

**EXTENSIÓN DE UNA PLATAFORMA QUE SOPORTE UN SISTEMA PARA LA
PROVISIÓN DE SERVICIOS EN UN ENTORNO DE REDES SUPERPUSTAS P2P**

ANEXOS



GABRIEL RAMIRO MUÑOZ SAMBONÍ

EDWIN FERNANDO LARA QUINCHUA

Director: Ing. PABLO AUGUSTO MAGÉ IMBACHÍ

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
LÍNEA DE INVESTIGACIÓN INGENIERÍA DE LA COLABORACIÓN E
INGENIERÍA DEL SOFTWARE BASADA EN LA COLABORACIÓN
DEPARTAMENTO DE SISTEMAS
POPAYÁN
JUNIO DE 2009**

TABLA DE CONTENIDO

ANEXOS.....¡Error! Marcador no definido.

Anexo 1. Capas de la arquitectura de JXTA.....¡Error! Marcador no definido.

1.1 Capa del núcleo (*JXTA Core*).....**¡Error! Marcador no definido.**

1.2 Capa de servicios (*JXTA Services*).....**¡Error! Marcador no definido.**

1.3 Capa de aplicación (*JXTA Applications*).....**¡Error! Marcador no definido.**

Anexo 2. Protocolos JXTA.....¡Error! Marcador no definido.

2.1 Peer Resolver Protocol (Protocolo de resolución de peers)**¡Error! Marcador no definido.**

2.2 Peer Discovery Protocol (Protocolo de descubrimiento de peers)**¡Error! Marcador no definido.**

2.3 Peer Information Protocol (Protocolo de información de peers)**¡Error! Marcador no definido.**

2.4 Pipe Binding Protocol (Protocolo de asociación de tuberías)**¡Error! Marcador no definido.**

2.5 Endpoint Routing Protocol (Protocolo de enrutamiento de terminal)**¡Error! Marcador no definido.**

2.6 Rendezvous Protocol (Protocolo de punto de encuentro)**¡Error! Marcador no definido.**

Anexo 3. Componentes básicos de JXTA.....¡Error! Marcador no definido.

3.1 Peer Group.....**¡Error! Marcador no definido.**

3.2 Módulos.....**¡Error! Marcador no definido.**

3.3 Pipes.....**¡Error! Marcador no definido.**

3.4 Mensajes.....**¡Error! Marcador no definido.**

3.5 Seguridad.....**¡Error! Marcador no definido.**

3.6 Advertisements.....**¡Error! Marcador no definido.**

3.7 Servicios de grupo de peers.....**¡Error! Marcador no definido.**

Anexo 4. Especificaciones técnicas – Agenda colaborativa P2P¡Error! Marcador no definido.

4.1 Actores de la aplicación.....**¡Error! Marcador no definido.**

4.2 Diagrama inicial de casos de uso de la aplicación p2p (agenda colaborativa)**¡Error! Marcador no definido.**

4.3 Relación de los casos de uso.....**¡Error! Marcador no definido.**

4.4 Especificación formal casos de uso.....**¡Error! Marcador no definido.**

4.4.1 Caso de uso conectarse a la aplicación.....**¡Error! Marcador no definido.**

4.4.2 Caso de uso crear grupo.....**¡Error! Marcador no definido.**

4.4.3 Caso de uso cambio de grupo.....**¡Error! Marcador no definido.**

4.4.4 Caso de uso envío y recepción de petición de chat**¡Error! Marcador no definido.**

4.4.5 Caso de uso envío y recepción de mensajes.....**¡Error! Marcador no definido.**

4.4.6 Caso de uso envío y recepción de archivos.....**¡Error! Marcador no definido.**

4.4.7 Caso de uso publicar actividad.....**¡Error! Marcador no definido.**

4.4.8 Caso de uso publicar evento.....**¡Error! Marcador no definido.**

4.4.9 Caso de uso emitir notificaciones.....**¡Error! Marcador no definido.**

4.5 Diagramas de Secuencia.....**¡Error! Marcador no definido.**

4.5.1 Caso de uso GE-001 conectarse a la aplicación.. **¡Error! Marcador no definido.**

4.5.2 Caso de uso GE-002 crear grupo.....**¡Error! Marcador no definido.**

4.5.3 Caso de uso GE-003 cambio de grupo.....**¡Error! Marcador no definido.**

4.5.4 Caso de uso GE-004 envío y recepción de petición de chat**¡Error! Marcador no definido.**

4.5.5 Caso de uso GE-005 envío y recepción de mensajes**¡Error! Marcador no definido.**

4.5.6 Diagrama de secuencia detección de mensaje entrante**¡Error! Marcador no definido.**

4.5.7 Caso de uso GE-006 envío y recepción de archivos**¡Error! Marcador no definido.**

4.5.8 Caso de uso GE-007 publicar actividad.....**¡Error! Marcador no definido.**

- 4.5.9 Caso de uso GE-008 publicar evento.....**¡Error! Marcador no definido.**
- 4.5.10 Caso de uso GE-009 emitir notificaciones**¡Error! Marcador no definido.**
- 4.6 Diagrama de clases de la aplicación p2p (Agenda Colaborativa)**¡Error! Marcador no definido.**

ANEXOS

Anexo 1. Capas de la arquitectura de JXTA

1.1 Capa del núcleo (*JXTA Core*)

La capa del núcleo encapsula las primitivas mínimas y esenciales para el adecuado funcionamiento de la red, es decir, brinda el soporte para las operaciones fundamentales de las aplicaciones, incluyendo descubrimiento, transporte (manejo de firewalls), creación de peers y *peer groups* y las primitivas de seguridad asociadas. Los elementos de esta capa constituyen el fundamento de JXTA y sobre ella se basan todos los aspectos de las capas de servicios y aplicación.

Los elementos presentes en el núcleo de JXTA son: Peers, *peer groups*, transporte de red (*endpoints*¹, *pipes*², mensajes), protocolos (descubrimiento, comunicación, monitoreo), *advertisements*, identificadores, primitivas de autenticación y seguridad.

Esta capa incluye además seis protocolos básicos y esenciales:

1. *Peer Resolver Protocol* (Protocolo de resolución de peers).
2. *Peer Discovery Protocol* (Protocolo de descubrimiento de peers).
3. *Peer Information Protocol* (Protocolo de información de peers).
4. *Pipe Binding Protocol* (Protocolo de asociación de tuberías).
5. *Endpoint Routing Protocol* (Protocolo de enrutamiento de terminal).
6. *Rendezvous Protocol* (Protocolo de punto de encuentro).

Aunque estos protocolos son implementados como servicios, están ubicados en la capa del núcleo y son designados como servicios del núcleo para distinguirlos de las soluciones de la capa de servicios.

1.2 Capa de servicios (*JXTA Services*)

Esta capa incluye los servicios de red, los cuales no son absolutamente necesarios para su buen funcionamiento, pero si son comunes y ofrecen funciones deseables para los entornos P2P.

¹ Son el origen o el destino de cualquier fragmento de datos que se transmite por la red. Un *endpoint* corresponde a las interfaces de red utilizadas para enviar y recibir datos.

² Son canales virtuales de comunicación asíncronos y unidireccionales o bidireccionales que conectan uno o más *endpoints*.

Entre los servicios de red más comunes están: búsqueda e indexado, directorio, sistemas de almacenamiento de información, intercambio de archivos, sistemas de archivos distribuidos, almacenamiento y alquiler de recursos, traducción de protocolos, autenticación, servicios de infraestructura de llave pública (PKI), entre otros.

1.3 Capa de aplicación (JXTA Applications)

Esta capa incluye la implementación de aplicaciones integradas, tales como mensajería, intercambio de documentos y recursos, reparto y gestión de contenidos, sistemas de correo electrónico, sistemas de subasta, y muchas otras aplicaciones P2P.

El límite entre servicios y aplicaciones no es rígido. Una aplicación para un cliente podría ser vista como un servicio para otro. El sistema en general está diseñado para ser modular, permitiendo a los desarrolladores escoger de entre una gran cantidad de aplicaciones y servicios, aquellos que mejor se ajusten a sus necesidades.

Anexo 2. Protocolos JXTA

JXTA define seis protocolos que pueden ser utilizados para construir aplicaciones P2P (Figura 5). Tales protocolos están diseñados con un *overhead* bajo, es decir, no asumen nada acerca de la topología de red subyacente sobre la cual se ejecutan las aplicaciones.

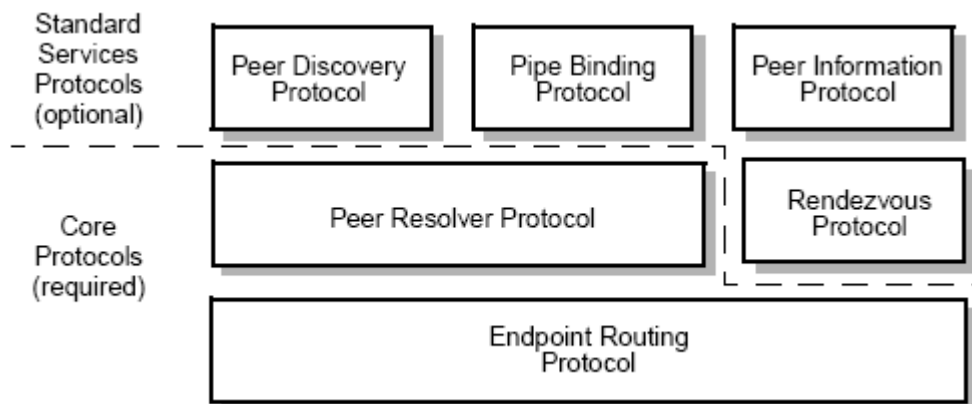


Figura 1. Protocolos JXTA [27]

El número de protocolos que un peer implementa depende en gran parte de sus capacidades. Los protocolos se pueden extender o reemplazar, dependiendo de los requerimientos del sistema a desarrollar.

Los protocolos estandarizan la manera en la cual los *peers*:

- Se encuentran unos a otros.
- Se organizan automáticamente en grupos.

- Publican y encuentran recursos.
- Se comunican unos con otros.
- Se monitorean unos a otros.

En la versión 2.0 de la especificación de los protocolos de JXTA [33] se definen los siguientes protocolos:

2.1 Peer Resolver Protocol (Protocolo de resolución de peers)

Es el mecanismo mediante el cual un peer envía una petición a uno o más peers, y recibe una o múltiples respuestas a dicha petición (protocolo petición/respuesta). El mensaje de respuesta se produce de acuerdo a un identificador único incluido en el cuerpo del mensaje. Las consultas pueden ser dirigidas a un grupo entero o a un peer en particular.

2.2 Peer Discovery Protocol (Protocolo de descubrimiento de peers)

A través de este protocolo un peer puede anunciar todos sus recursos y descubrir los recursos que otros peers ofrecen (*pipes*, *peer groups*, servicios, etc.). Cada recurso se describe y se publica usando un *advertisement*. Los *advertisements* son metadatos, independientes del lenguaje de programación, que describen los recursos de red y se representan mediante documentos XML.

2.3 Peer Information Protocol (Protocolo de información de peers)

Es el mecanismo que le permite a un peer obtener información sobre el estado de otros peers. Esta información puede incluir tiempo en línea (*uptime*), carga de tráfico, capacidades, estado, entre otros.

2.4 Pipe Binding Protocol (Protocolo de asociación de tuberías)

Es el mecanismo mediante el cual un peer puede establecer un canal de comunicación virtual o tubería (*pipe*) entre uno o más peers. Los peers utilizan este protocolo para asociar dos o más *endpoints* a una conexión. Los *pipes* son el mecanismo fundamental para establecer la comunicación entre los peers.

2.5 Endpoint Routing Protocol (Protocolo de enrutamiento de terminal)

Es el mecanismo mediante el cual un peer puede descubrir una ruta (secuencia de saltos) para enviar mensajes a otro peer. Si un peer A quiere enviar un mensaje a un peer C, y no hay una ruta directa entre ellos, entonces el peer A necesita encontrar los peers intermediarios que encaminen sus mensajes hacia el peer C. Este protocolo se utiliza

para determinar la información acerca de dicha ruta. Si la topología de red cambia y las rutas previamente establecidas ya no están disponibles, los peers pueden utilizar este protocolo para encontrar una ruta alternativa al destino.

2.6 Rendezvous Protocol (Protocolo de punto de encuentro)

Es el mecanismo mediante el cual los peers se registran o se retiran de un servicio de difusión. Dentro de un *peer group*, los peers pueden ser *peer rendezvous* o peers normales. Mediante este protocolo un peer puede enviar mensajes a todos los clientes del servicio *rendezvous*. El protocolo de resolución de peers y el protocolo de asociación de tuberías utilizan este protocolo cuando necesitan propagar sus propios mensajes de protocolo.

Anexo 3. Componentes básicos de JXTA

En el capítulo 2, sección 2.4, se describieron los elementos básicos de una red P2P. Por su naturaleza, JXTA implementa esta serie de elementos y añade otros tantos que son esenciales para su funcionamiento. A continuación se describen en detalle:

3.1 Peer Group

La naturaleza especializada de la gran mayoría de los sistemas P2P y sus protocolos asociados generan segmentación en la red. En JXTA, sin embargo, todos los peers ejecutan aplicaciones que implementan el mismo conjunto de protocolos, por ello se hace necesario el concepto de grupo. De esta manera un *peer group* se puede definir como un conjunto de peers conformado para servir a un interés común o a los objetivos estipulados por los peers involucrados de acuerdo a una serie de políticas establecidas.

Los *peer groups* sirven fundamentalmente para:

- Crear entornos seguros: Mediante el establecimiento de políticas de seguridad (*login/password*, PKI, etc.) o a través de la publicación de contenidos protegidos de acceso restringido.
- Crear entornos limitados: Creando dominios de especialización (divisiones lógicas) o limitando la búsqueda de los recursos.
- Crear entornos de monitorización: Monitorizando el estado de un grupo de peers.

3.2 Módulos

Los módulos son abstracciones utilizadas para representar cualquier pieza de código utilizado para implementar un comportamiento. Los servicios de red son el ejemplo más común de comportamiento que puede ser instanciado en un peer.

La abstracción del módulo no especifica que necesariamente sea el código, es decir, puede ser una clase Java, un Java jar, una dll dinámica (*dynamic library*), un conjunto de mensajes XML o un script.

Los módulos proveen una abstracción genérica para permitir a un peer instanciar un nuevo comportamiento. A medida que un peer busca o se une a un grupo, puede encontrar nuevos comportamientos que quiera instanciar, es decir, puede ser que tenga que aprender comportamientos que solo utiliza ese grupo de peers (por ejemplo un servicio de búsqueda), por lo que tendrá que instanciarlo. El modulo permite la representación y publicación de comportamientos independientes de la plataforma y permite a los peers describir e instanciar cualquier tipo de implementación de un comportamiento. Por ejemplo, un peer tiene la habilidad de instanciar un comportamiento de la implementación en Java o en C.

La habilidad de describir y publicar un comportamiento independiente de la plataforma es esencial para soportar grupos compuestos por peers heterogéneos.

El módulo de *advertisement* permite a los peers describir un comportamiento independiente de la plataforma.

La abstracción de un módulo incluye una clase, una especificación y una implementación de módulo.

- Clase: Es utilizada para publicar la existencia de un comportamiento. Cada clase de módulo esta identificada por un identificador único llamado *ModuleClassID*.
- Especificación: Contiene toda la información necesaria para acceder o invocar al módulo. En el caso de un servicio, por ejemplo, el modulo de especificación debe incluir un anuncio de *pipe*. En otras palabras, una especificación es un método utilizado para proveer la funcionalidad que este suministra. Cada especificación de módulo tiene su propio identificador llamado *ModuleSpecID*.
- Implementación: Es la implementación de una especificación dada. Pueden existir múltiples implementaciones por cada especificación. Cada modulo de implementación contiene un identificador llamado *ModuleSpecID* correspondiente a la especificación que implementa.

Los servicios JXTA pueden utilizar la abstracción de módulo para identificar la existencia de un servicio (modulo de clase), su especificación (modulo de especificación), o una implementación (modulo de implementación). Cada uno de estos componentes tiene asociado un *advertisement*, el cual puede ser publicado y descubierto por otros peers.

Los módulos son utilizados por los servicios de grupo y también pueden ser utilizados por los servicios *stand-alone*³.

³ Caracteriza a un terminal o un puesto de trabajo que está conectado directamente al ordenador o a un concentrador; es decir, que tiene una sola conexión a través de una línea externa.

3.3 Pipes

Los *pipes* son un mecanismo asíncrono y unidireccional para la transferencia de mensajes usado en el servicio de comunicación. Los *peers* utilizan *pipes* para enviar mensajes a otros *peers*. Los *pipes* no discriminan el contenido que transportan ya que soportan la transferencia de cualquier objeto, desde código binario y cadenas de datos (*strings*), hasta objetos.

En JXTA existen dos tipos de *pipes*: pipes de entrada (*input pipes*) y pipes de salida (*output pipes*). Los *pipes* sirven para conectar *peers* que no tienen un enlace físico directo; en tal caso, uno o más *endpoints* intermediarios son utilizados para transferir los mensajes entre los *peers*.

Los pipes ofrecen dos maneras de comunicación: punto a punto y propagación (Figura 6).

- Pipes punto a punto (*Unicast pipes*): Conecta dos terminales: un *input pipe* y un *output pipe*; para enviar y recibir mensajes respectivamente. También es posible que varios *peers* se unan a un único *input pipe*. En la carga del mensaje existe un identificador para determinar la secuencia de mensajes.
- Pipes de propagación (*Propagate pipes*): Conecta un *output pipe* con múltiples *input pipes*. Los mensajes fluyen desde el *output pipe* (la fuente de difusión) hacia los *input pipes*. Toda la difusión se lleva a cabo dentro de los límites de un grupo de *peers*, es decir, el *output pipe* y los *input pipes* deben pertenecer al mismo grupo.
- Pipes unicast seguros (*Secure unicast*): Es un tipo de *pipe* punto a punto que provee un canal de comunicación seguro y confiable.

Se pueden construir otros tipos de *pipes* a partir de los tipos básicos.

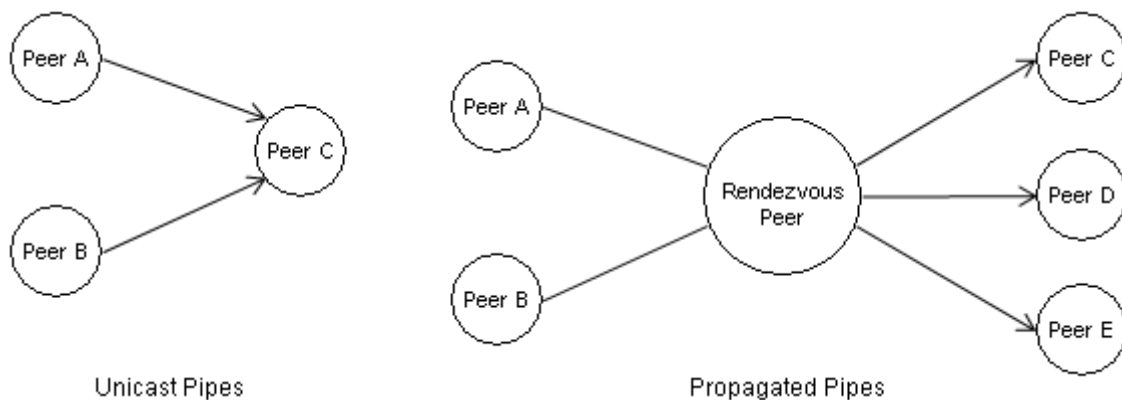


Figura 2. Pipes punto a punto y de propagación

3.4 Mensajes

Un mensaje es un objeto que es enviado entre dos o mas peers. Es la unidad básica para el intercambio de datos. Los mensajes son enviados y recibidos por los servicios *pipe* y *endpoint* respectivamente. Normalmente las aplicaciones utilizan el servicio *pipe* para crear, enviar y recibir mensajes.

Un mensaje esta compuesto por un conjunto de pares nombre/valor, llamados elementos (*elements*), con contenido de cualquier tipo. Hay dos representaciones de mensajes: XML y binario. El uso de mensajes XML para definir los protocolos permite que diferentes clases de peers participen en el protocolo. Como los datos están etiquetados, cada peer es libre de implementar el protocolo de la mejor manera, según sus habilidades y roles.

3.5 Seguridad

Los peers JXTA operan con un modelo de confianza basado en roles. Los peers actúan bajo la autoridad concedida por otros peers de confianza para realizar tareas específicas. JXTA proporciona cinco mecanismos básicos de seguridad:

1. Confidencialidad: Garantiza que el contenido de un mensaje no sea entregado a individuos no autorizados.
2. Autenticación: Garantiza que el emisor sea quien dice ser.
3. Autorización: Garantiza que el emisor del mensaje este autenticado o autorizado para enviar un mensaje.
4. Integridad de los datos: Garantiza que el mensaje no sea modificado accidental o deliberadamente en el transcurso de su viaje.
5. Refutabilidad: Garantiza que el mensaje sea transmitido por un emisor propiamente identificado y que no sea retransmitido de un mensaje previamente enviado.

Los mensajes XML proveen la habilidad de añadir metadatos como credenciales, certificados, resúmenes (*digests*) y llaves públicas, lo cual permite que se obtengan los requerimientos de seguridad básicos. Los *digests* de los mensajes garantizan la integridad de los mismos. Los mensajes también pueden ser encriptados (usando llaves públicas) y firmados (usando certificados) para la confidencialidad y refutabilidad. Las credenciales pueden ser usadas para permitir la autorización y autenticación de mensajes.

Una credencial es una ficha (*token*) utilizada para identificar al emisor y permite verificar si este tiene permisos suficientes para enviar un mensaje a un *endpoint* determinado [14]. La credencial debe ser presentada cada vez que se envía un mensaje. Cada implementación de la credencial se especifica como una configuración de fácil manejo, la cual permite múltiples configuraciones de autenticación coexistentes en la misma red.

Los protocolos de transporte de JXTA solo proveen la integridad y confidencialidad de la transferencia del mensaje entre dos peers que se comunican. Para proveer transferencia segura a través de múltiples saltos se debe establecer una asociación de confianza entre todos los peers intermediarios. La seguridad se verá comprometida si uno de los enlaces de comunicación no es seguro.

3.6 Advertisements

Un *advertisement* es un documento XML utilizado para representar de manera estructurada una entidad, servicio o recurso que publica un peer o un *peer group* a los demás miembros de la red P2P.

Todos los elementos básicos de JXTA se pueden describir mediante *advertisements*, a saber: *peers*, *peer groups*, *endpoints*, *pipes*, mensajes y servicios.

La representación de la información mediante *advertisements* simplifica la utilización de los protocolos al dictaminar la estructura y la representación de los datos. En lugar de pasar los datos de un lado a otro, los protocolos simplemente organizan el intercambio de *advertisements* que contienen la información requerida para llevar a cabo cualquier funcionalidad.

Los peers descubren recursos mediante la búsqueda de sus correspondientes *advertisements*, los cuales una vez descubiertos son almacenados en caché para facilitar la posterior localización del recurso respectivo.

Cada *advertisement* es publicado con un tiempo de vida (*lifetime*), que especifica la disponibilidad de sus recursos asociados. Esto permite eliminar recursos obsoletos sin la necesidad de requerir un control centralizado. Un *advertisement* puede ser publicado de nuevo para extender su tiempo de vida como recurso. Los protocolos JXTA definen los siguientes tipos de *advertisements* [14]:

- *Advertisement* de peer: Describe el recurso del peer. Se usa principalmente para mantener información específica del peer, como su nombre, identificador, *endpoints* disponibles y cualquier atributo que los servicios quieran publicar.
- *Advertisement* de *peer group*: Describe los recursos específicos de un grupo, como el nombre, identificador, descripción, especificación y parámetros de servicio.
- *Advertisement* de *pipe*: Describe un canal de comunicación (*pipe*), y es utilizado por el servicio de *pipe* para crea el *input pipe* y el *output pipe* asociados al *endpoint* correspondiente. Cada *pipe advertisement* contiene un identificador simbólico opcional, un tipo y un identificador único.
- *Advertisement* de clase de módulo: Describe la clase de un módulo. Su propósito principal es documentar formalmente la existencia de la clase de un módulo. Incluye nombre, descripción y un identificador único llamado *ModuleClassID*.
- *Advertisement* de módulo de especificación: Define la especificación de un módulo. Su propósito principal es proveer referencias a la documentación,

necesarias para crear implementaciones conformes a la especificación. Incluye nombre, descripción y un identificador único llamado *ModuleSpecID*.

- *Advertisement* de módulo de implementación: Define una implementación de un módulo de especificación determinado. Incluye nombre, un *ModuleSpecID* asociado, también el código, paquete y parámetros, los cuales permiten a un peer obtener los datos necesarios para ejecutar la implementación.
- *Advertisement* de *rendezvous*: Describe un peer que actúa como *rendezvous* para un grupo de peers dado.

3.7 Servicios de grupo de peers

Un grupo de peers provee una serie de servicios conocidos como servicios de grupo de peers. Los estándares para servicios de grupo de peers más comunes son [34]:

- Servicio de descubrimiento: Es utilizado por los miembros de un grupo de peers para buscar recursos como: grupos de peers, pipes, servicios y nodos.
- Servicio de afiliación: Es utilizado por los miembros de un grupo de peers para establecer identidades que permitan relaciones de confianza. Las identidades permiten a la aplicación y servicios determinar quien está enviando la petición de una operación y decide si esta operación debería ser permitida. Las aplicaciones pueden desempeñar su propio control de acceso o pueden usar el servicio de acceso. La definición de grupo especifica las políticas por las cuales un nodo puede establecer una identidad.
- Servicio de acceso: Es utilizado para validar peticiones hechas entre dos peers. El receptor de la petición chequea las credenciales del emisor que envía la petición y la información acerca de la petición será hecha para determinar si el acceso esta permitido.
- Servicio de pipe: Es utilizado para crear y administrar las conexiones entre los miembros de un grupo de peers.
- Servicio de monitoreo: Es utilizado para permitir que un nodo monitoree otros miembros del mismo grupo de peers.

Con el objetivo de modelar e implementar la extensión de la plataforma para dar soporte al sistema de provisión de servicios, a continuación se realiza un estudio de las diferentes tecnologías de gestión de redes, con el objetivo de identificar la plataforma de gestión que mejor se adapte a los requisitos de este proyecto.

Anexo 4. Especificaciones técnicas – Agenda colaborativa P2P

Anteriormente se definió los mecanismos básicos para la gestión de aplicaciones P2P mediante la extensión de una plataforma que soporta un sistema para la provisión de servicios P2P, ahora lo queda es llevarlo a un caso práctico en el cual se utilice dicha extensión para gestión de servicios creados en una aplicación particular que en nuestro caso será una Agenda Colaborativa, la cual sea útil para la comunicación, organización y revisión de un grupo de trabajo.

4.1 Actores de la aplicación

Usuario: Es el encargado de utilizar los diferentes servicios que le ofrece la aplicación P2P (Agenda).

4.2 Diagrama inicial de casos de uso de la aplicación p2p (agenda colaborativa)

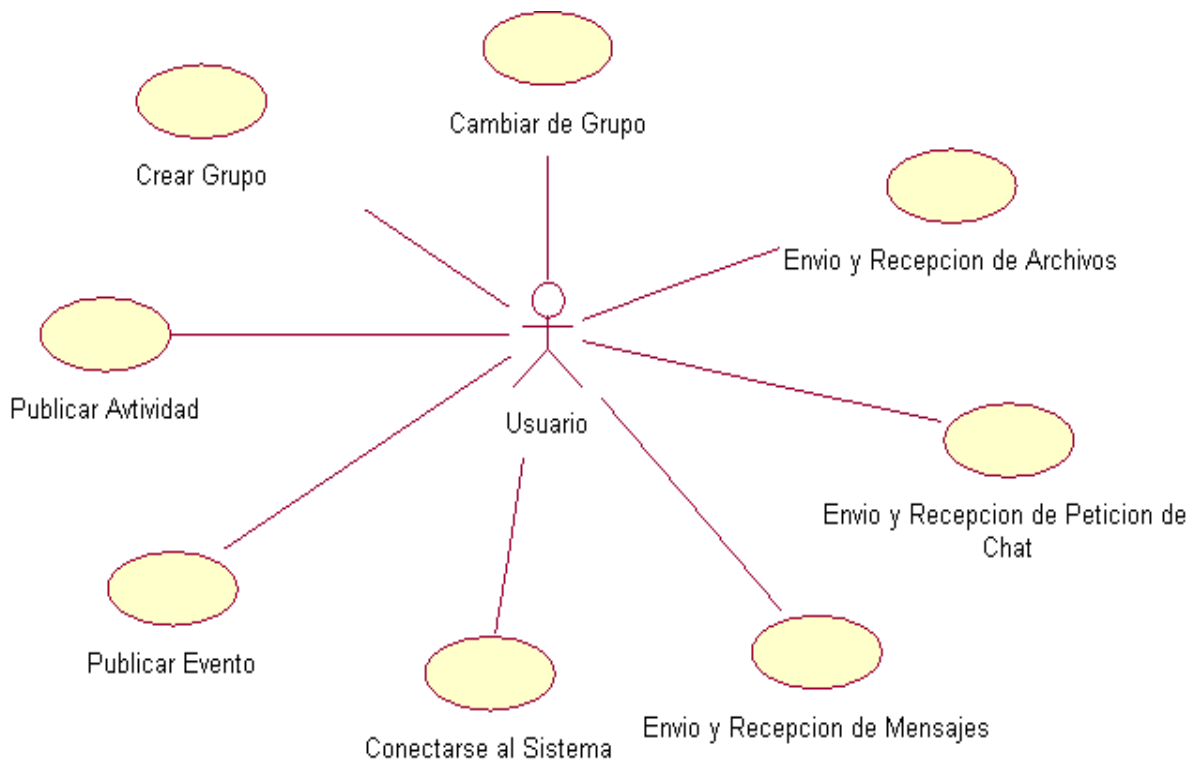


Figura 3. Diagrama Inicial de Casos de Uso Agenda P2P

En la Figura 3 se observa los casos de uso iniciales para el desarrollo del prototipo de la Agenda P2P. Para la implementación de la gestión de servicios de la agenda debemos implementar el caso de uso: Emitir notificaciones.

4.3 Relación de los casos de uso

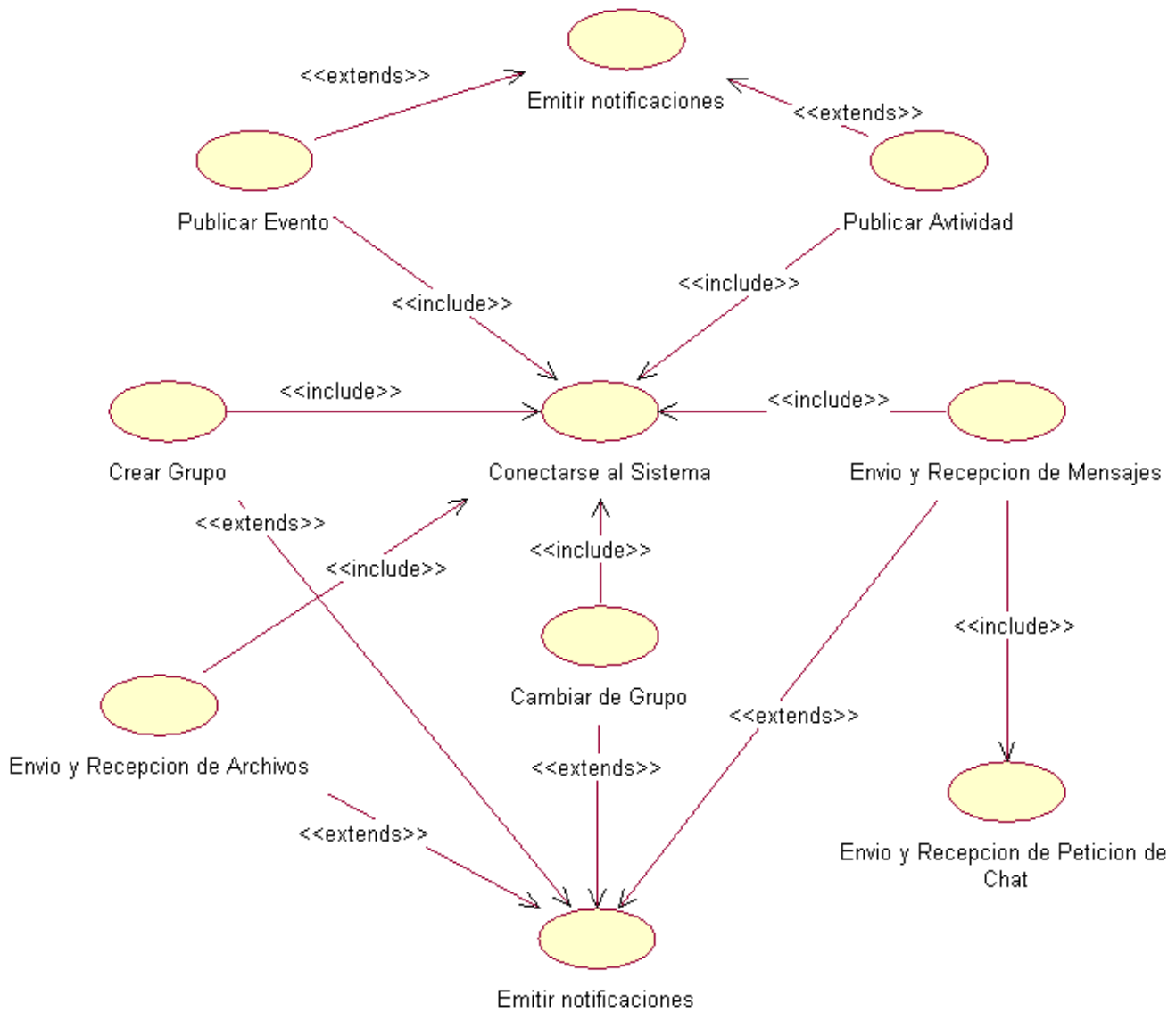


Figura 4. Diagrama Relaciones de Casos de Uso

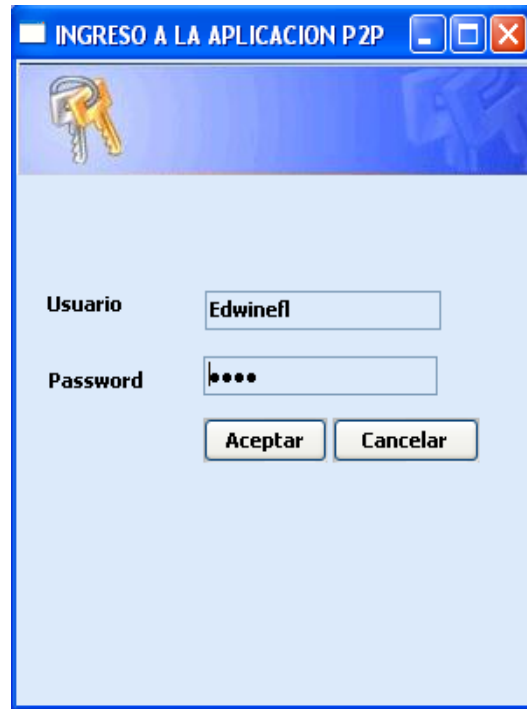
La Figura 4 representa la relación entre los principales casos de uso que componen la Agenda P2P.

4.4 Especificación formal casos de uso

4.4.1 Caso de uso conectarse a la aplicación

Código	Nombre			
CUS GE- 001	Caso de uso conectarse a la aplicación.			
Resumen	Esta funcionalidad permite al usuario ingresar a la aplicación.			
Prioridad: Alta.	Complejidad: Media.			
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Emitir notificaciones			
Precondiciones				
1	Debe existir una conexión válida.			
Postcondiciones				
1	El usuario ha ingresado a la agenda, visualiza una Interfaz con todas sus funcionalidades.			
Flujo Básico				
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Despliega una pantalla que permite realizar el ingreso del nombre de usuario y contraseña. Y dos botones de ingreso y cancelación.	<ul style="list-style-type: none"> Nombre Usuario Contraseña 	
2	El usuario ingresa su nombre de usuario y contraseña, oprime el botón de ingreso.	El sistema valida que los datos son obligatorios, de lo contrario ir a EX01.		
3	Fin.			
Excepciones				
Código:	Nombre:			
EX01				
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1		El sistema valido que los campos obligatorios sean digitados, de lo contrario muestra el mensaje: "Campo no puede ser nulo".		<ul style="list-style-type: none"> Mensaje: "Campo no puede ser nulo"

Prototipo de pantalla



The image shows a screenshot of a login dialog box titled "INGRESO A LA APLICACION P2P". The dialog box has a blue header bar with a key icon on the left and standard window control buttons (minimize, maximize, close) on the right. Below the header, the main area is light blue. It contains two input fields: "Usuario" with the text "Edwinefl" and "Password" with four black dots. At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Tabla 1. Caso de uso: Conectarse a la aplicación.

4.4.2 Caso de uso crear grupo

Código		Nombre		
GE-002		Crear un grupo		
Resumen		Esta funcionalidad permite al usuario crear un grupo.		
Prioridad: Alta.		Complejidad: Alta.		
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Emitir notificaciones.			
Precondiciones				
Postcondiciones				
1	El grupo se creó, el usuario se une al grupo para visualizar los usuarios.			
Flujo Básico				
Acción del Actor		Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Provee una opción de menú que despliegue una pantalla la cual permite realizar el ingreso de datos necesarios para la creación del grupo como son nombre, descripción y una contraseña. Además un botón de aceptar para concluir dicha acción y un botón cancelar para desistir de la creación del grupo.	<ul style="list-style-type: none"> Nombre Grupo Descripción Contraseña 	
2	El usuario ingresa los datos solicitados para la creación del grupo y oprime el botón aceptar.	El sistema valida los datos; si son validos con éxito se crea el grupo y asocia al usuario a dicho grupo. Si el grupo ya existe, ir a EX01. El sistema valida que los datos son obligatorios, de lo contrario ir a EX02.		
3	Fin.			


Excepciones				
Código:		Nombre:		
EX01		Validación no exitosa.		
Acción del Actor		Acción del Sistema		Datos de Salida
1		Si el sistema no valida con éxito al grupo muestra un mensaje "Grupo ya existe"		<ul style="list-style-type: none"> Mensaje: "Grupo ya existe"
2	Dar clic en Aceptar e intentar de nuevo.			
EX02		Campo no puede ser nulo		
Acción del Actor		Acción del Sistema		Datos de Salida
1		El sistema valido que los campos obligatorios sean digitados, de lo contrario muestra el mensaje: "Campo no puede ser nulo".		<ul style="list-style-type: none"> Mensaje: "Campo no puede ser nulo"
2	Dar clic en Aceptar e intentar de nuevo.			
Prototipo de pantalla				
				

Tabla 2. Caso de uso: Crear grupo.

4.4.3 Caso de uso cambio de grupo


Código	Nombre			
GE-003	Cambio de grupo			
Resumen	Esta funcionalidad permite al usuario cambiar de grupo.			
Prioridad: Alta.		Complejidad: Alta.		
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Emitir notificaciones.			
Precondiciones				
Postcondiciones				
1	El usuario ha cambiado de grupo, visualiza una Interfaz con el nombre del nuevo grupo.			
Flujo Básico				
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Desplegar una pantalla que permita seleccionar un grupo a unirse y realizar el cambio.	<ul style="list-style-type: none"> Nombre Grupo 	
2	El usuario selecciona el grupo al que desea cambiarse.	El sistema asocia el usuario al grupo que selecciono.		
3	Fin.			
Excepciones				
Código:		Nombre:		
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
Prototipo de pantalla				
				

Tabla 3. Caso de uso: Cambio de grupo.

4.4.4 Caso de uso envío y recepción de petición de chat

Código		Nombre		
GE-004		Envío y recepción de petición de chat		
Resumen		Esta funcionalidad permite al usuario enviar una petición de chat a cualquier usuario y detectar por cualquier petición entrante		
Prioridad: Alta.		Complejidad: Alta.		
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
Precondiciones				
Postcondiciones				
1	El usuario ha enviado una petición con éxito o ha detectado una petición de chat.			
Flujo Básico				
Acción del Actor		Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Desplegar una menú en pantalla que le permita seleccionar al usuario el envió de petición de chat.	<ul style="list-style-type: none"> Usuario Destino 	
2	El usuario oprime el botón de envío de petición.	<p>El sistema valida al usuario destino si es válido envía la petición y espera por la respuesta.</p> <p>Si es detectada una Petición de chat se lanza una interfaz de respuesta en donde el usuario puede escoger aceptar o no.</p> <p>Si el usuario destino es invalido, ir a EX01.</p>		
3	Fin de caso de uso			
Excepciones				
Código:		Nombre:		
EX01		Validación no exitosa		
Acción del Actor		Acción del Sistema	Datos de Entrada	Datos de Salida
1		Si el sistema no valida con éxito al usuario muestra un mensaje "Usuario no existe en el sistema"		<ul style="list-style-type: none"> Mensaje: "Usuario no existe en el sistema"
2	Dar clic en Aceptar e intentar de nuevo.			
Prototipo de pantalla				

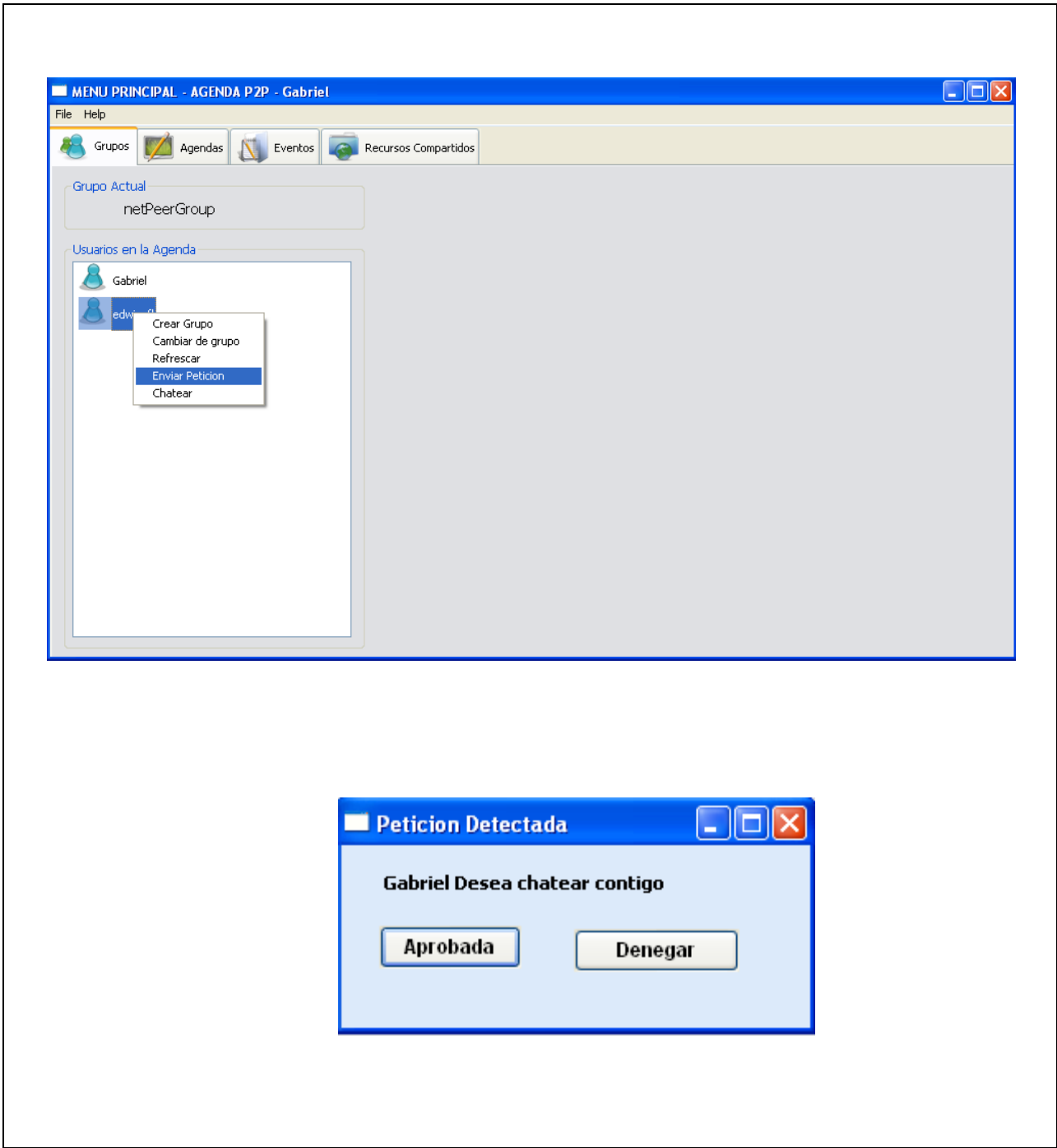


Tabla 4. Caso de uso: Envió y recepción de petición de chat.

4.4.5 Caso de uso envío y recepción de mensajes

Código	Nombre			
GE-005	Envío y recepción de mensajes			
Resumen	Esta funcionalidad permite al usuario enviar y recibir cualquier mensaje a cualquier Usuario de la Aplicación que haya aceptado una petición de chat previamente.			
Prioridad: Alta.		Complejidad: Alta.		
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Envío y recepción de petición de chat			
2	Emitir notificaciones			
Precondiciones				
1	Ya se han lanzado el caso de uso de envío y recepción de petición de chat.			
2	La Petición de chat ha sido aceptada.			
Postcondiciones				
1	El usuario recibe la respuesta del mensaje enviado y envía un mensaje sin error.			
Flujo Básico				
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Desplegar una pantalla que permita seleccionar el usuario destino realizar el ingreso de mensajes y un botón de envío.	<ul style="list-style-type: none"> Mensaje 	
2	El Usuario escribe el mensajes y oprime el botón de envío	El sistema envía el mensaje al nodo destino y está listo para recibir cualquier respuesta.		
3	Fin de caso de uso			
Excepciones				
Código:		Nombre:		
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida

Prototipo de pantalla

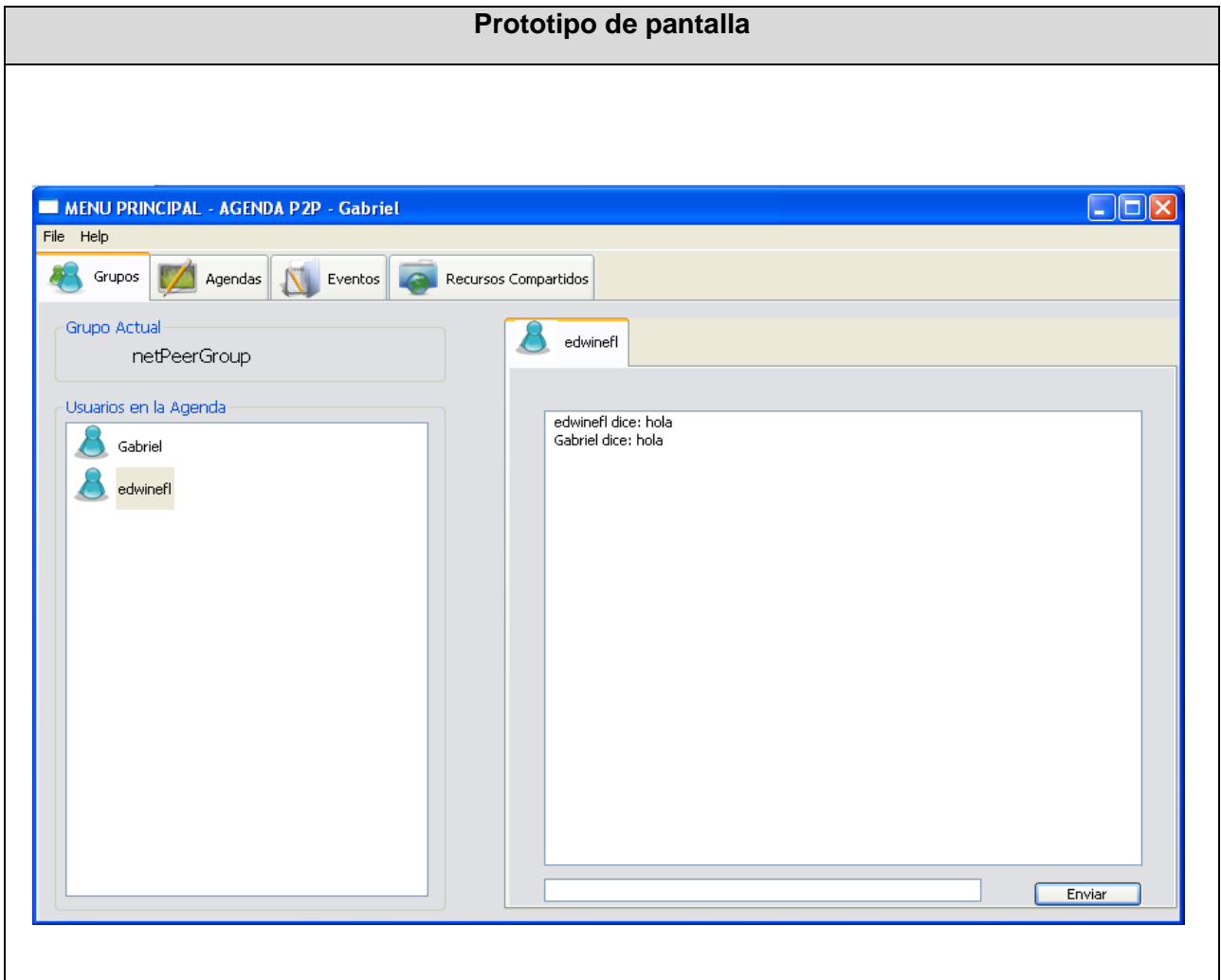


Tabla 5. Caso de uso: Envió y recepción de mensajes.

4.4.6 Caso de uso envío y recepción de archivos

Código	Nombre			
GE-006	Envío y Recepción de Archivos			
Resumen	Esta funcionalidad permite al usuario enviar y recibir archivos			
Prioridad: Alta.		Complejidad: Alta.		
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Emitir notificaciones			
Precondiciones				
1	El archivo debe estar cargado en memoria para poder enviarse.			
Postcondiciones				
1	El sistema visualiza una Interfaz el archivo enviado.			
Flujo Básico				
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Desplegar una lista con los archivos compartidos actualmente, y un botón para enviar el archivo a los miembros del grupo.		
2	El usuario oprime el botón	El sistema despliega en pantalla un mensaje de confirmación si el archivo fue enviado con éxito. El sistema despliega en pantalla un mensaje y refresca la lista de archivos compartidos si detecta algún archivo enviado por algún miembro del grupo.		
3	Fin de caso de uso			
Excepciones				
Código:		Nombre:		
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida

Prototipo de pantalla

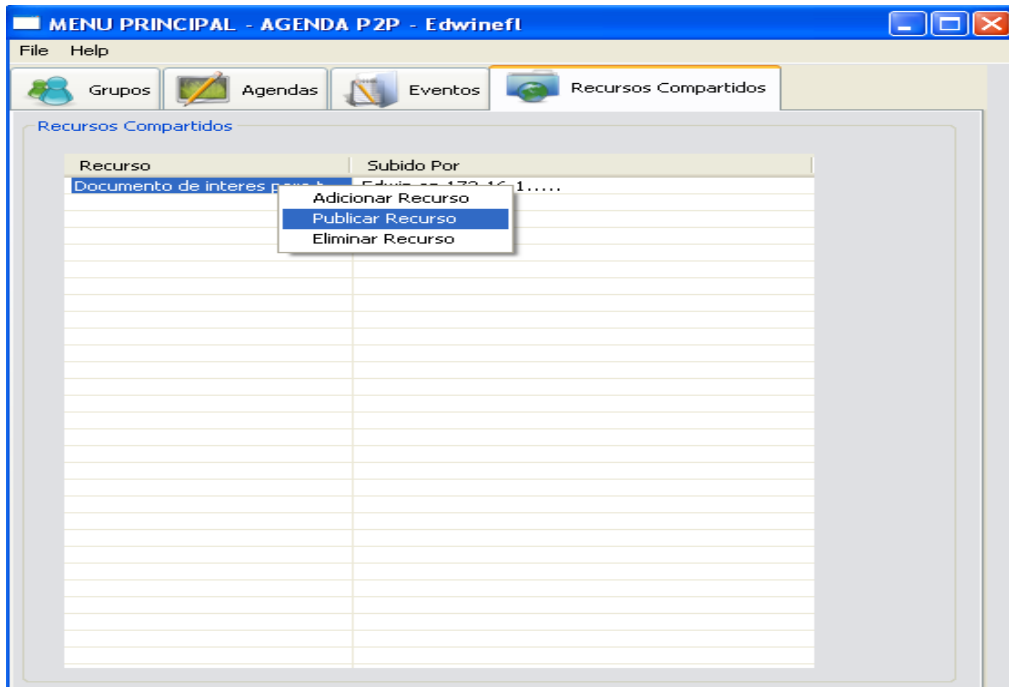
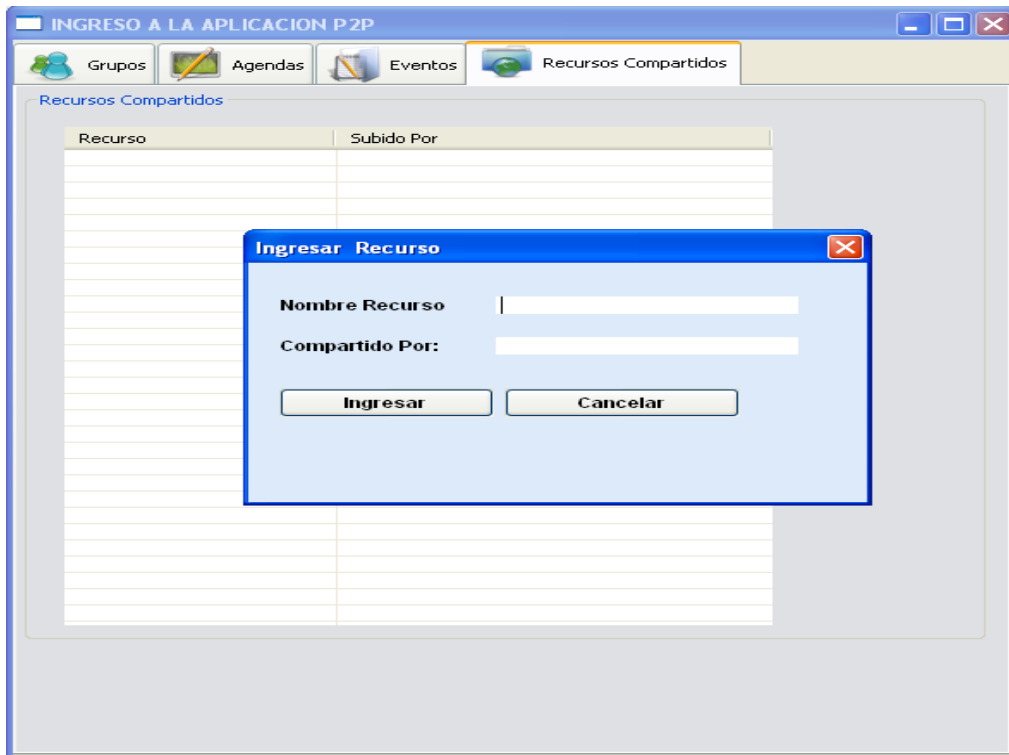


Tabla 6. Caso de uso: Envío y recepción de archivos.

4.4.7 Caso de uso publicar actividad

Código		Nombre		
USO GE-007		Publicar actividad		
Resumen		Esta funcionalidad permite al usuario ingresar y publicar una actividad sobre una agenda a todos los usuarios conectados.		
Prioridad: Alta.		Complejidad: Alta.		
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Emitir notificaciones			
Precondiciones				
Postcondiciones				
1	Los usuarios asociados a la agenda pueden conocer las actividades publicadas.			
Flujo Básico				
Acción del Actor		Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Desplegar en pantalla la funcionalidad requerida para crear actividades y publicarlas sobre una agenda establecida por un usuario o por varios usuarios		
2	El usuario selecciona la actividad y la publica a los otros usuarios.	El sistema publica las actividades que el usuario desea que se publiquen. Si las actividad no se pudo publicar ir EX01.		
3	Fin de caso de uso			
Excepciones				
Código:		Nombre:		
EX01		Publicación no exitosa.		
Acción del Actor		Acción del Sistema	Datos de Entrada	Datos de Salida
1		El sistema no publica la actividad, emite el mensaje "publicación no exitosa"		<ul style="list-style-type: none"> Mensaje Publicación no exitosa.
2	Dar clic en publicar e intentar de nuevo.			

Prototipo de pantalla

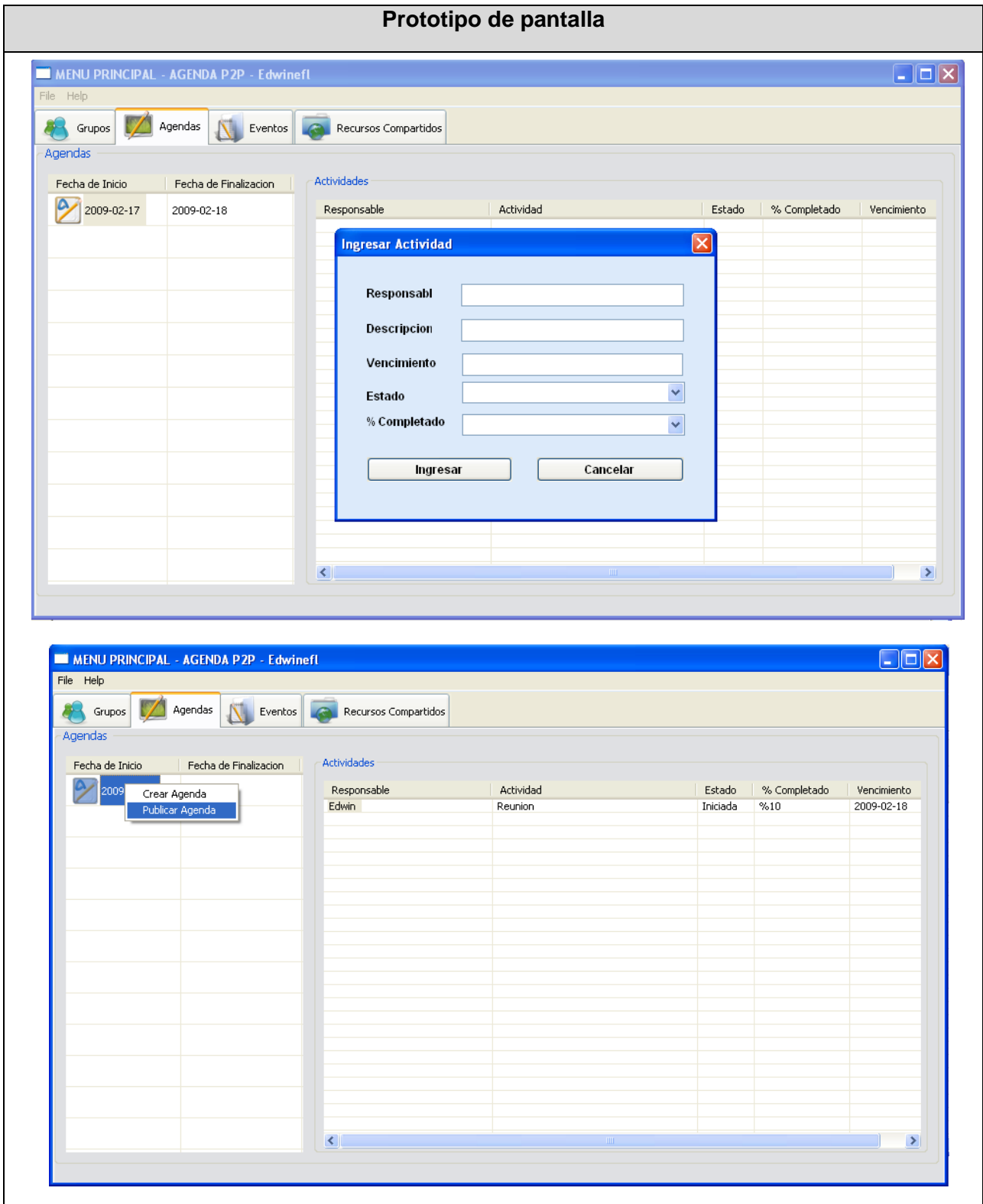


Tabla 7. Caso de uso: Publicar actividad.

4.4.8 Caso de uso publicar evento

Código	Nombre			
GE-008	Publicar evento			
Resumen	Esta funcionalidad permite al usuario ingresar y publicar un evento.			
Prioridad: Alta.	Complejidad: Alta.			
Actor(es) Involucrado(s)				
1	Usuario de la aplicación			
Caso(s) de Uso Relacionado(s)				
1	Emitir notificaciones			
Precondiciones				
Postcondiciones				
1	Los usuarios asociados a la agenda pueden conocer los eventos publicados.			
Flujo Básico				
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1	Ingresar al caso de uso	Desplegar en pantalla la funcionalidad requerida para crear eventos y publicarlos a otros usuarios.		
2	El usuario selecciona el evento y lo publica a los otros usuarios.	El sistema publica el evento. Si el evento no se pudo publicar ir EX01.		
3	Fin de caso de uso			
Excepciones				
Código:		Nombre:		
EX01		Publicación no exitosa.		
	Acción del Actor	Acción del Sistema	Datos de Entrada	Datos de Salida
1		El sistema no publica el evento, emite el mensaje "publicación no exitosa"		<ul style="list-style-type: none"> Mensaje Publicación no exitosa.
2	Dar clic en publicar e intentar de nuevo.			

Prototipo de pantalla

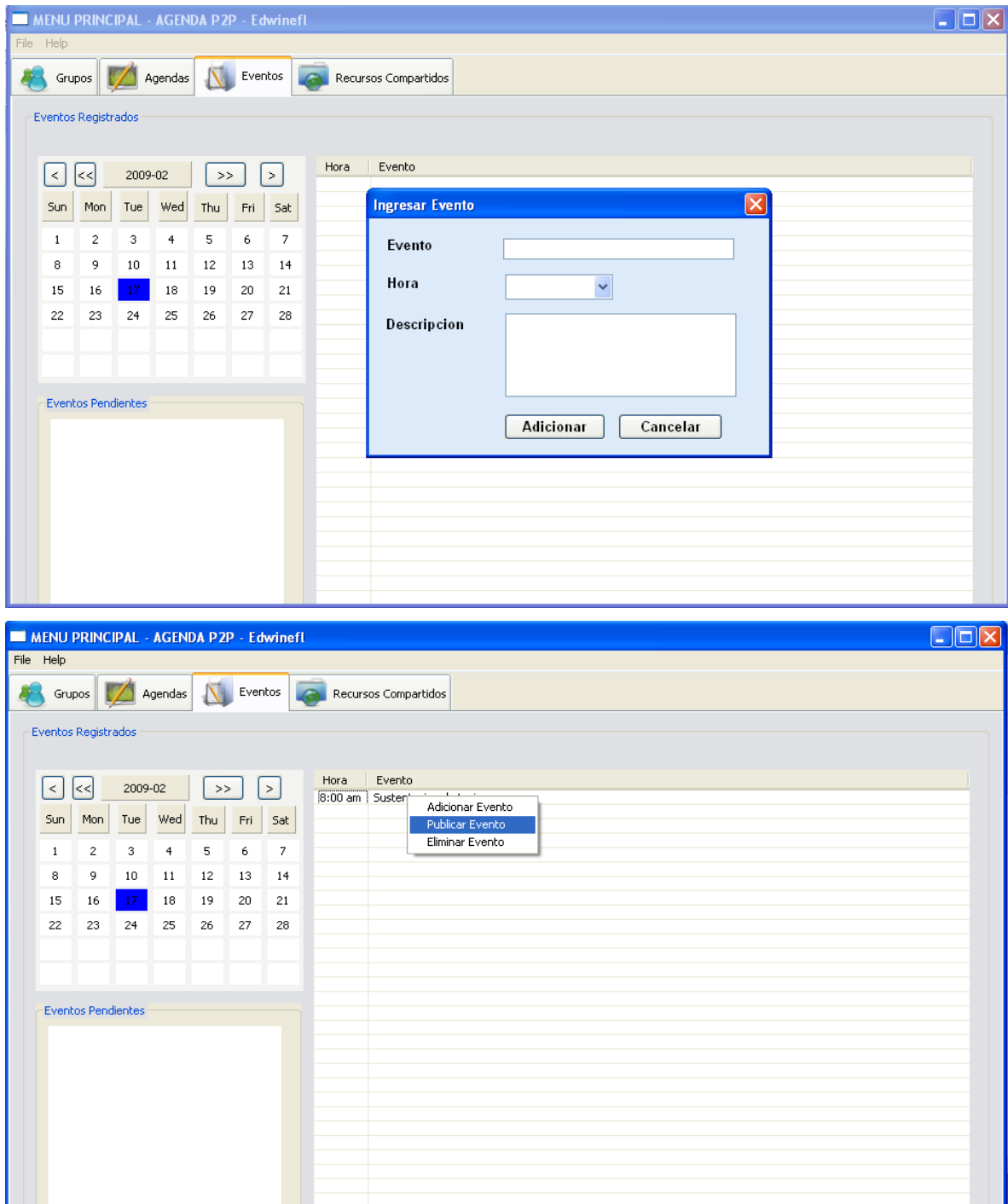


Tabla 8. Caso de uso: Publicar evento.

4.4.9 Caso de uso emitir notificaciones

Código	Nombre		
GE-009	Emitir notificaciones		
Resumen	Para que los servicios puedan definir su modelo de gestión de manera adecuada describen y emiten notificaciones de gestión.		
Prioridad: Alta.		Complejidad: Alta.	
Actor(es) Involucrado(s)			
1	Usuario de la aplicación		
Caso(s) de Uso Relacionado(s)			
1	Definir información de gestión		
Precondiciones			
	El servicio debe estar en ejecución.		
Postcondiciones			
Flujo Básico			
	Acción del Actor	Acción del Sistema	Datos de Salida
1	Ingresar al caso de uso	Emite notificaciones de gestión asociadas a un servicio.	
2	Fin de caso de uso		
Excepciones			
Código:		Nombre:	
	Acción del Actor	Acción del Sistema	Datos de Salida

Tabla 9. Caso de uso: Emitir notificaciones.

4.5 Diagramas de Secuencia

4.5.1 Caso de uso GE-001 conectarse a la aplicación

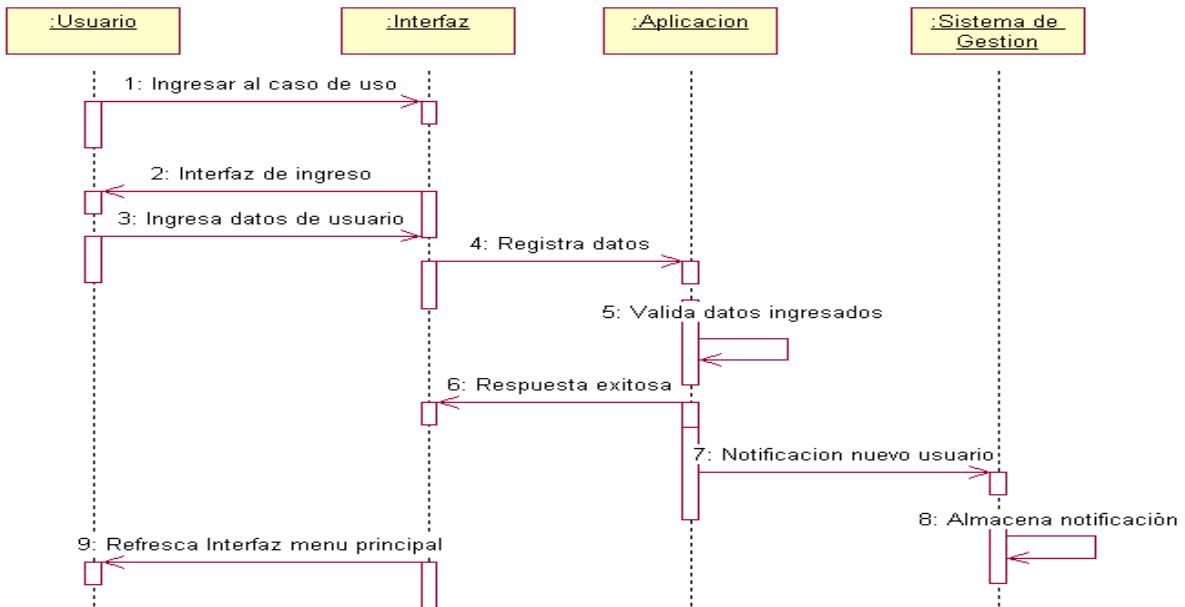


Figura 5. Diagrama de secuencia conectarse al sistema

La figura 5 representa la secuencia lógica del caso de uso conectarse al sistema.

4.5.2 Caso de uso GE-002 crear grupo

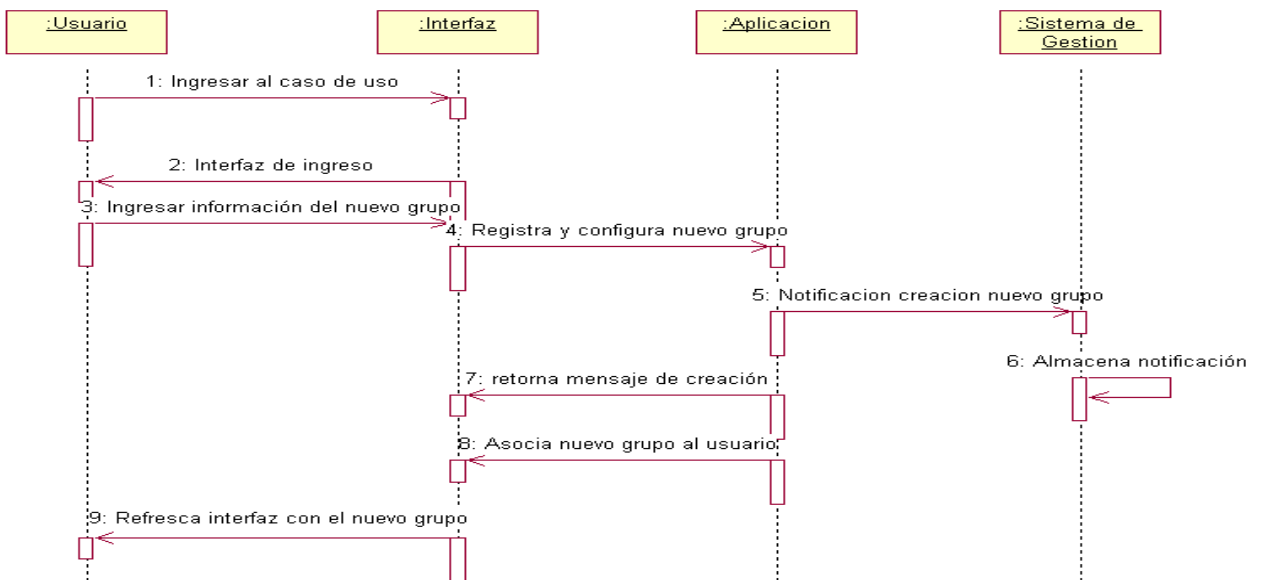


Figura 6. Diagrama de secuencia crear grupo.

La figura 6 representa la secuencia lógica del caso de uso crear grupo.

4.5.3 Caso de uso GE-003 cambio de grupo

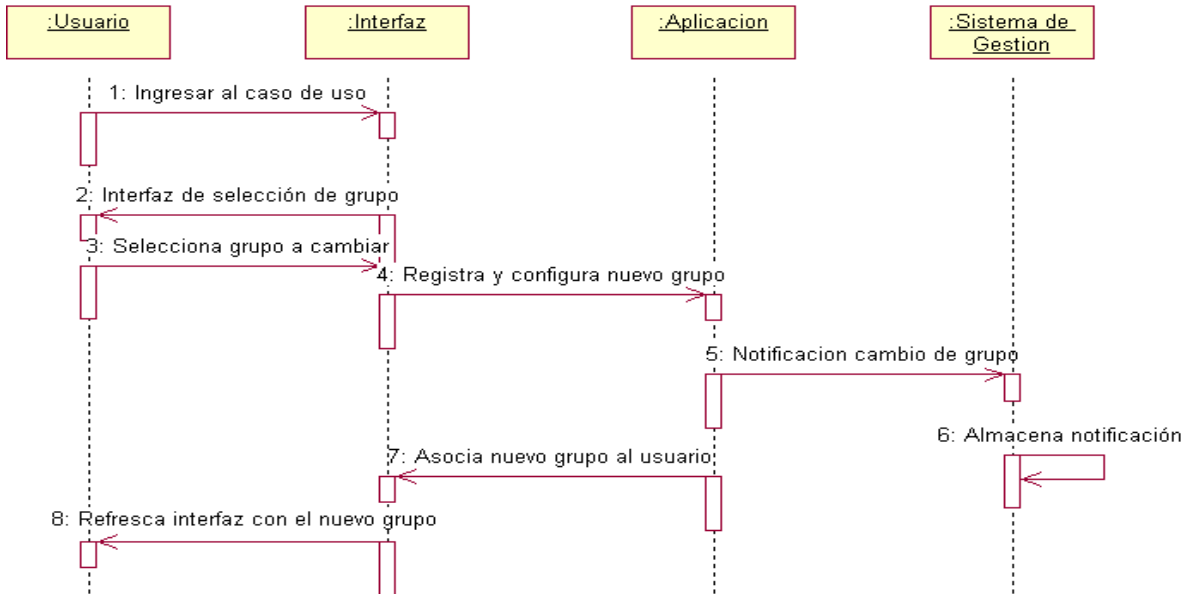


Figura 7. Diagrama de secuencia cambio de grupo.

La figura 7 representa la secuencia lógica del caso de uso cambio de grupo.

4.5.4 Caso de uso GE-004 envío y recepción de petición de chat

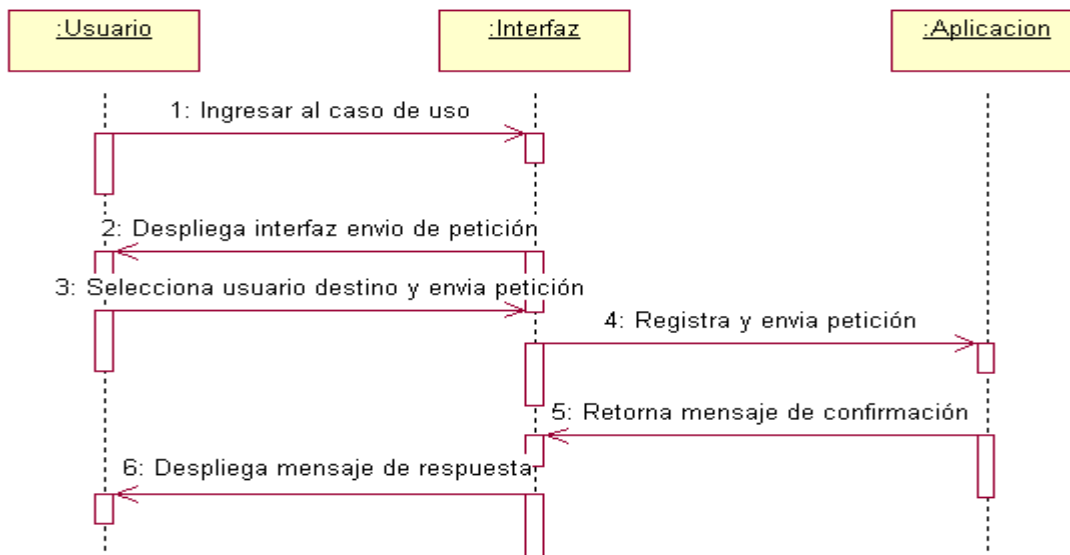


Figura 8. Diagrama de secuencia envío y recepción de petición de chat

La figura 8 representa la secuencia lógica del caso de uso envío y recepción de petición de chat

4.5.5 Caso de uso GE-005 envío y recepción de mensajes

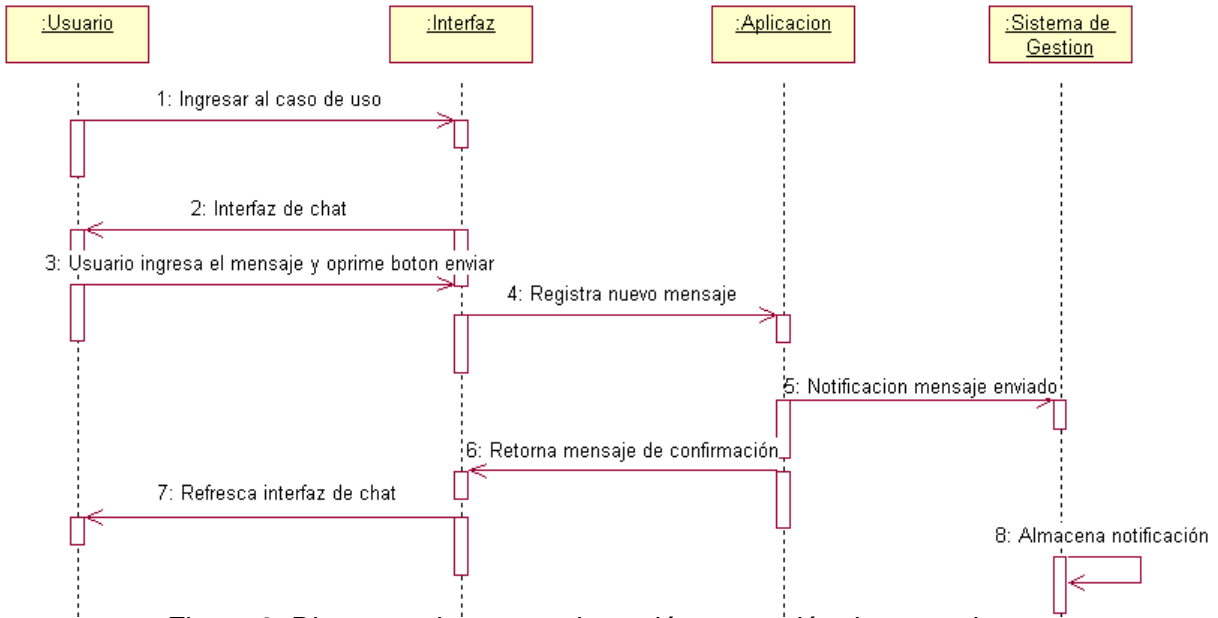


Figura 9. Diagrama de secuencia envío y recepción de mensajes.

La figura 9 representa la secuencia lógica del caso de uso envío y recepción de mensajes.

4.5.6 Diagrama de secuencia detección de mensaje entrante

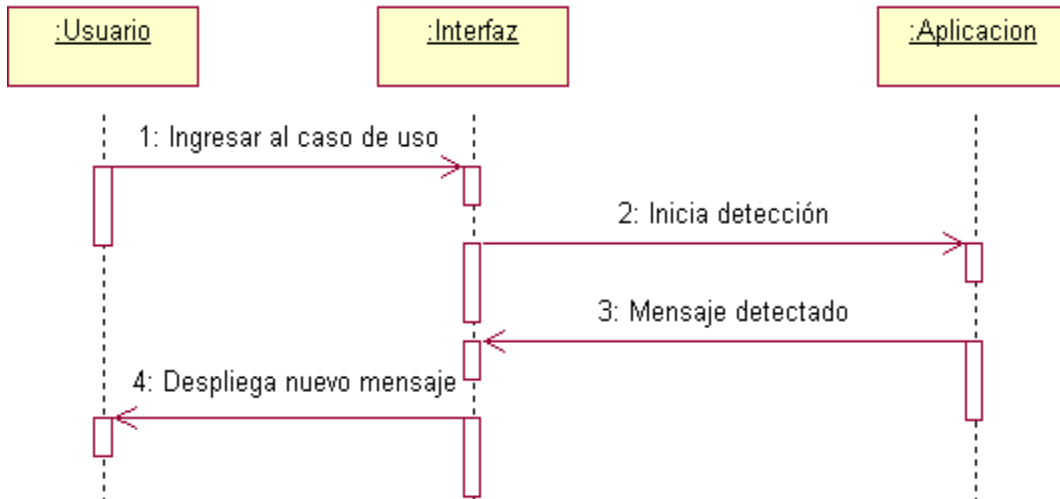


Figura 10. Diagrama de secuencia detección de mensaje entrante.

La figura 10 representa la secuencia lógica del caso de uso detección de mensaje entrante.

4.5.7 Caso de uso GE-006 envío y recepción de archivos

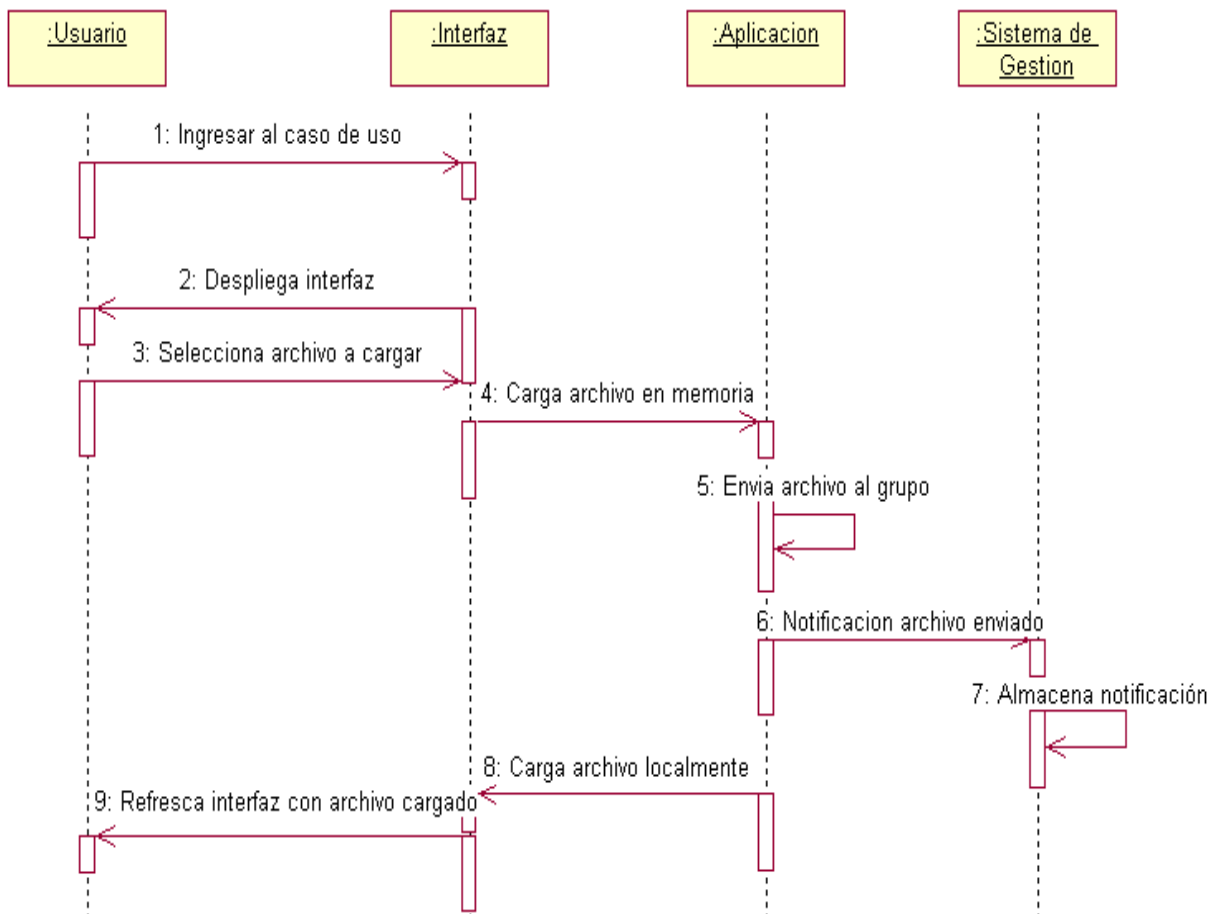


Figura 11. Diagrama de secuencia envi6 y recepci6n de archivos.

La figura 11 representa la secuencia l6gica del caso de uso envi6 y recepci6n de archivos.

4.5.8 Caso de uso GE-007 publicar actividad

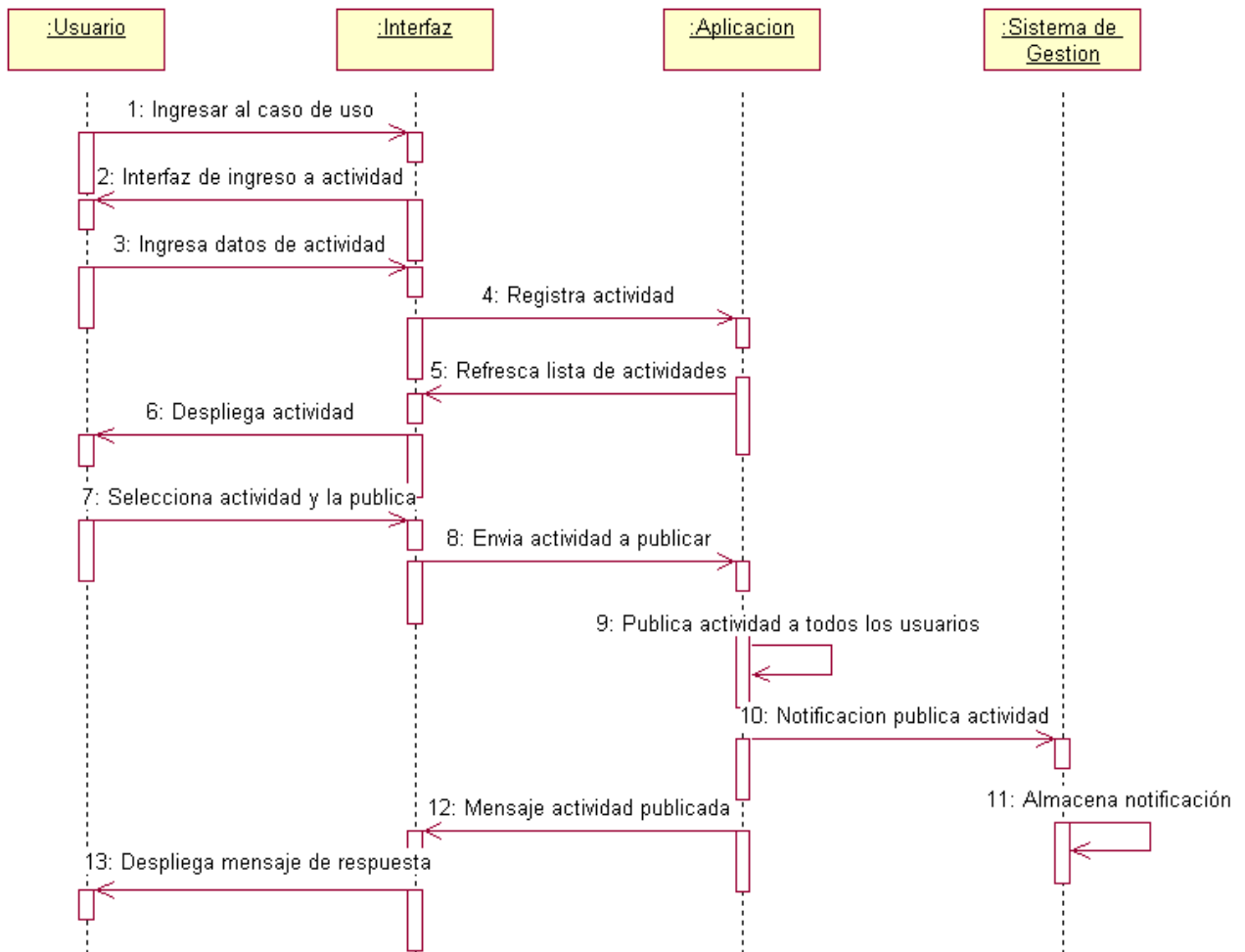


Figura 12. Diagrama de secuencia publicar actividad

La figura 12 representa la secuencia lógica del caso de uso publicar actividad.

4.5.9 Caso de uso GE-008 publicar evento

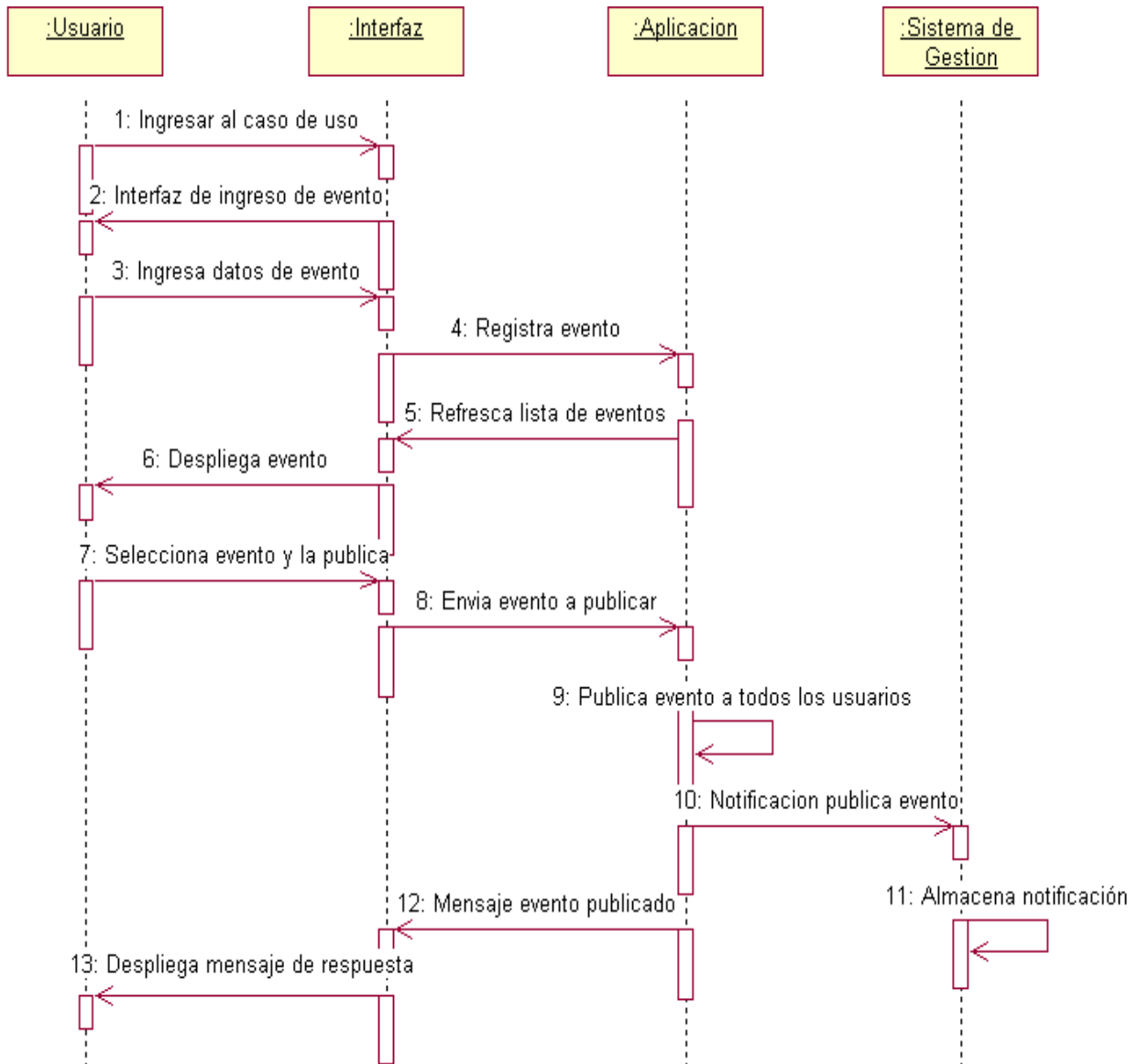


Figura 13. Diagrama de secuencia publicar evento.

La figura 13 representa la secuencia lógica del caso de uso publicar evento.

4.5.10 Caso de uso GE-009 emitir notificaciones

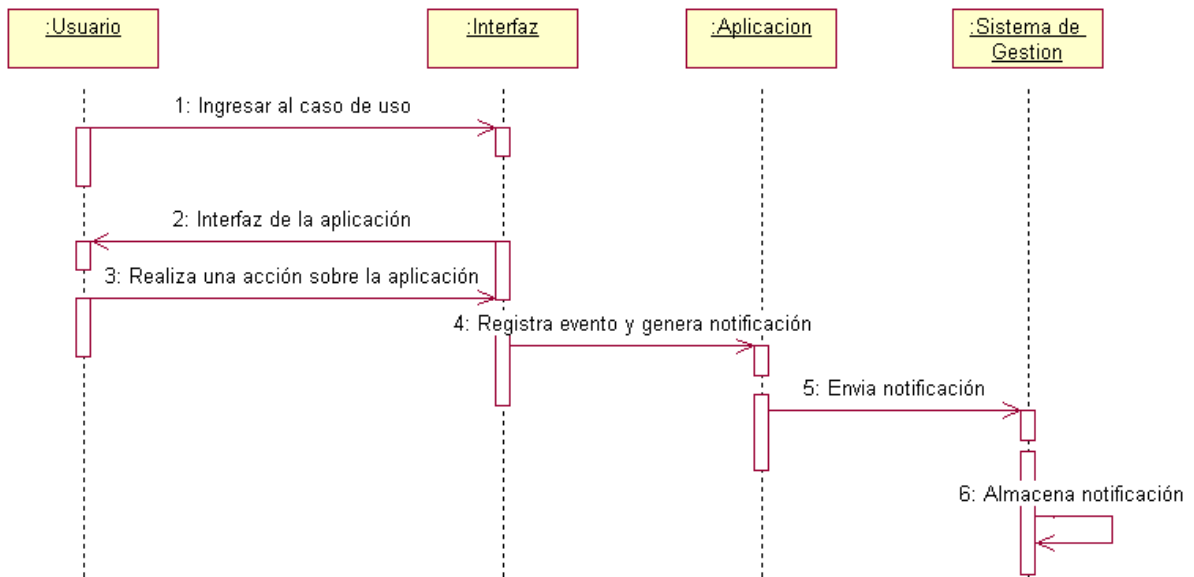


Figura 14. Diagrama de secuencia emitir notificaciones.

La figura 14 representa la secuencia lógica del caso de uso emitir notificaciones.

4.6 Diagrama de clases de la aplicación p2p (Agenda Colaborativa)

La siguiente figura representa las principales clases e interfaces que componen la aplicación P2P (Agenda), para proveer las funcionalidades centrales del mismo, ilustrando los componentes de la plataforma de servicios.

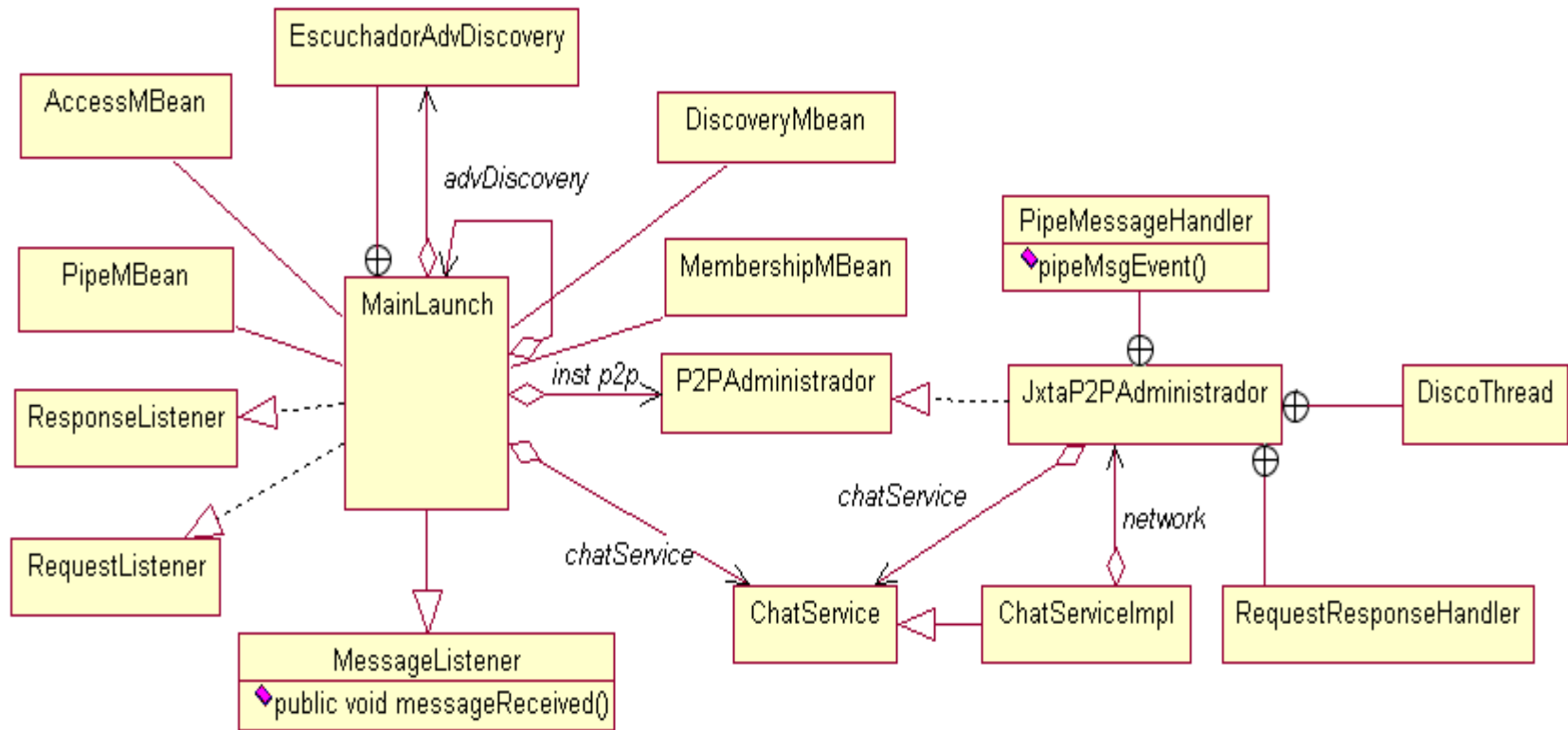


Figura 15. Diagrama de clases general de la aplicación P2P (Agenda Colaborativa)