

Trabajo Fin de Grado

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

Desarrollo de un sistema de videoconferencia en HTML 5.0

Autor: José Alberto Jinoria Fernández

Tutor: Javier Sánchez Pérez

Las Palmas de Gran Canaria

Diciembre 2013

Trabajo Fin de Grado realizado en la Escuela de Ingeniería Informática de la Universidad de Las Palmas de Gran Canaria, para la consecución del título de Ingeniero Informático.

Título: Desarrollo de un sistema de videoconferencia en HTML 5.0

Alumno: José Alberto Jinoria Fernández

Tutor: Javier Sánchez Pérez

Fecha: Diciembre, 2013

Este un proyecto sin ánimo de lucro basado en la plataforma de software libre WebRTC y HTML5.0.

Agradecimientos

A mi familia, por saberme guiar en la vida por el camino correcto y haberse sacrificado mucho, siempre pensando en mi porvenir.

A mis profesores, en especial a José Juan Hernández y a mi tutor Javier Sánchez por su ayuda en muestra de formación como profesional.

A mis amigos, en especial a Dariel Rodríguez y Daniel Ocaña.

A la Universidad de Las Palmas, por haberme acogido como estudiante y formado hasta la culminación de mis estudios.

Índice general

Prefacio	8
1 Introducción	9
1.1 Motivación y objetivos	9
1.2 Objeto de estudio.....	9
1.3 Campo de acción	9
1.4 Hipótesis.....	10
1.5 Aportaciones	10
1.6 Organización del documento	10
2 Estado actual del arte.....	12
2.1 WebRTC.....	12
2.1.1 Historia	12
2.1.2 Definición	12
2.1.3 Diseño.....	12
2.2 Sistemas de videoconferencias más usados a nivel mundial.....	13
2.2.1 Skype	13
2.2.2 Google Hangouts	14
3 Recursos utilizados.....	15
3.1 Recursos software	15
3.1.1 Herramientas.....	15
3.1.2 Lenguajes de programación	16
3.2 Recursos hardware.....	16
3.2.1 Hardware de los clientes	17
3.2.2 Hardware de los servidores.....	17
4 Planificación del trabajo.....	18
4.1 Metodología de desarrollo.....	18
4.2 Planificación y temporización	20
4.3 Presupuesto	21
4.3.1 Coste del hardware	21
4.3.2 Coste del software.....	21
4.3.3 Coste del personal.....	22
4.3.4 Presupuesto total del proyecto.....	22
5 Desarrollo del trabajo.....	23
5.1 Modelo del dominio	23

5.2	Lista de características	24
5.3	Requisitos del software	28
5.3.1	Actores	28
5.3.2	Modelo de casos de uso	28
5.3.3	Especificación de casos de uso	31
5.3.4	Prototipos de interfaz de usuario.....	35
5.4	Modelo de análisis	41
5.4.1	Diagrama de clases.....	41
5.4.2	Diagramas de colaboración	47
5.5	Modelo de diseño.....	53
5.5.1	Arquitectura del sistema	53
5.5.2	Diagramas de clases	54
5.5.3	Diagramas de secuencia	62
5.6	Implementación	70
5.6.1	Primeros pasos con CMS Elgg.....	70
5.6.2	Lo más importante está por llegar. WebRTC y HTML5	71
5.6.3	Integrar las dos aplicaciones para que parezcan una	71
5.7	Pruebas.....	72
6	Conclusiones y trabajo futuro	74
	Anexo I: Competencias.....	75
	Anexo II: Legislación vigente	77
	Anexo III: Manual de usuario	79
	Anexo IV: Entorno de Desarrollo.....	84
	Bibliografía	91

Prefacio

El documento expone un trabajo de fin grado de la carrera Grado en Ingeniería Informática, el cual tiene como objetivo desarrollar un sistema de videoconferencias a través de HTML5.0. Con este trabajo se pretende ampliar el uso de la web 2.0, la democratización de las comunicaciones, y lograr que los usuarios puedan comunicarse de una forma completamente libre, sin necesidad de instalación o actualización de software de terceros.

Basado en la metodología expuesta en el Proceso Unificado de Desarrollo de Software (PUD), se siguen todos sus procesos y flujos de trabajo, generando los productos y utilizando las técnicas y herramientas que proporciona la misma.

Se escoge el CMS¹ Elgg con el objetivo de lograr el comportamiento de una red social donde los usuarios puedan interactuar entre sí, libremente y con mayor usabilidad.

También se muestra la configuración del servidor web Node.js y de uno de sus módulos, socket.io, como herramienta de ayuda al navegador para lograr la señalización.

Además se hace uso de las APIs² de webRTC y de los elementos que brinda HTML5 para desarrollar el sistema de videollamadas.

¹ Sistema de gestión de contenidos.

² Interfaz de programación de aplicaciones.

1 Introducción

Una videoconferencia es una conexión multimedia entre dos o más personas que pueden verse, oírse e intercambiar recursos aunque estén separadas geográficamente. Su uso posibilita la realización de reuniones con grupos de personas situadas en lugares alejados entre sí, permitiendo el intercambio de información gráfica, transferencias de archivos, de vídeo, de voz, hacer presentaciones, servicio de atención al cliente, etc.

Las videoconferencias basan su eje tecnológico en la compresión digital de los flujos de audio y video en tiempo real. Su uso ofrece una solución accesible a la necesidad de comunicación. Los sistemas actuales permiten enviar y recibir información visual y sonora entre dos puntos que se encuentran en zonas diferentes, evitando así los gastos y pérdida de tiempo que implica el traslado físico de la persona. (1)

Estas ventajas hacen a la videoconferencia el segmento de mayor crecimiento en el área de las telecomunicaciones.

1.1 Motivación y objetivos

En la actualidad, las limitaciones técnicas tales como el sonido deficiente, la mala calidad de las imágenes, la poca fiabilidad, la complejidad y el coste han quedado atrás, dando lugar a videoconferencias con alta calidad de audio, video, transferencia de archivos y con un coste accesible a la mayoría de los interesados. (2)

Con HTML5.0, aparece la posibilidad de realizar videoconferencias sin la necesidad de instalación de software. Este lenguaje introduce soporte integrado para el contenido multimedia gracias a los elementos de audio y video, ofreciendo la posibilidad de incrustar este tipo de contenido en páginas HTML.

El objetivo principal del trabajo es: desarrollar una aplicación multiplataforma de videoconferencias en HTML5.0. Partiendo del mismo se trazaron los siguientes objetivos secundarios:

- ❖ Utilizar las tecnologías disponibles para el desarrollo de un sistema de comunicación en entorno web.
- ❖ Lograr una comunicación P2P privada entre dos personas.
- ❖ Crear una aplicación web que centralice la gestión de las comunicaciones entre los usuarios.

1.2 Objeto de estudio

Para la realización de este trabajo se plantea como objeto de estudio:

- ❖ La utilización de las APIs de WebRTC para el desarrollo de sistemas de comunicación en tiempo real.

1.3 Campo de acción

Dentro del objeto de estudio se define como campo de acción:

- ❖ El desarrollo de sistemas de comunicación con audio y video mediante la integración de las APIs de WebRTC y HTML5.

1.4 Hipótesis

Si se desarrolla una aplicación web para la interacción de un grupo de usuarios mediante audio y video, utilizando las ventajas que brinda WebRTC y su integración con HTML5.0, se obtendrá una solución Open Source que sirva como modelo de comunicación en tiempo real sin necesidad de descargar, instalar o actualizar ningún complemento de terceros.

1.5 Aportaciones

La aplicación a desarrollar permitirá interconectar, mediante sesiones interactivas, a dos interlocutores de forma que puedan verse y hablar entre sí, sin necesidad de descargar e instalar ningún plugin³ de terceros.

Libertad para compartir y conectarse

Las personas deben disfrutar de la libertad de compartir cualquier información que deseen, a través de cualquier medio y formato, tener el derecho de poder conectarse en línea con otros (personas, organizaciones o servicios) siempre que ambas partes consientan la conexión.

Comunicación P2P sin barreras

Las personas podrán comunicarse, a través de videoconferencias, libremente desde ordenadores o dispositivos móviles con quienes deseen sin restricciones administrativas en los dispositivos.

Flujo libre de información.

Las personas deben tener libertad de acceso a toda la información que otros pongan a su disposición. Las personas también deben disponer de las herramientas prácticas para compartir la información y el acceso a ella resulte más fácil, rápido y eficaz.

Valor social

Las personas deben tener libertad para darse a conocer a través de su identidad y sus conexiones, y no deben ser retiradas del servicio de Webcam App por razones diferentes de las descritas en la Declaración de derechos y responsabilidades de Webcam App.

Servicio fundamental

Las personas deben ser capaces de utilizar Webcam App de forma gratuita para establecer una presencia virtual, conectarse con otros y compartir información con ellos. Toda persona tiene que poder utilizar los servicios de Webcam App, independientemente de su nivel de participación o contribución.

1.6 Organización del documento

El documento contiene 6 capítulos, cada uno está compuesto por un conjunto de apéndices con información específica de los temas tratados.

En el primer capítulo se realiza una introducción del Trabajo de Fin de Grado, los objetivos a alcanzar, así como la motivación por la cual se ha decidido realizar este trabajo.

³ Programas que pueden anexarse a otros para extender sus funcionalidades.

En el segundo capítulo se hace una revisión del estado actual del arte en que se encuentran los sistemas de videoconferencias, además del surgimiento de WebRTC y se mencionan algunas de las aplicaciones de videoconferencias más famosas en Internet.

En el tercer capítulo se hace alusión a los recursos utilizados, tanto de hardware como de software, para desarrollar la aplicación y su posterior despliegue.

El capítulo número cuatro aborda la planificación y temporización del trabajo. Se explica detalladamente la metodología utilizada para realizar el sistema de videoconferencias, así como el presupuesto empleado para desarrollar el mismo.

En el capítulo cinco se detalla el desarrollo de la aplicación, donde se pone en práctica lo planteado por el Proceso Unificado de Desarrollo de Software (PUD) y se obtienen los productos asociados a cada flujo de trabajo.

Para finalizar, el capítulo seis expone las conclusiones obtenidas como resultado del Trabajo de Fin de Grado. Además, el mismo contiene los anexos y bibliografía empleada a lo largo de todo el proyecto, que sirvieron como soporte para la correcta elaboración del mismo.

2 Estado actual del arte

En la actualidad, la comunicación en tiempo real es muy limitada, se necesitan descargar programas, instalarlos y actualizarlos para efectuar la misma, además se realiza de una forma muy compleja. El uso de las tecnologías actuales para la transmisión de audio y video tiene una calidad limitada, además de ser costosas es muy difícil integrarlas con otros servicios web. Para el uso de estas tecnologías, se necesita licencia de software propietario o únicamente utilizando una aplicación desarrollada por uno mismo. Hasta la llegada de WebRTC y HTML5, no había forma de que los usuarios realizaran una comunicación de forma totalmente libre. WebRTC implementa estándares abiertos para la transmisión en tiempo real de video, audio y datos. (2)

2.1 WebRTC

2.1.1 Historia

El proyecto conocido como WebRTC, para la comunicación en tiempo real basada en el navegador, fue hecho Open Source por Google. Esto ha sido continuado por los trabajos en curso para estandarizar los protocolos pertinentes de la IETF⁴ y APIs del navegador en la W3C. (2)

El proyecto WebRTC de la W3C es un trabajo en progreso que cuenta con implementaciones avanzadas en Firefox y Chrome. El API se basa en el trabajo previo realizado en la WHATWG⁵. El grupo de trabajo de WebRTC espera que esta especificación evolucione significativamente con base en:

- ❖ Los resultados de los intercambios en el grupo RTCWEB de la IETF para definir el conjunto de protocolos que permitirán comunicaciones en tiempo real en los navegadores web.
- ❖ Los problemas de privacidad que surgen al exponer las capacidades y los flujos locales.
- ❖ Las discusiones técnicas dentro del grupo, sobre la implementación de canales de datos en particular.
- ❖ La experiencia adquirida a través de la experimentación temprana.
- ❖ Los comentarios recibidos de otros grupos e individuos. (3)

2.1.2 Definición

WebRTC ofrece a los desarrolladores de aplicaciones web la posibilidad de desarrollar excelentes aplicaciones multimedia en tiempo real a través de la web, sin necesidad de plugins, descargas o actualizaciones. Su propósito es ayudar a construir una sólida aplicación de comunicación en tiempo real, que funciona a través de múltiples navegadores web y plataformas. (4)

2.1.3 Diseño

Los principales componentes de WebRTC son:

- ❖ getUserMedia: Permite a un navegador web acceder a la cámara y el micrófono.
- ❖ PeerConnection: Establece las llamadas de audio / vídeo.
- ❖ DataChannels: Permiten a los navegadores compartir datos a través de p2p.

⁴ Grupo de Trabajo de Ingenieros de Internet.

⁵ Grupo de Trabajo de Aplicaciones Web

2.2 Sistemas de videoconferencias más usados a nivel mundial

2.2.1 Skype

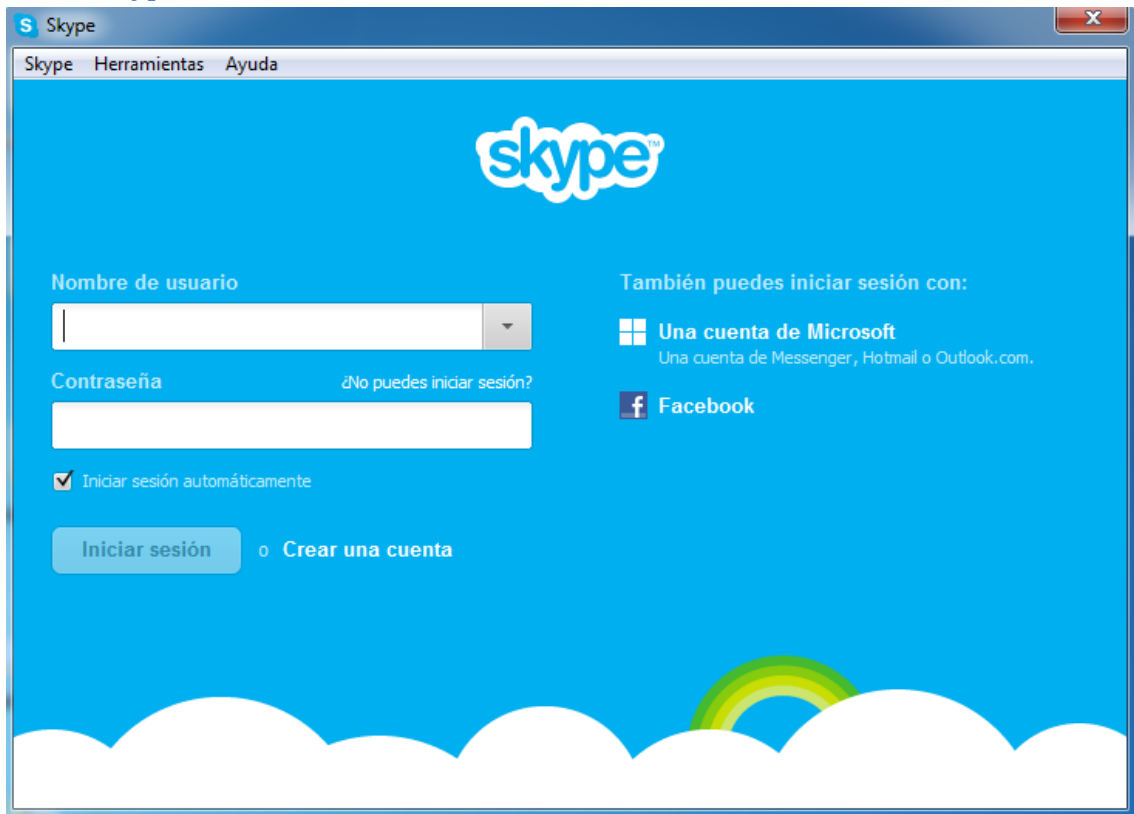


Ilustración 1: Aplicación Skype.

Skype es una aplicación mantenida por Microsoft que permite realizar videollamadas, enviar mensajes y compartir archivos con otras personas que se encuentren en diferentes localizaciones. Es un sistema muy potente, a pesar de brindar grandes beneficios tiene algunas limitaciones, de las cuales se mencionan las más significativas. (5)

Limitaciones:

- ❖ Necesita instalación de un plugin.
Para hacer uso de los servicios que brinda Skype, se necesita descargar, instalar y actualizar el plugin de la aplicación (ver Ilustración 2).

Skype™ 6.6	Skype Technologies S.A.	20/07/2013
Microsoft Office Enterprise 2007	Microsoft Corporation	20/07/2013
Synaptics Pointing Device Driver	Synaptics Incorporated	20/07/2013
Microsoft Security Essentials	Microsoft Corporation	20/07/2013
Complemento Guardar como PDF o XPS de Microsof...	Microsoft Corporation	19/07/2013
Java SE Development Kit 7 Update 25 (64-bit)	Oracle	19/07/2013
Java 7 Update 25 (64-bit)	Oracle	19/07/2013
Google Chrome	Google Inc.	19/07/2013
NetBeans IDE 7.3.1	NetBeans.org	19/07/2013
GlassFish Server Open Source Edition 4.0		19/07/2013

Ilustración 2: Instalación del plugin de Skype.

- ❖ No es totalmente gratuito.

Skype no es gratis al 100%, si es necesario hacer uso de los servicios más interesantes es preciso pagar, lo cual actualmente es una verdadera limitación (ver Ilustración 3).

Todo esto es gratis...

... y esto no lo es


 Tu cuenta de Skype y las descargas más recientes.	 Llamadas a fijos y móviles de cualquier lugar del mundo a tarifas económicas.
 Llamadas y videollamadas a cualquier persona en Skype.	 Envía mensajes SMS a precios económicos.
 Mensajería instantánea y compartir archivos.	 Videollamadas grupales con hasta 10 personas.

Ilustración 3: Servicios de Skype

2.2.2 Google Hangouts



The screenshot shows the Gmail interface with the Hangouts sidebar on the left. The sidebar includes sections for 'Recibidos (95)', 'Destacados', 'Importante', 'Enviados', and 'Borradores (38)'. At the bottom of the sidebar, there are icons for 'Buscar contactos', 'Iniciar videollamada', and 'Dropbox'. The main email list on the right shows several messages, including one from 'InfoJobs.net Ofertas' and another from 'Dropbox'.

Ilustración 4: Acceso al Hangouts de Google a través del Gmail.

A pesar de que WebRTC fue desarrollado por Google, actualmente el servicio que brinda esta compañía para realizar videoconferencias sigue teniendo un punto desfavorable.

Limitaciones:

- ❖ Al igual que Skype, necesita la instalación de un plugin, sin el cual no se puede hacer uso del servicio (ver Ilustración 5).

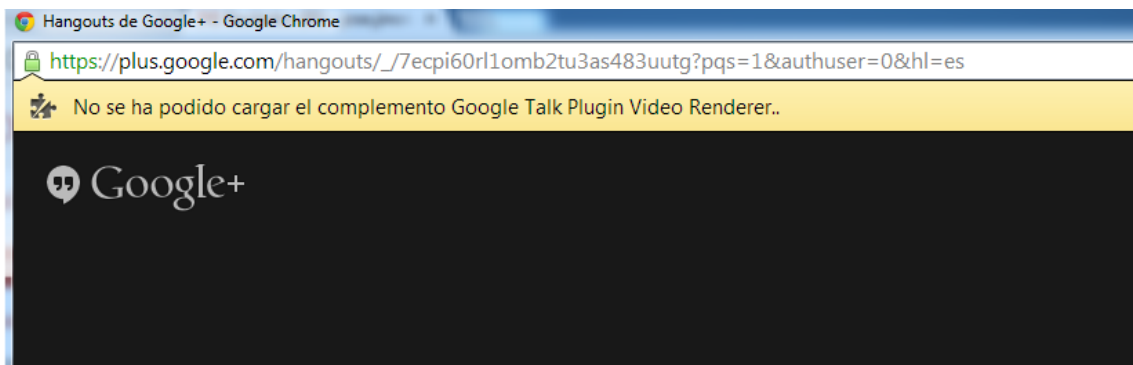


Ilustración 5: Mensaje de error al cargar el plugin de Hangouts.

3 Recursos utilizados

En este capítulo se mencionan los recursos utilizados para el desarrollo de la aplicación y los recursos mínimos para el correcto funcionamiento de la misma una vez que esté desplegada.

3.1 Recursos software

3.1.1 Herramientas

❖ **Google Chrome**

Navegador web desarrollado por Google, diseñado para ser veloz en todos los aspectos. Se inicia rápidamente desde el escritorio, carga las páginas web de forma instantánea y ejecuta aplicaciones web complejas a gran velocidad. (5)

❖ **Elgg**

Marco que proporciona las funcionalidades necesarias para que pueda desarrollarse una red social pública, como Facebook, o interna en una intranet, como Microsoft SharePoint. (6)

❖ **Node.js**

Node.js es un intérprete Javascript del lado del servidor que cambia la noción de cómo debe trabajar un servidor web. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código, para permitir el manejo de decenas de miles de conexiones simultáneas en una sólo una máquina física. (7)

❖ **jQuery**

jQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones. Contribuye a la manipulación del documento HTML, control de eventos, animación y Ajax mucho más simple, con un API fácil de usar que funciona perfectamente en muchos navegadores. Con una combinación de flexibilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben código JavaScript. (8)

❖ **Apache HTTP Server**

Es un servidor seguro, eficiente y extensible que proporciona servicios HTTP en sincronización con los estándares HTTP actuales. (9)

Apache httpd ha sido el servidor web más popular en Internet desde abril de 1996. El proyecto está gestionado por un grupo de voluntarios ubicados en todo el mundo y forma parte de la Fundación de Software Apache. Además, cientos de usuarios han contribuido con ideas, código y a la documentación del proyecto.

❖ **MySQL**

MySQL es un software de base de datos de código abierto, el más popular del mundo, con más de 100 millones de copias de su software descargados o distribuidos a lo largo de su historia. Con su velocidad, fiabilidad y facilidad de uso, MySQL se ha convertido en la opción preferida para la Web, Web 2.0, SaaS, ISV y empresas de telecomunicaciones, ya que elimina los principales problemas asociados con el tiempo de inactividad, el mantenimiento y administración para aplicaciones modernas online. El buque insignia de MySQL, es MySQL Enterprise, un conjunto completo de software, herramientas de monitorización proactivas y servicios de soporte premium disponibles en una suscripción anual asequible.(10)

❖ **NetBeans**

NetBeans es un proyecto de código abierto que provee productos consistentes para el desarrollo de software (NetBeans IDE y NetBeans Platform). Responde a las necesidades de los desarrolladores, los usuarios y las empresas que dependen de él como base para sus productos. El desarrollo de aplicaciones se realiza de forma rápida, eficaz y sencilla, aprovechando los puntos fuertes de la plataforma Java y otras normas relevantes de la industria. (11)

3.1.2 Lenguajes de programación

❖ Javascript

Javascript (a veces abreviado como JS) es un lenguaje ligero, interpretado y orientado a objetos, más conocido como el lenguaje script para páginas web. El estándar de JavaScript es ECMAScript. A partir de 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan al menos ECMAScript 3. Una sexta revisión del estándar está en proceso. (12)

❖ HTML5

El lenguaje de marcado de la World Wide Web ha sido siempre HTML. Fue diseñado principalmente para describir semánticamente documentos científicos, sin embargo, su diseño general y adaptaciones en los últimos años han permitido que sea utilizado para representar otros tipos de documentos. En 2006, el W3C manifestó su interés de participar en el desarrollo de HTML5 y en 2007 formó un grupo de trabajo para desarrollar la especificación HTML5.0.

Apple, Mozilla y Opera permitieron que W3C publicara la especificación bajo el copyright W3C, manteniendo una versión con licencia menos restrictiva en el sitio WHATWG. Esta nueva especificación incorpora interesantes capacidades Javascript, que aumentan la capacidad de almacenamiento frente a las cookies que dejaban recopilar algunos kilobytes. Ahora se puede conseguir un almacenamiento entre 5 y 10 megas, dependiendo de la plataforma. Además, HTML5 permite múltiples Javascripts ejecutándose en paralelo en la misma página, hace posible la inserción de audio y video de forma directa y la geolocalización del usuario. (13)

❖ CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla y la forma en que se va a imprimir. Esta forma de descripción, ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Permite a los desarrolladores controlar el estilo y el formato de múltiples páginas web simultáneamente. Cualquier cambio en el estilo marcado para un elemento en la CSS, afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (14)

❖ PHP

PHP (Hypertext Preprocessor) es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Las páginas web que usan PHP se tratan igual que páginas HTML comunes y corrientes, se pueden crear y editar de la misma manera que normalmente se crean páginas HTML. PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo pueden realizarse funcionalidades incluidas en otros lenguajes de programación, como recopilar datos de formularios, generar páginas con contenidos dinámicos, enviar y recibir cookies; aunque PHP puede hacer mucho más. (15)

3.2 Recursos hardware

El proyecto fue desarrollado desde un ordenador personal marca HP G62 con las siguientes características:

- ❖ Procesador Intel Core i5.
- ❖ Memoria RAM 4 GB DDR2.
- ❖ Pantalla de 15,4 pulgadas.
- ❖ Sistema Operativo: Windows 7 Home Edition.

3.2.1 Hardware de los clientes

- ❖ Ordenador portátil con webcam y micrófono incluido.
- ❖ Ordenador de sobre mesa con cámara web y micrófono incorporado.
- ❖ Tarjeta Wifi o Placa de Red Ethernet compatible.
- ❖ Conexión a internet a través de ADSL o 3G.
- ❖ Dispositivo móvil con Androide 2.4 o versiones posteriores y cámara frontal.

3.2.2 Hardware de los servidores

- ❖ Procesador: Pentium 4 o superior.
- ❖ Memoria RAM: 2 Gb.
- ❖ Disco Duro: 5GB disponibles.
- ❖ Placa de Red: Ethernet compatible.
- ❖ Velocidad de subida y bajada de datos superior a 5mb.
- ❖ Red LAN que soporte TCP/IP (en general, Internet).

4 Planificación del trabajo

En este capítulo se aborda la planificación y temporización del trabajo. Se explica la metodología utilizada para realizar el sistema de videoconferencias, la estimación del tiempo de ejecución de las tareas, así como un aproximado del coste para el desarrollo de la aplicación.

4.1 Metodología de desarrollo

❖ **Proceso Unificado de Desarrollo (PUD)**

Es una metodología de desarrollo de software basada en componentes e interfaces bien definidas, que unida al Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Es un proceso que puede especializarse para una gran variedad de sistemas de software en diferentes áreas de aplicación, distintos tipos de organizaciones, variados niveles de aptitud y tamaños de proyecto.

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Es el resultado de varios años de desarrollo y uso práctico, en el que se han unificado técnicas de desarrollo a través de UML y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0 (RUP), de ahí las siglas con las que se identifica a este proceso de desarrollo. (16)

❖ **Dirigido por Casos de Uso**

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona un resultado de valor para el usuario. Los casos de uso modelan los requerimientos funcionales del sistema. Todos los casos de uso juntos constituyen el modelo de casos de uso.

Los casos de uso también guían el proceso de desarrollo (diseño, implementación y prueba). Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. De este modo, los casos de uso no solo inician el proceso de desarrollo, sino que le proporcionan un hilo conductor que avanza a través de una serie de flujos de trabajo que parten de los casos de uso. (17)

❖ **Centrado en la Arquitectura**

La arquitectura de un sistema software se describe mediante diferentes vistas del sistema en construcción. El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles a un lado.

Los casos de uso y la arquitectura están profundamente relacionados. Los casos de uso deben encajar en la arquitectura y a su vez la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro (17).

❖ **Iterativo e Incremental**

Es práctico dividir el esfuerzo de desarrollo de un proyecto de software en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos a crecimientos en el producto. Las iteraciones deben estar controladas, esto significa que deben seleccionarse y ejecutarse de una forma planificada. Los desarrolladores basan la selección de

lo que implementarán en cada iteración en dos cosas: el conjunto de casos de uso que amplían la funcionalidad y los riesgos más importantes que deben mitigarse.

En cada iteración los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía para implementar dichos casos de uso. Si la iteración cumple sus objetivos, se continúa con la próxima, si no deben revisarse las decisiones previas y probar un nuevo enfoque. (17)

❖ **Ciclo de Vida**

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema.

❖ **Fases**

Cada ciclo constas de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide en iteraciones. En cada iteración se desarrolla en secuencia un conjunto de disciplinas o flujos de trabajos.

❖ **Disciplinas**

Cada disciplina es un conjunto de actividades relacionadas (flujos de trabajo) vinculadas a un área específica dentro del proyecto total. Las más importantes son: Requerimientos, Análisis, Diseño, Codificación y Prueba. El agrupamiento de actividades en disciplinas es principalmente una ayuda para comprender el proyecto desde la visión tradicional en cascada (17)(ver Ilustración 6).



Ilustración 6: Fases, iteraciones y disciplinas.

Cada disciplina está asociada con un conjunto de modelos que se desarrollan. Estos modelos están compuestos por artefactos. Los artefactos más importantes son los modelos que cada disciplina realiza: Modelo de Casos de Uso, Modelo de Diseño, Modelo de Implementación y Modelo de Prueba (17) (ver Ilustración 7).

Las disciplinas producen modelos

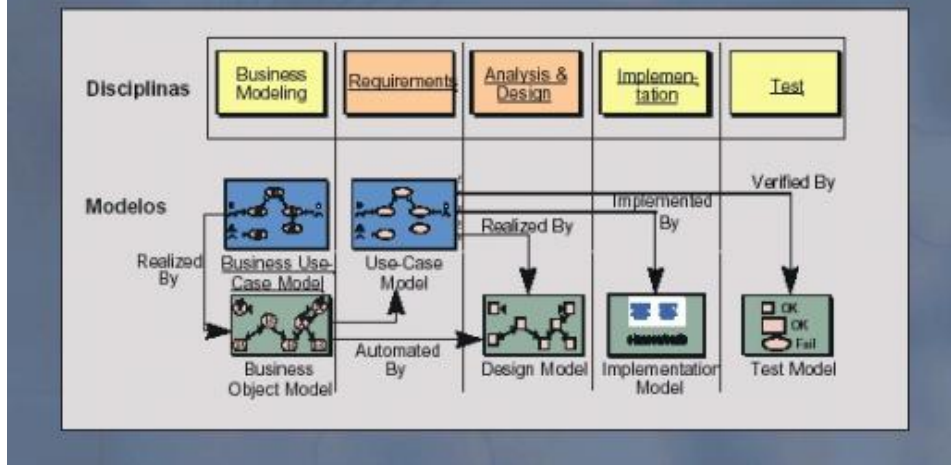


Ilustración 7: Modelos por disciplinas.

4.2 Planificación y temporización

En este apartado se detallan las horas empleadas para la realización del trabajo, siguiendo las cuatro fases que plantea el Proceso Unificado de Desarrollo de Software, así como las horas específicas para cada flujo de trabajo en cada fase (ver Tabla 1).

FASES	HORAS
INICIO	60H
Requerimientos	10H
Análisis y Diseño	50H
Codificación	0H
Pruebas	0H
ELABORACIÓN	85H
Requerimientos	10H
Análisis y Diseño	45H
Codificación	30H
Pruebas	5H
CONSTRUCCIÓN	140H
Requerimientos	0H
Análisis y Diseño	25H
Codificación	105H

Pruebas	10H
TRANSICIÓN	45H
Requerimientos	5H
Análisis y Diseño	10H
Codificación	25H
Pruebas	5H
Total	335H

Tabla 1: Planificación por fases y flujos.

4.3 Presupuesto

A continuación se detalla el presupuesto para efectuar el proyecto. Se destacan los elementos necesarios para la realización del mismo, así como el coste estimado para cada uno de ellos.

- ❖ **Coste del hardware**
Es el coste correspondiente a los equipos utilizados.
- ❖ **Coste del software**
Gastos derivados de la utilización de las herramientas empleadas para la realización del proyecto.
- ❖ **Coste del personal**
Gastos correspondientes a la remuneración de las personas que desarrollan el proyecto.

4.3.1 Coste del hardware

Para este proyecto se tienen en cuenta los siguientes elementos de hardware para su desarrollo (ver Tabla 2).

Elemento	Coste
Ordenador Portátil HP G62	450€
Servicio de Internet ADSL	160€
Servidores de hosting	Gratuito
Total Hardware	610€

Tabla 2: Desglose de los costes de hardware.

4.3.2 Coste del software

El coste de los siguientes elementos es el correspondiente a la licencia de los mismos (ver Tabla 3).

Elemento	Coste
Windows 7 Professional	160€
Microsoft Office 2007	130€
Total Software	290€

Tabla 3: Desglose de los costes de software.

4.3.3 Coste del personal

En la siguiente tabla se enumeran los trabajadores que intervienen en el desarrollo del proyecto, así como el número de horas estimadas para cada uno de ellos, el coste por hora y el coste total (ver Tabla 4).

Personal	Nº Horas	Coste / H	Coste
Jefe de proyecto	20H	40€	800€
Arquitecto	65H	35€	2275€
Analista	50H	20€	1000€
Programador	150H	20€	3000€
Ingeniero de pruebas	15H	20€	300€
Total	300H	-	7375€

Tabla 4: Desglose de los costes del personal.

4.3.4 Presupuesto total del proyecto

La siguiente tabla muestra un desglose de los costes calculados anteriormente y el coste total que representa el desarrollo del proyecto (ver Tabla 5).

Elemento	Coste
Coste del hardware	610€
Coste del software	290€
Coste del personal	7375€
Total	8275€

Tabla 5: Desglose de los costes.

5 Desarrollo del trabajo

5.1 Modelo del dominio

Un modelo del dominio se utiliza como fuente para el diseño de los objetos software y corresponde a una entrada necesaria para varios artefactos. El modelo del dominio muestra (a los modeladores) clases conceptuales significativas en un dominio del problema, es el artefacto más importante que se crea durante el análisis orientado a objetos. Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases u objetos software con responsabilidades. (18)

❖ Webcam App

Webcam App es una aplicación que tiene como principal objetivo social que las personas puedan realizar videoconferencias a través de la web de una forma gratuita y sencilla. Cuando un usuario se registra en Webcam App facilita su nombre, correo electrónico, entre otros datos. También, el usuario puede añadir una foto de perfil para que sus amigos puedan identificarlo fácilmente entre los miembros de la aplicación. Una vez registrado, el usuario puede proporcionar otra información relacionada, por ejemplo su ciudad de residencia, redes, actividades, intereses y lugares. También puede buscar a sus amigos sólo con introducir su nombre y agregarlos como amigos en Webcam App, con el fin de poder comunicarse y compartir sus datos con ellos. Otras de las finalidades principales del uso de Webcam App es compartir contenido con los demás, por ejemplo cargar o hacer una foto, cargar o grabar un video, compartir un enlace, crear un grupo, hacer un comentario, escribir una nota o enviar un mensaje. Parte del contenido que se comparte y de las acciones que se llevan a cabo se muestran en la página de actividades. Además, se puede limitar las personas que pueden ver los datos personales desde la configuración de privacidad (ver Ilustración 8).

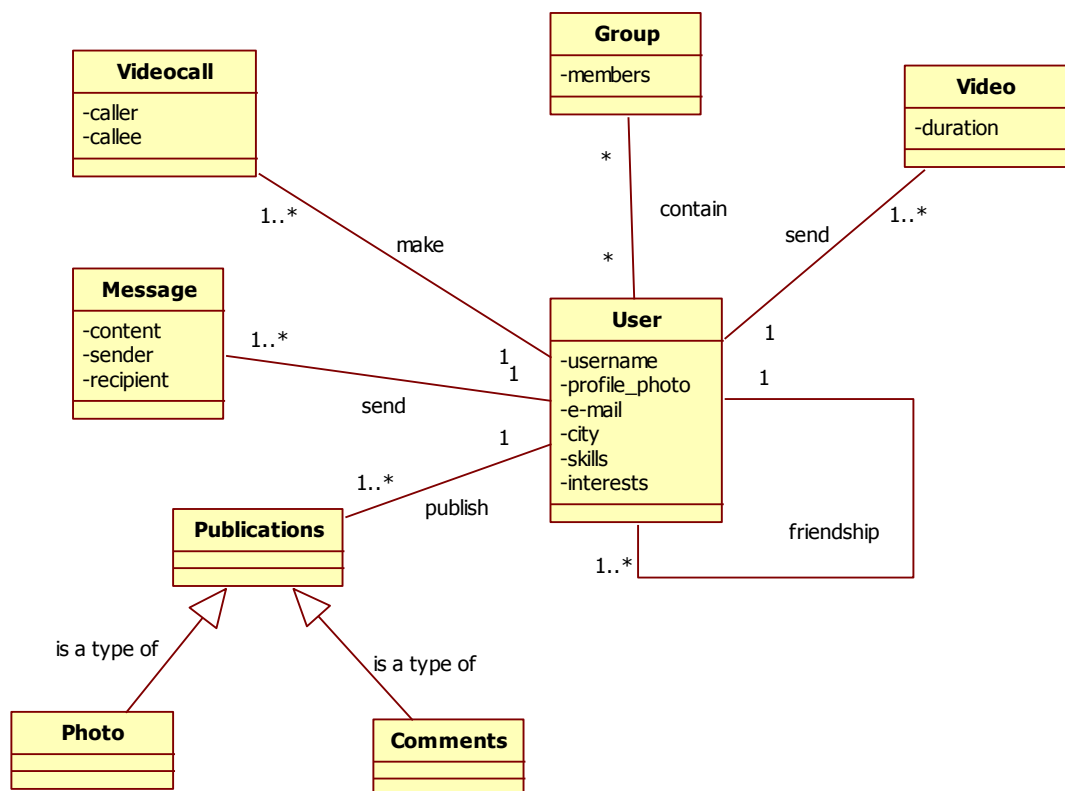


Ilustración 8: Modelo del dominio.

5.2 Lista de características

En esta sección se detalla la lista de características asociada a la aplicación Webcam App. La misma es un artefacto que se obtiene después de aplicar la tarea de “Enumerar los requisitos candidatos”, que se propone en la metodología del PUD (Proceso Unificado de Desarrollo), para la captura de requisitos.

Esta lista contiene las ideas de clientes, usuarios, analistas y desarrolladores sobre posibles aspectos a incluir en la aplicación, que posteriormente se pueden traducir en requisitos del software. Estas ideas se consideran requisitos candidatos a desarrollar en la versión actual del sistema o se pueden postergar a versiones futuras.

❖ Tipos de Usuarios

La clasificación de los usuarios es una primera aproximación para identificar y asociar a cada uno de los posibles usuarios, los tipos identificados no necesariamente tienen que ser los usuarios definitivos.

- ❖ **Anónimo:** Es el usuario que no está registrado y va a utilizar las funcionalidades básicas de navegación, como por ejemplo registrarse, sin opción a modificar el entorno.
- ❖ **Registrado:** Es el usuario más común, además de las funciones del usuario anterior puede acceder a la mayoría de las funcionalidades de la aplicación, añadir amigo, enviar mensajes, hacer videollamadas, eliminar amigo, agregar contenido, gestionar perfil, configurar privacidad, etc.
- ❖ **Administrador:** Este usuario puede acceder a todas las funcionalidades de la aplicación, entre las suyas propias están: bloquear cuentas de usuarios, eliminar cuentas de usuario, hacer a un usuario administrador, restablecer el password a los usuarios.

Para describir las características, se utiliza una tabla con los siguientes campos (ver Tabla6):

- ❖ **Código:** Es el identificador de la característica. Se especifica como LC + identificador del usuario + número de la característica.
 - ❖ Identificadores para cada tipo de usuario de la aplicación
 - ❖ R. Usuario Registrado.
 - ❖ A. Usuario Anónimo.
 - ❖ AD. Usuario Administrador.
- ❖ **Nombre:** Nombre de la característica.
- ❖ **Descripción:** Breve descripción de lo que comprende la característica.
- ❖ **Prioridad:** Se asigna una prioridad a cada característica con el fin de determinar el orden en que se van a ir desarrollando. Las prioridades que se usan están definidas por un valor numérico, que representa el nivel de prioridad, donde 0 es la más baja y 100 la más alta, luego están los rangos que se muestran a continuación:
 - ❖ 80-100 (Muy alta)
 - ❖ 60-80 (Alta)
 - ❖ 40-60 (Media)
 - ❖ 20-40 (Baja)
 - ❖ 0-20 (Muy baja)
- ❖ **Estado:** Cada característica tiene un estado asociado que va variando a medida que progresa el sistema. Los posibles estados son:
 - ❖ Aceptado: La característica se desarrollará en esta versión del producto.
 - ❖ Planificada: La característica ya ha sido planificada y se empezará a desarrollar en un plazo de tiempo corto.
 - ❖ En desarrollo: Ya se está desarrollando.
 - ❖ Finalizada: Se ha terminado de desarrollar.

- ❖ Postergada: No se desarrollará hasta una versión futura.
- ❖ Rechazada: Probablemente no se desarrollará en ninguna versión.

Código	Nombre	Descripción	Prioridad	Estado
LC-A.1	Registrar Usuario	La aplicación permitirá al usuario introducir sus datos personales para registrarse.	99	Finalizada
LC-R.1	Buscar Amigo	La aplicación permitirá a cualquier usuario introducir el nombre de una persona y mostrará los resultados de la búsqueda.	97	Finalizada
LC-R.2	Crear Perfil	Un usuario que se ha registrado puede introducir otros datos personales específicos que pueden variar como: (dirección de correo electrónico, localidad, etc.)	96	Finalizada
LC-R.3	Gestionar Perfil	Un usuario registrado puede modificar los datos personales introducidos en la información del perfil.	96	Finalizada
LC-R.4	Hacer Login	Se permitirá al usuario ya registrado introducir su dirección de correo o nombre de usuario y contraseña para identificarse en la aplicación.	99	Finalizada
LC-R.5	Añadir Amigo	Se permitirá al usuario registrado agregar una persona que esté registrada como amigo.	90	Finalizada
LC-R.6	Eliminar Amigo	Se permitirá al usuario registrado eliminar una persona de su lista de amigos.	90	Finalizada
LC-R.7	Enviar Mensajes	Un usuario registrado podrá enviar mensajes a cualquier persona que está registrada en la aplicación.	85	Finalizada
LC-R.8	Reportar	Un usuario registrado podrá reportar cualquier	75	Finalizada

	Usuario	anomalía que detecte en las cuentas de los otros usuarios.		
LC-R.9	Configurar Privacidad	Se permitirá al usuario configurar la privacidad de los datos personales de su cuenta.	78	Finalizada
LC-R.10	Gestionar Cuenta	Se permitirá al usuario gestionar los datos personales de su cuenta.	80	Postergada
LC-R.11	Crear Grupo	Se permitirá al usuario crear grupos, a los que posteriormente se le pueden incluir personas que estén registradas.	65	Finalizada
LC-R.12	Gestionar Grupo	Se permitirá al usuario que es propietario de un grupo eliminar miembros del grupo, añadir miembros al grupo y cambiar el nombre del grupo.	65	Finalizada
LC-R.13	Realizar Videollamadas	Se permitirá al usuario registrado la realización de videollamadas a los usuarios que estén en su listado de amigos.	100	Finalizada
LC-R.14	Compartir Contenido	Se permitirá al usuario registrado compartir contenidos con sus amigos o públicamente.	72	Finalizada
LC-R.15	Enviar video	En el caso de que algún amigo no se encuentre en línea y se desee enviar un video, el usuario puede grabarse un video desde la webcam y enviarlo al buzón de videos de su amigo.	60	Postergada
LC-R.15	Realizar Llamada Telefónica	Los usuarios registrados en Webcam App, podrán hacer llamadas telefónicas desde la aplicación.	53	Postergada
LC-R.16	Captura de	Los usuarios registrados podrán tomarse fotos	58	Postergada

	imágenes	desde la webcam aplicándole efectos tipo instagram y descargarlas en su ordenador.		
LC-R.17	Publicar Fotos	El usuario luego de capturar sus fotos desde la webcam, puede decidir compartirlas en otras redes sociales como Facebook, Tweeter o Instagram.	45	Postergada
LC-R.18	Publicar Video	El usuario puede capturar un fragmento de video y compartirlo en Facebook.	22	Rechazada
LC-R.19	Descargar video	El usuario puede capturar fragmentos de video y descargarlos en su ordenador.	60	Postergada
LC-R.20	Registrarse con redes sociales	El usuario puede elegir si desea hacer el registro usando los datos de otras redes sociales.	50	Postergada
LC-R.21	Gestionar Widget	Se le permitirá al usuario registrado insertar, eliminar o modificar los widgets que aparecen en su perfil.	70	Finalizada
LC-R.22	Eliminar Contenido	Se permitirá al usuario registrado que pueda eliminar el contenido que haya publicado.	72	Finalizada
LC-AD.1	Bloquear Cuenta de Usuario	El administrador puede suspender durante un período de tiempo una cuenta de usuario por violaciones de las normas.	99	Finalizada
LC-AD.2	Eliminar Cuenta de Usuario	El administrador puede eliminar una cuenta de usuario permanente debido a reiteradas violaciones de las normas.	99	Finalizada
LC-AD.3	Atención al usuario	En caso de que algún usuario presente algún problema online, podrá ponerse en contacto con	40	Postergada

		algún administrador a través del sistema de videoconferencia.		
--	--	---	--	--

Tabla 6: Lista de características.

5.3 Requisitos del software

5.3.1 Actores

❖ Actores del sistema

Existen tres clasificaciones de actores para los casos de uso de la aplicación, los cuales corresponden a los tipos de usuarios que tendría el sistema, ya que cada uno por separado tiene acceso a las distintas funcionalidades (ver Ilustración 9).

❖ Paquete de actores.

- ❖ Usuario Anónimo.
- ❖ Usuario Registrado.
- ❖ Usuario Administrador.

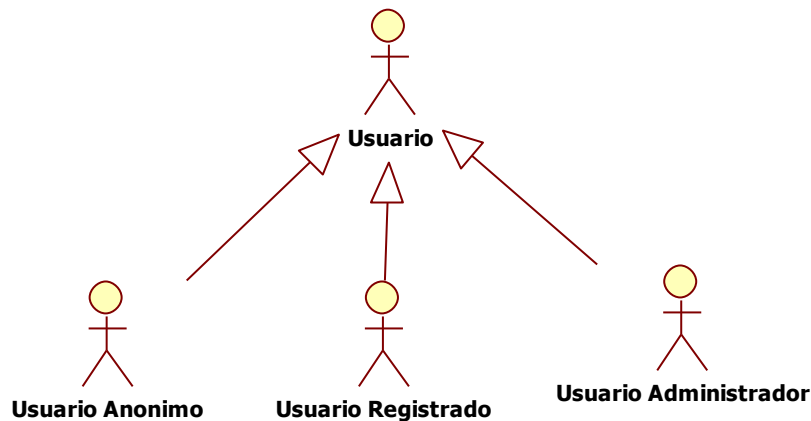


Ilustración 9: Paquete de actores del sistema

5.3.2 Modelo de casos de uso

❖ Identificación de los casos de uso

Caso de Uso	Prioridad
Registrarse	99
Login	99
Buscar Amigo	97
Crear Perfil	96
Gestionar Perfil	96
Gestionar Cuenta	80

Añadir Amigo	90
Eliminar Amigo	90
Reportar Usuario	75
Enviar Mensajes	86
Crear grupo	65
Gestionar Grupo	65
Eliminar Cuenta Usuario	99
Bloquear Cuenta Usuario	99
Realizar Videollamadas	100
Compartir Contenido	72
Eliminar Contenido	72
Gestionar Widgets	70

Tabla 7: Lista de casos de uso identificados.

❖ **Diagramas de casos de uso para el Usuario Anónimo**

En este diagrama se encuentran las funcionalidades básicas que presenta la aplicación en el momento de entrar por primera vez a la página de Webcam App. Para iniciar estos casos de uso no se requiere estar registrado, cualquier usuario puede utilizar estas funcionalidades (ver Ilustración 10).

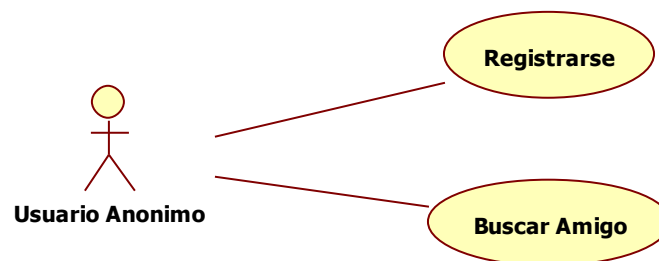


Ilustración 10: Diagrama de casos de uso para Usuario Anónimo.

❖ **Diagramas de casos de uso para el Usuario Registrado**

El usuario registrado puede hacer **Login** en cualquier momento, se validarán los datos de entrada y podrá acceder a su perfil (ver Ilustración 11).



Ilustración 11: Diagrama de casos de uso Login para Usuario Registrado.

En este diagrama se detallan las funcionalidades que puede realizar un usuario registrado con otras personas, al igual que el usuario anónimo puede acceder a la funcionalidad **Buscar Amigo** de la misma manera pero con la diferencia de si ya está identificado en el sistema se permite añadirlo como amigo (ver Ilustración 12).

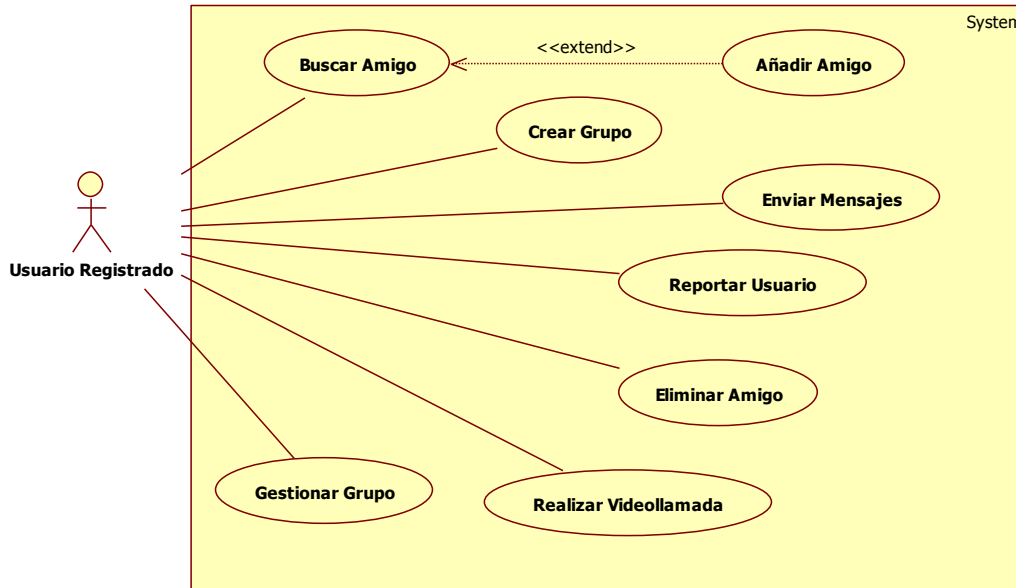


Ilustración 12: Diagrama de casos de uso para la relación de Usuario Registrado con otros usuarios.

Webcam App además de ser una aplicación para estar cerca de los amigos, permite gestionar contenidos o bien compartirlos con ellos (ver Ilustración 13).

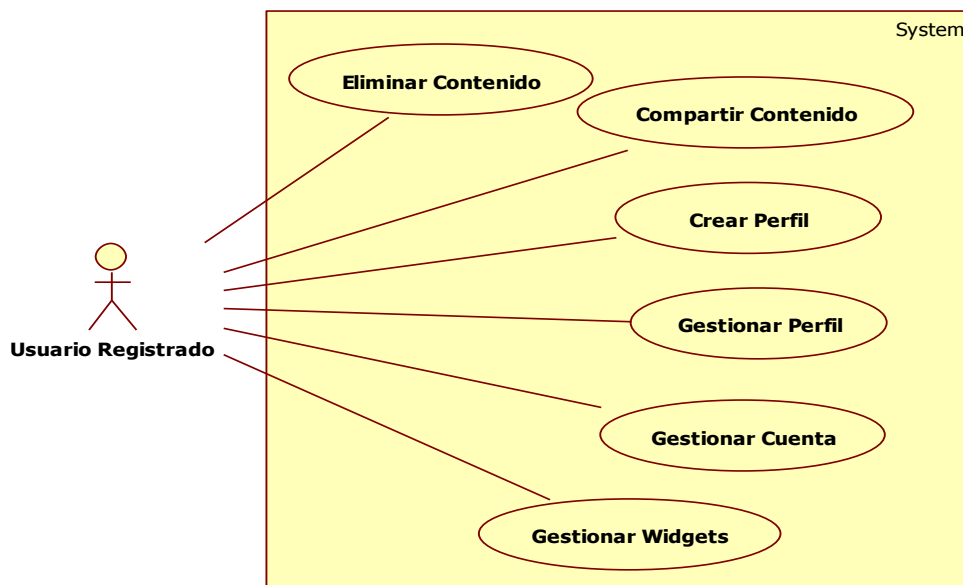


Ilustración 13: Diagrama de casos de uso para la gestión de contenidos.

❖ **Diagramas de casos de uso para el Usuario Administrador**

El usuario administrador tiene un control total sobre las cuentas de usuario (ver Ilustración 14).

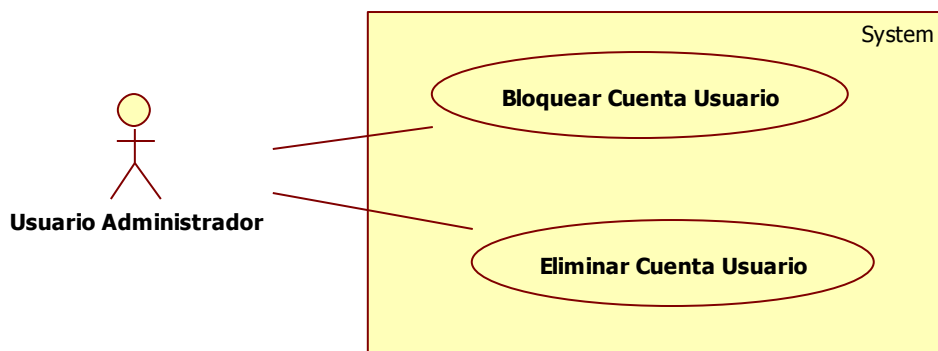


Ilustración 14: Diagrama de casos de uso para la gestión de usuarios.

5.3.3 Especificación de casos de uso

Descripción de los casos de uso

Caso de Uso	
Registrarse	
Actor	Usuario Anónimo
Precondición	1. La cuenta de correo electrónico que se usa para el registro no debe haberse utilizado antes para crearse una cuenta en Webcam App.
Post condición	1. Quedará registrado el usuario en la aplicación.
Flujo normal	1. La aplicación muestra un formulario en el cual el usuario puede incluir su información básica. 2. El sistema valida la información y la almacena en el servidor.
Flujo alternativo	2.1. Para el caso de la información relacionada con la contraseña, si la misma no cumple con los parámetros de seguridad, el sistema envía una notificación al usuario para que introduzca otra. Para el caso de la dirección de correo electrónico, el sistema envía una notificación al usuario para que introduzca una dirección válida, si la misma no cumple con las características que debe tener una dirección de correo.

Caso de Uso	
Buscar Amigo	
Actor	Usuario Registrado
Precondición	1. El usuario debe estar identificado en el sistema. 2. El usuario debe introducir el nombre de la persona que desea buscar.
Post condición	1. El sistema mostrará un listado de todos los contactos que coincidan

	con el nombre introducido.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario introduce el nombre de la persona que desea encontrar. 2. La aplicación muestra un listado de los contactos que están dados de alta en Webcam App con ese nombre. 3. Webcam App ofrece la posibilidad de agregarlos como "Amigos".
Flujo alternativo	3.1. El usuario puede omitir la opción de enviar una invitación de amistad a los usuarios encontrados.

Caso de Uso	
Eliminar Cuenta de Usuario	
Actor	Usuario Administrador
Precondición	1. Tiene que existir una o varias denuncias previas en contra del usuario y que se haya comprobado por los administradores de Webcam App que infringe las normas de uso de la aplicación.
Post condición	<ol style="list-style-type: none"> 1. Quedará eliminada permanentemente la cuenta de usuario. 2. Se elimina el usuario del listado de amigos y los grupos a los que la cuenta estaba asociada.
Flujo normal	<ol style="list-style-type: none"> 1. Si existen varias denuncias sobre un mismo perfil de usuario de la aplicación, el administrador puede decidir si el mismo es un posible usuario a eliminar, señalizando el nombre del usuario y la dirección de su perfil. 2. El administrador revisa los motivos de la denuncia. 3. El administrador pasa a eliminar la cuenta de usuario permanentemente.
Flujo alternativo	3.1. El administrador considera que las causas de las denuncias no proceden para eliminar la cuenta de usuario y de esta forma puede retirar la alerta generada.

Caso de Uso	
Compartir Contenido	
Actor	Usuario Registrado
Precondición	1. El usuario debe estar identificado en el sistema.
Post condición	<ol style="list-style-type: none"> 1. El contenido queda almacenado y asociado al perfil del usuario. 2. El perfil del usuario queda actualizado.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra el formulario asociado para subir el contenido. 2. El usuario introduce el contenido, así como un título o una descripción si así lo desea, además de elegir el nivel de seguridad que desea para su contenido.

	<ol style="list-style-type: none"> 3. El usuario decide las personas con las que desea compartir el contenido. 4. El contenido queda publicado en la sección de archivos. 5. El sistema valida y reconoce el tipo de contenido. La información se almacena en el servidor.
Flujo alternativo	5.1. El sistema detecta que la cuota límite del usuario está completa y rechaza la petición de publicar el contenido.

Caso de Uso	
Eliminar Contenido	
Actor	Usuario Registrado
Precondición	1. El usuario debe estar identificado en el sistema.
Post condición	<ol style="list-style-type: none"> 1. El contenido queda eliminado y desvinculado al perfil del usuario. 2. El perfil del usuario queda actualizado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el contenido a eliminar. 2. El sistema verifica que el contenido se quiere eliminar realmente. 3. El contenido se elimina del servidor.
Flujo alternativo	<ol style="list-style-type: none"> 2.1. El usuario no confirma la eliminación del contenido. 3.1. El contenido se mantiene alojado en el servidor.

Caso de Uso	
Gestionar Perfil	
Actor	Usuario Registrado
Precondición	1. El usuario debe estar identificado en el sistema
Post condición	<ol style="list-style-type: none"> 1. Los datos del perfil del usuario de Webcam App quedan modificados. 2. El perfil del usuario de Webcam App queda actualizado.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra toda la información relacionada con el perfil del usuario. 2. El usuario selecciona la opción de modificar los datos personales. 3. El sistema muestra un formulario asociado a la información de perfil que se desea editar. 4. El usuario edita los campos de la información que desea modificar. 5. El sistema valida que la nueva información es correcta. 6. El sistema almacena la nueva información en el servidor.
Flujo alternativo	5.1 En el caso que algún campo obligatorio no sea rellenado, el sistema mostrará un error y mantendrá el campo con el valor

	<p>anterior.</p> <p>6.1 El sistema no actualizará la información de dicho campo en el servidor.</p>
--	---

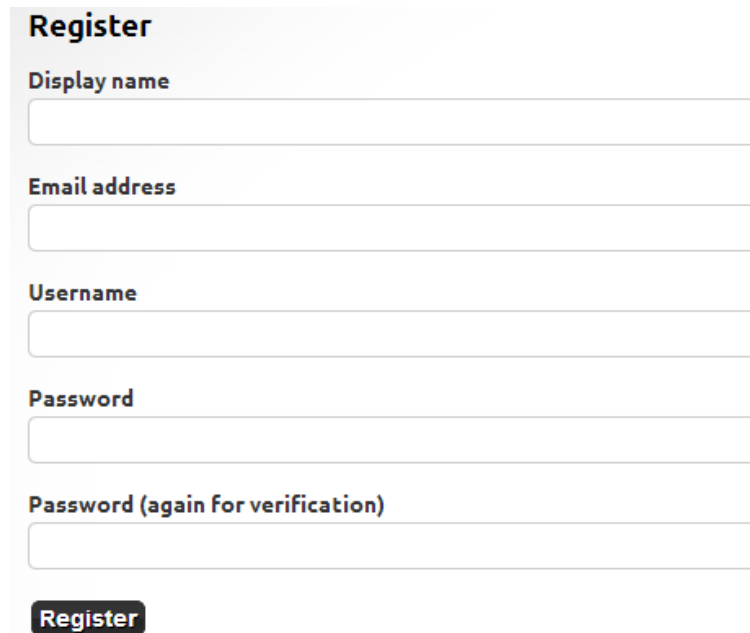
Caso de Uso	
Realizar Videollamada	
Actor	Usuario Registrado
Precondición	1. El usuario debe estar identificado en el sistema.
Post condición	1. La videollamada quedará realizada.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el amigo al cual desea hacerle una videollamada. 2. El usuario envía la solicitud a su amigo para ambos realizar una videollamada. 3. El sistema muestra un cuadro de diálogo para que el usuario que esté siendo llamado acepte la videollamada. 4. El usuario acepta la solicitud de videollamada. 5. La videollamada queda establecida entre los dos usuarios.
Flujo alternativo	<ol style="list-style-type: none"> 4.1. El usuario amigo rechaza la solicitud de videollamada que le están realizando. 5.1. La solicitud queda eliminada.

Caso de Uso	
Login	
Actor	Usuario Registrado
Precondición	1. El usuario debe haberse registrado con anterioridad.
Post condición	1. El usuario quedará identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra los campos para hacer el Login en la aplicación. 2. El usuario rellena los campos con los datos que proporcionó en el registro. 3. El sistema valida los datos y muestra la página del perfil.
Flujo alternativo	3.1. La aplicación pedirá al usuario que intente introducir sus datos nuevamente si no fueron introducidos correctamente.

5.3.4 Prototipos de interfaz de usuario

❖ Caso de Uso Registrarse

Para **Registrarse** el usuario debe introducir información básica con los campos que aparecen a continuación, luego podrá ampliar la información si lo desea (ver Ilustración 15).



Register

Display name

Email address

Username

Password

Password (again for verification)

Register

Ilustración 15: Interfaz para registrarse.

❖ Caso de Uso Buscar Amigo

Para el caso de uso **Buscar Amigo** el usuario podrá localizar a sus amigos registrados en la aplicación a través de su nombre (ver Ilustración 16).

Search members by name

Search

Ilustración 16: Interfaz de búsqueda de amigo.

Una vez introducido el nombre del amigo que se desea encontrar, la aplicación mostrará un listado con las personas las cuales su nombre coincide con el criterio de búsqueda, además la aplicación dará la opción de agregarlo como amigo en Webcam App (ver Ilustración 17).

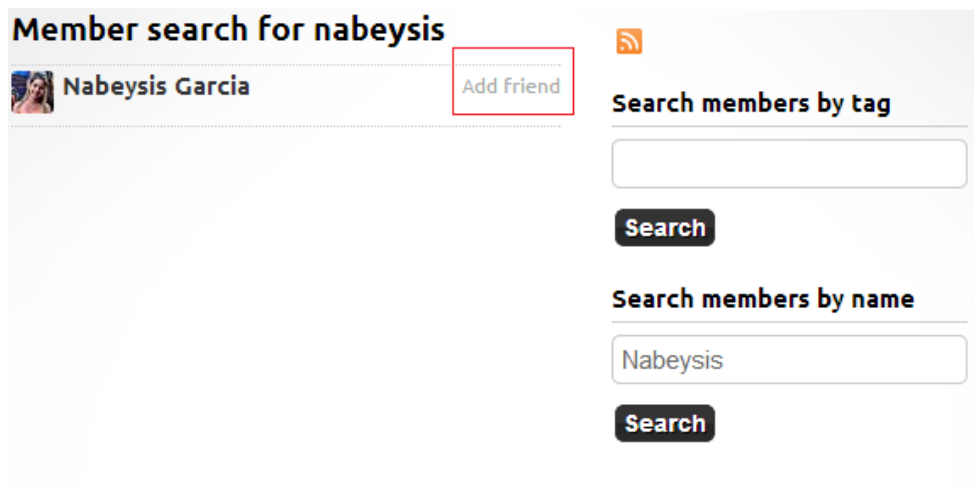


Ilustración 17: Interfaz para agregar un amigo.

❖ **Caso de uso Eliminar Cuenta de Usuario**

Cuando el administrador recibe una denuncia sobre el perfil de algún usuario, revisa los motivos del reporte. Si verdaderamente el usuario denunciado incumple las normas de uso del sitio, el administrador puede eliminar permanentemente su cuenta, de otro modo si es un reporte incorrecto, puede eliminar el reporte o simplemente archivarlo (ver Ilustración 18 y 19).

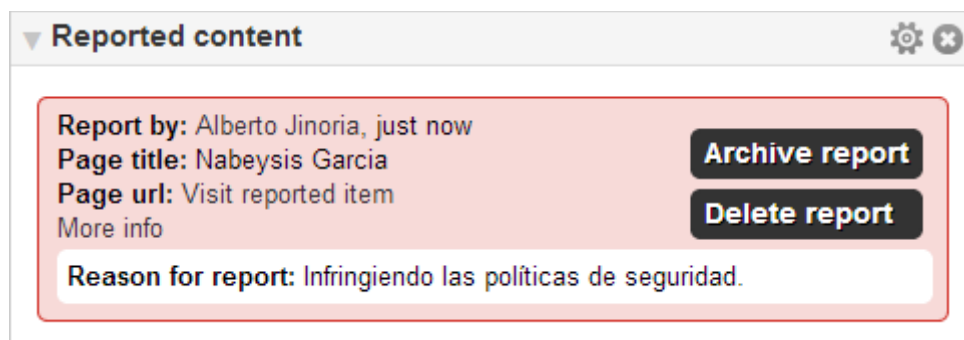


Ilustración 18: Interfaz de reporte de infracción.

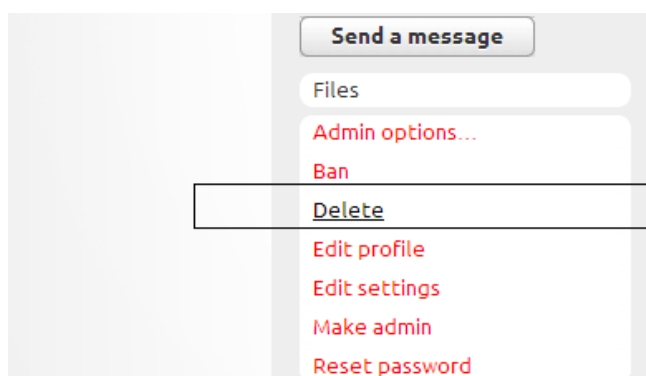


Ilustración 19: Interfaz de eliminar usuario.

❖ **Caso de Uso Agregar Contenido**

Cuando el usuario decide compartir algún tipo de contenido, se muestra el siguiente formulario (ver Ilustración 20).

Upload a file

File

Seleccionar archivo aplicaciones.png

Title

Aplicaciones de Google

Description

Son muy interesantes las aplicaciones de google

Ilustración 20: Interfaz para agregar contenido.

Una vez seleccionado el contenido a publicar, queda almacenado y asociado al perfil del usuario que lo agregó (ver Ilustración 21).






Aplicaciones de Google Download this

 *By Alberto Jinoria just now* Public Edit X Like

Son muy interesantes las aplicaciones de google

+José Alberto Gmail Imágenes 🔔 + 👤



  
+José Alberto Búsqueda YouTube
  
Maps Play Noticias
  
Gmail Drive Calendar

Más

Google es también en: català galego euskara

Ilustración 21: Interfaz del contenido ya agregado.

❖ Caso de Uso Eliminar Contenido

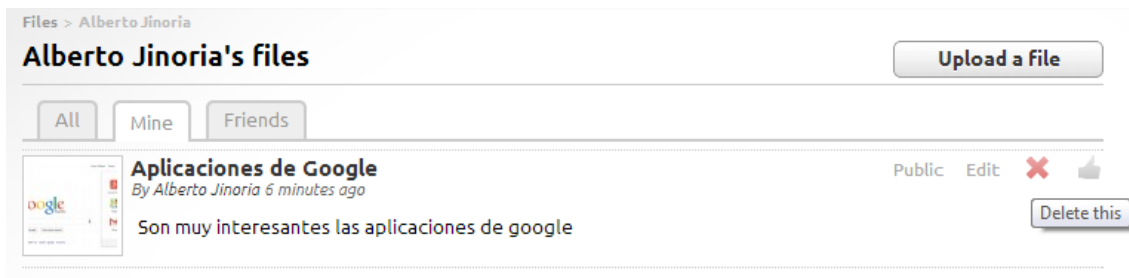


Ilustración 22: Interfaz para eliminar contenido.

Cuando el usuario decide eliminar el contenido el sistema muestra un mensaje de confirmación (ver Ilustración 23 y 24).

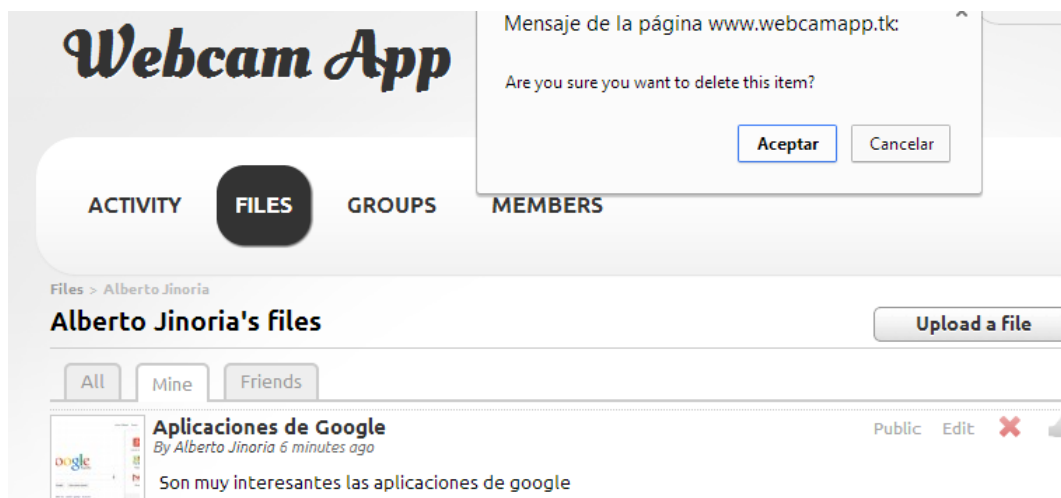


Ilustración 23: Interfaz de mensaje de confirmación de eliminación de contenido.

Una vez confirmado, el contenido queda eliminado permanentemente.

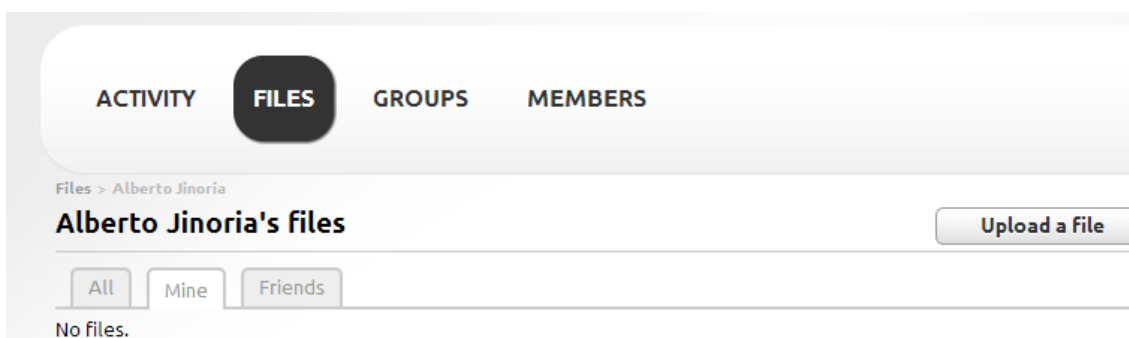


Ilustración 24: Interfaz que muestra el contenido ya eliminado.

❖ Caso de Uso Gestionar Perfil

Cuando el usuario decide gestionar su perfil, se muestra un formulario con todos los campos que tiene acceso a modificar, tales como su Nombre, Localidad, Número de Teléfono, etc. (ver Ilustración 25 y 26).



Ilustración 25: Interfaz del perfil del usuario creado.

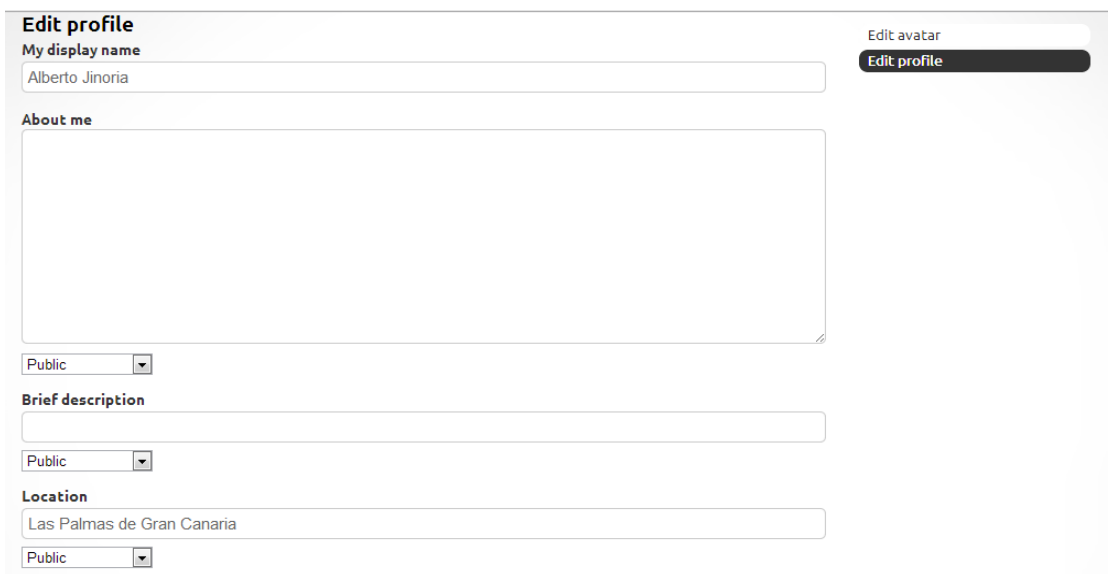


Ilustración 26: Interfaz para editar el perfil del usuario.

❖ **Caso de Uso Login**

Al visitar el sitio, el sistema muestra un formulario para que el usuario pueda identificarse en el mismo y de esta forma acceder a las principales funcionalidades de la aplicación (ver Ilustración 27).

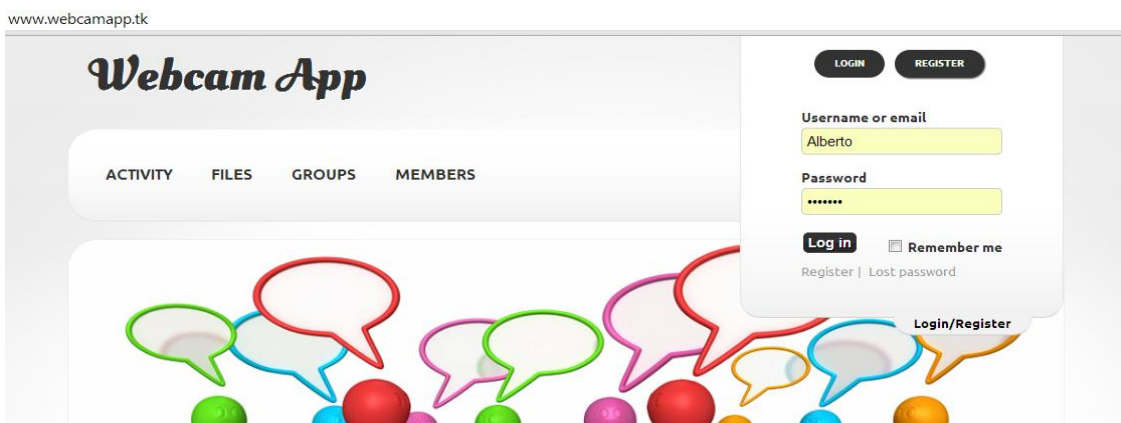


Ilustración 27: Interfaz de formulario de Login.

❖ **Caso de Uso Realizar Videollamada.**

Para el caso de uso realizar videollamada, se han definido dos prototipos de interfaz para realizar la misma acción, el primer prototipo sería desde el perfil del usuario que desea realizar la videollamada, desplegar el menú de opciones que tiene para cada amigo de su listado (ver Ilustración 28).

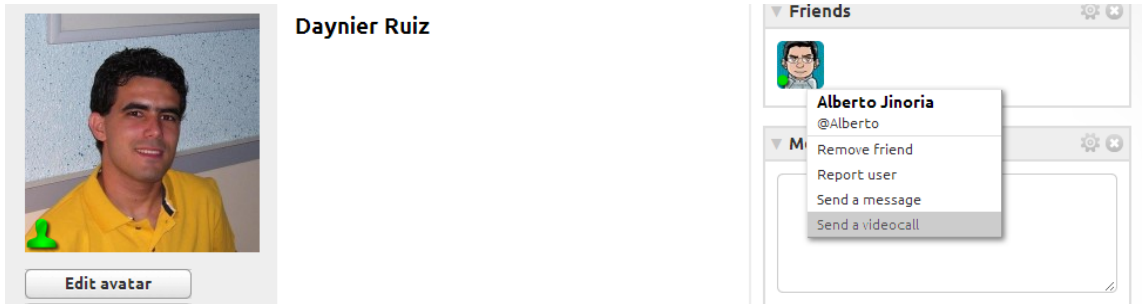


Ilustración 28: Interfaz para realizar videollamada.

El segundo prototipo sería visitando el perfil del usuario que se le desea realizar la videollamada y seleccionar esta opción en el menú que aparece debajo de su foto de perfil (ver Ilustración 29).



Ilustración 29: Interfaz para realizar videollamada.

Posteriormente de haber enviado la solicitud al usuario con el que se desea comunicar, en su aplicación aparecerá un cuadro de diálogo para que confirme o rechace la solicitud de videollamada (ver Ilustración 30).

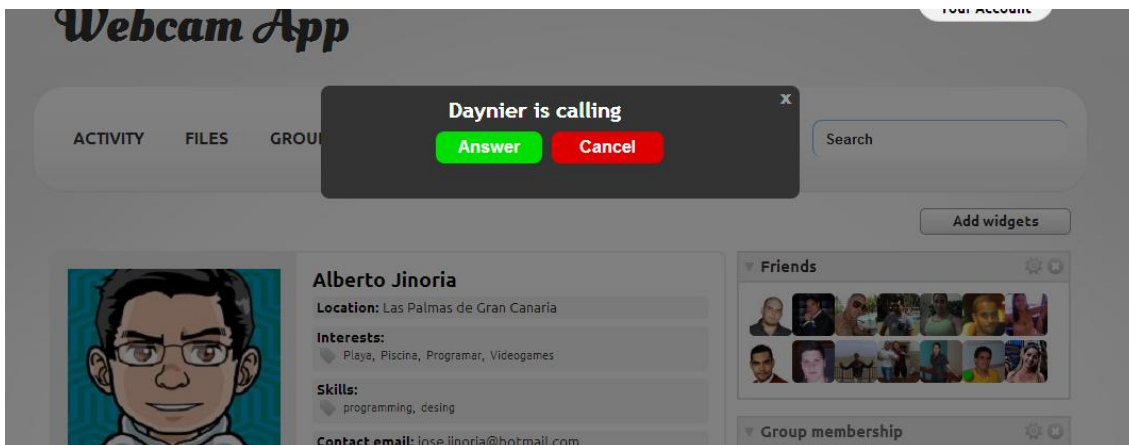


Ilustración 30: Cuadro de diálogo de notificación de videollamada.

5.4 Modelo de análisis

El modelo de análisis se utiliza como ayuda para refinar los requisitos. Analizar los requisitos en la forma de un modelo de análisis es importante por varios motivos:

- ❖ Un modelo de análisis ofrece una especificación más precisa de los requisitos que se obtienen como resultado de la captura de requisitos, incluyendo al modelo de casos de uso.
- ❖ Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores, se puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.
- ❖ Un modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación y en general su mantenimiento. Un modelo de análisis puede considerarse como una primera aproximación al modelo de diseño (aunque es un modelo por sí mismo) y es por tanto una entrada fundamental para darle forma al sistema en el diseño y en la implementación. (17)

5.4.1 Diagrama de clases

❖ Caso de Uso Registrarse

Relación de trazabilidad entre el caso de uso **Registrarse** del modelo de casos de uso y la realización del caso de uso **Registrarse** en el modelo de análisis (ver Ilustración 31).

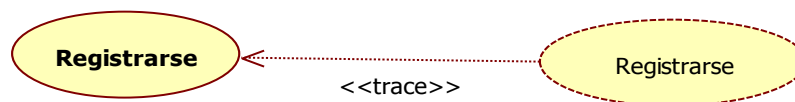


Ilustración 31: Relación de trazabilidad del caso de uso Registrarse.

Para el diagrama de clases del modelo de análisis se tienen las clases: **Control_Registro**, la cual se encarga de invocar a la clase **Interfaz_Registro**. La clase **Interfaz_Registro**, muestra un formulario para que el usuario pueda introducir sus datos personales y estos pasan a ser validados por la clase **Validar_Registro**. La clase **Validar_Registro** es una clase que tiene la función de comprobar que los datos insertados en el registro son correctos. Se identifica también **Control_Usuario**, la cual es una clase de control que recibe todos los datos del registro y se encarga de crear el usuario correspondiente. Por último, la clase **Cuenta_Usuario**, la cual es una entidad y representa al usuario en cuestión (ver Ilustración 32).

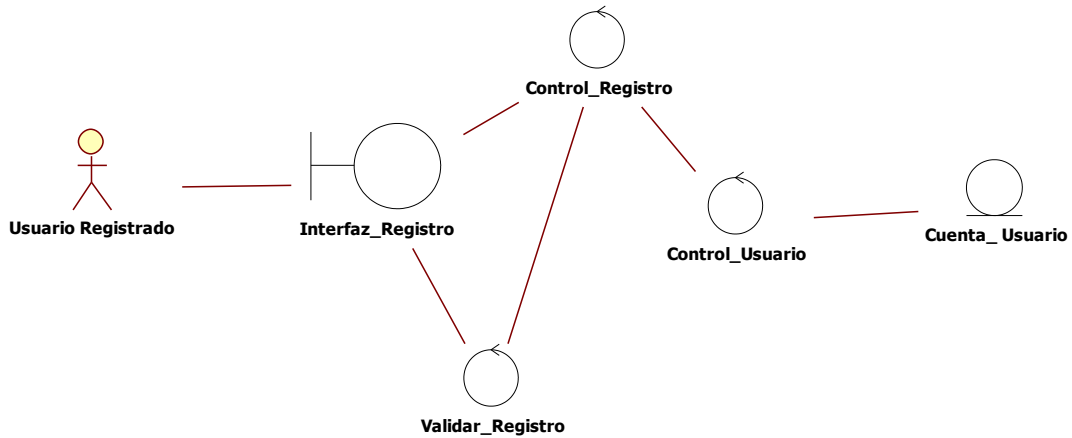


Ilustración 32: Diagrama de clases del análisis para el caso de uso Registrarse.

❖ **Caso de Uso Buscar Amigo**

Relación de trazabilidad entre el caso de uso **Buscar Amigo** del modelo de casos de uso y la realización del caso de uso **Buscar Amigo** en el modelo de análisis (ver Ilustración 33).

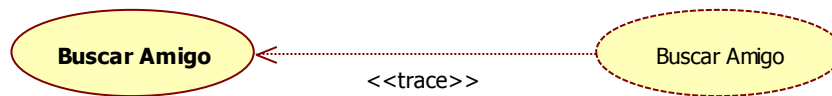


Ilustración 33: Relación de trazabilidad del caso de uso Buscar Amigo.

Este caso de uso tiene la clase controladora **Control_Búsqueda**, que es la encargada de generar las interfaces **Panel_Búsqueda** y **Listado_Resultados**, además está la clase **Control_Amigos** que es la encargada de la gestión de solicitudes de amistad. En el caso de uso también se encuentra identificada la clase entidad **Lista_Resultados**, que contiene lo que se muestra en la interfaz **Listado_Resultados**. Además de una clase entidad **Amigo** que es un usuario de Webcam App al cual se le envió la solicitud de amistad (ver Ilustración 34).

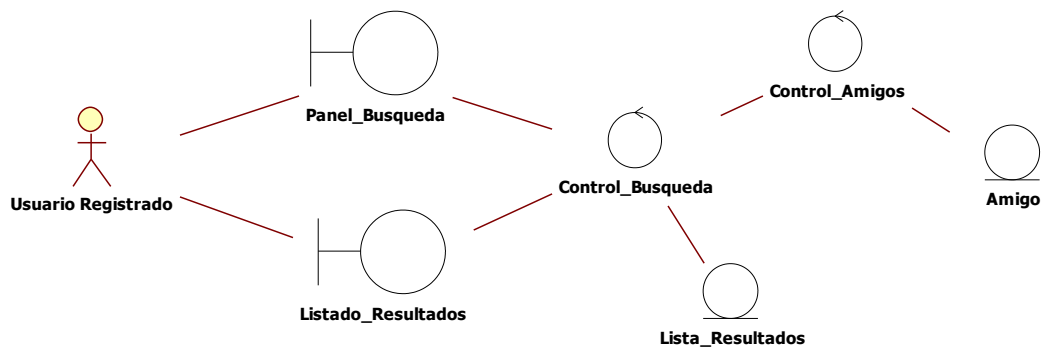


Ilustración 34: Diagrama de clases del análisis para el caso de uso Buscar Amigo.

❖ Caso de Uso Compartir Contenido

Relación de trazabilidad entre el caso de uso **Compartir Contenido** del modelo de casos de uso y la realización del caso de uso **Compartir Contenido** en el modelo de análisis (ver Ilustración 35).

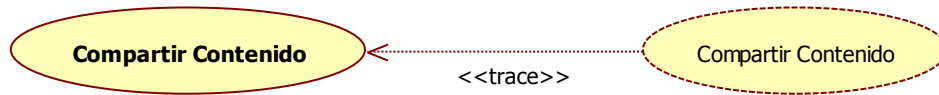


Ilustración 35: Relación de trazabilidad del caso de uso Compartir Contenido.

En el modelo de clases del análisis está la clase de interfaz **Contenido_Publicado**, se encarga de mostrar los archivos del usuario una vez que se haya agregado un nuevo contenido a la cuenta de usuario. Además, se encuentra la clase **Agregar_Contenido**, que es la interfaz que se muestra una vez que el usuario decide añadir un contenido nuevo. Se encuentran también las clases de control **Publicar_Contenido** y **Validar_Contenido**, la primera es la encargada tanto de agregar el contenido a la clase de entidad **Cuenta_Usuario** como de generar la interfaz llamada **Contenido_Publicado**; la clase **Validar_Contenido** se encarga de validar que el tamaño del nuevo archivo no es superior al espacio libre que la cuenta de usuario tiene asignado para publicar contenidos (ver Ilustración 36).

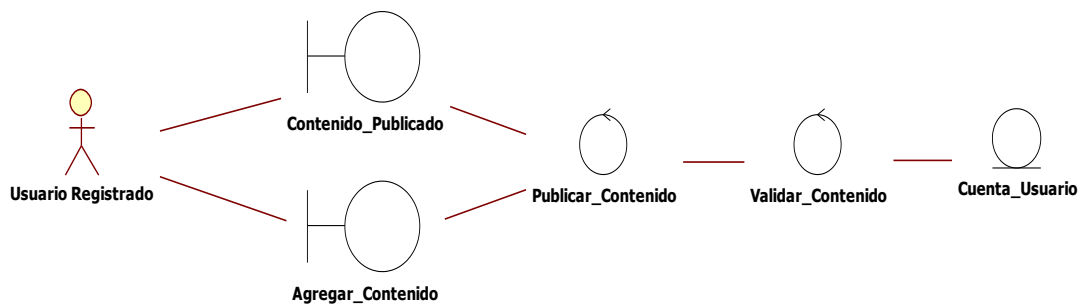


Ilustración 36: Diagrama de clases del análisis para el caso de uso Compartir Contenido.

❖ Caso de Uso Eliminar Contenido

Relación de trazabilidad entre el caso de uso **Eliminar Contenido** del modelo de casos de uso y la realización del caso de uso **Eliminar Contenido** en el modelo de análisis (ver Ilustración 37).

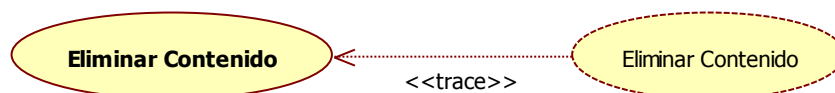


Ilustración 37: Relación de trazabilidad del caso de uso Eliminar Contenido.

En el modelo de clases del análisis se encuentra la clase interfaz **Confirmación_Eliminación** que es un formulario para saber si el usuario desea realmente eliminar el contenido seleccionado. Además, se localiza la clase interfaz **Contenido_Eliminado** la cual muestra que el contenido ha sido eliminado satisfactoriamente de los archivos del usuario. La clase de control

Eliminador_Contento_Control, la cual es la encargada de eliminar el contenido de la clase de entidad **Cuenta_Usuario**, además de generar las interfaces de **Confirmación_Eliminación** y **Contenido_Eliminado** (ver Ilustración 38).

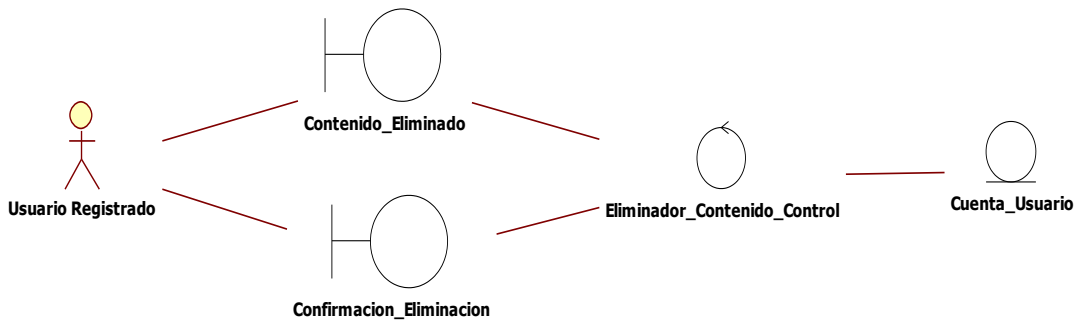


Ilustración 38: Diagrama de clases del análisis para el caso de uso Eliminar Contenido.

❖ **Caso de Uso Eliminar Cuenta Usuario**

Relación de trazabilidad entre el caso de uso **Eliminar Cuenta Usuario** del modelo de casos de uso y la realización del caso de uso **Eliminar Cuenta Usuario** en el modelo de análisis (ver Ilustración 39).

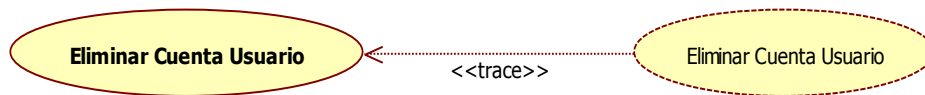


Ilustración 39: Relación de trazabilidad del caso de uso Eliminar Cuenta Usuario.

En el modelo de análisis para el caso de uso **Eliminar Cuenta Usuario** se utiliza la clase controladora **Gestor_Alertas**, que tiene como objetivo consultar las cuentas de usuarios y generar el listado de usuarios registrados que poseen alertas, además de un listado con las incidencias de cada usuario. La clase entidad encargada de guardar la información de usuarios registrados, **Cuenta_Usuario**. También existe otra clase entidad, **Listado_Alertas**, que guarda el listado de alertas generadas por la controladora **Gestor_Alertas**. La clase controladora llamada **Gestor_Usuarios**, se encarga de consultar el listado de alertas mencionado anteriormente y enviárselos a la clase interfaz **Lista_Alertas**. Además de enviar datos a la clase interfaz, esta clase es la encargada de eliminar o actualizar la información de la clase entidad **Cuenta_Usuario** (ver Ilustración 40).

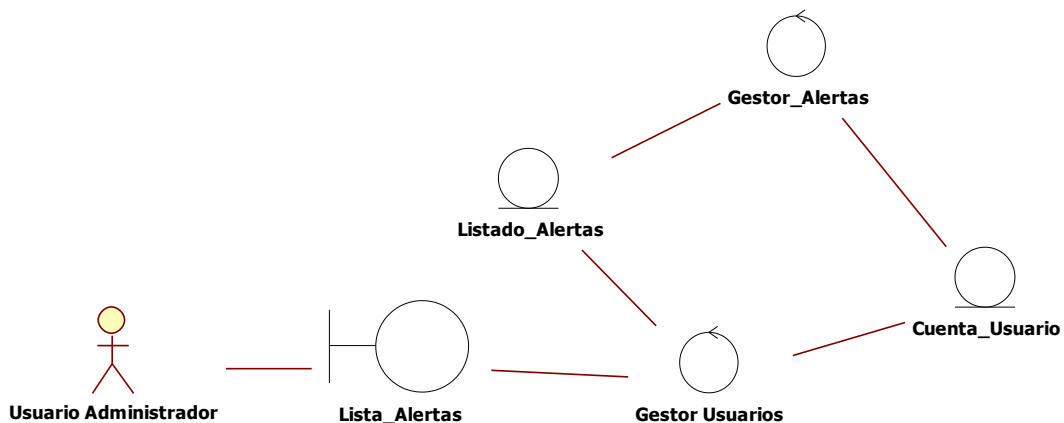


Ilustración 40: Diagrama de clases del análisis para el caso de uso Eliminar Cuenta Usuario.

❖ **Caso de Uso Login**

Relación de trazabilidad entre el caso de uso **Login** del modelo de casos de uso y la realización del caso de uso **Login** en el modelo de análisis (ver Ilustración 41).

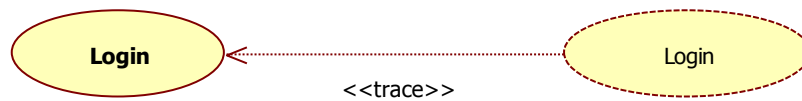


Ilustración 41: Relación de trazabilidad del caso de uso Login.

En el modelo de análisis para el caso de uso **Login** se utiliza la clase control **Gestor_Login**, la cual es la encargada de generar la clase interfaz **Formulario_Login** y de acceder a la cuenta de usuario. La clase **Formulario_Login** es la clase encargada de recoger los datos introducidos por el usuario. La clase entidad **Cuenta_Usuario**, que es la que guarda los datos de la cuenta, se utiliza por la clase control **Validador_Login** para verificar que los datos introducidos son correctos (ver Ilustración 42).

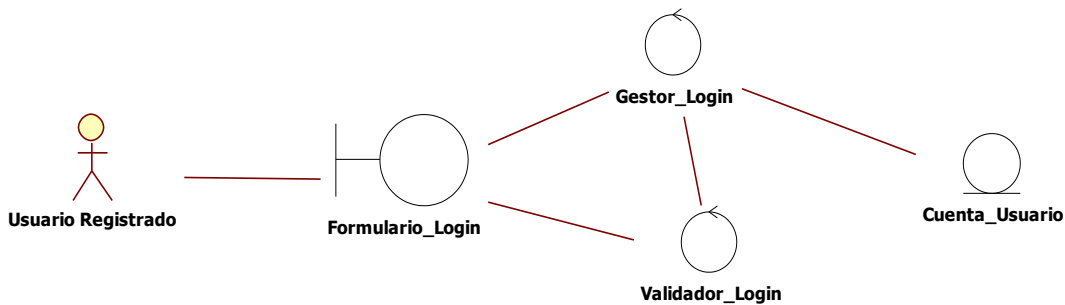


Ilustración 42: Diagrama de clases del análisis para el caso de uso Login.

❖ **Caso de Uso Realizar Videollamada**

Relación de trazabilidad entre el caso de uso **Realizar Videollamada** del modelo de casos de uso y la realización del caso de uso **Realizar Videollamada** en el modelo de análisis (ver Ilustración 43).

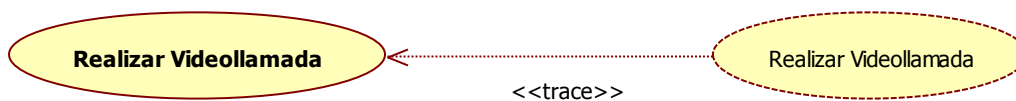


Ilustración 43: Relación de trazabilidad del caso de uso Realizar Videollamada.

En el modelo de análisis para el caso de uso **Realizar Videollamada** se encuentra la clase de control **Gestión_Videollamada**, dicha clase genera dos interfaces, la primera es **Panel_Videollamada** y la segunda **Ventana_Videollamada**. La primera interfaz muestra al usuario la opción de poder realizar videollamadas con sus amigos y la segunda es una interfaz que muestra una conexión con el servidor de aplicaciones Heroku. Además la clase **Gestión_Videollamada** es la encargada de generar la clase **Ventana_Diálogo**. La clase **Ventana_Diálogo** es la responsable de mostrarle al usuario toda la información de la videollamada que está recibiendo. La clase **Cuenta_Usuario** es la que contiene la información del usuario (ver Ilustración 44).

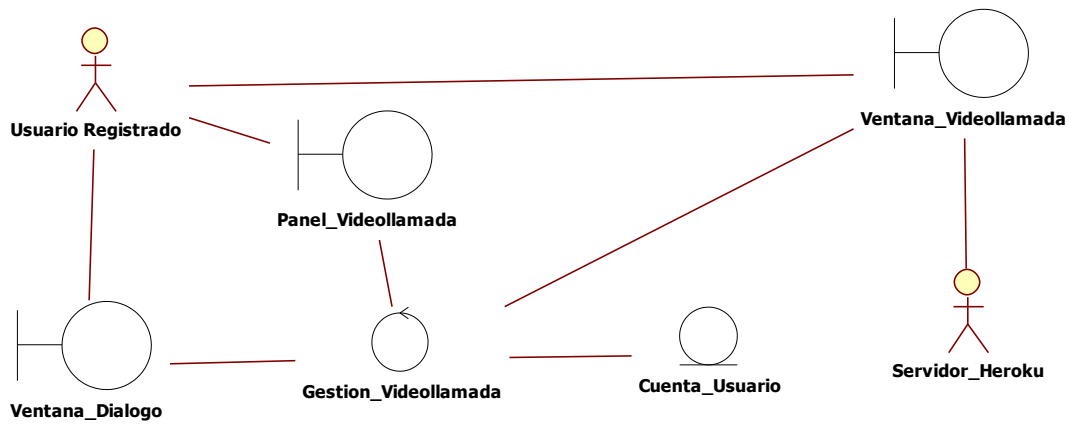


Ilustración 44: Diagrama de clases del análisis para el caso de Realizar Videollamada.

❖ **Caso de Uso Gestionar Perfil**

Una clase de análisis y sus objetos normalmente participan en varias realizaciones de casos de uso. Existe una relación de trazabilidad entre el caso de uso **Gestionar Perfil** del modelo de casos de uso y la realización del caso de uso **Gestionar Perfil** en el modelo de análisis (ver Ilustración 45).

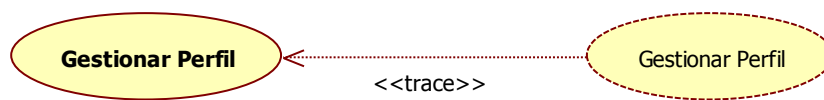


Ilustración 45: Relación de trazabilidad del caso de uso Gestionar Perfil.

En el modelo de análisis para el caso de uso **Gestionar Perfil** se encuentra la clase de control **Gestión_Usuario**, que se encarga de generar los datos que se muestran en la clase **Formulario_Datos_Perfil**, obtenidos de la cuenta de usuario. La clase **Formulario_Datos_Perfil** es la clase encargada de mostrar los datos asociados a la cuenta de usuario en los campos para ser editados. Los datos introducidos o editados son validados por la clase **Validar_Datos** la cual es una clase controladora que se encarga de verificar que los datos introducidos son correctos. Y por último la clase **Cuenta_Usuario** que es la clase que contiene los datos asociados al usuario (ver Ilustración 46).

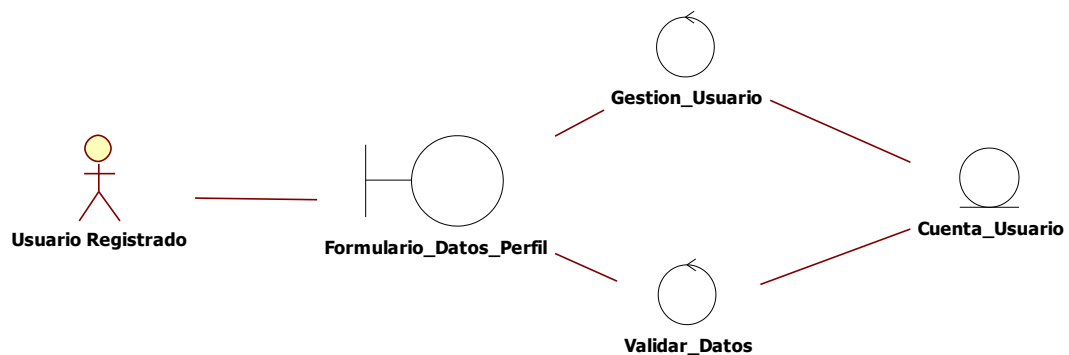


Ilustración 46: Diagrama de clases del análisis para el caso de uso Gestionar Perfil.

5.4.2 Diagramas de colaboración

❖ Caso de Uso Registrarse

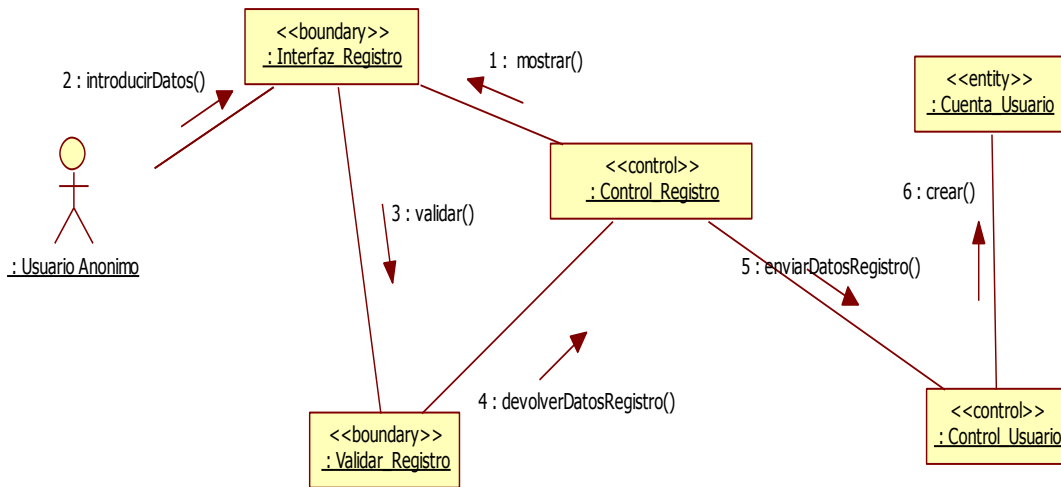


Ilustración 47: Diagrama de colaboración para el flujo normal del caso de uso Registrarse.

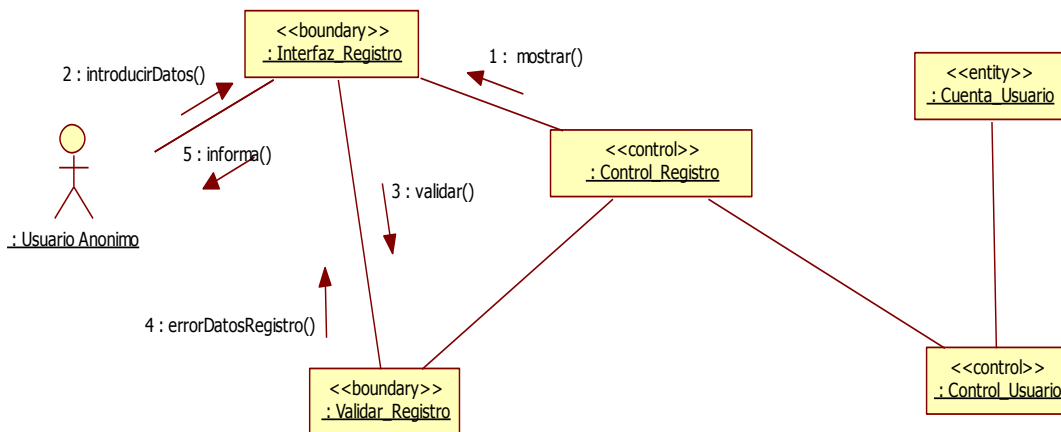


Ilustración 48: Diagrama de colaboración para el flujo alternativo del caso de uso Registrarse.

❖ Caso de Uso Buscar Amigo

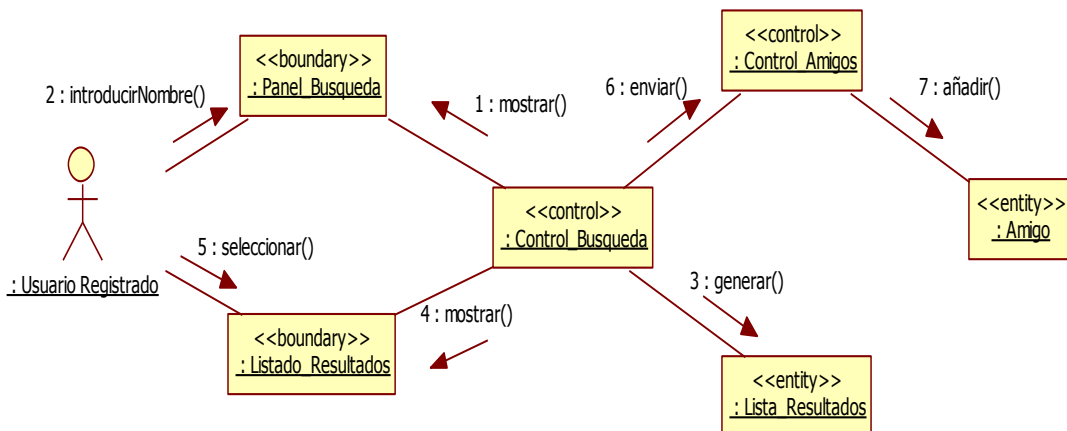


Ilustración 49: Diagrama de colaboración para el flujo normal del caso de uso Buscar Amigo.

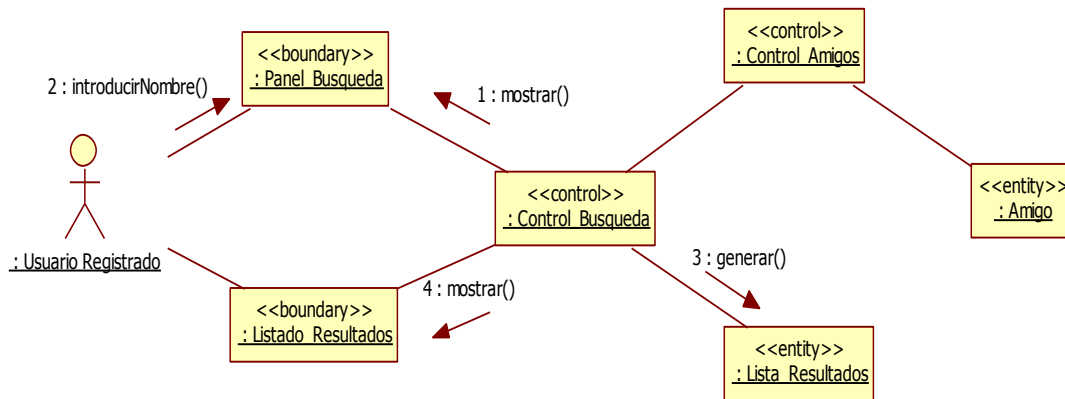


Ilustración 50: Diagrama de colaboración para el flujo alternativo del caso de uso Buscar Amigo.

❖ Caso de Uso Compartir Contenido

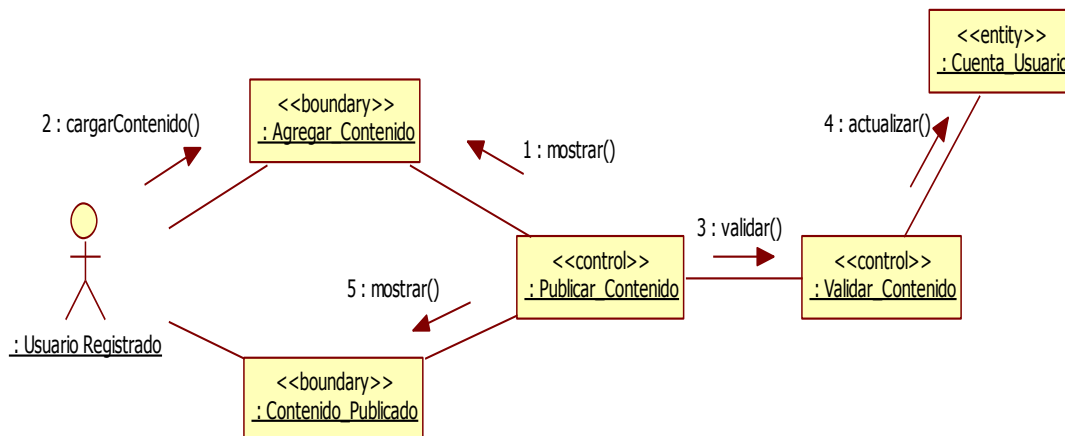


Ilustración 51: Diagrama de colaboración para el flujo normal del caso de uso Compartir Contenido.

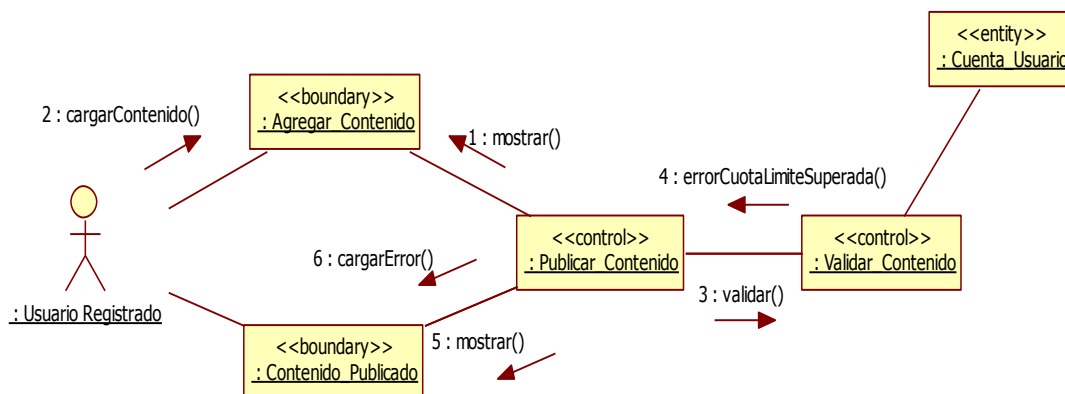


Ilustración 52: Diagrama de colaboración para el flujo alternativo del caso de uso Publicar Contenido.

❖ **Caso de Uso Eliminar Contenido**

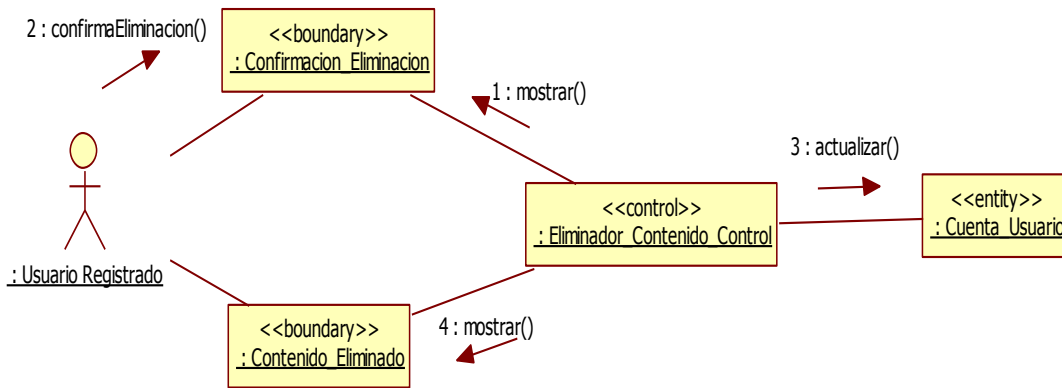


Ilustración 53: Diagrama de colaboración para el flujo normal del caso de uso Eliminar Contenido.

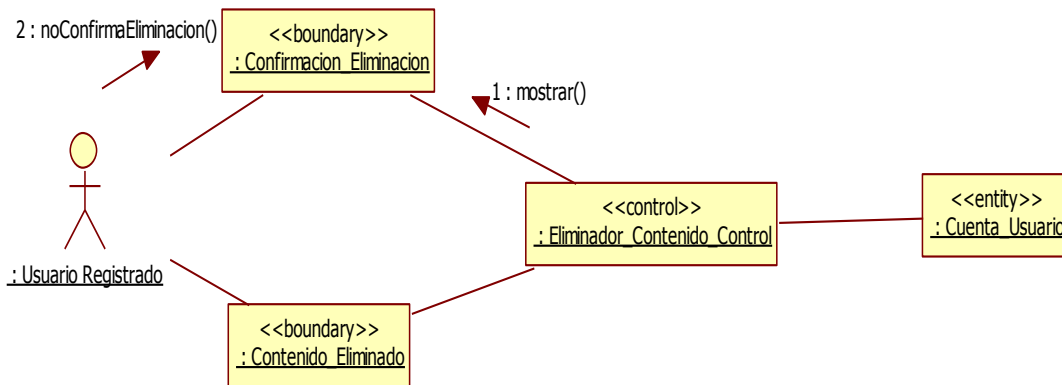


Ilustración 54: Diagrama de colaboración para el flujo alternativo del caso de uso Eliminar Contenido.

❖ **Caso de Uso Eliminar Cuenta Usuario**

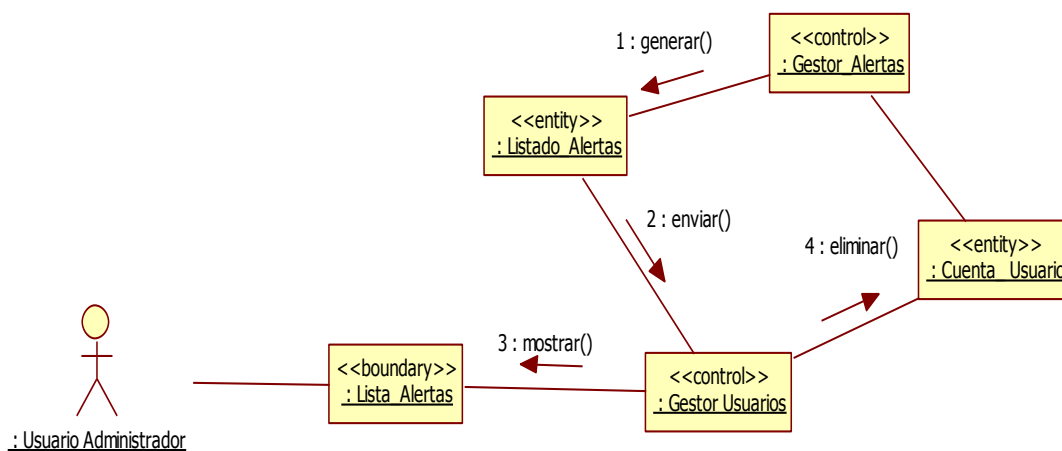


Ilustración 55: Diagrama de colaboración para el flujo normal del caso de uso Eliminar Cuenta Usuario.

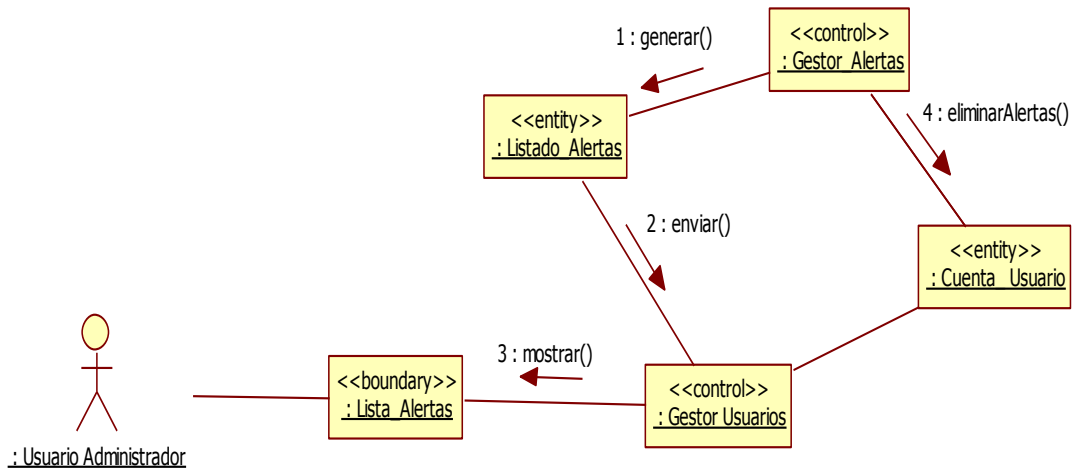


Ilustración 56: Diagrama de colaboración para el flujo alternativo del caso de uso Eliminar Cuenta Usuario.

❖ Caso de Uso Login

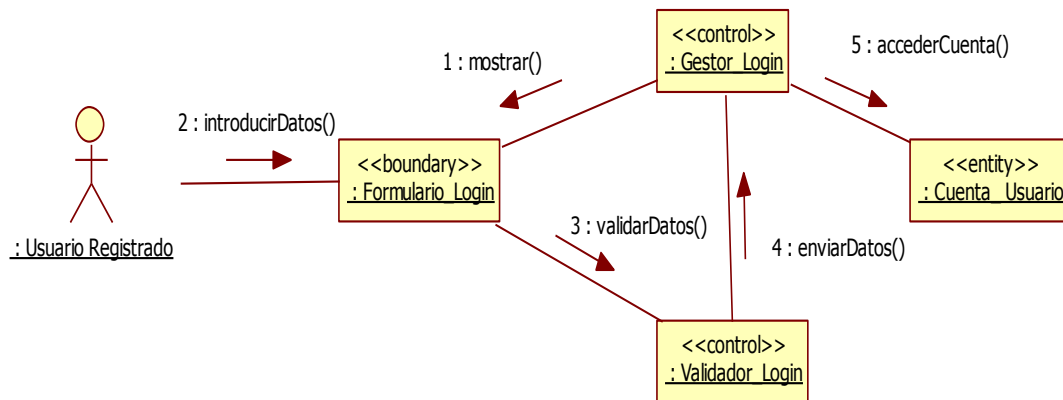


Ilustración 57: Diagrama de colaboración para el flujo normal del caso de uso Login.

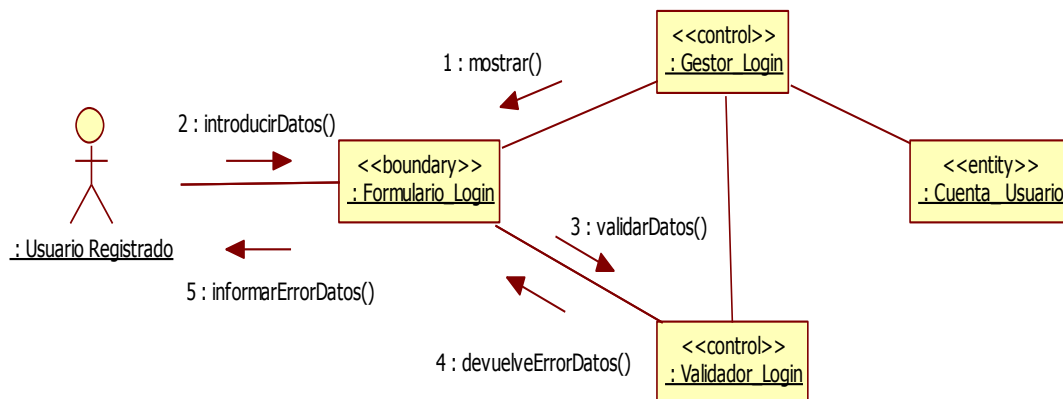


Ilustración 58: Diagrama de colaboración para el flujo alternativo del caso de uso Login.

❖ **Caso de Uso Realizar Videollamada**

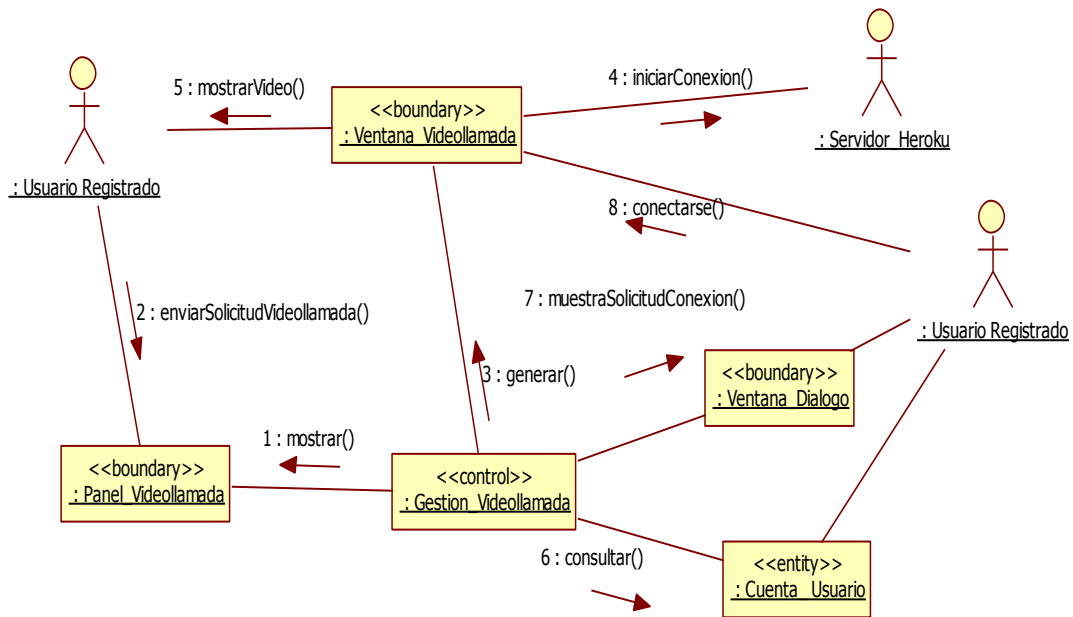


Ilustración 59: Diagrama de colaboración para el flujo normal del caso de uso Realizar Videollamada.

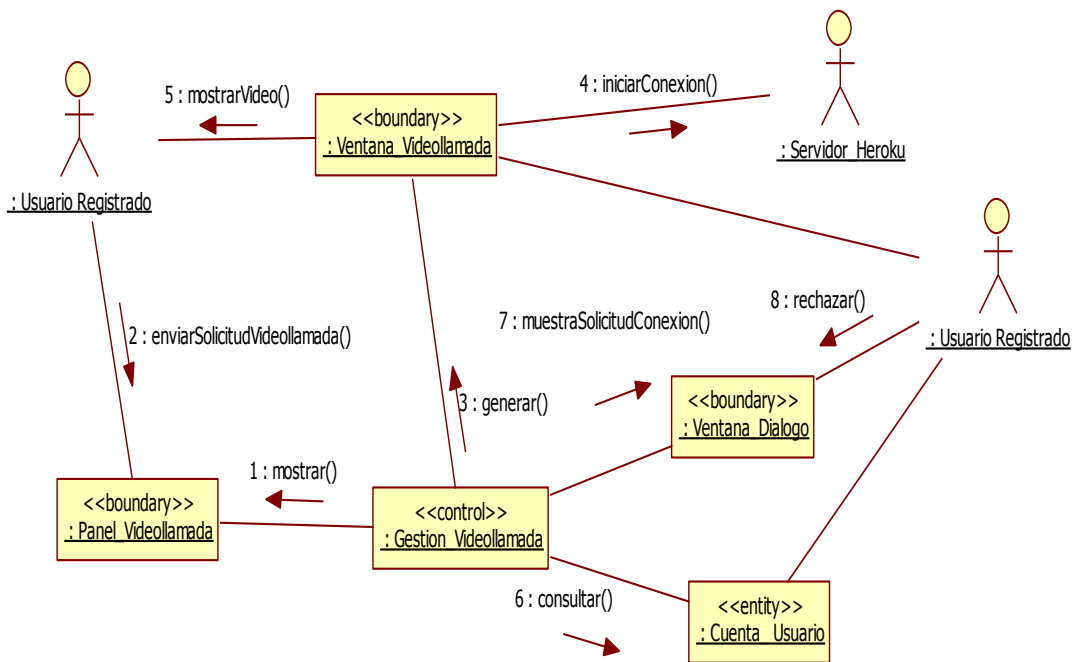


Ilustración 60: Diagrama de colaboración para el flujo alternativo del caso de uso Realizar Videollamada.

❖ **Caso de Gestionar Perfil**

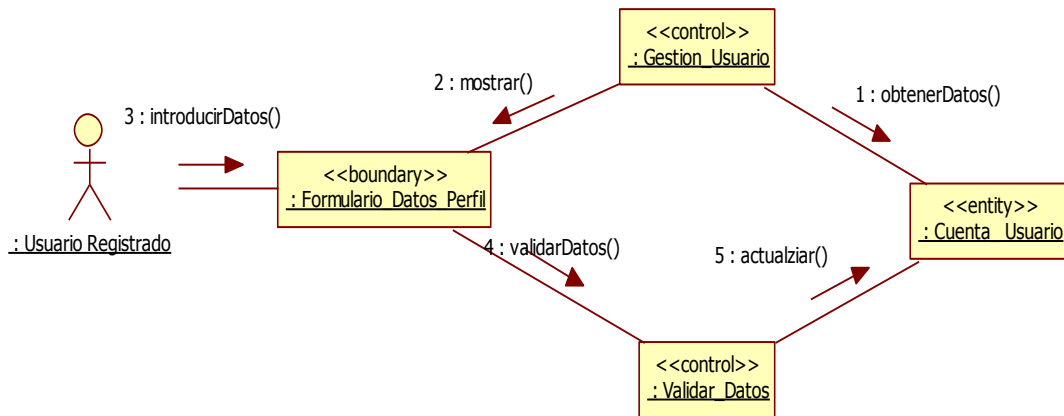


Ilustración 61: Diagrama de colaboración para el flujo normal del caso de uso Gestionar Perfil.

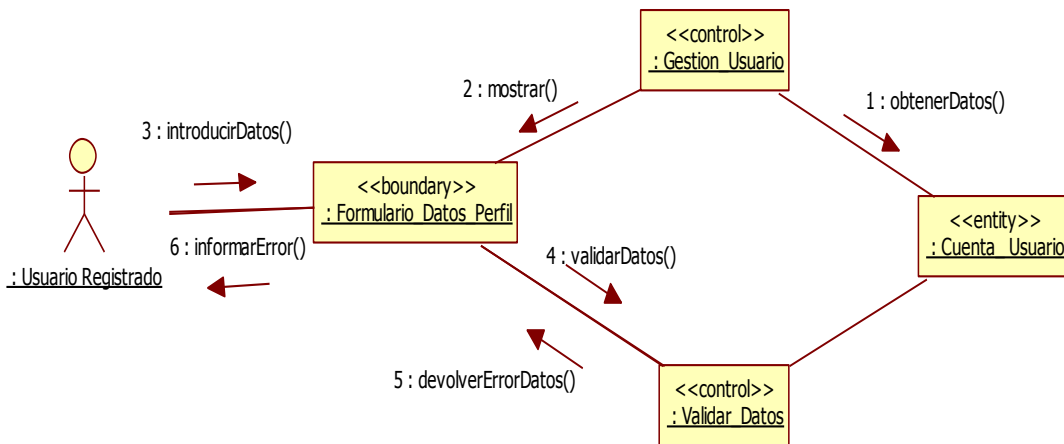


Ilustración 62: Diagrama de colaboración para el flujo alternativo del caso de uso Gestionar Perfil.

5.5 Modelo de diseño

El diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución. Prepara para la implementación y las pruebas del sistema. Pretende crear un plano del modelo de implementación, por lo que el grueso del esfuerzo está en las últimas iteraciones de elaboración y las primeras de construcción. El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos.

La Disciplina de Diseño tiene entre sus propósitos:

- ❖ Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y de interfaz de usuario.
- ❖ Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- ❖ Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- ❖ Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

5.5.1 Arquitectura del sistema

Para soportar el modelo de despliegue del sistema se definen los siguientes elementos:

- ❖ Dos balanceadores de carga con Apache Web Server 2
Las peticiones a los mismos se hará de forma alternada a través del Servidor DNS, que distribuirá las peticiones a cada balanceador. Cada balanceador distribuirá las conexiones a cada uno de los clústeres de Apache Tomcat y así estos se conectan a los servidores de DB. El objetivo de los dos balanceadores es no sobrecargar un solo Apache y en caso que deje de prestar servicio uno de ellos, los servicios no dejarán de funcionar. Los Apache Tomcat clústeres se conectarán a los sistemas externos, servidores de aplicaciones que soportan la tecnología de Node.js (ver Ilustración 63).

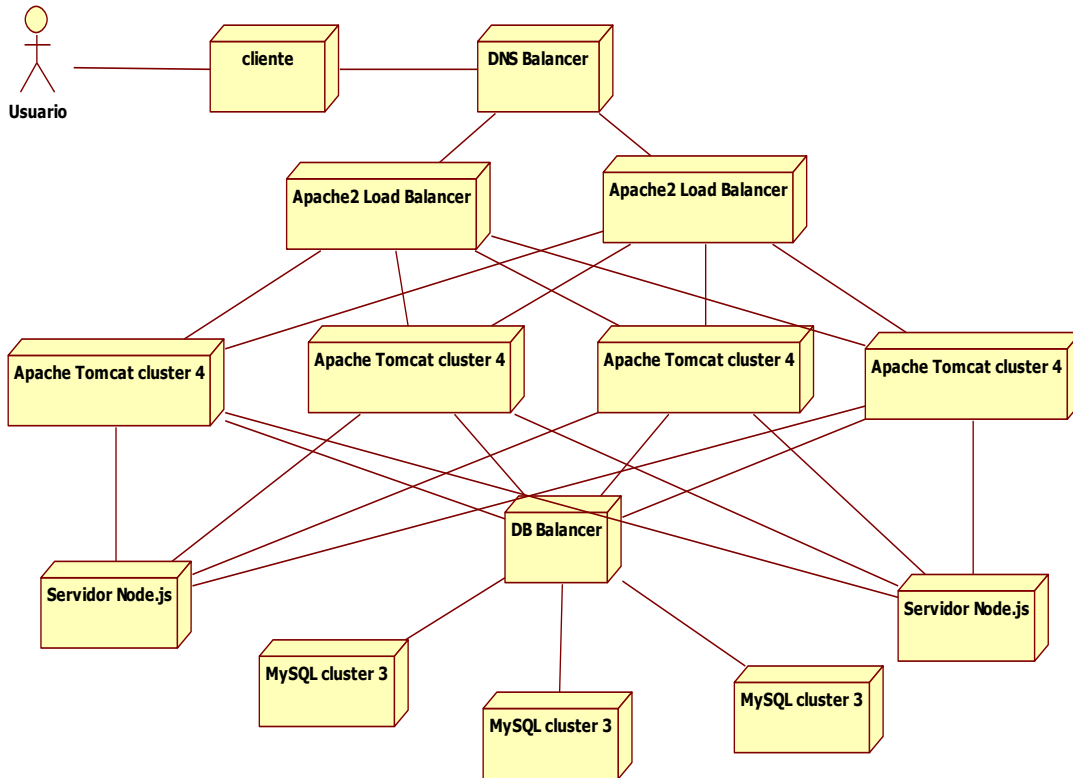


Ilustración 63: Modelo de despliegue.

5.5.2 Diagramas de clases

❖ Caso de Uso Registrarse

Una realización de caso de uso del diseño **Registrarse** proporciona una traza directa a una realización de caso de uso de análisis **Registrarse** en el modelo de análisis (ver Ilustración 64).

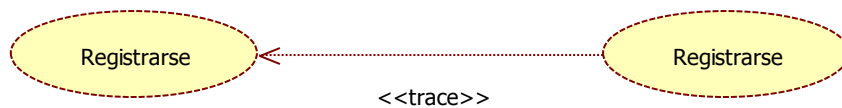


Ilustración 64: Relación de trazabilidad del caso de uso Registrarse.

En el diagrama de clases del modelo de diseño se identifican las siguientes clases: **Formulario**, clase genérica la cual puede representar cualquier formulario, en este caso con el atributo **Datos** y el método **enviarDatos ()**, para enviar al control los datos rellenos por el usuario. También encontramos la clase **UIFormulario_Registro** que hereda de la clase formulario. La clase **Controladora_Usuarios** se encarga de crear la cuenta de usuario correspondiente y almacenarla en la base de datos del sistema mediante el método **crear ()**.

Cuenta_Usuario es una clase entidad para representar a un usuario concreto, con un atributo datos que representa información acerca del usuario en cuestión. Además, la clase **Controladora_Registro** se encarga de mostrar el formulario de registro y recibir los datos del mismo. La clase **Validador** comprueba los datos recibidos de la clase **UIFormulario_Registro**, para verificar que efectivamente los datos introducidos son correctos (ver Ilustración 65).

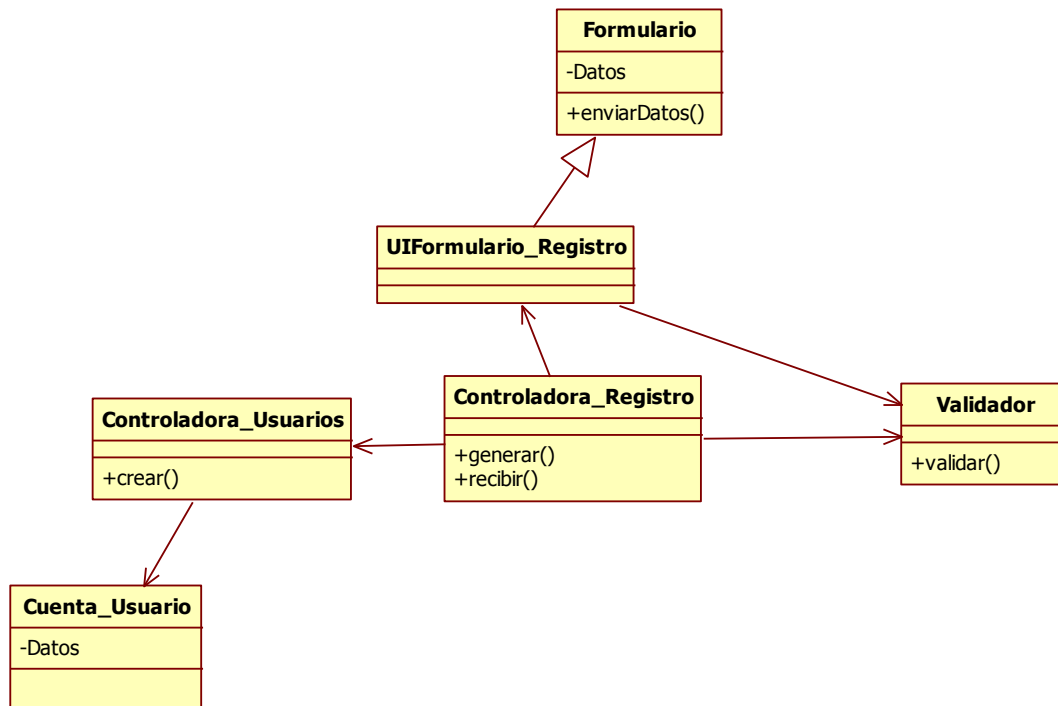


Ilustración 65: Diagrama de clases del diseño para caso de uso Registrarse.

❖ **Caso de Uso Buscar Amigos**

Una realización de caso de uso del diseño **Buscar Amigo** proporciona una traza directa a una realización de caso de uso de análisis **Buscar Amigo** en el modelo de análisis (ver Ilustración 66).

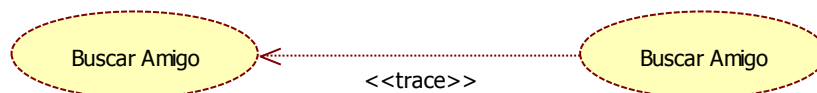


Ilustración 66: Relación de trazabilidad para el caso de uso Buscar Amigo.

En el modelo de clases de diseño del caso de uso **Buscar Amigo**, se identifica la clase **Control_Amigos** en la cual se encuentra el método **buscarAmigo ()**, dicho método localiza a los usuarios con el nombre introducido en el formulario de búsqueda. Se identifica la clase **Lista_Amigos** a la cual se le añaden los amigos que el usuario haya querido agregar, de ahí se genera la clase **Amigo** con sus datos, por ejemplo **Nombre**. Se tiene la clase **Lista_Resultados**, la cual contiene un listado con los contactos dados de alta en Webcam App que poseen el nombre del usuario que se desea encontrar. También por último la clase **UIBuscador_Amigos** que es la clase vista que se utiliza para mostrar los datos al usuario (ver Ilustración 67).

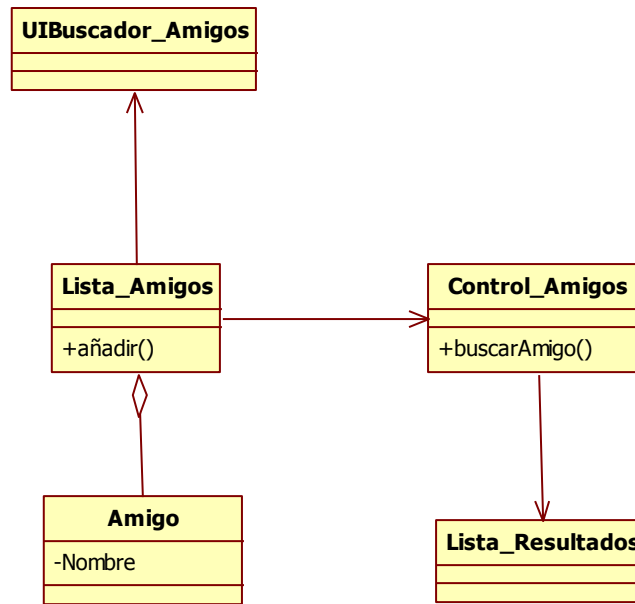


Ilustración 67: Diagrama de clases del diseño para el caso de uso Buscar Amigo.

❖ **Caso de Uso Compartir Contenido**

Una realización de caso de uso del diseño **Compartir Contenido** proporciona una traza directa a una realización de caso de uso de análisis **Compartir Contenido** en el modelo de análisis (ver Ilustración 68).

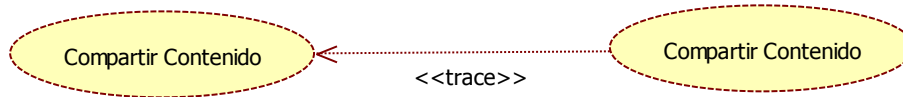


Ilustración 68: Relación de trazabilidad para caso de uso Compartir Contenido.

En el modelo de diseño está presente la clase **Gestor_Contenido**, que contiene los métodos **almacenar ()**, con el cual se almacena el contenido en la cuenta de usuario y el método **actualizar ()**, que actualiza los archivos del usuario después de agregar el contenido. La clase **Gestor_Contenido** tiene relación con las clases **Cuenta_Usuario** y **Validador**. La clase **Cuenta_Usuario** contiene los datos personales de la cuenta del usuario y sus archivos. La clase **Validador** contiene el método de **validar ()**, que comprueba que el usuario puede compartir el contenido. La clase **UIFormulario_Contenido** contiene el método **cargarContenido ()**, dicha clase envía los datos a la clase **Validador** (ver Ilustración 69).

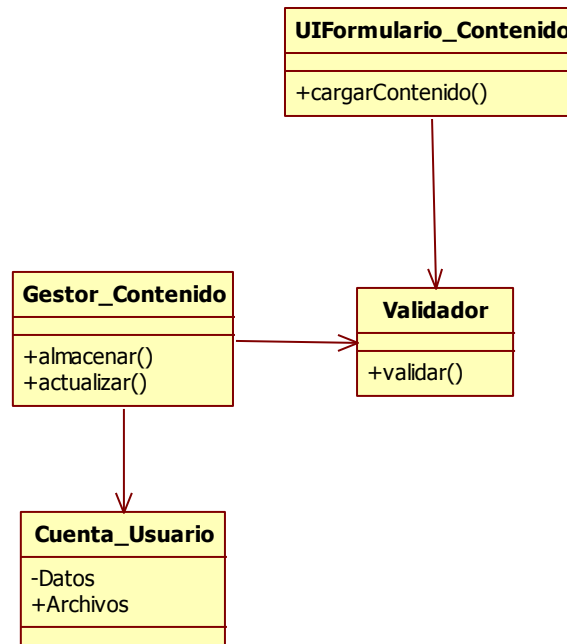


Ilustración 69: Diagrama de clases del diseño para el caso de uso Compartir Contenido.

❖ **Caso de Uso Eliminar Contenido**

Una realización de caso de uso del diseño **Eliminar Contenido** proporciona una traza directa a una realización de caso de uso de análisis **Eliminar Contenido** en el modelo de análisis (ver Ilustración 70).

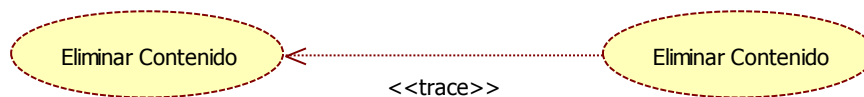


Ilustración 70: Relación de trazabilidad para el caso de uso Eliminar Amigo.

En el modelo de diseño del caso de uso Eliminar Contenido está la clase **Gestor_Contenido** que contiene el método **eliminar ()**, dicho método suprime el contenido de la cuenta de usuario y el método **actualizar ()** que actualiza los archivos del usuario después de eliminar el contenido. La clase **Gestor_Contenido** tiene relación con las clases **Cuenta_Usuario** y **Mensaje_Confirmación**. La clase **Cuenta_Usuario** contiene los datos personales de la cuenta del usuario y sus archivos. La clase **Mensaje_Confirmación** contiene el método **notificar ()**, dicha clase le muestra al usuario un mensaje para que el usuario confirme la eliminación (ver Ilustración 71).

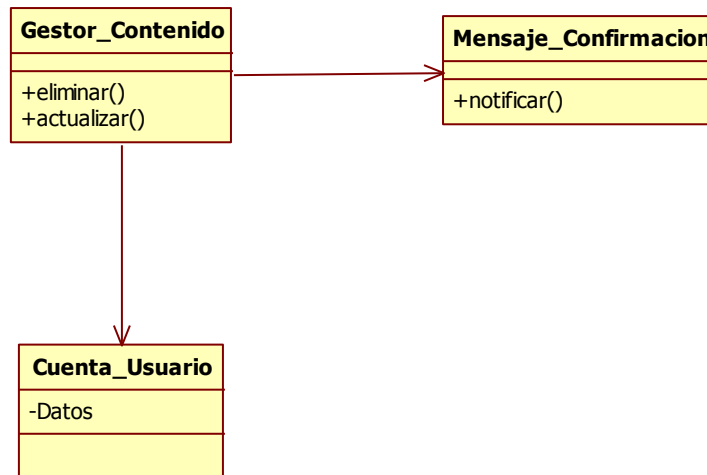


Ilustración 71: Diagrama de clases del diseño para el caso de uso Eliminar Contenido.

❖ **Caso de Uso Eliminar Cuenta Usuario**

Una realización de caso de uso del diseño *Eliminar Cuenta Usuario* proporciona una traza directa a una realización de caso de uso *Eliminar Cuenta Usuario* de análisis en el modelo de análisis (ver Ilustración 72).

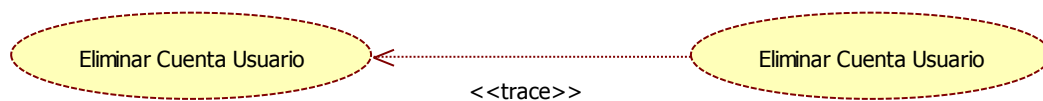


Ilustración 72: Relación de trazabilidad para el caso de uso Eliminar Cuenta Usuario.

Para el caso de uso Eliminar Cuenta Usuario en el modelo de clases de diseño se utiliza la clase *Gestor_Alertas*, la cual genera un listado de alertas a través del método *generarListaAlertas ()*. La clase que se encarga de guardar la información de usuarios registrados se llama *Cuenta_Usuario*. *Lista_Alertas_Usuarios* tiene como atributo la lista generada por la clase *Gestor_Alertas* y envía los datos a la clase *UIReporte_Alertas*, que es la encargada de mostrarle los datos al usuario administrador, para visualizar los datos implementa el método *mostrarAlertas ()*. La clase *Gestor_Usuarios*, es la encargada de eliminar o actualizar la información de la clase *Cuenta_Usuario*. Para dar solución a estas últimas funcionalidades la clase tiene implementados los métodos *eliminar ()* y *actualizar ()* (ver Ilustración 73).

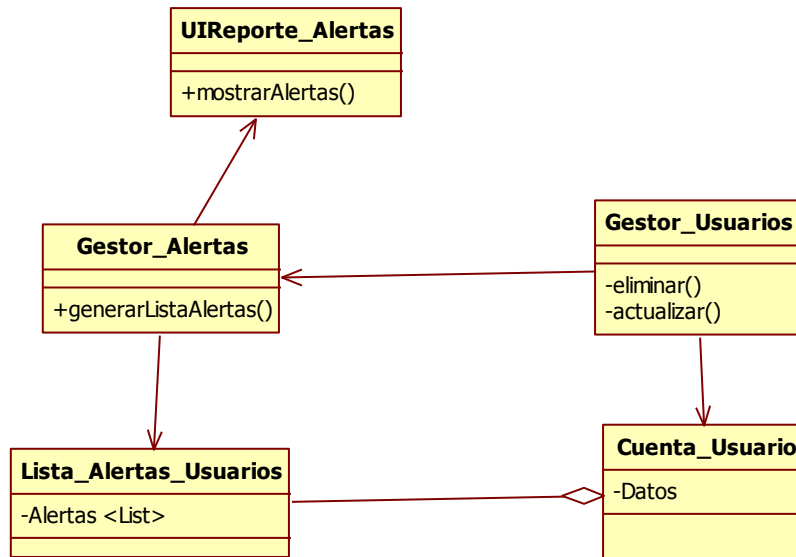


Ilustración 73: Diagramas de clases del diseño para el caso de uso Eliminar Cuenta Usuario.

❖ **Caso de Uso Login**

Una realización de caso de uso del diseño **Login** proporciona una traza directa a una realización de caso de uso de análisis **Login** en el modelo de análisis (ver Ilustración 74).

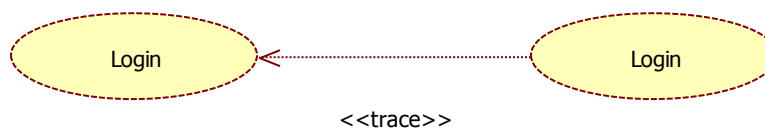


Ilustración 74: Relación de trazabilidad para el caso de uso Login

En el modelo de diseño se encuentran las clases **UIFormulario_Login** que contiene el método **enviarDatos ()**. La clase **Validador**, como su nombre lo indica, valida que sean correctos los datos que el usuario introduce para hacer el Login a través del método **validar ()**, dichos datos son enviados por la clase **UIFormulario_Login**. Además, en el caso de uso se encuentra la clase **Cuenta_Usuario**, la cual contiene todos los datos asociados a las cuentas que se han creado en la aplicación y a la cual deben pertenecer los datos que el usuario introduce. Y por último está la clase **Gestor_Login**, la cual es la encargada de acceder a los datos de la cuenta de usuario a través del método de **recibir ()** y de generar el formulario de **UIFormulario_Login** a través del método **generar ()** (ver Ilustración 75).

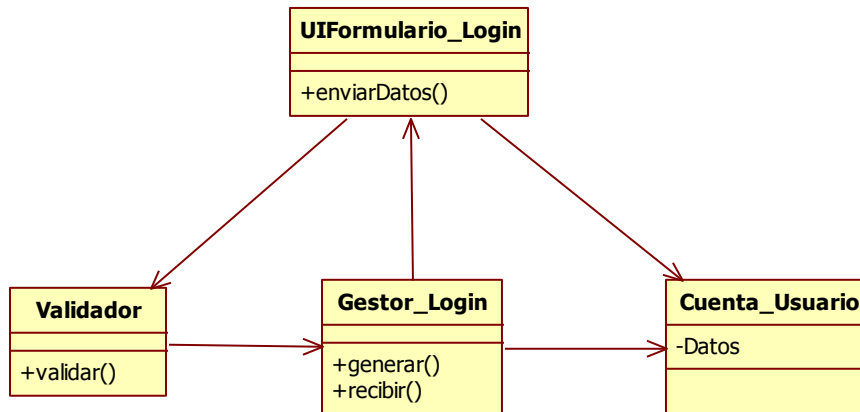


Ilustración 75: Diagrama de clases del diseño para el caso de uso Login.

❖ **Caso de Uso Realizar Videollamada**

Una realización de caso de uso del diseño **Realizar Videollamada** proporciona una traza directa a una realización de caso de uso de análisis **Realizar Videollamada** en el modelo de análisis (ver Ilustración 76).

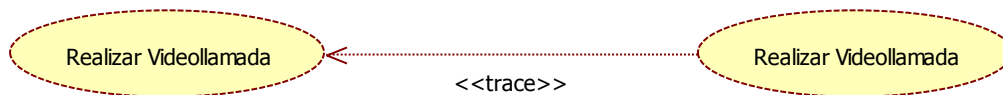


Ilustración 76: Relación de trazabilidad para el caso de uso Realizar Videollamada

En el modelo del diseño del caso de uso Realizar Videollamada, está la clase **Gestión_Videollamada**, la cual se encarga de generar las interfaces a través del método **generar ()**, además de informarle a través del método **enviarLlamada ()** a la clase **Ventana_Diálogo** cuando se ha realizado una videollamada nueva, para que le notifique al usuario que está siendo llamado. La clase **Cuenta_Usuario** contiene el nombre de usuario con el cual se establecerán las videollamadas. La clase **Ventana_Videollamadas**, contiene los videos, tanto el video local como el video del usuario remoto. La clase **Panel_Videollamada**, contiene las opciones para que el usuario pueda realizar videollamadas con sus amigos. Y por último en el caso de uso se utiliza un subsistema externo **Servidor_Aplicaciones_Heroku**, el cual está implementado el proceso del intercambio de videos entre los usuarios (ver Ilustración 77).

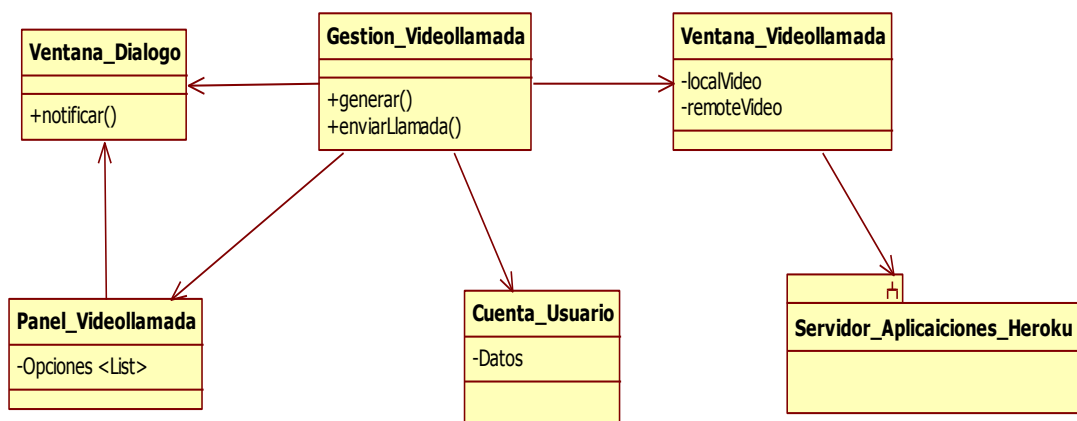


Ilustración 77: Diagrama de clases del diseño para el caso de uso Realizar Videollamada.

❖ Caso de Uso Gestionar Perfil

Una realización de caso de uso del diseño **Gestionar Perfil** proporciona una traza directa a una realización de caso de uso de análisis **Gestionar Perfil** en el modelo de análisis (ver Ilustración 78).

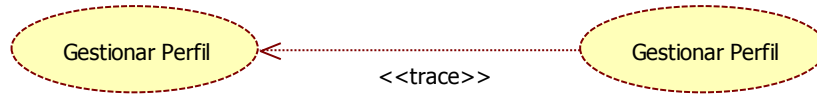


Ilustración 78: Relación de trazabilidad para el caso de uso Gestionar Perfil

En el modelo de diseño se encuentra la clase **UIFormulario_Datos_Perfil** que contiene el método **enviarDatos ()**, además de los datos actuales de la cuenta de usuario. La clase **Validador**, como su nombre lo indica, valida los datos que el usuario introduce sean correctos a través del método **validar ()**. Además en el caso de uso se encuentra la clase **Cuenta_Usuario**, la cual contiene todos los datos asociados a la cuenta que se ha creado en la aplicación y a la cual deben pertenecer los datos que el usuario introduce. Y por último está la clase **Gestor_Usuario**, la cual es la encargada de acceder a los datos de la cuenta de usuario a través del método de **recibir ()**, de generar el formulario de **UIFormulario_Datos_Perfil** a través del método **generar ()** y de cambiar los datos en la cuenta de usuario a través del método **actualizar ()** (ver Ilustración 79).

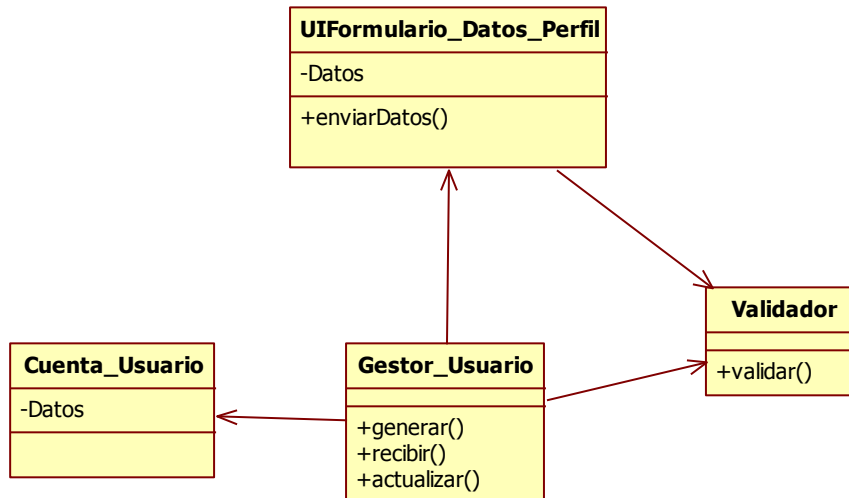


Ilustración 79: Diagrama de clases del diseño para el caso de uso Gestionar Perfil.

5.5.3 Diagramas de secuencia

❖ Caso de Uso Registrarse

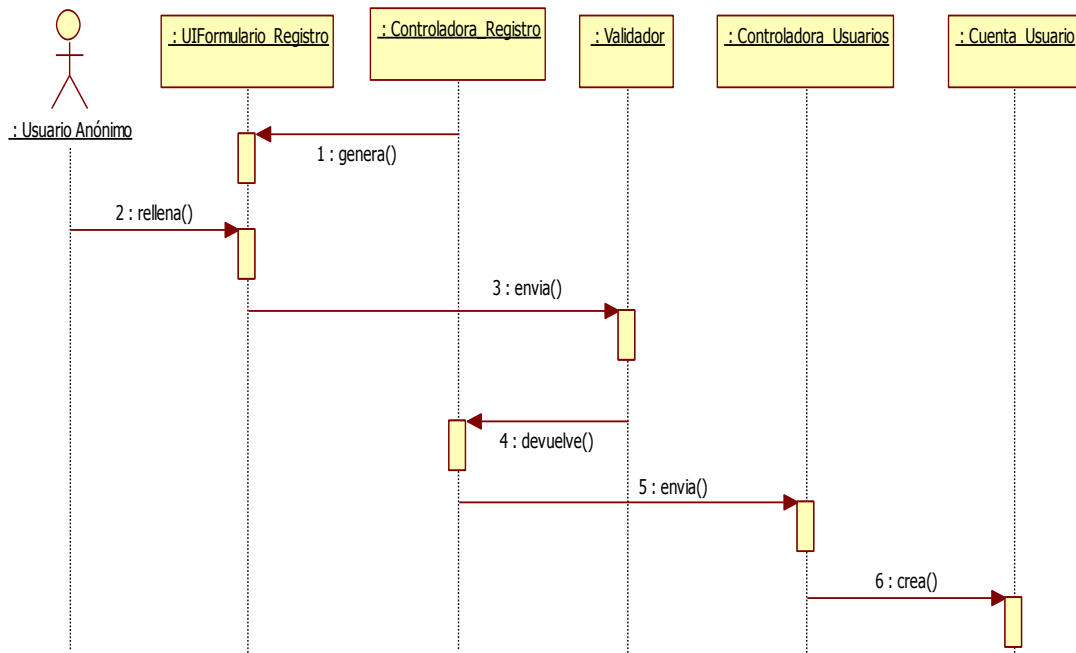


Ilustración 80: Diagrama de secuencia para el flujo normal del caso de uso Registrarse.

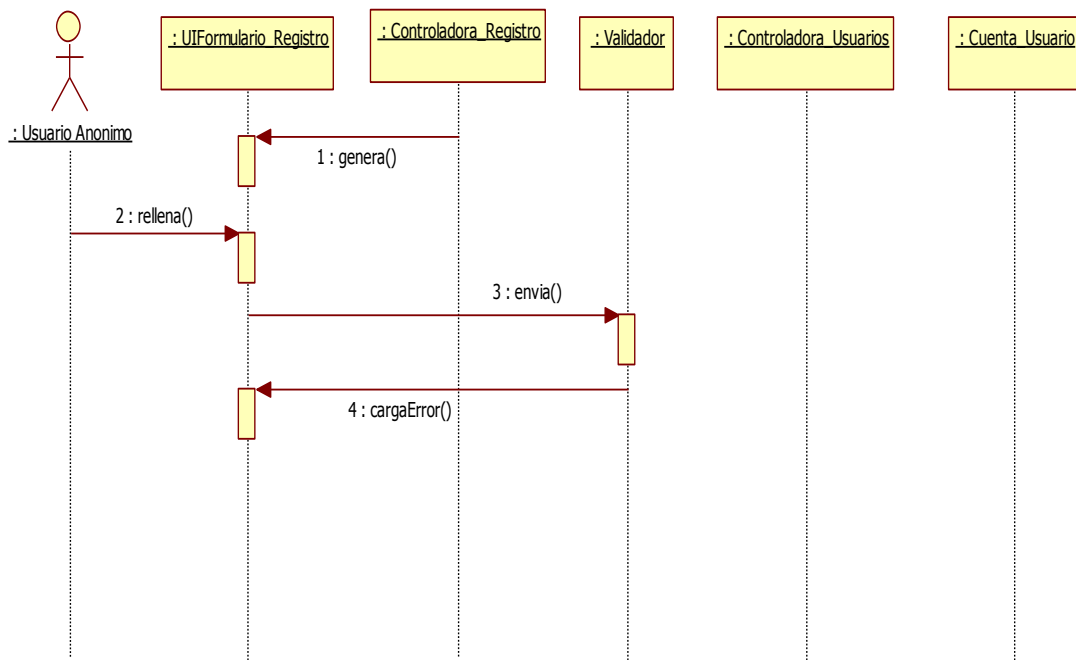


Ilustración 81: Diagrama de secuencia para el flujo alternativo del caso de uso Registrarse.

❖ **Caso de Uso Buscar Amigo**

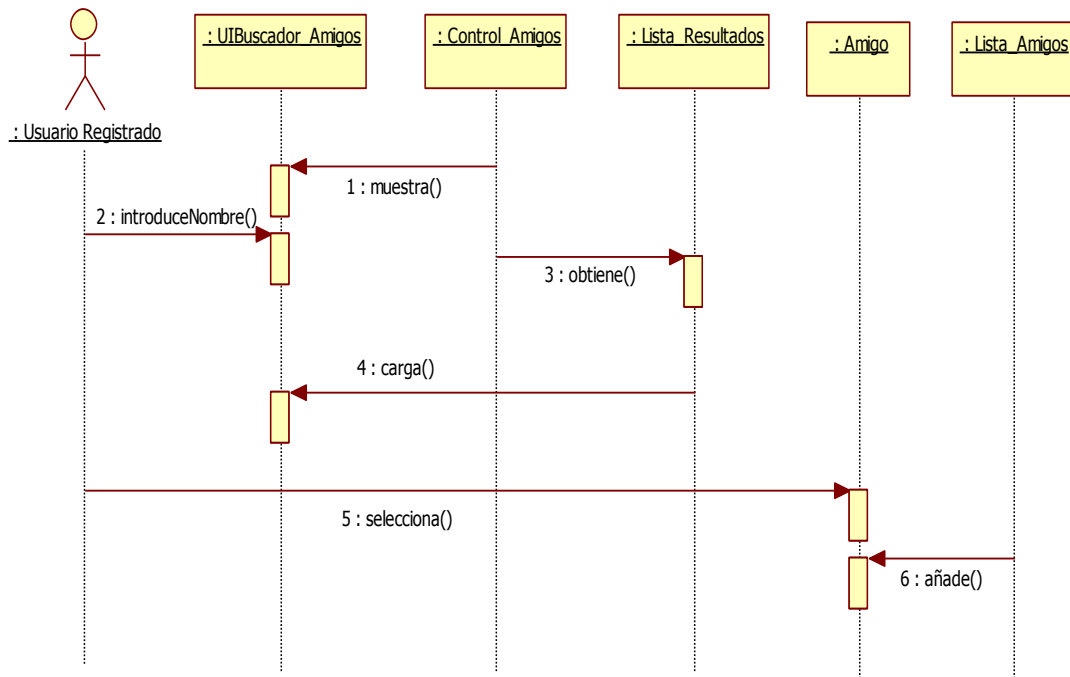


Ilustración 82: Diagrama de secuencia para el flujo normal del caso de uso Buscar Amigo.

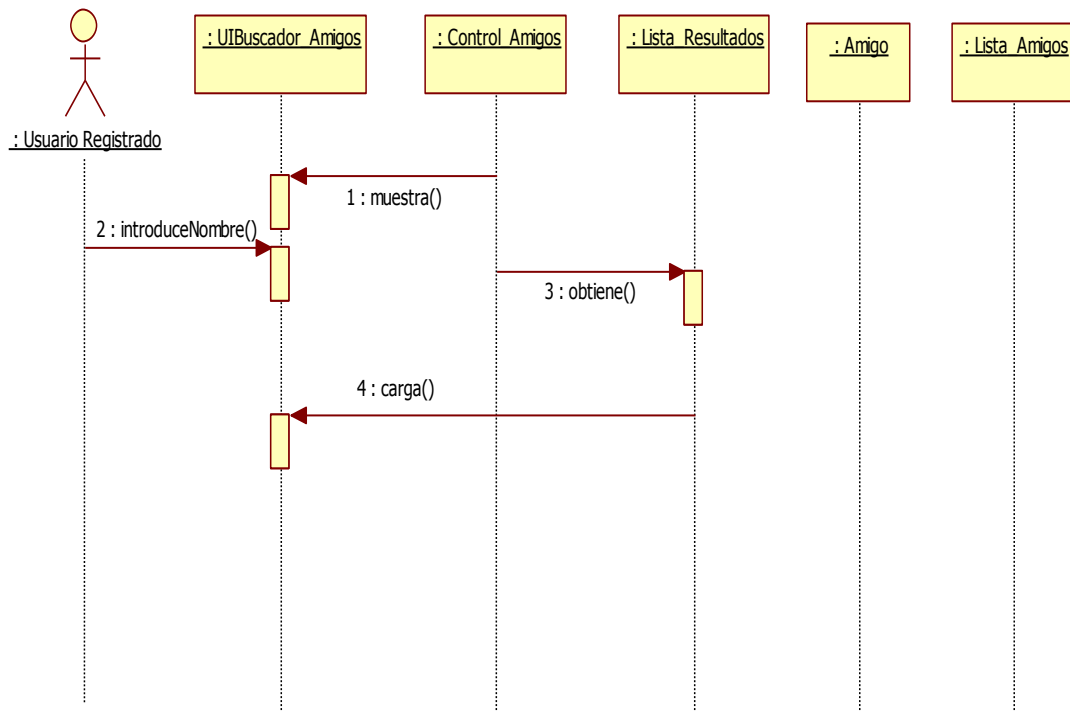


Ilustración 83: Diagrama de secuencia para el flujo alternativo del caso de uso Buscar Amigo.

❖ **Caso de Uso Compartir Contenido**

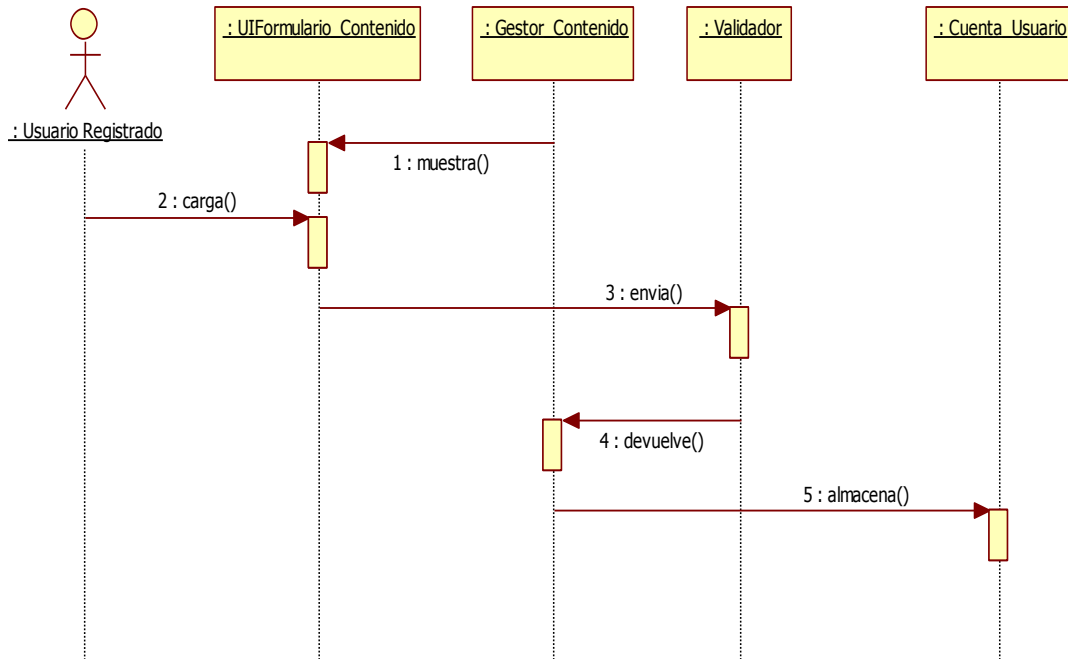


Ilustración 84: Diagrama de secuencia para el flujo normal del caso de uso Compartir Contenido.

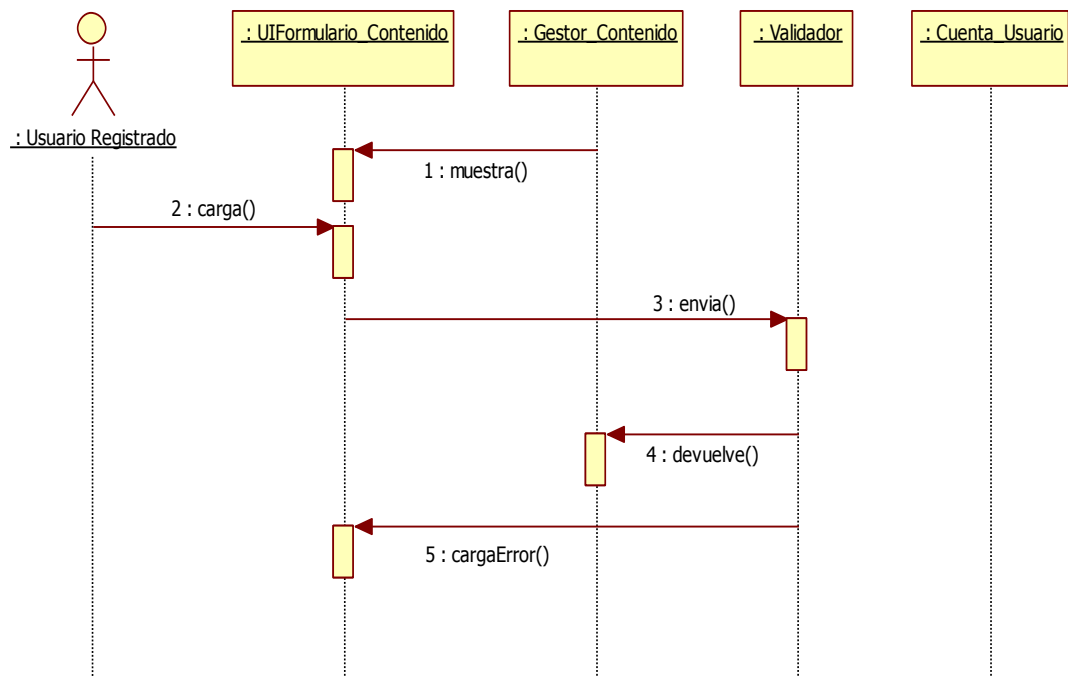


Ilustración 85: Diagrama de secuencia para el flujo alternativo del caso de uso Compartir Contenido.

❖ **Caso de Uso Eliminar Contenido**

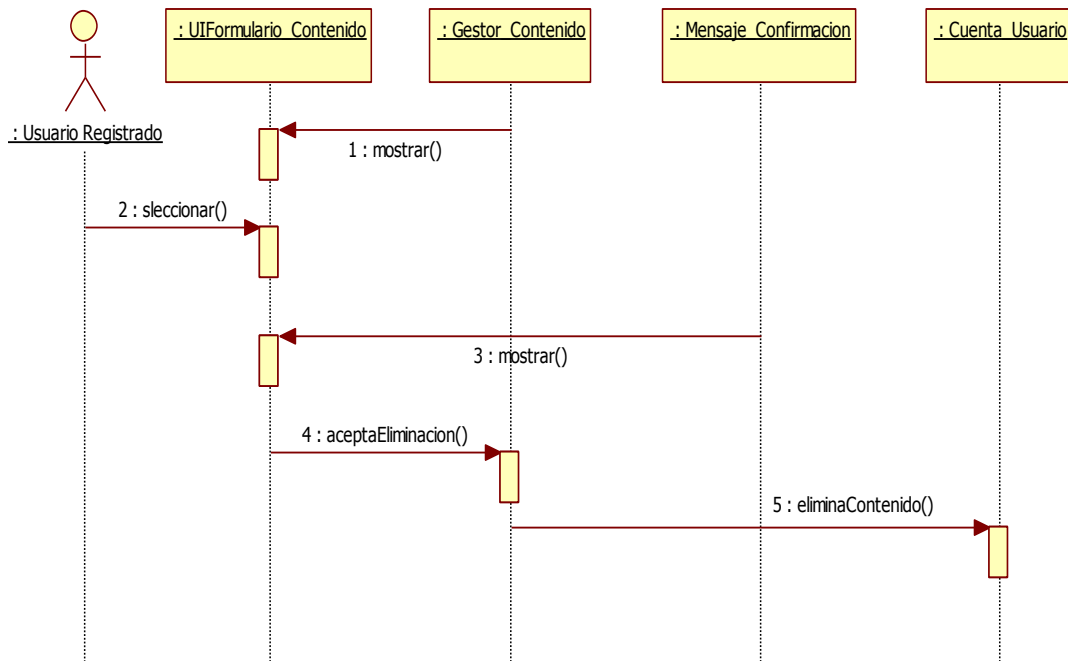


Ilustración 86: Diagrama de secuencia para el flujo normal del caso de uso Eliminar Contenido.

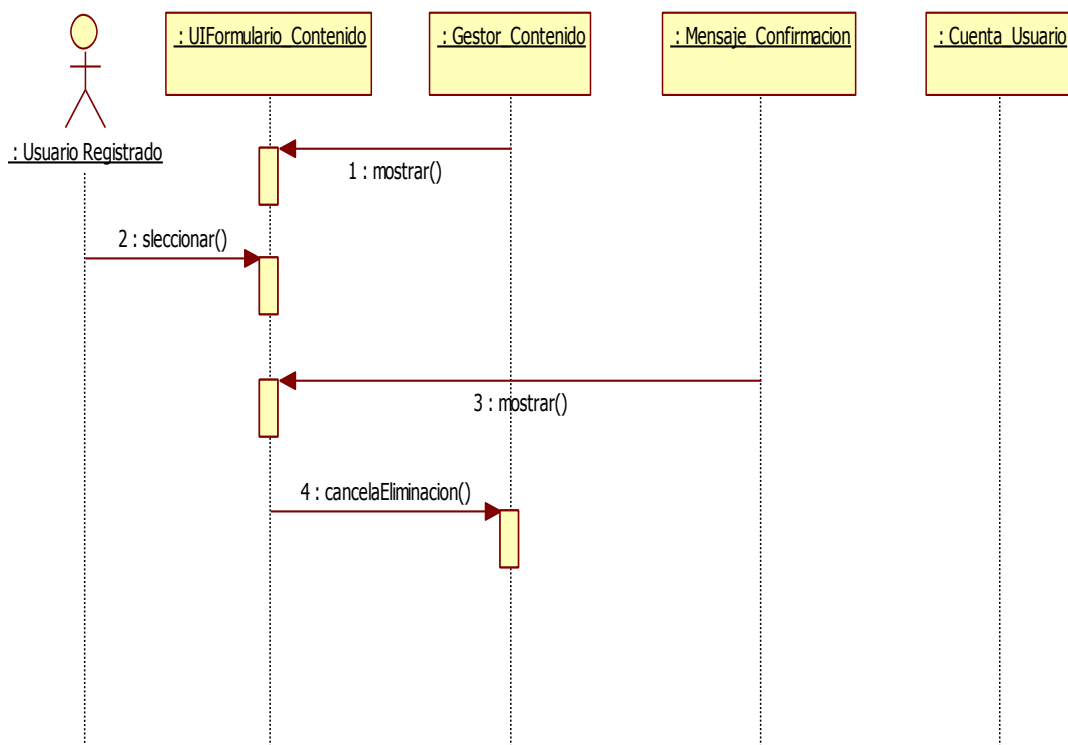


Ilustración 87: Diagrama de secuencia para el flujo alternativo del caso de uso Eliminar Contenido.

❖ **Caso de Uso Eliminar Cuenta Usuario**

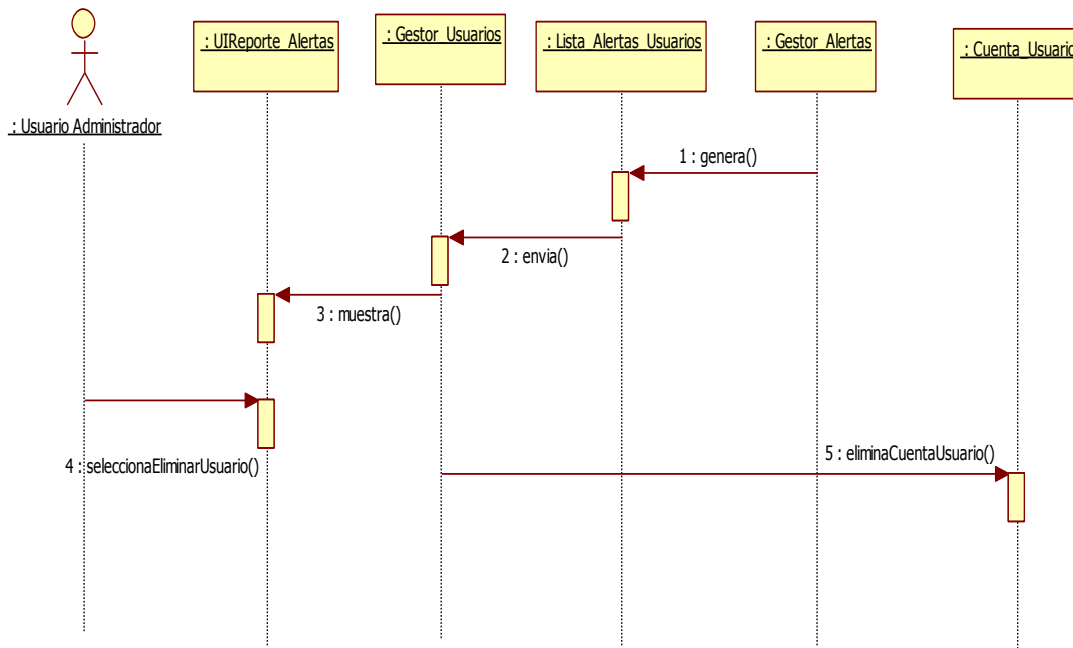


Ilustración 88: Diagrama de secuencia para el flujo normal del caso de uso Eliminar Cuenta Usuario.

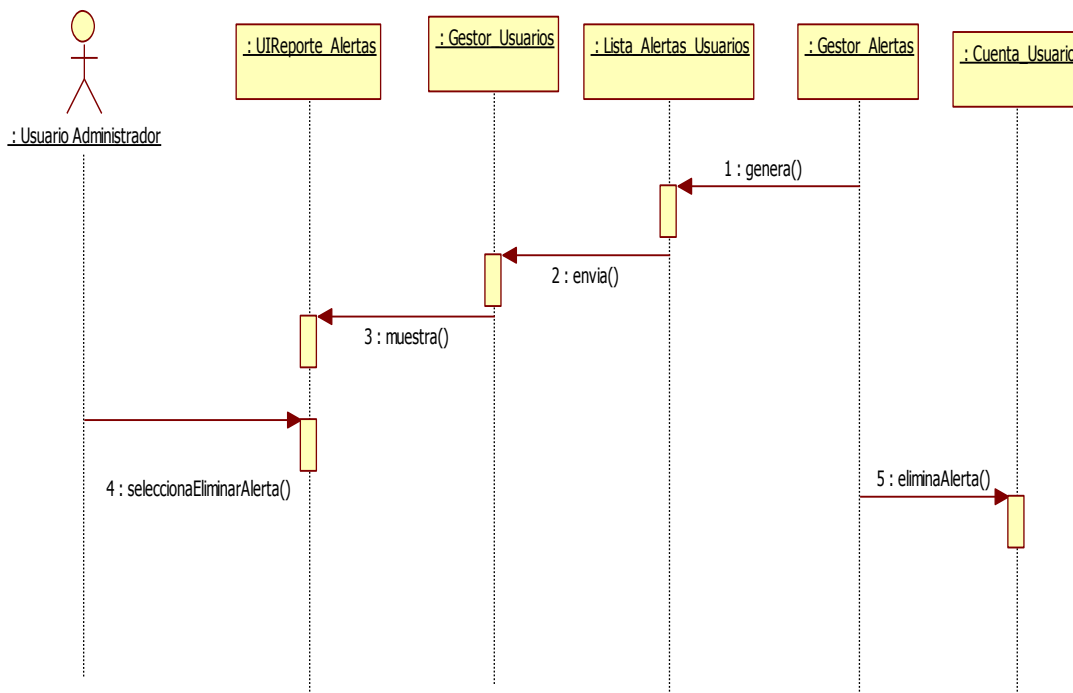


Ilustración 89: Diagrama de secuencia para el flujo alternativo del caso de uso Eliminar Cuenta Usuario.

❖ Caso de Uso Login

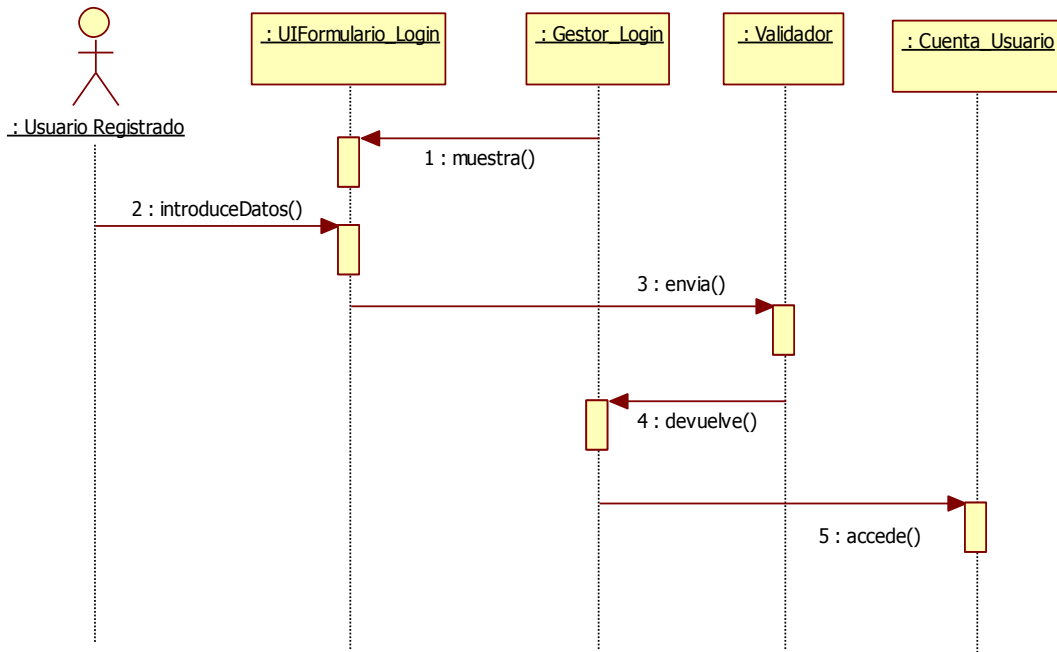


Ilustración 90: Diagrama de secuencia para el flujo normal del caso de uso Login.

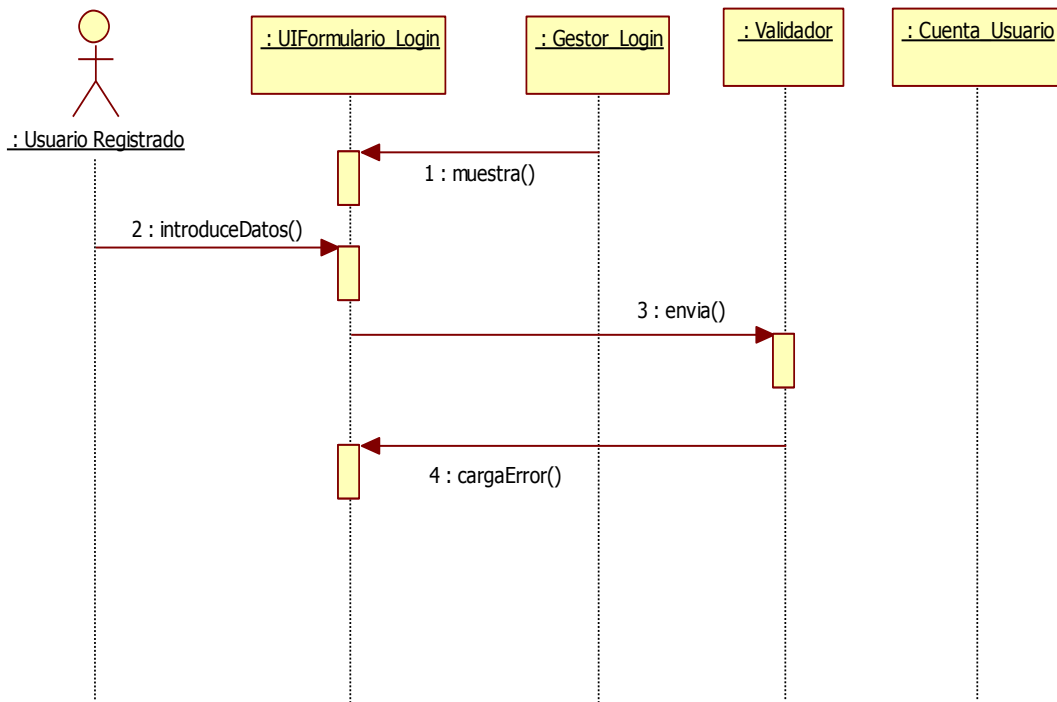


Ilustración 91: Diagrama de secuencia para el flujo normal del caso de uso Login.

❖ Caso de Uso Realizar Videollamada

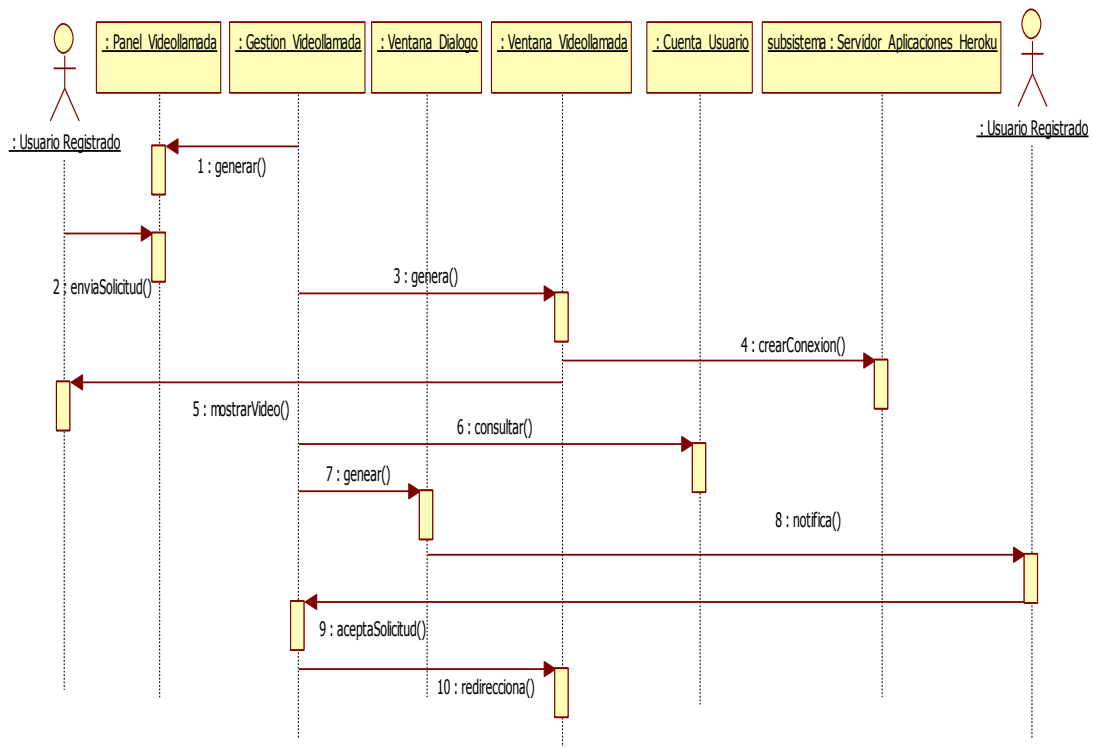


Ilustración 92: Diagrama de secuencia para el flujo normal del caso de uso Realizar Videollamadas.

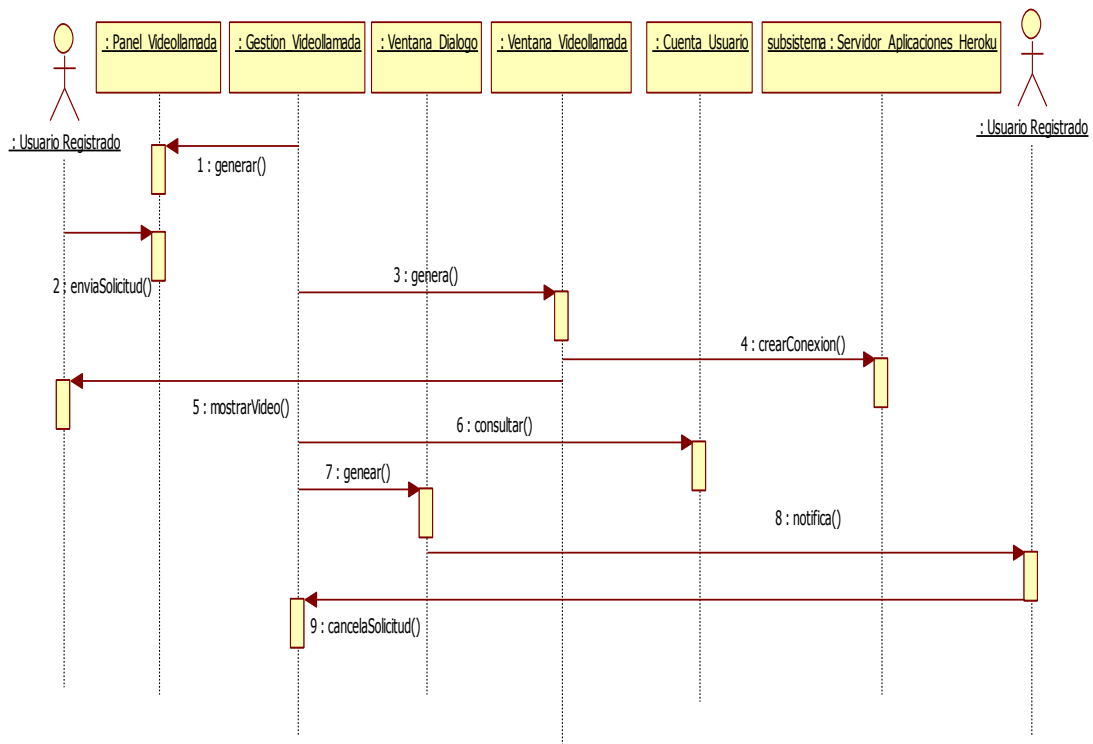


Ilustración 93: Diagrama de secuencia para el flujo alternativo del caso de uso Realizar Videollamada.

❖ **Caso de Uso Gestionar Perfil**

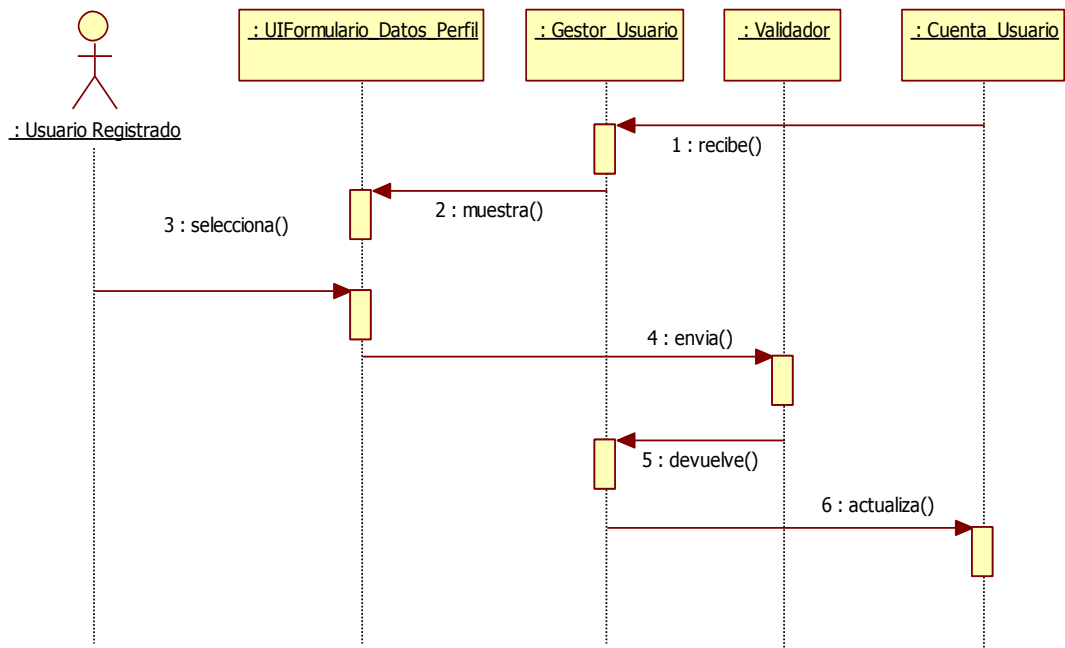


Ilustración 94: Diagrama de secuencia para el flujo normal del caso de uso Gestionar Perfil.

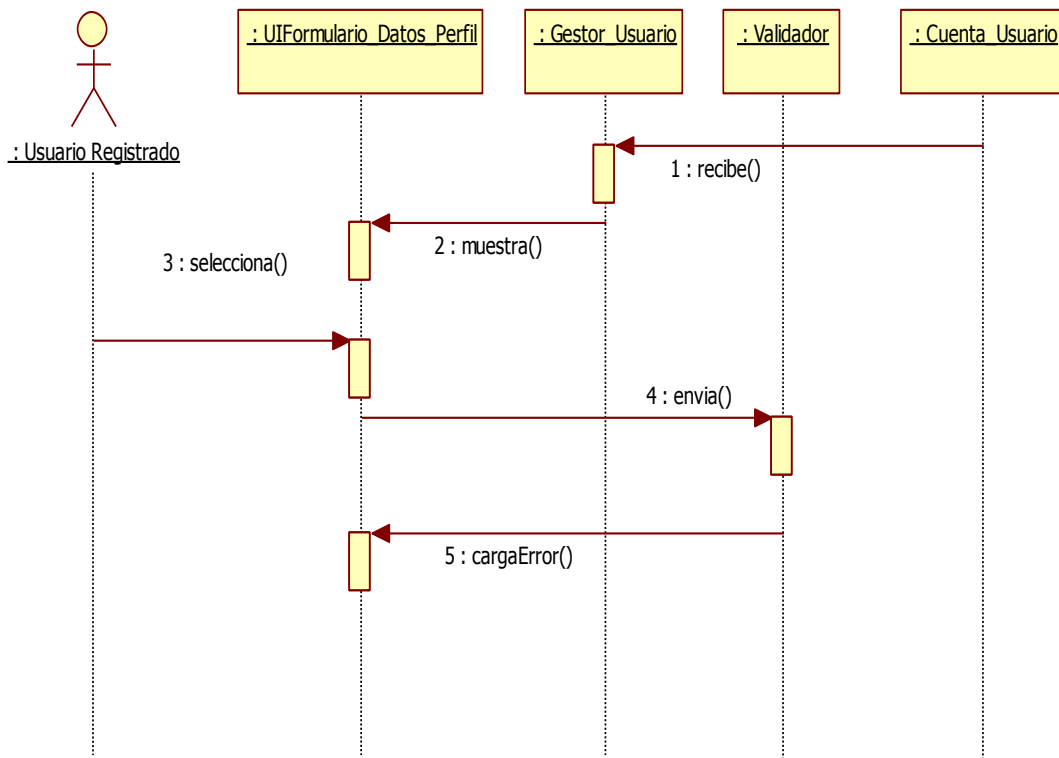


Ilustración 95: Diagrama de secuencia para el flujo alternativo del caso de uso Gestionar Perfil.

5.6 Implementación

5.6.1 Primeros pasos con CMS Elgg

Con el objetivo de desarrollar una aplicación que además de realizar videollamadas, fuera potente en funcionalidades como la gestión de usuarios, gestión de contenidos, gestión de amigos, entre otras, se eligió Elgg, que provee funcionalidades específicas para estos requerimientos.

❖ ¿Por qué elegirlo?

Elgg es un manejador de contenidos de código abierto, con plugins de gran utilidad, los cuales de manera muy potente permiten crear una red social. Se pueden desarrollar plugins que cumplan con los requerimientos de los usuarios del sitio web e incorporarlos de forma muy sencilla. Es una herramienta bastante segura ya que todos los objetos contenidos en el sitio de Elgg tienen un nivel de control de acceso configurable.

❖ Modificaciones en el CMS Elgg

Una vez cubiertas estas funcionalidades, se necesita desarrollar un plugin para que el usuario desde Elgg pueda realizar videollamadas a sus amigos. Para el desarrollo de esta funcionalidad se modificó el CMS Elgg. Utilizando funciones de jQuery, creando una nueva base de datos y a través de uso de Ajax, se desarrolla un plugin que permite:

- ❖ En el momento en que un usuario decide realizar una videollamada a alguno de sus amigos online, se escribe la petición en la base de datos.
- ❖ Después de escrita la petición en la base de datos, con una función que se ejecuta cada 5 segundos y desde el cliente de los usuarios que se encuentran conectados, lee la base de datos y detecta si algún usuario ha hecho alguna petición de videollamada.
- ❖ Una vez que lo anterior es detectado, el sistema visualiza un cuadro de diálogo desarrollado con jQuery, que simula al que muestran otros sistemas de videollamadas, para que el usuario que está siendo llamado conteste o rechace la solicitud de videollamada.

Elgg como otros manejadores de contenido, tras su instalación activa algunos plugins que vienen en el núcleo del CMS. Para lograr una mejor apariencia y mejorar algunas funcionalidades, se requirió la instalación de algunos plugins descargados del sitio web oficial de Elgg, los cuales se nombran a continuación:

❖ Invite Friends

Plugin que permite que las solicitudes de amistad se manejen tipo Facebook, es decir, que un usuario envíe una solicitud a otro y para que puedan ser amigos en la aplicación el otro debe aceptar dicha invitación. De otro modo no tendrán una relación de amistad y solo se podrá acceder a los datos que los usuarios compartan como público y a las funcionalidades generales entre usuarios.

❖ GalliStatus

El CMS a pesar de informarnos cuales son los usuarios que están en línea, no lo hace como otras redes sociales, por lo que el plugin GalliStatus permite saber el estado en el que se encuentra el usuario en la aplicación, en línea o desconectado, mostrando un icono verde sobre las fotos de perfil de los usuarios.

❖ **Pearl Premium Theme**

Plugin que permite modificar la apariencia del sitio. Tema que hace más agradable la presentación de los datos con el buen diseño que plantea y con características de las aplicaciones tradicionales.

5.6.2 **Lo más importante está por llegar. WebRTC y HTML5**

Es necesario desarrollar un sistema que pueda conectar a dos usuarios a través de sus respectivas webcams. Para esto se necesita implementar una pequeña aplicación en JavaScript y HTML5, que permita utilizar los elementos que brinda HTML5 para dar soporte al audio y el video. Además, es necesario utilizar las librerías de webRTC: MediaStream y RTCPeerConnection, que son fundamentales para la creación de un sistema de videoconferencias. Ahora bien, un sistema de videoconferencias necesita un servidor como proveedor de comunicación en tiempo real, lo cual es muy limitado si se usa Apache, esta parte de la aplicación se ha desarrollado enfocada a otro servidor, Node.js.

❖ **¿Por qué Node.js?**

Apache crea un nuevo hilo por cada conexión cliente-servidor. Esto funciona bien para pocas conexiones, pero crear nuevos hilos es algo costoso, así como los cambios de contexto. Apache funciona bien pero no es el mejor servidor para lograr máxima concurrencia. Uno de los puntos fuertes de Node.js es su capacidad de mantener muchas conexiones abiertas y esperando, por lo que su uso es ideal para que miles de usuarios estén enviando peticiones de videollamadas. Además de lo explicado anteriormente, para un sistema de videollamadas sobre HTML5 se necesita un servidor de señalización, el cual se puede construir con el módulo Sockets.io de Node.js porque las APIs de webRTC no proveen de este mecanismo.

5.6.3 **Integrar las dos aplicaciones para que parezcan una**

Tanto el usuario que realiza la solicitud de videollamada como el usuario que acepta la videollamada, son redirigidos al servidor de aplicaciones Heroku, es decir, los usuarios no deben usar las dos aplicaciones por separado, cada usuario puede usar los potenciales que ofrece Elgg y en el momento que desee realizar una videollamada el mismo CMS lo ayudará a conectarse al servidor Heroku de forma transparente.

5.7 Pruebas

Es necesario realizar pruebas para con los resultados obtenidos de las mismas poder verificar si el sistema cumple o no con los requerimientos planteados en el inicio del desarrollo. Las pruebas realizadas se basan en el funcionamiento de las APIs de WebRTC, HTML5 y Node.js. También, son ejecutadas con el objetivo de comprobar que la señalización se lleva a cabo correctamente y que dos usuarios con este servidor pueden intercambiar entre sí audio y video en tiempo real.

Luego de descargar el Node.js del sitio web oficial www.nodejs.org, el próximo paso es realizar la instalación. Una vez instalado, debe configurarse el archivo `server.js` o `app.js`. Posteriormente, se introducen los ficheros Javascript, HTML y CSS que contienen el código de la aplicación desarrollada, dentro de la carpeta del servidor Node.js. Una vez realizada toda la configuración pertinente, se inicia el servidor, lo cual se realiza de la siguiente manera (ver Ilustración 96).

```
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Jose>cd Desktop
C:\Users\Jose\Desktop>cd TFGS
C:\Users\Jose\Desktop\TFGS>cd "TFGApplication 6.0"
C:\Users\Jose\Desktop\TFGS\TFGApplication 6.0>node server.js_
```

Ilustración 96: CMD de Windows para activar servidor de Node.js.

Ya activado el servidor Node.js, solo resta abrir el navegador. Se introduce la dirección, el puerto y los parámetros requeridos (ver Ilustración 97).

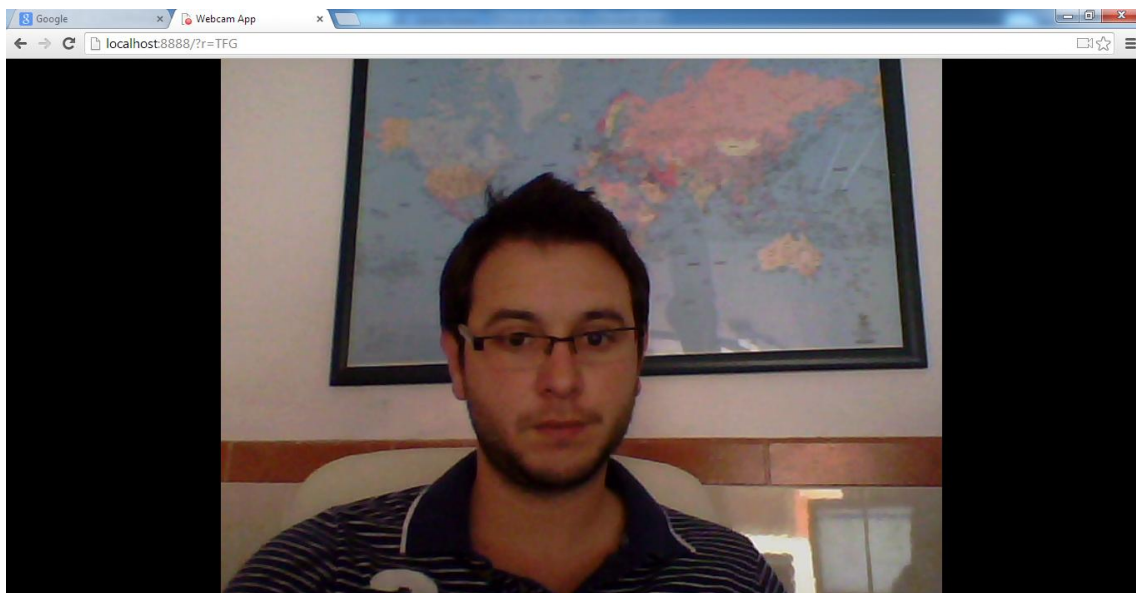


Ilustración 97: Capturando video desde la webcam a través de las APIs de WebRTC y HTML5 con Node.js como servidor de aplicaciones.

Una vez llegado a este punto y para comprobar que dos usuarios pueden intercambiar información, resta escribir exactamente la misma URL que se colocó anteriormente en otra ventana del navegador, lo cual simula que es otro usuario que está intentado conectarse con el primer usuario que creó la conexión.

Como se ha podido comprobar, el servidor funciona correctamente ya que ambos usuarios intercambian sus videos a través de las APIs de webRTC, usando los elementos de HTML5 para el manejo de multimedia, todo esto utilizando el servidor Node.js (ver Ilustración 98).

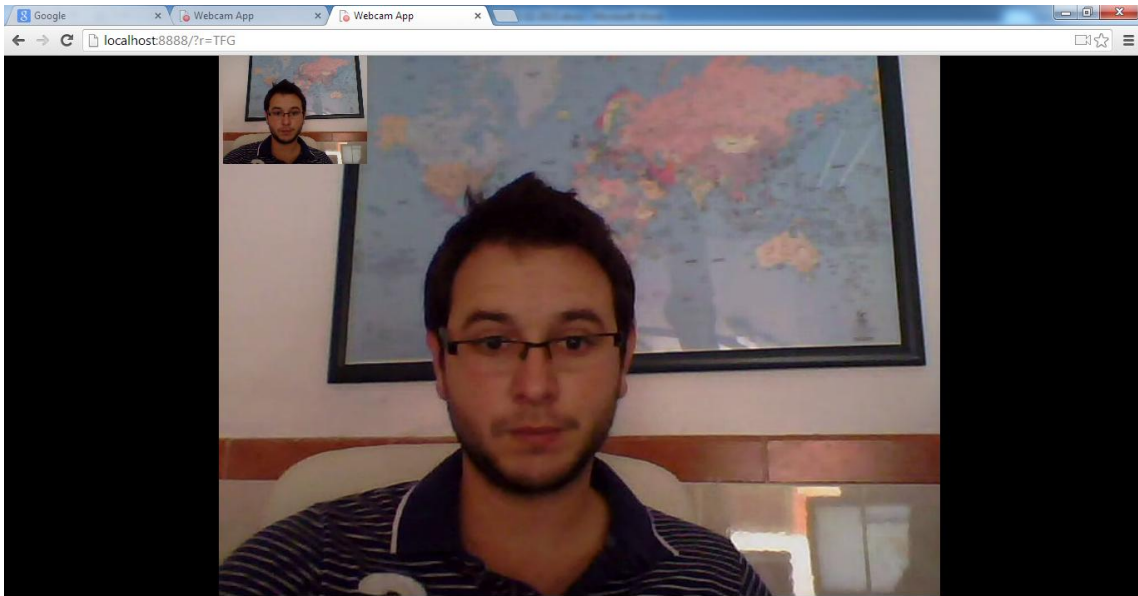


Ilustración 98: Realizando videollamada en el servidor local satisfactoriamente.

6 Conclusiones y trabajo futuro

En este Trabajo de Fin de Grado se cumplen todos los objetivos planteados al inicio y se obtienen las siguientes conclusiones:

- ❖ El uso de los sistemas de videoconferencias basados en HTML5 y WebRTC es fiable, a pesar de no estar estandarizado aún, queda demostrado con el desarrollo de la aplicación Webcam App que pueden desarrollarse sistemas potentes y de gran utilidad para los usuarios.
- ❖ El seguimiento de una metodología, específicamente Proceso Unificado de Desarrollo de Software, ha posibilitado una buena organización para la realización de este trabajo y aportado artefactos de gran utilidad.
- ❖ Con la unión de dos herramientas, en principio diferentes, se ha logrado crear una aplicación bastante completa, que a la vista de los usuarios funciona como una sola. Esto resulta beneficioso, pues si alguna de estas herramientas se encuentra en mejora constante, se puede mantener la aplicación actualizada sin afectar las otras partes.

Para el trabajo futuro, se recomienda añadir una funcionalidad nueva a esta aplicación y se propone: realizar una videoconferencia desde 2 usuarios hasta 10 usuarios, utilizando los mismos principios.

Anexo I: Competencias

Con el desarrollo de este Trabajo de Fin de Grado de la carrera Grado en Ingeniería Informática, se deben cubrir las competencias asignadas a este, las cuales son: CII01, CII02, CII04, CII18 y TFG01. A continuación, se listan cada una de ellas junto con una explicación de cómo se han cubierto.

- ❖ **CII01: capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.**

Para la realización de este trabajo fue necesario diseñar e implementar una aplicación web. De igual manera, fue necesario enfrentarse a la necesidad de tomar decisiones a la hora de la elección de un framework o tecnología a usar. Finalmente, se desarrolló una aplicación que cumple con la legislación y normativas vigentes para el desarrollo de software.

- ❖ **CII02: capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.**

Para la realización de este trabajo se realizó una correcta planificación y dirección del proyecto. El mismo consistió en el desarrollo de una aplicación web que se desplegó en un entorno real y se recopiló la valoración de los usuarios en cuanto a su utilidad. Además, se capturaron limitaciones y posibles mejoras para el futuro cercano, para aumentar la usabilidad a medida que se desarrollen funcionalidades nuevas.

- ❖ **CII04: capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.**

En el desarrollo de este documento se incluyeron en el capítulo número tres los requisitos de hardware y software necesarios para el correcto desarrollo de esta aplicación y para su despliegue. Además, se especificó el marco legal en que se ampara el presente proyecto.

- ❖ **CII18: conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.**

Para el desarrollo de este o cualquier proyecto, los desarrolladores deben estar al margen de lo legal y conocer la normativa y la regulación de la informática en el ámbito nacional, de la Unión Europea, e internacional.

- ❖ **TFG01: ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sintetizan e integran las competencias adquiridas en las enseñanzas.**

Luego de haber terminado el desarrollo del Trabajo de Fin de Grado, se debe hacer una presentación y defensa del mismo, ante el tutor y un tribunal constituidos por miembros del claustro de profesores, para cumplir con lo establecido con el TFG01. De esta manera se demuestra la consistencia en el ámbito de las tecnologías específicas de la Ingeniería Informática, entre las cuales se destacan la captura de requerimientos, análisis y diseño del software, así como la implementación, pruebas, manejo de base de datos, administración de servidores y diseños de interfaces de usuarios. Todos

estos conocimientos han sido adquiridos durante la formación que se obtuvo a lo largo de la carrera Grado en Ingeniería Informática.

Anexo II: Legislación vigente

Este Trabajo de fin de Grado se encuentra amparado por la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. Esta aplicación almacenará datos de carácter personal, como podría ser el caso de direcciones IP, de correo electrónico, nombre de las personas, localidad, número de teléfono, siempre y cuando se avise a los usuarios en la política de privacidad del sitio que la recogida de los mismos estaría en el marco de lo legal.

La presente Ley Orgánica será de aplicación a los datos de carácter personal registrados en soporte físico, que los haga susceptibles de tratamiento, y a toda modalidad de uso posterior de estos datos por los sectores público y privado.

❖ Se regirá por la presente Ley Orgánica todo tratamiento de datos de carácter personal

- a) Cuando el tratamiento sea efectuado en territorio español en el marco de las actividades de un establecimiento del responsable del tratamiento.
- b) Cuando al responsable del tratamiento no establecido en territorio español, le sea de aplicación la legislación española en aplicación de normas de Derecho Internacional público.
- c) Cuando el responsable del tratamiento no esté establecido en territorio de la Unión Europea y utilice en el tratamiento de datos medios situados en territorio español, salvo que tales medios se utilicen únicamente con fines de tránsito.

❖ A los efectos de la presente Ley Orgánica se entenderá por:

- a) Datos de carácter personal: cualquier información concerniente a personas físicas identificadas o identificables.
- b) Fichero: todo conjunto organizado de datos de carácter personal, cualquiera que fuere la forma o modalidad de su creación, almacenamiento, organización y acceso.
- c) Tratamiento de datos: operaciones y procedimientos técnicos de carácter automatizado o no, que permitan la recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias.
- d) Responsable del fichero o tratamiento: persona física o jurídica, de naturaleza pública o privada u órgano administrativo, que decida sobre la finalidad, contenido y uso del tratamiento.
- e) Afectado o interesado: persona física titular de los datos que sean objeto del tratamiento a que se refiere el apartado c) del presente artículo.
- f) Procedimiento de disociación: todo tratamiento de datos personales de modo que la información que se obtenga no pueda asociarse a persona identificada o identificable.
- g) Encargado del tratamiento: la persona física o jurídica, autoridad pública, servicio o cualquier otro organismo que, sólo o conjuntamente con otros, trate datos personales por cuenta del responsable del tratamiento.
- h) Consentimiento del interesado: toda manifestación de voluntad, libre, inequívoca, específica e informada, mediante la que el interesado consienta el tratamiento de datos personales que le conciernen.
- i) Cesión o comunicación de datos: toda revelación de datos realizada a una persona distinta del interesado.
- j) Fuentes accesibles al público: aquellos ficheros cuya consulta puede ser realizada, por cualquier persona, no impedida por una norma limitativa o sin más exigencia que, en su caso, el abono de una contraprestación. Tienen la consideración de

fuentes de acceso público, exclusivamente, el censo promocional, los repertorios telefónicos en los términos previstos por su normativa específica y las listas de personas pertenecientes a grupos de profesionales que contengan únicamente los datos de nombre, título, profesión, actividad, grado académico, dirección e indicación de su pertenencia al grupo. Asimismo, tienen el carácter de fuentes de acceso público los diarios y boletines oficiales y los medios de comunicación.

Anexo III: Manual de usuario

Webcam App es una aplicación para realizar videollamadas sencilla y fácil de usar. A continuación, se mostrarán los pasos a seguir para lograr realizar una videollamada con algún amigo que esté dado de alta en la aplicación.

Lo primero sería registrarse, para ello se debe acceder a la aplicación a través de la dirección www.webcamapp.tk (ver Ilustración 99).



Ilustración 99: Acceso al sitio web publicado en un dominio en internet.

Una vez allí, se despliega el menú que aparece en la parte superior derecha y se selecciona la opción **register** (ver Ilustración 100).

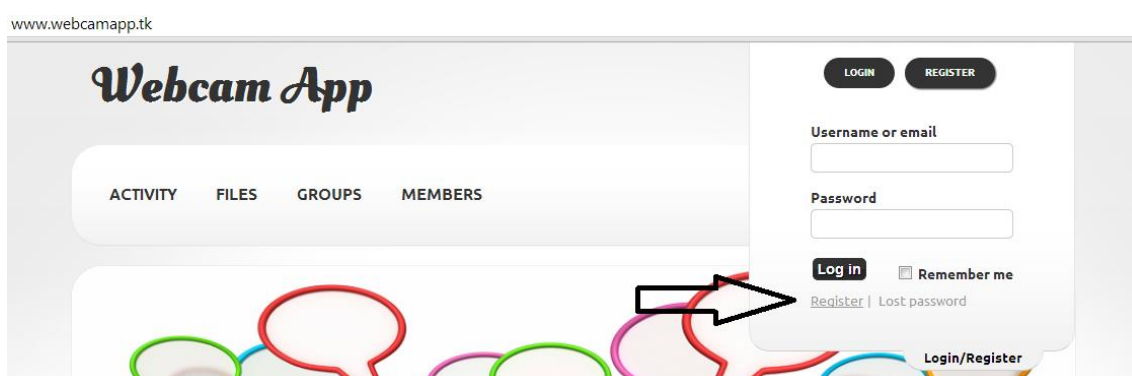
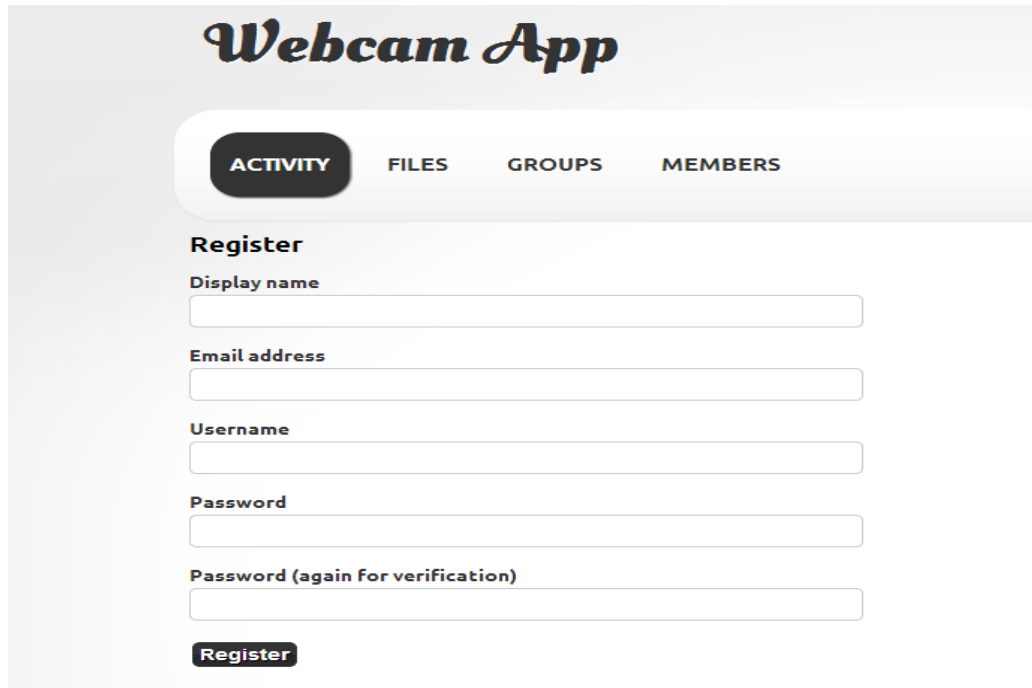


Ilustración 100: Panel para Registrarse o Iniciar Sesión.

Seleccionado esto, el sitio redirige a la página de registro donde se deben completar los siguientes campos.

- ❖ **Display Name:** es el nombre que se muestra en el perfil.
- ❖ **Username:** es el nombre de usuario que se desea tener el sitio, el cual es único en la aplicación, así que puede darse el caso que se necesite elegir otro al que se tenía pensado, si este ha sido elegido por un usuario anteriormente.

- ❖ **Email Address:** es la dirección de correo electrónico con la cual se desea registrarse en el sitio y donde se enviarán las notificaciones del mismo.
- ❖ **Password:** en este campo se debe introducir la contraseña con la cual se accederá a la cuenta que se creará tras haber finalizado el registro.
- ❖ **Password (again for verification):** este es un campo de verificación, para comprobar que se ha introducido correctamente la contraseña que se desea (ver Ilustración 101).



Webcam App

ACTIVITY FILES GROUPS MEMBERS

Register

Display name

Email address

Username

Password

Password (again for verification)

Register

Ilustración 101: Página de registro.

Cuando el registro esté finalizado, se puede empezar a hacer uso de todas las funcionalidades que brinda Webcam App, entre las cuales está la de localizar a los amigos que están registrados en el sitio. Para ello, se accede a la opción del menú **Members**, donde se muestra un listado con los miembros que están dados de alta en el sitio (ver Ilustración 102 y 103).

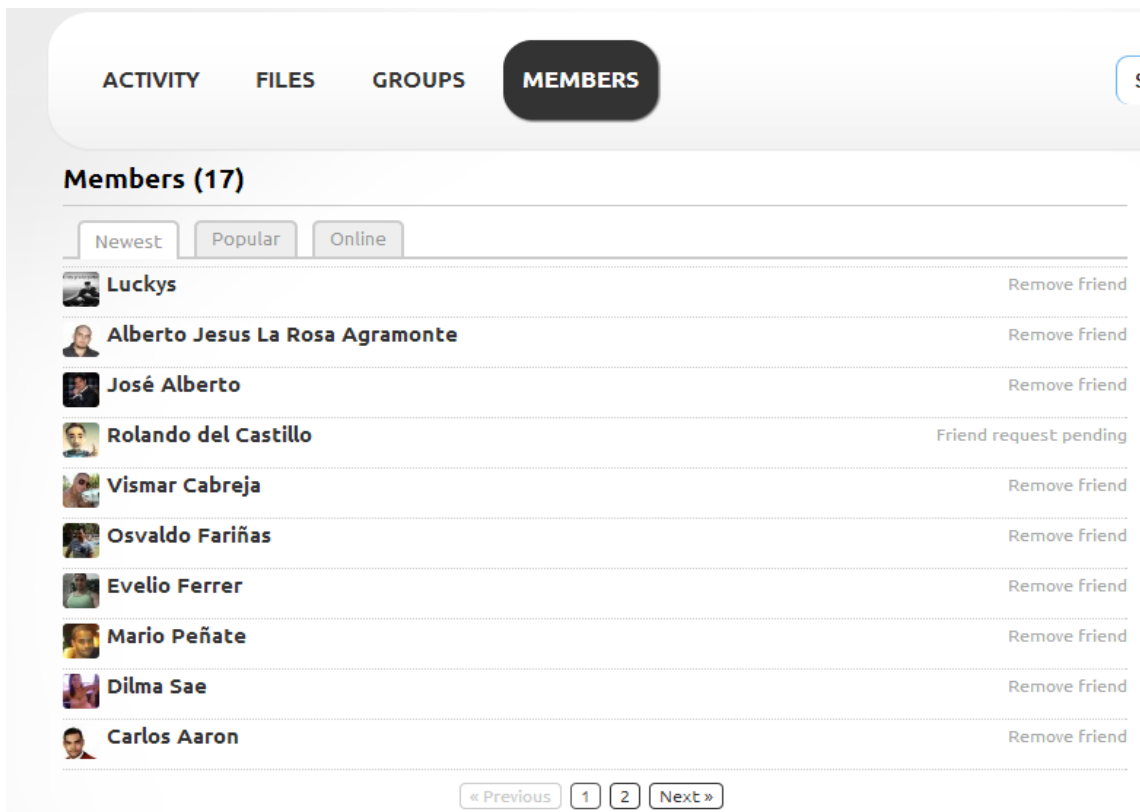


Ilustración 102: Listado de usuarios registrados.

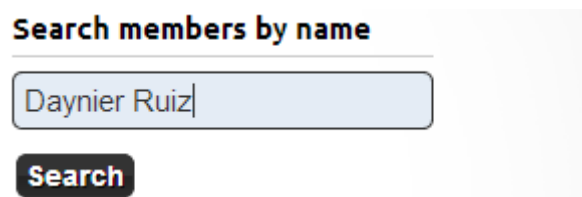


Ilustración 103: Panel de búsqueda.

Es necesario introducir el nombre del amigo que se desea buscar en el campo de texto y presionar el botón **Search**. La aplicación mostrará un listado con los usuarios encontrados con el nombre introducido, dando la opción de añadirlo como amigo en Webcam App a través del botón **Add Friend** (ver Ilustración 104).

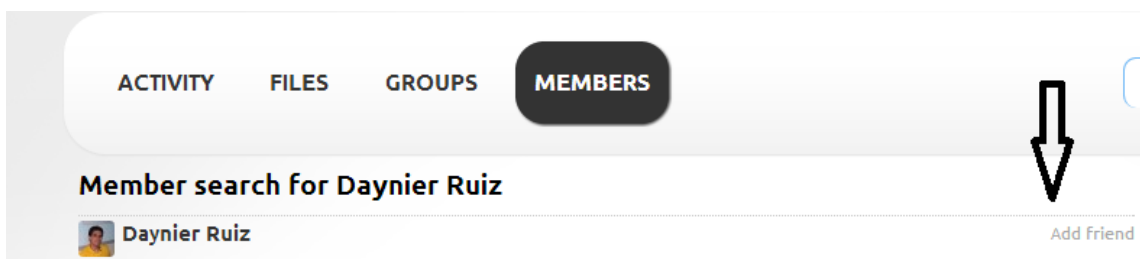


Ilustración 104: Resultados de la búsqueda de usuarios

Mientras la persona que desea llamar no acepte su solicitud de amistad, no se podrá realizar videollamadas con dicha persona (ver Ilustración 105).

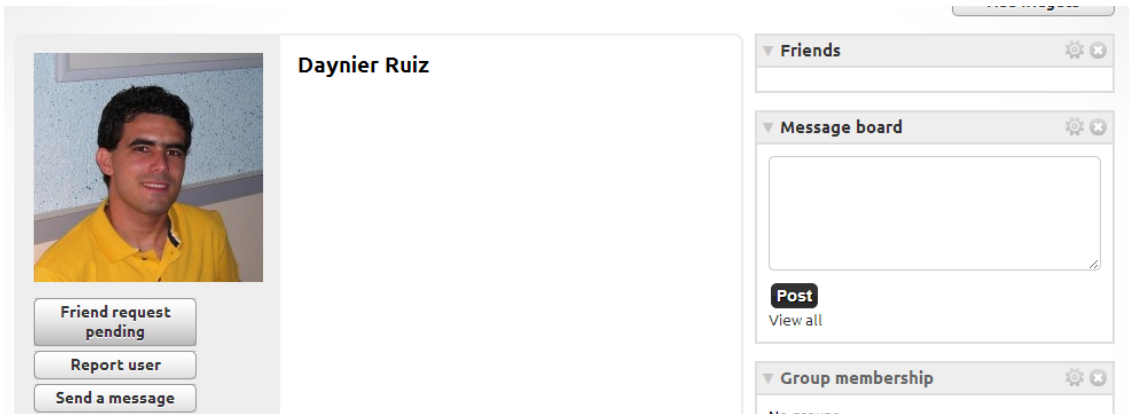


Ilustración 105: Perfil de un usuario que no ha aceptado la solicitud de amistad.

Una vez que el usuario acepte su solicitud de amistad, se tendrá disponible todas las funcionalidades incluida la de enviarle una solicitud de videollamada (ver Ilustración 106).

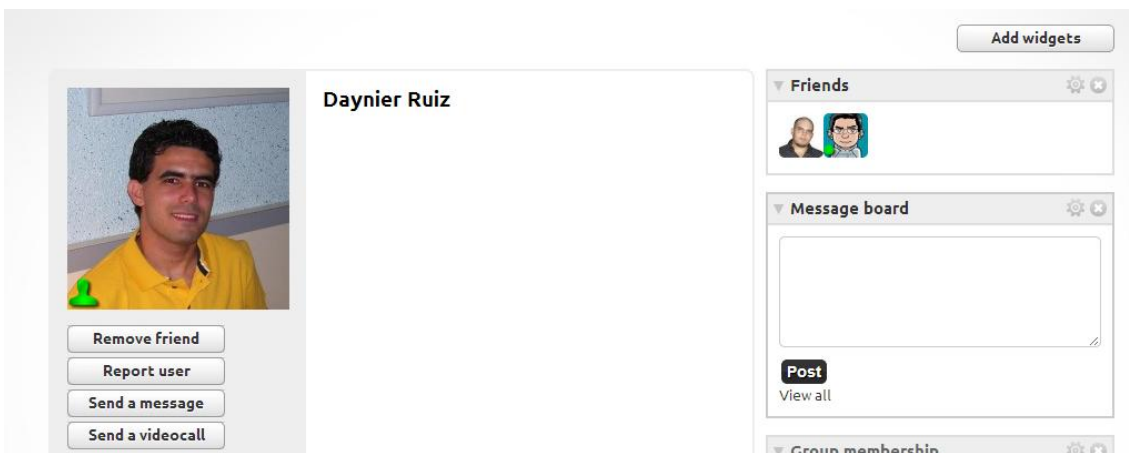


Ilustración 106: Perfil de un usuario que aceptó la solicitud de amistad

Por último, queda enviar la solicitud de videollamada y esperar que el usuario la acepte para poder comunicarse e intercambiar los videos obtenidos desde la webcam (ver Ilustración 107).

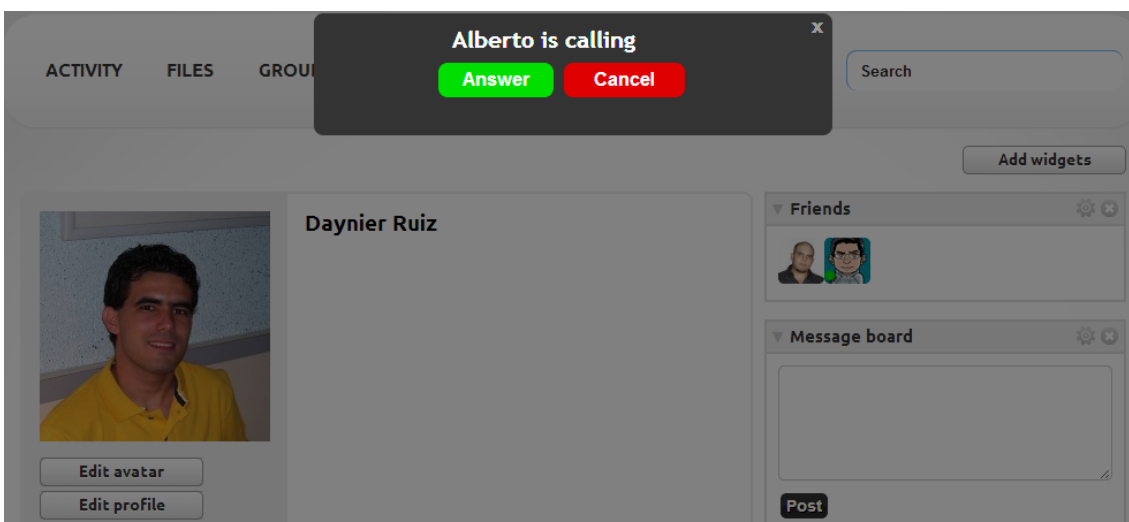


Ilustración 107: Ventana de diálogo que se muestra a los usuarios cuando reciben una solicitud de videollamadas.

Una vez que el usuario amigo acepte la solicitud la videollamada queda establecida (ver Ilustración108).

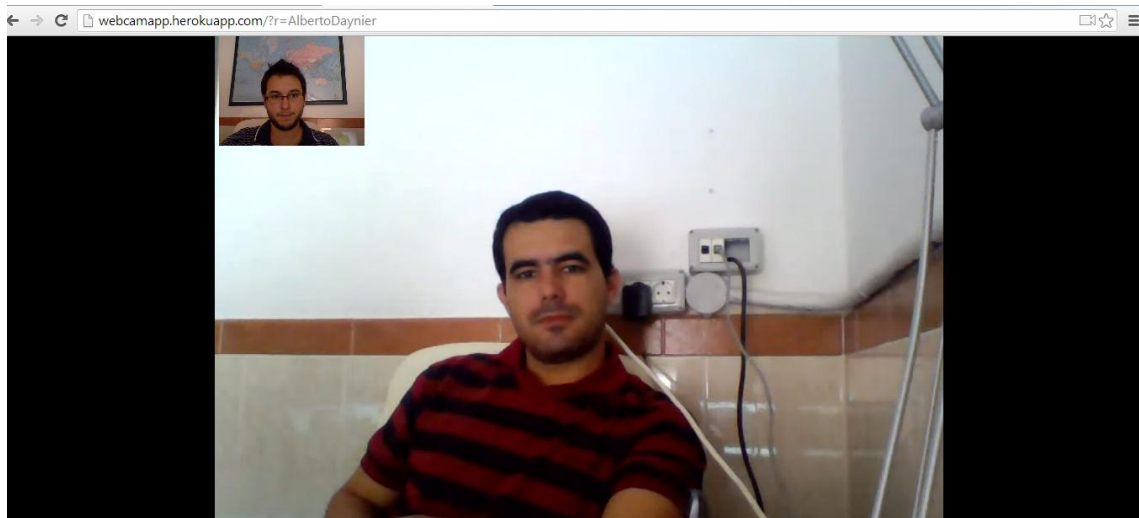


Ilustración 108: Videollamada establecida entre dos usuarios a través de la aplicación Webcam App.

Anexo IV: Entorno de Desarrollo

❖ Instalación de Elgg

Para la instalación de Elgg es necesario dos componentes esenciales, el primero es el CMS Elgg en sí, el cual se puede descargar del sitio web oficial www.elgg.org y como cualquier otro CMS necesita un entorno de desarrollo web, entre los más utilizados para Windows se tiene el Wamp y para Linux el Xamp. En este caso se utiliza Wamp.

❖ ¿Qué facilidades brinda Wamp?

Wamp provee de los cuatro elementos esenciales a la hora de construir un sitio web: un sistema operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y el soporte a un lenguaje de programación del lado del servidor (eje PHP).

Luego de tener instalado el Wamp en el ordenador, se continúa con la instalación del Elgg. Lo primero que se debe hacer es copiar la carpeta de Elgg para el servidor Wamp, dentro de la carpeta `/www` y tratar de acceder al servidor a través del navegador con la dirección <http://localhost:puerto>, en este caso el servidor web está configurado para que escuche en el puerto 88 (ver Ilustración 109 y 110).

Server Configuration
Apache Version : 2.4.2
PHP Version : 5.4.3
MySQL Version : 5.5.24

Loaded Extensions :

Core	bcmath	calendar	com_dotnet	ctype
date	ereg	filter	ftp	hash
iconv	json	mcrypt	SPL	odbc
pcre	Reflection	session	standard	mysqlnd
tokenizer	zip	zlib	libxml	dom
PDO	Phar	SimpleXML	wddx	xml
xmlreader	xmlwriter	apache2handler	mbstring	gd
mysql	mysqli	pdo_mysql	pdo_sqlite	mhash
xdebug				

Tools
phpinfo()
phpmyadmin

Your Projects
ckfinder
deportes
elgg-1.8.16

Ilustración 109: Página de inicio del servidor Wamp.

localhost:88/elgg-1.8.16/install.php

elgg

1. Welcome
2. Requirements check
3. Database installation
4. Configure site
5. Create admin account
6. Finished

Welcome

Installing Elgg has 6 simple steps and reading this welcome is the first one!

If you haven't already, read through the installation instructions included with Elgg (or click the instructions link at the bottom of the page).

If you are ready to proceed, click the Next button.

Next

Ilustración 110: Página de instalación de Elgg.

Como los demás CMS, Elgg necesita la creación de una base de datos, la cual se crea con el gestor de base datos MySQL, que brinda Wamp. La base de datos se crea de la siguiente manera: se accede al servidor a través del navegador y se selecciona la opción phpmyadmin, ubicada en la parte inferior de la página de inicio del Wamp (ver Ilustración 111).

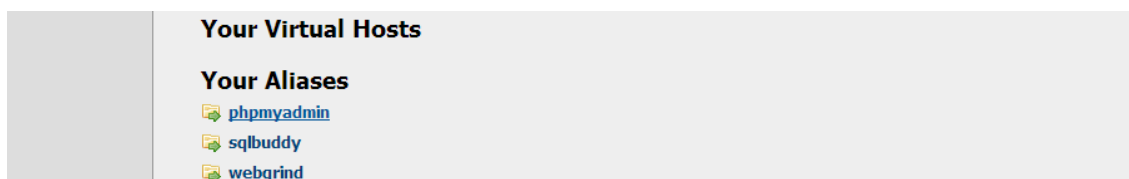


Ilustración 111: Acceso a phpmyadmin a través de la página de inicio de Wamp.

De esta manera se accede a phpMyAdmin, que no es más que una página en php para gestionar la base de datos a través del navegador (ver ilustración 112).

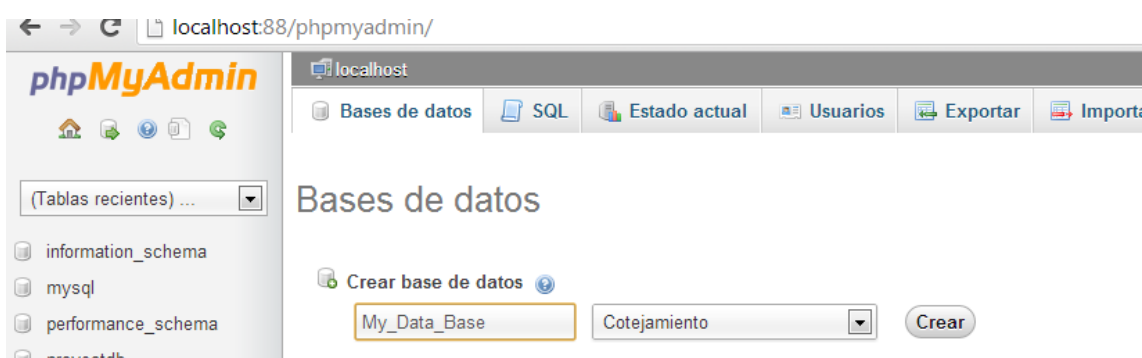


Ilustración 112: Creación de base de datos en phpMyAdmin.

De esta manera se crea la base de datos, ahora solo resta crear un usuario con todos los privilegios sobre ella.

Durante el proceso de instalación de Elgg, solicita proporcionar el nombre de la base de datos y el nombre de usuario con todos los privilegios, para que al completar la instalación se puedan generar todas las tablas y datos que trae consigo Elgg.

Para darle las funcionalidades adecuadas a la aplicación Webcam App, se utilizan algunos plugins específicos. Estos plugins ayudan a que el usuario alcance un alto nivel de usabilidad, pues aportan funciones similares a las que tienen otras redes sociales conocidas a escala mundial, la más destacada entre ellas es Facebook.

Para instalar un plugin en la aplicación, debe copiarse el plugin dentro de la carpeta **/mod** del directorio raíz del CMS y luego acceder al panel de administración del sitio web en el apartado **Plugins** y seleccionar la opción **Activate** (Ver Ilustración 113 y 114).

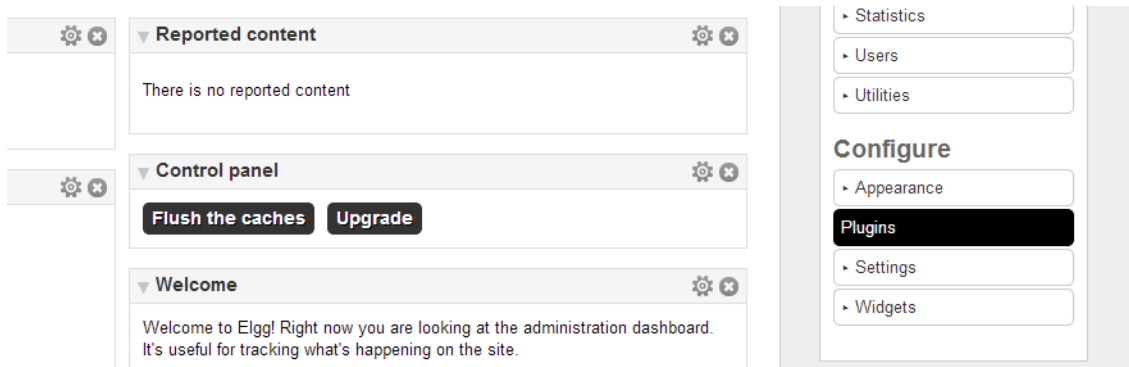


Ilustración 113: Acceso al panel de administración de plugins

Plugins

<p>Blog 1.8</p> <p>Adds simple blogging capabilities to your Elgg installation. Author: Core developers - http://elgg.org/ more info</p>	<p>Down Bottom</p> <input type="button" value="Activate"/>
<p>Bookmarks 1.8</p> <p>Adds the ability for users to bookmark internal and external sites. Other users can then comment on the bookmarks. Author: Core developers - http://www.elgg.org/ more info</p>	<p>Top Up Down Bottom</p> <input type="button" value="Activate"/>

Ilustración 114: Proceso de activación de un plugin.

❖ Publicación de la aplicación en un servidor de hosting en Internet

Luego de haber instalado el CMS correctamente, activados y configurados los plugins necesarios, se procede a la publicación del sitio web en un servidor de hosting en internet. Para esto se escoge 000webhost, el cual es un servidor de hosting gratuito y fácil de usar.

Lo primero es crearse una cuenta en la página web www.000webhost.com, luego acceder al panel de control del sitio web. Una vez allí, se crea una base de datos de igual manera que se hizo para el sitio que se crea en local, se exporta la base de datos que se encuentra en el ordenador y se importa a la base de datos del servidor de hosting.

Una vez hecho esto, queda indicarle al sitio web los datos de la nueva base de datos, puesto que ya no estará local, ahora debe encontrarse en el servidor 000webhost, como en todos los CMS, se localiza el archivo **settings.php**, que es el que contiene entre otras cosas los datos para la conexión a la base de datos y se modifican los valores por los nuevos parámetros.

Para finalizar la publicación del sitio, se copia a través de un cliente ftp o a través del que brinda 000webhost en su página web todos los archivos del sitio local a la carpeta **public_html** en el servidor de hosting.

Ahora bien, el sitio web está publicado en un servidor de hosting en internet pero aún no se puede navegar por él, pues es necesario un nombre de dominio. Para esto el usuario debe dirigirse al sitio www.dot.tk, en el cual se registra y puede obtener un nombre de dominio de forma gratuita si está disponible (ejemplo: www.webcamapp.tk). Luego de haber obtenido el nombre de dominio, solo resta entrar al panel de configuración e indicarle la dirección de Host Name que brinda 000webhost (ver Ilustración 115).

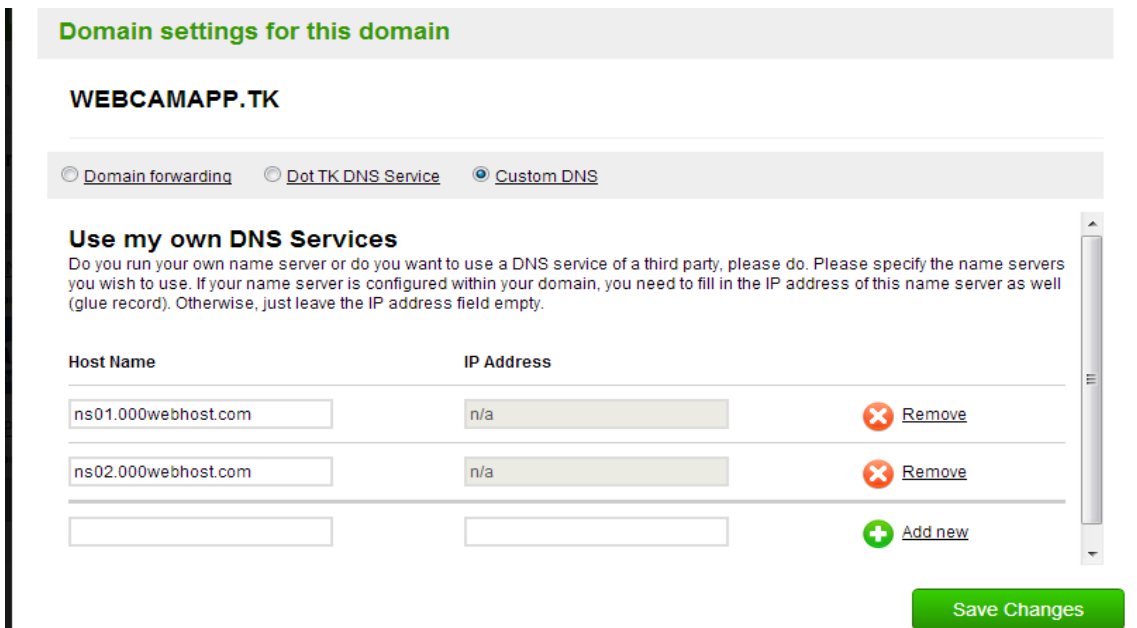


Ilustración 115: Panel de administración del dominio en dot.TK

Llegado a este punto, el sitio web se encuentra publicado en un servidor de hosting y con un nombre de dominio a través del cual se puede acceder a él. Se coloca el nombre de dominio creado en el navegador y se puede entrar fácilmente (ver Ilustración 116).



Ilustración 116: Webcam App desplegado en el servidor de hosting

❖ **Configuración de Node.js del lado del servidor**

Node.js no es como otros servidores web, necesita una configuración manual del fichero JavaScript que se ejecuta en el servidor llamado app.js ó server.js. Algo llamativo puede ser que si existe otro servidor web, por ejemplo el Apache escuchando en el puerto 88, se debe introducir un puerto diferente para que escuche el Node.js (ejemplo 8888) (ver Ilustración 117).

```

var static = require('node-static');
var http = require('http');
var file = new(static.Server)();
var app = http.createServer(function (req, res) {
  file.serve(req, res);
}).listen(8888);

var io = require('socket.io').listen(app);

```

Ilustración 117: Configuración del fichero server.js en el servidor.

En este caso se va a trabajar con el módulo Sockets.io por lo que es necesaria la configuración del mismo.

Con las salas (rooms) de Sockets.io, se puede enviar mensajes a un grupo de usuarios determinados, que serían los clientes que están conectados a ellas. Con el siguiente fragmento de código, se define con quien será el intercambio de mensajes. Con la función broadcast, se puede especificar quien se desea que reciba los mensajes que el cliente envíe, para este caso los mensajes se envían y se reciben por los usuarios que estén conectados a la misma sala (room), aunque bien podría configurarse para todas las salas de la aplicación. Si se quisiera mandar un mensaje de multidifusión a todos los usuarios, para eso se utiliza *socket.broadcast.emit ('message', message)* (ver Ilustración 118).

```

socket.on('message', function (message,room) {
  socket.broadcast.to(room).emit('message', message);
});

```

Ilustración 118: Definición de destinatario de los mensajes.

❖ Configuración del lado del cliente

Una vez configurado el servidor, es necesaria una configuración en el cliente para que pueda comunicarse con el mismo y poder intercambiar los mensajes, una sencilla configuración podría ser esta (ver Ilustración 119).

```

if (room != undefined) {
  socket= io.connect();
  socket.emit('create or join', room);
  socket.on('created', function (room){
    isInitiator = true;
  });
}

```

Ilustración 119: Configuración en el cliente.

Una vez que ambos usuarios estén conectados a la misma sala (room), es donde entra la API de webRTC, RTCPeerConnection. Para lograr que ambos logren intercambiar el video que obtienen desde su webcam, hay que vincular estas funciones con socket.io, el cual se encarga del envío de mensajes. Con la siguiente función, dos usuarios pueden establecer sus remoteDescription con la description que reciban del otro usuario y así posteriormente poder intercambiar sus videos obtenidos desde la webcam. Con la API MediaStream, a través de la función que está actualmente disponible en Google Chrome, Mozilla Firefox y Opera, definida como *getUserMedia()*, de una forma muy sencilla se puede obtener multimedia e incrustarla en el elemento video de HTML5.

Para aplicaciones publicadas en internet o para la comunicación con usuarios que estén fuera de una red local, es necesario añadir servidores que permiten que un host final pueda descubrir la dirección IP pública si se encuentra detrás de un NAT, durante la implementación del API `RTCPeerConnection` (ver Ilustración 120 y 121).

```
var pc_config = { 'iceServers': [{ 'url': 'stun:stun.l.google.com:19302' } ] };
```

Ilustración 120: Añadiendo servidores.

```
function createPeerConnection() {  
  try {  
    peerConnection = new RTCPeerConnection(pc_config);  
    peerConnection.onicecandidate = handleIceCandidate;  
    peerConnection.onaddstream = handleRemoteStreamAdded;  
    peerConnection.onremovestream = handleRemoteStreamRemoved;  
  } catch (e) {  
    alert('Cannot create RTCPeerConnection object.');    return;  
  }  
}
```

Ilustración 121: Creación de una `PeerConnection`.

❖ **APIs de webRTC utilizadas en la aplicación.**

- ❖ *MediaStream* (`getUserMedia`): La cual se utiliza para acceder a los datos de los usuarios (ejemplo: el video o el audio obtenidos desde la webcam).
- ❖ *RTCPeerConnection*: Esta API permite que a dos personas comunicarse directamente a través del navegador, usando un canal de señalización, el cual no está especificado en esta API.

❖ **Potencialidades de HTML5**

En la actualidad, la mayoría de los navegadores muestran videos a través de un plugin (ejemplo flash), claramente no es una solución que resuelva todos los problemas, ya que los navegadores pueden tener diferentes plugins, lo cual conlleva a que los videos incrustados en la web no estén siempre disponibles para todos los usuarios. Con la aparición de HTML5, se define un nuevo elemento que especifica un método estándar para incrustar un vídeo / película en una página web: el elemento `<video>`. A continuación se muestran los navegadores que soportan HTML5 (ver Ilustración 122).



Ilustración 122: Navegadores que soportan HTML5

❖ Actualmente, hay 3 formatos de video soportados para el elemento <video>: MP4, WebM y Ogg:

Navegador	MP4	WebM	Ogg
Internet Explorer	SÍ	NO	NO
Chrome	SÍ	SÍ	SÍ
Firefox	NO Versión: Firefox 21 de Windows 7, Windows 8, Windows Vista, y Android ahora soportan MP4	SÍ	SÍ
Safari	SÍ	SÍ	NO
Opera	SÍ	SÍ	SÍ

Tabla 8: Formato de video que soportan los navegadores actualmente.

Bibliografía

1. **Pazmiño, Fernando.** Monografías. [En línea] [Citado el: 10 de 9 de 2013.] <http://www.monografias.com/trabajos/videoconferencia/videoconferencia.shtml>.
2. **webRTC org.** webRTC org. [En línea] Google, 2011-2012. [Citado el: 20 de 9 de 2013.] <http://www.webrtc.org/>.
3. **Wikipedia.** Wikipedia org. [En línea] 1 de 10 de 2013. [Citado el: 5 de 10 de 2013.] <http://es.wikipedia.org/wiki/WebRTC>.
4. **Google.** Youtube. [En línea] GoogleDevelopers, 19 de 5 de 2013. [Citado el: 6 de 10 de 2013.] <http://www.youtube.com/watch?v=p2HzZkd2A40>.
5. —. Google. [En línea] Google, 2012. [Citado el: 9 de 10 de 2013.] <https://www.google.com/intl/es/chrome/browser/>.
6. **Elgg.** Elgg. [En línea] Elgg org, 2012. [Citado el: 10 de 10 de 2013.] <http://elgg.org/>.
7. **Node JS.** Node JS. [En línea] Joyent, Inc, 2011. [Citado el: 9 de 25 de 2013.] <http://nodejs.org/>.
8. **The jQuery Foundation.** jQuery. [En línea] 2013. [Citado el: 20 de 10 de 2013.] <http://jquery.com/>.
9. **The Apache Software Foundation.** Apache HTTP Server Project. [En línea] The Apache Software Foundation, 2012. [Citado el: 15 de 9 de 2013.] <http://httpd.apache.org/>.
10. **Oracle Corporation.** MySQL. [En línea] Oracle Corporation, 2013. [Citado el: 20 de 10 de 2013.] <http://www.mysql.com/>.
11. —. NetBeans IDE. [En línea] Oracle Corporation, 2013. [Citado el: 7 de 9 de 2013.] <https://netbeans.org/>.
12. **Mozilla Developer Network.** [En línea] Mozilla Developer Network, 2005-2013. [Citado el: 10 de 9 de 2013.] <https://developer.mozilla.org/es/docs/JavaScript>.
13. **W3C World Wide Web Consortium.** W3C. [En línea] W3C World Wide Web Consortium, 2013. [Citado el: 12 de 9 de 2013.] <http://www.w3.org/html/logo/>.
14. —. W3C. [En línea] W3C World Wide Web Consortium, 2013. [Citado el: 20 de 9 de 2013.] <http://www.w3.org/Style/>.
15. **The PHP Group.** PHP. [En línea] The PHP Group, 2001-2013. [Citado el: 28 de 10 de 2013.] <http://php.net/>.
16. Ecured. [En línea] Ecured, 2013. [Citado el: 22 de 10 de 2013.] http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo.
17. **Torossi, Gustavo.** *Proceso Unificado de Desarrollo de Software*.

18. **Larman, Craig.** *UML y Patrones.* s.l. : Prentice Hall, 2003 .