

# SELECCIÓN DE UN SISTEMA DE MENSAJERÍA INSTANTÁNEA SEGURO

SEPTIEMBRE - 2017



PÚBLICO

## **CSIRT-CV**

CSIRT-CV es el centro de seguridad TIC de la Generalitat Valenciana, el cual está adscrito a la Dirección General de Tecnologías de la Información y las Comunicaciones dentro de la Conselleria de Hacienda y Modelo Económico. Este centro ofrece servicios de seguridad principalmente a la Administración Pública de la Comunidad Valenciana.

El trabajo plasmado en el presente entregable ha sido sometido a un riguroso proceso de calidad que abarca desde la selección de recursos humanos hasta aspectos de auditoría y control. Confiamos en que cumpla con sus expectativas y, en caso de querer indicar algún aspecto relativo a la calidad de los servicios de CSIRT-CV, le rogamos se ponga en contacto con nosotros.



## **Licencia de uso**

Este documento es de dominio público bajo licencia Creative Commons Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): no se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



## Sumario

CSIRT-CV.....	4
Introducción.....	5
Estado del arte.....	5
Matrix.....	6
Riot.....	6
Openfire.....	7
Spark.....	7
Despliegue de MATRIX y RIOT.....	8
Ventajas.....	8
Manual de instalación.....	9
Matrix server.....	9
Cliente Riot.....	14
Referencias.....	20

## 1 CSIRT-CV

CSIRT-CV es el Centro de Seguridad TIC de la Comunitat Valenciana.

Nace en junio del año 2007, como una apuesta de la Generalitat de la Comunitat Valenciana por la seguridad en la red. Se trata de una iniciativa pionera al ser el primer centro de estas características que se crea en España para un ámbito autonómico. Actualmente CSIRT-CV está adscrito a la Dirección General de Tecnologías de la Información y las Comunicaciones dentro de la Consellería de Hacienda y Modelo Económico.

CSIRT-CV ofrece servicios dentro de la Comunitat Valenciana (Alicante, Castellón y Valencia), con vocación de servicio público y sin ánimo de lucro, por lo que sus servicios se ofrecen gratuitamente. Los colectivos destinatarios de estos servicios son:

- Los ciudadanos de la Comunidad Valenciana.
- Los profesionales y empresas privadas, especialmente las de menor tamaño.
- La Administración Pública, tanto local como autonómica. Principalmente esta última por la ubicación del centro.

El principal objetivo de CSIRT-CV es contribuir a la mejora de la seguridad de los sistemas de información dentro de su ámbito, así como promover una cultura de seguridad y buenas prácticas en el uso de las nuevas tecnologías de forma que se minimicen los incidentes de seguridad y se permita afrontar de forma activa las nuevas amenazas que pudieran surgir.

CSIRT-CV ha obtenido y mantiene la certificación ISO 27001:2013 demostrando el compromiso del centro con la seguridad de la información y se reconoce el buen trabajo que el equipo viene realizando en este sentido.

## 2 Introducción

CSIRT-CV ha llevado a cabo el presente estudio sobre algunas de las herramientas open source más utilizadas en mensajería instantánea, de cara a valorar sobre todo su seguridad y usabilidad.

## 3 Estado del arte

Actualmente en el mercado existen multitud de aplicaciones destinadas al intercambio de información de forma instantánea con características muy diferentes más allá de Whatsapp y Telegram, cuyos riesgos de uso ya ha mostrado el CCN-CERT en sendos informes<sup>1</sup>.

Para llevar a cabo este estudio, CSIRT-CV ha establecido una serie de requisitos que debían cumplirse para comenzar a valorarlas:

- Posibilidad de instalación independiente en un servidor propio
- Open source
- Versión gratuita para empresas
- Comunicaciones cliente/servidor cifradas
- Chats 1 a 1 cifrados
- Chats grupales cifrados
- Aplicación web y/o para Windows / Linux / Mac
- Salas de chat públicas / de acceso restringido

---

<sup>1</sup> <https://www.ccn-cert.cni.es/seguridad-al-dia/comunicados-ccn-cert/5047-principales-riesgos-en-el-uso-de-telegram.html>  
<https://www.ccn-cert.cni.es/seguridad-al-dia/comunicados-ccn-cert/4040-principales-riesgos-en-el-uso-de-whatsapp.html>

Tras analizar diferentes soluciones, finalmente se escogieron los siguientes conjuntos de aplicaciones cliente y servidor para su estudio: Matrix y Riot por un lado, y Openfire y Spark por el otro.

A continuación un breve resumen de las principales características de las herramientas analizadas.

## **Matrix**

Matrix<sup>2</sup> es un sistema de comunicación federado de software libre, programado principalmente en Python y Javascript, interoperable y descentralizado, que permite ser utilizado para mensajería instantánea, llamadas, video llamadas, grupos, enviar archivos de cualquier tamaño y permite alojar los servidores Matrix en un servidor propio.



Matrix permite cifrar las comunicaciones, aunque esta característica podría limitar almacenar registros en el historial y podría ser contraproducente en un ambiente donde se necesiten obtener registros anteriores.

Para configurar el servidor Matrix se necesita un equipo con sistema operativo Windows, Linux o MacOS.

## **Riot**

Riot<sup>3</sup> es un cliente con soporte para Matrix que permite diferentes conversaciones e interacciones de distintos clientes en una sola aplicación.

Construido en base a las salas de chat en grupo, Riot permite compartir mensajes, imágenes, videos, archivos, interactuar con sus herramientas y acceder a todas las diferentes comunidades y aplicaciones que la componen.



Riot principalmente programado en Javascript y HTML, es compatible con MacOS, Windows y Linux, además de que se puede utilizar en el navegador o instalarlo en iOS y

2 <https://matrix.org/>

3 <https://about.riot.im/>

Android. Es de código abierto, por lo que da la posibilidad a que cualquiera lo pueda auditar y contribuir con nuevas características. Con Riot podemos, siempre que el servidor lo soporte, cifrar todos los archivos y datos de extremo a extremo, lo que significa que nadie podrá interceptar las conversaciones entre los miembros del chat. Las salas de chat se pueden configurar con diferentes niveles de confidencialidad: público a cualquier persona, limitado a un grupo, o acceso gestionado mediante invitación.

## Openfire

Openfire<sup>4</sup> (anteriormente llamado Wildfire y Jive Messenger) es un servidor de mensajería instantánea con licencia de código abierto GLP, programado en Java y que utiliza el protocolo XMPP (anteriormente llamado Jabber) con soporte para Linux, Windows y Mac.



Openfire dispone de múltiples características entre las que se encuentran: un amigable panel de administración web con interfaz para agregar plugins (lo que permite la adaptación de la funcionalidad según nuestras necesidades), el uso de cifrado SSL/TLS, la posibilidad de crear conferencias, ver estadísticas de todo tipo del servidor (mensajes, paquetes, etc), realizar clúster con múltiples servidores, interactuar con otras plataformas como Google Talk, Yahoo messenger, AIM, ICQ, Jigle, transferir archivos, enviar mensajes offline, la autenticación vía certificados, Kerberos, LDAP, PAM y Radius, almacenamiento en Active Directory, LDAP, MS SQL, MySQL, Oracle y PostgreSQL.

## Spark

Spark<sup>5</sup> es un cliente gratuito de mensajería para XMPP/Jabber, programado en Java, y de código abierto, el cual cuenta con una sencilla interfaz de usuario orientada a empresas y organizaciones, y está disponible para Windows, Linux y MacOS. Entre sus características se encuentran grupos de chat, integración con servicios de telefonía, transferencias de archivos, corrección automática, conversaciones mediante pestañas y opciones de seguridad.



4 <https://igniterealtime.org/projects/openfire/index.jsp>

5 <https://igniterealtime.org/projects/spark/>

## 4 Despliegue de MATRIX y RIOT

Como hemos enumerado, en el estudio realizado se han encontrado diferentes aplicaciones que cumplen con los requisitos establecidos.

Tras valorar individualmente el funcionamiento de cada una de las aplicaciones citadas, finalmente se ha concluido que las que más se adecuan a las necesidades de comunicación segura y operatividad son la combinación SERVIDOR MATRIX con CLIENTE RIOT, por encima del SERVIDOR OPENFIRE con CLIENTE SPARK también valorados.

### **Ventajas**

Hemos elegido Matrix porque cumple con todos los requisitos que necesitamos: utiliza un estándar abierto para comunicaciones interoperable y descentralizado, es compatible con protocolos antiguos y nuevos como XMPP, Slack, Skype y Lync, tiene soporte de una gran comunidad, es muy usable, tiene un manejo sencillo y sobre todo permite cifrar las comunicaciones.

Además permite llevar un control de las comunicaciones ya que utilizamos un servidor propio para que la información este bajo nuestro control, y el software utilizado es de código abierto, permitiendo ser auditado con más facilidad.

Al utilizar esta herramienta de mensajería tenemos la posibilidad de crear grupos de chat, por ejemplo un grupo para cada proyecto en el que se esté trabajando.

Su uso está respaldado por grandes empresas como RedHat, Ericsson, Amdocs, Deutsche Telecom y Mozilla.



## 5 Manual de instalación

Para las pruebas se ha elegido hacer el despliegue de la plataforma Matrix en un equipo con Centos 7 y la instalación del cliente Riot en un equipo Windows 10 de 64 bits.

### Matrix server

A continuación se indican los pasos necesarios para la instalación del servidor Matrix en una distribución Centos 7. También existe la posibilidad de instalarlo en Ubuntu o Debian, ArchLinux, Mac Os X, Raspbian, openSUSE, e incluso OpenBSD.

Primero se tiene que instalar las dependencias requeridas por el servidor Matrix:

```
$ sudo yum install libtiff-devel libjpeg-devel libzip-devel \  
    freetype-devel lcms2-devel libwebp-devel tcl-devel tk-devel \  
    redhat-rpm-config python-virtualenv libffi-devel openssl-devel  
  
$ sudo yum groupinstall "Development Tools"
```

La instalación solicitará confirmación de descarga e instalación de paquetes, y también nos solicitará que confirmemos la importación de la clave GPG.

A continuación se instala el servidor synapse<sup>6</sup> :

```
$ virtualenv -p python2.7 ~/.synapse  
$ source ~/.synapse/bin/activate  
$ pip install --upgrade pip  
$ pip install --upgrade setuptools  
$ pip install https://github.com/matrix-org/synapse/tarball/master
```

Esto nos instalará el servidor en un entorno virtual ubicado en la carpeta `~/.synapse` como se indica en el primer comando. Este directorio puede modificarse a placer y ser cambiado a posteriori.

<sup>6</sup> <https://github.com/matrix-org/synapse#synapse-installation>

## Configuración inicial

Una vez concluye la instalación se debe generar la configuración inicial del servidor con los siguientes parámetros:

- El nombre del servidor determinará la ruta completa de los nombres de usuario, que tendrán el siguiente formato: @user:my.domain.name
- El directorio donde se guardará el archivo de configuración generado.
- Deshabilitar el envío de estadísticas anónimas de uso de la aplicación.

```
$ cd ~/.synapse
$ python -m synapse.app.homeserver \
  --server-name my.domain.name \
  --config-path homeserver.yaml \
  --generate-config \
  --report-stats=no
```

## Consideraciones sobre el fichero de configuración:

Antes de iniciar la aplicación es altamente recomendable revisar el fichero de configuración que se acaba de generar para adaptarlo a nuestras necesidades. A continuación exponemos algunos de los parámetros de configuración más significativos:

- Junto con el archivo de configuración se generarán un conjunto de claves de certificados SSL autofirmadas, las cuales se deberán resguardar o reemplazar con un conjunto de claves generadas por entidades de certificación (campos *tls\_certificate\_path*, *tls\_private\_key\_path* y *tls\_dh\_params\_path*).
- Los puertos por defecto que el servidor utilizará para escuchar (apartado "*listeners*" del fichero de configuración) son el puerto 8448 para https (*tls: true*) y el 8008 para http (*tls: false*). Estos puertos, así como las IP específicas en las que escucha el servidor, pueden ser modificadas en el archivo de configuración.
- El tamaño máximo de fichero que se puede subir al servidor está fijado por defecto en 10MB. Este valor se puede modificar en el parámetro *max\_upload\_size*.
- En el apartado *e-mail* (comentado por defecto) se puede configurar una pasarela de correo para enviar notificaciones que necesiten ser enviadas por el servidor.
- El parámetro *redact\_content* (comentado por defecto) del apartado *push* del fichero de configuración hace que las notificaciones *push* enviadas a dispositivos móviles (que deben ser enviadas a dichos dispositivos a través de servidores de terceros) no incluyan contenido de los mensajes, mejorando la privacidad global de

las conversaciones de los usuarios del servidor.

### Recomendaciones opcionales

- Para aumentar la seguridad podemos bloquear las federaciones, es decir el proceso mediante el cual los usuarios de diferentes servidores pueden participar en la misma sala comentando las siguientes líneas:

```
# trusted_third_party_id_servers:  
#   - matrix.org  
#   - vector.im  
#   - riot.im
```

### Inicio de la aplicación

Una vez adaptado el fichero de configuración a nuestras necesidades, podemos proceder a iniciar la aplicación.

```
A ejecutar sólo si no estamos dentro del virtualenv:  
$ source ~/.synapse/bin/activate  
  
$ synctl start ~/.synapse/homeserver.yaml
```

Antes de poder acceder a la aplicación desde un equipo remoto, deberemos crear las reglas de cortafuegos específicas para permitir el acceso externo a la aplicación, por lo que hay que asegurar la conectividad entre el servidor y los clientes comprobando que se comunican entre ellos. Para ello habilitaremos las siguientes reglas en el firewall de la maquina Centos 7 y habilitaremos en el firewall de la organización el tráfico entrante al servidor hacia del puerto 8448 (SSH) y el 8008 (HTTP opcional).

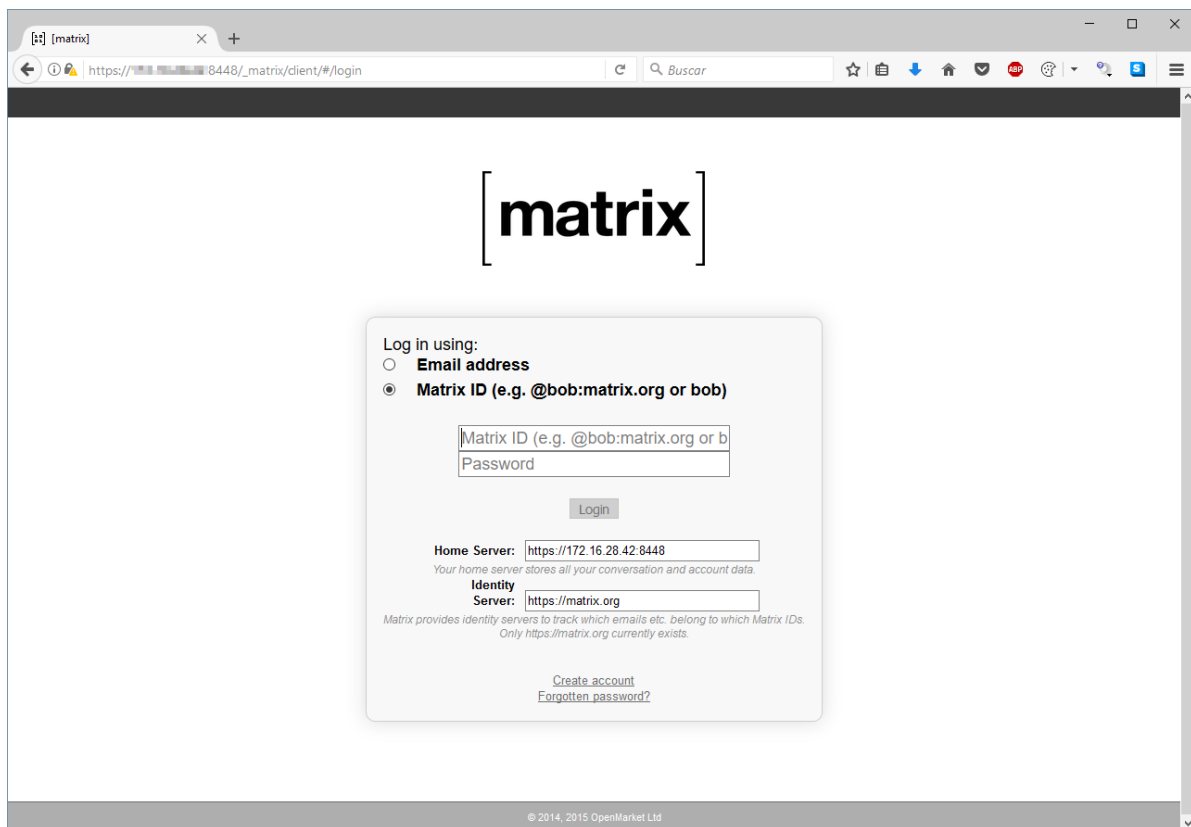
```
Comprobamos en que zona esta activo el firewall  
$ firewall-cmd --get-active-zones  
public  
  interfaces:enp0s8 enp0s3  
$ firewall-cmd --zone=public --add-port=8448/tcp --permanent  
$ firewall-cmd --zone=public --add-port=8008/tcp --permanent #opcional  
$ firewall-cmd --reload
```

Tras cualquier cambio en el fichero de configuración, deberemos reiniciar el servidor para que los cambios realizados tengan efecto:

```
A ejecutar sólo si no estamos dentro del virtualenv:  
$ source ~/.synapse/bin/activate  
  
$ synctl restart ~/.synapse/homeserver.yaml
```

Una vez iniciado el servicio, podemos acceder a él mediante un navegador web, en cualquiera de los puertos que hemos configurado. Las URL de acceso por defecto son:

- `http:// <ip> :8008`
- `https:// <ip> :8448`



*Ilustración 1: Ventana principal de Matrix*

## Registro de nuevos usuarios

El proceso para crear usuarios puede hacerse de dos maneras diferentes:

- Mediante la interfaz web. Para ello es necesario habilitar el registro de usuarios en el fichero de configuración creado anteriormente, modificando la siguiente línea: `enable_registration: true`
- Mediante comandos en la consola del servidor. Los pasos a seguir para crear un nuevo usuario mediante consola son los siguientes:

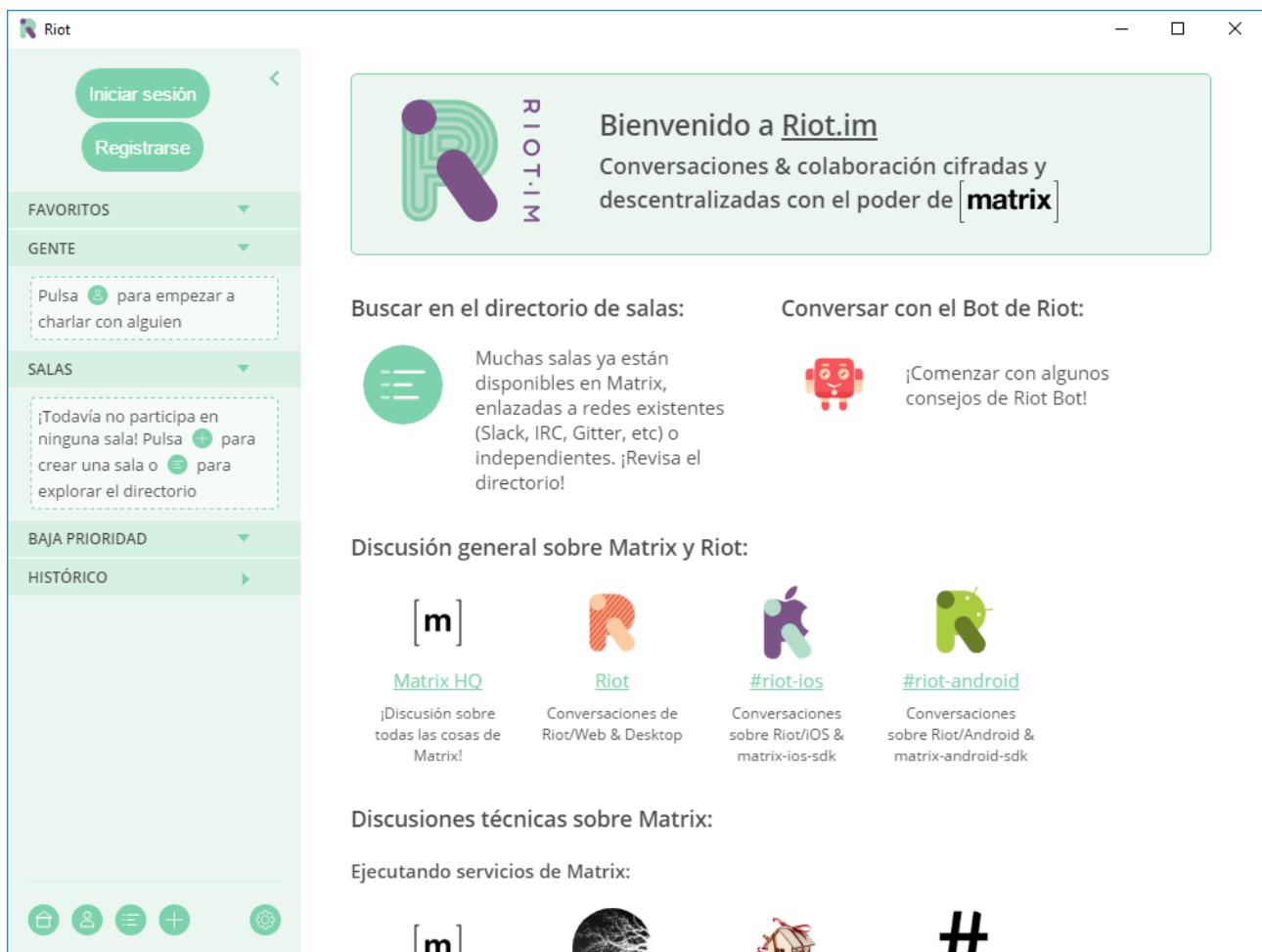
```
A ejecutar sólo si no estamos dentro del virtualenv:  
$ source ~/.synapse/bin/activate  
  
$ register_new_matrix_user -c ~/.synapse/homeserver.yaml  
http://localhost:8008  
New user localpart: csirt-cv  
Password:  
Confirm password:  
Make admin [no]: yes/no  
Success!
```

En esta parte se debe introducir un nombre de usuario, una contraseña, su confirmación, y elegir si será un usuario administrador.

## Ciente Riot

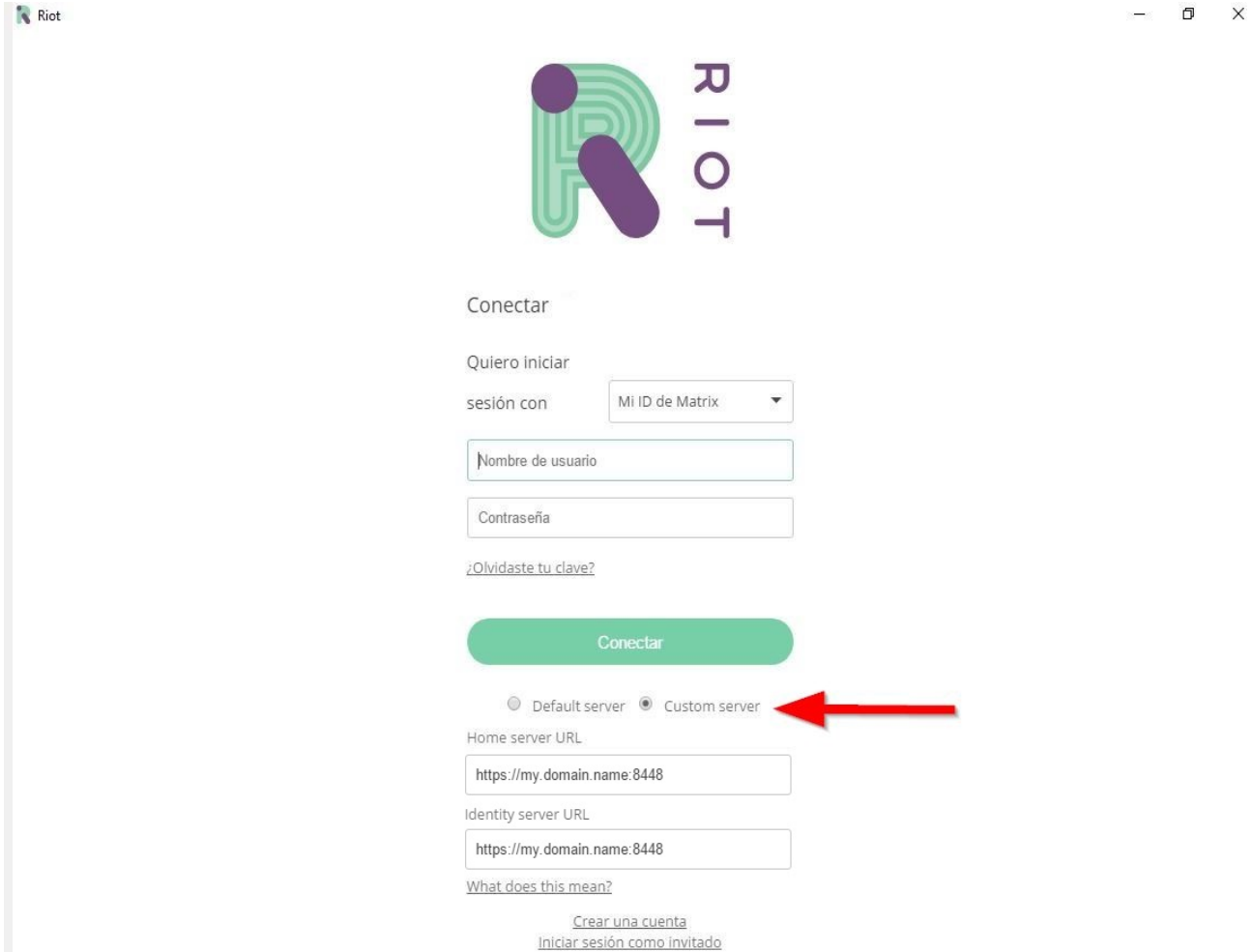
A continuación, se indican los pasos necesarios para la instalación del cliente Riot en un equipo con Windows 10 de 64 bits. El método de instalación y ejecución es similar al de otros sistemas operativos como iOS, Android, MacOS y Linux, e incluso a la versión de navegador.

1. Desde la página oficial de Riot (<https://about.riot.im>) es posible hacer la descarga directa de sus diferentes clientes.
2. Una vez descargado, se ejecuta el instalador que automáticamente instalará el cliente sin solicitar ninguna interacción y lo abrirá.




*Ilustración 2: Pantalla principal de la aplicación de escritorio Riot*

3. Para iniciar sesión es necesario indicar a la aplicación que queremos conectar a un servidor personalizado, e introducir los detalles de nuestro servidor:



Riot



Conectar

Quiero iniciar sesión con

[¿Olvidaste tu clave?](#)

Default server  Custom server

Home server URL

Identity server URL

[What does this mean?](#)

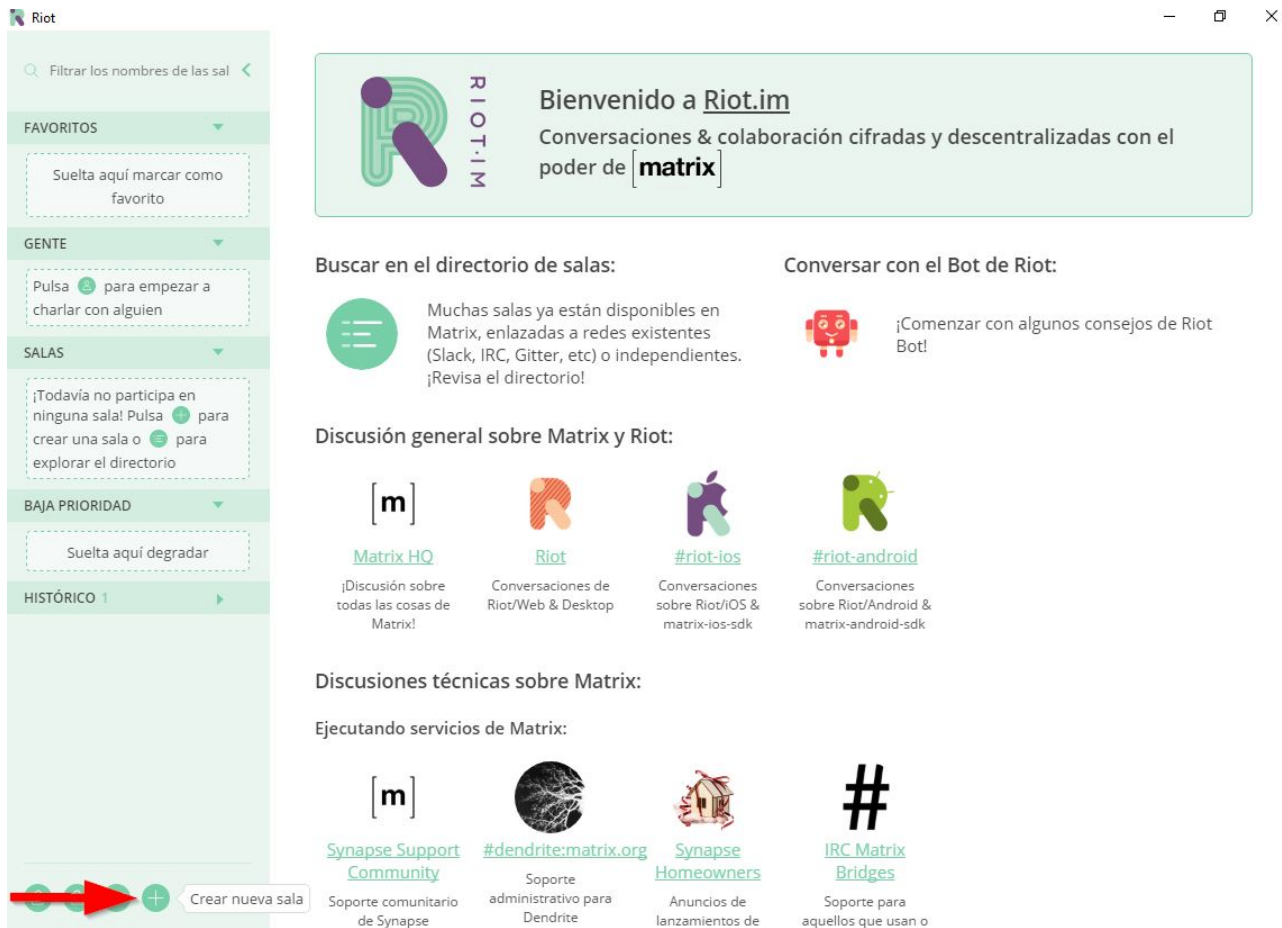
[Crear una cuenta](#)  
[Iniciar sesión como invitado](#)

*Ilustración 3: Pantalla de inicio de sesión de la aplicación Riot*

En *Home Server URL* y en *Identity Server URL* introduciremos el nombre de nuestro servidor previamente configurado, en nuestro caso (<https://my.domain.name:8448>)

Una vez tenemos todo el ecosistema de Matrix - Riot funcionando veamos cómo funciona:

## Creación de nueva sala



The screenshot shows the Riot.im web interface. On the left is a sidebar with navigation options: FAVORITOS, GENTE, SALAS, BAJA PRIORIDAD, and HISTÓRICO. The main content area displays a welcome message and several chat room suggestions. A red arrow points to the '+ Crear nueva sala' button at the bottom of the sidebar.

**Bienvenido a Riot.im**  
Conversaciones & colaboración cifradas y descentralizadas con el poder de **[matrix]**

**Buscar en el directorio de salas:**  
Muchas salas ya están disponibles en Matrix, enlazadas a redes existentes (Slack, IRC, Gitter, etc) o independientes. ¡Revisa el directorio!

**Conversar con el Bot de Riot:**  
¡Comenzar con algunos consejos de Riot Bot!

**Discusión general sobre Matrix y Riot:**

- [m] Matrix HQ**: ¡Discusión sobre todas las cosas de Matrix!
- Riot**: Conversaciones de Riot/Web & Desktop
- #riot-ios**: Conversaciones sobre Riot/iOS & matrix-ios-sdk
- #riot-android**: Conversaciones sobre Riot/Android & matrix-android-sdk

**Discusiones técnicas sobre Matrix:**

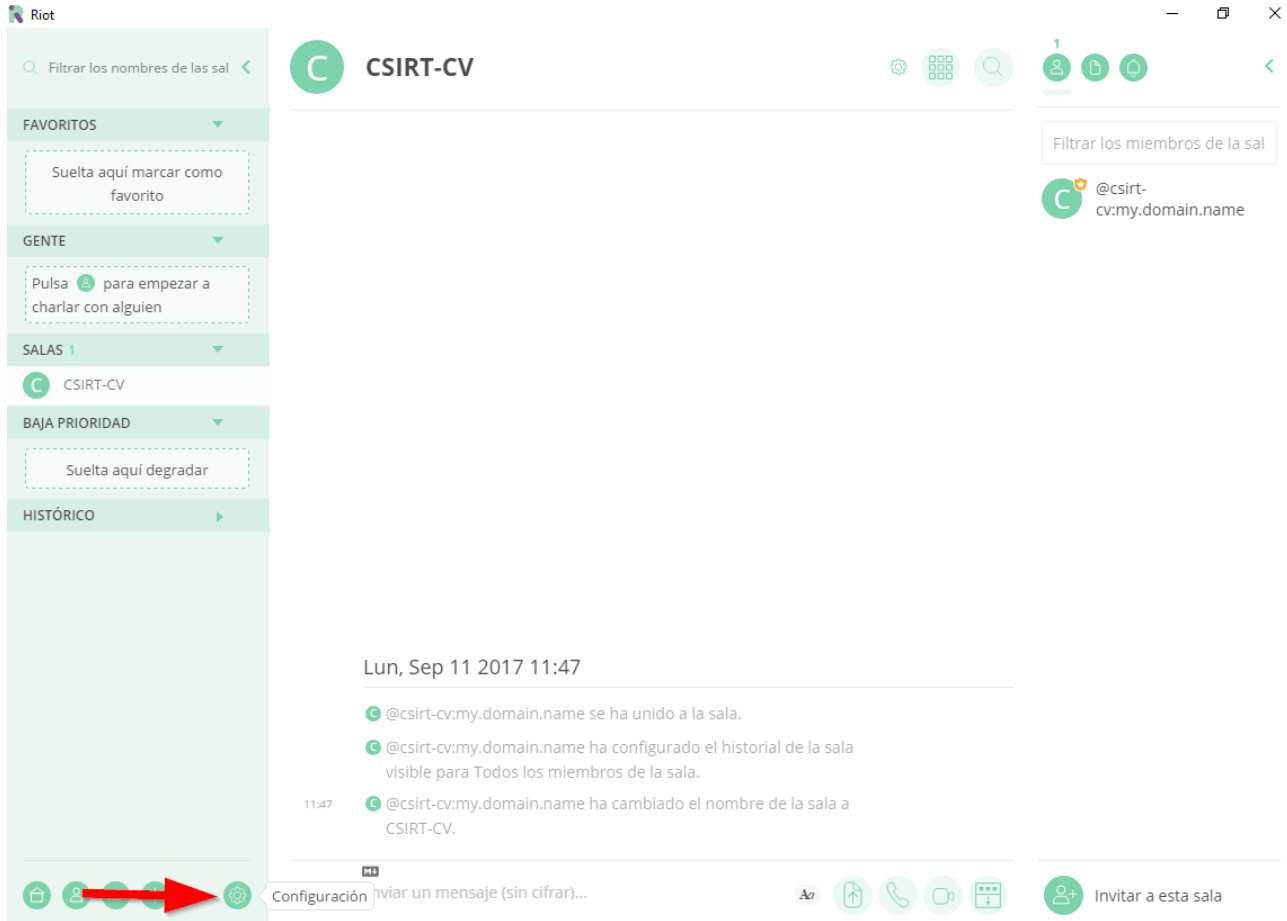
Ejecutando servicios de Matrix:

- [m] Synapse Support Community**: Soporte comunitario de Synapse
- #dendrite:matrix.org**: Soporte administrativo para Dendrite
- Synapse Homeowners**: Anuncios de lanzamientos de
- # IRC Matrix Bridges**: Soporte para aquellos que usan o

*Ilustración 4: Crear nueva sala*



Hacemos clic en *Crear nueva sala* y le indicamos un nombre, por ejemplo CSIRT-CV.



*Ilustración 5: Sala de chat creada*

Para ampliar la seguridad en la sala realizaremos las siguientes modificaciones:

*¿Quién puede acceder a esta sala?*

*Sólo los usuarios que han sido invitados (por defecto)*

*Seleccionar "Habilitar encriptación"*

*¿Quién puede leer el historial?*

*Sólo para miembros (desde que se conectaron)*

*URL previews*

*Seleccionar Disable URL previews*

*- Disable URL previews by default for participants in this room.*

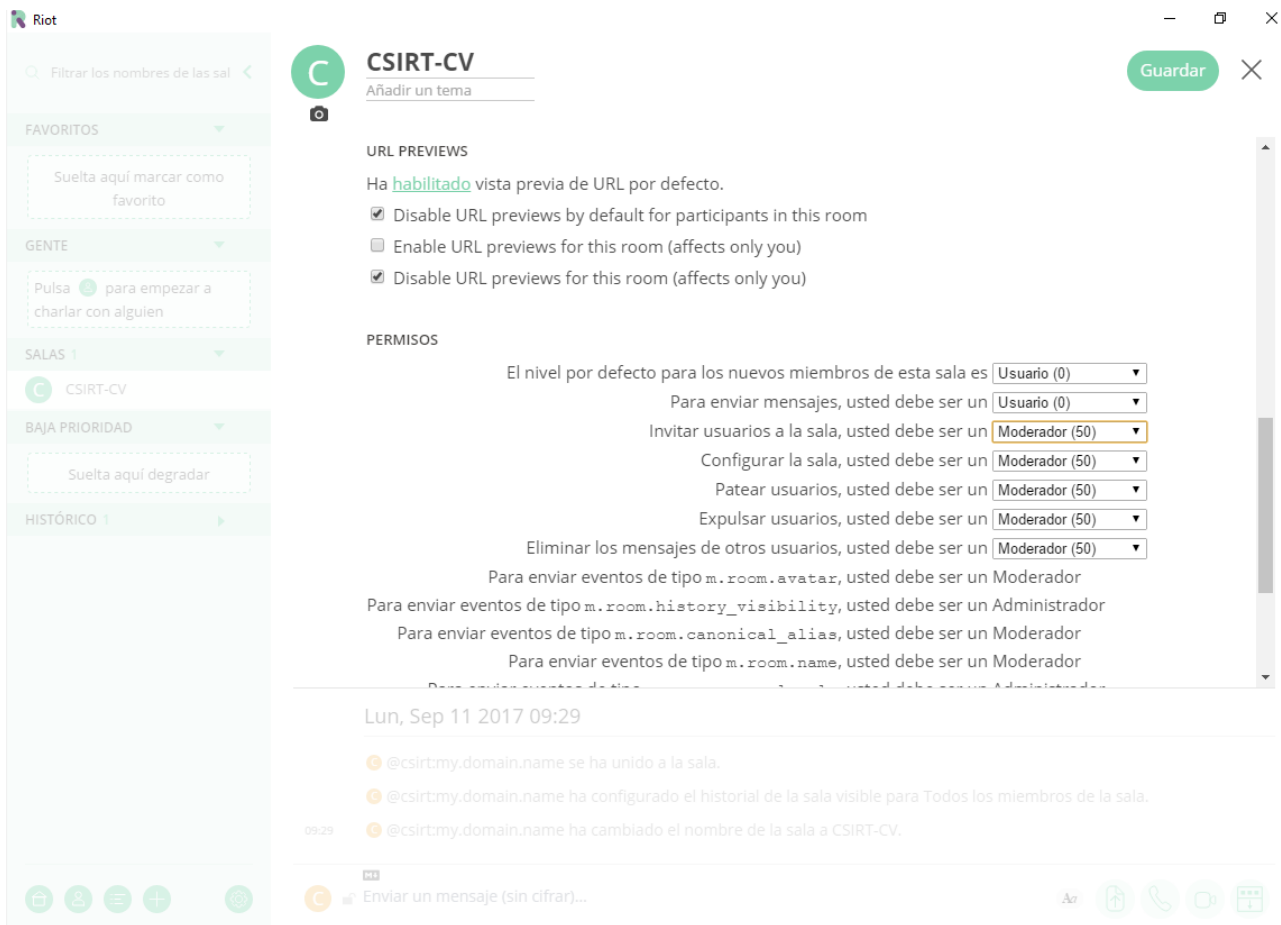
*- Disable URL previews for this room (affects only you)*

*Invitar usuarios a esta sala -> Moderador*

Hacer Clic en "Guardar"



*Ilustración 6: Modificación configuración seguridad*



*Ilustración 7: Modificación configuración seguridad*

## 6 Referencias

<https://matrix.org/>

<https://about.riot.im/>

<https://github.com/matrix-org/synapse#debian>

<https://blog.cryptoaustralia.org.au/2017/03/21/run-your-end-to-end-encrypted-chat-server-matrix-riot/>

<http://interorganic.com.ar/josx/riot.pdf>

<https://igniterealtime.org/projects/openfire/index.jsp>

<https://igniterealtime.org/projects/spark/>

<https://gist.github.com/gszathmari/e634f4156b872478ecb6a184521489c0>