

Enviar un mensaje UDP a un ordenador desde Python

Código del servidor (ordenador que escucha):

[crayon-6a9d6059e9ac2280487761/]

Código del cliente (ordenador que manda el mensaje):

[crayon-6a9d6059e9ac9157133116/]

Explicación línea por línea del código del servidor y del cliente:

Código del servidor (ordenador que escucha):

1. **Importar el módulo socket:** Permite crear y manipular sockets de red en Python, que son puntos finales para la comunicación.
2. **Crear el socket UDP:** Se instancia un socket configurado para usar IPv4 (AF_INET) y el protocolo UDP (SOCK_DGRAM), que es un protocolo sin conexión.
3. **Asignar la IP y el puerto al socket:** El socket se enlaza a la dirección 127.0.0.1 (localhost) y al puerto 2020, especificando dónde escuchar los mensajes entrantes.
4. **Imprimir mensaje de espera:** Se muestra un mensaje indicando que el servidor está listo y esperando recibir un mensaje.
5. **Recibir el mensaje:** El servidor espera hasta recibir un mensaje, que se almacena junto con la dirección del

remitente. La función `recvfrom(1024)` permite recibir hasta 1024 bytes.

6. **Mostrar el mensaje recibido:** El mensaje recibido se decodifica desde ASCII y se imprime.
7. **Cerrar el socket:** Finalmente, se cierra el socket para liberar los recursos asociados.

Código del cliente (ordenador que manda el mensaje):

1. **Importar el módulo socket:** Al igual que en el servidor, se utiliza para gestionar la comunicación en red.
2. **Crear el socket UDP:** Se crea un socket similar al del servidor, configurado para IPv4 y UDP.
3. **Preparar el mensaje a enviar:** El mensaje '0' se convierte en bytes usando codificación ASCII, ya que los sockets envían datos en formato binario.
4. **Enviar el mensaje al servidor:** El mensaje se envía al servidor especificando la dirección IP 127.0.0.1 y el puerto 2020.
5. **Cerrar el socket:** Se cierra el socket, liberando los recursos una vez enviado el mensaje.