Practica3$_Redes$

Gonzalez Pardo Adrian
Valdez Bernal Maria Fernanda
Valdez Esquivel Melani Betsabee

17 Marzo 2020

1. Introduccion

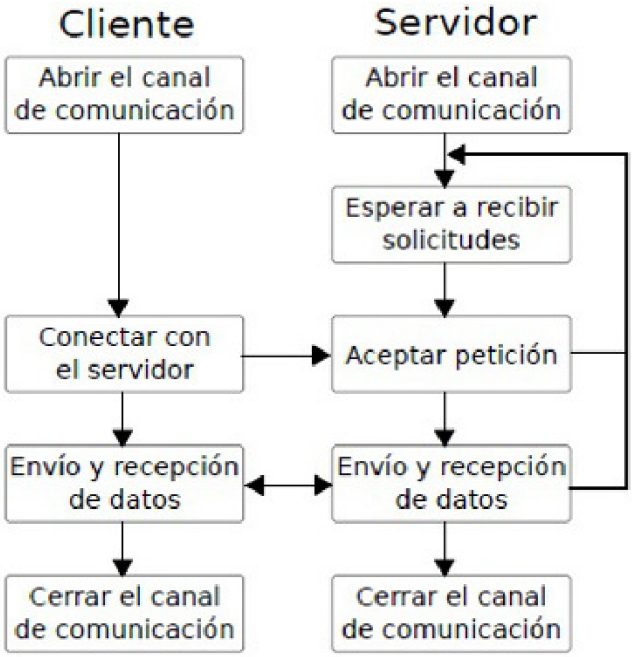
1.1. Sockets

Para que dos programas puedan comunicarse entre sí es necesario que se cumplan ciertos requisitos:
Que un programa sea capaz de localizar al otro.
Que ambos programas sean capaces de intercambiarse cualquier secuencia de octetos, es decir, datos relevantes a su finalidad.
Para ello son necesarios los dos recursos que originan el concepto de socket:
Un par de direcciones del protocolo de red (dirección IP, si se utiliza el protocolo TCP/IP), que identifican la computadora de origen y la remota.
Un par de números de puerto, que identifican a un programa dentro de cada computadora.
Los sockets permiten implementar una arquitectura cliente-servidor. La comunicación debe ser iniciada por uno de los programas que se denomina programa çliente". El segundo programa espera a que otro inicie la comunicación, por este motivo se denomina programa "servidor".
Un socket es un proceso o hilo existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

1.2. Arquitectura Cliente servidor

Arquitectura Cliente servidor. Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes.

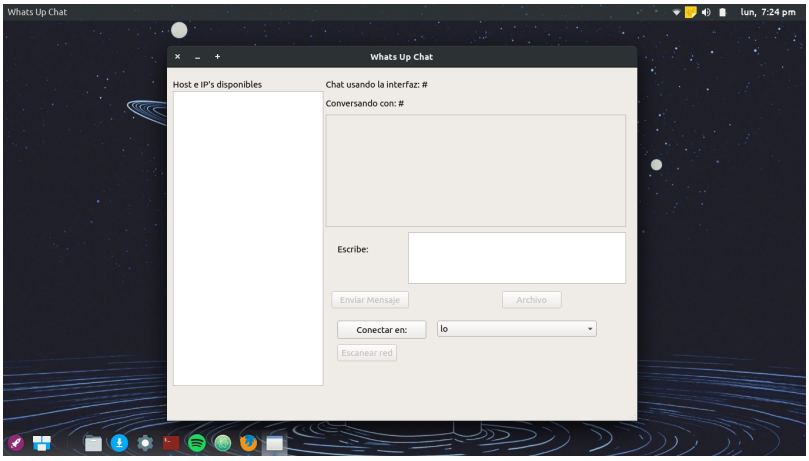
Ayuda a comprender las bases sobre las que están construidos los algoritmos distribuidos.



Ejemplo general de la Arquitectura Cliente Servidor

2. Desarollo

Lo que se procedio a realizar es en lenguaje de desarollo Python, debido a su facilidad para desarrollar interfaces graficas, ya que nosotros debemos construir un chat (una plaaforma de comunicacion estilo Whatss App).
Primero se desarollo lo que es la interfaz grafica (como semuestra en la siguiente figura)



Iterfaz del Chat

Una vez que se definio que es lo que se iba a utilizar (mensajes y archivos y como se debian enviar) se procedio a realizar la estructuradel cliente y del servidor.
Para elservidor se realizo uno de forma no bloqueante para pueda conectarse con tantos usuarios pueda soportar la computadora, aqui manejamos tambien laparte de seralizar datos para poder recibir y enviar los archivos que los clientes se esten compartiendo, los mensajes sin mayor problema se envian con la logica de que el servidor conoce sus IP ya que estamos en la misma red y podemos identificar quien es quien.
Para resolver el problema de los nombres del usuaio, se le pide en el codigo del cliente que mande un "ninckname"para tener un nombre de usuario y no solo como una IP.
En el cliente se toma que se puede enviar mensajes y conectar al servidor, tabien puede mandar archivos y ver con quien platica y platicar con mas de una persona a la vez, es decir puede tener varias ventanas de conversion abiertas a la vez.
Se implementaron hilos para poder tener varios "subprocesos.ªbiertos, entendiendo que los subprocesos son los usuaios y el hilo padre los va gestionando.

2.1. Diagrama de Flujo

3. Conclusiones

3.1. Gonzalez Pardo Adrian

3.2. Valdez Bernal Maria Fernanda

3.3. Valdez Esquivel Melani Betsabee