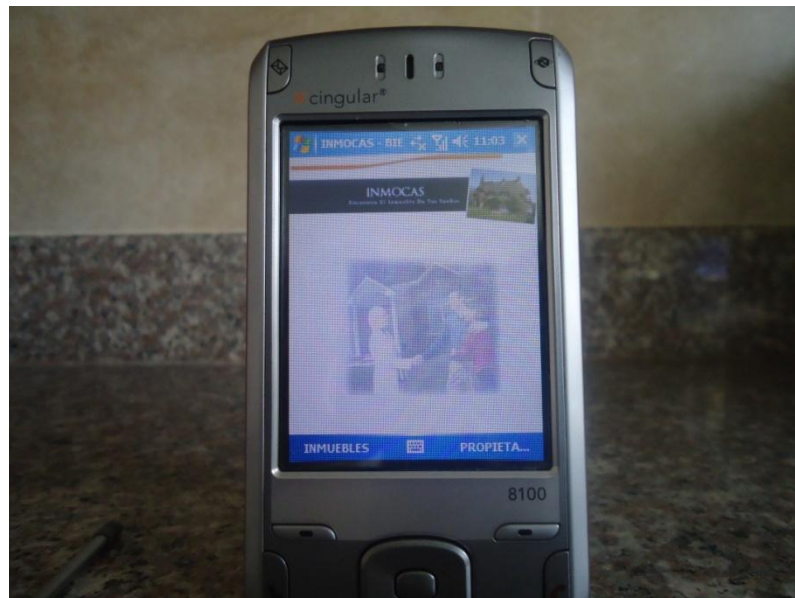
Fotografía 4.4: Instalación Satisfactoria

Ahora podemos ver al ícono de nuestro aplicativo en el menú de programas:



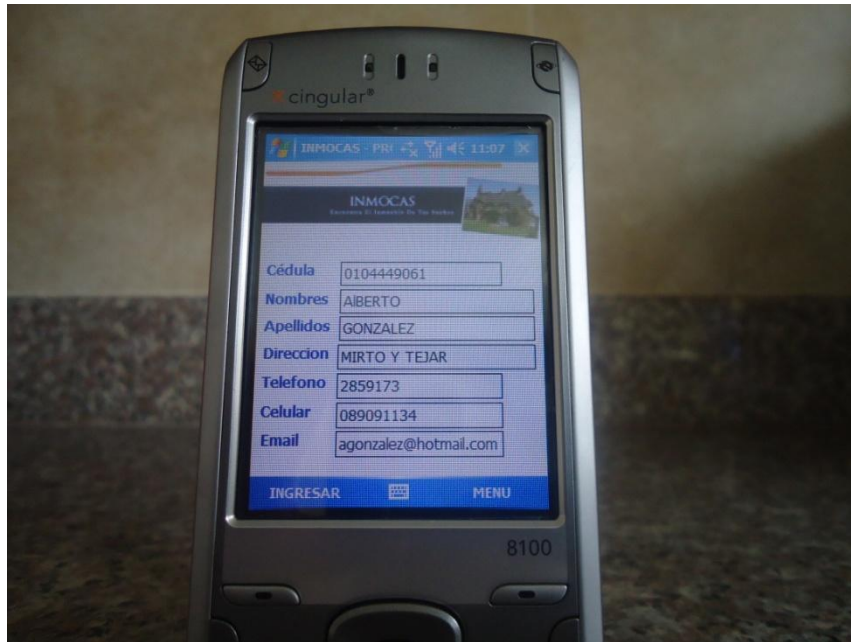
Fotografía 4.5: Ícono del Aplicativo

Damos doble clic para entrar en el Aplicativo para dar funcionamiento a nuestra aplicación. La siguiente imagen nos muestra la pantalla principal:

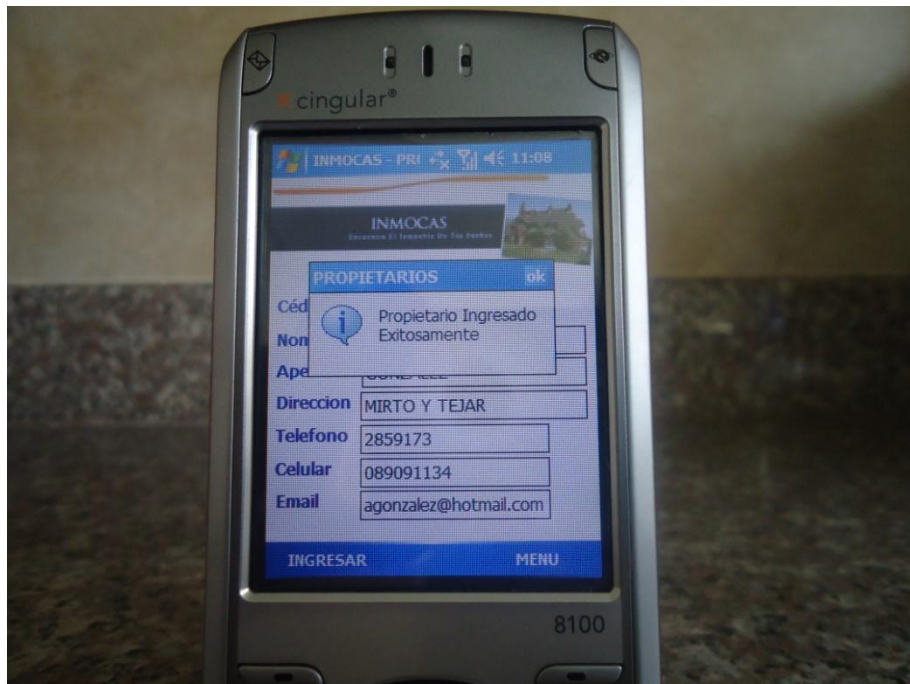


Fotografía 4.6: Pantalla Principal del Aplicativo

De ejemplo el formulario de Ingreso de Propietarios:



Fotografía 4.7: Formulario de Ingreso de Propietarios



Fotografía 4.8: Mensaje de Propietario Ingresado Exitosamente



### **4.3 Pruebas del Sistema**

En el transcurso del desarrollo de la Aplicación Web y Aplicativo Móvil, se hicieron pruebas para corroborar que los datos que estaban alimentando al Sistema producían los resultados esperados. Es vital realizar las pruebas desde el inicio del proyecto para detectar errores y buscar los elementos que se necesitan para solucionarlos.

En una aplicación Web se realizan distintos tipos de pruebas, que complementan al sistema tradicional de pruebas que se hace en Orientación a Objetos. Las pruebas que se realizaron a la Aplicación Web y al Aplicativo móvil fueron:

#### **4.3.1 Prueba de Contenido**

Se revisa la ortografía de todos los formularios, y la redacción en caso de existir texto. En esta prueba se verifica que el contenido de la WebApp transmita el mensaje que se quiere dar al usuario final, que en nuestro caso sería que sepa que tipos de inmuebles ofrecemos, los servicios en línea de los que puede ser beneficiario si se registra y los datos de las propiedades.

En esta prueba se vio si el estilo aplicado al contenido era acorde al estilo estético global de la aplicación. Otro punto importante era que los datos obtenidos desde la BD tenían el formato correcto al ser presentados al usuario. Esta prueba se realizó para determinar si los controles dinámicos que se crean mediante código, corresponden a la solicitud del administrador (ya que los controles dinámicos dependen del tipo de inmueble). Los mensajes de error presentados al usuario son personalizados de acuerdo al tipo de error que se presente.

Se revisaron y corrigieron los elementos del menú en el transcurso del desarrollo para que tengan nombres entendibles al usuario final y permitan una navegación fácil.

### **4.3.2 Prueba de la Interfaz del Usuario**

En esta prueba se revisa cada elemento que forma parte de la interfaz para comprobar que su funcionalidad es correcta, tiene facilidad de uso y es intuitivo al usuario final. Se verificó si el diseño de la interfaz ofrece una dirección clara y consistente, es decir que el usuario pueda navegar con facilidad hacia sus objetivos. Se debe determinar que los textos sean legibles y comprensibles, esto se logra con una tipografía clara que no incluya fuente de tipo simbología o similares y que sean del tamaño correcto.

La plantilla que diseñamos fue objeto de varias correcciones para mantener su formato mientras el usuario navega por el Sitio Web y para que se vea agradable en el explorador. Se probó que la personalización de la interfaz sea según cada usuario, es decir que se despliegue en pantalla el menú que debe ser. Se probó específicamente en dos exploradores Internet Explorer y Mozilla.

Nuestro objetivo con la interfaz de usuario fue diseñarla de tal forma que resulte clara, informativa, fácil de usar, uniforme y predecible para el usuario final, con el propósito de evitarle confusiones.

### **4.3.3 Prueba de Componentes o Unidades**

Cada Formulario Web es un componente que debe ser probado a nivel de interfaz y a nivel de funcionalidad.

La primera prueba al realizar un nuevo formulario web para mantenimiento es que el diseño esté bien aplicado y que los elementos estén ubicados correctamente dentro del form según el estándar de formularios, con el mismo formato de título, fuente, color e imágenes para los botones. La segunda prueba consiste en ingresar datos y verificar que se ingresen, modifiquen o eliminen correctamente de la base de datos y que las búsquedas estén mostrando los datos correctos según el criterio de búsqueda. Se probó que los controles de validación verifiquen si lo ingresado por el usuario cumple con los requisitos.

En caso de formularios para mostrar datos se verifica que el procedimiento para publicarlos en pantalla produzca los resultados esperados, lo que quiere decir que las consultas a la BD se están realizando satisfactoriamente.

Los formularios para importar información y descargar información pasaron por un proceso que determinaba si la descarga de datos era correcta y se estaban validando los datos antes de ingresarlos en la BD en las tablas que correspondían. Al momento de importar se verificó que los archivos XML estén bien estructurados y contengan los datos necesarios para el funcionamiento en el dispositivo móvil. En el dispositivo móvil se hicieron pruebas en el Administrador de Emuladores de Dispositivos antes de empaquetar la solución.

Tanto en la parte web como en la móvil las pruebas de funcionalidad se hicieron con la herramienta de depuración, que fue vital para la detección de errores ya que nos ayudó a ver como evolucionaban los datos en los diferentes procesos y eventos de los controles donde se realizó la programación.

#### **4.3.4 Pruebas de Navegación**

En las pruebas de navegación se comprobó que todos los vínculos para navegar dentro del sitio web estén funcionando correctamente. Funcionar correctamente en navegación quiere decir que al pulsar un enlace estén redirigiéndose a la página correcta que debe estar acorde al título del hipervínculo. En caso de incluir parámetros, se revisa que se están pasando los correctos para ser capturados en la página correspondiente.

#### **4.3.5 Pruebas de Integración**

Todos los componentes del sistema que se programaron por separado, van a estar unidos por vínculos o botones, por lo que una vez concluida la codificación en la aplicación web y en el dispositivo móvil se deben realizar las respectivas pruebas para ver si todo está funcionando como un reloj, es decir, que los datos con los se está alimentando al sistema produzcan los resultados correctos, ya que muchas veces los datos ingresados en un formulario son base para otros formularios, y los datos

que estén en la BD son vitales para el buen funcionamiento del aplicativo móvil. A su vez, los datos descargados desde el aplicativo móvil influirán en los resultados presentados al usuario final.

La Base de Datos fue encerrada para probar desde el principio cada uno de los procesos con datos coherentes y precisos que se asemejan a la realidad, y de esta manera ir detectando errores en la funcionalidad, interfaz o navegación del Sistema de Gestión Inmobiliaria.

## CONCLUSIONES Y RECOMENDACIONES

Al concluir la realización de este trabajo de graduación hemos concluido que:

Microsoft Visual Studio 2008 es un excelente entorno de trabajo que ofrece muchas posibilidades para la creación de aplicativos en diferentes lenguajes y ambientes. Fue de mucha utilidad su vasta paleta de controles que facilitó los procesos de validación, el uso de extensiones AJAX y controles HTML que permitieron facilitar el diseño de los formularios.

Visual Studio permite desarrollar aplicativos para móviles con un entorno amigable y comprensible. Es de gran ayuda el emulador de dispositivos para Pocket PC que ofrece al desarrollador una idea de un dispositivo real.

Otro punto es la importancia de los complementos adicionales como los utilizados para Google Maps y AJAX Control Toolkit, que son librerías que añaden funcionalidad al entorno de Visual Studio.

Las mayores dificultades se presentaron con la creación de los controles dinámicos y el evento Load en el caso de la aplicación Web. En el aplicativo móvil se encontró dificultad al momento de borrar los controles creados dinámicamente, lo cual se tuvo que realizar mediante código.

Un sistema para inmobiliaria tiene la posibilidad de interactuar con muchos elementos multimedia como: videos, fotografías y archivos de sonido. Como recomendación se puede implementar un video que muestre los espacios de cada propiedad y se puede refinar la galería fotográfica.

Otras recomendaciones que se pueden implementar en el futuro son:

- En el aplicativo móvil se podría ingresar las propiedades en línea sin la necesidad de archivos XML.

- En la parte Web se podría implementar un módulo para registrar las promesas de compra – venta y contratos que actualmente se realizan de forma manual. Además, incluir el manejo de arriendos de bienes inmuebles.

## BIBLIOGRAFÍA

WEITZENFELD, Alfredo. Ingeniería de Software orientada a objetos con UML, Java e Internet. Primera Edición. México: Thomson International, 2005. 704 p. ISBN: 9789706861900.

PILONE Dan, PITMAN NEIL. UML 2.0 in a Nutshell. Primera Edición. Estados Unidos de Nortamérica: O' Reilly, 2005. 234p. ISBN: 0-596-00795-7.

*Arquitectura y diseño de sistemas web modernos*. Ilustre Colegio de Ingenieros Informáticos de la Región de Murcia. 2004, no. 1. Murcia: Ilustre Colegio de Ingenieros Informáticos de la Región de Murcia. ISSN: 1698-884.

Cargar DataSet desde XML - 19/05/2010

[http://msdn.microsoft.com/es-es/library/fx29c3yd\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/fx29c3yd(VS.80).aspx)

CAPITULO 4: ANALISIS DIAGRAMAS DE FLUJOS DE DATOS - 27/05/2010

<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060030/lecciones/Capitulo%204/dfd.htm>

Cómo: Cargar archivos con el control FileUpload de servidor Web - 23/05/2010

<http://msdn.microsoft.com/es-es/library/ms227669%28v=VS.80%29.aspx>

Creación de Reportes con Crystal Reports - 23/05/2010

[http://msdn.microsoft.com/es-es/library/ms227780\(v=VS.90\).aspx](http://msdn.microsoft.com/es-es/library/ms227780(v=VS.90).aspx)

DataList (Clase) - 13/05/2010

<http://msdn.microsoft.com/es-es/library/system.web.ui.webcontrols.datalist%28VS.80%29.aspx>

Desarrollo de dispositivos inteligentes - 30/05/2010

[http://msdn.microsoft.com/es-es/library/sa69he4t\(VS.90\).aspx](http://msdn.microsoft.com/es-es/library/sa69he4t(VS.90).aspx)

Descripción de Visual Studio 2008 – 30/05/2010

<http://www.microsoft.com/spanish/msdn/latam/visualstudio2008/professional.aspx>

Ejemplo: Confirmar formato válido de correo electrónico - 14/04/2010

[http://msdn.microsoft.com/es-es/library/aa720056\(v=VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa720056(v=VS.71).aspx)

GoogleMaps con Visual Studio 2008 – 24/04/2010

<http://www.datanoia.com/googlemaps-con-visual-studio-2008-c-parte-01.html>

Información general sobre DropDownList (Control de servidor Web) - 15/04/2010

[http://msdn.microsoft.com/es-es/library/xy14hwwb\(v=VS.80\).aspx](http://msdn.microsoft.com/es-es/library/xy14hwwb(v=VS.80).aspx)

Información general sobre el control ScriptManager - 02/05/2010

<http://msdn.microsoft.com/es-es/library/bb398863%28VS.90%29.aspx>

Insertar marcas en Google Maps – 02/05/2010

<http://www.desarrolloweb.com/articulos/insertar-marca-en-mapa-google.html>

Integridad Referencial MYSQL - 14/04/2010

<http://www.mysql-hispano.org/page.php?id=27&pag=1>

Introducción a Visual Studio - 30/05/2010

[ms-help://MS.VSCC.v90/MS.MSDNQTR.v90.es/dv\\_vstoc/html/99997089-56ff-4d60-81a9-447062dc98ac.htm](ms-help://MS.VSCC.v90/MS.MSDNQTR.v90.es/dv_vstoc/html/99997089-56ff-4d60-81a9-447062dc98ac.htm)

La evolución de la programación hacia la ejecución y validación automática de modelos - 18/05/2010

[http://www.epidataconsulting.com/tikiwiki/tiki-read\\_article.php?articleId=15](http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15)

MySQL Connector/ODBC - 30/05/2010

<http://dev.mysql.com/doc/refman/5.0/es/myodbc-connector.html>

Paginación en un control GridView de servidor Web - 10/05/2010

<http://msdn.microsoft.com/es-es/library/5aw1xfh3%28VS.80%29.aspx>

Panorámica del sistema de gestión de base de datos MySQL - 29/05/2010

<http://dev.mysql.com/doc/refman/5.0/es/what-is.html>

Portal Web para Consultas de Equipos con Acceso a una Base de Datos de la Empresa Corporecna - 30/05/2010

<http://bibdigital.epn.edu.ec/bitstream/15000/1474/1/CD-2221.pdf>

Sintaxis declarativa del control de servidor Web CheckBoxList - 19/04/2010

[http://msdn.microsoft.com/es-es/library/8bw4x4wa\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/8bw4x4wa(VS.80).aspx)

Tutorial: Empaquetar una solución Smart Device para su implementación - 31/05/2010

<http://msdn.microsoft.com/es-es/library/zcebx8f8.aspx>

Unified Modeling Language™ (UML®) - 18/05/2010

[http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm)

Uso de Adrotator - 13/05/2010

<http://migue.pezweb.com/?s=adrotator>

Validar con respecto a un intervalo de valores para controles de servidor ASP.NET - 14/04/2010

[http://msdn.microsoft.com/es-es/library/cc438191\(v=VS.71\).aspx](http://msdn.microsoft.com/es-es/library/cc438191(v=VS.71).aspx)

Validación de Formularios WebForms - 14/04/2010

[http://msdn.microsoft.com/es-es/library/cc437880\(VS.71\).aspx](http://msdn.microsoft.com/es-es/library/cc437880(VS.71).aspx)

Validar entradas requeridas para controles de servidor ASP.NET - 14/04/2010

[http://msdn.microsoft.com/es-es/library/cc438198\(v=VS.71\).aspx](http://msdn.microsoft.com/es-es/library/cc438198(v=VS.71).aspx)



What is SQLyog? - 29/05/2010  
<http://www.webyog.com/en/>