

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**ADAPTACIÓN DEL MODELO DE NEGOCIO DE LA INDUSTRIA
INMOBILIARIA ECUATORIANA AL SISTEMA LIBRE OPENERP PARA
LA COMERCIALIZACIÓN DE BIENES INMUEBLES.**

Previa a la obtención del Título de:

INGENIEROS DE SISTEMAS E INFORMÁTICA

POR: WASHINGTON ISRAEL PAREDES REYES

KATHERINE DEL ROCÍO ROBLES ARÉVALO

SANGOLQUÍ, OCTUBRE DE 2013

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. WASHINGTON ISRAEL PAREDES REYES y la Srta. KATHERINE DEL ROCÍO ROBLES ARÉVALO como requerimiento parcial a la obtención del título de INGENIEROS DE SISTEMAS E INFORMÁTICA.

Sangolquí, Octubre de 2013

Ing. Jenny Ruiz

DIRECTORA DE TESIS

Ing. Diego Marcillo

CODIRECTOR DE TESIS

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
AUTORÍA DE RESPONSABILIDAD

Nosotros, WASHINGTON ISRAEL PAREDES REYES y KATHERINE DEL ROCÍO ROBLES ARÉVALO declaramos que el proyecto denominado “ADAPTACIÓN DEL MODELO DE NEGOCIO DE LA INDUSTRIA INMOBILIARIA ECUATORIANA AL SISTEMA LIBRE OPENERP PARA LA COMERCIALIZACIÓN DE BIENES INMUEBLES”, ha sido desarrollado con base a una investigación profunda, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, autenticidad y alcance científico del proyecto de grado en mención

Sangolquí, Octubre de 2013

WASHINGTON ISRAEL PAREDES REYES

KATHERINE DEL ROCÍO ROBLES ARÉVALO

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotros, WASHINGTON ISRAEL PAREDES REYES y KATHERINE DEL ROCÍO ROBLES ARÉVALO autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución, del trabajo “ADAPTACIÓN DEL MODELO DE NEGOCIO DE LA INDUSTRIA INMOBILIARIA ECUATORIANA AL SISTEMA LIBRE OPENERP PARA LA COMERCIALIZACIÓN DE BIENES INMUEBLES”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, Octubre de 2013

WASHINGTON ISRAEL PAREDES REYES

KATHERINE DEL ROCÍO ROBLES ARÉVALO

DEDICATORIA

Dedico esta tesis primeramente a Dios nuestro Señor, y a las personas más importantes de mi vida:

A Dios, por ser mi guía espiritual desde mi niñez hasta mi presente, y a pesar de los errores cometidos a lo largo de mi vida, siempre está a mi lado. Él es mi fortaleza para seguir adelante, y para saber que cada acción en el presente tiene consecuencias en el futuro porque él es un juez justo. Él está conmigo en las buenas y en las malas, y aunque existan situaciones difíciles de entender, es con un propósito para mi formación.

A mi madre: Dora Arévalo Castillo, por su apoyo constante, por su ejemplo y por confiar siempre en mí. Gracias a ella he cumplido con mis metas propuestas, pues siempre ha sido un pilar fundamental en mi desarrollo personal, espiritual y profesional. Gracias mami por estar siempre conmigo, por nunca dejarme desfallecer y por hacerme una persona de bien. Por enseñarme a dar siempre lo mejor de mí, y decirme que la vida es de retos y de las personas que deciden ser más, solo con la perseverancia se logra superar uno mismo día a día para lograr la felicidad anhelada.

A mi padre: Víctor Manuel Robles, por sus consejos y por creer en los proyectos que me he propuesto, por su gran amor, y por estar conmigo en la adversidad.

A mis hermanos: Víctor, Jenny, Alexandra, Bismarck, Luz María y Cristhian, por su amor y sabiduría, por los consejos que me supieron brindar en su momento y por su apoyo constante en cada decisión tomada.

A mi sobrina Jennifer Ledesma, que más que una sobrina es como mi hermana, gracias por tus sabias palabras, que a pesar de ser menor a mí,

siempre tienes las palabras precisas para levantarme el ánimo y para continuar adelante. Gracias Heidita por tu amor, por tu cariño, por tu amistad. Doy gracias a Dios por tenerte no solo como mi sobrina, sino como mi mejor amiga.

A mi compañero de tesis Israel Paredes, gracias Washito por tu constante cariño, por tu amor, por estar conmigo siempre, uno no elige la familia, pero si somos responsables de elegir a nuestras amistades, estar contigo fue una de las mejores decisiones de mi vida. Doy gracias a Dios por haber cruzado nuestras vidas, porque a pesar de los errores cometidos, siempre estas a mi lado apoyándome en cada decisión.

Katherine

DEDICATORIA

Dedico el presente proyecto a mi madre, por su paciencia y su confianza en la culminación del mismo.

A mi padre, quien con su apoyo y exigencia me ha demostrado que cualquier logro que me plantee lo puedo conseguir.

A mis hermanas y hermano, quienes me han demostrado un gran cariño siempre y me han alentado a seguir adelante cumpliendo las metas planteadas.

A Katherine, con quien tras largos días de desvelo y mucho esfuerzo hemos logrado conseguir este tan anhelado reto.

Israel

AGRADECIMIENTOS

A Dios, por estar presente en cada momento de mi vida.

A mi familia, por estar siempre pendientes de mí, y por guiarme por el camino del bien.

A la Ing. Jenny Ruiz y Diego Marcillo, por su acertada dirección en este proyecto. Por haberme guiado de una manera ordenada en el seguimiento del mismo. Por brindarme las bases necesarias para obtener un proyecto de calidad, y saber que día a día se aprende algo nuevo.

Al Ing. Mauricio Campaña, por sus enseñanzas, por su valiosa amistad, por no solo ser un maestro sino también un amigo. Por estar siempre dispuesto ayudar a sus alumnos y saberlos guiar por el camino del bien.

A la Universidad de las Fuerzas Armadas - ESPE, por ser la brillante institución que me formó como profesional.

A mi compañero de tesis Israel Paredes, por su gran ayuda en el desarrollo de este proyecto. Por compartir juntos este reto y saborear el éxito del mismo.

A mis amigos, por su apoyo en todas las etapas de mi vida.

Katherine

AGRADECIMIENTOS

A Dios, por permitirnos llegar a alcanzar este logro profesional en nuestras vidas, gracias señor por tantas enseñanzas que dejas en nuestras vidas al culminar mi etapa universitaria y porque siempre tengo la fe en que estarás a mi lado hasta el final de mis días.

A mi familia, que siempre ha estado a nuestro lado dotándonos de buena energía para continuar adelante con nuestro proyecto. Gracias por su alegría y porque orgullosamente puedo decir que soy parte de una familia cuyo principio es el amor, la fe y la confianza. Gracias por la gran acogida que

tuvimos cada instante que debíamos realizar este proyecto y también por los momentos de relajación que nos brindaron después de ver como nuestras capacidades llegaban a su límite.

A nuestros amigos, que siempre aunque sea con una palabra de aliento nos daban a conocer su entusiasmo por éste triunfo, gracias porque de alguna manera ustedes también son parte de este logro.

A Vinicio y Jorge, compañeros y amigos de trabajo que estuvieron siempre dispuestos a ayudarnos cuando así lo requeríamos, gracias por su comprensión y apoyo constante.

A Katherine, mi compañera en todos los sentidos, no solo en el ámbito laboral y académico, sino también de buenos y malos momentos, gracias Kathita por estar siempre a mi lado brindándome tu cariño y amor.

A nuestros directores, que con su acertada guía nos dieron las pautas para nuestro crecimiento profesional, gracias por la paciencia que tuvieron al darnos su guía y por los conocimientos que nos transmitieron.

Israel

INDICE DE CONTENIDO

CAPÍTULO 1	1
ADAPTACIÓN DEL MODELO DE NEGOCIO DE LA INDUSTRIA INMOBILIARIA ECUATORIANA AL SISTEMA LIBRE OPENERP PARA LA COMERCIALIZACIÓN DE BIENES INMUEBLES.	1
1.1 Tema.....	1
1.2 Introducción	1
1.3 Planteamiento del Problema.....	2
1.4 Justificación e Importancia.....	3
1.5 Alcance	5
1.6 Limitaciones.....	9
1.7 Objetivos	10
1.7.1 Objetivo General.....	10
1.7.2 Objetivos Específicos	11
CAPÍTULO 2.....	12
MARCO TEÓRICO	12
2.1 La industria inmobiliaria en el Ecuador	12
2.1.1 Introducción.....	12
2.1.2 Reseña Histórica	12
2.1.2.1 Mercado Inmobiliario en Quito.....	13
2.2 Python Versión 2.7.....	14
2.2.1 Introducción.....	14
2.2.2 Características	15
2.2.3 Conceptos y definiciones.....	15
2.2.3.1 Concepto.....	15
2.2.3.2 Definiciones	16
2.2.4 Ventajas y desventajas.....	16
2.2.5 Comparación con otros lenguajes	18
2.3 OpenERP (versión 6.0).....	18
2.3.1 Introducción.....	18
2.3.2 Arquitectura MVC de OpenObject	18
2.4 Arquitectura técnica OpenERP (Cliente - Servidor)	20
2.3.4 Entorno de desarrollo	22
2.3.5 Desarrollo de módulos.....	23

2.4	Herramientas.....	26
2.4.1	Base de Datos: PostgreSQL	26
2.4.2	Repositorio: Bitbucket.....	27
2.5	Metodologías ágiles	28
2.5.1	¿Qué es una Metodología Ágil?	28
2.5.2	¿Qué significa ser ágil?	28
2.5.3	EL MANIFIESTO ÁGIL	29
2.5.4	CRITERIOS DE COMPARACIÓN DE LAS METODOLOGÍAS ÁGILES	31
CAPÍTULO 3.....		36
ANÁLISIS.....		36
3.1	Levantamiento de requisitos utilizando el estándar IEEE 830	36
3.1.1	Introducción	36
3.1.1.1	Propósito.....	36
3.1.1.2	Alcance	36
3.1.1.3	Personal involucrado.....	41
3.1.1.4	Definiciones, acrónimos y abreviaturas.....	41
3.1.1.4.1	Definiciones.....	41
3.1.1.4.2	Abreviaturas y Acrónimos.....	42
3.1.1.5	Referencias.....	43
3.1.1.6	Visión General del Documento	44
3.1.2	DESCRIPCIÓN GENERAL.....	45
3.1.2.1	Perspectiva del producto.....	45
3.1.2.2	Funcionalidad del producto	45
3.1.2.3	Características de los usuarios	48
3.1.2.3.1	Administrador	48
3.1.2.3.2	Jefe de Diseño e Ingenierías	48
3.1.2.4	Vendedor.....	49
3.1.2.4	Restricciones.....	49
3.1.2.5	Suposiciones y dependencias	50
3.1.3	Requisitos específicos	50
3.1.3.1	Requisitos comunes de las interfaces	50
3.1.4	Requisitos funcionales.....	51
3.1.4.1	ADMINISTRACIÓN DE USUARIOS.....	52

3.1.4.2 PARAMETROS INICIALES	53
3.1.4.3 INICIO DEL PROCESO DE COMERCIALIZACIÓN DE UN BIEN INMUEBLE	54
3.1.5 VALIDACIONES	61
3.1.6 Requisitos no funcionales.....	61
3.2 Historias de Usuario.....	64
3.2.1 Administración de usuarios.....	64
3.2.2 Validación de usuarios	65
3.2.3 Administración de tipos de inmuebles	67
3.2.4 Administración de tipos de ubicaciones.....	68
3.2.5 Administración de proyectos	69
3.2.6 Registros de tipos de inmuebles por proyecto.....	71
3.2.7 Ubicaciones por proyecto	72
3.2.8 Ingreso de tipos de inmuebles por ubicación.	74
3.2.9 Ingreso de los inmuebles de forma individual.....	75
3.2.10 Importación de inmuebles.....	77
3.2.11 Administración de provincias, cantones, ciudades y parroquias..	78
3.2.12 Precios por Inmueble.....	79
3.2.13 Promociones de Proyectos.....	80
3.2.14 Políticas de ventas por proyectos.....	82
3.2.15 Iniciativas.....	83
3.2.16 Oportunidades	84
3.2.17 Pedidos de Venta	86
 CAPÍTULO 4.....	 88
 DISEÑO.....	 88
4.1 Modelo de Casos de Uso	88
4.2 ESPECIFICACIÓN DE CASOS DE USO.....	96
4.3 Modelo Conceptual de Datos	132
4.4 Modelo Lógico de Datos	134
4.5 Modelo Físico de Datos	136
 CAPÍTULO 5.....	 138
 CONSTRUCCIÓN Y PRUEBAS	 138
5.1 Anexo 1. ZEN DE PYTHON.....	138
5.2 Anexo 2. TABLA COMPARATIVA DE LAS METODOLOGIAS ÁGILES.....	139
5.3 Anexo 3. Plantilla de las Historias de Usuario.....	140
5.4 Anexo 4. Plantilla de los Casos de Uso	141

5.5	Anexo 5. Plan de Pruebas	143
5.6	Anexo 6. Casos de Pruebas	150
5.6.1	Caso de Pruebas de cada requerimiento	150
5.7	Anexo 7. Reporte de Errores	173
5.8	Anexo 8. Manuales	175
	CAPÍTULO 6.....	203
	CONCLUSIONES Y RECOMENDACIONES.....	203
6.1	Conclusiones.....	203
6.2	Recomendaciones	204
	BIBLIOGRAFÍA.....	206

INDICE DE FIGURAS

Figura 1:	MATRIZ BCG (Beneficio Costo Ganancia).....	13
Figura 2:	Modelo Vista Controlador	19
Figura 3:	Arquitectura técnica OpenERP	20
Figura 4:	Arquitectura técnica de OpenERP	22
Figura 5:	Manifiesto ágil.....	30
Figura 6:	Administración de Usuarios	65
Figura 7:	Validación Usuarios	66
Figura 8:	Administración de Tipos de Inmuebles	68
Figura 9:	Administración de Tipos de Ubicaciones	69
Figura 10:	Administración de Proyectos	71
Figura 11:	Registro Tipos de Inmuebles por Proyecto.....	72
Figura 12:	Tipos de ubicación en un proyecto	74
Figura 13:	Tipos Inmuebles enlazados a una ubicación	75
Figura 14:	Registro de inmuebles forma individual	77
Figura 15:	Importación Inmuebles.....	78
Figura 16:	Administración lugar ubicación	79
Figura 17:	Precios por inmueble	80
Figura 18:	Promociones de Proyectos	81
Figura 19:	Políticas de Venta por Proyecto.....	83
Figura 20:	Inicio Comercialización Inmuebles.....	84
Figura 21:	Inicio de Comercialización Inmuebles.....	85
Figura 22:	Inicio de Comercialización Inmuebles.....	87
Figura 23:	Administración Usuarios	89
Figura 24:	Validación Usuarios	89
Figura 25:	Administración Tipos Inmuebles	90

Figura 26: Administración Tipos Ubicaciones	90
Figura 27: Administración de Proyectos	91
Figura 28: Registro Tipos Inmueble por Proyecto	91
Figura 29: Ubicaciones por Proyecto	92
Figura 30: Tipos de Inmueble enlazados ubicación	92
Figura 31: Registro Inmuebles forma individual	93
Figura 32: Importación Inmuebles	93
Figura 33: Administración lugar ubicación	94
Figura 34: Precios por Inmuebles	94
Figura 35: Promociones de Proyectos	95
Figura 36: Políticas de Venta	95
Figura 37: Proceso Comercialización Inmuebles	96
Figura 38: Módulo de Construcciones	132
Figura 39: Módulo de Comercialización	133
Figura 40: Módulo de Construcciones	134
Figura 41: Módulo de Comercialización	135
Figura 42: Módulo de Construcciones	136
Figura 43: Módulo de Comercialización	137
Figura 44: Creación Tipo de Objeto	175
Figura 45: Tipo Objeto Guardado	175
Figura 46: Creación Objetos	176
Figura 47: Objeto Guardado	176
Figura 48: Creación Tipos de Inmueble	177
Figura 49: Creación Tipos de Ubicación	177
Figura 50: Creación Marcas	178
Figura 51: Ingreso de Datos Generales	179
Figura 52: Ingreso Proyección Ventas	179
Figura 53: Ingreso datos Preventa	180
Figura 54: Ingreso de los detalles de los tipos de inmueble	181
Figura 55: Detalles de Inmueble	181
Figura 56: Detalles Inmueble	182
Figura 57: Detalles	182
Figura 58: Ingreso Ubicación por Proyecto	183
Figura 59: Ingreso Entrega General de Inmuebles	184
Figura 60: Ingreso de Inmuebles	185
Figura 61: Importación de Inmuebles	186
Figura 62: Creación Iniciativa	186
Figura 63: Conversión Iniciativa a Prospecto	187
Figura 64: Creación Prospecto	187
Figura 65: Ingreso datos del prospecto	188
Figura 66: Añadir Inmuebles	188
Figura 67: Búsqueda Inmuebles	188
Figura 68: Ingreso de Documentos	189
Figura 69: Pasos para creación documentos	190

Figura 70: Registro de Documentos	190
Figura 71: Paso para registro de documentos	191
Figura 72: Convertir de Prospecto a Cliente	191
Figura 73: Reservación Inmueble	192
Figura 74: Creación Presupuesto	192
Figura 75: Pedido de Venta	193
Figura 76: Pedido de Venta	193
Figura 77: Término de Pago	194
Figura 78: Búsqueda Término de Pago	194
Figura 79: Datos de Término de Pago.....	194
Figura 80: Envío a la IFI Comercial.....	195

RESUMEN

Las inmobiliarias en el Ecuador no cuentan con un sistema que integre la gestión de inventarios con la gestión de los clientes. El presente proyecto de tesis describe el proceso de desarrollo del sistema denominado COMMUN. El objetivo es definir la adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles. El propósito es optimizar el proceso que realizan las inmobiliarias del Ecuador ubicadas en la ciudad de Quito para poder comercializar bienes inmuebles. El sistema cuenta con dos módulos, el módulo de construcción y el módulo de comercialización. Para su análisis y desarrollo, primero se realizó el levantamiento de los requerimientos del software utilizando la Norma IEEE 830, luego se utilizó la metodología ágil XP (Programación Extrema) para tener una mejor forma de entendimiento de los requerimientos y a la vez retroalimentarse. Como resultado del presente proyecto se obtuvo un sistema adaptado a OpenERP que integra el departamento de construcciones con el departamento comercial, mejorando la comunicación interna dentro de la empresa y los tiempos de sus procesos comerciales. Como conclusión, el presente proyecto es totalmente escalable y se puede integrar al resto de módulos del OpenERP permitiendo así la extensión a procesos más grandes como por ejemplo financiero o gestión de reclamos.

Palabras Clave: Inmobiliaria, OpenERP, módulo, metodología, XP.

CAPÍTULO 1

ADAPTACIÓN DEL MODELO DE NEGOCIO DE LA INDUSTRIA INMOBILIARIA ECUATORIANA AL SISTEMA LIBRE OPENERP PARA LA COMERCIALIZACIÓN DE BIENES INMUEBLES.

1.1 Tema

Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.

1.2 Introducción

Actualmente la mayoría de empresas productoras, comerciantes y de servicio, utilizan aplicaciones informáticas para el control de sus procesos. Generalmente estos sistemas incluyen módulos de salidas, entradas, traslados, toma de pedidos, existencias, etc. Además estos sistemas pueden encontrarse implementados en ambientes cliente / servidor o Web.

Este tipo de sistemas aportan muchas ventajas en comparación a los sistemas netamente especializados, ya que centralizan gran cantidad de los procesos como facturación, control de inventario, contabilidad, etc., además que permiten la generación de reportes.

En el artículo 1 del decreto número 1014 de la presidencia de la República del Ecuador publicada el 1 de agosto del 2007, se decreta establecer como política para las entidades de la Administración pública central la utilización de Software Libre en sus sistemas y equipamientos informáticos (Correa Delgado, 2007); además el presidente Rafael Correa, dentro del marco del discurso de los "100 primeros días de gestión", señaló que es necesario que se haga uso del software libre en todas las entidades

públicas. "De esta manera, garantizaremos la soberanía de nuestros Estados, dependeremos de nuestras propias fuerzas, no de fuerzas externas a nuestra región".

Esto ha generado un gran impulso en el uso de herramientas de software libre desarrollándose proyectos principalmente en empresas públicas, además que los costos por implementación son relativamente más bajos debido a que no se necesita adquirir licencia alguna, consecuentemente esto ha impulsado a las empresas privadas a optar por una solución basada en software libre.

1.3 Planteamiento del Problema

Las empresas se han visto en la necesidad de automatizar sus procesos por medio de aplicaciones informáticas. Existen muchas soluciones en el mercado, en su gran mayoría son privativas, cuyo soporte y mantenimiento le obliga al cliente a contratar nuevamente un proveedor específico convirtiéndose el soporte de software en un monopolio a costos demasiados elevados.

Para el desarrollo del sistema de comercialización de bienes inmuebles, se realizó una investigación basada en las necesidades de un inmobiliaria en el Ecuador, específicamente en la ciudad de Quito para saber los problemas a los cuales se enfrentan, para esto se programaron reuniones y levantamientos funcionales y se llegó a la conclusión de que las inmobiliarias presentan varios riesgos, entre estos el de no poseer un módulo que les permita comercializar su producto final basado en la producción de los mismos, en dicho caso los inmuebles, porque en la actualidad toda la

humanidad se encuentra en una era digital y ningún proceso debería ser manual, sino automatizado.

La mayoría de sistemas existentes se enfocan a las empresas de manufactura, otros simplemente sirven para llevar un inventario de los inmuebles obtenidos y los vendidos, dejando de lado la gran ayuda que se tendría al poseer un sistema completamente centralizado que contenga módulos para recursos humanos, contabilidad, gestión con el cliente, entre otros. En la actualidad la oferta inmobiliaria está en auge en el Ecuador, por lo que las perspectivas en el futuro del sector inmobiliario son favorables.

1.4 Justificación e Importancia

La elaboración del módulo de Comercialización de inmuebles responde a la necesidad de las empresas inmobiliarias en el Ecuador ubicada en la ciudad de Quito, ya que automatizará y permitirá tener un mejor control, sobre el flujo que se maneja para comercializar un inmueble; además cabe recalcar que al incorporarse a OpenERP podrá utilizar los módulos con los que cuenta; entre los cuales se destacan los módulos de Ventas, Compras, Inventarios, Contabilidad, CRM, Administración de Proyectos y Recursos Humanos.

Con el desarrollo del sistema de comercialización de bienes inmuebles, las inmobiliarias se favorecerían enormemente pues no solo ahorrarían tiempo en dicho proceso sino que también incrementarían su cuota de mercado, porque se podría atender a varios casos con diferentes probabilidades de negocio a la vez y la gestión sería más eficiente.

El proyecto propuesto, acoplará el producto final de una inmobiliaria, es decir sus inmuebles, a los módulos propios de OpenERP con la finalidad de

sacar todo el potencial de ésta herramienta, de esta manera las inmobiliarias optimizarán sus procesos y podrán ir centralizando cada uno de ellos en una sola herramienta.

A continuación se explica porque elegir la herramienta “OpenERP”, sobre otras plataformas:

- **Libertad:** OpenERP como producto no pertenece a ninguno de sus distribuidores, tiene libertad para elegir al proveedor que más le convenga según sus necesidades.
- **Filosofía Open:** Puede contratar únicamente lo que necesite. Lo habitual es tercerizar todos los procesos de la implantación, pero quizá su empresa ya disponga de algunos recursos y sólo requiera desarrollo de algún módulo específico o formación/soporte técnico de algún módulo oficial.
- **Código abierto:** Al ser software libre, podrá disponer del código para realizar cualquier mejora sobre los módulos ya existentes, o crear uno nuevo adaptado a sus necesidades.
- **Conectividad con otros productos:** Visualización de informes en Adobe PDF, importación/exportación con Microsoft Office u OpenOffice, Google Maps, Mozilla Thunderbird, Magento, Joomla, y otros muchos, puede conectarse por medio de cualquiera de los 3 protocolos de servicios que el servidor le ofrece.
- **Gratuito:** Aunque resulte difícil de creer, OpenERP es un producto que no tiene coste de licencias. No tiene que pagar dinero por usarlo en más puestos de trabajo o renovar las costosas licencias

anualmente. Ese dinero puede aprovecharlo para otras mejoras informáticas o en otros departamentos.

- **Multiplataforma:** Actualmente tiene clientes de escritorio funcionales para GNU/Linux, Mac OS X y Windows, e incluso interfaz web para poder trabajar en otros sistemas (tablets, smartphones).

- **OpenObject:** Dispone de un API abierto para desarrollo rápido de aplicaciones administrativas.

- **Variedad/Flexibilidad:** OpenERP es un sistema que actualmente cuenta con más de 500 módulos liberados, que se pueden combinar para construir cualquier tipo de aplicación administrativa.

- **PostgreSQL:** El motor de base de datos de OpenERP es un potente desarrollo libre dirigido por una comunidad de desarrolladores y organizaciones comerciales que cuenta con clientes como Skype, Sony Online, Departamento de Trabajo de U.S.A, etc...

- **Fácil migración:** La herramienta oficial importa y exporta datos en formato .CSV para que le resulte más sencillo seguir trabajando con los datos de su aplicación actual.

Aplicando este sistema se pretende automatizar y centralizar la comercialización de inmuebles, generando de esta forma un aumento en la productividad y ayudando a mejorar su visión del negocio.

1.5 Alcance

El presente sistema de automatización de procesos para la comercialización de bienes inmuebles abarca desde la creación de un

proyecto de construcción hasta que un inmueble haya sido reservado por un cliente.

Este sistema se encuentra orientado a la web, de esta forma sería mucho más amigable para el usuario final y no será necesario instalar software alguno en las terminales que se comunicarán con el servidor. Al ser una implementación HTTP, se podrá acceder a ella desde cualquier navegador.

A continuación se describe lo que contendría dicho sistema.

1. Un Módulo de Construcciones

- a. Proyectos de construcción
- b. Ubicaciones por proyecto
- c. Tipos de inmuebles por proyecto
- d. Inmuebles de un proyecto

2. Un Módulo de Comercialización

- a. Definición de plan de ventas por proyecto.
 - i. **Precios:** aquí se va a detallar los precios de los diferentes inmuebles a comercializar.
 - ii. **Promociones:** aquí se detallaran las diferentes promociones, que la inmobiliaria ofrece a sus clientes para comercializar un determinado inmueble.
 - iii. **Políticas de ventas:** en esta sección se detallarán las políticas a las cuales van a estar sujetos el cliente y la inmobiliaria, para que en un futuro no existan inconvenientes en el área legal y comercial del bien inmueble.

- b. Gestión de relaciones con el Cliente
 - i. **Registro de una iniciativa:** este proceso permite registrar un posible cliente, ya que el mismo puede inscribirse en una feria o en una isla de un centro comercial, pero mientras no se llegue a un arreglo, este solo quedaría almacenado como una iniciativa.
 - ii. **Paso de una iniciativa a una oportunidad:** este proceso ayuda a definir el posible cliente, que es cuando el contacto lleva papeles importantes que le permiten dar un avance hacia la reserva del inmueble, y de esta forma ya se tomaría como una oportunidad.
 - iii. **Creación del cliente:** es cuando el prospecto ha llevado el pago de la reserva entre otros documentos importantes que le permiten ser el ganador de la reserva de los inmuebles por los cuales está interesado. En este punto ya pasará a ser cliente de la inmobiliaria.
- c. Reservas
 - i. **Tabla de Negociación:** esta tabla sirve para tener los SLA¹ entre el cliente y la inmobiliaria para saber cómo se van a realizar los pagos y conocer las fechas máximas para realizarlos.

1 SLA (Acuerdos a Nivel de Servicio): Es un documento en el cual se detallan todos los acuerdos a los que lleguen el cliente y la inmobiliaria para la creación del módulo de Construcciones, pues en este mismo se basarán para hacer las pruebas funcionales del sistema, el cual les servirá de respaldo para ambas partes para que no existan ambigüedades.

ii. Reserva del inmueble: la reserva de inmueble genera un pedido de venta dentro del OpenERP² poder ser contabilizado con otros módulos, además que creará registros dentro de la tabla de producto la cual es base para el módulo de inventarios.

3. Un módulo de seguridades

a. Permisos: incluye permisos para poder ejecutar cada acción con roles dentro de la empresa. Estos roles son:

i. **Administrador:** con acceso a todo el sistema y con todos los privilegios sobre el mismo.

ii. **Usuario de construcciones:** el operador del módulo de construcciones. Puede crear proyectos, asignar las ubicaciones, determinar los tipos de inmuebles y crear los inmuebles.

iii. **Administrador de construcciones:** el administrador del módulo de construcciones. Este puede definir los tipos de inmuebles que maneja la empresa, las ubicaciones que puede poseer un proyecto y acceder a las configuraciones del módulo.

iv. **Usuario de comercialización:** son los vendedores de la empresa, ellos podrán gestionar sus iniciativas, sus oportunidades y sus reservas.

v. **Administrador de comercialización:** estos usuarios podrán definir los precios de los inmuebles, las

² ERP (Enterprise Resource Planning, ERP por sus siglas en inglés): Es un sistema que ayuda a la gestión empresarial de una empresa, optimizando sus recursos.

políticas de ventas, promociones de proyectos y demás configuraciones del módulo comercial.

4. **Reportes**

a. **Prospecto:** reporte que servirá para archivar el prospecto con todos sus datos, el lugar y el vendedor que lo atendió. Formato PDF.

b. **Proforma:** reporte en formato PDF que llevará el cliente como informativo de los valores que deberá pagar para la adquisición del inmueble.

c. **Formulario de reserva:** Formulario que deberá ser firmado por el cliente para hacer válida su reserva.

d. **Contrato de reserva:** el contrato que deberán firmar el cliente y la inmobiliaria.

e. **Cancelación de reserva:** en caso de existir una cancelación de reserva se deberá imprimir este formulario en el cual se detallan las causas y los valores a devolver con las respectivas penalidades.

1.6 **Limitaciones**

- El presente trabajo, contempla la implementación de dicho sistema en una inmobiliaria de la ciudad de Quito para verificar su correcto funcionamiento de acuerdo a los requisitos levantados dentro de la empresa, basándose en pruebas funcionales con datos reales, pero este sistema no contempla el desarrollo particular, es decir solventar funcionalidades que únicamente rijan dentro de la empresa, en caso de que exista algún proceso que así lo requiera, la

inmobiliaria deberá llegar a un acuerdo con Atikasoft fuera de este proyecto de tesis para la implementación del mismo, cabe recalcar que el software a construirse va a ser genérico, y estará disponible para todas las inmobiliarias, ya que es libre y de código abierto, y se pueden hacer modificaciones al mismo, dependiendo de las necesidades de cada inmobiliaria.

- Este sistema estará acoplado al módulo base de ventas y CRM de OpenERP de donde se derivan los módulos de inventarios, contabilidad, entre otros; por lo que se entiende que se podrá acoplar libremente a cualquier otro módulo del ERP, sin embargo el presente sistema no contempla el involucramiento del resto de módulos.

- A continuación se detallan los módulos que se van a utilizar del OpenERP, y cuales no se utilizarán.

- o Módulos que se van a utilizar
 - Ventas
 - CRM
- o Módulos que se no se van a utilizar, para el presente sistema.
 - Contabilidad
 - Inventarios

1.7 Objetivos

1.7.1 Objetivo General

Desarrollar un sistema que se adapte al modelo de negocio de la industria inmobiliaria ecuatoriana para la comercialización de bienes inmuebles con las funcionalidades que brinda OpenERP.

1.7.2 Objetivos Específicos

- Determinar las necesidades de la industria inmobiliaria ecuatoriana para adaptar su modelo al OpenERP.
- Investigar la plataforma OpenObject, el cual es la base de OpenERP para la implementación de sistemas.
- Ofrecer una alternativa al desarrollo de sistemas en el Ecuador, utilizando el lenguaje de programación Python.
- Investigar que módulos de OpenERP sirven de soporte para la implementación del sistema y su comportamiento dentro del ERP.
- Realizar las pruebas funcionales del sistema.

CAPÍTULO 2

MARCO TEÓRICO

2.1 La industria inmobiliaria en el Ecuador

2.1.1 Introducción

Los propietarios, promotores e inversionistas inmobiliarios de todo el mundo necesitan cumplir con los requerimientos reglamentarios y medir los riesgos financieros.

El sector inmobiliario abre nuevas oportunidades para invertir y construir. La recuperación de este mercado impulsa la actividad económica, motiva a compradores, inversionistas y constructores por los nuevos nichos que se abrieron. Con las debidas precauciones el futuro de la industria es de los mejores.

2.1.2 Reseña Histórica

El proceso inmobiliario en las principales ciudades del Ecuador, ha presentado situaciones y cambios que han permitido ajustar el desarrollo de la

industria en los últimos 15 años. El mismo que afrontó dos etapas críticas: una a finales de la década de los 90 y otra a partir del año 2008, para solventar estos eventos se produjeron ajustes que lograron apalancar y retomar nuevamente el crecimiento.

Posteriormente la industria inmobiliaria se vio reforzada por un importante grupo de nuevos promotores y constructores que robustecieron la oferta y que conjuntamente con promotores “sobrevivientes” y fortalecidos de la década de los 90, generaron una nueva dinámica.

Con los cambios, principalmente centrados en variaciones sustanciales de la tasa de interés a la baja y a una mayor disponibilidad de recursos disponibles para el crédito hipotecario (a partir del año 2004), fue posible que el sector se enrumbara en una etapa de consolidación. El aporte de promotores en otros segmentos, como por ejemplo el de los centros comerciales, imprimió nuevas dinámicas al sector y contribuyó a equilibrarlo en épocas de desaceleración de la actividad normal de la construcción.

El mercado del Ecuador ha tenido un crecimiento importante aunque con situaciones diferentes en términos de la etapa ciclo de vida en que se encuentra la industria en cada ciudad. En el caso de la ciudad de Quito, se encuentra en un nuevo ciclo productivo, después de afrontar un exceso en la oferta disponible de vivienda, que produjo una crisis en el mercado de la ciudad.

2.1.2.1 Mercado Inmobiliario en Quito

2.1.2.1.1 Reseña Histórica

La vivienda nueva continuará en el mediano plazo, como el mercado más atractivo por su tamaño (m²) y por su tasa de crecimiento sostenido (matriz BCG).

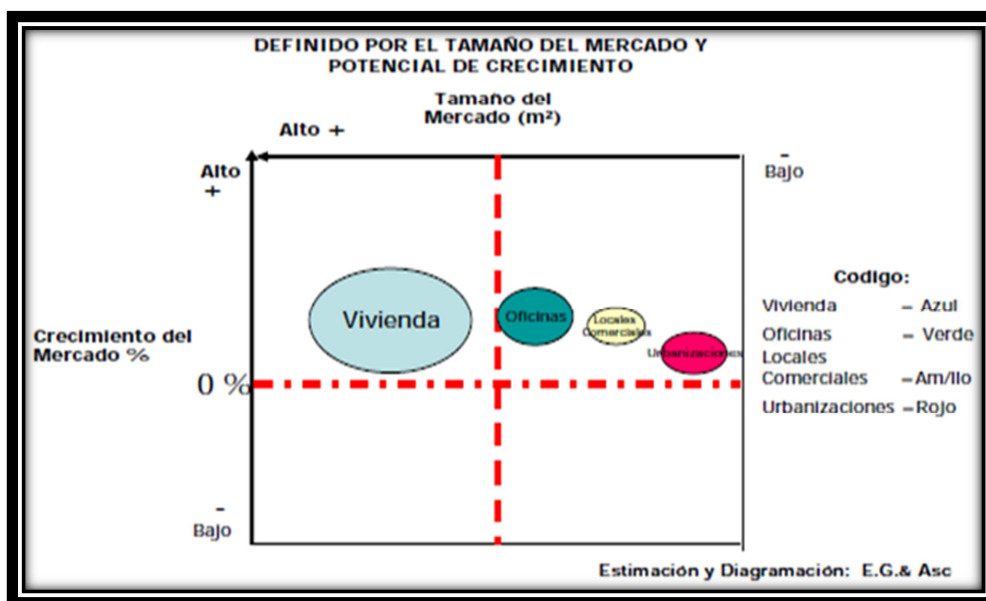


Figura 1: MATRIZ BCG (Beneficio Costo Ganancia)

A partir del año 2000 con la dolarización de la economía, el aumento de la oferta inmobiliaria disponible, la tendencia a la baja en los niveles de inflación, la disminución en las tasas de interés y la mayor disponibilidad de créditos para el comprador de vivienda, entre otros, crecieron las necesidades de vivienda lo que se reflejó en una mejora significativa en la demanda de los clientes de la industria.

2.2 Python Versión 2.7

2.2.1 Introducción

“Los programas deben escribirse para que los lean las personas, y sólo de forma circunstancial para que los ejecuten las máquinas”

Python es un lenguaje de programación interpretado y su nombre proviene de la afición de Van Rossum por el grupo de comedia británico llamado Monty Python. Como su autor principal, Guido mantiene poder de decisión sobre la dirección del equipo que desarrolla Python. Los dos grandes principios del software son la legibilidad y la transparencia, pero lo que el desarrollador Tim Peters ha compilado en El Zen de Python se ha convertido en una especie de rúbrica a seguir para quien programe usando este lenguaje.

Python se usa en programación de sistemas, cálculo numérico, desarrollo web, software para dispositivos móviles (Symbian, Android), desarrollo de aplicaciones de escritorio, educación, simulación, prototipados, GIS y un largo etc.

De las empresas más destacadas que utilizan Python se pueden nombrar Google, NASA, Facebook, US National Weather Service, Corel, Lockheed Martin, Pixar, Industrial Light and Magic, etc.

2.2.2. Características

- Python es un lenguaje de programación de propósito general, de muy alto nivel (esto es, un alto nivel de abstracción, con el uso de listas, tuplas, diccionarios).
- Python es un lenguaje interpretado (no es necesaria compilación), dinámico (no necesita identificar explícitamente los tipos de datos para inicializar variables, de modo que los tipos se validan durante la ejecución del programa) y fuertemente tipado (no pueden mezclarse tipos, es necesario hacer conversiones).

- Python es un lenguaje multiplataforma (Windows, Mac, Linux, etc.), multiparadigma (imperativo, orientado a objetos y en menor medida funcional) y con gestión automática de memoria.
- Por último cabe destacar que Python es un lenguaje de programación con una sintaxis clara y sencilla, fácil de aprender, donde se pueden mezclar los diferentes paradigmas de programación de los que dispone, ampliamente documentado, extensible, que intenta obligar al desarrollador de software a programar de la manera correcta en el menor tiempo posible.

2.2.3 Conceptos y definiciones

2.2.3.1 Concepto

Python, es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

2.2.3.2 Definiciones

Lenguaje interpretado o de script	Es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados).
Tipado dinámico	Se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo de valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.
Fuertemente tipado	No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente.
Multiplataforma	Está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.).
Orientado a	Es un paradigma de programación en el que los

Continúa...

objetos	conceptos del mundo real relevantes para el problema se trasladan a clases y objetos en el programa.
Filosofía	Los usuarios de Python se refieren a menudo a la Filosofía Python que es bastante análoga a la filosofía de Unix. El código que sigue los principios de Python de legibilidad y transparencia se dice que es "pythonico". Contrariamente, el código opaco u ofuscado es bautizado como "no pythonico". Estos principios fueron famosamente descritos por el desarrollador de Python Tim Peters en El Zen de Python (Anexo 1).

Tabla 1. Definiciones de python.

2.2.4 Ventajas y desventajas

2.2.4.1 Ventajas

- Una de las grandes ventajas de Python es que no obliga a los programadores a adoptar un estilo particular y permite la elección de varios paradigmas: programación orientada a objetos, programación estructurada y programación funcional, además de soportar otros paradigmas a través del uso de extensiones.
- La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y más portables.
- Es el lenguaje utilizado en el entorno gráfico Sugar y en la mayoría de sus actividades, por lo cual lo hace el lenguaje natural de la computadora XO.
- Facilita el intercambio con otros miembros de la comunidad de desarrolladores Sugar - OLPC ya que es el lenguaje más difundido para los desarrollos en Sugar.
- Es más sencillo el desarrollo, ya que es un lenguaje de script. Esto permite desarrollar en un entorno con un intérprete interactivo, lo que facilita experimentar con características del lenguaje o probar

funciones durante el desarrollo del programa. No requiere proceso de compilación.

- Es multiparadigma, por lo que permite varios estilos de programación: orientada a objetos, estructurada o funcional.
- Se pueden escribir nuevos módulos fácilmente en C o C++.
- Hay más programadores Python en el mercado que lo que hay, por ejemplo, programadores C++. El lenguaje tiene una barrera de entrada más accesible.

2.2.4.2 Desventajas

- Es menos performante que C++ debido a que es un lenguaje interpretado. Para cierto tipo de videojuegos que requieran mucha performance puede no ser adecuado el uso de Python.
- Si no se codifica prolijamente se podría dificultar la lectura del código o introducir errores indeseados.
- Puede llevar mucho tiempo encontrar determinado tipo de errores, característicos de los lenguajes de script.

2.2.5 Comparación con otros lenguajes

LENGUAJE DE PROGRAMACIÓN	FUNCIONALIDAD	ORIENTACIÓN A OBJETOS
PYTHON	No necesita compilación, ya que es un lenguaje interpretado.	Para la creación de objetos no es necesario instanciar la variable.
JAVA	Necesita compilación	Es necesaria la instancia de variables.
VISUAL BASIC	Necesita compilación	Es necesaria la instancia de variables.
C#	Necesita compilación	Es necesaria la instancia de variables.

Tabla 2. Comparación de python con otros lenguajes.

2.3 OpenERP (versión 6.0)

2.3.1 Introducción

Open ERP (Enterprise Resource Planning, ERP por sus siglas en inglés) es un sistema planeador de recursos empresariales que permite realizar una gestión integrada de los recursos. Entre sus características están la contabilidad analítica, contabilidad financiera, gestión de almacenes/inventario, gestión de ventas y compras, automatización de tareas, campañas de marketing, ayuda técnica (Helpdesk), y punto de venta, dentro de la construcción misma del software se hace uso intensivo de flujos de trabajo que se puede integrar con los módulos haciendo la modificación de aprobación y en general de cualquier proceso adaptable.

2.3.2 Arquitectura MVC de OpenObject

OpenERP utiliza el patrón de arquitectura de software Modelo Vista Controlador (Figura 2).

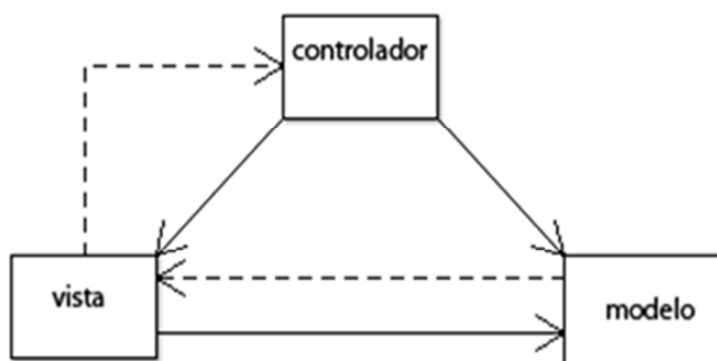


Figura 2: Modelo Vista Controlador

Modelo Vista Controlador (MVC), es un patrón de arquitectura de software que separa en tres componentes distintos:

- Modelo: Los datos de la aplicación
- Vista: La interfaz de usuario
- Controlador: La lógica de control

El patrón MVC se ve frecuentemente en aplicaciones web donde:

- La vista es la página HTML y el código que provee de datos dinámicos a la página.
- El modelo es el Sistema de Gestión de Base de Datos y la lógica de negocio.
- El controlador es el responsable de recibir los eventos de entrada desde la vista, consultar datos del modelo, realizar los cálculos necesarios y solicitar nuevas vistas.

2.3.2.1 Modelo Vista Controlador en OpenERP

En OpenERP, se puede aplicar este modelo-vista-controlador de la siguiente manera:

- **Modelo (model):** Los objetos de OpenERP con sus columnas que normalmente se guardan en las tablas de PostgreSQL con sus campos. Permite la creación/actualización automática de las tablas y acceder a las tablas sin usar SQL.
- **Vista (view):** Listas, formularios, calendarios, gráficos, definidas en archivos XML. En estos archivos también se definen menús, acciones, informes, asistentes, etc.
- **Controlador (controller):** Métodos Python definidos dentro de los objetos de OpenERP que proporcionan la lógica: Validación de datos, cálculos, etc.

2.4 Arquitectura técnica OpenERP (Cliente - Servidor)

OpenERP posee una arquitectura multiusuario de tres capas. La capa de negocio se define como un núcleo permitiendo así adicionar múltiples módulos, dando una funcionalidad diferente a OpenERP. (Figura 3)

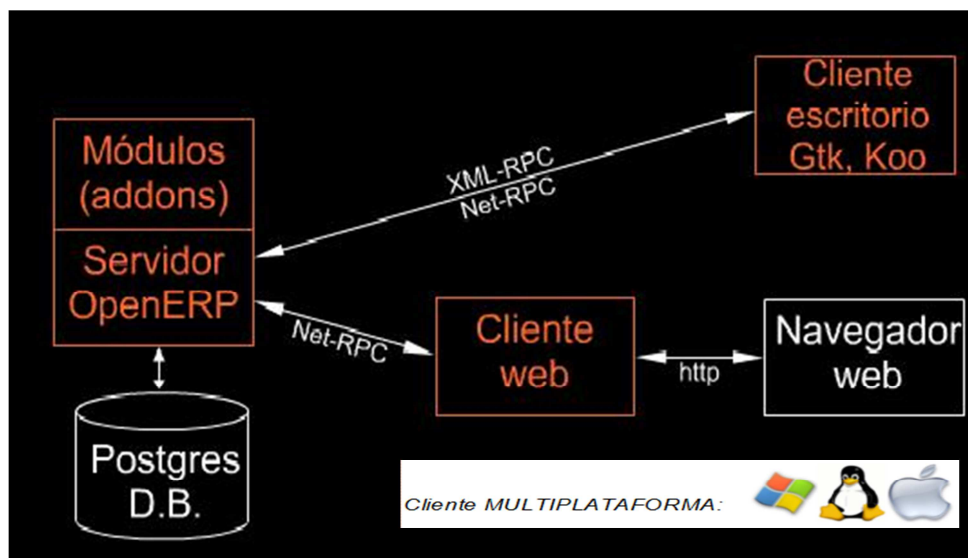


Figura 3: Arquitectura técnica OpenERP

El lenguaje para escribir los módulos es Python. La funcionalidad de los módulos es expuesta utilizando el protocolo XML-RPC y NET-RPC (configurable en el servidor). Los módulos hacen uso de la herencia de las clases ORM de OpenERP para persistir los datos en Postgres. Los módulos pueden insertar datos en su instalación por medio de archivos XML, CSV y YML.

2.4.1 El servidor OpenERP

OpenERP proporciona un servidor de aplicaciones en donde las aplicaciones de negocio son construidas a partir de la instalación de nuevos módulos. Permite un desarrollo completo pudiendo dar la funcionalidad requerida. Se destacan su ORM flexible, una arquitectura MVC, la carga de información extensible.

Desde el punto de vista de desarrollo, el servidor actúa como una librería de aplicaciones, ocultando los objetos de bajo nivel. Posee una forma sencilla de configurar y ejecutar las aplicaciones.

2.4.2 Los Módulos

Los módulos son la capa de negocio de OpenERP, sin ellos no tendría ninguna funcionalidad. OpenERP es solo un intérprete de los módulos los cuales darán la funcionalidad necesaria solventando las necesidades de las empresas.

Un módulo se compone generalmente de los modelos de datos, datos iniciales, definiciones de vistas, asistentes, las especificaciones de flujos de trabajo y los informes. La creación de los modelos de datos se definen en el core escrito en python mientras que los datos iniciales, definiciones de vistas y flujos de trabajo no son sino datos de los modelos de datos definidos en archivos XML.

2.4.3 Clientes

Los clientes pueden comunicarse con OpenERP por medio de los protocolos XML-RPC o el protocolo NET-RPC que pronto desaparecerá siendo reemplazado por JSON-RPC en versiones futuras. Tanto XML-RPX como a futuro JSON-RPC permiten la creación de clientes escritos en diferentes lenguajes de programación. OpenERP S.A desarrolla dos clientes, uno por escrito con GTK de python y un cliente web que permite la operación en cualquier navegador del mercado.

Gracias a que la lógica reside en el servidor, los clientes simplemente deben lanzar una petición y desplegar su respuesta, las versiones cliente desarrolladas por OpenERP permiten desplegar estos datos en diferentes

maneras como formularios, listas, calendarios, etc. Después de las acciones del usuario simplemente envían peticiones con modificaciones de datos.

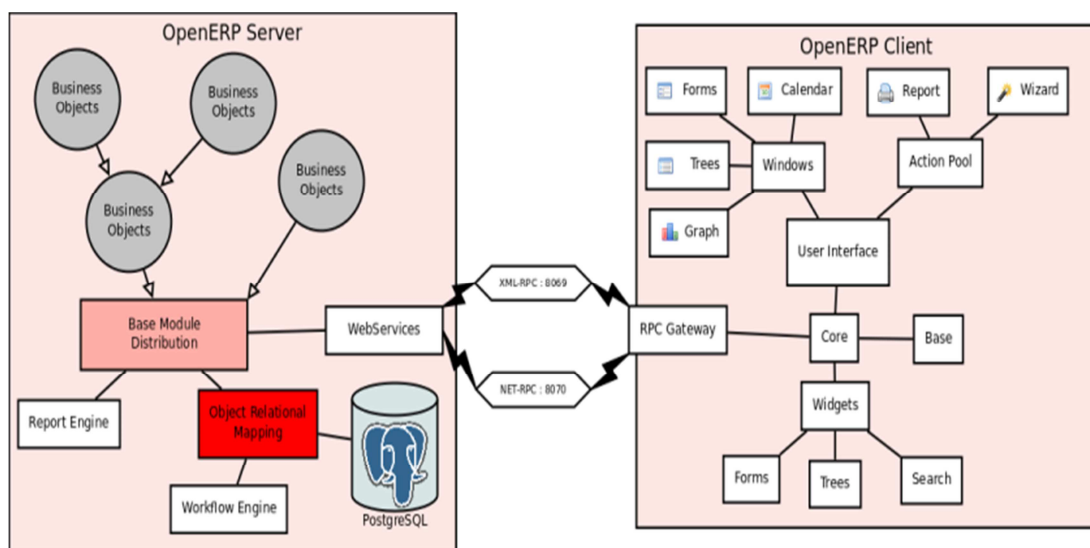


Figura 4: Arquitectura técnica de OpenERP

2.3.4 Entorno de desarrollo

Para desarrollar módulos para OpenERP se puede incluso usar el editor vi; sin embargo para efectos de éste proyecto nos basaremos en el entorno de desarrollo integrado de código abierto y multiplataforma llamado Eclipse el cual posee un proyecto externo enfocado al desarrollo en python llamado PyDev.

Las únicas configuraciones que necesita Eclipse para empezar a desarrollar módulos de OpenERP es la instalación del entorno para python de PyDev y el entorno para XML; sin embargo se pueden instalar las plantillas de código python y XML para OpenERP que se encuentran disponibles gracias al aporte de la comunidad en página oficial de OpenERP³

³ <http://openerp-eclipse-template.googlecode.com/svn/trunk/>

Para un entorno de desarrollo colaborativo y para el proyecto OpenERP se utiliza launchpad⁴. A más de ser un sitio de alojamiento de código que utiliza Bazaar⁵ como control de versiones, posee funcionales extendidas como un sistema de seguimiento de errores donde se puede informar de bugs⁶ en las aplicaciones, un sitio de traducciones colaborativa, y uno de ayuda a la comunidad entre las más destacadas.

2.3.5 Desarrollo de módulos

Como se mencionó anteriormente el núcleo de OpenERP y sus módulos están escritos en python. Los datos que cargan los módulos son definidos por medio de archivos XML, CPV y YML.⁷

Los módulos, también llamados addons, son una vía simple para estructurar un sistema completo, ellos extienden la funcionalidad de OpenERP.

2.3.5.1 Directorio y archivos de los módulos

Todos los módulos se encuentran dentro del directorio addons del servidor. Para crear un nuevo módulo es necesario:

- Crear un directorio con el nombre del módulo dentro del directorio addons
- Crear el archivo de descripción OpenERP del módulo (`__openerp__.py`)
- Crear el archivo de descripción python del módulo (`__init__.py`)

4 Launchpad: Sitio web que apoya al desarrollo de software libre.

5 Bazaar: Repositorio para el desarrollo colaborativo de software.

6 Bugs: Es un error de código que se corrige a través de un parche.

7 XML, CPV y YML: Son archivos, en los cuales se maneja la vista del módulo, aquí es donde se programa para ver como quedara el sistema en la web.

- Crear los archivos de python que contienen la descripción de los objetos

- Crear los archivos XML para la definición de datos (vistas, menús, etc.).

Opcionalmente y dependiendo de los requisitos del sistema se crearán los reportes, asistentes y flujos de trabajo.

2.3.5.2 Archivos XML

Los archivos XML son usados para crear/modificar los registros de la base de datos con varios propósitos, entre los cuales podemos citar:

- Datos (demo / necesarios)
- Declaración de vistas
- Declaración de reportes
- Declaración de asistentes
- Declaración de Flujos de trabajo

2.3.5.3 Archivo de descripción del módulo Python

El archivo `__init__.py`, como en cualquier módulo de python, ejecutado al iniciar el programa, indica los archivos python a importar necesarios para cargar.

Si se crea un archivo "modulo.py" que contiene la descripción de los objetos del módulo, necesariamente debemos importarlo en el archivo "`__init__.py`" para que sea cargado: "import modulo"

2.3.5.4 Archivo de descripción del módulo OpenERP

Este archivo es necesario para que un módulo de OpenERP sea considerado como tal. Además de ser especificada la información del módulo, es responsable de determinar los archivos (XML, CPV, YML) que

serán cargados con la instalación del módulo y también de especificar las dependencias hacia otros módulos.

El archivo consta de un solo diccionario de python con las siguientes claves:

- name: el nombre del módulo (por estándar debe ser en inglés)
- version: La versión del módulo
- description: Un string de texto que determina que hace el módulo
- author: el autor del módulo
- website: El sitio web del autor del módulo
- license: La licencia del módulo, por defecto GPL-2
- depends: Lista de módulos de los cuales el módulo a ser creado depende. Al menos el módulo “base” debe ser especificado debido a que posee datos necesarios para la creación de vistas, reportes, etc.
- init_xml: Lista de archivos .XML a cargar cuando el servidor se lanzado con la opción “--init=modulo”. Los paths son relativos al directorio del módulo.
- update_xml: Lista de archivos .xml a cargar cuando el servidor se lanzado con la opción “--update=modulo”. Los paths son relativos al directorio del módulo.
- demo_xml: Lista de archivos .xml a cargar cuando la instalación en la base de datos ha sido especificada con datos demo. Los paths son relativos al directorio del módulo.

- `installable`: Booleano que determina si el módulo será instalable o no.
- `active`: Booleano por defecto `false` que determina si el módulo se instala automáticamente con la creación de una base de datos.

Los archivos especificados en `init_xml` son los relativos a flujos de trabajo y datos a cargar en la instalación del módulo así también ciertos datos demostrativos.

Los archivos definidos en `update_xml` son los relativos a las vistas, reportes y asistentes.

2.4 Herramientas

2.4.1 Base de Datos: PostgreSQL

PostgreSQL es un sistema de base de datos relacional libre (open source) que utiliza OpenObject para representar el modelo relacional de los sistemas administrativos. PostgreSQL, tiene más de 15 años de activo desarrollo y arquitectura probada, por lo que se considera un sistema seguro y confiable para la integridad de los datos. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

PostgreSQL utiliza control de concurrencia de Multi-Versión (MVCC), puntos de recuperación (Silberchatz, 1998) a un momento dado, tablespaces, replicación asincrónica, transacciones jerarquizadas (savepoints), backups en línea, un sofisticado analizador/optimizador de consultas y write ahead loggin para tolerancia de falla.

Además, el código fuente de PostgreSQL está disponible bajo la licencia BSD. Esta licencia proporciona la libertad de usar, modificar y distribuir

PostgreSQL. Todas las modificaciones, mejoras, o cambios que se realicen quedan a decisión del usuario de publicar o no hacerlo.

2.4.2 Repositorio: Bitbucket

Para alojar el código fuente que se desarrollará en este tema, se utilizará Bitbucket el cual se basa en Mercurial para el control de versiones.

Se decide utilizar Bitbucket debido a que el sistema será desarrollado solamente por los participantes del proyecto de tesis, es decir que no es un trabajo colaborativo a grande escala como para utilizar Launchpad.

“Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git.” (Fundación Wikimedia, 2012).

Bitbucket ofrece cuentas comerciales y gratuitas. En la actualidad con una cuenta gratuita se puede tener un ilimitado número de repositorios privados en el cual pueden colaborar un máximo de cinco personas. Permite administrar los proyectos desde su propia interfaz en la web además que permite el control por equipos. Proporciona una manera fácil de revisar los cambios en cada versión.

“Mercurial es un sistema de control de versiones multiplataforma, para desarrolladores de software. Está implementado principalmente haciendo uso del lenguaje de programación Python, pero incluye una implementación binaria de diff escrita en C.” (Fundación Wikimedia, 2012)

2.5 Metodologías ágiles

2.5.1 ¿Qué es una Metodología Ágil?

Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores que las

metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

2.5.2 ¿Qué significa ser ágil?

El objetivo de los métodos ágiles es permitir que una empresa sea ágil, que significa ser capaz de “Entregar rápido, cambiar rápido, cambiar con frecuencia”⁸. La verdadera agilidad es más que un conjunto de prácticas, es un estado de ánimo.

De esta forma, el punto de vista de las metodologías ágiles que se pretende utilizar está más próximo al de esta cita:

“El factor más importante en el desarrollo de software no son las técnicas y las herramientas que emplean los programadores, sino la calidad de los propios programadores”⁹

2.5.3 EL MANIFIESTO ÁGIL

En un taller de dos días en Snowbird, Utah, en febrero de 2001 se reunieron los representantes de cada una de las metodologías ágiles. Había mucho en común, y este reconocimiento era mucho mayor que las diferencias entre los procesos. En la reunión se asignó el término “Métodos Ágiles” para definir los métodos que estaban surgiendo como alternativa a las metodologías formales, consideradas excesivamente “pesadas” y rígidas por su carácter normativo y su fuerte dependencia de planificaciones detalladas, previas al desarrollo.

8 Jim Highsmith.

9 Robert L. Glass. Facts and Fallacies of Software Engineering

El manifiesto ágil sugiere un enfoque orientado a la participación de los usuarios y clientes, más que hacia los procesos y herramientas, trabajando más en el software y menos en la documentación, colaborando más con los clientes en vez de estar negociando y respondiendo a los cambios sacrificando el plan de trabajo si es necesario.

En la ilustración 3, se observa la conclusión a la que llegaron los críticos de los modelos de mejora basados en procesos, que son un resumen en forma de cuatro principios, que actualmente se conoce como “Manifiesto Ágil”, que básicamente son los principios sobre los que se basan los métodos reconocidos como ágiles.

Manifiesto Ágil (<http://www.agilemanifesto.org>)

Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

- * A los individuos y su interacción, por encima de los procesos y las herramientas.
- * El software que funciona, por encima de la documentación exhaustiva.
- * La colaboración con el cliente, por encima de la negociación contractual.
- * La respuesta al cambio, por encima del seguimiento de un plan.

Firmado por: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.

Figura 5: Manifiesto ágil

2.5.3.1 Los cuatro principios del manifiesto ágil

1. Valoramos más a los individuos y su interacción que a los procesos y las herramientas.

En este principio no se quita la importancia que tienen tanto los procesos como las herramientas, solo que recalca que los individuos son lo más importante ya que los procesos deben ser una ayuda y un soporte para

guiar el trabajo, deben adaptarse a la organización, a los equipos y a las personas; y no al revés, y de esta manera se llega a obtener resultados extraordinarios porque se cuenta con personas preparadas para dicha área.

2. Valoramos más el software que funciona que la documentación exhaustiva.

En este principio no se manifiesta que la documentación no sea importante para que un sistema funcione de la manera deseada, ya que para cuestiones legales o normativas son obligatorios, pero ellos resaltan que son menos trascendentales porque no aportan un valor al producto, es decir que los documentos no pueden sustituir, ni pueden ofrecer la riqueza y generación de valor que se logra con la comunicación directa entre las personas a través de la interacción con los prototipos.

3. Valoramos más la colaboración con el cliente que la negociación contractual.

En este principio se valora más la colaboración del cliente, ya que el mismo es un miembro del equipo que se integra y colabora en el grupo de trabajo. Ya que ellos resaltan que un contrato no aporta valor al producto, sino que es una formalidad entre cliente y proveedor.

4. Valoramos más la respuesta al cambio que el seguimiento de un plan

En este principio valoran más las diferentes respuestas a los distintos cambios que puede existir durante el desarrollo del módulo, ya que surgen de un entorno inestable, antes que seguir un plan preestablecido, porque ellos dicen que no se puede seguir al pie de la letra un plan porque lo consideran trascendental.

2.5.4 CRITERIOS DE COMPARACIÓN DE LAS METODOLOGÍAS ÁGILES

En esta sección se hace referencia a los distintos criterios de comparación, para lo cual se hará un cuadro comparativo con las principales metodologías que existen.

Se tomaron como referencia tres puntos para hacer la comparación, que son el Ciclo de vida del proyecto, el estado actual y las herramientas de las diferentes metodologías, se hizo la comparación entre las principales las cuales son XP, Scrum, DSDM, AUP, AMDD, FDD.¹⁰

1. CICLO DE VIDA DEL PROYECTO

Para este criterio se va a considerar las siguientes etapas como parte del ciclo de vida de un proyecto estándar:

- Principio del proyecto.
- Especificación de requisitos.
- Análisis y Diseño.
- Codificación
- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de sistemas.
- Pruebas de aceptación.
- Sistema en uso o mantenimiento.

2. ESTADO ACTUAL DE LA METODOLOGÍA

Para este criterio se tomará en cuenta, los siguientes estados en los cuales se podría encontrar las metodologías en estudio:

¹⁰ XP, DSDM, AUP: Metodologías de Programación Extrema, AMDD: Agile model driven development, FDD: Feature Driven Development.

- **Recién nacida**, metodologías que tiene un año o menos y de la cual no existen evidencias ni estudios.
- **En construcción**, metodologías con más de un año de existencia, pero que no dispone de evidencia documentada.
- **Activa**, metodologías que llevan muchos años en el desarrollo del software y de las cuales se puede encontrar evidencias y estudios que corroboren su efectividad.
- **Olvidada**, metodologías que llevan tiempo sin ser utilizadas y de las cuales no se encuentra evidencia.

3. HERRAMIENTAS

Con este criterio se busca dar a conocer las distintas herramientas que las metodologías en estudio requieren para cumplir cada una de las tareas que especifica su enunciado. Para esto se enfatizó la búsqueda de herramientas de libre distribución.

Ver Tabla (Anexo 2)

CONCLUSIONES DE LA COMPARACIÓN

Al finalizar las comparativas de las herramientas seleccionadas se extrae las siguientes conclusiones:

1. No todas las metodologías ágiles contemplan todo el ciclo de vida tal y como lo hemos visto tradicionalmente, como se puede apreciar en la Tabla 1, presentada en la comparativa de ciclo de vida de un proyecto, la metodología que contempla todas las etapas es la metodología AUP, esto debido a que es una metodología que se basa en RUP, pero a la vez esta metodología sugiere que solo se utilice las

etapas que sean necesarias para el proyecto, con lo cual puede que se cumpla con toda o parte de las etapas tradicionales.

2. Al realizar la comparación, se realizó una ponderación, y se obtuvo que la metodología ágil más opcional sería AUP, porque cumple con todo el ciclo de vida del proyecto, se encuentra en un estado activa y utiliza diferente herramientas para su desarrollo e implementación, de ahí le sigue la metodología XP, con un puntaje de 24, no cumple con todo el ciclo de vida del proyecto ya que no cuenta con todas las fases como en el caso del principio del proyecto pero cumple con las otras bases en función PD y BA¹¹, aunque no cuenta con un SG¹², sin embargo dicha metodología se ha elegido para el proyecto de Tesis, ya que el desarrollo del mismo es un proyecto a mediano plazo, y esta metodología va de la mano con el desarrollo porque se realizan pruebas constantes en cada proceso del proyecto ya que se enfoca en la fase de pruebas, y existe una interacción continúa entre el cliente y los desarrolladores, y de esta manera ya se tendría un prototipo de lo que realmente el cliente desea con dicho sistema, ya que el mismo se encargara de verificar la calidad del mismo no solo en parámetros de fiabilidad sino de funcionalidad, en esta metodología no se maneja documentación exhaustiva, sin embargo se utiliza la documentación necesaria para el desarrollo del

11 PD: Se describe un proceso en el método que incluye esta etapa

12 BA: Buenas prácticas, actividades y artefactos considerados en la etapa. SG: Soporte a la gestión.

software. Cabe recalcar que para el levantamiento de requerimientos se utilizara IEEE 830.¹³

3. El estado actual de las metodologías ágiles es activo y va ganando cada vez más adeptos. Las metodologías que se han seleccionado no podían sino indicarnos esto, ya que fue uno de los criterios de selección de las mismas, y que son una realidad del panorama del software actual.

4. Con relación a las herramientas de distribución libre, como se ha podido apreciar hoy en día existen muchas herramientas que ayudan ya en el proceso de las metodologías ágiles, en el cuadro se puede observar que la mayor herramientas tiene es AUP, de ahí le sigue XP que cuenta con las herramientas para refactorización, administración de proyectos y compilación automáticas, repositorio de códigos y medición del rendimiento.

5. Con respecto a las áreas estudiadas, algunas metodologías están más enfocadas a presentar diferentes prácticas como es el caso de XP y AUP, otras a la gestión de proyectos, como Scrum, DSDM.

6. En base a la estado actual, se puede observar que las metodologías en comparación se encuentran activas, y de esta manera se puede llegar a la conclusión que las mismas ya tienen un grado de madurez elevado, o al menos a las que se está haciendo

13 IEEE 830: Es un estándar en donde indica que un buen documento de requisitos debe contemplar toda la información presentada en dicho estándar y, aunque propone una organización de dicha información, no exige estrictamente el formado de dicha información, aunque se lo puedo tomar como referencia para la Especificación de Requisitos Software.

referencia, las mismas empiezan a dar señales de que son una realidad palpable y que indican que un gran número de empresas están dando el salto y dejando de experimentar con ellas para utilizarlas seriamente.

CAPÍTULO 3

ANÁLISIS

3.1 Levantamiento de requisitos utilizando el estándar IEEE 830

3.1.1 Introducción

3.1.1.1 Propósito

El propósito del presente documento es ayudar tanto al analista como al cliente a obtener un documento en el cual se detalle todo acerca de los requisitos tanto funcionales como no funcionales del sistema, para que no existan ambigüedades del mismo. Porque el cliente no suele entender el proceso de diseño y desarrollo del software como para redactar una especificación de requisitos software (ERS) y los analistas no suelen entender completamente el problema del cliente, debido a que no dominan su área de trabajo. Por tal razón con este documento se pretende facilitar el entendimientos a ambas partes para lograr un sistema que cumpla con los puntos más relevantes del cliente al momento de comercializar un bien inmueble y de esta manera queden compensados tanto cliente como proveedor del servicio en este caso analista y programadores.

3.1.1.2 Alcance

El presente sistema de automatización de procesos para la comercialización de bienes inmuebles abarca desde que un bien inmueble

ha sido construido y está listo para su comercialización hasta que el mismo haya sido vendido y entregado al cliente final.

Este sistema estará en un ambiente web, de esta forma sería mucho más amigable para el usuario final y poseerá todos sus procesos centralizados, manejados desde una perspectiva más óptima.

A continuación se describe lo que contendría dicho sistema.

- Un Módulo de Construcciones

Este módulo abarcaría los siguientes puntos:

1. Construcción del inmueble

- Definición de plan de ventas por proyecto.
 - Precios: Aquí se va a detallar los precios de los diferentes inmuebles a comercializar.
 - Promociones: Aquí se detallaran las diferentes promociones, que la inmobiliaria ofrece a sus clientes para comercializar un determinado inmueble.
 - Políticas de ventas: En esta sección se detallarán las políticas a las cuales van a estar sujetos el cliente y la inmobiliaria, para que en un futuro no existan inconvenientes en el área legal y comercial del bien inmueble.

2. Construcción del Cliente

- a) Registro de un contacto: este proceso permite registrar un posible cliente, ya que el mismo puede inscribirse en una feria o en una isla de un centro comercial, pero mientras no se llegue a un arreglo, este solo quedaría almacenado como un contacto.

b) Paso de un contacto a prospecto: este proceso ayuda a definir el posible cliente, que es cuando el contacto lleva papeles importantes que le permiten dar un avance hacia la reserva del inmueble, y de esta forma ya se tomaría como un prospecto.

c) Creación del cliente: es cuando el prospecto ha llevado el pago de la reserva entre otros documentos importantes que le permiten ser el ganador del inmueble por el cual está interesado. En este punto ya pasará a ser cliente de la inmobiliaria.

3. Permisos: Incluye permisos para poder ejecutar cada acción con roles dentro de la empresa.

a) Estos roles serían:

a. Administrador

i. Acceso a todo el sistema y con todos los privilegios sobre el mismo.

b. Usuario de Construcciones

i. Acceso a la parte de creación de proyectos hasta la creación de los inmuebles.

c. Usuario de Comercialización

i. Es el usuario encargado de establecer los precios de los proyectos, las promociones y los descuentos.

d. Vendedor

i. Es el usuario encargado de crear los contactos, prospectos, y reservar inmuebles.

4. Tabla de Negociación: Esta tabla sirve para tener los SLA entre el cliente y la inmobiliaria para saber cómo se van a realizar los pagos y conocer las fechas máximas para realizarlos.

5. Registro de Pagos: Incluye el registro de pagos realizados, esta parte es independiente de la tabla de negociación.

6. Reserva del inmueble: La reserva de inmueble genera un pedido de venta dentro del OpenERP poder ser contabilizado con otros módulos, además que creará registros dentro de la tabla de producto la cual es base para el módulo de inventarios.

7. Reportes: la implementación también contempla reportes en todo su proceso, los reportes serán:

b) Prospecto: reporte que servirá para archivar el prospecto con todos sus datos, el lugar y el vendedor que lo atendió. Formato PDF.

c) Proforma: reporte en formato PDF que llevará el cliente como informativo de los valores que deberá pagar para la adquisición del inmueble.

d) Formulario de reserva: Formulario que deberá ser firmado por el cliente para hacer válida su reserva.

e) Contrato de reserva: el contrato que deberán firmar el cliente y la inmobiliaria.

f) Cancelación de reserva: en caso de existir una cancelación de reserva se deberá imprimir este formulario en el cual se detallan las causas y los valores a devolver con las respectivas penalidades.

El presente trabajo, contempla la implementación de dicho sistema en una inmobiliaria específica para verificar que funcione de la manera deseada

ya que se harán pruebas funcionales con datos reales, pero este sistema solo se lo implementara para dichos pruebas, en caso de que dicha inmobiliaria quisiera el sistema tendría que llegarse a un acuerdo para la comercialización de dicho sistema, cabe recalcar que el software a construirse va a ser genérico, y estará disponible para todas las inmobiliarias, ya que es libre y de código abierto, y se pueden hacer modificaciones al mismo, dependiendo de las necesidades de cada inmobiliaria.

Este sistema estará acoplado al módulo base de ventas de openERP de donde se derivan los módulos de inventarios, contabilidad, entre otros; por lo que se entiende que se podrá acoplar libremente a cualquier otro módulo del ERP, sin embargo el presente sistema no contempla el involucramiento del resto de módulos.

A continuación se detallan los módulos que se van a utilizar del OpenERP, y cuales no se utilizarán.

a. Módulos que se van a utilizar

- Ventas

b. Módulos que se no se van a utilizar, para el presente sistema.

- Contabilidad
- Inventarios

El objetivo es desarrollar un sistema que se adapte al modelo de negocio de la industria inmobiliaria ecuatoriana para la comercialización de bienes inmuebles con las funcionalidades que brinda OpenERP, aplicando este sistema se pretende automatizar y centralizar la comercialización de

inmuebles, generando de esta forma un aumento en la productividad y ayudando a mejorar su visión del negocio.

3.1.1.3 Personal involucrado

Nombre	Israel Paredes	Katherine Robles
Rol	Desarrollador	Desarrollador
Categoría profesional	Egresado	Egresada
Responsabilidades	Levantar los requisitos del sistema y desarrollar el módulo de comercialización para un bien inmueble. Levantar los requisitos del sistema y desarrollar el módulo de comercialización para un bien inmueble.	
Información de contacto	0984632744	0992712559

Tabla 3. Personal involucrado en el proyecto.

3.1.1.4 Definiciones, acrónimos y abreviaturas

3.1.1.4.1 Definiciones

Administradores: Usuarios del sistema, con todos los permisos para administrar el sistema.

Clientes: Son las personas que desean comprar un bien inmueble, dentro del sistema.

Inmueble: Son los inmuebles que se va a comercializar.

Producto: Es el producto final (módulo de comercialización)

Rol: Es el comportamiento determinado que se tiene dentro del sistema

Usuarios: Son los encargados de manipular el sistema, y son manejados en base a roles.

ERS: Es un documento, a manera de contrato entre el cliente y los desarrolladores al momento de desarrollar un software, para ayudar a la empresa a describir que es lo que realmente quiere, y que el desarrollador sepa exactamente qué debe hacer, para que al momento de la entrega final

del producto no existan dificultades, ya que se debería respaldar en los ERS para ver si se cumple o no con los requisitos del sistema.

SLA: Es un documento en el cual se detallan todos los acuerdos a los que lleguen el cliente y la inmobiliaria para la creación del módulo de Construcciones, pues en este mismo se basarán para hacer las pruebas funcionales del sistema, el cual les servirá de respaldo para ambas partes para que no existan ambigüedades.

ERP: Es un sistema que ayuda a la gestión empresarial de una empresa, optimizando sus recursos.

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema

3.1.1.4.2 Abreviaturas y Acrónimos

ERS: Especificación de Requisitos de Software

SLA: Acuerdos a Nivel de Servicio

ERP: Enterprise Resource Planning, ERP por sus siglas en inglés

UML: Lenguaje Unificado de Modelado

3.1.1.5 Referencias

Referencia	Titulo	Ruta	Fecha Última Actualización	Autor
Objetivos de los ERS	ERS	http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&sqj=2&ved=0CB4QFjAA&url=http%3A%2F%2Fsiml.googlecode.com%2Ffiles%2FFERS.pdf&ei=pz6lUJXBJYrm8QS1voGwDA&usg=AFQjCNFj2Xj8OMaKkue4Qcg5SHI5w4TYUg	2000	Chalmeta
Concepto UML	de Lenguaje Unificado Modelado	de http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado	24/10/2012	Wikipedia
Concepto Escalabilidad	de Escalabilidad	http://es.wikipedia.org/wiki/Escalabilidad	04/10/2012	Wikipedia
Concepto Open Source	de Código Abierto	http://es.wikipedia.org/wiki/C%C3%B3digo_abierto	25/10/2012	Wikipedia

Tabla 4. Referencias utilizadas en el proyecto.

3.1.1.6 Visión General del Documento

Basándose en este documento, el ingeniero de software procederá al modelado de la futura aplicación, ya teniendo una mejor visión de lo que realmente desea al cliente.

Además el presente documento tiene los principales objetivos:

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software: El cliente debe participar activamente en la especificación de requisitos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo. Asimismo, el cliente se siente partícipe del propio desarrollo.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente: En muchas ocasiones el cliente no sabe exactamente qué es lo que quiere. La ERS permite al cliente definir todos los requisitos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requisitos, los costes de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación.

Este documento consta de tres partes:

1. Visión General del Documento: Es la visión general del Sistema a desarrollar, en este caso del módulo de comercialización de un bien inmueble.
2. Descripción General: En esta parte, se describe el sistema, las principales funciones que se pretende que haga el sistema.

3. Requisitos Específicos: En esta parte, se detallan los requisitos tanto funcionales como no funcionales, que debe cumplir el sistema.

3.1.2 Descripción general

3.1.2.1 Perspectiva del producto

El producto (Módulo de comercialización) forma parte de un sistema mayor que es la Herramienta OpenERP que es un sistema planeador de recursos empresariales que permite realizar una gestión integrada de los recursos.

El producto final se integra a los módulos Sale y CRM propios de la plataforma los cuales involucran los procesos de creación de contactos, prospectos y clientes hasta realizar un pedido de venta.

3.1.2.2 Funcionalidad del producto

Para poder comercializar un bien inmueble, se debe registrar dicho inmueble con su respectivo cliente, para lo cual se procede hacer de la siguiente manera:

1. **Parámetros:** Se debe registrar un inmueble, en base a los siguientes parámetros:

- a) Tipos de Objeto
- b) Objeto
- c) Tipos de Inmueble
- d) Tipos de Ubicación
- e) Marcas

2. **Datos Generales:** Se debe llenar los datos generales, esta sección hace referencia a la provincia, parroquia donde se encuentra

el inmueble a comercializar, se debe crear dicha información como es el barrio o sector, cantón, ciudad, dirección del inmueble.

3. Proyección de Ventas: Aquí se debe llenar la información necesaria respecto a la proyección de ventas en función de:

- a) Número estimado de habitaciones
- b) Número estimado de locales comerciales
- c) Número estimado de estacionamientos
- d) Fecha estimada de aprobación de planos
- e) Fecha estimada de entrega Comercialización
- f) Fecha estimada inicio de construcción
- g) Fecha estimada de aprobación de planos
- h) Fecha estimada entrega/recepción comercialización
- i) Observación
- j) Otros

4. Preventa: Aquí se deben llenar los diferentes parámetros, es similar a lo hecho en Proyección de ventas, la diferencia aquí es que se debe escribir el número de registro.

5. Detalle de Tipo de Inmueble: En esta sección se va a detallar el tipo de inmueble, por ejemplo si sería una vivienda (casa) o un centro comercial o un departamento.

6. Ubicaciones por Proyecto: En esta sección se definen las distintas ubicaciones que puede poseer un proyecto. Cada ubicación puede contener a otra por ejemplo: El proyecto P1 posee 3 manzanas (M1, M2, M3, dentro de M1 existen los 2 Lotes (L1, L2) y en cada lote se construyeron 2 Torres (T1, T2).

7. Entrega General: En esta sección de deben llenar:

- a) La tipología (aquí se debe ingresar el número de pisos que tendrá la casa).
- b) Fecha de entrega del acta de entrega de recepción.
- c) Tipos de construcción.
- d) Observaciones.
- e) Se ingresará una nueva inspección (ingresar la fecha y su ubicación).
- f) Se debe ingresar un tipo de Inmueble, su ubicación el área y la cantidad aproximada.

8. Ingreso de Inmueble: En esta sección se debe ingresar manualmente los inmuebles uno por uno, con la siguiente información:

- a) Inmueble
- b) Número
- c) Área del terreno (dado en metros cuadrados)
- d) Área de construcción (dada en metros cuadrados)
- e) Alícuota
- f) Número de dormitorios
- g) Número de baños
- h) Predio

9. Importación de Inmuebles: Esta sección es de gran utilidad, ya que sirve para importar inmuebles que se tienen en un documento hecho por ejemplo un archivo de Excel.

10. Creación cliente: Para la creación del cliente, se va hacer uso del módulo propio del OpenERP (sale.order), en el cual se crea un contacto;

luego se lo evalúa en base a un porcentaje de éxito, esto significa el porcentaje de interés que tiene el contacto sobre un proyecto determinado: para luego convertirse en un prospecto; después pasa a convertirse en cliente, llenando la información necesaria como son los datos personales y algunos documentos requeridos como por ejemplo copia de la cédula.

11. Una ya vez ya creado el cliente, se procede añadir uno o varios inmuebles que ya se ingresó anteriormente, para luego proceder a reservar el mismo, y proceder con la comercialización de los mismos.

3.1.2.3 Características de los usuarios

El sistema manejará los siguientes usuarios:

3.1.2.3.1 Administrador

Es la persona encargada de realizar cualquier tipo de operación dentro del sistema, sin ningún tipo de restricción, con las siguientes características:

- Educación: Superior
- Nivel de Conocimiento: 3er nivel
- Conocimientos Técnicos: Informática, internet
- Formación: Ingeniero en Sistemas e Informática

3.1.2.3.2 Jefe de Diseño e Ingenierías

Es la persona encargada de manejar información respecto a los bienes inmuebles, ya sea en el ingreso de los mismos o también aprobando y revisando información, con las siguientes características:

- Educación: Superior
- Nivel de Conocimiento: 3er nivel
- Conocimientos Técnicos: Manejo de PC
- Formación: Ingeniero Civil

3.1.2.4 Vendedor

Es la persona encargada de comercializar el bien inmueble, con las siguientes características:

- Educación: Superior
- Nivel de Conocimiento: 3er nivel
- Conocimientos Técnicos: Ventas, marketing
- Formación: Ingeniero en Marketing

3.1.2.4 Restricciones

1. Para que el sistema funcione de la manera deseada, se recomienda que se trabaje sobre el sistema operativo Linux, aunque si funciona sobre Windows, pero en función de la ley del gobierno que obliga a utilizar software libre.

2. Las restricciones respecto al hardware, son:

- a) 1 GHz de procesador mínimo
- b) 2 GB de RAM.
- c) Disco Duro de 128 GB

3. Las restricciones respecto al lenguaje de programación:

- a) Únicamente se va a utilizar Python.

4. Se regirá únicamente a la metodología XP y para la SRS se utilizará el estándar IEEE 830.

5. Plataforma openobject.

6. No se debe pagar licencias, ya que para todo se va a utilizar tecnología Open Source.

3.1.2.5 Suposiciones y dependencias

3.1.2.5.1 Dependencias:

El módulo de comercialización podrá funcionar en navegadores web como:

- Internet Explorer 7.0
- Firefox Mozilla 3.0
- Google Chrome

3.1.3 Requisitos específicos

3.1.3.1 Requisitos comunes de las interfaces

En esta sección se va a detallar todas las entradas y salidas del sistema (módulo de comercialización).

3.1.3.1.1 Interfaces de usuario

- El sistema se va a manejar en la web, ya que es una forma más amigable para la interfaz del usuario, y es mucho más sencillo de manejar que una aplicación de escritorio.
- El producto aparecerá en la web, por medio de la configuración del puerto en donde quiero que aparezca el módulo, se manejara con menús y submenús porque es un sistema grande y para que de esta manera sea más sencillo su uso.

3.1.3.1.2 Interfaces de hardware

Windows:

- Windows XP
- Windows Vista
- Windows 7

Linux:

- Centos
- Ubuntu

3.1.3.1.3 Interfaces de software

- El sistema deberá funcionar sobre la plataforma de Ubuntu Server 10.04 o superior.
- Los clientes deben estar en la facultad de conectarse desde cualquier Sistema Operativo, usualmente Windows.
- El sistema se implementará sobre plataforma web de la siguiente manera:
 - **Módulo del cliente:** Abierto para todos los usuarios.
 - **Módulo de administración:** Será de acceso restringido, y únicamente tendrá acceso el administrador del sistema.

3.1.3.1.4 Interfaces de comunicación

- Se va a comunicar mediante el protocolo HTTP, por el puerto 8080.
- Adicionalmente si se requiere se puede comunicar por medio de servicios XML-RPC por medio del puerto 8069

3.1.4 Requisitos funcionales

Los requisitos funcionales, se los va a separar en cuatro secciones:

1. Administración Usuarios
2. Parámetros iniciales
3. Inicio del proceso de Comercialización de un bien inmueble

3.1.4.1 ADMINISTRACIÓN DE USUARIOS

<i>Número de requisito</i>	RF1		
<i>Nombre de requisito</i>	Administración de Usuarios		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El Sistema deberá permitir la correcta administración de perfiles de usuarios, esto se refiere hacer un “ crud ” a la base de datos que son la creación, lectura, eliminación y actualización de usuarios, esto se debe hacer con las seguridades pertinentes para que tengan acceso únicamente a ciertas opciones.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	<p>Los siguientes datos son:</p> <ul style="list-style-type: none"> - Nombre del Usuario - Iniciar sesión - Contraseña - Proyecto - Activo <p>Como datos también importantes dentro del sistema son:</p> <p>Contacto:</p> <ul style="list-style-type: none"> - Dirección - Unidad - Email <p>Acción:</p> <ul style="list-style-type: none"> - Inicio de acción - Menú action <p>Preferencias:</p> <ul style="list-style-type: none"> - Idioma - Zona Horaria - Interfaz <p>Firma</p> <p>Grupos</p>		
<i>Proceso</i>	<p>El sistema despliega el formulario para el registro de los usuarios en el cual se debe llenar toda la información pertinente de los datos que se tiene como entrada, ya que para poder guardar la información de un usuario se deben llenar los campos obligatorios ya que sí que no es así, el sistema lanzara un error indicando que se llenen dichos campos. (Para saber que un campo es obligatorio se lo podrá identificar al observar que el campo donde se debe llenar dicha información esta de color azul), y de esta forma se registra el usuario dentro del sistema.</p> <p>En el caso que se desee leer, eliminar o actualizar usuarios, el sistema lanzara el mismo formulario en el cual se llenaron los datos de entrada y el usuario debería llenar o corregir la información acorde a sus necesidades.</p>		
<i>Salida</i>	<p>En esta sección se va a tener diferentes salidas, ya que se están haciendo referencia al “crud”, estas serían:</p> <ul style="list-style-type: none"> • Lectura de usuarios existentes en el sistema. • Usuarios creados en el sistema con su respectiva información. • Usuarios eliminados. • Usuarios actualizados. 		

<i>Número de requisito</i>	RF2		
<i>Nombre de requisito</i>	Validación de Usuarios		
<i>Tipo</i>	<input type="checkbox"/> Requisito	<input checked="" type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El Sistema deberá permitir la validación de los usuarios que existen en el sistema.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Son los siguientes datos: <ul style="list-style-type: none"> • Login del usuario • Password del usuario 		
<i>Proceso</i>	El sistema despliega la pantalla principal, en la cual se debe ingresar los datos de entrada como son el login del usuario con su respectivo password, si dicho interesado existe podrá ingresar al sistema, caso contrario mostrará un error diciendo que dicho usuario no existe. De la misma forma se debe validar el rol que tiene el usuario dentro del sistema, y se debe mostrar una interfaz acorde a esos roles. El usuario principal para la creación de usuarios es el Administrador (este ya vendrá creado por defecto dentro del sistema), el cual tiene todos los permisos necesarios para hacer modificaciones al mismo.		
<i>Salida</i>	Correcta validación de dicho usuario y de esta manera podrá ingresar al sistema.		

3.1.4.2 PARÁMETROS INICIALES

<i>Número de requisito</i>	RF3		
<i>Nombre de requisito</i>	Administración de tipos de inmuebles.		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	Se requiere que los tipos de inmuebles sean administrables en el sistema.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos son: <ul style="list-style-type: none"> • Nombre • Descripción 		
<i>Proceso</i>	El sistema despliega el formulario para el registro de los tipos de inmuebles, en el cual se debe llenar los datos que se tiene como entrada.		
<i>Salida</i>	Una lista de los diferentes tipos de inmuebles que se crearon.		

<i>Número de requisito</i>	RF4		
<i>Nombre de requisito</i>	Administración de tipos de ubicaciones.		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	Se requiere que en el sistema se puedan crear las distintas ubicaciones que pueden existir por ejemplo Manzana, Bloque, Torre, Lote, etc.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Nombre • Descripción 		
<i>Proceso</i>	El sistema despliega el formulario para la creación de los tipos de ubicación, en el cual se debe llenar con los datos de entrada, en el campo nombre se debe colocar el nombre de la ubicación (por ejemplo Mz) y en el campo descripción se debe colocar una descripción específica de esa ubicación (por ejemplo para el nombre anterior sería Manzana).		
<i>Salida</i>	Una lista de los diferentes tipos de ubicación que va a tener cada inmueble. (Esto se visualizara en una vista de tipo árbol).		

3.1.4.3 INICIO DEL PROCESO DE COMERCIALIZACIÓN DE UN BIEN

INMUEBLE

<i>Número de requisito</i>	RF5		
<i>Nombre de requisito</i>	Administración de proyectos		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El Sistema deberá permitir la administración de proyectos de construcción.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Todos los datos del proyecto: <ul style="list-style-type: none"> • Código • Fecha de inicio del proyecto • Fecha de finalización del proyecto • Provincia • Cantón • Parroquia • Ciudad • Barrio o Sector • Dirección 		
<i>Proceso</i>	El sistema despliega el formulario para el registro del proyecto, en este se debe llenar toda la información pertinente con los datos de entrada, se validan en el caso de ser necesarios, y se crea dicho proyecto.		
<i>Salida</i>	Proyecto ingresado con su respectiva información.		

<i>Número de requisito</i>	RF6		
<i>Nombre de requisito</i>	Registros de tipos de inmueble por proyecto		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	Se requiere que se pueda registrar en el sistema todos los tipos de inmueble que fueron construidos en el proyecto.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Tipo de inmueble • Observaciones generales 		
<i>Proceso</i>	El sistema despliega el formulario para el registro de los diferentes tipos de inmueble por un proyecto específico, en esta sección se deben llenar los datos de entrada, en el caso del tipo de inmueble va a ser un combo box en el cual se va a escoger el inmueble creado anteriormente, y en observaciones generales se debe escribir las observaciones que se tengan respecto a dicho inmueble.		
<i>Salida</i>	Una lista de los diferentes tipos de inmuebles asignados a un proyecto específico. (Esto se visualizara en una vista de tipo árbol)		

<i>Número de requisito</i>	RF7		
<i>Nombre de requisito</i>	Ubicaciones por proyecto		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	Se debe permitir armar las ubicaciones dentro de un proyecto, una ubicación puede pertenecer a otra, por ejemplo que la Torre A se encuentre en el Bloque 1 de la Manzana Z, siendo su ubicación exacta MzZ-BI1-TA.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Ubicación a la que pertenece (Proyecto) • Nombre de la ubicación (Proyecto) • Tipo de ubicación • Nombre • Ubicación en el proyecto • Predio 		
<i>Proceso</i>	El sistema despliega el formulario para llenar las ubicaciones por proyecto, en el cual se deben llenar los datos que se tiene como entrada.		
<i>Salida</i>	Una lista de las diferentes ubicaciones por proyecto. (Esto se visualizara en una vista de tipo árbol)		

<i>Número de requisito</i>	RF8		
<i>Nombre de requisito</i>	Ingreso de tipos de inmuebles por ubicación.		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	Dentro de las ubicaciones se ingresan n inmuebles de un tipo de inmueble específico, se requiere que esto sea especificado en cada proyecto.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Tipo de inmueble • Ubicación en el proyecto • Área aproximada del terreno • Cantidad de inmuebles • Observación 		
<i>Proceso</i>	El sistema despliega el formulario para ingresar los datos que se tiene como entrada.		
<i>Salida</i>	Una lista de los inmuebles ingresados por cada ubicación registrada anteriormente. (Esto se visualizara en una vista de tipo árbol)		

<i>Número de requisito</i>	RF9		
<i>Nombre de requisito</i>	Ingreso de inmuebles individual.		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	Se deben registrar los inmuebles individualmente.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Estos datos: <ul style="list-style-type: none"> • Tipo de inmueble/ubicación • Número • Área de terreno • Área de construcción • Valor de la alícuota • Número de dormitorios • Número de Baños. • Predio 		
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe ingresar los datos que se tiene como entrada.		
<i>Salida</i>	Una lista de los diferentes inmuebles ingresados de forma individual, cada uno con su respectiva información.		

<i>Número de requisito</i>	RF10		
<i>Nombre de requisito</i>	Importación de inmuebles.		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir la importación en masa de los inmuebles a través de un archivo de Excel ya que por cada proyecto existirá un gran número de inmuebles.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los datos de los diferentes inmuebles a importar en los archivos de Excel en extensión .csv.		
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe elegir el proyecto al cual se hace referencia, el inmueble y a la vez se debe escoger el archivo en Excel, del cual se desea importar los datos.		
<i>Salida</i>	Datos importados en el sistema del archivo en formato Excel, los mismos que son leídos línea por línea, para luego ser almacenados en la base de datos.		

<i>Número de requisito</i>	RF11		
<i>Nombre de requisito</i>	Administración de provincias, cantones, ciudades y parroquias.		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El Sistema deberá permitir la administración de provincias, cantones, parroquias y ciudades; sabiendo que las parroquias pertenecen a las ciudades, las ciudades a los cantones y éstos a su vez pertenecen a las provincias.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Ingreso de ciudades con su respectiva jerarquía.		
<i>Proceso</i>	El sistema despliega el formulario, en el cual se deben ingresar las respectivas ciudades respetando su jerarquía.		
<i>Salida</i>	Ciudades ingresadas en el sistema, para su respectiva utilización.		

<i>Número de requisito</i>	RF12		
<i>Nombre de requisito</i>	Precios por inmuebles		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir la administración de los precios por inmuebles.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos:		

Continua...

	<ul style="list-style-type: none"> • Nombre • Precio por M2
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe ingresar los datos que se tiene como entrada.
<i>Salida</i>	Una lista de los diferentes precios por inmuebles ingresados en el sistema, cada uno con su respectiva información.

<i>Número de requisito</i>	RF13		
<i>Nombre de requisito</i>	Promociones de Proyectos		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir el ingreso de las distintas promociones de proyectos.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Nombre • Porcentaje de Descuento (%) • Fecha Inicio • Fecha Finalización • Estado 		
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe ingresar los datos que se tiene como entrada.		
<i>Salida</i>	Una lista de las diferentes promociones por proyectos ingresados en el sistema, cada uno con su respectiva información.		

<i>Número de requisito</i>	RF14		
<i>Nombre de requisito</i>	Políticas de Venta		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir el ingreso de las distintas políticas de venta.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Nombre • Porcentaje Total • Valor • Plazo (Días) 		
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe ingresar los datos que se tiene como entrada.		
<i>Salida</i>	Una lista de las diferentes políticas de venta ingresados en el sistema, cada uno con su respectiva información.		

<i>Número de requisito</i>	RF15		
<i>Nombre de requisito</i>	Iniciativas		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir el registro de prospectos con sus datos básicos los cuales son registrados por vendedores.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Nombres y apellidos del prospecto. • Cédula de identidad • Lugar de ubicación. 		
<i>Proceso</i>	Validación de cédula de identidad. Creación del registro en la base de datos.		
<i>Salida</i>	Una lista de los prospectos ingresados. (Esto se visualizara en una vista de tipo árbol).		

<i>Número de requisito</i>	RF16		
<i>Nombre de requisito</i>	Oportunidades		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir el paso de un prospecto a una oportunidad, para poder comercializar un bien inmueble.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Proyecto Principal • Información <ul style="list-style-type: none"> i) Cédula ii) Estudios iii) Estado Civil iv) Celular v) No. Hijos • Laboral <ul style="list-style-type: none"> i) Lugar de Trabajo ii) Teléfono iii) Dirección del Trabajo • Financiera <ul style="list-style-type: none"> i) Posee ahorros ii) Institución Financiera iii) Valor Precalificación iv) Ingreso Familiares Promedio v) Otros Ingresos • Datos del Cónyuge <ul style="list-style-type: none"> i) Nombre ii) No. Cédula 		

Continua...

	<ul style="list-style-type: none"> iii) Lugar de Trabajo iv) Teléfono v) Celular vi) Ingresos vii) Estudios <ul style="list-style-type: none"> • Inmuebles <ul style="list-style-type: none"> ○ Estos se cargan automáticamente, al momento de escoger el proyecto principal.
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe ingresar los datos que se tiene como entrada. Creación del registro ya como una oportunidad en la base de datos.
<i>Salida</i>	Una lista de las oportunidades ingresadas. (Esto se visualizara en una vista de tipo árbol).

<i>Número de requisito</i>	RF17		
<i>Nombre de requisito</i>	Pedidos de Venta		
<i>Tipo</i>	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
<i>Fuente del requisito</i>	El sistema debe permitir el ingreso del pedido de venta para la respectiva comercialización de un determinado bien inmueble.		
<i>Prioridad del requisito</i>	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<i>Entrada</i>	Los siguientes datos: <ul style="list-style-type: none"> • Cliente • Contacto que realiza el pedido • Dirección factura • Dirección de envío • Tarifa • Cuenta Analítica • Proyecto • Líneas de pedido <ul style="list-style-type: none"> ○ Estas se cargan automáticamente, porque ya vienen cargadas desde oportunidades. 		
<i>Proceso</i>	El sistema despliega el formulario en el cual se debe ingresar los datos que se tiene como entrada. Se valida el pedido de venta.		
<i>Salida</i>	Una lista de los pedidos de venta ingresados. (Esto se visualizara en una vista de tipo árbol).		

3.1.5 VALIDACIONES

3.1.5.1 RF18. Los usuarios tendrán acceso solo al módulo cliente, no tendrán acceso al módulo de administración del sistema.

3.1.5.2 RF19. El sistema deberá permitir ingresar la información necesaria para poder comercializar un bien inmueble, como es caso de los clientes, usuarios, inmuebles.

3.1.5.3 RF20. Los usuarios con el rol de administrador, podrán realizar la creación, eliminación, modificación de:

- Clientes
- Usuarios
- Inmuebles
- Proyectos

3.1.5.4 RF21. El sistema deberá permitir hacer búsqueda de clientes, usuarios, inmuebles.

3.1.6 Requisitos no funcionales

3.1.6.1 *Requisitos de rendimiento*

- Número de terminales
 - El sistema estará soportado dentro de un solo servidor.
- Número de usuarios conectados simultáneamente
 - El sistema trabaja con el servidor Cherry de Python, el mismo que permite que 150 usuarios se conecten simultáneamente.
- Número de transacciones por segundo que deberá soportar el sistema
 - El sistema podrá soportar el ingreso de 150 usuarios de forma simultánea.
- Volúmenes de información

- No son muy extensos, sin embargo el sistema debería ser escalable.
- Tiempo de respuesta
 - Debería ser no menos de 3 segundos al momento de realizar la petición (Esto puede variar dependiendo de la velocidad de la intranet.)

3.1.6.2 Seguridad

- Únicamente podrán ingresar al sistema, los usuarios que han sido creados, caso contrario el sistema lanzará un mensaje “que dicho usuario no existe”.
- El usuario administrador es el único autorizado para administrar el sistema, de esta manera se está protegiendo la integridad de los datos.
- Además respecto al código fuente del sistema, el cliente no tendrá acceso al mismo, las únicas personas autorizadas son los desarrolladores que son el Sr. Israel Paredes y la Srta. Katherine Robles.
- Como se va a contar con un menú para cada tarea, se creará roles a los usuarios para que únicamente tengan acceso a su respectiva información, de esta manera se protege la información que ingresa al sistema, y como se va a contar con un historial, se sabrá que usuario hizo cambios y de esta forma cada uno es responsable de los datos que ingresan y salen del sistema.

3.1.6.3 Fiabilidad

El sistema de comercialización de bienes inmuebles, respecto a la fiabilidad, se define en función a los siguientes factores:

- Tolerancia a fallos
- Al momento de ingresar datos, no se debe guardar basura ya que utilizaría demasiados recursos, que son necesarios para otros puntos importantes.
- El acceso al sistema debe ser 100% confiable.

3.1.6.4 Disponibilidad

El sistema estará funcionando las 24 horas, los 7 días de la semana, los 365 días del año (24x7x365). A no ser que se encuentre en mantenimiento o en corrección de errores, lo cual será previamente notificado a los usuarios.

3.1.6.5 Mantenibilidad

- Las tareas de mantenimiento de usuarios estarán a cargo del Administrador del Sistema (por ejemplo El Jefe de Sistemas de la Empresa).
- Las tareas de mantenimiento del software, estarán a cargo de los desarrolladores del sistema en este caso son, la Srta. Katherine Robles y el Sr. Israel Paredes.
- Las tareas de mantenimiento únicamente se harán cuando sean necesarias, por ejemplo cuando el sistema no esté disponible y este arrojando errores de programación, o cuando los roles no se hayan asignado de la forma correcta.

3.1.6.6 Portabilidad

El sistema es portable para cualquier plataforma, porque es tecnología libre y estará abierto a todas las plataformas.

3.2 Historias de Usuario

3.2.1 Administración de usuarios

No.	HU-001				
Proceso	Gerencia de Proyecto:	Módulo Base			
Cliente:	Empresa Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT – Analista/ Programador
Israel Paredes	ATIKASOFT – Analista/ Programador

Objetivos:

- Analizar el manejo de la administración de usuarios dentro del sistema

Trabajo Realizado:

La administración de usuarios es una de las etapas más importantes dentro del sistema porque se podrá crear, leer, eliminar y modificar dichos usuarios, ya que se asignarán roles de acuerdo al cargo que tengan y a la responsabilidad que tendrán para el manejo del módulo de comercialización de bienes inmuebles.

Cada usuario posee datos significativos dentro del sistema, que son:

- **Nombre del Usuario.** Es el nombre con el cual se va a identificar a un usuario específico dentro del sistema.
- **Iniciar sesión.** Es el login con el cual un usuario va a poder ingresar al sistema, (por ejemplo si el nombre del usuario es Katherine Robles para iniciar sesión sería krobles).
- **Contraseña.** Es la clave para poder entrar al sistema caso contrario debería saltar error.
- **Proyecto.** Se debe elegir un proyecto ya creado en el caso de que sea necesario caso contrario se dejara vacío.
- **Activo.** Es un check que validara si ese usuario está activo o no.
- **Contacto:**
 - Dirección. Es la dirección del usuario.
 - Unidad. Es la unidad a la cual pertenece dicho usuario, (por

Continúa...

ejemplo si ese usuario pertenece a la Unidad de Escrituración, en esta opción debería escoger dicha unidad)

Email._ Es la dirección de correo electrónico del usuario, esta opción es opcional.

- **Acción:**

Inicio de acción

Menú action

- **Preferencias:**

Idioma._ En esta sección se debe escoger el idioma con el cual el usuario, va a trabajar dentro del sistema.

Zona Horaria._ Se debe escoger la zona horario para dicho usuario.

Interfaz._ Esta opción es como el usuario visualizará el sistema. Existen dos opciones que son:

Extendido/ Simplificado: Al marcar esta casilla, el usuario dependiendo de los grupos a los que pertenece podrá además visualizar campos y funcionalidades extendidas en todos los módulos.

- **Firma:** Que es la firma electrónica del usuario

- **Grupos,** en esta sección se debe agregar los grupos a los cuales va a pertenecer dicho usuario.

La administración de usuarios, la podrá hacer únicamente el Jefe de Sistemas de la empresa.

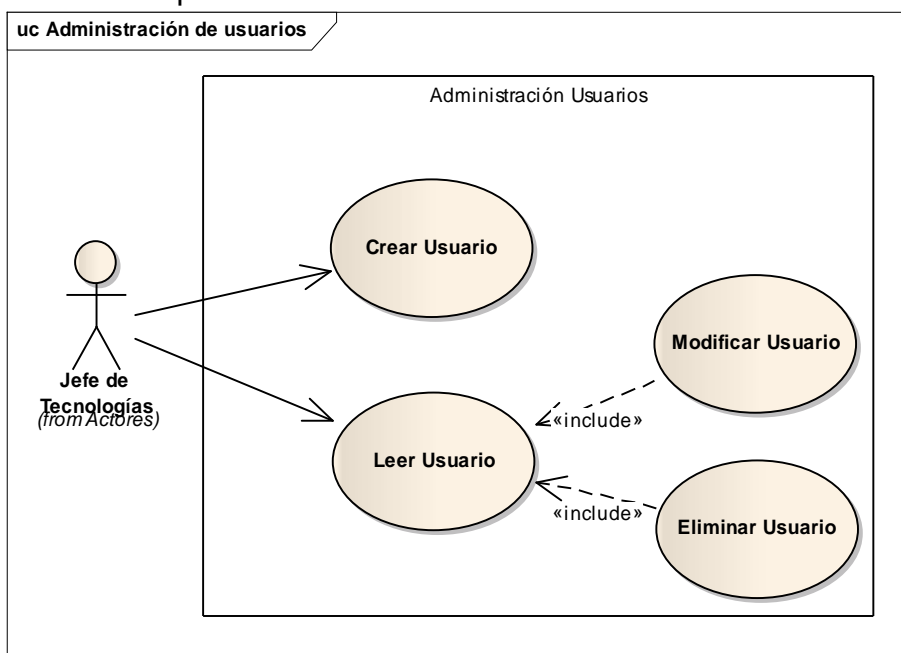


Figura 6: Administración de Usuarios

3.2.2 Validación de usuarios

No.	HU-002
Proceso Gerencia de Proyecto:	Módulo Base
Cliente:	Empresa Provincial de Viviendo E.P.
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.
Lugar:	Edificio ISPADE, Cuarto Piso

Continua...

Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00
---------------	------------	---------------------	-------	------------------	-------

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT – Analista / Programador
Israel Paredes	ATIKASOFT – Analista / Programador

Objetivos:

- Analizar cómo se va a realizar la validación de usuarios dentro del sistema.

Trabajo Realizado:

La validación de usuarios es una parte esencial dentro del sistema, ya que este debería validar el ingreso de usuarios.

Se tiene como datos:

- **Login.** Es el nombre del usuario dentro del sistema (por ejemplo para el Usuario Katherine Robles, sería krobles)
- **Password.** Es la contraseña de dicho usuario para ingresar al sistema.

Si el usuario que intenta ingresar al sistema no existe, el sistema debe indicar que dicho usuario y contraseña son incorrectos.

Todos los usuarios registrados en el sistema independientemente de su grupo podrán ingresar, el grupo únicamente define las reglas de acceso a los menús, objetos y funcionalidades del sistema.

Se requiere que en la base de datos la contraseña sea guardada con encriptación MD5 para garantizar la seguridad de los usuarios.

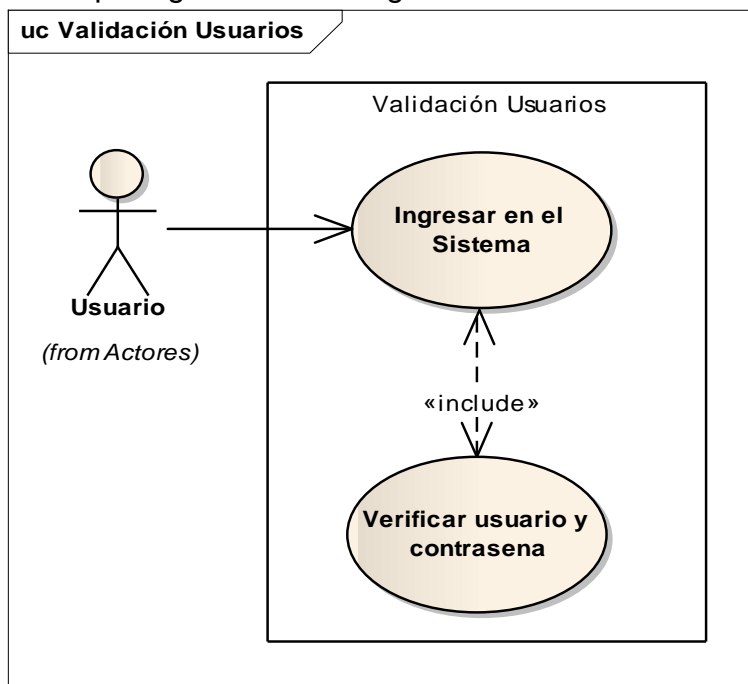


Figura 7: Validación Usuarios

3.2.3 Administración de tipos de inmuebles

No.	003				
Proceso Gerencia de Proyecto:	Módulo Construcciones				
Cliente:	Empresa Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

- Analizar el proceso de administración de Tipos de Inmuebles para su gestión en el Sistema.

Trabajo Realizado:

Todos los proyectos de construcción, previa la aceptación de su ejecución se someten a un análisis de factibilidad para obtener una información de posibles costos y la mejor opción para la construcción del mismo dependiendo del sector en el cual estará ubicado, por ejemplo casas o departamentos.

Una vez definido claramente cuál es la mejor opción para construir sobre un proyecto, se deben definir los Tipos de Inmuebles que se construirán en el proyecto de una lista de Tipos de inmuebles que COVIPROV como empresa oferta.

Por este motivo el Sistema automatizado debe poseer una administración de Tipos de Inmuebles, en donde se registran todos los tipos de inmueble que la empresa oferta y construye en sus proyectos.

Los Tipos de inmuebles poseen los siguientes datos de entrada.

- Nombre: define el nombre del tipo de inmueble, este campo será ingresado por el usuario y será el identificador de cada registro en el sistema.
- Descripción: Este debe ser un campo de texto largo para colocar una descripción globalizada del tipo de inmueble.

A continuación se presenta un ejemplo de cómo debe permitir el sistema registrar los tipos de inmuebles.

Nombre	Descripción
Casa Tipo 1	PLANTA BAJA Conformada por: sala, comedor, baño social; anaqueles altos y bajos de cocina; además cuenta con un área de jardinería y acceso peatonal y vehicular en la parte frontal, acceso lateral y patio posterior (lavandería). PLANTA ALTA

Continua...

Baño general, pequeña salita de estar, dormitorio principal con vestidor y baño, dormitorio 1 y dormitorio 2 con sus respectivos closets.

Cualquier usuario del área de construcciones podrá ingresar un nuevo tipo de inmueble.

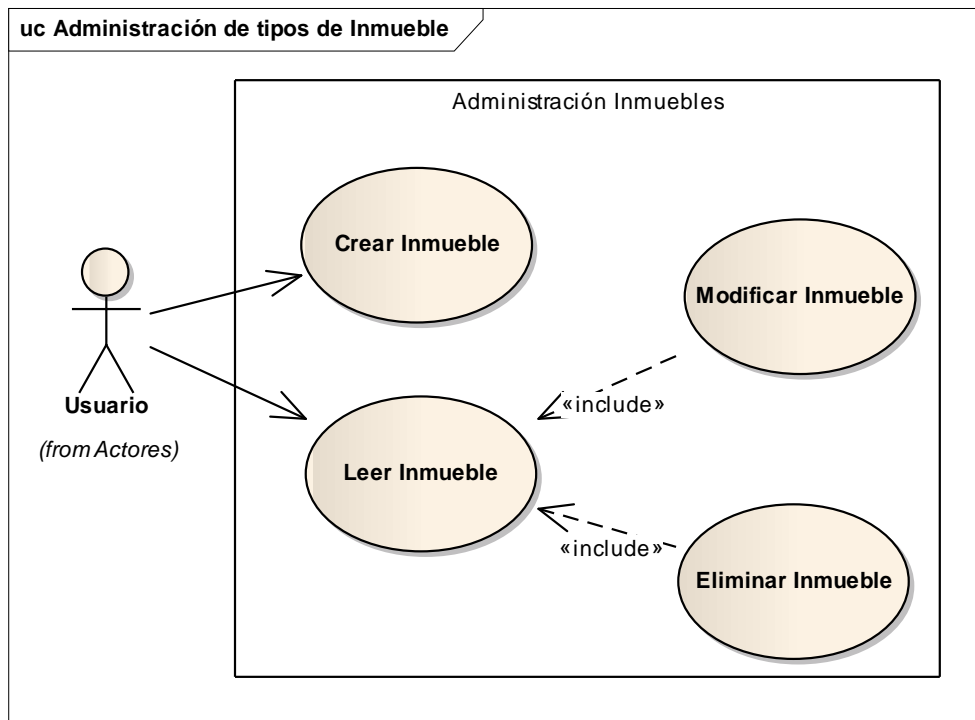


Figura 8: Administración de Tipos de Inmuebles

3.2.4 Administración de tipos de ubicaciones

No.	HU-004		
Proceso Gerencia de Proyecto:	Módulo Base		
Cliente:	Empresa Provincial de Viviendo E.P.		
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.		
Lugar:	Edificio ISPADE, Cuarto Piso		
Fecha:	07/01/2013	Hora inicio:	09:00
		Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	Jefe de Tecnologías - Covipro
Katherine Robles	Analista - Atikasoft
Israel Paredes Reyes	Analista - Atikasoft

Objetivos:

Determinar la manera de administrar los tipos de Ubicación.

Trabajo Realizado:

Cuando los proyectos de construcción que ya han pasado la etapa de

Continua...

factibilidad cuyo análisis a determinado cual es la mejor opción para construir, se define también la arquitectura general del proyecto, en donde se determinan las ubicaciones que tendrá el proyecto, entendiéndose por ubicación a las Manzanas, Bloques, Torres, entre otras que pueden existir.

Para ello, es necesario que en el sistema administre los tipos de ubicación que se manejan en la empresa.

Los datos de entrada requeridos por cada registro son:

- **Nombre:** es el nombre de la ubicación, este campo será escrito por el usuario e identificará el registro.
- **Abreviatura:** Es un campo de máximo 5 caracteres de longitud. Representa la abreviatura de la ubicación.

Solamente los usuarios de construcciones pueden administrar los tipos de ubicaciones.

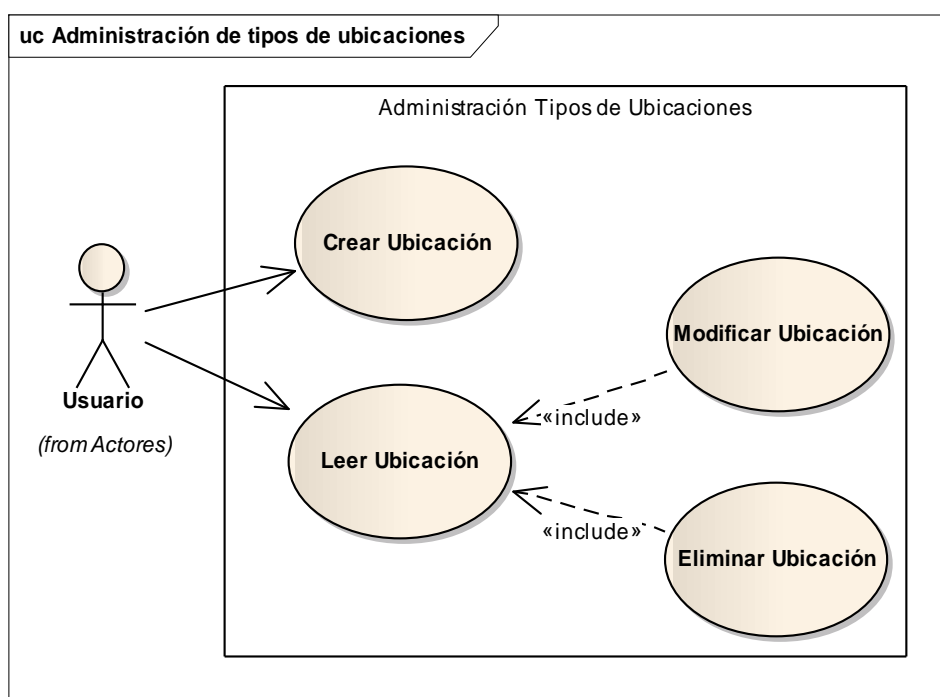


Figura 9: Administración de Tipos de Ubicaciones

3.2.5 Administración de proyectos

No.	HU-005				
Proceso Gerencia de Proyecto:	Módulo Construcciones				
Cliente:	Empres Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	Jefe de Tecnologías - Covipro

Continua...

Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

- Analizar y diseñar un modelo conceptual para la creación y administración de proyectos de construcción.

Trabajo Realizado:

Los proyectos de construcción son la base de todo el flujo de negocio que maneja la empresa. Sobre ellos se maneja la distribución de las calles e inmuebles que va a poseer.

Cada proyecto de construcción posee datos relevantes para todas las áreas de la empresa:

- Código: Identifica a cada proyecto, el sistema debe estar en la capacidad de asignar un código a cada proyecto
- Fecha de inicio: Indica la fecha en la cual el proyecto fue aprobado y los trabajos de construcción dieron comienzo.
- Fecha de finalización: Indica la fecha en la cual el proyecto fue totalmente construido acorde a lo planificado.
- Provincia: Indica la provincia en la cual fue realizado el proyecto.
- Cantón: Indica el cantón en el cual fue realizado el proyecto. Debe existir una validación para que solamente se puedan escoger cantones de la provincia antes ingresada.
- Ciudad: Indica la ciudad en la cual fue realizado el proyecto. Debe existir una validación para que solamente se pueda escoger una ciudad del cantón antes ingresado.
- Barrio o Sector: Indica el sector en el cual fue realizado el proyecto.
- Dirección: Indica la dirección exacta del proyecto.

El código que genere el sistema para cada proyecto debe ser único. Este debe estar compuesto por:

(Código de provincia)(Código del cantón)(Secuencia de 3 dígitos). Por ejemplo: PCHQUI001 siendo PCH el código de la provincia de Pichincha, QUI el código del cantón Quito y 001 el secuencial llenado por 3 dígitos.

La administración de las provincias solamente la puede realizar la Jefatura de Diseños e Ingenierías de la empresa, sin embargo los datos ingresados deben ser visibles para cualquier área que lo requiera.

Continua...

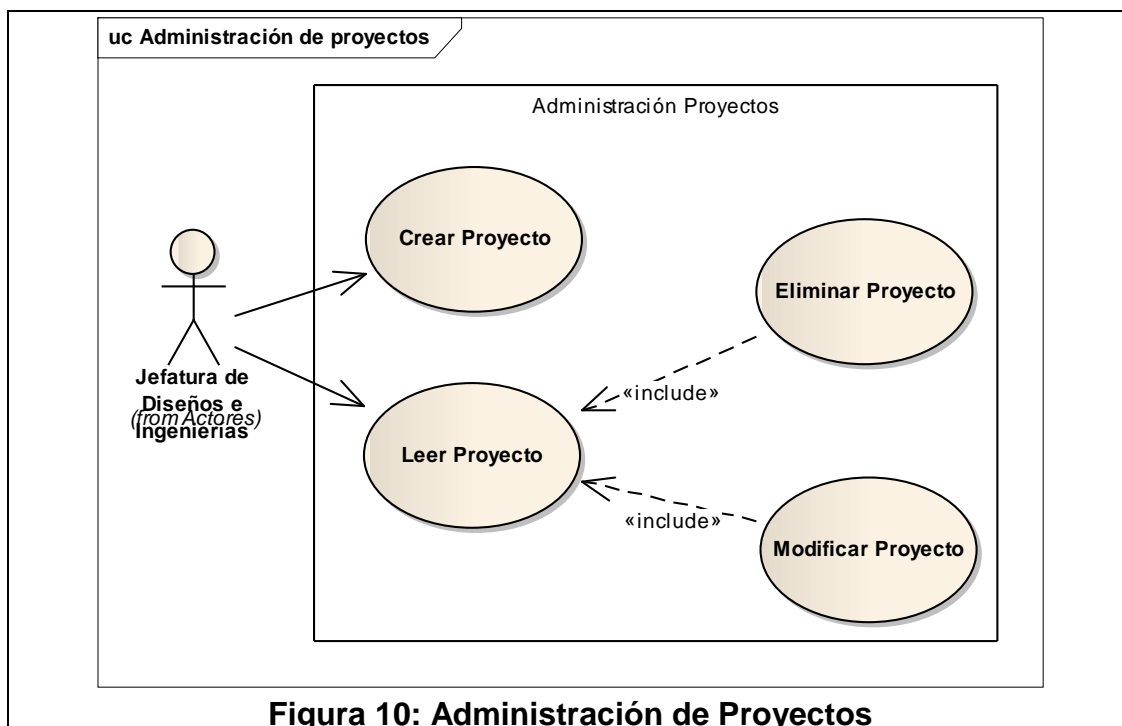


Figura 10: Administración de Proyectos

3.2.6 Registros de tipos de inmuebles por proyecto.

No.	HU-006				
Proceso Gerencia de Proyecto:	Módulo Construcciones				
Cliente:	Empres Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	Jefe de Tecnologías - Covipro
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

- Analizar y diseñar administración de los tipos de inmuebles ubicados por cada proyecto de construcción.

Trabajo Realizado:

Los proyectos de construcción son la base de todo el flujo de negocio que maneja la empresa. Sobre ellos se maneja la distribución de las calles e inmuebles que va a poseer.

Cada proyecto de construcción posee datos relevantes para todas las áreas de la empresa, sobre los cuales se debe ingresar los diferentes tipos de inmuebles que va a tener cada proyecto, los datos de entrada son:

Continua...

- Tipo de Inmueble: Es el tipo de inmueble que va a tener dicho proyecto.
- Observaciones Generales: Son las principales observaciones para el registro de los tipos de inmuebles por proyecto.

El registro de tipos de inmueble por proyectos, solamente la puede realizar la Jefatura de Diseños e Ingenierías de la empresa, sin embargo los datos ingresados deben ser visibles para cualquier área que lo requiera.

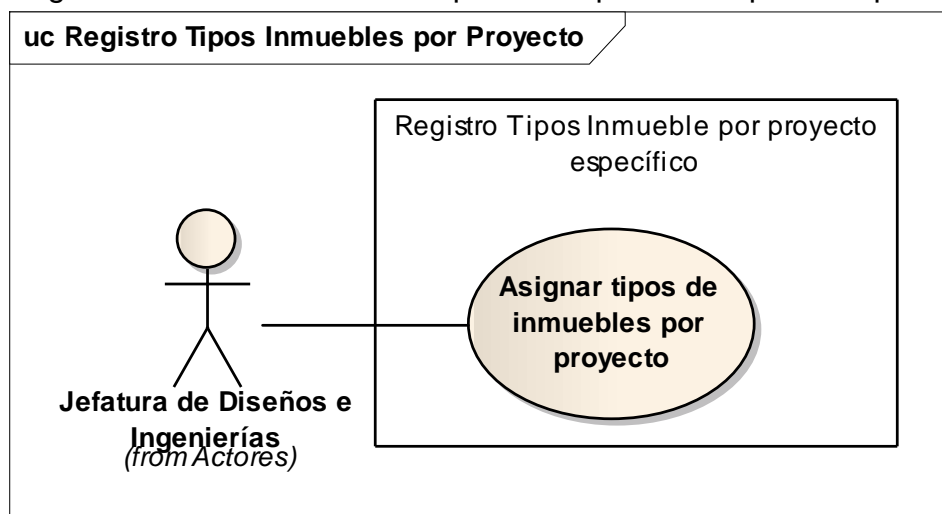


Figura 11: Registro Tipos de Inmuebles por Proyecto

3.2.7 Ubicaciones por proyecto

No.	HU-007				
Proceso Gerencia de Proyecto:	Módulo Construcciones				
Cliente:	Empres Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Victor Hugo Aguilar	Jefe de Tecnologías - Covipro
Katherine Robles Arévalo	Analista - Atikasoft
Israel Paredes Reyes	Analista - Atikasoft

Objetivos:

Analizar cómo el usuario registra los tipos de ubicación en un proyecto específico para la implementación dentro del sistema.

Trabajo Realizado:

Una vez aprobado por la jefatura de construcciones y ya hecho el análisis de factibilidad del proyecto, se debe esquematizar las ubicaciones que existen dentro de un proyecto acorde al análisis realizado.

Para ello en el sistema se necesita definir las ubicaciones por proyecto de construcción. Cada ubicación posee una nomenclatura específica y a su

Continua...

vez puede pertenecer a otra ubicación. Por lo que en el sistema se requerirá los siguientes datos de entrada:

- **Ubicación:** determina si la ubicación que se está registrando pertenece a otra, el usuario deberá seleccionar una de las ya registradas previamente.
- **Identificador:** es la sigla que determina la ubicación, el usuario debe registrar este campo cuyo valor debe ser máximo de 5 caracteres.
- **Tipo de ubicación:** este campo debe ser seleccionado por el usuario. La selección debe ser un listado de los tipos de ubicación previamente configurados.
- **Nomenclatura:** la nomenclatura es un campo que no podrá ingresar el usuario, este debe ser obtenido por el sistema. Se compone de la abreviatura del “Tipo de Ubicación” seguida por el “identificador” registrado por el usuario.
- **Ubicación general:** indica la ubicación general de un registro en el sistema. Este campo debe ser obtenido por el sistema y debe estar compuesto por la nomenclatura de las ubicaciones padre de cada registro.

A continuación se muestra una tabla como ejemplo de lo que se necesita registrar en el sistema:

Ubicación	Identificador	Tipo de ubicación	Nomenclatura	Ubicación general
	A	Mz	MzA	MzA
MzA	B	T	TB	MzA - TB
MzA - T1	10	Ps	Ps10	MzA - TB - Ps10
MzA - T1	20	Ps	Ps20	MzA - TB - Ps20

Según la tabla mostrada se tiene que para nuestro proyecto de ejemplo se tiene las siguientes ubicaciones:

- Una Manzana de identificador A.
- Una Torre de identificador B que se ubica en la Manzana A.
- Un Piso de identificador 10 que se ubica en la Torre B de la Manzana A.
- Un Piso de identificador 20 que se ubica en la Torre B de la Manzana A.

Solamente el usuario de construcciones podrá ingresar estos registros a cada proyecto.

Continua...

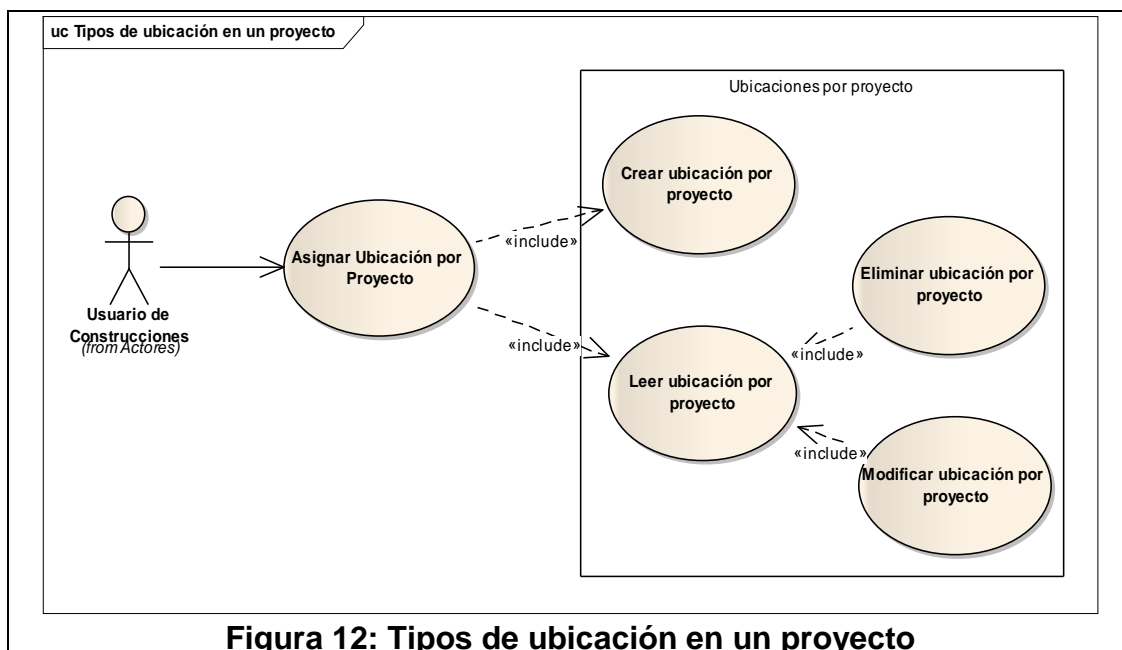


Figura 12: Tipos de ubicación en un proyecto

3.2.8 Ingreso de tipos de inmuebles por ubicación.

No.	HU-008				
Proceso Gerencia de Proyecto:	Módulo Construcciones				
Cliente:	Empres Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Victor Hugo Aguilar	Jefe de Tecnologías - Covipro
Katherine Robles Arévalo	Analista - Atikasoft
Israel Paredes Reyes	Analista - Atikasoft

Objetivos:

Conocer y analizar cómo los usuarios enlazan los tipos de inmuebles a una ubicación en un determinado proyecto para su implementación en el sistema.

Trabajo Realizado:

Cuando ya se encuentran definidos en el sistema los tipos de inmuebles al igual que las diferentes ubicaciones que posee un determinado proyecto, se empieza a unir cada ubicación con un tipo determinado de inmueble.

Cada ubicación final posee un determinado tipo de inmueble, independientemente de cuantos inmuebles de ese tipo sean construidos; por ejemplo, el proyecto "X" se conforma por 3 manzanas ("A", "B", "C"). En la manzana "A" se construirán únicamente "Casas Tipo 1"; en la manzana "B", "Casas Tipo 2"; en la manzana "C" nuevamente "Casas Tipo 1". En la manzana "A" por su superficie se construye cuatro casas, en la "B" seis

Continua...

casas y en la “C” ocho casas.

El sistema debe permitir registrar esta distribución general de los tipos de casas registrados en un proyecto a las diferentes ubicaciones previamente registradas para un proyecto.

Los datos que se necesitan registrar en esta parte del flujo de construcciones son los siguientes:

- **Ubicación del inmueble:** son las ubicaciones antes registradas para un proyecto específico.
- **Tipo de inmueble:** son los tipos de inmueble que posee un determinado proyecto.
- **Número de inmuebles:** es la cantidad de inmuebles de este tipo que poseerá una ubicación.
- **Área aproximada:** es el tamaño aproximado de la ubicación.
- **Observación:** un campo de texto para registrar cualquier observación que sucite.

El usuario encargado de hacer esta distribución será siempre el usuario que se encuentra a cargo del proyecto.

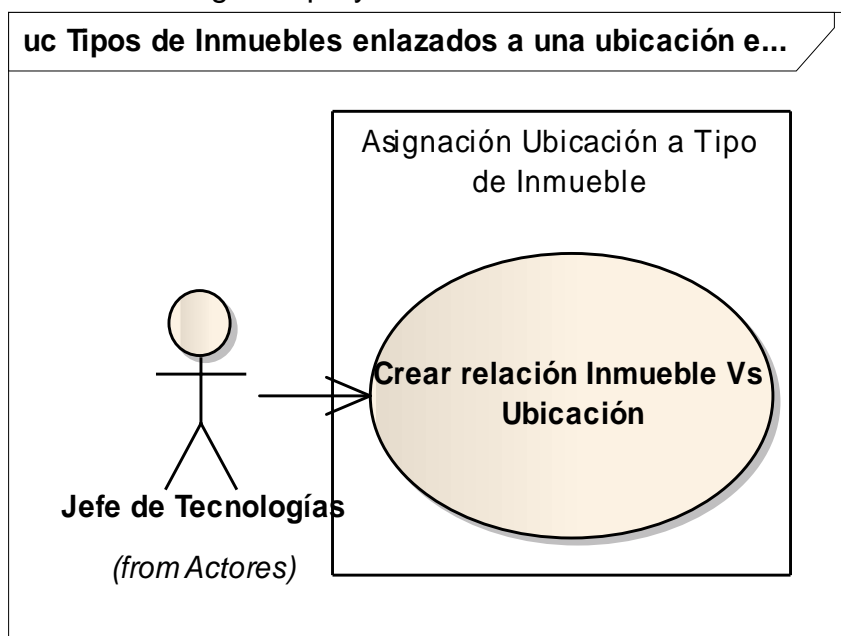


Figura 13: Tipos Inmuebles enlazados a una ubicación

3.2.9 Ingreso de los inmuebles de forma individual.

No.	HU-009			
Proceso Gerencia de Proyecto:	Módulo Construcciones			
Cliente:	Empres Provincial de Viviendo E.P.			
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.			
Lugar:	Edificio ISPADE, Cuarto Piso			
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin: 11:00

Participantes:

Nombre	Organización
Victor Hugo Aguilar	Jefe de Tecnologías - Covipro

Continua...

Katherine Robles Arévalo	Analista - Atikasoft
Israel Paredes Reyes	Analista - Atikasoft

Objetivos:

Conocer y analizar cómo los usuarios registran los inmuebles que posee un determinado proyecto para su implementación en el sistema.

Trabajo Realizado:**REGISTRO INDIVIDUAL**

El registro individual de inmuebles corresponde a las unidades habitacionales que se construyen en un proyecto. Representa exactamente al inmueble dentro del proyecto.

Previamente se ingresaron los tipos de inmuebles en una ubicación de manera general y la cantidad de inmuebles que poseerán, en esta ocasión se registran los inmuebles. A continuación un ejemplo más detallado en base al ejemplo del registro general.

Ubicación	Tipo de Inmueble	Cantidad	Área (mts ²)	Número	Área del terreno (mts ²)	Área de construcción (mts ²)	Alícuota (\$)	# Dormitorios	# Baños
MzA	Casa T1	4	400	A1	100	80	20	3	2
				A2	100	80	20	3	2
				A3	100	80	20	3	2
				A4	100	80	20	3	2
MzB	Casa T2	6	880	B1	150	200	20	4	3
				B2	150	200	20	4	3
				B3	150	200	20	4	3
				B4	150	200	20	4	3
				B5	150	200	20	4	3
				B6	130	180	20	4	3

Los campos que se necesitan registrar son:

- **Número:** Es el identificador del bien inmueble
- **Área del terreno (mts²):** es el área del terreno en el cual se construirá el inmueble.
- **Área de construcción (mts²):** es el área que posee el inmueble.
- **Alícuota (\$):** es la mensualidad que deberá pagar el propietario del bien a la administración del conjunto.
- **# Dormitorios:** es la cantidad de dormitorios que posee el inmueble.
- **# Baños:** es la cantidad de baños que posee el inmueble.

El registro de cada inmueble lo realizará el usuario que se encuentre a cargo del proyecto.

Continúa...

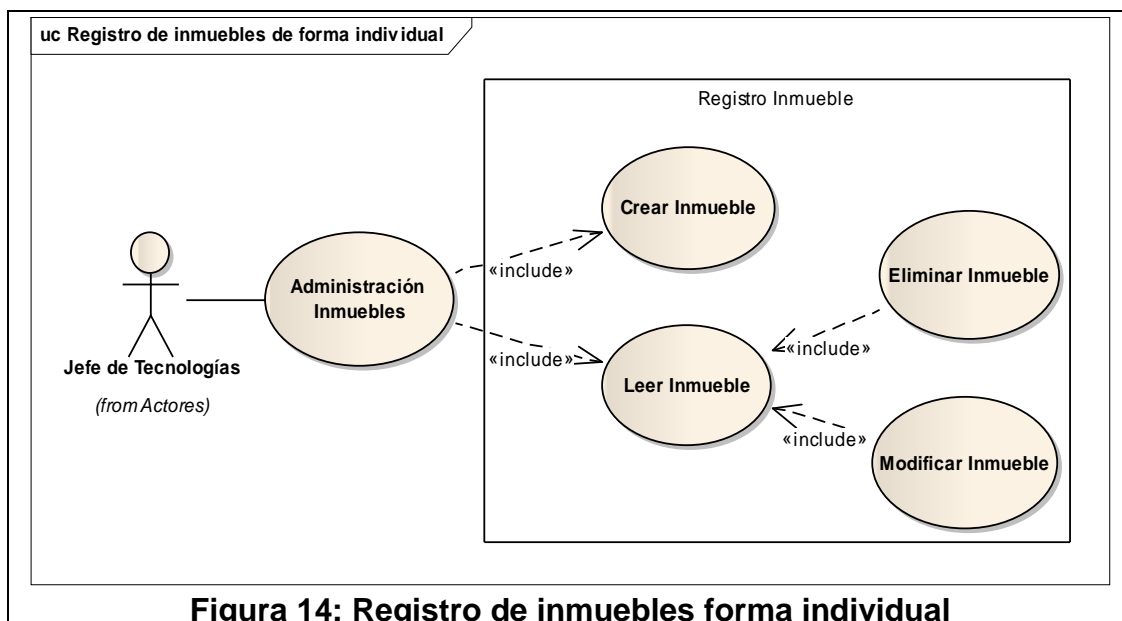


Figura 14: Registro de inmuebles forma individual
3.2.10 Importación de inmuebles

No.	HU-0010				
Proceso Gerencia de Proyecto:	Módulo Base				
Cliente:	Empres Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

- Analizar cómo se va a tener en el sistema la importación de inmuebles.

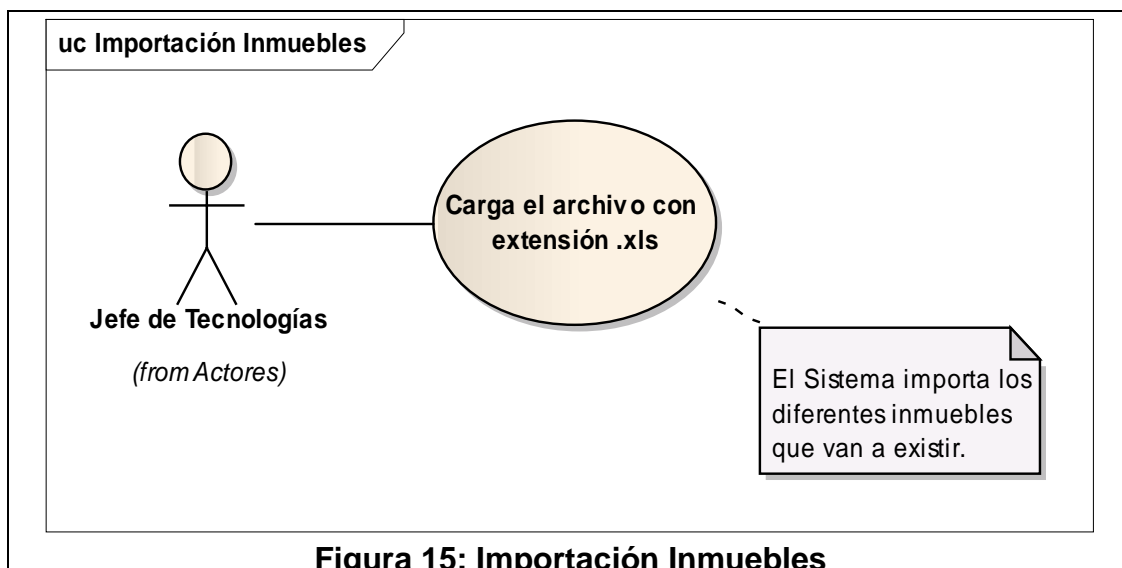
Trabajo Realizado:

Dentro del sistema va a existir la opción para poder importar en masa los diferentes inmuebles, esto se podrá hacer a través de un archivo en formato Excel ya que por cada proyecto existirá un gran número de inmuebles.

Una vez cargado dichos datos, se puede revisar que los mismos se hayan completado de la forma correcta en la opción de Registro de Inmuebles.

La importación de inmuebles, solamente la puede realizar el Jefe de Tecnologías, sin embargo los datos ingresados deben ser visibles para cualquier área que lo requiera.

Continua...



3.2.11 Administración de provincias, cantones, ciudades y

parroquias

No.	HU-0011		
Proceso Gerencia de Proyecto:	Módulo Base		
Cliente:	Empres Provincial de Viviendo E.P.		
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.		
Lugar:	Edificio ISPADE, Cuarto Piso		
Fecha:	07/01/2013	Hora inicio:	09:00
		Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

- Analizar cómo se va a tener en el sistema la administración de Provincias, cantones, parroquias y ciudades.

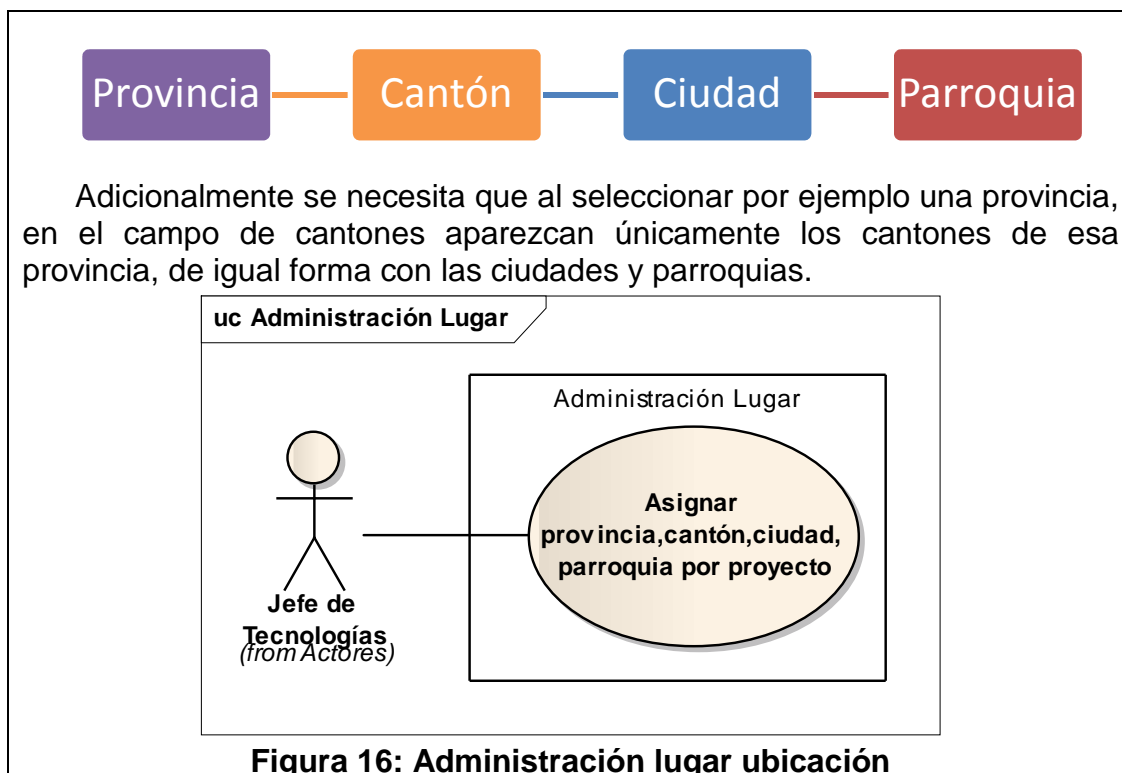
Trabajo Realizado:

Dentro de todo el sistema van a existir opciones para seleccionar la provincia, cantón, parroquias y ciudades.

Un ejemplo claro es en la ficha del proyecto en la cual se necesita que estos cuatro datos se encuentren bien definidos. Sin embargo en otras partes del sistema serán necesarios aunque no sean todos, por ejemplo en la ficha de los contactos deberá constar la ciudad en la que reside el contacto.

Para la administración de los mismos se requiere que exista un orden jerárquico de la siguiente manera:

Continua...



3.2.12 Precios por Inmueble

No.	HU-0012
Proceso Gerencia de Proyecto:	Módulo Comercialización
Cliente:	Empresa Provincial de Viviendo E.P.
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.
Lugar:	Edificio ISPADE, Cuarto Piso
Fecha:	07/01/2013
Hora inicio:	09:00
Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

<ul style="list-style-type: none"> Determinar el procedimiento de cómo se va a manejar el precio por tipo de inmuebles.
--

Trabajo Realizado:

<p>Dentro del proceso de construcción del inmueble, en el análisis de factibilidad de un proyecto se estima el precio de los inmuebles para determinar si un proyecto es o no factible.</p> <p>Ya en el proceso de comercialización de los inmuebles se realiza un análisis por el departamento de comercialización y ventas en el cual se</p>
--

Continua...

determina el precio de cada inmueble.

Dentro del sistema, debe existir la opción para colocar los precios para cada inmueble dependiendo del proyecto. Cabe recalcar que el precio de cada inmueble depende de su área de construcción ya que se fija un precio por cada tipo de inmueble, el precio es por metro cuadrado.

Los datos de entrada requeridos por cada registro son:

- **Nombre:** es el nombre del tipo de inmueble, este campo se debe escoger de los tipos de inmuebles antes creados que posee cada proyecto.
- **Precio por M2:** Es un campo de tipo numérico. Representa el precio del tipo de inmueble por m2.

Solamente los usuarios de comercialización y ventas (Jefatura) pueden registrar el precio de los tipos de inmuebles.

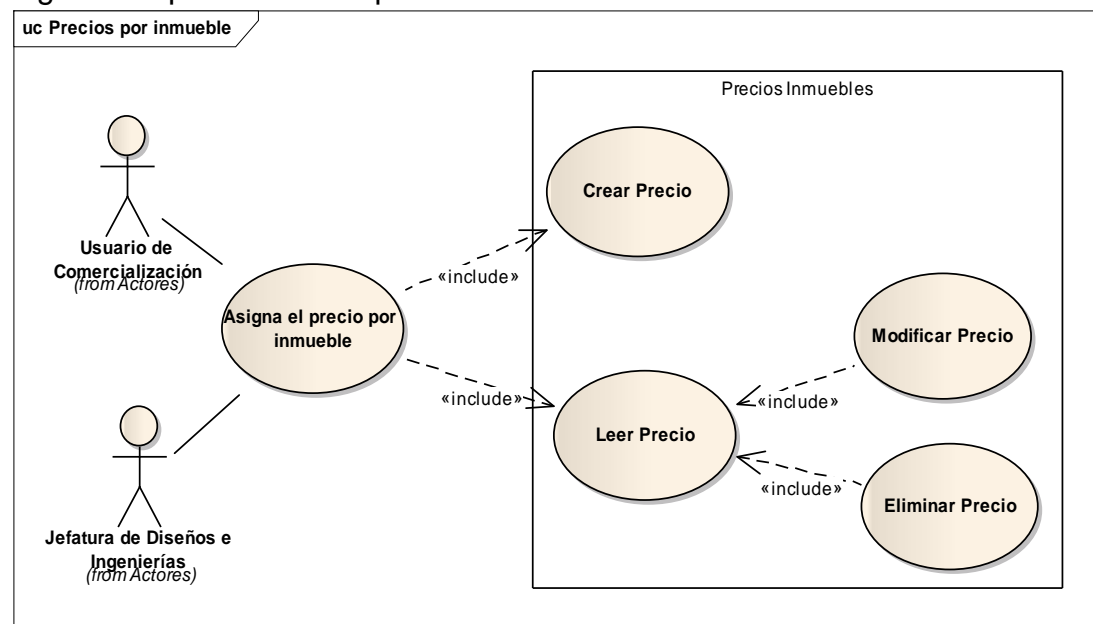


Figura 17: Precios por inmueble

3.2.13 Promociones de Proyectos

No.	HU-0013		
Proceso Gerencia de Proyecto:	Módulo Comercialización		
Cliente:	Empresa Provincial de Viviendo E.P.		
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.		
Lugar:	Edificio ISPADE, Cuarto Piso		
Fecha:	07/01/2013	Hora inicio:	09:00
		Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT – Analista

Continua...

Israel Paredes	ATIKASOFT – Analista
----------------	----------------------

Objetivos:

- Determinar el procedimiento la forma como se va a llevar a cabo las promociones de proyectos.

Trabajo Realizado:

En el análisis realizado por el departamento de comercialización y ventas, también consta las promociones que se van a dar en un proyecto dependiendo de a donde está enfocado el proyecto y a que mercado se apunta el mismo. Dentro de la empresa existen promociones ya determinadas que se han mantenido siempre para todos los proyectos, sin embargo pueden variar por cada proyecto. Existen dos tipos de promociones, una aplicado un valor fijo y otro que aplica un porcentaje de descuento al valor de inmueble a adquirir. Dentro del sistema, debe existir la opción para colocar las diferentes promociones que van a existir para los proyectos.

Los datos de entrada requeridos por cada registro son:

- **Nombre:** Es el nombre de la promoción para el proyecto, este campo lo debe escribir el usuario.
- **Porcentaje de descuento (%):** Es el porcentaje del descuento que se va a realizar por proyecto. Es un campo de tipo de dato entero.
- **Fecha inicio:** Es la fecha en la cual se va a iniciar la promoción para el proyecto. Es un tipo de dato date.
- **Fecha finalización:** Es la fecha final de la promoción para el proyecto. Es un tipo de dato date.
- **Estado:** Es el estado en el que se encuentra la promoción. Este campo no escoge el usuario sino que se crea automáticamente y es de solo lectura.

Solamente los usuarios de comercialización pueden registrar las promociones del proyecto.

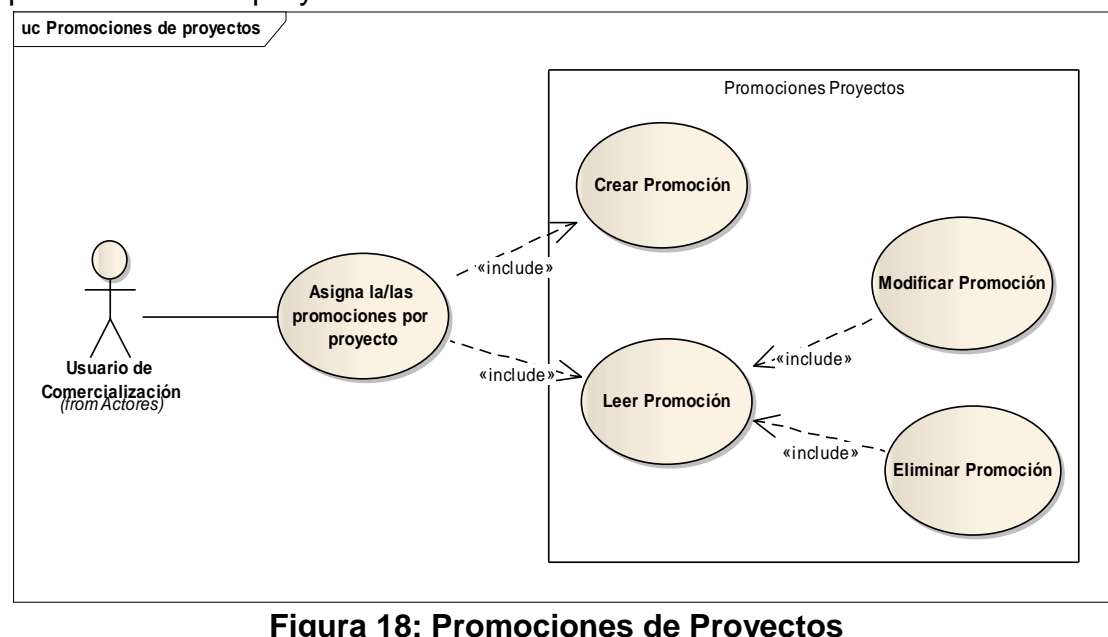


Figura 18: Promociones de Proyectos

3.2.14 Políticas de ventas por proyectos

No.	HU-0014		
Proceso Gerencia de Proyecto:	Módulo Comercialización		
Cliente:	Empresa Provincial de Viviendo E.P.		
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.		
Lugar:	Edificio ISPADE, Cuarto Piso		
Fecha:	07/01/2013	Hora inicio:	09:00
		Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT – Analista
Israel Paredes	ATIKASOFT – Analista

Objetivos:

- Analizar las diferentes políticas de venta que se van a manejar por proyecto.

Trabajo Realizado:

Dentro del sistema, va a existir la opción para colocar las diferentes políticas de venta que se van a manejar por proyecto.

Los datos de entrada requeridos por cada registro son:

- **Nombre:** Es el nombre de la política de venta por proyecto, este campo lo debe escribir el usuario.
- **Porcentaje de total (%):** Es el porcentaje total para la política de venta. Es un campo de tipo de dato entero.
- **Valor:** Es el valor que se le va a dar a dicha política de venta. Es un tipo de dato entero.
- **Plazo (Días):** Es el plazo que se va a otorgar para aplicar dicha política de venta por proyecto.

Solamente los usuarios de comercialización pueden registrar las políticas de venta.

Continua...

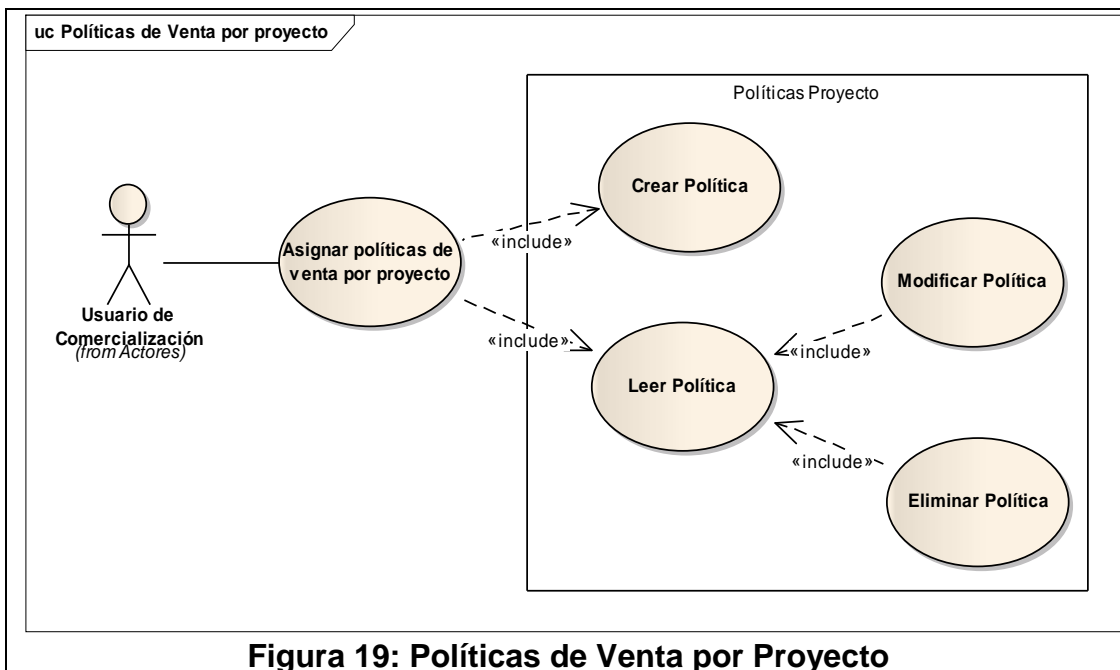


Figura 19: Políticas de Venta por Proyecto

3.2.15 Iniciativas

No.	HU-0015				
Proceso Gerencia de Proyecto:	Módulo Comercialización				
Cliente:	Empres Provincial de Viviendo E.P.				
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.				
Lugar:	Edificio ISPADE, Cuarto Piso				
Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

Objetivos:

- Analizar cómo se va a tener la parte de Iniciativa dentro de un proyecto.

Trabajo Realizado:

Dentro del sistema, va a existir la opción para colocar las iniciativas que son los datos de los posibles clientes interesados en un determinado inmueble. Los datos de entrada requeridos por cada registro son:

- **Asunto:** Es el tema de interés por el cual se desea realizar la compra del bien inmueble.
- **Interés:** Es el interés que tiene dicho cliente en un bien inmueble. Este campo se va a tomar en el rango de alto, bajo y medio.

Continua...

- **Comercial:** Es el usuario encargado del registro de la iniciativa.
- **Oficina:** Es el lugar donde se realiza la inscripción de la iniciativa.
- **Datos del Contacto:** Son los datos principales del interesado en un bien inmueble determinado. Dentro de esta sección existen varios datos, los principales son:

Nombre: Es el nombre completo de la persona interesada.

Cédula: Es la cédula de la persona interesada.

Proyecto Principal: Es el proyecto en el cual está interesado el posible cliente.

Calle: Es la dirección del posible cliente.

- **Información Extra**
- **Comunicaciones**
- **Proyectos de Interés**

Solamente los usuarios de comercialización pueden registrar el precio de los inmuebles.

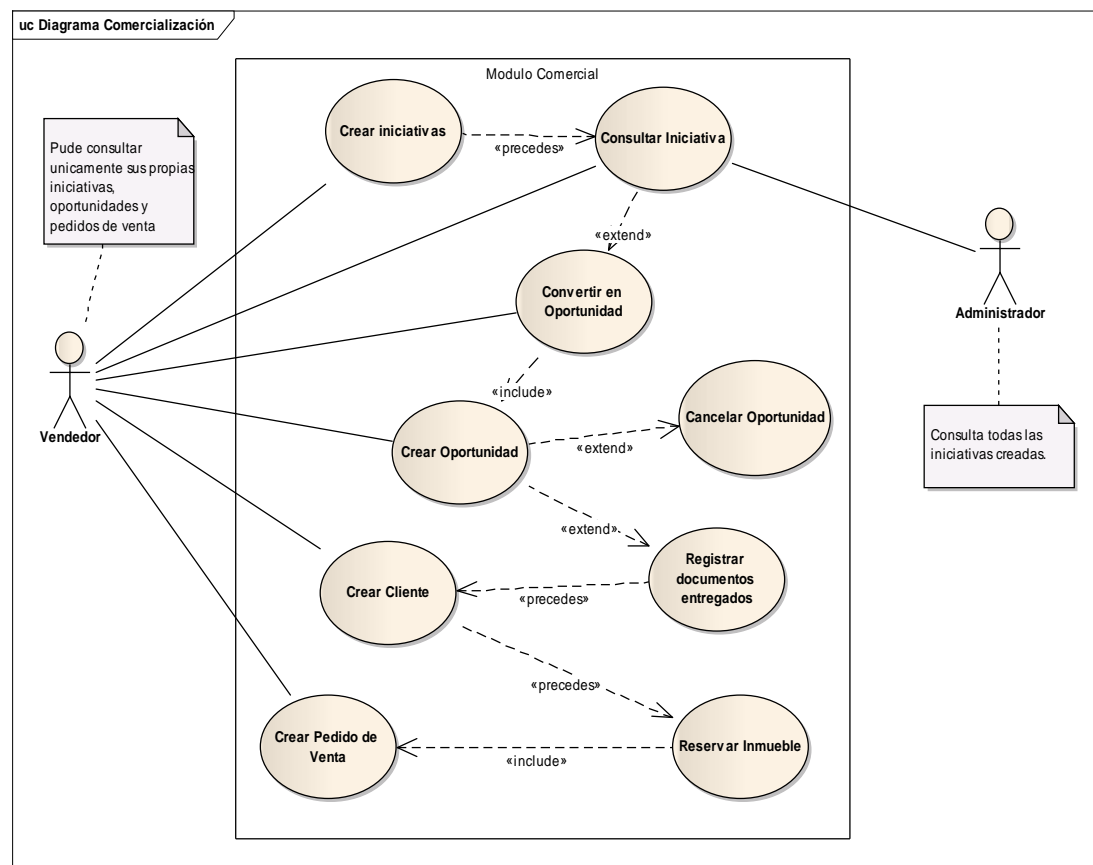


Figura 20: Inicio Comercialización Inmuebles

3.2.16 Oportunidades

No.	HU-0016
Proceso Gerencia de Proyecto:	Módulo Comercialización
Cliente:	Empres Provincial de Viviendo E.P.
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.
Lugar:	Edificio ISPADE, Cuarto Piso

Continua...

Fecha:	07/01/2013	Hora inicio:	09:00	Hora fin:	11:00
---------------	------------	---------------------	-------	------------------	-------

Objetivos:

- Analizar cómo se va a tener la parte de Oportunidades dentro de un proyecto.

Trabajo Realizado:

Dentro del sistema, va a existir la opción para colocar las oportunidades que son los prospectos interesados en un determinado inmueble.

Los datos de entrada requeridos por cada registro son:

- **Asunto Prospecto:** Es el tema de interés de un prospecto por el cual se desea realizar la compra del bien inmueble.
- **Porcentaje de éxito:** Es el porcentaje de éxito que se piensa que se va obtener con dicho prospecto.
- **Registrado por:** Es el usuario encargado del registro de la oportunidad.
- **Datos:** Son los datos ya importantes de dicho prospecto. Dentro de esta sección existen varios datos, los principales son:

Proyecto Principal: Es el proyecto en el cual está interesado el posible cliente.

Cédula: Es la cédula de la persona interesada.

Estado Civil: Es el estado civil del prospecto.

Estudios: Es el estudio al que ha llegado un prospecto.

No. Hijos: Es el número de hijos que tiene dicho prospecto.

Celular: Es el número de celular que tiene dicho prospecto.

Solamente los usuarios de comercialización pueden registrar el precio de los inmuebles.

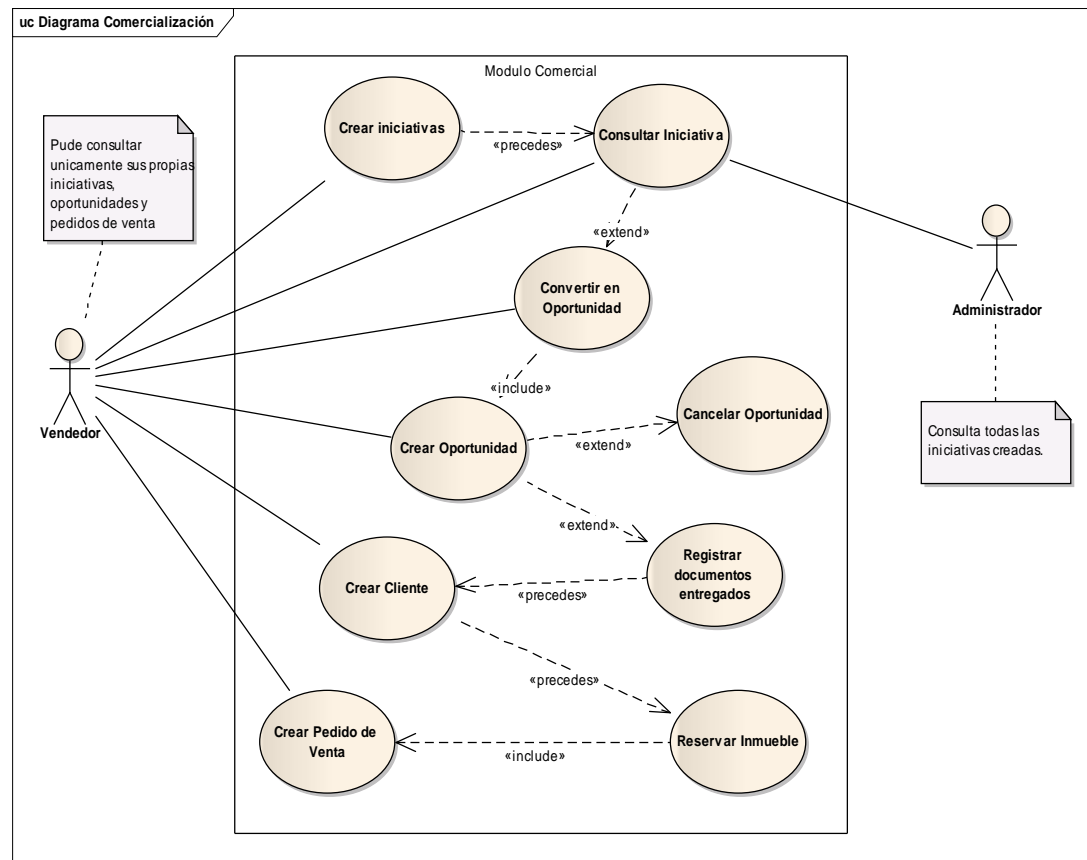


Figura 21: Inicio de Comercialización Inmuebles

3.2.17 Pedidos de Venta

No.	HU-0017		
Proceso Gerencia de Proyecto:	Módulo Comercialización		
Cliente:	Empres Provincial de Viviendo E.P.		
Proyecto:	Adaptación del modelo de negocio de la industria inmobiliaria ecuatoriana al sistema libre OpenERP para la comercialización de bienes inmuebles.		
Lugar:	Edificio ISPADE, Cuarto Piso		
Fecha:	07/01/2013	Hora inicio:	09:00
		Hora fin:	11:00

Participantes:

Nombre	Organización
Víctor Hugo Aguilar	COVIPROV – Jefe de Tecnología
Katherine Robles	ATIKASOFT - Analista
Israel Paredes	ATIKASOFT - Analista

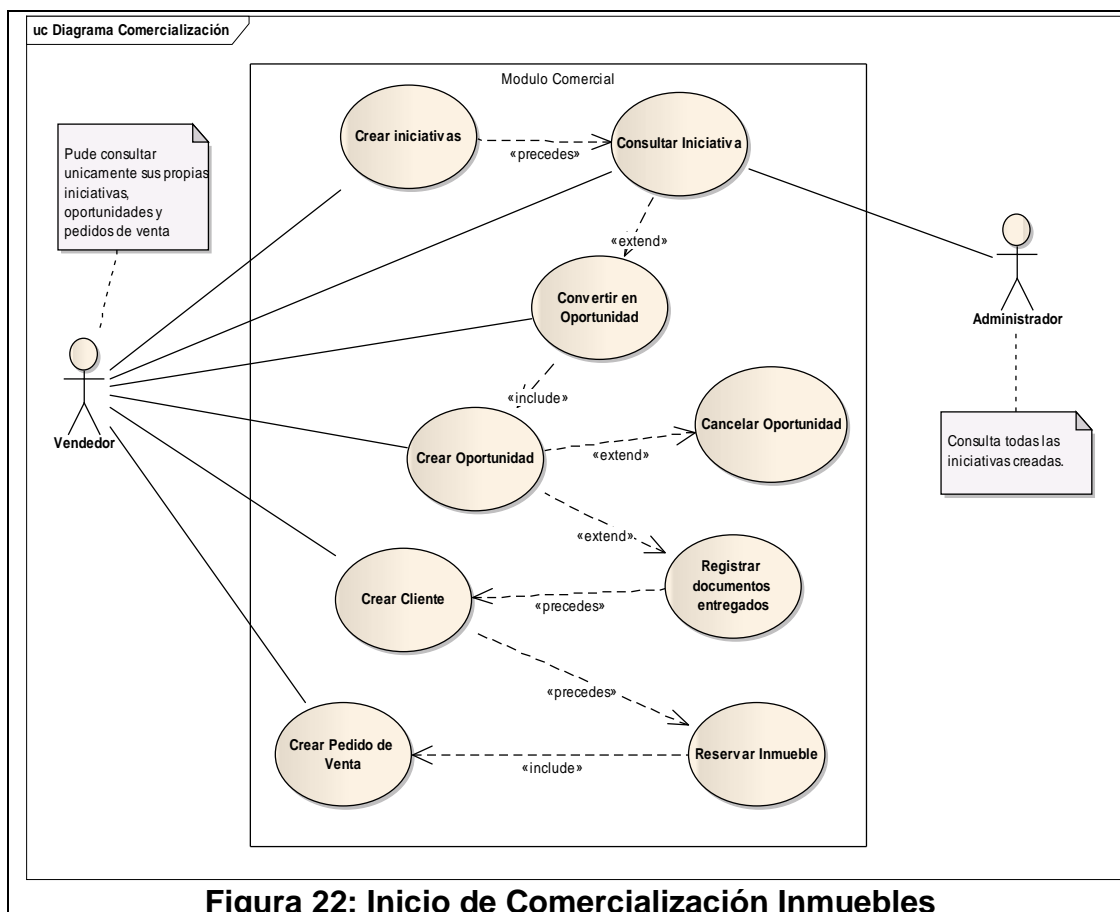
Objetivos:

- Analizar cómo se va a tener en el sistema los Pedidos de Venta.

Trabajo Realizado:

Una vez que un contacto potencial ha realizado los trámites solicitados y presenta con su asesor de ventas, éste último puede realizar la reserva del inmueble y registrar al contacto como cliente de la empresa. Se debe partir del hecho que sus datos ya fueron registrados en el sistema con anterioridad y se sabe por cuales inmuebles está aplicando. Es necesario que desde el sistema se cree automáticamente una reserva (pedido de venta) con los datos registrados.

Continua...



CAPÍTULO 4

DISEÑO

4.1 Modelo de Casos de Uso

4.1.1 Introducción

El presente documento es una descripción de los diferentes casos de uso que existen dentro del sistema para la comercialización de bienes inmuebles, además en el mismo consta una información detallada y concisa de la forma de interactuar entre los usuarios y el sistema.

Este documento es de suma importancia, ya que el mismo les sirve de ayuda a los diferentes usuarios para saber que funciones tienen que cumplir, y en el mismo constan los requerimientos funcionales que constan en el documento de la IEEE 830, que se lo hizo al momento del levantamiento de los requerimientos de software.

4.1.2 Descripción General de Actores

Para este proyecto se necesitan cuatro actores:

ACTORES	DESCRIPCIÓN
Jefe de Tecnologías	Es el administrador del sistema.
Jefatura de Diseño e Ingenierías	Es el usuario encargado de la parte de Diseño e Ingenierías dentro del sistema.
Usuario	Es el encargado de hacer diferentes procesos dentro del sistema, como por ejemplo la creación de un proyecto, creación de los diferentes tipos de inmuebles, etc.
Usuario de Construcciones	Es el encargado de la parte de Construcciones ya del bien inmueble para su respectiva comercialización.

Tabla 5. Actores del proyecto.

4.1.3 Modelo de Casos de Uso

4.1.3.1 Administración de Usuarios [CU-001]

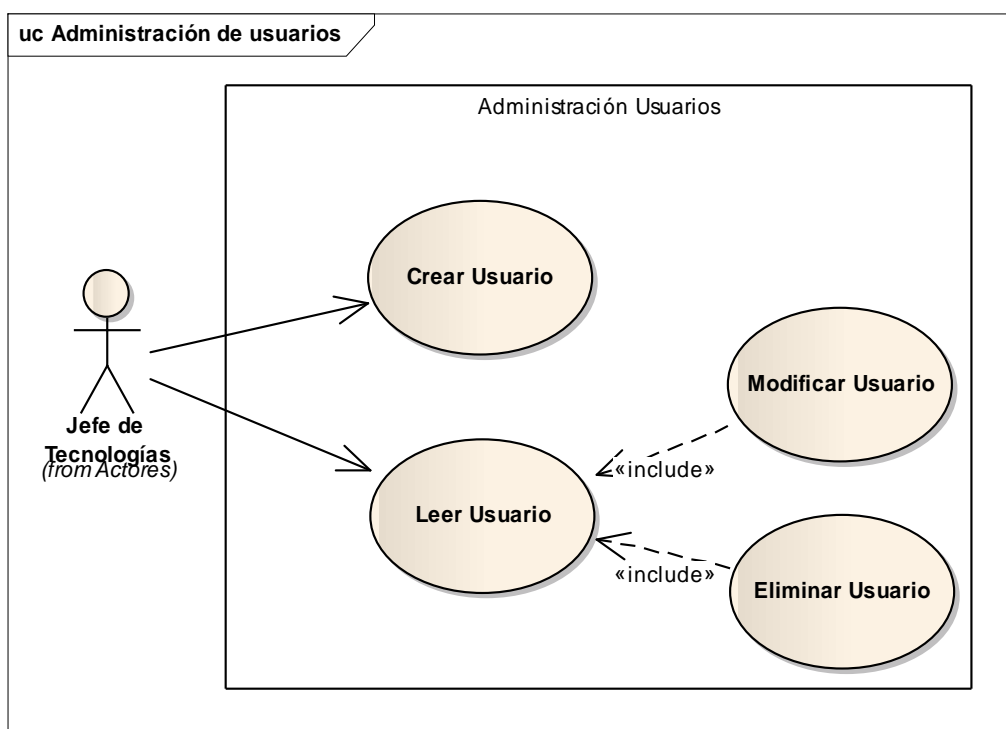


Figura 23: Administración Usuarios

4.1.3.2 Validación de Usuarios [CU-002]

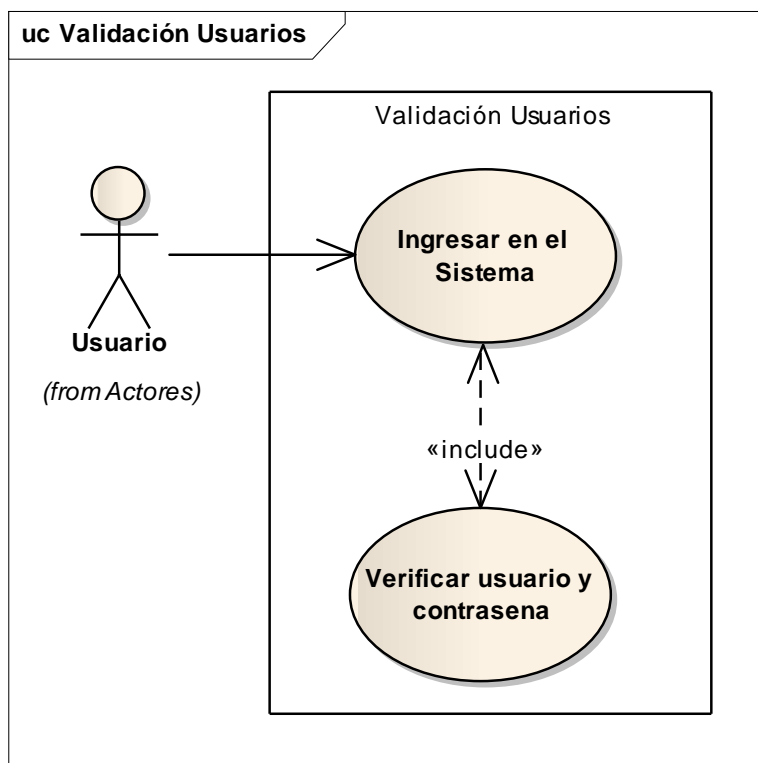


Figura 24: Validación Usuarios

4.1.4 Administración de Tipos de Inmuebles [CU-003]

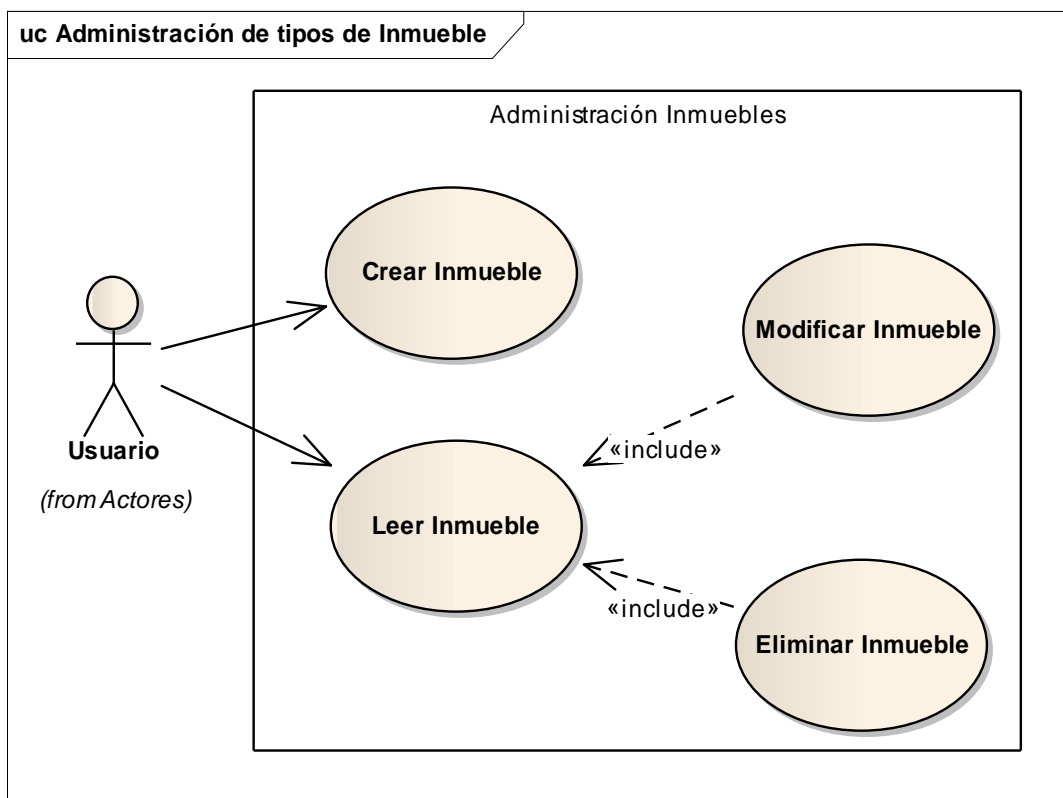


Figura 25: Administración Tipos Inmuebles

4.1.3.4 Administración de Tipos de Ubicaciones [CU-004]

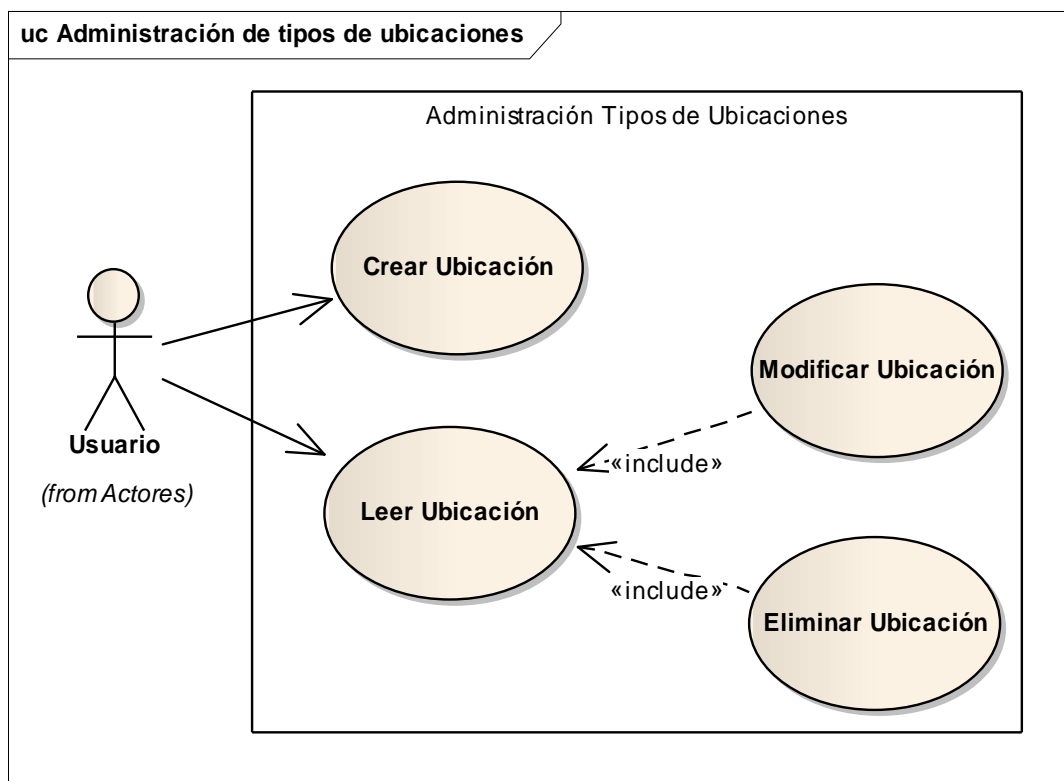


Figura 26: Administración Tipos Ubicaciones

4.1.3.5 Administración de Proyectos [CU-005]

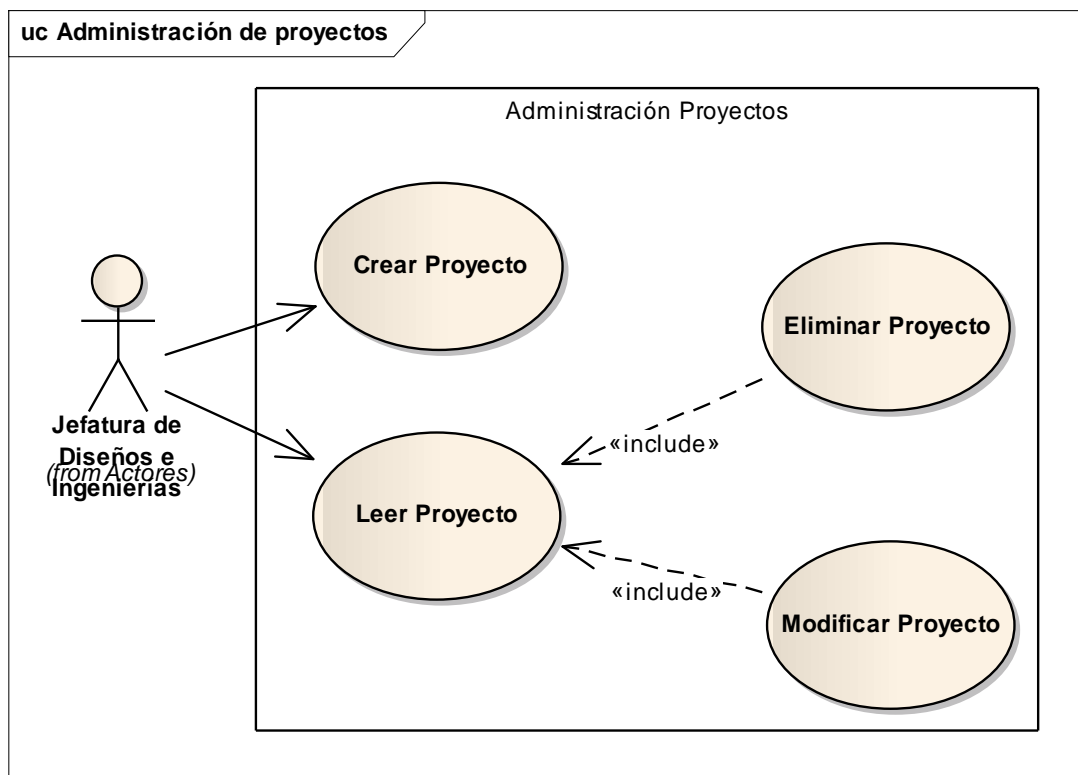


Figura 27: Administración de Proyectos

4.1.3.6 Registro de Tipos de Inmuebles por proyecto [CU-006]

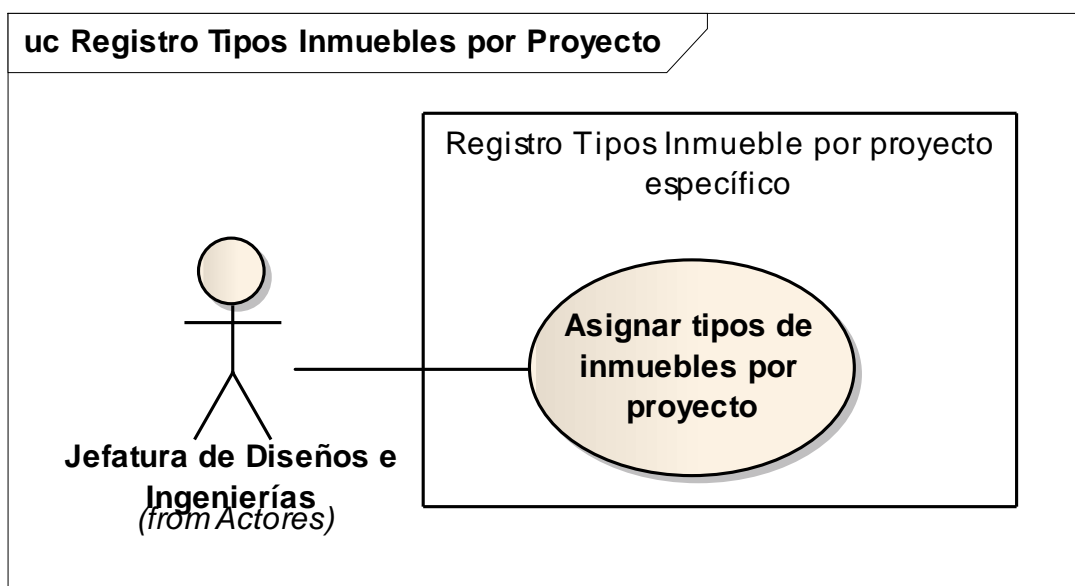


Figura 28: Registro Tipos Inmueble por Proyecto

4.1.3.7 Ubicaciones por Proyecto [CU-007]

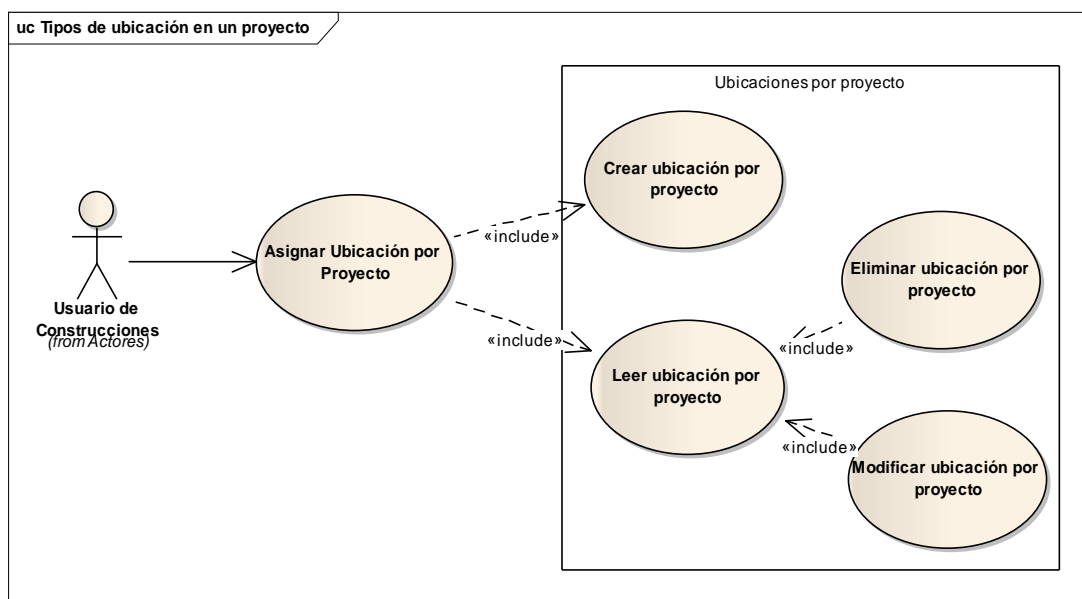


Figura 29: Ubicaciones por Proyecto

4.1.3.8 Ingreso de Tipos de Inmuebles por Ubicación [CU-008]

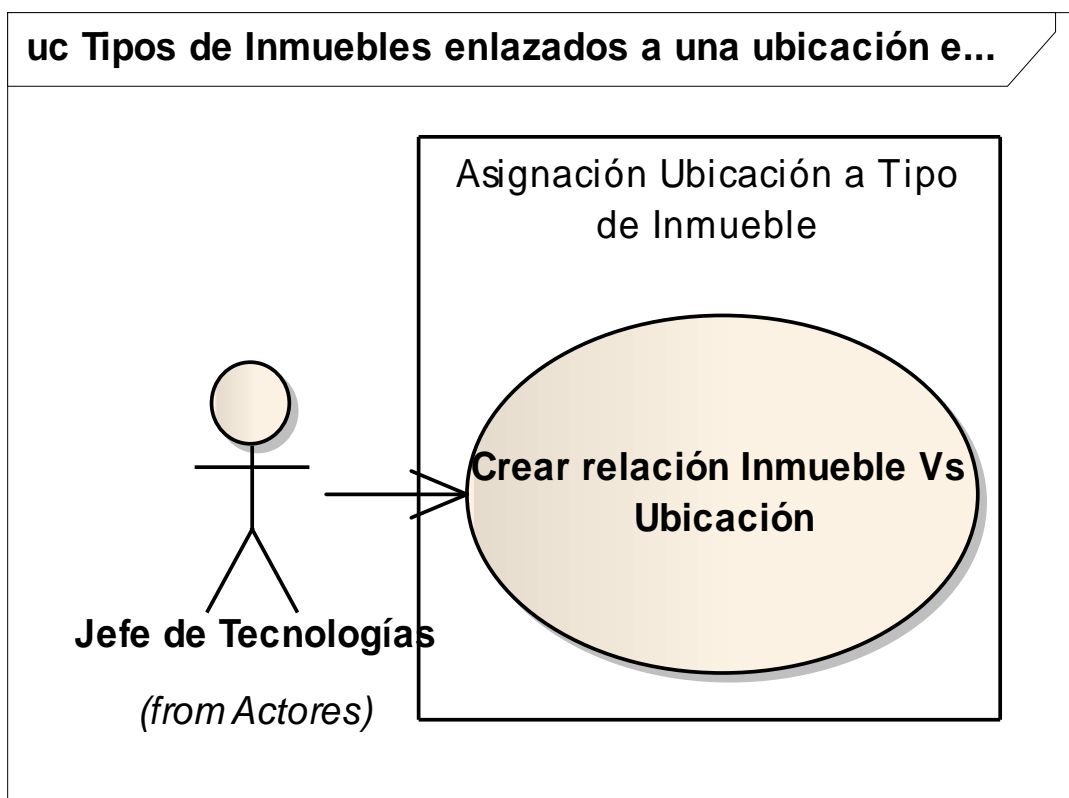


Figura 30: Tipos de Inmueble enlazados ubicación

4.1.3.9 Ingreso de los Inmuebles de forma individual [CU-009]

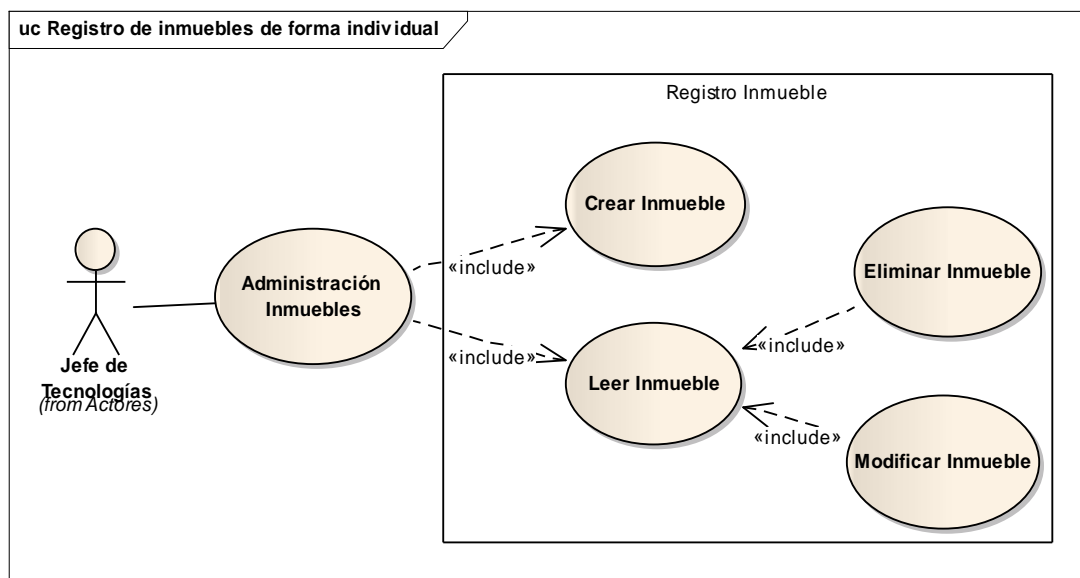


Figura 31: Registro Inmuebles forma individual

4.1.3.10 Importación de Inmuebles [CU-0010]

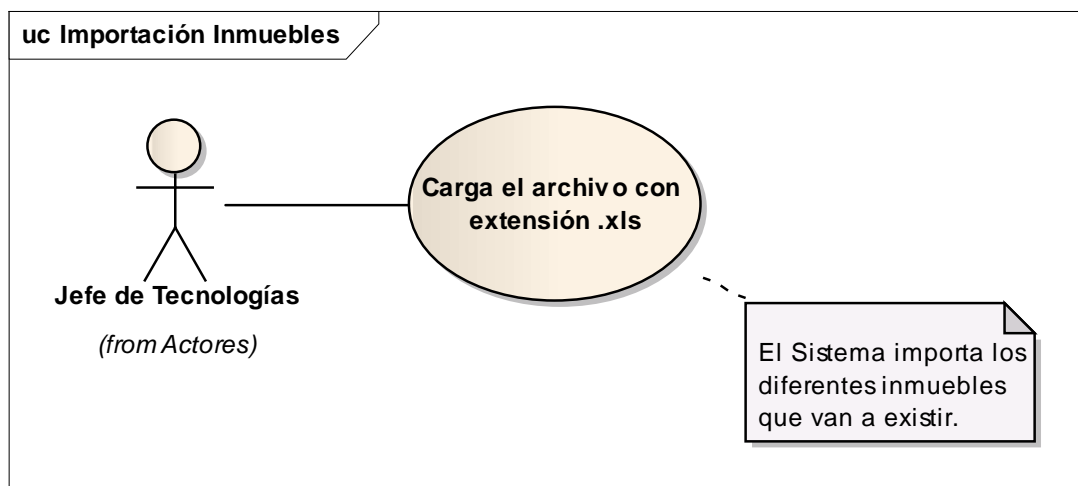


Figura 32: Importación Inmuebles

4.1.3.11 Administración de Provincias, Cantones, Ciudades y Parroquias [CU-0011]

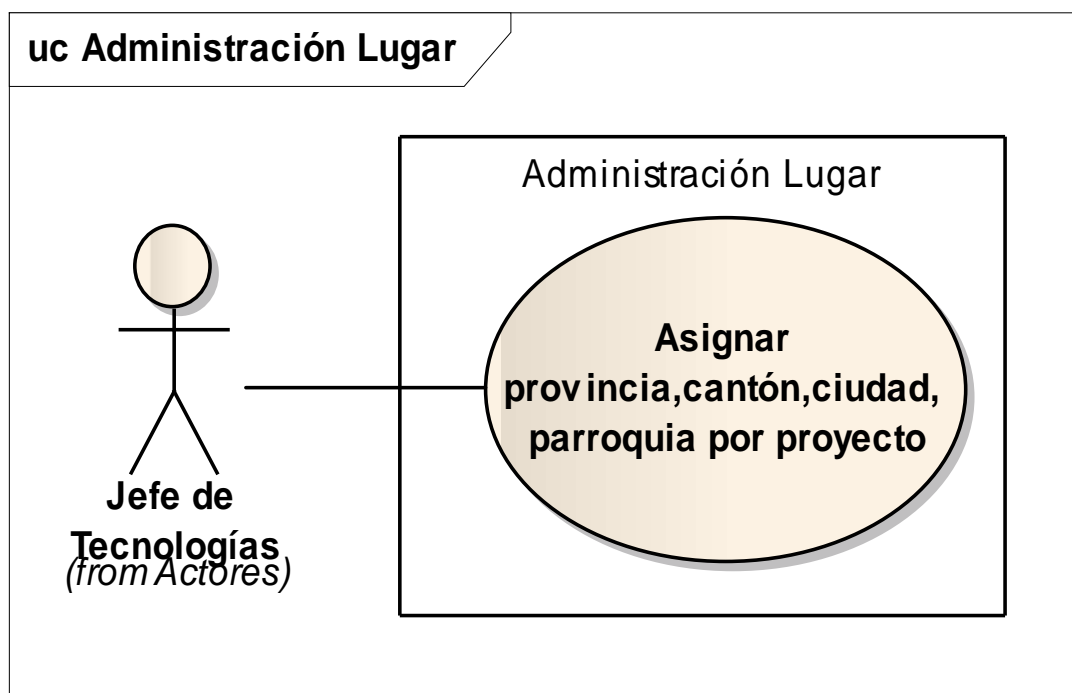


Figura 33: Administración lugar ubicación

4.1.3.12 Precios por inmuebles [CU-0012]

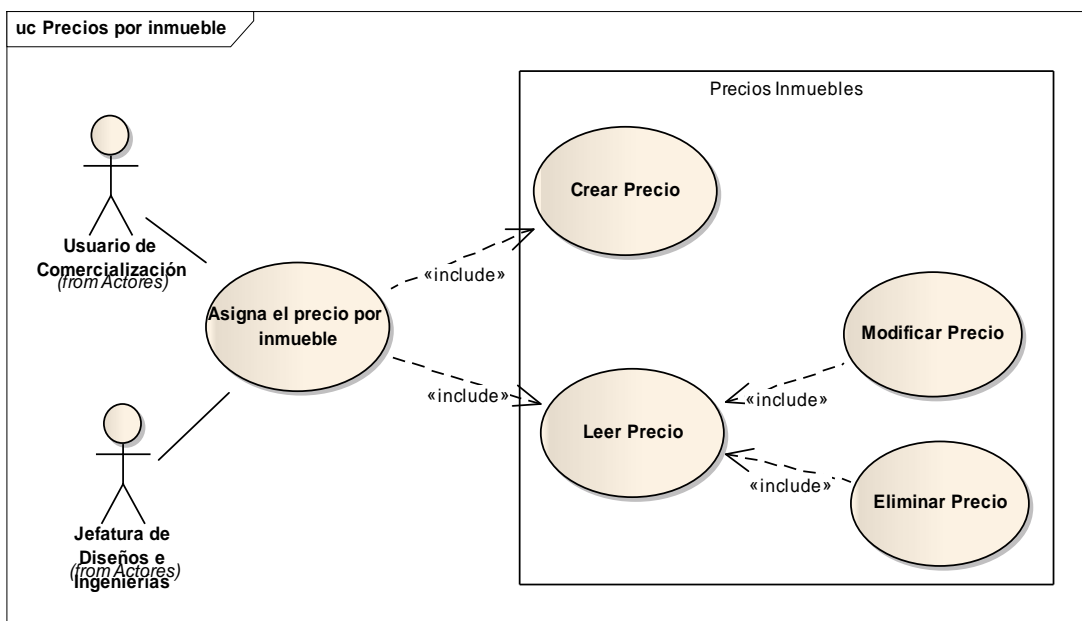


Figura 34: Precios por Inmuebles

4.1.3.13 Promociones de Proyectos [CU-0013]

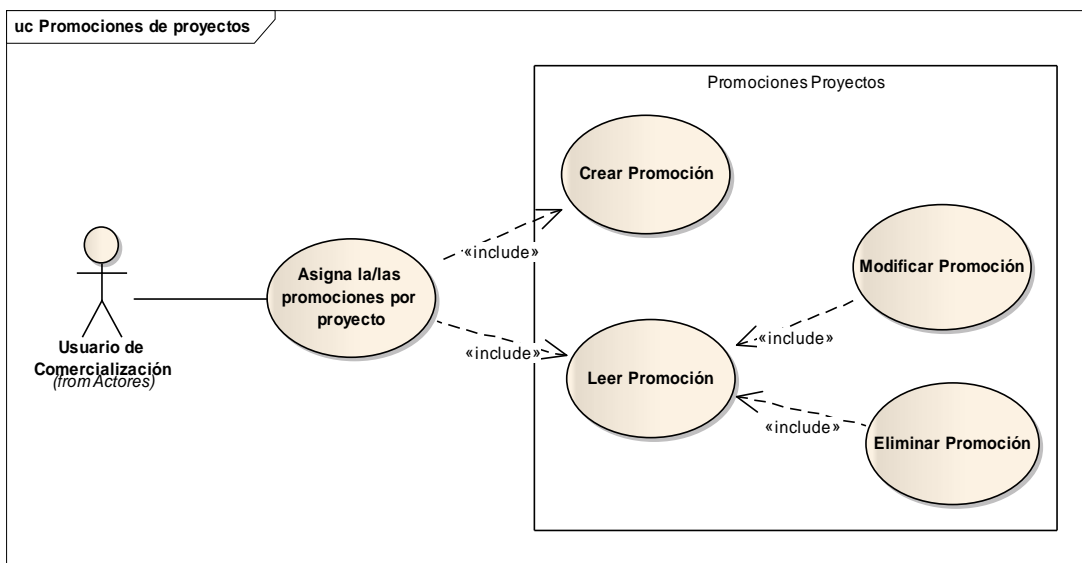


Figura 35: Promociones de Proyectos

4.1.3.14 Políticas de Venta [CU-0014]

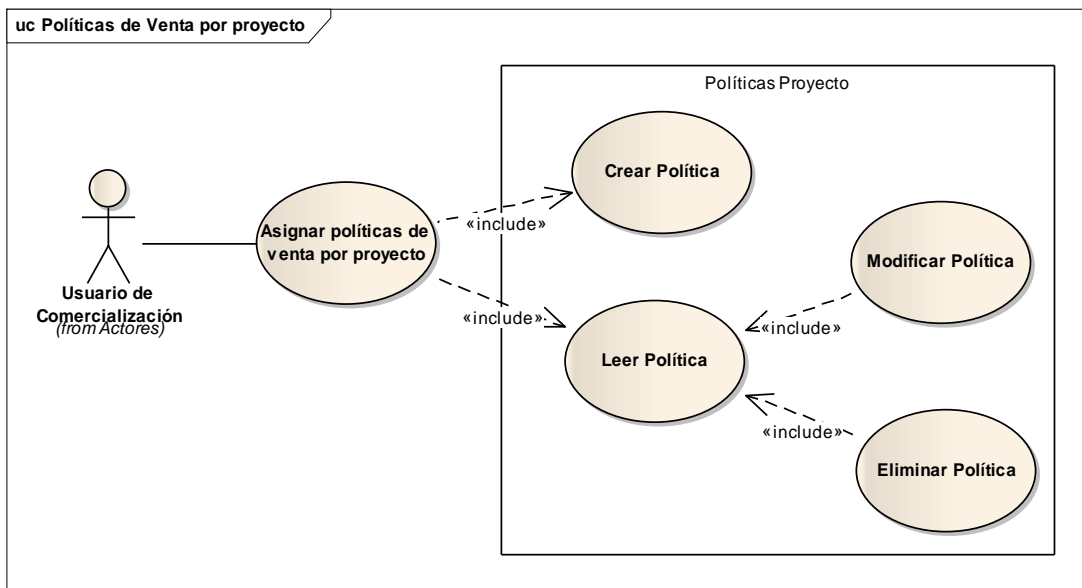


Figura 36: Políticas de Venta

1.1.3.12 Iniciativas [CU-0015], Oportunidades [CU-0016], Pedidos de Venta [CU-0017]

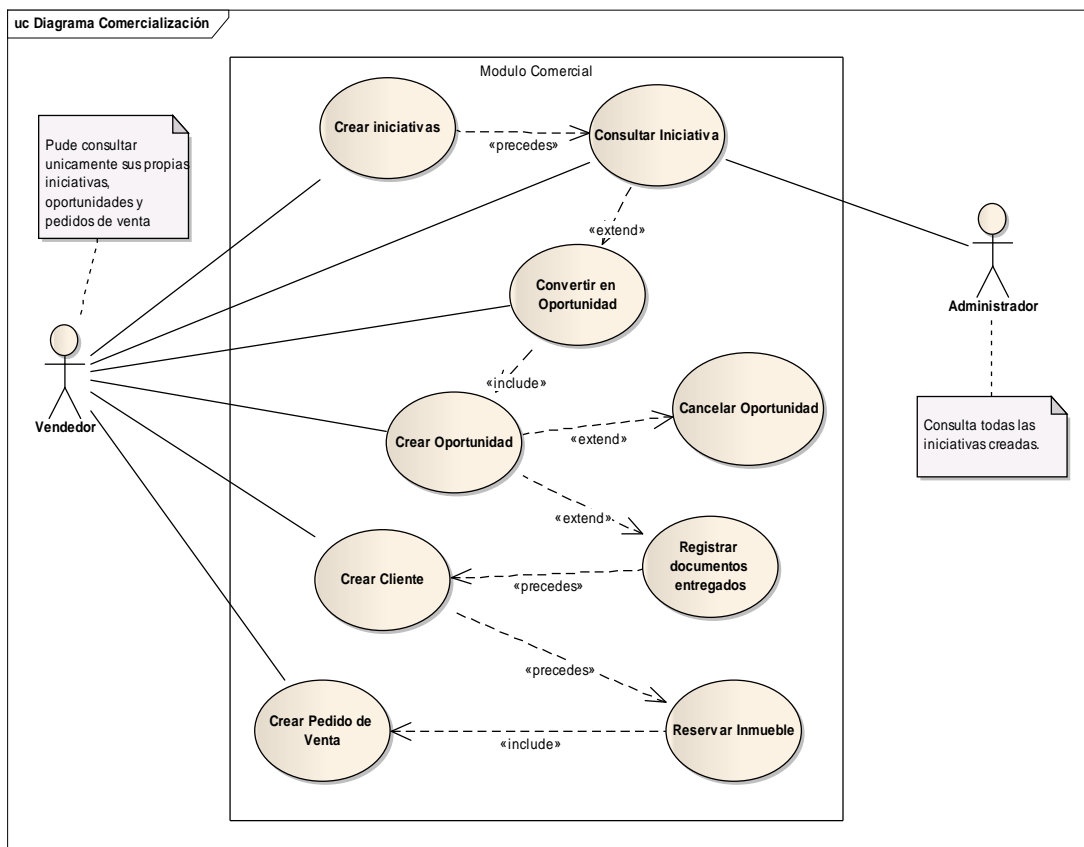


Figura 37: Proceso Comercialización Inmuebles

4.2 ESPECIFICACIÓN DE CASOS DE USO

4.2.1 ADMINISTRACIÓN DE USUARIOS [ECU-001]

4.2.1.1 Nombre

Crear Usuario

4.2.1.1.1 Descripción

Describe el proceso de la creación de un usuario.

4.2.1.1.2 Meta

Que el actor puede crear un usuario dentro del sistema.

4.2.1.1.3 Actores

Jefe de Tecnologías

4.2.1.1.4 Flujo de Eventos

4.2.1.1.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Administración” para proceder con la creación de un usuario.
2. El sistema despliega un formulario con los datos necesarios para la creación del usuario.
3. El actor procede a llenar la información del usuario, «Referencia, HU-001»
4. El sistema guarda la información ingresada.

4.2.1.1.4.2 *Flujo Alternativo*

1. El actor no llena la información requerida para la creación del usuario, «Referencia, HU-001»
2. El sistema no guarda la información.

4.2.1.1.5 Precondiciones

Que el usuario no se encuentre registrado en el sistema.

4.2.1.1.6 Condición de Éxito

El actor ha creado un usuario dentro del sistema.

4.2.1.1.7 Condición de Fallo

El actor no ha creado un usuario dentro del sistema.

4.2.1.2 Nombre

Leer Usuario

4.2.1.2.1 Descripción

Describe el proceso para la lectura de un usuario almacenado en el sistema.

4.2.1.2.2 Meta

Que el actor puede proceder con la lectura de usuarios almacenados en el sistema.

4.2.1.2.3 Actores

Jefe de Tecnologías

4.2.1.2.4 Flujo de Eventos

4.2.1.2.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Administración” para proceder con el proceso.
2. El actor pulsa sobre el menú “Usuarios” para la lectura del usuario.
3. El actor pulsa sobre el “nombre” del usuario.
4. El sistema procede a la lectura del usuario y despliega toda su información.

4.2.1.2.4.2 *Flujo Alternativo*

1. El actor no puede leer la información del usuario.

2. El sistema no despliega la información del usuario.

4.2.1.2.5 Precondiciones

Que el usuario se encuentre registrado en el sistema.

4.2.1.2.6 Condición de Éxito

El actor ha procedido con la lectura del usuario almacenado en el sistema.

4.2.1.2.7 Condición de Fallo

El actor no ha procedido con la lectura del usuario almacenado en el sistema.

4.2.1.3 Nombre

Modificar Usuario

4.2.1.3.1 Descripción

Describe el proceso de la modificación de un usuario.

4.2.1.3.2 Meta

Que el actor puede modificar usuarios almacenados en el sistema.

4.2.1.4 Actores

Jefe de Tecnologías

4.2.1.3.4 Flujo de Eventos

4.2.1.3.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Administración” para proceder con la modificación de un usuario.

2. El sistema despliega una lista de los usuarios creados en el sistema.

3. El actor procede a modificar los datos de un usuario.

4. El sistema guarda la información modificada.

4.2.1.3.4.2 *Flujo Alternativo*

1. El actor no modifica la información del usuario.
2. El sistema no actualiza la información.

4.2.1.3.5 Precondiciones

Que el usuario se encuentre registrado en el sistema.

4.2.1.3.6 Condición de Éxito

El actor procedió a la modificación de un usuario.

4.2.1.3.7 Condición de Fallo

El actor no procedió a la modificación de un usuario.

4.2.1.4 Nombre

Eliminar Usuario

4.2.1.4.1 Descripción

Describe el proceso de la eliminación de un usuario.

4.2.1.4.2 Meta

Que el actor puede eliminar usuarios almacenados en el sistema.

4.2.1.4.3 Actores

Jefe de Tecnologías

4.2.1.4.4 Flujo de Eventos

4.2.1.4.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Administración” para proceder con la eliminación de un usuario.
2. El sistema despliega una lista de los usuarios creados.
3. El actor procede a eliminar a un usuario.
4. El sistema elimina al usuario.

4.2.1.4.4.2 *Flujo Alternativo*

1. El actor no puede eliminar al usuario.
2. El sistema no eliminar al usuario.

4.2.1.4.5 Precondiciones

Que el usuario se encuentre registrado en el sistema.

4.2.1.4.6 Condición de Éxito

El actor procedió a la eliminación de un usuario.

4.2.1.4.7 Condición de Fallo

El actor no procedió a la eliminación de un usuario.

4.2.2 VALIDACIÓN DE USUARIOS [ECU-002]

4.2.2.1 Nombre

Validar Usuario

4.2.2.1.2 Descripción

Describe el proceso como un usuario se autentifica para entrar en el sistema.

4.2.2.1.3 Meta

Que el sistema puede validar el login y contraseña en el sistema, para poder ingresar al mismo.

4.2.2.1.4 Actores

Usuario

4.2.2.1.5 Flujo de Eventos

4.2.2.1.5.1 *Flujo Básico*

Este caso de uso, inicia cuando el usuario desea ingresar al sistema.

1. El sistema requiere que el actor ingrese su login y contraseña.
2. El actor ingresa el login y la contraseña.

3. El sistema valida el login y contraseña ingresados, e inicia la sesión del actor en el sistema.

4.2.2.1.5.2 *Flujo Alternativo*

1. Login y contraseña inválidos.

2. El sistema muestra un mensaje de error “Usuario/Contraseña incorrectos”.

4.2.2.1.6 Precondiciones

Que el usuario se encuentre registrado en el sistema.

4.2.2.1.7 Condición de Éxito

El actor ingreso al sistema.

4.2.2.1.8 Condición de Fallo

El actor no ingreso al sistema.

4.2.3 Administración de tipos de inmuebles [ECU-003]

4.2.3.1 Nombre

Crear Inmueble

4.2.3.1.1 Descripción

Describe el proceso de la creación de un inmueble.

4.2.3.1.2 Meta

Que el actor puede crear los diferentes tipos de inmuebles en el sistema.

4.2.3.1.3 Actores

Usuario

4.2.3.1.4 Flujo de Eventos

4.2.3.1.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Administración de Inmuebles” para proceder con el proceso.
2. El sistema despliega un formulario con los datos necesarios para la creación del inmueble.
3. El actor procede a llenar la información del inmueble.
«Referencia, HU-003»
4. El sistema guarda la información ingresada.

4.2.3.1.4.2 *Flujo Alternativo*

1. El actor no llena la información requerida para la creación del inmueble.
2. El sistema no guarda la información.

4.2.3.1.5 Precondiciones

Que el inmueble no se encuentre registrado en el sistema.

4.2.3.1.6 Condición de Éxito

El actor ha creado un inmueble dentro del sistema.

4.2.3.1.7 Condición de Fallo

El actor no ha creado un inmueble dentro del sistema.

4.2.3.2 Nombre

Leer Inmueble

4.2.3.2.1 Descripción

Describe el proceso para la lectura de un inmueble almacenado en el sistema.

4.2.3.2.2 Meta

Que el actor pueda leer los diferentes tipos de inmuebles en el sistema.

4.2.3.2.3 Actores

Usuario

4.2.3.2.4 Flujo de Eventos

4.2.3.2.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú "Parametrización de Inmuebles" para proceder con el proceso.
2. El actor pulsa sobre el inmueble del cual se desea obtener la información.
3. El sistema procede a la lectura del inmueble y despliega toda su información.

4.2.3.2.4.2 *Flujo Alternativo*

1. El actor no puede leer la información del inmueble.
2. El sistema no despliega la información del inmueble.

4.2.3.2.5 Precondiciones

Que el inmueble se encuentre registrado en el sistema.

4.2.3.2.6 Condición de Éxito

El actor ha procedido con la lectura del inmueble almacenado en el sistema.

4.2.3.2.7 Condición de Fallo

El actor no ha procedido con la lectura del inmueble almacenado en el sistema.

4.2.4 Nombre

Modificar Inmueble

4.2.4.1 Descripción

Describe el proceso de la modificación de un inmueble.

4.2.4.2 Meta

Que el actor pueda modificar los diferentes tipos de inmuebles en el sistema.

4.2.4.3 Actores

Usuario

4.2.4.4 Flujo de Eventos

4.2.4.4.1 Flujo Básico

1. El actor pulsa sobre el menú “Parametrización de Inmuebles” para proceder con el proceso.
2. El sistema despliega una lista de los inmuebles creados en el sistema.
3. El actor procede a modificar los datos de un inmueble.
4. El sistema guarda la información modificada.

4.2.4.4.2 Flujo Alternativo

1. El actor no modifica la información del inmueble.
2. El sistema no actualiza la información.

4.2.4.5 Precondiciones

Que el inmueble se encuentre registrado en el sistema.

4.2.4.6 Condición de Éxito

El actor procedió a la modificación de un usuario.

4.2.4.7 Condición de Fallo

El actor no procedió a la modificación de un usuario.

4.2.3.4 Nombre

Eliminar Inmueble

4.2.3.4.1 Descripción

Describe el proceso de la eliminación de un inmueble.

4.2.3.4.2 Meta

Que el actor pueda eliminar los diferentes tipos de inmuebles en el sistema.

4.2.3.4.3 Actores

Usuario

4.2.3.4.4 Flujo de Eventos

4.2.3.4.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú "Parametrización de Inmuebles" para proceder con el proceso.
2. El sistema despliega una lista de los inmuebles creados.
3. El actor procede a eliminar a un inmueble.
4. El sistema elimina el inmueble.

4.2.3.4.4.2 *Flujo Alternativo*

1. El actor no puede eliminar al inmueble.
2. El sistema no elimina al inmueble.

4.2.3.4.5 Precondiciones

Que el inmueble se encuentre registrado en el sistema.

4.2.3.4.6 Condición de Éxito

El actor procedió a la eliminación de un inmueble.

4.2.3.4.7 Condición de Fallo

El actor no procedió a la eliminación de un inmueble.

4.2.4 Administración de Tipos de Ubicación [ECU-004]

4.2.4.1 Nombre

Crear Tipo de Ubicación

4.2.4.1.1 Descripción

Describe el proceso de la creación de un tipo de ubicación de un inmueble.

4.2.4.1.2 Meta

Que el actor pueda crear un tipo de ubicación para los diferentes inmuebles dentro del sistema.

4.2.4.1.3 Actores

Usuario

4.2.4.1.4 Flujo de Eventos

4.2.4.1.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Tipos de Ubicación” para proceder con el proceso.
2. El sistema despliega un formulario con los datos necesarios para la creación de un tipo de ubicación para un inmueble específico.
3. El actor procede a llenar la información del tipo de ubicación para un inmueble.
4. El sistema guarda la información ingresada.

4.2.4.1.4.2 *Flujo Alternativo*

1. El actor no llena la información requerida para la creación del tipo de ubicación de un inmueble.
2. El sistema no guarda la información.

4.2.4.1.5 Precondiciones

Que el tipo de ubicación para un inmueble no se encuentre registrado en el sistema.

4.2.4.1.6 Condición de Éxito

El actor ha creado un tipo de ubicación para un determinado inmueble dentro del sistema.

4.2.4.1.7 Condición de Fallo

El actor no ha creado un tipo de ubicación para un determinado inmueble dentro del sistema.

4.2.4.2 Nombre

Leer Tipo de Ubicación

4.2.4.2.1 Descripción

Describe el proceso para la lectura de un tipo de ubicación que puede tener un inmueble almacenado en el sistema

4.2.4.2.2 Meta

Que el actor pueda leer un tipo de ubicación para los diferentes inmuebles dentro del sistema.

4.2.4.2.3 Actores

Usuario

4.2.4.2.4 Flujo de Eventos

4.2.4.2.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Tipos de Ubicaciones” para proceder con el proceso.
2. El actor pulsa sobre la ubicación del inmueble del cual se desea obtener la información.

3. El sistema procede a la lectura de la ubicación del inmueble y despliega toda su información.

4.2.4.2.4.2 *Flujo Alternativo*

1. El actor no puede leer la información del tipo de ubicación del inmueble.

2. El sistema no despliega la información de la ubicación del inmueble.

4.2.4.2.5 Precondiciones

Que el tipo de ubicación se encuentre registrado en el sistema.

4.2.4.2.6 Condición de Éxito

El actor ha procedido con la lectura del tipo de ubicación del inmueble almacenado en el sistema.

4.2.4.2.7 Condición de Fallo

El actor ha procedido con la lectura del tipo de ubicación del inmueble almacenado en el sistema.

4.2.4.3 Nombre

Modificar Tipo de Ubicación

4.2.4.3.1 Descripción

Describe el proceso de la modificación de un tipo de ubicación para un inmueble.

4.2.4.3.2 Meta

Que el actor pueda modificar un tipo de ubicación para los diferentes inmuebles dentro del sistema.

4.2.4.4 Actores

Usuario

4.2.4.3.4 Flujo de Eventos

4.2.4.3.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Tipos de Ubicaciones” para proceder con el proceso.
2. El sistema despliega una lista con los diferentes tipos de ubicación asignados a los distintos inmuebles creados.
3. El actor procede a modificar los datos del tipo de ubicación asignado a un inmueble.
4. El sistema guarda la información modificada.

4.2.4.3.4.2 *Flujo Alternativo*

1. El actor no modifica la información del tipo de ubicación de un inmueble.
2. El sistema no actualiza la información.

4.2.4.3.5 Precondiciones

Que el tipo de ubicación de un inmueble se encuentre registrado en el sistema.

4.2.4.3.6 Condición de Éxito

El actor procedió a la modificación de un tipo de ubicación de un inmueble específico.

4.2.4.3.7 Condición de Fallo

El actor no procedió a la modificación de un tipo de ubicación de un inmueble específico.

4.2.4.3 Nombre

Eliminar Tipo de Ubicación

4.2.4.3.1 Descripción

Describe el proceso de la eliminación de un tipo de ubicación para un inmueble determinado.

4.2.4.3.2 Meta

Que el actor pueda eliminar un tipo de ubicación para los diferentes inmuebles dentro del sistema.

4.2.4.4 Actores

Usuario

4.2.4.3.4 Flujo de Eventos

4.2.4.3.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Tipos de Ubicación” para proceder con el proceso.
2. El sistema despliega una lista con los diferentes tipos de ubicación asignados a los distintos inmuebles creados.
3. El actor procede a eliminar un tipo de ubicación de un inmueble.
4. El sistema elimina el tipo de ubicación de un inmueble determinado.

4.2.4.3.4.2 *Flujo Alternativo*

1. El actor no puede eliminar el tipo de ubicación del inmueble.
2. El sistema no elimina el tipo de ubicación del inmueble.

4.2.4.3.5 Precondiciones

Que el tipo de ubicación para un inmueble se encuentre registrado en el sistema.

4.2.4.3.6 Condición de Éxito

El actor procedió a la eliminación de un tipo de ubicación de un determinado inmueble.

4.2.4.3.7 Condición de Fallo

El actor no procedió a la eliminación de un tipo de ubicación de un determinado inmueble.

4.2.5 Administración de Proyectos [ECU-005]

4.2.5.1 Nombre

Crear Proyecto

4.2.5.1.1 Descripción

Describe el proceso de la creación de un proyecto de construcción.

4.2.5.1.2 Meta

Que el actor pueda crear un proyecto dentro del sistema.

4.2.5.1.3 Actores

Jefe de Tecnologías

4.2.5.1.4 Flujo de Eventos

4.2.5.1.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Creación Proyectos” para proceder con el proceso.
2. El sistema despliega un formulario con los datos necesarios para la creación de un proyecto de construcción.
3. El actor procede a llenar la información del proyecto de construcción. «Referencia, HU-005»
4. El sistema guarda la información ingresada.

4.2.5.1.4.2 *Flujo Alternativo*

1. El actor no llena la información requerida para la creación del proyecto de construcción. «Referencia, HU-005»

2. El sistema no guarda la información.

4.2.5.1.5 Precondiciones

Que el proyecto de construcción no se encuentre registrado en el sistema.

4.2.5.1.6 Condición de Éxito

El actor ha creado un proyecto de construcción dentro del sistema.

4.2.5.1.7 Condición de Fallo

El actor no ha creado un proyecto de construcción dentro del sistema.

4.2.5.2 Nombre

Leer Proyecto

4.2.5.2.1 Descripción

Describe el proceso para la lectura de un proyecto almacenado en el sistema.

4.2.5.2.2 Meta

Que el actor pueda leer un proyecto dentro del sistema.

4.2.5.2.3 Actores

Jefe de Tecnologías

4.2.5.2.4 Flujo de Eventos

4.2.5.2.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Creación Proyectos” para proceder con el proceso.

2. El actor pulsa sobre el proyecto de construcción del cual se desea obtener la información.

3. El sistema procede a la lectura del proyecto de construcción y despliega toda su información.

4.2.5.2.4.2 *Flujo Alternativo*

1. El actor no puede leer la información del proyecto de construcción.

2. El sistema no despliega la información del proyecto de construcción.

4.2.5.2.5 Precondiciones

Que el proyecto de construcción no se encuentre registrado en el sistema.

4.2.5.2.6 Condición de Éxito

El actor ha procedido con la lectura del proyecto de construcción almacenado en el sistema.

4.2.5.2.7 Condición de Fallo

El actor no ha procedido con la lectura del proyecto de construcción almacenado en el sistema.

4.2.5.3 Nombre

Modificar Proyecto

4.2.5.3.1 Descripción

Describe el proceso de la modificación de un proyecto de construcción.

4.2.5.3.2 Meta

Que el actor pueda modificar un proyecto dentro del sistema.

4.2.5.4 Actores

Jefe de Tecnologías

4.2.5.3.4 Flujo de Eventos

4.2.5.3.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Creación Proyectos” para proceder con el proceso.

2. El sistema despliega una lista con los diferentes proyectos de construcción creados en el sistema.

3. El actor procede a modificar los datos del proyecto de construcción.

4. El sistema guarda la información modificada.

4.2.5.3.4.2 *Flujo Alternativo*

1. El actor no modifica la información del proyecto de construcción.

2. El sistema no actualiza la información.

4.2.5.3.5 Precondiciones

Que el proyecto de construcción se encuentre registrado en el sistema.

4.2.5.3.6 Condición de Éxito

El actor procedió a la modificación de un proyecto de construcción determinado.

4.2.5.3.7 Condición de Fallo

El actor procedió a la modificación de un proyecto de construcción determinado.

4.2.5.4 Nombre

Eliminar Proyecto

4.2.5.4.1 Descripción

Describe el proceso de la eliminación de un proyecto de construcción.

4.2.5.4.2 Meta

Que el actor pueda eliminar un proyecto dentro del sistema.

4.2.5.4.3 Actores

Jefe de Tecnologías

4.2.5.4.4 Flujo de Eventos

4.2.5.4.4.1 *Flujo Básico*

1. El actor pulsa sobre el menú “Creación Proyectos” para proceder con el proceso.
2. El sistema despliega una lista con los diferentes proyectos de construcción creados en el sistema.
3. El actor procede a eliminar un proyecto de construcción.
4. El sistema elimina un proyecto de construcción determinado.

4.2.5.4.4.2 *Flujo Alternativo*

1. El actor no puede eliminar un proyecto de construcción.
2. El sistema no elimina el proyecto de construcción.

4.2.5.4.5 Precondiciones

Que el proyecto de construcción se encuentre registrado en el sistema.

4.2.5.4.6 Condición de Éxito

El actor procedió a la eliminación de un proyecto de construcción determinado.

4.2.5.4.7 Condición de Fallo

El actor no procedió con la eliminación de un proyecto de construcción determinado.

4.2.6 Registros de tipos de inmueble por proyecto [ECU-006]

4.2.6.1 Nombre

Registros de tipos de inmueble por proyecto

4.2.6.1.1 Descripción

Describe el proceso para el registro de los tipos inmuebles antes ingresados por cada proyecto.

4.2.6.1.2 Meta

Que el actor pueda registrar los tipos de inmueble por proyecto dentro del sistema.

4.2.6.1.3 Actores

Jefatura de Diseños e Ingenierías

4.2.6.1.4 Flujo de Eventos

4.2.6.1.4.1 *Flujo Básico*

Este caso de uso, inicia cuando el usuario desea asignar los tipos de inmuebles por cada proyecto que existe.

1. El sistema despliega un formulario.
2. El actor ingresa los datos de entrada. «Referencia, HU-006»
3. El sistema asigna los tipos de inmueble por cada proyecto existente.

4.2.6.1.4.2 *Flujo Alternativo*

1. El actor no asigna ningún tipo de inmueble por proyecto.
2. El sistema no puede asignar una un tipo de inmueble por proyecto.

4.2.6.1.5 Precondiciones

Que ya este registrado en el sistema los tipos de inmueble y los proyectos.

4.2.6.1.6 Condición de Éxito

El actor asigna uno o varios tipos de inmueble por proyecto dentro del sistema.

4.2.7 Ubicaciones por proyecto [ECU-007]

4.2.7.1 Nombre

Asignar ubicaciones por proyecto

4.2.7.1.1 Descripción

Describe el proceso para la asignación de una ubicación para un proyecto de construcción determinado.

4.2.7.1.2 Meta

Que el actor pueda asignar ubicaciones por proyecto dentro del sistema.

4.2.7.1.3 Actores

Usuario de Construcciones

4.2.7.1.4 Flujo de Eventos

4.2.7.1.4.1 Flujo Básico

Este caso de uso, inicia cuando el usuario desea definir una ubicación por proyecto.

4. El sistema despliega un formulario.

5. El actor ingresa los datos de entrada. «Referencia, HU-007»

6. El sistema asigna una ubicación a un proyecto. Cada ubicación posee una nomenclatura específica y a su vez puede pertenecer a otra ubicación.

4.2.7.1.4.2 Flujo Alternativo

3. El actor ingresa información errónea al momento de la asignación de una ubicación para un proyecto de construcción.

4. El sistema no puede asignar una ubicación al proyecto.

4.2.7.1.5 Precondiciones

Que ya este registrado en el sistema la ubicación para un proyecto de construcción determinado.

4.2.7.1.6 Condición de Éxito

El actor asigno una ubicación para un proyecto dentro del sistema.

Por ejemplo:

Ubicación	Identificador	Tipo de ubicación	Nomenclatura	Ubicación general
	A	Mz	MzA	MzA
MzA	B	T	TB	MzA – TB
MzA - T1	10	Ps	Ps10	MzA - TB - Ps10
MzA - T1	20	Ps	Ps20	MzA - TB - Ps20

Tabla 6. Ejemplo de ubicaciones de un proyecto.

4.2.7.1.7 Condición de Fallo

El actor no asigno una ubicación para un proyecto dentro del sistema.

4.2.8 Ingreso de tipos de inmuebles por ubicación

[ECU-008]

4.2.8.1 Nombre

Asignar los tipos de inmuebles a una ubicación en un determinado proyecto

4.2.8.1.1 Descripción

Describe el proceso para enlazar un tipo de inmueble a una ubicación específica en un determinado proyecto.

4.2.8.1.2 Meta

Que el actor pueda asignar tipos de inmuebles a una ubicación específica dentro del sistema.

4.2.8.1.3 Actores

Jefe de Tecnologías

4.2.8.1.4 Flujo de Eventos

4.2.8.1.4.1 *Flujo Básico*

Este caso de uso, inicia cuando el usuario desea relacionar un tipo de inmueble a una ubicación en un determinado proyecto.

1. El sistema despliega un formulario.
2. El actor ingresa los datos de entrada. «Referencia, HU-008»
3. El sistema asigna una ubicación a un tipo determinado de inmueble en proyecto específico. Cada ubicación final posee un determinado tipo de inmueble, independientemente de cuantos inmuebles de ese tipo sean construidos.

4.2.8.1.4.2 *Flujo Alternativo*

1. El actor ingresa información errónea al momento de la asignación de una ubicación a un tipo de inmueble en un determinado proyecto de construcción.
2. El sistema no puede asignar dicha ubicación.

4.2.8.1.5 Precondiciones

Que en el sistema ya se encuentren registrados los diferentes tipos de inmuebles, las distintas ubicaciones para un proyecto de construcción determinado.

4.2.8.1.6 Condición de Éxito

El actor asigna una ubicación a un tipo de inmueble en un determinado proyecto dentro del sistema.

4.2.8.1.7 Condición de Fallo

El actor asigna una ubicación a un tipo de inmueble en un determinado proyecto dentro del sistema.

4.2.9 Ingreso de los inmuebles de forma individual [ECU-009]

4.2.9.1 Nombre

Registro individual de inmuebles en un determinado proyecto

4.2.9.1.1 Descripción

Describe el proceso para el registro individual de inmuebles que corresponde a las unidades habitacionales que se construyen en un proyecto. Representa exactamente al inmueble dentro del proyecto.

4.2.9.1.2 Meta

Que el actor pueda registrar inmuebles de forma individual dentro del sistema.

4.2.9.1.3 Actores

Jefe de Tecnologías

4.2.9.1.4 Flujo de Eventos

4.2.9.1.4.1 *Flujo Básico*

Este caso de uso, inicia cuando el usuario desea registrar un inmueble de forma individual en el sistema.

1. El sistema despliega un formulario.
2. El actor ingresa los datos de entrada. «Referencia, HU-009»

3. El sistema asigna un inmueble a un determinado proyecto.

4.2.9.1.4.2 *Flujo Alternativo*

1. El actor ingresa varios inmuebles a un proyecto específico.

2. El sistema no puede asignar dichos inmuebles.

4.2.9.1.5 Precondiciones

Que en el sistema ya se encuentren registrados los diferentes tipos de inmuebles en una ubicación y la cantidad de inmuebles que van a tener.

4.2.9.1.6 Condición de Éxito

El actor ingreso un inmueble de forma individual dentro del sistema.

4.2.9.1.7 Condición de Fallo

El actor no ingreso un inmueble de forma individual dentro del sistema.

4.2.10 *Importación de Inmuebles [CU-0010]*

4.2.10.1 Nombre

Importación de Inmuebles

4.2.10.1.1 Descripción

Describe el proceso para la importación de inmuebles desde un archivo en Excel.

4.2.10.1.2 Meta

Que el sistema pueda importar inmuebles.

4.2.10.1.3 Actores

Jefe de Tecnologías

4.2.10.1.4 Flujo de Eventos

4.2.10.1.4.1 Flujo Básico

Este caso de uso, inicia cuando el usuario desea importar inmuebles en masa.

1. El sistema despliega un formulario.
2. El actor ingresa los datos de entrada. «Referencia, HU-0010»
3. El sistema importa dichos inmuebles en masa.

4.2.10.1.4.2 Flujo Alternativo

1. El actor no ingresa los inmuebles.
2. El sistema no puede importar los inmuebles.

4.2.10.1.5 Precondiciones

Que en el sistema ya se encuentren registrados los diferentes proyectos y los inmuebles.

4.2.10.1.6 Condición de Éxito

El sistema importo los distintos inmuebles en masa para continuar con el proceso de construcción y comercialización de bienes inmuebles.

4.2.10.1.7 Condición de Fallo

El sistema no pudo importar los inmuebles.

4.2.11 Administración de Provincias, Cantones, Ciudades y Parroquias [ECU-0011]

4.2.11.1 Nombre

Administración Provincias, Cantones, Ciudades y Parroquias dentro del Sistema de Comercialización de Bienes Inmuebles.

4.2.11.1.1 Descripción

Describe el proceso para la administración de provincias, cantones, ciudades y parroquias dentro del sistema.

4.2.11.1.2 Meta

Que el actor pueda administrar provincias con sus respectivos cantones, ciudades y parroquias dentro del sistema.

4.2.11.1.3 Actores

Jefe de Tecnologías

4.2.11.1.4 Flujo de Eventos

4.2.11.1.4.1 *Flujo Básico*

1. El sistema despliega un formulario.
2. El actor escoge una provincia con su respectivo cantón, ciudad y parroquia para un determinado proyecto.
3. El sistema asigna la información anterior a un proyecto específico.

4.2.11.1.4.2 *Flujo Alternativo*

1. El actor no asigna ninguna provincia y por ende ninguna de las subsiguientes.

4.2.11.1.5 Precondiciones

Que en el sistema ya estén creados proyectos, para la respectiva asignación

4.2.11.1.6 Condición de Éxito

El actor asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.11.1.7 Condición de Fallo

El actor no asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.12 Precios por Inmueble [ECU-0012]

4.2.12.1 Nombre

Precios por Inmueble

4.2.12.1.1 Descripción

Describe el proceso para la administración de los precios por inmueble dentro del sistema.

4.2.12.1.2 Meta

Que el actor pueda ingresar los distintos precios por inmuebles dentro del sistema.

4.2.12.1.3 Actores

Jefe de Tecnologías

4.2.12.1.4 Flujo de Eventos

4.2.12.1.4.1 *Flujo Básico*

1. El sistema despliega un formulario.
2. El actor ingresa los datos de entrada para fijar los distintos precios por inmueble.
3. El sistema asigna los precios por inmueble.

4.2.12.1.4.2 *Flujo Alternativo*

1. El actor no asigna ningún precio al inmueble.

4.2.12.1.5 Precondiciones

Que en el sistema ya este asignado un precio a ese inmueble, y por ende no se va a ingresar datos.

4.2.12.1.6 Condición de Éxito

El actor asignó los precios por inmueble.

4.2.12.1.7 Condición de Fallo

El actor no asignó los precios por inmueble.

4.2.13 Promociones de Proyectos [ECU-0013]

4.2.13.1 Nombre

Promociones de Proyectos dentro del Sistema de Comercialización de Bienes Inmuebles.

4.2.13.1.1 Descripción

Describe el proceso para la administración de las distintas promociones que puedan existir para los distintos proyectos.

4.2.13.1.2 Meta

Que el actor pueda administrar las diferentes promociones de proyectos dentro del sistema.

4.2.13.1.3 Actores

Jefe de Tecnologías

4.2.13.1.4 Flujo de Eventos

4.2.13.1.4.1 Flujo Básico

1. El sistema despliega un formulario.
2. El actor ingresa las distintas promociones que van a existir para los proyectos.
3. El sistema almacena la información de las promociones por proyectos.

4.2.13.1.4.2 Flujo Alternativo

1. El actor no ingresa ninguna promoción.

4.2.13.1.5 Precondiciones

Que en el sistema ya estén almacenadas las promociones para todos los proyectos existentes.

4.2.13.1.6 Condición de Éxito

El actor ingreso las distintas promociones para los proyectos.

4.2.13.1.7 Condición de Fallo

El actor no ingresa ninguna promoción.

4.2.14 Políticas de Venta [ECU-0014]

4.2.14.1 Nombre

Políticas de Venta dentro del Sistema de Comercialización de Bienes Inmuebles.

4.2.14.1.1 Descripción

Describe el proceso para la administración de las diferentes políticas de venta que deben existir para poder comercializar un determinado inmueble dentro del sistema.

4.2.14.1.2 Meta

Que el actor pueda ingresar las políticas de venta dentro del sistema.

4.2.14.1.3 Actores

Jefe de Tecnologías

4.2.14.1.4 Flujo de Eventos

4.2.14.1.4.1 Flujo Básico

1. El sistema despliega un formulario.
2. El actor ingresa las políticas de venta existentes.
3. El sistema almacena la información ingresada.

4.2.14.1.4.2 Flujo Alternativo

1. El actor no ingresa ninguna política de venta.

4.2.14.1.5 Precondiciones

Que en el sistema ya estén creadas las principales políticas de venta para poder comercializar un bien inmueble.

4.2.14.1.6 Condición de Éxito

El actor ingreso las políticas de venta para la comercialización de un inmueble determinado.

4.2.14.1.7 Condición de Fallo

El actor no ingreso ninguna política.

4.2.15 *Iniciativas [ECU-0015]*

4.2.15.1 Nombre

Iniciativas

4.2.15.1.1 Descripción

Describe el proceso para la administración de prospectos dentro del sistema.

4.2.15.1.2 Meta

Que el actor pueda administrar provincias con sus respectivos cantones, ciudades y parroquias dentro del sistema.

4.2.15.1.3 Actores

Jefe de Tecnologías

4.2.15.1.4 Flujo de Eventos

4.2.15.1.4.1 Flujo Básico

5 El sistema despliega un formulario.

6 El actor escoge una provincia con su respectivo cantón, ciudad y parroquia para un determinado proyecto.

7 El sistema asigna la información anterior a un proyecto específico.

4.2.15.1.4.2 *Flujo Alternativo*

8 El actor no asigna ninguna provincia y por ende ninguna de las subsiguientes.

4.2.15.1.5 Precondiciones

Que en el sistema ya estén creados proyectos, para la respectiva asignación

4.2.15.1.6 Condición de Éxito

El actor asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.15.1.7 Condición de Fallo

El actor no asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.16 Oportunidades [ECU-0016]

4.2.16.1 Nombre

Oportunidades

4.2.16.1.1 Descripción

Describe el proceso para la administración de provincias, cantones, ciudades y parroquias dentro del sistema.

4.2.16.1.2 Meta

Que el actor pueda administrar provincias con sus respectivos cantones, ciudades y parroquias dentro del sistema.

4.2.16.1.3 Actores

Jefe de Tecnologías

4.2.16.1.4 Flujo de Eventos

4.2.16.1.4.1 *Flujo Básico*

1. El sistema despliega un formulario.
2. El actor escoge una provincia con su respectivo cantón, ciudad y parroquia para un determinado proyecto.
3. El sistema asigna la información anterior a un proyecto específico.

4.2.16.1.4.2 *Flujo Alternativo*

1. El actor no asigna ninguna provincia y por ende ninguna de las subsiguientes.

4.2.16.1.5 Precondiciones

Que en el sistema ya estén creados proyectos, para la respectiva asignación

4.2.16.1.6 Condición de Éxito

El actor asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.16.1.7 Condición de Fallo

El actor no asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.17 Pedidos de Venta [ECU-0017]

4.2.17.1 Nombre

Pedidos de Venta

4.2.17.1.1 Descripción

Describe el proceso para la administración de provincias, cantones, ciudades y parroquias dentro del sistema.

4.2.17.1.2 Meta

Que el actor pueda administrar provincias con sus respectivos cantones, ciudades y parroquias dentro del sistema.

4.2.17.1.3 Actores

Jefe de Tecnologías

4.2.17.1.4 Flujo de Eventos

4.2.17.1.4.1 *Flujo Básico*

1. El sistema despliega un formulario.
2. El actor escoge una provincia con su respectivo cantón, ciudad y parroquia para un determinado proyecto.
3. El sistema asigna la información anterior a un proyecto específico.

4.2.17.1.4.2 *Flujo Alternativo*

1. El actor no asigna ninguna provincia y por ende ninguna de las subsiguientes.

4.2.17.1.5 Precondiciones

Que en el sistema ya estén creados proyectos, para la respectiva asignación

4.2.17.1.6 Condición de Éxito

El actor asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.2.17.1.7 Condición de Fallo

El actor no asignó una provincia, cantón, ciudad y parroquia a un proyecto específico.

4.3 Modelo Conceptual de Datos

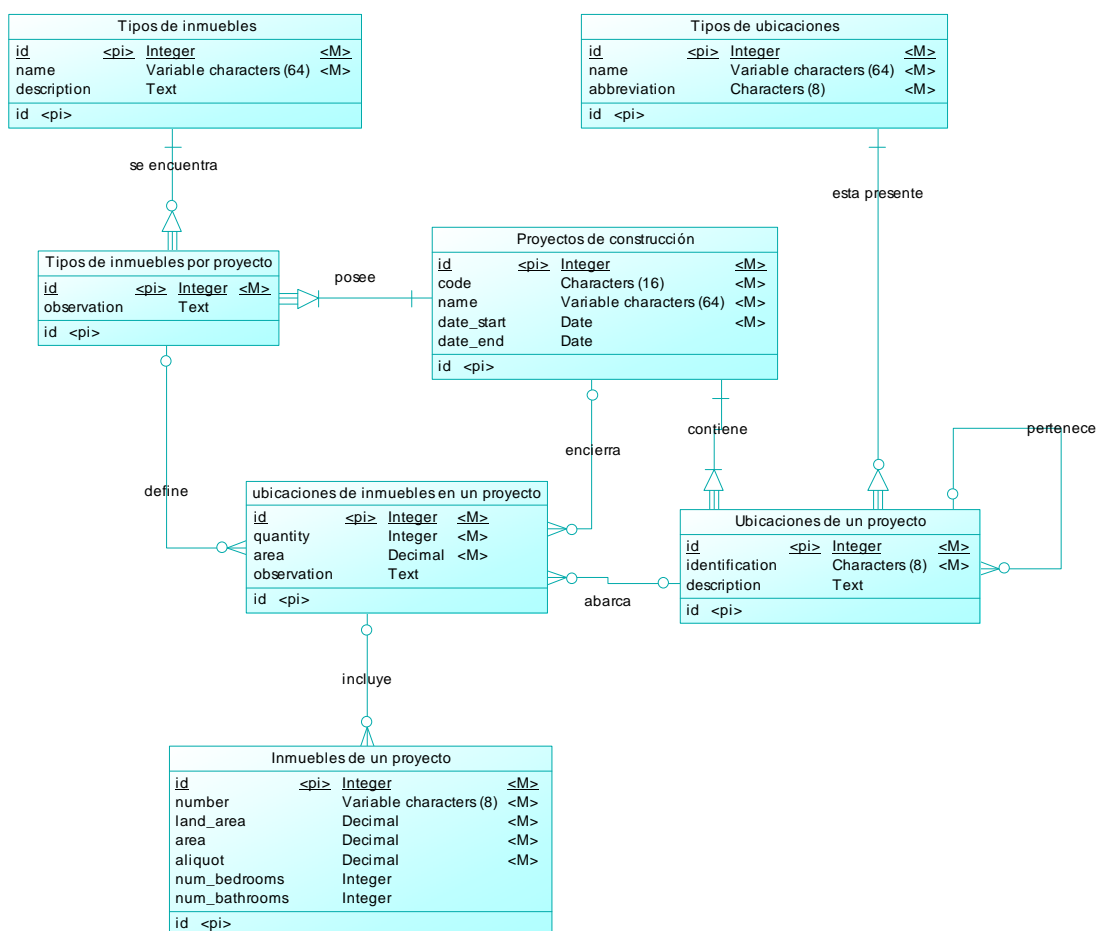


Figura 38: Módulo de Construcciones

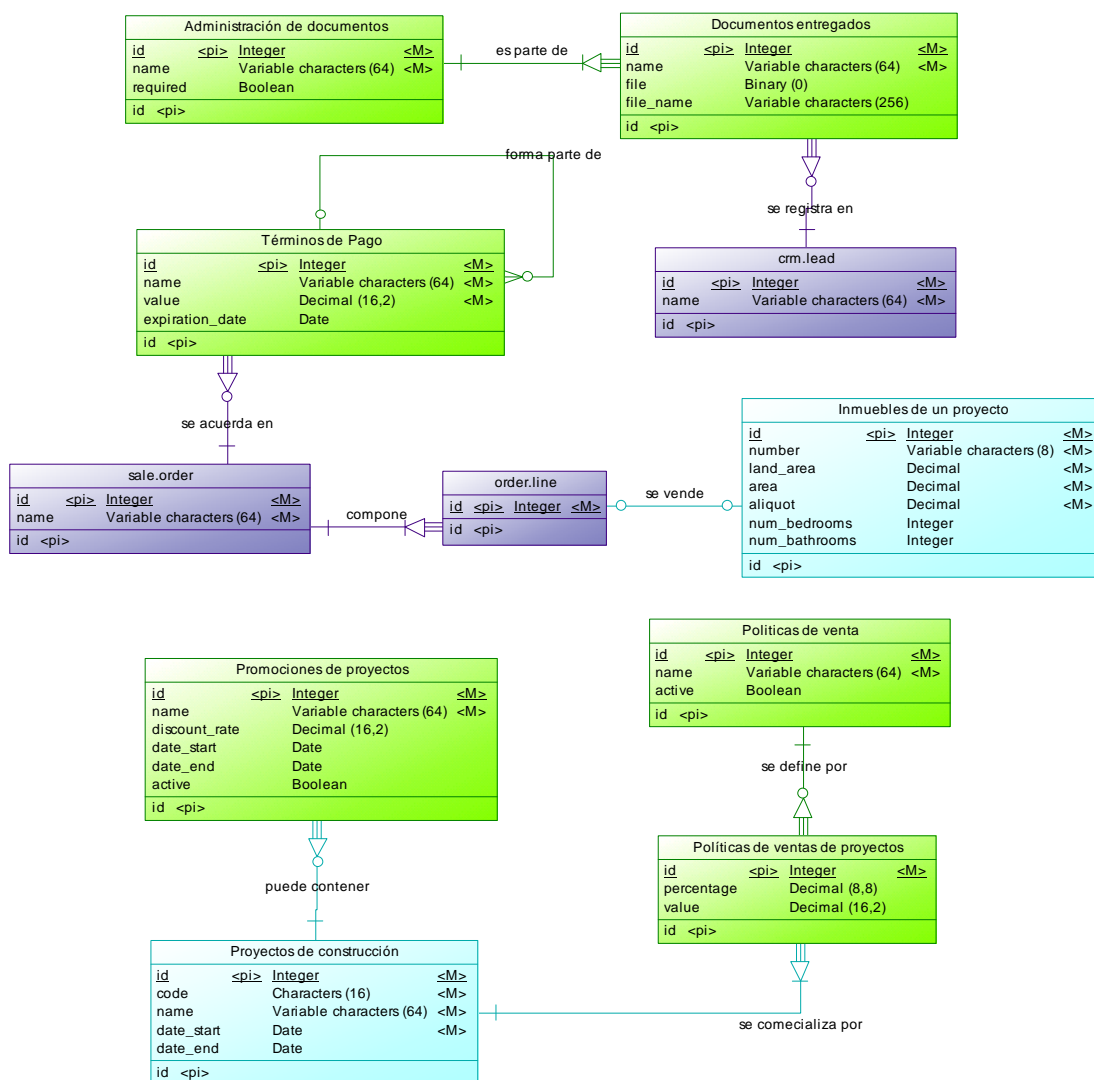


Figura 39: Módulo de Comercialización

4.4 Modelo Lógico de Datos

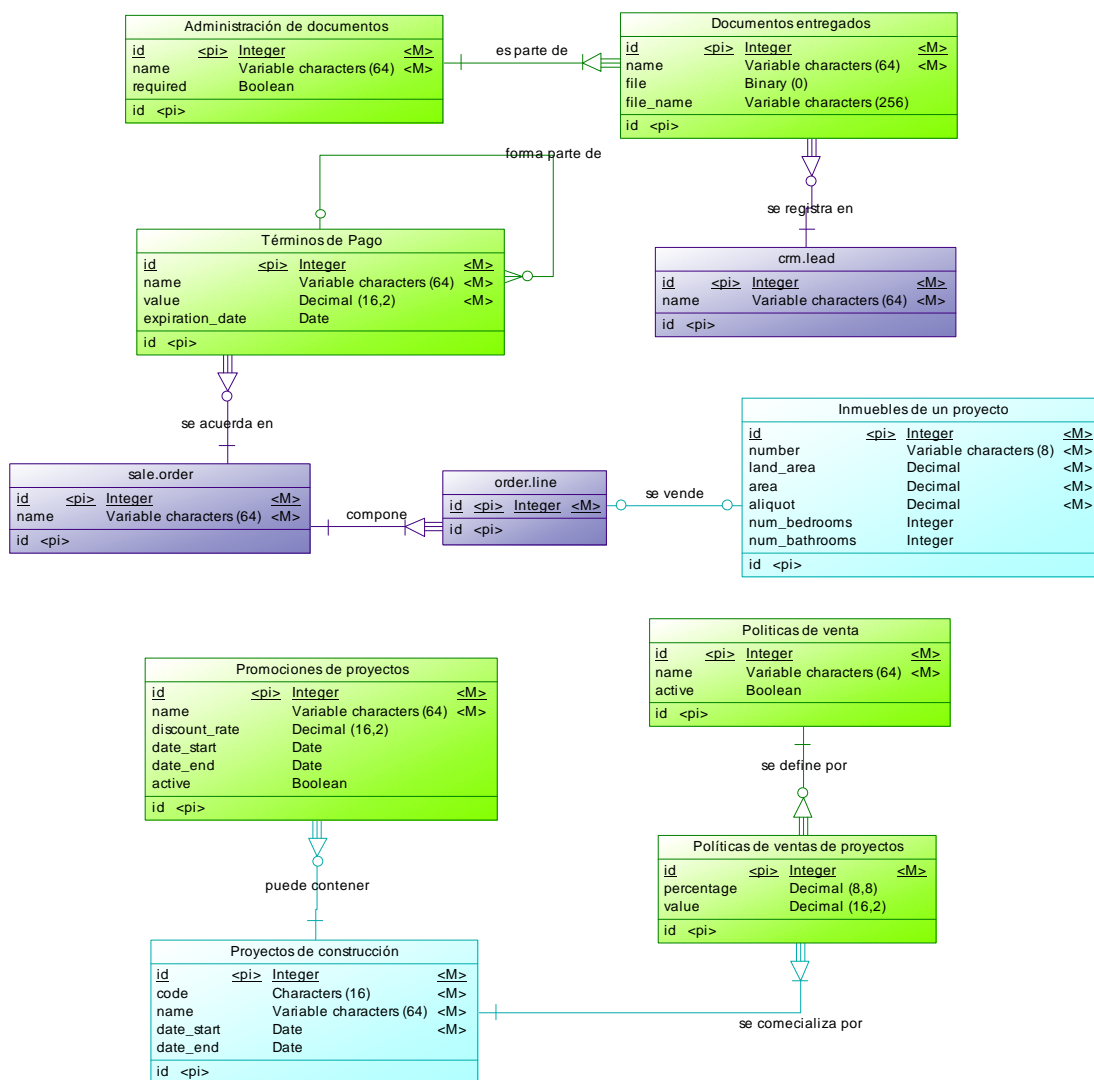


Figura 40: Módulo de Construcciones

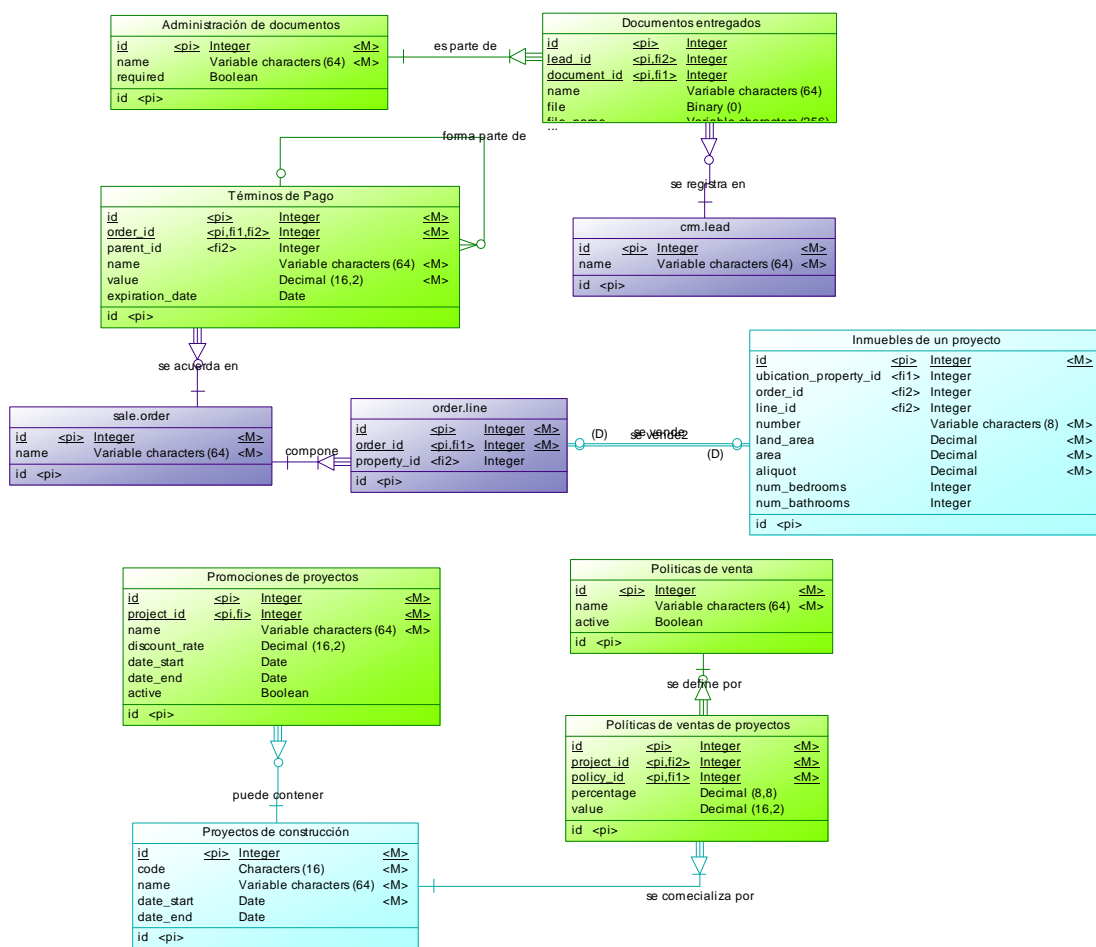


Figura 41: Módulo de Comercialización

4.5 Modelo Físico de Datos

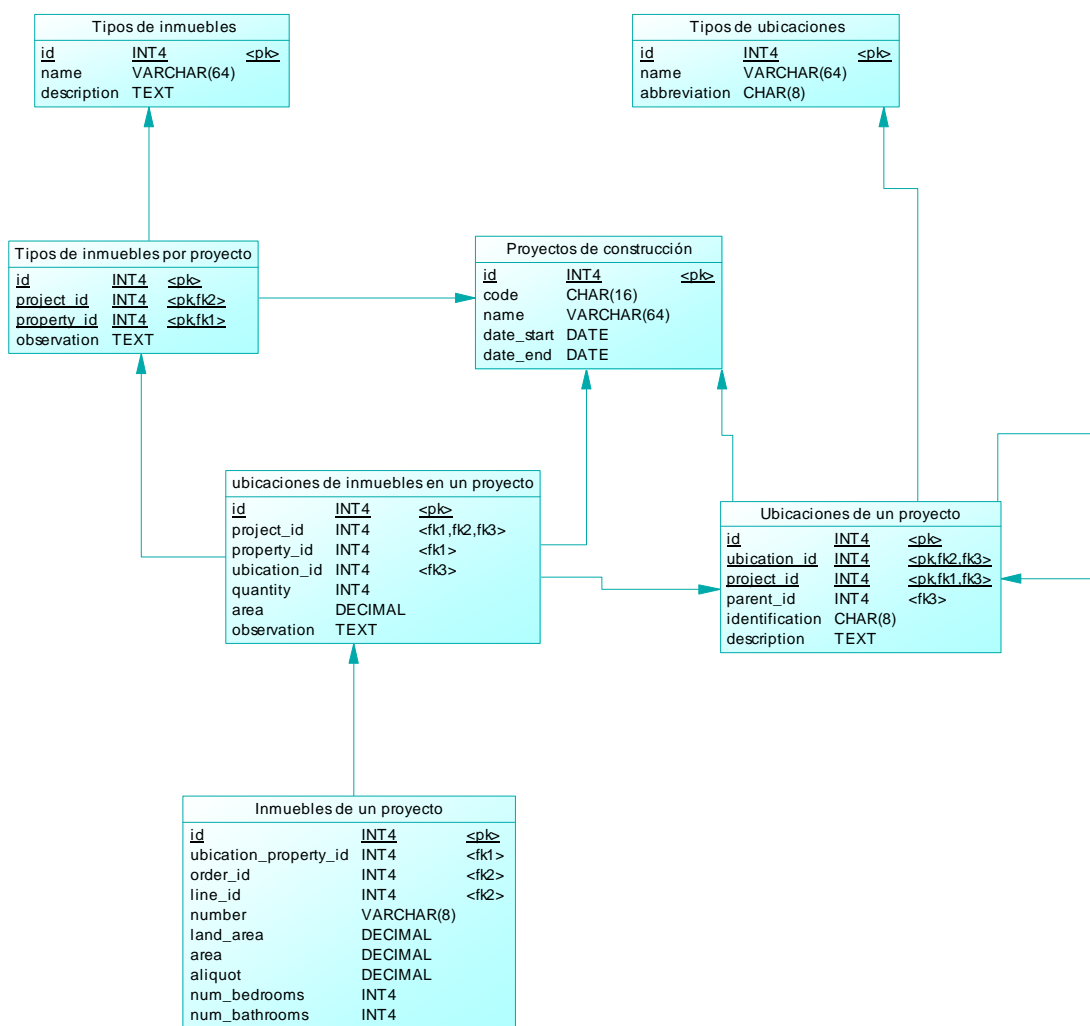


Figura 42: Módulo de Construcciones

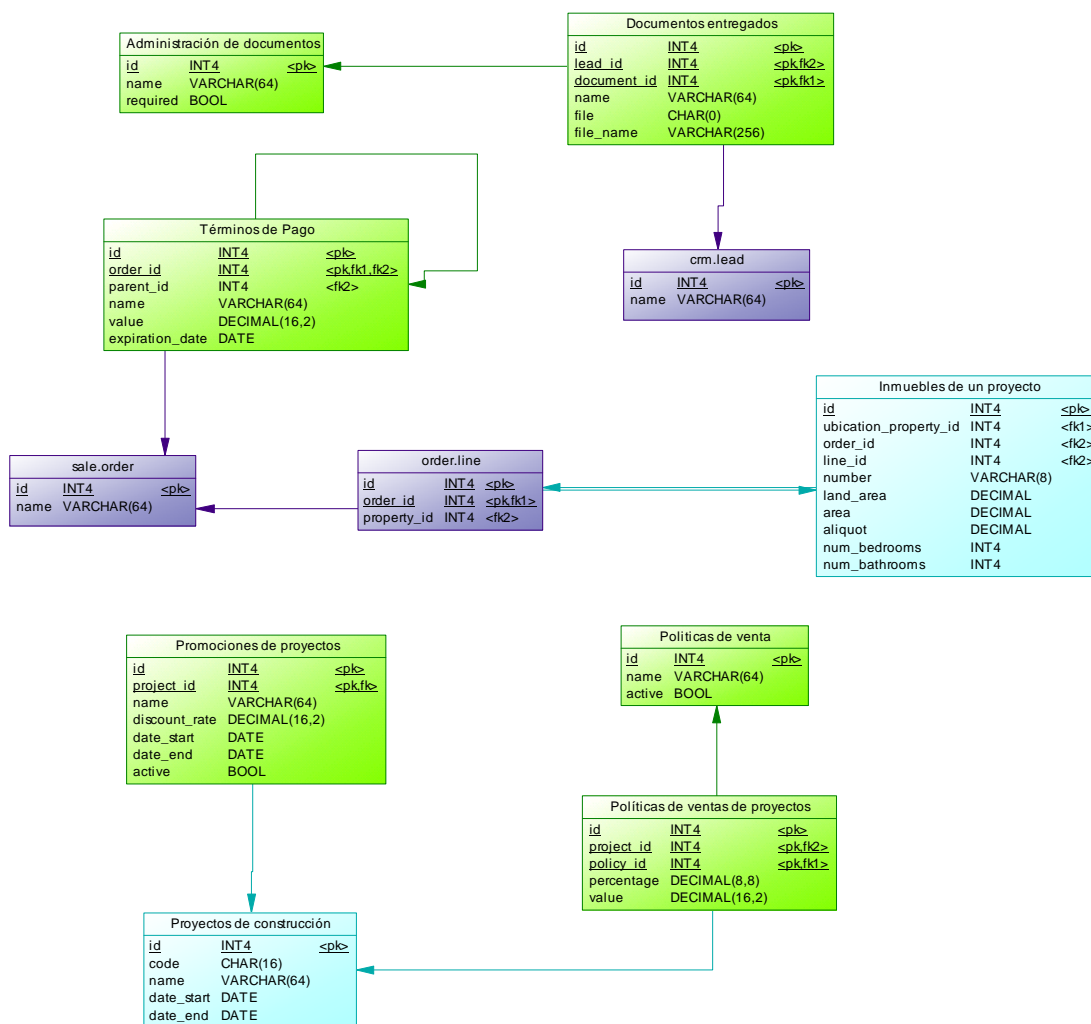


Figura 43: Módulo de Comercialización

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

1.1 Conclusiones

- Con el desarrollo del ERP en las inmobiliarias en el Ecuador, se logró optimizar los recursos para el proceso de construcción y comercialización de un bien inmueble y mejorar su rendimiento y reducción de costos. El tiempo de respuesta es favorable y oscila entre los 3-5 segundos por petición. El software es escable y cuenta con mantenimiento en caso de presentarse errores por parte de los desarrolladores.

- Al usar la plataforma OpenObject para el desarrollo de sistemas, facilitó el proceso de programación porque se logró que el manejo de la base de datos sea transparente para los desarrolladores gracias a las funciones ORM definidas por la plataforma.

- OpenObject al ser una plataforma de desarrollo, utiliza sus propias definiciones de campos para mostrar al usuario final cada campo dependiendo de su tipo, si se necesita cambiar la forma de visualizar la misma se debe modificar el núcleo del cliente Web.

Al ser una plataforma, OpenObject la definición de los campos posee su propio widget dependiendo del campo, es decir ya definidos por la propia plataforma, lo que involucra que si el usuario desea modificar la forma como se presentan los mismos hay que modificar el core de OpenObject lo cual no está permitido.

- Al utilizar módulos propios del OpenERP, se logró adquirir la ventaja de tener funciones adicionales para el desarrollo del sistema que los usuarios finales supieron aprovechar.

- El uso del lenguaje de programación Python facilitó el desarrollo del sistema, porque es un lenguaje más legible al momento de programar y cuenta con indexado obligatorio. A diferencia de otros lenguajes las sentencias no terminan con un punto y coma; tampoco necesita compilación, en el caso de existir errores se dan en tiempo real.

- Al utilizar el estándar IEEE 830 para el levantamiento de los requerimientos del software, se obtuvo el resultado deseado por parte del cliente y se logró obtener el software tal como se lo esperaba para

poder construir y comercializar un bien inmueble visto desde la parte funcional.

1.2 Recomendaciones

- Utilizar la norma IEEE 830 para el levantamiento de los requerimientos del software, porque facilitará el correcto análisis del sistema, de esta forma se tiene una base estable y concreta de la forma en cómo quieren que se presente el software en su parte funcional por parte del cliente antes de poner en marcha cualquier código fuente.

- Investigar el uso de la plataforma OpenObject para el desarrollo de sistemas, porque mejora los tiempos de programación pues se implementa únicamente la lógica del negocio, además que permite la adaptación de sus módulos de manera rápida y segura..

- Se recomienda que los datos del sistema solo se manejen dentro de la intranet de la empresa, tal como se está manejando actualmente, para protección e integridad de los datos.

- Investigar los módulos propios de OpenERP en su parte funcional, actualmente cuenta con 500 módulos, que cuenta con los requisitos no funcionales como son rendimiento, flexibilidad, seguridad, mantenibilidad, portabilidad, gratuito, multiplataforma, código abierto, entre otras bondades, entre los principales módulos están contabilidad, ventas, inventarios CRM, Recursos Humanos, los mismos que se pueden adecuar a módulos propios desarrollados desde cero, dependiendo de las necesidades de cada empresa.

BIBLIOGRAFÍA

Beck, K. (1999). Extreme Programming Explained. Embrace Change.

Booch, G., Rumbaugh, J., & Jacobson, I. (s.f.). El lenguaje unificado de modelado. Addison Wesley.

González Duque, R. (s.f.). Python para todos.

Joseph, S. (s.f.). Aprendiendo UML en 24 horas. Prentice Hall.

Martelli, A. (2008). Python Guía de Referencia. Anaya Multimedia Oreilly.

Martínez Guzmán, J. P. (18 de Abril de 2011). La verdadera historia de la burbuja inmobiliaria ecuatoriana. Obtenido de Realidad Ecuador.

Pinckaers, F., Gardiner, G., & Van Vossel, E. (2011). Open ERP Version 6: a modern approach to integrated business management.

Robles Prado, T. J. (2002). Introducción a Python y PostgreSQL.

Tiny SPRL. (2009). Open Object Developer Book.

Van Rossum, G. (2005). Guía de Aprendizaje de Python. Python Software Foundation.

NETGRAFÍA

Fundación Wikimedia, I. (22 de Septiembre de 2012). Wikipedia. Recuperado el 11 de Octubre de 2012, de Wikipedia: <http://es.wikipedia.org/wiki/Bitbucket>

Fundación Wikimedia, Inc. (22 de 03 de 2013). Wikipedia. Recuperado el 02 de 04 de 2013, de Mercurial: <http://es.wikipedia.org/wiki/Mercurial>

OpenERP. (2011). OpenERP. Retrieved 01 23, 2013, from Installation: <https://doc.openerp.com/6.0/es/install/#installation-link>

OpenERP. (2011). OpenERP. Retrieved 02 10, 2013, from OpenERP Book: <https://doc.openerp.com/book/#books-link>