UNIVERSIDAD DE ORIENTE

NÚCLEO DE ANZOÁTEGUI

ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS

DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**PROFESOR(A): INTEGRANTES:**

ING. MUJICA VICTOR C.I. 20 503 457 LÓPEZ ALFONSO

C.I. 19 248 706 RODRÍGUEZ GABRIEL

C.I. 18 401 499 TINEO FANNY

C.I. 21 173 595 MOBILIO GABRIEL

C.I. 19 316 427 PERDOMO HELYSON

C.I. 19 508 203 ROJAS EYLÍN

C.I. 20 448 909 FELICCE RICARDO

C.I. 19 940 338 DELGADO JAVIER

C.I. 19 169 037DÍAZ MARIÁNGELA

C.I. 19 316 936 LEZAMA KATHERINE

BARCELONA, 8 DE JULIO DE 2014

RESUMEN

ÍNDICE

CONTENIDO

|  |  |
| --- | --- |
| **RESUMEN**  **ÍNDICE**  **CAPITULO I. PLANTEAMIENTO DEL PROBLEMA**   * 1. Introducción      1. Problema      2. Propósito      3. Importancia      4. Alcance      5. Justificación      6. Originalidad   2. Objetivos      1. Objetivo general      2. Objetivo especifico   **CAPÍTULO II FASE DE INICIO**  2.1 Análisis de los requerimientos  2.2 Requerimientos del sistema  2.2.1 Requisitos de software.  2.2.2 Requisitos de hardware  2.2.3 Requisitos funcionales del sistema.  2.2.4 Requisitos no funcionales del sistema  2.3 Modelo de dominio del sistema.  2.3.1 Modelo de dominio del sistema general.  2.3.2 Modelo de dominio página principal  2.3.3 Modelo de dominio Edificio.  2.3.4 Modelo de dominio Apartamento.  2.3.5 Modelo de dominio Usuario  2.3.6 Modelo de dominio cartelera  2.3.7 Modelo de dominio mensajería.  2.3.8 Modelo de dominio servicios.  2.3.9 Modelo de dominio gastos.  2.3.10 Modelo de dominio pagos.  2.4 Diagrama de casos de uso  2.4.1 Definición de actores utilizados en el sistema  2.4.2 Diagrama de casos de uso general.  2.4.2.1 Descripción detallada del caso de uso general.  2.4.3 Diagrama de casos de uso página principal  2.4.3.1 Descripción detallada del caso de uso página principal.  2.4.4 Diagrama de casos de uso Edificio.  2.4.5 Diagrama de casos de uso Apartamento.  2.4.6 Diagrama de casos de uso Usuario  2.4.7 Diagrama de casos de uso cartelera  2.4.7.1 Descripción detallada del caso de uso módulo cartelera.  2.4.7.2 Descripción del Flujo de Sucesos del Caso de Uso “Publicar Anuncios”.  2.4.7.3 Descripción del Flujo de Sucesos del Caso de Uso “Eliminar anuncios”.  2.4.7.4 Descripción del Flujo de Sucesos del Caso de Uso “Ver publicaciones”.  2.4.7.5 Descripción del Flujo de Sucesos del Caso de Uso “Responder publicaciones”.  2.5 Diagrama de comunicación.  2.5.1 Diagrama de comunicación del sistema general.  2.5.2 Diagrama de comunicación página principal  2.5.3 Diagrama de comunicación Edificio.  2.5.4 Diagrama de comunicación Apartamento.  2.5.5 Diagrama de comunicación Usuario  2.5.6 Diagrama de comunicación cartelera  2.5.6.1 Descripción del proceso de comunicación “Publicar Anuncios”.  2.5.6.2 Descripción del proceso de comunicación “Ver publicaciones”.  2.5.6.3 Descripción del proceso de comunicación “Eliminar anuncios”  2.5.6.4 Descripción del proceso de comunicación “Responder publicaciones”  2.5.7 Diagrama de comunicación mensajería.  2.5.8 Diagrama de comunicación servicios.  2.5.9. Diagrama de comunicación gastos.  2.5.10 Diagrama de comunicación pagos.  2.6 Diagrama de clases  2.6.1 Diagrama de clases general.  2.6.2 Diagrama de clases página principal  2.6.3 Diagrama de clases Edificio.  2.6.4 Diagrama de clases Apartamento.  2.6.5 Diagrama de clases Usuario  2.6.6 Diagrama de clases cartelera  2.6.7 Diagrama de comunicación mensajería.  2.6.8 Diagrama de comunicación servicios.  2.6.9. Diagrama de comunicación gastos.  2.6.10 Diagrama de comunicación pagos.  2.7 Diagrama de entidad-relación.  2.7.1 Diagrama de entidad relación general.  2.7.2 Diagrama de entidad relación página principal  2.7.3 Diagrama de entidad relación Edificio.  2.7.4 Diagrama de entidad relación Apartamento.  2.7.5 Diagrama de entidad relación Usuario  2.7.6 Diagrama de entidad relación Cartelera  2.7.7 Diagrama de entidad relación mensajería.  2.7.8 Diagrama de entidad relación servicios.  2.7.9. Diagrama de entidad relación gastos.  2.7.10 Diagrama de entidad relación pagos.  **CAPITULO III. DISEÑO DEL SISTEMA**  3.1 Diseño de la base de datos  3.1.2 Diseño del modelo conceptual de la base de datos  3.1.3 Estructuración de los datos  3.1.3.1 Entidad “apartamento”.  3.1.3.2 Entidad “apartamentotipo”.  3.1.3.3 Entidad “apartamento\_usuario”.  3.1.3.4 Entidad “edificio”.  3.1.3.5 Entidad “paginaprincipal”.  3.1.3.6 Entidad “pagos”.  3.1.3.7 Entidad “pagos\_bancos”.  3.1.3.8 Entidad “pagos\_estados”.  3.1.3.9 Entidad “pagos\_tipos”.  3.1.3.10 Entidad “post”.  3.1.3.11 Entidad “publicacion”.  3.1.3.12 Entidad “servicios”.  3.1.3.13 Entidad “servicioscostos”.  3.1.3.14 Entidad “serviciospagos”.  3.1.3.15 Entidad “solicituddeservicios”.  3.1.3.16 Entidad “telefono”.  3.1.3.17 Entidad “tipodeservicios”.  3.1.3.18 Entidad “tipopagodeservicios”.  3.1.3.19 Entidad “usuario”.  3.2 Interfaz del sistema.  3.2.1 Interfaz general  3.2.2 Interfaz página principal  3.2.3 Interfaz Edificio.  3.2.4 Interfaz Apartamento.  3.2.5 Interfaz Usuario  3.2.6 Interfaz Cartelera  3.2.6.1 Vista del Administrador  3.2.6.1.1Descripción detallada Interfaz cartelera vista administrador  3.2.6.1.2 Descripción detallada Interfaz cartelera “publicar un nuevo anuncio”  3.2.6.1.3 Descripción detallada Interfaz cartelera “restringir publicación”  3.2.6.1.4 Descripción detallada Interfaz cartelera “publicar un nuevo anuncio”  3.2.6.1.4 Descripción detallada Interfaz cartelera “restringir publicación por pisos”  3.2.6.1.5 Descripción detallada Interfaz cartelera “Restringir publicación por apartamentos”  3.2.6.1.6 Descripción detallada Interfaz cartelera “Restringir publicación por apartamentos según el piso”  3.2.6.2 Vista del Usuario  3.2.7 Interfaz mensajería.  3.2.8 Interfaz servicios.  3.2.9. Interfaz gastos.  3.2.10 Interfaz pagos.  3.3 Pruebas de integración.  3.3.1 Prueba de caja blanca.  3.3.2 Prueba de caja negra. | 2  3  4 |

**CAPITULO I. PLANTEAMIENTO DEL PROBLEMA**

1. **Introducción**

Desde sus inicios el ser humano ha sido nómada, buscando siempre las mejores condiciones para sí, de esta manera estaba constantemente en busca de un lugar capaz de proveerle agua, refugio y comida, a medida que evolucionaba como especie, y sus habilidades iban creciendo, ya no solo la recolección era una forma de conseguir alimento sino también la siembra, además de la acostumbrada caza y pesca (en las zonas aptas para esto).

Otra característica que acompaña al humano desde sus principios es que, como muchos otros mamíferos, vive en comunidades, lo que le ha permitido enfrentarse a las adversidades naturales que se le presentaron desde el momento de su aparición en la tierra y a lo largo de todo su proceso evolutivo hasta llegar al día de hoy.

Hoy en día aun cuando varios miles de años nos separan de nuestros primeros ancestros compartimos varias características con ellos, seguimos ese proceso migratorio en busca de mejores oportunidades, familiares, económicas, académicas, y hasta por simple comodidad, y por supuesto seguimos viviendo en comunidades, las cuales van creciendo generalmente en torno al principal motor de hoy en día, la industria y la economía que genera. Esto deriva en ciudades con una demanda de vivienda cada vez mayor, la cual es resuelta mediante la construcción de conjuntos residenciales y edificios que le ofrecen a los que allí viven una serie de facilidades y comodidades que hacen de su día a día un poco más sencillo y agradable.

Pero detrás de estos servicios y comodidades que brinda vivir en un conjunto residencial existen personas trabajando para hacerlo posible, y es donde surgen ciertas complicaciones, transmitir la información a todas las personas, administrar pagos, llevar control de deudas, de pagos parciales, un control de todos los servicios que ofrece el conjunto, etc, esto a veces llevado por una dos personas, para múltiples edificios. Es por esto que nace la necesidad de crear un software que permita facilitar el trabajo de estas personas, así como las vías de comunicación entre los administradores e inquilinos.

Además de facilitar el trabajo de los administradores se desea que este software permita a los propietarios un acceso total a sus cuentas y a la información desde la comodidad de su casa, en el siguiente proyecto nos aventuraremos a crear un programa capaz de brindar estas comodidades para simplificar la vida de usuarios y administradores de los diferentes condominios.

* + 1. **Problema**

Existe la necesidad de controlar y tener un mejor control en un conjunto residencial, de tener un registro automatizado de los dueños de cada apartamento de un edificio, de llevar un control sobre los gastos y pagos que ha hecho el dueño de dicho apartamento, de controlar y permisar los servicios adquiridos o solicitados por el usuario, permitir la comunicación de usuario-administrador de mantener comunicado a todos inquilinos de un terminado piso u edificio o todo el conjunto residencial.

Es complejo mantener y satisfacer todos los requisitos y necesidades que demanda gerenciar un condominio, ya que todo está controlado bajo sistemas manuales lo que agrava el problema si se trata de un conjunto residencial grande.

* + 1. **Propósito**

El fin de este trabajo es el desarrollo del Sistema que automatice el proceso administrativo de cualquier condominio de edificios que decida aplicarla en un entorno Web utilizando tecnologías de Base de Datos, lenguajes de cuarta generación y una arquitectura Cliente/Servidor que funcione sobre la plataforma en internet. Este trabajo se realizará aplicando la disciplina de Ingeniería de Software, la cual se puede obtener mediante la combinación de métodos completos para todas las fases de desarrollo de software.

En el sistema a desarrollar se utilizará el Lenguaje Unificado para Modelado UML, ya que este lenguaje define una notación y un proceso para construir Sistemas de Software complejos y ofrece un rico conjunto de métodos lógicos y físicos con los cuales se puede razonar sobre diferentes aspectos del sistema.

* + 1. **Importancia**
    2. **Alcance**

El Sistema automatizará todos los procesos referentes a un condominio que desee aplicarlo, produciendo un mayor rendimiento tiempo/trabajo.

Con la realización de este proyecto se elaboró una herramienta de gestión para automatizar las tareas y procesos que se realizan en un condominio.

* + 1. **Originalidad**
    2. **Justificación**

**1.2 Objetivos**

**1.2.1 Objetivos General**

* Desarrollar una página web que automatice el proceso administrativo de cualquier condominio de edificios que decida aplicarla.

**1.2.2 Objetivos Específicos:**

* Determinar las problemáticas comunes de los condominios de edificios
* Identificar las herramientas principales para la simplificación del proceso administrativo
* Identificar las herramientas principales para brindar a los propietarios información sobre el condominio así como acceso inmediato a sus estados de pago, fechas de cobro, montos, saldo promedio, etc.
* Modelar la estructura del nuevo sistema y su base de datos.
* Codificar los distintos módulos e interfaces.

**CAPÍTULO II FASE DE INICIO**

**2.1 Análisis de los requerimientos**

En la etapa de Análisis de los Requerimientos se realizó el análisis del sistema, en el cual se determinaron los principales requisitos necesarios para su buen funcionamiento. La finalidad del análisis de los requerimientos es el de conocer la necesidades de los usuarios para así realizar el sistema y poder satisfacer los requerimientos de los usuarios, y así mostrar cada una de las funciones que realiza el sistema.

La manera en la que se llevan a cabo los procesos en los sistemas de condominio de un edificio y los problemas encontrados en cuanto al administración de los mismos, hace necesario crear una herramienta automatizada que proporcione grandes beneficios, en cuanto a velocidad de transmisión de información, pagos, actualizaciones, prestamos de servicios, entre otras actividades. Por lo tanto, se recurrió al diseño de una página web que automatice el proceso administrativo de cualquier condominio de edificios que decida aplicarla, permitiendo así, optimizar las operaciones y el incremento de la productividad.

Para lograr este propósito se utilizaron las herramientas del UML; como son: el diagrama de casos de usos el cual permitirá mostrar las funciones que se esperan del sistema, identificar los actores, diseñar los principales procesos involucrados y describir cómo funciona cada uno de los procesos. El diagrama de comunicación, para tener en cuenta la forma en que se procesa cada requerimiento o caso de uso realizado por un acto. Se utilizará el diagrama de clase de análisis el cual permitirá identificar las principales clases de análisis que intervienen en la realización de cada caso de uso y la interacción que existe entre los distintos objetos de análisis, entre otros diagramas que se documentaran y especificaran según corresponda.

**2.2 Requerimientos del sistema**

1. La interfaz de usuario del sistema debe ser fácil y amigable.
2. La base de datos debe estar ubicada en un servidor en el cual se puedan acceder los datos a través de la red de área local.
3. Mostrar los datos asociados a la administración de los que interactúan con el servicio en Tiempo real.
4. Permitir ingresar edificios y apartamentos, para la adaptación de cualquier condominio.
5. Contar con un sistema de seguridad que le permita al usuario acceder al sistema solo a los usuarios autorizados, esto va permitir disminuir el riesgo de que entre cualquier usuario al sistema.
6. Automatizar el proceso de control de pagos.
7. Gestionar la publicación de anuncios de información para apartamentos en específicos según la necesidad del administrador.
8. Envió de mensajes y recepción de manera óptima
9. Automatizar procesos de alquiler de servicios.

### **2.2.1 Requisitos de software.**

### **2.2.2 Requisitos de hardware**

### **2.2.3 Requisitos funcionales del sistema**

* El sistema debe permitir el acceso sólo a los usuarios pertenecientes a la junta de condominio e indicar el tipo de usuario.
* El Sistema permite enviar mensajes a todos los apartamentos, pisos o edificios el administrador del sistema y para un usuario permite solo enviar mensajes al administrador.
* El administrador solo puede crear, modificar o eliminar anuncios en la cartelera que serán visibles por usuarios determinados.
* El administrador podrá aprobar o rechazar los pagos consignados por los usuarios
* Solo podrá agregar, modificar, eliminar usuarios en el sistema, la persona previamente autorizada por el administrador del mismo.
* Solo puede crear, modificar, eliminar secciones, modificar el footer cambiar fondo de imagen de la página principal el administrador.
* El sistema debe permitir agregar edificios, modificar y eliminar.
* El sistema permite asignar dueños a los apartamentos.

**2.2.4 Requisitos no funcionales del sistema**

* El sistema debe estar constituido y estructurado como una aplicación WEB que podrá ser usada a nivel externo (Internet).
* La estructura y diseño del sistema es escalable lo que significa que debe adaptarse fácilmente a cualquier cambio o mejora.
* El Sistema debe poseer una interfaz amigable de fácil acceso y manejo
* El Sistema debe poseer un manual de usuario para la instalación y el correcto uso del sistema.

**2.3 Modelo de dominio del sistema**

Este modelo se encarga de capturar los objetos más importantes vinculados al contexto del sistema. Estos objetos representan los eventos que tienen lugar en el entorno donde se desenvuelve el sistema.

Un modelo de dominio es una representación visual de las entidades del mundo real en un área de interés. También se les denomina modelos conceptuales, modelos delo de objetos del dominio y modelos de objetos de análisis. Para la descripción delo de dominio deben realizarse diagramas de clases basados en el Lenguaje de modelado UML. En la figura 1 se muestra el

**2.3.1 Modelo de dominio del sistema general.**

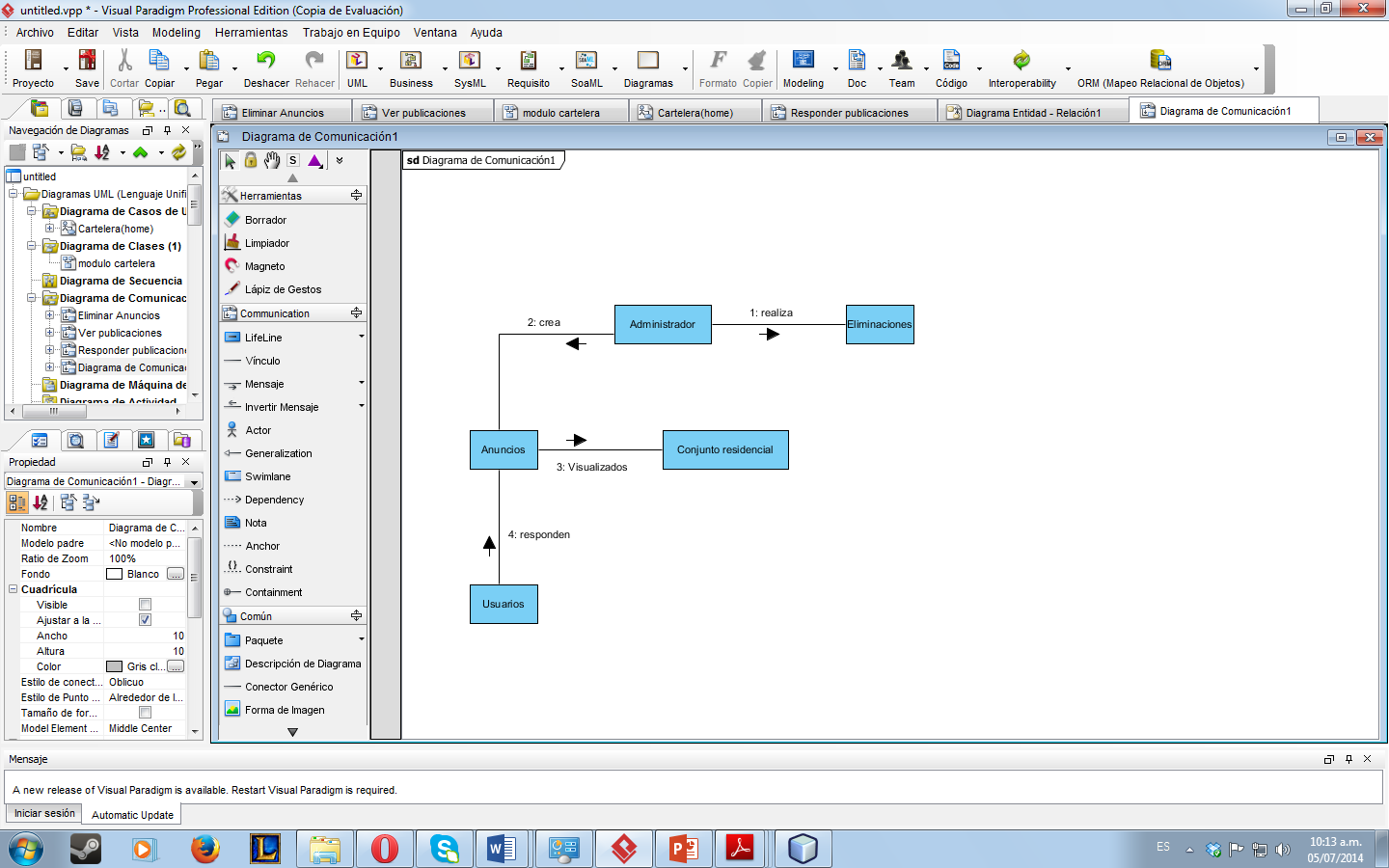
**2.3.2 Modelo de dominio página principal**

**2.3.3 Modelo de dominio Edificio.**

**2.3.4 Modelo de dominio Apartamento.**

**2.3.5 Modelo de dominio Usuario**

**2.3.6 Modelo de dominio cartelera**



1 0….”

1

0….”

1

1

0…” 1

Figura modelo de dominio cartelera

|  |
| --- |
| Glosario de términos modelo de dominio cartelera |
| Administrador: Las personas que poseen este privilegio el sistema les permite crear publicaciones, mediante el ingreso de la información correspondiente, referida al título, contenido y la privacidad de la publicación, además de la posibilidad de hacer eliminaciones. |
| Anuncios: Todo lo concerniente a la publicación en general, titulo, contenido, nombre del administrado que realizó la publicación y fecha. |
| Usuario: Los inquilinos de la junta de condominio, que no son administradores del sistema, los cuales pueden respondes los anuncios realizados por los administradores. |
| Conjunto residencial: lo conforman todos los inquilinos logueados en el sistema, así como los administradores del mismo, los cuales pueden visualizar cualquier anuncio del sistema. |
| Eliminaciones: Se pueden borrar los registros de anuncios realizados por los administradores. |

**2.3.7 Modelo de dominio mensajería.**

**2.3.8 Modelo de dominio servicios.**

**2.3.9 Modelo de dominio gastos.**

**2.3.10 Modelo de dominio pagos.**

**2.4 Diagrama de casos de uso**

Un caso de uso describe el comportamiento del sistema en diferentes condiciones mientras este responde a la petición de uno de sus usuarios. Está formado principalmente por actores y el caso de uso.

Un actor es algún elemento que se comunica con el sistema o producto y que es externo a esté, estos realizan una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando, que es lo que se conoce como caso de uso.

**2.4.1 Definición de actores utilizados en el sistema**

El proceso que conlleva el desarrollo de cualquier software, es el requerimiento de sistemas, en el cual se describen las funciones que el usuario quiere que el nuevo sistema realice. Para la realización del sistema es necesario saber quiénes van a hacer los actores que van a interactuar con el sistema, los cuales van a permitir conocer las necesidades que ellos poseen y esperan que se vayan a solucionar con el desarrollo del nuevo software.

Esto se va representar utilizando una de las herramientas robustas para el modelado de sistema conocido como UML, para ello se va hacer uso de los diagramas de caso de uso, ya que éstos van a permitir definir cuáles son las limitantes del sistema, definir quiénes serán los usuarios del sistema, las funciones que los usuarios representan, además va a servir para crear los diagramas siguientes, y así tener una visión general del sistema.

**Tabla 1 actores del sistema.**

|  |  |
| --- | --- |
| Actor | Descripción |
| Usuario |  |
| Administrador |  |

**2.4.2 Diagrama de casos de uso general.**

**2.4.2.1 Descripción detallada del caso de uso general.**

**2.4.3 Diagrama de casos de uso página principal**

**2.4.3.1 Descripción detallada del caso de uso página principal.**

**2.4.4 Diagrama de casos de uso Edificio.**

**2.4.5 Diagrama de casos de uso Apartamento.**

**2.4.6 Diagrama de casos de uso Usuario**

**2.4.7 Diagrama de casos de uso cartelera**

Tabla Descripción de casos de uso Módulo Cartelera

|  |  |  |
| --- | --- | --- |
| Caso de uso | Descripción | Actor |
| Publicar Anuncios | Permite crear anuncios, estos pueden configurarse según el tipo de privacidad desea, se debe llenar todos los campos recibidos. Además de publicar anuncios estos pueden ser eliminados si así se requiere. | Administrador |
| Eliminar anuncios | Permite eliminar anuncios. | Administrador |
| Ver publicaciones | Permite ven los mensajes publicados una vez logueado en el sistema, mediante la interfaz que se proporciona. | Usuario, Administrador. |
| Responder Publicaciones | Permite responder los mensajes hechos por el o los administradores del sistema, mediante un enlace de comunicación con el de mensajería. | Usuario |

A continuación en la figura se muestra el caso de uso, del módulo cartelera donde se muestran cada caso de uso los actores y las relaciones que existen entre ellos.

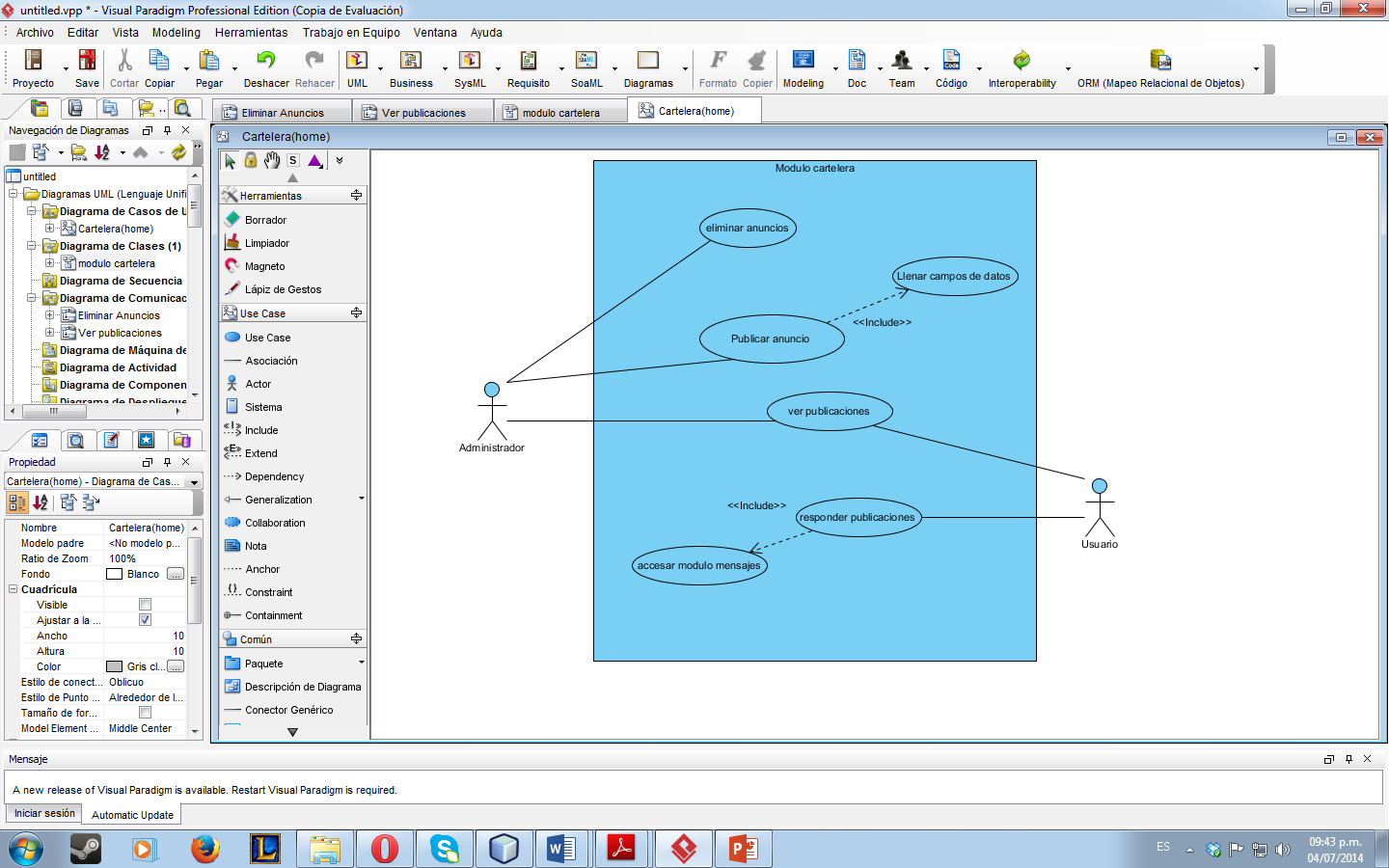


Figura modelo de caso de uso módulo cartelera.

**2.4.7.1 Descripción detallada del caso de uso módulo cartelera.**

En el diagrama del contexto del módulo cartelera (Ver figura ), se puede observar el caso de uso, el cual nos permite tener una visión más clara de las acciones que puede realizar el sistema al interactuar con los actores, es decir, al ser invocado por el actor.

Además se describirán las secuencias de acciones a seguir por el caso de uso al ser invocado por un actor, de tal manera que puedan describir el flujo de suceso desde que se inicia hasta que finaliza el caso de uso.

**2.4.7.2 Descripción del Flujo de Sucesos del Caso de Uso “Publicar Anuncios”.**

|  |
| --- |
| Nombre del caso 1: Publicar Anuncios |
| Actores: Administrador. |
| Descripción: Crea anuncios para que los usuarios correspondientes los puedan leer |
| Pre-condición: El usuario debe estar logueado como administrador, y hacer click en la opción “Publicar anuncios” |
| Flujo de eventos:  Flujo principal: -El sistema muestra las casillas siguiente: Titulo, un campo para colocar el anuncio y un botón restringir publicación.  -Se deben rellenar ambos campos de texto.  -Se debe pulsar el botón restringir publicación.  -El sistema muestra un modal, con distintas pestañas (Por apartamento, por pisos, por edificio), y dos botones publicar y cancelar.  -Se debe indicar la opción correspondiente, según lo deseado por el actor.  -Luego pulsar el botón publicar.  -Se genera un mensaje “La publicación ha sido realizada de manera correcta”  -Finaliza el caso de uso.  Flujo alterno:  -Volver a la interfaz cartelera. |

**2.4.7.3 Descripción del Flujo de Sucesos del Caso de Uso “Eliminar anuncios”.**

|  |
| --- |
| Nombre del caso 1: Eliminar anuncios |
| Actores: Administrador. |
| Descripción: Permite tener la opción de borrar anuncios de la cartelera. |
| Pre-condición: Se debe hacer click en el botón “X”. |
| Flujo de eventos:  -Se presentara un modal con un aviso, y dos botones Aceptar y cancelar.  -Se debe presionar el botón aceptar.  -Se genera el mensaje “La publicación ha sido eliminada exitosamente”.  - Finaliza el caso de uso.  Flujo alterno:  -Volver a la interfaz cartelera. |

**2.4.7.4 Descripción del Flujo de Sucesos del Caso de Uso “Ver publicaciones”.**

|  |
| --- |
| Nombre del caso 1: Ver publicaciones |
| Actores: usuario, administrador. |
| Descripción: Genera los mensajes mediante una interfaz, que permite visualizar los distintos anuncios o publicaciones creadas |
| Pre-condición: Los distintos usuarios deben loguearse en el sistema a través del módulo de usuarios. |
| Flujo de eventos:  -Al ingresar el sistema se muestran los mensajes.  -Al acceder al botón inicio, se accederá a este caso de uso  -Este caso de uso finaliza cuando se accedan a otros módulos u opciones del software. |

**2.4.7.5 Descripción del Flujo de Sucesos del Caso de Uso “Responder publicaciones”.**

|  |
| --- |
| Nombre del caso 1: Responder publicaciones. |
| Actores: Usuarios. |
| Descripción: Permite tener la opción de que los usuarios respondan publicaciones hecha por los administradores. |
| Pre-condición: Se debe hacer click en el icono de mensaje. |
| Flujo de eventos:  -Se accede al módulo mensajería.  -Finaliza caso de uso. |

**2.4.8 Diagrama de casos de uso mensajería.**

**2.4.9 Diagrama de casos de uso servicios.**

**2.4.10 Diagrama de casos de uso gastos.**

**2.4.11 Diagrama de casos de uso pagos.**

**2.5 Diagrama de comunicación.**

En los distintos diagramas correspondientes, se visualizara la forma en que se procesa la información correspondiente para realizar las distintas opciones que poseen los actores en este módulo, es decir, la forma en que se transmite la información según los requerimientos de cada actor.

Además se describirán las secuencias de acciones a seguir según el proceso o solicitud requerida por el actor, desde el inicio de los requerimientos hasta la obtención de los resultados.

**2.5.1 Diagrama de comunicación del sistema general.**

**2.5.2 Diagrama de comunicación página principal**

**2.5.3 Diagrama de comunicación Edificio.**

**2.5.4 Diagrama de comunicación Apartamento.**

**2.5.5 Diagrama de comunicación Usuario**

**2.5.6 Diagrama de comunicación cartelera**

Tabla Descripción de diagrama de comunicación Módulo Cartelera

|  |  |
| --- | --- |
| Diagrama de comunicación | Actor |
| Publicar Anuncios | Administrador |
| Eliminar anuncios | Administrador |
| Ver publicaciones | Usuario, Administrador. |
| Responder Publicaciones | Usuario |

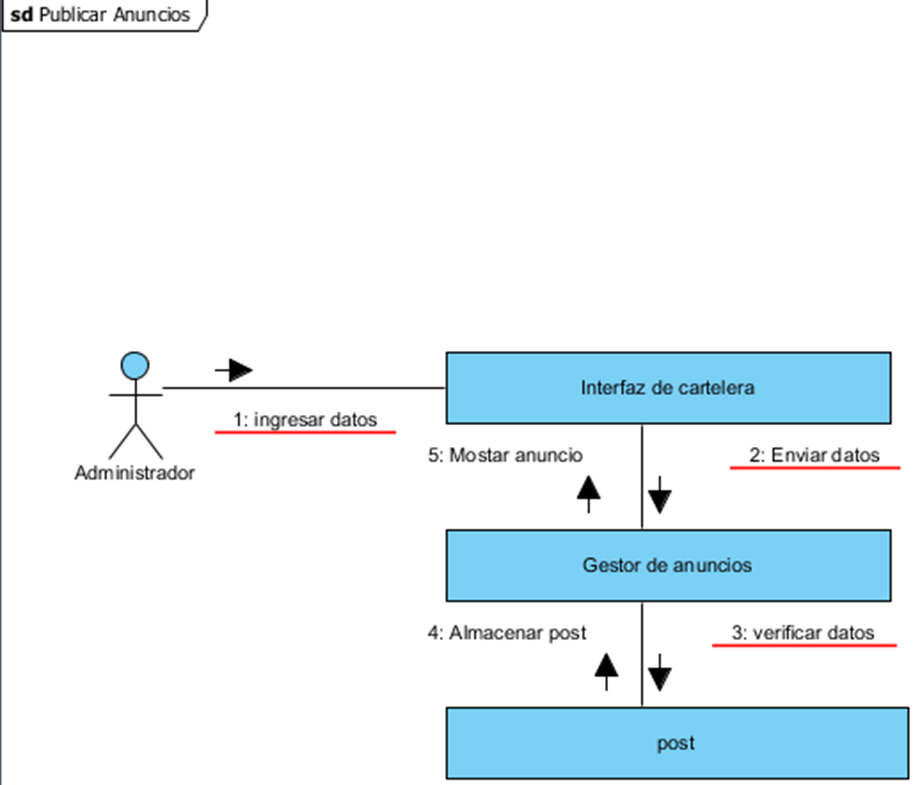
****

Figura diagrama de comunicación “Publicar Anuncios”

**2.5.6.1 Descripción del proceso de comunicación “Publicar Anuncios”**

|  |
| --- |
| Nombre del Diagrama: Publicar anuncios. |
| Actores: Usuarios. |
| Procesos:  1: Ingresar datos: Se ingresan los datos requeridos a través de la interfaz cartelera.  2: Enviar datos: Los datos son procesados mediantes el gestor de anuncios  3: Verificar datos: Una vez procesados los datos, se evalúan que estos corresponda al requerimiento según la bdd.  4: Almacenar post: se almacena la información en la tabla correspondiente y se envía la respuesta del requerimiento la cual es procesada por el gestor de anuncios.  5: Mostrar anuncios: se muestra la publicación creada mediante la interfaz cartelera. |

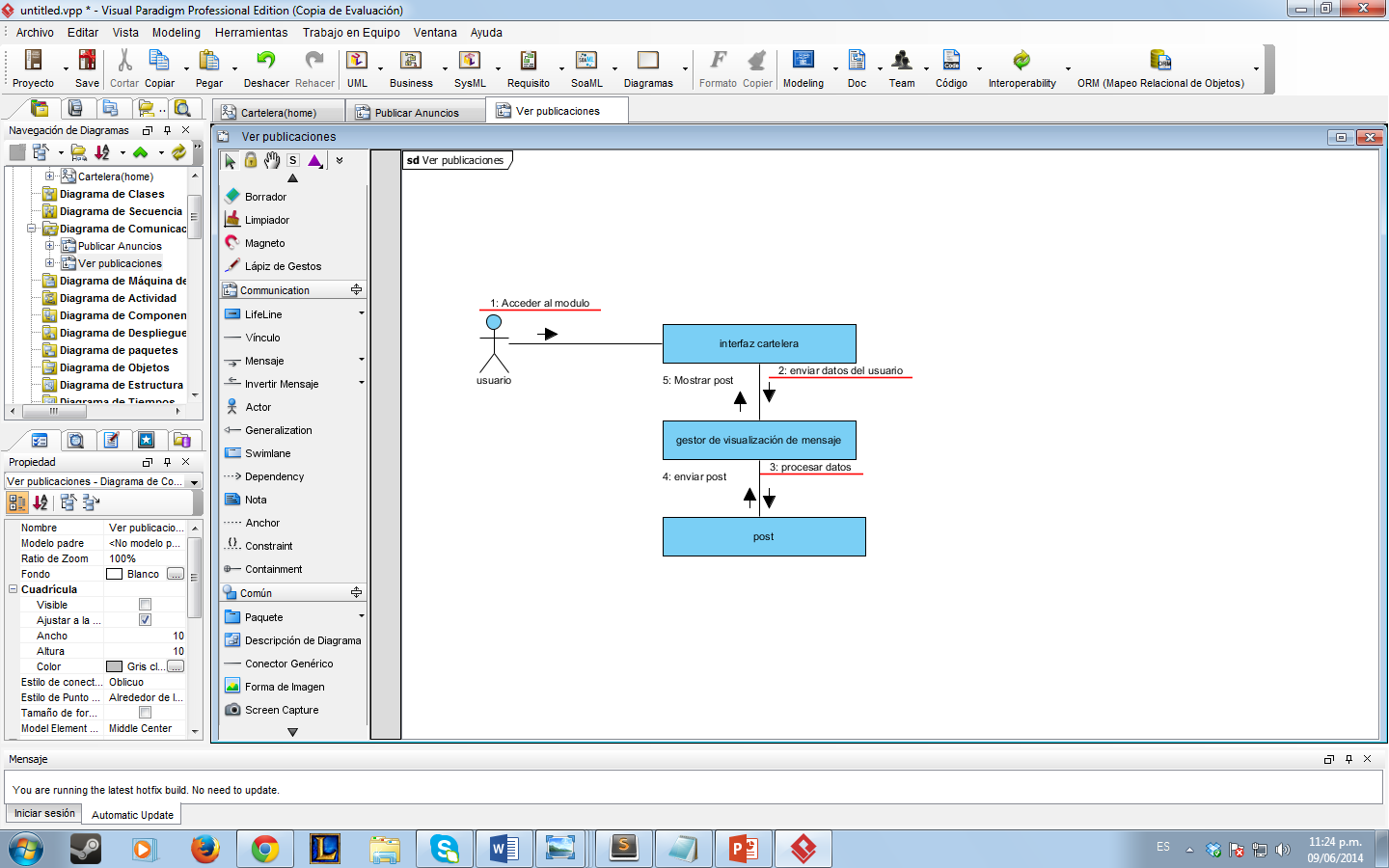


Figura diagrama de comunicación “Ver publicaciones”

**2.5.6.2 Descripción del proceso de comunicación “Ver publicaciones”**

|  |
| --- |
| Nombre del Diagrama: Ver publicaciones. |
| Actores: Usuarios. |
| Procesos:  1: Acceder al módulo: Se ingresan a la interfaz cartelera.  2: Enviar datos del usuario: Los datos del usuaria son procesados mediantes el gestor de anuncios.  3: Procesar datos: Se evalúan que publicaciones tiene permiso de ver el usuario.  4: Enviar post: Se envía todas las publicaciones resultantes de la evaluación.  5: Mostrar anuncios: se muestra estas publicaciones mediante la interfaz cartelera. |

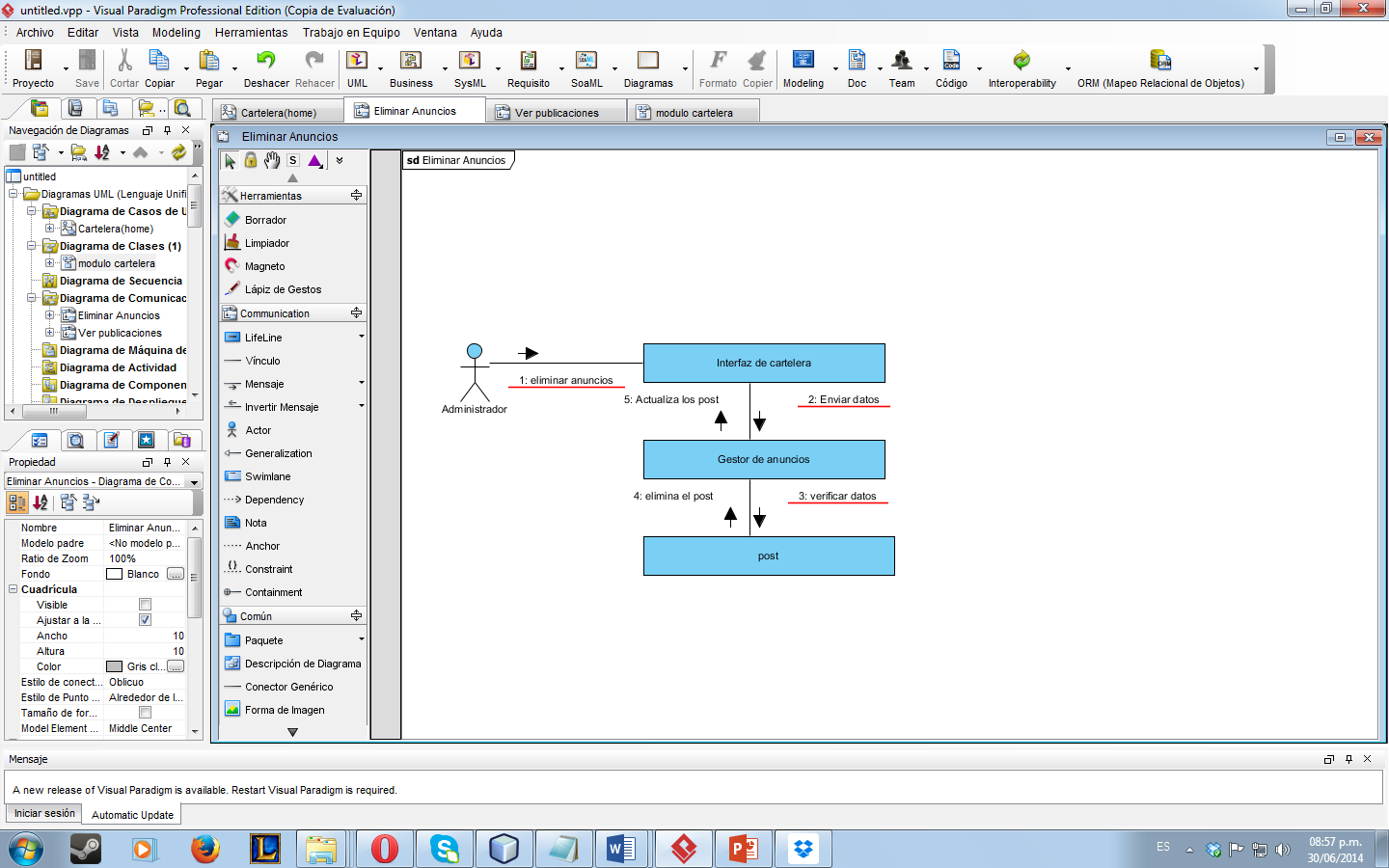


Figura diagrama de comunicación “Eliminar anuncios”

**2.5.6.3 Descripción del proceso de comunicación “Eliminar anuncios”**

|  |
| --- |
| Nombre del Diagrama: Eliminar anuncios. |
| Actores: Usuarios. |
| Procesos:  1: Eliminar anuncios: Se accede al requerimiento desde la interfaz cartelera.  2: Enviar datos: Los datos de la publicación son procesados mediantes el gestor de anuncios.  3: Verificar datos: Se procesan los datos del post.  4: Eliminar post: Se ejecuta el requerimiento.  5: Mostrar anuncios: se actualizan las publicaciones y se muestran mediante la interfaz cartelera. |

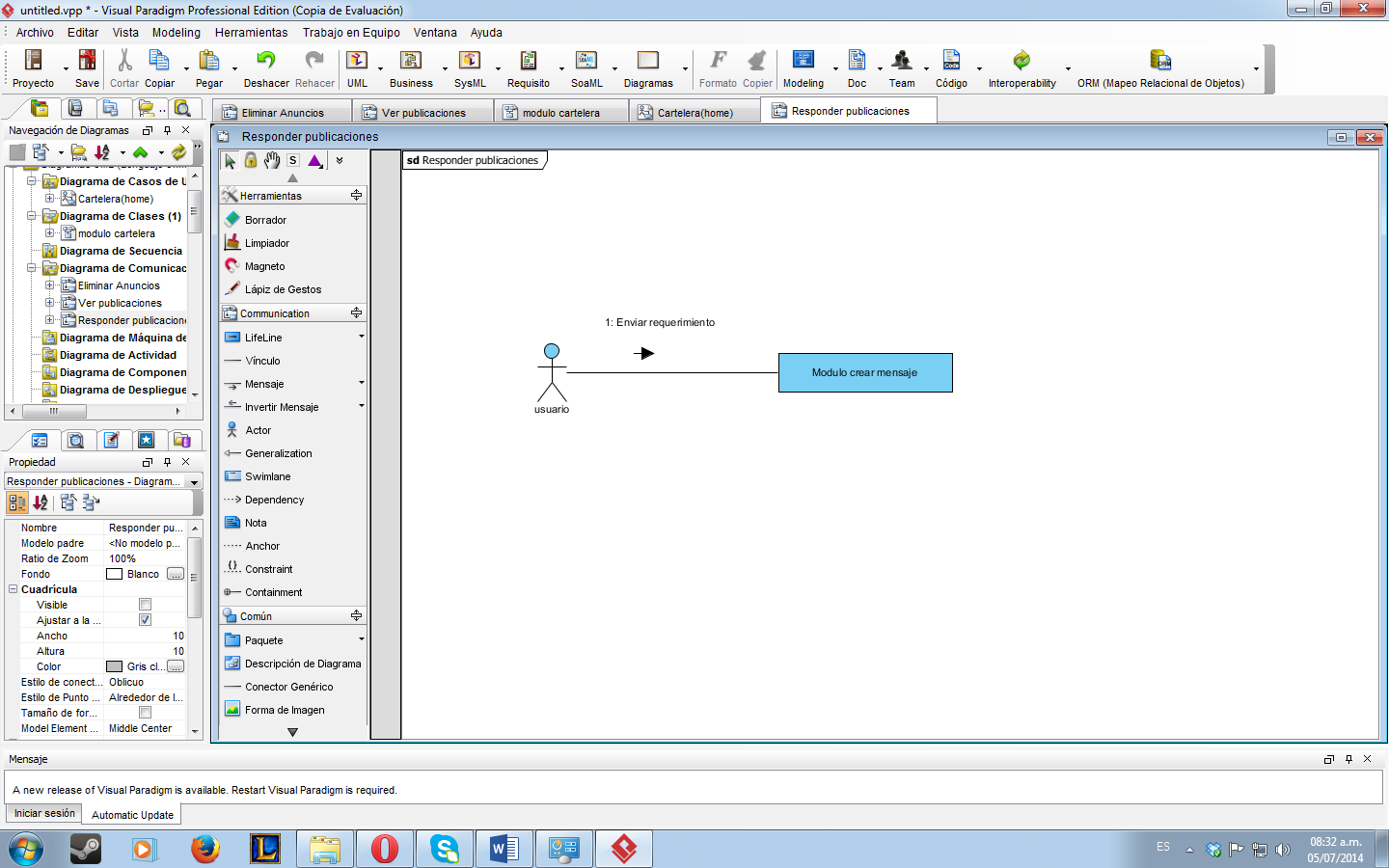


Figura diagrama de comunicación “Responder publicaciones”

**2.5.6.4 Descripción del proceso de comunicación “Responder publicaciones”**

|  |
| --- |
| Nombre del Diagrama: Responder publicaciones. |
| Actores: Usuarios. |
| Procesos:  1: Enviar requerimiento: Se procesa la solicitud re direccionando al módulo crear mensaje, donde se llevara a cabo la solicitud |

**2.5.7 Diagrama de comunicación mensajería.**

**2.5.8 Diagrama de comunicación servicios.**

**2.5.9. Diagrama de comunicación gastos.**

**2.5.10 Diagrama de comunicación pagos.**

**2.6 Diagrama de clases.**

El Diagrama de clases es utilizado por los desarrolladores para especificar los requerimientos funcionales, considerando una o varias clases, o subsistemas del sistema a desarrollar.

UML existen tres estereotipos que permiten distinguir el ámbito de las diferentes clases:

* **Estereotipo de Entidad**: se utiliza para modelar información que posee una vida larga y que es a menudo persistente, es la típica entidades de los modelos de entidad-relación tradicionales, accedido normalmente por varios casos de uso y suelen asociárseles una base de datos.
* **Estereotipo de Interfaz:** se utiliza para modelar las interacciones entre el sistema y sus actores (es decir, usuarios y sistemas externos). Esta interacción a menudo implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y sistemas externos.

#### **Estereotipo de Control** Representa coordinación, secuencia, transacciones y control de otros objetos y se usa con frecuencia para encapsular el control de un caso de uso en concreto.

**2.6.1 Diagrama de clases general.**

**2.6.2 Diagrama de clases página principal**

**2.6.3 Diagrama de clases Edificio.**

**2.6.4 Diagrama de clases Apartamento.**

**2.6.5 Diagrama de clases Usuario**

**2.6.6 Diagrama de clases cartelera**

**2.6.7 Diagrama de comunicación mensajería.**

**2.6.8 Diagrama de comunicación servicios.**

**2.6.9. Diagrama de comunicación gastos.**

**2.6.10 Diagrama de comunicación pagos.**

**2.7 Diagrama de entidad-relación.**

**2.7.1 Diagrama de entidad relación general.**

**2.7.2 Diagrama de entidad relación página principal**

**2.7.3 Diagrama de entidad relación Edificio.**

**2.7.4 Diagrama de entidad relación Apartamento.**

**2.7.5 Diagrama de entidad relación Usuario**

**2.7.6 Diagrama de entidad relación Cartelera**

**2.7.7 Diagrama de entidad relación mensajería.**

**2.7.8 Diagrama de entidad relación servicios.**

**2.7.9. Diagrama de entidad relación gastos.**

**2.7.10 Diagrama de entidad relación pagos.**

**CAPITULO III. DISEÑO DEL SISTEMA**

**3.1 Diseño de la base de datos**

Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada. Es diseñada para dar servicios eficientemente a muchas aplicaciones a centralizar los datos y minimizar aquellos que son redundantes. En lugar de separar los datos en archivos por separados para cada aplicación, los datos son almacenados físicamente para aparecer los datos almacenados en una sola ubicación: una sola base de dato sirve a muchas aplicaciones.

**3.1.2 Diseño del modelo conceptual de la base de datos**

El modelo de datos utilizado para el diseño de la base de datos del sistema es el modelo conceptual de datos, en donde se representan todos los datos en la base de datos como sencillas tablas de dos dimensiones llamadas relaciones.

La estructura principal del modelo conceptual es la relación, es decir una tabla bidimensional compuesta por líneas y columnas. Cada línea, que en terminología relacional se llaman tupla, representa una entidad que se quiere memorizar en la base de datos, las características de cada entidad están definidas por las columnas de las relaciones, que se llaman atributos.

En la (Figura ), se observa el modelo conceptual de datos aplicable para cualquier condominio de edificios. Él cual está formado por 19 entidades identificadas con sus nombres, dichas entidades reúnen un conjunto específico de campos entre los cuales se encuentra el campo clave (subrayado y resaltado en negrita), los campos comunes y los restantes; estos campos hacen referencia a los atributos que reflejan las características principales de cada una de las entidades. Dichas entidades están asociadas por líneas que expresan la relación existente entre ellas además de indicar su cardinalidad.

**3.1.3 Estructuración de los datos**

Las tablas que conforman la base de datos de la aplicación web para cualquier condominio de edificios aplicó previamente el proceso de normalización para asegurar que las relaciones no poseen datos redundantes.

La estructura física de los datos está compuesta por datos, los cuales representan los atributos de cada tabla. En cada campo se especifican las siguientes características: nombre del campo, tipo de datos, tamaño del campo y definición de clave primaria.

En cada entidad de este modelo conceptual, se definieron las propiedades importantes de sus campos a través de una tabla donde se indica el nombre del campo, el tipo de datos que posee, su longitud, además de una breve explicación del campo.

A continuación se muestra la estructura física de cada una de las tablas utilizadas para el crear la aplicación web para cualquier condominio de edificios.

**3.1.3.1 Entidad “apartamento”.**

**3.1.3.2 Entidad “apartamentotipo”.**

**3.1.3.3 Entidad “apartamento\_usuario”.**

**3.1.3.4 Entidad “edificio”.**

**3.1.3.5 Entidad “paginaprincipal”.**

**3.1.3.6 Entidad “pagos”.**

**3.1.3.7 Entidad “pagos\_bancos”.**

**3.1.3.8 Entidad “pagos\_estados”.**

**3.1.3.9 Entidad “pagos\_tipos”.**

**3.1.3.10 Entidad “post”.**

Esta entidad almacena los datos correspondientes a los anuncios hechos por los administradores del condominio de edificios. El campo clave es **idPost,**  esta entidad está relacionada con la tabla usuarios, con una relación de uno a muchos, ya que un usuario puede realizar muchos anuncios, además de esto guarda relación con la tabla apartamentos, pero no directamente ya que estos son almacenados en un arreglo para evitar crear otra tabla cuyo volumen de datos, daría problemas al gestor de base de datos, además de manejo ineficiente y consumo de tiempo de procesamiento.

Las propiedades de esta entidad se indican en la tabla que se muestra a continuación:

Tabla propiedades de la entidad “post”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre del campo | Dato | Tamaño | | Descripción |
| idPost | Numérico | 7 | Clave autoincrementable | |
| titulo | Varchar | 25 | Tirulo del anuncio | |
| contenido | Varchar | 500 | Contenido del anuncio | |
| aptos | Text |  | Restricciones de privacidad de los mensajes. | |
| fecha | Datatime |  | Fecha y hora en que se realizó el anuncio. | |
| usuario | Varchar | 25 | Nombre del usuario que realiza el anuncio. | |
| apellido | Varchar | 25 | Apellido del usuario que realiza el anuncio | |

**3.1.3.11 Entidad “publicacion”.**

**3.1.3.12 Entidad “servicios”.**

**3.1.3.13 Entidad “servicioscostos”.**

**3.1.3.14 Entidad “serviciospagos”.**

**3.1.3.15 Entidad “solicituddeservicios”.**

**3.1.3.16 Entidad “telefono”.**

**3.1.3.17 Entidad “tipodeservicios”.**

**3.1.3.18 Entidad “tipopagodeservicios”.**

**3.1.3.19 Entidad “usuario”.**

**3.2 Interfaz del sistema.**

**3.2.1 Interfaz general**

**3.2.2 Interfaz página principal**

**3.2.3 Interfaz Edificio.**

**3.2.4 Interfaz Apartamento.**

**3.2.5 Interfaz Usuario**

**3.2.6 Interfaz Cartelera**

**3.2.6.1 Vista del Administrador**

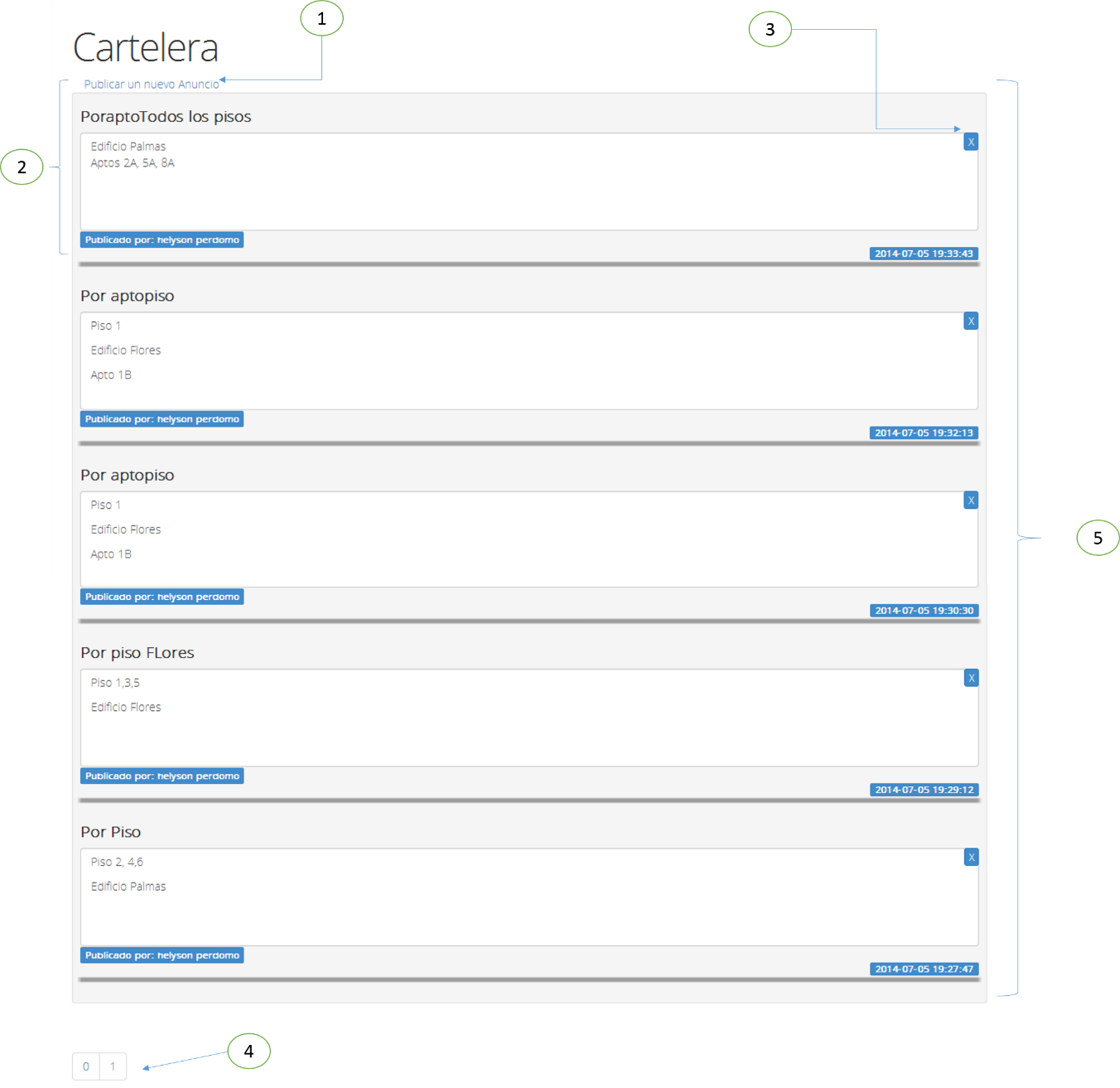


Figura Interfaz cartelera general

**3.2.6.1.1Descripción detallada Interfaz cartelera vista administrador**

|  |
| --- |
| Interfaz: cartelera. |
| Vista: Administrador |
| Nombre: General |
| Componentes:  1: Link para acceder a publicar un mensaje (ver figura interfaz cartelera “publicar un nuevo anuncio”).  2: Anuncio: compuesto por el título, contenido, nombre y apellido de quien publica el mensaje, un botón para eliminar mensaje.  3: Botón para eliminar el mensaje, realiza la solicitud para llevar a cabo el requerimiento (ver figura interfaz eliminar anuncios).  4: Paginación botones que permiten ver más publicaciones.  5: cantidad de mensajes que se pueden visualizar por páginas. |
|  |

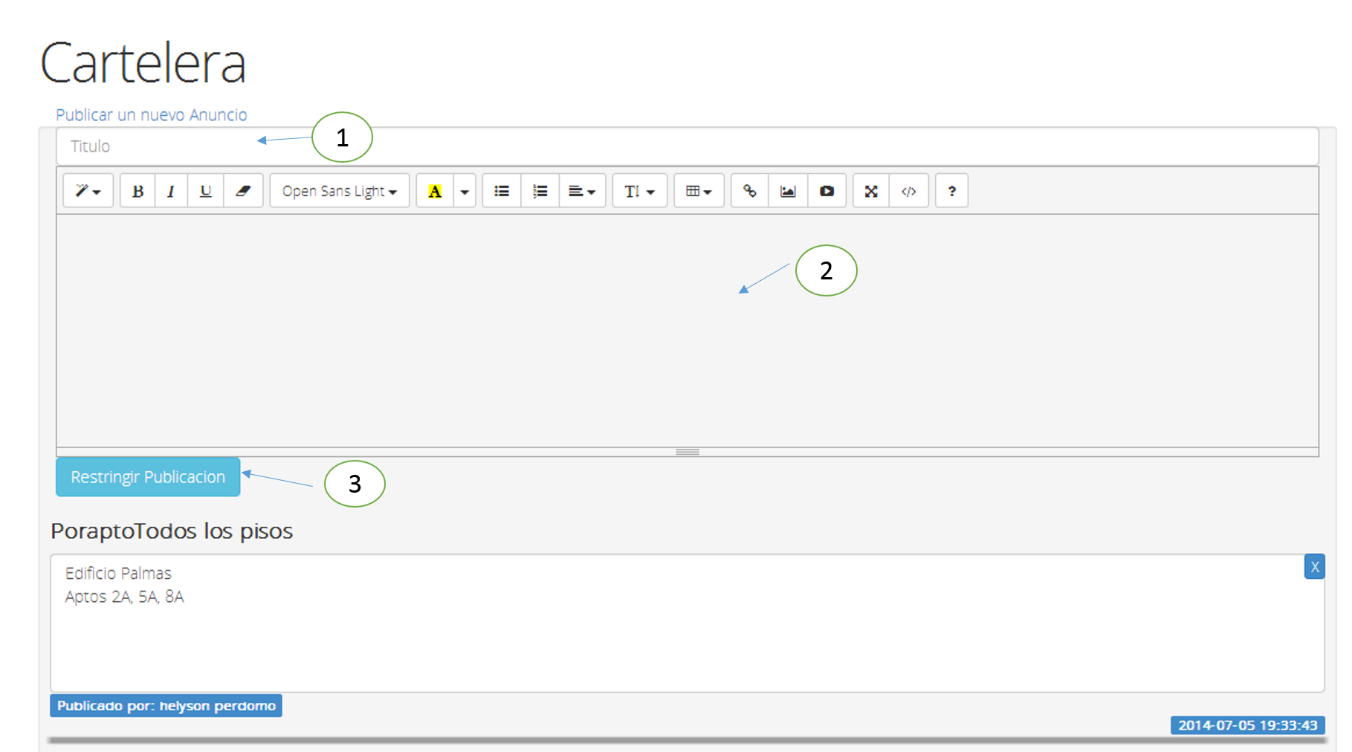


Figura Interfaz cartelera “publicar un nuevo anuncio”

**3.2.6.1.2 Descripción detallada Interfaz cartelera “publicar un nuevo anuncio”**

|  |
| --- |
| Interfaz: cartelera. |
| Vista: Administrador |
| Nombre: Publicar un nuevo anuncio |
| Componentes:  1: Título del anuncio correspondiente.  2: Contenido de la publicación.  3: Botón para seleccionar las restricciones del mensaje (ver figura interfaz cartelera “restringir publicación”). |

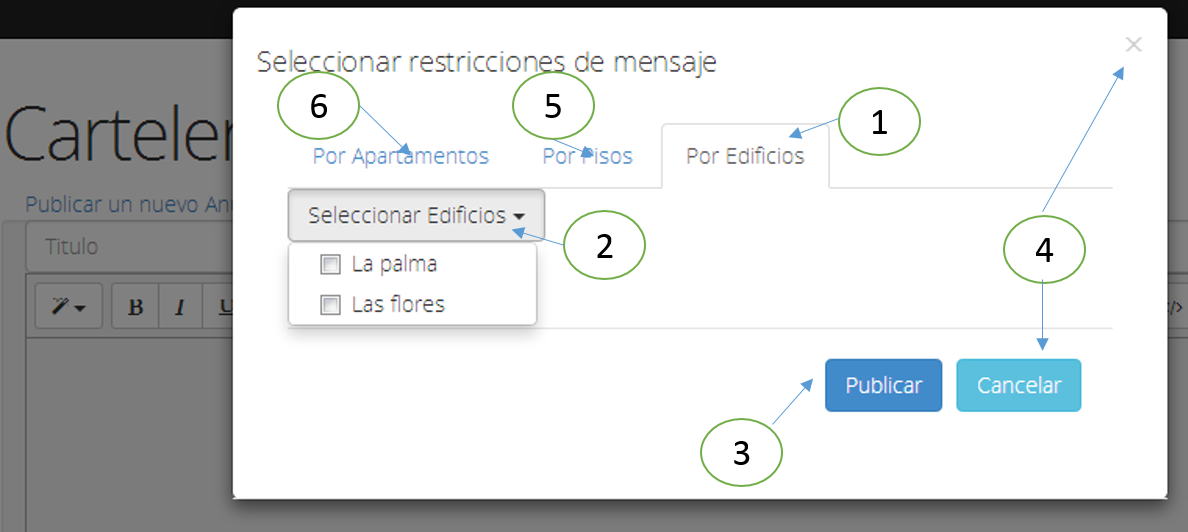


Figura Interfaz cartelera “restringir publicación”.

**3.2.6.1.3 Descripción detallada Interfaz cartelera “restringir publicación”**

|  |
| --- |
| Interfaz: cartelera. |
| Vista: Administrador |
| Nombre: Restringir publicación |
| Componentes:  1: Restringir publicación según los edificios del conjunto residencial.  2: Select múltiple, en donde se listan los edificios que se encuentran en el conjunto residencial.  3: Botón para publicar el mensaje.  4: Botones para cerrar el modal restringir mensaje.  5: Restringir publicación según los pisos pertenecientes a un edificio del conjunto residencial (ver figura interfaz cartelera restringir por pisos).  6: Restringir publicación según los apartamentos pertenecientes a un edificio del conjunto residencial (ver figura interfaz cartelera restringir por apartamento). |

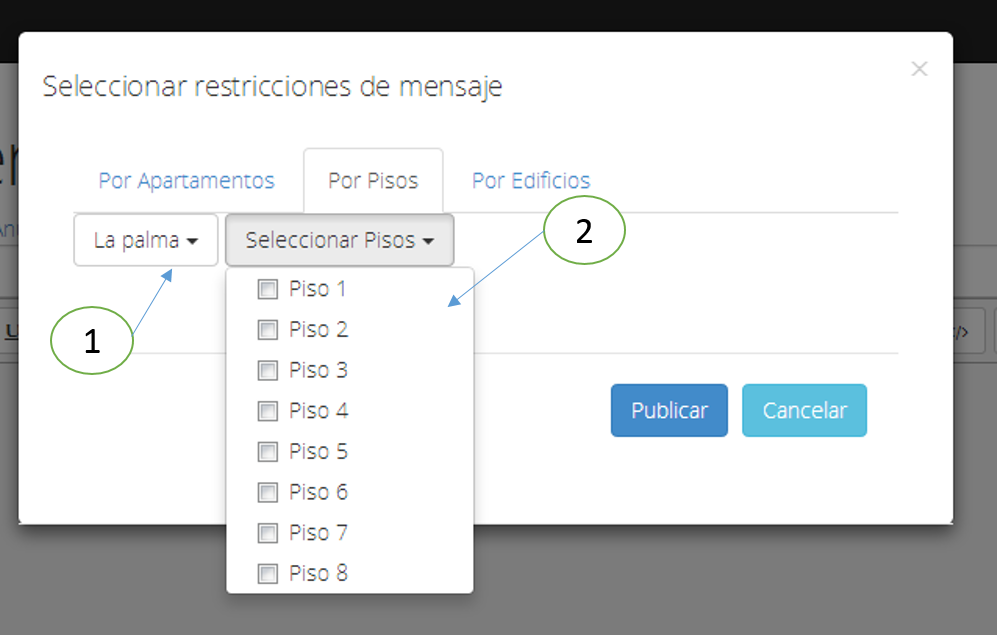


Figura Interfaz cartelera “restringir publicación por pisos”.

**3.2.6.1.4 Descripción detallada Interfaz cartelera “restringir publicación por pisos”**

|  |
| --- |
| Interfaz: cartelera. |
| Vista: Administrador |
| Nombre: Restringir publicación por pisos |
| Componentes:  1: Select simple, en donde se listan los edificios que se encuentran en el conjunto residencial.  2: Select múltiple, en donde se listan los pisos del edificio seleccionado |

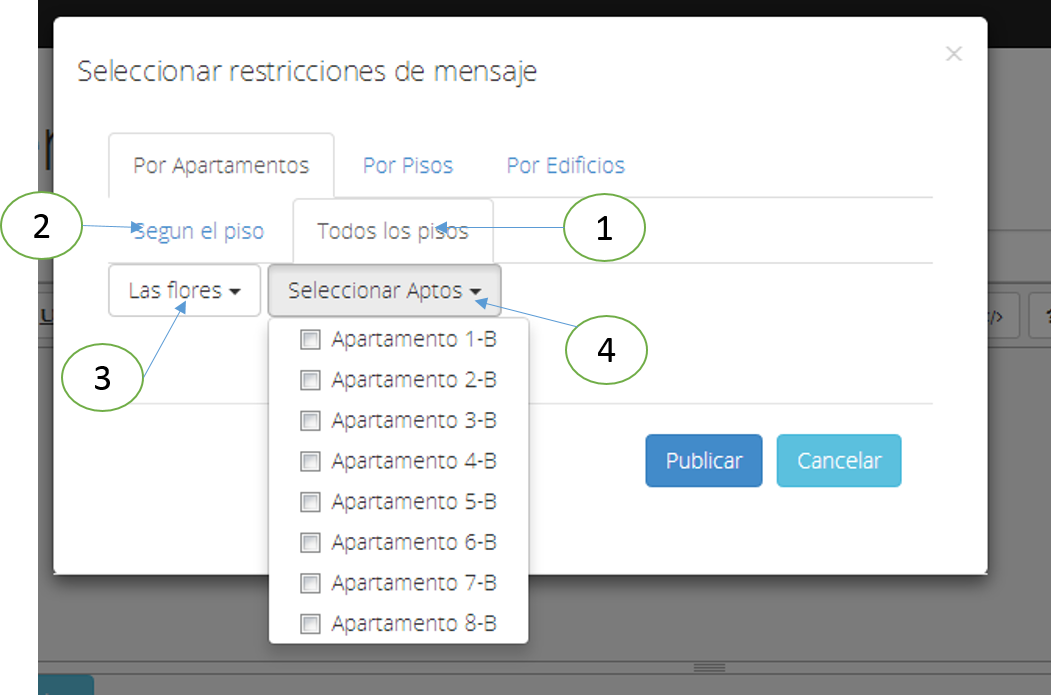


Figura Interfaz cartelera “restringir publicación por apartamentos”.

**3.2.6.1.5 Descripción detallada Interfaz cartelera “Restringir publicación por apartamentos”**

|  |
| --- |
| Interfaz: cartelera. |
| Vista: Administrador |
| Nombre: Restringir publicación por apartamentos |
| Componentes:  1: Restringir por apartamento según todos los pisos del edificio.  2: Restringir por apartamento según el piso de un edificio (ver interfaz cartelera restringir por apartamento según el piso).  3: Select simple, en donde se listan los edificios que se encuentran en el conjunto residencial.  4: Select múltiple, en donde se listan todos los apartamentos del edificio seleccionado |



Figura Interfaz cartelera “restringir publicación por apartamentos según el piso”.

**3.2.6.1.6 Descripción detallada Interfaz cartelera “Restringir publicación por apartamentos según el piso”**

|  |
| --- |
| Interfaz: cartelera. |
| Vista: Administrador |
| Nombre: Restringir publicación por apartamentos según el piso |
| Componentes:  1: Select simple, en donde se listan los edificios que se encuentran en el conjunto residencial.  2: Select simple, en donde se listan los pisos del edificio seleccionado.  3: Select múltiple, en donde se listan todos los apartamentos del piso del edificio seleccionado. |

**3.2.6.2 Vista del Usuario**

**3.2.7 Interfaz mensajería.**

**3.2.8 Interfaz servicios.**

**3.2.9. Interfaz gastos.**

**3.2.10 Interfaz pagos.**

**3.3 Pruebas de integración.**

Conociendo el funcionamiento interno del producto, se aplican pruebas para asegurarse que “todas las piezas encajan”, en otras palabras, están integradas correctamente. Se basa en un examen cercano al detalle procedimental. Estas pruebas se aplicaron a todos los componentes de software desarrollados, comprobando así la exitosa integración de la aplicación. Además, se probaron los enlaces de las páginas, que no hubiesen enlaces rotos o vacios, ni enlaces que direccionarán hacia alguna página indebida. A continuación se presentan ejemplos de algunos casos de prueba de integración que se aplicaron al sistema.

**3.3.1 Prueba de caja blanca.**

**3.3.2 Prueba de caja negra.**