

1. Procedimientos remotos es un concepto que hace referencia al evento en el que la ejecución de un programa en una computadora ocasiona que un procedimiento se ejecute en otra computadora, de manera que ambos procedimientos forman parte de un solo flujo de ejecución.
2. Una IP es una secuencia numérica que sirve para identificar un dispositivo en una red de Internet, sigue un estándar que establece la semántica y sintaxis de los números en la secuencia.
3. Un puerto es un número que sirve como identificador de procesos o servicios dentro de un sistema operativo, es parte del mecanismo que se lleva a cabo en la capa de transporte para ofrecer los servicios de comunicación entre el hardware y el software de la aplicación.
4. Sí se pueden tener dos clientes, pero también deberían haber uno o dos procesos servidores, que sirvan para escuchar los mensajes de cada cliente.
5. Reporte del código
  - a. El flujo es el siguiente: inicia la interfaz de login para autenticar al usuario y las direcciones IP de los servidores local y remoto, posteriormente se inicia la ventana de chat y los hilos correspondientes al cliente y al servidor, el servidor local recibe mensajes de los dos clientes, con la diferencia de que el cliente remoto hace que los guarde en un buffer y el cliente local obtiene los mensajes de ese buffer.
  - b. En el directorio GUI se localizan los archivos \*.ui, que contiene la especificación gráfica de los elementos que se dibujan en la interfaz del usuario, además de los archivos: Login.py, que se encarga de tomar usuario, ip propia e ip destino, realizar validaciones e iniciar la interfaz de chat y el servidor que escucha los mensajes; Chat.py, incluye una clase con métodos que dibujan y recuperan mensajes en cuadros de texto dentro de la ventana de chat.
  - c. En el directorio Channel se encuentran tres archivos: Channel.py, que contiene la clase encargada de iniciar el thread de escucha y de enviar mensajes al servidor remoto; ApiClient.py, que especifica una clase que sirve para definir la manera en que se envían los mensajes al servidor remoto; ApiServer.py, que define la interfaz con la que puede interactuar el cliente remoto para comunicarse con el servidor local.
  - d. Principales problemas:
    - i. Entender cómo se debería organizar el código para que cumpliera la estructura deseada. La solución, que después se convirtió en un problema, fue desarrollar inicialmente la aplicación de manera que entendiéramos el flujo de ejecución sin considerar el diseño estructural.
    - ii. Lograr que el programa funcione con la estructura especificada, ya que al desarrollar inicialmente la aplicación sin estructura de directorios y bajo un esquema de organización diferente, no hubo problemas de ejecución, sino hasta que se trató de realizar la implementación del programa funcionando bajo el esquema requerido para la práctica.
  - e. Problemas no solucionados:

- i. Lograr que el programa funcionase con los cambios que se realizaron a la implementación inicial para que se adaptara a la estructura de directorios especificada.