

# **EJABBERD & PIDGIN**

## **TMD**



## I.1 Parte teórica

La mensajería instantánea ha evolucionado con múltiples protocolos y aplicaciones. Entre ellos, XMPP y servidores como ejabberd han sido clave para sistemas seguros y descentralizados. Además, clientes como Pidgin permiten interactuar con estos protocolos.

Ejabberd es un servidor XMPP de código abierto escrito en Erlang, diseñado para ser altamente escalable y confiable. Se utiliza en plataformas de mensajería instantánea y comunicaciones en tiempo real. Su arquitectura permite manejar millones de conexiones simultáneas, siendo ideal para grandes implementaciones.

Existen alternativas como Prosody, un servidor ligero escrito en Lua, y Openfire, basado en Java, que ofrece una interfaz gráfica intuitiva.

Ya que mencionamos el protocolo XMPP vamos a hablar sobre él. XMPP (Extensible Messaging and Presence Protocol) es un protocolo abierto basado en XML para mensajería instantánea, presencia y comunicación en tiempo real. Se diseñó para proporcionar un sistema descentralizado y seguro, permitiendo la interoperabilidad entre diferentes servidores y clientes. XMPP es la base de muchas plataformas de chat, como WhatsApp en sus inicios.

Además XMPP nos servirá para nosotros a la hora de configurar la segunda parte de la nuestra actividad, Pidgin. Pidgin es un cliente de mensajería instantánea de código abierto que soporta múltiples protocolos, incluidos XMPP, IRC y AIM. Su ventaja principal es permitir el uso de diferentes cuentas de chat en una sola aplicación.

Alternativas a Pidgin incluyen Gajim, un cliente XMPP ligero para Linux y Windows, y Dino, diseñado con un enfoque en la seguridad. Y respecto a las otras alternativas más conocidas no podemos dejar pasar desapercibidos tales como Signal, Telegram y WhatsApp. Signal está centrado en la privacidad con cifrado de extremo a extremo, Telegram distingue por su rapidez y los canales públicos, Whatsapp es popular por su facilidad de uso y cifrado integrado.

## I.2 Después de la configuración

Aunque esta parte viene después de la configuración la hemos unido a la parte teórica para la comodidad del lector.

Al configurar ejabberd es clave conocer los puertos que utiliza, vamos a ver los puertos 5222, 5223, 5280, 5443 y 5269.

El puerto 5222 es un puerto estándar para conexiones de cliente XMPP/Jabber, simple o STARTTLS.

El puerto 5223 sirve para las conexiones cliente Jabber utilizando el antiguo método SSL.

El puerto 5280 es usado para interfaces web de administración de servidores XMPP.

El puerto 5443 es un puerto estándar cifrado utilizado para la interfaz web de ejabberd.

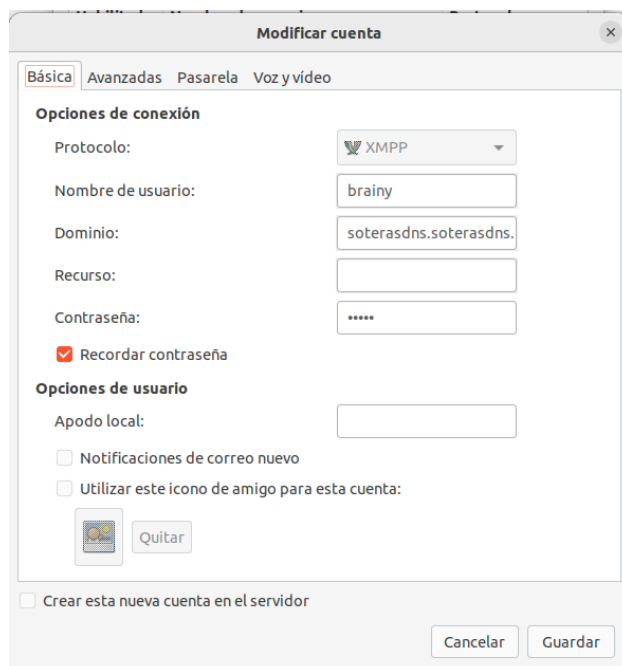
El puerto 5269 es el puerto para comunicación entre servidores XMPP.

## I.3 Pidgin - preguntas frecuentes

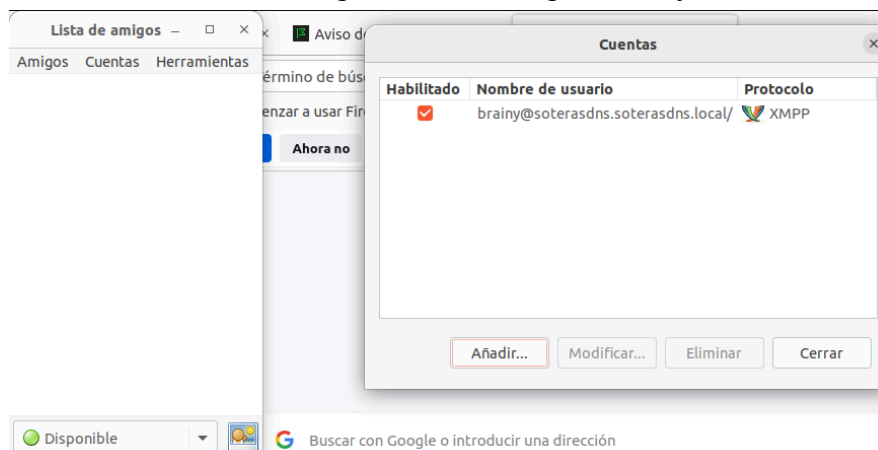
En este apartado vamos a solucionar los problemas comunes y hablaremos de la configuración básica de Pidgin.

Para conectarse a un servidor XMPP en Pidgin hay que seguir unos pasos concretos:

Abre Pidgin y dirígete a Cuentas > Gestionar cuentas. Haz clic en Añadir.

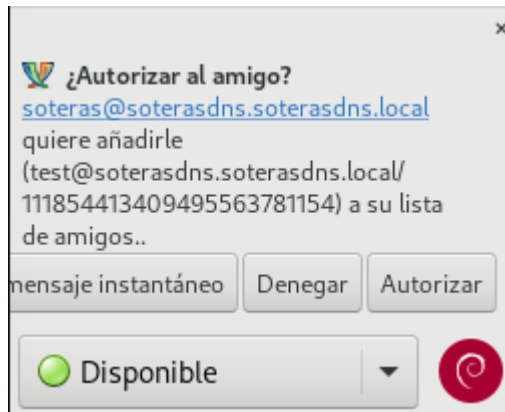


Selecciona XMPP como protocolo e ingresa tu nombre de usuario, dominio y contraseña. Por último guarda la configuración y conéctate.

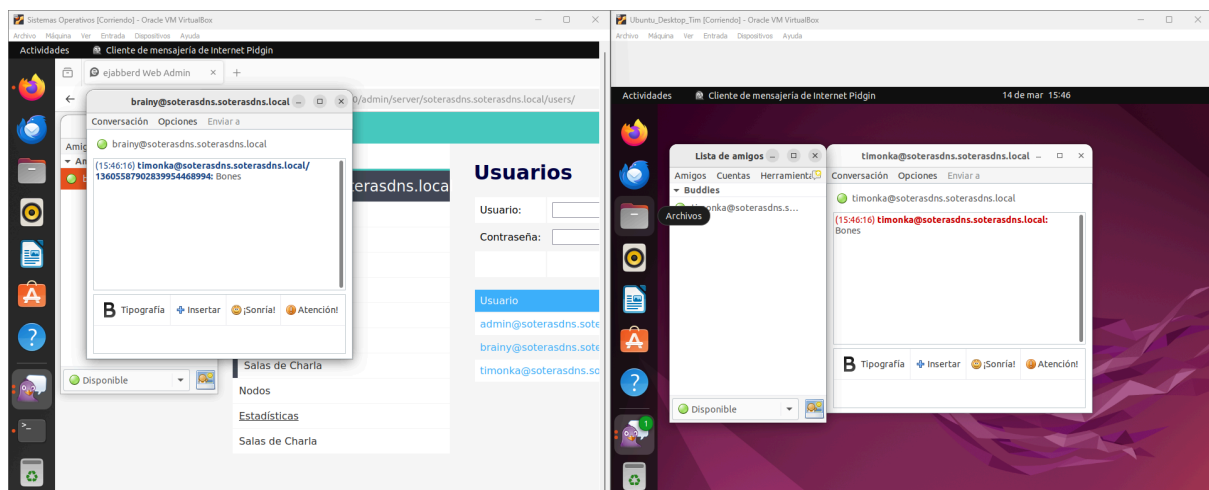


Para establecer la conexión con otro usuario y poder empezar a chatear abre la sección de “amigos” y clicas en “añadir un amigo”. Introduce el nombre de usuario del amigo que se quiere añadir junto al dominio, opcionalmente se puede añadir el apodo.

Ejemplo de una solicitud de amistad:



Finalmente cuando tenemos una conexión establecida entre ambos clientes el resultado final debería ser así:



En la captura de pantalla se pueden apreciar dos máquinas virtuales diferentes con el chat funcional entre ambas.

## SOLUCIÓN DE PROBLEMAS

Si pidgin no logra conectarse al servidor XMPP, sigue unos simples pasos para encontrar y mitigar el problema. Primero, verifica las credenciales y el dominio del servidor, a continuación, asegúrate de que el servidor XMPP está funcionando. Luego, revisa la configuración de firewall, que no provoque conflictos de conexión. Finalmente, asegúrate de que los puertos están bien configurados en Pidgin.

## Instalación en Docker

Para instalar ejabberd en docker vamos a completar los siguientes pasos para no perder nada de vista.

Primero comprueba el estado de docker y docker-compose mediante los comandos “docker --version” y “docker-compose --version”.

A continuación vamos a crear un archivo docker-compose.yml definiendo la configuración del ejabberd.

Modelo ejemplo del archivo docker-compose.yml

```
version: '3'

services:
  ejabberd:
    build: .
    container_name: ejabberd
    environment:
      - EJABBERD_HOST=servergea.servergea.local
      - EJABBERD_DOMAIN=servergea.servergea.local
      - EJABBERD_ADMIN=admin@servergea.servergea.local
      - EJABBERD_ADMIN_PASSWORD=password
    ports:
      - "5222:5222"
      - "5269:5269"
      - "5280:5280"
    volumes:
      - ejabberd_data:/var/lib/ejabberd
      - ejabberd_logs:/var/log/ejabberd
    restart: always

volumes:
  ejabberd_data:
  ejabberd_logs:
```

A continuación iniciamos el servicio ejabberd con “docker-compose up -d”. Este comando descargará la imagen de ejabberd, creará los volúmenes y ejecutará el contenedor. Para verificar que el contenedor está en ejecución escribimos “docker ps”. El resto de la configuración es idéntica a la configuración de ejabberd estándar.

## Conclusiones

Aunque no necesitamos la aplicación de un servicio de mensajería instantánea en nuestro proyecto, me ha resultado interesante su configuración y los resultados obtenidos. Lo considero bastante útil desde la perspectiva de su aplicación práctica en futuros proyectos.

## Webgrafía

[Ejabberd](#)

[XMPP](#)

[Pidgin](#)

[Puertos](#)