



UNIVERSIDADE DA CORUÑA



JoinMe!
Realización Casos de Uso
Version 1.0

José Antonio López Sebio
Pablo Paz Varela
Grupo ER-12-03

Histórico

Fecha	Version	Descripción	Autor
15/04/2014	1.0	Primera Revisión	ER-12-03

Índice

1. Introduction	3
1.1. Objetivo	3
1.2. Ámbito	3
1.3. Definiciones, Acrónimos y Abreviaturas	3
1.4. Referencias	3
2. CU01 - Registrar usuario	3
2.1. Flujo de Eventos	3
2.2. Diagrama de interacción	4
2.3. Objetos participantes	4
2.4. Diagrama de clases	4
2.5. Requisitos derivados	4
3. CU03 - Aceptar solicitud de amistad	4
3.1. Flujo de Eventos	5
3.2. Diagrama de interacción	5
3.3. Objetos participantes	5
3.4. Diagrama de clases	5
3.5. Requisitos derivados	5
4. CU04 - Rechazar solicitud de amistad	6
4.1. Flujo de Eventos	6
4.2. Diagrama de interacción	6
4.3. Objetos participantes	6
4.4. Diagrama de clases	6
4.5. Requisitos derivados	7
5. CU05 - Ver amigos	7
5.1. Flujo de Eventos	7
5.2. Diagrama de interacción	7
5.3. Objetos participantes	7
5.4. Diagrama de clases	8
5.5. Requisitos derivados	8
6. CU06 - Enviar solicitud de amistad	8
6.1. Flujo de Eventos	8
6.2. Diagrama de interacción	8
6.3. Objetos participantes	9
6.4. Diagrama de clases	9
6.5. Requisitos derivados	9
7. CU011 - Crear entrada	9
7.1. Flujo de Eventos	9
7.2. Diagrama de interacción	9
7.3. Objetos participantes	11
7.4. Diagrama de clases	11
7.5. Requisitos derivados	11

1. Introduction

1.1. Objetivo

Este documento es una visión del sistema empleando diagramas, mostrando así la interacción entre las distintas partes del mismo.

1.2. Ámbito

JoinMe! es una red social, basado en una arquitectura cliente servidor que permite a los usuarios mantener el contacto con sus amigos, y a las empresas anunciarse de una manera más eficiente.

1.3. Definiciones, Acrónimos y Abreviaturas

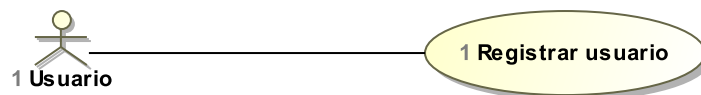
Las definiciones, acrónimos y abreviaturas se encuentran en el documento *Glosario*.

1.4. Referencias

1. JoinMe! - Glosario
2. JoinMe! - Modelo de casos de uso
3. JoinMe! - Especificación suplementaria
4. JoinMe! - Modelo del dominio
5. JoinMe! - Arquitectura lógica
6. JoinMe! - DSS

2. CU01 - Registrar usuario

Este caso de uso permite a un futuro usuario poder registrarse en el sistema y así poder acceder a todas las funcionalidades disponibles en **JoinMe!**. Para ello el usuario tendrá que proporcionar sus datos personales o utilizar un certificado digital como puede ser el DNI-e.



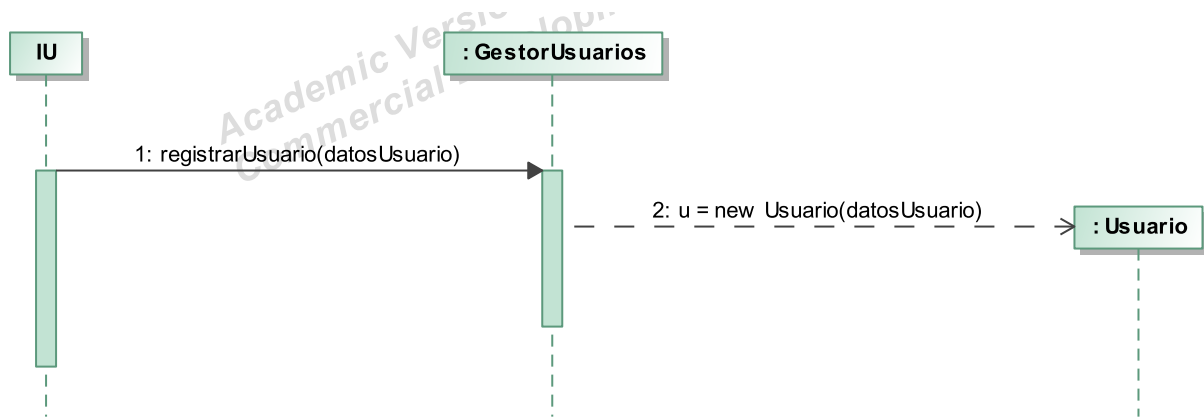
2.1. Flujo de Eventos

El usuario introduce sus datos personales necesarios para el registro o utilizar un certificado digital.

2.2. Diagrama de interacción

En el caso de REGISTRAR USUARIO se han identificado las siguientes operaciones:

- El usuario introduce sus datos y los envía al sistema mediante la operación **registrarUsuario**.



2.3. Objetos participantes

Clase	Descripción
GestorUsuarios	Clase que se encarga de realizar las operaciones permitidas sobre los usuarios, tales como registro, modificación y baja.
Usuario	Clase que representa al usuario.

2.4. Diagrama de clases

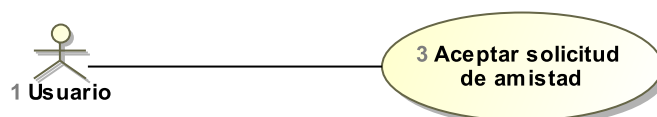
Se ha optado por realizar un diagrama de clases completo, éste, puede verse en la sección 8 Diagrama de Clases.

2.5. Requisitos derivados

- Recuperación robusta cuando el usuario inserta datos incorrectos.
- Información por pantalla en caso de error.

3. CU03 - Aceptar solicitud de amistad

Este caso de uso representa la acción de aceptar una solicitud de amistad enviada por otro usuario de la red social **JoinMe!** que quiere añadir al usuario en cuestión a su círculo de amistades.



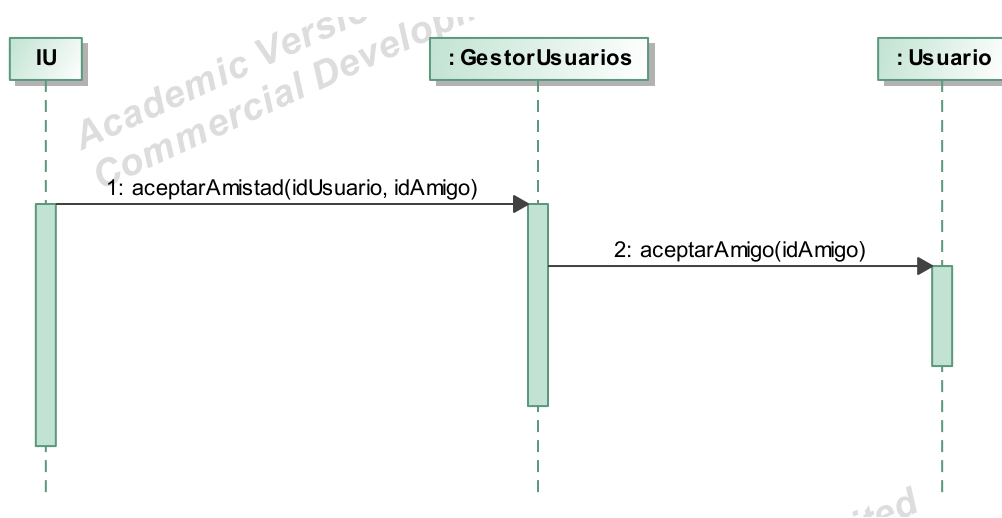
3.1. Flujo de Eventos

El usuario acepta la solicitud de amistad que tiene en su bandeja de solicitudes haciendo click en el botón oportuno.

3.2. Diagrama de interacción

En el caso de ACEPTAR SOLICITUD DE AMISTAD se han identificado las siguientes operaciones:

- El usuario acepta la petición de amistad a través de la operación **aceptarAmistad**.
- El sistema registra la aceptación y añade el amigo a lista de amigos con la operación **aceptarAmigo**.



3.3. Objetos participantes

Clase	Descripción
GestorUsuarios	Clase que se encarga de realizar las operaciones permitidas sobre los usuarios, tales como registro, modificación y baja.
Usuario	Clase que representa al usuario.

3.4. Diagrama de clases

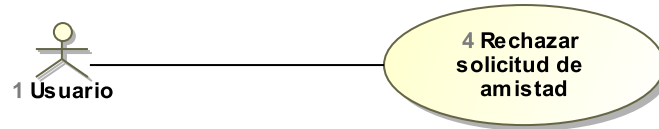
Se ha optado por realizar un diagrama de clases completo, éste, puede verse en la sección 8 Diagrama de Clases.

3.5. Requisitos derivados

- Recuperación robusta en caso de fallo del sistema.
- Información por pantalla en caso de error.

4. CU04 - Rechazar solicitud de amistad

Caso de uso que representa la acción de rechazar una petición de amistad enviada anteriormente por otro usuario de **JoinMe!**.



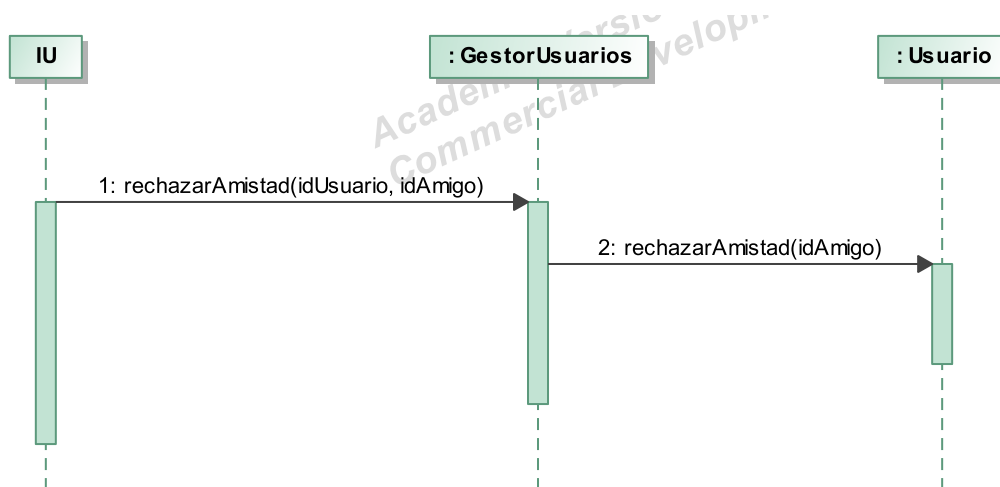
4.1. Flujo de Eventos

El usuario rechaza la solicitud de amistad que tiene en la bandeja de solicitudes de amistad haciendo uso del botón de rechazar solicitud.

4.2. Diagrama de interacción

En el caso de RECHAZAR SOLICITUD DE AMISTAD se han identificado las siguientes operaciones:

- El usuario rechazar la solicitud con la operación **rechazarAmistad**.



4.3. Objetos participantes

Clase	Descripción
GestorUsuarios	Clase que se encarga de realizar las operaciones permitidas sobre los usuarios, tales como registro, modificación y baja.
Usuario	Clase que representa al usuario.

4.4. Diagrama de clases

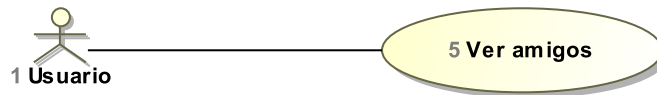
Se ha optado por realizar un diagrama de clases completo, éste, puede verse en la sección 8 Diagrama de Clases.

4.5. Requisitos derivados

- Recuperación robusta cuando el usuario inserta datos incorrectos.
- Información por pantalla en caso de error.

5. CU05 - Ver amigos

Caso de uso que representa la opción de listar todos los amigos que tiene un usuario en la red social.



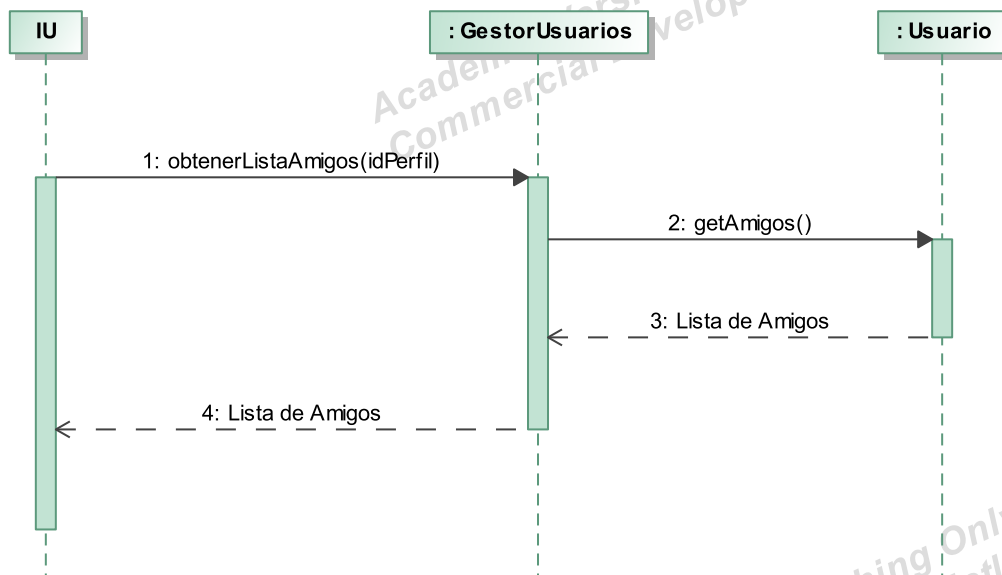
5.1. Flujo de Eventos

El usuario mediante la opción disponible en la interfaz, solicita consultar su lista de amigos.

5.2. Diagrama de interacción

En el caso de VER AMIGOS se han identificado las siguientes operaciones:

- El usuario obtiene la lista de amigos haciendo uso de la operación **obtenerListaAmigos**.



5.3. Objetos participantes

Clase	Descripción
GestorUsuarios	Clase que se encarga de realizar las operaciones permitidas sobre los usuarios, tales como registro, modificación y baja.
Usuario	Clase que representa al usuario.

5.4. Diagrama de clases

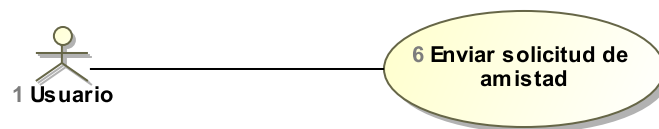
Se ha optado por realizar un diagrama de clases completo, éste, puede verse en la sección 8 Diagrama de Clases.

5.5. Requisitos derivados

- Recuperación robusta cuando el usuario inserta datos incorrectos.
- Información por pantalla en caso de error.

6. CU06 - Enviar solicitud de amistad

Caso de uso que representa la funcionalidad de enviar una solicitud a otro usuario de la res social que deseamos tener entre nuestros contactos en la red.



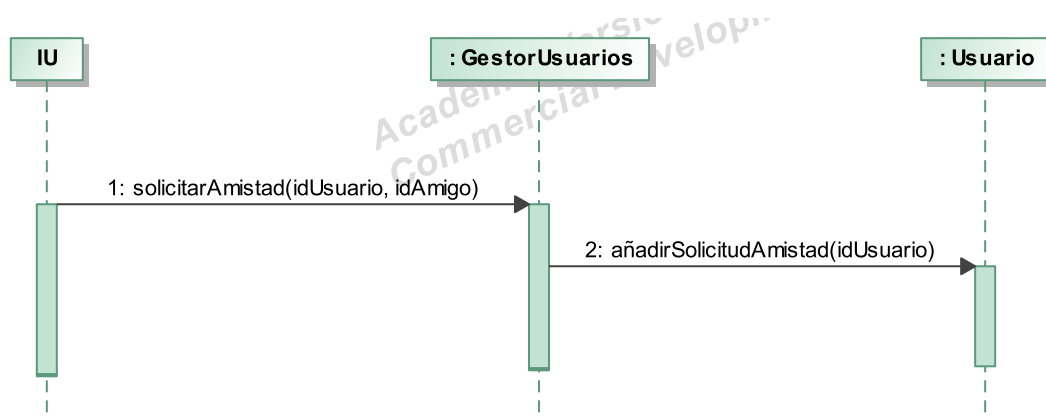
6.1. Flujo de Eventos

El usuario, envía una solicitud de amistad haciendo click en el botón de enviar solicitud que está en la interfaz del perfil del usuario que desea agregar.

6.2. Diagrama de interacción

En el caso de ENVIAR SOLICITUD DE AMISTAD se han identificado las siguientes operaciones:

- El usuario envía la solicitud de amistad al otro usuario haciendo uso de la operación `solicitarAmistad`.



6.3. Objetos participantes

Clase	Descripción
GestorUsuarios	Clase que se encarga de realizar las operaciones permitidas sobre los usuarios, tales como registro, modificación y baja.
Usuario	Clase que representa al usuario.

6.4. Diagrama de clases

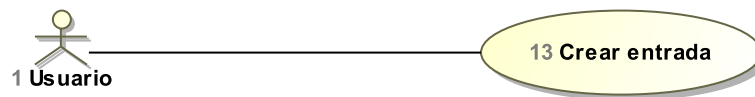
Se ha optado por realizar un diagrama de clases completo, éste, puede verse en la sección 8 Diagrama de Clases.

6.5. Requisitos derivados

- Recuperación robusta cuando el usuario inserta datos incorrectos.
- Información por pantalla en caso de error.

7. CU011 - Crear entrada

Este caso de uso muestra la funcionalidad de crear una entrada que será publicada en la red social. Una entrada que podrá contener, imágenes o vídeos y podrá ser comentada por otros usuarios.

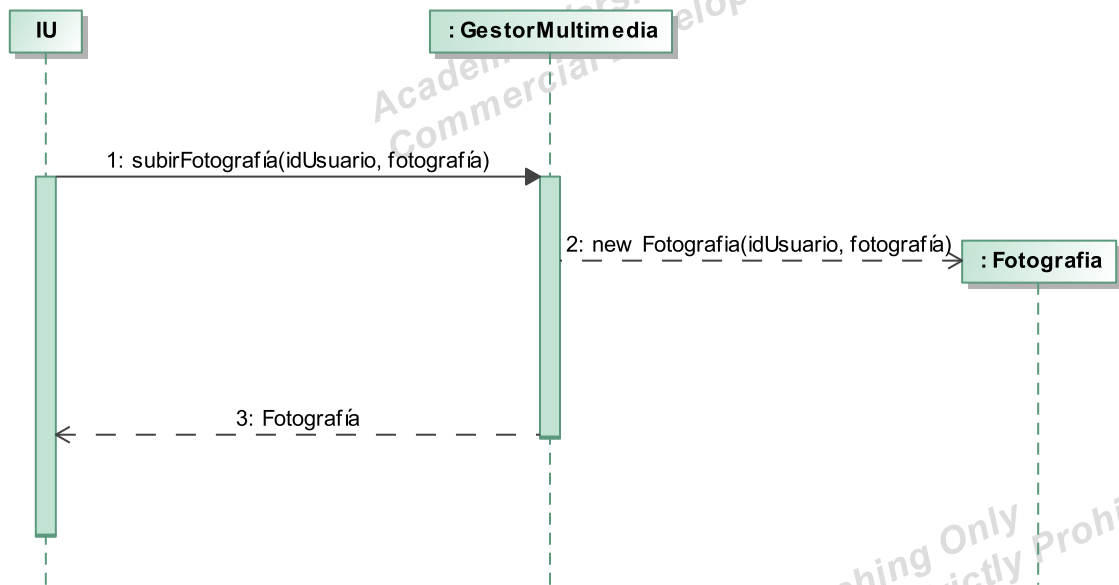
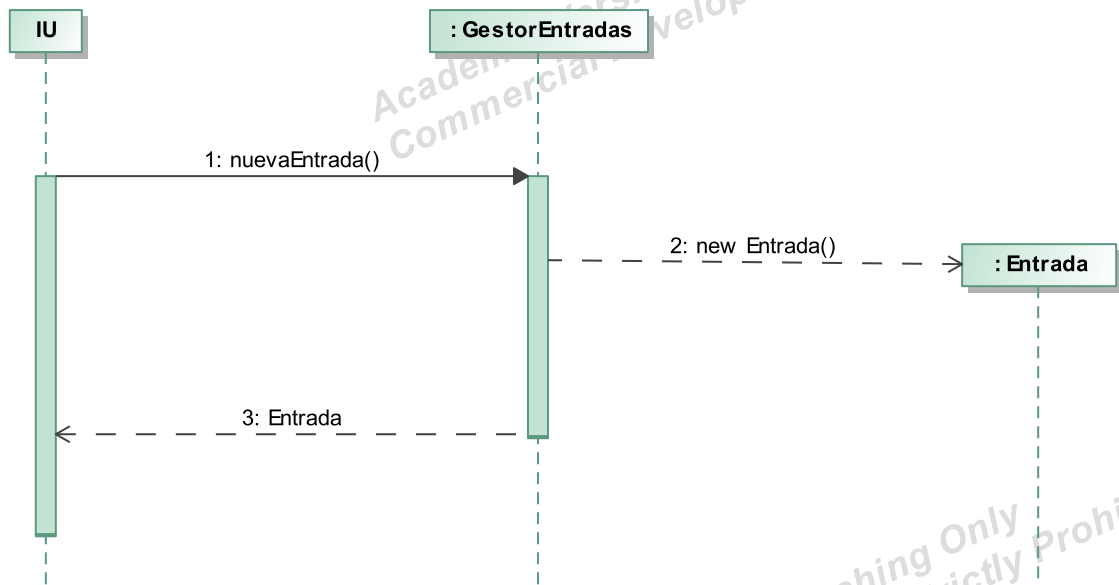


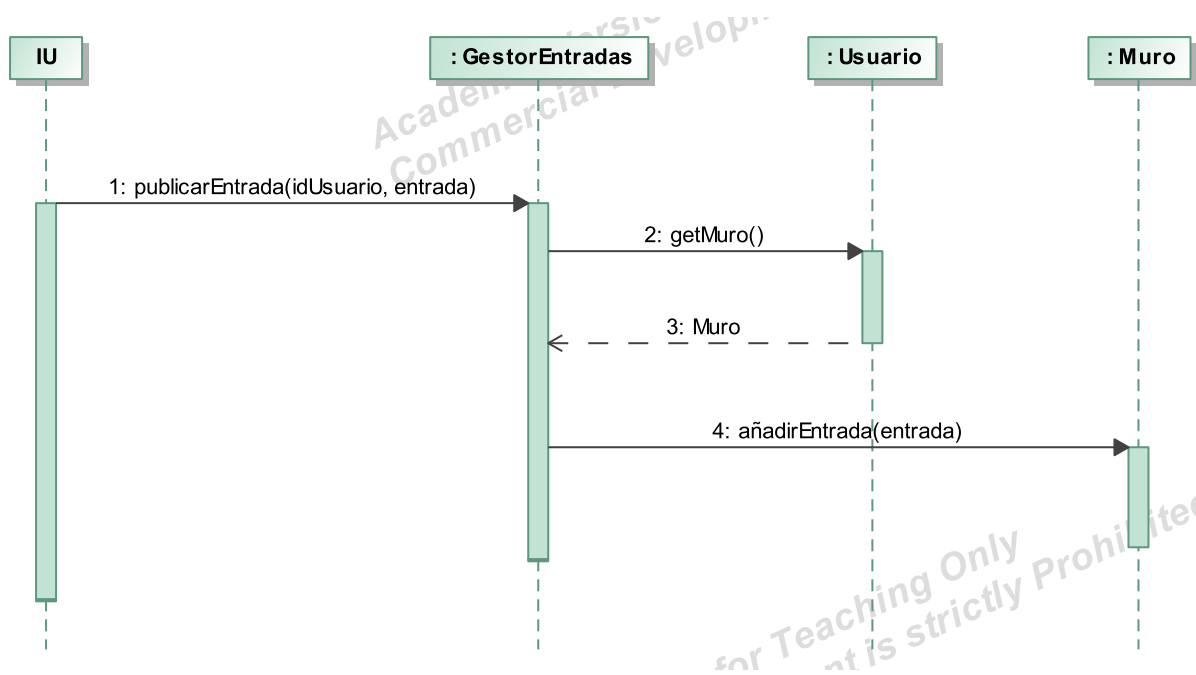
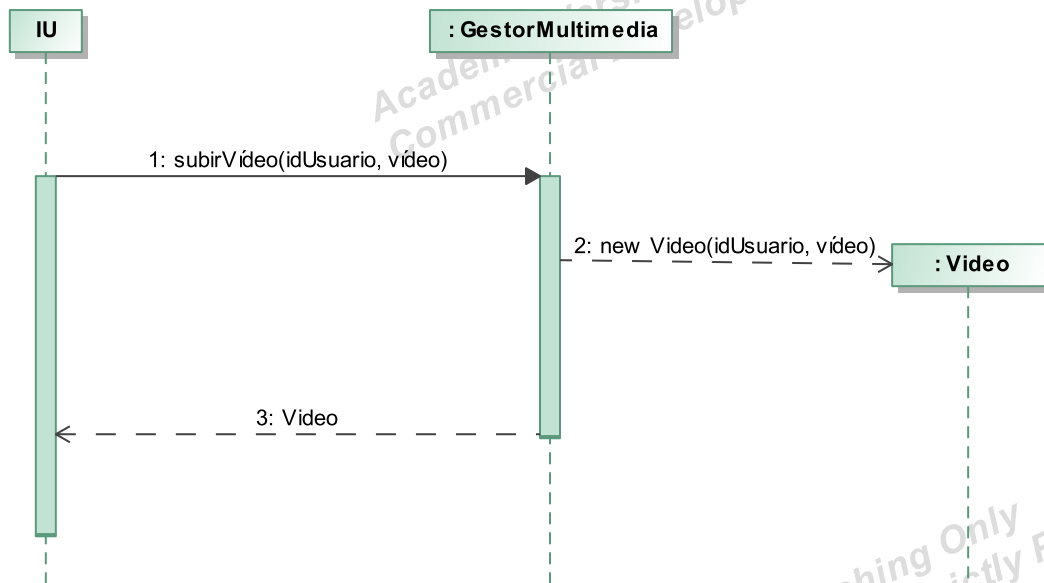
7.1. Flujo de Eventos

7.2. Diagrama de interacción

En el caso de CREAR ENTRADA se han identificado las siguientes operaciones:

-





7.3. Objetos participantes

Clase	Descripción
-------	-------------

7.4. Diagrama de clases

Se ha optado por realizar un diagrama de clases completo, éste, puede verse en la sección 8 Diagrama de Clases.

7.5. Requisitos derivados

- Recuperación robusta cuando el usuario inserta datos incorrectos.

- Información por pantalla en caso de error.

8. Diagrama de clases

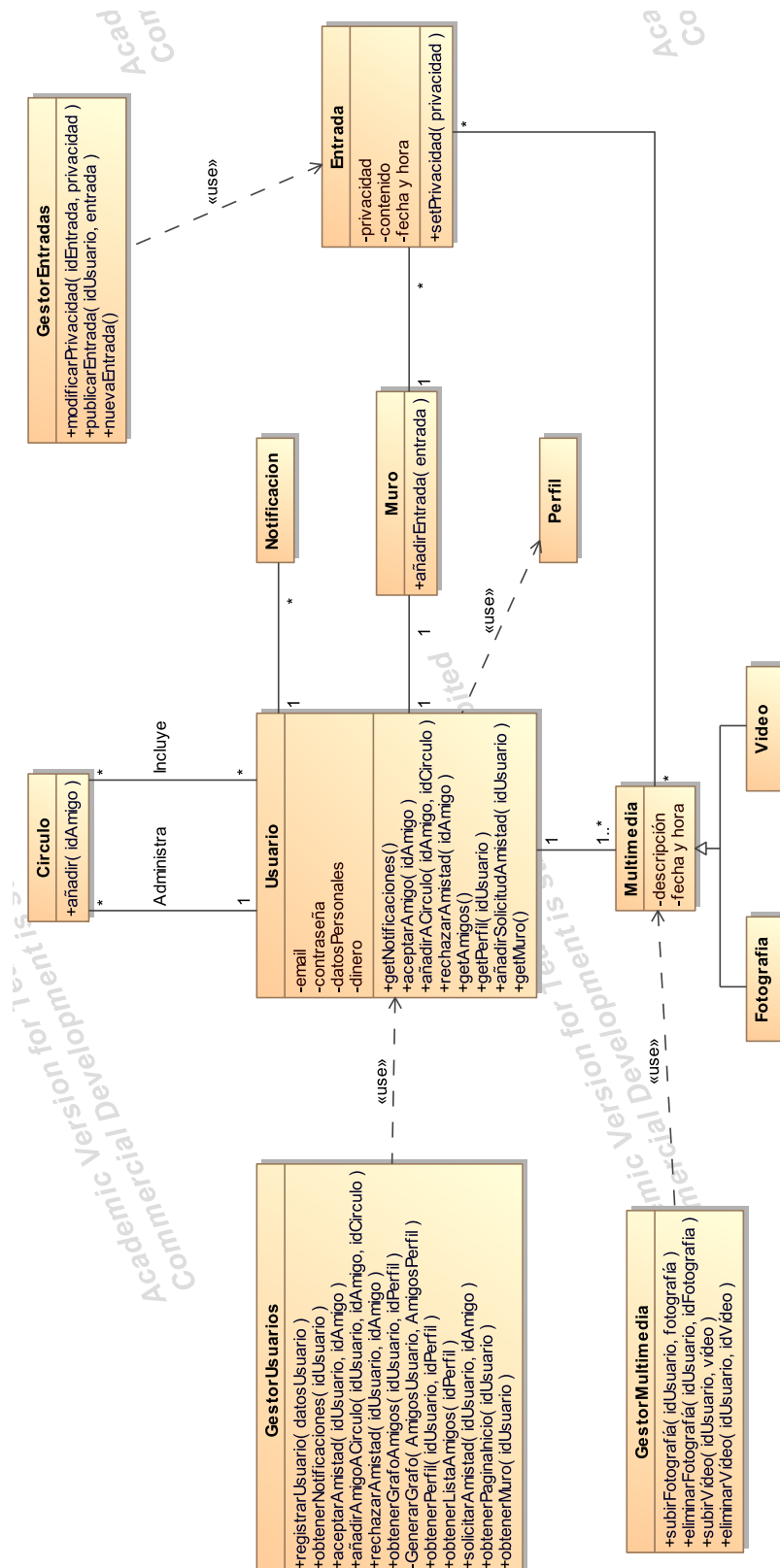


Figura 1: Diagrama de clases