**Proyecto de Mensajería Segura: BirBeep**

Nombre del centro: IES Gaspar Melchor de Jovellanos

Nombre del ciclo: Desarrollo de aplicaciones multiplataforma

Título: Aplicación de mensajería instantánea

Autores: Rubén García, Sebastián Faranna y Luis Miguel Sanchez-Clemente Batres

Profesor coordinador: Miguel ángel Herrera Muñoz

Índice

1. Agradecimientos……………………………………………………
2. Resumen……………………………………………………………….
3. Datos de la aplicación……………………………………………
   1. Datos técnicos…………………………………………….
   2. Funcionalidad……………………………………………..
   3. Usabilidad……………………………………………………
4. Bibliografía…………………………………………………………….

1.Agradecimientos

2. Resumen

La información a día de hoy es uno de los activos más importantes que las empresas necesitan proteger. Por otra parte, para resolver los diferentes proyectos, es cada vez más habitual disponer de equipos multidisciplinares dispersos geográficamente, que obviamente precisan de mecanismos que les posibiliten una comunicación fluida. Por todo ello, un sistema de mensajería que comunique al personal de la organización de manera confidencial, garantizando al mismo tiempo la autenticación de emisores y receptores de la información, aportaría sin duda un valioso servicio a la empresa; es lo que vamos a desarrollar en este proyecto. Se utilizará una arquitectura cliente-servidor para resolver el sistema:

• El servidor actuará de intermediario en la distribución de cada mensaje.

• Los clientes serán los remitentes o destinatarios de cada comunicación, y se utilizarán sobres digitales para asegurar la autenticación de emisor, la confidencialidad, la integridad y el no repudio.

3. Datos de la aplicación

a)Datos técnicos

* **Nombre de la aplicación**

El nombre de la aplicación es Birbeep.

* **Plataforma**

El lenguaje elegido es Java, ya que es el que más hemos utilizado en clase.

La plataforma que utilizamos es Windows. Para usar la aplicación debemos tener instalado JRE( Java Runtime Enviroment) para que puedan correr los programascreados con lenguaje Java.

El servidor corre bajo el JRE, que es un entorno en tiempo de ejecución de Java creado por la Máquina Virtual de Java (JVM). La Máquina Virtual de Java es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java. El JRE actúa como mediador entre el sistema operativo y Java.

Las aplicaciones que han sido utilizadas para desarrollar la aplicación son Eclipse, para el desarrollo del código de aplicación junto con el servidor e interfaz gráfica de usuario, y phpMyAdmin, para gestionar la base de datos.

Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java DevelopmentToolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Le hemos utilizado mayormente para la creación del código de la aplicación y del servidor.

PhpMyAdmines una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas.

* **Requisitos mínimo del sistema**

Estos serían los requisitos mínimos para lazar esta aplicación:

Windows:

* + Windows:Windows 10 (8u51 y superiores), Windows 8.x (escritorio), Windows 7 SP1, Windows Vista SP2, Windows Server 2008 R2 SP1 (64 bits) o Windows Server 2012 y 2012 R2 (64 bits)
  + RAM: 128 MB
  + Espacio en disco: 124 MB para JRE y 2 MB para Java Update
  + Procesador: Mínimo Pentium 2 a 266 MHz
  + Exploradores: Internet Explorer 9 y superior, Firefox

Linux:

* + Linux:Oracle Linux 5.5+1, Oracle Linux 6.x (32 bits), 6.x (64 bits)2, Oracle Linux 7.x (64 bits)2 (8u20 y superiores), Red Hat Enterprise Linux 5.5+1, 6.x (32 bits), 6.x (64 bits)2, Red Hat Enterprise Linux 7.x (64 bits)2 (8u20 y superiores), Suse Linux Enterprise Server 10 SP2+, 11.x, Suse Linux Enterprise Server 12.x (64 bits)2 (8u31 y superiores), Ubuntu Linux 12.04 LTS, 13.x, Ubuntu Linux 14.x (8u25 y superiores), Ubuntu Linux 15.04 (8u45 y superiores) o Ubuntu Linux 15.10 (8u65 y superiores)
  + Exploradores: Firefox 3.6 y posterior

Mac OS X:

* + Mac con Intel que ejecuta Mac OS X 10.7.3+, 10.8.3+, 10.9+
  + Privilegios de administrador para la instalación
  + Explorador de 64 bits
* **Manual de instalación**

b)Funcionalidad

Las funcionalidades son:

* El uso de certificados digitales, que es un fichero informático con una estructura de datos que contienen información sobre una entidad, firmado por otra entidad, considerada por una colección de otras entidades como una autoridad para este tipo de contenido.

Se generan mediante la herramienta keytool. Tiene diferentes pasos:

1. Creamos el certificado del Servidor con su almacén de certificados correspondiente.
2. Creamos el certificado del primer cliente con su almacén correspondiente.
3. Exportamos y metemos el certificado del cliente en el almacén de certificados de confianza del Servidor.
4. Se repiten los pasos 2 y 3 para cada cliente

* La utilización del protocolo SSL, que es un protocolo criptográfico, que proporciona comunicaciones por una red.

Consiste en el uso de criptología asimétrica para autentificar a la contraparte con quien se están comunicando y para intercambiar una llave simétrica. para autentificar a la contraparte con quien se están comunicando y para intercambiar una llave simétrica. Esta sesión es luego usada para cifrar el flujo de datos entre los dos partes. Esto permite la confidencialidad del mensaje, códigos de autentificación de mensajes para integridad y como producto lateral, autentificación del mensaje.

Lo utilizamos para

* Utilizamos los sobres digitales, que se basa en que su cifrado y su descifrado utilizan la clave pública y la clave privada. Lo usamos para el envío de mensajes y en la seguridad en la aplicación.

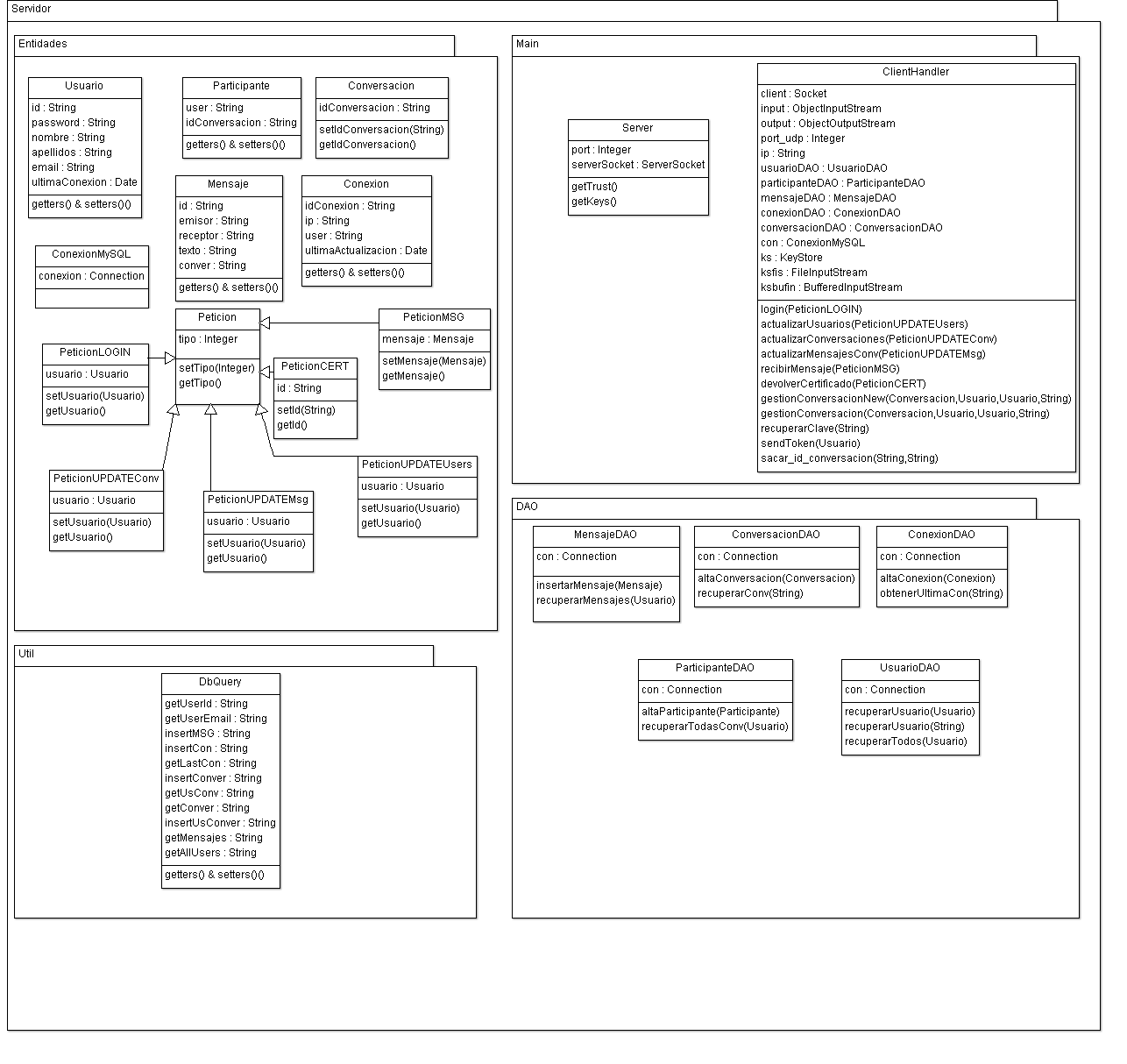
Las fases para cifrar son:

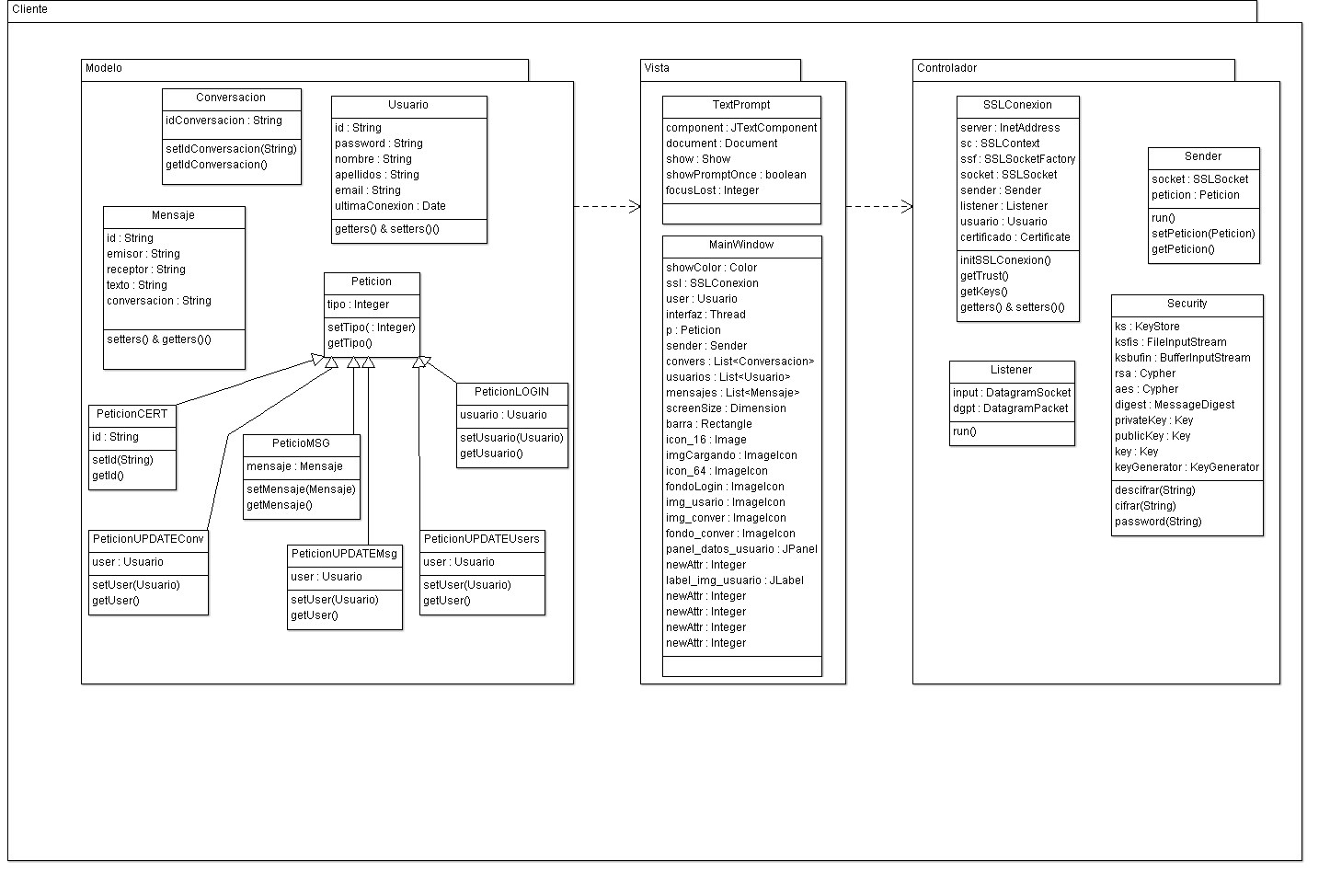
1. Se calcula el hash con el mensaje
2. Se obtiene la clave privada del emisor, que se encuentra en el diccionario de certificados
3. A continuación, se firma el hash con la clave privada, para autentificar que ese mensaje está escrito por el emisor y que no ha sido modificado en la trayectoria de envío.
4. Se genera la clave AES para hacer el cifrado simétrico y se cifra el mensaje con esa clave.
5. Obtenemos la suma del mensaje cifrado con sus hash cifrado. Después cogemos la clave pública del receptor y ciframos la clave generada
6. Obtenemos el sobre firmado

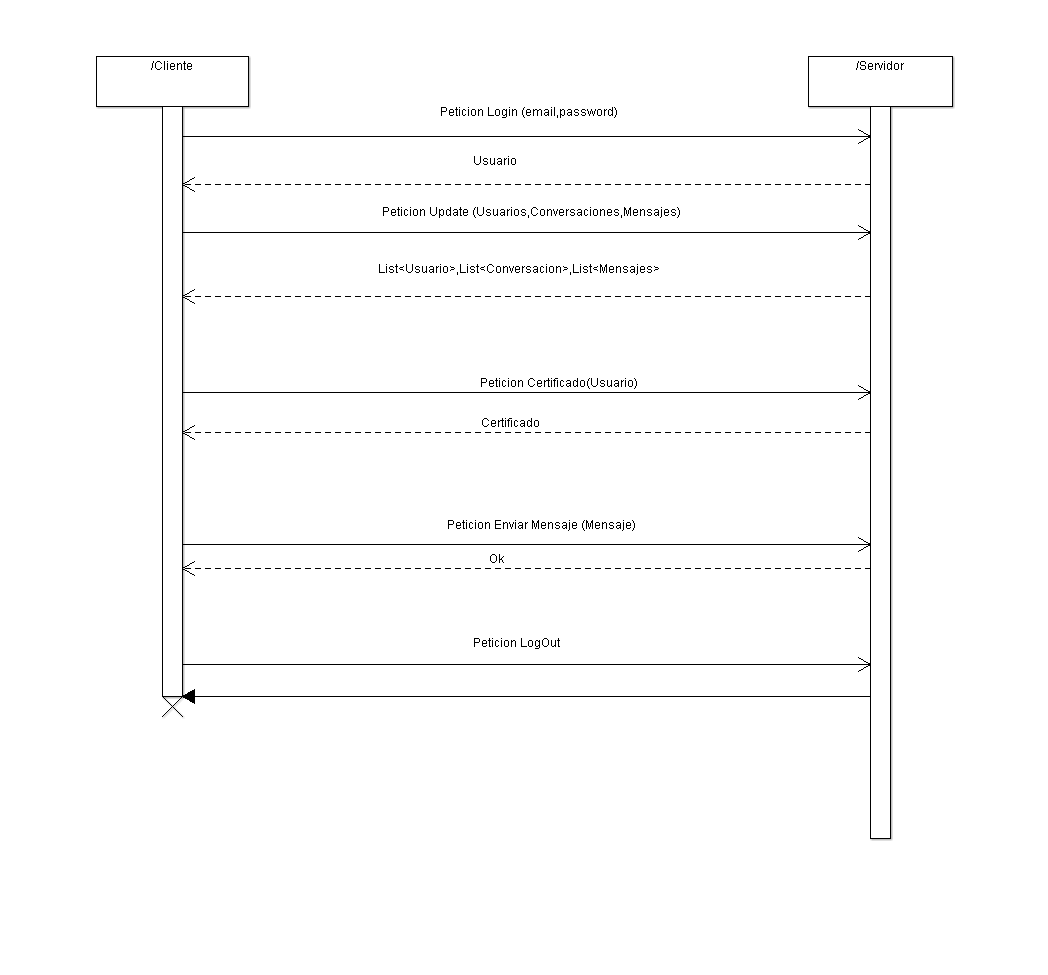
Las fases de descifrado son:

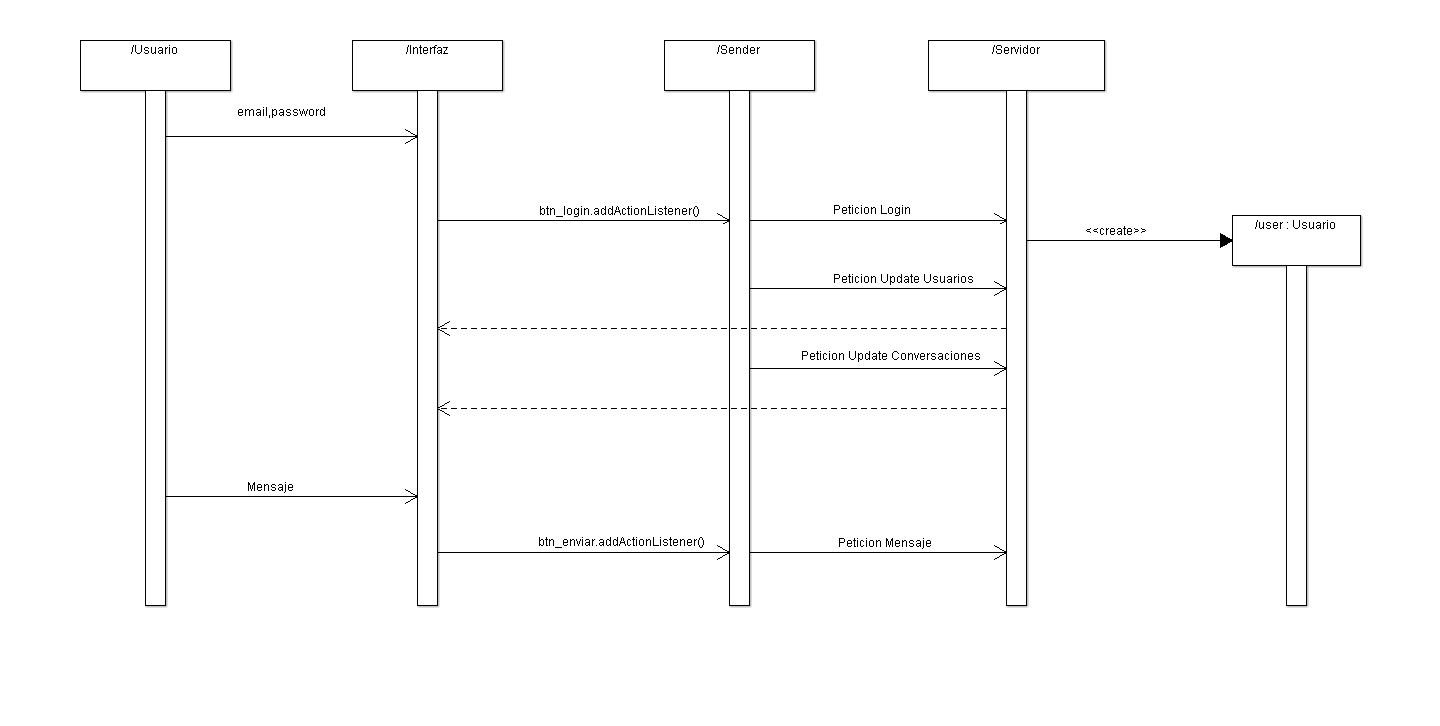
1. Obtenemos el sobre digital enviado por el emisor del mensaje. Este sobre contiene el mensaje encriptado, el hash firmado y la clave encriptada.
2. Obtenemos la clave privada del receptor y desciframos la clave simétrica y se obtiene la clave pública del emisor
3. Desciframos el mensaje y calculamos el hash del mensaje
4. Comparamos el hash generado con el que viene del sobre digital para saber si ha sido modificado el mensaje.

* El servidor corre en la máquina virtual de Java. Cada vez que se inicia la aplicación y se entra en la aplicación, se crea una conexión SSL con el servidor que devuelve un token UDP para saber que ya está conectado. Si no se consigue conectar con el servidor se envía un mensaje de error.

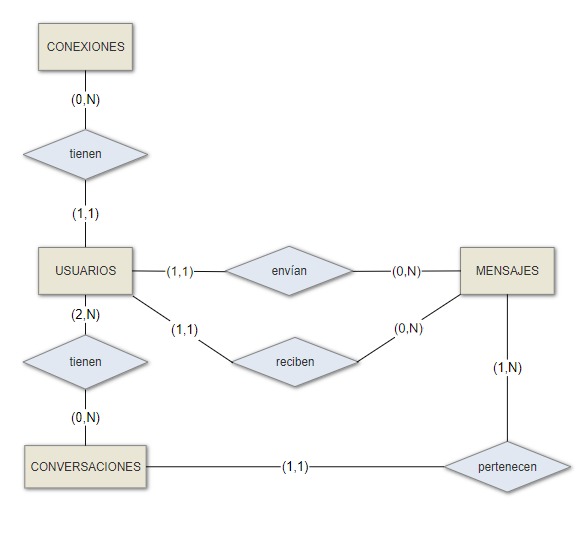


* El cliente puede hacer diferentes peticiones al servidor, que son actualizarse, actualizar mensajes, actualizar conversaciones, actualizar usuarios, petición mensaje, pedir un certificado, login y logout. Para el uso de estas peticiones tenemos que deserializar los datos que nos devuelven, el cual utilizamos flexjson-2.1.
  + La petición actualizarse es llamada cuando el servidor envía un token UDP, que avisa que hay nuevos mensajes para el usuario, y hace que se actualice.
  + La petición actualizar mensajerecupera todos los mensajes de una conversación.
  + La petición actualizar conversaciones recupera todas las conversaciones en las que el usuario ha sido participante.
  + La petición actualizar usuarios recupera todos los usuarios que están dados de alta en la aplicación.
  + La petición mensaje es la que hace el envío al receptor y al mismo emisor y el guardado del mensaje en la base de datos. El envío del mensaje al mismo emisor es para saber qué mensajes envía.
  + La petición pedir un certificado se llama cuando tenemos que usar la clave pública de un usuario para cifrar un mensaje.
  + La petición login se usa cuando inicia la aplicación porque es la que comprueba si existe ese usuario con esa contraseña en la base de datos. Si ese usuario está en la base de datos, el servidor crea un hilo para que se encargue de ese usuario hasta que cierre la aplicación.
  + La petición logout se usa cuando se cierra la aplicación. Cierra el hilo que nos asigna el servidor ,para estar en contacto con él.
* Hay un flujo de información entre el cliente y el servidor. Este flujo está basado en peticiones, que han sido explicadas en el punto previo, y en la serialización. La serialización consiste en un proceso de codificación de un objeto en un medio de con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible, que en nuestro caso en JSON.Cada petición devuelve un objeto, que hay que deserializar para poder usarlo.

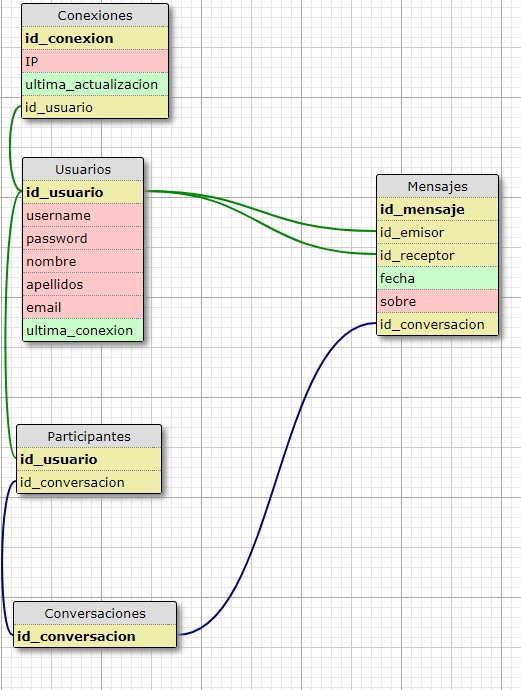




* Modelo de E/R de la base de datos



* Diagrama relacional de la base de datos



Para usar las funcionalidades de la aplicación, se tendrá que estar registrado en el servidor con un correo y una contraseña.

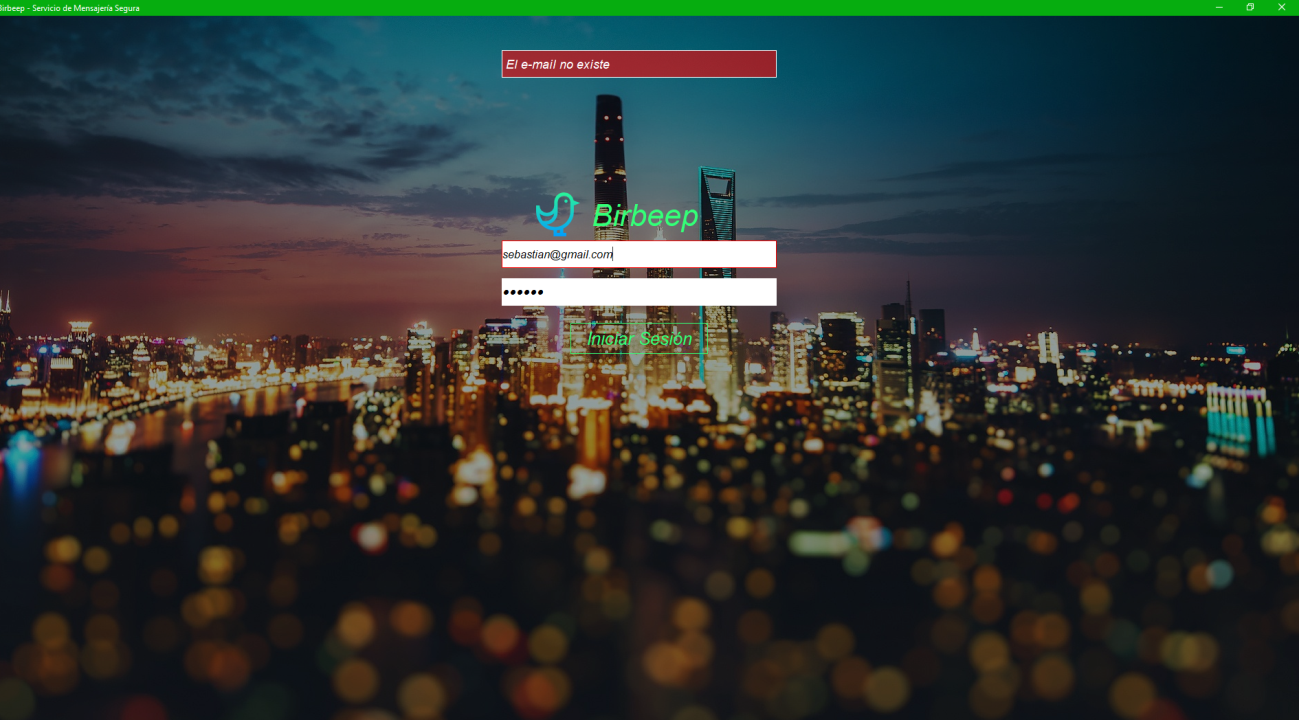
El servidor almacena en la base de datos los siguientes datos, que son el nombre, los apellidos, la ip desde donde se conecta, una id para cada usuario y el email que le sirve para entrar a la aplicación junto con una contraseña, que también es guardada en la base de datos.

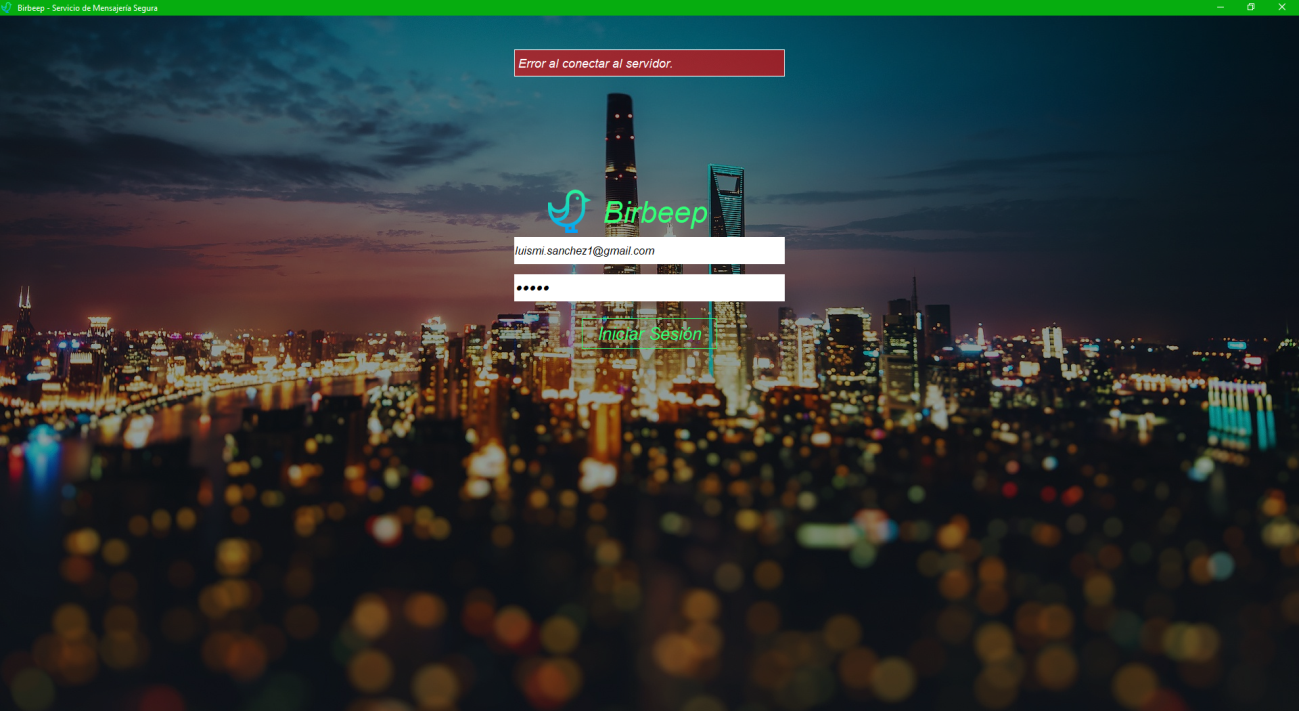
c) Usabilidad

Esta sería la pantalla del inicio de sesión. 

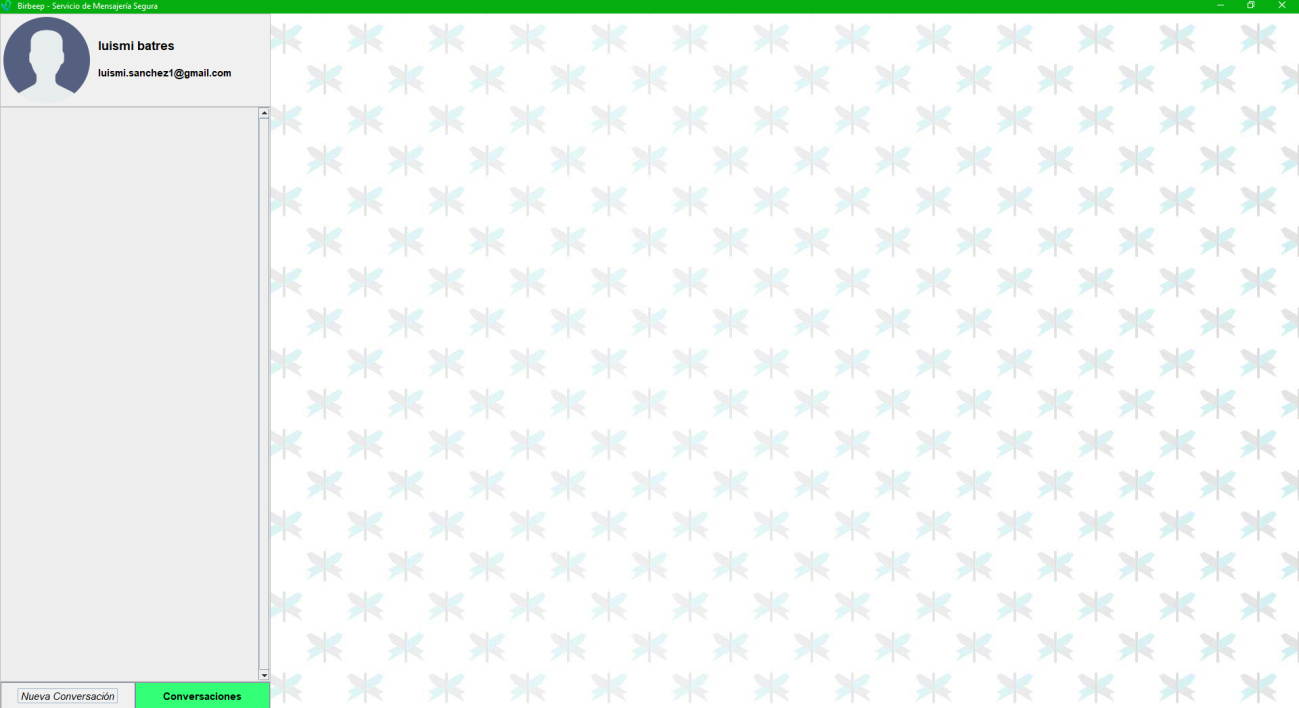
El inicio de sesion comprueba si faltan algun campo por rellenar, comprueba si existe ese email y esa contraseña en la base de datos, y por ultima, comprueba si esta conectado el servidor







Esta sería la pantalla principal de la aplicación.



6.bibliografia

<https://es.wikipedia.org/wiki/Eclipse_(software)>

<https://es.wikipedia.org/wiki/PhpMyAdmin>

<https://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java>

<https://www.java.com/es/download/help/sysreq.xml>