

ComAppÑeros

Especificación de Requisitos Software

IEEE Std. 830-1998

Sebastián Águila
Diego del Corral
Miguel Franco
Mario Michiels
Rodrigo de Miguel
Álvaro Navas
Sergio Pino
Cristian Pinto
Emanuel Ramírez
Fernando Rivilla
Cristhian Rodríguez



ComAppÑeros

Universidad Complutense de Madrid

Gestión de Proyectos Software y Metodologías de Desarrollo

Historial de versiones

Versión	Fecha
SRS-ComAppÑeros-v0.0.1	21/03/2016
SRS-ComAppÑeros-v0.1.1	04/04/2016
SRS-ComAppÑeros-v0.1.2	08/04/2016
SRS-ComAppÑeros-v0.1.3	25/05/2016

Tabla 0: Control de versiones

Índice

1 Introducción	4
1.1 Propósito	4
1.2 Alcance	4
1.3 Definiciones, Acrónimos y Abreviaturas	4
1.4 Referencias	5
1.5 Resumen	6
2 Descripción general	7
2.1 Perspectiva del producto	7
2.2 Funciones del producto	7
2.2.1 Funciones Generales	7
2.2.2 Buscar piso	9
2.2.3 Buscar inquilino	9
2.2.4 Administrador	9
2.3 Características de los usuarios	10
2.4 Restricciones	10
2.5 Suposiciones y dependencias	10
2.6 Requisitos futuros	10
3 Requisitos específicos	11
3.1 Interfaces externas	11
3.1.1 Interfaces de usuario	11
3.1.2 Interfaces de Software	13
3.1.3 Interfaces de Hardware	13
3.2 Requisitos Funcionales	14
3.2.1 Requisitos Generales	14
3.2.2 Requisitos de Buscar Piso	58
3.2.3 Requisitos de Buscar inquilino	68
3.2.4 Requisitos de Adiministración	78
3.3 Requisitos de Rendimiento	84
3.4 Requisitos lógicos de la BBDD	84
3.5 Restricciones de diseño	85
3.6 Atributos del sistema	86

1. Introducción

En el presente documento se analizan y detallan en gran detalle los requisitos y funciones del sistema ComAppÑeros.

1.1. Propósito

El propósito de este documento es hacer la especificación de requisitos (SRS) siguiendo el estándar IEEE Std. 830-1998 para que nuestro sistema permita el correcto funcionamiento de una aplicación para buscar piso o un compañero con el que compartir uno. Así mismo, el documento es una guía tanto para el equipo de desarrollo de ComAppÑeros como para el cliente que podrá realizar una validación del producto final y conocer si la implementación de las funciones es la pactada.

1.2. Alcance

Nuestro sistema se llama “ComAppÑeros” y su principal objetivo es facilitar la búsqueda de compañeros de piso adecuados, así como de los pisos más indicados para cada usuario mediante la aplicación de filtros específicos. La aplicación será de plataforma web responsive, para facilitar el uso desde dispositivos móviles.

Dicha aplicación ofrecería la posibilidad de registrarse una única vez por usuario, así como de poder dar de baja la cuenta.

Se tendrá la posibilidad de priorizar a unos usuarios sobre otros mediante una verificación de sus perfiles en la aplicación usando las redes sociales (Facebook, o Google+). También se desea implantar un catálogo tanto de usuarios que buscan ser inquilinos, como de pisos ofertados, en el que se pueda consultar la información disponible, de forma que se facilite el objetivo principal de la aplicación.

Los datos publicados a través del catálogo deberán estar actualizados frecuentemente, siendo deseable que se sincronicen periódicamente con las bases de datos internas, con la finalidad de satisfacer la experiencia de usuario ofreciendo información de calidad actualizada y filtrada en cualquier momento.

1.3. Definiciones, Acrónimos y Abreviaturas

- **SW:** Software.
- **IEEE:** Institute of Electrical And Electronics Engineers (Instituto De ingenieros eléctricos y electrónicos).

- **Compañeros potenciales:** Retrata la relación de querer compartir piso de forma consentida por ambas personas.
- **Compañeros actuales:** Establece que ambas personas viven actualmente en el mismo piso.
- **Inquilino:** Persona que vive en un piso.
- **Propietario:** Dueño de un piso. Puede o no ser inquilino del mismo.
- **BBDD:** Base de datos. Almacén o banco de información es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- **Tag:** también denominado etiqueta, es una palabra clave que se usa para describir de manera corta y precisa algún elemento.
- **API:** abreviatura de Interfaz de Programación de Aplicaciones.
- **MongoDB:** MongoDB (de la palabra en inglés “humongous” que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.
- **RRSS:** redes sociales.
- **Drupal:** es un marco de gestión de contenidos o CMS (por sus siglas en inglés, Content Management System) libre, modular, multipropósito y muy configurable que permite publicar artículos, imágenes, archivos y que también ofrece la posibilidad de otros servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.
- **XAMPP:** XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- **MySQL:** MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo^{1 2}, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.
- **GUI:** la interfaz gráfica de usuario.
- **Geolocalización:** La geolocalización o georreferenciación es un concepto relativamente reciente que hace referencia a la identificación de una situación geográfica de manera automática.
- **Filezilla:** FileZilla es un cliente FTP multiplataforma de código abierto y software libre, licenciado bajo la Licencia Pública General de GNU. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS).

1.4. Referencias

Nuestra referencia es el Standard de IEEE 830-1998, 20 Oct. 1998 de la IEEE Computer Society así como apuntes y material audiovisual de la asignatura Gestión de Proyectos Software y Metodologías de Desarrollo (GPS) y sus predecesoras, Ingeniería del Software (IS) y Modelado de Software (MS).

1.5. Resumen

Inicialmente se determina el ámbito de la aplicación, especificando como se integra en el sistema. En secciones posteriores se identifican funciones principales del sistema de forma muy general así como restricciones y los usuarios a los que va destinada.

Finalmente se hace un análisis exhaustivo de los requisitos haciendo una partición del sistema en las funciones identificadas, así como una visión de cómo podrían ser las interfaces externas de la misma.

2. Descripción general

El software facilitará la búsqueda de compañeros de piso adecuados, así como de los pisos más indicados para cada usuario mediante la aplicación de filtros específicos en base a la preferencias de los usuarios. Se podrán buscar tanto usuarios como pisos para poder conseguir el compañero ideal de piso.

Los usuarios podrán interactuar entre ellos para hacer una búsqueda conjunta de piso, así como los inquilinos o propietarios de un piso podrán buscar a gente para compartirlo.

También se implementará un sistema de mensajería entre usuarios y de notificaciones con el fin de facilitar y mantener al día a todos los usuarios de la aplicación.

2.1. Perspectiva del producto

El sistema que vamos a desarrollar es para un proyecto universitario de la asignatura "GPS – Gestión de Proyectos Software y Metodologías de Desarrollo" por lo que es independiente ya que no lo relacionamos con otros productos.

2.2. Funciones del producto

2.2.1. Funciones Generales

- **Registro de cuenta:** Los usuarios podrán registrarse en el sistema con sus datos personales, además de poder usar las Redes Sociales (Facebook y Google+). En el registro se determinará el estado actual del usuario, el cual posteriormente puede cambiar: "Busca piso", "Tiene piso y busca compañero", "Es propietario y busca inquilinos".
- **Búsqueda automática por filtro (para piso y usuario):** buscará de forma automática los pisos/usuarios en base a los perfiles de cada uno (ejem: No se mostrarán los pisos de no fumador a un fumador que esté buscando piso).
- **Buscar usuarios por valoración:** Filtra la búsqueda de los usuarios por la valoración indicada.
- **Mostrar información de un usuario:** Muestra la información de un usuario.
- **Modificación de una cuenta:** Permite modificar los datos del usuario.
- **Eliminar cuenta del sistema:** Al usuario se le dará la opción de darse de baja (baja lógica) o de eliminar su cuenta por completo (baja física). Si el usuario elige dar de baja su cuenta, prescindirá así de los servicios que ofrece de la aplicación, pero podrá volver a reactivarla. Eliminar la cuenta provocará la eliminación de todos los datos del usuario en el servidor (Datos de perfil, preferencias, filtros... etc).

- **Listar con filtrado automático usuario/piso:** Da un resultado en forma de lista, con todos los pisos de entre los disponibles con los filtros previamente establecidos, ordenados de mayor/menor valoración.
- **Mostrar información de un piso:** Muestra los datos de un piso.
- **Búsqueda por tag (para piso y para usuario):** Búsqueda en la base de datos del sistema que muestra los usuarios/pisos ofertados mediante tags.
- **Valorar un usuario:** El usuario puede valorar usuarios utilizando un sistema de 0 a 5 estrellas, después de haber concretado el alquiler.
- **Mensajería entre usuarios:** Permite el envío y recepción de mensajes entre un usuario y el buzón de otro usuario.
- **Mostrar información de un usuario:** Muestra los datos de un usuario.
- **Valorar un piso/usuario:** El usuario puede valorar pisos utilizando un sistema de 0 a 5 estrellas, luego de haber concretado el alquiler.
- **Aceptar/Rechazar petición de Compañero:** El “inquilino” ó “Buscador de un piso”, podrá dar su visto bueno a las peticiones de los usuarios que crea convenientes cuyos perfiles sean validados por él mismo para poder residir en ese piso en concreto, siempre que haya plazas disponibles.
- **Enviar solicitud de compañero a otro usuario:** Se enviará la petición de querer compartir piso con otro usuario registrado. Se usará para poder hacer peticiones conjuntas de un piso.
- **Asignar compañero/s actual:** El sistema automáticamente establecerá el vínculo de ser compañeros actuales, al nuevo usuario que pase a vivir en el mismo domicilio con el solicitante de la petición.
- **Desasignar compañero/s actual:** El sistema automáticamente romperá el vínculo de ser compañeros actuales, a los usuarios que dejen de vivir actualmente en el mismo domicilio.
- **Eliminar Compañero potencial:** Cuando un usuario ha aceptado la petición de otro usuario, para ser compañeros potenciales de un piso, es posible que cualquiera de los dos cambie de opinión en cualquier momento “deshaciendo el grupo” creado para encontrar el piso adecuado a ambos.
- **Geolocalización de los pisos con Google Maps:** Utilizando un mapa embebido usando la API de *Google Maps*, el usuario puede ver la localización del piso.
- **Sistema de notificaciones:** Proporciona flexibilidad a la aplicación, y mejora la experiencia de usuario gratuitamente. Notifica a un usuario particular de movimientos en la aplicación que puedan interesarle (e.g: la solicitud de miembros de la aplicación interesados en el piso ofertado por el usuario, un posible match entre usuarios que buscan el mismo piso y cuyos intereses son afines, etc.).
- **Buzón de sugerencias y problemas técnicos:** Vía de contacto de los usuarios con los administradores.

2.2.2. Buscar piso

- **Marcar piso como "Deseado":** El usuario marca un piso como "deseado", es decir, lo añade a su lista de pisos favoritos en los que le gustaría alojarse. La lista de pisos deseados es individual para cada usuario.
- **Mostrar lista de pisos deseados de un usuario:** Se mostrará la lista de pisos "deseados" de un usuario, es decir, los pisos favoritos en los que le gustaría alojarse.
- **Eliminar piso como "Deseado":** Una vez el usuario ha seleccionado un piso como "deseado", también será posible deshacer esta selección, eliminando así la preferencia del usuario por el piso.
- **Solicitar piso:** Tendrá la opción de poder añadir a otros usuarios (con la condición de ser compañeros) para poder hacer peticiones conjuntas.
- **Mostrar pisos por valoración:** Da un resultado en forma de lista, con todos los pisos de entre los disponibles con los filtros previamente establecidos, ordenados de mayor/menor valoración.

2.2.3. Buscar inquilino

- **Registrar piso:** El usuario que registre el piso podrá indicar si es propietario y/o inquilino del mismo. No está necesariamente obligado a ofertarlo (hacerlo público).
- **Modificar datos de piso:** Permite modificar los datos de un piso.
- **Ofertar piso:** El usuario, puede ofertar su piso (establecerá el precio exigido y otros datos), para compartirlo con otros usuarios.
- **Eliminar oferta de piso:** El inquilino de un piso, puede eliminar la oferta realizada de su piso en cualquier instante, lo que provocaría un borrado de los intereses de los usuarios que estuviesen interesados en ese piso en particular.
- **Aceptar/Rechazar inquilino:** Permite acepta o rechaza una petición de inquilino.

2.2.4. Administrador

- **Eliminar cuenta:** Elimina una cuenta de usuario del sistema.
- **Eliminar anuncios y publicaciones:** Elimina el anuncio de un piso del sistema.
- **Ánalysis de datos (usuarios, anuncios, estadísticas):** Analiza los datos de los usuarios del sistema y de los pisos ofertados para calcular unas estadísticas que sirvan de utilidad para los usuarios.

2.3. Características de los usuarios

El producto estará destinado a cualquier persona con edades entre los 17 y los 35 años de edad, de habla española o inglesa, que posea un dispositivo móvil o web. Pero, sobretodo, personas que quieran buscar compañero con quien compartir un piso.

La aplicación se verá sencilla e intuitiva para que no exista ningún problema a la hora de usarla.

2.4. Restricciones

La implementación del programa se hará con tecnologías web. Para el almacenamiento de datos se usará una base de datos relacional [o MongoDB al usar RRSS].

El desarrollo de la versión web se realizará con Drupal, un sistema de gestión de contenidos de software libre que se utiliza para crear sitios web dinámicos y con gran variedad de funcionalidades.

Para poder visualizar la página web y depurarla necesitaremos XAMPP (de software libre) con el que podemos interpretar líneas de código.

Y por último Filezilla, que lo usaremos para subir archivos desde nuestro ordenador local a los repositorios alojados en el servidor dedicado.

La fecha de la entrega es una restricción temporal, por lo que se deben tener en cuenta a la hora de la planificación del mismo. Además habrá posibles entradas previas de cara al profesor que habrá que cumplir.

2.5. Suposiciones y dependencias

El sistema correrá sobre una plataforma web, por lo que se utilizarán las tecnologías actuales para llevarlo a cabo. Se necesitarán navegadores con versiones relativamente actuales, de no ser así pueden surgir problemas a la hora de navegar por el sistema.

El sistema depende de una BBDD relacional (MySQL) [o MongoDB en caso de usarla] para el almacenamiento de los datos de los usuarios, los pisos, las características de ambos y sus relaciones.

2.6. Requisitos futuros

A fecha de la redacción de ésta SRS del proyecto “ComAppNeros” no se contemplan futuros requisitos.

3. Requisitos específicos

3.1. Interfaces externas

Nuestra página web tiene una estrecha relación con Facebook, Google+ y Google Maps. Los usuarios que lo deseen pueden registrarse en nuestra web a través Facebook y Google+ si ellos lo desean, para ahorrarse tener que repetir su registro. También, utilizamos Google Maps para representar en un mapa la localización exacta de los pisos de los cuales tenemos sus coordenadas exactas.

3.1.1. Interfaces de usuario

A continuación se presentan unos prototipos de como sería la interfaz gráfica de la aplicación, no son definitivos pero si dan una idea muy aproximada de como serán. En el Apartado 5 del documento de Calidad (SQA) se tiene una pequeña guía de estilo de como se hará la aplicación.

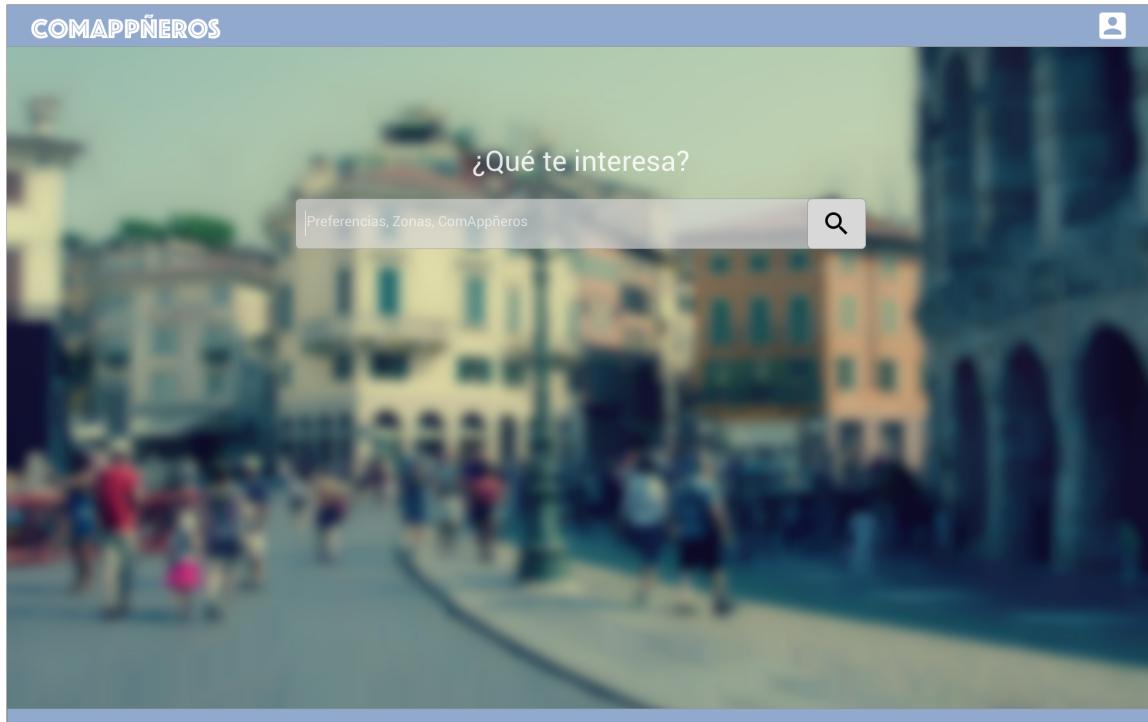


Tabla 1: Pagina de búsqueda

The screenshot shows a user profile page for 'Lorena Ipsum'. At the top, there is a placeholder for a profile picture with the text 'ComAppNeros'. Below the placeholder, the user's name 'Lorena Ipsum' is displayed. A profile picture of a woman with blonde hair wearing a hat is shown. To the right of the profile picture is a text box containing placeholder text: 'Hola soy Lorena Ipsum, y quiero decir que Lorem ipsum dolor sit amet, consectetur ad ipiscing elit. In eu nunc mollis, condimentum erat sed, faucibus nunc. In ornare vitae.' Below this text box are four circular icons representing different filters or interests: a paw print, a smoking person, a smiling face, and a baby carriage. Below these icons is a green circular icon with a plus sign and a person silhouette. To the right of the profile picture is a form with fields for 'Edad', 'Profesión', 'Lugar de Nacimiento', and 'Que busco'. At the bottom right of the profile area is a text input field labeled 'Escríbeme un mensaje' with a send button.

Tabla 2: Perfil con la información de un usuario

The screenshot shows a search interface for 'ComAppNeros'. The search bar at the top contains the placeholder text 'Preferencias, Zonas, ComAppñeros'. Below the search bar is a large question mark icon with the text '¿Qué te interesa?'. Below this are three cards representing different items: a modern apartment interior, another apartment interior, and a person's profile picture. Each card has a set of five circular icons below it, likely for filtering. At the bottom of the screen, there are three smaller cards showing parts of apartment interiors and a building exterior.

Tabla 3: Listado de usuarios y pisos

3.1.2. Interfaces de Software

Disponemos de una BBDD en la que almacenamos todos los datos relevantes para nuestra página web (pisos, usuarios, etc.).

3.1.3. Interfaces de Hardware

El hardware necesario para ejecutar nuestra página web es el siguiente:

- **Ordenador** (Características mínimas):

- Procesador: 1GHZ o superior.
- Memoria Ram: 1GB o superior.
- Espacio en disco: 420 MB como mínimo.
- Conexión a internet.

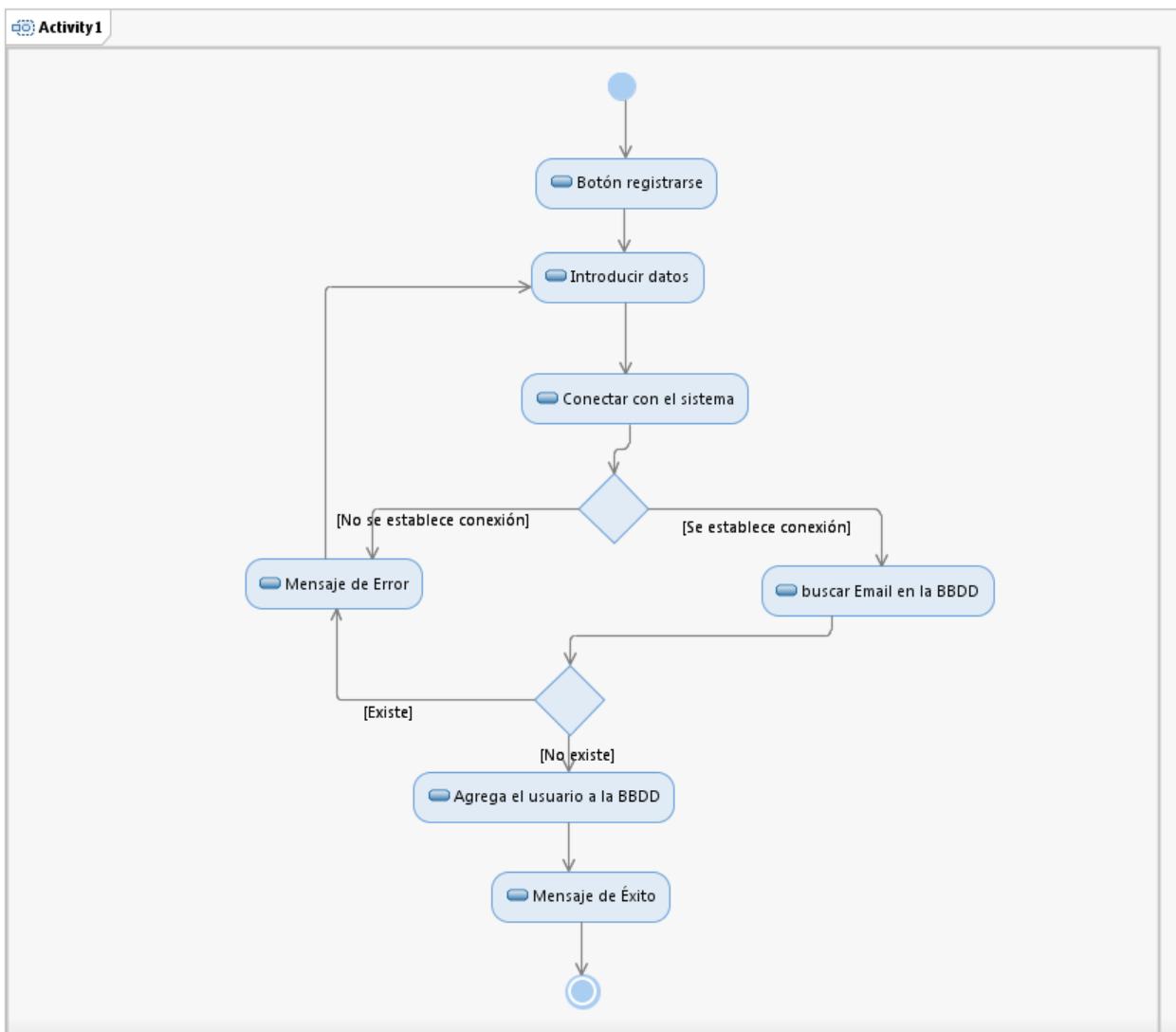
- **Smartphone o tablet** (Características mínimas):

- Procesador: 1GHz o superior.
- Memoria RAM: 512MB o superior.
- Espacio libre: 420 MB como mínimo.
- Conexión a internet.

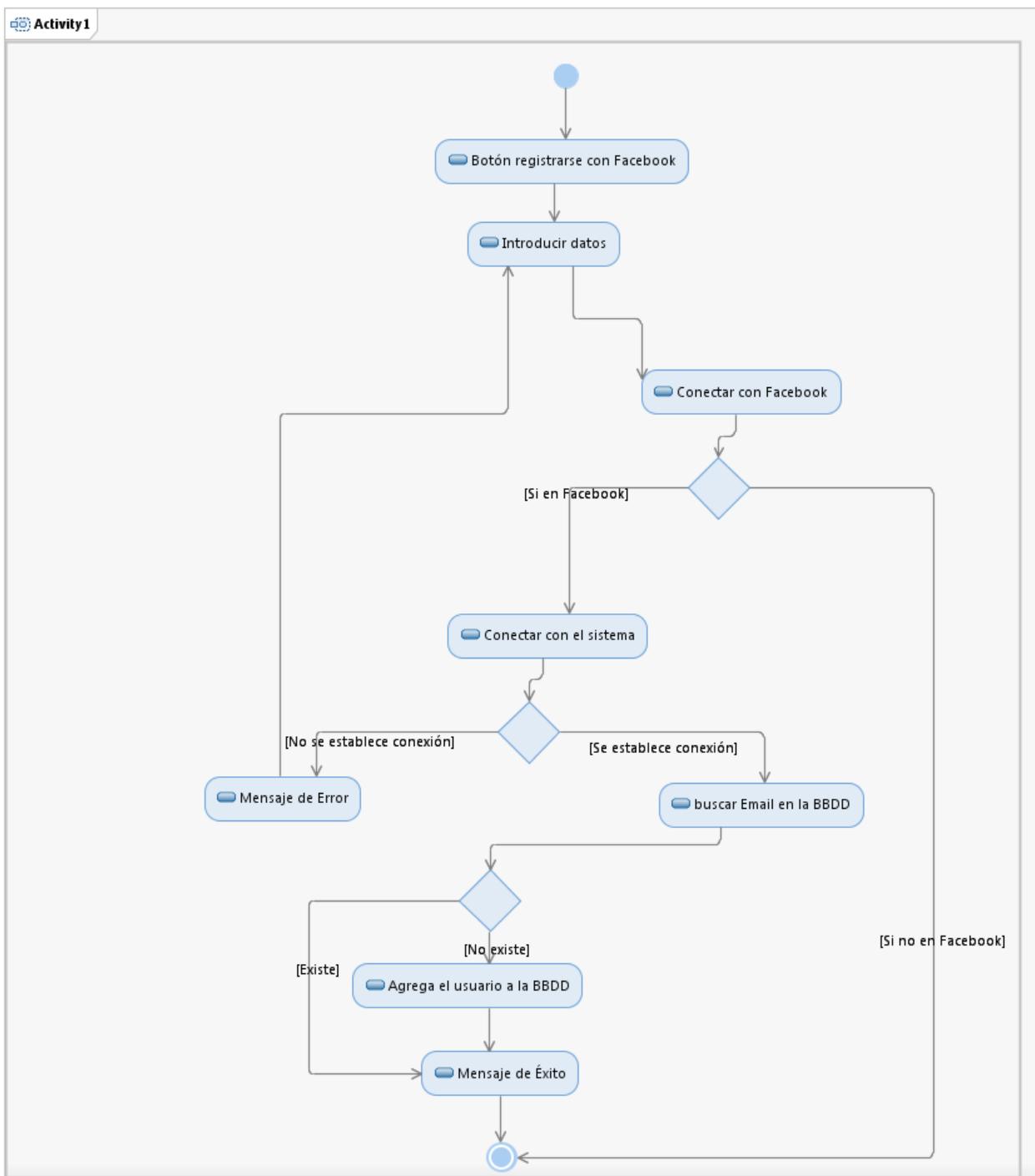
3.2. Requisitos Funcionales

3.2.1. Requisitos Generales

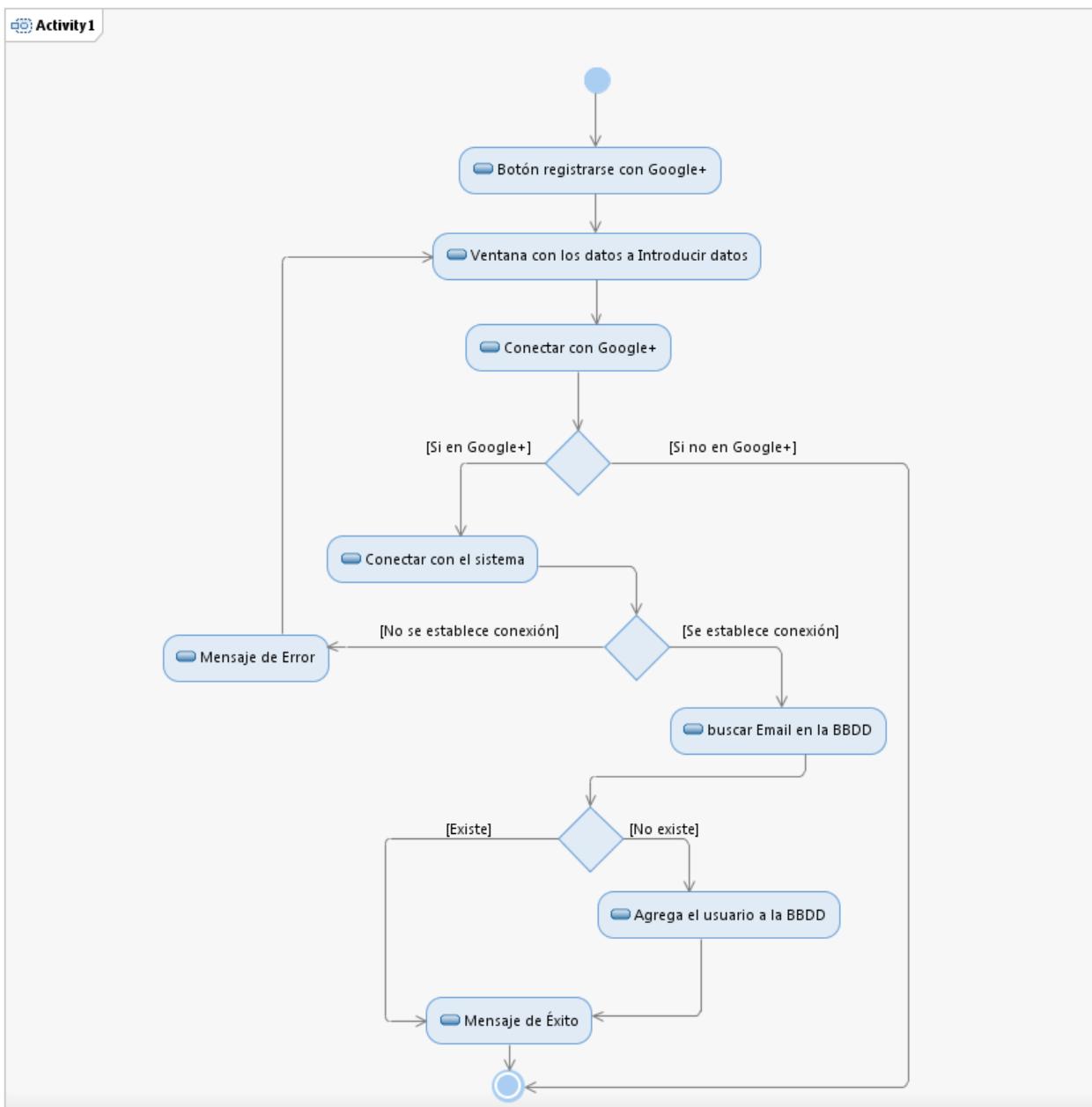
Función	Registrar cuenta de usuario
Código	RG01
Prioridad	Alta
Estabilidad	Alta
Descripción	Añade un usuario al sistema.
Entrada	Email, Nombre, Apellidos, Fecha de nacimiento, Password, Estado, Género.
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Que el email no exista en el sistema.
Postcondición	Usuario añadido al sistema.
Acción	Crea un nuevo usuario en el sistema
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos en el botón registrarse. 2. La interfaz cambia y nos muestra un pequeño cuadro para introducir los datos del usuario. 3. Una vez insertados los datos pulsamos el botón registrar. 4. El sistema comprueba si existe el Email del usuario. <ol style="list-style-type: none"> a) No se establece conexión: se envía un mensaje de error y vuelve al paso 2. b) Se establece conexión: se agrega el usuario al sistema para terminar se notifica con un mensaje de éxito.

Tabla 4: Diagrama de actividad *Registrar cuenta*

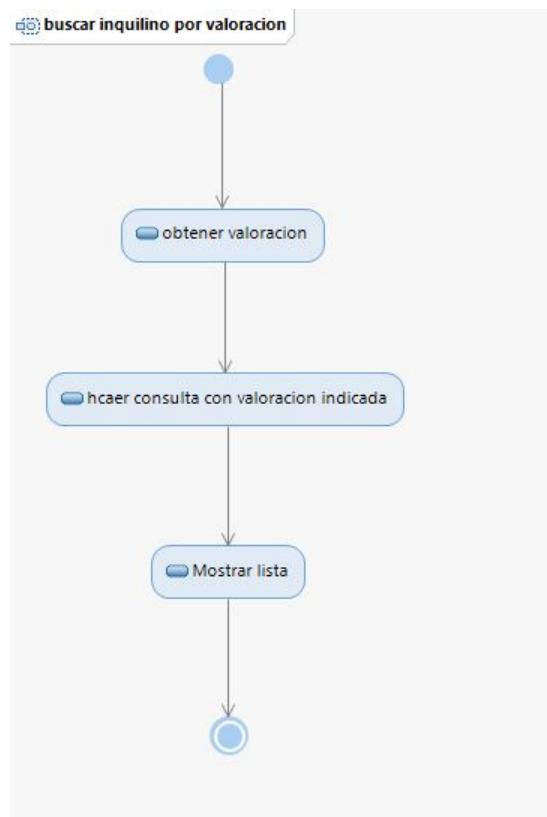
Función	Registrar cuenta con Facebook
Código	RG02
Prioridad	Alta
Estabilidad	Alta
Descripción	Registrar un usuario mediante la cuenta de Facebook.
Entrada	Email.
Salida	Mensaje de éxito o error.
Origen	GUI
Destino	Sistema
Precondición	Que tenga una cuenta de Facebook.
Postcondición	Usuario añadido al sistema.
Acción	Crea un nuevo usuario en el sistema con los datos recibidos de Facebook.
Flujo de evento	<p>1. En la GUI pulsamos en el botón registrarse con Facebook.</p> <p>2. La interfaz cambia y nos muestra un pequeño cuadro para introducir el usuario y la contraseña.</p> <p>3. Una vez insertados los datos pulsamos el botón registrar con Facebook.</p> <p>a) Si en Facebook</p> <ul style="list-style-type: none"> ▪ Si existe en el sistema: se notifica con un mensaje de éxito. ▪ Si no existe en el sistema: se agrega el usuario al sistema, se notifica con un mensaje de éxito. <p>b) Si no en Facebook</p> <ul style="list-style-type: none"> ▪ Se envía un mensaje de error y vuelve al paso 2.

Tabla 5: Diagrama de actividad *Registrar cuenta con Facebook*

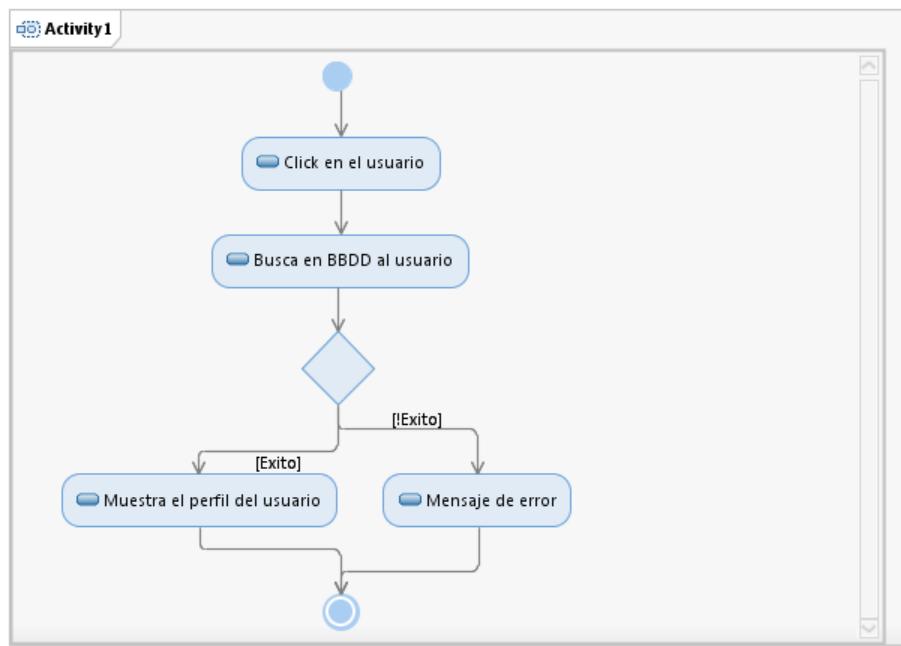
Función	Registrar cuenta con Google+
Código	RG03
Prioridad	Alta
Estabilidad	Alta
Descripción	Registrar un usuario mediante la cuenta de Google+.
Entrada	Email.
Salida	Mensaje de éxito o error.
Origen	GUI
Destino	Sistema
Precondición	Que tenga una cuenta de Google+.
Postcondición	Usuario añadido al sistema.
Acción	Crea un nuevo usuario en el sistema.
Flujo de evento	<p>1. En la GUI pulsamos en el botón registrarse con Google+.</p> <p>2. La interfaz cambia y nos muestra un pequeño cuadro para introducir el usuario y la contraseña.</p> <p>3. Una vez insertados los datos pulsamos el botón registrar con Google+.</p> <p>a) Si en Google+</p> <ul style="list-style-type: none"> ▪ Si existe en el sistema: se notifica con un mensaje de éxito. ▪ Si no existe en el sistema: se agrega el usuario al sistema, se notifica con un mensaje de éxito. <p>b) Si no en Google+</p> <ul style="list-style-type: none"> ▪ Se envía un mensaje de error y vuelve al paso 2.

Tabla 6: Diagrama de actividad *Registrar cuenta con Google+*

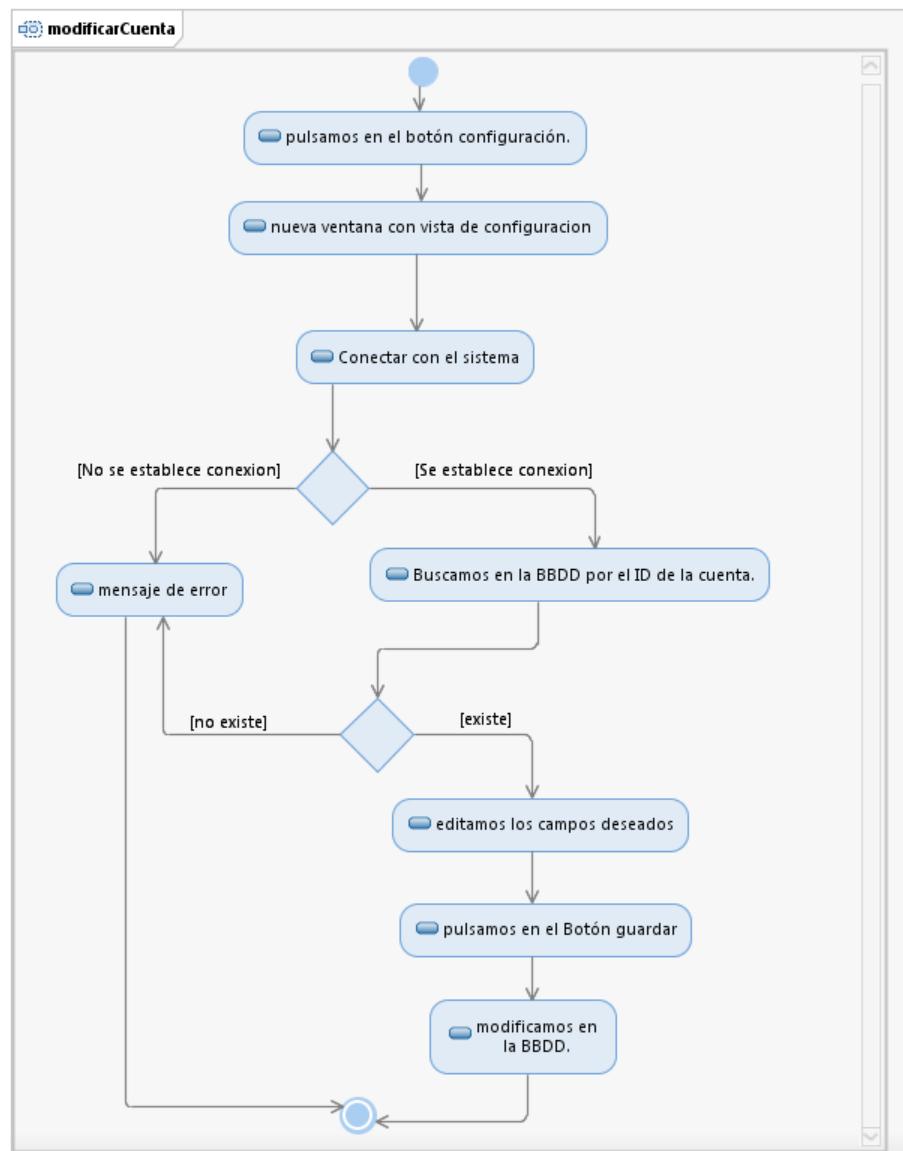
Función	Buscar usuarios por valoración
Código	RG04
Prioridad	Alta
Estabilidad	Alta
Descripción	Filtrar la búsqueda de los usuarios por la valoración indicada
Entrada	Valor de la valoración
Salida	Lista de usuarios con o por encima de dicha valoración
Origen	GUI
Destino	Sistema
Precondición	Valoración dentro del rango de las posibles valoraciones
Postcondición	Lista filtrada de usuarios
Acción	Crea un nuevo usuario en el sistema
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos en el botón registrarse. 2. La interfaz cambia y nos muestra un pequeño cuadro para introducir los datos del usuario. 3. Una vez insertados los datos pulsamos el botón registrar. 4. El sistema comprueba si existe el Email del usuario. <ol style="list-style-type: none"> a) No se establece conexión: se envía un mensaje de error y vuelve al paso 2. b) Se establece conexión: se agrega el usuario a la BBDD para terminar se notifica con un mensaje de éxito.

Tabla 7: Diagrama de actividad *Buscar usuarios por valoración*

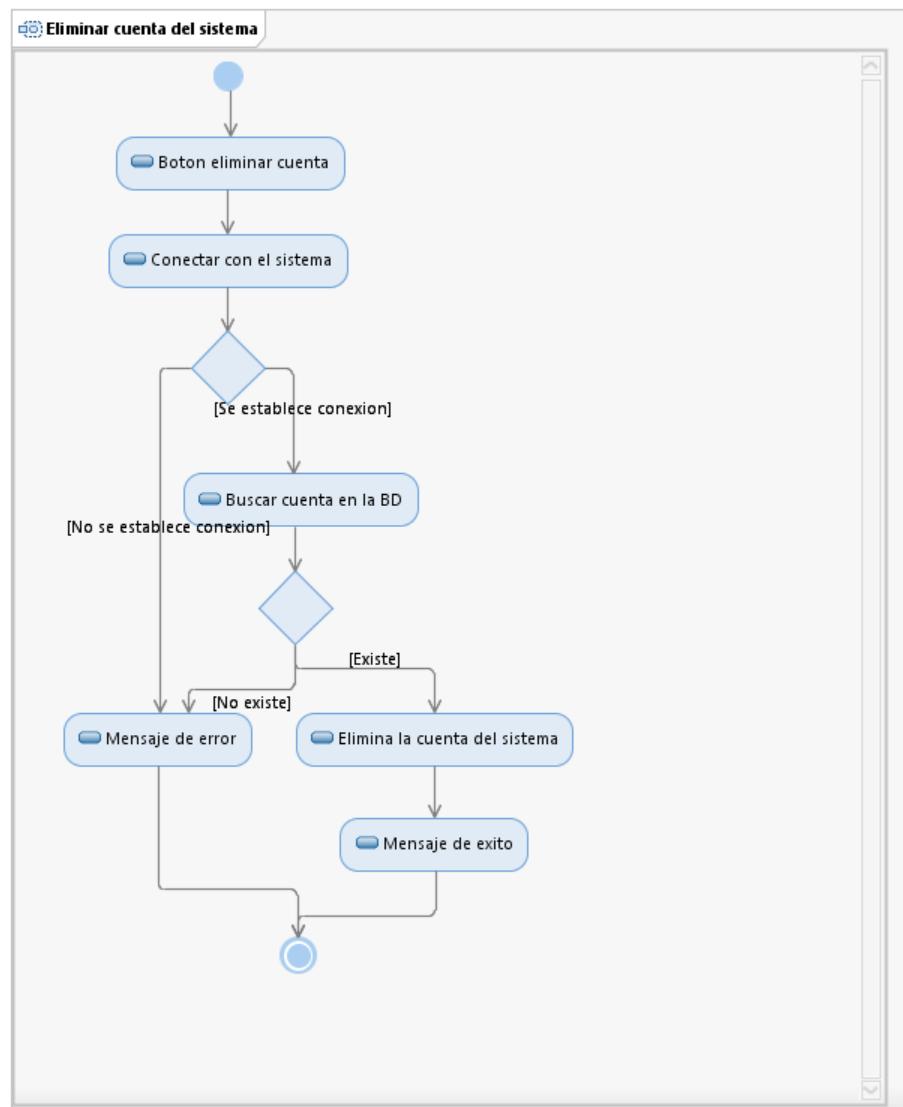
Función	Mostrar información de usuario
Código	RG05
Prioridad	Alta
Estabilidad	Alta
Descripción	Muestra los datos del usuario.
Entrada	ID del usuario
Salida	Mensaje de éxito o error.
Origen	GUI
Destino	GUI
Precondición	Usuario existente.
Postcondición	Los datos del usuario son mostrados en pantalla.
Acción	Muestra la información del usuario.
Flujo de evento	<p>1. Pinchamos dos veces en el usuario del que queremos visualizar su información.</p> <p>2. Comprobamos si el usuario está almacenado en el sistema. A raíz de esto pueden ocurrir dos cosas:</p> <ul style="list-style-type: none"> a) Si el usuario existe en el sistema, aparecerá una nueva ventana con la información del usuario. b) Si el usuario no existe en el sistema, aparecerá una ventana de error comunicando que el usuario no existe.

Tabla 8: Diagrama de actividad *Mostrar información de usuario*

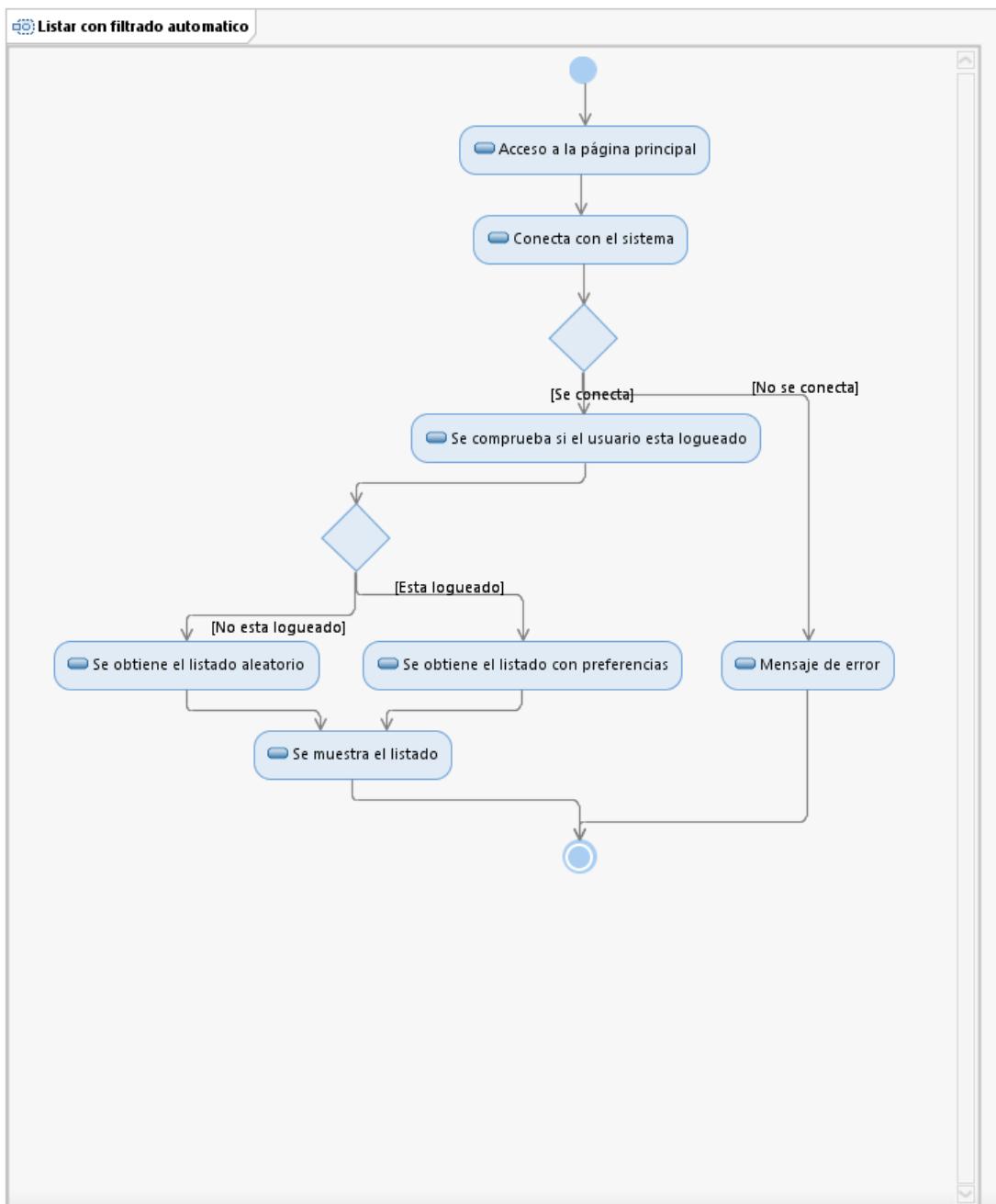
Función	Modificar cuenta
Código	RG06
Prioridad	Alta
Estabilidad	Alta
Descripción	Modifica los datos del usuario.
Entrada	ID, Email, Nombre, Apellidos, Fecha de nacimiento, Password, Estado, Género, Tags.
Salida	Mensaje de éxito o error.
Origen	GUI
Destino	Sistema
Precondición	Cuenta existente.
Postcondición	Se han modificado los datos del usuario en el sistema.
Acción	Modifica la cuenta del usuario.
Flujo de evento	<ol style="list-style-type: none"> 1. En la ventana principal pulsamos en el botón Modificar cuenta. 2. La interfaz cambia a la pantalla “Modificar cuenta” y nos muestra un pequeño cuadro para poder editar los datos de la cuenta. Solo se puede editar su nombre, teléfono, edad, género, ocupación, fumador, tiene mascota. 3. El operador del sistema puede elegir: <ol style="list-style-type: none"> a) Cancelar no se modifica ningún dato y se regresa a la pantalla de configuración de la cuenta. b) Aceptar se actualizan los cambios en el sistema y se muestra un mensaje de éxito.

Tabla 9: Diagrama de actividad *Modificar cuenta*

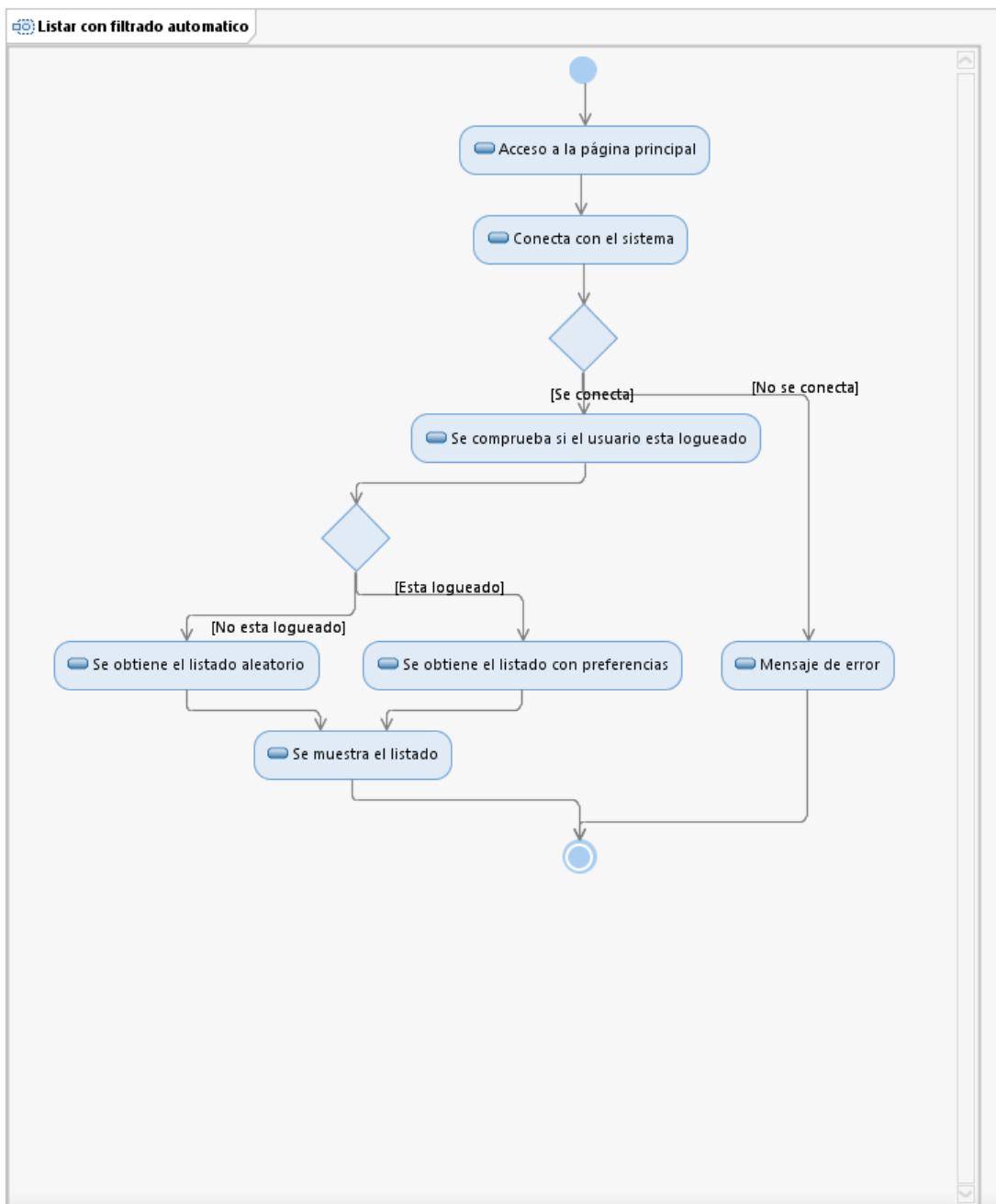
Función	Eliminar cuenta del sistema
Código	RG07
Prioridad	Alta
Estabilidad	Alta
Descripción	Elimina una cuenta de usuario del sistema.
Entrada	ID de la cuenta.
Salida	Mensaje.
Origen	GUI
Destino	Sistema
Precondición	Cuenta existente.
Postcondición	Eliminar cuenta de la BBDD.
Acción	Eliminar la cuenta del sistema.
Flujo de evento	<ol style="list-style-type: none"> 1. En la pagina de configuración de la cuenta de usuario, el usuario pulsa sobre el botón “Eliminar cuenta”. 2. Se muestra un mensaje de confirmación: <ol style="list-style-type: none"> a) Eliminar cuenta: la cuenta y todos los datos personales del usuario que estuvieran en la BBDD son eliminados. b) Cancelar: la cuenta no se elimina.

Tabla 10: Diagrama de actividad *Eliminar cuenta del sistema*

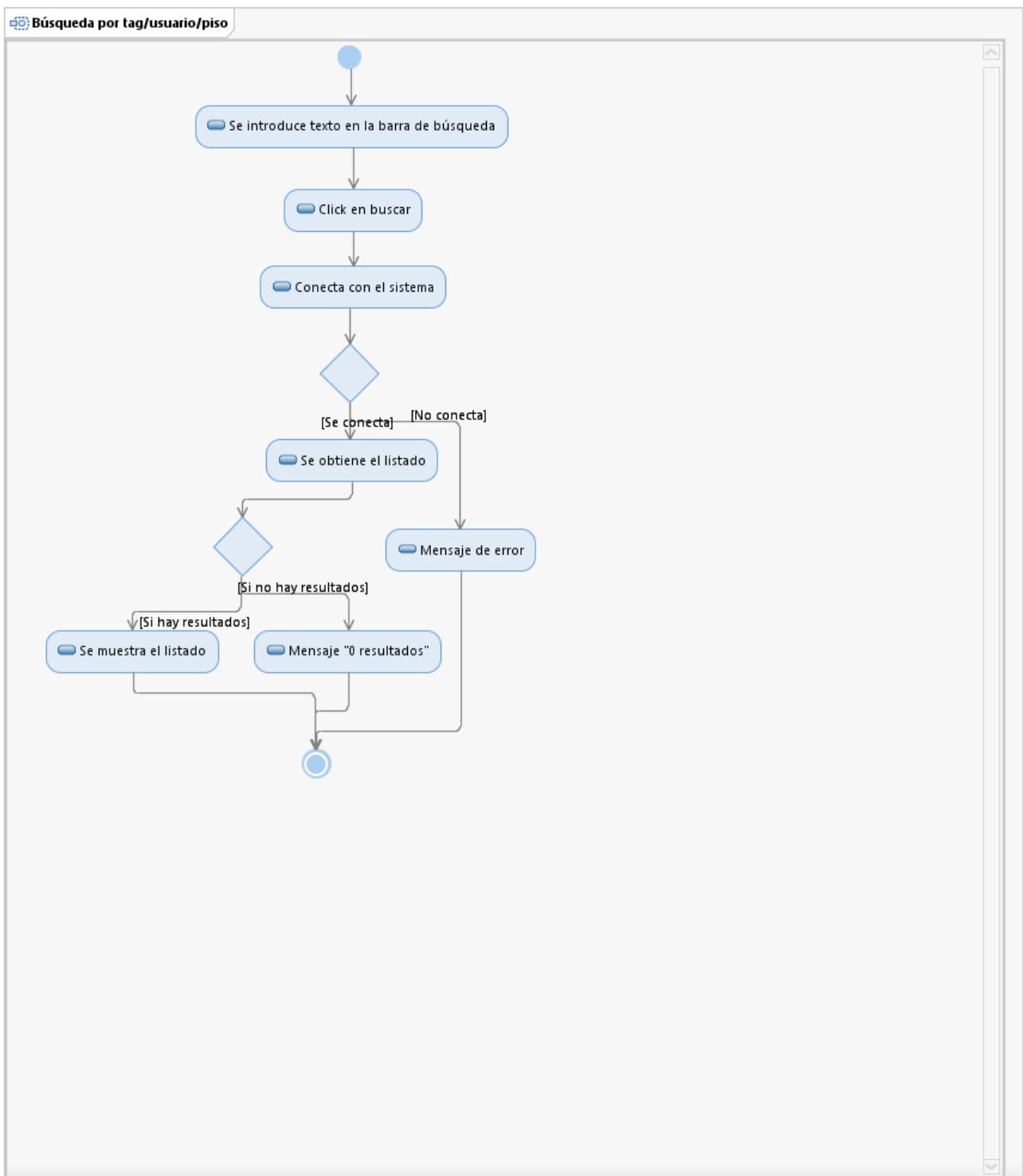
Función	Listar con Filtrado Automático usuario
Código	RG08
Prioridad	Media
Estabilidad	Alta
Descripción	Recoge los datos del usuario que ya está logueado en el sistema y filtra automáticamente pisos en función de las características del propio usuario.
Entrada	Usuario.
Salida	Lista con los usuarios de mayor compatibilidad con el usuario activo.
Origen	Sistema
Destino	Sistema
Precondición	Usuario logueado en el sistema.
Postcondición	Se muestra el listado.
Acción	Realizar la búsqueda de los usuarios y mostrarlos en pantalla.
Flujo de evento	<p>1. Se inicia la aplicación.</p> <p>a) El usuario logueado hace click en el botón de filtración automática de la lista de usuarios.</p> <p>b) El sistema recoge los datos del usuario y los manda a la función de filtrado.</p> <p>c) Esta función devuelve una lista que se mostrará por pantalla con los usuarios del sistema que sean más compatibles.</p>

Tabla 11: Diagrama de actividad *Listar con Filtrado Automático usuario*

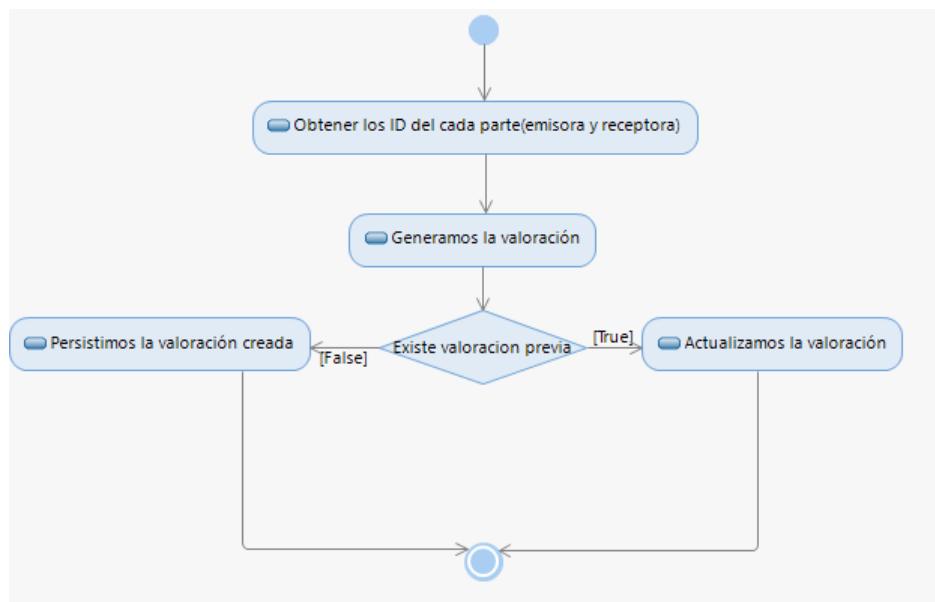
Función	Listar con Filtrado Automático piso
Código	RG09
Prioridad	Media
Estabilidad	Alta
Descripción	Recoge los datos del usuario que ya está logueado en el sistema y filtra automáticamente pisos en función de las características del propio usuario.
Entrada	Usuario.
Salida	Lista con los pisos de mayor compatibilidad con el usuario activo.
Origen	Sistema
Destino	Sistema
Precondición	Usuario logueado en el sistema.
Postcondición	Se muestra el listado.
Acción	Realizar la búsqueda de los pisos y mostrarlos en pantalla.
Flujo de evento	<p>1. Se inicia la aplicación.</p> <p>a) El usuario logueado hace click en el botón de filtración automática de la lista de pisos.</p> <p>b) El sistema recoge los datos del pisos y los manda a la función de filtrado.</p> <p>c) Esta función devuelve una lista que se mostrará por pantalla con los pisos del sistema que sean más compatibles.</p>

Tabla 12: Diagrama de actividad *Listar con Filtrado Automático piso*

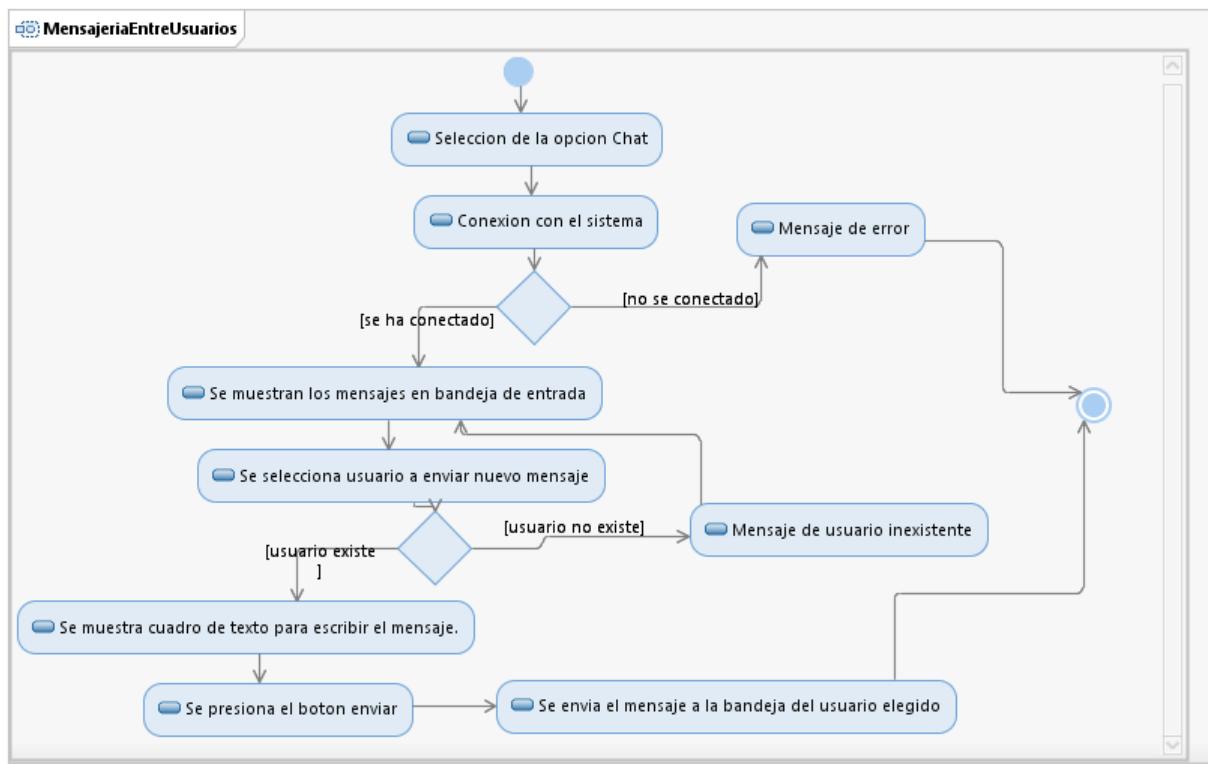
Función	Búsqueda por Tag/Usuario/Piso
Código	RG010
Prioridad	Media
Estabilidad	Alta
Descripción	Realiza una búsqueda por tag, usuario, piso.
Entrada	El tag, usuario o piso que se quiere buscar.
Salida	Página con el resultado de la búsqueda.
Origen	GUI
Destino	Sistema
Precondición	Que haya texto en la barra de búsqueda.
Postcondición	Se muestra el resultado de la búsqueda.
Acción	El sistema realiza la búsqueda y filtrado correspondiente.
Flujo de evento	<ol style="list-style-type: none"> 1. Se introduce texto en la barra de búsqueda. 2. Se pulsa sobre el botón “buscar”. <ol style="list-style-type: none"> a) Si existe algún resultado se muestra en una página al usuario. b) En caso contrario se muestra un página advirtiendo al usuario de no haber encontrado ningún resultado.

Tabla 13: Diagrama de actividad *Búsqueda por Tag/Usuario/Piso*

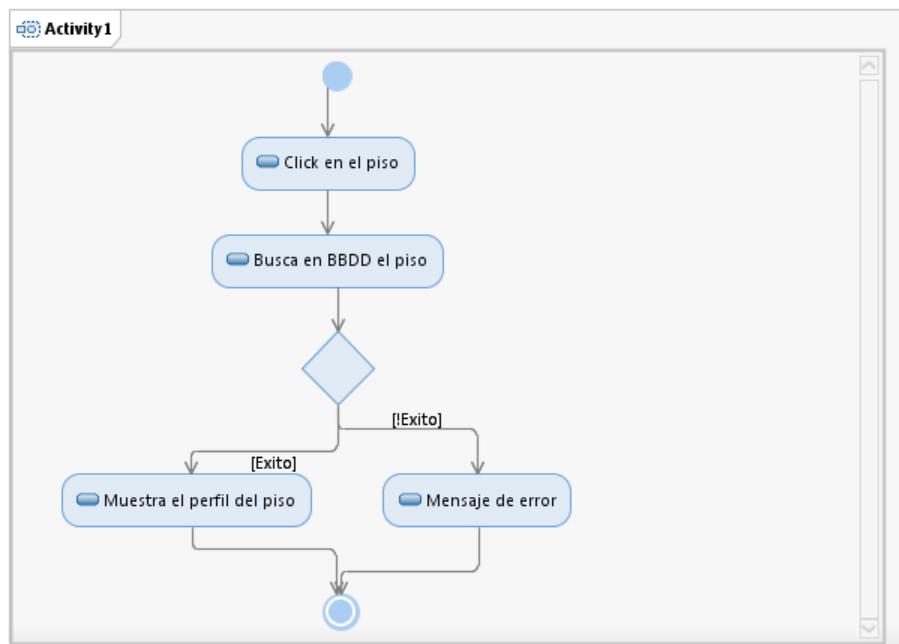
Función	Valorar un usuario
Código	RG011
Prioridad	Media
Estabilidad	Media
Descripción	El usuario puede valorar usuarios utilizando un sistema de 0 a 5 estrellas, después de haber concretado el alquiler.
Entrada	Valoración de 0 a 5 del usuario.
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	ID usuario válido y petición no alterada
Postcondición	Valoración creada y persistida.
Acción	Añade una valoración a un usuario.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos el icono de valoración. 2. Elegimos el valor de 0 a 5 estrellas. 3. La valoración de envía. 4. El sistema comprueba si existe valoración previa. <ol style="list-style-type: none"> a) False : Se añade la valoración a la media del usuario. b) True : Se reemplaza la valoración previa y se recalcula la media de valoración del usuario.

Tabla 14: Diagrama de actividad *Valorar un usuario*

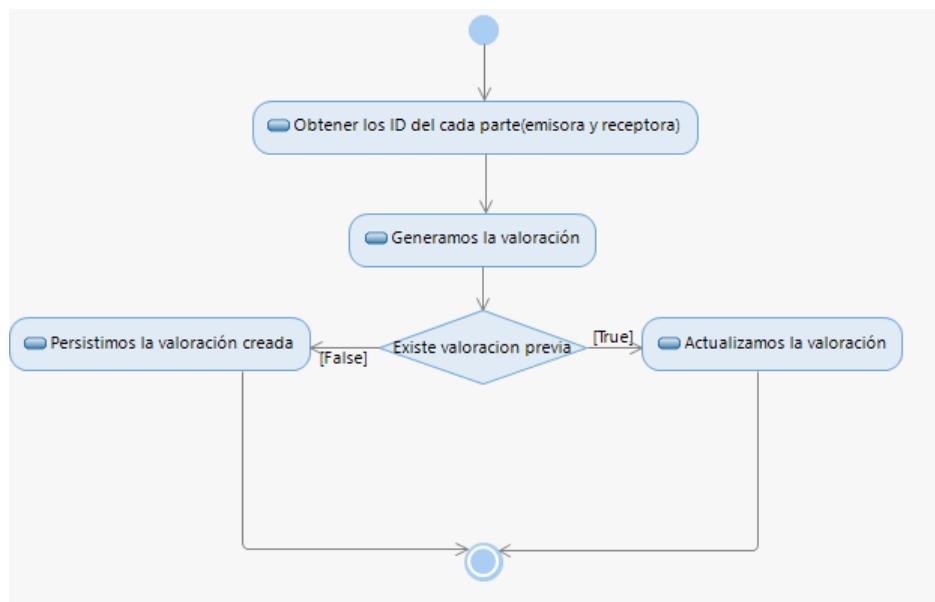
Función	Mensajería entre usuarios
Código	RG12
Prioridad	Media
Estabilidad	Media
Descripción	Permite el envío y recepción de mensajes entre un usuario y el buzón de otro usuario.
Entrada	ID de los usuarios, el asunto del mensaje y el contenido del mensaje
Salida	
Origen	GUI
Destino	Sistema
Precondición	Que ambos usuarios existan.
Postcondición	- -
Acción	Un usuario escribe un mensaje para enviar al buzón de otro usuario y cualquier usuario puede leer los mensajes en su propio buzón.
Flujo de evento	<ol style="list-style-type: none"> 1. El usuario entra en la función chat desde la pagina principal. 2. El usuario elige al usuario al que desea enviar el mensaje. 3. Se muestra una ventana con un cuadro de texto donde escribir el mensaje. 4. El sistema envía el mensaje al buzón de mensajes del usuario seleccionado (BBDD). 5. El usuario receptor puede leer todos los mensajes recibidos en su buzón.

Tabla 12: Diagrama de actividad *Mensajería entre usuarios*

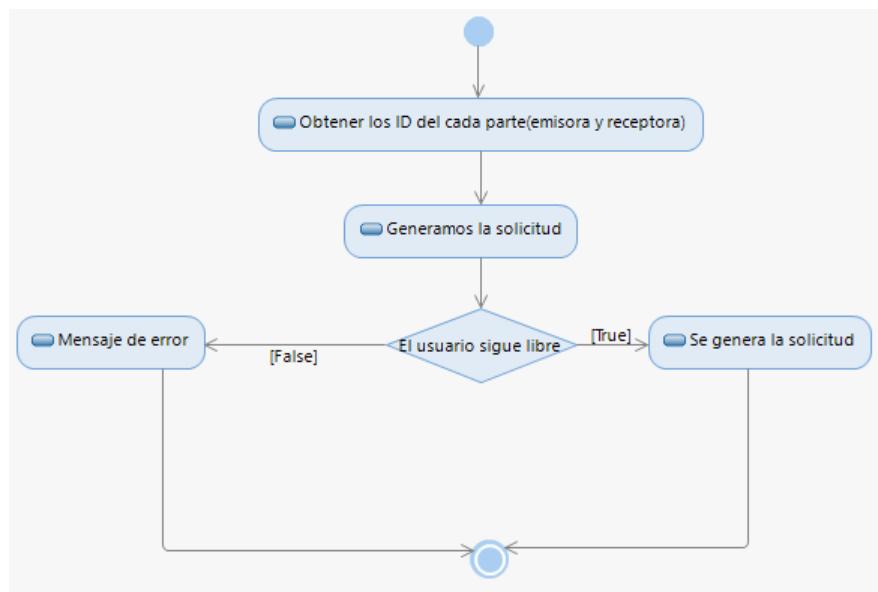
Función	Mostrar información de un piso
Código	RG13
Prioridad	Alta
Estabilidad	Alta
Descripción	Muestra los datos del piso.
Entrada	ID del piso.
Salida	Mensaje de éxito o error.
Origen	GUI
Destino	GUI
Precondición	Piso existente.
Postcondición	Los datos del piso son mostrados en pantalla.
Acción	Muestra la información del piso.
Flujo de evento	<p>1. Pinchamos dos veces en el piso del que queremos visualizar su información.</p> <p>2. Comprobamos si el piso está almacenado en el sistema. A raíz de esto pueden ocurrir dos cosas:</p> <ul style="list-style-type: none"> a) Si el piso existe en el sistema, aparecerá una nueva ventana con la información del piso. b) Si el piso no existe en el sistema, aparecerá una ventana de error comunicando que el piso no existe.

Tabla 16: Diagrama de actividad *Mostrar información de un piso*

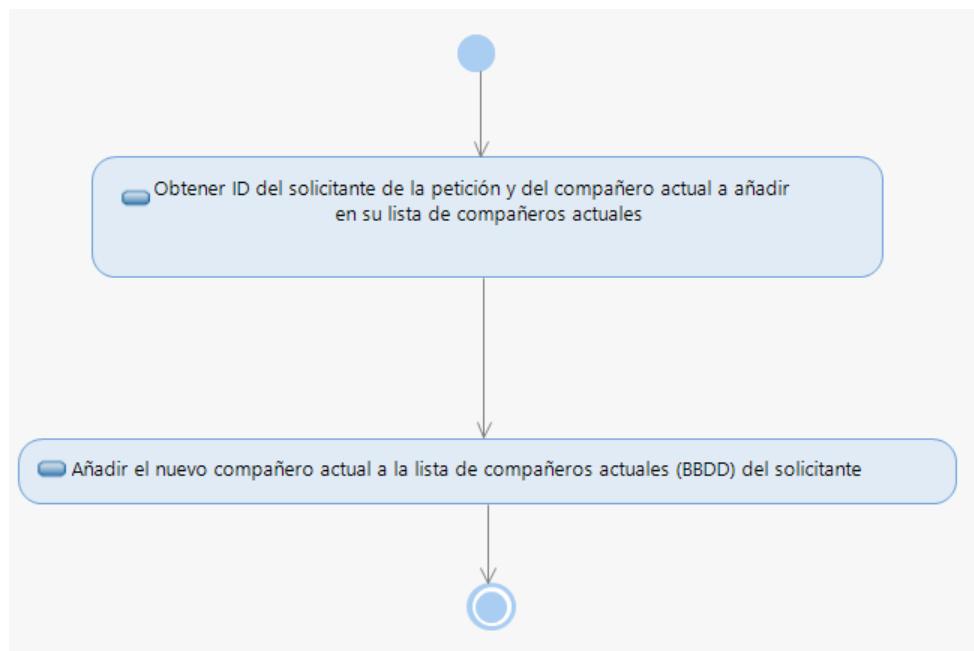
Función	Valorar un piso
Código	RG14
Prioridad	Media
Estabilidad	Media
Descripción	El usuario puede valorar usuarios utilizando un sistema de 0 a 5 estrellas, después de haber concretado el alquiler.
Entrada	Valoracion del piso.
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Piso válido y petición no alterada
Postcondición	Valoración creada y persistida.
Acción	Añade una valoración a un piso.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos el icono de valoración. 2. Rellenamos todos los campos de la valoración. 3. Enviamos la valoración. 4. El sistema comprueba si existe previa. <ol style="list-style-type: none"> a) False : persistimos la valoración creada. b) True : actualizamos la anterior.

Tabla 17: Diagrama de actividad *Valorar un piso*

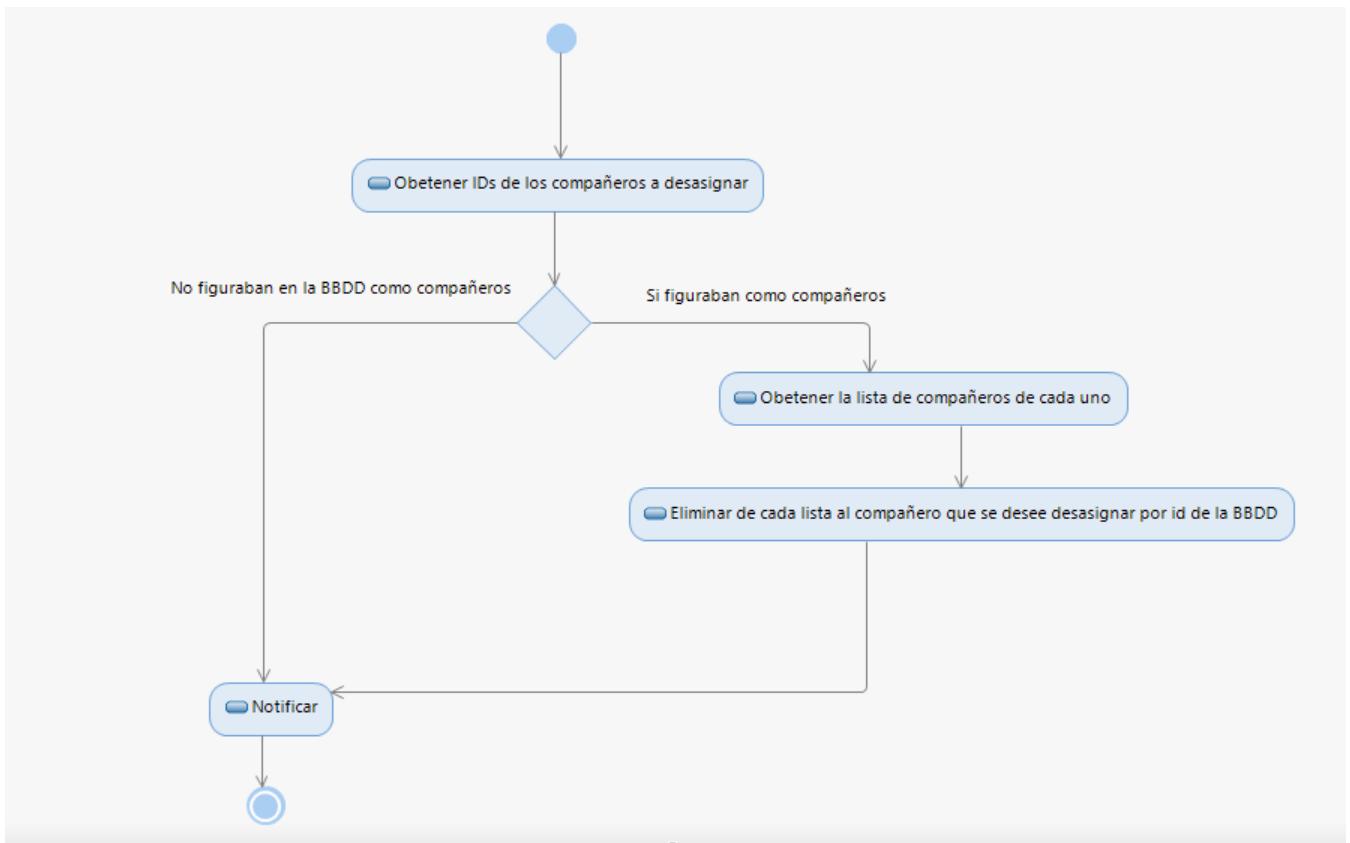
Función	Enviar solicitud de compañero potencial a otro usuario
Código	RG15
Prioridad	Alta
Estabilidad	Alta
Descripción	Se enviará la petición de querer compartir piso con otro usuario registrado. Se usará para poder hacer peticiones conjuntas de un piso.
Entrada	Nombre del usuario.
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	ID de usuario válido
Postcondición	Solicitud enviada.
Acción	Añade una petición de compañero en la lista de peticiones del usuario.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos el icono de solicitud de compañero. 2. Rellenamos la solicitud de compañero. 3. Enviamos la solicitud. 4. El sistema comprueba si ese usuario sigue solicitando compañero. <ol style="list-style-type: none"> a) False : mensaje indicando que el usuario ya no solicita compañero. b) True : persistimos la solicitud.

Tabla 18: Diagrama de actividad *Enviar solicitud de compañero potencial a otro usuario*

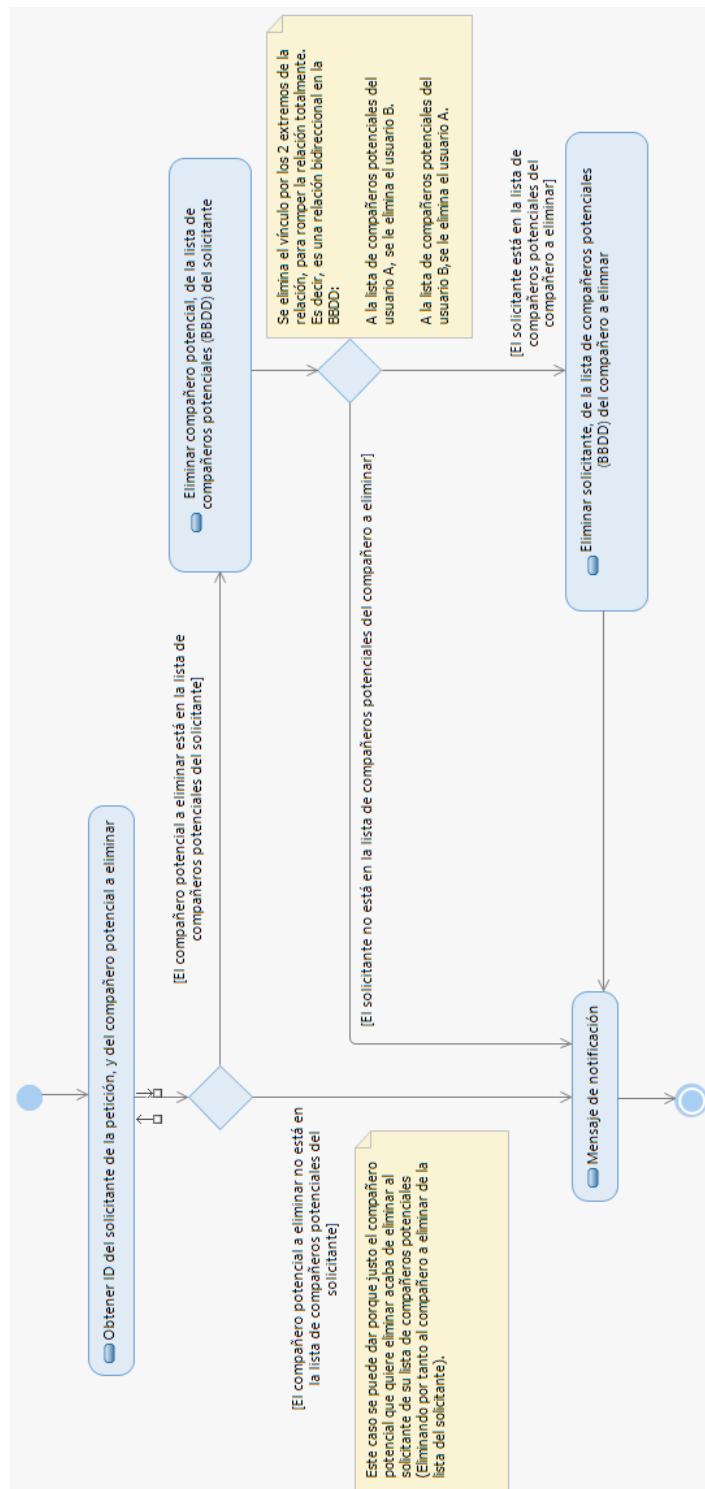
Función	Asignar compañero/s actual
Código	RG16
Prioridad	Alta
Estabilidad	Alta
Descripción	El sistema automáticamente establecerá el vínculo de ser compañeros actuales, al nuevo usuario que pase a vivir actualmente en el mismo domicilio, con el solicitante de la petición(Que se asignará en la función Aceptar Inquilino, pudiendo ser cualquier inquilino que ya viví ahí, ó algún nuevo inquilino). Al fin y al cabo, esta función terminará llamándose en tantas ocasiones que todos los inquilinos del mismo piso pasarán a ser compañeros actuales.
Entrada	Nombre del solicitante y del compañero actual que se va a añadir en su lista de compañeros.
Salida	Mensaje de éxito/error.
Origen	Sistema
Destino	Sistema
Precondición	Nuevos inquilinos en alguna vivienda.
Postcondición	Relación de compañeros actuales consolidada entre el solicitante de la petición y del compañero actual a añadir en su lista de compañeros actuales.
Acción	Crear vínculo de compañeros actuales entre los usuarios que viven en el mismo piso.
Flujo de evento	<ol style="list-style-type: none"> 1. Obtener el id del solicitante y del compañero actual a añadir en su lista de compañeros actuales. 2. Añadir el nuevo compañero actual a la lista de compañeros actuales.

Tabla 19: Diagrama de actividad *Asignar compañero/s actual*

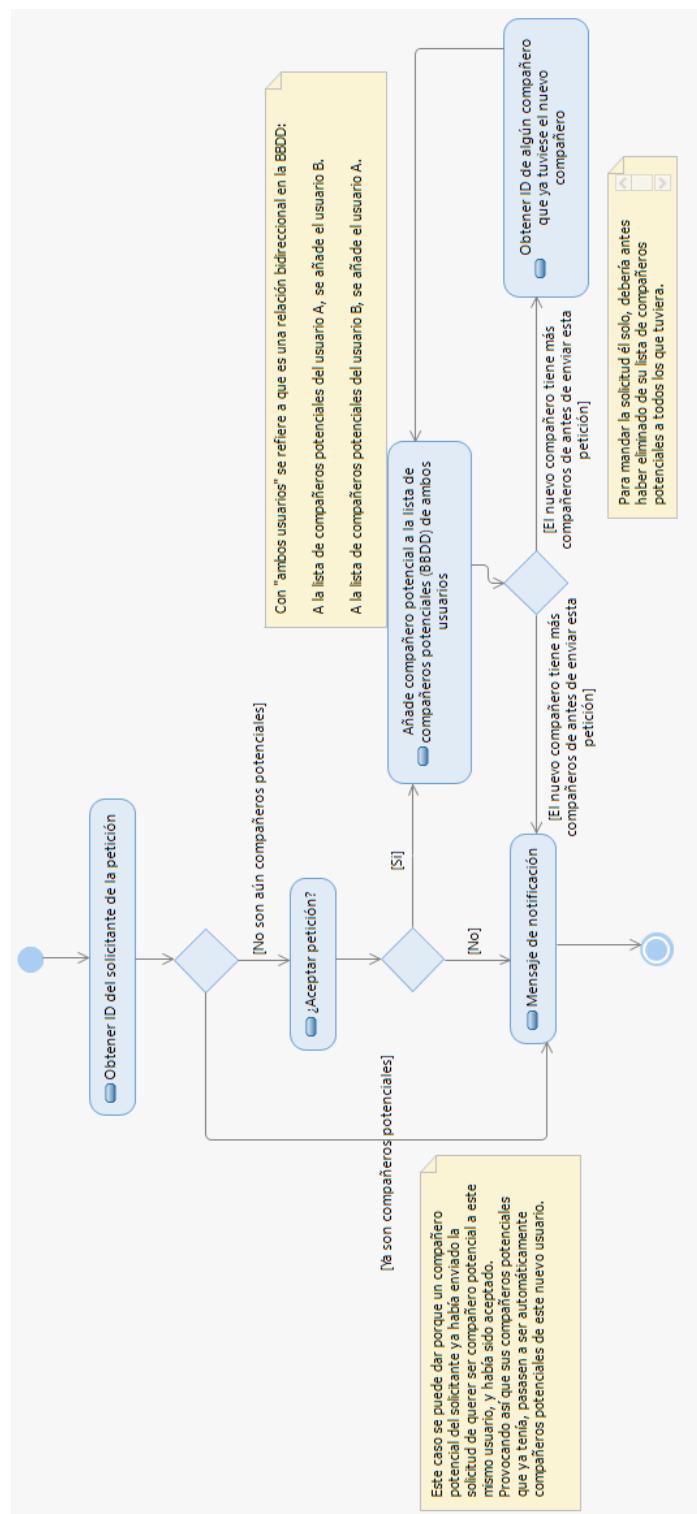
Función	Desasignar compañero/s actual
Código	RG17
Prioridad	Alta
Estabilidad	Alta
Descripción	El sistema automáticamente romperá el vínculo de ser compañeros actuales, a los usuarios que dejen de vivir actualmente en el mismo domicilio.
Entrada	-
Salida	-
Origen	Sistema
Destino	Sistema
Precondición	Varios inquilinos en alguna vivienda, y que alguno deje de vivir en dicha vivienda.
Postcondición	Usuarios implicados ya no están relacionados como compañeros.
Acción	Romper vínculo de compañeros actuales entre los usuarios que vivían en ese piso.
Flujo de evento	<ol style="list-style-type: none"> 1. Obtener el ID de los compañeros a desasignar. 2. El compañero figura en la base de datos. <ol style="list-style-type: none"> a) False : Notificamos el error. b) True : <ol style="list-style-type: none"> 1) Obtener la lista de compañeros de cada uno. 2) Eliminar de cada lista su asignado 3. Notificación

Tabla 20: Diagrama de actividad *Desasignar compañero/s actual*

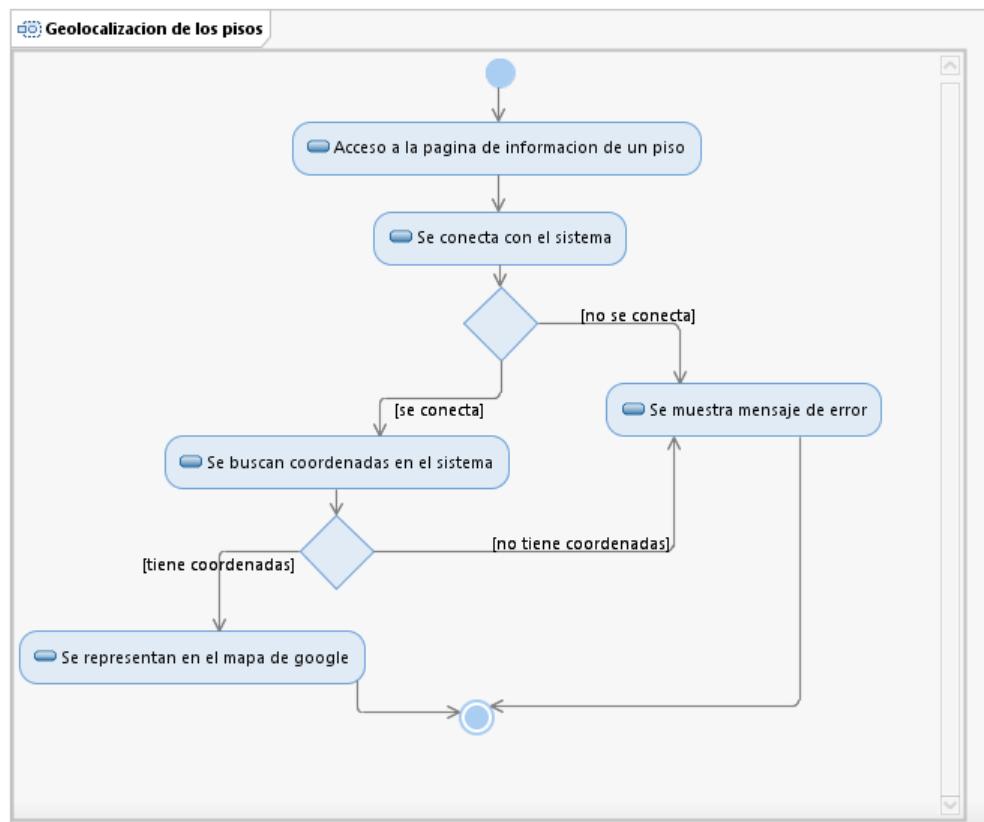
Función	Eliminar Compañero potencial
Código	RG18
Prioridad	Alta
Estabilidad	Alta
Descripción	Cuando un usuario ha aceptado la petición de otro usuario, para ser compañeros potenciales de un piso, es posible que cualquiera de los dos cambie de opinión en cualquier momento y rompan su vínculo creado para encontrar el piso adecuado a ambos.
Entrada	ID usuario que solicita eliminar a un compañero potencial suyo, ID compañero potencial a eliminar
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Vínculo existente entre el compañero potencial a eliminar y el usuario que solicita eliminarlo
Postcondición	Vínculo de compañero potencial roto, entre el compañero potencial a eliminar y el usuario que solicita eliminarlo
Acción	Eliminar el vínculo de compañeros potenciales entre el compañero potencial a eliminar y el usuario que solicita eliminarlo.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos en el usuario el botón eliminar el usuario potencial. 2. Obtenemos los id del emisor y el receptor. 3. Eliminamos el usuario de la lista de usuarios potenciales. 4. Enviamos una notificación.

Tabla 21: Diagrama de actividad *Eliminar Compañero potencial*

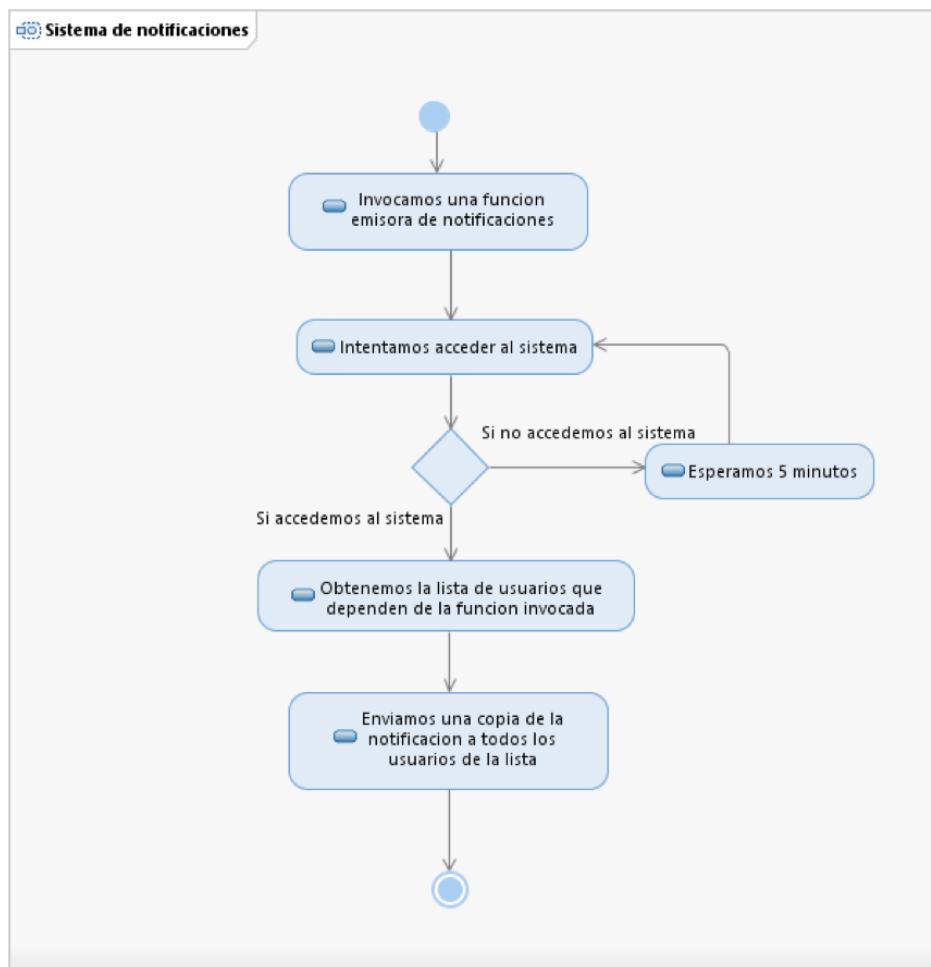
Función	Aceptar/Rechazar petición de compañero potencial
Código	RG19
Prioridad	Alta
Estabilidad	Alta
Descripción	El “inquilino” ó “Buscador de un piso”, podrá aceptar/rechazar las peticiones de los usuarios que crea convenientes , en el caso de aceptarlos, convirtiéndose así en compañeros potenciales(es decir, que aún no son realmente compañeros, pero les gustaría serlo)
Entrada	ID del usuario al que va dirigido la petición, ID del solicitante de la petición
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Haber recibido una oferta para alojarse en nuestro piso ofertado
Postcondición	Relación de compañeros potenciales consolidada entre solicitante y usuario al que va dirigida la petición, y automáticamente, si algún usuario tenía ya algún compañero potencial, esos compañeros también pasarán a ser compañeros del nuevo compañero potencial.
Acción	Crear vínculo de compañero potencial entre solicitante(y sus compañeros potenciales) y usuario(y sus compañeros potenciales) al que va dirigida la petición
Flujo de evento	<ol style="list-style-type: none"> 1. Obtener el ID del solicitante de la petición y el ID al que va dirigido. 2. Consultar si son compañeros potenciales. <ol style="list-style-type: none"> a) True : Asignar compañero. b) False : Notificación. 3. Acepta la petición

Tabla 22: Diagrama de actividad *Aceptar/Rechazar petición de compañero potencial*

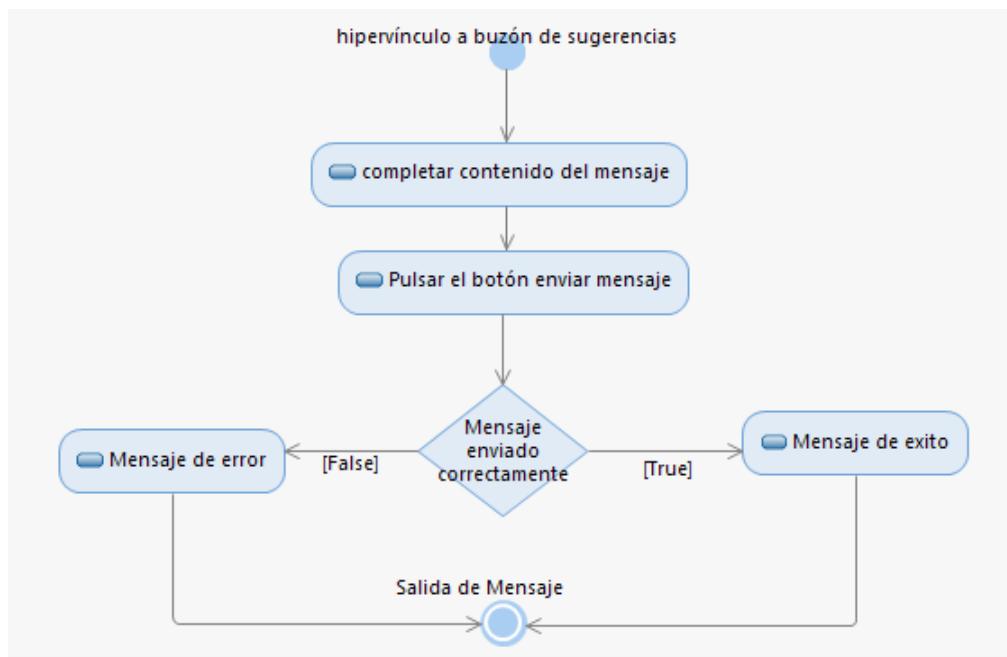
Función	Geolocalización de los pisos con <i>Google Maps</i>
Código	RG20
Prioridad	Alta
Estabilidad	Alta
Descripción	Permite ver la localización del piso utilizando Google Maps embebido en la pagina.
Entrada	ID del piso.
Salida	Representación de las coordenadas en un mapa de Google.
Origen	Sistema
Destino	GUI
Precondición	Piso existente y coordenadas del piso almacenadas en el sistema.
Postcondición	El piso es geolocalizado en un mapa de Google Maps.
Acción	Muestra la ubicación del piso seleccionado.
Flujo de evento	<ol style="list-style-type: none"> 1. Pinchamos dos veces en el piso del que queremos visualizar su localización. 2. Comprobamos si el piso está almacenado en el sistema. A raíz de esto pueden ocurrir dos cosas: <ol style="list-style-type: none"> a) Si el usuario existe en el sistema, aparecerá una nueva ventana con la información del piso y pasamos en punto 3. b) Si el usuario no existe en el sistema, aparecerá una ventana de error comunicando que el usuario no existe. 3. Comprobamos si las coordenadas del piso están almacenadas en el sistema. Pueden ocurrir dos cosas: <ol style="list-style-type: none"> a) Si las coordenadas del piso están en la sistema, se solicita el mapa a Google, quien nos devuelve el mapa listo, y lo situamos debajo de la información del piso. b) Si las coordenadas del piso no están en la sistema, aparecerá mensaje de error comunicando que no están en el sistema.

Tabla 23: Diagrama de actividad *Geolocalización de los pisos con Google Maps*

Función	Sistema de notificaciones
Código	RG21
Prioridad	Media
Estabilidad	Media
Descripción	Permite el envío y recepción de notificaciones a un usuario al utilizarse otra funciones del sistema
Entrada	- -
Salida	Notificaciones distintas para cada función relevante.
Origen	Sistema
Destino	Sistema
Precondición	- -
Postcondición	Notificación recibida por los usuarios dependientes de la función invocada.
Acción	Notifica a usuarios de acciones tomadas por otros usuarios o el sistema.
Flujo de evento	<ol style="list-style-type: none"> 1. Invocamos una función emisora de notificaciones. 2. Accedemos al sistema. Si no conseguimos acceder al sistema, esperamos 5 minutos y lo volvemos a intentar. 3. Obtenemos una lista de los usuarios dependen de la función invocada. 4. Enviamos una copia de la notificación a cada uno de los usuarios de la lista.

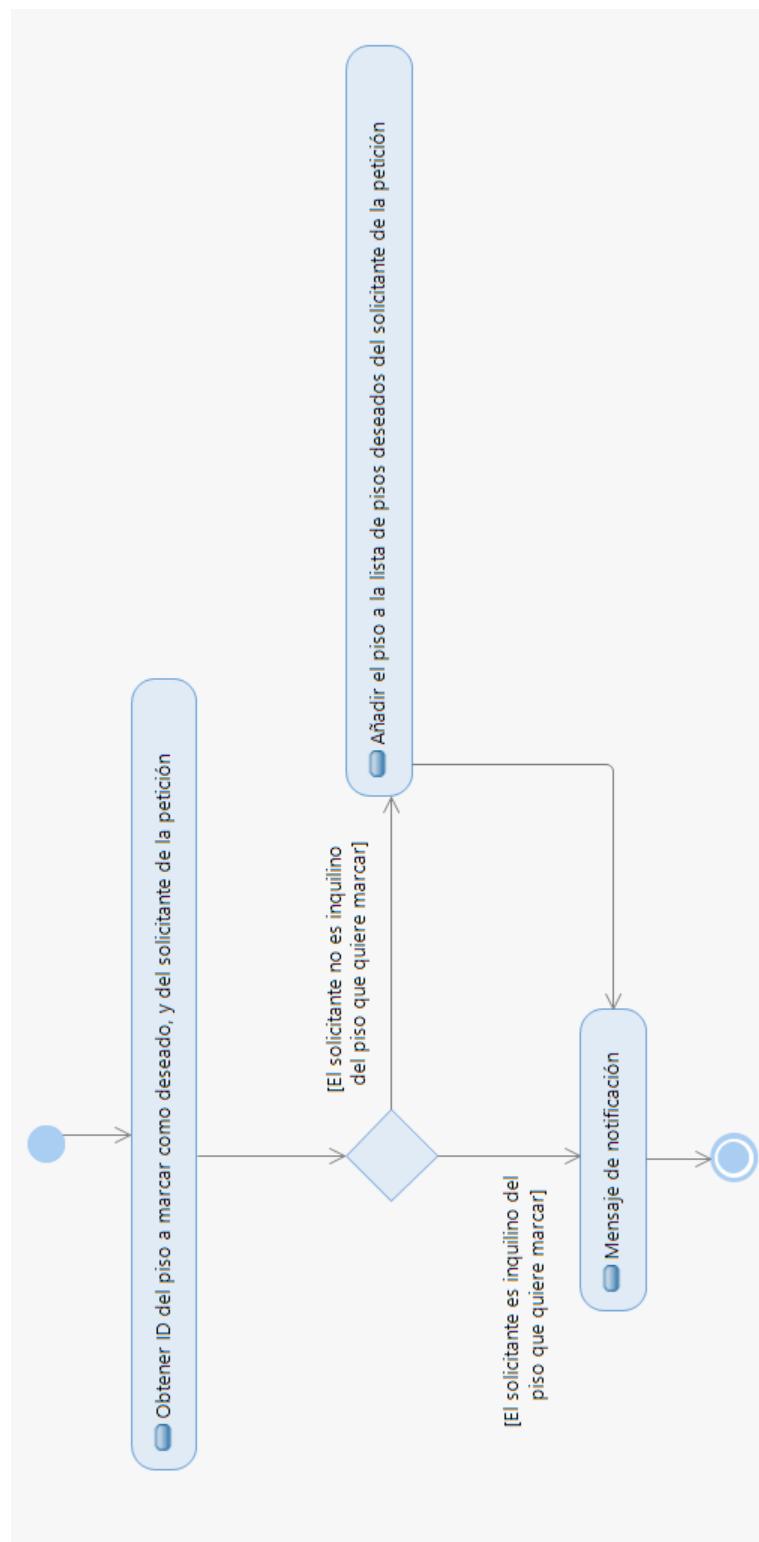
Tabla 24: Diagrama de actividad *Sistema de notificaciones*

Función	Buzón de sugerencias y problemas técnicos
Código	RG22
Prioridad	Media
Estabilidad	Media
Descripción	Vía de contacto de los usuario con los administradores
Entrada	ID de usuario
Salida	Mensaje de éxito / error.
Origen	GUI.
Destino	Sistema.
Precondición	Usuario logeado
Postcondición	Se envía su mensaje al buzón de sugerencias y problemas técnicos.
Acción	El sistema almacena el mensaje en la lista de sugerencias y problemas técnicos.
Flujo de evento	<ol style="list-style-type: none"> 1. Se abre el hipervínculo desde la GUI hacia el buzón de sugerencias y problemas técnicos. 2. Se completa el contenido del mensaje. 3. Se pulsa el botón enviar mensaje. 4. Si el mensaje se envía correctamente: <ol style="list-style-type: none"> a) True : Mensaje de éxito. b) False : Mensaje de error.

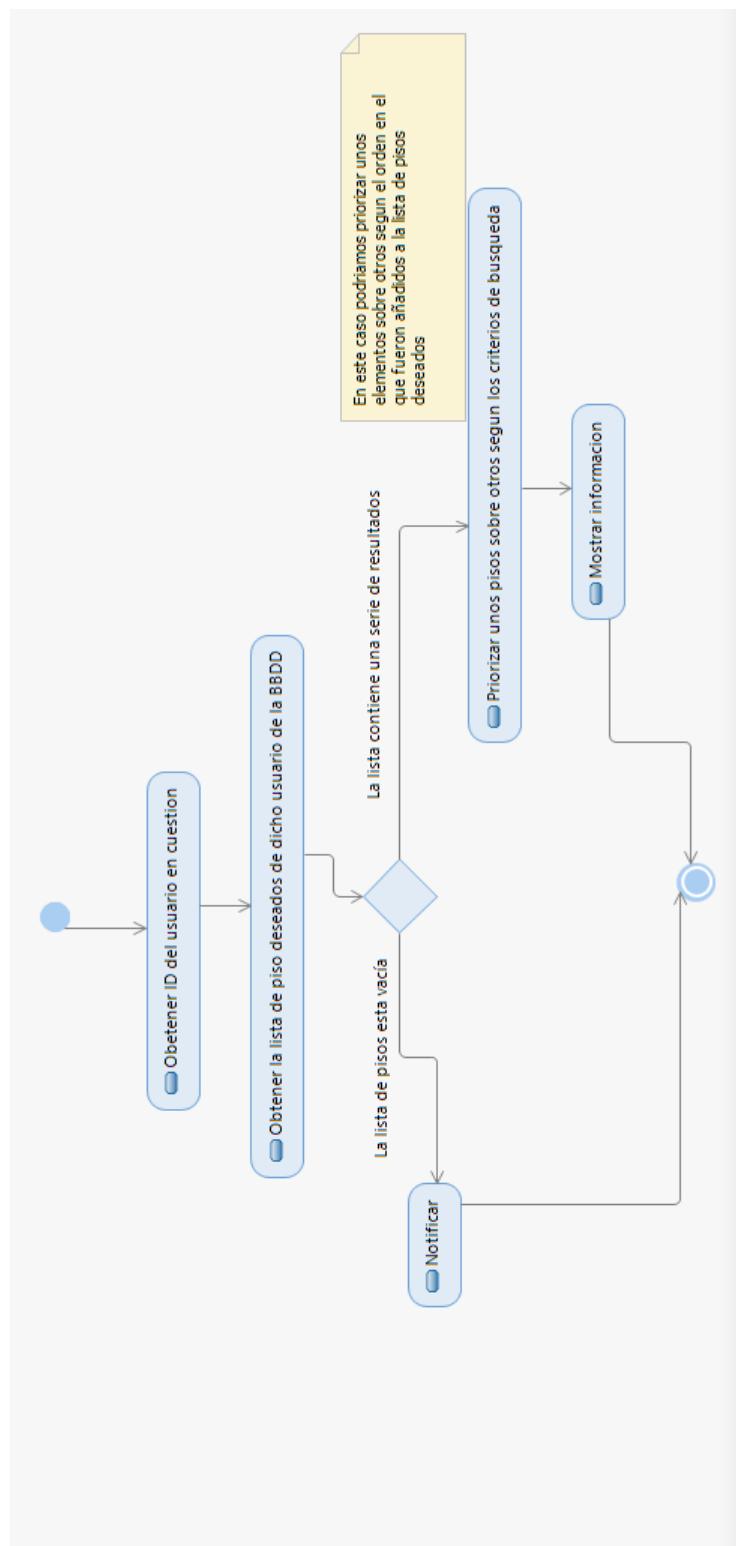
Tabla 25: Diagrama de actividad *Buzón de sugerencias y problemas técnicos*

3.2.2. Requisitos de Buscar Piso

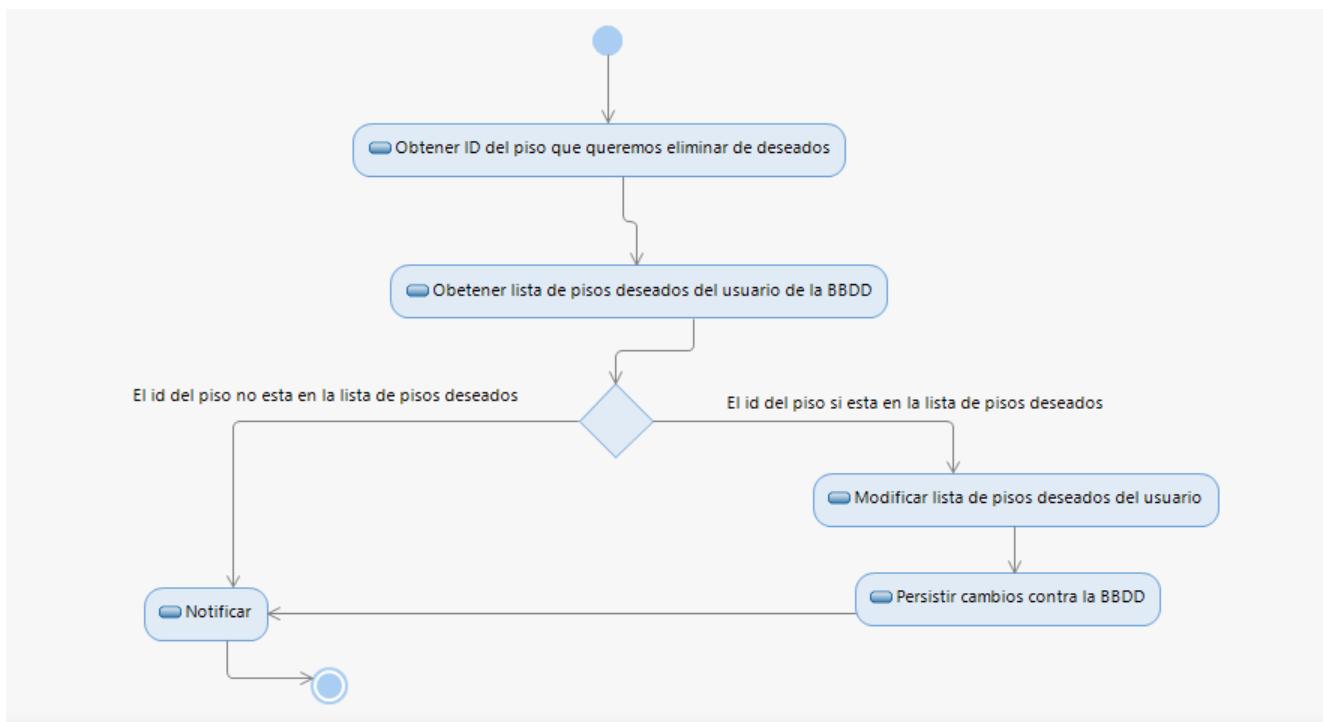
Función	Marcar piso como deseado
Código	RBP01
Prioridad	Media
Estabilidad	Media
Descripción	El usuario marca un piso como “deseado”, es decir, lo añade a su lista de pisos favoritos en los que le gustaría alojarse. La lista de pisos deseados es individual para cada usuario.
Entrada	ID del piso deseado
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Dicho piso no lo teníamos ya agregado a la lista de pisos deseados.
Postcondición	Vínculo de compañero potencial roto, entre el compañero potencial a eliminar y el usuario que solicita eliminarlo
Acción	Eliminar el vínculo de compañeros potenciales entre el compañero potencial a eliminar y el usuario que solicita eliminarlo.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos el botón de marcar piso como deseado. 2. Obtenemos los id del emisor y de la casa. 3. El solicitante es inquilino de la misma casa. <ol style="list-style-type: none"> a) True : Mensaje de notificación indicando el error. b) False : Añadimos el piso a la lista de deseados del usuario.

Tabla 26: Diagrama de actividad *Marcar piso como deseado*

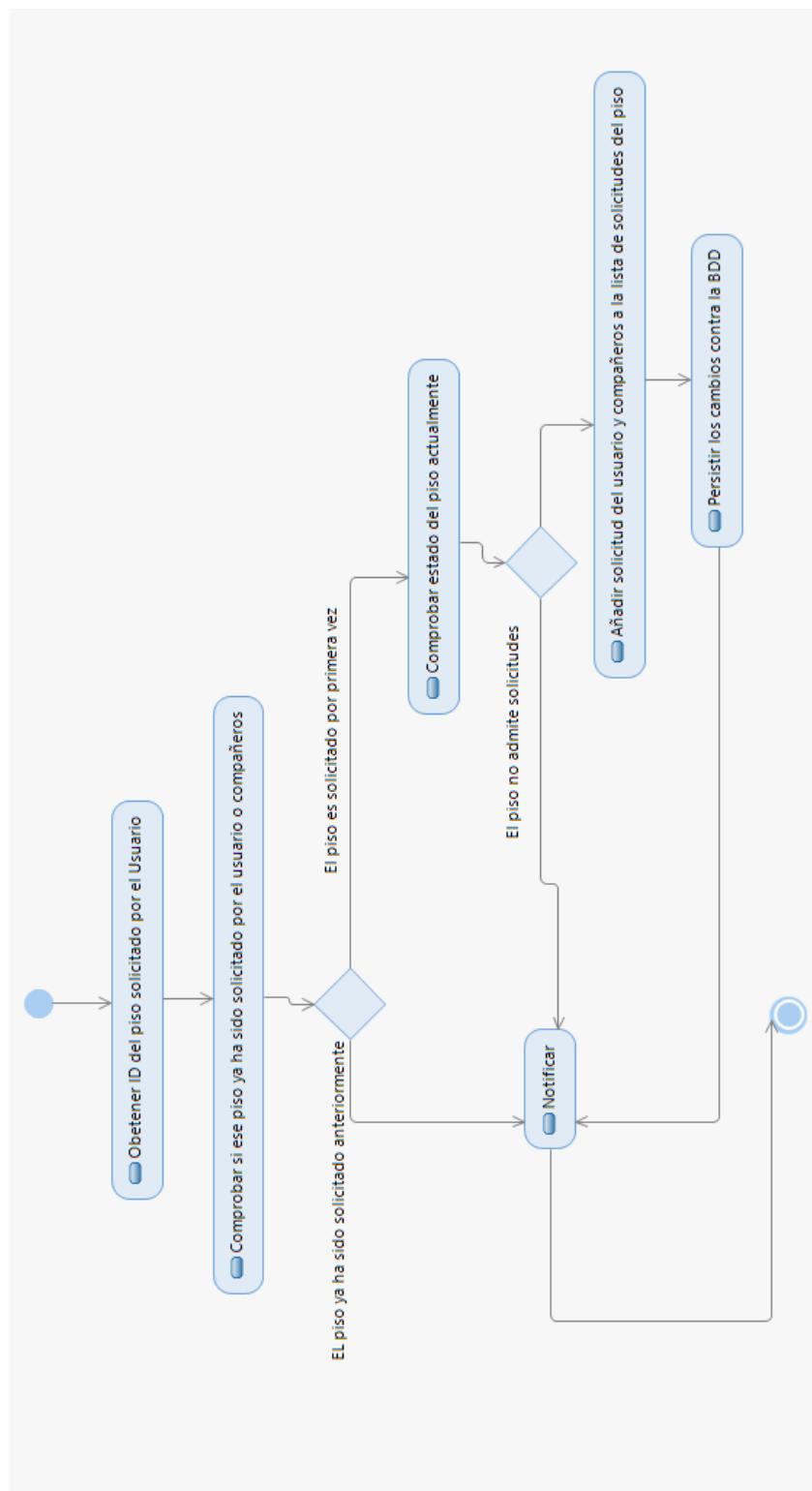
Función	Mostrar lista de pisos deseados de un usuario
Código	RBP02
Prioridad	Baja
Estabilidad	Media
Descripción	Se mostrará la lista de pisos “deseados” de un usuario, es decir, los pisos favoritos en los que le gustaría alojarse.
Entrada	ID del usuario
Salida	Lista de pisos deseados.
Origen	GUI
Destino	GUI
Precondición	El usuario debe estar logueado y lista de pisos deseados existente.
Postcondición	La lista de pisos deseados por el usuario fue mostrada.
Acción	Se muestra la lista de pisos deseados del usuario introducido.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos el botón para mostrar todos los pisos deseados. 2. Obtenemos el ID del usuario y buscamos en la BBDD su lista de deseados: <ol style="list-style-type: none"> a) Si la lista existe, mostramos los pisos deseados. b) Si la lista no existe, mostramos un mensaje de error.

Tabla 27: Diagrama de actividad *Mostrar lista de pisos deseados de un usuario*

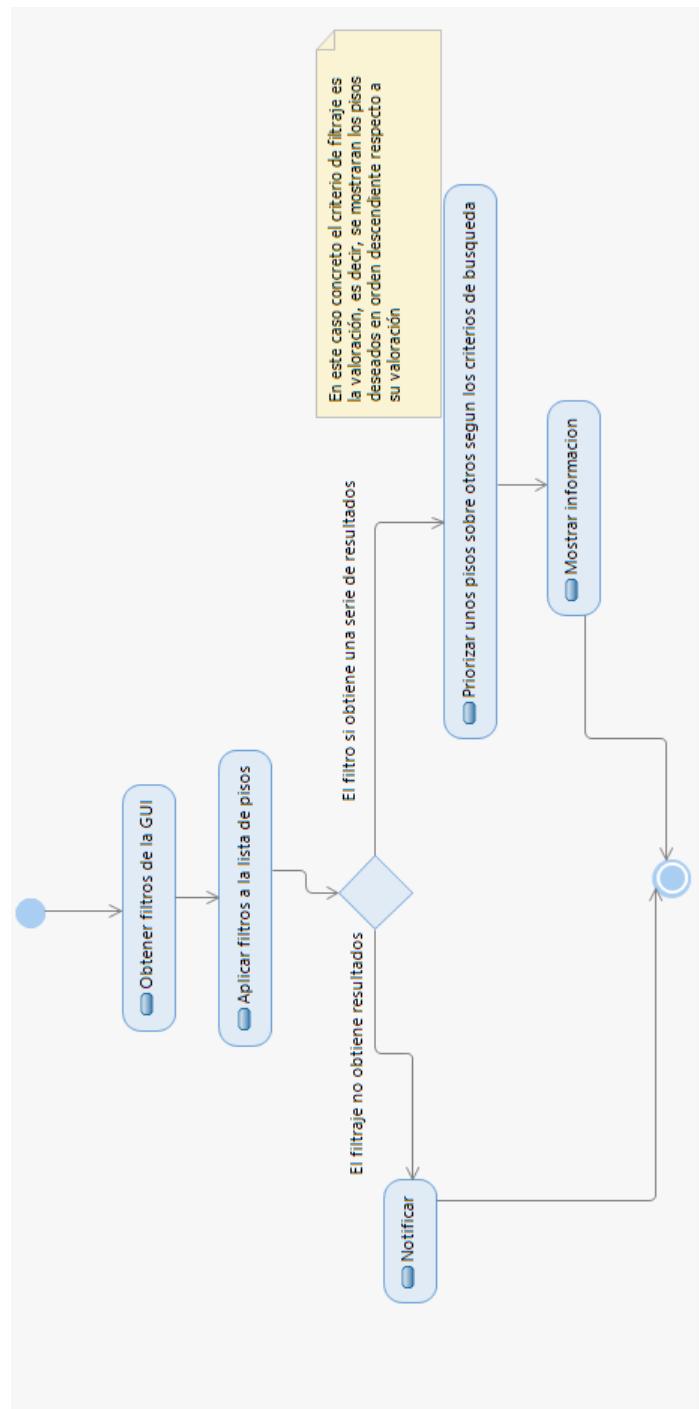
Función	Eliminar piso como deseado
Código	RBP03
Prioridad	Media
Estabilidad	Media
Descripción	Una vez el usuario ha seleccionado un piso como “deseado”, también será posible deshacer esta selección, eliminando así la preferencia del usuario por el piso.
Entrada	ID del piso deseado, ID del usuario.
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Piso este agregado en la lista de pisos deseados.
Postcondición	Piso eliminado de la lista de pisos deseados.
Acción	Se elimina la preferencia por ese piso de la lista de pisos deseados.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos el botón de desmarcar piso deseado. 2. Se obtienen el ID del piso y el ID del usuario. 3. El ID del piso no se encuentra en la lista de deseados. <ol style="list-style-type: none"> a) True : Mensaje de notificación indicando el error. b) False : Eliminamos el piso de la lista de deseados.

Tabla 28: Diagrama de actividad *Eliminar piso como deseado*

Función	Solicitar piso
Código	RBP04
Prioridad	Alta
Estabilidad	Alta
Descripción	Enviar una propuesta al ofertante del piso (que podría ser inquilino o propietario) para acceder al piso
Entrada	Nombre del usuario solicitante.
Salida	Mensaje
Origen	GUI
Destino	Sistema
Precondición	Piso actualmente no solicitado por dicho usuario o por sus compañeros potenciales.
Postcondición	La oferta es enviada al ofertante del piso y a todos los compañeros potenciales que tenga el solicitante; los compañeros potenciales quedan marcados como solicitantes del piso.
Acción	Se envía la solicitud al ofertante.
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos en el botón de solicitar piso. 2. Obtenemos el ID del piso solicitado por el usuario. 3. Una vez obtenido el ID comprobamos si el piso ha sido solicitado anteriormente <ol style="list-style-type: none"> a) En caso afirmativo notificamos con un error al usuario b) En otro caso añadimos al usuario solicitante y a sus compañeros a la lista de solicitantes del piso

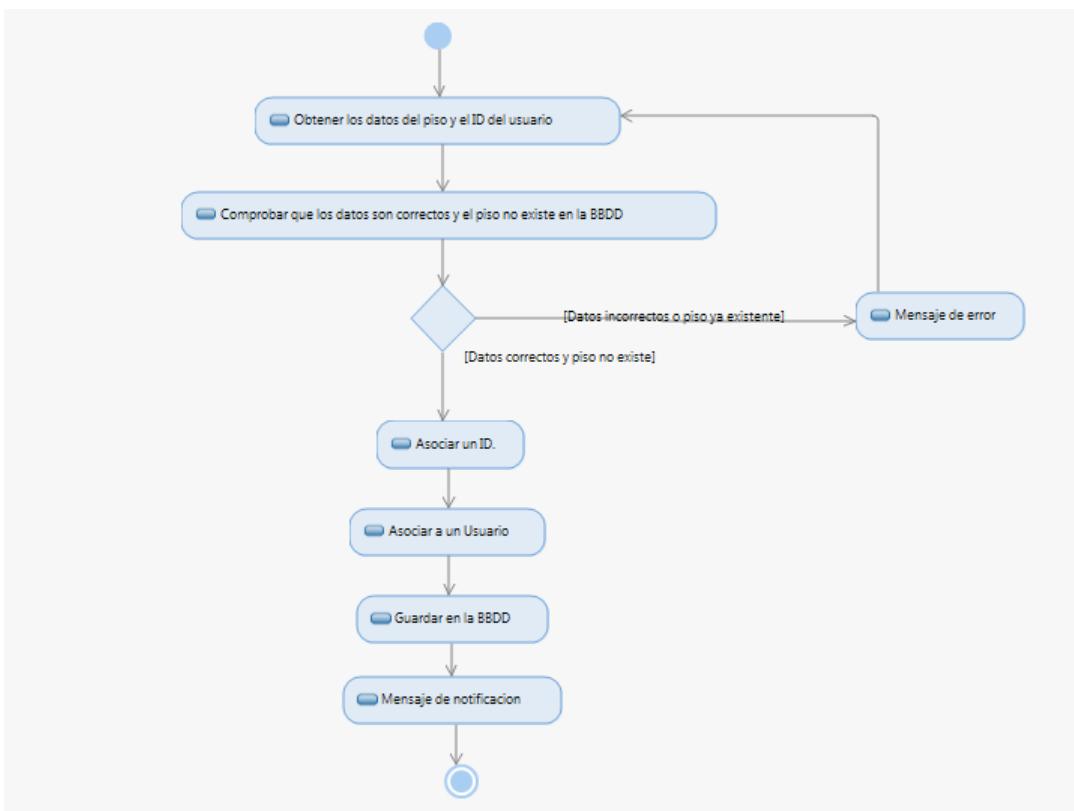
Tabla 29: Diagrama de actividad *Solicitar piso*

Función	Mostrar pisos por valoración
Código	RBP05
Prioridad	Media
Estabilidad	Baja
Descripción	Da un resultado en forma de lista, con todos los pisos de entre los disponibles con los filtros previamente establecidos, ordenados de mayor/menor valoración
Entrada	Filtros previos (Podríamos haber buscado sin filtros)
Salida	Lista de pisos ordenados de mayor/menor valoración
Origen	GUI
Destino	Sistema
Precondición	-
Postcondición	Lista de pisos ordenados de mayor/menor valoración
Acción	Se muestra la lista de pisos disponibles ordenados de mayor/menor valoración
Flujo de evento	<ol style="list-style-type: none"> 1. En la GUI pulsamos en el botón de buscar(una vez los filtros son aplicados) 2. Una vez aplicado el filtro existen dos posibilidades <ol style="list-style-type: none"> a) En caso de que el filtraje no produzca resultados se lo comunicamos al usuario b) En otro caso priorizamos unos pisos sobre otros según los criterios que haya considerado el usuario, y los mostramos por pantalla.

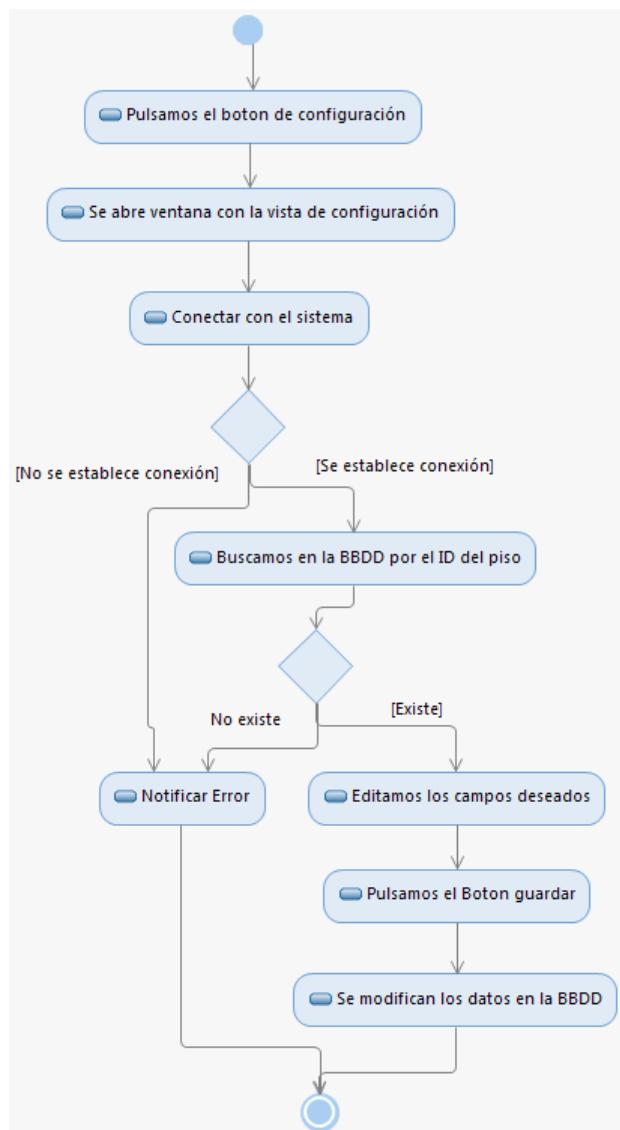
Tabla 30: Diagrama de actividad *Mostrar pisos por valoración*

3.2.3. Requisitos de Buscar inquilino

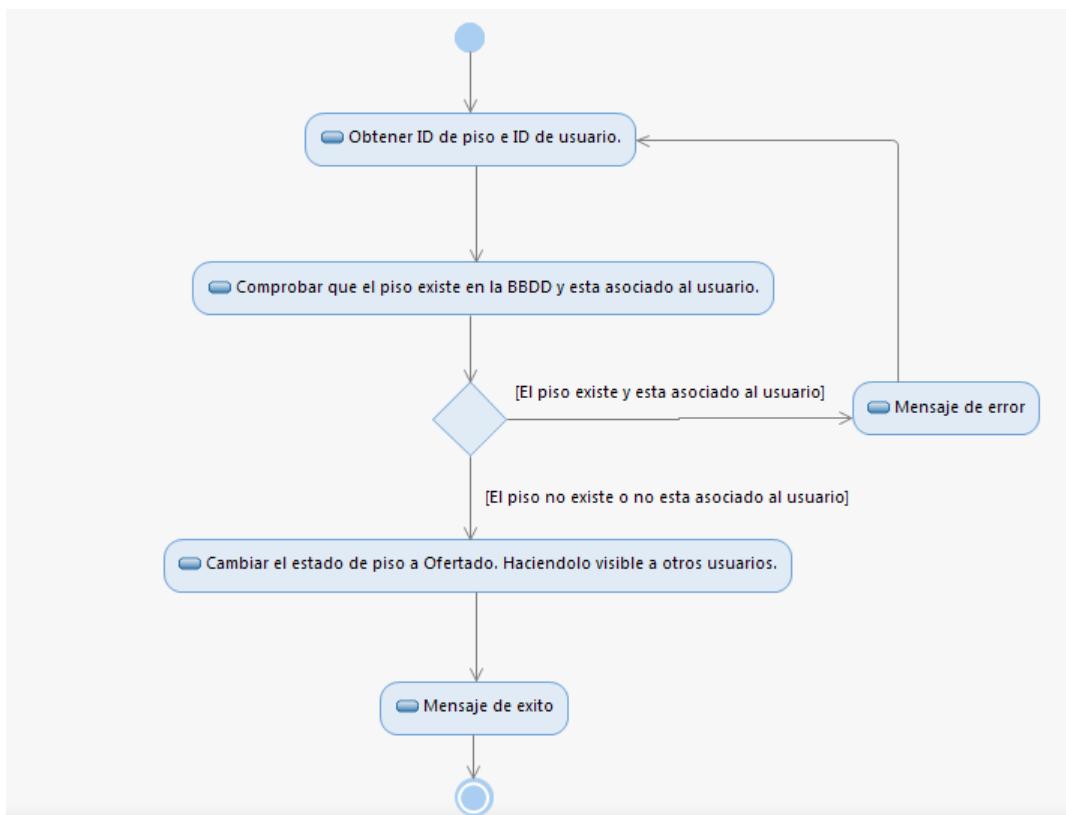
Función	Registrar Piso
Código	RBI01
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario registra un piso en el sistema.
Entrada	Datos del piso, ID del usuario
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Los datos del piso son correctos. El piso no puede estar ya registrado en el sistema.
Postcondición	Añade el piso a la BBDD
Acción	Registra un piso en el sistema
Flujo de evento	<ol style="list-style-type: none"> 1. En la pantalla principal de pisos, el usuario selecciona el botón registrar piso. 2. La interfaz cambia para mostrar la pantalla de “Registrar piso”. El usuario introduce los datos del piso. 3. Una vez insertados los datos, el usuario selecciona el botón añadir al sistema. 4. La aplicación comprueba que los datos son correctos y no existe ningún piso con esos datos (calle, número, piso, código postal) en la BBDD: <ol style="list-style-type: none"> a) Si existe: se muestra mensaje de error y vuelve al 2. b) Si no existe: internamente se le asocia un ID y se agrega el piso a la BBDD. Para finalizar se muestra un mensaje notificando el éxito de la operación.

Tabla 31: Diagrama de actividad *Registrar piso*

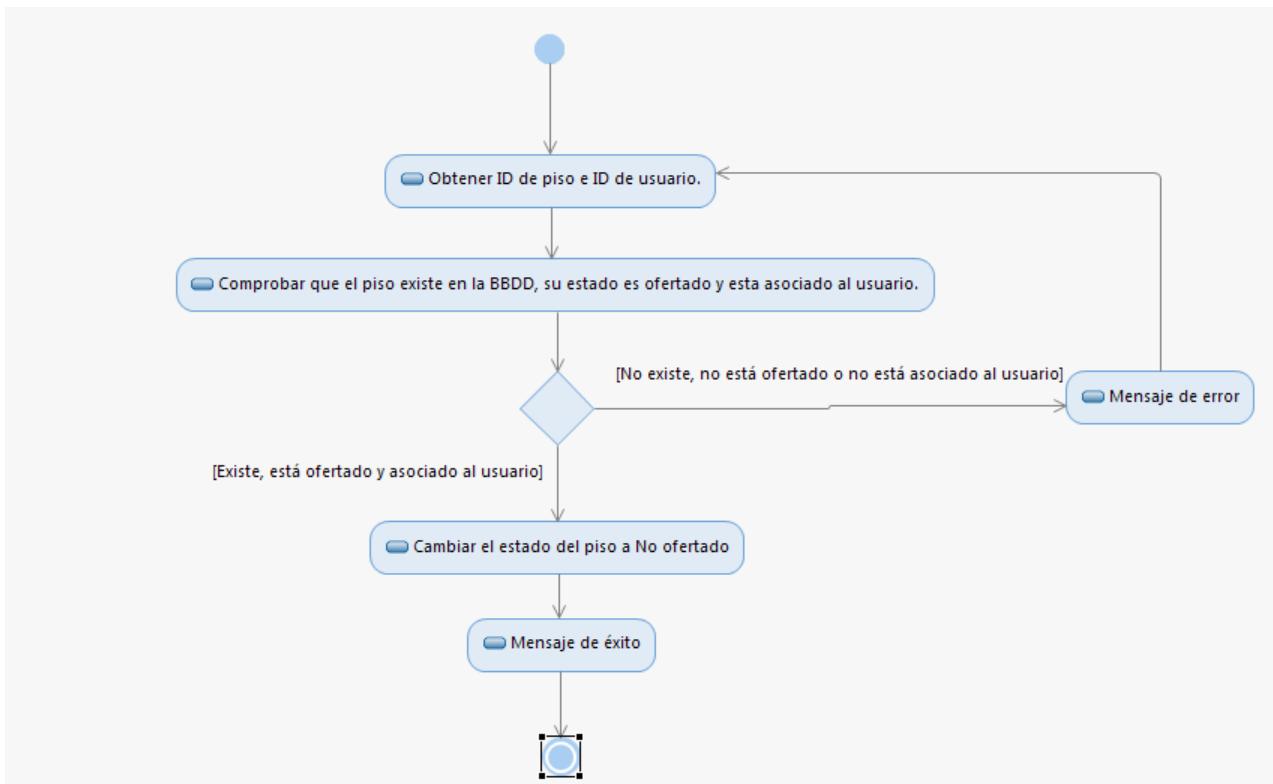
Función	Modificar datos de piso
Código	RBI02
Prioridad	Alta
Estabilidad	Alta
Descripción	Modifica los datos de un piso.
Entrada	Datos modificables del piso
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	Que el piso exista en el sistema
Postcondición	Datos del piso modificados correctamente
Acción	Modifica los datos de un piso
Flujo de evento	<ol style="list-style-type: none"> 1. En el listado de pisos propios pulsamos en el botón Modificar piso 2. La interfaz cambia a la pantalla “Modificar piso” y nos muestra un pequeño cuadro para poder editar los datos del piso. Solo se puede editar las características, contrato y extras. 3. El operador del sistema puede elegir: <ol style="list-style-type: none"> a) Descartar cambios: la aplicación no modifica nada y vuelve a la pantalla principal de configuración de la cuenta. b) Guardar cambios: la aplicación actualiza la información en la BBDD y muestra un mensaje notificando el éxito de la operación.

Tabla 32: Diagrama de actividad *Modificar datos de piso*

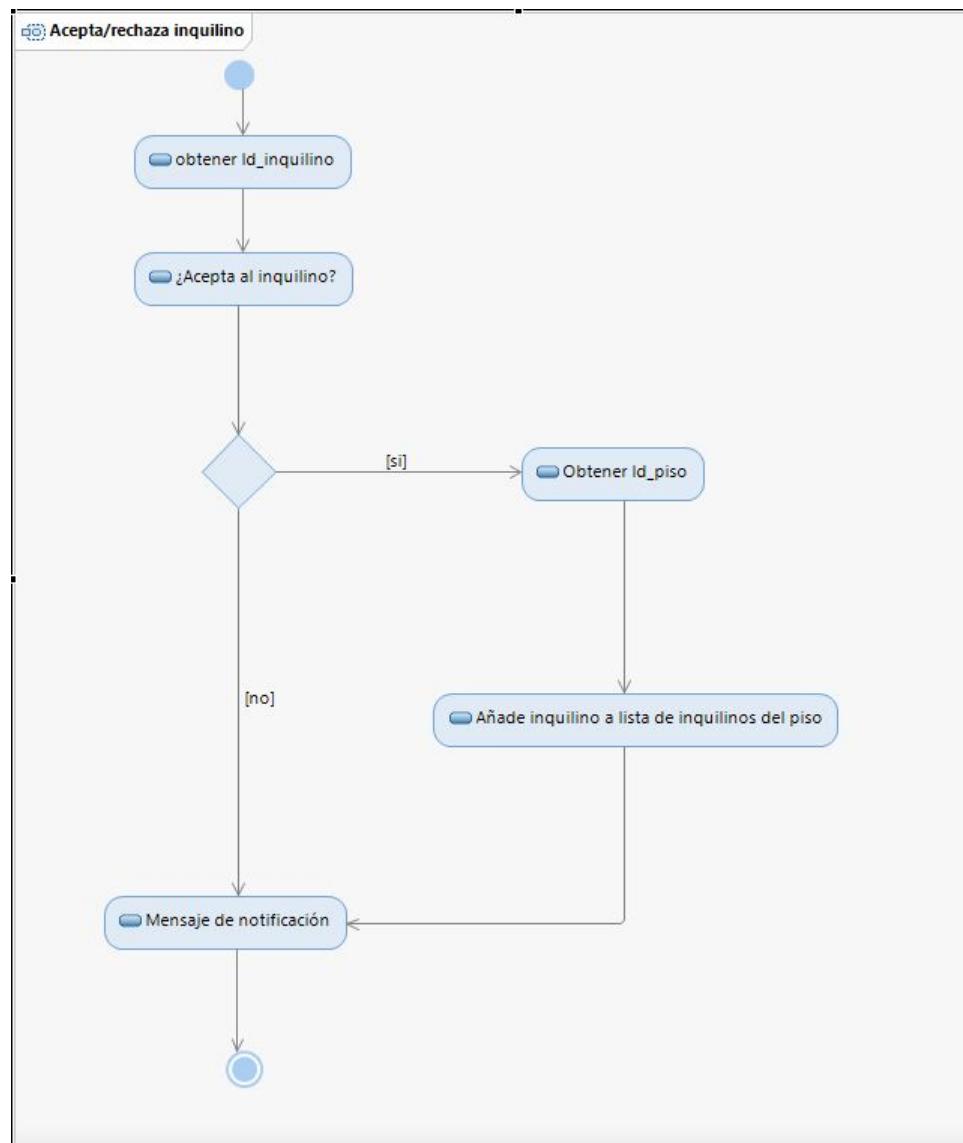
Función	Ofertar Piso
Código	RBI03
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario oferta un piso ya registrado.
Entrada	ID Piso y de ID Usuario
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	El ID del piso y del usuario existen en la BBDD. El piso ha sido registrado por ese mismo usuario previamente.
Postcondición	Cambia el estado del piso a ofertado, haciendo visible el piso para el resto de los usuarios.
Acción	Oferta un piso en el sistema.
Flujo de evento	<ol style="list-style-type: none"> 1. En la pantalla principal de pisos, el usuario selecciona el botón Ofertar piso. 2. La interfaz cambia para mostrar la pantalla de “Ofertar piso”. El usuario selecciona el piso a ofertar. 3. Una vez insertados los datos, el usuario selecciona el botón aceptar. 4. La aplicación comprueba que el piso existe en la BBDD, está asociado a ese usuario y que tiene espacio para al menos un inquilino más: <ol style="list-style-type: none"> a) No correcto: Se muestra mensaje de error y vuelve al 2. b) Todo correcto: Se cambia el estado del piso a “Ofertado” lo que lo hace visible para los demás usuarios del sistema. Para finalizar se muestra un mensaje notificando el éxito de la operación.

Tabla 33: Diagrama de actividad *Ofertar piso*

Función	Eliminar oferta de piso
Código	RBI04
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario cambia el estado de un piso previamente ofertado a “No ofertado”.
Entrada	ID Piso y de ID Usuario
Salida	Mensaje de éxito/error.
Origen	GUI
Destino	Sistema
Precondición	El piso existe en la base de datos. El piso ha sido ofertado previamente por el mismo usuario.
Postcondición	El estado del piso es “No ofertado”, por tanto, invisible el piso para el resto de los usuarios. Las posibles peticiones de inquilino sobre el piso serán eliminadas con notificación a los interesados.
Acción	Elimina la oferta de un piso.
Flujo de evento	<ol style="list-style-type: none"> 1. En la pantalla principal de pisos, el usuario selecciona el botón “Eliminar piso ofertado”. 2. La interfaz cambia para mostrar la pantalla de “Eliminar piso ofertado”. El usuario selecciona el piso a “eliminar”. 3. Una vez insertados los datos, el usuario selecciona el botón aceptar. 4. La aplicación comprueba que el piso existe en la BBDD, está ofertado y está asociado a ese usuario: <ol style="list-style-type: none"> a) No correcto: Se muestra mensaje de error y vuelve al 2. b) Todo correcto: <ol style="list-style-type: none"> 1) Se cambia el estado del piso a “No ofertado” lo que lo hace invisible para los demás usuarios del sistema. 2) Después, se eliminan las posibles “Solicitudes de inquilino” que tuviese dicho piso y se envían notificaciones a los interesados. 3) Para finalizar se muestra un mensaje notificando el éxito de la operación.

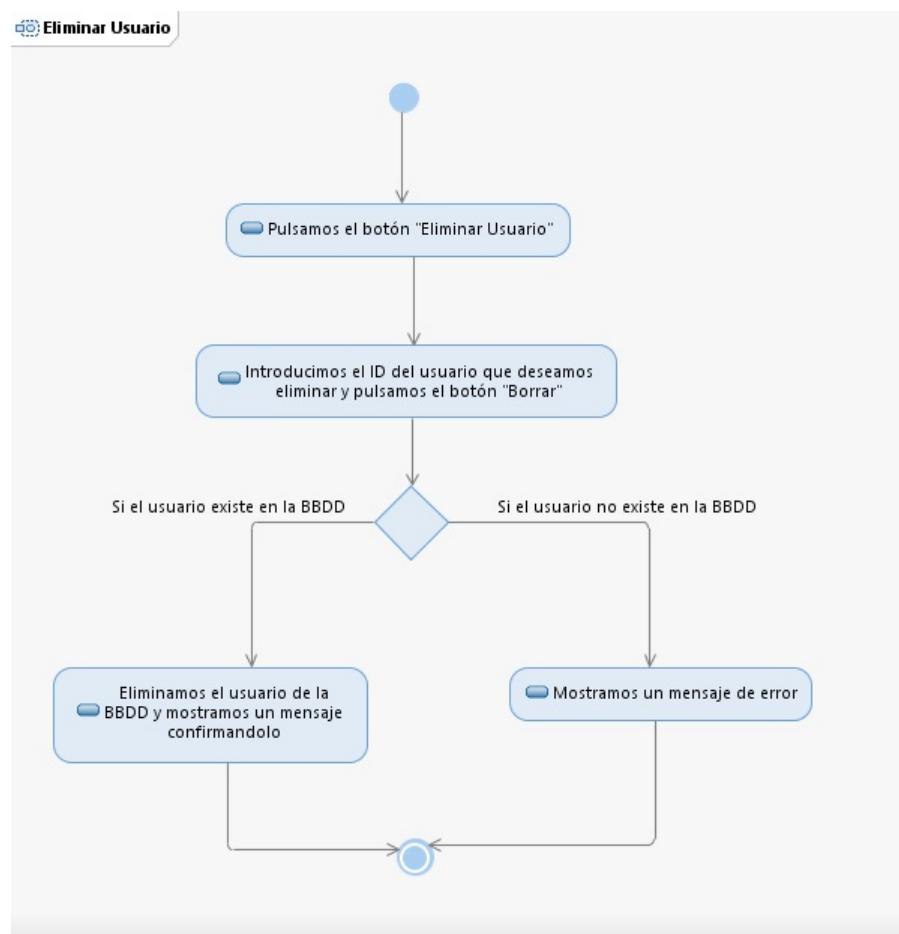
Tabla 34: Diagrama de actividad *Eliminar oferta de piso*

Función	Aceptar/rechazar inquilino
Código	RBI05
Prioridad	Alta
Estabilidad	Alta
Descripción	Acepta o rechaza una petición de inquilino
Entrada	ID del piso, del usuario que solicita y del que recibe la petición
Salida	Mensaje
Origen	GUI
Destino	Sistema
Precondición	El piso tiene espacio para al menos un inquilino más. El inquilino no puede estar alojado en ningún otro piso.
Postcondición	Asociado o no un usuario a un piso y sus inquilinos
Acción	Asocia o no un inquilino más al piso
Flujo de evento	<p>1. El ofertante del piso recibe sus peticiones de inquilino en su bandeja de peticiones. Cada una de ellas contendrá los datos de cada uno de los usuarios que han enviado dichas peticiones junto con dos botones</p> <p>a) "Aceptar": Si el ofertante pulsa este, el demandante pasará a aparecer como inquilino del piso y se le enviará una notificación. Sin embargo, el demandante tendrá que contactar con el ofertante para concretar los detalles sobre el contrato, pudiendo el ofertante rechazar al inquilino si no llegasen a un acuerdo.</p> <p>b) "Rechazar": Si el ofertante pulsa este botón, se eliminará la petición del demandante y se le notificará mediante un mensaje.</p>

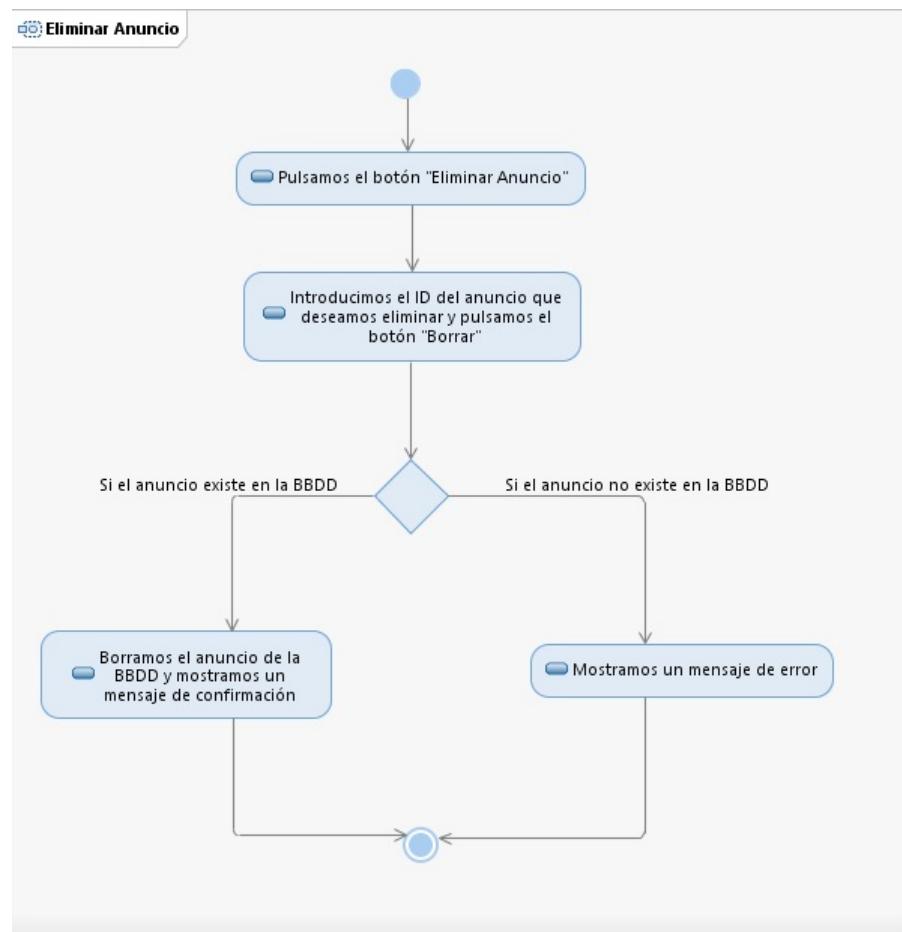
Tabla 35: Diagrama de actividad *Aceptar/rechazar inquilino*

3.2.4. Requisitos de Adiministración

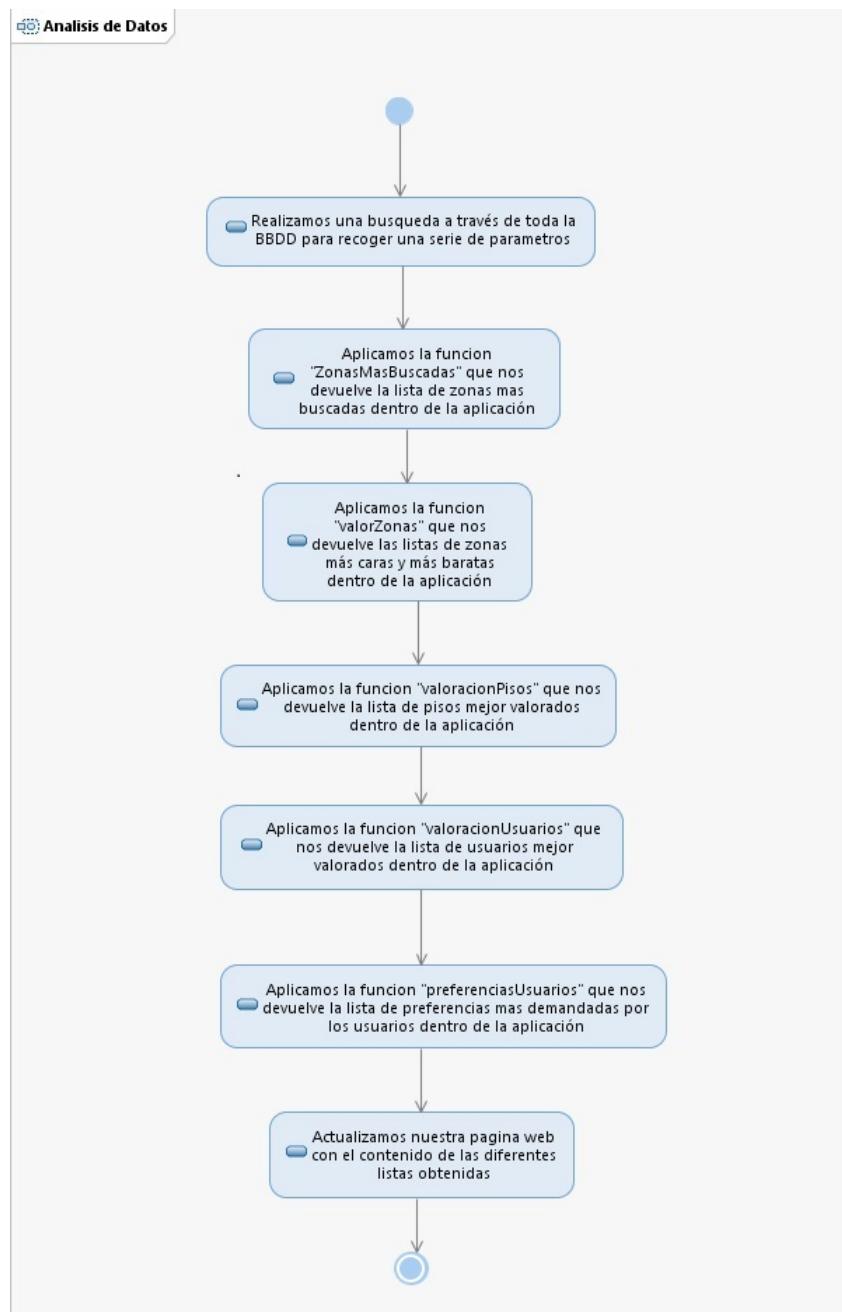
Función	Eliminar usuario
Código	RBI06
Prioridad	Alta
Estabilidad	Alta
Descripción	Elimina un usuario del sistema.
Entrada	ID del usuario
Salida	Mensaje de aviso y, al finalizar, de éxito
Origen	GUI
Destino	Sistema
Precondición	El usuario debe estar dado de alta en el sistema (BBDD).
Postcondición	Eliminar al usuario de la BBDD y las ofertas de piso que tenga.
Acción	Borra un usuario del sistema y las ofertas de piso que tenga.
Flujo de evento	<ol style="list-style-type: none"> 1. El administrador del sistema pulsa el botón de “Eliminar Usuario”. 2. A continuación, saldrá una ventana emergente que nos pedirá un ID de usuario: <ol style="list-style-type: none"> a) Introducimos el ID del usuario que queremos eliminar y pulsamos el botón ”Borrar”: <ol style="list-style-type: none"> 1) Si el usuario existe y cumple las precondiciones, se elimina del sistema. 2) Si el usuario no existe, se notifica mediante una ventana emergente. b) Pulsamos el botón Cancelar o cerramos la ventana: volvemos al menú del Administrador.

Tabla 36: Diagrama de actividad *Eliminar usuario*

Función	Eliminar anuncio
Código	RBI07
Prioridad	Alta
Estabilidad	Alta
Descripción	Elimina el anuncio de un piso del sistema.
Entrada	ID del anuncio
Salida	Mensaje de aviso y, al finalizar, de éxito
Origen	GUI
Destino	Sistema
Precondición	El anuncio debe existir en la BBDD.
Postcondición	Elimina el anuncio de la BBDD
Acción	Borra un anuncio del sistema.
Flujo de evento	<p>1. El administrador del sistema pulsa el botón de “Eliminar Anuncio”.</p> <p>2. A continuación, saldrá una ventana emergente que nos pedirá un ID del anuncio:</p> <p>a) Introducimos el ID del anuncio que queremos eliminar y pulsamos el botón ”Borrar”:</p> <p>1) Si el anuncio existe se elimina del sistema.</p> <p>2) Si el anuncio no existe, se notifica mediante una ventana emergente.</p> <p>b) Pulsamos el botón Cancelar o cerramos la ventana: volvemos al menú del Administrador.</p>

Tabla 37: Diagrama de actividad *Eliminar anuncio*

Función	Análisis de datos
Código	RBI08
Prioridad	Baja
Estabilidad	Media
Descripción	Analiza los datos de los usuarios del sistema y de los pisos ofertados para calcular unas estadísticas que sirvan de utilidad para los usuarios.
Entrada	Datos de la BBDD
Salida	<p>Listas de:</p> <ul style="list-style-type: none"> ▪ Zonas más buscadas ▪ Zonas más baratas ▪ Zonas más caras ▪ Pisos mejor valorados ▪ Usuarios mejor valorados ▪ Preferencias de los usuarios
Origen	Sistema
Destino	Sistema
Precondición	- -
Postcondición	- -
Acción	Analiza diferentes datos referentes al sistema relativos a los usuarios y a los pisos ofertados.
Flujo de evento	Se realiza una búsqueda de información en la BBDD que luego es procesada por diferentes funciones. Estas funciones nos devolverán una serie de listas de información que serán publicadas posteriormente en nuestra página web.

Tabla 38: Diagrama de actividad *Análisis de datos*

3.3. Requisitos de Rendimiento

La BBDD estará alojada en un servidor dedicado 24/7 contratado para evitar tener que acarrear los gastos y adaptaciones de la infraestructura por motivos de la protección de datos.

La BBDD se modificará cada vez que se produzca una variación en la información almacenada en dicha BBDD.

Los datos se almacenarán hasta el máximo que nos permita nuestra BBDD, aunque antes de llegar a tal extremo se ampliará el contrato con el propietario del servidor para mejorar el servicio.

Además el sistema deberá ser capaz de afrontar multitud de peticiones simultaneas y mantener la integridad de los datos.

3.4. Requisitos lógicos de la BBDD

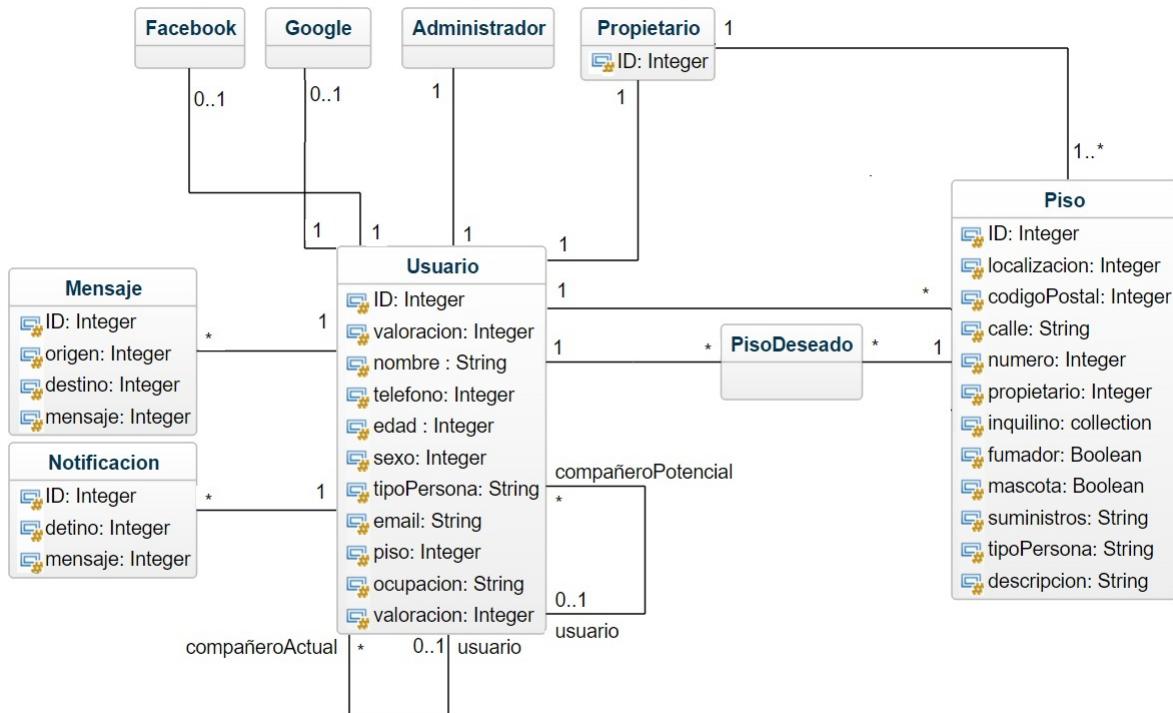


Tabla 39: *Modelo del Dominio*

3.5. Restricciones de diseño

■ Restricciones I

- Es obligatorio aplicar técnicas de IS.

■ Restricciones II

- La entrega del proyecto (SRS y PdP y demás documentación pertinente) será el día 31/5/2016. En el «Plan de Proyecto» se detalla todo lo referente a la entrega del proyecto.

■ Restricciones III

- La arquitectura de la aplicación será multicapa.

■ Restricciones IV

- El proyecto debe realizarse en equipo.
- El número de miembros de cada equipo debe ser once (11).
- En casos excepcionales, y previa consulta con el profesor de esta asignatura se podría permitir la existencia de algún equipo de otro tamaño.
- En caso de existir equipos de alumnos de tamaño distinto de once (11), el profesor se reserva el derecho de modificar la composición de estos equipos.
- No se admiten entregas después de acabar la sesión de la asignatura correspondiente al día de cada entrega.

■ Restricciones V

- El día de la entrega segunda y tercera se procederá a una ejecución del proyecto para comprobar la implementación de los requisitos.
- Todas las entregas deben hacerse mediante una carpeta compartida con el profesor en Google Drive.
- La existencia de cualquier tipo de virus en cualquier soporte informático entregado al profesor invalidará la entrega.
- A pesar de no exigir actas de las reuniones realizadas por el equipo, se exhorta y anima a los alumnos a su realización.
- Los alumnos (en concreto el responsable de la comunicación equipo-Profesores) son los únicos responsables del contenido almacenado en la carpeta compartida con los profesores en Google Drive. La presencia de contenido inadecuado en dicha carpeta conllevará su borrado y la invalidación del proyecto.

3.6. Atributos del sistema

▪ **Fiabilidad**

- El sistema debe tener una baja probabilidad de fallos.

▪ **Disponibilidad**

- La disponibilidad del sistema debe de ser continuo, aunque el sistema de quejas solo estará disponible en horario laboral.

▪ **Seguridad**

- Garantizar la seguridad del sistema con respecto a la información contenida en la base de datos.
- Garantizar la confiabilidad, la seguridad de que la información almacenada en la base de datos podrán ser consultados y actualizados permanentemente, sin que se pierdan datos ni que afecte al tiempo de respuesta del sistema.

▪ **Mantenibilidad**

- Además, la implementación del programa vendrá elaborada mediante patrones de programación lo que hará que en un futuro la aplicación sea fácilmente actualizable y pueda ser mantenida con mínimo coste y esfuerzo.
- Contamos también con que dispondremos de una documentación fácilmente actualizable al igual que el código, que permitirá ir haciendo operaciones de mantenimiento.

▪ **Interfaz amigable con el usuario**

- La interfaz será intuitiva para facilitar al máximo el trabajo del usuario.