



# Documento funcional APP - Chat

De:

**Juan Ignacio Avendaño Huergo**

[juan.avendano@tecandweb.com](mailto:juan.avendano@tecandweb.com)

Escoles universitaries Gimbernat



## Para Tecnologia Multimèdia i Dispositius Mòbils

Fecha de presentación: viernes, 21 de abril de 2017



## 1. Contenido

---

1.	Contenido .....	3
2.	Requerimientos no funcionales .....	7
3.	Casos de uso .....	8
1.	Registro .....	9
	Flujo principal .....	9
	Flujos alternativos .....	9
a.	Alta usuario en servidor .....	10
	Flujo principal .....	10
	Flujos alternativos .....	10
	Reglas de negocio .....	10
	Diagrama de interacción .....	11
2.	Autenticación .....	12
	Flujo principal .....	12
	Flujos alternativos .....	12
a.	Validación de login con servidor .....	13
	Flujo principal .....	13
	Flujos alternativos .....	13
	Reglas de negocio .....	13
	Diagrama de interacción .....	14
b.	Guardar datos de registro en teléfono e iniciar escucha en servidor .....	15
	Flujo principal .....	15
	Flujos alternativos .....	15
3.	Abrir o actualizar actividad principal .....	16
	Flujo principal .....	16
	Flujos alternativos .....	16
a.	Construir conversaciones .....	17
	Flujo principal .....	17
	Flujos alternativos .....	17
	Diagrama de interacción .....	18



4.	Cerrar sesión.....	19
	Flujo principal .....	19
	Flujos alternativos .....	19
a.	Componente de Cierre de sesión .....	20
	Flujo principal .....	20
	Flujos alternativos .....	20
	Reglas de negocio.....	20
	Diagrama de interacción .....	21
5.	Dar de baja usuario .....	22
	Flujo principal .....	22
	Flujos alternativos .....	22
	Diagrama de interacción .....	23
6.	Leer mensaje .....	24
	Flujo principal .....	24
	Flujos alternativos .....	25
	Diagrama de interacción .....	25
a.	Recibir mensaje .....	26
	Flujo principal .....	26
	Flujos alternativos .....	26
b.	Abrir actividad de conversación .....	27
	Flujo principal .....	27
	Flujos alternativos .....	27
7.	Escribir mensaje .....	28
	Flujo principal .....	28
	Flujos alternativos .....	29
	Diagrama de interacción .....	30
8.	Modificar datos de usuario.....	31
	Flujo principal .....	31
	Flujos alternativos .....	31
	Reglas de negocio.....	31
	Diagrama de interacción .....	32



9.	Consultar Información de amigo.....	33
	Flujo principal .....	33
	Flujos alternativos .....	33
	Reglas de negocio.....	33
	Diagrama de interacción .....	34
10.	Borrar amigo.....	35
	Flujo principal .....	35
	Flujos alternativos .....	35
	Diagrama de interacción .....	36
11.	Añadir amigo. ....	37
	Flujo principal .....	37
	Flujos alternativos .....	37
	Diagrama de interacción .....	38
a.	Recibir aceptación de amistad .....	39
	Flujo principal .....	39
	Flujos alternativos .....	39
12.	Aceptar o rechazar solicitud de amistad. ....	40
	Flujo principal .....	40
	Flujos alternativos .....	40
	Diagrama de interacción .....	41
a.	Recibir solicitud de amistad .....	42
	Flujo principal .....	42
	Flujos alternativos .....	42
13.	Reiniciar clave.....	43
	Flujo principal .....	43
	Flujos alternativos .....	43
a.	Petición de reinicio de clave a servidor.....	44
	Flujo principal .....	44
	Flujos alternativos .....	44
	Diagrama de interacción .....	45
4.	Diagrama de clases servidor .....	46



5.	Diagrama de clases Dispositivo.....	47
6.	Wireframes Generales .....	48
	Actividad de Login.....	48
	Actividad de registro .....	49
	Actividad de inicio .....	50
	Actividad de solicitudes.....	51
	Actividad de conversación .....	52
	Actividad Perfil de amigo.....	53
	Actividad de perfil .....	54
7.	Workflow .....	55
8.	Wireframes específicos .....	55
1.	RegisterActivity.....	56
	Perfil de Amigo.....	57
	PerfilActivity.....	58
2.	LoginActivity .....	59
3.	InicioActivity.....	60
4.	ConversationActivity .....	61
5.	FriendsActivity.....	62
	Solicitudes.....	62
	Amigos .....	63



## 2. Requerimientos no funcionales

---

### RNF1: Baja de usuario

La petición de baja del usuario en el servidor ha de ser asíncrona.

- Prioridad: Media

### RNF2: Recepción de mensajes asíncrona

La escucha al servidor de nuevos mensajes y su consecuente recepción de mensajes en el dispositivo ha de ser asíncrona.

- Prioridad: Alta

### RNF3: Orden de mensajes

Las conversaciones se ordenarán según la que contenga el mensaje más nuevo, ya sea emitido o recibido por el usuario de la aplicación.

- Prioridad: alta

### RNF4: Conversaciones

Los mensajes estarán dentro de conversaciones. Estas conversaciones serán las ventanas de chat, que dependen de con quien estemos hablando. Cada contacto con el que hayamos conversado tendrá una conversación asociada, que se mostrarán en el menú principal.

- Prioridad: Alta

### RNF5: Borrado de amigo

La petición de borrado de amigo en el servidor ha de ser asíncrona.

- Prioridad: Media

### RNF6: Usuario y Server

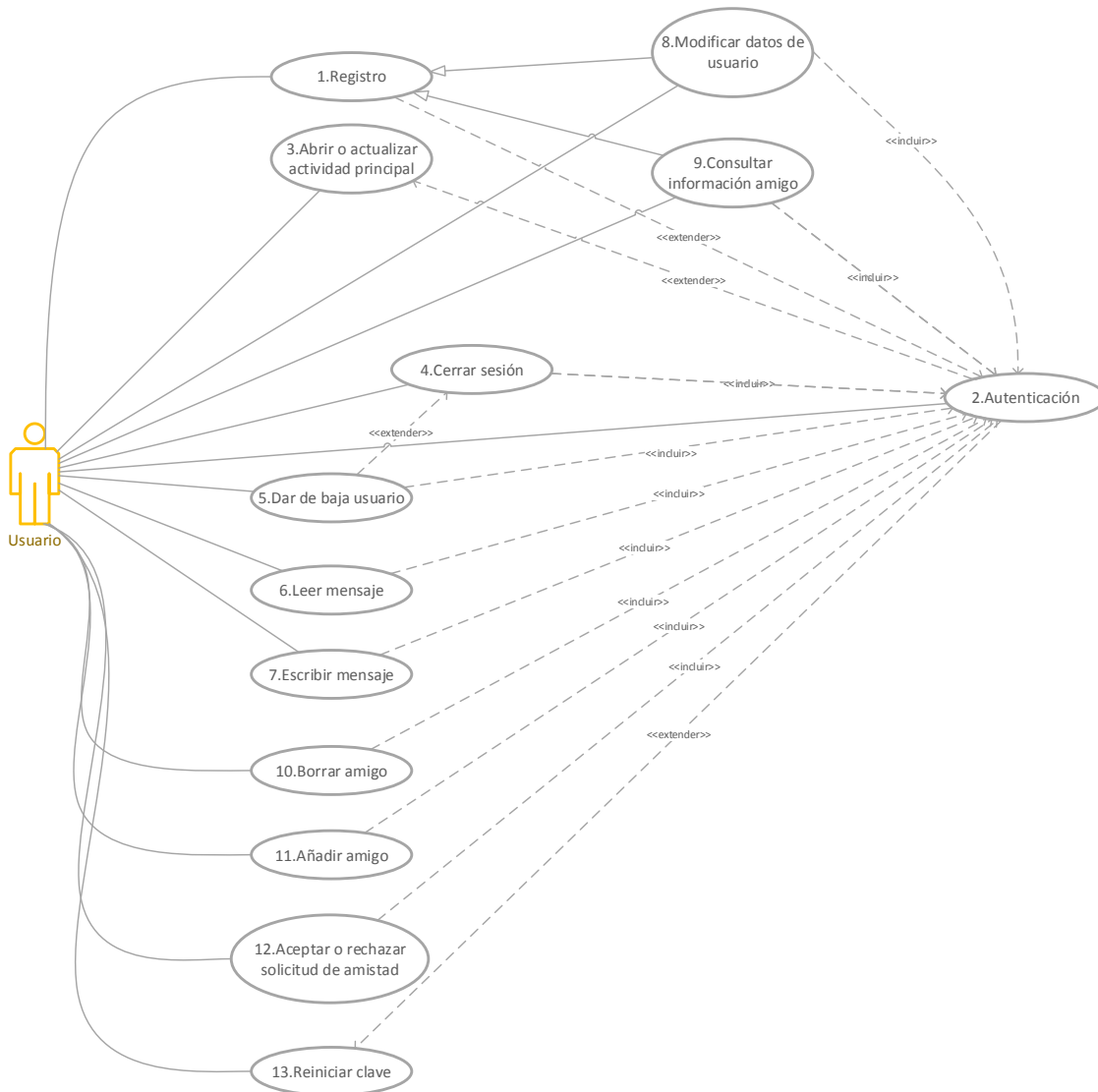
Estas clases y ServerListener serán singleton

- Prioridad: Alta



### 3. Casos de uso

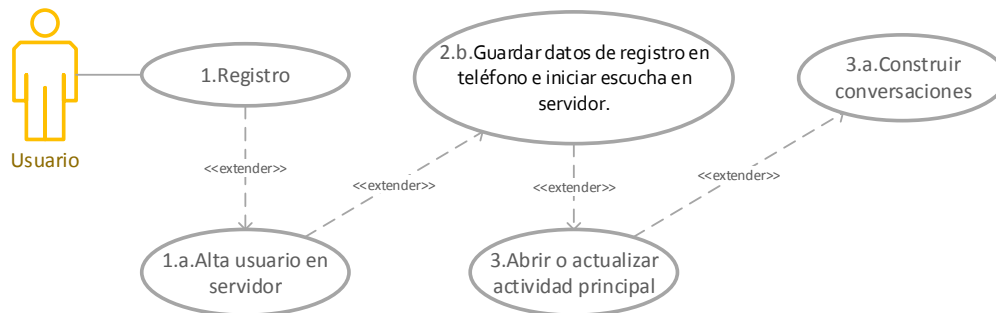
Casos de uso APP - Chat  
Juan Ignacio Avendaño Huergo







## 1. Registro



- **Actor:** Usuario
- **Propósito:** Dar de alta el usuario en el sistema.
- **PreCondiciones:**
  - No tener usuario en el sistema.
  - Tener instalada la aplicación en el móvil.
- **PostCondiciones:** Usuario habilitado en el sistema y listo para interactuar con la herramienta.
- **Resumen:** El usuario ha de darse de alta en el sistema con el fin de poder autenticarse, lo que le dará libertad al usuario de poder usar la herramienta, poder ser reconocido como emisor o receptor de mensajes, y poder hacer un seguimiento de la actividad del usuario a nivel administración.
- **Requerimientos asociados:**

### Flujo principal

1. El usuario hace clic sobre icono en menú de aplicaciones. La aplicación se abre.
2. La aplicación abre formulario inicio de sesión. Usuario no tiene datos de login, luego el usuario hace clic en registro.
3. Aplicación abre formulario de registro. Usuario rellena datos de registro.
4. Usuario hace clic en botón de alta. Se dan de alta los datos del usuario en el sistema. Se guardan datos de login en teléfono. (**Se extiende caso de uso 1.a, alta usuario en servidor**).
5. Aplicación abre actividad de inicio. El usuario puede interactuar con la aplicación.

### Flujos alternativos

Línea 4: Si el usuario ya existe, el sistema dará error diciendo que el usuario ya existe, notificándolo por pantalla. El usuario habrá de insertar otro correo o número de teléfono, o volver a la pantalla anterior para iniciar sesión, donde pasaría al caso de uso 2, autenticación.

Línea 4: Si por algún motivo este paso no se puede realizar, por ejemplo, porque se pierde la conectividad, el sistema nos notificará el error por pantalla.



#### a. Alta usuario en servidor

- **Actor:** Usuario
- **Proposito:** Dar de alta el usuario en el sistema.
- **PreCondiciones:**
  - No tener usuario en el servidor.
  - No haber usado la aplicación previamente
- **PostCondiciones:** Usuario dado de alta en servidor.
- **Resumen:** El usuario ha de darse de alta en el sistema con el fin de poder autenticarse, lo que le dará libertad al usuario de poder usar la herramienta, poder ser reconocido como emisor o receptor de mensajes, y poder hacer un seguimiento de la actividad del usuario a nivel administración.
- **Requerimientos asociados:**

#### Flujo principal

1. Usuario hace clic en botón de alta. Actividad de registro (RegisterActivity) solicita alta a servidor.
2. Servidor procesa alta de usuario. **Extiende caso de uso 2.b, guardar datos de registro en teléfono e iniciar escucha en servidor.**

#### Flujos alternativos

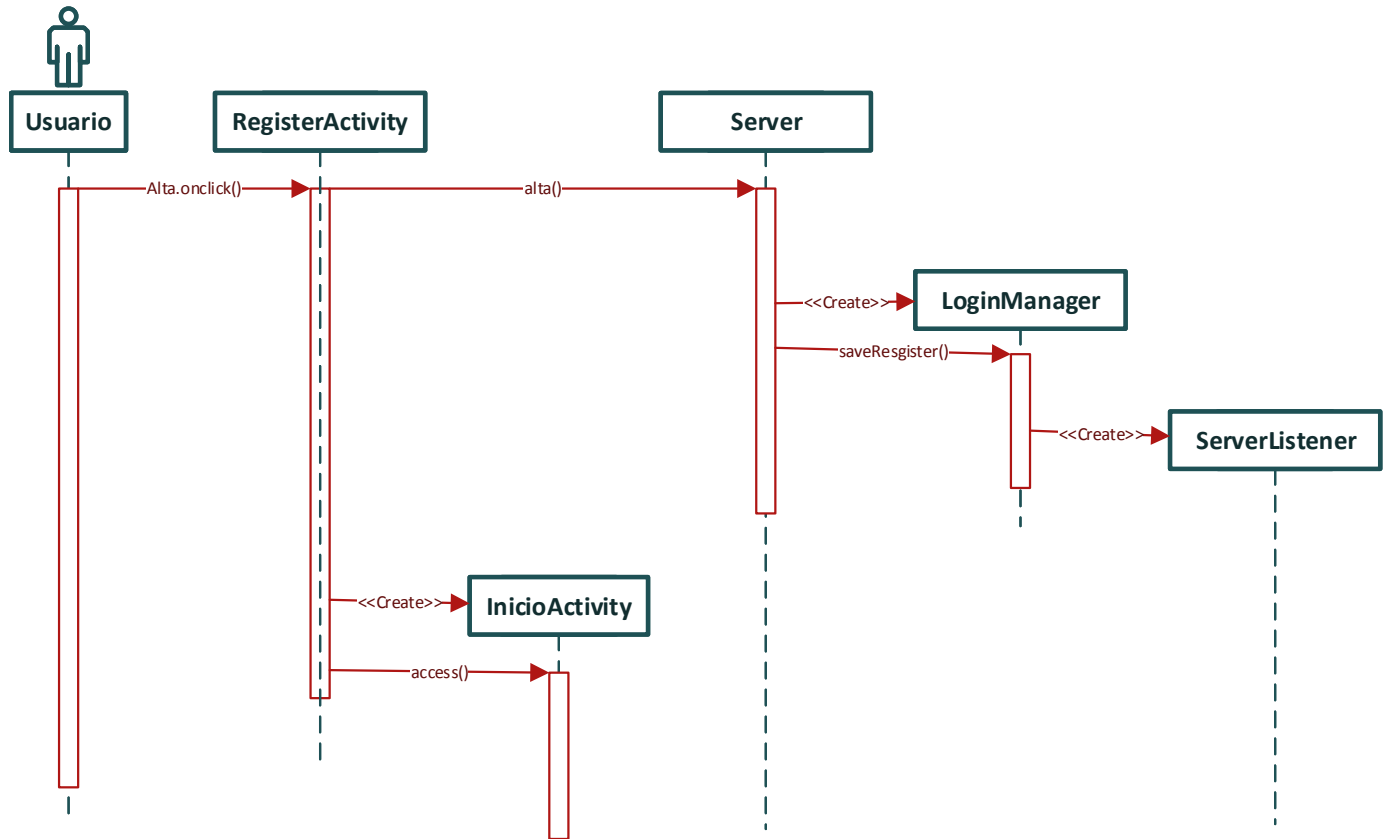
Línea 2: Si por algún motivo alguno de estos pasos no se puede realizar, por ejemplo, porque se pierde la conectividad, el sistema nos notificará el error por pantalla.

#### Reglas de negocio

1. Si el usuario ya existe, la validación ha de dar error.

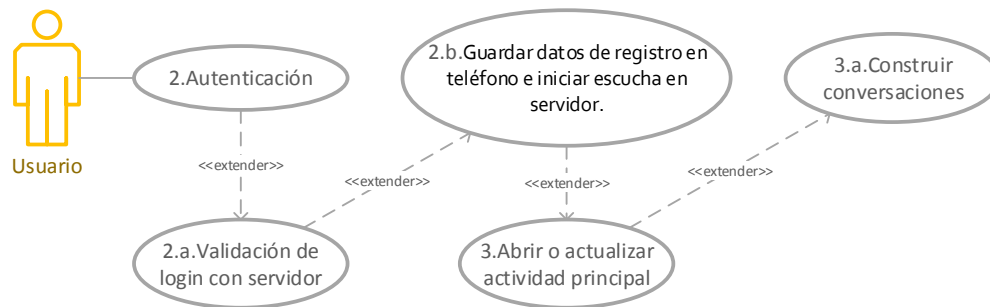


Diagrama de interacción





## 2. Autenticación



- **Actor:** Usuario
- **Propósito:** Iniciar sesión en el sistema en caso de ya disponer una cuenta en el sistema.
- **PreCondiciones:**
  - Tener usuario en el sistema.
  - Tener instalada la aplicación en el móvil.
  - No haber utilizado antes la aplicación.
- **PostCondiciones:** El usuario podrá interactuar con la herramienta con total libertad y acceder a sus funcionalidades.
- **Resumen:** El usuario ha de poder autenticarse para poder usar la herramienta, para así poder ser reconocido como emisor o receptor de mensajes, y poder hacer un seguimiento de la actividad del usuario a nivel administración.
- **Requerimientos asociados:**

### Flujo principal

1. El usuario hace clic sobre icono en menú de aplicaciones. La aplicación se abre.
2. La aplicación abre formulario inicio de sesión. Usuario rellena los datos del formulario.
3. El usuario hace clic en entrar. Se envían los datos al servidor para ser comprobados.
4. Validación con servidor es correcta. Sus datos de inicio se almacenan en el dispositivo. Se inicia escucha de servidor en segundo plano (**extiende caso de uso 2.a, Validación de login con servidor**).

### Flujos alternativos

Línea 2: Si usuario no recuerda datos de acceso, **se extiende el caso de uso 13, reiniciar Clave**.

Línea 3: Si por algún motivo algunos de estos pasos no se pueden realizar, por ejemplo, porque se pierde la conectividad, el sistema notificará por pantalla el error.

Línea 4: Si los datos no son validados correctamente, se ha de mostrar cual es el dato que falla.

1. Si falla el usuario, se ha de notificar al usuario y el usuario podrá iniciar el caso de uso de registro.
2. Si falla la clave, se le ha de notificar al usuario y, con el correo insertado, podrá iniciar el caso de uso 12, reiniciar clave.



#### a. Validación de login con servidor

- **Actor:** Usuario
- **Propósito:** Iniciar sesión en el dispositivo para iniciar servicios de escucha de servidor y acceder a nuestros datos.
- **PreCondiciones:**
  - Tener usuario en el sistema.
  - No tener almacenado en dispositivo datos de login.
- **PostCondiciones:** El usuario podrá interactuar con la herramienta con total libertad y acceder a sus funcionalidades.
- **Resumen:** El usuario ha de poder autenticarse para poder usar la herramienta, para así poder ser reconocido como emisor o receptor de mensajes, y poder acceder a su información.
- **Requerimientos asociados:**

#### Flujo principal

1. Usuario hace clic en botón de Entrar. Actividad de login (LoginActivity) solicita comprobación de datos con servidor.
2. Servidor valida login. **Extiende caso de uso 2.b, guardar datos de registro en teléfono e iniciar escucha en servidor.**

#### Flujos alternativos

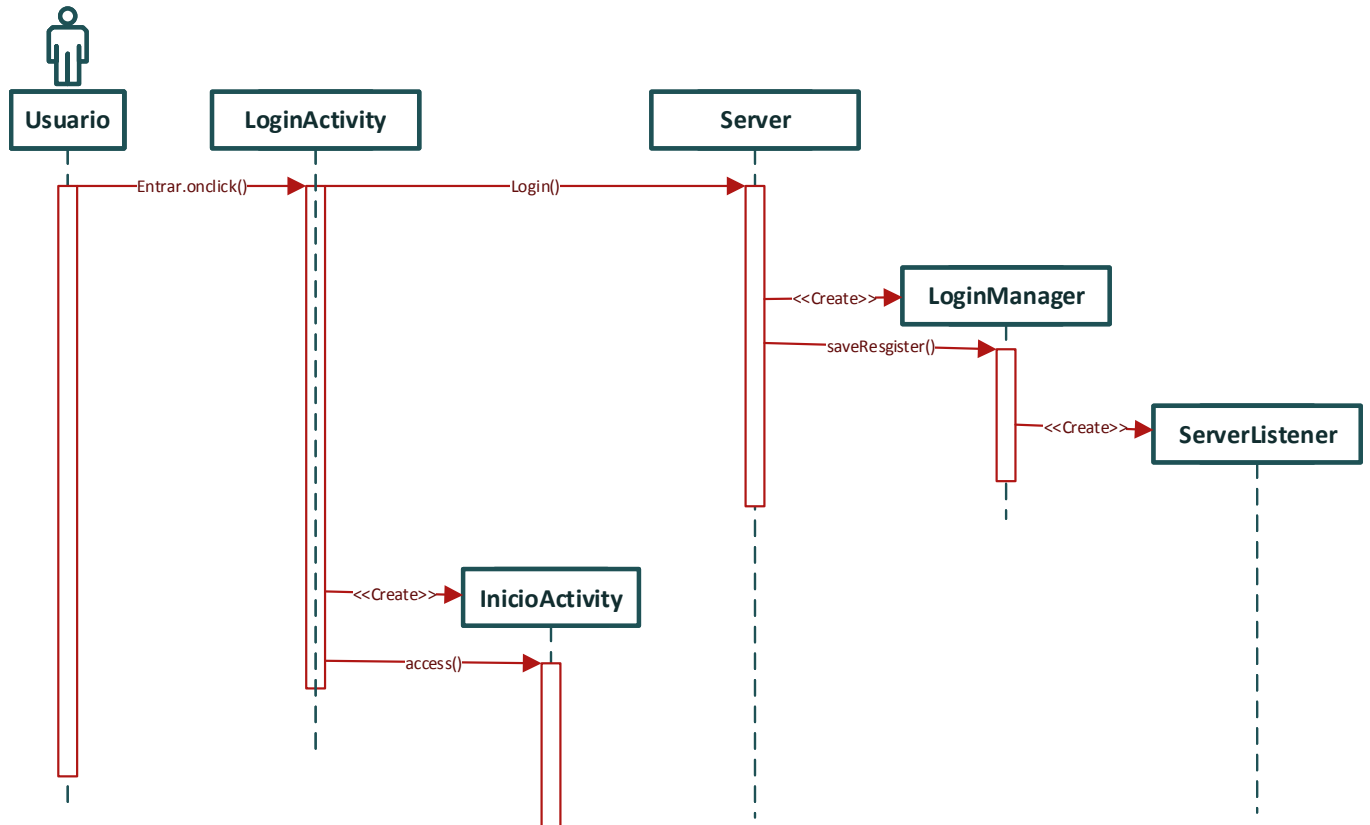
Línea 2: Si por algún motivo algunos de estos pasos no se pueden realizar, por ejemplo, porque se pierde la conectividad, el sistema notificará por pantalla el error.

#### Reglas de negocio

1. Si el usuario no existe, se ha de notificar el error por pantalla.
2. Si falla la clave, se le ha de notificar el error por pantalla.



Diagrama de interacción





#### b. Guardar datos de registro en teléfono e iniciar escucha en servidor

- **Actor:** Usuario
- **Propósito:** Componente del sistema que guarda datos de registro e inicia escucha asíncrona al servidor.
- **PreCondiciones:**
  - Haberse autenticado o registrado en el sistema.
  - Estar procesando el alta de un usuario o el inicio de sesión en dispositivo.
- **PostCondiciones:** Los datos de registro estarán almacenados en el teléfono y el servicio de escucha al servidor estará listo para funcionar.
- **Resumen:** Componente del sistema que guarda datos de registro e inicia escucha asíncrona al servidor.
- **Requerimientos asociados:**

#### Flujo principal

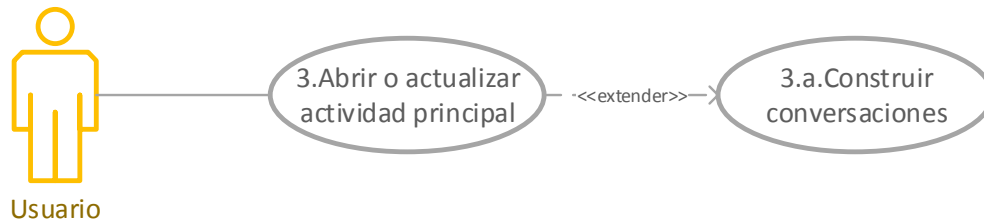
1. Gestor de login guarda datos de registro en teléfono. Gestor de login inicia ServerListener en segundo plano.
2. ServerListener inicia escucha en segundo plano al servidor para recibir mensajes entrantes y solicitudes de amistad. **Extiende caso de uso 3, abrir actividad de inicio.**

#### Flujos alternativos

No se han detectado.



### 3. Abrir o actualizar actividad principal



- **Actor:** Usuario
- **Propósito:** Componente del sistema que abre actividad principal.
- **PreCondiciones:**
  - Estar autenticado en el sistema.
- **PostCondiciones:** Actividad mostrará conversaciones, indicando cuantos mensajes no leídos hay en cada una, de haber.
- **Resumen:** Componente del sistema que abre actividad principal, donde se nos muestran las conversaciones con nuestros amigos.
- **Requerimientos asociados:**

#### Flujo principal

1. Actividad de inicio pide a Server conversaciones. **Extiende actividad 3.a, Construir conversaciones.** Server devuelve a actividad conversaciones con usuarios.
2. ServerListener almacena conversaciones nuevas en móvil. Actividad se muestra.

#### Flujos alternativos

No se han detectado.





#### a. Construir conversaciones

- **Actor:** Usuario
- **Propósito:** Componente del sistema que construye conversaciones, que luego se mostrarán en actividad principal.
- **PreCondiciones:**
  - Estar autenticado en el sistema.
- **PostCondiciones:** Server devuelve lista de conversaciones con usuarios.
- **Resumen:** Componente del sistema que construye conversaciones, que luego se mostrarán en actividad principal.
- **Requerimientos asociados:**

#### Flujo principal

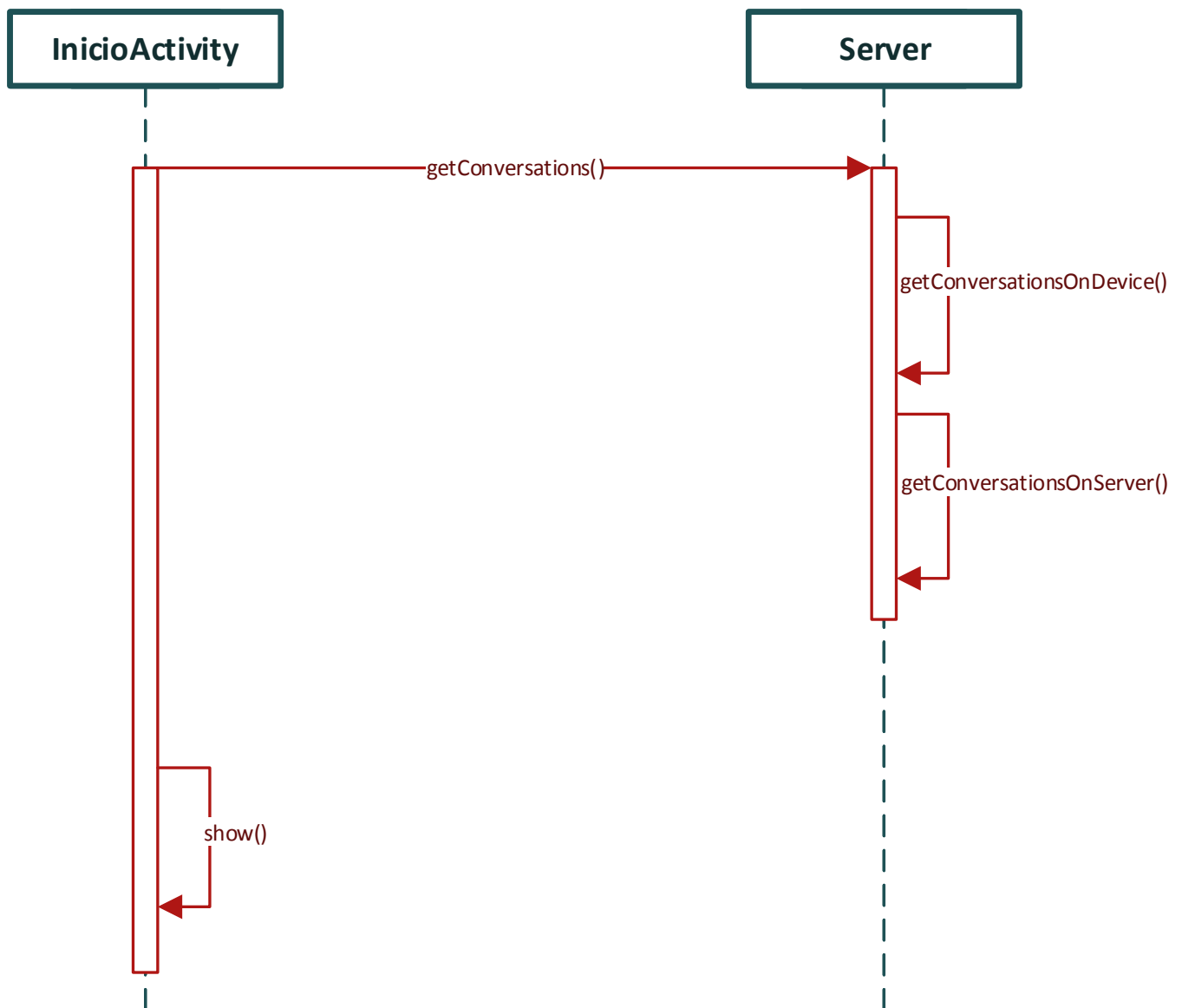
1. Server busca en dispositivo si hay conversaciones guardadas. Si las hay las recupera y las agrega a una lista.
2. Server busca en servidor nuevas conversaciones que no estén en dispositivo. Si las hay, las recupera y las agrega a una lista.

#### Flujos alternativos

No se han detectado.

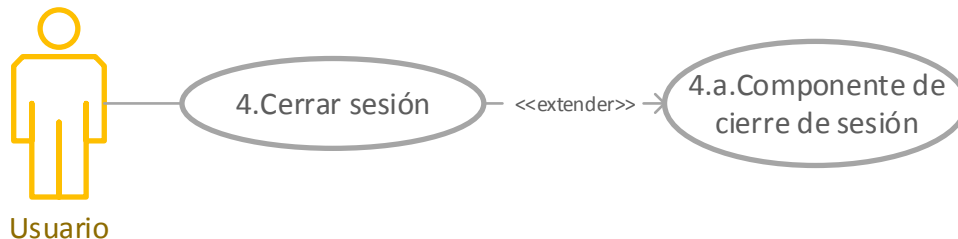


Diagrama de interacción





#### 4. Cerrar sesión



- **Actor:** Usuario
- **Propósito:** Permitirle al usuario de forma sencilla controlar el usuario y sus datos.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en el menú principal.
- **PostCondiciones:** El usuario ya no estará autenticado. Para volver a usar la aplicación con normalidad, deberá hacer el caso de uso 2.
- **Resumen:** El usuario ha de poder cerrar la sesión activa. Esta opción lo que hará será, además de cerrar la sesión de escucha del servidor, borrará los datos de autenticación del dispositivo automáticamente.
- **Requerimientos asociados:** RF3

##### Flujo principal

1. El usuario hace clic sobre su usuario. Dispositivo abre actividad de perfil.
2. El usuario hace clic en cerrar sesión. Se le pide a usuario confirmación.
3. Usuario confirma cierre. Los datos de credenciales de usuario son borrados del dispositivo. Se cierra el servicio en segundo plano que escucha los mensajes (**extiende caso de uso 4.a, Componente de cierre de sesión**).
4. Servicio en según plano es cerrado. Se reinicia aplicación.

##### Flujos alternativos

No se han detectado.



#### a. Componente de Cierre de sesión

- **Actor:** Usuario
- **Propósito:** Permitirle al usuario controlar el usuario y sus datos.
- **PreCondiciones:**
  - Haberse autenticado en el dispositivo.
  - Abrir la actividad de perfil de usuario.
- **PostCondiciones:** El usuario ya no estará autenticado y sus datos de registro serán borrados del dispositivo. Para volver a usar la aplicación con normalidad, deberá hacer el caso de uso 2.
- **Resumen:** El usuario ha de poder cerrar la sesión activa. Esta opción lo que hará será, además de cerrar la sesión de escucha del servidor, borrará los datos de autenticación del dispositivo automáticamente.
- **Requerimientos asociados:**

#### Flujo principal

1. El usuario hace clic en cerrar sesión. Actividad pide confirmación de cierre de sesión.
2. Usuario confirma cierre de sesión. Actividad solicita a Gestor de login cierre de sesión.
3. Gestor de login recibe cierre de sesión. Gestor de login borra datos de registro en teléfono. Gestor de login cierra la instancia de ServerListener. Gestor de login reinicia aplicación.

#### Flujos alternativos

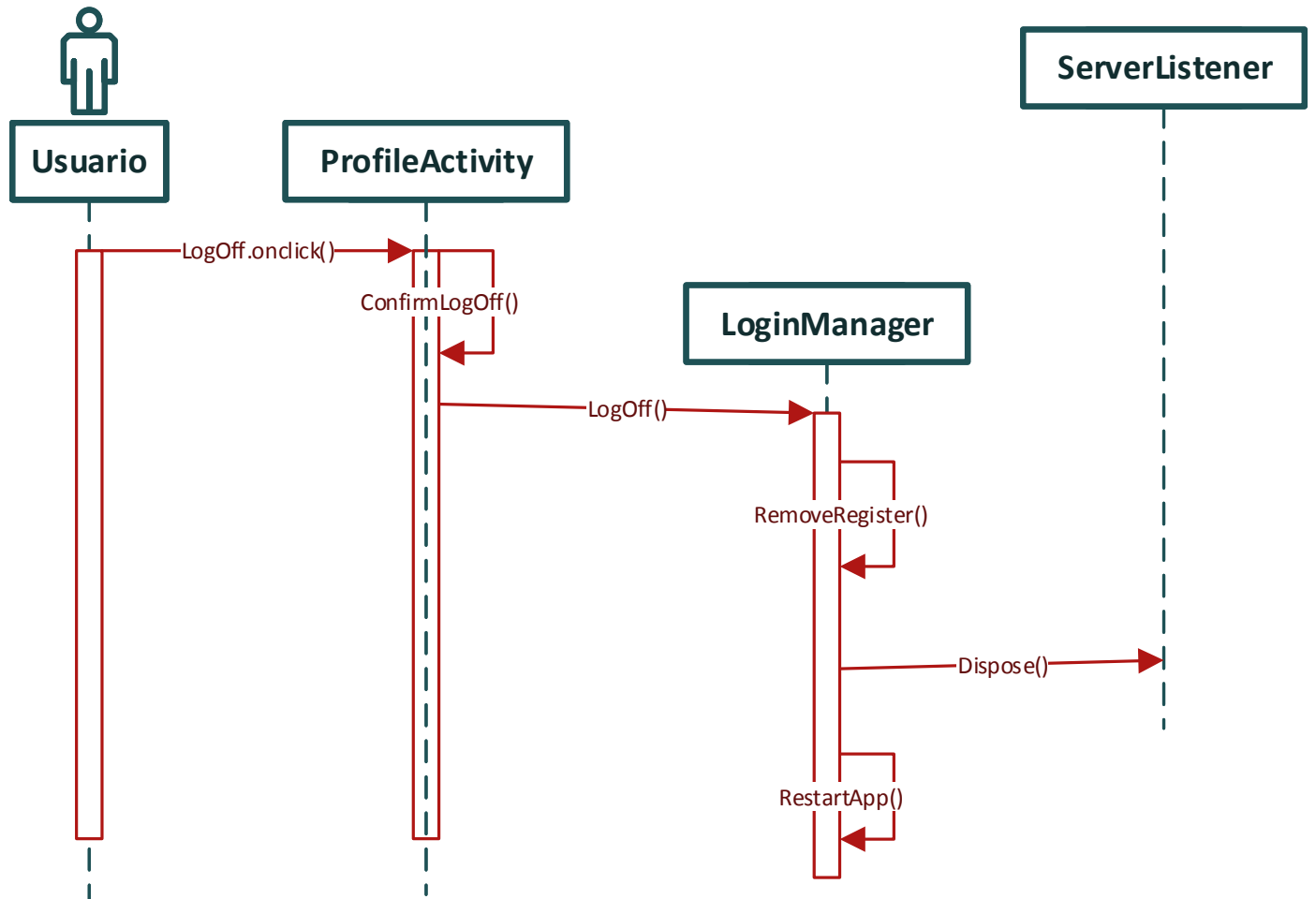
No se han detectado.

#### Reglas de negocio

No se han detectado.

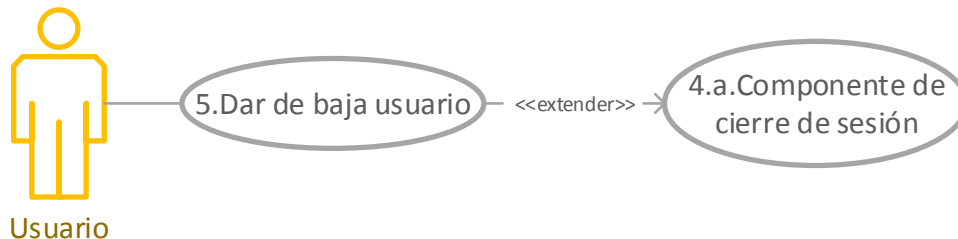


Diagrama de interacción





## 5. Dar de baja usuario



- **Actor:** Usuario
- **Propósito:** Permitirle al usuario de forma sencilla controlar el usuario y sus datos.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en el menú principal.
- **PostCondiciones:** El usuario ya no existirá en el sistema y toda su información asociada habrá sido borrada. Para volver a usar la aplicación con normalidad, deberá hacer el caso de uso 1 para volver a darse de alta.
- **Resumen:** El usuario ha de poder darse de baja del sistema, con el fin de darle al usuario control total sobre su información. Esta opción lo que hará será, además de cerrar la sesión de escucha del servidor, borrará los datos de autenticación del dispositivo y de nuestros sistemas, así como toda su información asociada.
- **Requerimientos asociados:** RNF1

### Flujo principal

1. El usuario hace clic sobre su usuario. Dispositivo abre actividad de perfil de usuario.
2. El usuario hace clic en baja. Se le pide confirmación al usuario.
3. Usuario confirma baja. Actividad solicita a servidor baja.
4. Servidor inicia solicitud asíncrona de baja. **Extiende caso de uso 4.a, componente de cierre de sesión.**
5. En servidor borra toda la información asociada al usuario, incluyendo mensajes. Se le envía correo a usuario de que la operación ha tenido éxito.

### Flujos alternativos

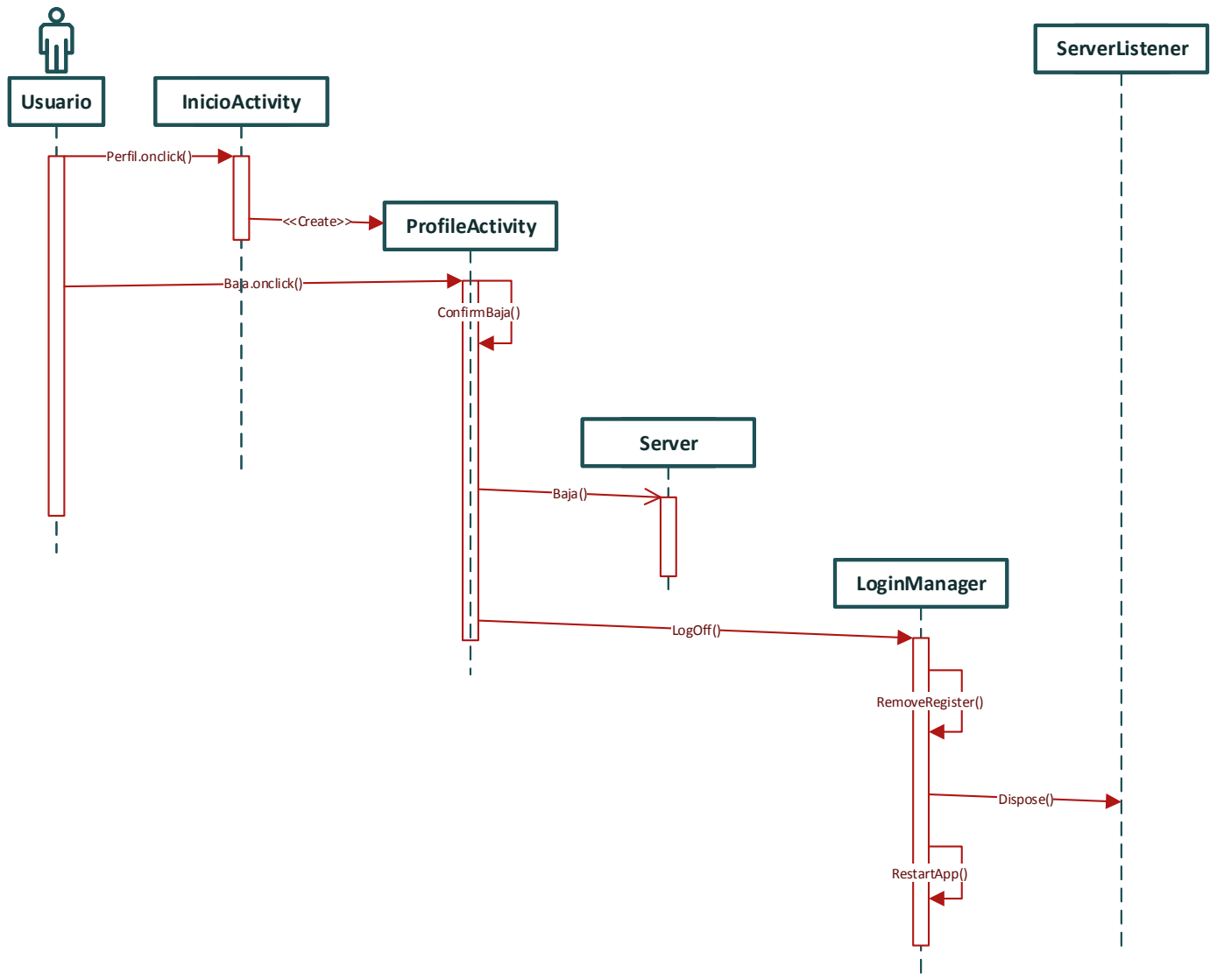
Línea 4: En caso de que no haya conexión a internet, la operación no se llevará a cabo. Se le notificará al usuario del error.

Línea 5: En caso que haya habido un problema con el borrado en servidor, inmediatamente se iniciará una incidencia técnica, notificándole a los administradores de la aplicación y al usuario de lo ocurrido para que:

- El usuario realice el debido seguimiento, bajo su derecho.
- Los técnicos se pongan manos a la obra para eliminar la información restante.

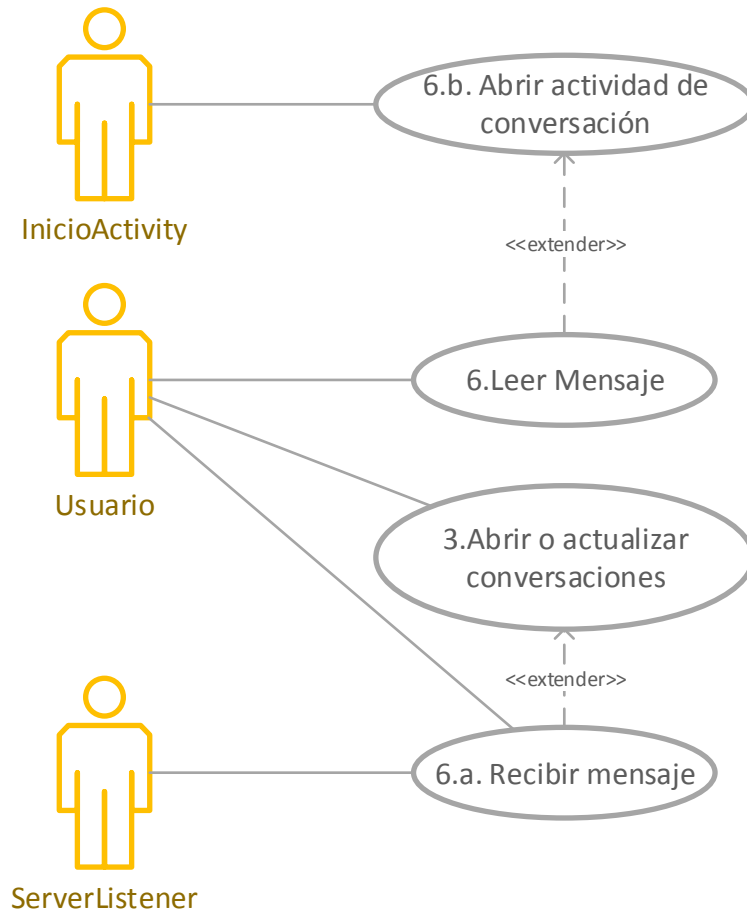


Diagrama de interacción





## 6. Leer mensaje



- **Actor:** Usuario
- **Propósito:** Poder ver los mensajes que se le envía al usuario que utiliza la aplicación.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Tener mensajes que leer.
  - Estar en actividad de inicio o haber recibido una notificación push de nuevo mensaje.
- **PostCondiciones:** Los mensajes leídos que antes eran nuevos ya no será nuevo, sino que figurarán como leídos.
- **Resumen:** El usuario ha de poder ver todos los mensajes recibidos. Ya sean nuevos o no. Como si fuera un correo electrónico, o WhatsApp.
- **Requerimientos asociados:** RNF2, RNF3, RNF4

### Flujo principal

1. Usuario hace clic en una conversación. Aplicación abre actividad de conversación. **Se extiende caso de uso 6.b abrir actividad de conversación.**
2. Actividad de conversación muestra conversación con amigo. Usuario puede interactuar con actividad.

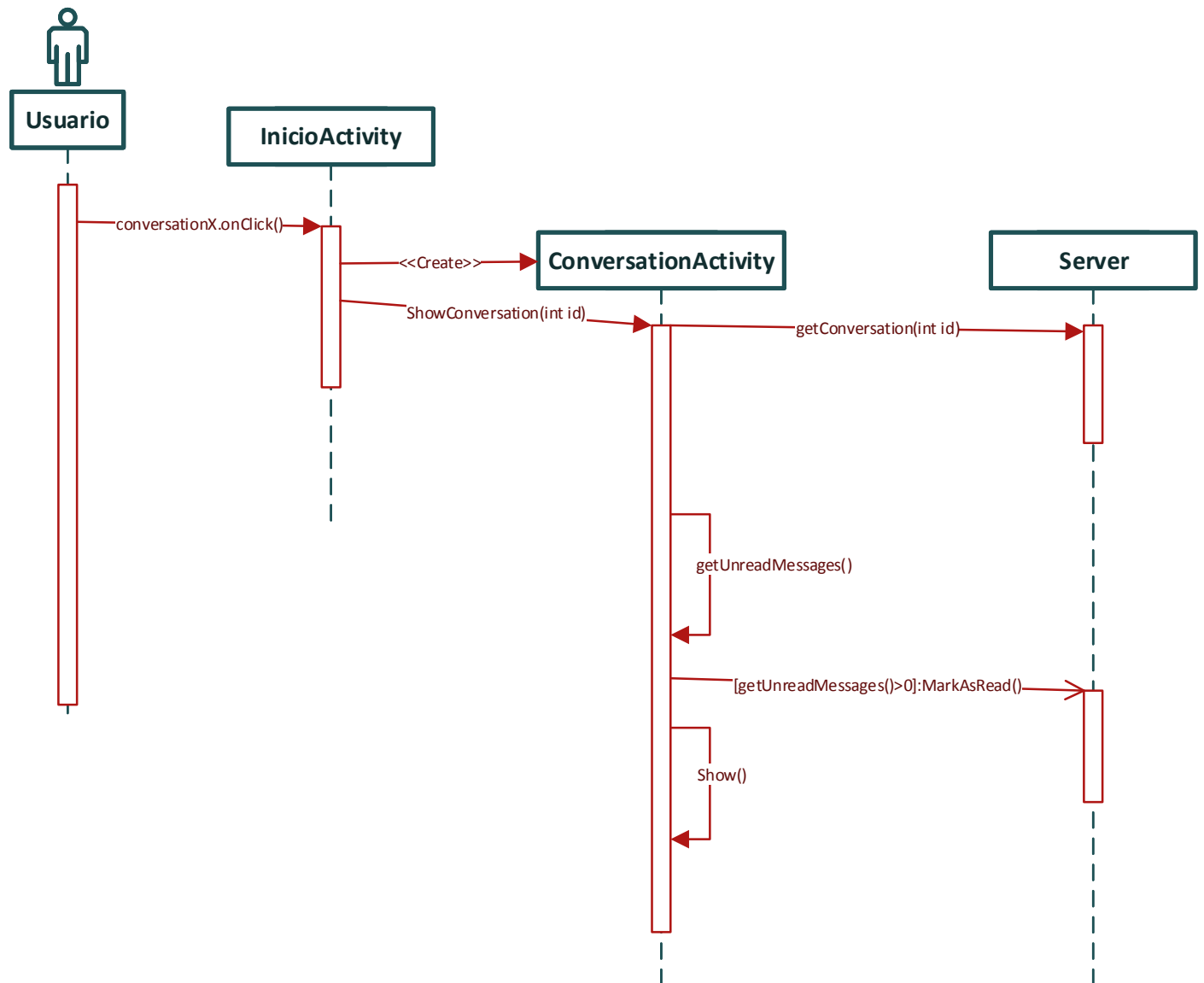




### Flujos alternativos

Línea 1: En caso de abrirlo mediante el mensaje push, se abrirá directamente la actividad de conversación y esta recuperará la conversación con amigo.

### Diagrama de interacción





#### a. Recibir mensaje

- **Actor:** Usuario
- **Propósito:** Poder recibir los mensajes que se le envía al usuario que utiliza la aplicación.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
- **PostCondiciones:** Dispositivo informará al usuario de los nuevos mensajes.
- **Resumen:** Para el usuario poder ver los mensajes, el usuario ha de poder recibir nuevos mensajes.
- **Requerimientos asociados:**

#### Flujo principal

1. ServerListener detecta nuevo mensaje. ServerListener solicita a dispositivo notificación push.
2. Si actividad de inicio está abierta, **se extiende caso de uso 3, Abrir o actualizar conversaciones**. Actividad de inicio muestra conversaciones con actualizaciones.

#### Flujos alternativos

Línea 1: Si aplicación está abierta, ServerListener ha de solicitar a dispositivo realizar un pequeño sonido o pitido y no enviar notificación.

Línea 2: Si actividad de conversación está abierta y el mensaje nuevo es en esa conversación, Actividad de conversación actualiza conversación mostrando los nuevos mensajes.



#### b. Abrir actividad de conversación

- **Actor:** Actividad de conversación.
- **Propósito:** Gestionar la apertura de la actividad correctamente.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Iniciar instancia de nueva conversación.
- **PostCondiciones:** Conversación estará lista para mostrar.
- **Resumen:** Hemos de gestionar correctamente la lógica de mostrar conversación, la cual no es arbitraria. Este caso de ejecuta durante el create de la actividad de conversación.
- **Requerimientos asociados:**

##### Flujo principal

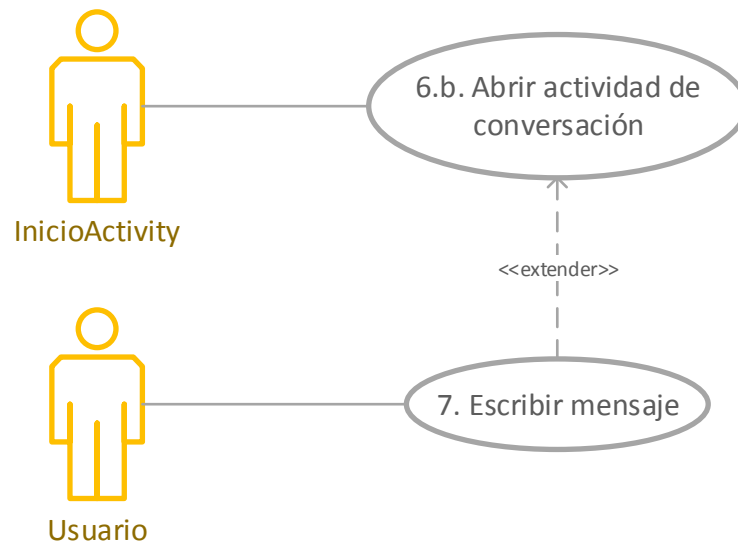
1. Actividad solicita a servidor conversación con amigo. Servidor devuelve conversación con amigo.
2. Si hay mensajes nuevos, actividad envía solicitud a servidor para marcar mensajes en lista como leídos. Servidor marca mensajes no leídos como leídos.

##### Flujos alternativos

Líneas 1 y 2: Si no es posible contactar con servidor, se le ha de notificar al usuario del error.



## 7. Escribir mensaje



- **Actor:** Usuario
- **Propósito:** Poder ver los mensajes que se le envía al usuario que utiliza la aplicación.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
- **PostCondiciones:** La conversación tendrá un nuevo mensaje emitido por él hacia un amigo suyo, el asociado a esa conversación.
- **Resumen:** El usuario ha de poder enviar un mensaje a un amigo.
- **Requerimientos asociados:**

### Flujo principal

1. El usuario hará clic en el signo más. Aplicación muestra desplegable con opciones.
2. En el menú desplegable, usuario hace clic en mensaje. Aplicación abre actividad de nueva conversación.
3. Actividad de nueva conversación solicita a servidor lista de amigos. Servidor devuelve listado. Actividad lista amigos en orden alfabético.
4. Usuario elige de la lista a amigo con el cual iniciar la conversación. Actividad solicita a servidor generar nueva conversación.
5. Servidor genera nueva conversación, Actividad abre actividad de conversación.
6. **Extiende caso de uso 6b. abrir actividad de conversación.** Actividad de conversación muestra conversación.
7. Usuario escribe mensaje con o sin adjuntos y hace clic en enviar. Actividad solicita envío de mensaje a servidor.
8. Servidor envía mensaje de forma exitosa. Mensaje aparece como enviado en la actividad. Campo de texto de mensaje de reinicia para enviar nuevo mensaje.



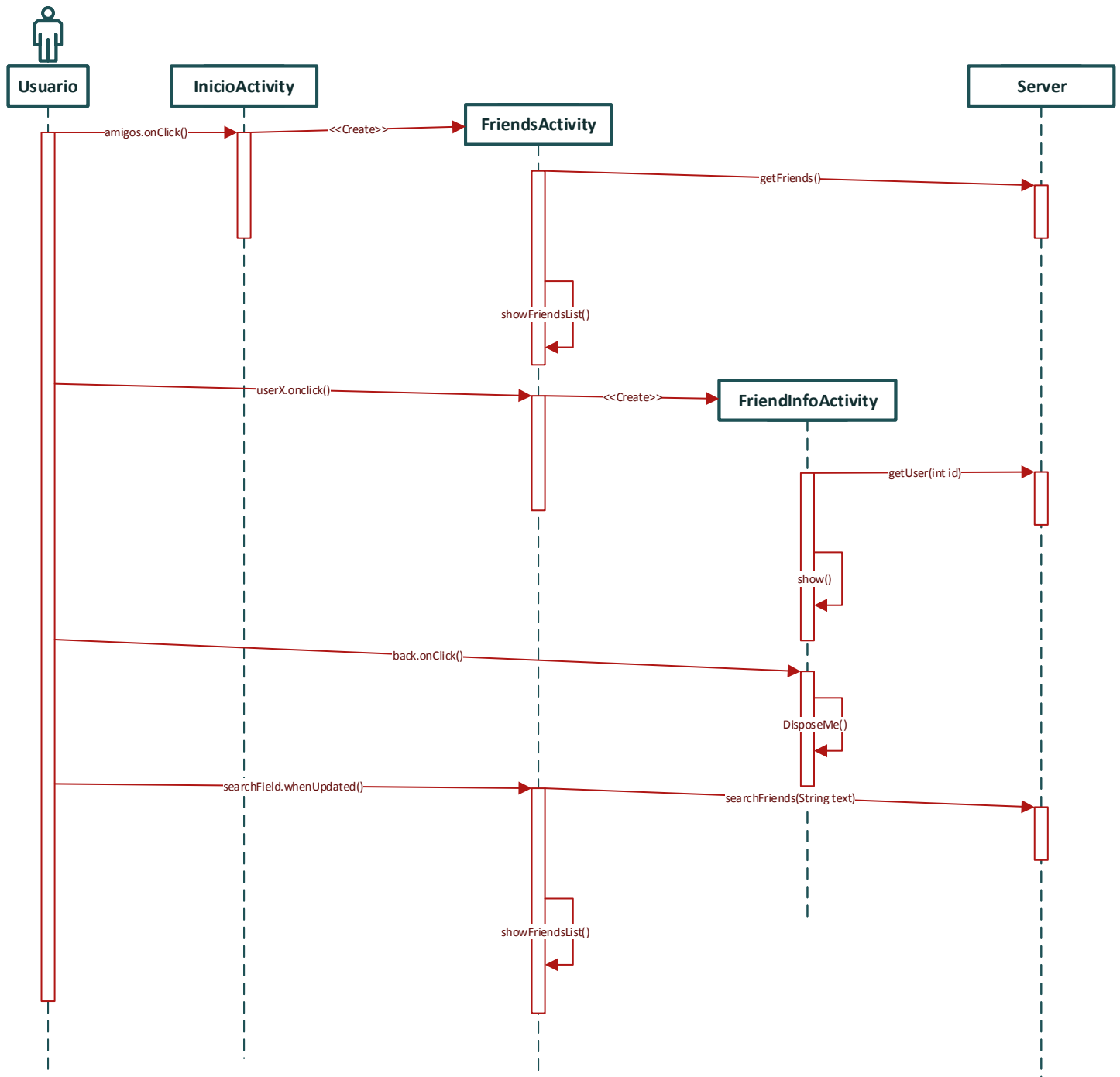
#### Flujos alternativos

Línea 1: También se puede, en el menú principal, acceder a una conversación ya existente y nos saltaríamos los pasos 2, 3, 4, y 5. Actividad abriría directamente la actividad de conversación. Lo mismo sucedería si accediéramos a una conversación desde un mensaje push.

Línea 7: Si el envío del mensaje no es exitoso, se le notifica al usuario por pantalla.



## Diagrama de interacción





## 8. Modificar datos de usuario

- **Actor:** Usuario
- **Propósito:** Poder editar los datos de su perfil, con el fin de tener total control de sus datos.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en la actividad principal.
- **PostCondiciones:** La información del usuario serán diferente.
- **Resumen:** El usuario podrá editar los datos de su perfil, con el fin de tener total control de sus datos.
- **Requerimientos asociados:**

### Flujo principal

1. El usuario hace clic en su nombre. Se abre actividad de perfil.
2. Actividad de perfil solicita datos de perfil a servidor. Servidor devuelve datos y son mostrados por pantalla.
3. Usuario modifica datos y hace clic en actualizar. Servidor valida datos.
4. Servidor modifica datos. Aplicación abre actividad principal y notifica éxito de operación.

### Flujos alternativos

Línea 3: Si la validación es errónea, se le notifica el error al usuario por pantalla y el Caso de uso no continua.

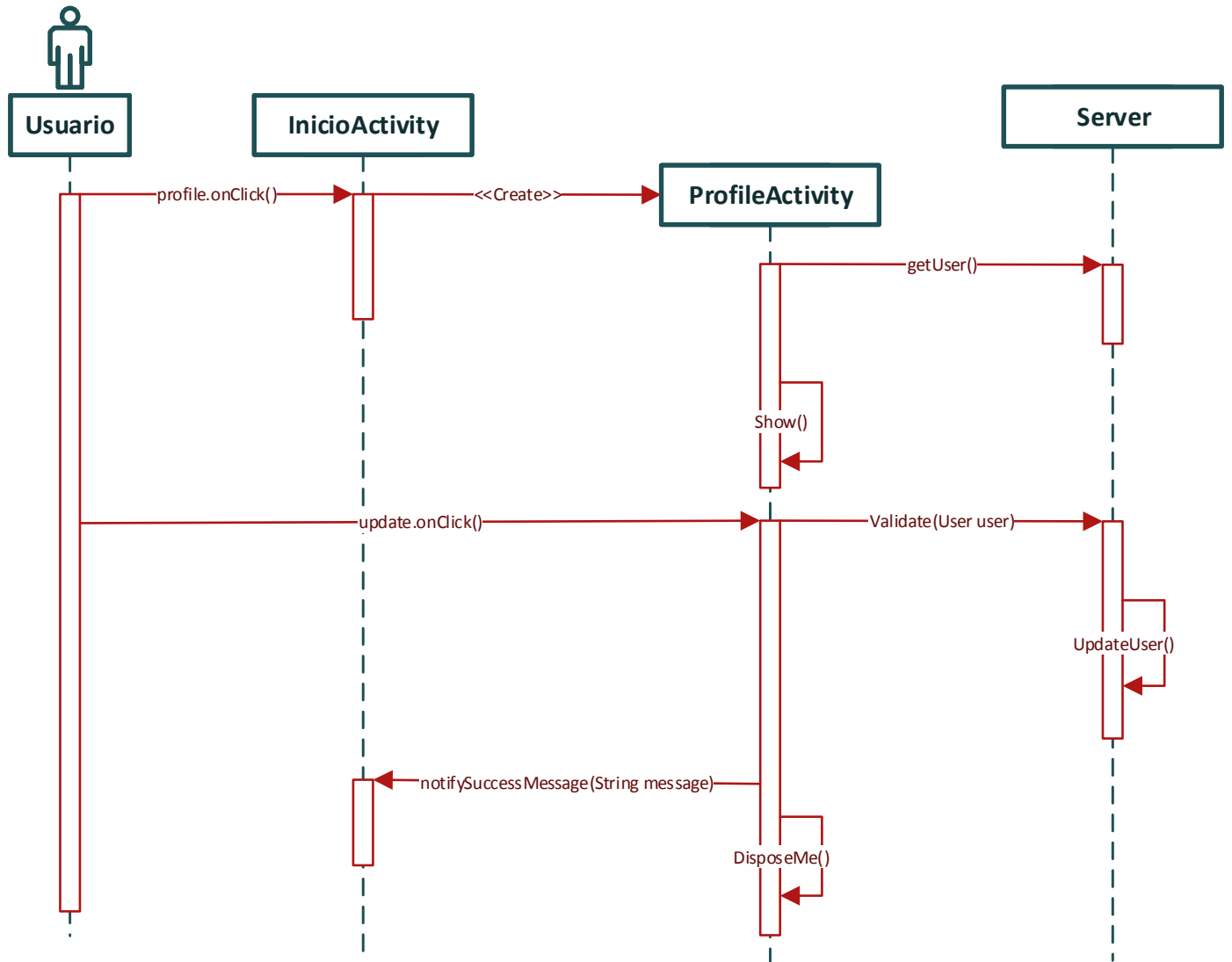
Línea 2 o 3: Si no se puede enviar o recuperar datos del servidor. Se le ha de notificar el error al usuario y el caso de uso no continua.

### Reglas de negocio

1. Si hemos recibido clave, hemos de cambiar la clave.
2. Si se insertas datos únicos de otro usuario, la validación ha de fallar.



Diagrama de interacción







## 9. Consultar Información de amigo

- **Actor:** Usuario
- **Propósito:** El usuario podrá ver la información de perfil de sus amigos, no editarla.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en actividad de inicio.
- **PostCondiciones:** El usuario estará informado de los datos de su amigo.
- **Resumen:** El usuario podrá consultar la información de su amigo, solo con fines informativos.
- **Requerimientos asociados:**

### Flujo principal

1. El usuario hace clic en gestión de amigos (el icono de persona). Actividad de inicio abre actividad de gestión de amigos. Actividad realiza búsqueda de usuarios en servidor. Actividad lista amigos en orden alfabético.
2. Usuario busca y selecciona un amigo. Actividad abre actividad de información de amigo. Actividad de información de amigo solicita información a servidor.
3. Actividad de información de amigo recupera información de amigo. Actividad muestra información.
4. Al acabar de ver los datos, usuario hace clic en volver. Actividad se cierra. Actividad realiza búsqueda de usuarios en servidor. Actividad lista amigos en orden alfabético.

### Flujos alternativos

#### *Desde la actividad de conversación*

1. Usuario hace clic en la conversación con el amigo deseado. Actividad abre actividad de conversación.
2. En la actividad de conversación, el usuario hará clic sobre el nombre de su amigo. Actividad redirige a actividad de información de amigo. Actividad de información de amigo solicita información a servidor.
3. Actividad de información de amigo recupera información de amigo. Actividad muestra información.
4. Al acabar de ver los datos, usuario hace clic en volver. Actividad se cierra.

#### *Excepciones a ambos flujos*

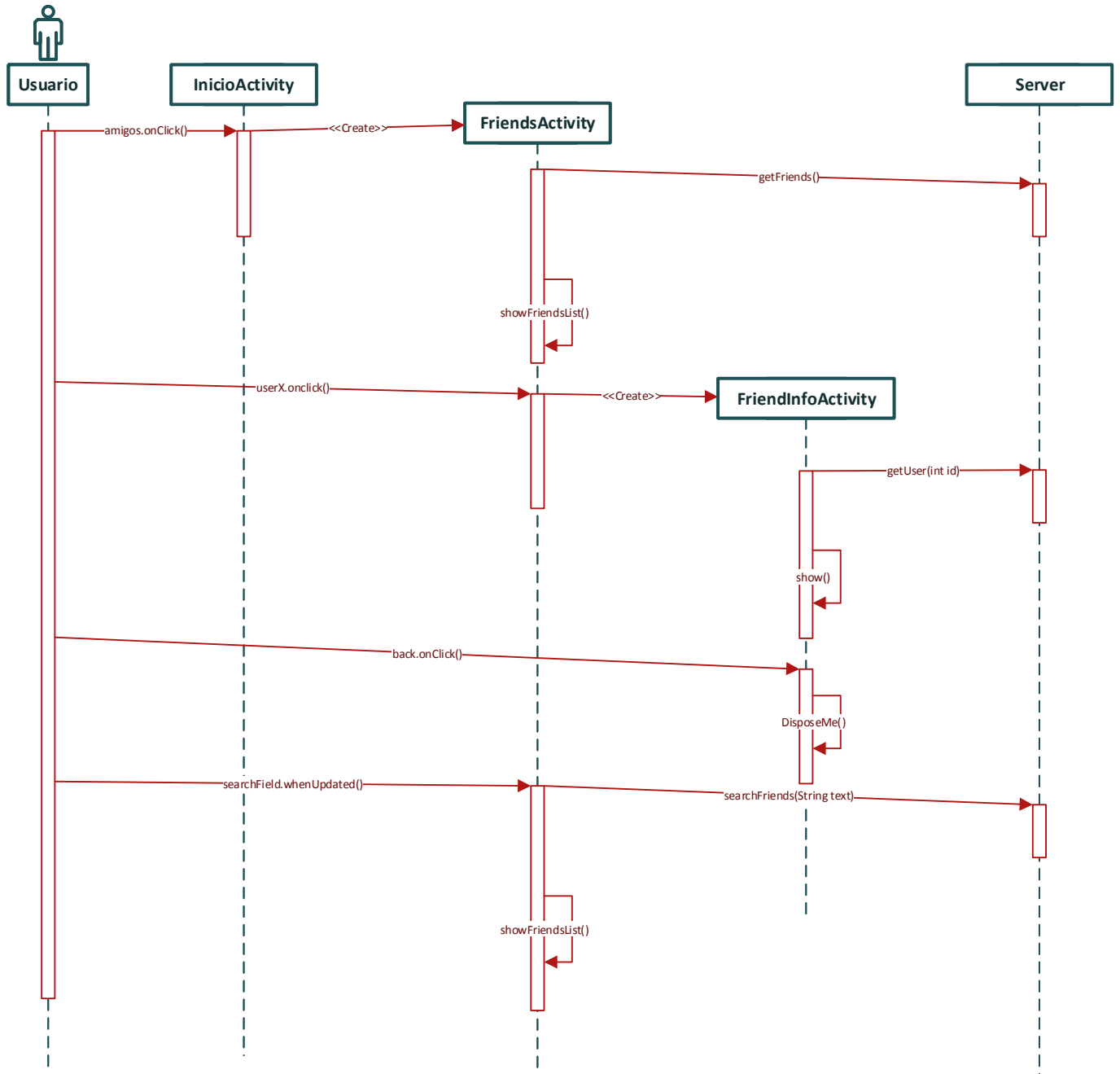
Línea 1, 2 o 3: En caso de no tener conexión a internet u otro problema para realizar la solicitud de información, se le notificará al usuario el error, y el caso de uso no continua.

### Reglas de negocio

1. Si el campo de búsqueda está vacío, ha de traer todos nuestros amigos. Si no, ha de buscar los amigos según el texto indicado.



## Diagrama de interacción





## 10. Borrar amigo.

- **Actor:** Usuario
- **Propósito:** El usuario podrá borrar a su amigo y su correspondiente conversación de su móvil.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en la actividad de gestión de amigos/mis amigos.
- **PostCondiciones:** El usuario habrá borrado a su amigo, conjunto con las conversaciones almacenadas en servidor.
- **Resumen:** El usuario podrá borrar a su amigo y su correspondiente conversación de su móvil y del sistema.
- **Requerimientos asociados:** RNF5

### Flujo principal

1. Usuario hace clic en botón de borrar amigo. Actividad notifica a usuario de las consecuencias de la acción.
2. Usuario confirma borrado. Servidor inicia borrado asíncrono de amigo. Server borra conversación y mensajes asociados del dispositivo. Actividad elimina amigo de listado.
3. Cuando el borrado en servidor se acabe, Servidor enviará correo de fin de acción. Usuario recibe información por correo.

### Flujos alternativos

Línea 2. Si usuario no confirma borrado, cancelamos caso de uso.

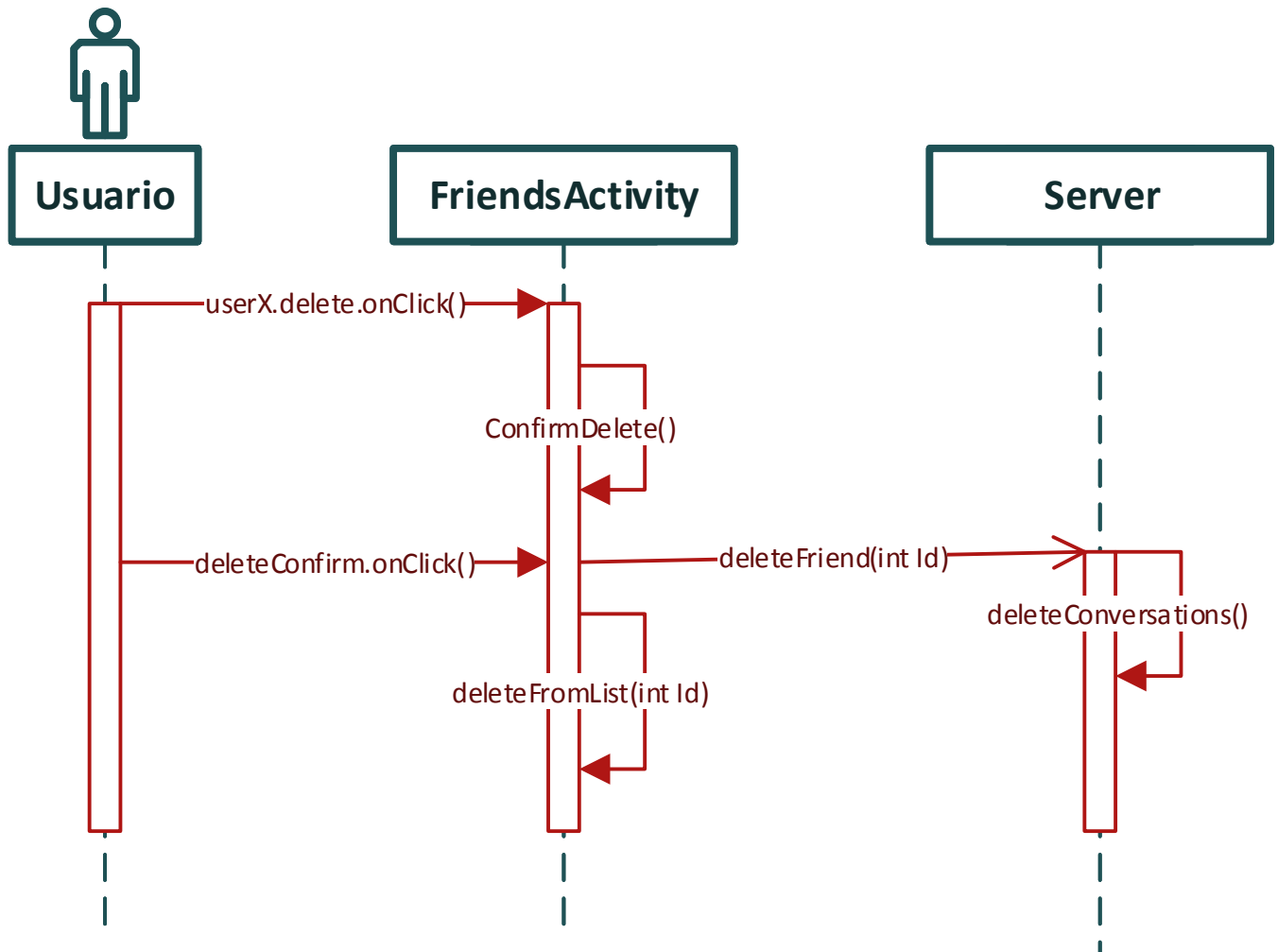
Línea 2: En caso de haber algún problema para realizar la solicitud de información, se le notificará al usuario el error, y el caso de uso no continua.

Línea 3: En caso que haya habido un problema con el borrado en servidor, inmediatamente se iniciará una incidencia técnica, notificándole a los administradores de la aplicación y al usuario de lo ocurrido para que:

- El usuario realice el debido seguimiento, bajo su derecho.
- Los técnicos se pongan manos a la obra para eliminar la información restante.



Diagrama de interacción





## 11. Añadir amigo.



- **Actor:** Usuario
- **Propósito:** El usuario podrá elegir a sus contactos en la aplicación. Para lo cual, tendrá que añadirlos.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en la actividad principal o en gestión de amigos.
- **PostCondiciones:** El usuario tendrá un nuevo amigo en el sistema.
- **Resumen:** El usuario podrá elegir a sus contactos en la aplicación. Para lo cual, tendrá que añadirlos.
- **Requerimientos asociados:**

### Flujo principal

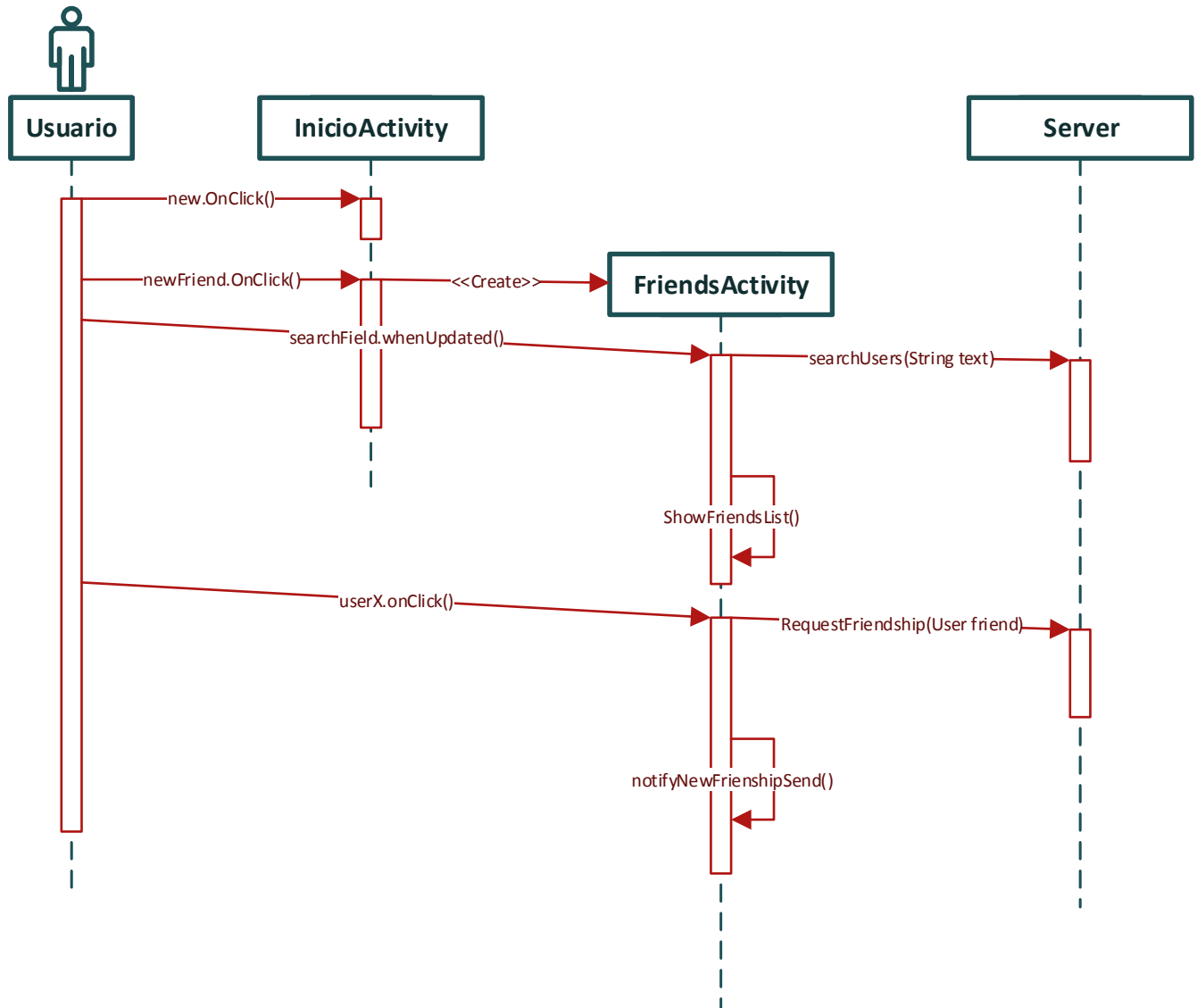
1. Usuario hacemos clic sobre el icono de +. Actividad muestra desplegable.
2. Usuario selecciona amigo. Actividad abre actividad de gestión de amigos. Actividad abre pestaña de nuevo amigo.
3. Usuario inserta texto por el cual buscar al usuario (nombres y/o apellidos, un correo, o bien su número de teléfono) y hace clic en buscar. Actividad realiza búsqueda de usuarios en servidor. Servidor devuelve lista de usuarios. Actividad muestra lista de usuarios.
4. Usuario selecciona amigo. Servidor inicia solicitud de amistad. Actividad notifica envío de solicitud

### Flujos alternativos

Línea 3 o 4: En caso de no tener conexión a internet u otro problema para realizar las búsquedas, se le notificará al usuario el error, y el caso de uso no continua.



## Diagrama de interacción





#### a. Recibir aceptación de amistad

- **Actor:** Usuario
- **Propósito:** Poder recibir las aceptaciones de amistad generadas a partir de solicitudes hechas por él.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
- **PostCondiciones:** El usuario estará informado de su nuevo amigo.
- **Resumen:** El dispositivo le informará al usuario de las solicitudes de amistad enviadas aceptadas.
- **Requerimientos asociados:**

#### Flujo principal

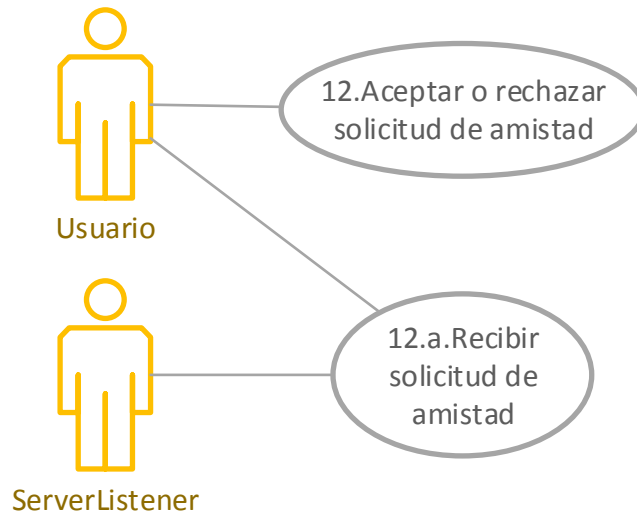
1. ServerListener detecta solicitud aceptada. ServerListener solicita a dispositivo notificación push.
2. Usuario hace clic en notificación push. Aplicación abre actividad de gestión de amigos, mostrando el nuevo amigo.

#### Flujos alternativos

Línea 1: Si actividad de gestión de amigos está abierta, Actividad actualiza listado de amigos.



## 12. Aceptar o rechazar solicitud de amistad.



- **Actor:** Usuario
- **Propósito:** El usuario podrá aceptar o rechazar solicitudes de amistad entrantes, con el fin de proteger su privacidad.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
  - Estar en la actividad principal o bien recibir una notificación push de solicitud.
- **PostCondiciones:** El usuario tendrá un nuevo amigo en el sistema o una solicitud de un desconocido o persona no deseada en la herramienta rechazada.
- **Resumen:** El usuario podrá aceptar o rechazar solicitudes de amistad entrantes, con el fin de proteger su privacidad.
- **Requerimientos asociados:**

### Flujo principal

1. El usuario hace clic sobre el icono de persona. Actividad abre actividad de gestión de amigos.
2. Usuario hace clic en la pestaña mis solicitudes. Actividad abre pestaña.
3. Actividad solicita a servidor listado de solicitudes pendientes. Servidor devuelve solicitudes pendientes. Actividad muestra solicitudes pendientes.
4. Usuario hace clic en aceptar solicitud. Actividad solicita a servidor actualización de amistad. Actividad elimina solicitud de lista. Actividad notifica por pantalla aceptación exitosa.

### Flujos alternativos

Línea 1: En caso de iniciar el caso de uso desde una notificación push, el paso uno sería entrar a la aplicación desde la notificación y el siguiente paso sería la línea 3. Es decir, el usuario entraría a la aplicación ya en la actividad de gestión de solicitudes y en la pestaña correcta.

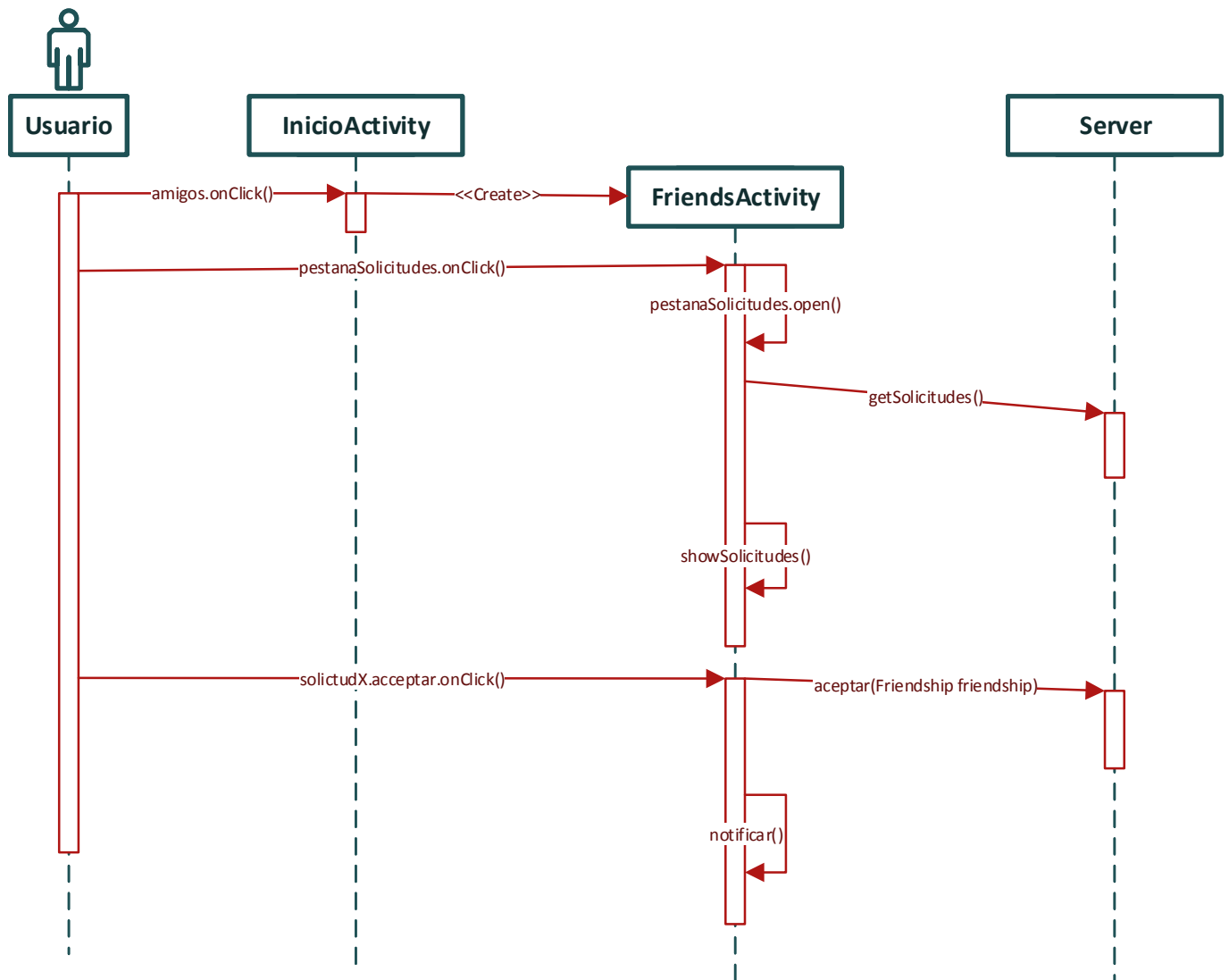
Línea 3 o 4: En caso de haber algún problema para listar las solicitudes o enviar a servidor aceptación o rechazo de amistad, se le notificará al usuario el error, y el caso de uso no continua.





Línea 4: Usuario hace clic en rechazo. Actividad solicita a servidor eliminación de amistad. Actividad elimina solicitud de lista. Actividad notifica por pantalla rechazo exitoso.

Diagrama de interacción





#### a. Recibir solicitud de amistad

- **Actor:** Usuario
- **Propósito:** Poder recibir las aceptaciones de amistad generadas a partir de solicitudes hechas por él.
- **PreCondiciones:**
  - Haberse autenticado en el sistema.
- **PostCondiciones:** El usuario estará informado de su nuevo amigo.
- **Resumen:** El dispositivo le informará al usuario de las solicitudes de amistad enviadas aceptadas.
- **Requerimientos asociados:**

#### Flujo principal

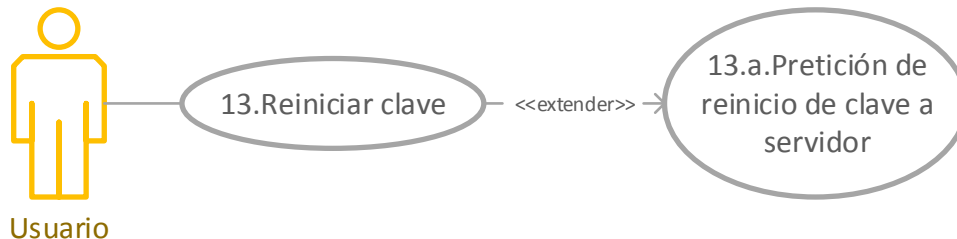
3. ServerListener detecta nueva solicitud. ServerListener solicita a dispositivo notificación push.
4. Usuario hace clic en notificación push. **Extiende caso de uso 12, Aceptar o rechazar solicitud de amistad.**

#### Flujos alternativos

Línea 1: Si actividad de gestión de amigos está abierta en la pestaña correcta, Actividad actualiza listado de solicitudes.



### 13. Reiniciar clave



- **Actor:** Usuario
- **Propósito:** En caso de que el usuario haya olvidado la clave, el usuario podrá reiniciarla y crear una nueva.
- **PreCondiciones:**
  - Haberse registrado anteriormente.
  - Estar en la actividad de login.
- **PostCondiciones:** El usuario tendrá una nueva clave con la cual podrá hacer el caso de uso 2, autenticación.
- **Resumen:** En caso de que el usuario haya olvidado la clave, el usuario podrá reiniciarla y crear una nueva.
- **Requerimientos asociados:** RF13

#### Flujo principal

1. El usuario hace clic sobre icono en menú de aplicaciones. La aplicación se abre.
2. La aplicación abre formulario inicio de sesión. Usuario inserta email.
3. Usuario hace clic en reiniciar clave. **Extiende caso de uso 12.a, petición de reinicio de clave a servidor.**
4. Actividad muestra notificación que nos dirá que un correo ha sido enviado a nuestra casilla para continuar el proceso. Si el usuario tiene acceso a su casilla de correo, podrá acceder a una web donde podrá reiniciar la clave.

#### Flujos alternativos

Línea 3: En caso de haber problemas de conexión u otros para realizar el envío de la solicitud, se le notificará al usuario el error, y el caso de uso no continuará. Nos quedaríamos en la actividad de login.

Línea 3: En caso de no existir el correo, nos aparecerá un error pidiéndonos que nos registremos en la herramienta. El caso no continúa y nos quedamos en la actividad de login.

Línea 3: En caso de no insertar correo, nos aparecerá un error pidiéndonos que lo insertemos.



#### a. Petición de reinicio de clave a servidor.

- **Actor:** Usuario
- **Propósito:** En caso de que el usuario haya olvidado la clave, el usuario podrá reiniciarla y crear una nueva.
- **PreCondiciones:**
  - Haberse registrado anteriormente.
  - Estar en la actividad de login.
- **PostCondiciones:** El usuario tendrá una nueva clave con la cual podrá hacer el caso de uso 2, autenticación.
- **Resumen:** En caso de que el usuario haya olvidado la clave, el usuario podrá reiniciarla y crear una nueva.
- **Requerimientos asociados:** RF13

#### Flujo principal

1. Usuario hace clic en reinicio de clave. Actividad solicita reinicio de clave a servidor.
2. Servidor valida solicitud. Actividad notifica envío de correo.

#### Flujos alternativos

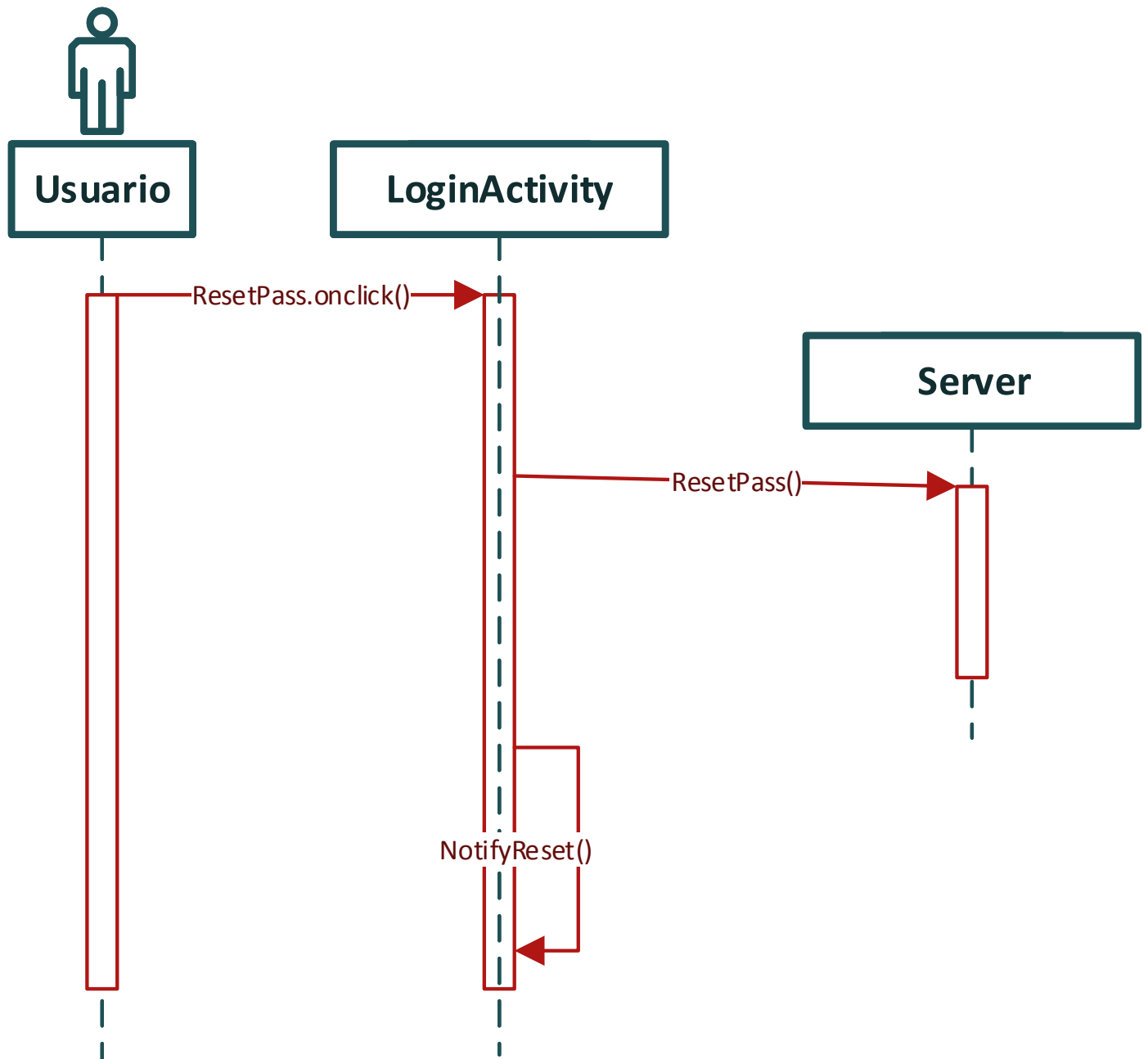
Línea 2: En caso de haber problemas de conexión u otros para realizar el envío de la solicitud, se le notificará al usuario el error, y el caso de uso no continuará. Nos quedaríamos en la actividad de login.

Línea 2: En caso de no existir el correo, nos aparecerá un error pidiéndonos que nos registremos en la herramienta. El caso no continua y nos quedamos en la actividad de login.

Línea 1: En caso de no haber insertado correo en caso de uso 12, nos aparecerá un error pidiéndonos que lo insertemos.

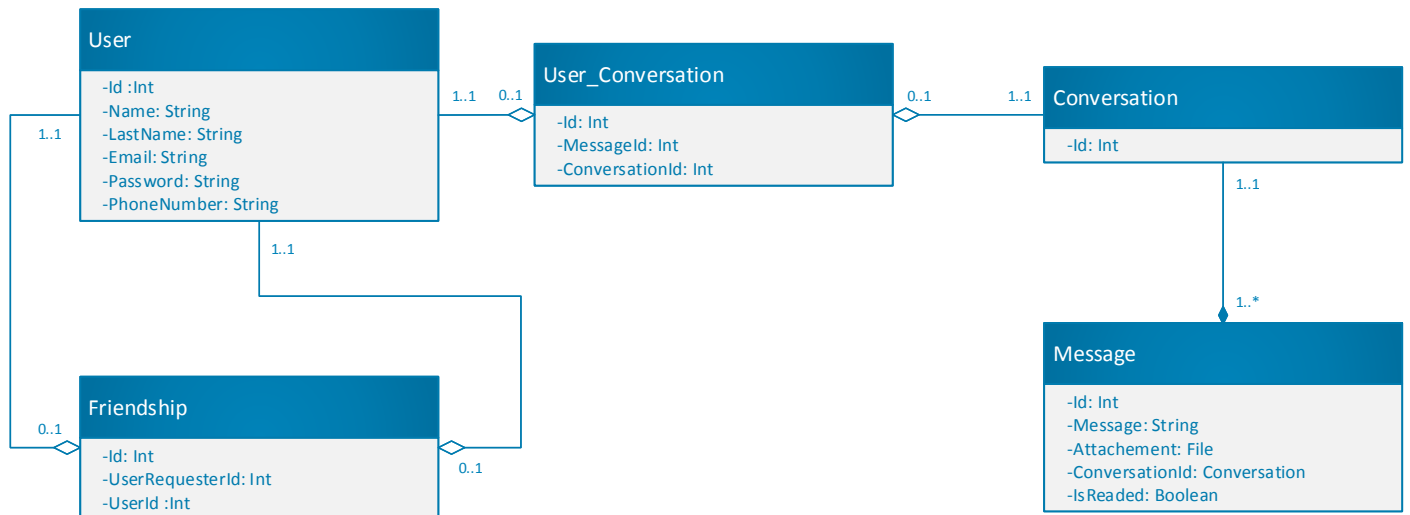


Diagrama de interacción



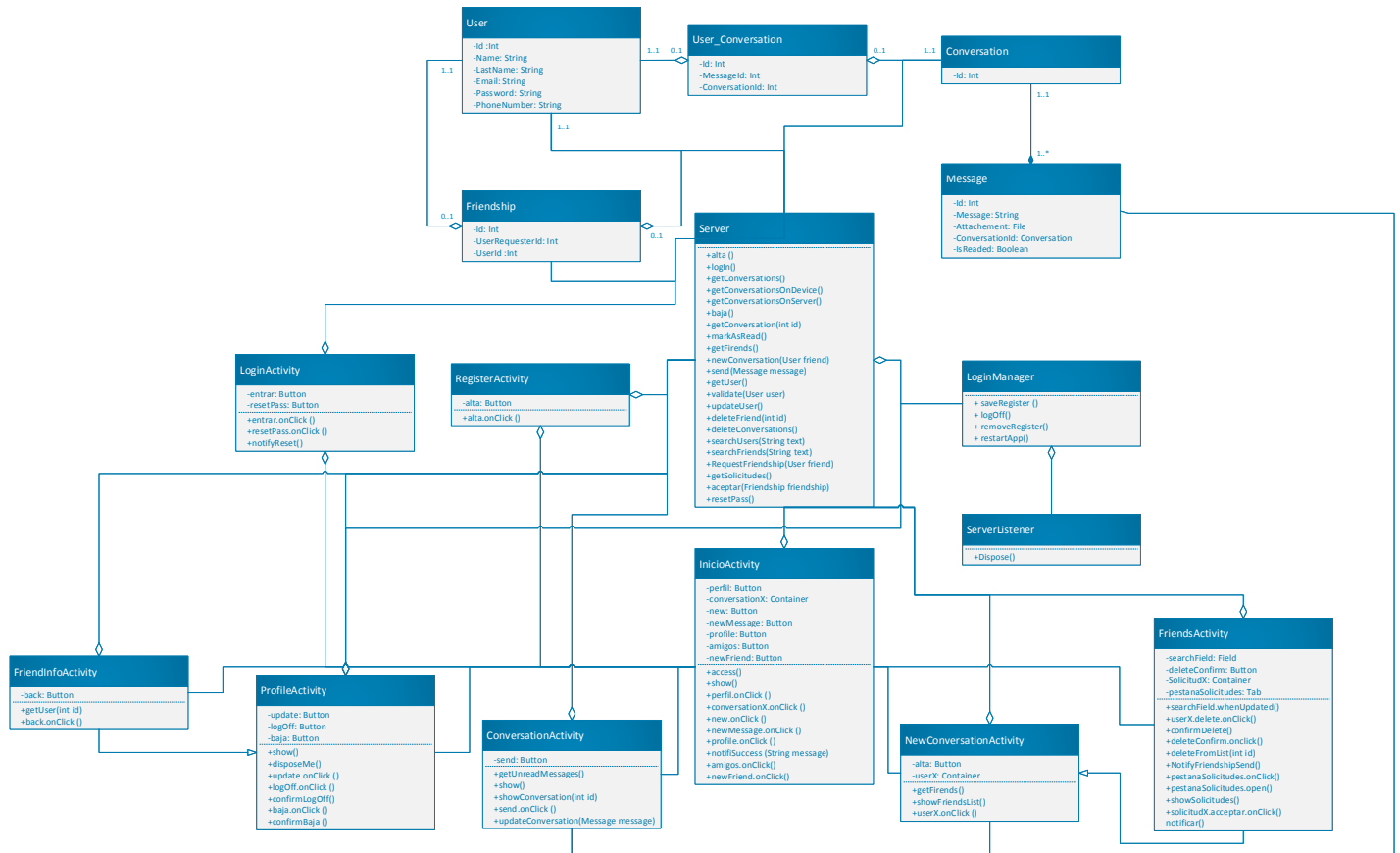


## 4. Diagrama de clases servidor





## 5. Diagrama de clases Dispositivo

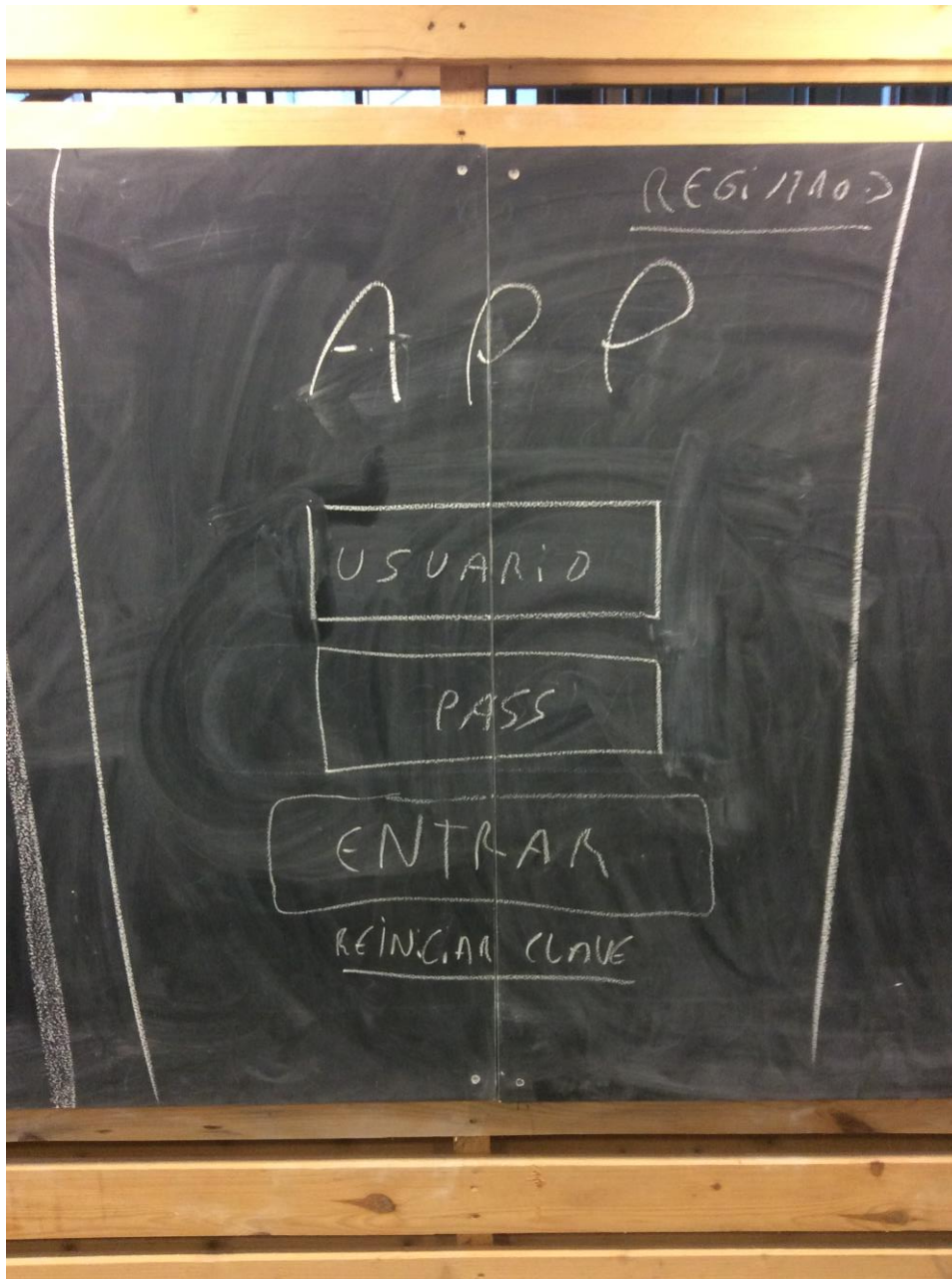




## 6. Wireframes Generales

---

### Actividad de Login







## Actividad de registro

APP  
REGISTRO

NOMBRE

APELLIDO

EMAIL

TELÉFONO

PAIS

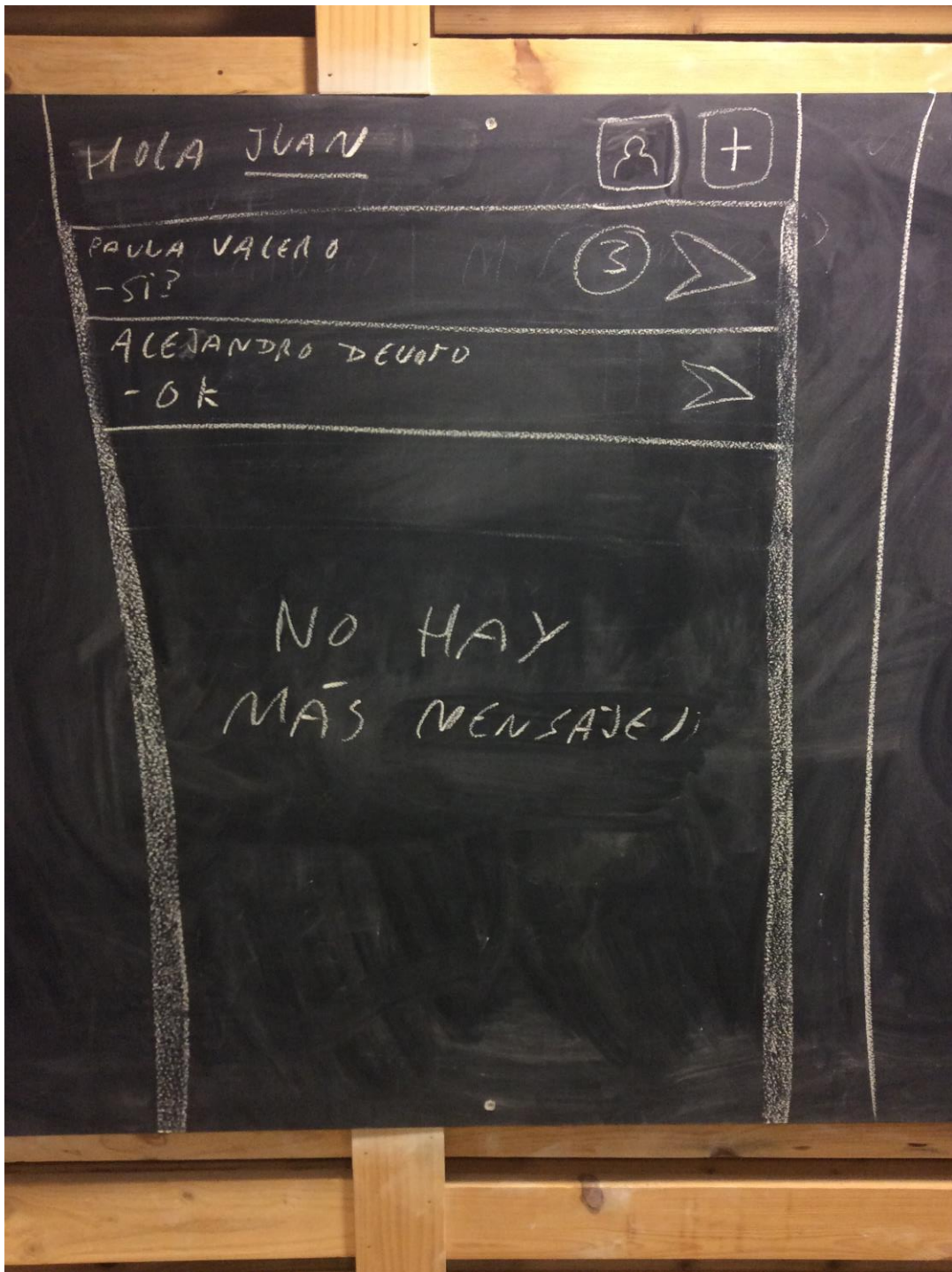
AERIDA PAIS

ENVIAR



#### Actividad de inicio

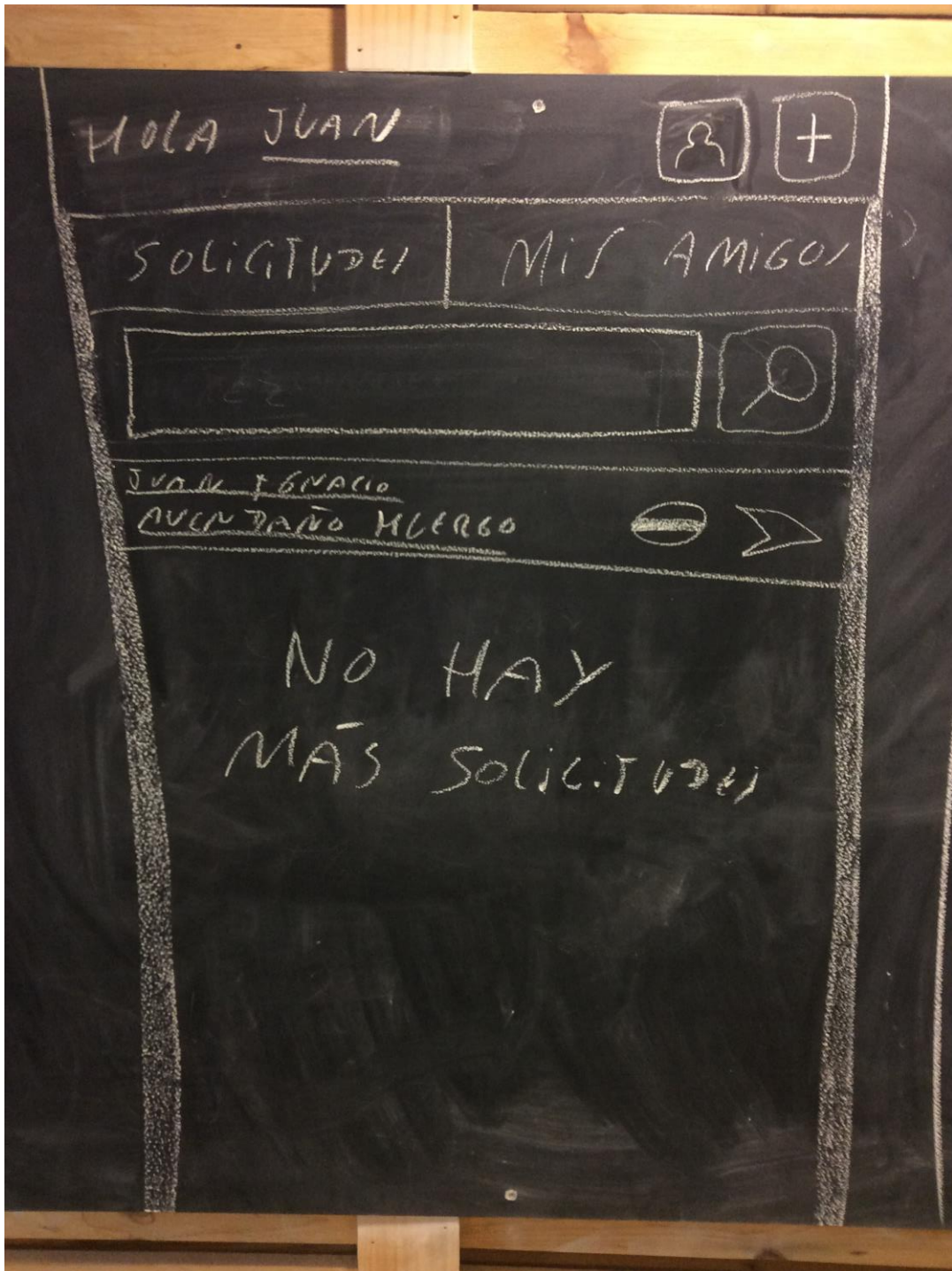
A veces referenciada como pantalla principal, menú principal, actividad principal, o inicioActivity.







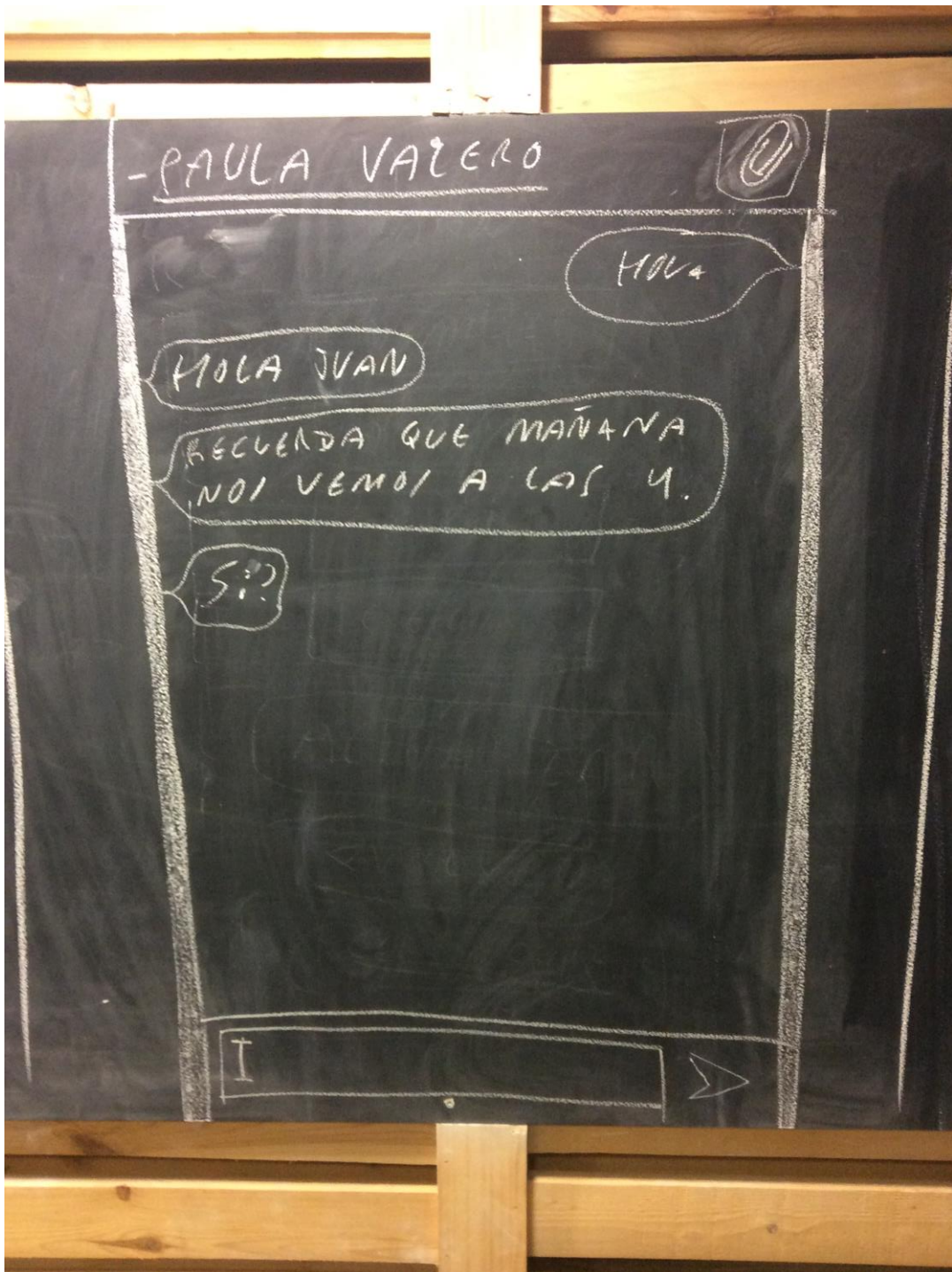
Actividad de solicitudes





### Actividad de conversación

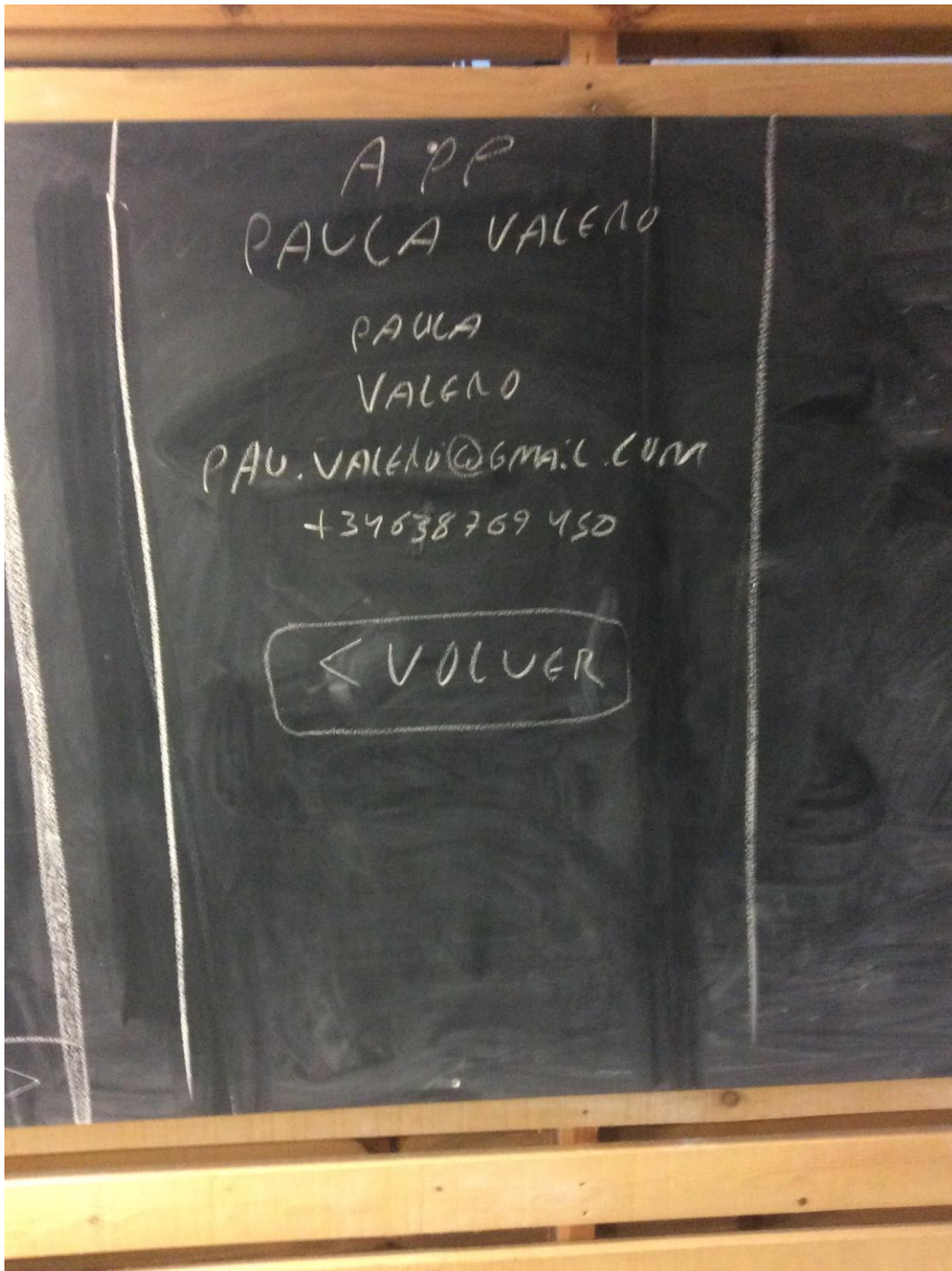
A veces referenciada como pantalla de conversación, o conversation activity.







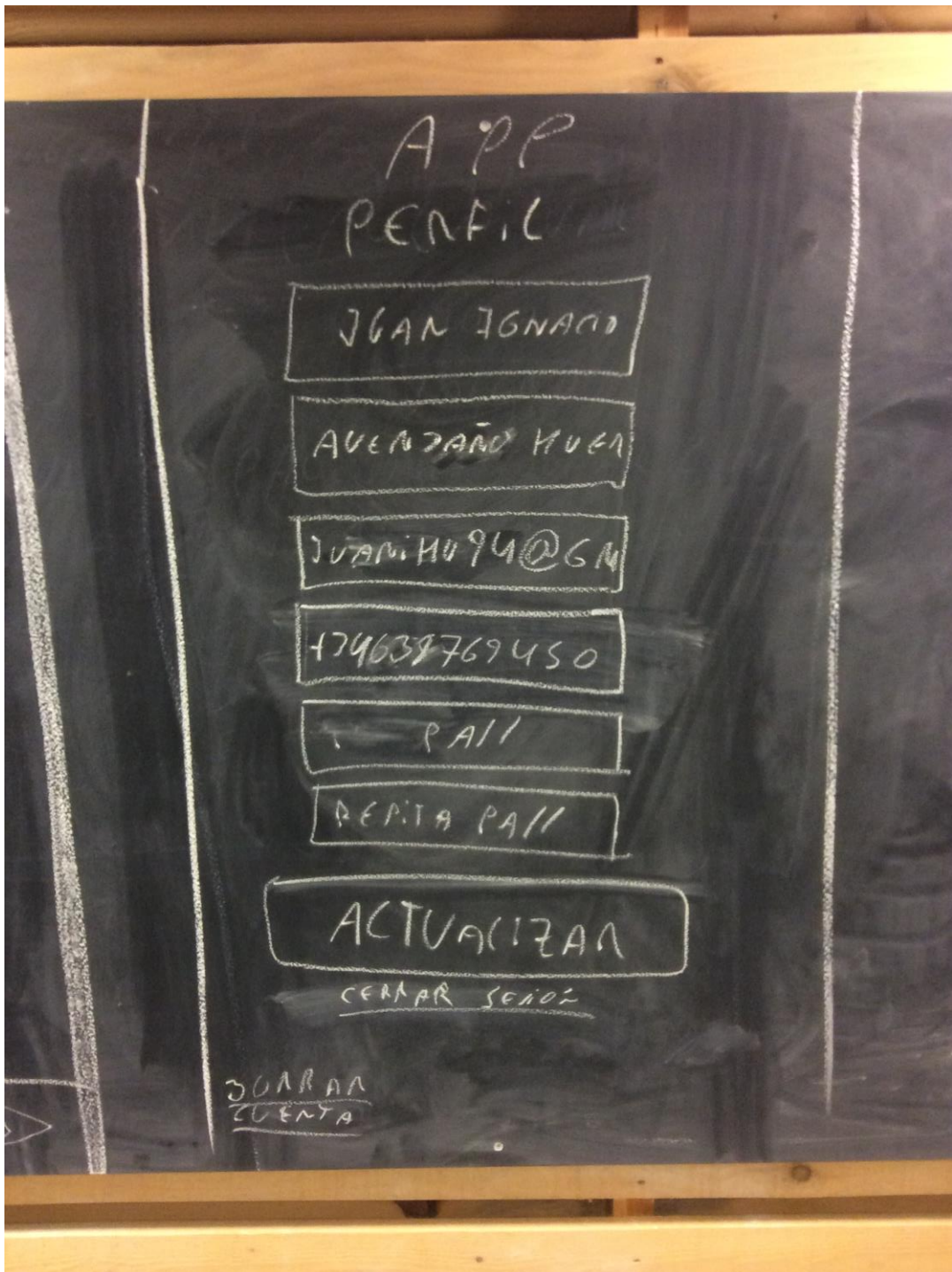
### Actividad Perfil de amigo





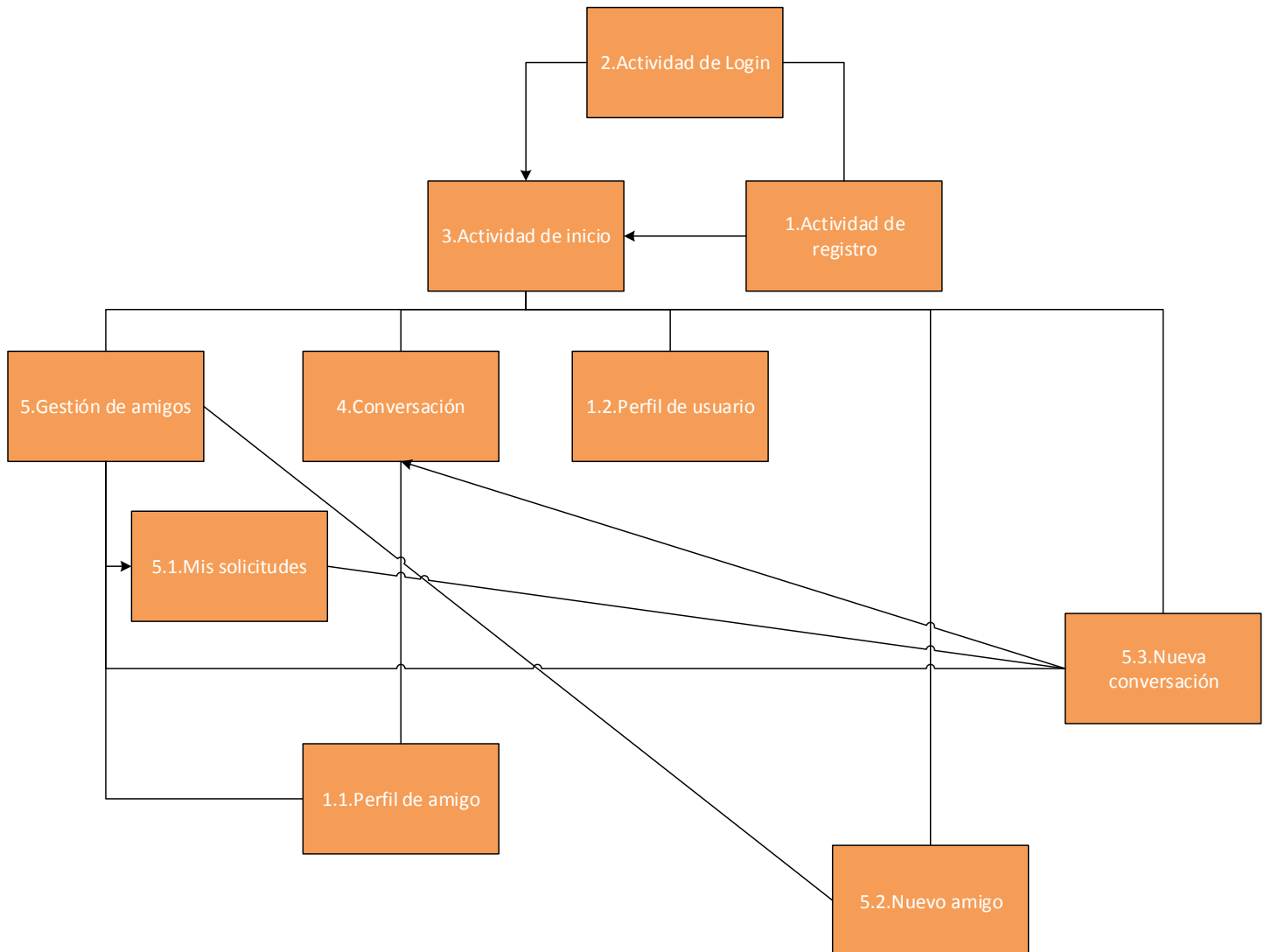
### Actividad de perfil

A veces referenciada como pantalla de perfil, o Perfil Activity.





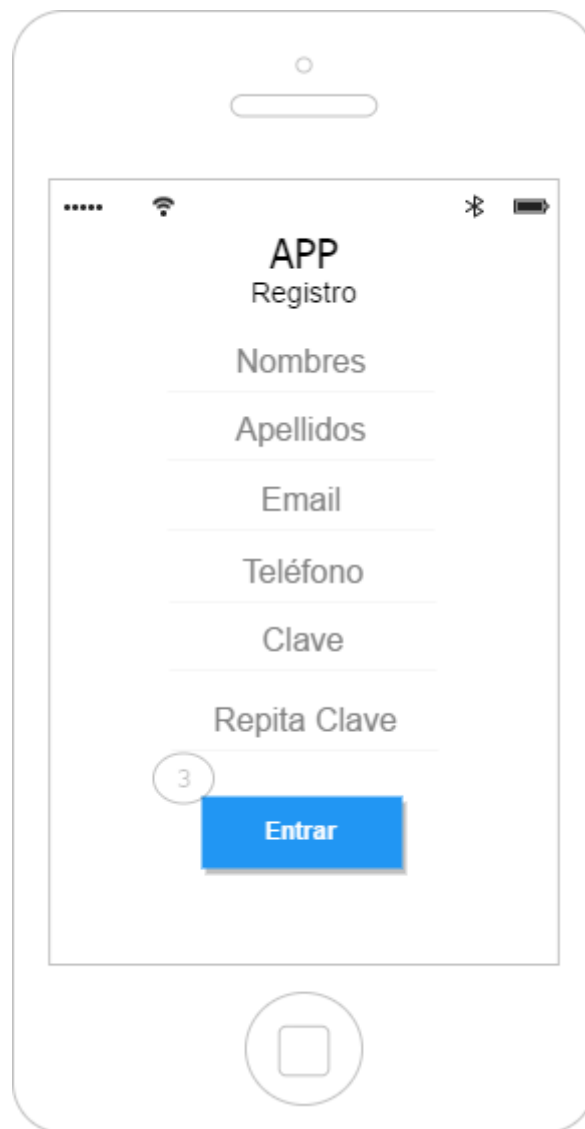
## 7. Workflow



## 8. Wireframes específicos



## 1. RegisterActivity







## Perfil de Amigo

The mockup shows a mobile app interface for a friend's profile. At the top, there's a status bar with signal strength, Wi-Fi, Bluetooth, and battery icons. The main content area is titled 'APP Perfil'. Below the title, there are several input fields for personal information: 'Juan Ignacio', 'Avendaño Huergo', 'juanihu94@gmail.c', and '+34638769451'. There are also fields for 'Clave' (Password) and 'Repita Clave' (Repeat Password). A blue 'Actualizar' (Update) button is positioned below the password fields, with a small circle containing the number '3' next to it. Below the 'Actualizar' button is a link that says 'Cerrar Sesión' (Log Out). At the bottom left of the form, there is a link that says 'Borrar Cuenta' (Delete Account). The entire form is enclosed in a rounded rectangle, and the whole screen is framed by a larger rounded rectangle representing the phone's body, with a home button at the bottom.

APP  
Perfil

Juan Ignacio

Avendaño Huergo

juanihu94@gmail.c

+34638769451

Clave

Repita Clave

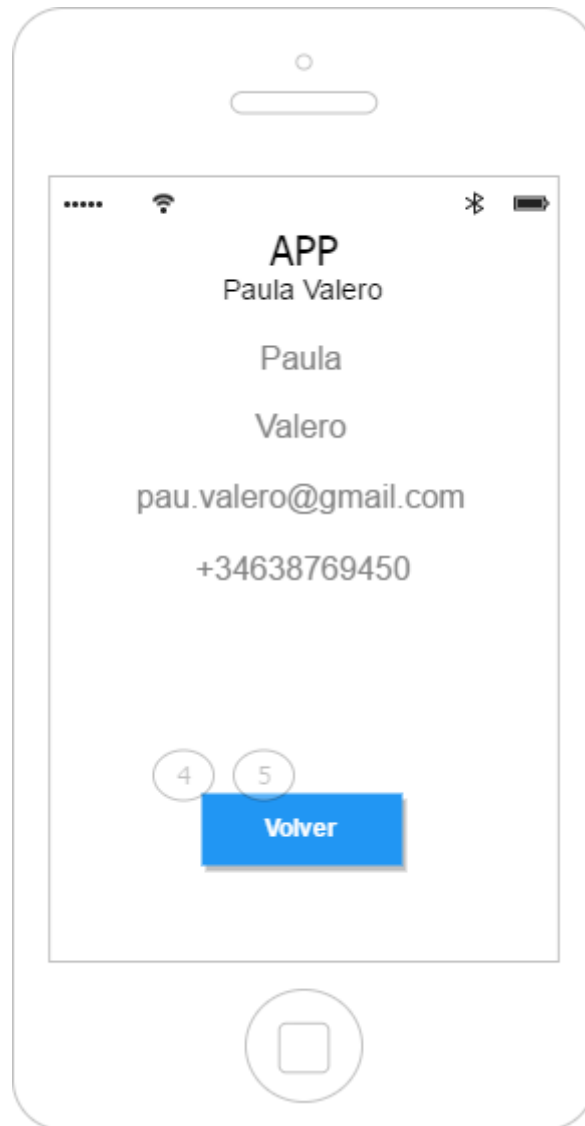
3 Actualizar

Cerrar Sesión

Borrar Cuenta

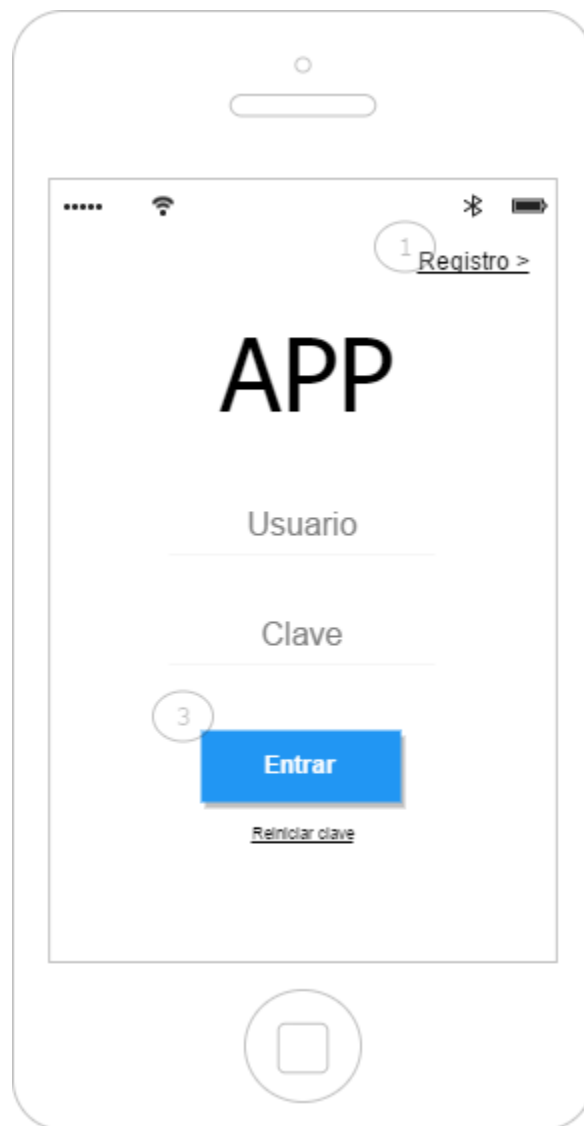


PerfilActivity





## 2. LoginActivity



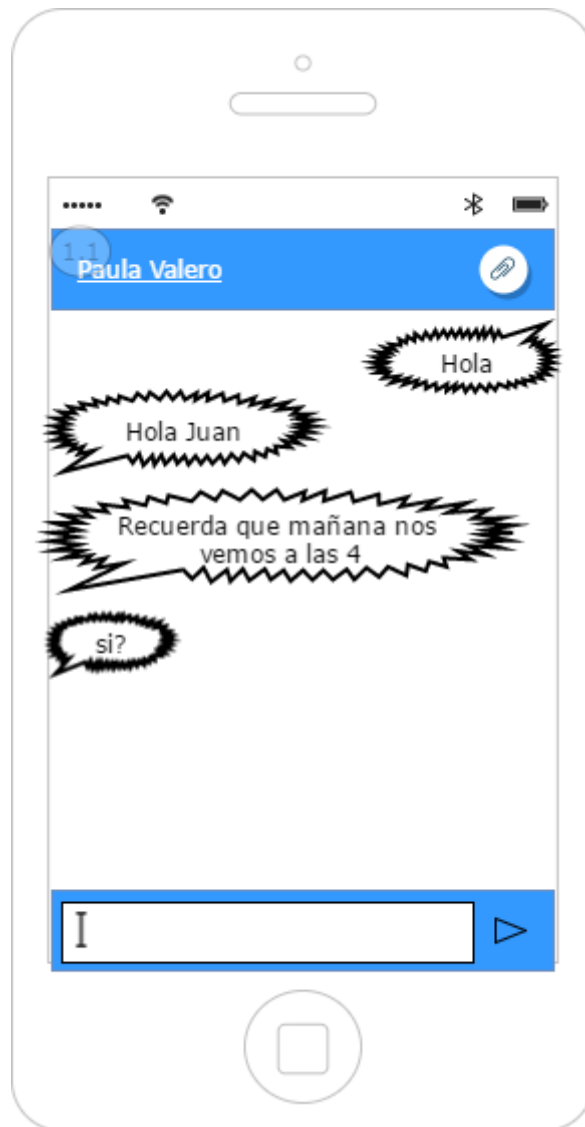


### 3. InicioActivity





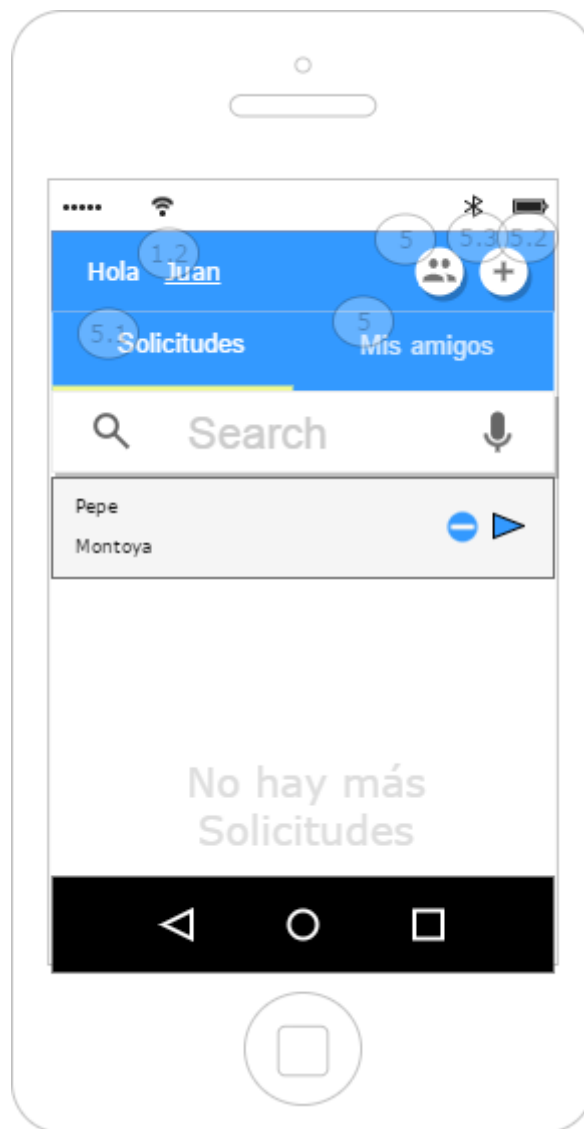
#### 4. ConversationActivity





## 5. FriendsActivity

### Solicitudes





## Amigos

