Universidad de Valladolid

E.U. DE INFORMÁTICA (SEGOVIA)

Grado en Ingeniería Informática de Servicios y Aplicaciones

Portal web para comunicaciones seguras

Alumno: Rodrigo Jurado Villalobos

Tutor: José Ignacio Farrán Martín

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Descripción del proyecto

1.1. Introducción

El proyecto "Portal web para comunicaciones seguras" o *"SecureCom"* es un proyecto teórico-práctico de investigación, enfocado en obtener un mayor conocimiento del funcionamiento de los protocolos y funciones de seguridad criptográfica así como explorar las limitaciones y buscar nuevas vías de implementar la seguridad de las comunicaciones. Esto se conseguirá a través del desarrollo de una aplicación que dé un uso en un entorno "real" de estos protocolos de seguridad.

Esta idea viene de la preocupación ante la inseguridad de la red actual, en la que los paquetes viajan por la red conteniendo los mensajes en texto plano, de forma que cualquier usuario de la red capaz de capturar estos paquetes puede leer su contenido, comprometiendo así la privacidad de toda comunicación.

Aprovechando el trabajo realizado para la asignatura de "Protocolos y Comunicaciones Seguras", en el que se desarrollaron un conjunto de protocolos y métodos criptográficos, este proyecto emplea dicho conjunto de herramientas criptográficas desarrolladas por mi mismo siguiendo las especificaciones de cada función criptográfica, evitando confiar en paquetes criptográficos desarrollados por terceros.

La principal característica que diferencia esta aplicación de otros del mismo estilo es la fuerte carga de responsabilidad en la seguridad que se otorga al lado cliente, minimizando el tránsito de claves por la red, así como su ámbito web que evita la necesidad de descargar e instalar una aplicación en el cliente, consiguiendo así que siempre se utilice la última versión y permitiendo que sea independiente del sistema operativo del usuario.

1.2. Descripción general del proyecto

1.2.1. Objetivos

Este proyecto tiene una orientación teórico-práctica, cuyos objetivos se centran en el estudio de los protocolos ciptográficos en un entorno de aplicación "real", y no en el desarrollo de un producto funcional cerrado para su posterior explotación o comercialización . Además, se pretende estudiar las posibilidades y limitaciones en el ámbito de la seguridad de las tecnologías de programación para clientes web, como JavaScript junto con las últimas APIS de HTML5. Desde el punto de vista teórico se busca:

- Obtener un mayor conocimiento de los protocolos criptográficos en un entorno de uso "real"
- Estudiar la posibilidad de la seguridad web basada en el cliente
- Estudiar las posibilidades de las últimas tecnologías web para el ámbito de seguridad informática
- Buscar alternativas o ampliaciones al uso de https para la seguridad web

Desde el punto de vista de la aplicación se busca:

- Proporcionar un canal de comunicación privado y seguro
- Multiplataforma sin necesidad de descarga de software adicional (acceso web mediante navegador)

1.2.2. Entorno de aplicación

En el marco tecnológico actual hay gran cantidad de aplicaciones de comunicación, sin embargo son pocas las que tienen un enfoque en la privacidad y seguridad. A continuación revisaremos las más importantes.

- **WhatsApp:** El ejemplo más inmediato de aplicación de comunicación actual es la aplicación para *smartphones* es Whatsapp. Esta aplicación permite a los usuario realizar una comunicación con otros usuarios individualmente o crear grupos de usuarios para conversaciones grupales. Sin embargo, no fue hasta finales del año 2014 que su equipo de desarrollo comenzó a tomarse en serio la seguridad e incluyó un sistema de cifrado de los mensajes, habiendo estado siendo transmitidos en plano hasta entonces, de forma que un usuario malicioso podría obtener dichos mensajes sin mayor complicación.

- **Telegram:** La alternativa más conocida del anteriormente mencionado Whatsapp el Telegram, el cual ofrece las mismas características salvo que tuvo un enfoque hacia la seguridad desde el inicio, incluyendo cifrado de los mensajes y otras funcionalidades como las conversaciones privadas que son destruidas a elección del usuario

- **Cryptocat:** Tal vez la aplicación con más parecido a este proyecto, se trata de una aplicación de código libre y gratuita enfocada en la comunicación cifrada. Esta aplicación se encuentra en formato de complemento para diversos navegadores, así como una app para iOS y OSX que también permite enviar fotos y archivos.

- **MeWe:** Se trata de una red social web similar a facebook, con la diferencia de que se centra en dar al usuario una herramienta de comunicación enfocada a la privacidad, permitiendo limitar tu presencia a ojos del resto de la comunidad fuera de tus círculos, crear post privados... Además alegan que no realizan ningún tipo de análisis, rastreo o compartición de los datos personales.

  La mayor diferencia de las aplicaciones anteriores con este proyecto se centra en la aparente incompatibilidad entre sitios web y cifrado. Las aplicaciones anteriores que incluyen cifrado al margen de https (Whatsapp, Telegram y Cryptocat) son aplicaciones que requieren la descarga e instalación de un software, por lo que restringen más la compatibilidad entre plataformas. En el caso de MeWe, vemos que proporciona privacidad, mientras que su seguridad reposa en el uso de https.

1.2.3. Metodología

Proceso de desarrollo

Para el proceso de desarrollo he decidido escoger un **enfoque iterativo**. Siendo este un proyecto web de cierta innovación, el distribuir el desarrollo de los objetivos del proyecto en iteraciónes consecutivas va a permitir llevar un mayor control sobre el avance del proyecto y la interacción de sus diversas partes.

Debido también a tratarse de un proyecto web, el enfoque de desarrollo no será orientado a objetos, sino que se seguirá un enfoque orientado a eventos, más adecuado a las características de las aplicaciones web.

Pese a no emplear ningún framework, durante el desarrollo se pretende alcanzar la mayor separación posible entre la interfaz y los procesos de de datos, diferenciando claramente las vistas de los servicios de datos del servidor.

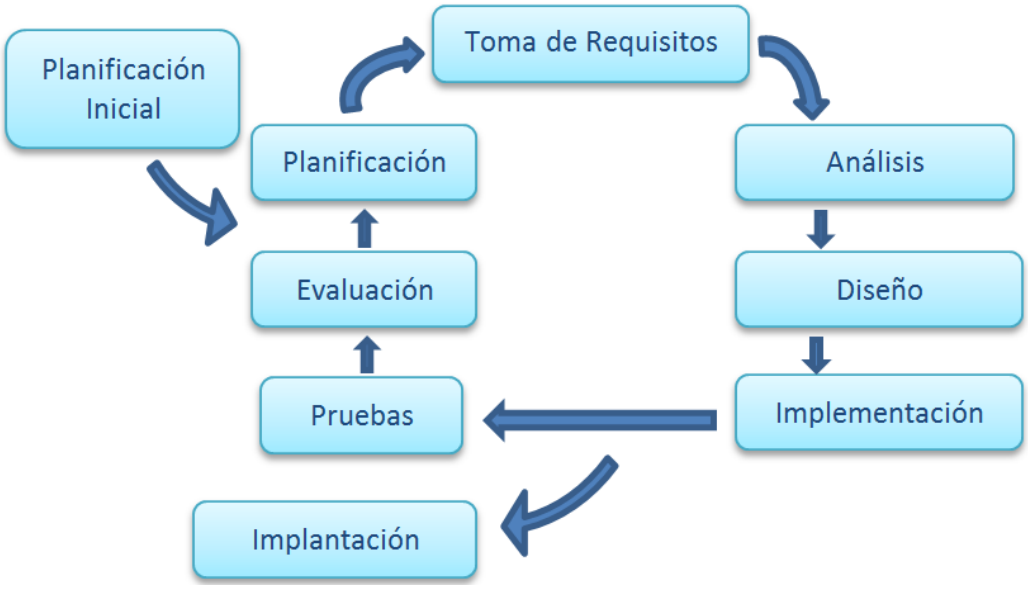

Figura 1.1: Modelo de desarrollo iterativo

Estimación temporal y de costes

Para la planificación temporal y de costes he empleado dos acercamientos. El primero se basa en la estimación del esfuerzo mediante **puntos de función** y su transformación en lineas de código para finalmente calcular el coste mediante **COCOMO II**. Sin embargo, la dependencia de esta última fase sobre el lenguaje de programación empleado (que en el caso de este proyecto engloba no solo un lenguaje, sino varios) me impulsa a emplear otro método de estimación para poder realizar una comparación.

El segundo método consiste en la estimación por **puntos de Caso de Uso**, la cual se basa en un análisis previo de los casos de uso de la aplicación. Este método permite realizar una estimación del esfuerzo al margen de las líneas de código, aunque posee la desventaja de no reflejar con todo detalle la complejidad de los procesos, lo cual podría llevar a una mala estimación en un proyecto como este.

Análisis

El proceso de análisis tendrá tres partes principalmente:

- **Determinación de la visión y el alcance**: esta primera etapa es el primer paso antes de comenzar a realizar ninguna otra tarea del proyecto. Consiste en definir los objetivos, funciones principales y limitaciónes

del proyecto que se va a llevar a cabo. Una vez que tengamos claro el proyecto que vamos a desarrollar podemos comenzar con el resto de tareas.

- **Identificación de casos de uso**: el siguiente paso del análisis consiste en identificar los actores que van a interactuar con nuestro sistema y dichas interacciónes, definiendo la respuesta del sistema ante estos casos de uso sin entrar en mucho detalle.

- **Análisis de requisitos**: a partir de los casos de uso identificados se refinan los requisitos funcionales de la aplicación, así como otros requisitos no funcionales que deban tenerse en cuenta para el diseño de la aplicación.

Diseño

El proceso de diseño consistirá principalmente en especificar las secuencias de mensajes que se intercambiarán los diferentes elementos del sistema para cada acción del usuario o evento del sistema, además de definir la estructura del sitio web y la organización de los subsistemas de la aplicación.

1.3. Alcance de la aplicación

El proyecto se trata de una aplicación de comunicación textual segura entre usuarios. A continuación definimos las principales características de esta, así como aquellas funciones que se excluyen por diseño:

1.3.1. Características principales

1. **FE-1: Gestión de usuarios**. El sistema permitirá el registro de usuarios controlado por los administradores, para su posterior identificación en la aplicación.

2. **FE-2: Gestión de grupos**. El sistema permitirá crear grupos de usuarios y administrar dichos grupos a su líder.

3. **FE-3: Mensajería entre usuarios**. El sistema permitirá el intercambio de mensajes de texto entre usuarios en forma de diálogo o mensaje grupal en tiempo real.

4. **FE-4: Transferencia de archivos**. El sistema permitirá a los usuarios adjuntar archivos a sus mensajes de texto.

5. **FE-5: Seguridad**. El sistema poseerá las capacidades criptográficas para asegurar la privacidad de las comunicaciónes así como evitar la suplantación de identidad.

6. **FE-6: Internacionalización**. La aplicación ofrecerá varios idiomas para mostrar la interfaz de usuario.

7. **FE-7: Acceso**. La aplicación será accesible mediante cualquier navegador web compatible.

8. **FE-8: Tiempo real**. La aplicación será capaz de mantener informado al usuario del estado de sus comunicaciones en tiempo real.

1.3.2. Limitaciones y exclusiones

Se tendrá que atender a ciertas limitaciones así como excluir algunas funciónes posibles dentro del marco de la comunicación entre usuarios.

1. **LI-1: Comunicación en formato de audio**. El sistema no soportará servicios de voz sobre IP.

2. **LI-2: Mensajería pública**. El sistema no permitirá enviar mensajes de forma pública accesibles por cualquier miembro del sistema.

3. **LI-3: Tamaño de archivos**. El sistema no permitirá adjuntar archivos de tamaño mayor de 15MB.

1.3.3. Consideraciones de desarrollo

Adicionalmente, este proyecto pretende innovar centrando gran parte de su seguridad en el lado del cliente, lo cual permite la transmisión cifrada de los datos en todo momento y reduce drásticamente las posibilidades de suplantación de identidad.

1.4. Cuestiones de implementación

1.4.1. Lenguajes de programación

Para este proyecto se emplearán los siguientes lenguajes:

- **HTML5+CSS**: toda la interfaz del usuario se diseñará empleando este estándar .

- **JavaScript**: más que un simple apoyo, este proyecto tiene una grán carga de proceso en el lado del cliente, y esta se llevará a cabo empleando JavaScript. Todos los protocolos de seguridad que se comunicarán con el servidor se controlarán mediante este lenguaje.

- **Java**: el lado del servidor se desarrollará empleando el lenguaje Java, el cual posee la potencia necesaria para los procesos criptográficos que empleará la aplicación. También nos permite llevar dicha potencia al lado cliente mediante applets.

1.4.2. Frameworks

A pesar de que hay potentes frameworks para el desarrollo web basados en Java, la falta de experiencia con ellos ha llevado ha escoger un esquema de desarrollo Java puro. Sin embargo, para facilitar el desarrollo de la interfaz, se ha empleado el framework de HTML+CSS+JavaScript de **Bootstrap**. Este posee muchos componentes de interfaz predefinidos y un sistema de clases que facilitan la creación de la interfaz.

1.4.3. Tecnologías

Para el desarrollo de este proyecto se han escogido algunas tecnologías reseñables del entorno web que se detallan a continuación:

- **WebSockets**: La especificación de HTML5 ha traído consigo una serie de APIs para javascript de gran utilidad, entre las que se encuentra WebSockets. Se trata de un API que permite trasladar la tecnología de sockets al ámbito de javascript, permitiendo realizar una comunicación bidireccional y full-duplex entre el navegador y el servidor. Esto permite realizar varias conexiones sobre una misma conexión TCP y da mayor versatilidad a la hora de realizar operaciones asíncronas.

- **Local Storage**: Esta API de HTML5 permite almacenar datos en el lado del cliente sin la necesidad del uso de cookies y sin la brecha de seguridad que supondría permitir el acceso al sistema de archivos desde javascript. Al contrario que las cookies, los datos nunca se transmiten entre cliente y servidor, no tienen caducidad y el limite de capacidad es bastante superior (entre 5 y 10 MB contra los 4KB de las cookies). El almacenamiento de los datos se realiza por dominio, no pudiendo ser accedido desde cualquier otro dominio que no sea aquel al que pertenecía la página que guardó los datos en el Local Storage, lo que lo hace más robusto contra ataques maliciosos.

- **WebWorkers**: El API de HTML5 WebWorkers es una tecnología que permite el empleo de múltiples hilos dentro del esquema javascript. Hasta la fecha, todo el procesamiento de javascript se realizaba en un único hilo, lo que causaba que los procesos largos pudieran llegar a bloquear el navegador por completo. El API de WebWorkers remedia esto permitiendo crear *"Workers"*, hilos separados de javascript que se definen en archivos separados y realizan procesos en respuesta a llamadas que se realizan desde el hilo principal y devuelven un valor resultante. Estos procesos se ejecutan en un entorno separado, por lo que no tienen acceso a las variables del hilo principal.

- **JQuery**: La librería de JavaScript JQuery posee una gran cantidad de herramientas que facilitan la interacción de JavaScript con el árbol DOM de a página web, así como otros procesos como AJAX.

- **JSP**: esta tecnología permite añadir algo de dinamicidad a las páginas html. Su funcionamiento es similar al de PHP, incrustando código Java entre el HTML, de forma que el servidor genere el HTML resultante de las operaciónes Java antes de servirlo al usuario. Esta tecnología va a facilitar el desarrollo multilenguaje de la aplicación.

- **Tomcat8**: es el servidor Java más empleado actualmente. Un detalle importante es el empleo de la última versión, Tomcat8, la cual es compatible con la tecnología de WebSockets a diferencia de Tomcat7.

1.4.4. Herramientas

Las herramientas que se emplearan para el desarrollo de este proyecto serán las siguientes:

- **Linux Mint**: si bien no es una herramienta, creo que es importante destacar que todo el proceso de desarrollo de este proyecto ha sido llevado a cabo empleando este sistema gratuito y libre, lo cual va a condicionar las herramientas empleadas y el despliegue de la propia aplicación.

- **Eclipse**: eclipse es un IDE enfocado en Java gratuito y de código libre. Es el entorno más potente actualmente para el desarrollo Java y la familiaridad con él lo hacen la opción más adecuada para emplearlo en este proyecto.

- **Brackets**: para el desarrollo de la parte puramente web (HTML, JavaScript, CSS...) la herramienta Brackets posee una gran cantidad de utilidades y complementos que facilitan el desarrollo y la depuración.

- **Texmaker**: esta aplicación ofrece un apoyo a la construcción de documentos con el lenguaje LaTeX.
- **ArgoUML**: se trata de una aplicación gratuita y libre para el dibujo de diagramas UML basada en Java.

1.5. Arquitectura

1.5.1. Arquitectura lógica

La arquitectura lógica de la aplicación se plantea como una arquitectura por capas clásica, es decir, una arquitectura cliente-servidor. Por un lado tenemos un cliente, el cual se comunicará mediante el uso de un navegador web con el servidor, el cual le servirá las vistas tras procesar los JSP procedentes junto con la lógica de cliente encapsulada en los componentes JavaScript.

Dicho cliente, a través de la lógica de los JavaScript, se comunicará con el servidor para acceder a los servicios del servidor, que contendrán los procesos y el acceso a los datos de la BD.

1.5.2. Arquitectura física

La arquitectura física planteada inicialmente es la más básica, compuesta únicamente por el cliente y un servidor que englobaría tanto la lógica como la base de datos. Sin embargo es posible separar la lógica de la base de datos en servidores separados para mejorar el rendimiento y la seguridad.

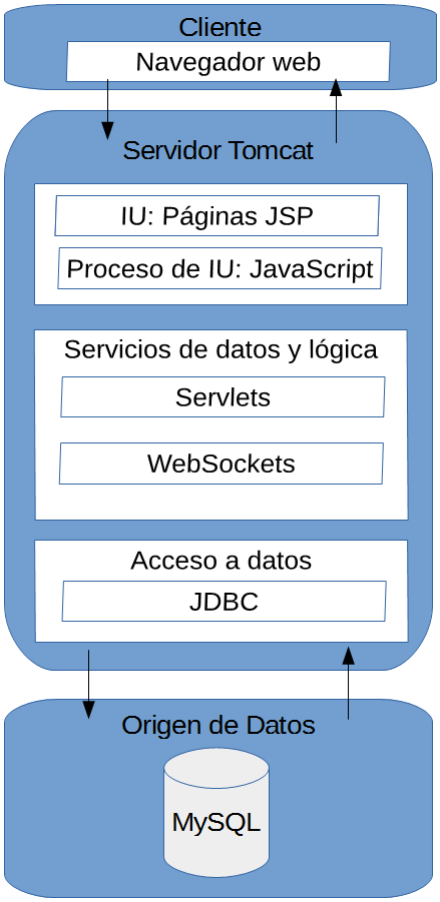

Figura 1.2: Arquitectura lógica

1.6. Planificación

1.6.1. Estimación del esfuerzo

Para realizar una planificación efectiva en primer lugar haremos una estimación del esfuerzo, para lo que emplearemos dos métodos cuyos resultados posteriormente compararemos.

Puntos de Función + COCOMO II

Para obtener los puntos de función no ajustados, primero debemos identificar los elementos pertenecientes a cinco clasificaciones y asignarles un nivel de complejidad en función de ciertos parámetros (vease la sección 5.1.1):

- **Entradas externas**:
  1. Datos de registro de usuario: complejidad Baja.
  2. Datos de login de usuario: complejidad Baja.
  3. Contraseña de seguridad de claves: complejidad Baja.
  4. Fichero de claves importado: complejidad Baja.
  5. Mensaje de diálogo: complejidad Baja.
  6. Mensaje de grupo: complejidad Baja.
  7. Secreto de grupo: complejidad Baja.
  8. Archivo adjunto a un mensaje: complejidad Baja.
  9. Nombre de nuevo grupo: complejidad Baja.
- **Salidas externas**:
  1. Mail de activación de usuario: complejidad Baja.
  2. Clave RSA de usuario: complejidad Baja.
  3. Estado de diálogos activos: complejidad Baja.
  4. Estado de grupos activos: complejidad Baja.
  5. Invitaciones a diálogo: complejidad Baja.
  6. Invitaciones a grupo: complejidad Baja.
  7. Solicitudes de registro: complejidad Baja.
  8. Usuarios de la aplicación: complejidad Baja.
  9. Grupos del usuario: complejidad Baja.

10. Miembros de grupo: complejidad Media.
11. Mensajes diálogo: complejidad Baja.
12. Mensajes de grupo: complejidad Baja.
13. Secretos de grupo: complejidad Baja.
14. Archivos adjuntos a mensajes: complejidad Baja.
15. Backup de archivo de claves: complejidad Baja.

- **Consultas externas**:

  1. Búsqueda de usuarios: complejidad Media.
  2. Búsqueda de grupos: complejidad Baja.
  3. Búsqueda de usuarios no miembros: complejidad Media.

- **Archivos lógicos internos**:

  1. Usuarios: complejidad Baja.
  2. Mensajes de diálogo: complejidad Baja.
  3. Grupos: complejidad Baja.
  4. Mensajes de grupo: complejidad Baja.
  5. Secretos: complejidad Baja.
  6. Archivos adjuntos a mensajes: complejidad Baja.

- **Archivos de interfaz externos**:

  Ninguno

A continuación asignamos valores numéricos en puntos de función para cada elemento anteriormente identificado en función de su complejidad siguiendo las tablas de valor correspondientes (vease la sección 5.1.2):

A partir de estos puntos de función no ajustados continuación debemos obtener los puntos de función ajustados mediante un factor de ajuste. .

Tipo de Función	Cantidad por complejidad			PF por complejidad			Suma
	Baja	Media	Alta	Baja	Media	Alta	
Entradas externas	9	0	0	27	0	0	**27**
Salidas externas	14	1	0	56	5	0	**61**
Consultas externas	1	2	0	3	8	0	**11**
Archivos lógicos internos	6	0	0	42	0	0	**42**
Archivos de interfaz externa	0	0	0	0	0	0	**0**
Total de puntos de función no ajustados							**141**

Cuadro 1.1: Cálculo de los puntos de función no ajustados

Factores de ajuste	**Influencia (0 a 5)**
Comunicación de datos	5
Procesamiento distribuido	4
Rendimiento	2
Configuración del equipamiento	0
Volumen de transacciones	4
Entrada de datos on-line	2
Diseño para la eficiencia del usuario final	3
Actualización de datos on-line	3
Procesamiento complejo	4
Reusabilidad	2
Facilidad de implementación	0
Facilidad de operación	2
Múltiples localizaciónes	0
Facilidad de cambios	2
Nivel de influencia	**33**

Cuadro 1.2: Nivel de influencia para el cálculo del factor de ajuste

Factor de ajuste=(Nivel de influencia*0,01)+0,65

FA=(33*0,01)+0,65=0,98

Empleando este valor de ajuste, podemos calcular los puntos de función ajustados con los que luego calcularemos el esfuerzo.

Puntos de función ajustados=Factor de ajuste*Puntos de función no ajustados

PFA=0,98*141=138,18

Una vez obtenidos los puntos de función ajustados empleamos COCOMO II para obtener el esfuerzo. El proyecto se puede definir como **Semiacoplado**, puesto que desarrolla una cierta innovación técnica. Teniendo esto en cuenta emplearemos los valores correspondientes (vease la sección 5.1.3) para calcular el esfuerzo nominal.

Para estimar las líneas de código correspondientes a los puntos de función calculados deberíamos utilizar datos históricos de la empresa para obtener el valor más fiable. Sin embargo eso no es posible en este caso, por lo que acudiremos a un valor medio calculado estadísticamente basado en la experiencia de muchas empresas (Bibliografía-[2] *Tabla de puntos de función por lenguaje*).

En este proyecto se van a emplear diversos lenguajes de programación, por lo que la equivalencia en líneas de código se va a calcular asignando un porcentaje de peso sobre el proyecto de cada lenguaje. Empleando el porcentaje estimado de empleo de cada lenguaje y la equivalencia entre puntos de función y líneas de código obtendremos un valor de equivalencia combinado:

Lenguaje	**Peso estimado**	**LDC/PF**	Valor ajustado
HTML	10 %	34	3,4
Java	40 %	53	21,2
JavaScript	50 %	47	23,5
Combinado	-	-	**48,1 LDC/PF**

Cuadro 1.3: Cálculo de la relación puntos de función-lineas de código con varios lenguajes

Empleando el valor calculado:
Lineas de código = 138,18*48,1 = 6646 LDC
Esfuerzo $E = a * KLDC^b$
$E = 3 * 6{,}646^{1,12} = 25{,}02 personas/mes$
Tiempo de desarrollo $TD = c * E^d$
$TD = 2{,}5 * 25{,}02^{0,35} = 7,71 meses$
Nº medio de personas necesarias = 25.02/7,71=3,24

Esto quiere decir que este proyecto con 3,24 empleados se completaría en 7,71 meses. Con una sola persona trabajando en él llevaría **25.02 meses** en completarse.

Estimación por puntos de Caso de Uso

El segundo método de estimación se basa en un análisis previo de los Casos de Uso del proyecto (vease la sección 2.1.1). Primero se deben calcular los puntos de caso de uso no ajustados a partir de los actores y casos de uso identificados, asignándoles un peso atendiendo a unos valores predefinidos (vease la sección 5.1.4).

- **Factor de peso de los actores sin ajustar (UAW)**:

  1. Usuario: tipo Complejo - Factor 3.
  2. Miembro identificado: tipo Complejo - Factor 3.
  3. Administrador: tipo Complejo - Factor 3.
  4. Líder de Grupo: tipo Complejo - Factor 3.

  Total: UAW=12

- **Factor de peso de los casos de uso sin ajustar (UUCW)**:

  1. Solicitar registro: tipo Simple - Factor 5.
  2. Activar cuenta: tipo Complejo - Factor 15.
  3. Iniciar sesión: tipo Medio - Factor 10.
  4. Aceptar solicitud de registro: tipo Simple - Factor 5.
  5. Denegar solicitud de registro: tipo Simple - Factor 5.
  6. Bloquear registro: tipo Simple - Factor 5.
  7. Solicitar diálogo: tipo Medio - Factor 10.
  8. Aceptar diálogo: tipo Medio - Factor 10.
  9. Acceder a diálogo: tipo Simple - Factor 5.
  10. Listar usuarios: tipo Simple - Factor 5.
  11. Filtrar usuarios: tipo Simple - Factor 5.
  12. Mostrar estado de comunicaciones: tipo Medio - Factor 5.
  13. Mandar mensaje de diálogo: tipo Simple - Factor 5.
  14. Mandar mensaje de grupo: tipo Simple - Factor 5.

15. Mandar secreto: tipo Simple - Factor 5.
16. Proponer desvelar secreto: tipo Simple - Factor 5.
17. Aceptar desvelar secreto: tipo Simple - Factor 5.
18. Rechazar desvelar secreto: tipo Simple - Factor 5.
19. Adjuntar archivo: tipo Simple - Factor 5.
20. Crear grupo: tipo Medio - Factor 10.
21. Dejar grupo: tipo Simple - Factor 5.
22. Aceptar invitación de grupo: tipo Medio - Factor 10.
23. Listar grupos: tipo Simple - Factor 5.
24. Filtrar grupos: tipo Simple - Factor 5.
25. Acceder a grupo: tipo Simple - Factor 5.
26. Invitar a grupo: tipo Simple - Factor 5.
27. Transferir liderazgo: tipo Simple - Factor 5.
28. Expulsar de grupo: tipo Simple - Factor 5.
29. Importar datos locales: tipo Simple - Factor 5.
30. Exportar datos locales: tipo Simple - Factor 5.
31. Cerrar sesión: tipo Simple - Factor 5.
32. Descargar archivo: tipo Simple - Factor 5.

Total: UUCW=195

Puntos de caso de uso sin ajustar **UUCP=UAW+UUCW=12+195=207**

Una vez obtenidos los puntos de caso de uso sin ajustar debemos obtener los puntos de caso de uso ajustados. Para ello debemos establecer dos factores, de complejidad técnica y de entorno.

Factor	**Descripción**	**Peso**	**Influencia**	**Resultado**
R1	Sistema distribuido	2	4	R1=2*4=**8**
R2	Objetivos de rendimiento	2	2	R2=2*2=**4**
R3	Eficiencia respecto al usuario final	1	3	R3=1*3=**3**
R4	Procesamiento complejo	1	4	R4=1*4=**4**
R5	Código reutilizable	1	2	R5=1*2=**2**
R6	Instalación sencilla	0.5	0	R6=0,5*0=**0**
R7	Fácil utilización	0.5	4	R7=0,5*4=**2**
R8	Portabilidad	2	2	R8=2*2=**4**
R9	Fácil de cambiar	1	2	R9=1*2=**2**
R10	Uso concurrente	1	4	R10=1*4=**4**
R11	Características de seguridad	1	5	R11=1*5=**5**
R12	Accesible por terceros	1	0	R12=1*0=**0**
R13	Se requiere formación especial	1	0	R13=1*0=**0**
TCF=$0{,}6 + (0{,}01 * \sum_{i=1}^{i=13} R_i)$	-	-	-	TCF=0,6+(0,01*38) **TFC=0,98**

Cuadro 1.4: Cálculo de los factores de Complejidad Técnica

Factor	Descripción	Peso	Influencia	Resultado
R1	Familiaridad con el proyecto	1.5	2	R1=1,5*2=**1,5**
R2	Experiencia con la aplicación	0.5	0	R2=0,5*0=**0**
R3	Experiencia con orientación a objetos	1	3	R3=1*3=**3**
R4	Capacidades de análisis	0.5	3	R4=0,5*2=**1**
R5	Motivación	1	4	R5=1*4=**4**
R6	Requisitos estables	2	4	R6=2*4=**8**
R7	Trabajadores a tiempo parcial	-1	0	R7=-1*0=**0**
R8	Dificultad de lenguaje de programación	-1	3	R8=-1*3=**-3**
EF=$1{,}4 - (0{,}03 * \sum_{i=1}^{i=8} R_i)$	-	-	-	$EF = 1{,}4-(0{,}03*14{,}5)$ **EF=0,965**

Cuadro 1.5: Cálculo de los factores de Entorno

Puntos de caso de uso ajustados **UCP=UUCP*TCF*EF=207*0,98*0,965=195,7599**

Se estima que cada punto de caso de uso requiere 20 horas/persona, por lo tanto:

$Totaldehoras/persona = 195,7599 * 20 = 3915,198 horas/persona$

Si consideramos que un mes tiene 208 horas laborales el esfuerzo será **E=3820,628/208=18,82 personas/mes**

Esto quiere decir que este proyecto desarrollado por una sola persona llevaría **18,82** meses en completarse.

Comparación de resultados

Una vez realizada la estimación del esfuerzo, podemos ver que se obtienen valores dispares. Teniendo en cuenta que este proyecto será desarrollado por una sola persona, la estimación por puntos de función concluye que llevará **25.02 meses** en completarse, mientras que el análisis por puntos de caso de uso concluye que llevará **18,82 meses** en completarse.

Las diferencias de estimación del esfuerzo son debidas a la base de las diferentes técnicas que se han empleado para ello. La primera estimación basa en puntos de función identifica los elementos que componen el producto, teniendo en cuenta la comunicación de entrada y salida de datos con el usuario, con otros sistemas externos y los elementos de datos que el sistema va a manejar, calculando el esfuerzo a través de una valoración de la complejidad

de cada elemento, mientras que la segunda estimación se basa en un análisis previo de los casos de uso, que solo van a reflejar la interacción del usuario con el sistema, lo que puede dar lugar a una valoración errónea de la complejidad del sistema, ya que se obvian el resto de factores dentro de los procesos del sistema.

Por otra parte, la estimación por puntos de función se basa en la transformación a líneas de código según el lenguaje de programación que se vaya a emplear, obteniendo el valor de conversión de los datos históricos de gran cantidad de organizaciones, mientras que la técnica de casos de uso basa la conversión a esfuerzo en un valor estándar.

Teniendo en cuenta que el objetivo del proyecto no es lograr un producto plenamente funcional, completo y seguro sino la investigación de los sistemas de seguridad, que la estimación del esfuerzo se basa en datos estadísticos y las limitaciones temporales impuestas por el carácter académico de este proyecto, determinamos que el proyecto (no la aplicación completa) se puede llevar a cabo en 6 meses.

1.6.2. Estimación presupuestaria

Para la estimación del presupuesto tendremos en cuenta las herramientas utilizadas (tanto hardware como software) así como los recursos humanos necesarios. Calcularemos el presupuesto estimado siguiendo las dos estimaciones del esfuerzo y el presupuesto real con el tiempo real de desarrollo de 6 meses.

Presupuesto hardware

Presupuesto Hardware Total:

- **Estimación 1**: 846,62 €
- **Estimación 2**: 630,57 €
- **Coste real**: 219,5 €

Teniendo en cuenta que ambos PC se utilizan para más usos que el desarrollo de este proyecto y estimando una vida útil para ambos de 5 años, los porcentajes de uso se basan en la proporción de tiempo de desarrollo comparado con esa vida útil.

Presupuesto software

Ya que todo el software utilizado para el desarrollo de este proyecto es gratuito el presupuesto software es de 0 €.

Hardware	Coste total	Uso	Coste relativo
Portatil VANT MOOVE Pro2 C408T	570 €	41 %	228 €
		30 %	171 €
		10 %	57 €
PC Sobremesa por componentes	685 €	41 %	280,85 €
		30 %	205,5 €
		10 %	68,5 €
Conexión a internet	45€/mes*25.02mes=1125,9 €	30 %	337,77 €
	45€/mes*18,82mes=846,9 €	30 %	254,07 €
	45€/mes*6mes=315 €	30 %	94,5 €

Cuadro 1.6: Presupuesto hardware

Recursos humanos

En este proyecto solo vamos a tener una persona trabajando con el rol de analista-programador. Estimamos que el salario de este un analista-programador se encontrará sobre los 10 €/h al tener poca experiencia. Dicha persona trabajando una media de 208 horas al mes (8h*26dias):

	Horas de trabajo	Sueldo/hora	Total
Analista-programador	25.02 meses=5204,16 h	10 €/h	52041,6 €
	18.82 meses=3914,56 h		39145,6 €
	6 meses=1248 h		12480 €

Cuadro 1.7: Presupuesto de recursos humanos

Presupuesto total

	Hardware	**Software**	**Humano**	**Total**
Estimación 1	1301,62 €	0	118092 €	*119393,62 €*
Estimación 2	630,57 €	0	58718,4 €	*57938,9 €*
Coste real	219,5 €	0	18720 €	*18939,5 €*

Cuadro 1.8: Presupuesto total

1.6.3. Planificación temporal

Para realizar una planificación temporal, dado el carácter iterativo del proyecto, he dividido las tareas de análisis, diseño, implementación y pruebas en diversas iteraciones que cubrirán las principales áreas que compondrán el producto final.

El tiempo de desarrollo que tomaré para realizar la planificación sera el tiempo real del proyecto de 6 meses.

		Nombre	Duración	Inicio
1		**Iteración1: Gestión de usuarios**	22 days	**2/03/15 8:00**
2		Análisis	3 days	2/03/15 8:00
3		Diseño	5 days	5/03/15 8:00
4		Implementación	10 days	12/03/15 8:00
5		Pruebas	4 days	26/03/15 8:00
6		**Iteración 2: Gestión de grupos**	22 days	**1/04/15 8:00**
7		Análisis	3 days	1/04/15 8:00
8		Diseño	5 days	6/04/15 8:00
9		Implementación	10 days	13/04/15 8:00
10		Pruebas	4 days	27/04/15 8:00
11		**Iteracón 3: Servicios de comunicacion básicos**	22 days	**1/05/15 8:00**
12		Análisis	3 days	1/05/15 8:00
13		Diseño	5 days	6/05/15 8:00
14		Implementación	10 days	13/05/15 8:00
15		Pruebas	4 days	27/05/15 8:00
16		**Iteración 4: Transferencia de archivos**	16 days	**2/06/15 8:00**
17		Análisis	2 days	2/06/15 8:00
18		Diseño	4 days	4/06/15 8:00
19		Implementación	7 days	10/06/15 8:00
20		Pruebas	3 days	19/06/15 8:00
21		**Iteración 5: Depuración y ampliación**	22 days	**24/06/15 8:00**
22		Análisis	3 days	24/06/15 8:00
23		Diseño	5 days	29/06/15 8:00
24		Implementación	10 days	6/07/15 8:00
25		Pruebas	4 days	20/07/15 8:00
26		**Documentación**	27 days	**24/07/15 8:00**
27		Documentación técnica	25 days	24/07/15 8:00
28		Manuales	2 days	28/08/15 8:00

Figura 1.3: Planificación de las tareas del proyecto

El diagrama de Gantt correspondiente permite observar de forma rápida de un solo vistazo la disposición temporal de dichas tareas.

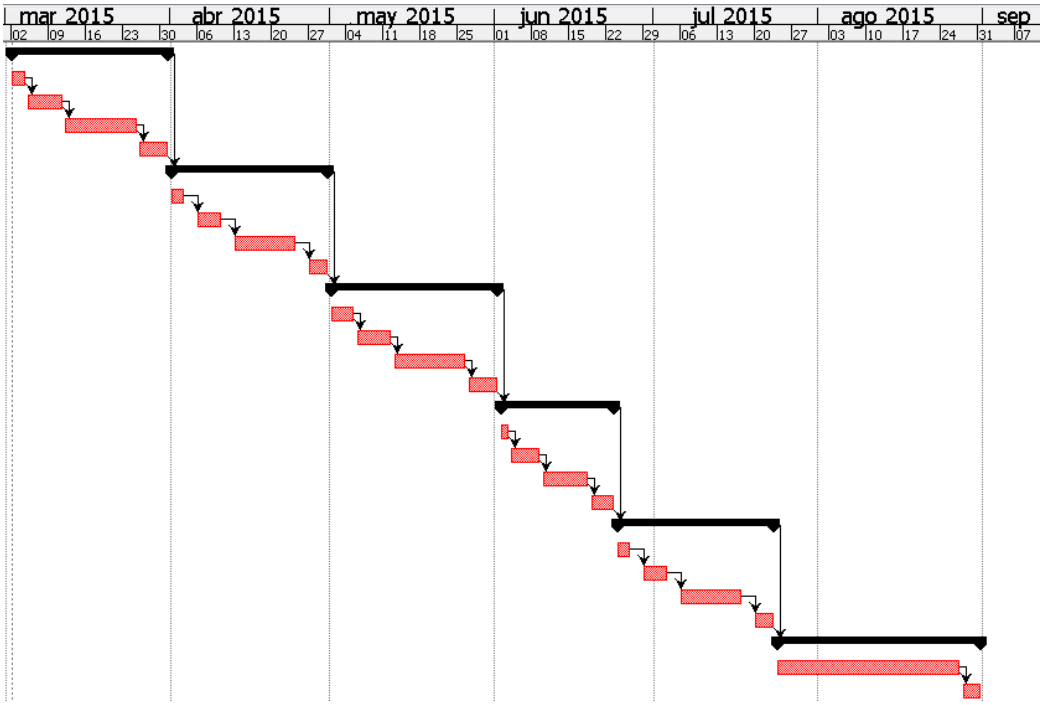

Figura 1.4: Diagrama de Gantt del proyecto

Capítulo 2

Documentación técnica

2.1. Análisis

2.1.1. Casos de Uso

A continuación se describen y detallan los casos de uso de la aplicación. Dichos casos de uso representan la forma en que los diferentes actores externos interactúan con nuestra aplicación y su descripción permite determinar el comportamiento del sistema en respuesta a dichas interacciónes.

Para facilitar comprensión dividiremos los casos de uso entre los principales grupos de características o subsistemas de la aplicación.

Actores

El primer paso es identificar los actores que van a interactuar con nuestro sistema y definir sus características.

Actor	**Descripción**	**Herencia**
Usuario	Usuario estándar que no ha iniciado una sesión en la aplicación. Su interacción con el sistema es muy limitada	-
Miembro identificado	Usuario con una cuenta en la aplicación que ha iniciado una sesión. Podrá realizar la mayor parte de las actividades.	-
Líder de grupo	Miembro de un grupo con el estatus de líder. Podrá llevar a cabo las operaciónes de gestión de los miembros del grupo.	Miembro identificado
Administrador	Usuario de la aplicación que posee permisos especiales para gestionar las solicitudes de registro de la aplicación.	Miembro identificado

Cuadro 2.1: Puntos de función de archivos internos y externos según su complejidad

Gestión de usuarios

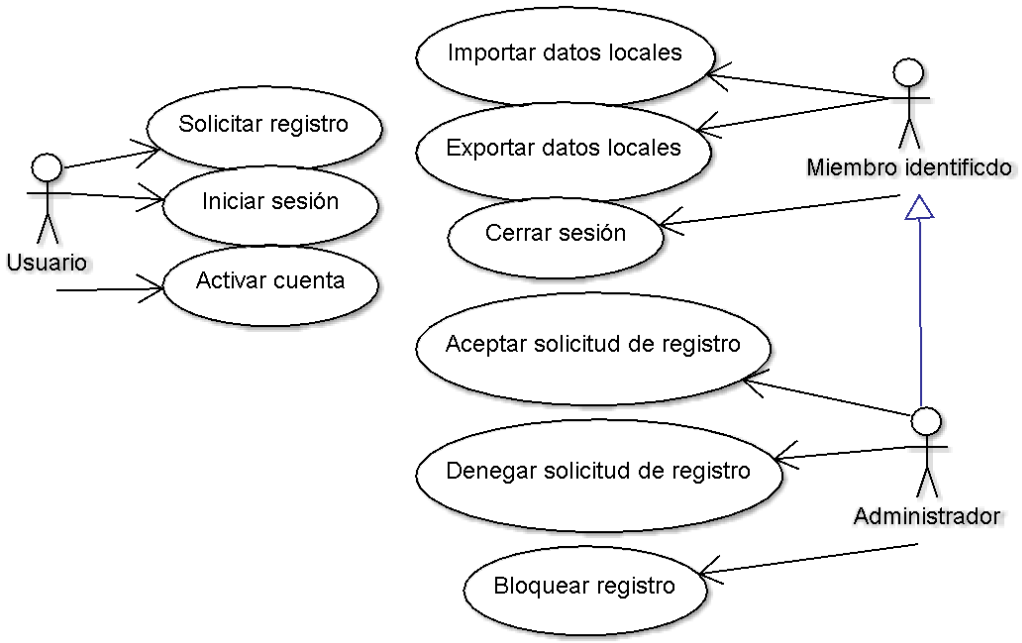

Figura 2.1: Diagrama de casos de uso de gestión de usuarios

Nombre e ID del CU	CU-01. Solicitar registro
Actor	Usuario
Descripción	El usuario enviará sus datos de usuario al sistema, generando una solicitud de registro que deberá ser gestionada por un administrador.
Precondiciónes	PRE-1. El usuario no está identificado en el sistema.
Postcondiciones	POST-1. La solicitud queda almacenada en el sistema.
Flujo normal	FN1 El actor introduce sus datos de usuario e indica al sistema que quiere solicitar el registro. FN2 El sistema comprueba los datos introducidos. FN3 Si los datos son correctos, el sistema almacena la solicitud de registro e informa al actor del resultado.

Flujo alternativo 1	FA3 Si los datos son incorrectos se informa al usuario del error y no se procede a almacenar la solicitud.
Flujo alternativo 2	FA3 Si los datos son incorrectos y no hay otros usuarios, el usuario pasa automáticamente al estado de activación y tiene el rol de administrador.
Excepciones	E1 El usuario ha dejado campos requeridos sin rellenar. E2 El usuario o correo electrónico ya existen. E3 El usuario está bloqueado.
Prioridad	Alta
Otra info	El primer usuario registrado en la aplicación será el que tome el rol de administrador.

Cuadro 2.2: CU-01. Solicitar registro

Nombre e ID del CU	CU-02. Activar cuenta
Actor	Usuario
Descripción	El usuario recibirá un correo electrónico con un enlace de activación. Mediante dicho enlace el sistema comprobará la identidad del usuario, generará una clave RSA y la almacenará protegida por una contraseña de forma local en el cliente.
Precondiciónes	▪ PRE-1. La solicitud de registro del usuario ha sido aceptada por un administrador.
Postcondiciones	▪ POST-1. El usuario queda activado ▪ POST-2. La clave RSA queda almacenada en el cliente.

Flujo normal	FN1 El actor accede al link de activación. FN2 El sistema recoge los datos de activación incluidos en el link y los comprueba. FN3 Si los datos son correctos, el sistema genera una clave RSA y la envía al cliente. FN4 El sistema comprueba la correcta recepción de la clave. FN5 Si la clave se ha recibido correctamente, los datos públicos se almacenan en la BD, se activa al usuario y se continúa el flujo normal. FN6 El usuario introduce una calve de seguridad. FN7 El sistema comprueba la clave. FN8 Si la clave es correcta el sistema cifra la clave RSA y la almacena en el cliente. FN9 El sistema almacena la clave pública del usuario en la BD. FN10 El sistema cambia el estado del usuario a activo y elimina sus datos de activación. FN11 Se redirige al usuario al inicio.
Flujo alternativo 1	FA3 Si los datos de activación son incorrectos se informa al usuario del error y no se procede a la activación.
Flujo alternativo 2	FA5 Si la clave no se ha recibido correctamente se recarga la página, reiniciando el caso de uso.
Flujo alternativo 3	FA8 Si la contraseña introducida es incorrecta se informa al usuario del error y este debe corregirlo para continuar el flujo normal.

Excepciones	E1 Los datos de activación son incorrectos o el usuario ya está activado. E2 La clave recibida es incorrecta. E3 El usuario no introduce contraseña. E4 La contraseña introducida no es válida
Prioridad	Alta
Otra info	Toda la comunicación entre el cliente y el servidor se realizará de forma segura mediante una clave de usar y tirar

Cuadro 2.3: CU-02. Activar cuenta

Nombre e ID del CU	CU-03. Iniciar sesión
Actor	Usuario
Descripción	El usuario introducirá su contraseña y su clave RSA, de forma que el sistema identifique el usuario mediante un mecanismo de desafío-respuesta.
Precondiciónes	▪ PRE-1. El usuario no está identificado en el sistema.
Postcondiciones	▪ POST-1. El usuario inicia una sesión en el sistema.
Flujo normal	FN1 El actor introduce su correo, contraseña y clave RSA FN2 El sistema verifica la existencia del usuario FN3 Si el usuario existe y está activado el servidor verifica la identidad del usuario mediante un mecanismo de desafío-respuesta. FN4 El lado cliente del sistema responde al desafío empleando la clave RSA descifrada FN5 Si la respuesta al desafío es correcta se almacenan los datos de sesión y se redirige al usuario a la página principal.

Flujo alternativo 1	FA3 Si el usuario no existe o no está activado el sistema notifica el error al actor y no se inicia la sesión.
Flujo alternativo 2	FA5 Si la respuesta al desafío es incorrecta el sistema notifica el error al actor y no se inicia la sesión.
Excepciones	E1 El usuario no existe E2 El usuario no está activado E3 La respuesta al desafío es incorrecta
Prioridad	Alta
Otra info	El desafío-respuesta consiste en que el servidor genera un desafío, el cual cifra con la clave pública RSA del usuario y la envía al cliente. El lado cliente del sistema lo descifra y devuelve al sistema. Si la respuesta coincide con el desafío se puede asegurar que el usuario es quien dice ser.

Cuadro 2.4: CU-03. Iniciar sesión

Nombre e ID del CU	CU-04. Aceptar solicitud de registro
Actor	Administrador
Descripción	El administrador acepta una solicitud de registro de un usuario
Precondiciónes	▪ Existe al menos una solicitud de registro ▪ El usuario está identificado en el sistema
Postcondiciones	▪ El usuario que había solicitado el registro queda en estado pendiente de activación ▪ Quedan almacenados los datos de activación del usuario

Flujo normal	FN1 El administrador indica que desea aceptar la solicitud de registro FN2 El sistema comprueba que el actor tiene el estatus de administrador FN3 Si el actor es un administrador el sistema genera un código de activación y lo almacena FN4 El sistema envía un correo electrónico a la dirección del usuario que realizó la solicitud con un enlace de activación
Flujo alternativo	FA3 Si el actor no es un administrador el sistema deniega la operación
Excepciones	E1 El actor no es un administrador
Prioridad	Alta
Otra info	N/A

Cuadro 2.5: CU-04. Aceptar solicitud de registro

Nombre e ID del CU	CU-05. Denegar solicitud de registro
Actor	Administrador
Descripción	El administrador deniega una solicitud de registro de un usuario
Precondiciónes	▪ Existe al menos una solicitud de registro ▪ El usuario está identificado en el sistema
Postcondiciones	▪ Los datos del usuario que había solicitado el registro son eliminados de la BD

Flujo normal	FN1 El administrador indica que desea denegar la solicitud de registro FN2 El sistema comprueba que el actor tiene el estatus de administrador FN3 Si el actor es un administrador el sistema elimina los datos del usuario de la BD
Flujo alternativo	FA3 Si el actor no es un administrador el sistema deniega la operación
Excepciones	E1 El actor no es un administrador
Prioridad	Media
Otra info	N/A

Cuadro 2.6: CU-05. Denegar solicitud de registro

Nombre e ID del CU	CU-06. Bloquear registro
Actor	Administrador
Descripción	El administrador blouea una solicitud de registro de un usuario
Precondiciónes	▪ Existe al menos una solicitud de registro ▪ El usuario está identificado en el sistema
Postcondiciones	▪ El usuario que había solicitado el registro queda bloqueado para impedir posteriores solicitudes
Flujo normal	FN1 El administrador indica que desea denegar la solicitud de registro FN2 El sistema comprueba que el actor tiene el estatus de administrador FN3 Si el actor es un administrador el sistema establece el estado del usuario solicitante como bloqueado

Flujo alternativo	FA3 Si el actor no es un administrador el sistema deniega la operación
Excepciones	E1 El actor no es un administrador
Prioridad	Media
Otra info	N/A

Cuadro 2.7: CU-06. Bloquear registro

Nombre e ID del CU	CU-07. Exportar datos locales
Actor	Miembro identificado
Descripción	El actor exporta los datos locales a un archivo.
Precondiciónes	▪ El usuario está identificado en el sistema.
Postcondiciones	▪ Los datos locales quedan almacenados un archivo protegido.
Flujo normal	FN1 El actor indica que quiere realizar una copia de seguridad de sus datos locales. FN2 El sistema genera un archivo con los datos locales que se guarda en la localización del dispositivo del actor determinada por el mismo.
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Baja
Otra info	Los datos locales se almacenan cifrados en un espacio reservado para la aplicación mediante LocalStorage, el cual puede ser borrado al limpiar la caché o emplear un dispositivo distinto.

Cuadro 2.8: CU-07. Exportar datos locales

Nombre e ID del CU	CU-08. Importar datos locales

Actor	Miembro identificado
Descripción	El actor importa los datos locales de un archivo.
Precondiciónes	▪ El usuario está identificado en el sistema.
Postcondiciones	▪ Los datos del archivo quedan almacenados en en el espacio local
Flujo normal	FN1 El sistema detecta que no hay datos locales y ofrece la opción de importarlos al usuario FN2 El actor selecciona el archivo que desea importar. FN3 El sistema comprueba la estructura del archivo de datos FN4 Si el archivo tiene la estructura correcta sus datos se guardan en el espacio local mezclándose con los actuales.
Flujo alternativo 1	FA1 El usuario indica que quiere importar los datos de un archivo FA2 El caso de uso continúa en el paso 2 del flujo normal
Flujo alternativo 2	FA3 Si el archivo tiene una estructura incorrecta su importación se interrumpe.
Excepciones	E1 El archivo tiene una estructura incorrecta
Prioridad	Baja
Otra info	La estructura del archivo se definirá mas adelante

Cuadro 2.9: CU-08. Importar datos locales

Nombre e ID del CU	CU-09. Cerrar sesión
Actor	Miembro identificado
Descripción	El actor cierra la sesión

Precondiciónes	▪ El usuario está identificado en el sistema.
Postcondiciones	▪ La sesión queda cerrada ▪ Todos los datos de sesión se eliminan
Flujo normal	FN1 El actor indica que desea cerrar la sesión FN2 El sistema elimina los datos de sesión, tanto del servidor como del cliente FN3 El actor es redirigido a la página de acceso
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Alta
Otra info	N/A

Cuadro 2.10: CU-09. Cerrar sesión

Gestión de grupos

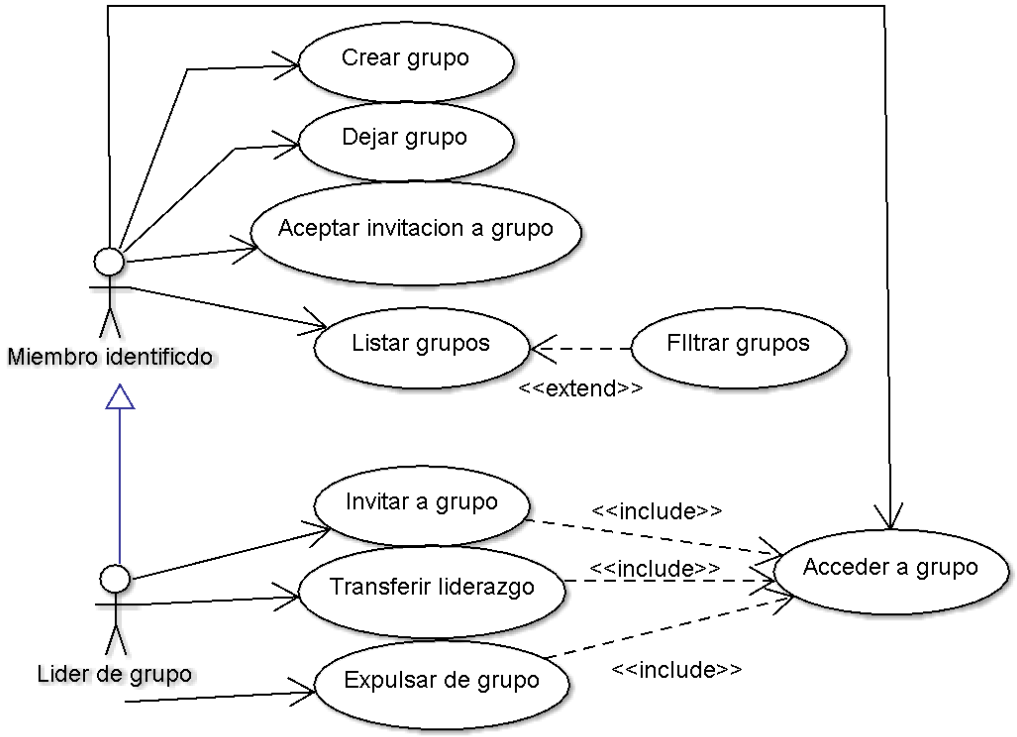

Figura 2.2: Diagrama de casos de uso de gestión de grupos

Nombre e ID del CU	CU-10. Crear grupo
Actor	Miembro identificado
Descripción	El actor crea un nuevo grupo que posteriormente gestionará como lider.
Precondiciónes	▪ PRE-1. El usuario está identificado en el sistema
Postcondiciones	▪ POST-1. El nuevo grupo queda almacenado en la BD ▪ POST-2. El usuario creador queda almacenado como miembro líder ▪ POST-3. La clave de grupo se almacena en el espacio local del actor

Flujo normal	FN1 El actor introduce un nombre de grupo FN2 El sistema comprueba que no existe un grupo con mismo nombre FN3 Si el grupo es nuevo, el sistema almacena el nuevo grupo en la BD. FN4 El sistema almacena el usuario creador como miembro lider del grupo FN5 El sistema genera una clave de grupo y la almacena localmente de forma segura FN6 Se redirige al actor a la página del nuevo grupo
Flujo alternativo	FA3 Si el grupo ya existe se informa al usuario y se termina el CU
Excepciones	E1 Ya existe un grupo con ese nombre E2 El nombre indicado por el actor está vacío
Prioridad	Alta
Otra info	N/A

Cuadro 2.11: CU-10. Crear grupo

Nombre e ID del CU	CU-11. Dejar grupo
Actor	Miembro identificado
Descripción	El actor crea un nuevo grupo que posteriormente gestionará como lider.
Precondiciónes	▪ PRE-1. El usuario está identificado en el sistema ▪ PRE-2. El usuario es miembro de al menos un grupo
Postcondiciones	▪ POST-1. El actor deja de ser miembro del grupo ▪ POST-2. Las claves de grupo se eliminan del almacén local

Flujo normal	FN1 El actor indica que desea dejar el grupo FN2 El sistema comprueba la identidad del actor FN3 Si el actor se identifica correctamente el sistema elimina sus datos de miembro del grupo FN4 El sistema comprueba que el actor no sea el líder de grupo FN5 Si el actor no es líder de grupo el actor deja de ser miembro del grupo FN6 El sistema elimina las claves de grupo del almacén local
Flujo alternativo 1	FA3 Si el actor no se identifica correctamente se cancela el CU y el sistema notifica el error al actor.
Flujo alternativo 2	FA5 Si el actor es el líder del grupo y es el único miembro, el sistema elimina el grupo.
Flujo alternativo 3	FA5 Si el actor es el líder del grupo pero hay más miembros, el sistema notifica al usuario de que debe transferir el liderazgo antes de salir.
Excepciones	E1 La identidad del actor no se puede verificar. E2 El actor es líder de grupo y hay otros miembros.
Prioridad	Media
Otra info	Se debe comprobar la identidad del usuario para evitar la suplantación

Cuadro 2.12: CU-11. Dejar grupo

Nombre e ID del CU	CU-12. Aceptar invitación a grupo
Actor	Miembro identificado
Descripción	El actor acepta una invitación a un grupo y se convierte en miembro del mismo.

Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor tiene al menos una invitación pendiente
Postcondiciones	▪ POST-1. El actor pasa a ser miembro del grupo ▪ POST-2. El sistema almacena localmente la clave de grupo ▪ POST-3. La invitación queda eliminada del sistema
Flujo normal	FN1 El actor indica que desea aceptar la invitación a un grupo FN2 El sistema comprueba la identidad del actor FN3 Si el actor se identifica correctamente el sistema guarda la clave de grupo localmente FN4 El sistema almacena los datos de miembro del grupo del actor
Flujo alternativo	FA3 Si el actor no se identifica correctamente se cancela el CU y el sistema notifica el error al actor.
Excepciones	E1 La identidad del actor no se puede verificar.
Prioridad	Alta
Otra info	N/A

Cuadro 2.13: CU-12. Aceptar invitación a grupo

Nombre e ID del CU	CU-13. Invitar a grupo
Actor	Líder de grupo
Descripción	El actor envía una invitación para unirse al grupo a otro usuario.

Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor es líder del grupo ▪ PRE-3. El usuario invitado no es miembro del grupo ▪ PRE-4. El usuario no posee una invitación al grupo
Postcondiciones	▪ POST-1. El sistema almacena la invitación
Flujo normal	FN1 El actor indica que desea invitar a otro usuario no miembro a formar parte del grupo FN2 El sistema almacena la invitación al grupo en el servidor
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Alta
Otra info	La invitación contendrá la clave del grupo

Cuadro 2.14: CU-13. Invitar a grupo

Nombre e ID del CU	CU-14. Transferir liderazgo
Actor	Líder de grupo
Descripción	El actor transfiere el liderazgo del grupo a otro miembro.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema. ▪ PRE-2. El actor es líder del grupo. ▪ PRE-3. El nuevo líder seleccionado es miembro del grupo.
Postcondiciones	▪ POST-1. El actor pasa a ser un miembro normal ▪ POST-2. El miembro seleccionado pasa a ser el lider de grupo

Flujo normal	FN1 El actor indica que desea transferir el liderazgo a otro miembro FN2 El sistema actualiza el estado del actor y el miembro seleccionado
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.15: CU-14. Transferir liderazgo

Nombre e ID del CU	CU-15. Expulsar de grupo
Actor	Líder de grupo
Descripción	El actor expulsa a in miembro del grupo.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema. ▪ PRE-2. El actor es líder del grupo. ▪ PRE-3. El usuario a expulsar es miembro del grupo.
Postcondiciones	▪ POST-1. El miembro seleccionado deja de ser miembro del grupo
Flujo normal	FN1 El actor indica que desea expulsar a un miembro FN2 El sistema elimina los datos del miembro seleccionado del grupo
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.16: CU-15. Expulsar de grupo

Nombre e ID del CU	CU-16. Listar grupos

Actor	Miembro identificado
Descripción	El actor obtiene la lista de los grupos a los que pertenece.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema. ▪ PRE-2. El actor pertenece al menos a un grupo.
Postcondiciones	▪ POST-1. El actor visualiza la lista de los grupos a los que pertenece
Flujo normal	FN1 El actor indica que desea listar sus grupos FN2 El sistema muestra la lista de grupos a los que pertenece el actor
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Media
Otra info	N/A

Cuadro 2.17: CU-16. Listar grupos

Nombre e ID del CU	CU-17. Filtrar grupos
Actor	Miembro identificado
Descripción	El actor obtiene la lista filtrada de los grupos a los que pertenece.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema. ▪ PRE-2. El actor pertenece al menos a un grupo.
Postcondiciones	▪ POST-1. El actor visualiza la lista filtrada de los grupos a los que pertenece

Flujo normal	FN1 El actor indica que desea listar sus grupos FN2 El actor introduce el criterio de filtro FN3 El sistema muestra la lista de grupos a los que pertenece el actor cuyo nombre se ajusta al filtro introducido
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.18: CU-17. Filtrar grupos

Nombre e ID del CU	CU-18. Acceder a grupo
Actor	Miembro identificado
Descripción	El actor accede a un grupo del que es miembro, obteniendo todos los datos relacionados.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema. ▪ PRE-2. El actor es miembro de al menos del grupo.
Postcondiciones	▪ POST-1. El actor se encuentra conectado al grupo ▪ POST-2. El sistema actualiza localmente los datos relevantes de grupo
Flujo normal	FN1 El actor indica que desea acceder a un grupo FN2 El sistema verifica que el actor es miembro del grupo FN3 Si el actor es miembro del grupo el sistema crea una sesión segura para la comunicación de los datos de grupo FN4 El sistema mantiene actualizados los datos de grupo locales periódicamente
Flujo alternativo	FA3 Si el actor no es miembro del grupo el sistema deniega el acceso

Excepciones	E1 El actor no es miembro del grupo.
Prioridad	Alta
Otra info	Los datos de grupo relevantes son los mensajes, miembros, secretos, etc...

Cuadro 2.19: CU-18. Acceder a grupo

Mensajería

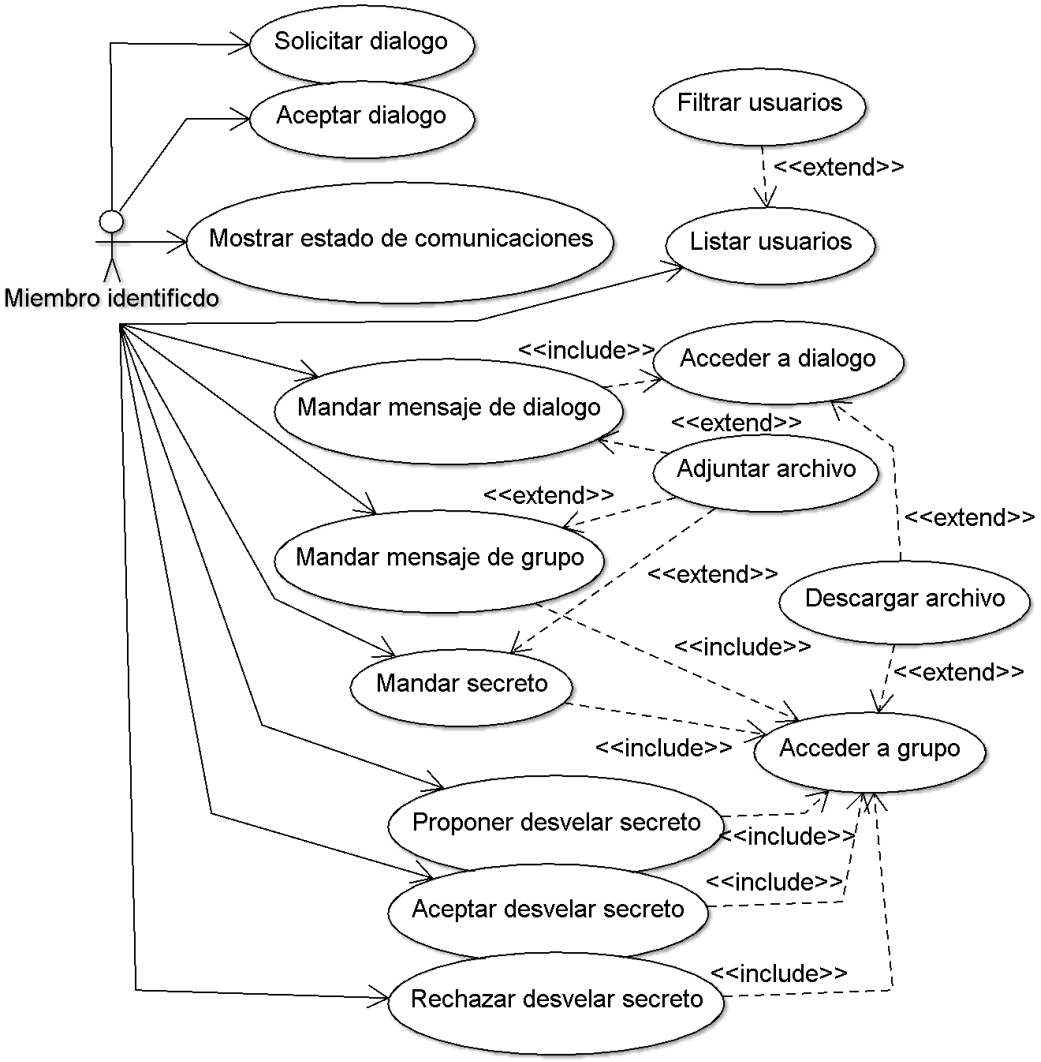

Figura 2.3: Diagrama de casos de uso de mensajería

Nombre e ID del CU	CU-19. Listar usuarios
Actor	Miembro identificado
Descripción	El actor obtiene la lista de los activos de la aplicación.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema.

Postcondiciones	▪ POST-1. El actor visualiza la lista de los usuarios activos del sistema
Flujo normal	FN1 El actor indica que desea listar los usuarios FN2 El sistema muestra la lista de los usuarios activos del sistema
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Alta
Otra info	N/A

Cuadro 2.20: CU-19. Listar usuarios

Nombre e ID del CU	CU-20. Filtrar usuarios
Actor	Miembro identificado
Descripción	El actor obtiene la lista filtrada de los activos de la aplicación.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema.
Postcondiciones	▪ POST-1. El actor visualiza la lista filtrada de los usuarios activos del sistema
Flujo normal	FN1 El actor indica que desea listar los usuarios FN2 El actor introduce un criterio de filtro FN3 El sistema muestra la lista de los usuarios activos del sistema cuyo nombre de usuario, nombre o apellidos cumplen el criterio de filtro
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.21: CU-20. Filtrar usuarios

Nombre e ID del CU	CU-21. Solicitar diálogo
Actor	Miembro identificado
Descripción	El actor envía una solicitud de diálogo a otro usuario.
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema.
Postcondiciones	▪ POST-1. El sistema almacena la solicitud de diálogo
Flujo normal	FN1 El actor indica que quiere solicitar el diálogo con otro usuario FN2 El sistema genera una clave de diálogo y la transfiere al cliente FN3 El sistema comprueba la correcta recepción de la clave FN4 Si la clave se ha recibido correctamente el sistema almacena la clave de diálogo en el espacio local del actor y la invitación a diálogo en la BD
Flujo alternativo	FA4 Si la clave es incorrecta se informa del error al usuario y se cancela el CU
Excepciones	E1 La clave no se recibe correctamente
Prioridad	Alta
Otra info	N/A

Cuadro 2.22: CU-21. Solicitar diálogo

Nombre e ID del CU	CU-22. Aceptar diálogo
Actor	Miembro identificado
Descripción	El actor acepta la invitación a diálogo de otro usuario
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema. ▪ PRE-2. El usuario posee al menos una invitación a diálogo

Postcondiciones	▪ POST-1. El sistema almacena el nuevo diálogo entre el actor y el usuario que mandó la invitación. ▪ POST-2. La invitación queda eliminada de la BD. ▪ POST-3. La clave de diálogo queda almacenada en el almacenamiento local del actor.
Flujo normal	FN1 El usuario indica que quiere aceptar una invitación de diálogo FN2 El sistema comprueba la identidad del actor FN3 Si el actor es identificado, el sistema envía al cliente la clave de diálogo FN4 El sistema almacena la clave localmente y elimina la invitación de la BD
Flujo alternativo 1	FA3 Si la identidad del actor no se puede verificar el CU se cancela y se notifica el error al actor
Excepciones	E1 La identidad del actor no puede ser confirmada
Prioridad	Alta
Otra info	N/A

Cuadro 2.23: CU-22. Aceptar diálogo

Nombre e ID del CU	CU-23. Mostrar estado de comunicaciones
Actor	Miembro identificado
Descripción	El sistema mantiene informado a tiempo real al actor del estado de sus comunicaciones
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema
Postcondiciones	▪ POST-1. El actor visualiza el estado de sus comunicaciones

Flujo normal	FN1 El actor indica que quiere visualizar el estado de sus comunicaciones FN2 El sistema muestra a tiempo real el estado de las comunicaciones al actor
Flujo alternativo	N/A
Excepciones	N/A
Prioridad	Media
Otra info	El estado de las comunicaciones incluye los diálogos, grupos, invitaciones, mensajes no leídos, etc...

Cuadro 2.24: CU-23. Mostrar estado de comunicaciones

Nombre e ID del CU	CU-24. Acceder a diálogo
Actor	Miembro identificado
Descripción	El actor accede al diálogo con un usuario, donde podrá enviar mensajes y visualizar los mensajes del otro usuario
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor tiene un diálogo con otro usuario
Postcondiciones	▪ POST-1. El actor se conecta al diálogo ▪ POST-2. El sistema actualiza localmente los datos relevantes de diálogo
Flujo normal	FN1 El usuario indica que quiere acceder al diálogo FN2 El sistema verifica la identidad del actor FN3 Si la identidad del actor se verifica, el sistema inicia sesión segura para la comunicación de los datos de diálogo FN4 El sistema mantiene actualizados los datos de diálogo locales periódicamente
Flujo alternativo	FA3 Si la identidad del actor no puede ser verificada el sistema deniega el acceso al diálogo

Excepciones	E1 La identidad del usuario no puede ser verificada
Prioridad	Alta
Otra info	Los datos relevantes de diálogo incluyen los mensajes, la hora del último mensaje recibido, etc...

Cuadro 2.25: CU-24. Acceder a diálogo

Nombre e ID del CU	CU-25. Mandar mensaje de diálogo
Actor	Miembro identificado
Descripción	El actor envía un mensaje de diálogo
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido al diálogo con un usuario
Postcondiciones	▪ POST-1. El mensaje queda almacenado en la BD de forma segura
Flujo normal	FN1 El actor introduce el mensaje que desea enviar FN2 El actor indica que desea enviar el mensaje FN3 El sistema comprueba si se ha adjuntado algún archivo FN4 Si no hay archivos adjuntos el sistema cifra el mensaje con la clave de diálogo y lo envía al servidor FN5 El sistema almacena el mensaje en la BD
Flujo alternativo	FA4 Si el actor ha adjuntado algún archivo el sistema lo cifra con la clave de diálogo y lo sube al servidor FA5 El cifra el mensaje con la clave de diálogo y lo envía al servidor FA6 El sistema almacena el mensaje en la BD

Excepciones	E1 El archivo seleccionado no existe E2 El mensaje está vacío
Prioridad	Alta
Otra info	N/A

Cuadro 2.26: CU-25. Mandar mensaje de diálogo

Nombre e ID del CU	CU-26. Mandar mensaje de grupo
Actor	Miembro identificado
Descripción	El actor envía un mensaje de grupo
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo
Postcondiciones	▪ POST-1. El mensaje queda almacenado en la BD de forma segura
Flujo normal	FN1 El actor introduce el mensaje que desea enviar FN2 El actor indica que desea enviar el mensaje FN3 El sistema comprueba si se ha adjuntado algún archivo FN4 Si no hay archivos adjuntos el sistema cifra el mensaje con la clave de grupo y lo envía al servidor FN5 El sistema almacena el mensaje en la BD
Flujo alternativo	FA4 Si el actor ha adjuntado algún archivo el sistema lo cifra con la clave de grupo y lo sube al servidor FA5 El cifra el mensaje con la clave de grupo y lo envía al servidor FA6 El sistema almacena el mensaje en la BD

Excepciones	E1 El archivo seleccionado no existe E2 El mensaje está vacío
Prioridad	Alta
Otra info	N/A

Cuadro 2.27: CU-26. Mandar mensaje de grupo

Nombre e ID del CU	CU-27. Mandar secreto
Actor	Miembro identificado
Descripción	El actor envía un secreto al grupo
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo
Postcondiciones	▪ POST-1. El secreto queda almacenado en la BD de forma segura ▪ POST-2. Las partes necesarias para desvelar secreto quedan almacenadas en la BD de forma segura
Flujo normal	FN1 El actor introduce el mensaje secreto que desea enviar y las partes minimas necesarias para desvelarlo FN2 El actor indica que desea enviar el mensaje FN3 El sistema comprueba si se ha adjuntado algún archivo FN4 Si no hay archivos adjuntos el sistema cifra el mensaje con la clave de grupo y lo envía al servidor FN5 El sistema genera una clave secreta y cifra el mensaje secreto con ella FN6 El sistema genera el secreto y sus partes a partir de la clave secreta y las almacena en la BD

Flujo alternativo	FA4 Si el actor ha adjuntado algún archivo el sistema lo cifra con la clave de grupo y lo sube al servidor FA5 El cifra el mensaje con la clave de grupo y lo envía al servidor FA6 El sistema genera una clave secreta y cifra el mensaje y el archivo secretos con ella FA7 El sistema genera el secreto y sus partes a partir de la clave secreta y las almacena en la BD
Excepciones	E1 El archivo seleccionado no existe E2 El mensaje está vacío
Prioridad	Media
Otra info	Las partes necesarias para desvelar el secreto se deben almacenar de forma segura hasta que el miembro correspondiente la obtenga. En ese momento debe eliminarse. Se generará una parte por cada miembro del grupo.

Cuadro 2.28: CU-27. Mandar secreto

Nombre e ID del CU	CU-28. Adjuntar archivo
Actor	Miembro identificado
Descripción	El actor adjunta un archivo a un mensaje de diálogo, grupo o secreto
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo o diálogo
Postcondiciones	▪ POST-1. El archivo queda seleccionado para su posterior envío

Flujo normal	FN1 El actor indica que quiere adjuntar un archivo FN2 El actor selecciona el archivo deseado FN3 El sistema almacena la localización del archivo seleccionado
Flujo alternativo	FA2 El actor cancela la selección y el archivo no se adjunta
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.29: CU-28. Adjuntar archivo

Nombre e ID del CU	CU-29. Proponer desvelar secreto
Actor	Miembro identificado
Descripción	El actor propone desvelar un secreto oculto
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo ▪ PRE-3. Existe al menos un secreto no desvelado en el grupo
Postcondiciones	▪ POST-1. La parte compartida del actor queda almacenada en la BD de forma segura ▪ POST-2. El estado del secreto queda actualizado a *pendiente*
Flujo normal	FN1 El actor indica que quiere proponer desvelar un secreto FN2 El sistema envía al servidor la parte compartida del actor y la almacena FN3 El sistema actualiza el estado del secreto a *pendiente*
Flujo alternativo	N/A
Excepciones	N/A

Prioridad	Media
Otra info	N/A

Cuadro 2.30: CU-29. Proponer desvelar secreto

Nombre e ID del CU	CU-30. Aceptar desvelar secreto
Actor	Miembro identificado
Descripción	El actor acepta desvelar un secreto oculto propuesto por otro usuario
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo ▪ PRE-3. Existe al menos un secreto en estado *pendiente* en el grupo
Postcondiciones	▪ POST-1. La parte compartida del actor queda almacenada en la BD de forma segura ▪ POST-2. El secreto queda revelado
Flujo normal	FN1 El actor indica que acepta desvelar un secreto FN2 El sistema envía al servidor la parte compartida del actor y la almacena FN3 Si el número de usuarios que han aceptado desvelar es suficiente el sistema calcula la clave secreta y revela el mensaje secreto FN4 El sistema actualiza el estado del secreto como *desvelado*
Flujo alternativo	FA3 Si el número de usuarios que han aceptado desvelar es insuficiente el sistema no desvela el secreto y su estado se mantiene *pendiente*
Excepciones	N/A
Prioridad	Media
Otra info	N/A

Cuadro 2.31: CU-30. Aceptar desvelar secreto

Nombre e ID del CU	CU-31. Rechazar desvelar secreto
Actor	Miembro identificado
Descripción	El actor rechaza desvelar un secreto oculto propuesto por otro usuario
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo ▪ PRE-3. Existe al menos un secreto en estado *pendiente* en el grupo
Postcondiciones	▪ POST-1. La parte compartida del actor queda almacenada en la BD de forma segura ▪ POST-2. El estado del secreto queda actualizado a *oculto*
Flujo normal	FN1 El actor indica que rechaza desvelar un secreto FN2 El sistema almacena el rechazo FN3 El sistema compara las partes necesarias con las ya aceptadas y rechazadas FN4 Si el número de usuarios que han rechazado desvelar el secreto es suficiente como para que no se pueda desvelar independientemente de la decisión de los miembros que aún no han decidido, el sistema actualiza el estado del secreto de nuevo a *oculto* FN5 El sistema elimina las partes compartidas que habían enviado los usuarios para desvelar y los rechazos
Flujo alternativo	FA3 Si el secreto aún tiene posibilidad de ser revelado el sistema mantiene su estado *pendiente*
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.32: CU-31. Rechazar desvelar secreto

Nombre e ID del CU	CU-32. Descargar archivo

Actor	Miembro identificado
Descripción	El actor descarga un archivo adjunto compartido por otro usuario
Precondiciónes	▪ PRE-1. El actor está identificado en el sistema ▪ PRE-2. El actor ha accedido a un grupo o diálogo ▪ PRE-3. Existe al menos un mensaje con archivo adjunto en el diálogo o grupo
Postcondiciones	▪ POST-1. El usuario obtiene el archivo adjunto
Flujo normal	FN1 El actor indica que desea descargar un archivo FN2 El sistema descarga y descifra el archivo FN3 El sistema da la oportunidad de almacenar el archivo descargado al usuario
Flujo alternativo	FA3 Si el archivo es una imagen, el sistema la mostrará directamente al usuario *pendiente*.
Excepciones	N/A
Prioridad	Baja
Otra info	N/A

Cuadro 2.33: CU-32. Descargar archivo

2.1.2. Análisis de Requisitos

A partir de la definición de la visión y alcance junto con el análisis previo de los casos de uso se van a definir los requisitos funcionales y no funcionales de la aplicación, junto con otras restricciones y suposiciones.

Entorno Operativo

OE-1. El producto deberá funcionar correctamente en los navegadores Firefox 38+, Chrome 42+, Safari 8+ y Opera30+ al menos

OE-2. El producto deberá funcionar sobre un servidor web Tomcat8

Restricciones de diseño e implementación

CO-1. El código HTML debe respetar el estandar HTML5

CO-2. Toda a comunicación de datos susceptibles entre el cliente y el servidor debe transmitirse cifrada

CO-3. El sistema debe usar MySQL como tecnología de base de datos

CO-4. El código del servidor debe implementarse en lenguaje Java para facilitar la reutilización de código

Requisitos funcionales

Esta sección define los requisitos funcionales del sistema agrupados por cada caso de uso del análisis anterior. Algunos requisitos funcionales afectan a más de un caso de uso, pero para evitar la repetición, solo se especificará su primera aparición.

CU-01. Solicitar registro (Cuadro 2.2)

RF-1.1: Comprobar credenciales:
El sistema comprobará que el usuario ha introducido todos los datos de registro.

RF-1.2: Comprobar usuario existente:
El sistema comprobará que, en el momento del registro, el nombre de usuario y correo electrónico no han sido registrados previamente.

RF-1.3: Comprobar usuario existente:
El sistema comprobará que el usuario no se encuentre en estado *bloqueado*.

RF-1.4: Almacenar solicitud de registro:
Si los datos de registro son correctos, el sistema almacena los datos del usuario. El usuario queda a la espera de aprobación por parte de un administrador.

RF-1.5: Registro de administrador:
Si al registrarse el usuario no hay otros usuarios registrados en la aplicación, sus datos se almacenan, pasa directamente al estado de *activación* y tiene permisos de administrador.

RF-1.6: Denegar solicitud de registro:
Si los datos de registro son incorrectos, el sistema no almacena la solicitud e informa al usuario del error correspondiente.

CU-02. Activar cuenta (Cuadro 2.3)

RF-2.1: Comprobar datos de activación:

El sistema comprobará que el código incluido en el enlace de activación coincide con los datos de activación almacenados en la BD.

RF-2.2: Generar clave RSA:

Si los datos de activación son correctos el sistema debe generar una clave RSA para el usuario.

RF-2.3: Comprobar contraseña:

El sistema comprobará que la contraseña introducida por el usuario para cifrar la clave RSA se ajusta a las normas de seguridad.

RF-2.4: Cifrar clave RSA:

Si la contraseña introducida por el usuario es válida sistema deberá cifrar la clave RSA del usuario empleando la contraseña introducida por este.

RF-2.5: Almacenar clave RSA:

El sistema debe almacenar la clave RSA del usuario cifrada en un archivo en la localización elegida por el usuario.

RF-2.6: Activar usuario:

Si la clave RSA se almacena de forma segura en el dispositivo del usuario, el sistema debe cambiar el estado del usuario a *activo*, almacenar su clave pública en la BD y eliminar sus datos de activación.

RF-2.7: Error de activación:

El sistema debe informar al usuario de cualquier error que impida el proceso normal de activación y abortarla.

CU-03. Iniciar sesión (Cuadro 2.4)

RF-3.1: Comprobar datos de inicio de sesión:

El sistema comprobará la introducción de todos los datos de inicio de sesión necesarios.

RF-3.2: Comprobar existencia de usuario:

El sistema comprobará que los credenciales de inicio de sesión se corresponden con un usuario.

RF-3.3: Comprobar estado de usuario:

Si el usuario existe, el sistema comprobará que se encuentra en estado *activo*.

RF-3.4: Descifrar clave RSA:

Si los datos de login son correctos, el sistema debe emplear la contraseña introducida por el usuario para descifrar la clave RSA del archivo indicado por el usuario.

RF-3.5: Comprobar identidad:

El sistema comprobará la identidad del usuario empleando la clave introducida por el usuario y la clave pública correspondiente a los credenciales introducidos por el usuario.

RF-3.6: Iniciar sesión:

Si los credenciales introducidos por el usuario, el sistema permite el acceso como *miembro identificado* al usuario.

RF-3.7: Impedir login:

Si los credenciales introducidos por el usuario no son correctos, el sistema denegará el acceso e informará del error correspondiente al usuario.

CU-04. Aceptar solicitud de registro (Cuadro 2.5)

RF-4.1: Comprobar administrador:

El sistema comprobará certificará la identidad del usuario para confirmar su estatus de administrador.

RF-4.2: Listar solicitudes de registro

El sistema mostrará al administrador la lista de solicitudes de registro.

RF-4.3: Aceptar solicitud:

Si se verifica la identidad del usuario administrador, el sistema genera un código de activación y lo almacena.

RF-4.4: Notificar activación:

Si la activación se acepta correctamente, el sistema genera un enlace de activación con el código y lo notifica al usuario vía correo electrónico.

RF-4.5: Operación restringida:

Si no se verifica la identidad del usuario administrador, el sistema deniega la operación e informa al usuario del error.

CU-05. Denegar solicitud de registro (Cuadro 2.6)

RF-5.1: Denegar solicitud:

Si se verifica la identidad del usuario administrador, el sistema elimina los datos de registro del usuario que envió la solicitud.

CU-06. Bloquear registro (Cuadro 2.7)

RF-6.1: Bloquear registro:

Si se verifica la identidad del usuario administrador, el sistema cambia el estado del usuario que envió la solicitud a *bloqueado*, impidiendo posteriores solicitudes con esos mismos datos.

CU-07. Exportar datos locales (Cuadro 2.8)

RF-7.1: Exportar datos locales:

El sistema permitirá exportar los datos del almacén local a un archivo en la localización elegida por el usuario.

CU-08. Importar datos locales (Cuadro 2.9)

RF-8.1: Comprobar estructura de archivo:

El sistema comprobará que la estructura del archivo a importar es correcta.

RF-8.2: Importar datos locales:

Si la estructura del archivo indicado por el usuario es correcta, el sistema importará los datos contenidos en el mismo dentro del almacén local.

CU-09. Cerrar sesión (Cuadro 2.10)

RF-9.1: Cerrar sesión:

El sistema elimina los datos de sesión del usuario y cierra la sesión, denegando el acceso a las funciones accesibles como *miembro identificado*.

CU-10. Crear grupo (Cuadro 2.11)

RF-10.1: Comprobar validez de nombre:

El sistema comprobará que el usuario haya introducido un nombre de grupo.

RF-10.2: Comprobar existencia de grupo:

El sistema comprobará que no exista un grupo con el nombre de grupo indicado por el usuario.

RF-10.3: Crear grupo:

Si el nombre de grupo es válido, el sistema almacenara en la BD el nuevo grupo y el usuario creador como miembro líder.

RF-10.4: Generar clave de grupo:

El sistema generará una clave para el nuevo grupo y la guardará en el almacén local del usuario creador.

RF-10.5: Denegar creación de grupo:

Si ya existe un grupo con el nombre indicado por el usuario, el sistema cancela la creación del grupo e informa al usuario.

CU-11. Dejar grupo (Cuadro 2.12)

RF-11.1: Verificar identidad:

El sistema verificará la identidad del usuario al acceder a la acción de dejar el grupo.

RF-11.2: Dejar grupo:

Si la identidad del usuario se verifica y el usuario no es líder del grupo, el sistema elimina su información de miembro del grupo seleccionado. El sistema elimina la clave de grupo del almacén local.

RF-11.3: Evitar salida de lider:

Si el usuario que desea dejar el grupo es lider del mismo y hay otros miembros en el grupo, el sistema impide que abandone el grupo y le notifica que debe escoger otro líder antes de abandonar el grupo.

RF-11.4: Disolver grupo:

Si el usuario es el único miembro del grupo, al dejarlo el sistema elimina el grupo.

CU-12. Aceptar invitación a grupo (Cuadro 2.13)

RF-12.1: Aceptar invitación a grupo:

Si se verifica la identidad del usuario, el sistema almacena la clave de grupo localmente y los datos de miembro de grupo en la BD.

CU-13. Invitar a grupo (Cuadro 2.14)

RF-13.1: Comprobar lider:

El sistema comprobará que el usuario es líder del grupo antes de ejecutar tareas de gestión de grupo.

RF-13.2: Invitar a grupo:

Si el usuario que ejecuta la acción es líder del grupo, el sistema almacenará la invitación en la BD junto con la clave de grupo de forma segura.

CU-14. Transferir liderazgo (Cuadro 2.15)

RF-14.1: Transferir liderazgo:

Si el usuario que ejecuta la acción es líder del grupo, el sistema eliminará los permisos de líder y se los otorgará al otro miembro seleccionado.

CU-15. Expulsar de grupo (Cuadro 2.16)

RF-15.1: Expulsar de grupo:

Si el usuario que ejecuta la acción es líder del grupo, el sistema eliminará los datos del miembro seleccionado.

CU-16. Listar grupos (Cuadro 2.17)

RF-16.1: Listar grupos:

El sistema mostrará una lista de los grupos a los que pertenece actualmente el usuario

RF-16.2: Sin grupos:

Si el usuario no pertenece a ningún grupo, el sistema mostrara un mensaje informando de ello.

CU-17. Filtrar grupos (Cuadro 2.18)

RF-17.1: Filtrar grupos:

El sistema mostrará una lista por nombre atendiendo al criterio de filtro que introduzca el usuario de los grupos a los que pertenece.

CU-18. Acceder a grupo (Cuadro 2.19)

RF-18.1: Comprobar membresía:

El sistema comprobará que el usuario que intenta acceder al grupo es miembro del mismo.

RF-18.2: Denegar acceso a grupo:

Si el usuario no es miembro del grupo el sistema denegará el acceso al grupo.

RF-18.3: Acceder a grupo:

Si el usuario es miembro del grupo, el sistema permitirá el acceso al grupo.

RF-18.4: Mostrar mensajes de grupo:

El sistema descifrará los mensajes del grupo con la clave de grupo y los mostrará.

RF-18.5: Listar miembros:

El sistema mostrará la lista de los miembros del grupo y sus características.

RF-18.6: Listar usuarios no miembros:

Si el actor es líder del grupo, el sistema mostrará una lista de los usuarios no miembros.

RF-18.7: Mostrar secretos:

El sistema mostrará la lista de secretos del grupo y su estado.

RF-18.8: Mostrar archivos adjuntos de grupo:

El sistema mostrará para cada mensaje o secreto revelado si tienen un archivo adjunto.

RF-18.9: Descargar archivo adjunto de grupo:

El sistema descargará y descifrará el archivo adjunto.

RF-18.10: Descargar partes compartidas de secretos:

El sistema descargará, descifrará y almacenará localmente las partes de secretos de grupo y las eliminará de la BD.

RF-18.11: Actualizar último mensaje recibido de miembro:

El sistema actualizará la información de último mensaje recibido por el usuario.

RF-18.12: Actualización periódica de grupo:

El sistema mantendrá actualizado al cliente con los datos del grupo periódicamente.

CU-19. Listar usuarios (Cuadro 2.20)

RF-19.1: Listar usuarios:

El sistema muestra una lista de todos los usuarios de la aplicación.

RF-19.2: No hay otros usuarios:

Si no existen otros usuarios, el sistema muestra un mensaje informándolo.

CU-20. Filtrar usuarios (Cuadro 2.21)

RF-20.1: Filtrar usuarios:

El sistema mostrará una lista por filtrada por nombre de usuario, nombre, apellido o correo electrónico atendiendo al criterio de filtro que introduzca el usuario de los otros usuarios del sistema.

CU-21. Solicitar diálogo (Cuadro 2.22)

RF-21.1: Generar clave de diálogo:

El sistema generará y almacenará localmente una clave para el nuevo diálogo.

RF-21.2: Comprobar validez de clave de diálogo:

El sistema comprobará que la clave de diálogo se ha recibido correctamente en el cliente.

RF-21.3: Almacenar solicitud:

Si la clave es correcta, el sistema almacenará la solicitud de diálogo junto con la clave de diálogo de forma segura.

CU-22. Aceptar diálogo (Cuadro 2.23)

RF-22.1: Aceptar solicitud de diálogo:

Si la identidad del usuario se verifica, el sistema almacenará la clave de diálogo localmente y creará el nuevo diálogo. El sistema elimina la solicitud de la BD.

CU-23. Mostrar estado de comunicaciónes (Cuadro 2.24)

RF-23.1: Listar diálogos:

El sistema mostrará un listado de los diálogos activos del usuario.

RF-23.2: Mostrar estado de diálogos:

El sistema mostrará la fecha del último mensaje y el número de mensajes sin leer de cada diálogo.

RF-23.3: Mostrar estado de grupos:

El sistema mostrará la fecha del último mensaje y el número de mensajes sin leer de cada grupo.

RF-23.4: Listar solicitudes de diálogo:

El sistema mostrará una lista de las solicitudes de diálogo del usuario.

RF-23.5: Listar invitaciones de grupo:

El sistema mostrará una lista de las invitaciones de grupo del usuario.

CU-24. Acceder a diálogo (Cuadro 2.25)

RF-24.1: Comprobar identidad:

El sistema comprobará que el usuario es una de las dos partes del diálogo.

RF-24.2: Denegar acceso a diálogo:

Si el usuario no es parte del diálogo, el sistema denegará el acceso.

RF-24.3: Acceder a diálogo:

Si el usuario no es parte del diálogo, el sistema permitirá el acceso.

RF-24.4: Mostrar mensajes de diálogo:

El sistema descifrará los mensajes del diálogo con la clave de diálogo y los mostrará.

RF-24.5: Mostrar archivos adjuntos de grupo:

El sistema mostrará para cada mensaje o secreto revelado si tienen un archivo adjunto.

RF-24.6: Descargar archivo adjunto de grupo:

El sistema descargará y descifrará el archivo adjunto.

RF-24.7: Actualizar último mensaje recibido de diálogo:

El sistema actualizará la información de último mensaje recibido por el usuario.

RF-24.8: Actualización periódica de diálogo:

El sistema mantendrá actualizado al cliente con los datos del diálogo periódicamente.

CU-25. Mandar mensaje de diálogo (Cuadro 2.26)

RF-25.1: Comprobar mensaje:

El sistema comprobará que el mensaje no esté vacío.

RF-25.2: Mandar mensaje de diálogo:

Si el mensaje no está vacío, el sistema envía el mensaje cifrado con la clave de diálogo al servidor y lo almacena en la BD.

RF-25.3: Mandar archivo adjunto de diálogo

Si el usuario ha seleccionado un archivo para adjuntar, el sistema lo cifra y envía al servidor junto con el mensaje de diálogo.

CU-26. Mandar mensaje de grupo (Cuadro 2.27)

RF-26.1: Mandar mensaje de diálogo:

Si el mensaje no está vacío, el sistema envía el mensaje cifrado con la clave de grupo al servidor y lo almacena en la BD.

RF-26.2: Mandar archivo adjunto de grupo

Si el usuario ha seleccionado un archivo para adjuntar, el sistema lo cifra y envía al servidor junto con el mensaje de grupo.

CU-27. Mandar secreto (Cuadro 2.28)

RF-27.1: Mandar mensaje secreto:

Si el mensaje no está vacío, el sistema lo envía al servidor, genera una clave secreta y lo cifra con ella. El sistema almacena el mensaje secreto en la BD en estado *oculto*.

RF-27.2: Dividir secreto:

El sistema divide la clave secreta en tantas partes como miembros del grupo y las almacena en la BD de forma segura.

RF-27.3: Mandar archivo adjunto de secreto:

Si el usuario ha seleccionado un archivo para adjuntar, el sistema lo cifra y envía al servidor junto con el mensaje secreto.

CU-28. Adjuntar archivo (Cuadro 2.29)

RF-28.1: Adjuntar archivo:

El sistema almacena la localización del archivo seleccionado por el usuario para su posterior envío junto con un mensaje de diálogo, grupo o secreto.

RF-28.2: Cancelar adjuntar:

Si el usuario cancela la selección de archivo, este no se adjunta al mensaje.

CU-29. Proponer desvelar secreto (Cuadro 2.30)

RF-29.1: Secreto pendiente:

El sistema actualiza el estado del secreto a *pendiente*.

RF-29.2: Almacenar parte compartida:

El sistema envía la parte compartida del usuario al servidor y la almacena en la BD indicando que el usuario está a favor de desvelar el secreto

CU-30. Aceptar desvelar secreto (Cuadro 2.31)

RF-30.1: Aceptar desvelar secreto:

El sistema envía la parte compartida del usuario al servidor y la almacena en la BD indicando que el usuario está a favor de desvelar el secreto.

RF-30.2: Comprobar posibilidad de desvelar:

El sistema comprueba que las partes compartidas enviadas por los miembros del grupo son suficientes para desvelar el secreto.

RF-30.3: Desvelar secreto:

Si las partes compartidas enviadas por los miembros del grupo son suficientes, el sistema recupera la clave secreta, descifra con ella el mensaje secreto y lo almacena en la BD, actualizando su estado a *desvelado*.

CU-31. Rechazar desvelar secreto (Cuadro 2.32)

RF-31.1: Rechazar desvelar secreto:

El sistema almacena el rechazo a desvelar el secreto por parte del usuario.

RF-31.2: Comprobar imposibilidad de desvelar:

El sistema comprueba que el secreto aún tiene posibilidad de desvelarse teniendo en cuenta las partes compartidas ya enviadas por miembros del gupo, las partes restantes y los miembros que quedan por elegir.

RF-31.3: Cancelar desvelar secreto:

Si el secreto no tiene oportunidad de ser desvelado, el sistema elimina de la BD las partes compartidas enviadas por los miembros que estaban a favor y actualiza el estado del secreto a *oculto*.

CU-32. Descargar archivo (Cuadro 2.33)

RF-32.1: Descargar archivo:

El sistema descarga el archivo adjunto seleccionado.

RF-32.2: Descifrar archivo:

El sistema descifrará el archivo empleando la clave de diálogo o grupo correspondiente.

RF-32.3: Mostrar archivo:

El sistema dará la oportunidad al usuario de descargar el archivo una vez se haya descifrado.

RF-32.4: Mostrar imagen:

Si el archivo descargado es una imagen, el sistema la mostrará junto con el mensaje al que va adjunto.

Requisitos de interfaz de usuario

UI-1: El sistema permitirá al usuario en todo momento cambiar el idioma en que se muestra el texto del sistema.

Requisitos de interfaz de comunicación

SI-1: El sistema enviará un correo electrónico a la dirección registrada por un usuario cuando un administrador acepte su solicitud, incluyendo un enlace de activación de cuenta.

Requisitos de información

A continuación se indican los datos relevantes de cada una de las entidades del sistema.

IRQ-1: Datos de usuario:

a) Nombre de usuario
b) Nombre
c) Apellidos
d) Email
e) Estado
f) Administrador
g) Clave pública
h) Clave privada

IRQ-2: Datos de grupo:

a) Nombre
b) Clave

IRQ-3: Datos de diálogo:

a) Clave

IRQ-4: Datos de mensajes:

a) Mensaje
b) Hora enviado

IRQ-5: Datos de achivos:

a) Nombre
b) Ubicación
c) Tamaño original
d) Tipo mime

IRQ-6: Datos de secreto:

a) Mensaje
b) Archivo adjunto

c) Hora creado

d) Mínimo para desvelar

e) Numero de partes

f) Nº primo secreto

IRQ-7: Datos de pates compartidas:

a) Contenido

b) Numero de parte

IRQ-8: Datos de invitaciónes:

a) Clave

2.1.3. Atributos de calidad

Rendimiento

PER-1: El tiempo de espera de un usuario al acceder a un grupo o diálogo debe ser inferior a 5s

PER-2: El tiempo de envío de los mensajes debe ser inferior a 2s

PER-3: El tiempo de envío de los archivos debe ser inferior a 2min

PER-4: El tiempo de espera al enviar o aceptar cualquier invitación debe ser inferior a 5s

Seguridad

SEC-1: La contraseña de protección de la clave RSA de usuario deben tener al menos 6 caracteres

SEC-2: La contraseña de protección de la clave RSA de usuario nunca se transmitirá por la red.

SEC-3: Toda transmisión de datos susceptibles (en especial claves) debe realizarse cifrada.

SEC-4: La identidad del usuario debe verificarse antes de realizar operaciones susceptibles a la suplantación de identidad.

SEC-5: Las claves se almacenarán en la BD solo cuando sea estrictamente necesario y siempre cifradas.

SEC-6: El sistema de cifrado de clave pública empleado será RSA

SEC-7: Las claves RSA generadas serán de 1024 bits

SEC-8: El sistema de cifrado simétrico empleado será AES128

SEC-9: El sistema de compartición de secretos será Shamir mejorado

SEC-10: El sistema solo almacenará la clave pública RSA en la BD

SEC-11: Los datos de sesión serán eliminados siempre al cerrar la sesión

SEC-12: Los datos guardados en el almacén local del sistema siempre se guardarán cifrados

SEC-13: Toda transacción con la BD del sistema estará contra SQLInjection

SEC-14: La negociación de claves de usar y tirar se efectuará siguiendo el protocolo Diffie-Hellman

Robustez

ROB-1: Si el cliente pierde la conexión con el servidor durante una operación, el sistema debe abortar la operación e informar al usuario.

ROB-2: El sistema debe sugerir al usuario que exporte sus datos locales si se han realizado cambios desde la ultima vez que los exportó.

2.1.4. Modelos de análisis

Diagramas de estados

Los siguientes diagramas muestran los estados de *usuarios* y *secretos* y las transiciones posibles entre sus diferentes estados.

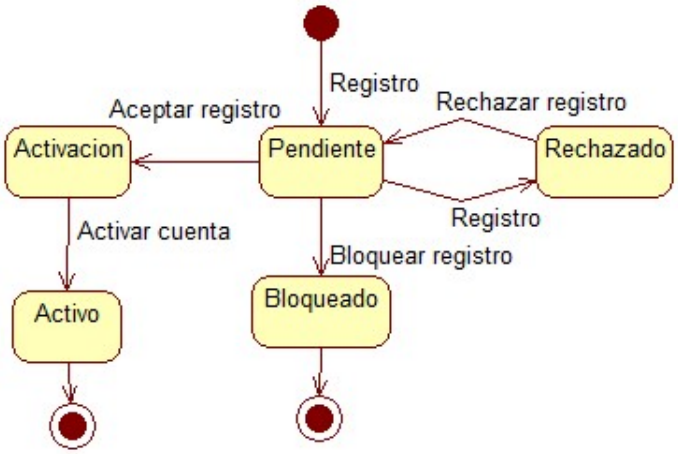

Figura 2.4: Diagrama de estados de usuarios

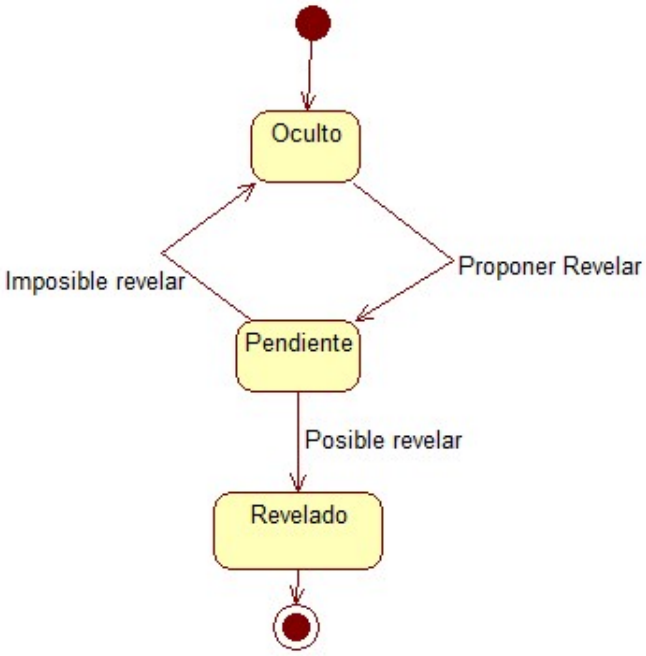

Figura 2.5: Diagrama de estados de secretos

2.2. Diseño

Para el diseño de la aplicación he seguido un proceso orientado a eventos y, dado que la seguridad se reparte entre el cliente y el servidor, va a haber una gran interacción entre estos, siendo necesario diseñar principalmente dichas comunicaciones, junto con los datos necesarios para ello.

2.2.1. Diseño de datos

A partir de los requisitos de información se deriva un modelo conceptual de los datos requeridos por la aplicación y sus relaciones, que posteriormente será implementado en la base de datos real.

Modelo entidad-relación

El diagrama entidad-relación representa entidades, elementos del mundo real que poseen una serie de atributos que se desea almacenar en la BD, y las relaciones entre ellos. Es un modelo conceptual que posteriormente debe ser traducido para implementarlo en la BD real.

Aparte de los atributos detallados en los requisitos de información, las entidades poseen atributos necesarios para la lógica de negocio que inicialmente no habían sido planteados, como la hora del último mensaje leído por un miembro.

Modelo relacional

A partir del modelo entidad-relación se deriva el modelo relacional. Este modelo representa los datos en una estructura que pueda ser implementada en una base de datos, representando las entidades como tablas y sus atributos. Además representa las relaciones entre las tablas. Puede darse el caso de que una relación del modelo entidad-relación se transforme también en una tabla. Esto ocurre cuando la relación es N:M o en el caso de una agregación (que representa una relación entre relaciones, como es el caso del miembro de grupo del diagrama **??**).

El diagrama también representa la acción que debe tomar el gestor de bases de datos al actualizar o eliminar una tupla con la que se relaciona otra. Esto se define con dos letras; la primera indica la acción (actualización U y eliminación D) y la segunda la reacción (en cascada C, rechazar R o anular la clave foranea N).

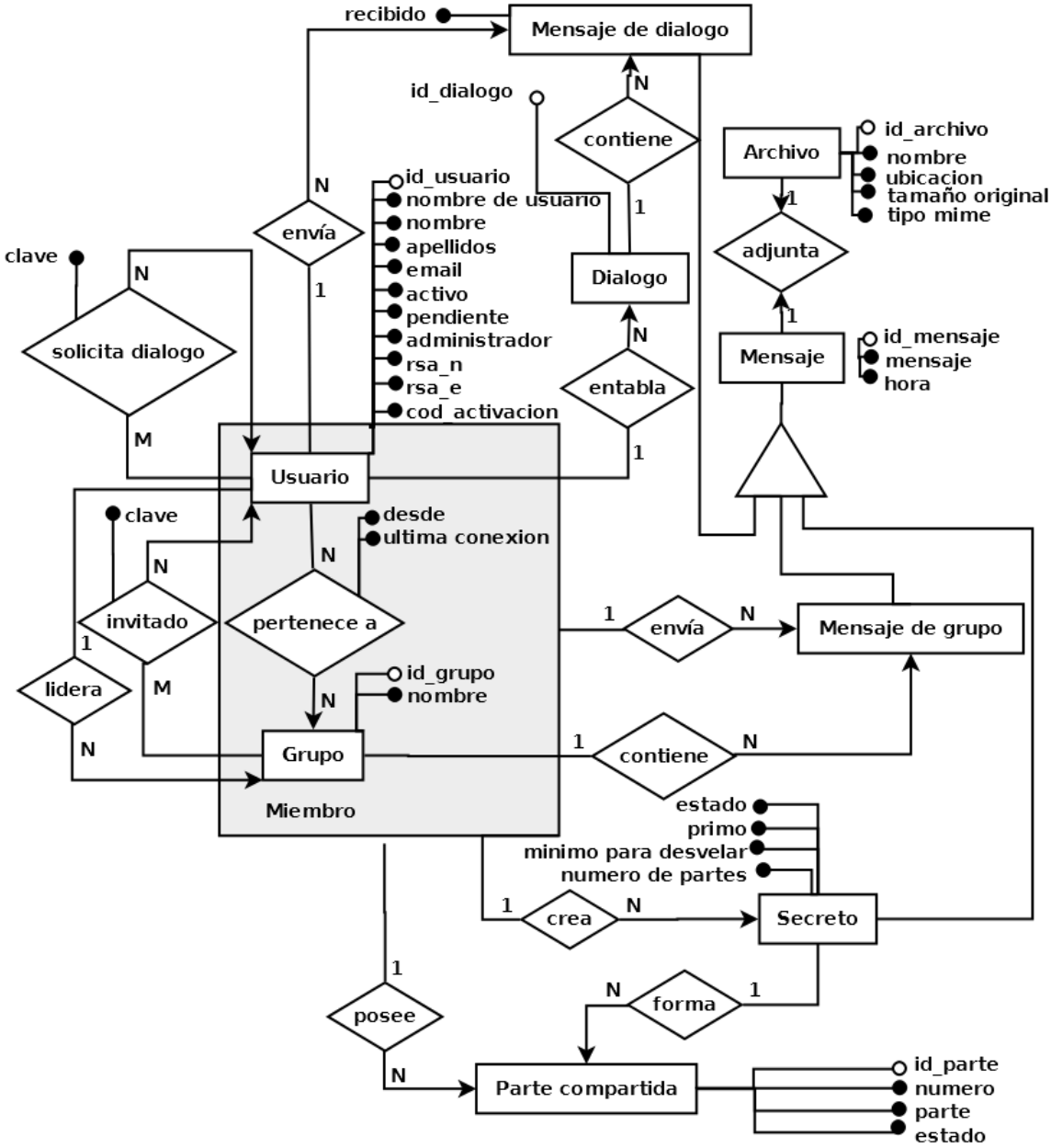

Figura 2.6: Diagrama Entidad-Relación de la BD

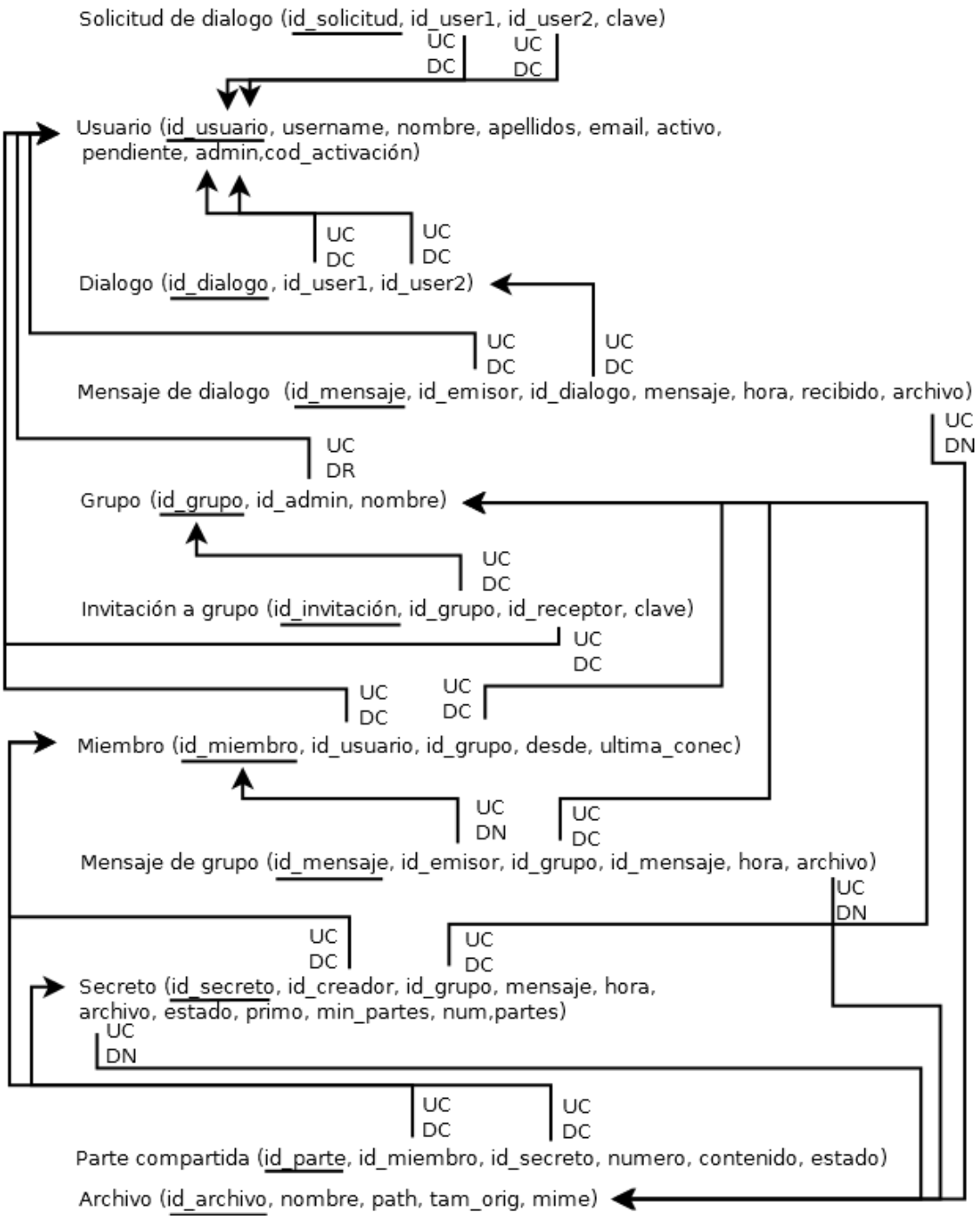

Figura 2.7: Modelo Relacional de la BD

Diccionario de datos

El diccionario de datos representa con más detalle la estructura de la BD detallando los atributos.

Relación "solicitud de diálogo" (*dialog-invite*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único de solicitud
id_from	int(11)	No	FK(user->id)	-	Id del emisor de solicitud
id_to	int(11)	No	FK(user->id)	-	Id del receptor de solicitud
key	varbinary(512)	No	-	-	Clave de diálogo cifrada

Cuadro 2.34: Datos de la entidad "solicitud de diálogo"

Entidad "usuario" (*user*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del usuario
username	varchar(30)	No	UNIQUE	-	Nombre de usuario
rsa_n	varbinary(512)	No	-	0x00	Modulo de la clave RSA del usuario
rsa_e	varbinary(512)	No	-	0x10001	Parte pública de la clave RSA de usuario
name	varchar(30)	No	-	-	Nombre propio del usuario
surname	varchar(120)	No	-	-	Apellidos del usuario
email	varchar(50)	No	UNIQUE	-	Correo electrónico del usuario
active	tinyint(1)	No	-	0	Booleano que indica si el usuario está activo
pending	tinyint(1)	No	-	1	Booleano que indica si el registro de usuario está pendiente de ser gestionada por un administrador
admin	tinyint(1)	No	-	0	Booleano que indica si el usuario es administrador
cod_activ	varchar(100)	Si	-	NULL	Código de activación del usuario

Cuadro 2.35: Datos de la entidad "usuario"

Entidad ”diálogo” (*dialog*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del diálogo
id_user1	int(11)	No	-	-	Id del usuario 1 del diálogo
id_user2	int(11)	No	-	-	Id del usuario 1 del diálogo

Cuadro 2.36: Datos de la entidad ”diálogo”

Entidad ”Mensaje de diálogo” (*dialog-message*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del mensaje
id_from	int(11)	No	FK(user->id)	-	Id del emisor del mensaje
id_dialog	int(11)	No	FK(dialog->id)	-	Id del diálogo al que pertenece el mensaje
message	text	No	-	-	Contenido textual del mensaje cifrado y en base64
time	timestamp	No	-	CUR_TIME	Fecha y hora de creación del mensaje
received	tinyint(1)	No	-	0	Booleano que representa si el mensaje ha sido recibido por el receptor
file	int(11)	Si	FK(file->id)	NULL	Id del archivo adjunto

Cuadro 2.37: Datos de la entidad ”Mensaje de diálogo”

Entidad ”grupo” (*group*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del grupo
id_admin	int(11)	No	FK(user->id)	-	Id del usuario líder del grupo
name	varchar(30)	No	-	-	Nombre del grupo

Cuadro 2.38: Datos de la entidad ”grupo”

Entidad ”invitación a grupo” (*group-invite*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único de la invitación
id_group	int(11)	No	FK(group->id)	-	Id del grupo de la invitación
id_to	int(11)	No	FK(user->id)	-	Id del usuario invitado

Cuadro 2.39: Datos de la entidad ”invitación a grupo”

Relación ”miembro” (*member*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del miembro
id_user	int(11)	No	FK(user->id)	-	Id del usuario miembro del grupo
id_group	int(11)	No	FK(group->id)	-	Id del grupo del que es miembro
since	timestamp	No	-	CUR_TIME	Fecha y hora en que se unió al grupo
lastreceived	int(11)	No	-	0000-00-00 00:00:00	Hora en que se le mostró el último mensaje

Cuadro 2.40: Datos de la entidad ”miembro”

Entidad ”Mensaje de grupo” (*group-message*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del mensaje
id_from	int(11)	No	FK(user->id)	-	Id del emisor del mensaje
id_group	int(11)	No	FK(group->id)	-	Id del grupo al que pertenece el mensaje
message	text	No	-	-	Contenido textual del mensaje cifrado y en base64
time	timestamp	No	-	CUR_TIME	Fecha y hora de creación del mensaje

file	int(11)	Si	FK(file->id)	NULL	Id del archivo adjunto

Cuadro 2.41: Datos de la entidad ”Mensaje de grupo”

Entidad ”secreto” (*secret*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del secreto
id_from	int(11)	No	FK(user->id)	-	Id del usuario que envió el secreto
id_group	int(11)	No	FK(group->id)	-	Id del grupo al que pertenece el secreto
message	text	No	-	-	Contenido textual del mensaje cifrado y en base64
time	timestamp	No	-	CUR_TIME	Fecha y hora de creación del secreto
file	int(11)	Si	FK(file->id)	NULL	Id del archivo adjunto
status	tinyint(2)	No	-	0	Estado del secreto. 0=Oculto, 1=Pendiente, 2=Revelado
p	varbinary(256)	No	-	-	Numero primo necesario para desvelar el secreto
min_shares	int(11)	No	-	3	Minimo de partes necesarias para desvelar el secreto
num_shares	int(11)	No	-	3	Partes en las que se ha dividido el secreto

Cuadro 2.42: Datos de la entidad ”secreto”

Entidad ”parte compartida” (*share*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único de la parte compartida
id_member	int(11)	No	FK(member->id)	-	Id del miembro al que pertenece la parte compartida

id_secret	int(11)	No	FK(secret->id)	-	Id del secreto al que corresponde la parte compartida
number	int(11)	No	-)	-	Numero de la parte compartida
share	varbinary(256)	No	-	-	Contenido de la parte compartida
status	tinyint(2)	No	-	0	Estado de la parte compartida. 0=No recibida, 1=Acepta revelar, 2=Rechaza revelar

Cuadro 2.43: Datos de la entidad ”parte compartida”

Entidad ”archivo” (*file*)					
Atributo	**Tipo**	**Nulo**	**Restricción**	**Valor inicial**	**Descripción**
id	int(11)	No	PK	Auto incremental	Id único del archivo
name	varchar(100)	No	-	-	Nombre del archivo
path	varchar(400)	No	-	-	Localización en el sistema de archivos del servidor del archivo
orig_size	int(11)	No	-	-	Tamaño original del archivo en bytes
mime	varchar(50)	No	-	-	Tipo MIME del archivo

Cuadro 2.44: Datos de la entidad ”archivo”

Almacen local

El almacen local es el espacio de almacenamiento dentro del cliente de cada usuario que la aplicación empleará para almacenar ciertos datos necesarios para garantizar que todo mensaje se envíe cifrado. El almacén local se implementará aprovechando el API de HTML5 LocalStorage, el cual provee un espacio de almacenamiento gestionado por el navegador. Los datos que se guardarán son los siguientes:

- **Claves de diálogo**: la clave de cada diálogo activo.
- **Claves de grupo**: la clave de cada grupo al que pertenece el usuario.
- **Partes compartidas**: el contenido y número de las partes de secreto compartidas con el usuario.
- **Cambios**: un booleano que indica si ha habido cambios desde el último backup de los datos locales.

Todos los elementos guardados en este almacén local deberán estar siempre cifrados.

Usuarios de la BD

Como capa adicional de seguridad, la BD tendrá definidos varios usuarios con acceso restringido a ciertas tablas, y se emplearán de forma selectiva dependiendo de las necesidades de cada operación.

Usuarios de la BD		
Usuario	**Permisos**	**Actividades**
secureaccept	▪ **member**: INSERT ▪ **group-invite**: SELECT, DELETE ▪ **dialog**: INSERT ▪ **user**: SELECT(id,rsa_n,rsa_e) ▪ **dialog-invite**: SELECT, DELETE	Para la acción de aceptar invitaciones

secureactiv	▪ **user**: SELECT(id, pending, active, cod_activ), UPDATE (pending, active, cod_activ)	Para la comprobar la el código de activación de un usuario
secureadmin	▪ **user**: SELECT, UPDATE (pending, active, cod_activ, admin)	Para realizar operaciónes de administrador
securecom	▪ **dialog-message**: SELECT, INSERT, UPDATE (received), DELETE ▪ **user**: SELECT ▪ **file**: SELECT, INSERT, UPDATE (path), DELETE ▪ **secret**: SELECT, INSERT, UPDATE (message, p, status), DELETE ▪ **dialog**: SELECT, INSERT, UPDATE ▪ **member**: SELECT, INSERT, UPDATE (lastreceived), DELETE ▪ **share**: SELECT (number, id_member, status, id, share, id_secret), INSERT (number, id_member, status, id, share, id_secret), UPDATE (status, share), DELETE ▪ **group-invite**: SELECT (id_to, id, id_group), INSERT, DELETE ▪ **group-message**: SELECT, INSERT, DELETE ▪ **dialog-invite**: SELECT(id_to, id, id_from), INSERT, DELETE ▪ **group**: SELECT, INSERT, UPDATE (name), DELETE	Para operaciones comunes de miembro identificado

securegp	▪ **user**: SELECT (rsa_e, id, rsa_n) ▪ **member**: SELECT, DELETE ▪ **group-invite**: SELECT (id_to, id, id_group), INSERT, DELETE ▪ **group**: SELECT, UPDATE (name, id_admin), DELETE	Para operaciones comunes de miembro identificado
securelogin	▪ **user**: SELECT, INSERT	Para operaciones comunes de miembro identificado

Cuadro 2.45: Usuarios de la BD

2.2.2. Diagramas de secuencia

Para un proyecto web como este, el mejor método de diseñar la aplicación es empleando diagramas de secuencia. Estos diagramas describen la secuencia de pasos que se llevan a cabo cuando un actor interactúa de alguna manera con el sistema.

Cada diagrama de secuencia suele ir unido con un caso de uso, y define los diferentes flujos del mismo con mayor detalle.

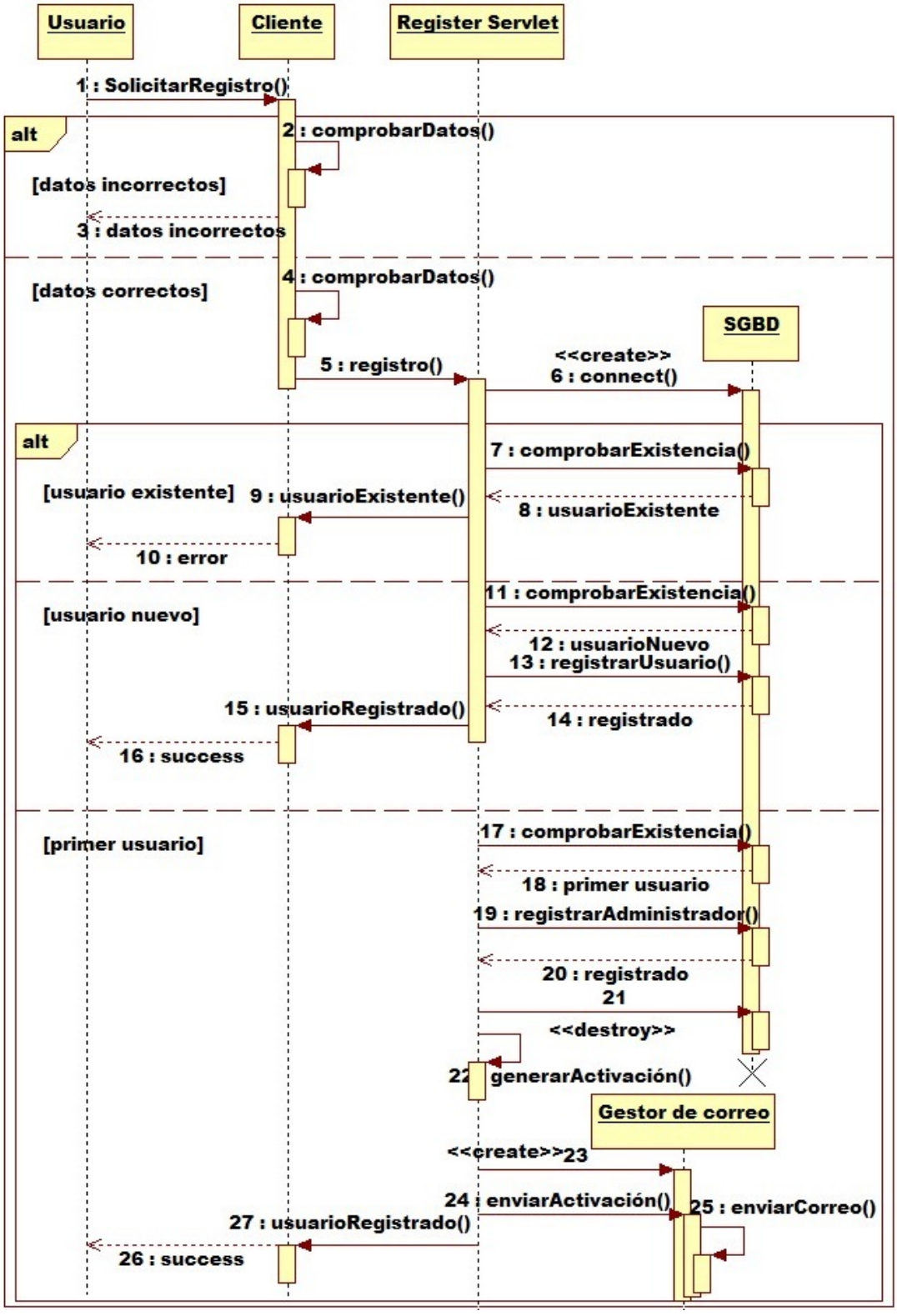

Figura 2.8: Diagrama de secuencia del CU-01 Solicitar Registro

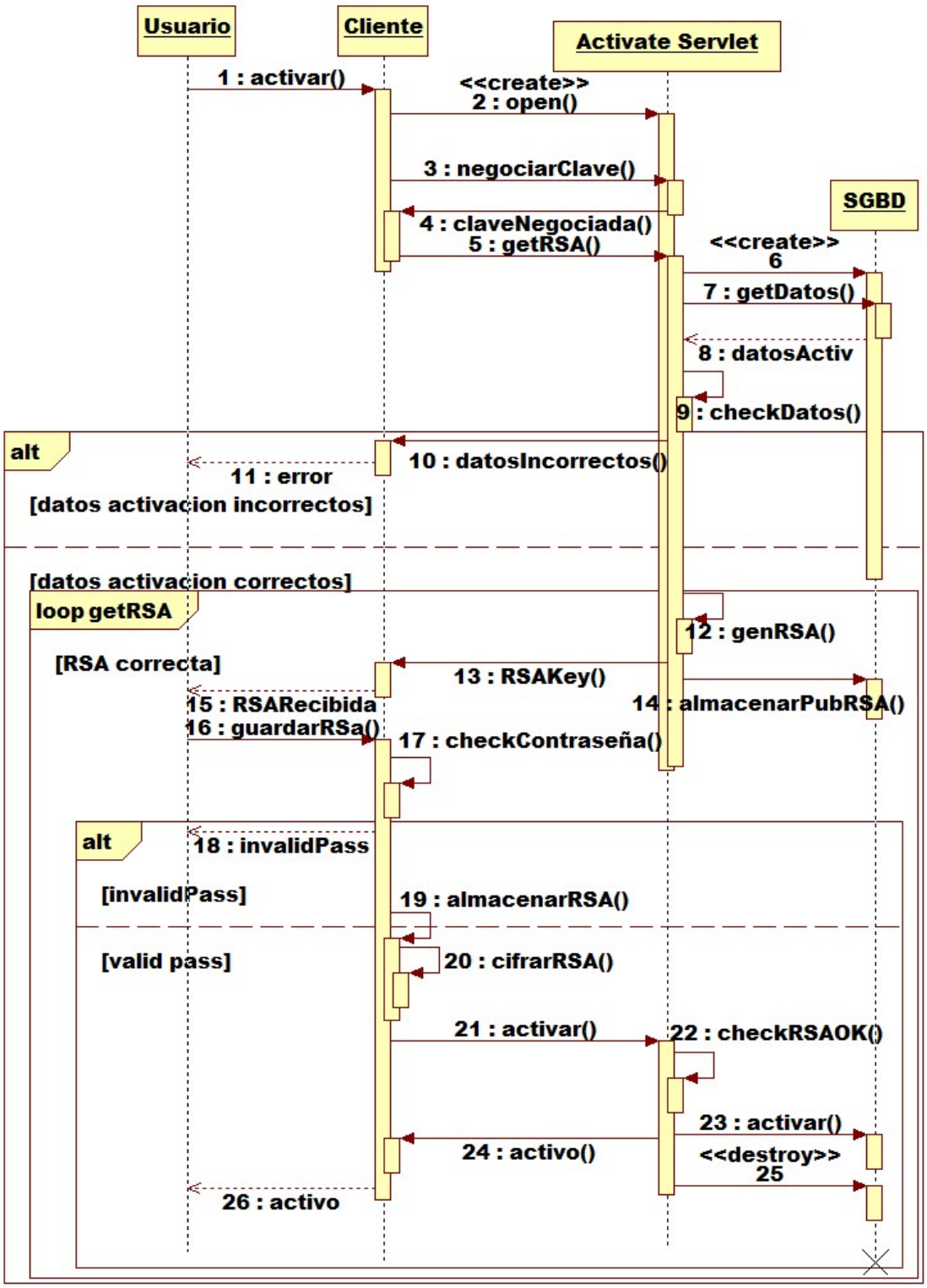

Figura 2.9: Diagrama de secuencia del CU-02 Activar Cuenta

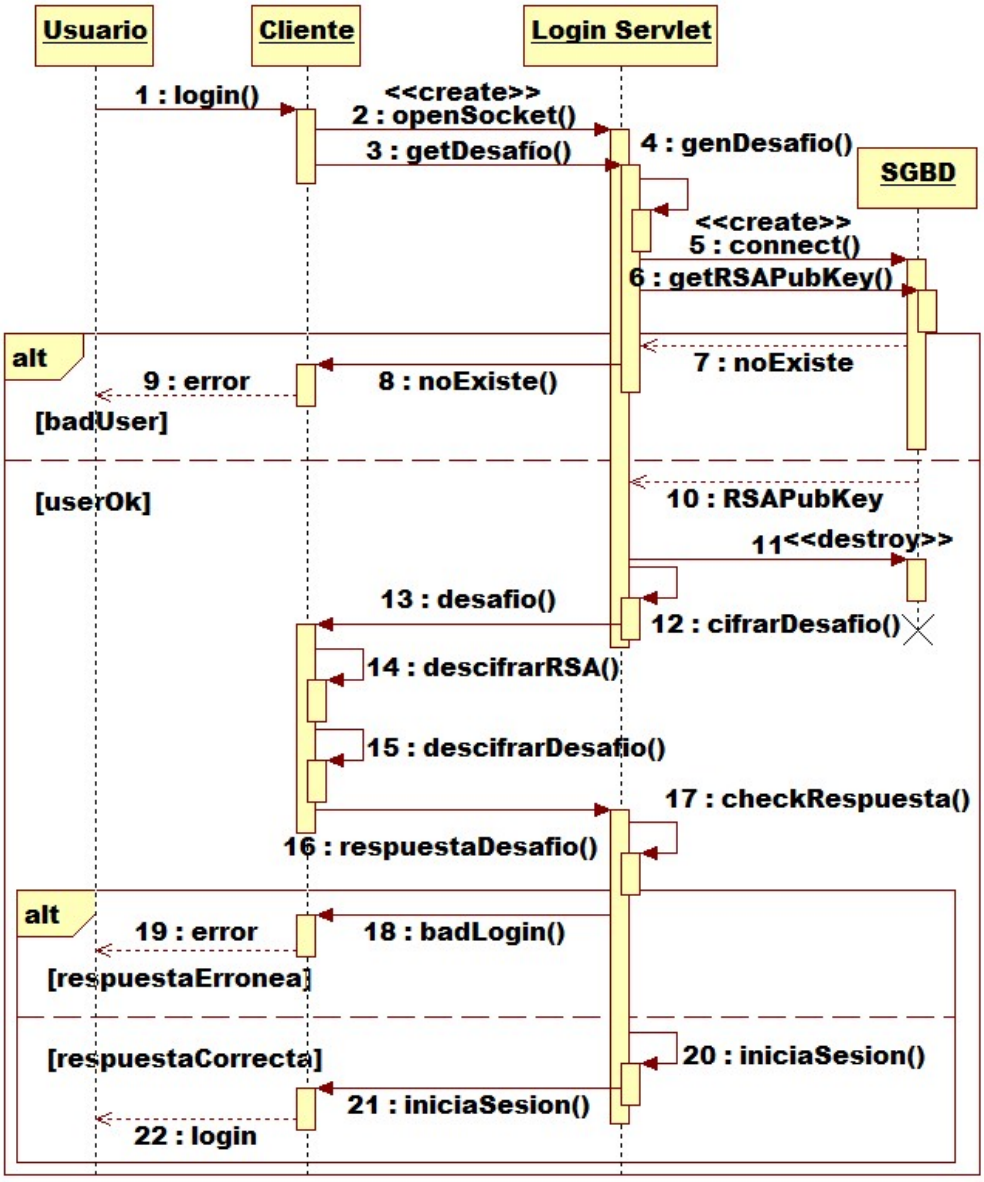

Figura 2.10: Diagrama de secuencia del CU-03 Iniciar sesión

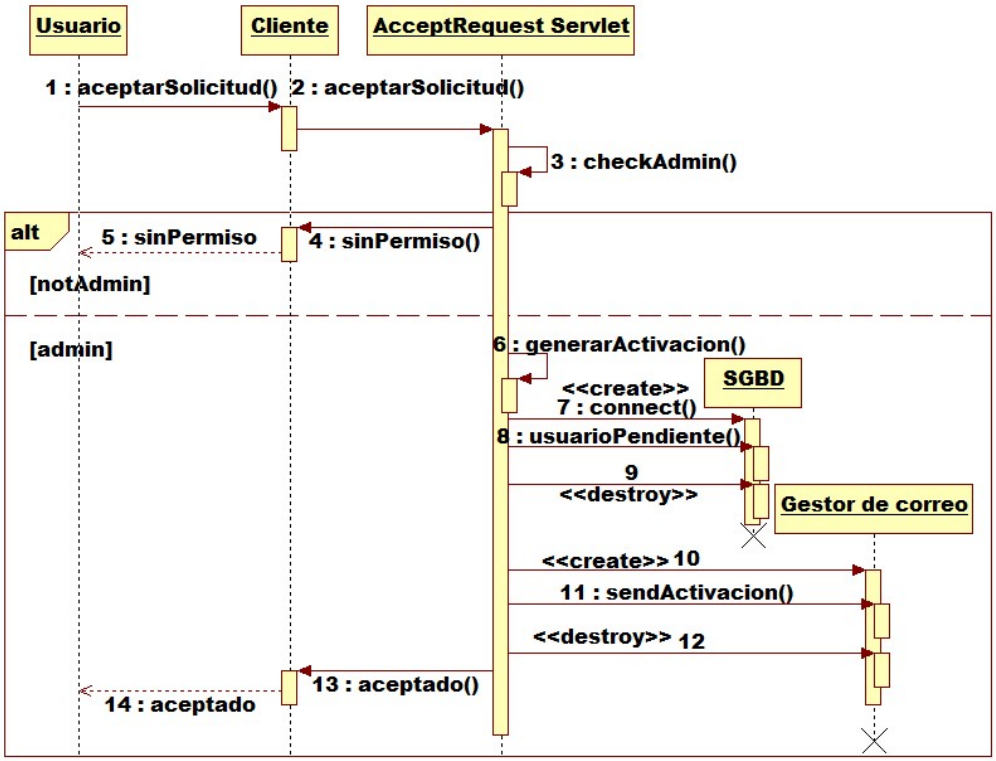

Figura 2.11: Diagrama de secuencia del CU-04 Aceptar solicitud de registro

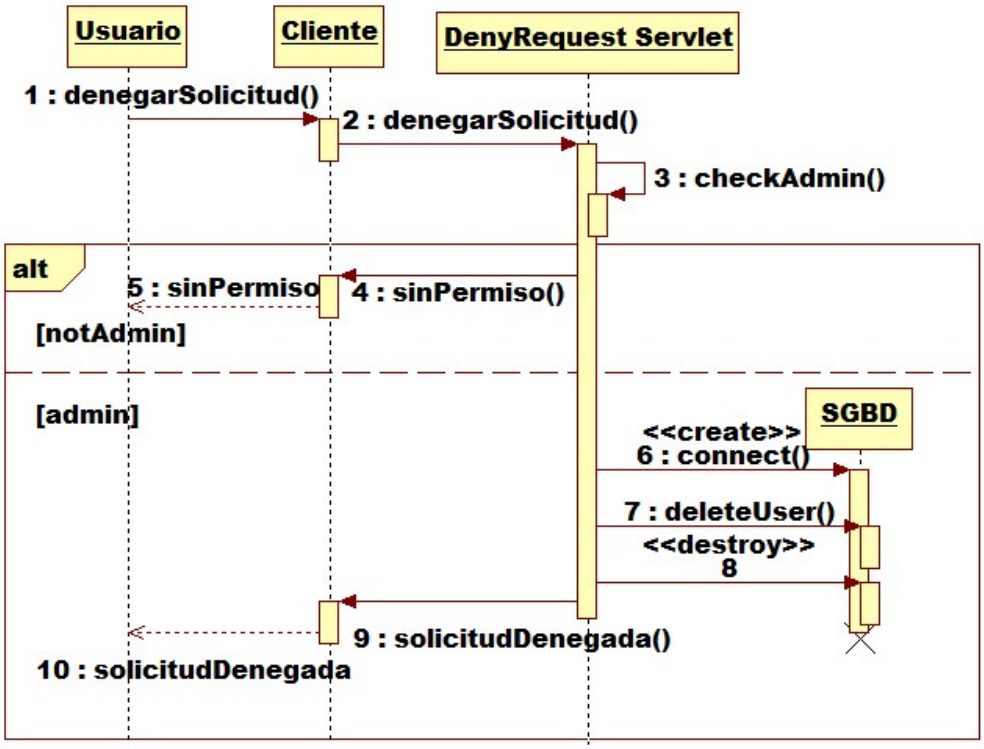

Figura 2.12: Diagrama de secuencia del CU-05 Denegar solicitud de registro

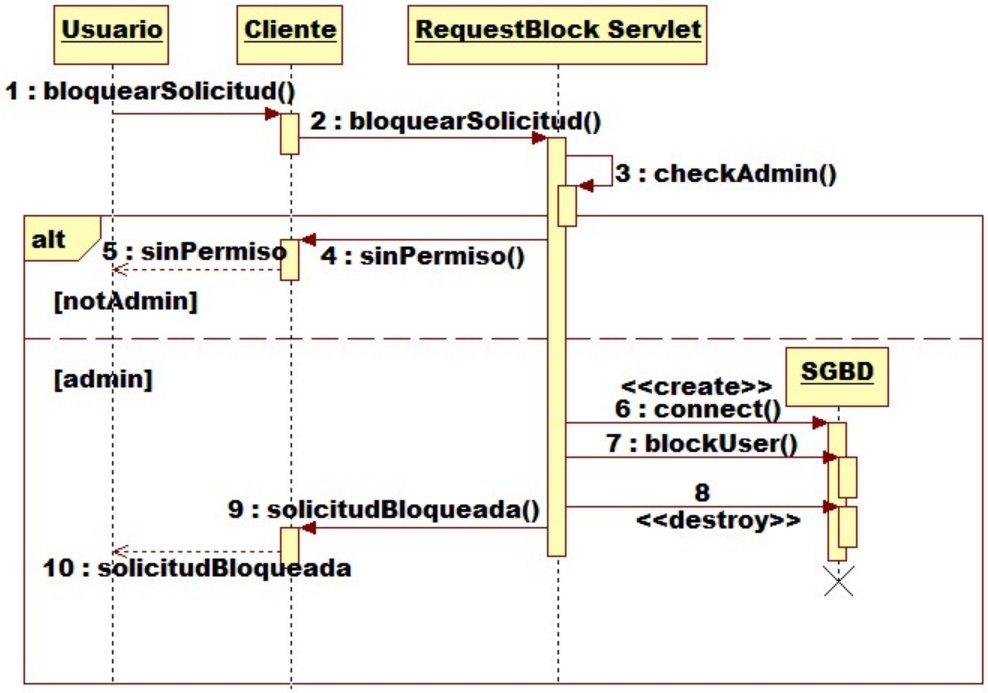

Figura 2.13: Diagrama de secuencia del CU-06 Bloquear registro

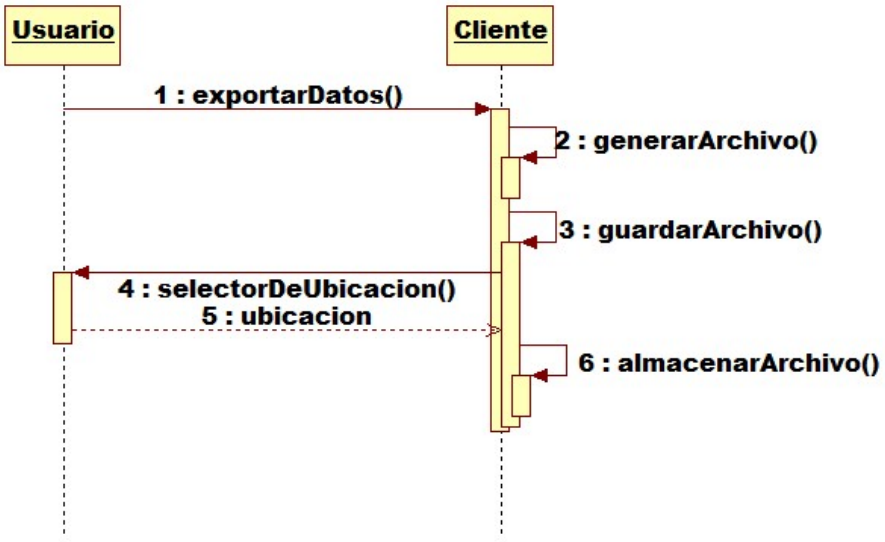

Figura 2.14: Diagrama de secuencia del CU-07 Exportar datos locales

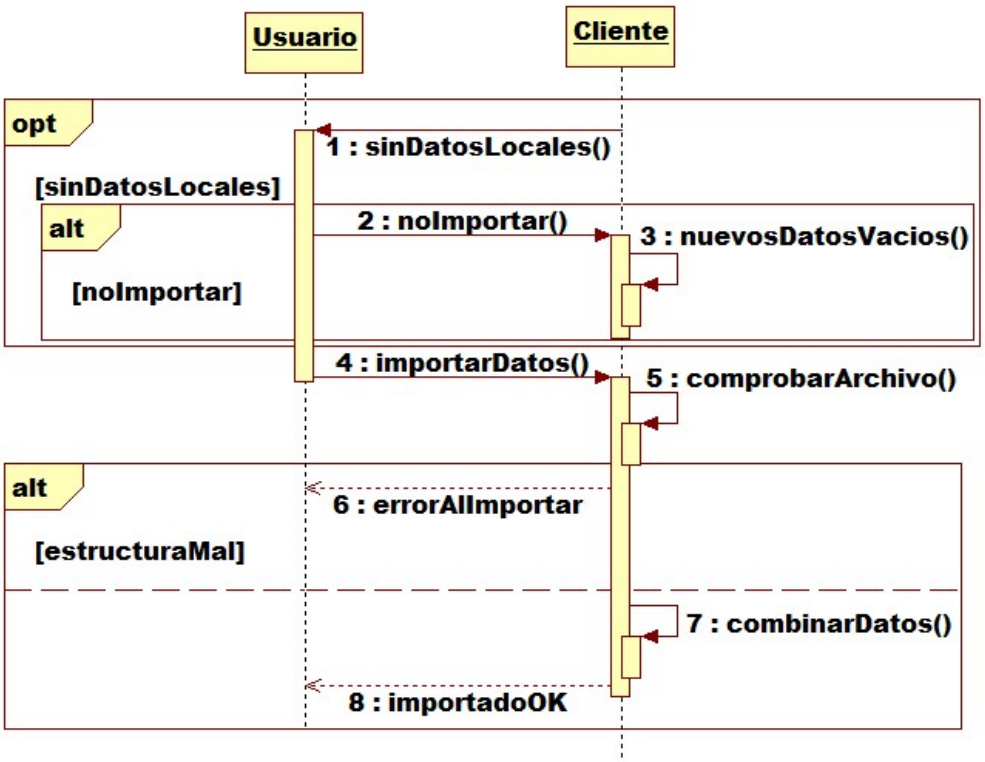

Figura 2.15: Diagrama de secuencia del CU-08 Importar datos locales

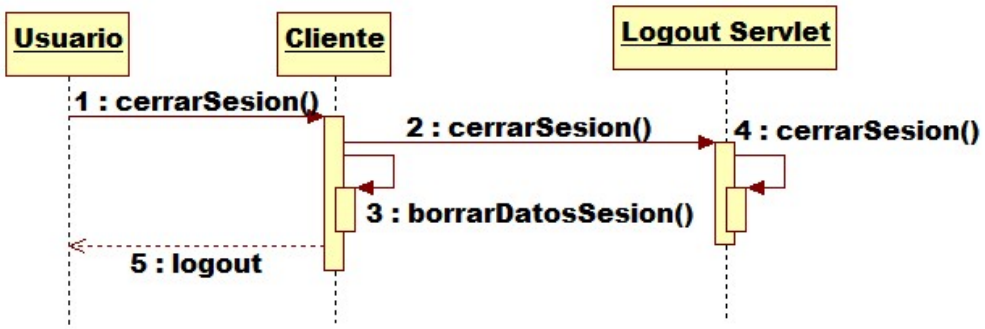

Figura 2.16: Diagrama de secuencia del CU-09 Cerrar sesión

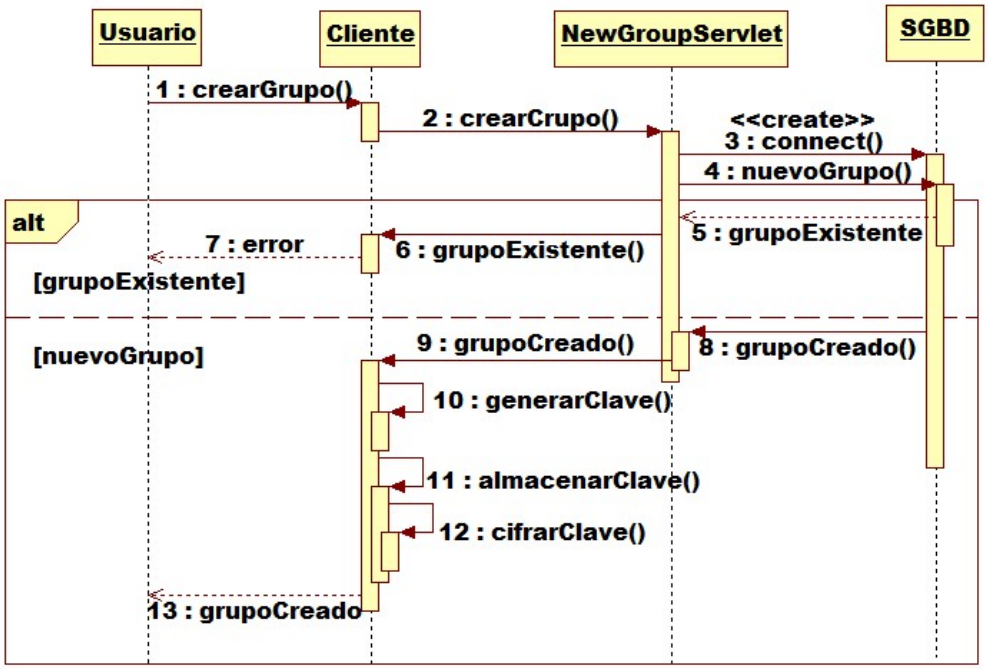

Figura 2.17: Diagrama de secuencia del CU-10 Crear grupo

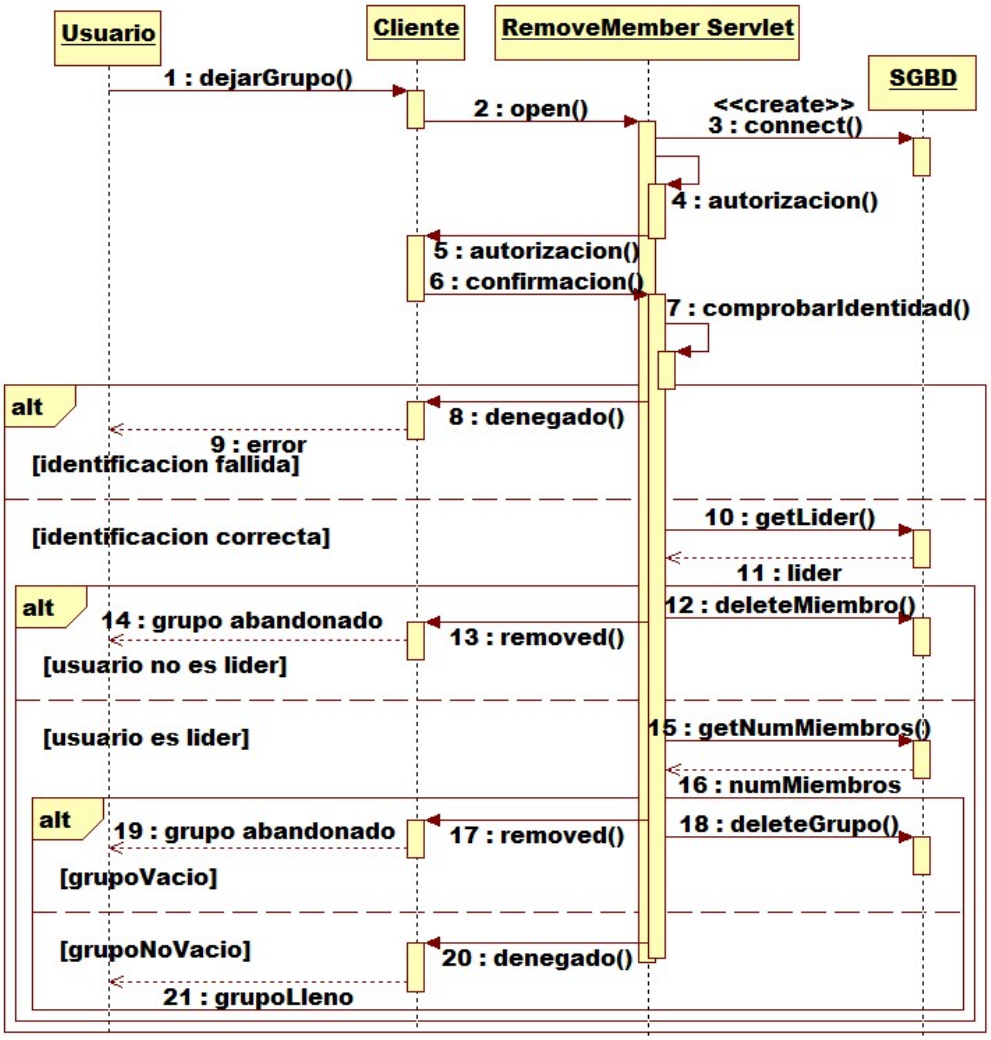

Figura 2.18: Diagrama de secuencia del CU-11 Dejar grupo

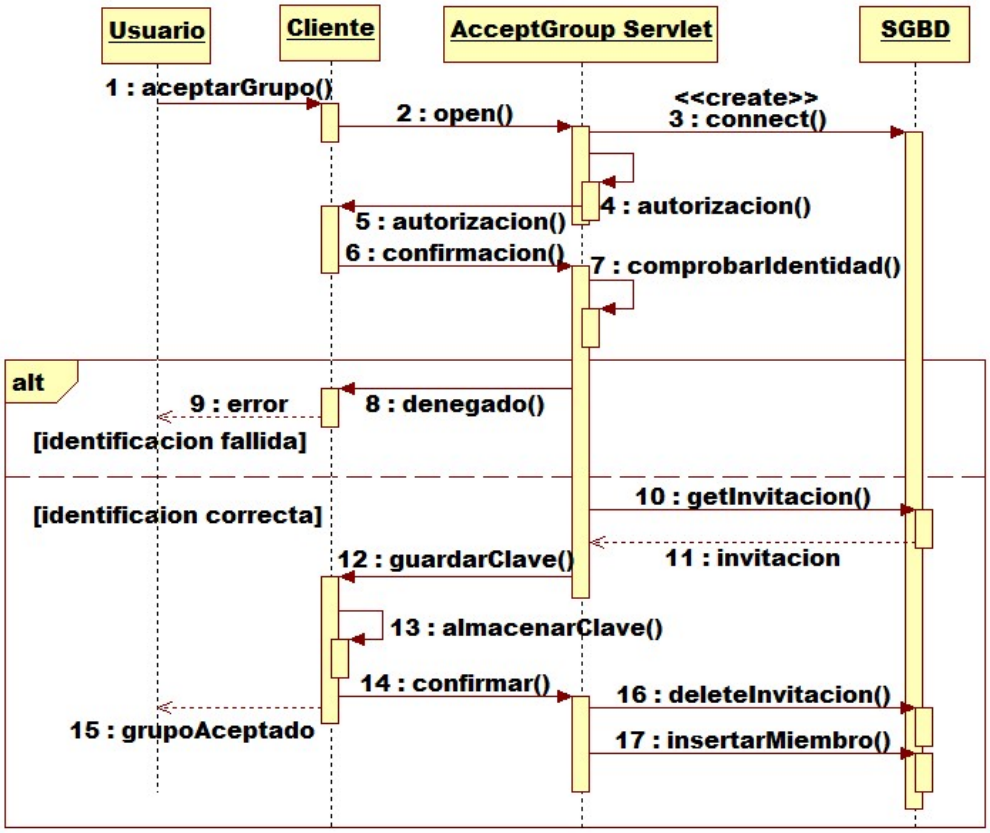

Figura 2.19: Diagrama de secuencia del CU-12 Aceptar invitación a grupo

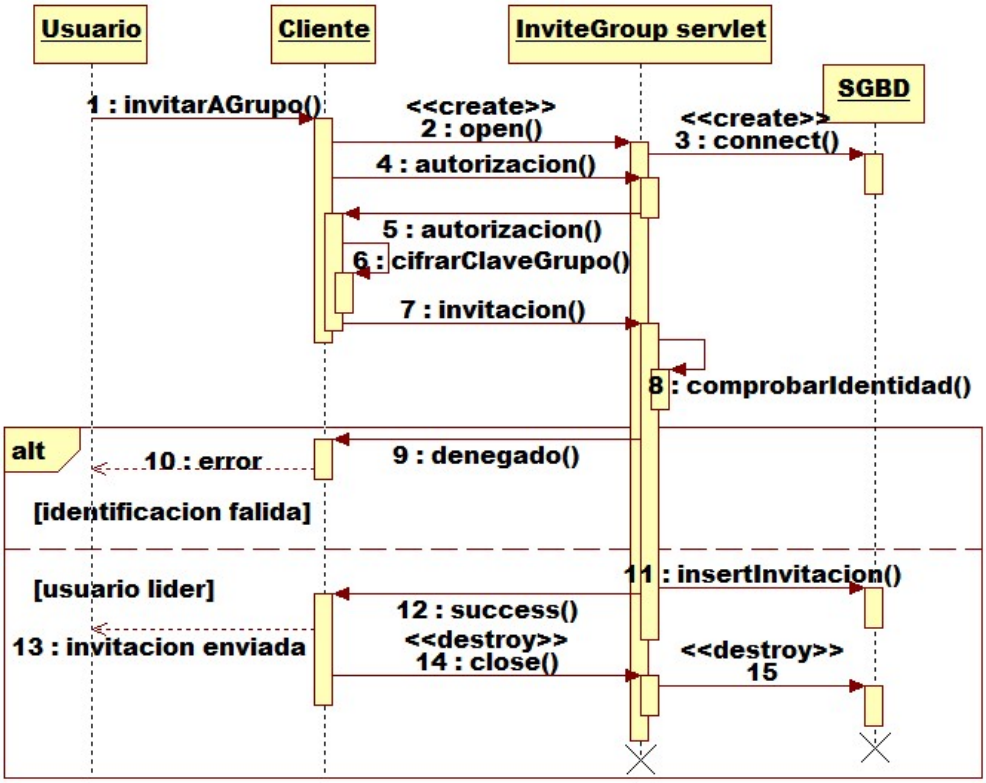

Figura 2.20: Diagrama de secuencia del CU-13 Invitar a grupo

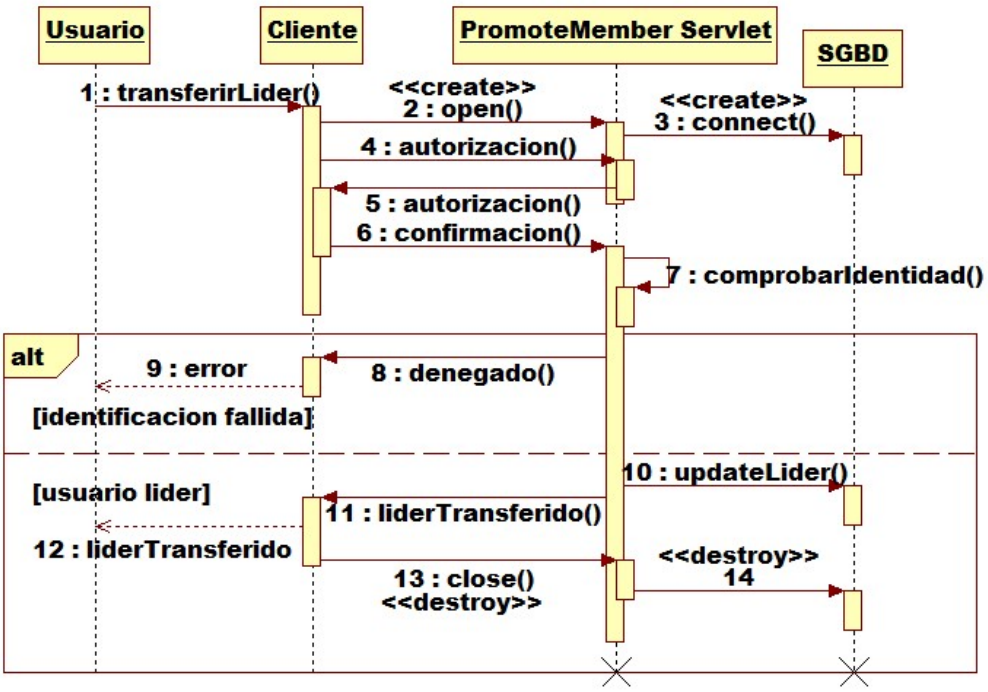

Figura 2.21: Diagrama de secuencia del CU-14 Transferir liderazgo

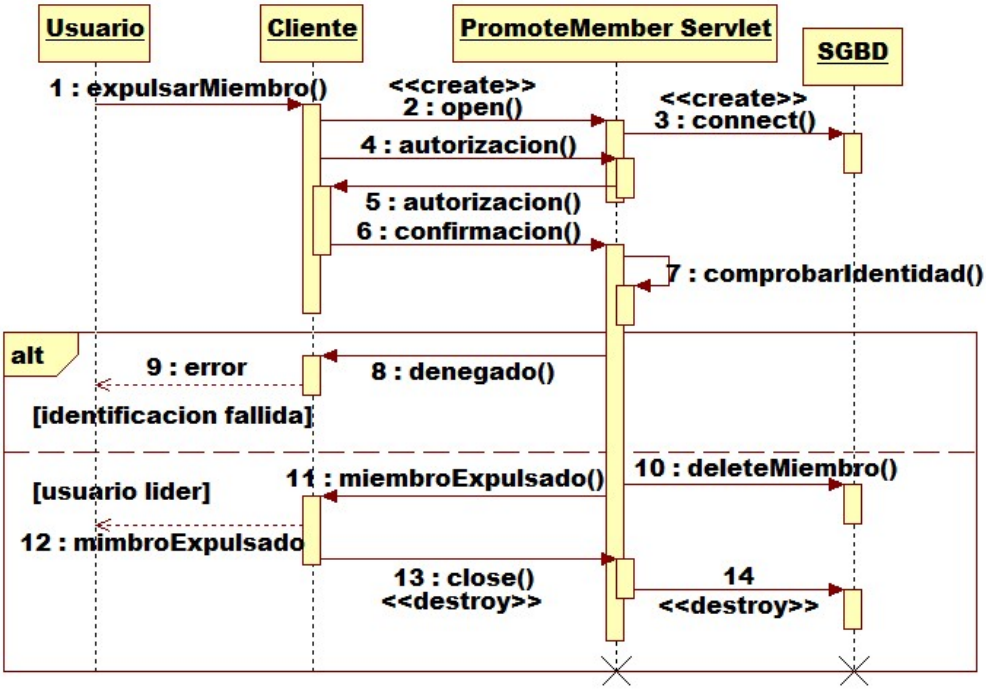

Figura 2.22: Diagrama de secuencia del CU-15 Expulsar de grupo

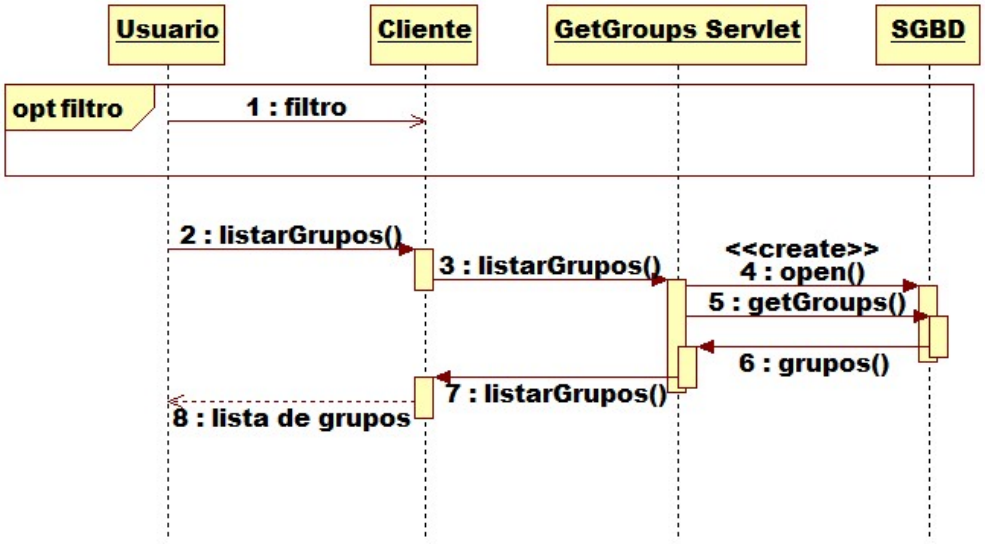

Figura 2.23: Diagrama de secuencia del CU-16 Listar grupos y CU-17 Filtrar grupos

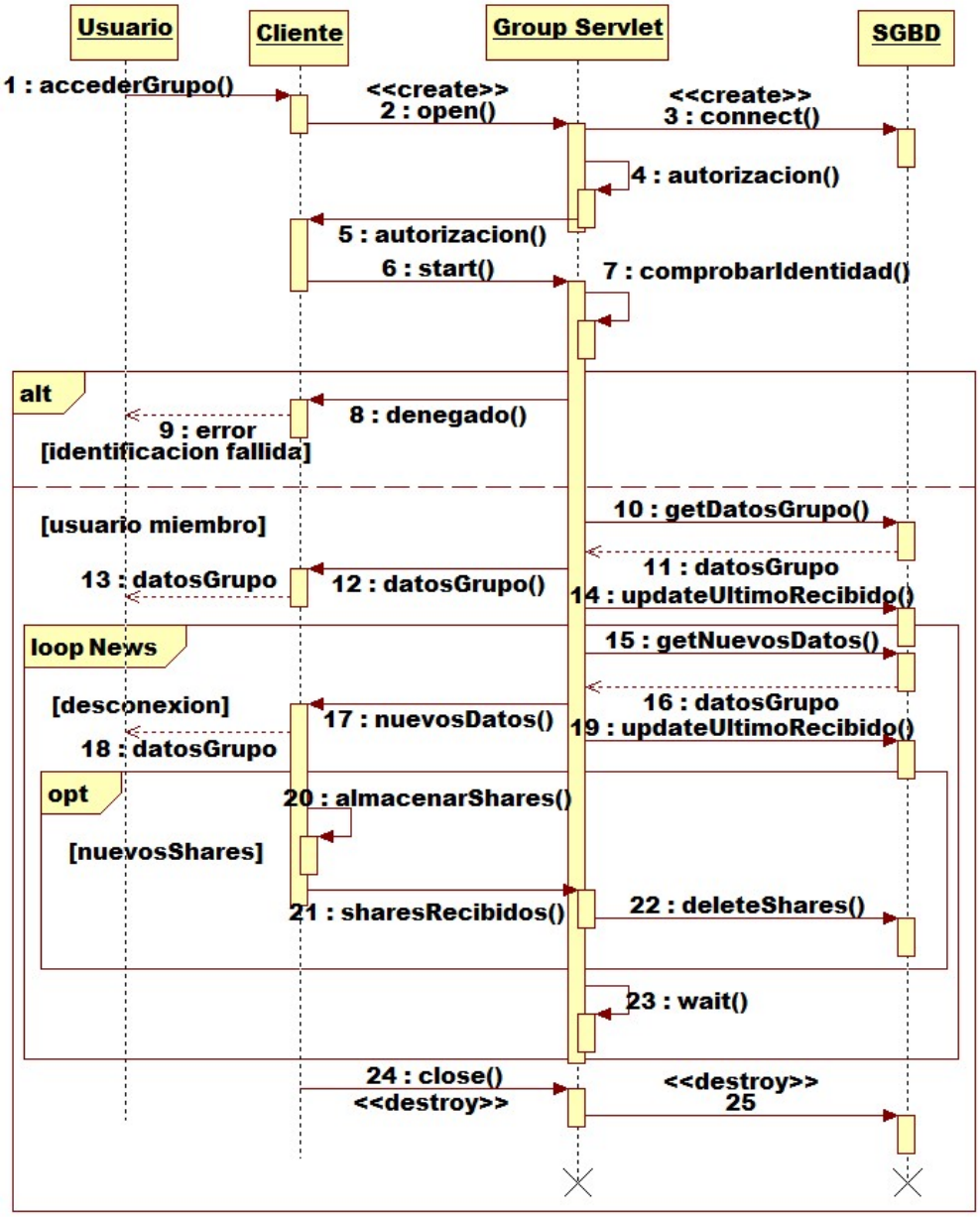

Figura 2.24: Diagrama de secuencia del CU-18 Acceder a grupo

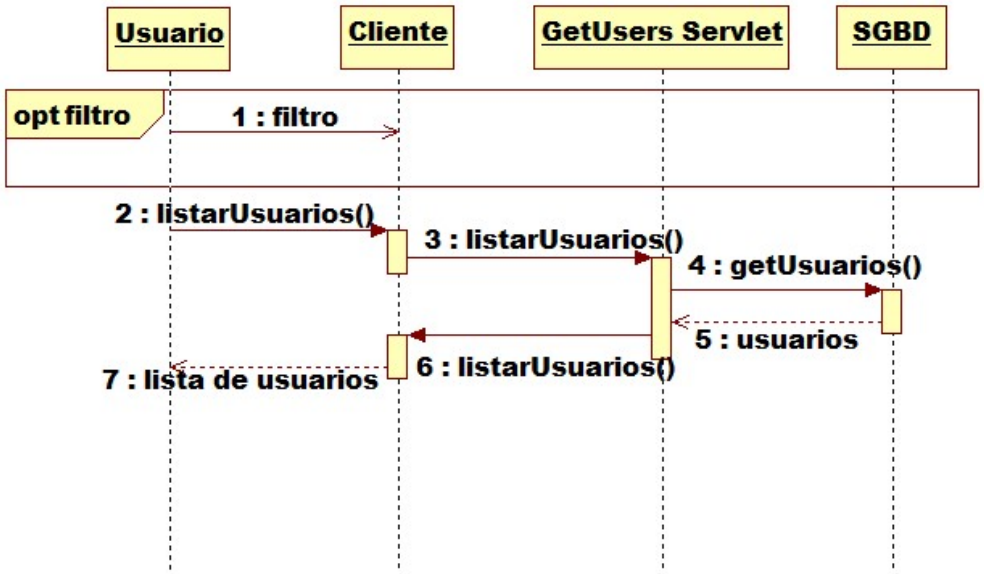

Figura 2.25: Diagrama de secuencia del CU-19 Listar usuarios y CU-20 Filtrar usuarios

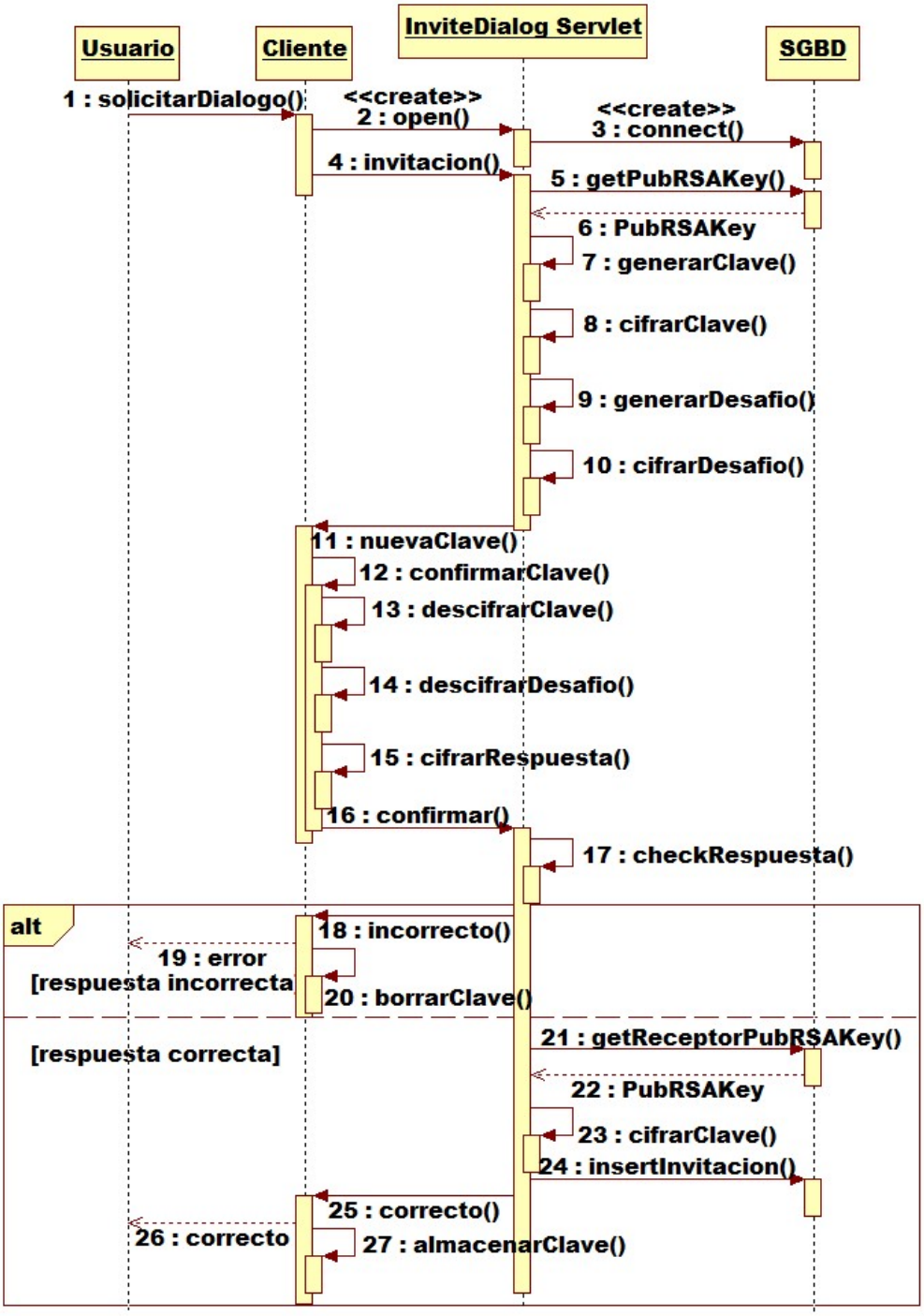

Figura 2.26: Diagrama de secuencia del CU-21 Solicitar diálogo

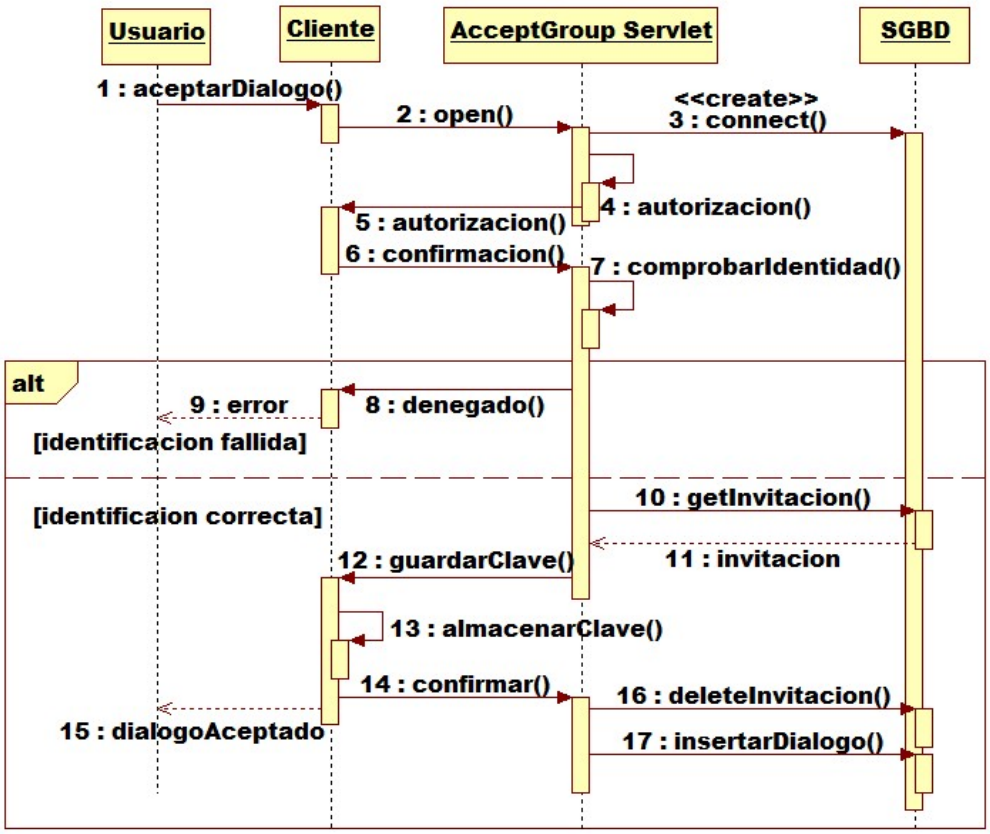

Figura 2.27: Diagrama de secuencia del CU-22 Aceptar diálogo

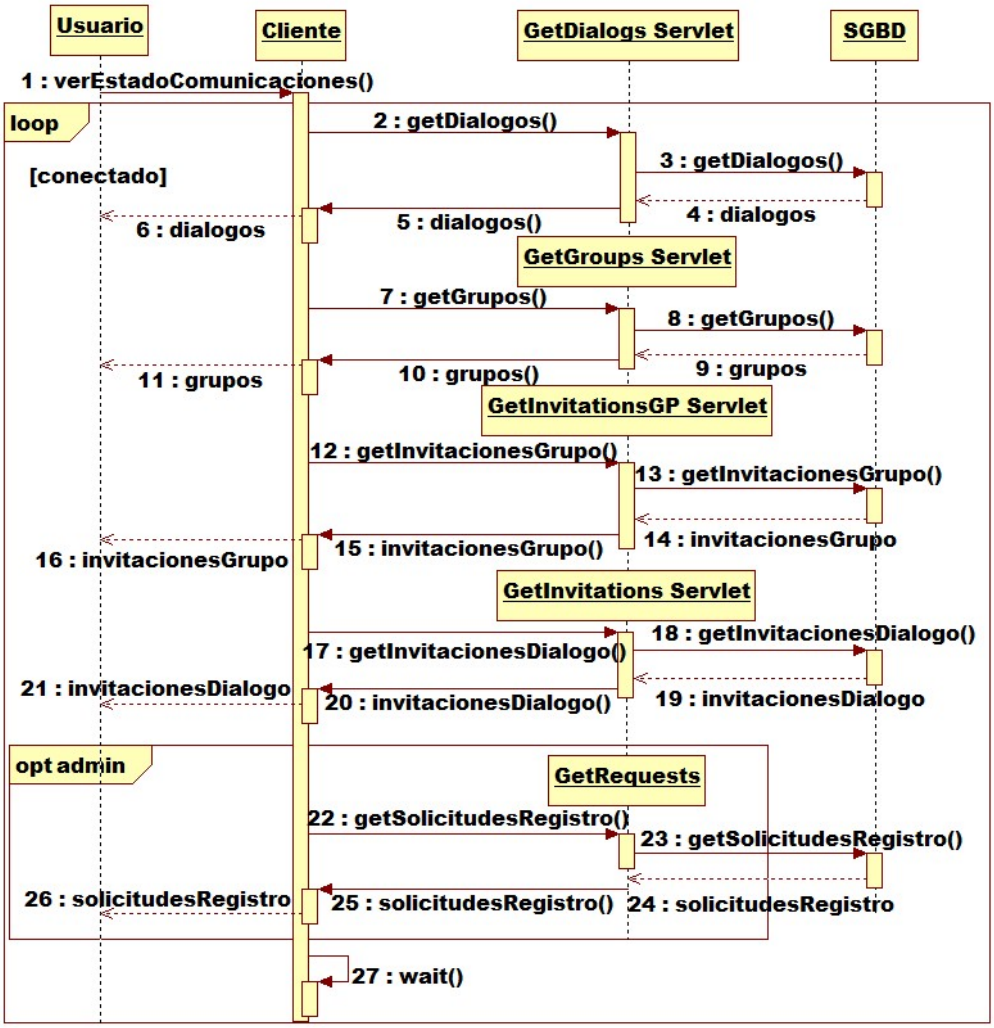

Figura 2.28: Diagrama de secuencia del CU-23 Mostrar estado de comunicaciones

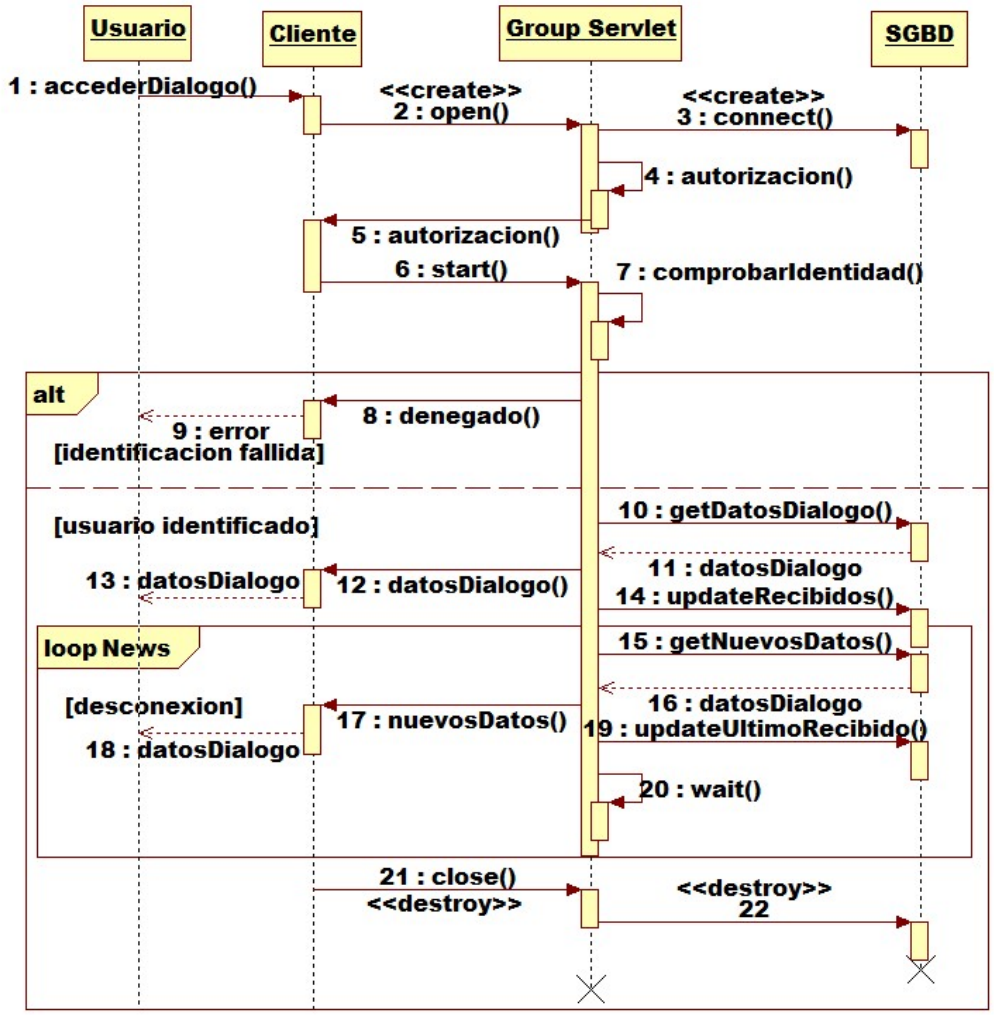

Figura 2.29: Diagrama de secuencia del CU-24 Acceder a diálogo

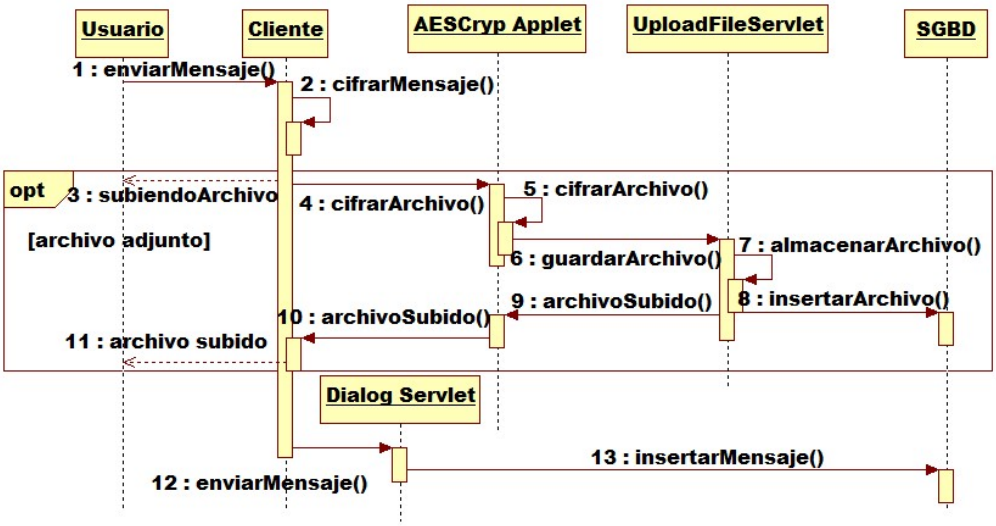

Figura 2.30: Diagrama de secuencia del CU-25 Mandar mensaje de diálogo

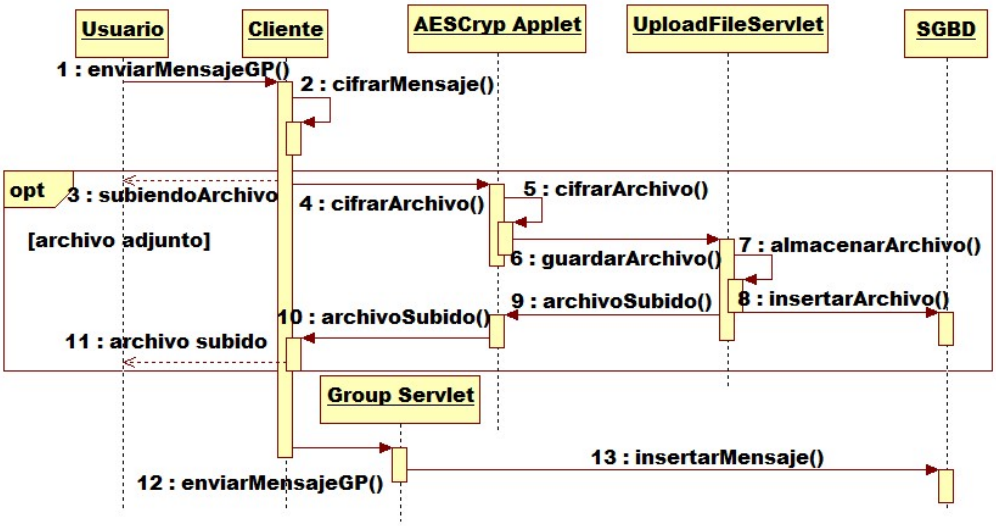

Figura 2.31: Diagrama de secuencia del CU-26 Mandar mensaje de grupo

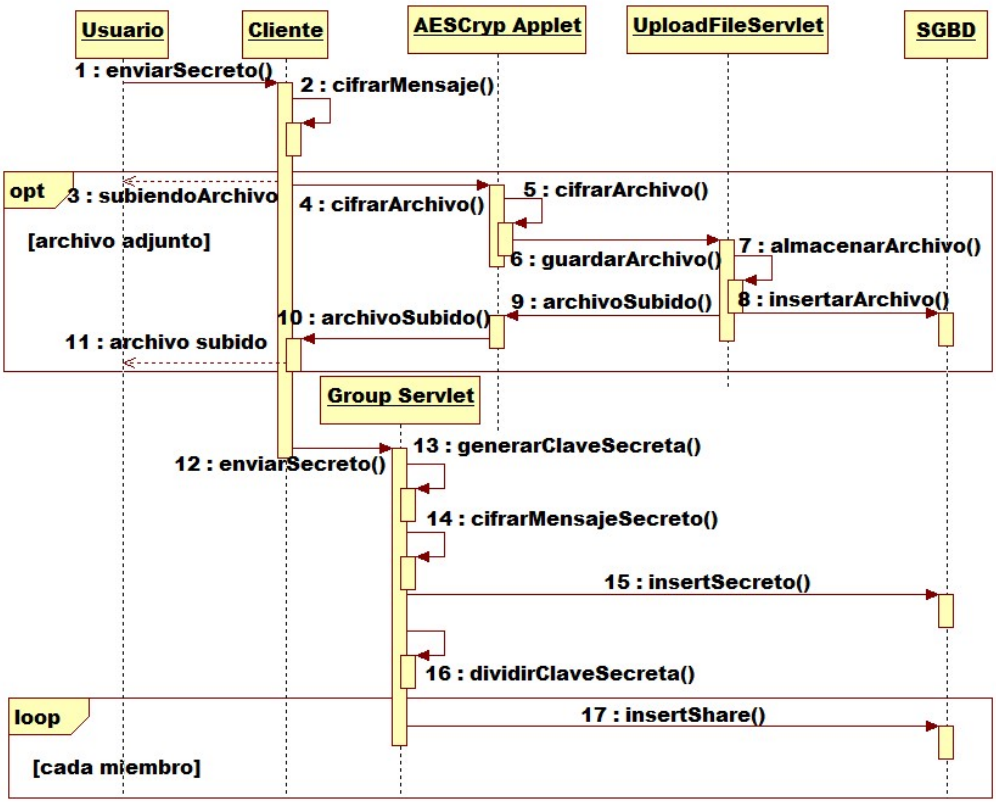

Figura 2.32: Diagrama de secuencia del CU-27 Mandar secreto

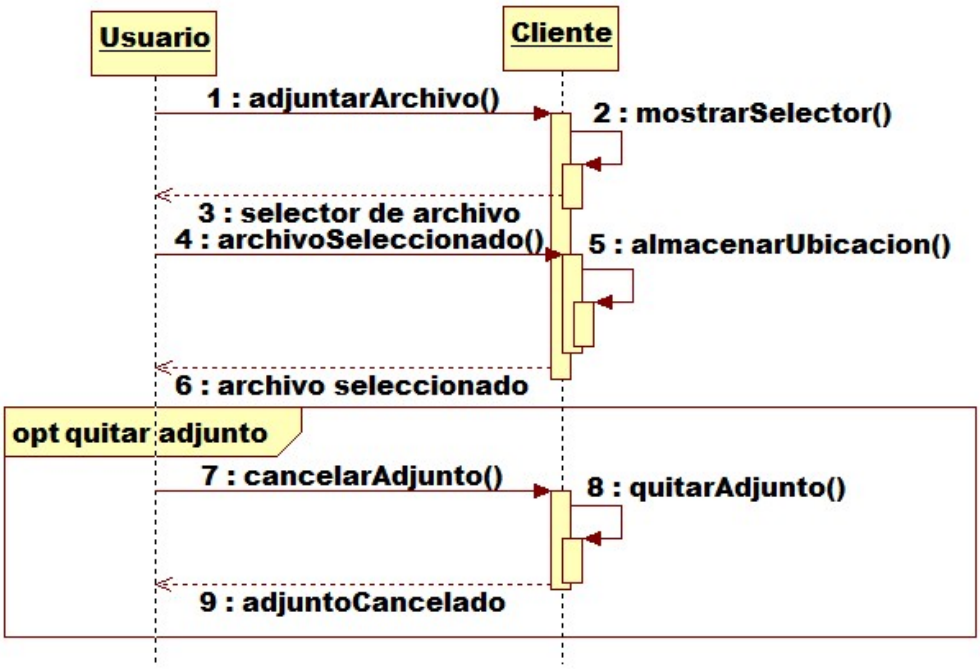

Figura 2.33: Diagrama de secuencia del CU-28 Adjuntar archivo

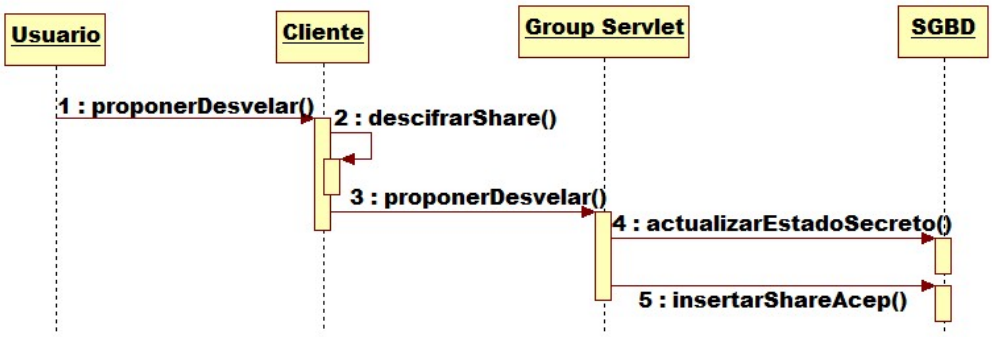

Figura 2.34: Diagrama de secuencia del CU-29 Proponer desvelar secreto

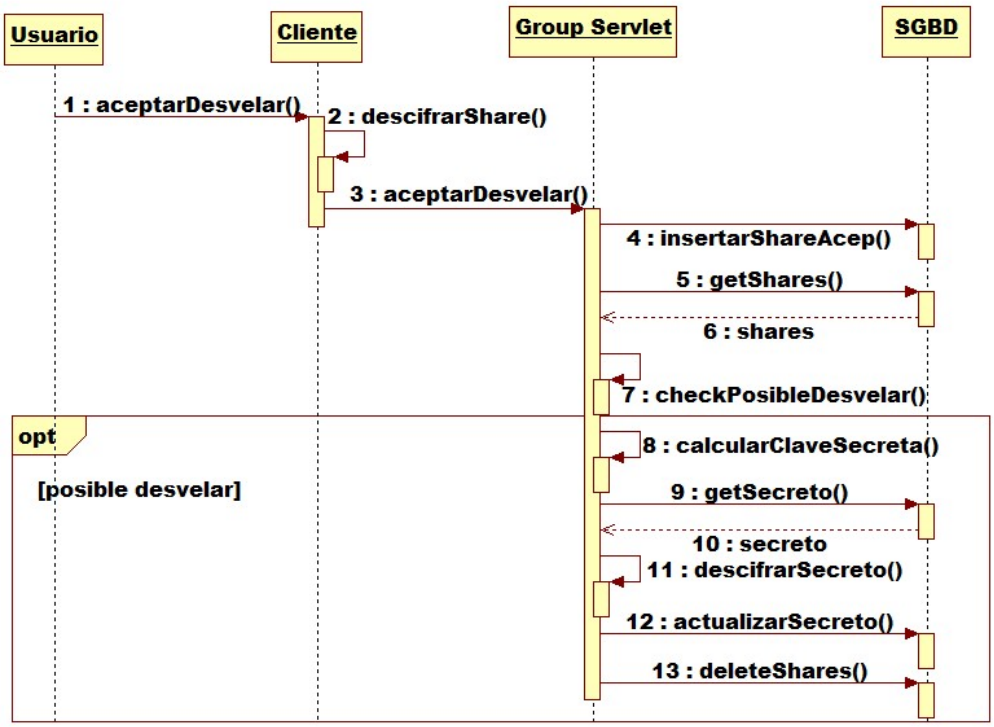

Figura 2.35: Diagrama de secuencia del CU-30 Aceptar desvelar secreto

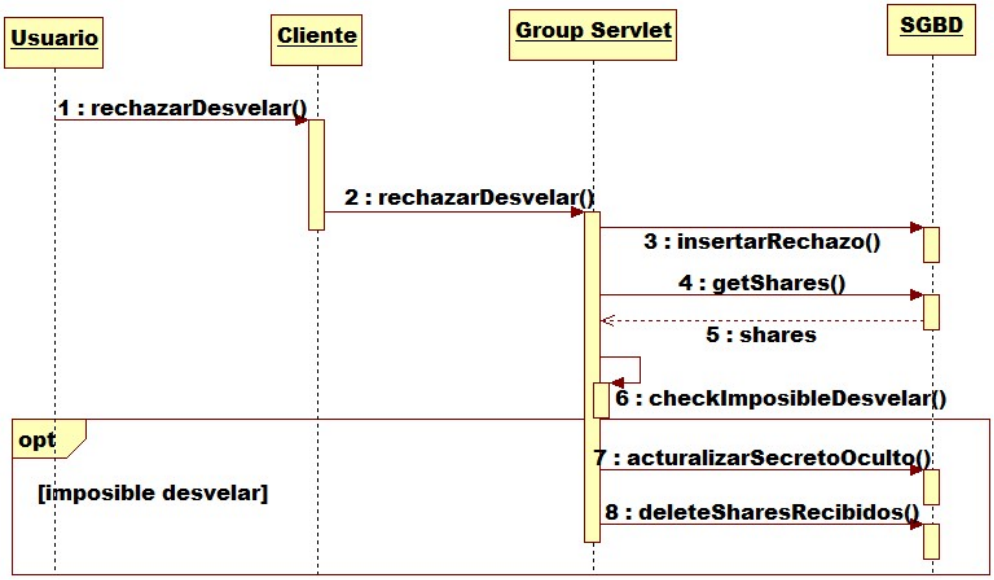

Figura 2.36: Diagrama de secuencia del CU-31 Rechazar desvelar secreto

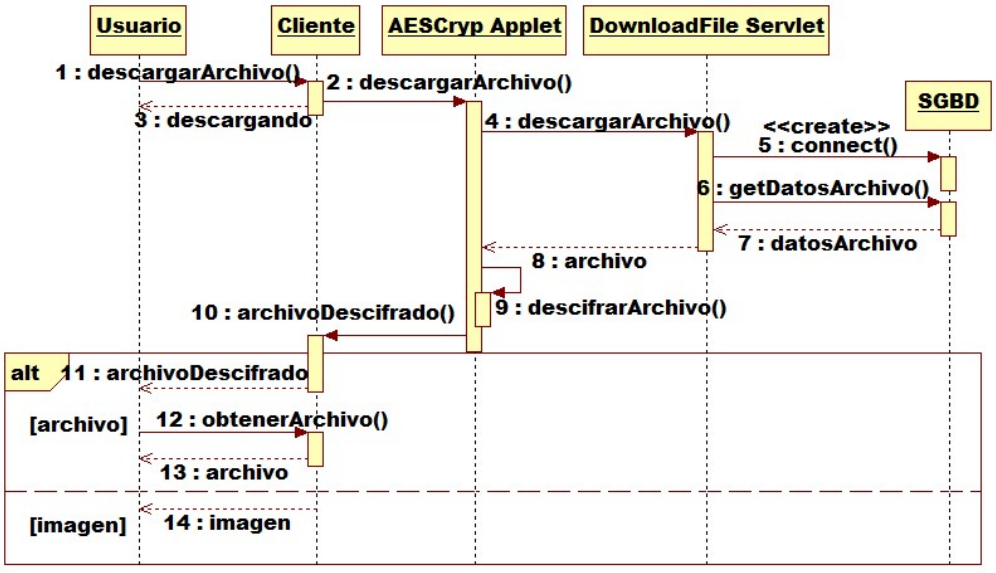

Figura 2.37: Diagrama de secuencia del CU-32 Descargar archivo

Detalle de procesos

El siguiente diagrama muestra más detalladamente el proceso de autorización empleado en algunos de los casos de uso, por el cual el sistema genera un canal seguro de comunicación mediante una clave de usar y tirar siguiendo el protocolo Diffie-Hellman, y verifica la identidad del usuario mediante un desafío-respuesta empleando su clave RSA.

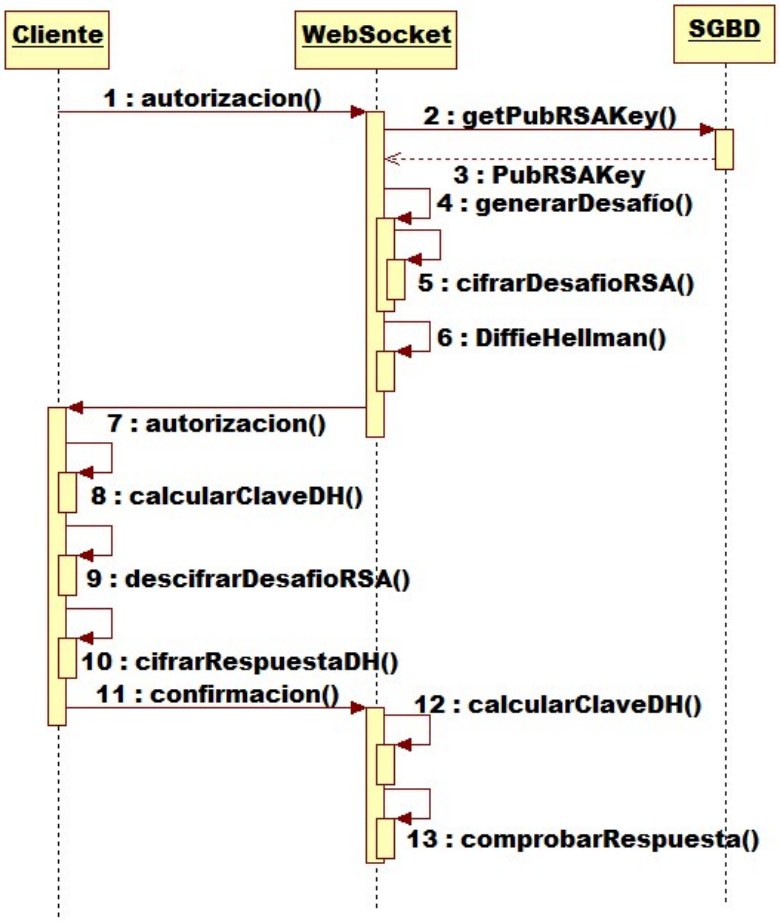

Figura 2.38: Diagrama de secuencia del proceso de autorización

2.2.3. Clases

A pesar de que la mayor parte de los elementos de este proyecto siguen un esquema orientado a eventos, ciertos componentes del sistema son empleados siguiendo un paradigma orientado a objetos con la estructura que se muestra a continuación.

Clases del servidor

El siguiente diagrama muestra las clases del servidor sin entrar en mucho detalle. La mayoría de clases del servidor consisten en servlets o WebSockets que responden a los mensajes que envía el cliente. Las clases de criptografía son usadas por estas, y se encuentran separadas en un paquete a parte. Además existe un applet que será el encargado de cifrar los archivos que el usuario adjunte a los mensajes.

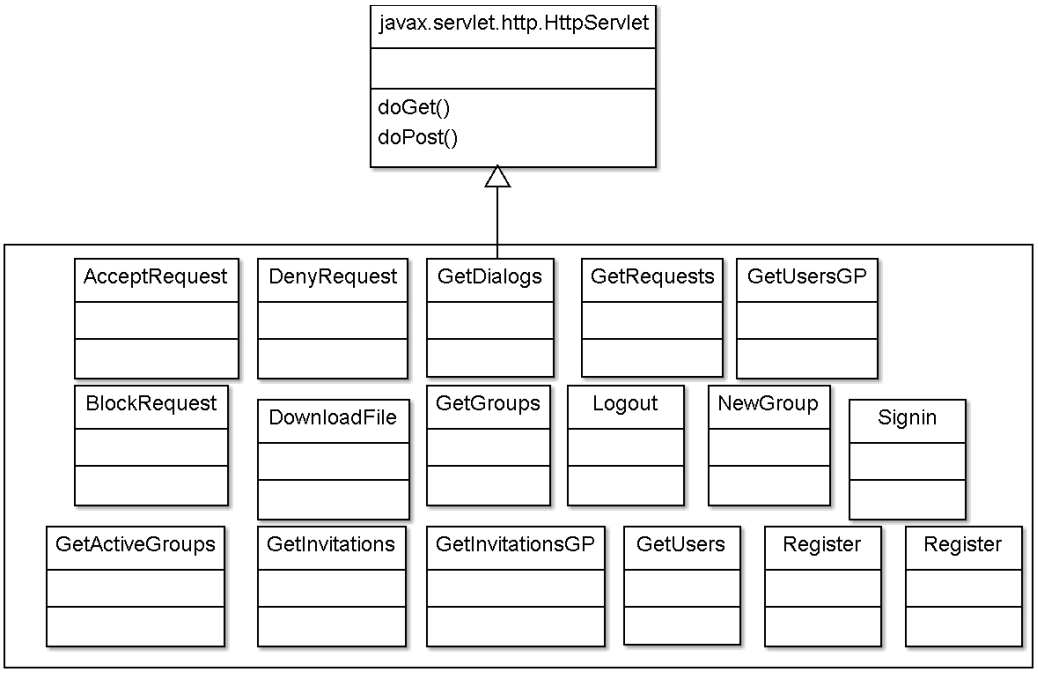

Figura 2.39: Clases Servlet del servidor

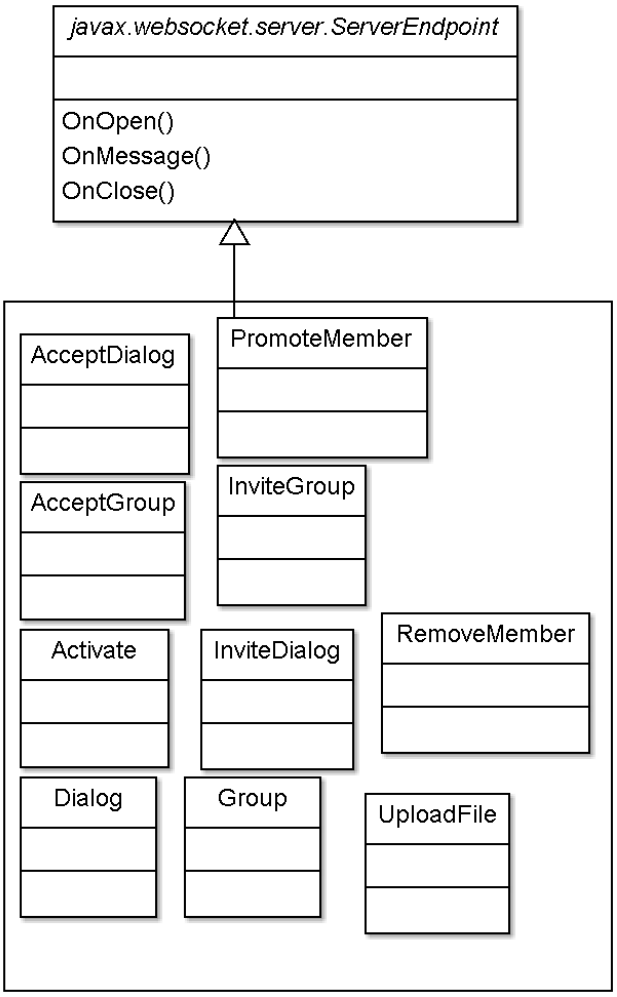

Figura 2.40: Clases WebSocket del servidor

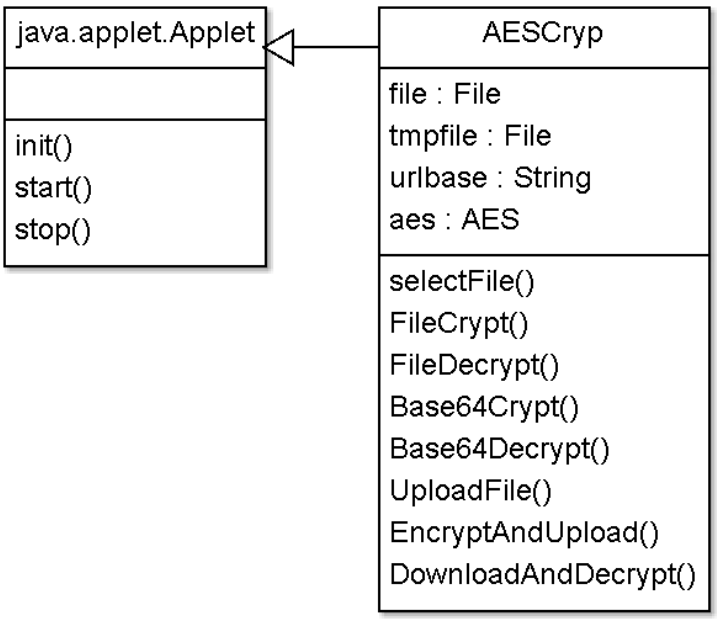

Figura 2.41: Applet del servidor

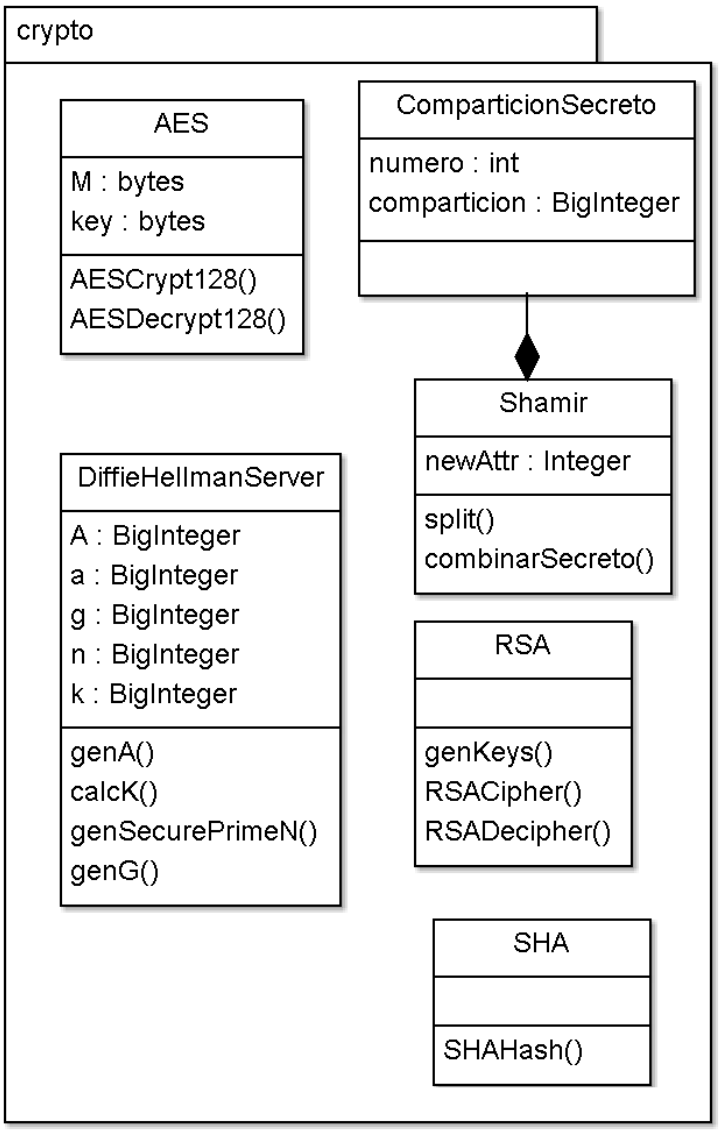

Figura 2.42: Clases de criptografía del servidor

Clases del cliente

El servidor está mucho más orientado a objetos ya que la tecnología que emplea es Java, mientras que el cliente emplea JavaScript, que está mucho mas orientado a eventos.

Por tanto, el cliente poseerá un archivo de JavaScript por cada pantalla de la interfaz con todo el código necesario para llevar a cabo los procesos detallados anteriormente, a excepción de librerías necesarias como jQuery y los objetos criptográficos, que serán una replica en JavaScript de las clases correspondientes Java del servidor.

2.2.4. Mapa del sitio

Adicionalmente he decidido construir un mapa del sitio como apoyo durante el desarrollo. El mapa del sitio permite establecer las pantallas que se van a presentar al usuario y definir las acciones que puede llevar a cabo en cada una. Gracias a ello podemos asegurarnos de que no quede ningún caso de uso descolgado.

El mapa del sitio está compuesto por tres tipos de entidades:

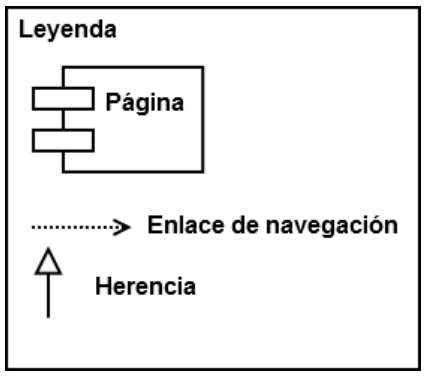

Figura 2.43: Leyenda del mapa del sitio

1. **Páginas**: representan las diferentes páginas o pantallas a las que puede acceder el usuario mientras navega el sitio web. Están representadas con un rectángulo. En su interior se detalla el nombre de la página y la lista de los casos de uso accesibles desde ella.

2. **Enlaces de navegación**: representan la navegación que el usuario realiza de unas páginas a otras, indicando su sentido. Están representados por una flecha de linea discontinua. La dirección de la flecha indica la dirección de la navegación.

3. **Herencia:** representa la relación entre dos paginas, donde una de ellas (hija) hereda los casos de uso y enlaces de navegación de la otra (padre). Está representada por una línea continua con un triángulo en un extremo que conecta con una o más páginas. El extremo del triángulo indica el padre de la herencia, mientras que los otros extremos conectan con las hijas.

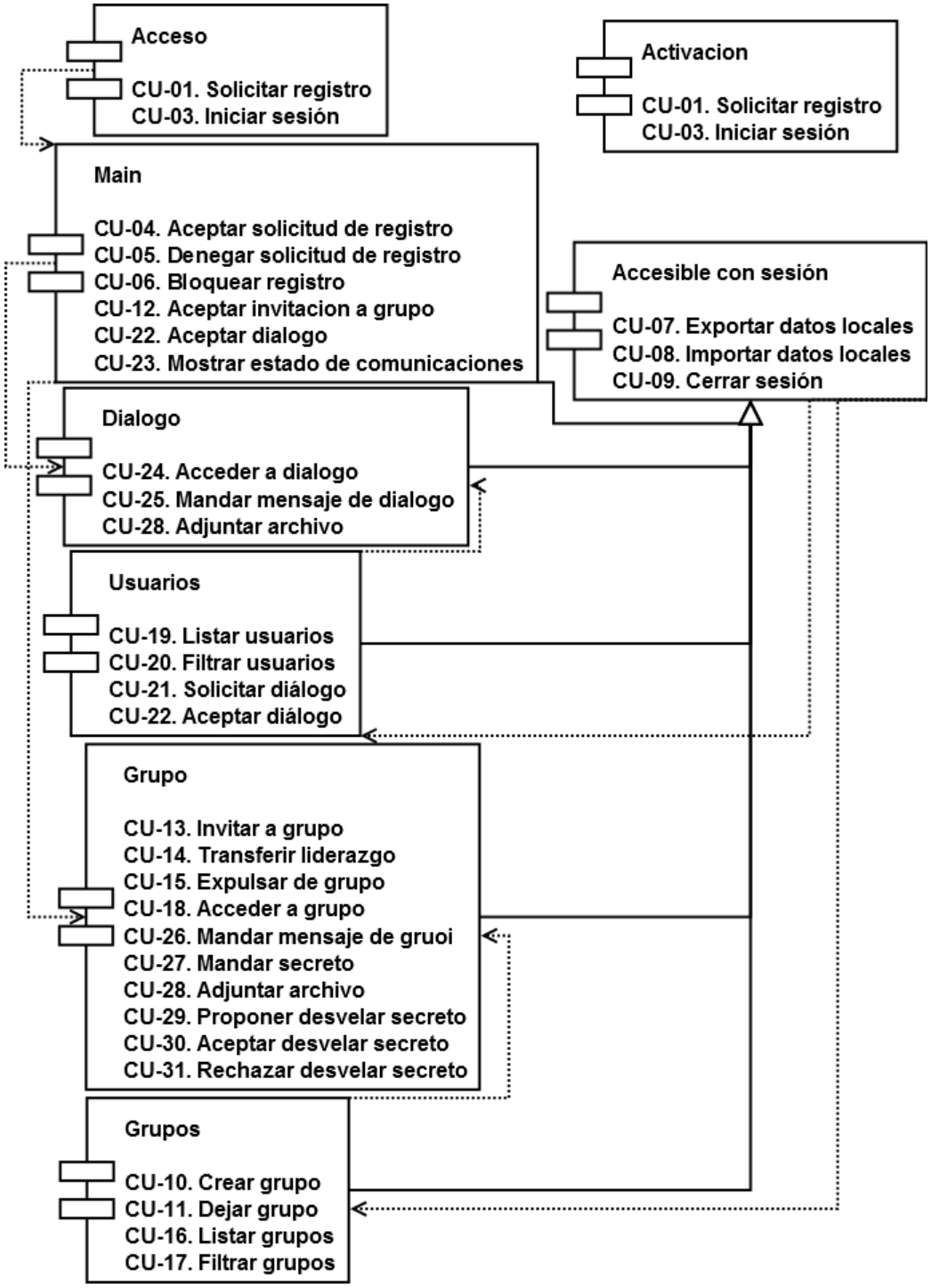

Figura 2.44: Mapa del sitio

2.3. Pruebas

Las pruebas realizadas sobre este sistema pretenden comprobar que la implementación del mismo se ajusta a las características definidas en el análisis, comprobando que hace lo que debe hacer.

Agruparemos las pruebas por caso de uso:

CU-01. Solicitar registro:

a) El sistema comprueba los datos de registro correctamente: *Si*

b) El sistema almacena los datos del nuevo usuario correctamente: *Si*

c) El sistema inserta el primer usuario como administrador y este pasa a la fase de activación automáticamente: *Si*

CU-02. Activar cuenta:

a) El sistema genera la clave RSA de usuario correctamente: *Si*

b) La clave se envía correctamente al cliente: *No - En ocasiones el sistema no envía la clave al cliente ->SOLUCIONADO*

c) El sistema cifra la clave RSA utilizando la contraseña de usuario correctamente: *Si*

d) La clave se comprueba correctamente: *No - En ocasiones el servidor falla al descifrar la respuesta del cliente a su desafío ->SOLUCIONADO*

CU-03. Iniciar sesión:

a) El sistema comprueba correctamente que todos los datos han sido introducidos: *Si*

b) El sistema comprueba correctamente a existencia del usuario: *Si*

c) El sistema comprueba correctamente la identidad del usuario mediante su clave RSA: *Si*

d) El sistema inicia la sesión correctamente si la identidad del usuario se confirma: *Si*

CU-04. Aceptar solicitud de registro:

a) El sistema restringe el acceso a usuarios no administradores: *Si*

b) El sistema modifica el estado del usuario y genera un código de activación correctamente: *Si*

c) El sistema genera y envía un correo de activación a la dirección del usuario seleccionado: *Si, pero el correo presenta el contenido doble ->SOLUCIONADO*

CU-05. Denegar solicitud de registro

a) El sistema restringe el acceso a usuarios no administradores: *Si*

b) El sistema elimina los datos del usuario correctamente: *Si*

CU-06. Denegar solicitud de registro

a) El sistema restringe el acceso a usuarios no administradores: *Si*

b) El sistema modifica el estado del usuario correctamente: *Si*

CU-07. Exportar datos locales

a) El sistema genera y almacena los datos locales correctamente: *Si*

CU-08. Importar datos locales

a) El sistema comprueba el formato del archivo de datos correctamente: *Si*

b) El sistema mezcla los datos importados con los datos locales correctamente: *Si*

CU-09. Crear grupo

a) El sistema comprueba que no exista un grupo con el mismo nombre que el que se desea crear: *Si*

b) El sistema genera la clave de grupo correctamente: *Si*

c) El sistema crea el nuevo grupo correctamente: *Si*

CU-10. Dejar grupo

a) El sistema impide que el usuario líder no salga si hay otros usuarios: *Si*

b) El sistema elimina el grupo si el líder lo abandona y no hay otros usuarios: *Si*

c) El sistema elimina el miembro del grupo correctamente: *Si*

CU-11. Aceptar invitación a grupo

a) El sistema almacena la clave de grupo localmente de forma correcta: *Si*

b) El sistema añade la nueva información de miembro del grupo: *Si*

CU-12. Invitar a grupo

a) El sistema almacena correctamente la invitación al grupo: *Si*

b) El sistema restringe el acceso a la funcionalidad a los miembros no lideres de grupo: *Si*

CU-13. Transferir liderazgo

a) El sistema actualiza el líder del grupo correctamente: *Si*

b) El sistema restringe el acceso a la funcionalidad a los miembros no lideres de grupo: *Si*

CU-14. Expulsar de grupo

a) El sistema elimina correctamente la información de miembro del grupo: *Si*

b) El sistema restringe el acceso a la funcionalidad a los miembros no lideres de grupo: *Si*

CU-15. Listar grupos

a) El sistema elimina correctamente la lista de los grupos a los que pertenece el usuario: *Si*

CU-16. Filtrar grupos

a) El sistema filtra correctamente la lista de los grupos del usuario siguiendo el criterio que este introduzca: *Si*

CU-17. Acceder a grupo

a) El sistema verifica la identidad del usuario correctamente: *Si*

b) El sistema negocia la clave de usar y tirar correctamente: *Parcial - en ocasiones la negociación falla ->SOLUCIONADO*

c) El sistema deniega el acceso al grupo a los no miembros: *Si*

d) El sistema muestra correctamente los mensajes de grupo: *Si*

e) El sistema muestra correctamente los secretos de grupo y su estado: *Si*

f) El sistema lista los miembros del grupo correctamente: *Si*

g) El sistema lista los usuarios no miembros correctamente: *Si*

h) El sistema muestra correctamente los archivos adjuntos a mensajes: *Si*

i) El sistema descarga los archivos adjuntos a los mensajes correctamente: *Parcial - al descargar archivos desde fuera del localhost, el archivo se descarga y descifra parcialmente. ->SOLUCIONADO*

j) El sistema descarga y almacena as partes compartidas de secretos correctamente: *Si*

k) El sistema elimina las partes compartidas de secretos de la BD una vez recibidas: *Si*

l) El sistema actualiza los datos del grupo periódicamente de forma correcta: *Si*

CU-18. Listar usuarios

a) El sistema elimina correctamente la lista de los usuarios del sistema: *Si*

CU-19. Filtrar grupos

a) El sistema filtra correctamente la lista de los usuarios del sistema siguiendo el criterio que este introduzca: *Si*

CU-20. Solicitar diálogo

a) El sistema genera correctamente una clave de diálogo: *Si*

b) El sistema comprueba que la clave sea recibida en el cliente de forma correcta: *Si*

c) El sistema almacena la solicitud de diálogo en la BD de forma correcta: *Si*

d) El sistema almacena la clave de diálogo localmente de forma segura correctamente: *Si*

CU-21. Aceptar dialogo

a) El sistema verifica la identidad del receptor correctamente: *Si*

b) El sistema almacena la clave de diálogo localmente de forma correcta: *Si*

c) El sistema crea la información del nuevo diálogo correctamente: *Si*

d) El sistema elimina la invitación a dialogo correctamente: *Si*

CU-22. Mostrar estado de comunicaciones

a) El sistema muestra correctamente la información de diálogos activos del usuario: *Si*

b) El sistema muestra correctamente la información de grupos del usuario: *Si*

c) El sistema muestra correctamente la información de solicitudes de diálogo pendientes del usuario: *Si*

d) El sistema muestra correctamente la información de invitaciones a grupo del usuario: *Si*

e) El sistema muestra correctamente la lista de las solicitudes de registro: *Si*

CU-23. Acceder a dialogo

a) El sistema deniega el acceso al dialogo a usuarios que no pertenezcan a él: *Si*

b) El sistema muestra correctamente los mensajes de dialogo: *Si*

c) El sistema muestra correctamente los archivos adjuntos a mensajes: *Si*

d) El sistema descarga los archivos adjuntos a los mensajes correctamente: *Parcial - al descargar archivos desde fuera del localhost, el archivo se descarga y descifra parcialmente solo.*

e) El sistema actualiza los datos del dialogo periódicamente de forma correcta: *Si*

CU-24. Mandar mensaje de diálogo

a) El sistema cifra correctamente los mensajes de diálogo: *Si*

b) El sistema almacena correctamente los mensajes de diálogo: *Si*

c) El sistema cifra correctamente los archivos adjuntos: *Si*

d) El sistema sube y almacena correctamente los archivos adjunto: *Parcial - El applet no sube los archivos desde fuera del localhost - SOLUCIONADO*

e) El sistema informa al usuario del estado de sus mensajes enviados correctamente: *Si*

CU-25. Mandar mensaje de grupo

a) El sistema cifra correctamente los mensajes de grupo: *Si*

b) El sistema almacena correctamente los mensajes de grupo: *Si*

c) El sistema cifra correctamente los archivos adjuntos: *Si*

d) El sistema sube y almacena correctamente los archivos adjunto: *Parcial - El applet no sube los archivos desde fuera del localhost - SOLUCIONADO*

e) El sistema informa al usuario del estado de sus mensajes enviados correctamente: *Si*

CU-26. Mandar secreto

a) El sistema cifra correctamente los secretos: *Si*

b) El sistema almacena correctamente los secretos: *Si*

c) El sistema cifra correctamente los archivos adjuntos: *Si*

d) El sistema sube y almacena correctamente los archivos adjunto: *Parcial - El applet no sube los archivos desde fuera del localhost - SOLUCIONADO*

e) El sistema genera y divide correctamente la clave secreta: *Si*

CU-27. Adjuntar archivo

a) El sistema muestra el selector de archivo correctamente: *Parcial - Se requiere Java*

b) El sistema muestra correctamente el archivo seleccionado: *Si*

CU-28. Proponer desvelar secreto

a) El sistema almacena correctamente la parte compartida al proponer desvelar un secreto: *Si*

b) El sistema actualiza el estado del secreto a *Pendiente* al proponer desvelarlo: *Si*

CU-29. Aceptar desvelar secreto

a) El sistema almacena correctamente la parte compartida al aceptar desvelar un secreto: *Si*

b) El sistema desvela el secreto correctamente cuando es posible: *Si*

CU-30. Rechazar desvelar secreto

a) El sistema almacena correctamente el rechazo a desvelar el secreto: *Si*

b) El sistema devuelve al estado *Oculto* al secreto al ser imposible su desvelo: *Si*

Capítulo 3

Conclusiones

3.1. Conclusiones

Durante el desarrollo de este proyecto me han ido surgiendo dudas sobre la viabilidad del planteamiento del mismo, es decir, el reposar la responsabilidad de la seguridad en gran medida en el lado cliente de un sitio web. Por un lado he visto que es posible, sin embargo conlleva una serie de ventajas e inconvenientes.

- **Ventajas**:
  - Todos los mensajes parten del cliente ya cifrados, lo que asegura la privacidad de las comunicaciones frente a terceros indeseados que puedan estar espiando.
  - La seguridad aumenta al emplear protocolos personalizados de seguridad. El protocolo Diffie-Hellman permite negociar de forma rápida claves de usar y tirar, lo que dificulta la obtención de los mensajes por parte de los atacantes.
  - La resistencia contra el robo de contraseñas se intensifica. Al basar la identificación en un protocolo de desafío respuesta con claves RSA, y almacenar dicha clave RSA cifrada con otra contraseña personal de cada usuario, un atacante debería primero conseguir esa clave RSA (que solo se transmite una vez por la red, cifrada, a la hora de la activación), y posteriormente tratar de descifrar la clave RSA.

    Por otro lado un ataque de fuerza bruta requeriría generar un numero enorme de claves, lo que computacionalmente es inabordable actualmente.

    - El código responsable de la seguridad es 100 % accesible y corregible en cualquier momento.

- **Desventajas:**

    - La responsabilidad de conservar las claves es del usuario, no habiendo forma sencilla de recuperaras en caso de pérdida, y pudiendo dar lugar a que no pueda recuperar sus mensajes.
    - El código del cliente viaja sin cifrar por la red. El gran punto débil del proyecto es que los archivos con el código JavaScript viajan sin cifrar, pudiendo ser interceptados y modificados, lo que compromete toda la seguridad de la aplicación.
    - No hay forma de asegurar la identidad del servidor. Un tercero malicioso lo suficientemente hábil podría hacerse pasar por el servidor y de esta manera obtener las claves de los usuarios.
    - El rendimiento se reduce. JavaScript es un lenguaje muy útil para añadir dinamicidad a las páginas web, pero a la hora de realizar procesos computacionalmente mas costosos es muy lento y tiene limitaciones de memoria considerables, pudiendo llegara bloquear el navegador en procesos largos.

Los problemas de seguridad mencionados en las desventajas serían fácilmente solucionables incorporando al sistema el uso del protocolo HTTPS, que se encarga de certificar la identidad del servidor a través de una entidad certificadora y cifrar los mensajes transmitidos entre cliente y servidor. Ya que uno de los objetivos del proyecto era que todos los procesos de seguridad estuviesen implementados por mi mismo, no consideré esta opción. Sin embargo, si el objetivo fuese la viabilidad del producto, el empleo del protocolo HTTPS sería indispensable.

El desarrollo de este proyecto me ha enseñado a no subestimar los proyectos de software, ya que el esfuerzo requerido para llevarlo a cabo ha sido superior al esperado. En un principio, el objetivo era llevarlo a cabo durante el segundo cuatrimestre de clase, compaginadolo con las asignaturas. Sin embargo, llegado el momento vi necesario ampliar el tiempo para no presentar un proyecto incompleto.

Desde el punto de vista del desarrollo, lo más costoso ha sido implementar los protocolos de comunicación entre el cliente y el servidor, y en especial la coordinación entre los tipos de datos de Java y JavaScript, que han consumido gran parte de mi tiempo intentando resolver conflictos. También la implementación de los procesos criptográficos en JavaScript me consumió más tiempo del esperado ya que, a pesar de tenerlos implementados en Java,

replicar el proceso en JavaScript de forma que el resultado fuera el deseado fue todo un reto.

3.2. Posibles ampliaciónes

Lo primero que debería hacerse si se continuase con este proyecto sería realizar pruebas a fondo para asegurar la seguridad. Habría que implementar sistemas para asegurar la robustez ante la pérdida de contraseñas por parte del usuario, por ejemplo con un sistema de petición de contraseñas al otro usuario del diálogo o grupo.

El proyecto también podría ampliarse con más funciones de seguridad que se quedaron en el tintero a la hora de plantear el proyecto, como la firma de mensajes o conversaciones privadas, cuyos mensajes no se almacenen en la BD y se pierdan al abandonar la conversación.

También sería interesante considerar añadir más opciones de configuración de perfil, permitiendo tener una foto de perfil y de grupo, cambiar la contraseña que protege la clave RSA, etc...

Otra posibilidad sería desarrollar una aplicación de escritorio y una aplicación móvil, aunque se salga del concepto de sistema web sin instalación.

Capítulo 4

Manual de usuario

4.1. Manual de despliegue

En este manual seguiremos los pasos necesarios para el despliegue de la aplicación en un servidor. Los requisitos para que la aplicación funcione son los siguientes:

1. Servidor Tomcat8
2. Gestor de bases de datos MySQL
3. Java JRE v1.8

El proceso de despliegue se puede realizar en cualquier sistema operativo que cumpla los requisitos anteriores. En este manual se detallará el proceso en el sistema *Linux Mint17*, sin embargo, el proceso es igual o similar en otros sistemas, variando la instalación de los elementos anteriores y la localización en el sistema de archivos de los directorios donde se guardará la aplicación.

4.1.1. Instalación de Tomcat8

Para instalar el servidor Tomcat8 debemos descargar la versión adecuada para nuestro sistema operativo de su página oficial (Bibliografía *Descargar Tomcat8* [1]. Una vez descargado, lo extraeremos en el directorio que deseemos. Recuerde la localización donde lo haya extraído, pues será necesario para el resto del despliegue.

4.1.2. Instalación de MySQL

Para instalar MySQL en linx, debemos ejecutar el comando **`sudo apt-get install mysql-server mysql-client`** seguiremos los pasos para con-

figurar la cuenta de administrador de MySQL.

Una vez instalado, podemos instalar phpmyadmin para facilitar la gestión de la BD ejecutando el comando **sudo apt-get install phpmyadmin**.

Para acceder a phpmyadmin abriremos un navegador web e introduciremos la dirección *localhost/phpmyadmin*.

En caso de que recibamos el mensaje 404 not found, debemos ejecutar el comando **sudoln -s /us/share/phpmyadmin /vat/www/html/** y ya podremos acceder.

4.1.3. Instalación de Java 1.8

Para instalar Java 1.8 debemos añadir el repositorio de Java8 de oracle e instalar el paquete correspondiente ejecutando los comandos siguientes:

1. **sudo add-apt-repository ppa:webupd8team/java**
2. **sudo apt-get update**
3. **sudo apt-get install oracle-java8-installer**

4.1.4. Preparar la aplicación

Una vez que está todo instalado debemos extraer el contenido del archivo *aplicacion.zip* dentro del directorio *webapps* que se encuentra dentro del directorio donde extraíamos el servidor Tomcat8.

Para que la aplicación funcione, deberemos extraer el contenido del archivo comprimido *libs.zip* dentro del directorio lib del directorio de instalación de Tomcat8.

A continuación debemos importar la base de datos. Para ello podemos hacerlo mediante la linea de comandos o usando el gestor phpmyadmin.

Mediante linea de comandos, deberemos abrir un terminar y navegar hasta el directorio donde se encuentra el archivo *securecom.sql* y ejecutar el comando **mysql -u root -p < securecom.sql** e introducir la contraseña de administrador de mysql.

Utilizando el gestor phpmyadmin debemos identificarnos con nuestra cuenta de administrador de MySQL y acceder a la pestaña *Importar*. Una vez ahí, seleccionaremos el archivo *securecom.sql* y pulsaremos importar.

4.1.5. Configuración

Con esto la aplicación está lista para funcionar. Sin embargo, podemos ajustar algunos parámetros de configuración si lo deseamos (o si es necesario).

Usuarios de la BD

Junto con el resto de la BD se habrán creado los usuarios necesarios para que la aplicación funcione. Sin embargo, si deseamos modificar sus contraseñas por defecto, debemos ir al directorio donde extrajimos la aplicación (dentro del directorio webapps/TFG de Tomcat8) y modificar el archivo *context.xml* dentro del directorio *META-INF*.En este archivo podemos encontrar toda la configuración de la aplicación.

Cada usuario de la BD está definido dentro de un recurso. Debemos buscar el atributo *password* y sustituir su contenido por el que deseemos.

La contraseña debe coincidir con la que especifiquemos en el gestor de bases de datos MySQL para que funcione.

Adicionalmente, si nuestra base de datos se va a alojar en una dirección diferente a la máquina donde se ejecutará el servidor Tomcat8, debemos especificar dicha dirección en el atributo *url* de cada usuario.

Cuenta de correo

Por defecto, la aplicación empleará una cuenta de correo específica para enviar los correos de activación. Si deseamos cambiar la cuenta de correo por defecto, debemos editar el archivo *context.xml* mencionado anteriormente y modificar el recurso *mail/Mailer*, estableciendo el nombre de usuario de nuestro correo en los atributos *username* y *mail.user*, la contraseña en *password* y *mail.password* y el host SMTP de nuestra cuenta de correo en *mail.smtp.host*.

Clave del servidor

Para algunas operaciones, el servidor emplea una clave simétrica de 128 bits. Si deseamos cambiar la clave por defecto, debemos editar el archivo *context.xml* mencionado anteriormente y modificar el parámetro con el nombre *AESK*, estableciendo su valor a la nueva clave que deseemos.

Nota: la clave debe estar compuesta por 16 caracteres alfanuméricos, ni mas, ni menos.

Directorio de archivos

El sistema necesita un directorio para almacenar los archivos compartidos por los usuarios. Para establecer el directorio donde queremos que se almacenen debemos modificar el archivo *context.xml* mencionado anteriormente y modificar el parámetro con el nombre *UPLOADS_PATH*, y establecer su

valor con la dirección de la carpeta de nuestro sistema de archivo que deseemos.

Este paso será obligatorio si va a desplegar la apliación en un sistema diferente a linux, ya que por defecto la carpeta indicada sigue el esquema de almacenamiento de linux.

Nota: debemos asegurarnos de que el usuario con quien ejecutemos el servidor Tomcat8 tenga permisos de escritura sobre el directorio escogido.

4.1.6. Despliegue de la aplicación

Por fin podemos desplegar la aplicación. Para ello debemos ir al directorio *bin* dentro del directorio de Tomcat8 y ejecutar el archivo *startup.sh* (en linux) o *startup.bin*(en windows). La aplicación quedará desplegada.

Para probar que la aplicación funciona, debemos abrir un navegador web y dirigirnos a la dirección *localhost:8080/TFG/access*. El navegador debería acceder a la página de acceso de la aplicación.

Para más información sobre el funcionamiento de la aplicación vea el manual siguiente.

4.2. Manual de uso

La aplicación está diseñada para que sea muy sencilla e intuitiva usar. El único requisito

El primer paso del proceso es el registro. La primera pantalla que nos encontramos al acceder a la aplicación es la pantalla de acceso.

4.2.1. Acceso a la aplicación

Figura 4.1: Pantalla de acceso

Para registrar un nuevo usuario debe introducir sus datos en el formulario de registro y pulsar el botón *Enviar Solicitud*. El sistema procesará la solicitud de registro y, una vez que un administrador la haya aceptado, recibirá un correo electrónico con un link de activación que deberá seguir para proceder con la activación.

En la pantalla de activación deberá esperar a que el sistema genere una clave de usuario personal. Una vez que se haya generado, deberá introducir su contraseña de usuario y pulsar el botón *Descargar*. El sistema protegerá su clave de usuario con su contraseña y descargará el archivo con extensión *.cod* conteniendo la clave.

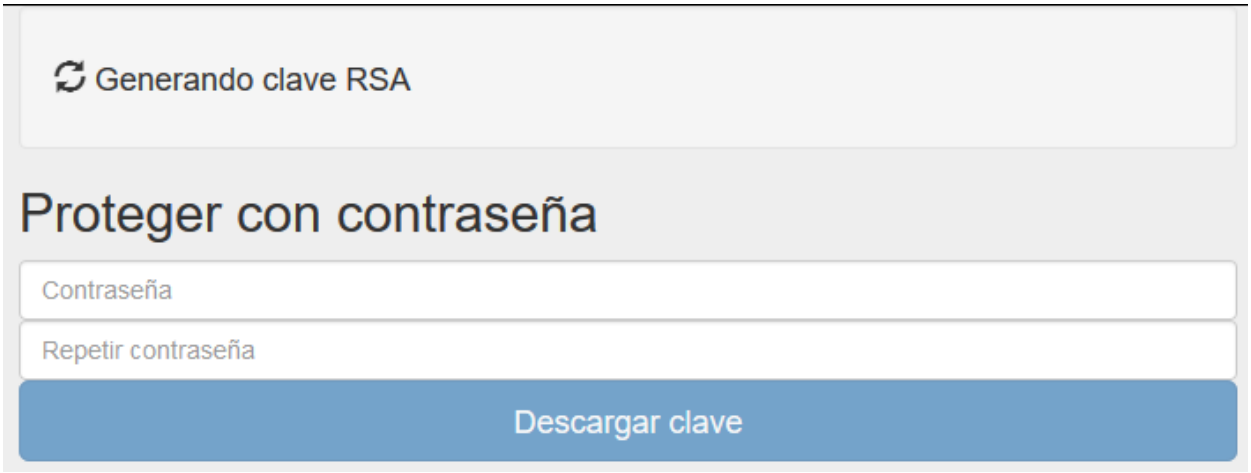

Figura 4.2: Pantalla de acceso

¡Importante! - asegúrese de guardar su clave, pues una vez activada la cuenta, no tendrá oportunidad de recuperarla, y es su vía de acceso al resto de la aplicación.

Una vez que haya activado la cuenta, vuelva a la pantalla de acceso e introduzca sus datos en el formulario de inicio de sesión para acceder a la aplicación.

4.2.2. Pantalla principal

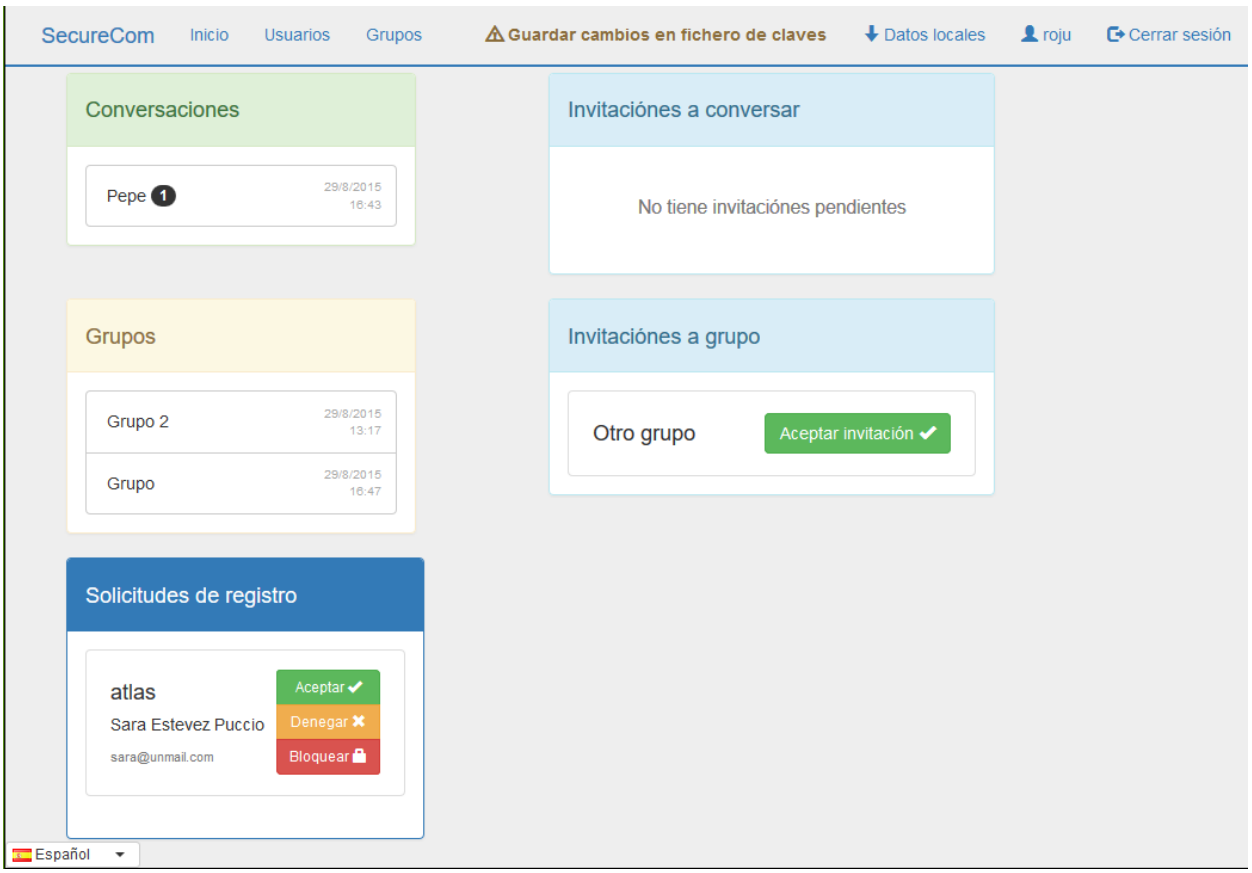

Figura 4.3: Pantalla principal

Si es nuevo en la aplicación o ha cambiado de dispositivo o navegador, la aplicación le pedirá que le indique un fichero para importar sus claves o bien crear un nuevo archivo.

La pantalla principal le informará del estado de sus comunicaciones. Dicha pantalla presenta cuatro áreas donde se le mostrarán. Las dos de la izquierda muestran sus diálogos y grupos, indicando la fecha y hora del último mensaje junto con la cantidad de mensajes sin leer.

Las dos de la derecha muestran las invitaciones a dialogo y a grupos, permitiendo aceptarlas si lo desea.

En caso de que su usuario sea el administrador, se mostrará un quinto área donde se le informará de las solicitudes de registro pendientes, y se le permitirá decidir que hacer con ellas; aceptarlas, rechazarlas o bloquearlas

para no recibir más solicitudes con el mismo correo electrónico o nombre de usuario.

La barra superior se mantendrá en el resto de las páginas de la aplicación, y le permitirá navegar por la aplicación, así como importar o exportar su archivo de claves y cerrar la sesión. Procure cerrar siempre la sesión antes de dejar de usar la aplicación como medida de seguridad.

¡Importante! - el archivo de claves se almacena en un espacio gestionado por el navegador. Procure realizar una copia de seguridad de sus claves frecuentemente para evitar su pérdida ya que podrían borrarse al limpiar la caché de su navegador. Además, si va a utilizar la aplicación desde un dispositivo diferente, necesitará sus claves.

También, en todo momento en la esquina inferior izquierda encontrará un control para seleccionar el idioma en que quiere que se muestre la página.

4.2.3. Pantalla de usuarios

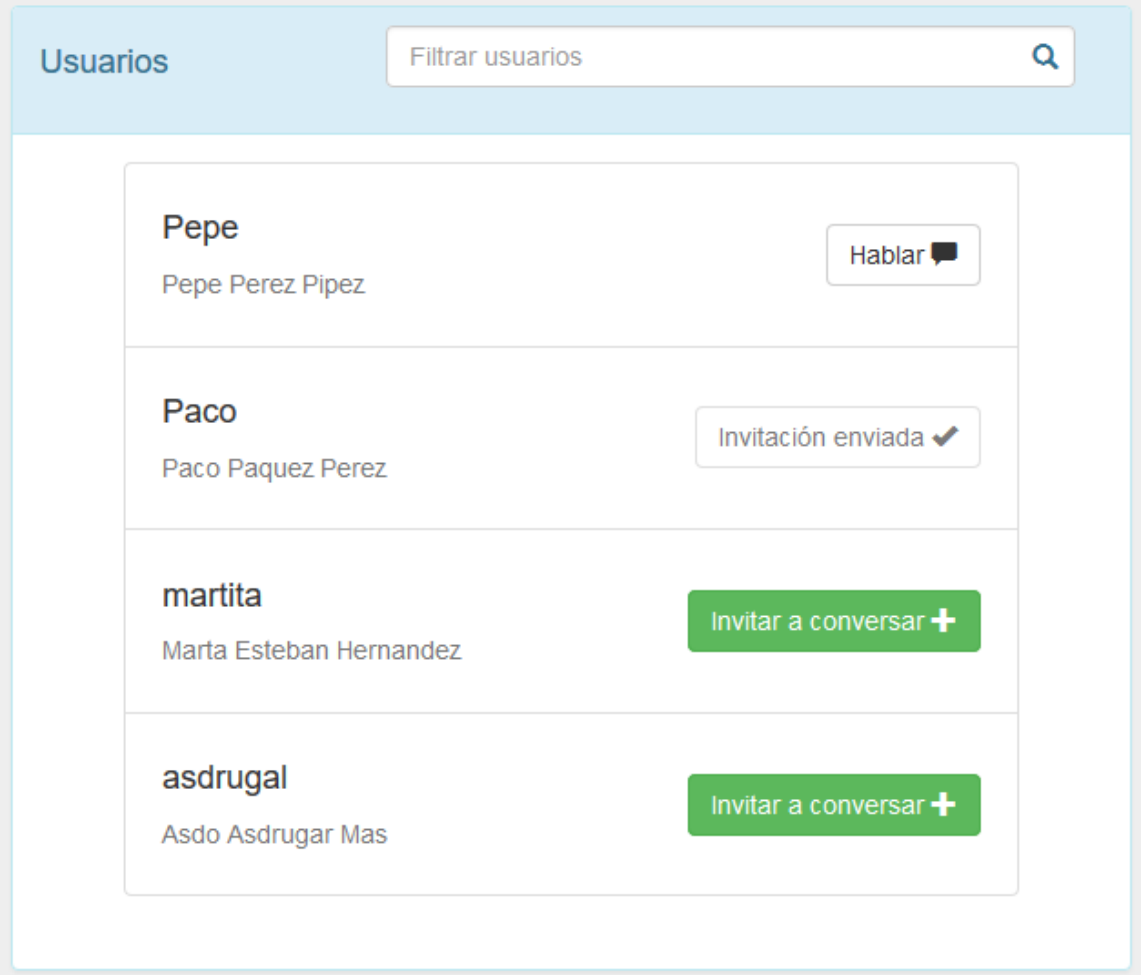

Figura 4.4: Pantalla de usuarios

La pantalla de usuarios es accesible a través de los enlaces de la barra de navegación.

En esta página podrá visualizar los usuarios que forman parte de la aplicación actualmente, y filtrar los mismos por nombre de usuario, nombre, apellidos o correo electrónico.

Para cada usuario se mostrarán los controles para invitarle a dialogar, aceptar su invitación si el usuario ya le invitó a dialogar o acceder a dialogar con el, si ya se ha aceptado un diálogo.

4.2.4. Pantalla de grupos

La pantalla de grupos es accesible a través de los enlaces de la barra de navegación.

En esta página podrá visualizar los grupos de los que es miembro o a los que ha sido invitado y filtrar los mismos por nombre.

Para cada grupo se mostrarán los controles para aceptar la invitación a formar parte del mismo si ha sido invitado, los controles para acceder al grupo o dejarlo, en caso de que ya forme parte de él.

Adicionalmente, en esta pantalla podrá crear nuevos grupos introduciendo un nombre y pinchando en el botón *Crear grupo*.

Figura 4.5: Pantalla de grupos

4.2.5. Pantalla de diálogo

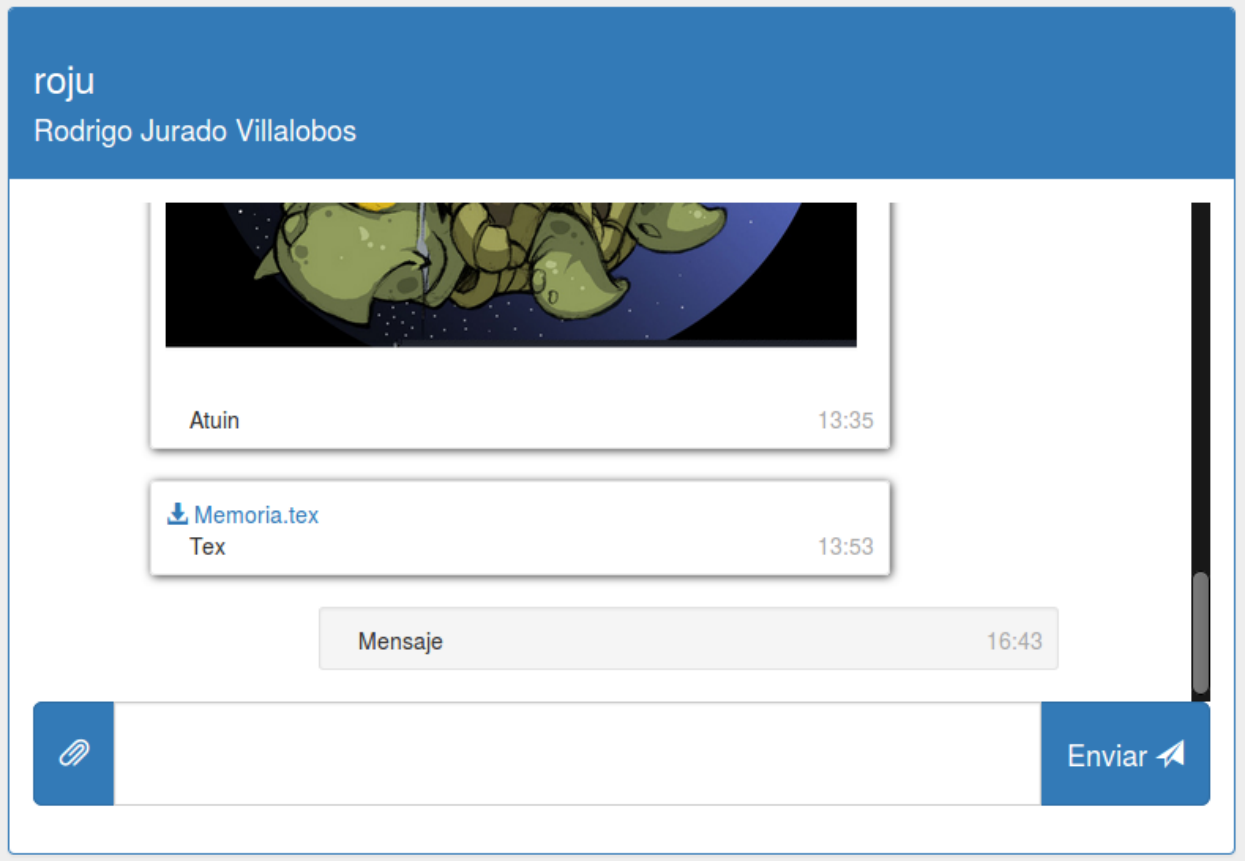

Figura 4.6: Pantalla de diálogo

La pantalla de diálogo le permitirá intercambiar mensajes con otro usuario. Al acceder a ella deberá esperar un momento a que la aplicación verifique su identidad y cree un canal seguro. Una vez haya terminado, podrá ver los datos del usuario con quien dialoga en la cabecera. El cuerpo del diálogo mostrará los mensajes que se intercambien a tiempo real, y en la parte inferior podrá escribir mensajes y enviarlos a su remitente.

Adicionalmente podrá enviar archivos y ver los archivos que le ha enviado el otro usuario. Cuando envíe un archivo adjunto, deberá esperar a que se cifre y se suba al servidor. Si abandona la página mientras se procesa, el mensaje no se mandará.

Cada mensaje puede incorporar un solo archivo adjunto, y no se descargará hasta que usted pinche sobre el archivo adjunto.

Nota: para utilizar las funciones de archivos debemos emplear un navegador compatible con Java, tener el plugin de Java activado (suele venir activado, si no en la página oficial de Java) y dar permiso al navegador de ejecutar el applet.

Además, dado que la aplicación es un proyecto estudiantil, el applet no posee un certificado, por lo que hay que agregar las páginas de dialogo y grupo a la lista de excepciones de Java

Para añadir a excepciones de Java las páginas de dialogo y grupo debemos abrir el panel de control de Java (en windows busque *"Configurar java"* en el menú de inicio, en linux ejecute el comando **jcontrol**) y vaya a la pestaña *Seguridad*. En la parte inferior del panel encontrará la lista de sitios. Deberá editar la lista de sitios y añadir la dirección de la página de diálogo y la página de grupo (la url completa sin incluir el id de grupo que aparece en la barra de dirección del navegador cuando se conecta).

Una vez hecho esto, si su navegador es compatible con Java podrá adjuntar archivos a sus mensajes.

(Para más información visitar las páginas oficiales de Java *¿Cómo puedo activar Java en el navegador?* [9] y *¿Cómo puedo configurar la lista de excepciones de sitios?* [10]).

4.2.6. Pantalla de grupo

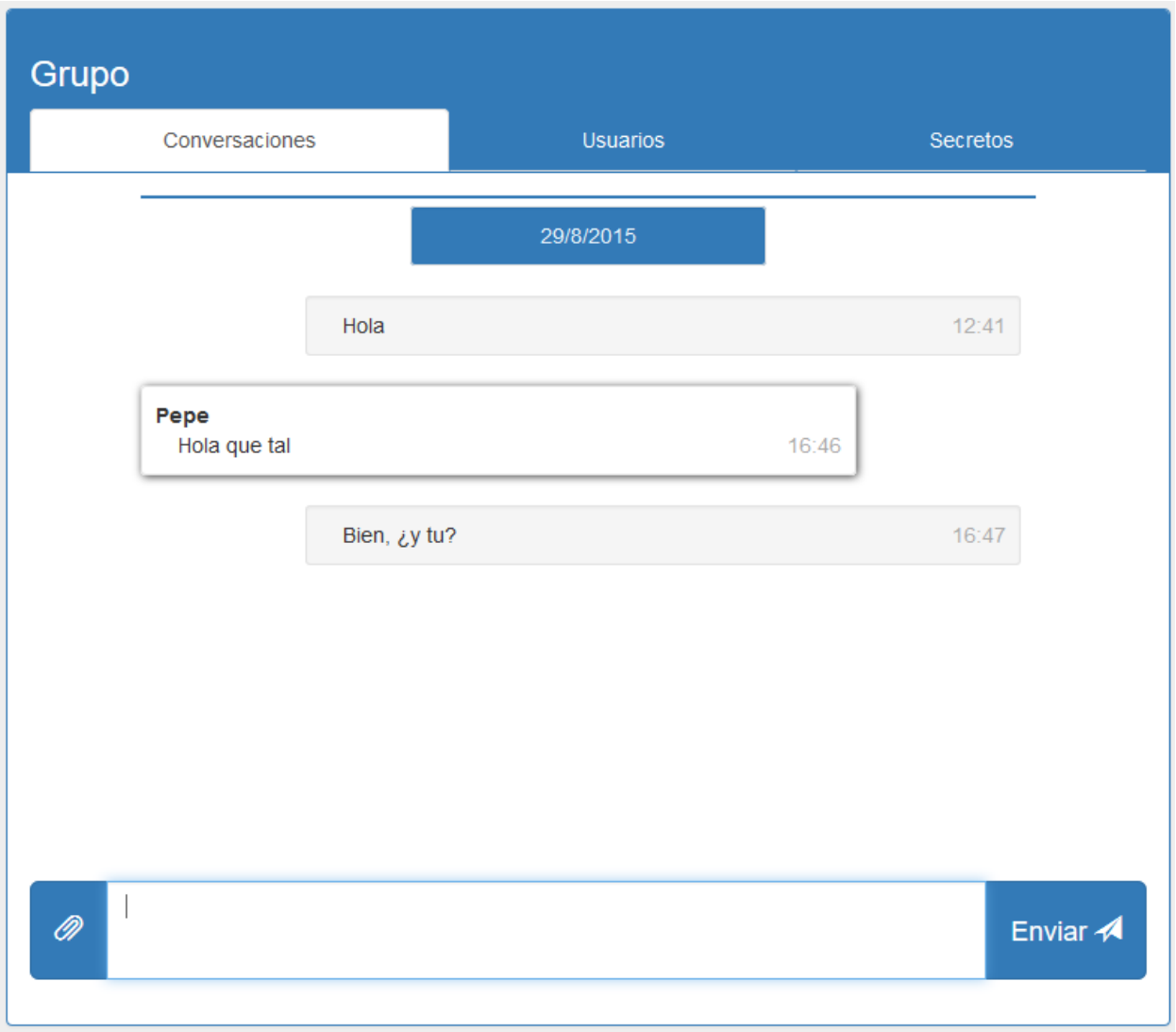

Figura 4.7: Pantalla de grupo

La pantalla de grupo es muy similar a la de diálogo con algunas diferencias. La primera de ellas es que en cada mensaje se indicará el nombre del miembro del grupo que lo envió.

Aparte en la cabecera del grupo hay tres pestañas que corresponden a los mensajes, miembros y secretos del grupo.

La pestaña de miembros del grupo permite visualizar los miembros que forman parte del grupo.

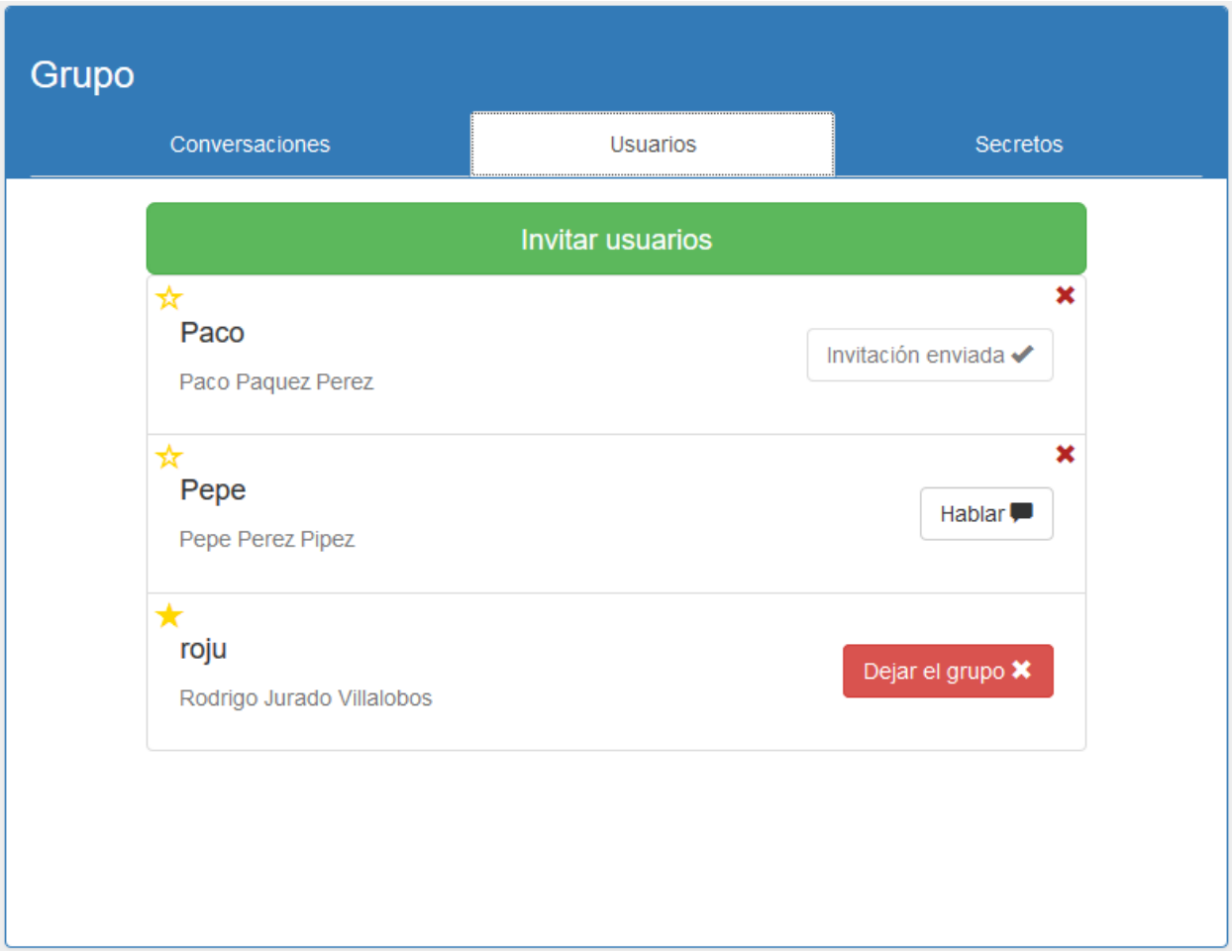

Figura 4.8: Pantalla de miembros del grupo

Adicionalmente, si es líder del grupo, esta pestaña le permitirá administrar los miembros del grupo, expulsando miembros pinchando en la cruz roja o darle el puesto de líder pinchando en la estrella que hay junto a su nombre.

También le permitirá invitar a otros usuarios al grupo, pinchando en el botón *Invitar usuarios* y posteriormente en el botón *Invitar al grupo* junto al nombre del usuario a quien quieras invitar.

La pantalla de secretos le mostrará los secretos creados por los miembros del grupo y su estado.

Si el secreto fue *desvelado* se le mostrará su contenido.

Si el secreto está *oculto* se le dará la oportunidad de proponer desvelarlo, entonces tendrá que esperar a que el resto de miembros acepte o rechace la proposición.

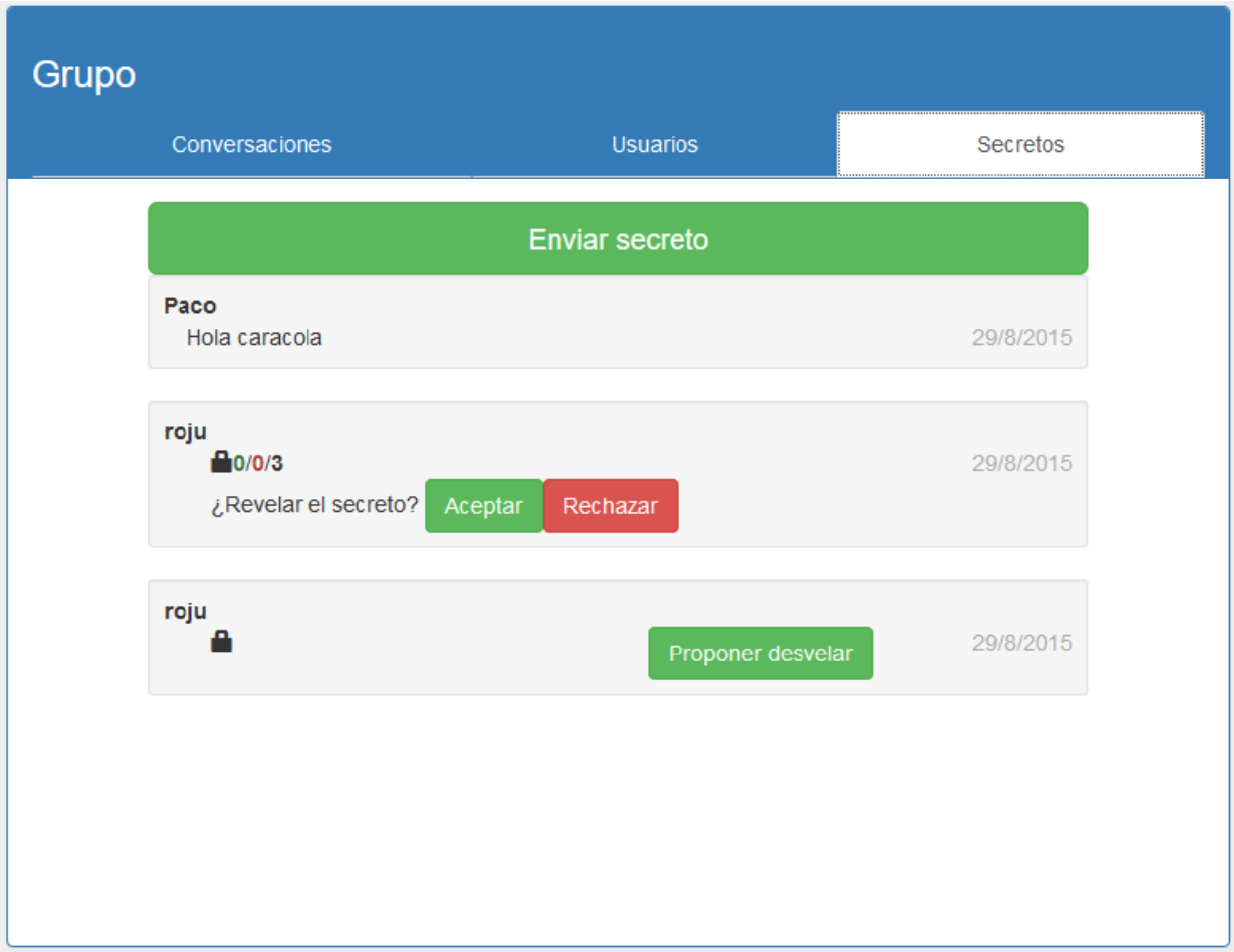

Figura 4.9: Pantalla de secretos del grupo

Si alguien propuso que se desvele el secreto podrá ver cuantos miembros aceptaron (en color verde), cuantos rechazaron (en rojo) y cuantos faltan por decidir (en negro), y podrá decidir aceptar o rechazar la proposición.

Si ya ha decidido aceptar o rechazar la proposición se le mostrará su decisión.

Capítulo 5

Anexos

5.1. Tablas de referencia para la estimación del esfuerzo

5.1.1. Complejidad

Archivos referenciados	Elementos de datos		
	1-4	5-15	>15
0-1	Baja	Baja	Media
2	Baja	Baja	Media
3 o más	Media	Alta	Alta

Cuadro 5.1: Criterios de complejidad para Entradas Externas

Archivos referenciados	Elementos de datos		
	1-5	6-19	>19
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
>3	Media	Alta	Alta

Cuadro 5.2: Criterios de complejidad para Salidas Externas

Archivos referenciados	Elementos de datos		
	1-5	6-19	>19
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
>3	Media	Alta	Alta

Cuadro 5.3: Criterios de complejidad para Consultas Externas

Tipos de Registro	Tipos de Datos		
	1-19	20-50	>50
1	Baja	Baja	Media
2-5	Baja	Media	Alta
>5	Media	Alta	Alta

Cuadro 5.4: Criterios de complejidad para Archivos Lógicos Internos

5.1.2. Relación valor-complejidad por elemento

Clasificación	Valores		
	Entradas Externas	Salidas Externas	Consultas Externas
Baja	3	4	3
Media	4	5	4
Alta	6	7	6

Cuadro 5.5: Puntos de función de entradas, salidas y consultas externas según su complejidad

Clasificación	Valores	
	Archivo Lógico Interno	Archivo de Interfaz Externo
Baja	7	5
Media	10	7
Alta	15	10

Cuadro 5.6: Puntos de función de archivos internos y externos según su complejidad

5.1.3. Valores para el cálculo del esfuerzo con COCOMO

Modo de desarrollo	Persona-mes	Tiempo de desarrollo
Orgánico	$E = 3{,}2KLDC^{1,05}$	$TD = 2.PM^{0,38}$
Semiacoplado	$E = 3{,}0KLDC^{1,12}$	$TD = 2.PM^{0,35}$
Integrado	$E = 2{,}8KLDC^{1,2}$	$TD = 2.PM^{0,32}$

Cuadro 5.7: Funciones para la estimación por COCOMO según el tipo de proyecto

5.1.4. Valores para la asignación de peso para puntos de Caso de Uso

Tipo de Actor	Descripción	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API)	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3

Cuadro 5.8: Peso de los actores sin ajustar (UAW)

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10
Complejo	El Caso de Uso contiene más de 8 transacciones	15

Cuadro 5.9: Peso de los Casos de Uso sin ajustar (UUCW)

Bibliografía

[1] *Descargar Apache Tomcat8*. Apache. Agosto 2015. URL: https://tomcat.apache.org/download-80.cgi.

[2] *Function Point Languages Table*. Quantitative Software Management. Agosto 2015. URL: http://www.qsm.com/resources/function-point-languages-table.

[3] *jQuery API*. jQuery Foundation. Agosto 2015. URL: http://api.jquery.com/.

[4] NIST. *FIPS PUB 197: the official AES Standard*. 2001.

[5] L. Adleman R. Rivest A. Shamir. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. 1977.

[6] Amparo Fúster Sabater. *Técnicas criptográficas de protección de datos*. Madrid : RA-MA, 2003.

[7] Adi Shamir. "How to share a secret". En: *Communications of the ACM* 22.11 (1979).

[8] *W3Schools*. Refsnes Data. Agosto 2015. URL: www.w3schools.com.

[9] *¿Como puedo activar java en el explorador web?* Oracle Corporation. Agosto 2015. URL: https://www.java.com/es/download/help/enable_browser.xml.

[10] *¿Cómo puedo configurar la lista de excepciones de sitios?* Oracle Corporation. Agosto 2015. URL: https://www.java.com/es/download/faq/exception_sitelist.xml.