

**Universidad de las Ciencias Informáticas**

**Facultad 1**



**Título:** “Sistema de mensajería instantánea para el Gestor de Documentos Administrativos eXcriba”

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:**

Dayan Arquímedes Bermúdez Cruz

**Tutor:**

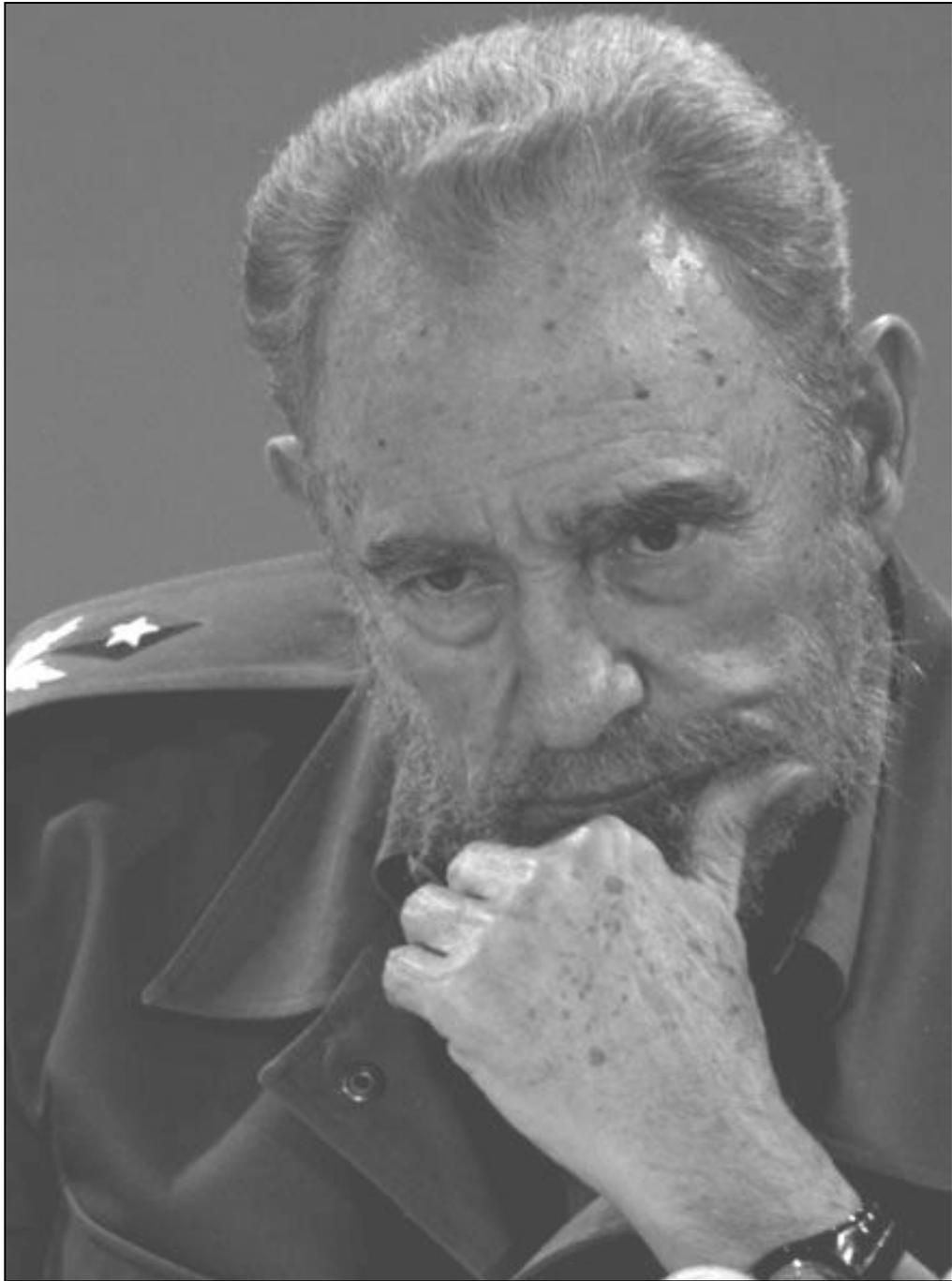
Ing. Alexander Hernández Chapman

**Cotutor:**

Ing. Julio Antonio Zamora Rosabal

**La Habana, Cuba**

**Junio 2013**



*...quien no sea capaz de luchar por otros, no será nunca  
suficientemente capaz de luchar por sí mismo...*

## *Declaración de autoría*

Declaro que soy el único autor de esta investigación y autorizo a la Facultad 1 de la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del autor

Dayan Arquímedes Bermúdez Cruz

---

Firma del tutor

Ing. Alexander Hernández Chapman

---

Firma del tutor

Ing. Julio Antonio Zamora Rosabal

## *Datos del contacto*

### **Datos del tutor**

Nombre: Alexander Hernández Chapman

Especialidad de graduación: Ingeniero informático

E-mail: [alx@uci.cu](mailto:alx@uci.cu)

Años de graduado: 7

### **Datos del cotutor**

Nombre: Julio Antonio Zamora Rosabal

Especialidad de graduación: Ingeniero informático

E-mail: [jazamora@uci.cu](mailto:jazamora@uci.cu)

Años de graduado: 1

## *Agradecimientos*

*Son muchas las personas a las que agradecer, pero primero que nada agradecerle a Dios, El Señor Todopoderoso por haberme ayudado, pues cuando le pedía que me salieran bien las cosas siempre salía adelante y con una sonrisa en la cara, muchas gracias Señor por dejarme vivir este momento. Agradecer a mi familia completa por darme fuerzas y palabras de aliento, especialmente a mi abuela Lidis, mi madre Alexis, mi tío Jesús, mi primo Adrián y mi prima Ariadna. A mis tutores (Alexander y Julio) y a mi amigo Adrián, que hicieron posible la realización de esta tesis. A mujeres de alta importancia y alto calibre que han pasado por mi vida cuyos nombres son honorables de solo mencionarlos: Yaniset, Dianelys, Yaimarelis, Yulién, Yuliet, Yolaine y Yaima. A 3 personas que han sido súper consideradas, serviciales y comprensivas conmigo: Célida, Eric y Gladys. A los chamacos que se han criado conmigo en el barrio: Yoelby, Pocholo, Edelito, Sergito (Mandarrio), Milán, Yunier Jaca, Yunier (El de Chichi), Iván, Maruán, Juan Carlos, Lincoln, Asley, Yuliany, Raudel, Esteban, Maikel, Armando, Ulises, Pelao, Enmanuel, Aniubis, Hipsis y Ana Haidée. A las personas mayores de mi zona que también me han dado fuerzas para seguir adelante: Pedro, Jorge Ramos, Teresa, Melón, Silveria, Iris, Román, Mayelín, Nilda, Yanisbel, Chiquitico, Mildrey, Berta, Risquet, Orbelis, Saray, Maray, Coco, Orlando (Temetrepo), Carmen, El Cusi, Blanca, El Coto, Abelito, Dalila, Marlenis, Eddy, Ricardito, Celia, Leticia, Yudisbel, Erasmo, Yolanda, Chiri, Jorge, Máximo, Gustavo (El Carpintero), Odalys, Deisy, Evaristo, El Ñato, Oscar, El Yoyo, Serafín, José Espinosa (Que en paz descanse), Hilda y Magaly. A mis amistades de la UCI que han compartido miles de momentos conmigo durante toda la carrera: Yexenia, Dina, Yadi, Patricia, Lisbet, Elisa, Lisandra, Mahelis, Arianna, Yamilka, Yinelys, Claudia, Yanelis, Yaima, Mily, Sonia, Yadira, Iliana, Yennifer, Viviana, María, Odette, Lili, Dainelys, Jineth, Yarineth, Yelienny, Amanda, Hilario, Riverón, Alexis, Carlos Michel, Jose Miguel, El Indio, Lachy, Magdiel, Julio César (El Chino), El Mangui, Juan Manuel (Jota), Yoani, Lairon, Yerandy, Ernesto, Odeinis, Luis Alfredo (El Chino), Luis Andrés (El Luiso), El Santi, Jose Alberto, Henry, Tomás, Los Yasmany, Los Intelectuales, Danny, Raciél, Yoandri, Yaidel (El Flaco), los DJ y en especial a la gente de “La Mosca” (Juan Carlos y Marcelo). A todo aquel que me ha ayudado directa o indirectamente, que aunque no los ponga en estos agradecimientos porque son incontables, han sido de vital importancia durante mi vida en esta universidad ya que algunos me han enseñado lo bueno y otros lo malo, pero en general si no hubiera sido por todos ellos no sería un profesional más completo.*

**Dayan Arquímedes Bermúdez Cruz**

## *Dedicatoria*

*A mi abuela Lidis por ser la persona que más me quiere en este mundo sin importarle nunca nada, ni lo que me pase, ni lo que diga o haga, ni como me sienta y mucho menos como piense; a ella principalmente está dedicada esta tesis.*

*A mi mamá Alexis que se ha sacrificado siempre por mí en situaciones que nadie se imagina, si no fuera por ella y por su confianza en mí no hubiera sido posible la realización de este hombre y de esta tesis.*

*A mi papá Omar que no se encuentra entre nosotros pero que siempre tuvo la convicción de que iba a ser grande en algún momento de mi vida, papá, este es uno de mis momentos en los que me siento de esa manera.*

*A Adrián por ser la única persona que de verdad me ha demostrado que el concepto de amistad verdadera existe todavía cuando siempre pensé que no existía para mí.*

*A la Universidad de las Ciencias Informáticas por ser la universidad más versátil que existe en nuestro país y por permitirme conocer a miles de personas que gracias a ellas y sus vivencias, han hecho de mí el ser humano que soy.*

*Dayan Arquímedes*

## *Resumen*

El GDA eXcriba 2.0 no ofrece una funcionalidad que le permita a sus usuarios intercambiar mensajes de texto al instante. Para realizar dicha acción hay que acudir a otras aplicaciones como los clientes de mensajería instantáneas ajenos al sistema. El uso de estas aplicaciones y otros medios de comunicación alternativos como el teléfono, la intranet y el correo electrónico, trae aparejado que al desplegarse el eXcriba en una empresa o entidad que carezca de ellos, la posibilidad de intercambiar mensajes en tiempo real o información referente a los procesos que se desarrollan en el eXcriba se verá limitada o disminuida.

El presente trabajo tiene como objetivo desarrollar un módulo que al ser integrado al GDA eXcriba permita el intercambio de mensajes de texto en tiempo real entre sus usuarios autenticados. Para su desarrollo, se utiliza la metodología RUP asociada al lenguaje de modelado UML, Zend Studio 8.0 como entorno de desarrollo integrado y JQuery 1.3.2 como framework. También se utilizan PHP 5.3 y JavaScript 2.2 como lenguajes de programación y como Sistema Gestor de Base de Datos PostgreSQL 8.4. Finalmente fueron implementadas las funcionalidades básicas y realizadas sus pruebas correspondientes, obteniendo resultados satisfactorios.

## *Palabras clave*

Cientes de mensajería instantánea, GDA eXcriba, JavaScript, mensajes de texto, PHP, PostgreSQL.

# Índice

<i>Introducción</i> .....	<b>1</b>
<i>Capítulo 1. Fundamentación teórica</i> .....	<b>6</b>
<b>Introducción</b> .....	<b>6</b>
<b>1.1 Estudio de los sistemas de mensajería instantánea</b> .....	<b>6</b>
1.1.1 Pidgin .....	6
1.1.2 PSI .....	7
1.1.3 Empathy .....	8
1.1.4 EBuddy .....	9
1.1.5 Meebo.....	9
<b>1.2 Estudio de los sistemas de gestión documental</b> .....	<b>11</b>
1.2.1 Alfresco .....	11
1.2.2 Knowledge Tree .....	12
1.2.3 LetoDMS.....	12
1.2.4 OpenKM .....	13
<b>1.3 Metodología de desarrollo RUP con nivel 2 de CMMI</b> .....	<b>14</b>
<b>1.4 Lenguajes, herramientas y tecnologías a utilizar</b> .....	<b>15</b>
1.4.1 Lenguaje de modelado UML .....	15
1.4.2 pgAdmin 3 .....	15
1.4.3 Visual Paradigm .....	16
1.4.4 Lenguajes de programación .....	16
1.4.5 Entorno de desarrollo Zend Studio .....	17
1.4.6 Framework JQuery .....	18
1.4.7 Sistema Gestor de Base de Datos PostgreSQL .....	18
<b>1.5 Conclusiones del capítulo</b> .....	<b>19</b>
<i>Capítulo 2. Características del módulo</i> .....	<b>20</b>
<b>Introducción</b> .....	<b>20</b>
<b>2.1 Propuesta de solución</b> .....	<b>20</b>
<b>2.2 Modelo de dominio</b> .....	<b>20</b>
<i>Figura 1: Modelo de dominio</i> .....	21
<b>2.3 Especificación de los requisitos de software</b> .....	<b>21</b>
2.3.1 Técnicas para la captura de los requisitos .....	22
2.3.2 Requerimientos funcionales .....	23
2.3.3 Requerimientos no funcionales .....	24
2.3.4 Técnicas de validación de requisitos.....	25
<b>2.4 Captura de requisitos como casos de uso</b> .....	<b>26</b>

2.4.1	Definición de los actores.....	26
	<i>Tabla 1: Definición de los actores del módulo</i> .....	27
2.4.2	Diagrama de casos de uso.....	27
	<i>Figura 2: Diagrama de casos de uso</i> .....	27
2.4.3	Validación de requerimientos mediante la matriz de trazabilidad.....	27
<b>2.5</b>	<b>Conclusiones del capítulo</b> .....	<b>28</b>
	<i>Capítulo 3. Análisis y diseño del módulo</i> .....	<b>29</b>
	<b>Introducción</b> .....	<b>29</b>
<b>3.1</b>	<b>Análisis y diseño</b> .....	<b>29</b>
<b>3.2</b>	<b>Modelo de análisis</b> .....	<b>29</b>
3.2.1	Diagrama de clases del análisis.....	30
	<i>Figura 3: Diagrama de clases del análisis para el caso de uso "Enviar mensaje"</i> .....	30
3.2.2	Diagrama de interacción.....	30
	<i>Figura 4: Diagrama de colaboración para el caso de uso "Enviar mensaje"</i> .....	31
<b>3.3</b>	<b>Modelo de diseño</b> .....	<b>31</b>
3.3.1	Arquitectura definida para el sistema.....	32
3.3.2	Análisis de la arquitectura del GDA eXcriba 2.0.....	34
	<i>Figura 5: Vista de la arquitectura del eXcriba</i> .....	34
3.3.3	Arquitectura del módulo.....	35
3.3.4	Patrones de diseño.....	36
3.3.5	Diagrama de clases del diseño.....	37
	<i>Figura 6: Diagrama de clases del diseño del caso de uso "Enviar mensaje"</i> .....	38
3.3.6	Modelo físico de la base de datos.....	39
	<i>Figura 7: Modelo físico de la base de datos del módulo de mensajería instantánea</i> .....	39
3.3.7	Estándares de codificación.....	39
	<i>Figura 8: Ejemplo del uso de los estándares de codificación</i> .....	40
	<i>Figura 9: Ejemplo del uso de los estándares de codificación</i> .....	40
	<i>Figura 10: Ejemplo del uso de los estándares de codificación</i> .....	41
	<i>Figura 11: Ejemplo del uso de los estándares de codificación</i> .....	41
	<i>Figura 12: Ejemplo del uso de los estándares de codificación</i> .....	42
3.3.8	Diagrama de despliegue.....	42
	<i>Figura 13: Diagrama de despliegue</i> .....	43
3.3.9	Diagrama de componentes.....	44
	<i>Figura 14: Diagrama de componentes</i> .....	45
3.3.10	Prototipo del sistema.....	45
	<i>Figura 15: Prototipo de interfaz de usuario del módulo de mensajería instantánea</i> .....	46
<b>3.4</b>	<b>Conclusiones del capítulo</b> .....	<b>46</b>
	<i>Capítulo 4. Validación de los resultados del módulo</i> .....	<b>47</b>
	<b>Introducción</b> .....	<b>47</b>

<b>4.1</b>	<b>Casos de prueba de caja negra .....</b>	<b>47</b>
<b>4.2</b>	<b>Casos de prueba de caja blanca.....</b>	<b>47</b>
	<i>Figura 16: Representación del algoritmo updateLoggedUsers() .....</i>	<i>48</i>
	<i>Figura 17: Grafo del flujo asociado al algoritmo updateLoggedUsers().....</i>	<i>49</i>
	<i>Tabla 4: Caminos básicos.....</i>	<i>50</i>
<b>4.3</b>	<b>Resultados de las pruebas realizadas .....</b>	<b>51</b>
	<i>Figura 18: Gráfica de no conformidades detectadas y corregidas correspondientes a las pruebas realizadas al módulo .....</i>	<i>51</i>
<b>4.4</b>	<b>Conclusiones del capítulo .....</b>	<b>51</b>
	<i>Conclusiones .....</i>	<i>53</i>
	<i>Recomendaciones.....</i>	<i>54</i>
	<i>Referencias bibliográficas .....</i>	<i>55</i>
	<i>Bibliografía .....</i>	<i>58</i>
	<i>Glosario de términos.....</i>	<i>60</i>
	<i>Anexos.....</i>	<i>64</i>

## *Introducción*

La tecnología es un valioso instrumento utilizado por la mayoría de las empresas y entidades actuales para establecer un buen comportamiento de las comunicaciones, tanto para uso interno entre las áreas de una misma entidad, como para comunicarse con otras entidades o empresas.

Gracias a una buena comunicación, se eliminan problemas como la poca o inadecuada comprensión de lo que se asigna en cuanto a tareas se refiere; la ineffectividad y la falta de conocimiento de lo que se quiere en realidad. Estos problemas no existen únicamente en el ámbito de las empresas, existen áreas de la gestión que los presentan; más específicamente, en el área de la gestión documental.

Según la Norma ISO 15489-1, la gestión documental es el área de gestión responsable del control eficaz y sistemático de la creación, recepción, mantenimiento, uso y disposición de documentos de archivo. Dentro de la gestión documental se encuentran los sistemas de gestión documental, los cuales son sistemas computarizados de información que incorporan, gestionan y facilitan el acceso a los documentos de archivo a lo largo del tiempo (Proyecto UNE-ISO 15489-1, 2005).

A nivel mundial existen sistemas que realizan la gestión documental, ejemplo de ello son el Gestor de Contenido Empresarial (por sus siglas en inglés *ECM*) Alfresco, OpenKM y Nuxeo. Cuba no está ajena al desarrollo de la gestión documental, en la Universidad de Ciencias Informáticas (UCI), específicamente en el Centro de Informatización Universitaria (CENIA) se encuentra el Departamento de Gestión Documental. En este departamento se desarrolla Gestor de Documentos Administrativos (GDA) eXcriba, cuyo núcleo es el ECM Alfresco.

Uno de los pilares fundamentales de la gestión de contenido empresarial es la colaboración, la cual es el arte de trabajar juntos (AIIM, 2013). Una de las formas de garantizar la colaboración es permitiendo la comunicación entre las personas que participan en los procesos que se desarrollan en el eXcriba. El GDA eXcriba 2.0 posee varias funcionalidades, sin embargo, no tiene implementada una que sea capaz de proporcionar a sus usuarios un espacio para la comunicación interna. El intercambio de mensajes de texto en tiempo real es un aspecto que no gestiona el GDA eXcriba.

La inexistencia de este valor agregado es un inconveniente para empresas o entidades que tengan

desplegado el eXcriba y a su vez carezcan de medios de comunicación como la intranet, el teléfono, el correo electrónico, etc. De esta forma, las personas implicadas en los procesos que se desarrollan en el eXcriba no tendrían una alternativa para comunicarse. Estas personas tendrían que realizar acciones extras para establecer la comunicación como la convocatoria de reuniones y el abandono del puesto de trabajo con la necesidad de solicitar información. Esto tributaría a una pérdida de tiempo y a un retraso en las tareas a realizar por parte de estos usuarios.

A partir de la **situación problemática** existente, se plantea el siguiente **problema de investigación**:  
¿Cómo lograr la comunicación interna entre los usuarios que se encuentren autenticados en el GDA eXcriba 2.0?

El **objeto de estudio** se enmarca en la comunicación instantánea para la *web* y el **campo de acción** de la investigación va dirigido específicamente a los clientes de mensajería instantánea. Con el fin de darle solución a la problemática antes mencionada se define como **objetivo general**, implementar un sistema de mensajería instantánea para el GDA eXcriba 2.0 que permita el intercambio de mensajes de texto en tiempo real entre sus usuarios autenticados. Dicho objetivo general se desglosa en los siguientes **objetivos específicos**:

1. Fundamentar los antecedentes y aspectos teóricos referentes a los sistemas de mensajería instantánea.
2. Diseñar un módulo de mensajería instantánea para el GDA eXcriba 2.0.
3. Implementar un módulo de mensajería instantánea para el GDA eXcriba 2.0.
4. Validar el correcto funcionamiento del módulo realizado mediante pruebas de caja blanca y caja negra.

Para organizar el desarrollo del presente trabajo y cumplir con el objetivo propuesto se plantearon las siguientes **tareas**:

1. Identificación de los fundamentos teóricos relacionados a la mensajería instantánea.
2. Caracterización del estado actual de los sistemas de mensajería instantánea en las aplicaciones

web.

3. Validación de las herramientas, lenguajes, tecnologías y metodología a utilizar para el desarrollo del módulo.
4. Identificación de los artefactos que genera la metodología seleccionada.
5. Descripción de los artefactos que genera la metodología seleccionada.
6. Implementación del módulo con las funcionalidades propuestas.
7. Validación del correcto funcionamiento del módulo mediante las pruebas de caja blanca y caja negra.

- **Métodos de investigación científica**

Los métodos utilizados a lo largo de la investigación son los siguientes:

### **Métodos Teóricos**

Los métodos teóricos utilizados en la investigación son:

- ❖ **Analítico sintético:** Permite un análisis mental del objeto o fenómeno en sus múltiples relaciones o componentes facilitando su estudio, así como unir partes previamente analizadas descubriendo características generales y relaciones entre ellas. Este método se utiliza en la investigación con el objetivo de extraer elementos, definiciones y conceptos importantes que tienen que ver con los sistemas de mensajería instantánea.
- ❖ **Histórico lógico:** Analizan la trayectoria del fenómeno e investiga las leyes generales del funcionamiento y desarrollo de los fenómenos. Utilizado en la investigación para dividir la información obtenida, y de esta forma tener un análisis más a fondo de la evolución que los sistemas de mensajería instantánea ofrecen.

### **Métodos Empíricos**

- ❖ **Observación:** Es el instrumento universal del científico, se realiza de forma consciente y orientada

a un objetivo determinado. Usado para percibir directamente la realidad y actualidad que presentan los sistemas de mensajería instantánea en el mundo, se estudió su comportamiento y se recogió la mayor información necesaria que ofrecían.

## **Aportes prácticos esperados del trabajo**

Con la presente investigación se pretende realizar un servicio de mensajería instantánea para el GDA eXcriba. Con su culminación, se esperan obtener los siguientes resultados:

- ✓ Mayor comunicación entre los usuarios que participan en los procesos que se desarrollan en el GDA eXcriba 2.0.
- ✓ Requisitos funcionales y no funcionales del módulo.
- ✓ Artefactos de análisis y diseño del módulo.
- ✓ Propuesta de arquitectura.
- ✓ Módulo de mensajería instantánea desarrollado en el GDA eXcriba 2.0 con sus funcionalidades básicas implementadas.

## **Estructuración del trabajo por capítulos:**

El presente trabajo se estructura en cuatro capítulos como se muestra a continuación:

- **Capítulo I:** Fundamentación teórica.

Se exponen las principales características de las tecnologías y herramientas seleccionadas para la solución de la problemática actual, además de un estudio del estado del arte de los sistemas de mensajería instantánea.

- **Capítulo II:** Características del módulo.

Con el fin de comprender el contexto donde será implantado el sistema se realiza el modelado del negocio. Se identifican los requerimientos funcionales, no funcionales y se validan mediante la matriz de trazabilidad. Se realiza el modelo de casos de uso del sistema, así como la descripción de cada caso de

uso y se capturan los artefactos que propone la metodología de desarrollo seleccionada.

- **Capítulo III:** Análisis y diseño del módulo.

Se hace un análisis de los requisitos capturados, haciendo una estructuración y clarificación de los mismos con el objetivo de llegar a una comprensión más detallada de dichos requisitos. Se modela el diseño logrando una arquitectura estable y sólida, ayudando a crear un plano de lo que se implementará. Finalmente se construye la propuesta de solución realizando el diagrama de despliegue y de componentes.

- **Capítulo IV:** Validación de los resultados del módulo.

El *software* es sometido a un proceso de pruebas para así comprobar su correcto funcionamiento. Se valida la solución del módulo que se propone, además se evalúa la calidad de la especificación de los requisitos y el diseño realizado.

## *Capítulo 1. Fundamentación teórica*

### **Introducción**

En este capítulo se muestran algunos conceptos fundamentales que ofrecen un entendimiento del contexto a desarrollar. Se expone un breve resumen de las principales características de las tecnologías, herramientas, lenguajes y técnicas de desarrollo propuestas para la implementación del módulo de mensajería instantánea. Además, se realiza un estudio de algunos de los sistemas de mensajería instantánea y sistemas de gestión documental.

### **Conceptos fundamentales**

**Mensajería instantánea:** Forma de comunicación en tiempo real basada en texto que entablan dos o más personas involucradas que han aceptado comunicarse entre sí. La mensajería instantánea requiere el uso de un cliente de mensajería instantánea que realiza el servicio y se diferencia del correo electrónico en que las conversaciones se realizan en tiempo real (Castro, Luis, 2013).

**Sistema de mensajería instantánea:** Es un programa o aplicación que se utiliza para enviar mensajes en tiempo real permitiéndole al usuario hacer uso de la mensajería instantánea (Castro, Luis, 2013).

#### **1.1 Estudio de los sistemas de mensajería instantánea**

Actualmente existen varios sistemas de mensajería instantánea en el mundo que son usados en diferentes sistemas operativos. A continuación se expondrán las características de algunos de los sistemas de mensajería instantánea que han sido escogidos para su análisis, tanto los que se utilizan en las aplicaciones *web* como *desktop*<sup>1</sup>.

##### **1.1.1 Pidgin**

El Pidgin anteriormente llamado Gaim, es el cliente universal de mensajería instantánea, multiplataforma, capaz de conectarse a múltiples redes (multiprotocolo) y cuentas (multicuenta) de manera simultánea.

---

<sup>1</sup> Se refiere a las aplicaciones de escritorio

Compatible con otros clientes como el AIM<sup>2</sup>, MSN Messenger, Yahoo y Google Talk por tan solo mencionar algunos (Leo, Junior, 2009). Entre sus características principales se encuentran:

- ✓ Transferencia de archivos o mensajes privados.
- ✓ Dispone de *plug-ins* de seguridad y mejoras.
- ✓ Corrector ortográfico.
- ✓ Conversaciones con pestañas.

Este cliente de mensajería no puede ser incorporado al GDA eXcriba 2.0, pues este tendría que utilizar alguno de los protocolos que Pidgin usa (XMPP/Jabber, SIMPLE, Yahoo!, entre otros), de los cuales algunos son propietarios. Además, Pidgin no ofrece el código fuente para modificarlo y de esta forma permitir la relación con el eXcriba referente a la autenticación de sus usuarios.

### 1.1.2 PSI

PSI es un atractivo cliente de mensajería instantánea multiplataforma, que se basa por completo en el protocolo Jabber (XMPP<sup>3</sup> (XMPP Standards Foundation, 2013)). Sin necesidad de saber qué red usan los usuarios involucrados, a la hora de chatear es una excelente alternativa. El PSI es compatible con ICQ, MSN Messenger, Yahoo y AIM. Dispone de una gama de funcionalidades cubriendo así las necesidades de comunicación (Castagnino, Pablo, 2010), entre las que se encuentran:

- ✓ Mensajes privados.
- ✓ Grupos de chat.
- ✓ Envío de ficheros.
- ✓ Diversos estados de conexión (Castagnino, Pablo, 2010).

Ofrece ciertas características como:

---

<sup>2</sup> America-On-Line Instant Messenger (Cliente de Mensajería Instantánea de America On Line)

<sup>3</sup> Extensible Messaging and Presence Protocol (Protocolo Extensible de Mensajería y Comunicación de Presencia)

- ✓ Altamente personalizable.
- ✓ Soporte para perfiles.
- ✓ Ventanas de chat con solapas (Castagnino, Pablo, 2010).

PSI no ha sido considerado para incorporarse al GDA eXcriba 2.0 pues se basa exclusivamente en un protocolo (XMPP). Este protocolo se utiliza en la UCI mediante la forma usuario (usuario@jabber.uci.cu) y contraseña correspondiente al usuario. El módulo de mensajería propuesto no se rige por este protocolo ya que una de las características principales a considerar es prescindir de los servidores pertenecientes a la red UCI o cualquier red en general. Mediante un servidor de base de datos utilizado en el eXcriba para manejar los datos de los usuarios se estaría solucionando este problema.

### 1.1.3 Empathy

Empathy es un poderoso programa de mensajería instantánea cuyo objetivo principal es la estrecha relación con el escritorio, basado en los clientes de mensajería *Telepathy* y *Nokia's Mission Control* utilizando la interfaz de usuario de *Gossip*. La librería *libempathy-gtk*, el corazón del programa, no es más que un conjunto de *widgets* que pueden ser insertados en cualquier aplicación de GNOME<sup>4</sup> (Castagnino, Pablo, 2010). Sus características más llamativas son:

- ✓ Multiprotocolo, pues posee todos los protocolos soportados por Pidgin.
- ✓ Interfaz de usuario específica para cada protocolo.
- ✓ Reconectado automático.
- ✓ Registro de las conversaciones (Castagnino, Pablo, 2010).

Este cliente no puede ser insertado al GDA eXcriba 2.0 porque utiliza todos los protocolos que soporta Pidgin y posee una interfaz especializada para la mayoría de ellos. Además, permite editar la información de sus contactos, aspecto que el módulo a desarrollar no debe permitir ya que la información de cada usuario del eXcriba es personal y no debe ser cambiada.

---

<sup>4</sup> GNU Network Object Model Environment

### 1.1.4 EBuddy

EBuddy es un cliente de mensajería a través de la *web* que se conecta a las redes más conocidas del mundo (MSN, Yahoo, GTalk, Facebook, ICQ, MySpace y AIM). Al conectarse con sus credenciales en EBuddy una ventana aparte se abrirá, haciendo las veces de la aplicación *web* de mensajería. En la pestaña principal se tendrán los contactos organizados por grupos. Se podrán ver sus avatares, nombres y mensajes de estado, además de poder cambiar los propios (Bueno, Abel, 2013).

En pestañas distintas se tienen las conversaciones de EBuddy que se abran o las que te abran. En ellas se pueden personalizar los textos, compartir enlaces o videos en YouTube<sup>5</sup>. Las ventanas se pueden configurar mediante coloridos temas, aunque la publicidad de los *banners* puede saturar un poco. Las opciones de privacidad y configuración de EBuddy completan un cliente muy versátil y comparable a los de escritorio, pero accesible desde cualquier lugar (Bueno, Abel, 2013). Muestra algunas ventajas como:

- Centraliza todas las redes en una aplicación.
- Aspecto cuidado, similar a cliente de escritorio.
- Permite conversaciones múltiples (Bueno, Abel, 2013).

Este cliente de mensajería no ha sido considerado para ser añadido al GDA eXcriba 2.0 pues se rige por su propio protocolo de mensajería (viajando sobre POP3<sup>6</sup> (Universidad de Zaragoza, 2008)). Esto implicaría que el eXcriba necesariamente tenga que regirse por este protocolo, por lo que no es factible el uso de eBuddy ya que no deben realizarse modificaciones que tengan que ver con el funcionamiento interno del GDA eXcriba.

### 1.1.5 Meebo

Meebo es una aplicación *web* con la que es posible conectar a múltiples cuentas de mensajería instantánea y charlar con tus contactos a través de texto, voz y video, exactamente como en Messenger y otras aplicaciones *desktop*. Meebo es compatible con las redes de Facebook, Windows Live Messenger, AIM y Google Talk, además de otras menos conocidas (Ferri-Benedetti, Fabrizio, 2013).

---

<sup>5</sup> Sitio web en el cuál los usuarios pueden subir y compartir videos

<sup>6</sup> Post Office Protocol (Protocolo de Oficina Postal)

En el escritorio virtual de Meebo, los cuadros de chat y la lista de contactos se pueden mover, minimizar y maximizar como ventanas normales. Por suerte, Meebo te avisará de nuevos mensajes mediante un sonido y el parpadeo del título de la pestaña. El surtido de emotíconos de Meebo es el habitual. Juegos, chat de voz y video y envío de archivos, si bien no funcionan igual de bien que en programas de escritorio, cumplen su cometido. Y para mayor comodidad, Meebo pone a tu disposición su propia cuenta centralizada, opción interesante si quieres conectar desde otros ordenadores sin volver a escribir los datos una y otra vez (Ferri-Benedetti, Fabrizio, 2013). Sus ventajas más significativas son:

- Soporte para muchos protocolos.
- Notificaciones con sonido.
- Soporte para chat con voz y video.
- Guarda los historiales de conversación.
- Interfaz personalizable (Ferri-Benedetti, Fabrizio, 2013).

Meebo no puede ser incorporado al GDA eXcriba 2.0, pues soporta varios protocolos que el eXcriba no utiliza en su entorno de desarrollo. Además, el usuario que interactúe con Meebo debe crearse una cuenta para utilizar sus servicios. Este paso extra es obviado completamente en el módulo a desarrollar, pues el usuario que interactúe con este, solamente tendría que estar autenticado al eXcriba para utilizarlo.

Se concluye que ninguno de los sistemas estudiados puede integrarse al GDA eXcriba 2.0, ya que todos utilizan protocolos que no necesariamente tienen que estar habilitados donde el eXcriba se despliega. Por tanto, es conveniente que el módulo a desarrollar pueda prescindir de los protocolos de mensajería instantánea. Además, estos clientes no ofrecen información de su código fuente, para de esta manera modificarlo e integrarlo con el código del eXcriba en función de la autenticación de sus usuarios.

Por otra parte, estos sistemas brindan algunas características que son consideradas para incorporarlas al módulo, estas son:

- Conversaciones múltiples por pestañas.
- Multiplataforma (accesibles desde varios sistemas operativos).

- Permiten crear grupos de chat.
- Chatear desde el navegador.
- Envío y recibo de mensajes de texto de forma instantánea.

A continuación se realiza un estudio de los sistemas de gestión documental y su relación con la mensajería instantánea.

## 1.2 Estudio de los sistemas de gestión documental

Alrededor del mundo se utilizan los sistemas de gestión documental, cada uno de ellos muestran características particulares que los distinguen a cada uno. Para un desarrollo más detallado de la investigación, es de vital importancia verificar si alguno de estos sistemas se relaciona con la mensajería instantánea.

### 1.2.1 Alfresco

Alfresco es un sistema de administración de contenidos de código abierto, es decir libre, desarrollado en Java<sup>7</sup>, basado en estándares abiertos y de escala empresarial para sistemas operativos Windows, Unix, Solaris y algunas versiones de Linux. Es utilizado como *software* de gestión documental para documentos, páginas *web*, registros, imágenes y desarrollo colaborativo de contenido (Alfresco Software, 2013).

Alfresco incluye un repositorio de contenidos, un *framework*<sup>8</sup> de portal *web* para administrar y usar contenido estándar en portales, una interfaz CIFS<sup>9</sup> que provee compatibilidad de sistemas de archivos en Windows y Unix, un sistema de administración de contenido *web*, capacidad de visualizar aplicaciones *web* y sitios estáticos vía Apache Tomcat, búsquedas vía el motor Lucene<sup>10</sup> y flujo de trabajo en jBPM<sup>11</sup> (Alfresco Software, 2013).

---

<sup>7</sup> Lenguaje de programación

<sup>8</sup> Marco de trabajo

<sup>9</sup> Common Internet File System (Sistema de Archivos de Internet Común)

<sup>10</sup> Interfaz de programación de aplicaciones de código abierto para recuperación de información

<sup>11</sup> Flexible Business Process Management: Plataforma para lenguajes de procesos ejecutables

Para la comunicación, Alfresco ofrece mecanismos de comunicación adecuados para la integración como los servicios web tipo SOAP<sup>12</sup> (W3C, 2010) o REST (Binovo Hit Human Project, S.L., 2013). Estos servicios permiten la interacción entre dispositivos a través del protocolo HTTP. Alfresco no mantiene relación con la mensajería instantánea para la comunicación entre usuarios, pues no contempla un sistema de mensajería instantánea y no utiliza ningún protocolo de mensajería.

### 1.2.2 Knowledge Tree

Sistema de gestión documental basado en entorno *web*. Proporciona un entorno estructurado y seguro, óptimo para la gestión de información y procesos de misión crítica. Permite registrar, compartir y seguir el flujo de los documentos al interior de la organización de manera segura. Al ser un programa libre, permite la adaptación y mejora de sus componentes, así como una sencilla adaptación a la infraestructura actual de cualquier organización, entregando así una solución flexible y de bajo costo (Knowledge Tree, 2013).

Gracias a la organización de los documentos de archivo que posee Knowledge Tree, la gestión de estos por parte de los usuarios se genera de manera colaborativa. Este sistema de gestión documental permite la integración con LDAP<sup>13</sup> (The PHP Group, 2013) para centralizar los niveles de acceso a directorios y tecnologías dentro de la organización pero al igual que Alfresco no presenta desarrollado un módulo o funcionalidad referente a la comunicación interna entre sus usuarios.

### 1.2.3 LetoDMS

Este sistema de gestión documental incluye todas las funcionalidades necesarias para la correcta administración de grandes volúmenes de documentación. Destaca la completa gestión de usuarios y grupos, así como el sistema de gestión de privilegios. El sistema de seguimiento de la documentación y el control de versiones hacen que esta herramienta sea adecuada para cualquier tipo de entorno (LetoDMS, 2009).

Los usuarios que interactúan con este sistema tienen un control sobre los archivos que se gestionan. LetoDMS ofrece varias características que contribuyen al desarrollo colaborativo entre usuarios para la

---

<sup>12</sup> Simple Object Access Protocol

<sup>13</sup> Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios)

gestión de documentos, pero no contempla una que esté relacionada con la comunicación interna para que sus usuarios interactúen dentro del mismo sistema.

### 1.2.4 OpenKM

El OpenKM permite la gestión y almacenamiento de los documentos y contenidos que se vayan generando dentro de las empresas. Entre sus funciones generales, el programa cuenta con indexación<sup>14</sup> de documentos de texto, Office, OpenOffice, PDF<sup>15</sup>, bases de datos, etc. El motor de búsqueda indexa los documentos para posteriormente poder realizar búsquedas concretas sobre los contenidos guardados dentro del sistema. OpenKM integra en una sola aplicación fácil de usar todas las funcionalidades para colaborar, gestionar y buscar documentos (OpenKM, 2011).

Las actividades que giran en torno a la gestión de la información ayudan a conectar a las personas que interactúan con el OpenKM. Actividades como comentarios y foros sobre un documento determinado son las que permiten este tipo de conexión entre usuarios, pero al igual que los demás sistemas de gestión documental estudiados, el OpenKM no ofrece ninguna característica que tribute a la comunicación interna de sus usuarios.

Se puede afirmar que todos los sistemas de gestión documental estudiados poseen características que se relacionan con la gestión de la información y mediante esta forma existe la comunicación entre sus usuarios, pero estos no tienen relación con la mensajería instantánea.

Por todo lo antes estudiado, se hace necesario la implementación de un módulo de mensajería instantánea para el GDA eXcriba 2.0 que se ajuste al sistema y presente características particulares como:

- A medida que los usuarios se autentiquen en el GDA eXcriba 2.0, automáticamente estarían autenticados en el módulo.
- Los usuarios que interactuarán con el módulo son específicamente los que pertenecen al eXcriba, lo que significa que no es necesario agregarlos mediante solicitudes de amistad.

---

<sup>14</sup> Registro ordenado de información

<sup>15</sup> Portable Document Format (Formato de Documento Portable)

- Mediante un servidor de base de datos interno que sea capaz de manejar los datos de todos los usuarios que interactúan con el módulo; solo sería necesario utilizar la conexión a este servidor para establecer la comunicación. Esta conexión no necesitaría ningún protocolo de mensajería ni dominio de red.
- Los usuarios que utilizan el módulo no podrán ser eliminados mediante este, pues estos son los que pertenecen e interactúan con el eXcriba.

A continuación, se hace un breve análisis de las herramientas, metodologías, lenguajes y *framework* a utilizar durante del desarrollo del trabajo de diploma.

### 1.3 Metodología de desarrollo RUP con nivel 2 de CMMI

La realización y desarrollo de un *software* no es trabajo fácil. La prueba de ello es que existen numerosas propuestas de metodologías que sirven de guía para el proceso de desarrollo de un *software*. Ejemplo de ello, son las metodologías de desarrollo de un *software*. A continuación se exponen algunas de las características de la metodología de desarrollo a utilizar en el presente trabajo de diploma.

El Proceso Unificado Racional (por sus siglas en inglés *RUP*) es más que un simple conjunto de actividades necesarias para transformar los requerimientos del usuario en el sistema de *software*, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de *software*. Este define claramente quién, cómo, cuándo y qué debe hacerse en un proyecto. El proceso unificado combinado con UML como lenguaje de modelado conforma la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientado a objetos (Jacobson, y otros, 2004).

RUP mantiene un desarrollo basado en componentes, en el que permite que el sistema se vaya creando a medida que se desarrollan los mismos. Es guiado por casos de uso, iterativo e incremental y centrado en la arquitectura. Divide el proceso de desarrollo del *software* en cuatro fases: inicio, elaboración, construcción y transición, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto (Jacobson, y otros, 2004).

Conociendo las particularidades de la solución a diseñar y las capacidades de la metodología, se decide usar RUP con Nivel 2 de CMMI, pues en el proyecto eXcriba se utiliza dicha metodología. Esta permite una buena documentación del producto y una reducción de los riesgos que puedan existir en el transcurso

del desarrollo del mismo.

### 1.4 Lenguajes, herramientas y tecnologías a utilizar

El Departamento de Gestión Documental de la Facultad 1 hace uso de la plataforma de código abierto con diferentes sistemas operativos, rigiéndose por algunos de los principios y funcionalidades del centro. A continuación se hace referencia a un grupo de lenguajes, herramientas y tecnologías que se usan en el presente trabajo de diploma.

#### 1.4.1 Lenguaje de modelado UML

El Lenguaje Unificado de Modelado (por sus siglas en inglés *UML*) es el lenguaje de modelado más conocido y utilizado en la actualidad para el desarrollo de *software*. UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas de *software* como para la arquitectura de *hardware* donde se ejecuten. Permite visualizar, construir y documentar los elementos que forman un sistema de *software* orientado a objetos. Este lenguaje indica cómo escribir y leer los modelos, no dice cómo crearlos, pues eso es objetivo de la metodología que se utilice (Larman, 1999).

Se seleccionó UML como lenguaje para modelar los artefactos seleccionados durante el proceso de desarrollo del *software*. Además, este lenguaje se emplea asociado a la metodología de desarrollo RUP.

#### 1.4.2 pgAdmin 3

PgAdmin es una herramienta de código abierto para la administración de bases de datos PostgreSQL. Incluye interfaz administrativa gráfica y consultas SQL. Se diseñó para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.

Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Windows, Linux, FreeBSD, Mac OSX y Solaris. Además de lo todo lo anteriormente mencionado, se utiliza pgAdmin en la investigación porque soporta todas las características de PostgreSQL y hace simple la administración (Reingart, Mariano, 2010).

### 1.4.3 Visual Paradigm

Visual Paradigm para UML es una herramienta líder de las llamadas herramientas CASE<sup>16</sup> que provee el modelado de procesos de negocio, además de un generador de mapeo de objetos relacionales para los lenguajes de programación Java, .NET y PHP. Esta herramienta mantiene el ciclo de vida completo del desarrollo de *software* que son análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Posibilita la importación y exportación de ficheros XML. Está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas y analistas de negocios (Visual Paradigm, 2013).

Para la construcción de los diagramas que se generen se utilizará Visual Paradigm en su versión 8.0, ya que es una herramienta multiplataforma para modelado UML.

### 1.4.4 Lenguajes de programación

#### JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas *web* dinámicas. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, lo que lo hace una opción elegible y fácil de usar (Eguíluz Pérez, 2009).

Se escoge este lenguaje de programación que será usado en su versión 2.2 para la implementación de las interfaces de usuario. Además, este lenguaje es el que se propuso para el desarrollo del eXcriba.

#### PHP

PHP<sup>17</sup> es un lenguaje *open source* (*código abierto*) interpretado de alto nivel empapado en páginas HTML<sup>18</sup> y ejecutado en el servidor. PHP es un lenguaje de programación interpretado, diseñado

---

<sup>16</sup> Computer Aided Software Engineering

<sup>17</sup> Hypertext Pre-processor (Pre-procesador de Hipertexto)

originalmente para la creación de páginas *web* dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica. Se utilizan para complementar el HTML (lenguaje para crear páginas *web*) para hacer una *web* más interactiva y funcional. Gracias a estos lenguajes se pueden desarrollar cualquier tipo de servicios a través de su sitio *web*. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual (Saether Bakken, 2002).

Para el trabajo de diploma se decide usar PHP con su versión 5.3 para la implementación del módulo, ya que es un lenguaje “código abierto”, multiplataforma y cuenta con un gran soporte para las bases de datos.

### 1.4.5 Entorno de desarrollo Zend Studio

Zend Studio es un completo IDE<sup>19</sup> para el lenguaje de programación PHP. Está escrito en Java y disponible para las plataformas Microsoft Windows, Mac OSX y GNU/Linux. Dentro de sus funcionalidades posee editor, análisis, depuración, optimizadores de código y herramientas de base de datos. Permite agilizar el desarrollo *web* y permite simplificar proyectos complejos (Zend Technologies Ltd, 2013). Entre sus características más significativas se encuentran:

- Excelente completamiento de código.
- Coloreado en la sintaxis del código.
- Administración avanzada de proyectos.
- Soporte para múltiples lenguajes.
- Posee un manual de PHP.

---

<sup>18</sup> HyperText Markup Language (Lenguaje de Marcado Hipertextual)

<sup>19</sup> Integrated Development Environment (Entorno de Desarrollo Integrado)

- Soporte para servicios *web*, PHP 4, PHP 5 y SQL<sup>20</sup> (Zend Technologies Ltd, 2013).

Para desarrollar el módulo propuesto se escogió Zend Studio en su versión 8.0 como entorno de desarrollo integrado debido a las características mencionadas anteriormente.

### 1.4.6 Framework JQuery

JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con las páginas HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX<sup>21</sup> a las páginas *web*. Además, permite el intercambio asíncrono de datos entre cliente y servidor de manera sencilla. Esta biblioteca incluye los efectos y eventos de AJAX, las funcionalidades de DOM<sup>22</sup>, y se puede manipular el estilo CSS<sup>23</sup> de las páginas o elementos HTML (Álvarez, Miguel Ángel, 2013).

El eXcriba usa este *framework* desde sus inicios y los desarrolladores del proyecto tienen dominio del mismo desde la versión inicial. La versión a utilizar de este *framework* para el desarrollo de las interfaces del lado del cliente es la 1.3.2 ya que es el que se propone en la arquitectura del eXcriba.

### 1.4.7 Sistema Gestor de Base de Datos PostgreSQL

Originalmente nombrado Postgres, PostgreSQL fue creado en la Universidad de California en Berkeley, por un profesor de Ciencia de la Computación llamado Michael Stonebaker. PostgreSQL es un potente gestor de base de datos relacional libre (liberado bajo licencia BSD<sup>24</sup>). Cuenta con más de 15 años de desarrollo activo y arquitectura probada que ha ganado mucha reputación dada su confidencialidad e integridad en los datos (PostgreSQL, 2013).

---

<sup>20</sup> Structured Query Language (Lenguaje de Consulta Estructurado)

<sup>21</sup> Asynchronous JavaScript and XML (JavaScript Asíncrono y XML)

<sup>22</sup> Document Object Model (Modelo de Objetos del Documento)

<sup>23</sup> Cascading Style Sheets. Lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos

<sup>24</sup> Berkeley Software Distribution, licencia de *software* libre que permite el uso del código fuente en el *software* privativo.

Para el trabajo con la base de datos se usará la versión 8.4 de PostgreSQL, el cual tiene una excelente documentación y está bien organizada, además, cuenta con una comunidad de usuarios y desarrolladores a los que acudir en caso de presentar algún problema.

### **1.5 Conclusiones del capítulo**

Luego de haber planteado las principales características de las herramientas y tecnologías que serán utilizadas en el desarrollo de esta investigación, además de la realización de un estudio de los diferentes sistemas de mensajerías instantáneas y de los sistemas de gestión documental; es posible afirmar que:

- No es conveniente incorporar al GDA eXcriba 2.0 ninguno de los clientes de mensajería estudiados.
- Ninguno de los sistemas de gestión documental estudiados tiene incorporado ningún cliente de mensajería instantánea.
- El módulo propuesto será independiente de cualquier protocolo de mensajería instantánea.
- Se implementará un módulo de mensajería instantánea en el GDA eXcriba 2.0 utilizando las herramientas y metodologías propuestas en el capítulo.

### *Capítulo 2. Características del módulo*

#### **Introducción**

En el presente capítulo se elabora una propuesta del módulo de mensajería instantánea a implementar, se exponen las características del sistema para un mejor entendimiento, así como la descripción y modelación de los procesos, detallando las historias de usuarios que proporcionan una mejor visión sobre lo que se desea. Se puntualizan los requisitos funcionales y no funcionales que debe cumplir el sistema así como las funcionalidades que este presenta.

#### **2.1 Propuesta de solución**

Para darle solución al problema de investigación, se propone desarrollar un módulo para el GDA eXcriba 2.0 que cuente con las funcionalidades siguientes:

- Gestionar grupos de usuarios.
- Enviar mensajes de texto en tiempo real a los usuarios que estén autenticados y hayan accedido al módulo.
- Recibir mensajes de texto en tiempo real.
- Cambiar el nombre de los usuarios conectados.
- Abrir y cerrar conversaciones referentes a los usuarios conectados.

#### **2.2 Modelo de dominio**

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos de dominio o clases (para emplear una terminología más precisa) pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio (Jacobson, y otros, 2004). Las clases del dominio aparecen de tres formas típicas:

- Objetos del negocio que representan cosas que se manipulan en el negocio (pedidos, cuentas o contratos).
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento (aviación enemiga, misiles o trayectorias).
- Sucesos que ocurrirán o han ocurrido (llegada de un avión o su salida).

El modelo de dominio se describe mediante diagramas UML (especialmente mediante diagrama de clases). A continuación se muestra el modelo de dominio y la descripción de sus entidades:

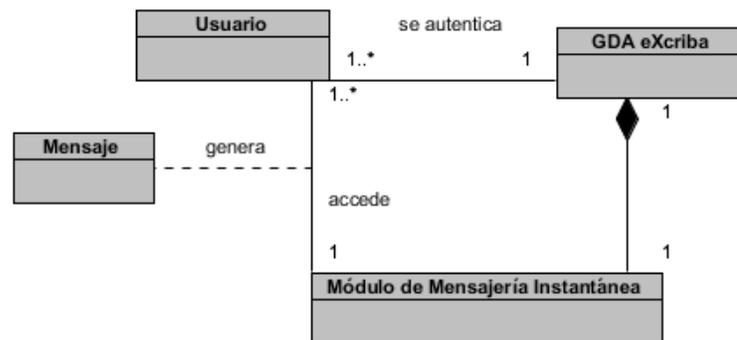


Figura 1: Modelo de dominio

### Descripción de las entidades

*Usuario*: Es la persona que interactúa con el módulo, luego de autenticarse al GDA eXcriba 2.0.

*GDA eXcriba*: Es el sistema donde será desarrollado el módulo, donde el usuario se debe autenticar.

*Módulo de mensajería instantánea*: Es el sistema con el cual el usuario va a interactuar luego de autenticarse al GDA eXcriba 2.0.

*Mensaje*: Es lo que envía y recibe el usuario a medida que interactúa con el módulo para comunicarse con los demás usuarios conectados.

## 2.3 Especificación de los requisitos de *software*

De acuerdo con lo planteado por el estándar 1233 de la IEEE: Guía para el Desarrollo de Especificaciones

de Requerimientos de Sistemas, un requisito tiene la siguiente definición:

- Condición o capacidad que necesita un usuario para resolver un problema, y de esta manera, lograr un objetivo.
- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

Se puede llegar a la conclusión de que los requisitos de *software* son características y funcionalidades que un sistema debe cumplir. Estos están enfocados en lo que debe hacer el *software* y los usuarios que interactúan con el mismo. Los requisitos son clasificados en funcionales y no funcionales.

### 2.3.1 Técnicas para la captura de los requisitos

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de *software* extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo. Por su complejidad, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa (Escalona, y otros, 2002).

Las técnicas a usar para la captura de requisitos en el proceso de desarrollo del módulo son:

- **Tormenta de ideas:** Técnica de reuniones en grupo, cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas (Escalona, y otros, 2002). Esta técnica se llevó a cabo mediante un encuentro con algunos funcionarios del eXcriba. Cada uno de ellos aportó sus ideas para la realización del módulo y a partir de ahí se definieron los primeros requisitos.
- **Análisis de otros sistemas:** Mediante esta técnica se pudieron estudiar sistemas de mensajería instantánea parecidos al módulo que se desea realizar. Ya estudiados dichos sistemas, su funcionamiento, sus características, ventajas y desventajas, se hace más simple la identificación de los requisitos que necesita el sistema para su implementación (Escalona, y otros, 2002).

### 2.3.2 Requerimientos funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos dependen del tipo de *software* que se desarrolle, de los posibles usuarios del *software* y del enfoque general tomado por la organización al redactar requerimientos. Cuando se expresan como requerimientos del usuario, habitualmente se describen de una forma bastante abstracta. Sin embargo, los requerimientos funcionales del sistema, describen con detalle la función de este, sus entradas y salidas, excepciones, etcétera (Sommerville, 2005).

**RF.01- Mostrar usuarios conectados:** El módulo debe permitirle al usuario mostrar una lista con el nombre completo de los usuarios que estén autenticados al GDA eXcriba 2.0 y conectados al módulo en ese momento.

**RF.02- Gestionar grupo de usuarios:** El módulo debe permitirle al usuario crear grupos de usuarios, agregar usuarios a un grupo creado y eliminar un grupo de usuarios.

- **RF.02.1- Crear grupo de usuarios:** Crear grupos de usuarios con un nombre dado.
- **RF.02.2- Agregar usuarios a un grupo:** El módulo debe permitirle al usuario agregar los usuarios conectados a un grupo de usuarios creado previamente.
- **RF.02.3- Eliminar grupo de usuarios:** El módulo debe permitirle al usuario borrar un grupo de usuarios creado previamente.

**RF.03- Cambiar nombre de los usuarios:** El módulo debe permitirle al usuario cambiar el nombre de cualquier usuario que esté utilizando el módulo en ese momento.

**RF.04- Enviar mensaje:** El módulo debe permitirle al usuario enviar mensajes de texto en tiempo real a uno o varios usuarios que estén utilizando el módulo en ese momento.

**RF.05- Recibir mensaje:** El módulo debe permitirle al usuario recibir mensajes de texto en tiempo real de uno o varios usuarios que estén utilizando el módulo en ese momento.

**RF.06- Mostrar conversaciones:** El módulo debe mostrar las conversaciones que el usuario realice.

**RF.07- Cerrar conversación:** El módulo debe permitirle al usuario cerrar las conversaciones activas.

### 2.3.3 Requerimientos no funcionales

Los requerimientos no funcionales, como su nombre lo dice, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema (Sommerville, 2005).

#### Usabilidad

**RNF.01-** Tipo de aplicación informática: aplicación *web*.

**RNF.02-** Finalidad: El objetivo del módulo es la comunicación entre los usuarios que lo utilicen.

#### Idioma

**RNF.03-** Utilizar el idioma español para los mensajes y textos de la interfaz.

#### Portabilidad

**RNF.04-** Se podrá utilizar la aplicación en los sistemas operativos Linux y Windows.

#### Soporte

**RNF.05-** La estación de trabajo del cliente debe tener instalado el navegador Mozilla Firefox 6.0 o superior.

#### Legales

**RNF.06-** Las herramientas seleccionadas para el desarrollo del producto están respaldadas por licencias libres, bajo las condiciones de *software* libre.

#### Restricciones de diseño

**RNF.07-** Utilizar JQuery 1.3.2 como marco de trabajo.

**RNF.08-** Implementar el módulo en el lenguaje de programación PHP 5.3.X.

**RNF.09-** Utilizar el servidor web Apache 2.2.X.

**RNF.10-** Para el correcto funcionamiento del módulo utilizar el servidor de base de datos PostgreSQL 8.4.

### **Capacidad**

**RNF.11-** Soporte de múltiples conexiones al módulo de manera simultánea.

### **Hardware**

**RNF.12-** El hardware donde se instalará el sistema debe tener al menos una interfaz de red con la velocidad de transferencia igual o superior a los 100 megabytes por segundo.

### **Confiabilidad**

**RNF.13-** La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en la base de datos y de la información introducida por los usuarios del sistema.

## **2.3.4 Técnicas de validación de requisitos**

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos (Escalona, y otros, 2002).

Para realizar la validación de requisitos en la presente solución se utilizaron diferentes métodos, entre los que se encuentran:

- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos (Escalona, y otros, 2002). Mediante la matriz de trazabilidad se comprobó la relación que existía entre los casos de uso del

sistema y los requisitos funcionales, comprobando su correspondencia se garantiza que los requisitos se reflejen en los casos de uso aplicados en la solución logrando cumplir con los objetivos trazados.

- **Prototipos:** Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final (Escalona, y otros, 2002). Con esta técnica de validación se reunieron varias ideas de cómo podría llegar a quedar la interfaz del módulo. Finalmente se llegó a un acuerdo de cómo llegaría a ser la interfaz final, y de esta manera, cómo sería finalmente el módulo. Vale destacar que lo pactado puede estar sujeto a cambios a medida que se vaya trabajando en el diseño de la aplicación en general.

### 2.4 Captura de requisitos como casos de uso

Aunque inicialmente se desarrollaron como técnica para la definición de requisitos, algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema. Un actor puede participar en varios casos de usos y un caso de uso puede interactuar con varios actores (Escalona, y otros, 2002).

La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades.

#### 2.4.1 Definición de los actores

En el sistema interactúa el usuario, ya sea administrador, analista, desarrollador o cualquier rol que se le haya asignado en el proyecto, siempre y cuando este usuario pertenezca al GDA eXcriba 2.0.

Tabla 1: Definición de los actores del módulo

Actores	Descripción
Usuario	Persona que interactúa directamente con el módulo luego de autenticarse al GDA eXcriba 2.0. Puede realizar diferentes funcionalidades como abrir y cerrar conversaciones, enviar y recibir mensajes de texto, cambiar el nombre de los usuarios y crear grupos de usuarios con un nombre dado.

### 2.4.2 Diagrama de casos de uso

Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre éstos y los casos de uso. Estos últimos se muestran en óvalos y los actores son figuras estilizadas. Hay líneas de comunicaciones entre los casos y los actores; las flechas indican el flujo de la información o el estímulo. El diagrama tiene por objeto ofrecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan (Larman, 1999).

En la figura 2 se muestra el diagrama de casos de uso del sistema, el cual ofrece claramente las acciones que puede realizar el usuario cuando interactúa con el módulo.

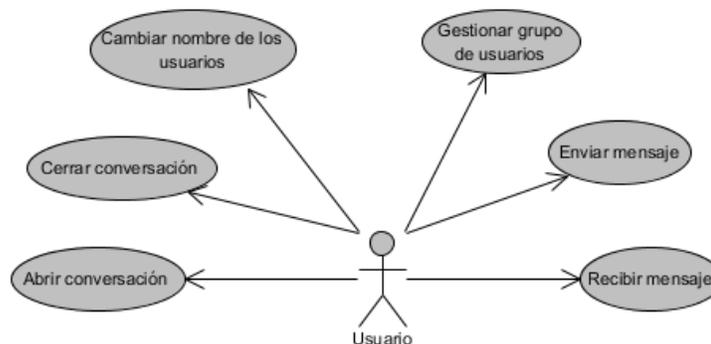


Figura 2: Diagrama de casos de uso

### 2.4.3 Validación de requerimientos mediante la matriz de trazabilidad

Mediante la matriz de trazabilidad se muestra la correspondencia de los requisitos funcionales con los casos de uso identificados. A continuación se menciona el nombre de cada caso de uso para un posterior entendimiento de dicha matriz:

CU.01- Abrir conversación

CU.02- Cerrar conversación

CU.03- Gestionar grupo de usuarios

CU.04- Cambiar nombre de los usuarios

CU.05- Enviar mensaje

CU.06- Recibir mensaje

*Tabla 2: Matriz de trazabilidad*

	CU.01	CU.02	CU.03	CU.04	CU.05	CU.06
RF.01	X		X	X	X	
RF.02			X			
RF.03				X		
RF.04	X				X	
RF.05						X
RF.06	X	X			X	
RF.07		X				

Como se muestra en la tabla anterior, cada requisito identificado ha sido cubierto por al menos un caso de uso del sistema, lo que significa que la especificación de casos de uso propuesta satisface todas las necesidades. La descripción de los casos de uso se pueden consultar en el anexo 1.

### 2.5 Conclusiones del capítulo

Con la propuesta de solución del módulo de mensajería instantánea para el GDA eXcriba 2.0 se tiene una mayor comprensión y entendimiento de la solución a desarrollar antes de su implementación, evitando que se desarrollen innecesarias funcionalidades. La descripción de los requisitos funcionales y no funcionales permitió determinar las principales funcionalidades y condiciones que brindará el módulo, dándole cumplimiento a los objetivos propuestos. Con el desarrollo de este capítulo se han creado las bases para un posterior diseño del módulo, quedando claro y entendible lo que se quiere y cómo se quiere realizar.

### *Capítulo 3. Análisis y diseño del módulo*

#### **Introducción**

En el presente capítulo se abordará el flujo de trabajo de análisis y diseño. Se expondrá a través de un grupo de artefactos cómo será llevada la solución del módulo que se propone hasta el diseño. Para su modelado se utilizan los diagramas de clases del análisis, de interacción, el diseño de clases y el esquema de la base de datos que se necesita para el almacenamiento de la información a persistir. Por último, se expone la arquitectura base del GDA eXcriba 2.0, ya que es la utilizada para desarrollar la propuesta de solución planteada en la investigación.

#### **3.1 Análisis y diseño**

Este flujo de trabajo se acerca más al problema a resolver, debido a que las actividades contempladas en el análisis representan una vista interna del sistema en el que, usando el lenguaje de los desarrolladores se refinan los requisitos y se estructuran en base a clases y paquetes. Este proceso continúa en el diseño hasta obtener los objetos que interactúan para cumplir los requisitos funcionales y no funcionales obtenidos. Es importante destacar que aunque RUP contempla análisis y diseño en una misma disciplina por estar muy relacionadas, son actividades diferentes con artefactos diferentes (Jacobson, y otros, 2004). Los objetivos del flujo de trabajo análisis y diseño son:

- Transformar los requerimientos en un diseño de cómo va a ser implementado el sistema.
- Evolucionar hacia una arquitectura del *software* robusta.
- Adaptar el diseño para que coincida con el ambiente de implementación, diseñando el sistema con un enfoque hacia el rendimiento (Jacobson, y otros, 2004).

#### **3.2 Modelo de análisis**

El modelo del análisis es un artefacto usualmente utilizado para entender mejor los requisitos y realizar mejor el diseño. Es típicamente un artefacto temporal, que brinda una aproximación al diseño y se realiza

normalmente cuando no se tiene una idea clara de los procesos y requerimientos. Es decir, se realiza el análisis para lograr un mejor acercamiento a lo que se desea construir, para sentar las bases del diseño (Jacobson, y otros, 2004).

El objetivo del modelo de análisis es comprender perfectamente los requisitos del *software* y no precisar cómo se implementará la solución, en él se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas y con esta información se construye el diagrama de clases del análisis (Jacobson, y otros, 2004).

### 3.2.1 Diagrama de clases del análisis

El diagrama de clases del análisis constituye una vista estática de las clases que conforman el modelo del análisis y las asociaciones entre las mismas. Es una vista de la futura composición de clases de *software* (Jacobson, y otros, 2004). A continuación se especifica el diagrama de clases del análisis para el caso de uso 'Enviar mensaje', los demás diagramas de clases del análisis para los casos de uso que restan, están en los anexos, específicamente en el anexo 2.

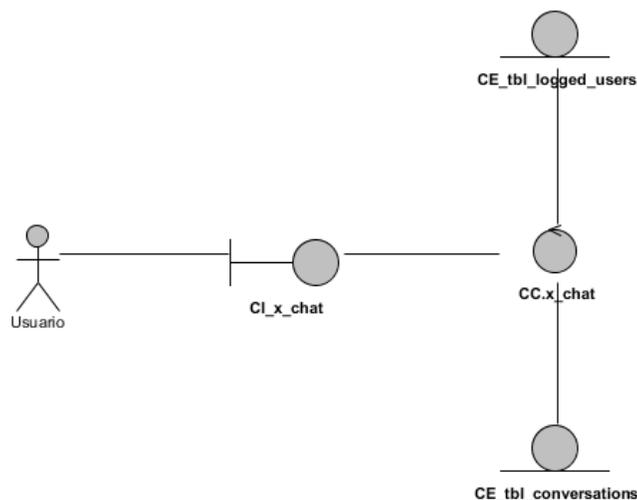


Figura 3: Diagrama de clases del análisis para el caso de uso "Enviar mensaje"

### 3.2.2 Diagrama de interacción

Un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias (y las

clases) del modelo de estas. El punto de partida de las interacciones es el cumplimiento de las condiciones de los contratos de operación (Larman, 1999).

El UML define dos tipos de estos diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje:

- Los diagramas de colaboración describen las interacciones entre los objetos en un formato de grafo o red. Presentan una excepcional expresividad y una capacidad de comunicar más información contextual. Estos nos permiten decir más en un espacio que los diagramas de secuencia, como por ejemplo, el tipo de visibilidad entre los objetos (Larman, 1999).
- El diagrama de secuencia de un sistema muestra gráficamente los eventos que fluyen de los actores del sistema. La creación de estos diagramas forman parte de la investigación para conocer el sistema en sí (Larman, 1999).

Para la presente investigación, se utilizó el diagrama de colaboración correspondiente a los casos de uso definidos. En la figura siguiente se muestra el diagrama de colaboración para el caso de uso “Enviar mensaje”, los restantes se pueden consultar en el anexo 3.

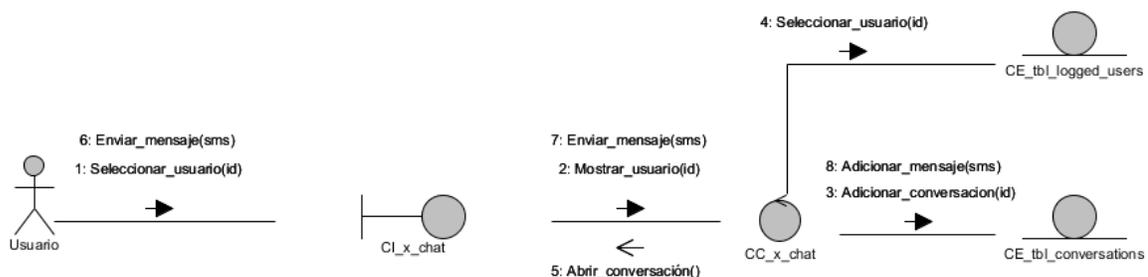


Figura 4: Diagrama de colaboración para el caso de uso "Enviar mensaje"

### 3.3 Modelo de diseño

Los modelos de diseño muestran los objetos o clases en un sistema y, donde sea apropiado, los diferentes tipos de relaciones entre estas entidades. Los modelos de diseño son esencialmente el diseño mismo. Son el puente entre los requerimientos y la implementación del sistema (Jacobson, y otros, 2004).

Existen dos tipos de modelos de diseño para describir un diseño orientado a objetos:

- Modelos estáticos que describen la estructura estática del sistema en términos de las clases del sistema y sus relaciones. Las relaciones importantes que se documentan en esta etapa son de generalización, utiliza/utilizado-por y de composición (Jacobson, y otros, 2004).
- Modelos dinámicos que describen la estructura dinámica del sistema y que muestran las interacciones entre los objetos del sistema (no entre las clases). Las interacciones que se documentan incluyen la secuencia de servicios solicitados por los objetos y la forma en que el estado del sistema se relaciona con estas interacciones de objetos (Jacobson, y otros, 2004).

### 3.3.1 Arquitectura definida para el sistema

La arquitectura de *software* es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema (Billy Reynoso, 2004).

El objetivo principal de la arquitectura del *software* es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto. Para conseguirlo, la arquitectura del *software* construye abstracciones, materializándolas en forma de diagramas comentados (Billy Reynoso, 2004).

La arquitectura basada en capas es un miembro de la familia de estilos de llamada y retorno, esta arquitectura se define como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. En la arquitectura multicapa, las capas inferiores brindan servicios a las capas superiores (independientemente del nivel en que se encuentren). En esta arquitectura una capa solamente debe utilizar lo que la interfaz de la o las capas inferiores le brindan, de este modo se pueden intercambiar las capas respetando la interfaz, que viene a ser como un “contrato entre capas” (Larman, 1999).

Para el diseño del módulo de mensajería instantánea se utilizó el patrón arquitectónico n-capas o basada en capas. Esta arquitectura se escogió por ser la utilizada en el GDA eXcriba 2.0, además de que ofrece

múltiples ventajas como son:

- Se puede entender una sola capa como un todo sin depender de las demás.
- Las capas se pueden sustituir con implementaciones alternativas de los mismos servicios básicos.
- Se minimizan dependencias entre capas.
- Posibilidad de estandarización de servicios.
- Luego de tener una capa construida, puede ser usada por otros servicios de mayor nivel (Larman, 1999).

### 3.3.2 Análisis de la arquitectura del GDA eXcriba 2.0

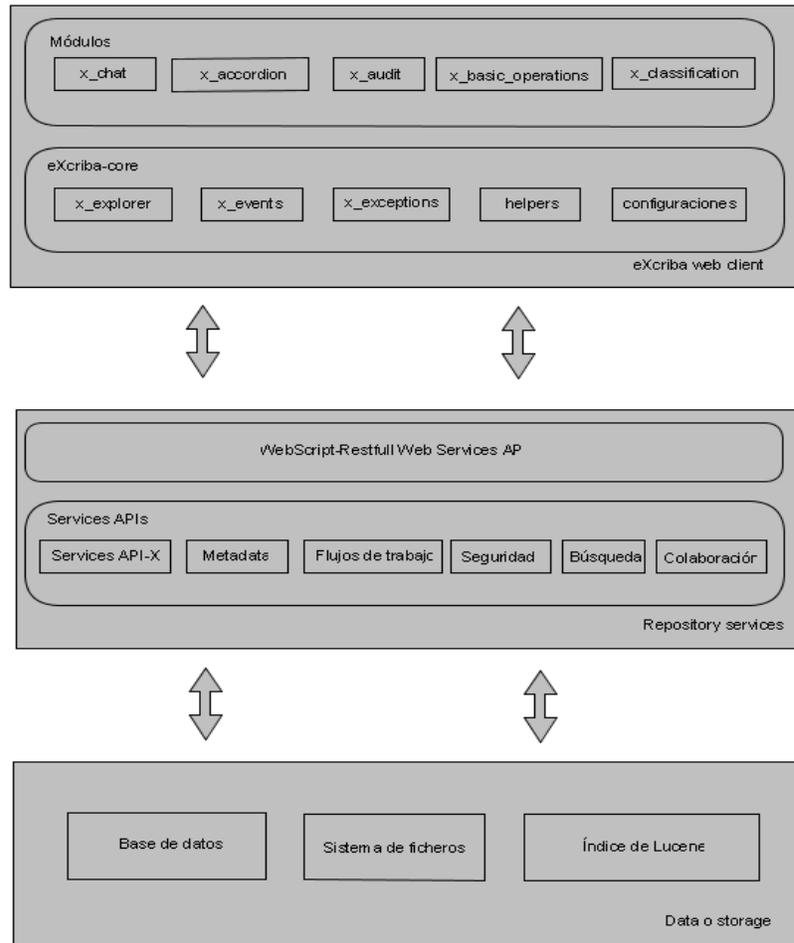


Figura 5: Vista de la arquitectura del eXcriba

Como se puede apreciar en la figura anterior, el GDA eXcriba 2.0 se compone de 3 capas arquitectónicas fundamentales (*eXcriba web client*, *repository services* y *data storage*) que interactúan entre sí durante la realización de cada una de las funcionalidades que brinda el sistema.

La capa superior cliente *web* de eXcriba (en inglés *eXcriba web client*), se compone de un conjunto de módulos que dan soporte a las diferentes operaciones o funcionalidades con las que interactúan los usuarios del sistema. Su principal componente es el módulo eXcriba-core, este brinda un conjunto de

servicios, interfaces, configuraciones y funcionalidades genéricas de las cuales dependen (o pueden ser usadas) por el resto de los módulos que conforman la aplicación *web*.

Las otras dos capas, forman parte del núcleo del ECM Alfresco. La capa de servicios de acceso al repositorio (en inglés *repository services*), a través del *framework WebScript* provee un API<sup>25</sup> completa de objetos de repositorio y servicios basados en el estilo arquitectónico REST. Estos servicios juegan un papel fundamental en la incorporación de las funcionalidades al sistema al tiempo que garantizan que desde el cliente *web* se pueda acceder a los datos que se almacenan en el repositorio de contenido.

La capa de almacenamiento de datos (en inglés *data storage*), la más baja de la arquitectura, constituye precisamente los cimientos sobre los cuales se construye o desarrolla el resto de la aplicación, esta capa se compone de la base de datos (donde se almacena de manera estructurada toda la información referente a los documentos en forma de metadatos, a los usuarios y demás objetos de negocio) y un sistema de ficheros que se compone de documentos que persisten en formato binario.

### Relación entre los componentes de la arquitectura

Como se observa en la figura anterior, al implementar un nuevo módulo, en este caso *x\_chat*, este módulo pasa a formar parte del subsistema “modules” dentro de la capa *eXcriba web client*. Generalmente, estos módulos hacen uso de algunas de las funcionalidades o configuraciones que ofrece el núcleo *eXcriba-core*. Para acceder a los datos del repositorio se hace uso de los servicios que se encuentran disponibles en el subsistema “services” dentro de la capa *Repository services*, servicios que en la mayoría de los casos hacen uso de las APIs que provee el ECM Alfresco para el desarrollo de determinadas funcionalidades.

#### 3.3.3 Arquitectura del módulo

Para la implementación del módulo, se mantiene la misma arquitectura propuesta por el *eXcriba*, siendo la arquitectura en capas que ayuda en la organización del sistema. Las capas mostradas son descritas de esta forma:

---

<sup>25</sup> Application Programming Interface (Interfaz de Programación de Aplicaciones)

- **Presentación:** Ventanas, reportes, etcétera (Larman, 1999). Esta capa muestra las interfaces de usuario, lo que el usuario percibe de la aplicación. Contiene todo lo relativo a la presentación hacia el usuario y toda la interacción con el mismo a través del teclado y el ratón.
- **Lógica de aplicaciones:** Tareas y reglas que rigen el proceso (Larman, 1999). En esta capa se muestran los algoritmos, procesos y flujos de trabajo de la aplicación, es la esencia de la aplicación en sí. Se realizan los procesos de negocio que han sido previamente implementados.
- **Almacenamiento:** Mecanismo de almacenamiento persistente (Larman, 1999). Esta capa contiene los datos que se manejan en la aplicación.

### 3.3.4 Patrones de diseño

El patrón de diseño es el tercer nivel de abstracción de la arquitectura de *software*, cuya finalidad es la de precisar en detalle los subsistemas y componentes de la aplicación. Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptado para resolver un problema de diseño general en un contexto particular (Bahit, 2011).

Los patrones GRASP<sup>26</sup> constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos. Estos son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades (Larman, 1999). De estos patrones de asignación de responsabilidades se proponen utilizar en el diseño:

- **El patrón Experto:** ¿Quién asumirá la responsabilidad en el caso general? Este patrón mantiene el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Esto conlleva a un bajo acoplamiento, lo que da lugar a sistemas más robustos y fáciles de mantener (Larman, 1999). Se utiliza este patrón en el módulo cuando se invoca algún método de una clase, por ejemplo, la clase X\_chat tiene la información necesaria para hacer la llamada a los servicios “actualizar mensaje” y “actualizar usuarios logueados”.
- **El patrón Controlador:** ¿Quién administra un evento del sistema? Un controlador es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema, define el método para la operación del sistema (Larman, 1999). En el módulo se observa la

---

<sup>26</sup> General Responsibility Assignment Software Patterns (Patrones de Software para la Asignación General de Responsabilidad)

utilización de este patrón cuando se realiza una petición de la vista al modelo, o sea, cuando una clase que forma parte de la capa de presentación envía los datos para la clase controladora donde se realiza el procesamiento de la operación. Por ejemplo, la clase `X_chat_lib_loader` envía los datos para la clase controladora `X_chat` donde se realiza el procesamiento de la operación.

- **Alta Cohesión:** ¿Cómo mantener controlable la complejidad? Asignar las responsabilidades de modo que se mantenga una alta cohesión (Larman, 1999). Se muestra la utilización de este patrón en el módulo a la hora de no sobrecargar las clases con métodos innecesarios, lo que quiere decir que no se realizan métodos que no tengan relación con la clase correspondiente.

También existen los patrones `GoF`<sup>27</sup> que son una herramienta fundamental para cualquier programador. Estos patrones de diseño son una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular (Eckel, 2004). A continuación se evidencia como estos patrones son utilizados en la construcción del módulo:

- **Patrones creacionales:** Dentro de estos patrones se utiliza el patrón `Fábrica Abstracta`, este tiene la responsabilidad de crear instancias de objetos de otras clases (Eckel, 2004). El uso de este patrón se verifica en la creación de instancias a las clases que contienen la configuración del cliente. Un ejemplo de ello en la clase `X_chat`, el método `update_message` crea una instancia del método `get_conversation` de la clase `X_chat_lib_loader`.

### 3.3.5 Diagrama de clases del diseño

Los diagramas de clases del diseño son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de este tipo normalmente contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de atributos.

---

<sup>27</sup> Gang of Four (Pandilla de los Cuatro)

- Navegabilidad.
- Dependencias.

Se utilizan para modelar la vista de diseño estática de un sistema, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas (Larman, 1999).

A continuación se muestra el diagrama de clases del diseño referente al caso de uso 'Enviar mensaje'. Este diagrama se empleó en la propuesta de solución con el objetivo de enumerar los elementos que se tienen en el modelo de diseño. Con esto se tiene una idea a simple vista del propósito que tienen las relaciones entre los paquetes, las clases y sus atributos a través de los links. Estos links ejemplifican que existe una interconexión entre clases, específicamente relacionadas con la web. Los demás diagramas de clases del diseño están reflejados en el anexo 4.

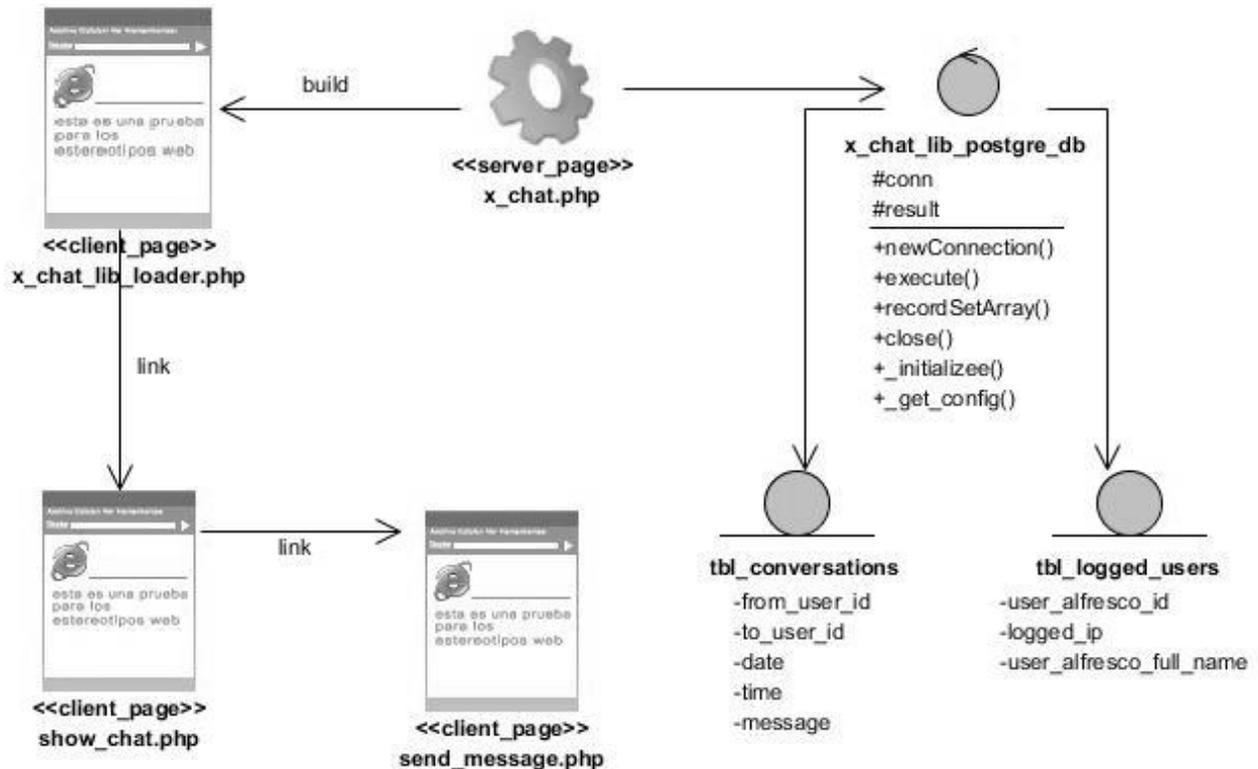


Figura 6: Diagrama de clases del diseño del caso de uso "Enviar mensaje"

### 3.3.6 Modelo físico de la base de datos

En el desarrollo de un sistema informático, el diseño de la base de datos es de gran importancia, ya que en ella se almacenan todos los datos que son necesarios en la modelación del problema que se desea resolver, además esta es la fuente de obtención de toda la información que se quiera recuperar del sistema. Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información.

A continuación se muestra el modelo físico de la base de datos utilizado en la investigación mediante un diagrama Entidad-Relación, modelo que sirve para un mayor entendimiento de la base de datos utilizada y sus tablas. La descripción de este modelo se encuentra en el anexo 5.

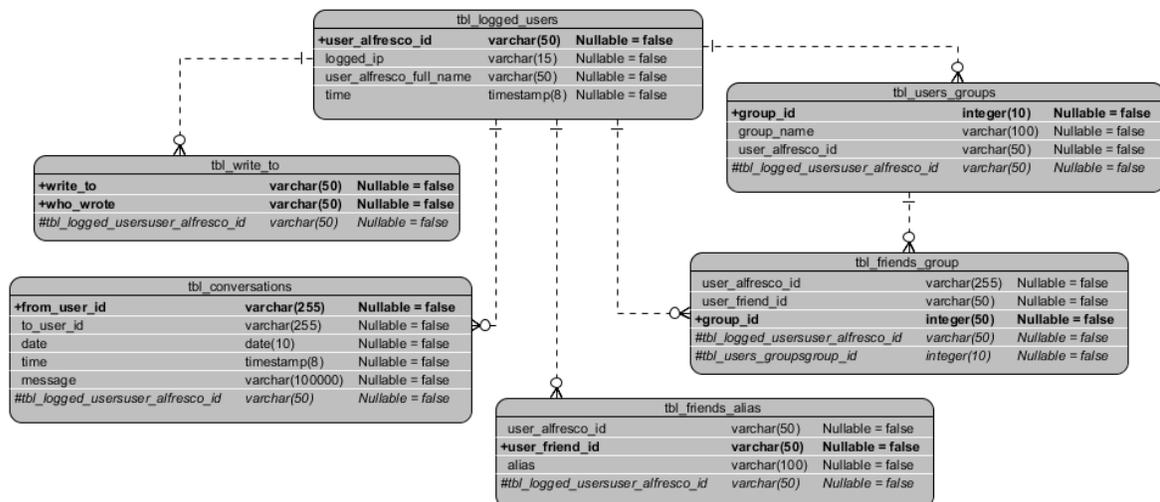


Figura 7: Modelo físico de la base de datos del módulo de mensajería instantánea

### 3.3.7 Estándares de codificación

El uso de técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del *software* y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y posteriormente; se efectúan revisiones del código de rutinas, esto trae consigo que la aplicación se convierta en un sistema de *software* fácil de comprender y

de mantener (Deitel, y otros, 2008).

Con el objetivo de facilitar el mantenimiento del *software*, se emplearon los estándares de codificación que a continuación se ejemplifican.

**Indentación, llaves de apertura y cierre y tamaño de líneas:** Usar una indentación sin tabulaciones, con un equivalente a cuatro espacios. El uso de las llaves {} es seguido del nombre del método. La longitud de las líneas de código no exceden los 70 caracteres.

```
public function update_message()  
{  
    validate_session ();  
    $lib = & init_lib ( 'x_chat_lib_loader' );  
    echo $lib->get_conversation();  
}
```

Figura 8: Ejemplo del uso de los estándares de codificación

### Convención de nomenclatura

**Variables:** Las variables deben contener solo letras minúsculas, usar guión bajo (“\_”) en caso de ser nombres compuestos para esclarecer el propósito y contenido de la variable. Las variables cortas (no palabras) se deben utilizar en bucles for ().

```
for ($i = 0; i < 10; i ++)  
    $sql  
    $date  
    $time  
    $user_id  
    $db_users
```

Figura 9: Ejemplo del uso de los estándares de codificación

**Clases:** Los nombres de las clases deben tener siempre la primera letra mayúscula y todos los métodos de la clase deben ser en minúsculas.

```
class X_chat_lib_loader
{
    private function _initializee()
    {
        @include get_config_file ( 'x_chat' );
        if (isset($config) && count($config))
            return $config;
    }
}
```

Figura 10: Ejemplo del uso de los estándares de codificación

**Funciones:** Comienzan con minúscula y en caso de nombres compuestos se separan por un guión bajo (“\_”). Siempre deben llevar la palabra “public” delante para evitar errores de compilación.

```
public function logout_user()
```

```
private function initializee()
```

Figura 11: Ejemplo del uso de los estándares de codificación

### Estructuras de control

Se incluye if, for, foreach, while y switch. Entre las estructuras de control y los paréntesis debe existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales, esto disminuye la probabilidad de errores lógicos.

```
public function prueba ($var)
{
    //...
}

foreach ($arr as $key => $val)
{
    //...
}

if ($var == $key)
{
    //...
}

for ($i = 0; $i < 10; $i ++ )
{
    for ($j = 0; $j < 10; $j ++ )
    {
        //...
    }
}
```

Figura 12: Ejemplo del uso de los estándares de codificación

### 3.3.8 Diagrama de despliegue

El diagrama de despliegue muestra las relaciones físicas entre los componentes de *hardware* y *software* en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación donde cada nodo puede contener instancias de componentes. En general un nodo puede ser una unidad de computación de algún tipo, desde un sensor hasta un mainframe y las instancias de componentes de *software* pueden estar unidas por relaciones de dependencia (Pressman, 2002).

A continuación se muestra el correspondiente diagrama de despliegue para el desarrollo del módulo de mensajería instantánea propuesto, así como la descripción de sus nodos y sus respectivas conexiones de comunicación o protocolos.

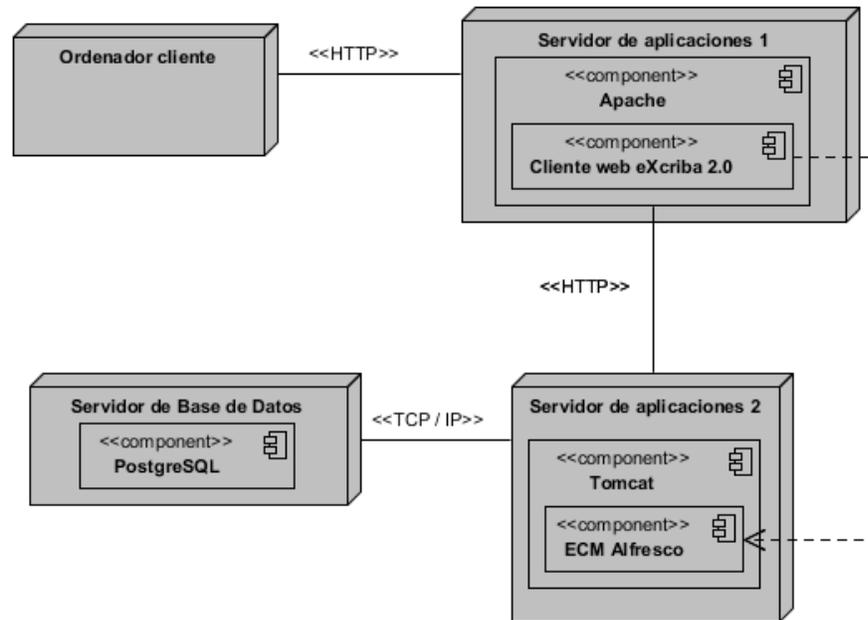


Figura 13: Diagrama de despliegue

### Descripción de cada uno de los componentes del diagrama

Nodo **Ordenador cliente**: Nodo donde se procesan todas las interfaces de usuario que han sido implementadas previamente. Se encuentra el navegador *web* que se utilice el usuario donde este accederá a las interfaces del módulo utilizando el protocolo HTTP.

Nodo **Servidor de aplicaciones 1**: Nodo donde se desarrolla el cliente *web* en un servidor Apache 2.0 y con PHP 5.3.

Nodo **Servidor de aplicaciones 2**: Es donde se encuentra desplegado el núcleo del eXcriba. En el caso de este módulo es donde se encuentran desplegados los servicios que han sido previamente implementados, los cuales permiten modificar los objetos del repositorio.

Nodo **Servidor de base de datos**: Nodo donde se muestra donde está desplegada la base de datos (PostgreSQL 8.4) el cuál utiliza el ECM Alfresco donde es almacenada la información.

Primeramente el usuario desde la PC cliente interactúa con el servidor de aplicaciones 1 donde se encuentra el cliente *web* del GDA eXcriba 2.0 mediante la comunicación que establece el protocolo HTTP

con el ordenador y el servidor. Luego se realiza la conexión con el servidor de aplicaciones 2 para capturar los datos que son solicitados por el usuario, los cuales son llevados a cabo por el protocolo HTTP. El servidor de aplicaciones 2 accede a la información que es almacenada en la base de datos mostrando la respuesta a la petición que es realizada por el servidor.

### 3.3.9 Diagrama de componentes

Los diagramas de componentes son diagramas de implementación que muestran las dependencias del compilador y del “*runtime*” entre los componentes del *software*; por ejemplo, los archivos del código fuente y los dll. Estos describen los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo código fuente, binario y ejecutable (Larman, 1999).

A continuación se muestra el diagrama de componentes que está estrechamente relacionado con los elementos que posee el código fuente del módulo implementado.

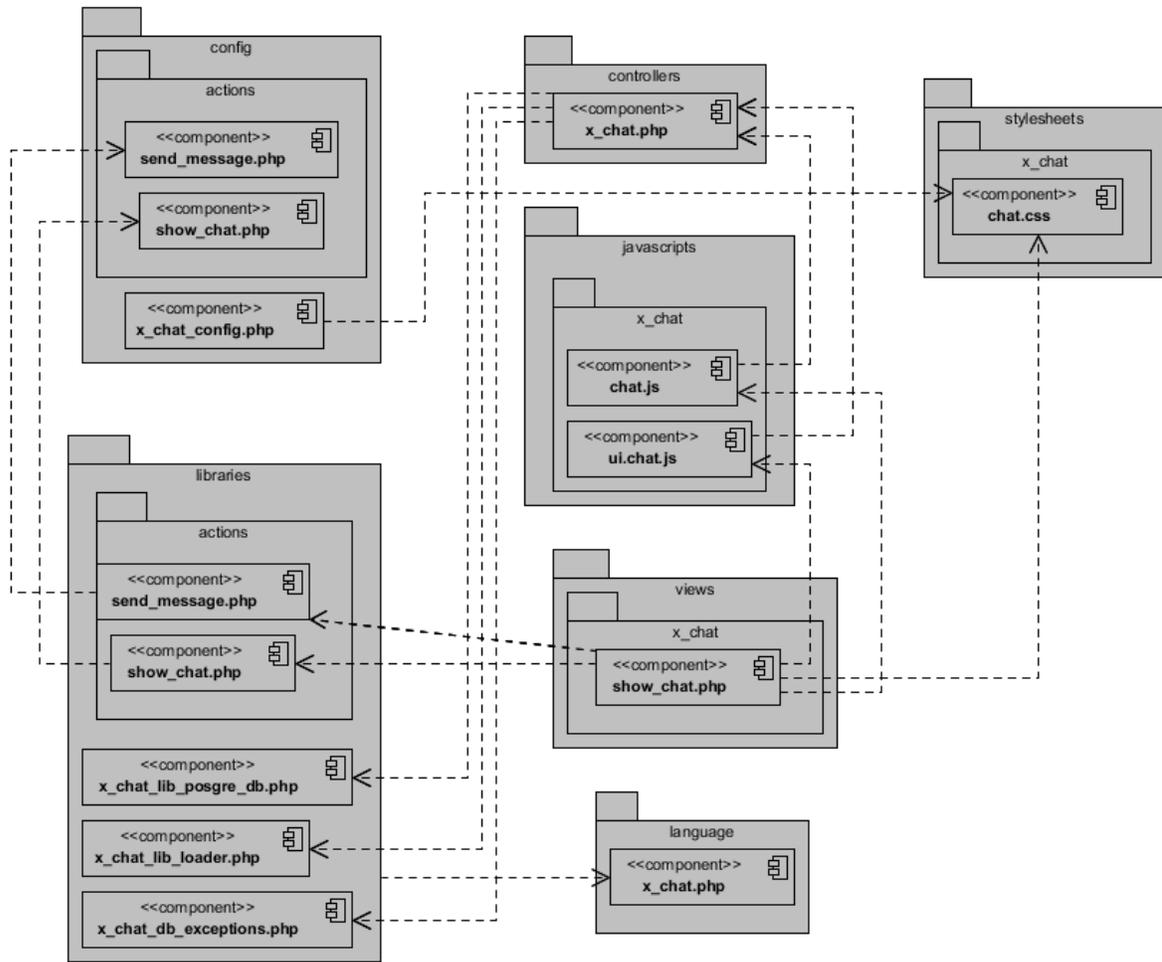


Figura 14: Diagrama de componentes

### 3.3.10 Prototipo del sistema

A continuación se presenta el prototipo de interfaz de usuario correspondiente al módulo de mensajería instantánea que se propone para el GDA eXcriba 2.0.

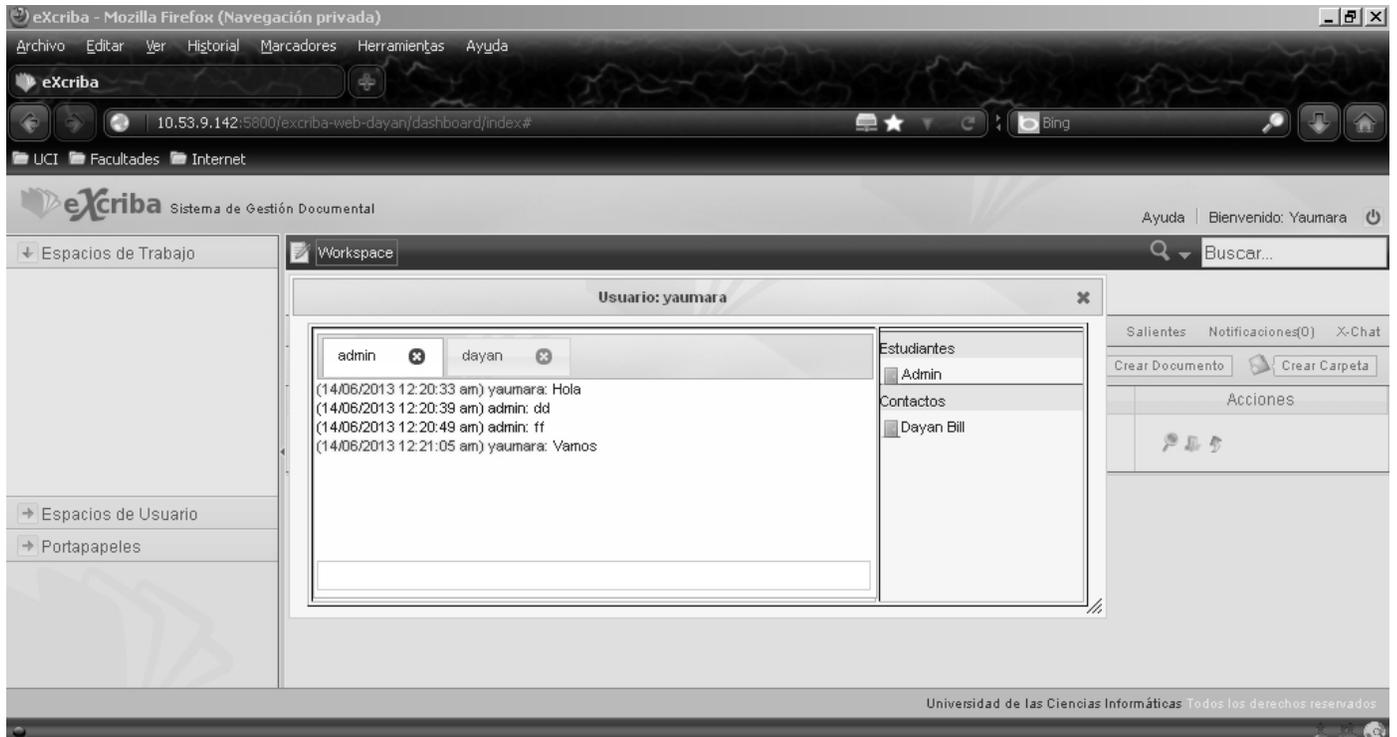


Figura 15: Prototipo de interfaz de usuario del módulo de mensajería instantánea

### 3.4 Conclusiones del capítulo

Luego de haber elaborado este capítulo, se concluye que la realización del análisis de forma detallada facilitó sentar las bases para confeccionar el diseño y posterior construcción del *software*. Se detalló la propuesta de arquitectura y los estándares de codificación utilizados en la implementación de la aplicación. Se propuso la arquitectura del módulo y se expusieron los patrones de diseño utilizados. Además, se presentaron los diagramas de despliegue y de componentes, así como el prototipo de interfaz del módulo.

### *Capítulo 4. Validación de los resultados del módulo*

#### **Introducción**

En el ciclo de desarrollo de un *software*, para obtener un producto terminado con la calidad requerida y que cumpla con las expectativas del cliente, un aspecto importante es la realización de las pruebas. Su objetivo es localizar el mayor número de deficiencias lo antes posible para corregirlas, ahorrando tiempo y recursos en el desarrollo. Por lo analizado anteriormente, en el presente capítulo se desarrollarán pruebas unitarias, a través de la aplicación de casos de pruebas de caja negra y de caja blanca con el objetivo de verificar y validar los resultados de la implementación del módulo.

#### **4.1 Casos de prueba de caja negra**

Las pruebas de caja negra no consideran la codificación dentro de sus parámetros a evaluar, es decir, no están basadas en el diseño interno del programa. El objetivo que tienen estas pruebas es el de revelar el incorrecto o incompleto funcionamiento del sistema, así como los errores de interfaz, rendimiento y errores de inicialización y terminación. Estas pruebas se enfocan en los requerimientos establecidos y en las funcionalidades del sistema.

A través de esta técnica, los casos de prueba desarrollados en el presente trabajo pretenden demostrar que las funciones que ofrece el módulo son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Los resultados de esta prueba generalmente han sido satisfactorios. Los casos de prueba para las historias de usuario se encuentran en el anexo 6.

#### **4.2 Casos de prueba de caja blanca**

En la prueba de caja blanca se realiza un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, bucles y condiciones; y examinando el estado del programa en varios puntos. Para realizar estas pruebas podemos apoyarnos en varias técnicas para validar la solución:

- **Prueba de Condición:** Ejercita las condiciones lógicas contenidas en el módulo de un programa.
- **Prueba de Flujo de Datos:** Se seleccionan caminos de prueba de un programa de acuerdo con la

ubicación de las definiciones y los usos de las variables del programa.

- **Prueba de Bucles:** Es una técnica que se centra exclusivamente en la validez de las construcciones de bucles.
- **Prueba del Camino Básico:** Permite obtener una medida de la complejidad lógica de un diseño, centrándose en encontrar y probar los caminos básicos por los que circula el flujo.

Para la realización de los casos de prueba de caja blanca se decidió realizar las pruebas del camino básico debido a que garantiza que se verifiquen todos los caminos por los que circula el flujo del algoritmo. A continuación se muestran los pasos a seguir para aplicar esta técnica:

- A partir del código fuente se dibuja el grafo de flujo asociado.
- Se calcula la complejidad ciclomática del grafo.
- Se determina un conjunto básico de caminos independientes.
- Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

La figura siguiente muestra las sentencias de código enumeradas del procedimiento realizado sobre el método `updateLoggegUsers()`.

```
var updateLoggedUsers = function()
{
    $.post(uncache(CI_ROOT + 'x_chat/update_logged_users'), function(data)
    {
        var json = eval('(' + data + ')'); 1
        if( json != undefined && json != null ) 2
        {
            loggedUsersDiv = document.getElementById("loggedUsersToChat"); 3
            if( loggedUsersDiv != null && loggedUsersDiv != undefined ) 4
            {
                document.getElementById("loggedUsersToChat").innerHTML = json.html; 5
                for( var i in blinkList ) 6
                blink( blinkList[i], true ); 7
            } 8
        } 9
    });
}
```

Figura 16: Representación del algoritmo `updateLoggedUsers()`

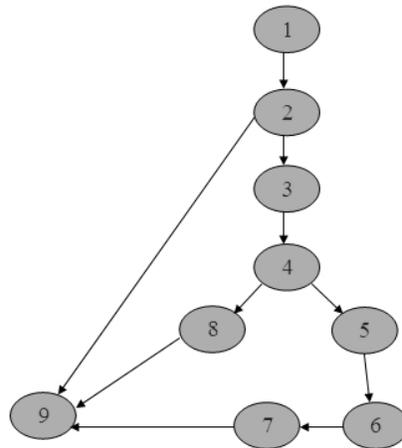


Figura 17: Grafo del flujo asociado al algoritmo `updateLoggedUsers()`

Fórmulas para calcular la complejidad ciclomática

$$V(G) = (A - N) + 2$$

$$V(G) = (10 - 9) + 2$$

$$V(G) = 3$$

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

Se puede usar también:

$$V(G) = P + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$

Siendo “P” la cantidad total de nodos predicados (nodos de los cuáles parten dos o más aristas).

$$V(G) = R$$

$$V(G) = 3$$

Siendo “R” la cantidad total de regiones, para cada fórmula, “V (G)” representa el valor del cálculo.

El resultado obtenido mediante las fórmulas anteriores representa los posibles caminos por los que

transitará el flujo, así como la cantidad mínima de casos de prueba para el procedimiento escogido. A continuación se representan los caminos básicos que se identificaron en el flujo:

*Tabla 3: Caminos básicos*

Camino 1	1-2-9
Camino 2	1-2-3-4-8-9
Camino 3	1-2-3-4-5-6-7-9

Una vez extraídos los caminos básicos, se procede a realizar los casos de pruebas, teniendo en cuenta que se debe de realizar al menos un caso de prueba por cada camino básico:

### **Caso de prueba para el camino básico 1**

Camino 1: [1-2-9].

Descripción: Se verifica que el código HTML mostrado sea válido.

Condición de ejecución: El código HTML referente a los usuarios logueados no está definido o es nulo.

Entrada: json = "";

Resultados esperados: No se muestran los usuarios logueados.

### **Caso de prueba para el camino básico 2**

Camino 2: [1-2-3-4-8-9]

Descripción: Se verifica que el id de los usuarios sean mostrados en una tabla mediante el código definido, ya que cada id es único.

Condición de ejecución: No hay usuarios en la tabla que contiene los usuarios logueados.

Entrada: json = "Código HTML referente a los usuarios logueados", loggedUsersDiv = "".

Resultados esperados: Se actualiza la tabla pero no se muestran los usuarios logueados.

### **Caso de prueba para el camino básico 3**

Camino 3: [1-2-3-4-5-6-7-9]

Descripción: Se verifican los usuarios logueados y los usuarios que parpadean.

Condición de ejecución: Concatena el código HTML con el id de los usuarios a mostrarse en la tabla y parpadean los usuarios que han enviado mensajes.

Entrada: json = “Código HTML”, loggedUsersDiv = “Dayan Bermúdez, Hilario Mastrapa, Michel David”.

Resultados esperados: Se actualiza la tabla con los usuarios logueados y con los usuarios que han enviado mensajes.

### 4.3 Resultados de las pruebas realizadas

A continuación se muestra una gráfica que contiene la relación de las no conformidades detectadas con las corregidas en cada una de las iteraciones realizadas.

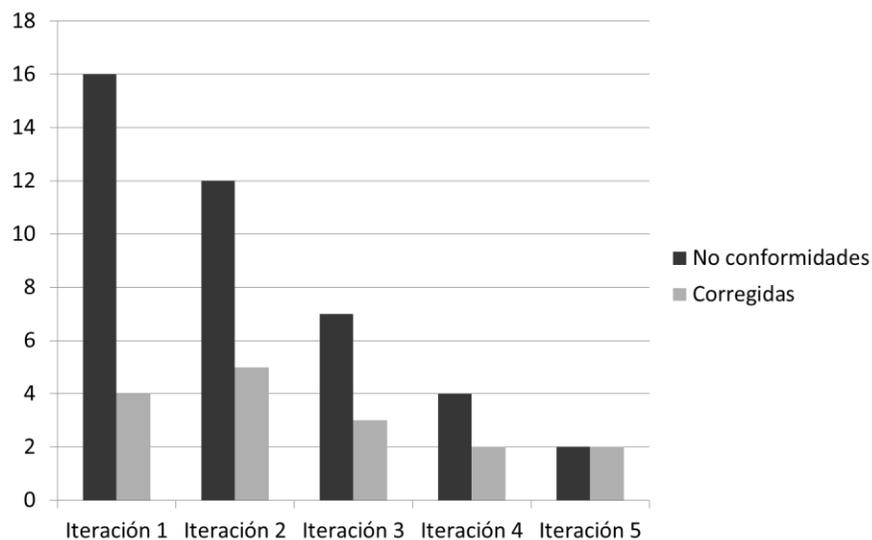


Figura 18: Gráfica de no conformidades detectadas y corregidas correspondientes a las pruebas realizadas al módulo

Como se aprecia en la figura anterior, se presentan las no conformidades detectadas y corregidas en cada una de las iteraciones de pruebas realizadas al sistema, lo que demuestra que a medida que se realizaron estas pruebas, fueron disminuyendo las no conformidades, que sirvieron para corregir los errores que existían. La temprana solución de estos problemas ayudó a que la aplicación tuviera una mejor calidad y cumpliera con los requisitos planteados.

### 4.4 Conclusiones del capítulo

En el capítulo se diseñaron y aplicaron los casos de prueba usados durante el desarrollo de la solución y

se describieron las pruebas unitarias a las que fue sometida la aplicación, con la finalidad de asegurar que esta cumpla con los requerimientos funcionales. En todos los casos, las mismas devolvieron resultados satisfactorios.

## *Conclusiones*

A partir del análisis de los sistemas de mensajería instantánea y de un breve análisis de los sistemas de gestión documental, se lograron identificar elementos teóricos que caracterizan el problema existente. Se decidió desarrollar como solución al problema de investigación un sistema de mensajería instantánea para el GDA eXcriba 2.0.

Con el análisis y diseño se identificaron las funcionalidades básicas a tener en cuenta para comenzar la construcción del *software*. Con la implementación de estas funcionalidades, se obtuvo el producto final que permitió validar la arquitectura propuesta. Con este módulo desarrollado se logra la comunicación interna entre los usuarios del GDA eXcriba 2.0, ya que ninguno de los sistemas de mensajería estudiados se adapta para ser incorporados al sistema. También se lograron realizar las pruebas unitarias al sistema comprobando que los requerimientos seleccionados fueron exitosamente cumplidos.

## *Recomendaciones*

Se recomienda:

- Comenzar a utilizar el módulo dentro del proyecto, tal que pueda ser una opción para las actividades productivas.
- Utilizar el módulo en los sitios *web* que poseen las facultades de la universidad.
- Agregar para futuras versiones, la conexión del módulo con los demás clientes de mensajería instantánea que posee la universidad.

## *Referencias bibliográficas*

- AIIM. 2013.** AIIM. [En línea] AIIM, 2013. [Citado el: 30 de Mayo de 2013.] <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>.
- Alfresco Software. 2013.** Alfresco. [En línea] Alfresco Software, 2013. [Citado el: 15 de Abril de 2013.] <http://www.alfresco.com/es/recorrido-por-alfresco>.
- Álvarez, Miguel Ángel. 2013.** desarrolloweb.com. [En línea] 2013. [Citado el: 28 de Mayo de 2013.] <http://www.desarrolloweb.com/manuales/manual-jquery.html>.
- Bahit, Eugenia. 2011.** *POO y MVC en PHP*. 2011.
- Billy Reynoso, Carlos. 2004.** *Introducción a la Arquitectura de Software*. Buenos Aires : s.n., 2004.
- Binovo Hit Human Project, S.L. 2013.** binovo. [En línea] Binovo Hit Human Project, S.L., 2013. [Citado el: 10 de Junio de 2013.] [http://www.binovo.es/index.php?option=com\\_content&view=article&id=110&Itemid=89](http://www.binovo.es/index.php?option=com_content&view=article&id=110&Itemid=89).
- Bueno, Abel. 2013.** Softonic. [En línea] INTERSHARE, S.L., 2013. [Citado el: 11 de Marzo de 2013.] <http://ebuddy.softonic.com/aplicaciones-web>.
- Castagnino, Pablo. 2010.** Usemos Linux. [En línea] BY-NC-SA, 15 de Marzo de 2010. [Citado el: 15 de Febrero de 2013.] <http://usemoslinux.blogspot.com/2010/03/los-12-mejores-clientes-de-chat.html#>.
- Castro, Luis. 2013.** About.com. [En línea] IAC/InterActiveCorp, 2013. [Citado el: 26 de Mayo de 2013.] <http://aprenderinternet.about.com/od/ChatsForosEtc/a/Que-Es-Im-O-Mensajeria-Instantanea-Y-Como-Funciona.htm>.
- Deitel, P. J. y Deitel, H. M. 2008.** *¿Cómo programar en Java?* México : Pearson Educación, 2008. ISBN: 978-970-26-1190-5.
- Eckel, Bruce. 2004.** *Pensar en C++*. s.l. : Maindview Inc., 2004. Vol. 2.
- Eguíluz Pérez, Javier. 2009.** *Introducción a JavaScript*. 2009.
- Escalona, María José y Koch, Nora. 2002.** *Ingeniería de requisitos en aplicaciones para la Web - Un estudio comparativo*. Sevilla : s.n., 2002.
- Ferri-Benedetti, Fabrizio. 2013.** Softonic. [En línea] INTERSHARE, S.L., 2013. [Citado el: 27 de Mayo de 2013.] <http://meebo.softonic.com/aplicaciones-web>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004.** *El Proceso Unificado de Desarrollo de Software*. La Habana : Félix Varela, 2004.

**Knowledge Tree. 2013.** software.com.ar. [En línea] Targetware Informática S.A.C., 2013. [Citado el: 20 de Febrero de 2013.] <http://www.software.com.ar/knowledgetree.html>.

**Larman, Craig. 1999.** *UML y patrones. Introducción al análisis y diseño orientado a objetos*. Monterrey, México : Pearson Educación, 1999. ISBN: 970-17-0261-1.

**Leo, Junior. 2009.** AplicacionesEmpresariales.com. [En línea] 23 de Febrero de 2009. [Citado el: 15 de Febrero de 2013.] <http://www.aplicacionesempresariales.com/pidgin-el-cliente-universal-de-mensajeria.html>.

**LetoDMS. 2009.** LetoDMS. [En línea] Trilex Labs, 2009. [Citado el: 19 de Febrero de 2013.] <http://www.letodms.com/module.php?modname=content&cid=3>.

**Nuxeo. 2013.** Nuxeo / Content Management Platform For Business Applications. [En línea] Nuxeo, 2013. [Citado el: 10 de Marzo de 2013.] <http://www.nuxeo.com/en/products/document-management>.

**OpenKM. 2011.** OpenKM. Knowledge Management. [En línea] OpenKM, 2011. [Citado el: 21 de Febrero de 2013.] <http://www.openkm.com/es/>.

**PostgreSQL. 2013.** PostgreSQL. [En línea] The PostgreSQL Global Development Group, 2013. [Citado el: 24 de Febrero de 2013.] <http://www.postgresql.org/about/>.

**Pressman, Roger S. 2002.** *Ingeniería de Software, un enfoque práctico.2001*. Quinta edición. s.l. : S.l: McGraw-Hill Companies, 2002. 8448132149.

**Proyecto UNE-ISO 15489-1. 2005.** *Información y documentación. Gestión de documentos. Parte 1: Generalidades*. Madrid : s.n., 2005.

**Reingart, Mariano. 2010.** ArPUG. Grupo de usuarios PostgreSQL de Argentina. [En línea] PostgreSQL Powered, 2010. [Citado el: 14 de Mayo de 2013.] <http://www.arpug.com.ar/trac/wiki/PgAdmin>.

**Saether Bakken, Stig. 2002.** CALITAE. [En línea] 4 de Noviembre de 2002. [Citado el: 20 de Febrero de 2013.] <http://www.calitae.com/manuales/manual-php.pdf>.

**Sommerville, Ian. 2005.** *Ingeniería del software*. [ed.] Miguel Martín-Romo. [trad.] María Isabel Alfonso Galipienso, y otros. Séptima Edición. Madrid : s.n., 2005.

**The PHP Group. 2013.** php. [En línea] The PHP Group, 2013. [Citado el: 19 de Mayo de 2013.] <http://php.net/manual/es/book.ldap.php>.

**Universidad de Zaragoza. 2008.** Servicio de Informática y Comunicaciones Universidad de Zaragoza. [En línea] Universidad de Zaragoza, 2008. [Citado el: 28 de Mayo de 2013.] [https://webmail.unizar.es/mail\\_protocolos.php](https://webmail.unizar.es/mail_protocolos.php).

**Visual Paradigm. 2013.** software.com.ar. [En línea] Targetware Informática S.A.C., 2013. [Citado el: 21 de Febrero de 2013.] <http://www.software.com.ar/visual-paradigm-para-uml.html>.

**W3C. 2010.** W3C. [En línea] W3C, 2010. [Citado el: 2 de Junio de 2013.]  
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

**XMPP Standars Foundation. 2013.** XMPP Standars Foundation. [En línea] XMPP Standars Foundation, 11 de Junio de 2013. [Citado el: 23 de Mayo de 2013.] <http://xmpp.org/>.

**Zend Technologies Ltd. 2013.** Zend. [En línea] Zend Technologies Ltd, 2013. [Citado el: 26 de Mayo de 2013.]  
<http://www.zend.com/en/products/studio/>.

## Bibliografía

- **Catálogo Biblioteca UCI en línea.** [Consultado el: 21 de Marzo de 2013]. Disponible en: <http://catalogoenlinea.uci.cu/>.
- **Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas.** [Consultado el: 15 de Febrero de 2013]. Disponible en: <https://eva.uci.cu/>.
- **Larman, Craig.** *UML y patrones.* Monterrey, México. Pearson Education, 1999. ISBN: 970-17-0261-1.
- **PostgreSQL.** [Consultado el: 19 de Febrero de 2013]. Disponible en: <http://www.postgresql.org/docs/8.4/interactive/index.html>.
- **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico.* Quinta Edición. La Habana. McGraw Hill, 2002.
- **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico.* Sexta Edición. La Habana. Félix Varela, 2005.
- **Sommerville, Ian.** *Ingeniería del software.* Séptima Edición. Madrid, España. Pearson Educación, 2005.
- **Sunshine.uci.cu.** [Consultado el: 14 de Abril de 2013]. Disponible en: <http://sunshine.prod.uci.cu/>.
- **Eguíluz Pérez, Javier.** *Introducción a JavaScript.* 2009.
- **IEEE Guide for Developing System Requirements Specifications.** 1998.
- **Proyecto UNE-ISO 15489-1.** *Información y documentación. Gestión de documentos. Parte 1: Generalidades.* 2005
- **¿Qué es la gestión de contenido empresarial?** [Consultado el: 6 de Mayo de 2013]. Disponible en: <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>.
- **DMS: Document Management System.** [Consultado el: 13 de Marzo de 2013]. Disponible en: <http://openpyme.osl.ull.es/DMS/applications>.
- **Mena Mugica, Mayra.** *Gestión documental y organización de archivos.* La Habana. Félix Varela, 2009.
- **Booch, Grady.** *El proceso de desarrollo de desarrollo de software.* La Habana. Félix Varela, 2004.
- **PHP. General Information.** [Consultado el: 9 de Abril de 2013]. Disponible en: <http://www.php.net/manual/en/faq.general.php#faq.general.what>.

- **JavaScript Ya.** [Consultado el: 10 de Abril de 2013]. Disponible en: <http://www.javascriptya.com.ar>.
- **Arias Calleja, Manuel.** *Estándares de codificación.* Madrid, España. 2013.

## *Glosario de términos*

### **A**

AJAX: Técnica de desarrollo *web* para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el navegador de los usuarios mientras se mantiene la comunicación con el servidor en segundo plano. Esta técnica es válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DOM.

API: Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca (libraries) para ser utilizado por otro *software* por una capa de abstracción.

### **B**

Banner: En español banderola. Formato publicitario en Internet. Esta forma de publicidad online consiste en incluir una pieza publicitaria dentro de una página *web* cuyo objetivo es atraer tráfico hacia el sitio *web* del anunciante que paga por su inclusión.

### **C**

CIFS: Nombre que adoptó la empresa Microsoft en 1998 para el protocolo SMB (Server Message Block), protocolo que permite compartir archivos e impresoras (entre otras cosas) entre nodos de una red.

### **D**

Desktop: Refiérase a las aplicaciones que son creadas para ejecutarse en un ordenador de escritorio, sobre un sistema operativo con interfaz visual como Windows o Linux.

DLL: Siglas en inglés de Dynamic-Link Library. Biblioteca de encale dinámico, término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

DOM: Una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML. A través del DOM, los programas pueden acceder y

modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

### **E**

ECM: Siglas en inglés de Enterprise Content Management.

### **G**

GNOME: Entorno de escritorio e infraestructura de desarrollo para sistemas operativos Linux y Unix, compuesto enteramente de *software* libre.

### **H**

HTML: Lenguaje de marcado predominante para la elaboración de páginas *web*. Usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP: Siglas en inglés de Hypertext Transfer Protocol.

### **I**

ICQ: Cliente de mensajería instantánea mediante el cual es posible chatear y enviar mensajes instantáneos a otros usuarios conectados a la red de ICQ.

### **M**

Miranda IM: El Miranda Instant Messenger es un cliente de mensajería instantánea libre y multiprotocolo para el sistema operativo Microsoft Windows, licenciado bajo la licencia GNU General Public License.

MSN Messenger: Cliente de mensajería instantánea y su nombre se utiliza para diferenciar todos los programas de mensajería de Microsoft.

MySpace: Servicio de red social (como Facebook), propiedad de la empresa Specific Media LLC y la estrella del pop Justin Timberlake.

### P

Plug-in: Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver (controlador) en una aplicación para hacer así funcionar un dispositivo en otro programa. Esta aplicación adicional es ejecutada por la aplicación principal.

### R

Repositorio: Es un término utilizado en el dominio de las herramientas CASE. El repositorio podría definirse como la base de datos fundamental para el diseño; no solo guarda datos, sino también algoritmos de diseño y, en general, elementos *software* necesarios para el trabajo de programación.

RUP: Siglas en inglés de Rational Unified Process.

### S

Skype: *Software* que permite comunicaciones de texto, voz y video sobre Internet.

SQL: Lenguaje de acceso a base de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

### T

TCP/IP: Conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadora. Los dos protocolos que la componen: Protocolo de Controla de Transmisión (TCP) y Protocolo de Internet (IP).

Telefonía IP: Tecnología que permite integrar en una misma red – basada en el protocolo IP – las comunicaciones de voz y datos.

### U

UML: Siglas en inglés de Unified Modeling Language.

### V

VoIP: Grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP.

### **W**

Widgets: Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños ejecutados por un motor de widgets o Widget Engine. Pueden ser vistosos relojes en pantalla, notas, calculadoras, calendarios, agendas, juegos, etcétera.

Windows Live Messenger: Anteriormente MSN Messenger, fue un cliente de mensajería instantánea creado por Microsoft y diseñado para funcionar en PC con Microsoft Windows y en dispositivos móviles.

### **X**

XML: Lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). A diferencia de otros lenguajes, XML da soporte a base de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

XMPP: Protocolo abierto y extensible basado en XML, originalmente ideado para la mensajería instantánea.

## Anexos

### Anexo 1. Descripción de los casos de uso

A continuación, se tienen los casos de uso identificados y su descripción.

Tabla 1. 1: Descripción del caso de uso “Abrir conversación”

<b>Objetivo</b>	Abrir conversación	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el actor selecciona un usuario de la lista de usuarios conectados y de esta forma abrir una nueva conversación. El caso de uso termina cuando se abre una nueva conversación identificada por una pestaña que tiene id del usuario seleccionado.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El actor debe tener un usuario conectado.	
<b>Postcondiciones</b>	El actor abre una conversación identificada con una pestaña que contiene el id del usuario que seleccionó.	
<b>Flujo de eventos</b>		
<b>Flujo básico: “Abrir conversación”</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario selecciona de la lista de usuarios conectados, el usuario con el cuál desea abrir una nueva conversación.	2. El sistema muestra la interfaz identificada con una nueva pestaña, a su vez esta muestra el id del usuario que se seleccionó previamente.	
<b>Flujos alternos</b>		

<b>Relaciones</b>	<b>CU Incluidos</b>	<i>Abrir conversación</i>
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b> no	RNF.01, RNF.02 y RF.03.	
<b>Asuntos pendientes</b>		

Tabla 1. 2: Descripción del caso de uso "Cerrar conversación"

<b>Objetivo</b>	Cerrar conversación	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el actor tiene una o varias conversaciones que desea cerrar y termina cuando el actor cierra la conversación.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El actor debe tener al menos una conversación abierta.	
<b>Postcondiciones</b>	El actor cierra la conversación.	
<b>Flujo de eventos</b>		
<b>Flujo básico: "Cerrar conversación"</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario cierra una conversación activa.	2. El sistema muestra todas las conversaciones menos la que el usuario ha cerrado.	
<b>Flujos alternos</b>		

<b>Relaciones</b>	<b>CU Incluidos</b>	<i>Cerrar conversación</i>
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b>	<b>no</b>	
<b>Asuntos pendientes</b>		

Tabla 1. 3: Descripción del caso de uso "Crear grupo de usuarios" perteneciente al caso de uso "Gestionar grupo de usuarios"

<b>Objetivo</b>	Crear grupo de usuarios	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el actor desea crear un grupo de usuarios y termina cuando el grupo de usuarios es creado.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El actor debe tener al menos un usuario conectado.	
<b>Postcondiciones</b>	El grupo de usuarios es creado con el nombre que el actor le ha asignado.	
<b>Flujo de eventos</b>		
<b>Flujo básico: "Crear grupo de usuarios"</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario da clic derecho encima del usuario seleccionado.	2. El sistema muestra un menú con varias opciones a escoger.	
3. El usuario escoge la opción "Grupo de usuarios".	4. El sistema muestra un submenú con la opción "Crear nuevo grupo".	

5. El usuario escoge la opción "Crear nuevo grupo".		6. El sistema muestra una nueva interfaz titulada "Nombre del grupo" con los botones "Aceptar" y "Cancelar".
7. El usuario le especifica un nombre al nuevo grupo en la interfaz mostrada y presiona el botón "Aceptar".		8. El sistema muestra el grupo de usuarios creado con el usuario seleccionado como único usuario dentro del grupo creado.
<b>Flujos alternos</b>		
<b>Sección 1: "Nombre de grupo no especificado"</b>		
7.1	El usuario no le especifica un nombre al nuevo grupo en la interfaz mostrada y presiona el botón "Aceptar".	El sistema muestra un mensaje de alerta: "Por favor, especifique un nombre".
7.2	El usuario presiona el botón "Cancelar".	El sistema no crea el grupo de usuarios.
<b>Relaciones</b>	<b>CU Incluidos</b>	<i>Crear grupo de usuarios</i>
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b>	<b>no</b>	
<b>Asuntos pendientes</b>		

Tabla 1. 4: Descripción del caso de uso "Cambiar nombre de los usuarios"

<b>Objetivo</b>	Cambiar nombre de los usuarios
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el actor desea cambiarle el nombre a un usuario conectado. El caso de uso termina cuando el nombre del usuario seleccionado ha sido cambiado.
<b>Complejidad</b>	Media

<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe tener al menos un usuario conectado para cambiarle el nombre.	
<b>Postcondiciones</b>	El nombre del usuario seleccionado es cambiado por el alias que el usuario le ha asignado.	
<b>Flujo de eventos</b>		
<b>Flujo básico: "Cambiar nombre de los usuarios"</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario da clic derecho encima del usuario seleccionado con el que desea cambiarle el nombre (Crearle un alias).	2. El sistema muestra un menú con varias opciones a escoger.	
3. El usuario escoge la opción "Cambiar nombre".	4. El sistema muestra una nueva interfaz titulada "Cambiar nombre". Esta interfaz muestra el nombre original del usuario seleccionado. Además muestra los botones "Aceptar" y "Cancelar".	
5. El usuario le asigna un alias (le pone un nombre nuevo) al usuario seleccionado dentro de la interfaz nueva mostrada y presiona el botón "Aceptar".	6. El sistema muestra el usuario seleccionado con el nombre que se le ha asignado (alias).	
<b>Flujos alternos</b>		
<b>Sección 1: "Alias no especificado"</b>		
5.1	El usuario no le especifica un alias al usuario seleccionado y presiona el botón "Aceptar".	El sistema muestra un mensaje de alerta: "No se ha cambiado el nombre". El módulo no muestra ningún cambio realizado sobre el usuario seleccionado, no le ha cambiado el nombre.
5.2	El usuario presiona el botón "Cancelar".	El sistema no muestra ningún cambio sobre el usuario seleccionado.

<b>Relaciones</b>	<b>CU Incluidos</b>	Cambiar nombre de los usuarios
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b>	no	
<b>Asuntos pendientes</b>		

Tabla 1. 5: Descripción del caso de uso "Recibir mensaje"

<b>Objetivo</b>	Recibir mensaje	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando un mensaje es recibido y termina cuando el mensaje recibido es mostrado por el sistema.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El actor debe tener al menos un usuario conectado parpadeando.	
<b>Postcondiciones</b>	El usuario que parpadea es seleccionado para recibir el mensaje.	
<b>Flujo de eventos</b>		
<b>Flujo básico: "Recibir mensaje"</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario selecciona el usuario que parpadea.	2. El sistema muestra el mensaje que el usuario ha enviado.	
<b>Flujos alternos</b>		

<b>Relaciones</b>	<b>CU Incluidos</b>	<i>Recibir mensaje</i>
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b> no		
<b>Asuntos pendientes</b>		

Tabla 1. 6: Descripción del caso de uso "Agregar usuarios a un grupo creado" perteneciente al caso de uso "Gestionar grupo de usuarios"

<b>Objetivo</b>	Agregar usuarios a un grupo creado	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el actor desea agregar un usuario a un grupo de usuarios creado previamente. El caso de uso termina cuando el usuario se muestra en el grupo al que ha sido agregado.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El actor debe tener al menos un usuario conectado para añadirlo a un grupo de usuarios creado.	
<b>Postcondiciones</b>	El usuario se muestra en el grupo de usuarios al que ha sido añadido.	
<b>Flujo de eventos</b>		
<b>Flujo básico: "Agregar usuarios a un grupo creado"</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario selecciona un usuario y le da clic derecho encima de él.	2. El sistema muestra un menú con varias opciones a escoger.	

3. El usuario escoge la opción "Grupo de usuarios".	4. El sistema muestra un submenú con varias opciones a escoger.
5. El usuario selecciona el nombre del grupo de usuarios creado al que desea agregarle el usuario escogido.	6. El sistema muestra el usuario dentro del grupo de usuarios al cual ha sido agregado.
<b>Flujos alternos</b>	
<b>Relaciones</b>	<b>CU Incluidos</b>
	<i>Agregar usuarios a un grupo creado</i>
	<b>CU Extendidos</b>
<b>Requisitos funcionales</b>	<b>no</b>
<b>Asuntos pendientes</b>	

Tabla 1. 7: Descripción del caso de uso "Eliminar grupo de usuarios" perteneciente al caso de uso "Gestionar grupo de usuarios"

<b>Objetivo</b>	Eliminar grupo de usuarios
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el actor desea borrar un grupo de usuarios creado. El caso de uso termina cuando el grupo de usuarios es borrado. Sólo se borra el grupo de usuarios, los usuarios dentro de cada grupo borrado son incorporados nuevamente a la lista de usuarios conectados por defecto.
<b>Complejidad</b>	Baja
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	El actor debe tener al menos un grupo de usuarios creado.
<b>Postcondiciones</b>	El grupo de usuarios seleccionado es borrado.
<b>Flujo de eventos</b>	

Flujo básico: "Eliminar grupo de usuarios"		
Actor	Sistema	
1. El usuario selecciona un grupo de usuarios que desea borrar y le da clic derecho encima de él.	2. El sistema muestra un menú con la opción "Eliminar grupo".	
3. El usuario escoge la opción "Eliminar grupo".	4. El sistema muestra un mensaje de alerta: "¿Desea borrar el grupo de usuarios "Nombre del grupo"?" con los botones "Aceptar" y "Cancelar".	
5. El usuario presiona el botón "Aceptar".	6. El sistema actualiza los usuarios y los grupos de usuarios que tiene el usuario, significa que no muestra el grupo borrado.	
Flujos alternos		
Sección 1: "Grupo no borrado"		
5.1	El usuario presiona el botón "Cancelar".	El sistema no muestra ningún cambio realizado.
Relaciones	CU Incluidos	<i>Eliminar grupo de usuarios</i>
	CU Extendidos	
Requisitos funcionales	<b>no</b>	
Asuntos pendientes		

Tabla 1. 8: Descripción del caso de uso "Enviar mensaje"

<b>Objetivo</b>	Enviar mensaje
<b>Actores</b>	Usuario

<b>Resumen</b>	El caso de uso se inicia cuando el actor selecciona un usuario para de esta forma enviar un mensaje. El caso de uso termina cuando el mensaje es enviado.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El actor debe seleccionar un usuario para enviar un mensaje.	
<b>Postcondiciones</b>	El mensaje es enviado.	
<b>Flujo de eventos</b>		
<b>Flujo básico: "Enviar mensaje"</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El usuario selecciona de la lista de usuarios conectados el usuario o los usuarios a los cuáles le va a enviar los mensajes de texto.	2. El sistema muestra una interfaz que contiene el nombre o los nombres de los usuarios que el usuario seleccionó previamente.	
3. El usuario escribe los mensajes de texto que desee al usuario o los usuarios seleccionados.	4. El sistema envía los mensajes satisfactoriamente.	
<b>Flujos alternos</b>		
<b>Relaciones</b>	<b>CU Incluidos</b>	<i>Enviar mensaje</i>
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b> <b>no</b>	RNF.06, RNF.08.	
<b>Asuntos pendientes</b>		

## Anexo 2. Diagramas de clases del análisis

A continuación, se muestran los diagramas de clases del análisis para los casos de uso restantes.

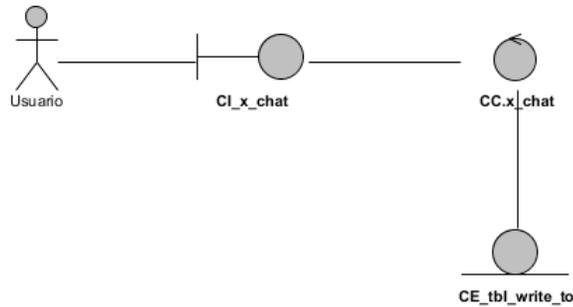


Figura 2. 1: Diagrama de clases del análisis para el caso de uso "Abrir conversación"

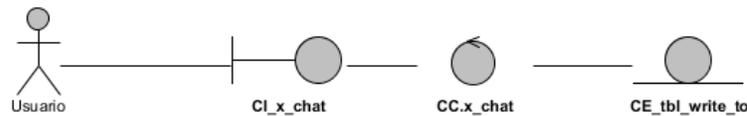


Figura 2. 2: Diagrama de clases del análisis para el caso de uso "Cerrar conversación"

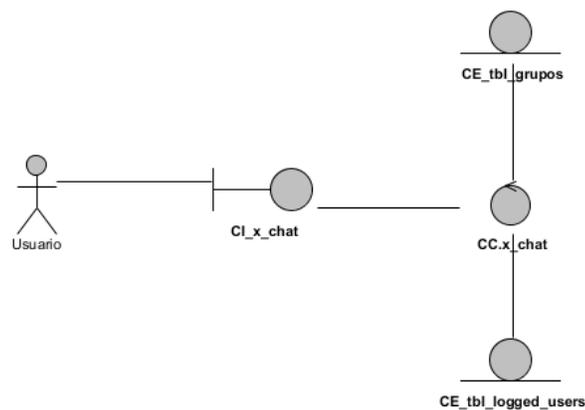


Figura 2. 3: Diagrama de clases del análisis para el caso de uso "Crear grupo de usuarios" perteneciente al caso de uso "Gestionar grupo de usuarios"

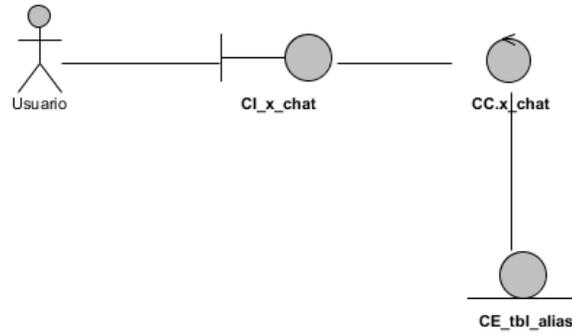


Figura 2. 4: Diagrama de clases del análisis para el caso de uso "Cambiar nombre de los usuarios"

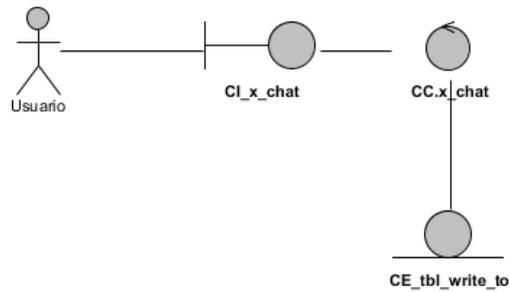


Figura 2. 5: Diagrama de clases del análisis para el caso de uso "Recibir mensaje"

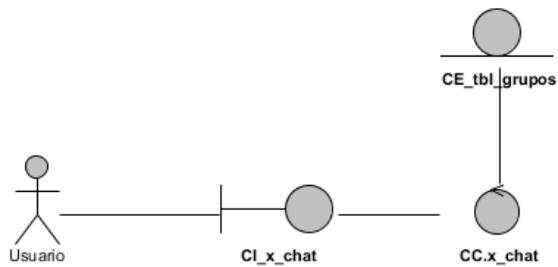


Figura 2. 6: Diagrama de clases del análisis para el caso de uso "Agregar usuarios a un grupo creado" perteneciente al caso de uso "Gestionar grupo de usuarios"

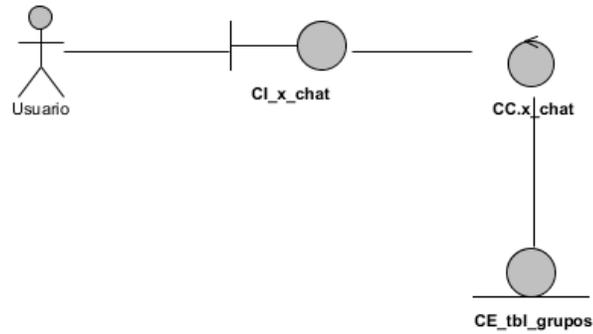


Figura 2. 7: Diagrama de clases del análisis para el caso de uso "Eliminar grupo de usuarios" perteneciente al caso de uso "Gestionar grupo de usuarios"

### Anexo 3. Diagramas de colaboración

A continuación, se muestran los diagramas de colaboración para los casos de uso restantes.

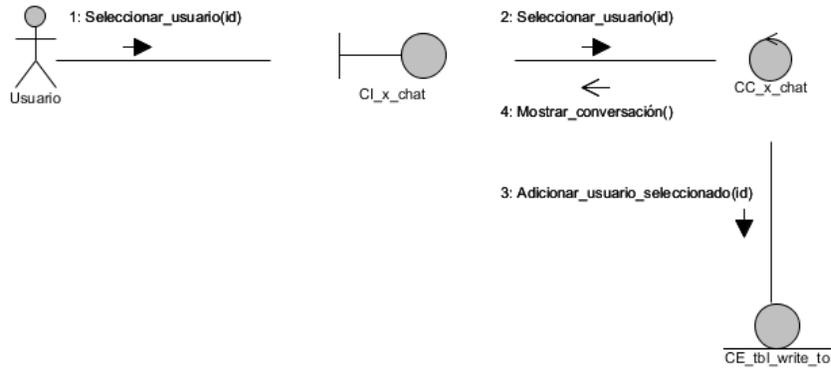


Figura 3. 1: Diagrama de colaboración para el caso de uso "Abrir conversación"

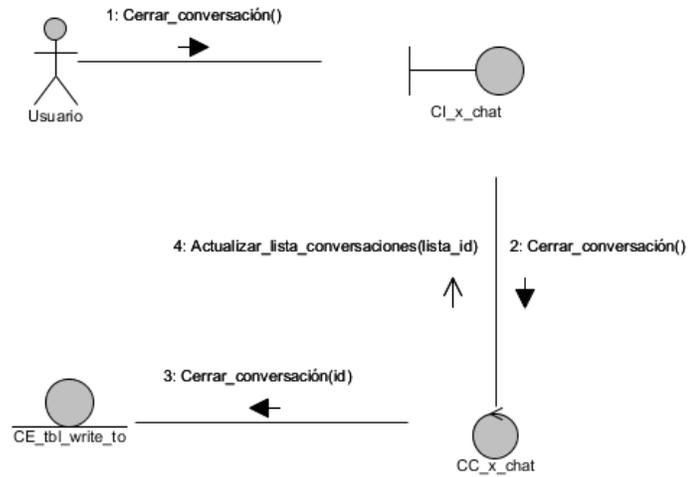


Figura 3. 2: Diagrama de colaboración para el caso de uso "Cerrar conversación"

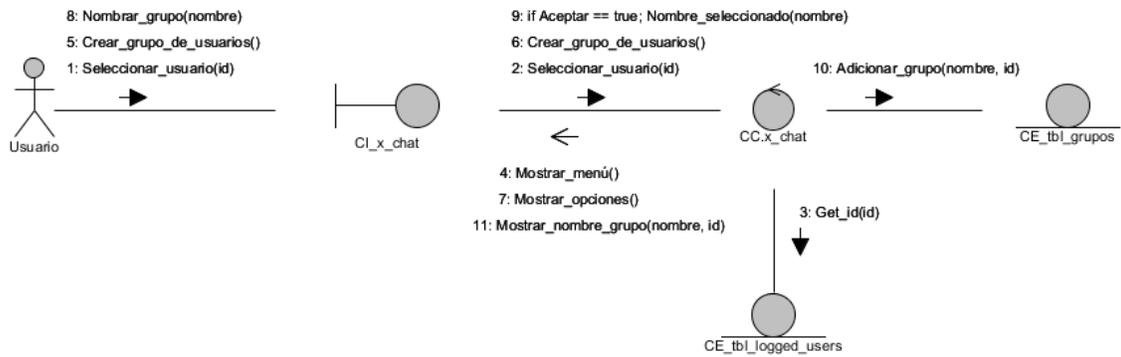


Figura 3. 3: Diagrama de colaboración para el caso de uso "Crear grupo de usuarios" perteneciente al caso de uso "Gestionar grupo de usuarios"

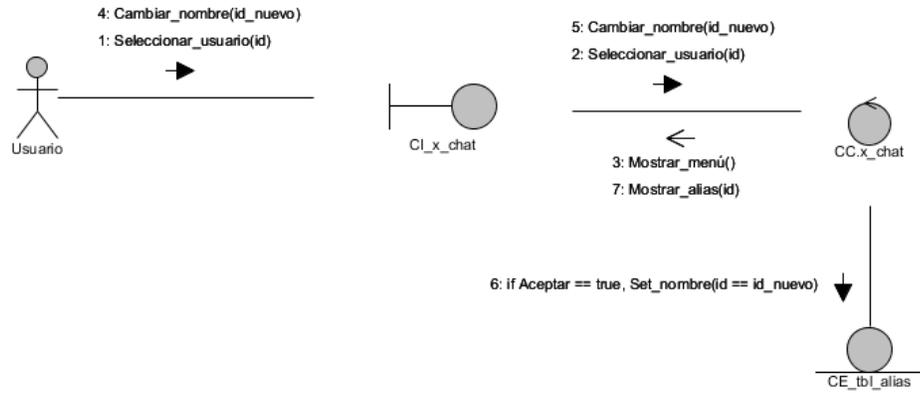


Figura 3. 4: Diagrama de colaboración para el caso de uso "Cambiar nombre de los usuarios"

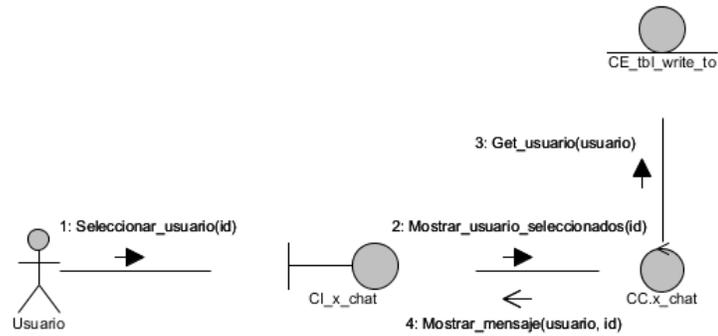


Figura 3. 5: Diagrama de colaboración para el caso de uso "Recibir mensaje"

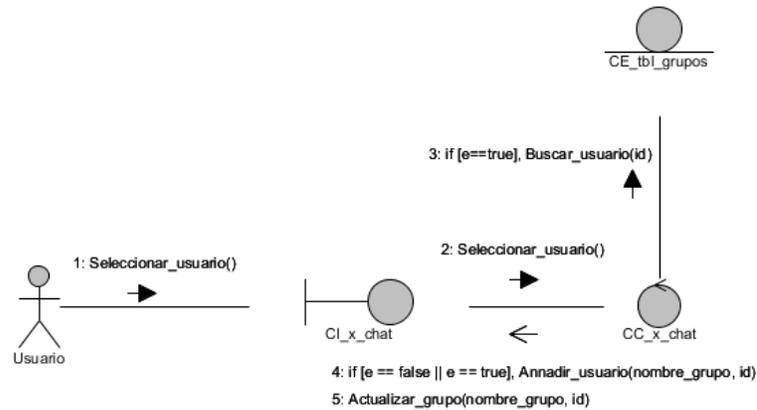


Figura 3. 6: Diagrama de colaboración para el caso de uso "Agregar usuarios a un grupo" perteneciente al caso de uso "Gestionar grupo de usuarios"

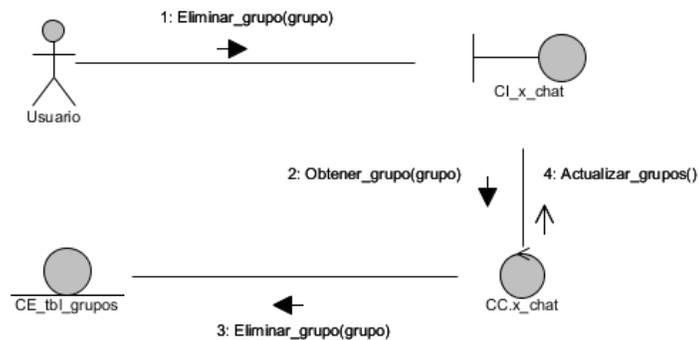


Figura 3. 7: Diagrama de colaboración para el caso de uso "Eliminar grupo de usuarios" perteneciente al caso de uso "Gestionar grupo de usuarios"

## Anexo 4. Diagramas de clases del diseño

A continuación, los diagramas de clases del diseño para los casos de uso restantes.

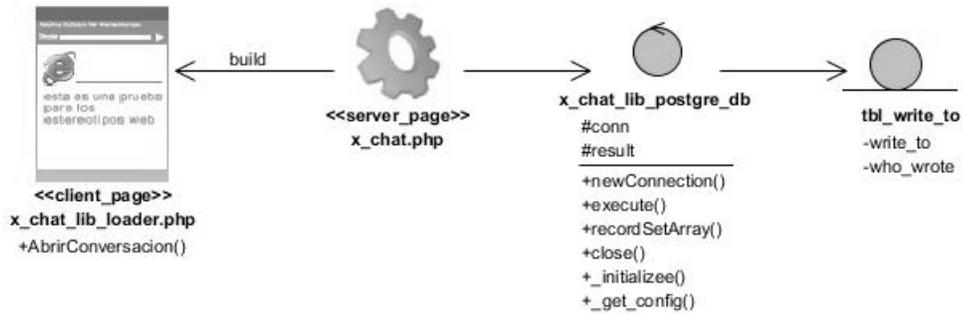


Figura 4. 1: Diagrama de clases del diseño del caso de uso “Abrir conversación”

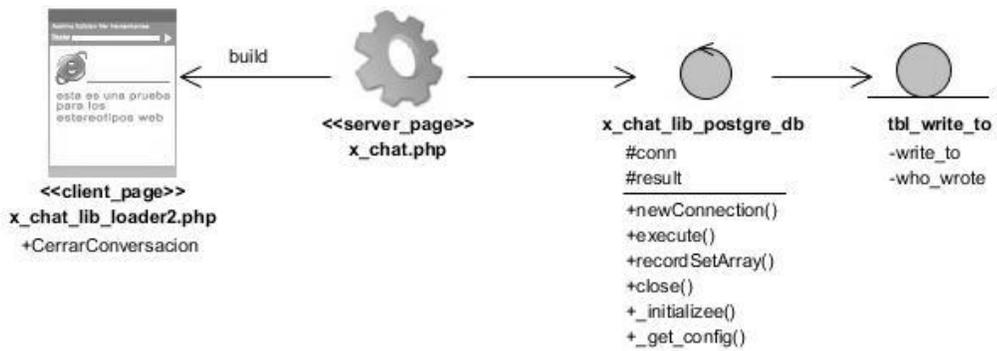


Figura 4. 2: Diagrama de clases del diseño del caso de uso “Cerrar conversación”

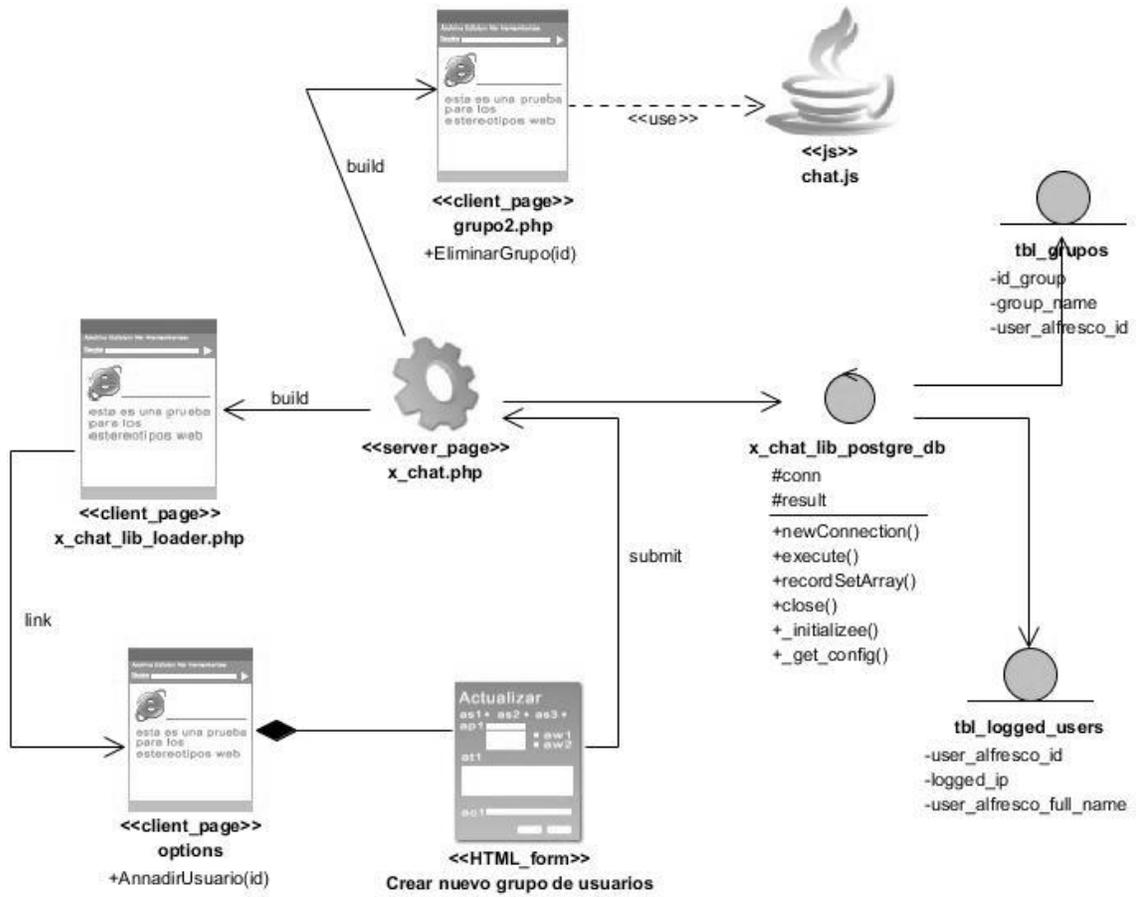


Figura 4. 3: Diagrama de clases del diseño del caso de uso "Gestionar grupo de usuarios"

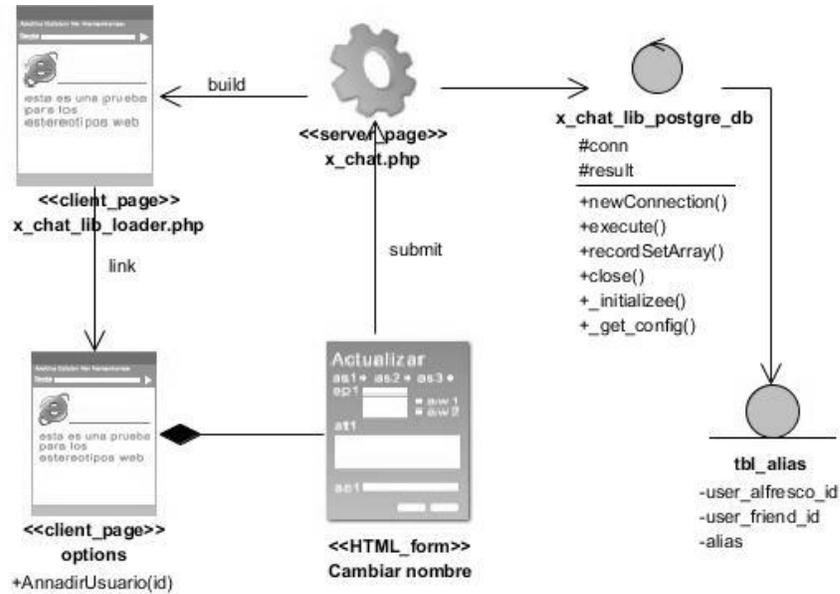


Figura 4. 4: Diagrama de clases del diseño del caso de uso “Cambiar nombre de los usuarios”

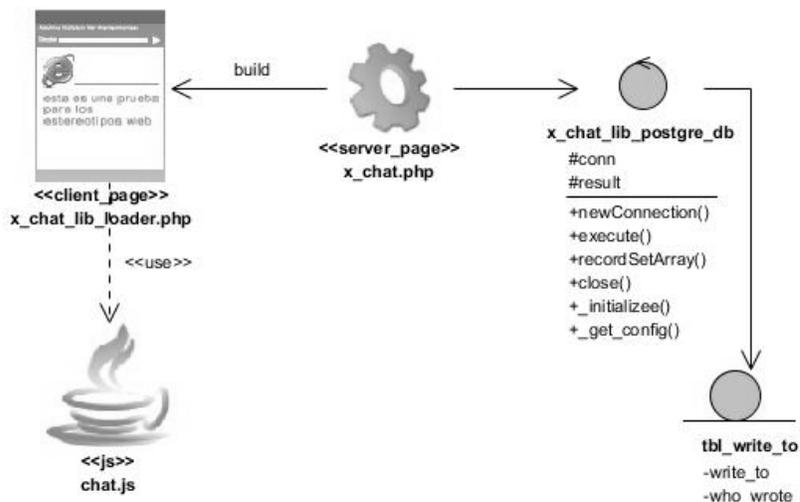


Figura 4. 5: Diagrama de clases del diseño del caso de uso “Recibir mensaje”

## Anexo 5. Descripción del modelo físico de la base de datos

A continuación se muestra la descripción de las tablas y las columnas de cada tabla que pertenecen a la

base de datos.

Tabla 5. 1: Descripción de la tabla "tbl\_friends\_alias"

Nombre de la tabla	Descripción			
tbl_friends_alias	Esta tabla contiene los datos que se vinculan al cambio de nombre de los usuarios.			
Columnas				
Nombre	Tipo de dato	Tipo de llave	Anulable	Descripción
user_alfresco_id	varchar	ninguna	No	Representa el id del usuario el cual interactúa con el módulo (Ejemplo: dabermudez).
user_friend_id	varchar	primaria	No	Representa el id del usuario al cual se le cambiará el nombre (Ejemplo: jazamora).
alias	varchar	ninguna	No	Representa el nombre impuesto por el usuario al usuario seleccionado (Ejemplo: Julio).
tbl_logged_usersuser_alfresco_id	varchar	foránea	No	Representa la llave primaria "user_alfresco_id" de la tabla "tbl_logged_users".

Tabla 5. 2: Descripción de la tabla "tbl\_users\_groups"

Nombre de la tabla	Descripción			
tbl_users_groups	Esta tabla contiene los datos que tiene cada uno de los grupos de usuarios creados.			
Columnas				
Nombre	Tipo de dato	Tipo de llave	Anulable	Descripción
group_id	integer	primaria	No	Representa el id del grupo donde está el usuario (Ejemplo: 1, 2, 3...).
user_alfresco_id	varchar	ninguna	No	Representa el id de cada usuario que pertenece a un grupo.
group_name	varchar	ninguna	No	Representa el nombre que el usuario le ha asignado al grupo.
tbl_logged_usersuser_alfresco_id	varchar	foránea	No	Representa la llave primaria "id_group" de la tabla "tbl_grupos".

Tabla 5. 3: Descripción de la tabla "tbl\_logged\_users"

Nombre de la tabla	Descripción
tbl_logged_users	Esta tabla contiene todos los datos de los usuarios logueados al sistema.
Columnas	

Nombre	Tipo de dato	Tipo de llave	Anulable	Descripción
user_alfresco_id	varchar	primaria	No	Representa el id del usuario que lo utiliza para autenticarse junto con su correspondiente contraseña (Ejemplo: dabermudez).
logged_ip	numeric	única	No	Representa el IP de la computadora donde se encuentra el usuario interactuando con el módulo (Ejemplo: 10.53.9.22).
user_alfresco_full_name	varchar	única	No	Representa el nombre completo que posee el usuario en el sistema (Ejemplo: Dayan Arquímides Bermúdez Cruz).
time	timestamp	ninguna	No	Representa el tiempo real en hora, minutos y segundos.

Tabla 5. 4: Descripción de la tabla "tbl\_conversations"

Nombre de la tabla	Descripción			
tbl_conversations	Esta tabla contiene los datos que conforman una conversación.			
Columnas				
Nombre	Tipo de dato	Tipo de llave	Anulable	Descripción
from_user_id	varchar	primaria	No	Representa el id del usuario que está interactuando con el módulo, es quien envía los mensajes de texto (Ejemplo: dabermudez).
to_user_id	varchar	única	No	Representa el id del usuario a quien se le envían los mensajes de texto (Ejemplo: jazamora).
date	date	única	No	Representa la fecha actual (Ejemplo: 26/06/2013).
time	timestamp	ninguna	No	Representa la hora actual (Ejemplo: 12:35:22 pm).
message	varchar	ninguna	No	Representa el mensaje de texto enviado por el usuario.
tbl_logged_usersuser_alfresco_id	varchar	foránea	No	Representa la llave primaria "user_alfresco_id" de la tabla "tbl_logged_users".

Tabla 5. 5: Descripción de la tabla "tbl\_write\_to"

Nombre de la tabla	Descripción			
tbl_write_to	Esta tabla contiene el nombre de dos usuarios que generan una conversación.			
Columnas				
Nombre	Tipo de dato	Tipo de llave	Anulable	Descripción

write_to	varchar	primaria	No	Representa el nombre del usuario receptor de los mensajes enviados (Ejemplo: Julio Antonio Zamora Rosabal).
who_wrote	varchar	primaria	No	Representa el nombre del usuario el cual envía mensajes de texto (Ejemplo: Dayan Arquímedes Bermúdez Cruz).
tbl_logged_usersuser_alfresco_id	varchar	foránea	No	Representa la llave primaria "user_alfresco_id" de la tabla "tbl_logged_users".

Tabla 5. 6: Descripción de la tabla "tbl\_friends\_group"

Nombre de la tabla	Descripción			
tbl_friends_group	Esta tabla contiene los datos de los usuarios que pertenecen a un grupo.			
Columnas				
Nombre	Tipo de dato	Tipo de llave	Anulable	Descripción
user_alfresco_id	varchar	única	No	Representa el id del usuario que lo utiliza para autenticarse junto con su correspondiente contraseña (Ejemplo: dabermudez).
user_friend_id	varchar	única	No	Representa el id del usuario que presenta el nombre original cambiado.
group_id	integer	primaria	No	Representa el id del grupo al que pertenece el usuario (Ejemplo: 1, 2, 3...).
tbl_logged_usersuser_alfresco_id	varchar	foránea	No	Representa la llave primaria "user_alfresco_id" de la tabla "tbl_logged_users".
tbl_users_groupsgroup_id	integer	foránea	No	Representa la llave primaria "group_id" de la tabla "tbl_users_groups".

## Anexo 6. Casos de prueba para las historias de usuario

A continuación, los casos de prueba restantes para las historias de usuario.

### Caso de prueba. HU Cerrar conversación

Descripción de la funcionalidad: Permite al usuario poder cerrar las conversaciones.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos una conversación abierta para poder cerrarla.

Flujo central: El usuario una vez que haya accedido al módulo y haya abierto una o varias conversaciones por pestañas, se le muestra en cada pestaña el botón Cerrar.

Escenario: Cerrar conversación.

Tabla 6. 1: Caso de prueba. HU Cerrar conversación

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Mostrar conversaciones por pestañas”, selecciona el id “hmastrapa” del usuario “Hilario Mastrapa Montero” y cierra la conversación.	El módulo actualiza las pestañas con las demás conversaciones, excluyendo la conversación con el id “hmastrapa”.	Satisfactoria	
El usuario estando en la interfaz “Mostrar conversaciones por pestañas”, selecciona el id “dabermudez” del usuario “Dayan Bermúdez Cruz” y cierra la conversación.	El módulo actualiza las pestañas con las demás conversaciones, excluyendo la conversación con el id “dabermudez”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
<b>Evaluación de la prueba</b>	Satisfactoria		

### Caso de prueba. HU Crear grupos de usuarios

Descripción de la funcionalidad: Permite al usuario crear grupos de usuarios dentro del módulo.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un usuario conectado al módulo para iniciar un grupo de usuarios.

Flujo central: Una vez que el usuario haya accedido al módulo puede seleccionar un usuario y dar clic derecho encima de él. A este se le muestra un menú con varias opciones de las que escoge “Grupo de usuarios” y más adelante escoge de un submenú la opción “Crear nuevo grupo”.

Escenario: Crear grupo de usuarios.

Tabla 6. 2: Caso de prueba. HU Crear grupo de usuarios

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Mostrar usuarios conectados” le da clic derecho encima del usuario “Hilario Mastrapa Montero”, escoge la opción “Grupo de usuarios” y luego la opción “Crear nuevo grupo”.	El módulo muestra la nueva interfaz titulada “Nombre del grupo” con los botones “Aceptar” y “Cancelar”.	Satisfactoria	
El usuario estando en la interfaz “Nombre del grupo” especifica el nombre “Estudiantes” y presiona el botón “Aceptar”.	El módulo muestra el grupo creado con el nombre “Estudiantes” especificado por el usuario. El módulo muestra la interfaz “Crear grupos de usuarios”.	Satisfactoria	
El usuario estando en la interfaz “Mostrar usuarios conectados” le da clic derecho encima del usuario “Julio Antonio Zamora Rosabal”, escoge la opción “Grupo de usuarios” y luego la opción “Crear nuevo grupo”.	El módulo muestra la nueva interfaz titulada “Nombre del grupo” con los botones “Aceptar” y “Cancelar”.	Satisfactoria	

El usuario estando en la interfaz “Nombre del grupo” especifica el nombre “Profesores” y presiona el botón “Aceptar”.	El módulo muestra el grupo creado con el nombre “Profesores” especificado por el usuario. El módulo muestra la interfaz “Crear grupos de usuarios”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Nombre del grupo” no especifica un nombre y presiona el botón “Aceptar”.	El sistema muestra un mensaje de alerta: “Por favor especifique un nombre”.	Satisfactoria	
El usuario estando en la interfaz “Nombre del grupo” presiona el botón “Cancelar”.	El módulo no crea el grupo de usuarios.	Satisfactoria	
El usuario estando en la interfaz “Nombre del grupo” especifica el nombre “Estudiantes” o “Profesores” y presiona el botón “Cancelar”.	El módulo no crea el grupo de usuarios.	Satisfactoria	
<b>Evaluación de la prueba</b>	Satisfactoria		

### Caso de prueba. HU Cambiar nombre de los usuarios

Descripción de la funcionalidad: Permite al usuario cambiar el nombre de los usuarios conectados en el módulo.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un usuario conectado al módulo para cambiarle el nombre.

Flujo central: Una vez que el usuario haya accedido al módulo puede seleccionar un usuario y dar clic derecho encima de él. A este se le muestra un menú con varias opciones de las que escoge “Cambiar nombre”.

Escenario: Cambiar nombre.

Tabla 6. 3: Caso de prueba. HU Cambiar nombre de los usuarios

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Mostrar usuarios conectados” le da clic derecho encima del usuario “Hilario Mastrapa Montero”, escoge la opción “Cambiar nombre”.	El módulo muestra la nueva interfaz titulada “Cambiar nombre” con los botones “Aceptar” y “Cancelar”. Además tiene el nombre del usuario “Hilario Mastrapa Montero” focalizado.	Satisfactoria	
El usuario estando en la interfaz “Cambiar nombre” especifica el nombre “Hilario” y presiona el botón “Aceptar”.	El módulo muestra el nombre “Hilario” cambiado por el nombre del usuario “Hilario Mastrapa Montero”.	Satisfactoria	
El usuario estando en la interfaz “Mostrar usuarios conectados” le da clic derecho encima del usuario “Dayan Bermúdez Cruz”, escoge la opción “Cambiar nombre”.	El módulo muestra la nueva interfaz titulada “Cambiar nombre” con los botones “Aceptar” y “Cancelar”. Además tiene el nombre del usuario “Dayan Bermúdez Cruz” focalizado.	Satisfactoria	
El usuario estando en la interfaz “Cambiar nombre” especifica el nombre “Dayan” y presiona el botón “Aceptar”.	El módulo muestra el nombre “Dayan” cambiado por el nombre del usuario “Dayan Bermúdez Cruz”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Cambiar nombre” no le especifica un nombre al usuario “Dayan Bermúdez Cruz” o “Hilario Mastrapa Montero” y presiona el botón “Aceptar”.	El módulo muestra un mensaje de alerta: “No se ha cambiado el nombre”. El módulo no muestra ningún cambio realizado sobre el usuario seleccionado, no le ha cambiado el nombre.	Satisfactoria	
El usuario estando en la interfaz “Cambiar nombre” no le especifica un nombre al usuario “Dayan Bermúdez Cruz” o “Hilario Mastrapa Montero” y presiona el botón “Cancelar”.	El módulo no muestra ningún cambio realizado sobre el usuario seleccionado, no le ha cambiado el nombre.	Satisfactoria	

El usuario estando en la interfaz “Cambiar nombre” le especifica el nombre “Dayan” al usuario “Dayan Bermúdez Cruz” y presiona el botón “Cancelar”.	El módulo no muestra ningún cambio realizado sobre el usuario seleccionado, no le ha cambiado el nombre.	Satisfactoria	
El usuario estando en la interfaz “Cambiar nombre” le especifica el nombre “Hilario” al usuario “Hilario Mastrapa Montero” y presiona el botón “Cancelar”.	El módulo no muestra ningún cambio realizado sobre el usuario seleccionado, ya que no le ha cambiado el nombre.	Satisfactoria	
<b>Evaluación de la prueba</b>	<b>Satisfactoria</b>		

### Caso de prueba. HU Enviar mensaje

Descripción de la funcionalidad: Permite al usuario enviar mensajes de texto a los usuarios conectados al módulo.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un usuario conectado al módulo y seleccionado para enviarle mensajes.

Flujo central: Una vez que el usuario haya accedido al módulo, seleccionado un usuario y esté abierta una pestaña con el id del usuario seleccionado, ya puede enviar mensajes de texto al usuario seleccionado.

Escenario: Enviar mensaje.

Tabla 6. 4: Caso de prueba. HU Enviar mensaje

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
----------------	--------------------	------------------------	---------------

El usuario estando en la interfaz “Abrir conversaciones por pestañas” le envía uno o varios mensajes de texto al usuario con el id “dabermudez”.	El módulo muestra el mensaje o los mensajes enviados por el usuario a la izquierda del mismo. El módulo muestra la interfaz “Mostrar conversaciones por pestañas”.	Satisfactoria	
El usuario estando en la interfaz “Abrir conversaciones por pestañas” le envía uno o varios mensajes de texto al usuario con el id “hmastrapa”.	El módulo muestra el mensaje o los mensajes enviados por el usuario a la izquierda del mismo. El módulo muestra la interfaz “Mostrar conversaciones por pestañas”.	Satisfactoria	
El usuario estando en la interfaz “Abrir conversaciones por pestañas” le envía uno o varios mensajes de texto al usuario con el id “jazamora”.	El módulo muestra el mensaje o los mensajes enviados por el usuario a la izquierda del mismo. El módulo muestra la interfaz “Mostrar conversaciones por pestañas”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
<b>Evaluación de la prueba</b>	<b>Satisfactoria</b>		

### Caso de prueba. HU Recibir mensaje

Descripción de la funcionalidad: Permite al usuario recibir mensajes de texto de los usuarios conectados al módulo.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un usuario conectado al módulo y parpadeando en rojo con el color de fondo amarillo.

Flujo central: Una vez que el usuario haya accedido al módulo y este muestre uno o varios usuarios

parpadeando puede recibir los mensajes enviados.

Escenario: Recibir mensaje.

Tabla 6. 5: Caso de prueba. HU Recibir mensaje

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Mostrar usuarios conectados” le parpadea el usuario “Dayan Bermúdez Cruz”.	El módulo muestra el usuario “Dayan Bermúdez Cruz” parpadeando.	Satisfactoria	
El usuario selecciona el usuario “Dayan Bermúdez Cruz” que está parpadeando.	El módulo muestra en la interfaz “Mostrar conversaciones por pestañas” una pestaña identificada con el id “dabermudez” y con el mensaje de texto enviado por el usuario “Dayan Bermúdez Cruz”.	Satisfactoria	
El usuario estando en la interfaz “Mostrar usuarios conectados” le parpadea el usuario “Hilario Mastrapa Montero”.	El módulo muestra el usuario “Hilario Mastrapa Montero” parpadeando.	Satisfactoria	
El usuario selecciona el usuario “Hilario Mastrapa Montero” que está parpadeando.	El módulo muestra en la interfaz “Mostrar conversaciones por pestañas” una pestaña identificada con el id “hmastrapa” y con el mensaje de texto enviado por el usuario “Hilario Mastrapa Montero”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
<b>Evaluación de la prueba</b>	Satisfactoria		

### Caso de prueba. HU Agregar usuarios a un grupo creado

Descripción de la funcionalidad: Permite al usuario agregar usuarios a un grupo de usuarios ya creado

previamente.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un grupo de usuarios creado y al menos un usuario conectado.

Flujo central: Una vez que el usuario haya accedido al módulo y tenga algún grupo de usuarios creado puede agregarle usuarios a dicho grupo.

Escenario: Agregar usuarios.

*Tabla 6. 6: Caso de prueba. HU Agregar usuarios a un grupo creado*

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Crear grupos de usuarios” le da clic derecho encima del usuario “Dayan Bermúdez Cruz”, escoge la opción “Grupo de usuarios” y luego el nombre del grupo de usuarios nombrado “Estudiantes”.	El módulo muestra el usuario “Dayan Bermúdez Cruz” dentro del grupo de usuarios “Estudiantes”.	Satisfactoria	
El usuario estando en la interfaz “Crear grupos de usuarios” le da clic derecho encima del usuario “Julio Antonio Zamora Rosabal”, escoge la opción “Grupo de usuarios” y luego el nombre del grupo de usuarios nombrado “Profesores”.	El módulo muestra el usuario “Julio Antonio Zamora Rosabal” dentro del grupo de usuarios “Profesores”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones

Evaluación de la prueba	Satisfactoria
-------------------------	---------------

**Caso de prueba. HU Eliminar grupo de usuarios**

Descripción de la funcionalidad: Permite al usuario borrar un grupo de usuarios ya creado.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un grupo de usuarios creado.

Flujo central: Una vez que el usuario haya accedido al módulo y tenga algún grupo de usuarios creado tiene la posibilidad de borrarlo. El usuario puede borrar el grupo de usuarios, pero no puede borrar los usuarios que tiene el grupo.

Escenario: Eliminar grupo.

*Tabla 6. 7: Caso de prueba. HU Eliminar grupo de usuarios*

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Crear grupos de usuarios” le da clic derecho encima del grupo de usuarios “Estudiantes” y escoge la opción “Eliminar grupo”.	El módulo muestra un mensaje de alerta: “¿Desea borrar el grupo Estudiantes?” con los botones “Aceptar” y “Cancelar”.	Satisfactoria	
El usuario presiona el botón “Aceptar”.	El módulo no muestra el grupo de usuarios “Estudiantes”, se verifica que el grupo ha sido borrado.	Satisfactoria	

El usuario estando en la interfaz “Crear grupos de usuarios” le da clic derecho encima del grupo de usuarios “Profesores” y escoge la opción “Eliminar grupo”.	El módulo muestra un mensaje de alerta: “¿Desea borrar el grupo Profesores?” con los botones “Aceptar” y “Cancelar”.	Satisfactoria	
El usuario presiona el botón “Aceptar”.	El módulo no muestra el grupo de usuarios “Profesores”, lo que significa que el grupo ha sido borrado.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario presiona el botón “Cancelar”.	El grupo de usuarios “Estudiantes” o “Profesores” persiste en el módulo.	Satisfactoria	
<b>Evaluación de la prueba</b>	Satisfactoria		

### Caso de prueba. HU Abrir conversación

Descripción de la funcionalidad: Permite al usuario poder abrir conversaciones nuevas dentro del módulo.

Condiciones de ejecución:

- El usuario debe estar autenticado en el GDA eXcriba 2.0.
- El usuario debe acceder al módulo.
- El usuario debe tener al menos un usuario conectado en el módulo.

Flujo central: El usuario una vez que accede al módulo, selecciona uno o varios usuarios de la lista de usuarios conectados, esta se encuentra en la parte derecha del módulo. Luego de realizar esta acción, en la parte izquierda del módulo aparecen una o varias pestañas con el id del nombre de los usuarios seleccionados previamente.

Escenario: Abrir conversación.

Tabla 6. 8: Caso de prueba. HU Abrir conversación

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario estando en la interfaz “Usuarios conectados”, selecciona el nombre del usuario “Dayan Bermúdez Cruz” ubicado en la parte derecha del módulo junto con los demás usuarios que aparecen conectados.	El módulo muestra dentro de la misma interfaz, pero a la izquierda, una pestaña que contiene el id “dabermudez” correspondiente al nombre del usuario seleccionado. Este ofrece la nueva interfaz “Abrir conversaciones por pestañas”.	Satisfactoria	
El usuario estando en la interfaz “Usuarios conectados”, selecciona el nombre del usuario “Hilario Mastrapa Montero” ubicado en la parte derecha del módulo junto con los demás usuarios que aparecen conectados.	El módulo muestra dentro de la misma interfaz, pero a la izquierda, una pestaña que contiene el id “hmastrapa” correspondiente al nombre del usuario seleccionado. Este ofrece la nueva interfaz “Abrir conversaciones por pestañas”.	Satisfactoria	
El usuario estando en la interfaz “Usuarios conectados”, selecciona el nombre del usuario “Julio Antonio Zamora Rosabal” ubicado en la parte derecha del módulo junto con los demás usuarios que aparecen conectados.	El módulo muestra dentro de la misma interfaz, pero a la izquierda, una pestaña que contiene el id “jazamora” correspondiente al nombre del usuario seleccionado. Este ofrece la nueva interfaz “Abrir conversaciones por pestañas”.	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
<b>Evaluación de la prueba</b>	Satisfactoria		