

**Universidad Tecnológica de Pereira**  
**Ingeniería de Sistemas y Computación**  
**Sistemas Distribuidos**

**PROYECTO FINAL SISTEMAS DISTRIBUIDOS**

**Por favor lea completamente el archivo antes de formular alguna pregunta.**

**SERVIDOR DE MENSAJERIA INSTANTANEA**

El proyecto consiste en crear un servidor y un cliente (debe ser instanciado varias veces en máquinas diferentes, mínimo 3) de un programa de mensajería instantánea (CHAT).

Una persona se conecta al servidor utilizando el programa cliente. Cuando se conecta por primera vez al servidor debe crear una cuenta. El servidor almacenará los datos de dicha cuenta localmente y persistente (archivo o base de datos). La información requerida será: Nombres, apellidos, login, password, edad y género. Una vez registrada, una persona debe poderse conectar utilizando su usuario y contraseña.

El servidor cuenta con diferentes salas de conversaciones. Estas salas son creadas por los clientes. Cuando se inicia el servidor, este cuenta con una sola sala que es la sala por defecto (las salas no serán persistentes, es decir, si un usuario crea una sala y luego se desconecta la sala será borrada). Cuando un cliente se conecta queda ubicado en dicha sala. Todos los mensajes que envíe un cliente se le reenvían a todos los clientes que se encuentren en dicha sala, al mensaje se le antepondrá el nombre del usuario que origina el mensaje.

Los clientes pueden enviar mensajes en el servidor. La forma de diferenciarlos es que los comandos iniciaran con el carácter '#'. Los comandos disponibles son:

#cR <nombreSala> . Crear sala con el nombreSala. El servidor de forma automática ingresa a este cliente a la sala que creo.

#gR <nombreSala> Entrar a la sala nombreSala.

#eR. Salir de la sala en que se encuentra. El servidor enviara al cliente a la sala por defecto. Si el cliente ingresa este comando estando en la sala por defecto, no tendrá ningún efecto.

#exit. Desconectará al cliente del servidor

#lR Lista los nombres de todas las sala disponibles y el número de participantes de cada una.

#dR <nombreSala>. Elimina la sala nombreSala. Un cliente solo puede eliminar las salas que creo.

#show users: Muestra el listado de todos los usuarios en todo el sistema

#\private<nombreusuario>: Envía un mensaje privado a un usuario determinado sin importar en qué sala se encuentre

**Requerimientos Básicos:**

El servidor y los clientes deben estar en máquinas diferentes (mínimo 3).

El proyecto debe ser desarrollado con hilos y sockets

Cuando un cliente se conecta con el servidor éste debe ser sincronizado con la hora del servidor.

El servidor tiene que ser implementado en lenguaje Python, los clientes pueden ser implementados en el lenguaje que desee (igual, sugiero Python).

El sistema operativo es de su elección.

## **Mecanismo de sustentación:**

Para la entrega final si bien pueden trabajar de manera local por grupos, opcionalmente el servidor puede ser subido a la nube(gratuita por supuesto) de tal manera que los clientes puedan hacer conexión con dicho servidor.

Dicho lo anterior les apporto un listado de proveedores con servicios gratuitos en la nube:

<https://www.dbigcloud.com/bigdata/84-listado-de-proveedores-con-servicios-gratuitos-en-la-nube.html>

De los cuales se destacan:

Parse: <https://parseplatform.org/>

Nitrous.IO: <https://www.nitrous.io/>

Openshift: <https://www.openshift.com/>

Appfog: <https://www.appfog.com/>

Heroku: <https://www.heroku.com/>

Cloudno.de: <https://cloudno.de/>

Personalmente recomiendo "Heroku"

Es importante que los vayan revisando para que como equipo escojan alguno e inicien con la instalación y/o configuración de las diferentes tecnologías que van a utilizar en el desarrollo (lenguajes, motor de bases de datos...)

NOTA: otra alternativa es por ejemplo Amazon(AWS).

## **Bonificaciones (OPCIONAL):**

50% si el servidor es subido a la nube

20% de la nota si el proyecto es implementado con modo gráfico.

10% de la nota si el almacenado de los datos de usuarios es implementado con una base de datos (la de Su preferencia).

## **GRUPOS:**

Los mismos de las exposiciones.

## **Fecha de entrega y sustentación:**

jueves semana de finales, 4-6pm