PSP

SOCKETS

TIPOS

* STREAM (TCP)
  + Cliente
    - Métodos:
      * socket()
      * connect() -> Parámetros (HOSTNAME, PUERTO).
      * send() -> Para enviar datos al servidor.
      * sendAll()
      * recieve() -> Para recibir datos del servidor.
      * recv()
      * close() -> Cierra el socket.
  + Servidor
    - Métodos:
      * socket()
      * bind() -> Parámetros (HOSTNAME, PUERTO). Vincula el socket a una dirección.
      * listen()
      * accept() -> Devuelve una tupla (CLIENTE, DIRECCION)
        + Cliente -> Se crea un nuevo socket que representa la conexión con el cliente.
        + Dirección -> Contiene la IP y el PUERTO del socket cliente.
      * recieve() -> Para recibir los datos que envía el cliente.
      * recv()
      * send() -> Para enviar datos al cliente.
      * sendAll()
      * close() -> Cierra el socket.
  + 3-Way-Handshake
    - 1 – Synchronize
      * Cliente envía Falag SYN activado.
        + Contiene ISN (Initial Sequence Number)
    - 2 – SYN-ACK
      * Servidor recibe Flag SYN
      * Servidor responde Frag SYN activado y Flag ACK activado.
      * Servidor elige ISN del servidor.
    - 3 – ACK
      * Cliente recibo SYN-ACK
      * Cliente envía Flag ACK activado.
      * Puede enviar datos con este ACK.
  + socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)
* DGRAM
  + Cliente
    - Métodos:
      * socket()
      * sendto()
        + El cliente debe saber los datos del servidor (HOSTNAME, IP). Es siempre el cliente quien empieza la conversación.
      * recvfrom()
        + Para recibir los datos que nos envía el servidor. Se guarda una tupla (mensaje, addr del servidor)
      * close() -> Cierra el socket.
  + Servidor
    - Métodos:
      * socket()
      * bind() -> Parámetros (HOSTNAME, PUERTO). El socket se vincula a una dirección.
      * recvfrom()
        + Cuando un cliente intente una conexión devolverá una tupla (mensaje, addr)

Mensaje -> Datos que envía el cliente.

Addr -> IP del cliente.

* + - * sendto() -> Envía datos a la dirección del cliente.
      * close()
  + socket.socket(socket.AF\_INET, socket.DGRAM)