

## Evaluación del Proyecto – Presentación

## SISTEMA DE MENSAJERÍA INSTANTÁNEA DEL EJÉRCITO ARGENTINO (SiMIDEA)

CT HORACIO GERMÁN FUENTES

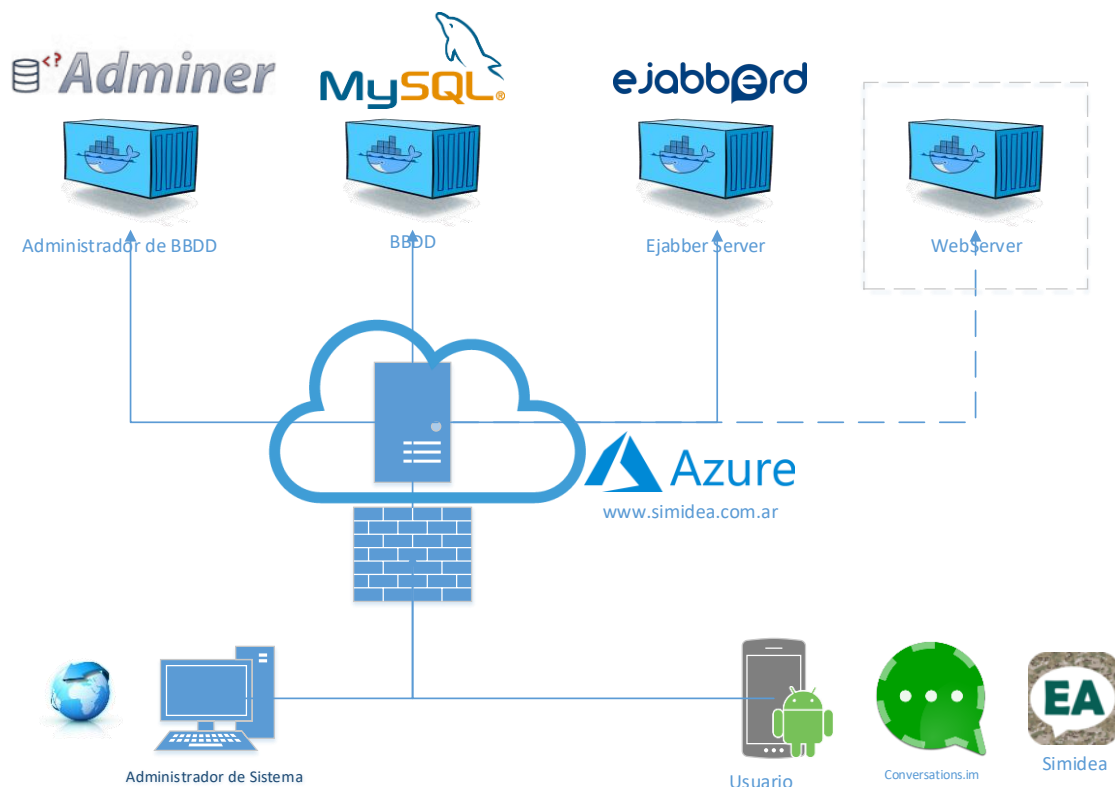
Fecha de Presentación

05Nov2021

## Concepto General del Proyecto

*SiMIDEA es un proyecto para la implementación de un sistema de mensajería instantánea privado, a ser utilizado por los integrantes del Ejército Argentino con la finalidad de tramitar las comunicaciones rutinarias propias de las actividades guarnicionales. Este sistema presenta como características principales la seguridad del mismo, implementando protocolos de cifrado confiables y una experiencia de usuario similar a la utilización de aplicaciones de mensajería públicas. A su vez, presenta la ventaja del control total del sistema a través de la gestión de los servidores de mensajería, asegurando de ésta forma el control de los datos que a través del mismo se tramitan o se crean.*

## 1. Enfoque del proyecto



Como puede verse en la imagen anterior, el componente central del sistema es el servidor de mensajería instantánea XMPP Ejabberd 21.04, montado sobre una arquitectura de contenedores. Junto con el servidor de mensajería, se implementa una base de datos de tipo MySQL externa al servidor, permitiendo mayores capacidades de

almacenamiento y de escalamiento. Además, la administración de la citada base de datos se realiza a través de Adminer, un gestor de base de datos MySQL.

Ejabberd es un servidor de mensajería instantánea de código abierto (GNU GPL) para Linux que ofrece una simultaneidad masiva 2.000.000 de usuarios en 1 nodo. Aplicaciones como Whatsapp y Facebook utilizan este servidor para la mensajería ya que permite hacer uso de, entre otras funcionalidades, chats de par a par, chats grupales, transmisión de archivos, de voz sobre IP, videollamadas, etc. El sistema a implementar ofrece las funcionalidades de chats de par a par, chats grupales, transmisión de archivos, de voz sobre IP, todo ello bajo la seguridad que ofrece el protocolo de encriptado de extremo a extremo OMEMO y del protocolo de cifrado de la conexión SSL/TLS.

Por otro lado, el sistema propuesto cuenta con una aplicación cliente para smartphones con sistema operativo Android, la cual se conectará al servidor de mensajería haciendo uso de su usuario y contraseña especificados en dicho servidor. Esta aplicación cliente está basada en la herramienta open source Conversations, una aplicación desarrollada por una comunidad abierta escrita en Java, como una de las más seguras que existen en la actualidad en la categoría de clientes XMPP open source.

Es importante destacar que el sistema tiene como pilar fundamental a la seguridad del mismo, la cual es transversal a los diferentes componentes, viéndose reflejada a través de la aplicación de diferentes protocolos en cada uno de dichos componentes. Así, el sistema hace uso de los protocolos TLS y OMEMO a fin de asegurar la confidencialidad, integridad y disponibilidad de los datos que a través de él se gestionan, logrando el encriptado de los mensajes de extremo a extremo y el cifrado de la conexión. Cabe destacar que además de los protocolos de cifrado, el sistema demanda contar con cierta infraestructura física que permita mantener la seguridad de los diferentes componentes o servidores, entendiéndose por ello de forma mínima un firewall que filtre el tráfico e impida el acceso no autorizado al servidor que gestiona el sistema.

#### a. Selección de herramientas

Al momento de analizar las diferentes variantes para cumplimentar los requisitos del sistema, se tuvo en cuenta las ventajas y desventajas que ofrecían cada una de las herramientas implementadas en cada componente. Basado en lo conocido como el Paradigma del Software Libre, se decidió hacer uso herramientas opensource ya desarrolladas y testeadas y partiendo de las mismas realizar modificaciones para que se adapten a los requerimientos establecidos. Por ello, inicialmente se optó por la utilización del Protocolo XMPP, protocolo abierto que posee toda una fundación que otorga soporte y actualización permanente (actualizaciones constatadas en <https://xmpp.org/>). La utilización de este protocolo llevo a seleccionar un servidor con las características mencionadas anteriormente tal como lo es Ejabberd, el cual también cuenta con toda una organización que brinda soporte, soporte al cual se puede acceder fácilmente a través de su página web. Por el lado del almacenamiento en base de datos, la utilización de MySQL da la confianza de una de las base de datos más utilizadas a lo largo de la historia, ampliamente probada y que posee un sin número de bibliografía para consultar ante cualquier inconveniente. Finalmente, la utilización del cliente XMPP

como aplicación base sobre la cual realizar modificaciones, aseguró tanto la usabilidad como la eficiencia del sistema. Lo más destacable de este cliente de mensajería a la hora de tomar la decisión de elección del mismo radicó en la facilidad de acceso a soporte por parte de la comunidad de desarrolladores con los que cuenta lo cual da un respaldo o apoyo de seguridad al sistema, llegando a realizar consultas propias a esta comunidad y recibir respuestas en forma inmediata. Esto habilita un canal de comunicación para reportar eventuales inconvenientes o problemas de seguridad y poder darles una solución en forma conjunta.

b. Eficiencia y pruebas de estrés:

Se considera importante citar ciertas referencias que responden a la eficiencia de las herramientas y componentes del sistema, respondiendo principalmente a las capacidades demostradas por estos componentes ante diferentes pruebas de estrés o comprobaciones realizadas por organismos especializados.

Respecto al servidor de mensajería Ejabberd, una prueba realizada en el año 2016 demuestra su alto rendimiento pudiendo tener conectados hasta dos millones de usuarios en forma simultánea en un único nodo. Este informe realizado por Mickaël Rémond buscaba el objetivo de llegar a dos millones de usuarios conectados, cada uno con 18 contactos en la lista y una sesión que dura alrededor de 1 hora, siendo ejecutada en una instancia EC2 de Amazon, de 40 vCPU y 160 GiB de RAM (considerablemente superior a la instancia utilizada en esta etapa inicial en el proyecto). Puede encontrarse más detalles de tal prueba en <https://www.process-one.net/blog/ejabberd-massive-scalability-1node-2-million-concurrent-users/>

También es importante destacar que el servidor de mensajería fue testeado a través del XMPP Compliance Tester, un servicio web para verificar y visualizar el estado de cumplimiento con las XEP (XMPP Extension Protocols) de los servidores XMPP, alcanzando el cumplimiento del 95% de las XEP que allí se comprueban, un valor extremadamente alto para esta etapa en la que se encuentra el proyecto. Dicho testeo se puede observar en <https://compliance.conversations.im/server/simidea.com.ar/>, siendo adjuntado el informe de tal comprobación.

Si bien el despliegue del sistema sobre la nube de Azure puede verse modificado en una etapa futura, la realidad es que este proveedor de servicios cloud ofrece una amplia muestra de las capacidades aseguradas respecto a los servicios contratados. Entre la documentación oficial, existe un apartado que responde a los límites, cuotas y restricciones de suscripción y servicio de Azure, a la cual se puede acceder visitando <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#virtual-machines-limits>.

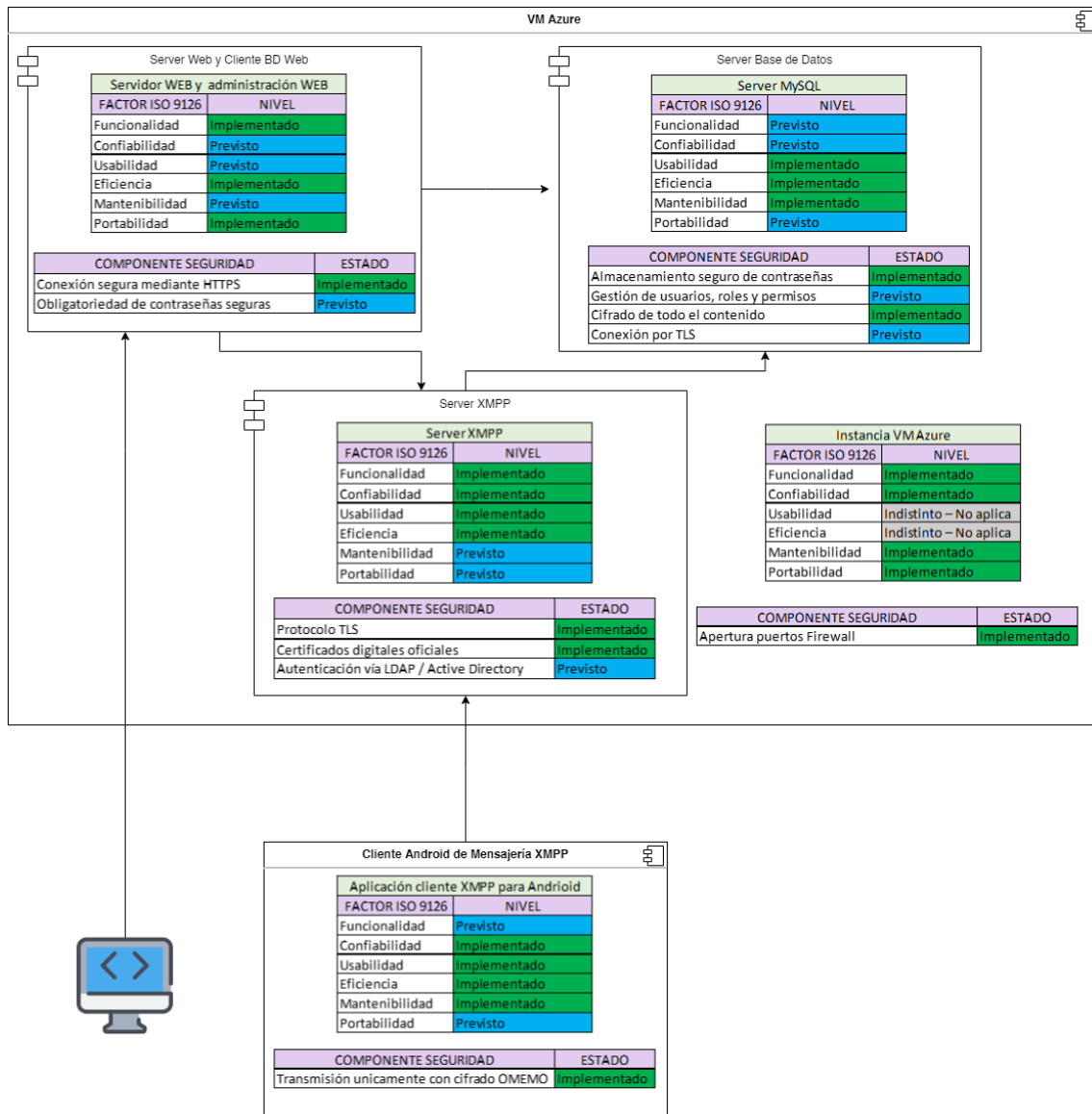
Además, se pueden encontrar una gran cantidad de artículos relacionados con pruebas de estrés realizadas sobre estos servicios cloud, destacando entre ellos uno desarrollado por Iiro Vanninen, en el que se exponen las métricas obtenidas en una prueba de estrés ejecutada sobre un servidor web montado en dicho proveedor. Si bien el servicio sobre el que se ejecuta dicha prueba es diferente al de un servidor de mensajería instantánea, se da una noción general de la capacidad del servicio y de la forma de ejecutar tal prueba en el caso de que en un futuro se desee optar por este

servicio cloud. Dicho artículo puede consultarse en <https://zure.com/blog/limits-of-the-cloud-stress-testing-in-azure-part-2/>.

c. Escalas de seguridad:

La propia arquitectura del sistema otorga una serie de escalas de seguridad otorgada por cada una de las herramientas en las que ha sido desplegado. En primer lugar, la propia instancia en el cloud ofrece ciertas particularidades de seguridad, pudiendo ser consultadas en <https://docs.microsoft.com/en-us/azure/security/>. En segundo lugar, el sistema operativo Linux Ubuntu Server posee sus propias particularidades de acceso y seguridad, destacando entre éstos al módulo de seguridad AppArmor, basado en el modelo de control de acceso obligatorio (MAC), siendo uno de los módulos de seguridad más eficientes para el kernel de Linux. Pueden encontrarse más detalles de seguridad de Ubuntu Server en <https://ubuntu.com/server/docs/security-introduction>. Finalmente, la propia tecnología de contenedores ofrece una nueva capa de seguridad para el sistema. Docker destaca en su documentación oficial relacionada al tema (<https://docs.docker.com/engine/security/>). Independientemente de esta seguridad, se recomienda tener en cuenta el control de versiones respecto al SO del propio contenedor y de los módulos o herramientas que el mismo pueda contener a fin de asegurar aún más tal seguridad. Una solución a este problema consiste en el uso de herramientas de escaneo de seguridad en contenedores.

2. Esquema general de los **componentes** del sistema y perfil de calidad de cada uno de ellos.



**NOTA:** los valores de los niveles presentados son los considerados por el Jefe de Proyecto, siendo sometidos a la aprobación o modificación por el organismo correspondiente.

3. Listado de los atributos tenidos en cuenta en cada componente y su estado de implementación alcanzado, junto con los motivos considerados para alcanzar tal nivel de implementación.

### 3.1. CALIDAD (Conforme a ISO 9126)

#### 3.1.1. Servidor de mensajería XMPP

Servidor de mensajería XMPP		
Atributo de Calidad	Estado	Motivo
Funcionalidad	Implementado	Especificado en 3.1.1.1
Confiabilidad	Implementado	Especificado en 3.1.1.2
Usabilidad	Implementado	Especificado en 3.1.1.3
Eficiencia	Implementado	Especificado en 3.1.1.4
Mantenibilidad	Previsto	Especificado en 3.1.1.5

Portabilidad	Previsto	Especificado en 3.1.1.6
--------------	----------	-------------------------

- 3.1.1.1. Funcionalidad: La característica de trabajar en base a un estándar internacional para la mensajería instantánea asegura la funcionalidad del sistema. Por otro lado, claramente el servidor Ejabberd es una herramienta específica para la necesidad de un sistema de mensajería instantánea, creado para tal fin lo que lo hace idóneo para la función específica. Además, permite la interoperabilidad con otros servidores de mensajería, siempre y cuando trabajen con el protocolo XMPP.
- 3.1.1.2. Confiabilidad: Frente a otros servidores XMPP, Ejabberd ofrece una mayor disponibilidad y tolerancia a fallos. Además, la propia arquitectura de contenedores, permite tener una mayor disponibilidad y capacidad de recuperación, aspectos que pueden incrementar su nivel si se gestionan los contenedores a través de un orquestador de contenedores, algo sencillo de implementar en etapas futuras gracias a la arquitectura utilizada.
- 3.1.1.3. Usabilidad: La principal característica de usabilidad que posee el servidor es el soporte y la amplia documentación existente, aspectos que incrementan sustancialmente la capacidad de aprendizaje que se tiene sobre el mismo. La operabilidad del mismo se ve reducida en la complejidad que presenta la configuración a través de archivos de configuración. Sin embargo, para el administrador del sistema, ofrece una plataforma web que permite gestionar cuentas con facilidad.
- 3.1.1.4. Eficiencia: El servidor XMPP se caracteriza por ofrecer un gran nivel de capacidad con una utilización de recursos media. El propio hecho de la utilización de un protocolo de mensajería instantánea abierto y extensible, incrementa la capacidad general del sistema. Además de ello, ofrece posibilidades de integración con clientes web y de cifrado de extremo a extremo, lo que le permite al sistema ampliar sus capacidades de escalabilidad a futuro en relación a los clientes XMPP.
- 3.1.1.5. Mantenibilidad: Si bien el propio servidor Ejabberd no permite realizar ciertas modificaciones o configuración mientras se encuentra activo, la utilización de tecnología de contenedores permiten alcanzar un alto nivel de modularidad (dada por la propia arquitectura del servidor Ejabberd y por la configuración aplicada entorno a la persistencia de los datos), reusabilidad, capacidad de ser modificado, capacidad de ser probado (tres características principalmente generadas por la definición y ejecución de varios contenedores con aplicaciones en Docker a través de un docker-compose) y analizabilidad (herramientas

ofrecidas por el servidor Ejabberd a través de la plataforma web para el administrador)

- 3.1.1.6. Portabilidad: En forma muy similar a la mantenibilidad, la tecnología de contenedores en la cual se encuentra inserta el servidor Ejabberd y la propia forma de ejecución (a través de la herramienta docker-compose) permiten una amplia adaptabilidad, facilidad de instalación y capacidad de ser reemplazado.

### 3.1.2. Servidor WEB y módulos para administración WEB

Servidor WEB y módulos para administración WEB		
Atributo de Calidad	Estado	Motivo
Funcionalidad	Implementado	Especificado en 3.1.2.1
Confiabilidad	Previsto	Especificado en 3.1.2.2
Usabilidad	Previsto	Especificado en 3.1.2.3
Eficiencia	Implementado	Especificado en 3.1.2.4
Mantenibilidad	Previsto	Especificado en 3.1.2.5
Portabilidad	Implementado	Especificado en 3.1.2.6

- 3.1.2.1. Funcionalidad: el acceso web a los diferentes componentes que hacen al sistema permiten la interoperabilidad de los sistemas y el acceso de una forma sencilla sobre la administración de la gestión de usuarios del servidor XMPP y de la base de datos MySQL

- 3.1.2.2. Confiabilidad: Si bien el servidor web de la página principal y el de la administración de base de datos se ejecutan en forma independiente, la interface de administración web del servidor XMPP forma parte del servicio que ofrece tal servidor, por lo cual se encontrará disponible siempre y cuando el servidor XMPP se encuentre en funcionamiento. Por otro lado, debido al alcance limitado del sistema para estas etapas no se aplica ningún método o herramienta que permita la tolerancia a fallos

- 3.1.2.3. Usabilidad: Los aspectos visuales y de accesibilidad de todas las interfaces web son bajos, propios de una etapa inicial, siendo necesario que personal capacitado en el diseño multimedial mejore la experiencia de usuarios en etapas venideras. Pese a ello, la operabilidad de las interfaces resultan fáciles de aprender y utilizar.

- 3.1.2.4. Eficiencia: La baja funcionalidad de las interfaces permite una mínima utilización de recursos tanto del lado del cliente como del servidor, teniendo acceso eficiente en base a las funciones que cumple. Si bien las capacidades que ofrecen la mayoría de las interfaces web son limitadas, cumplen con las necesidades de administración para la etapa en la que se encuentra el sistema.

3.1.2.5. **Mantenibilidad:** Si bien la página principal de acceso web puede ser modificada sin inconvenientes, el resto de las interfaces no pueden ser modificadas de manera sencilla, por lo cual su adaptabilidad a nuevas necesidades resultaría compleja. Por otro lado, la interface de administración de la base de datos ofrece la reusabilidad sobre otras bases de datos que el sistema pueda incorporar en un futuro.

3.1.2.6. **Portabilidad:** Si bien la interface de administración web del servidor XMPP no es controlable por fuera de éste, tanto la página web principal como el cliente web de base de datos pueden ser fácilmente instalables, son adaptables y tienen la capacidad de ser reemplazado, todo ello gracias a la tecnología de contenedores sobre la que se encuentran insertos.

### 3.1.3. Servidor de Base de Datos

Servidor de Base de Datos		
Atributo de Calidad	Estado	Motivo
Funcionalidad	Previsto	Especificado en 3.1.3.1
Confiabilidad	Previsto	Especificado en 3.1.3.2
Usabilidad	Implementado	Especificado en 3.1.3.3
Eficiencia	Implementado	Especificado en 3.1.3.4
Mantenibilidad	Implementado	Especificado en 3.1.3.5
Portabilidad	Previsto	Especificado en 3.1.3.6

3.1.3.1. **Funcionalidad:** la base de datos MySQL con la que trabaja el servidor de mensajería instantánea proveer un adecuado conjunto de funciones que cumplen las tareas y objetivos específicos que tienen. Si bien para una etapa inicial del proyecto la arquitectura es correcta, se prevé tener una baja interoperabilidad en caso de necesitar a futuro una base de datos distribuida.

3.1.3.2. **Confiabilidad:** Si bien la disponibilidad de este tipo de base de datos es muy alta y presenta un nivel de madurez superior al resto, en su configuración e implementación dentro del sistema no se aplican procedimientos para la recuperación de datos, tal como puede ser el replicado de los mismos, aspecto tenido en cuenta para etapas venideras.

3.1.3.3. **Usabilidad:** Al tratarse de un servidor de base de datos, la operabilidad del mismo dependerá del grado de capacitación que tenga tal operador, por lo cual, para un conocedor de este tipo de base de datos resultará sencillo realizarlo desde la línea de comandos. Sin embargo, a fin de simplificar la operabilidad de la base de datos es que el sistema cuenta con un cliente web de base



de datos, incrementando las características estéticas y de accesibilidad frente a la interface de línea de comandos.

3.1.3.4. Eficiencia: Un servidor de base de datos MySQL ofrece una capacidad de gestión de base de datos muy alta, siendo una de las mejores para la implementación de bases de datos relacionales. Si bien para una etapa inicial el consumo de recursos resulta bajo, se prevé un nivel medio para las próximas etapas.

3.1.3.5. Mantenibilidad: este tipo de base de datos ofrece la capacidad de ser modificada y probada en “en caliente”, aspecto que podría verse mejorado con la implementación de una base de datos en etapas posteriores. Las herramientas propias de administración de base de datos de MySql permiten el análisis permanente de la misma.

3.1.3.6. Portabilidad: La propia tecnología de contenedores aseguran la portabilidad de la base de datos, siendo adaptable y de fácil instalación dentro de otras instancia o hardware. Por otro lado, la inexistencia de réplicas, le impiden a esta base de datos en esta etapa la capacidad de ser reemplazada en forma sencilla, situación que podrá ser evaluada en futuras etapas.

#### 3.1.4. Instancia VM Azure

Instancia VM Azure (*)		
Atributo de Calidad	Estado	Motivo
Funcionalidad	Implementado	Especificado en 3.1.4.1
Confiabilidad	Implementado	Especificado en 3.1.4.2
Usabilidad	Indistinto	Especificado en 3.1.4.3
Eficiencia	Indistinto	Especificado en 3.1.4.4
Mantenibilidad	Implementado	Especificado en 3.1.4.5
Portabilidad	Implementado	Especificado en 3.1.4.6

(\*) Inicialmente cabe destacar que, al ser una instancia virtualizadas en uno de los mayores proveedores cloud existentes hoy en día, ofrece las características respecto a la calidad que ofrece tal proveedor. El concepto central del proyecto es la virtualización del servidor que permita darle un nivel de portabilidad al sistema, a fin que el Ejército Argentino decida posteriormente si mantiene el sistema sobre un proveedor cloud que le suministre la infraestructura o si lo emplea sobre su propia infraestructura.

3.1.4.1. Funcionalidad: Al tratarse de una instancia virtualizada en Azure (pudiendo ser también otro proveedor), se permite que los componentes de la misma puedan interactuar con otros elementos, incrementando la interoperabilidad de este hardware virtualizado.

- 3.1.4.2. Confiabilidad: La propia arquitectura de virtualización y la contratación de un proveedor cloud aseguran la disponibilidad de los recursos y de esta instancia. Si bien no se desarrolla para esta etapa del proyecto, es importante desarrollar una política de recuperación del almacenamiento de la instancia, lo que permitiría recuperar la instancia en cualquier otro sistema de virtualización.
- 3.1.4.3. Usabilidad: No aplica – Si bien el acceso y la creación mediante este proveedor resulta sencilla, ello aplica puntualmente a Azure, pudiendo ser diferente en otros casos.
- 3.1.4.4. Eficiencia: No aplica – La eficiencia entorno a recursos de la instancia dependen de las características contratadas para la misma. Para etapa del proyecto se realizó una contratación mínima que otorga una capacidad baja, siendo modificado en el momento en que el sistema pase a producción.
- 3.1.4.5. Mantenibilidad: La virtualización ofrece un alto nivel de modularidad, dando la posibilidad de modificar las capacidades de la instancia, pudiendo ser muchos de estos cambios mientras el servicio se encuentra en funcionamiento.
- 3.1.4.6. Portabilidad: Es la principal característica de calidad de esta instancia, la cual ofrece una amplia capacidad para ser reemplazado debido a la tecnología de virtualización. Dependiendo del servicio puntual que se contrate o con el que se cuente, se dará o no la facilidad de instalación.

### 3.1.5. Aplicación cliente XMPP para Android

Aplicación cliente XMPP para Android		
Atributo de Calidad	Estado	Motivo
Funcionalidad	Previsto	Especificado en 3.1.5.1
Confiabilidad	Implementado	Especificado en 3.1.5.2
Usabilidad	Implementado	Especificado en 3.1.5.3
Eficiencia	Implementado	Especificado en 3.1.5.4
Mantenibilidad	Implementado	Especificado en 3.1.5.5
Portabilidad	Previsto	Especificado en 3.1.5.6

- 3.1.5.1. Funcionalidad: Al trabajar basada en el Protocolo XMPP, se asegura la conformidad del sistema. Además, la aplicación móvil permite la interoperabilidad con otras aplicaciones, sean móviles, web o de escritorio. Si bien con esto se asegura el escalamiento del sistema a nuevas plataformas, también se permite el acceso al servidor de mensajería mediante otra aplicación cliente XMPP, la cual pueda llegar a resultar insegura, por lo que se prevé para siguientes etapas del proyecto diseñar un mecanismo que

imposibilite el acceso al servicio de mensajería desde otras aplicaciones cliente.

3.1.5.2. Confiabilidad: El hecho de desarrollar la aplicación a partir de un software open source ampliamente utilizado y reconocido, se asegura un nivel de madurez alto, con capacidades de recuperación y tolerancia a fallas. Además, propio del tipo de aplicación, se ofrece una permite disponibilidad de la misma, debido a la necesidad de mantenerse en funcionamiento en segundo plano. Tal disponibilidad ha sido probada en forma experimental.

3.1.5.3. Usabilidad: La aplicación ofrece una interface de usuario sencilla y clara (siendo uno de los requerimientos su similitud a WhatsApp), lo que asegura un aprendizaje sencillo y diseño estético atractivo y reconocible. Dicha similitud a la aplicación de mensajería instantánea más utilizada del mundo permite alcanzar una operabilidad alta por parte de los usuarios. Por otro lado, no ofrece grandes características entorno a la accesibilidad, siendo un aspecto previsto para futuras etapas (con baja prioridad para el Ejército Argentino).

Es importante destacar que la aplicación original Conversations.im es un cliente de mensajería XMPP ampliamente utilizado y testeado, que asegura la calidad necesaria respecto a la usabilidad.

3.1.5.4. Eficiencia: Si bien la utilización de recursos dependerá del dispositivo en el que se encuentre desplegada, en la mayoría de los utilizados en la actualidad ésta será media. Las capacidades ofrecidas para la comunicación entre diferentes usuarios son amplias (transmisión de mensajes, archivos, llamadas de voz y audio), pudiendo mejorar en otras etapas del proyecto su comportamiento temporal al momento del envío de datos adjuntos.

3.1.5.5. Mantenibilidad: El proyecto sobre el que se basa la aplicación ha sido diseñado en base a los conceptos de modularidad necesarios. Además, brinda una capacidad de análisis de logs media que permiten conocer el funcionamiento general de la aplicación en los momentos de deploy. Presenta una amplia capacidad para ser modificada, aunque compleja y poco documentada. Además, las pruebas unitarias sobre los diferentes componentes del sistema no han sido realizadas.

3.1.5.6. Portabilidad: Para esta etapa del proyecto, la instalación de la aplicación sobre la plataforma móvil se realiza mediante un archivo de paquetes apk, teniendo previsto ser montada sobre la plataforma de Playstore en siguientes etapas para una fácil

instalación, lo que permite una mayor capacidad de ser reemplazado o actualizado.

### 3.2. SEGURIDAD

Antes de detallar los componentes de seguridad destacados en el informe, resulta necesario destacar que los mismos no responden a ningún estándar o proyecto de seguridad informática (por ejemplo OWASP Top Ten) debido a las características particulares del sistema, siendo tales proyectos particularizados a una temática específica (aplicaciones web en el caso de OWASP)

#### 3.2.1. Servidor de mensajería XMPP

Servidor de mensajería XMPP		
Atributo de Calidad	Estado	Motivo
Protocolo TLS	Implementado	Especificado en 3.2.1.1
Certificados digitales	Implementado	Especificado en 3.2.1.2
Autenticación vía LDAP / Active Directory	Previsto	Especificado en 3.2.1.3

##### 3.2.1.1. Protocolo TLS:

En sí, el servidor de mensajería Ejabberd permite la aplicación de STARTTLS, una extensión a los protocolos de comunicaciones que ofrece establecer una conexión cifrada sobre el mismo puerto en el que se realiza la comunicación no cifrada. Se basa en TLS, incorporando ciertos pasos que facilitan la utilización del puerto ya utilizado. Aplicando el mismo, es posible que las aplicaciones clientes de mensajería que se conecten al servidor lo harán mediante el protocolo TLS.

Para la configuración del protocolo TLS, resulta necesario establecer en el archivo de configuración del servidor de mensajería el direccionamiento de los certificados digitales que permitirán la negociación TLS y establecer en los puertos de conexión que la misma se debe realizar aplicando TLS o STARTTLS.

```
certfiles:
- /etc/letsencrypt/live/simidea.com.ar/fullchain.pem
- /etc/letsencrypt/live/simidea.com.ar/key.pem

listen:
-
  port: 5222
  ip: "::"
  module: ejabberd_c2s
  max_stanza_size: 262144
  shaper: c2s_shaper
  access: c2s
  starttls_required: true
```

La configuración de estas variables permitirá que el servidor realice sus conexiones mediante TLS, como puede verse en los logs siguientes:

```

ejabberd | 2021-10-23 16:16:45.891233+00:00 [info] ejabberd 21.7.0 is started in the node ejabberd@bb9aed239e47 in S7.1s
ejabberd | 2021-10-23 16:16:45.896912+00:00 [info] Start accepting UDP connections at [::]:3478 for ejabberd_stun
ejabberd | 2021-10-23 16:16:45.898589+00:00 [info] Start accepting TCP connections at [::]:5222 for ejabberd_c2s
ejabberd | 2021-10-23 16:16:45.898803+00:00 [info] Start accepting TLS connections at [::]:5223 for ejabberd_c2s
ejabberd | 2021-10-23 16:16:45.899017+00:00 [info] Start accepting TCP connections at [::]:5269 for ejabberd_s2s_in
ejabberd | 2021-10-23 16:16:45.899597+00:00 [info] Start accepting TLS connections at [::]:5443 for ejabberd_http
ejabberd | 2021-10-23 16:16:45.899861+00:00 [info] Start accepting TCP connections at [::]:5280 for ejabberd_http
ejabberd | 2021-10-23 16:16:45.900887+00:00 [info] Start accepting TCP connections at [::]:1883 for mod_mqtt
ejabberd | 2021-10-23 16:16:45.913877+00:00 [info] Start accepting TCP connections at [::]:16010 for mod_stream65

```

Pudiendo observar el establecimiento de la conexión mediante TLS en los momentos en que los usuarios se conectan al servidor:


```

.3]:5222
ejabberd | 2021-10-23 17:37:10.294848+00:00 [info] (tls|<0.754.0>) Accepted c2s SCRAM-SHA-1 authentication for horacio@simidea.com.ar by
y sql backend from ::ffff:181.170.201.105
ejabberd | 2021-10-23 17:37:10.673249+00:00 [info] (tls|<0.755.0>) Accepted c2s SCRAM-SHA-1 authentication for hari2@simidea.com.ar by
sql backend from ::ffff:181.170.201.105
ejabberd | 2021-10-23 17:37:10.998719+00:00 [info] (tls|<0.754.0>) Resumed session for horacio@simidea.com.ar/SiMiDeA.OPoX
ejabberd | 2021-10-23 17:37:11.135225+00:00 [info] (tls|<0.755.0>) Resumed session for hari2@simidea.com.ar/SiMiDeA.WYrC
ejabberd | 2021-10-23 17:39:51.502197+00:00 [info] (<0.758.0>) Accepted connection [::ffff:181.170.201.105]:32876 -> [::ffff:192.168.80
.3]:5222

```

### 3.2.1.2. Certificados digitales:

A fin de poder establecer las correspondientes conexiones a través del Protocolo TLS, resulta necesario contar con certificados digitales firmados por una autoridad competente. Para ello se creó un certificado que fue firmado por Let's Encrypt, Una autoridad de certificación sin fines de lucro que proporciona certificados TLS de manera gratuita. En la siguiente imagen, pueden verse las características del certificado digital:


**Información del certificado**

**Este certif. está destinado a los siguientes propósitos:**

- Prueba su identidad ante un equipo remoto
- Asegura la identidad de un equipo remoto
- 2.23.140.1.2.1
- 1.3.6.1.4.1.44947.1.1.1

\*Para ver detalles, consulte la declaración de la entidad de ce

---

**Emitido para:** simidea.com.ar

**Emitido por:** R3

**Válido desde** 23/10/2021 **hasta** 21/1/2022

Campo	Valor
Versión	V3
Número de serie	03864450511b3f36df3bfd8a2...
Algoritmo de firma	sha256RSA
Algoritmo hash de firma	sha256
Emisor	R3, Let's Encrypt, US
Válido desde	sábado, 23 de octubre de 202...
Válido hasta	viernes, 21 de enero de 2022 ...
Sujeto	simidea.com.ar

Es importante destacar que el certificado digital fue generado únicamente para el dominio principal, motivo por el cual el servidor de mensajería instantánea demanda certificados para una serie de subdominios que utiliza para funciones particulares. Se prevé contar con el certificado con los subdominios correspondientes en etapas posteriores, a fin de asegurar todo tipo de conexión TLS que el sistema establezca:

```

ejabberd 2021-10-23 16:16:45.889364+00:00 [info] Waiting for TLSv1.3 synchronization to complete
ejabberd 2021-10-23 16:16:45.889364+00:00 [warning] No certificate found matching pubsub.simidea.com.ar
ejabberd 2021-10-23 16:16:45.889364+00:00 [warning] No certificate found matching proxy.simidea.com.ar
ejabberd 2021-10-23 16:16:45.889364+00:00 [warning] No certificate found matching conference.simidea.com.ar
ejabberd 2021-10-23 16:16:45.889364+00:00 [warning] No certificate found matching upload.simidea.com.ar
ejabberd 2021-10-23 16:16:45.891233+00:00 [info] ejabberd 21.7.0 is started in the node ejabberd@bbfaed239e47 in 5.71s
ejabberd 2021-10-23 16:16:45.896912+00:00 [info] Start accepting UDP connections at [::]:3478 for ejabberd_stun
ejabberd 2021-10-23 16:16:45.898589+00:00 [info] Start accepting TCP connections at [::]:5222 for ejabberd_c2s
ejabberd 2021-10-23 16:16:45.898589+00:00 [info] Start accepting TLS connections at [::]:5223 for ejabberd_c2s_in
ejabberd 2021-10-23 16:16:45.899017+00:00 [info] Start accepting TCP connections at [::]:5269 for ejabberd_s2s_in
ejabberd 2021-10-23 16:16:45.899597+00:00 [info] Start accepting TLS connections at [::]:5443 for ejabberd_http
ejabberd 2021-10-23 16:16:45.899861+00:00 [info] Start accepting TCP connections at [::]:5280 for ejabberd_http
ejabberd 2021-10-23 16:16:45.900887+00:00 [info] Start accepting TLS connections at [::]:1883 for mod_mqtt
ejabberd 2021-10-23 16:16:45.917857+00:00 [info] Start accepting TCP connections at 192.168.48.3:7777 for mod_proxy65_stream
ejabberd 2021-10-23 16:16:45.920445+00:00 [info] Requesting new certificate for pubsub.simidea.com.ar, proxy.simidea.com.ar and 2 mor
e hosts from https://acme-v02.api.letsencrypt.org/directory

```

Relacionado a lo anteriormente descripto, se recomienda la compra de un certificado digital a una autoridad certificante, que posibilite mayor seguridad sobre el mismo y soporte ante inconvenientes que se puedan presentar.

### 3.2.1.3. Autenticación vía LDAP / Active Directory:

Ejabberd permite la autenticación a través de Active Directory mediante el protocolo LDAP. De esta forma, si el Ejército Argentino quisiera realizar la autenticación mediante este servicio (pudiendo hacer por ejemplo que todos los usuarios se autenticquen con su cuenta de correo), podría usar tal servicio (el cual utiliza) para realizar una asignación personal de cuentas y delegar sobre el mismo la autenticación

Si bien fueron gestionadas los permisos y cuentas necesarias para realizar tal autenticación sobre el servicio Active Directory del Ejército, debido a encontrarse el servidor por fuera de la red interna resulta necesario solicitar autorizaciones especiales que podrían dar lugar a ciertas vulnerabilidades para esta etapa del proyecto. Además, sería necesario el cambio del dominio del servidor de mensajería con el que se trabaja y con ello los certificados digitales necesarios (aspecto que demanda ciertas autorizaciones por parte del Ejército Argentino).

Independientemente de lo comentado anteriormente, en caso de, en futuras etapas, decidir delegar la autenticación sobre este servicio, resultaría necesario establecerla en el archivo de configuración:

```

auth_method: [ldap]
ldap_servers: [ldap.ea.mil.ar]
ldap_base: "DC=ea,DC=mil,DC=ar"
ldap_rootdn: "OU=Puestos Fijos,OU=Usuarios,DC=ea,DC=mil,DC=ar"
ldap_password: "*****"
ldap_uids: [sAMAccountName]
ldap_filter: "(memberOf=*)"

modules:
...
mod_vcard:
  db_type: ldap
  ldap_vcard_map:
    GIVEN: {"%s": [givenName]}
    EMAIL: {"%s": [mail]}
    ORGNAME: {"%s": [company]}
    ORGUNIT: {"%s": [department]}
    TITLE: {"%s": [title]}
  ldap_search_fields:
    User: "%u"
    Name: givenName
    Email: mail

```

```

Company: company
Department: department
Role: title
ldap_search_reported:
  "Full Name": FN
  Email: EMAIL
...

```

### 3.2.2. Servidor WEB y módulos para administración WEB

Servidor WEB y módulos para administración WEB		
Atributo de Calidad	Estado	Motivo
Conexión segura mediante HTTPS	Implementado	Especificado en 3.2.2.1
Obligatoriedad de contraseñas seguras	Previsto	Especificado en 3.2.2.2

#### 3.2.2.1. Conexión segura mediante HTTPS:

El sitio web principal se encuentra montado sobre un servidor web Nginx, que, entre otras herramientas, permite la configuración del Protocolo HTTPS a fin de contar con un sitio web seguro. Para ello resultó necesario solicitar la firma del certificado digital a la autoridad competente. Además, se encuentra configurado de manera tal que en forma permanente censa la legalidad del certificado, solicitando renovaciones en caso de que el mismo esté cerca de vencer.

Además, redirecciona la conexión desde el puerto 80 (conexión en protocolo HTTP) al puerto 443 (conexión en protocolo HTTPS) para todas las solicitudes que se hagan sobre el protocolo no seguro.

#### 3.2.2.2. Obligatoriedad de contraseñas seguras:

La creación de usuarios para el sistema de mensajería se realiza de la interfaz web del servidor de mensajería. Esta interfaz permite crear usuarios y contraseñas sin ningún tipo de restricción u obligatoriedad respecto a la extensión y contenido de la contraseña, razón por la que se pueden asignar contraseñas inseguras para los usuarios. Ante esta problemática, y en caso de no delegar la autenticación al servicio Active Directory del Ejército, se prevé realizar una modificación a la interfaz web del servidor a fin de directamente no permitir la utilización de una contraseña débil.

### 3.2.3. Servidor de Base de datos

Servidor de Base de datos		
Atributo de Calidad	Estado	Motivo
Almacenamiento seguro de contraseñas	Implementado	Especificado en 3.2.3.1
Gestión de usuarios, roles y permisos	Previsto	Especificado en 3.2.3.2

Cifrado del contenido transmitido	Implementado	Especificado en 3.2.3.3
Protocolo TLS	Previsto	Especificado en 3.2.3.4

### 3.2.3.1. Almacenamiento seguro de contraseñas:

El servidor de mensajería Ejabberd permite ser configurado de manera tal que la autenticación se realice mediante el mecanismo SCRAM-SHA-1 (Salted Challenge Response Authentication Mechanism) una familia de mecanismos de autenticación de desafío-respuesta modernos y basados en contraseñas que brindan autenticación de un usuario en un servidor. Este mecanismo asegura que el almacenamiento de claves se realice en forma segura, no siendo almacenadas las mismas en forma de texto plano.

Para establecer tal mecanismo, es necesario habilitarlo en el módulo de base de datos del archivo de configuración del servidor a través de la variable "auth\_password\_format: scram", quedando configurado dicho módulo de la siguiente forma:

```
host_config:
  simidea.com.ar:
    sql_port: 3306
    sql_type: mysql
    sql_server: "mysql"
    sql_pool_size: 1
    default_db: sql
    auth_method: sql
    auth_password_format: scram
```

A continuación, puede observarse una consulta a la base de datos a través de un cliente web, en el cual se observa la forma de almacenamiento de la clave, la cual claramente no es la misma con la que el usuario se autentica:

username	password	serverkey	salt	iterationcount	created_at
horacio	/7y/Ja42cEPWhFXpQeyO+Neulu0=	z/6mBhhOdJRN8SdZqYc60VrJpNg=	JDn+0XQ4TU1yPDHEerDNpQ==	4096	2021-10-13 21:02:47

### 3.2.3.2. Gestión de usuarios, roles y permisos:

Inicialmente, la gestión de la base de datos será ejercida por un único usuario que contará con todos los permisos para ello. Sin embargo, se prevé para una futura etapa la creación de dos usuarios que puedan administrar la base de datos, uno de ellos con los permisos necesarios para cumplir el rol de "administrador de usuarios" (con permisos de lectura y modificación a las tablas relacionadas con los usuarios), y un segundo usuario con los permisos necesarios para el cumplimiento del rol de "administración de metadatos" (con permisos sobre las tablas que almacenan los metadatos de conexión y demás).

De esta forma, existirá un DBA que se encargue de la administración y mantenimiento de toda la base de datos (con todos los permisos), un administrador de



cuentas de usuario y un administrador de metadatos. De esta forma se establecerá un modelo de seguridad basado en roles para ayudar al acceso seguro a la base de datos.

### 3.2.3.3. Cifrado del contenido transmitido:

El sistema de mensajería en forma general trabaja con cifrado de extremo a extremo, lo que permite que los datos del contenido de los mensajes que permanecen almacenados en el servidor no puedan ser accedidos por quien no tenga las credenciales correspondientes.

De esta forma, el servidor de mensajería almacena metadatos (una de las características más destacada del sistema por la mínima cantidad de metadatos que genera en comparación con aplicaciones de mensajería comerciales) dentro de la base de datos. Dentro de esos datos almacenará el contenido del mensaje, el cual, al encontrarse cifrado por el Protocolo OMEMO, no podrá ser descryptado, almacenándose una frase que informe lo comentado. Ante la necesidad del cliente de mensajería de recuperar algún mensaje, podrá realizarlo sin inconvenientes, siendo descryptados en el extremo correspondiente.

A continuación, puede verse un ejemplo de un campo almacenado en la base de datos de un mensaje transmitido a través del sistema:

<u>username</u>	<u>timestamp</u>	<u>peer</u>	<u>bare_peer</u>	<u>xml</u>	<u>txt</u>	<u>id</u>	<u>kind</u>	<u>created_at</u>
hari2	1635000284776064	hari@simidea.com.ar/SiMIDEA.-BNs	<a href="mailto:hari@simidea.com.ar">hari@simidea.com.ar</a>	<message xml:lang='en' to='hari2@simidea.com.ar' from='hari@simidea.com.ar/SiMIDEA.-BNs' type='chat'...	I sent you an OMEMO encrypted message but your client doesn't seem to support that. Find more inform...	2	chat	2021-10-23 14:44:49

### 3.2.3.4. Protocolo TLS:

En la conexión entre el servidor de mensajería y el servidor de base de datos MySQL es posible aplicar el Protocolo TLS a fin de cifrar dicha conexión entre estas dos entidades. Para realizar tal implementación es necesario contar con certificados digitales firmados por una autoridad competente para el subdominio de base de datos del sistema. Debido a que el certificado digital tramitado es sólo para el dominio principal, esta implementación no es posible realizarla.

Es por ello que se prevé realizar esta implementación del protocolo TLS para la conexión entre la base de datos y el servidor de mensajería en una etapa futura, para lo cual será necesario tramitar los certificados digitales correspondientes.

### 3.2.4. Instancia VM Azure

Previamente a detallar los atributos de seguridad que ofrece la Instancia de Azure, es importante hacer referencia a que Microsoft Azure cumple, entre otros, con

la Normas de gestión de seguridad de la información ISO/IEC 27001, siendo auditada anualmente y pudiendo observar los informes de dichas auditorías en <https://servicetrust.microsoft.com/>.

Instancia VM Azure		
Atributo de Calidad	Estado	Motivo
Gestión del Firewall	Implementado	Especificado en 3.2.4.1

#### 3.2.4.1. Gestión del Firewall:

Como medida de seguridad, es importante mantener abiertos y en escucha sólo aquellos puertos que sean necesarios para las funciones del servidor. El resto de los puertos deben mantenerse cerrados, imposibilitando el acceso a cualquier ciberdelincuente a través del mismo. Como puede verse en la siguiente imagen, sólo se realiza la apertura de los puertos necesarios para el sistema, a saber (se explican aquellos propios del sistema):

```
adminSimidea@ServerSimidea:~/ejabberd2$ sudo netstat -plnut
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5443             0.0.0.0:*                 LISTEN      6389/docker-proxy
tcp        0      0 0.0.0.0:33060            0.0.0.0:*                 LISTEN      6229/docker-proxy
tcp        0      0 0.0.0.0:5222             0.0.0.0:*                 LISTEN      6450/docker-proxy
tcp        0      0 0.0.0.0:3306             0.0.0.0:*                 LISTEN      6249/docker-proxy
tcp        0      0 0.0.0.0:8080             0.0.0.0:*                 LISTEN      6633/docker-proxy
tcp        0      0 0.0.0.0:5269             0.0.0.0:*                 LISTEN      6429/docker-proxy
tcp        0      0 127.0.0.0:53:53          0.0.0.0:*                 LISTEN      642/systemd-resolve
tcp        0      0 0.0.0.0:22               0.0.0.0:*                 LISTEN      868/sshd: /usr/sbin
tcp        0      0 0.0.0.0:5280             0.0.0.0:*                 LISTEN      6409/docker-proxy
tcp6       0      0 :::5443                  :::*                     LISTEN      6395/docker-proxy
tcp6       0      0 :::33060                  :::*                     LISTEN      6235/docker-proxy
tcp6       0      0 :::5222                   :::*                     LISTEN      6456/docker-proxy
tcp6       0      0 :::3306                   :::*                     LISTEN      6255/docker-proxy
tcp6       0      0 :::8080                   :::*                     LISTEN      6638/docker-proxy
tcp6       0      0 :::5269                   :::*                     LISTEN      6435/docker-proxy
tcp6       0      0 :::22                     :::*                     LISTEN      868/sshd: /usr/sbin
tcp6       0      0 :::5280                   :::*                     LISTEN      6415/docker-proxy
udp        0      0 127.0.0.0:53:53          0.0.0.0:*                 642/systemd-resolve
udp        0      0 10.2.0.4:68              0.0.0.0:*                 640/systemd-network
udp        0      0 127.0.0.1:323            0.0.0.0:*                 790/chronyd
udp6       0      0 :::1:323                  :::*                     790/chronyd
```

- 5443: usado por el módulo ejabberd\_http (Este módulo habilita la comunicación XMPP a través de la conexión Websocket)
- 5222: usado por el módulo ejabberd\_c2s (Maneja conexiones c2s)
- 5269: usado por el módulo ejabberd\_c2s\_in (Maneja conexiones entrantes c2s)
- 5280: usado para el acceso a la interfaz web de configuración
- 8080: utilizado por el cliente web de base de datos
- 3306/33060: puertos del servidor de base de datos MySQL

#### 3.2.5. Aplicación cliente XMPP para Android

Aplicación cliente XMPP para Android		
Atributo de Calidad	Estado	Motivo
Transmisión solamente mediante cifrado OMEMO	Implementado	Especificado en 3.2.5.1

#### 3.2.5.1. Transmisión solamente mediante cifrado OMEMO:

Tal como ya fue comentado, el sistema hace uso del protocolo OMEMO para el cifrado de extremo a extremo de la comunicación. Sin embargo, la aplicación Android original Conversations permite utilizar otro protocolo de cifrado de extremo a extremo o incluso tramitar la comunicación sin cifrado. Debido a la importancia para el sistema Simidea de este cifrado, se estableció por default este cifrado y se eliminaron el resto de posibilidades de cifrado, a fin de que sólo sea posible cursar una comunicación por el sistema de forma cifrada de extremo a extremo.