

MPT02. Servicios de mensajería instantánea

1. Introducción

El Servicio de mensajería instantánea es una forma de comunicación en tiempo real entre dos o mas personas basada en texto. El texto es enviado a través de dispositivos conectados ya sea a una red como Internet, o datos móviles sin importar la distancia que exista entre los dos (o mas) dispositivos conectados.

La mayoría usan redes propietarias de los diferentes software que ofrecen este servicio en cada maquina diferente. También hay programas de mensajería que sirven de gran ayuda para que llegue su envío de forma instantánea que utilizan el protocolo abierto XMPP, en este caso es el tipo de programa que vamos a utilizar en esta practica.

El protocolo XMPP (Extensible Messaging and Presence Protocol) anteriormente llamado Jabber. Es un protocolo abierto y extensible basado en XML, ideado para mensajería instantánea.

2. OpenFire

Este es el sistema de mensajería instantánea que usaremos, hecho en java y que utiliza el protocolo XMPP, donde podemos tendremos nuestro propio servidor donde podremos administrar usuarios, compartir archivos, mensajes broadcast, grupos, etc.

Para ello hemos utilizado un Ubuntu Server 20.04 LTS y hemos instalado un servicio de mensajería instantánea con openfire. Las características principales de openfire es el panel de administracion web, adaptable según las necesidades, interaccion con MSN, Google Talk, Yahoo messenger, control total de clientes, compresion de datos, etc.

La instalacion es muy sencilla lo primero de todo descargamos openfire con **'wget --content-disposition https://www.igniterealtime.org/downloadServlet?filename=openfire/openfire_4_6_2.tar.gz**

,

Extraemos el contenido del paquete que acabamos de descargar en su ubicación de instalación que sería la carpeta **'/opt/'**, **'sudo tar xf openfire_4_6_2.tar.gz -C /opt/'** creamos un enlace simbólico para configurar lo como un servicio del sistema **'sudo ln -s /opt/openfire/bin/openfire /etc/init.d/'**

Creamos los scripts básicos para la administración del nuevo servicio **'sudo update-rc.d openfire defaults'** y lanzamos el servicio con **'sudo systemctl start openfire'**.

Para que el servicio se inicie de forma automática cada vez que iniciemos el servidor utilizamos **'sudo systemctl enable openfire'**

Habilitamos el puerto 9090 con el firewall de ubuntu **'sudo ufw allow 9090/tcp'**

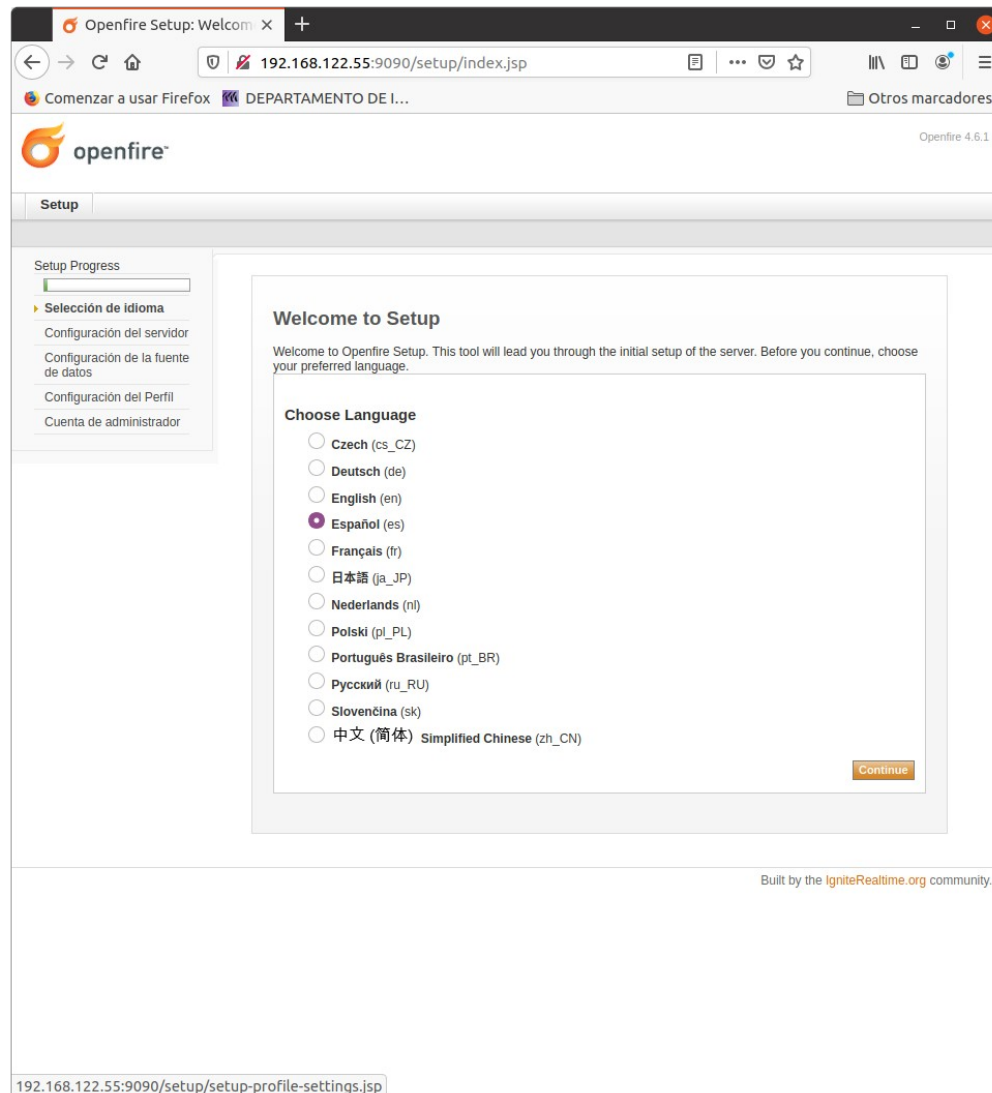
Ahora crearemos y configuramos la base de datos que utilizara openfire en este caso PostgreSQL.

La instalación sería con **'sudo apt install postgresql postgresql-contrib'**

Y para openfire creamos un rol con contraseña: **'sudo -u postgres createuser -P openfire'**

y su base correspondiente: **'sudo -u postgres createdb openfire -O openfire'**

Una vez hecho todo esto podremos acceder al instalador web de OpenFire entrando con el dominio asignado o con <http://localhost:9090> :



Después de elegir el idioma configuramos el dominio del servicio de mensajería:

Openfire Configuración: x +

192.168.122.55:9090/setup/setup-host-settings.jsp

Comenzar a usar Firefox DEPARTAMENTO DE I...

Otros marcadores

openfire® Openfire 4.6.1

Configuración

Progreso de la Instalación

✓ Selección de idioma

Configuración del servidor

Configuración de la fuente de datos

Configuración del Perfil

Cuenta de administrador

Configuración del Servidor

A continuación se muestra la configuración de red del servidor.

Dominio:

Host del Servidor (FQDN):

Puerto de la Consola de Administración:

Puerto de la Consola de Administración Segura:

Cifrar Propiedades con:

☒ Blowfish

☐ AES

Clave de Cifrado de Propiedades:

Built by the IgniteRealtime.org community.

192.168.122.55:9090/setup/setup-profile-settings.jsp

Configuramos después la fuente de datos:

La opción recomendada es la conexión estándar, ya que utilizara el servicio de bases de datos que hayamos configurado anteriormente, lo que aumentara el rendimiento.

Openfire Configuración: X

192.168.122.55:9090/setup/setup-datasource-settings.jsp

Comenzar a usar Firefox

DEPARTAMENTO DE I...

Otros marcadores

Openfire 4.6.1

Configuración

Progreso de la Instalación

✓ Selección de idioma

✓ Configuración del servidor

Configuración de la fuente de datos

Configuración del Perfil

Cuenta de administrador

Configuración de la fuente de datos

Elija como quiere conectarse a la base de datos Openfire.

☒ **Conexión Estándar**
Usa una base de datos externa con el pool de conexiones interno.

☐ **Base de datos interna**
Usa una base de datos interna (HSQLDB). Esta opción no requiere la configuración de una base de datos externa y permite poner al servidor en producción rápidamente. Sin embargo dicha base de datos no se desempeña tan bien como una base de datos externa.

Continuar

Built by the IgniteRealtime.org community.

Openfire Configuración: X

192.168.122.55:9090/setup/setup-datasource-standard.jsp

Comenzar a usar Firefox

DEPARTAMENTO DE I...

Otros marcadores

Openfire 4.6.1

Configuración

Progreso de la Instalación

✓ Selección de idioma

✓ Configuración del servidor

Configuración de la fuente de datos

Configuración del Perfil

Cuenta de administrador

Configuración de la fuente de datos - Conexión Estándar

Indique un driver JDBC y las propiedades de la conexión a su base de datos. Si necesita más información sobre este proceso por favor vea la documentación incluida sobre bases de datos con Openfire.
Nota: La distribución de Openfire incluye scripts de configuración para lasbases de datos más populares en [Openfire_HOME]/resources/database.

Drivers Predefinidos : PostgreSQL

Clase del Driver JDBC: org.postgresql.Driver

URL de la Base de Datos: jdbc:postgresql://192.168.122.55:5432/openfire

Nombre de usuario: openfire

Contraseña:

Minimum Connections: 5

Maximum Connections: 25

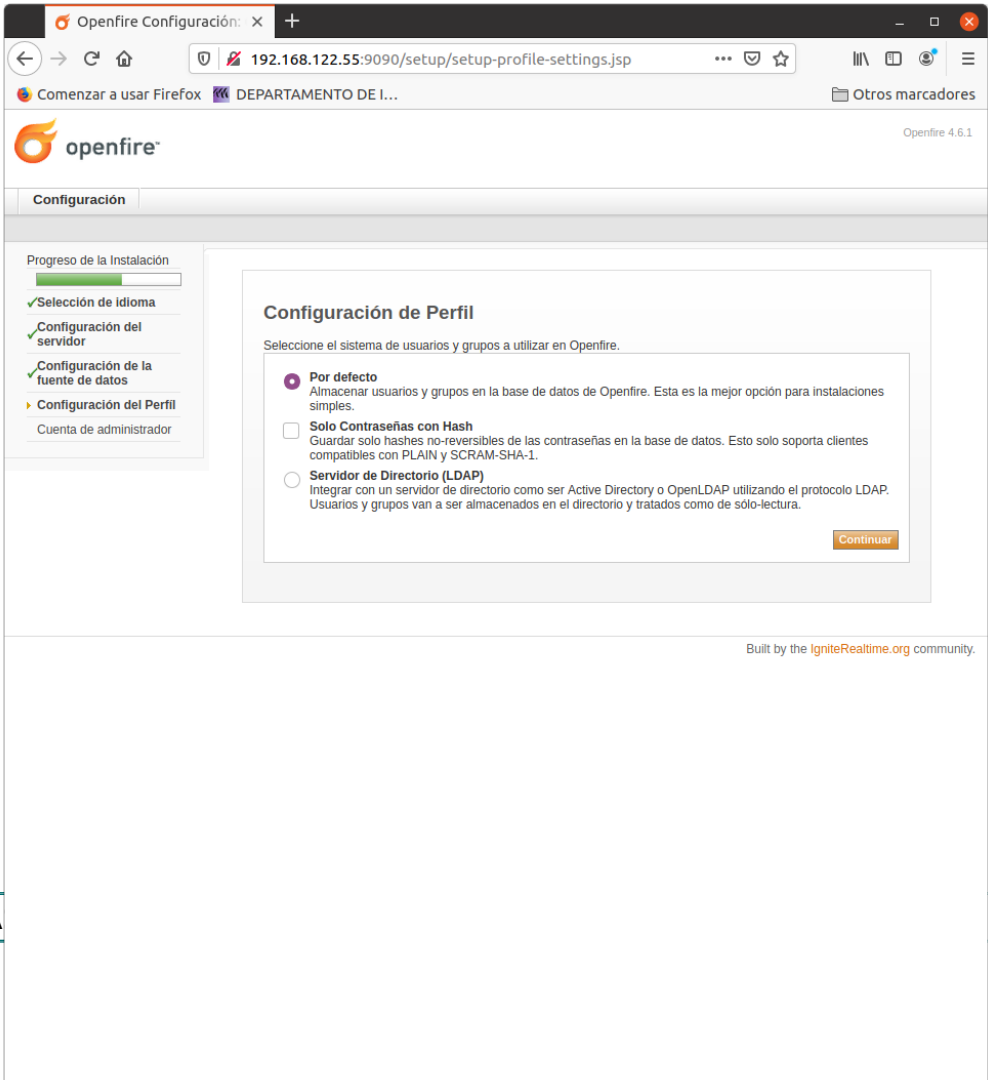
Tiempo de Vida de la Conexión: 1.0 Days

Nota: la conexión a la base de datos puede tardar entre 30 y 60 segundos.

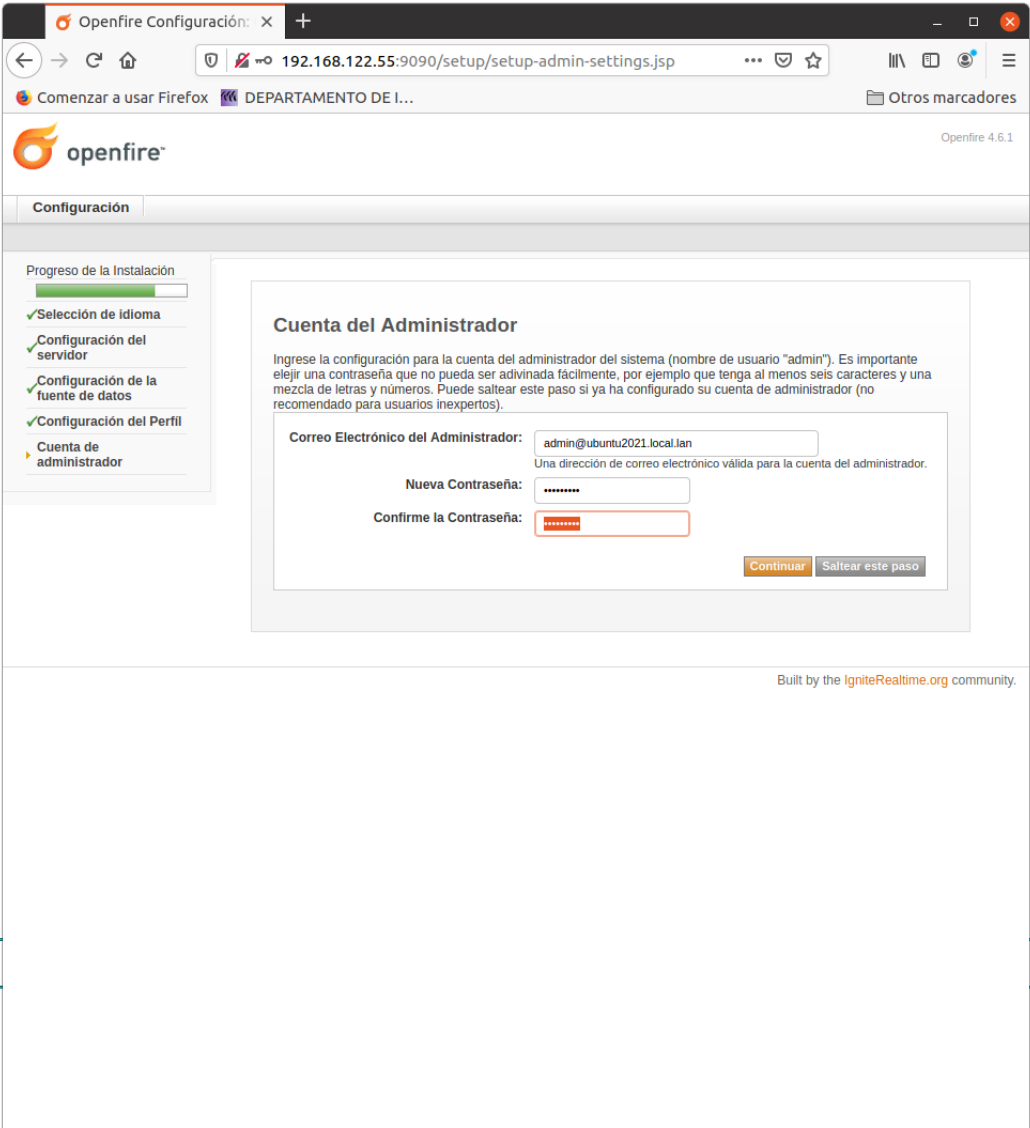
Continuar

Built by the IgniteRealtime.org community.

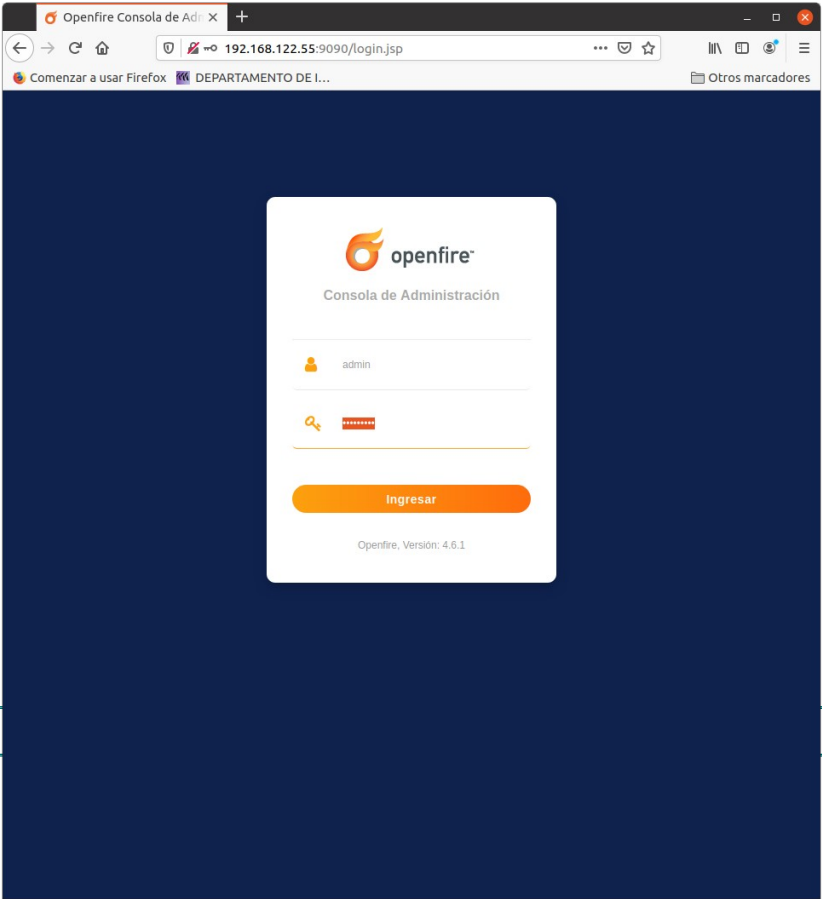
192.168.122.55:9090/setup/setup-datasource-settings.jsp



Continuamos
con el modo de
almacenar
usuarios y
contraseñas:



Añadimos
usuario
administrador
y contraseña y
la instalación
estaría
finalizada.



Openfire Consola de Adm X

192.168.122.55:9090/index.jsp

Comenzar a usar Firefox

DEPARTAMENTO DE I...

Otros marcadores

Openfire 4.6.1
Conectado como admin - Salir
Estado de clustering - Deshabilitado

Servidor

Usuarios/Grupos

Sesiones

Conferencias

Plugins

Administración del Servidor

Configuración del Servidor

Certificados TLS/SSL

Servicios de Multimedia

PubSub

Configuración del Servidor

Propiedades del Sistema

Idioma y Tiempo

Clúster

Caches del Sistema

Base de Datos

Registro

Configuración de Correo

Configuración de SMS

Ver Auditoría de Seguridad

Configuración del Servidor

A continuación están las propiedades de este servidor. Presione en el botón "Editar Propiedades" para cambiar algunas de las propiedades del servidor. Algunas configuraciones no pueden ser cambiadas.

Propiedades del Servidor

Tiempo de Actividad del Servidor: 23 minutos -- started 27 ene. 2021 9:03:22

Versión: Openfire 4.6.1

Ruta al servidor: /opt/openfire

Nombre del Servidor: ubuntu2021.local.lan

Ambiente

Versión de Java: 11.0.9.1 Ubuntu -- OpenJDK 64-Bit Server VM

Servidor de Aplicaciones: jetty/9.4.35.v20201120

Nombre del Host: ubuntu2021.local.lan

SO / Hardware: Linux / amd64

Idioma / Huso Horario: es / tiempo universal coordinado (0 GMT)

Dueño del proceso del SO: root

Memoria de Java

40,68 MB of 953,19 MB (4,3%) used

Novedades de Ignite Realtime

Openfire 4.6.1 is released, 6 ene. 2021

Monitoring Openfire plugin v2.2.0 released, 6 ene. 2021

Openfire Meetings becomes Pádé, 31 dic. 2020

Openfire Pádé 1.2.0 released!, 10 dic. 2020

Smack 4.4.0 released, 9 dic. 2020

Openfire Pádé 1.1.1 released!, 24 nov. 2020

Ignite Realtime Bamboo Usage and Access, 21 nov. 2020

Puertos del Servidor

Interfaz	Puerto	Tipo	Descripción
todas	5222	Cliente-Servidor	El puerto estandar utilizado por clientes para conectarse al servidor. En este puerto se establecen conexiones de texto plano las cuales derivan de las conexiones de...

192.168.122.55:9090/system-cache.jsp

3. OpenLDAP

La implementación de OpenLDAP sería instalando el servidor de OpenLDAP y la conexión la daríamos en la opción de el modo de almacenamiento de usuarios que en vez de darle a por defecto le daremos a servidor LDAP. [Mirar instalación de LDAP en trabajo PostgreSQL.](#)

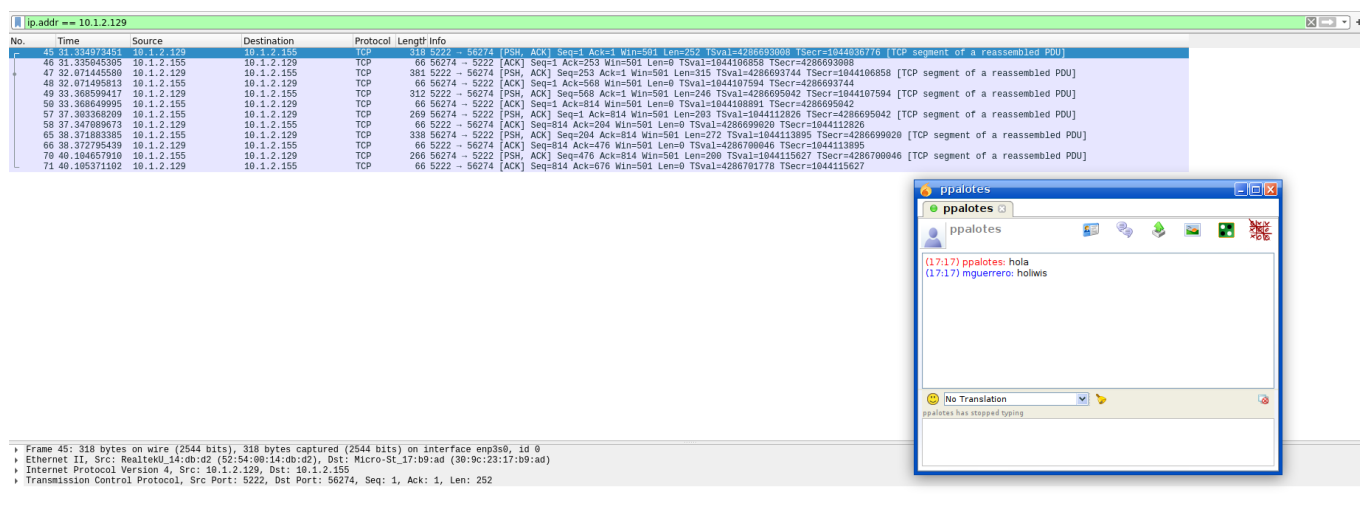
Al tener una configuración ya hecha de Openfire, no podremos directamente implementar la autenticación con OpenLDAP para ello nos iremos al fichero de **'sudo nano**

/opt/openfire/conf/openfire.xml'. Dentro del fichero buscamos la etiqueta **<setup>** y cambiamos el true por false, así podemos reconfigurar Openfire y poder implementar OpenLDAP.

Una vez hayamos hecho esto, el usuario admin deja de existir y debemos entrar con el primer usuario creado en nuestro grupo de OpenLDAP.

4. Captura de tráfico con Wireshark

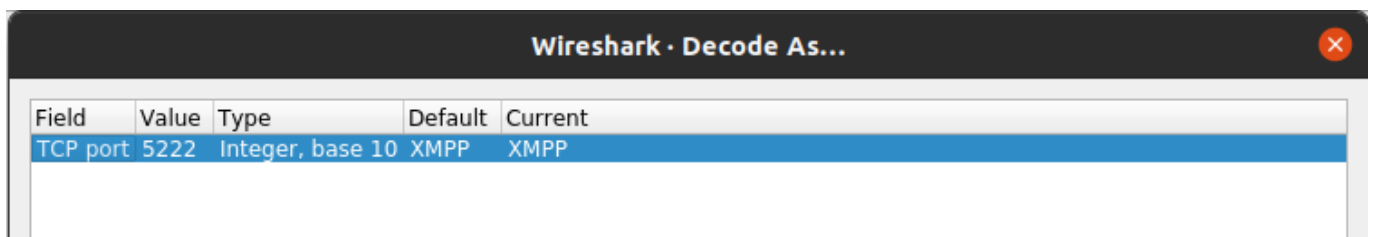
Concluiremos este trabajo con una captura del tráfico gracias a la herramienta de **Wireshark** durante el momento de enviar mensajes entre dos usuarios en el servidor.:



Como podemos comprobar los usuarios se envían un mensaje cada uno y, si aindex.jsp aplicamos un filtro como por ejemplo la dirección IP de uno de los usuarios participantes, se genera un tráfico de protocolo TCP entre los dos.

10.1.2.155 corresponde a 'mgurrero' y 10.1.2.129 corresponde a 'ppalotes'. Cuando ppalotes envía un mensaje a mgurrero, este se envía mediante paquetes con protocolo **TCP** al usuario index.jsp destino del puerto 5222 al puerto 56274. Cuando el usuario destino mgurrero recibe el paquete, este devuelve

una trama de confirmación **ACK** también por protocolo **TCP** al usuario de vuelta por donde ha venido confirmando que el mensaje ha llegado a su destino. Si el mensaje fuera en sentido contrario sería exactamente igual, pero la dirección de los puertos sería opuesta.



Si hacemos click derecho y ‘*Decode as...*’ en el protocolo **TCP** nos saltará una ventana la cual nos dice la disección del protocolo que hemos usado y, como podemos ver, esta disección index.jspado en protocolo **XMPP**, el cual es el que usa este servicio de mensajería instantánea. Cada protocolo tiene su propia disección y ese conjunto forma el protocolo. En nuestro caso solo tiene uno a través del puerto 5222.