



3º Grado en Ingeniería Informática

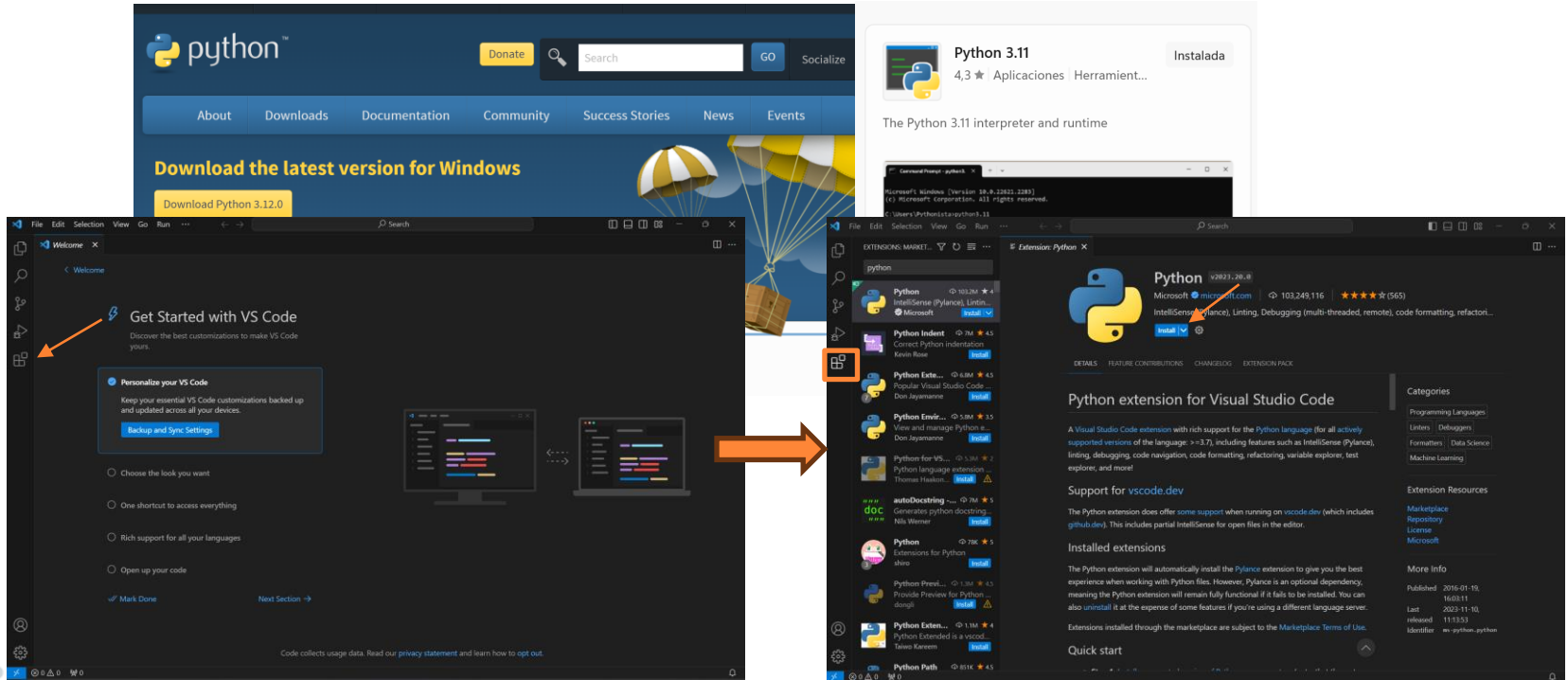
Fundamentos de Redes

Seminario 4: Creación de aplicaciones Cliente/Servidor

Julia Caleyá Sánchez – jcaleyas@ugr.es

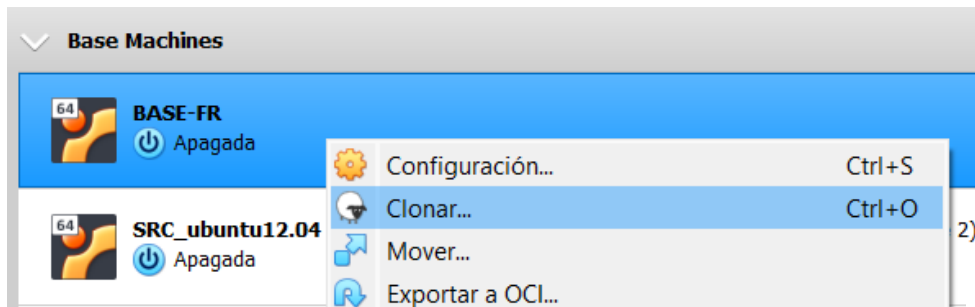


Instalación Python y Visual Studio Code



Realizar dos clonaciones enlazadas

- 🕒 Realizad 1 clonación enlazadas de la máquina virtual de seminarios (FR-cliente/servidor).



Nuevo nombre de máquina y ruta

Seleccione un nombre y opcionalmente una carpeta para la nueva máquina virtual. La nueva máquina será un clon de la máquina **BASE-FR**.

Nombre:

Ruta:

Política de dirección MAC:

Opciones adicionales: ☐ Mantener nombres de disco

☐ Mantener UUIDs hardware

Tipo de clonación

Seleccione el tipo de clonación que desea crear.

Si selecciona **Clonación completa**, una copia exacta (incluyendo todos los archivos de disco duro virtual) de la máquina original serán creados.

Si selecciona **Clonación enlazada**, una nueva máquina será creada, pero los archivos de las unidades de disco duro virtuales serán vinculados a los archivos de disco duro virtual de la máquina original y no podrá mover la nueva máquina virtual a una computadora diferente sin mover los originales también.

Si crea una **Clonación enlazada** entonces una nueva instantánea será creada en la máquina virtual original como parte del proceso de clonación.

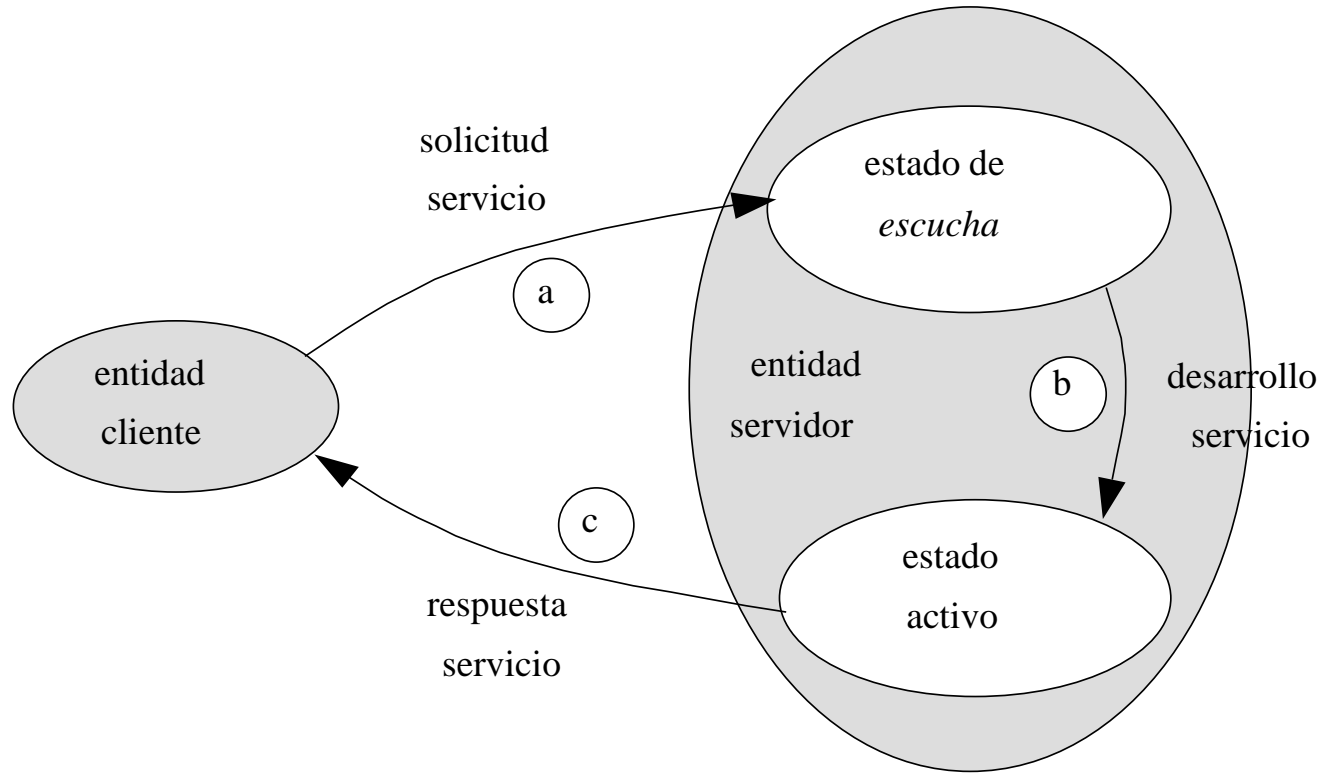
☐ Clonación completa

☒ Clonación enlazada

Clonar

Cancelar

Paradigma cliente/servidor



Paradigma cliente/servidor

Tipo de conexiones

- Orientada a conexión

TCP

- No orientada a conexión

UDP

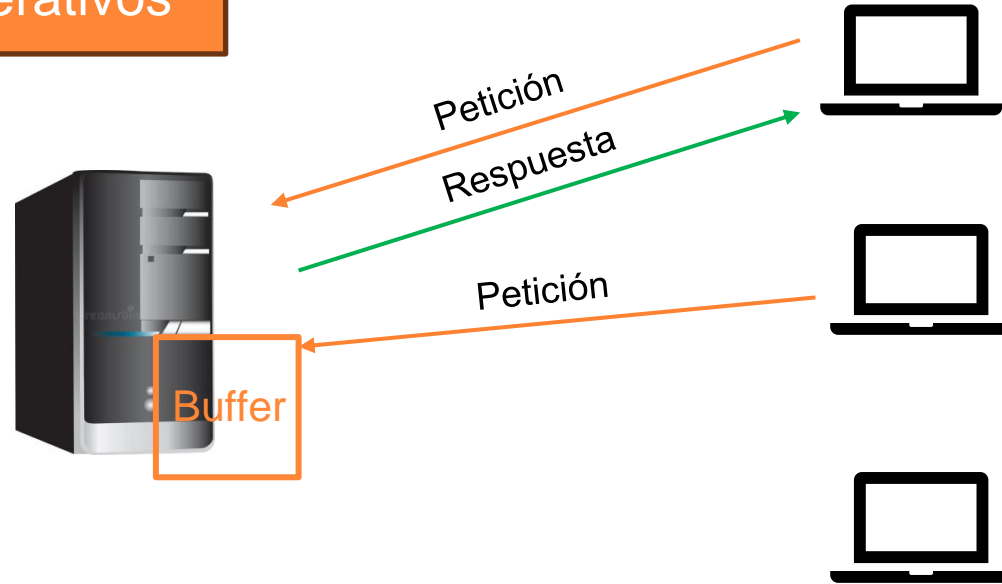
Tipo de servidores

- Servidores iterativos

- Servidores concurrentes

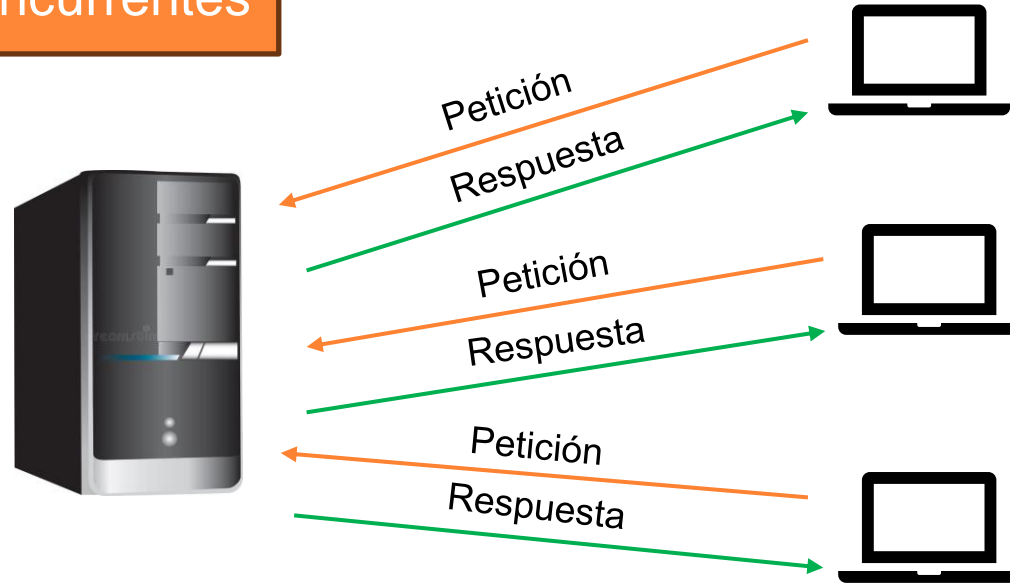
Tipos de servidores

Servidores iterativos

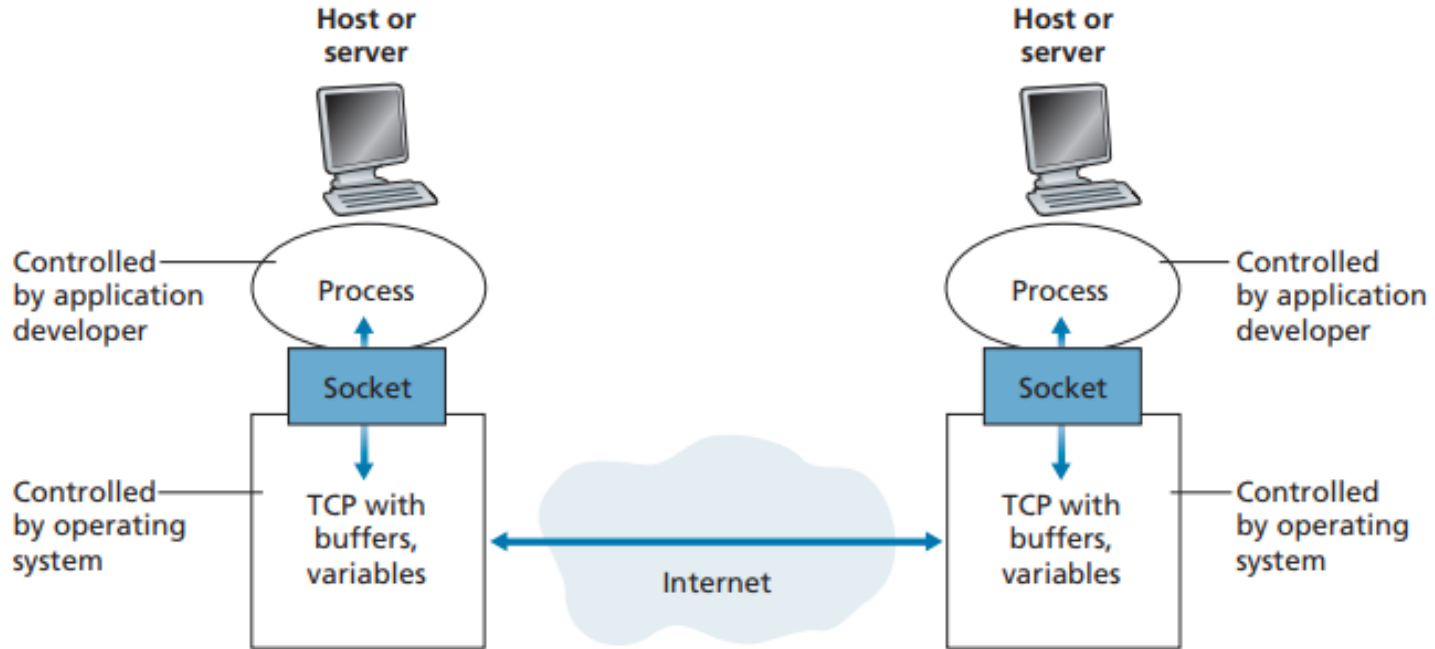


Tipos de servidores

Servidores concurrentes

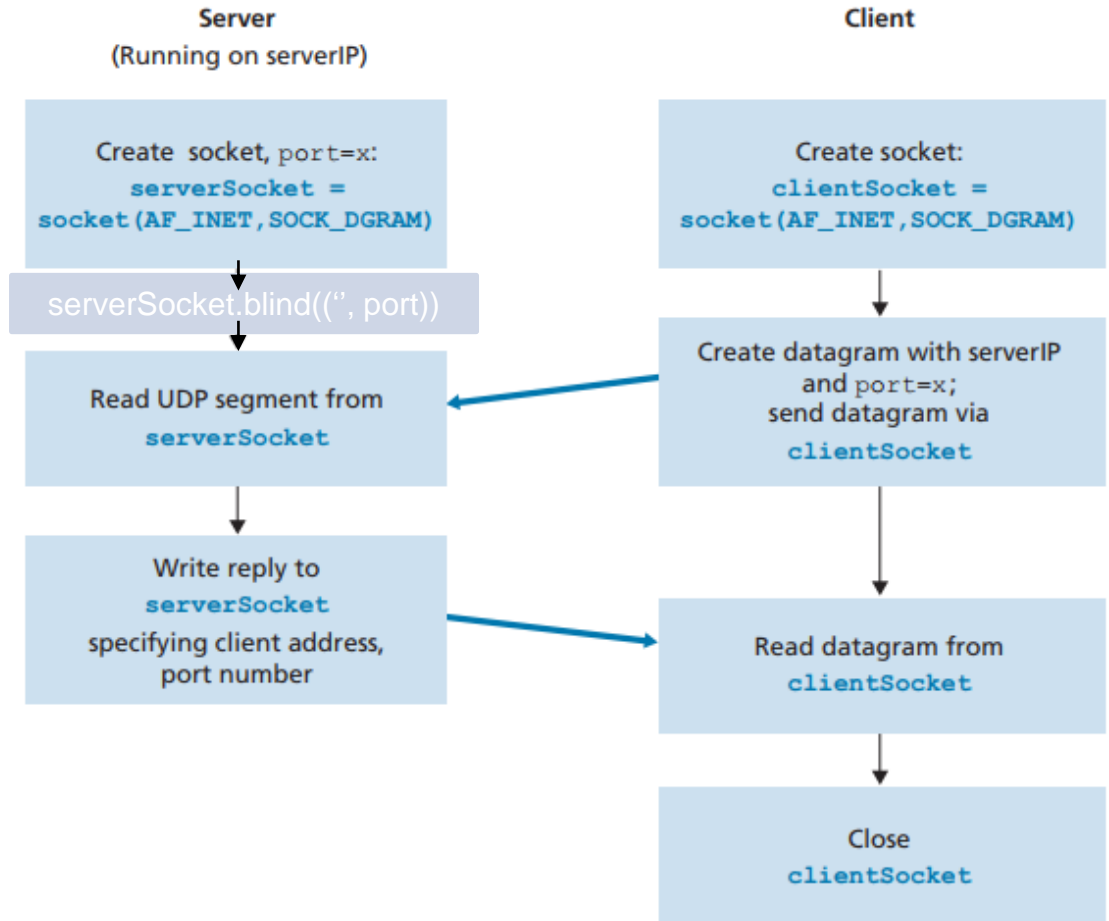


Modelo cliente/servidor: Sockets



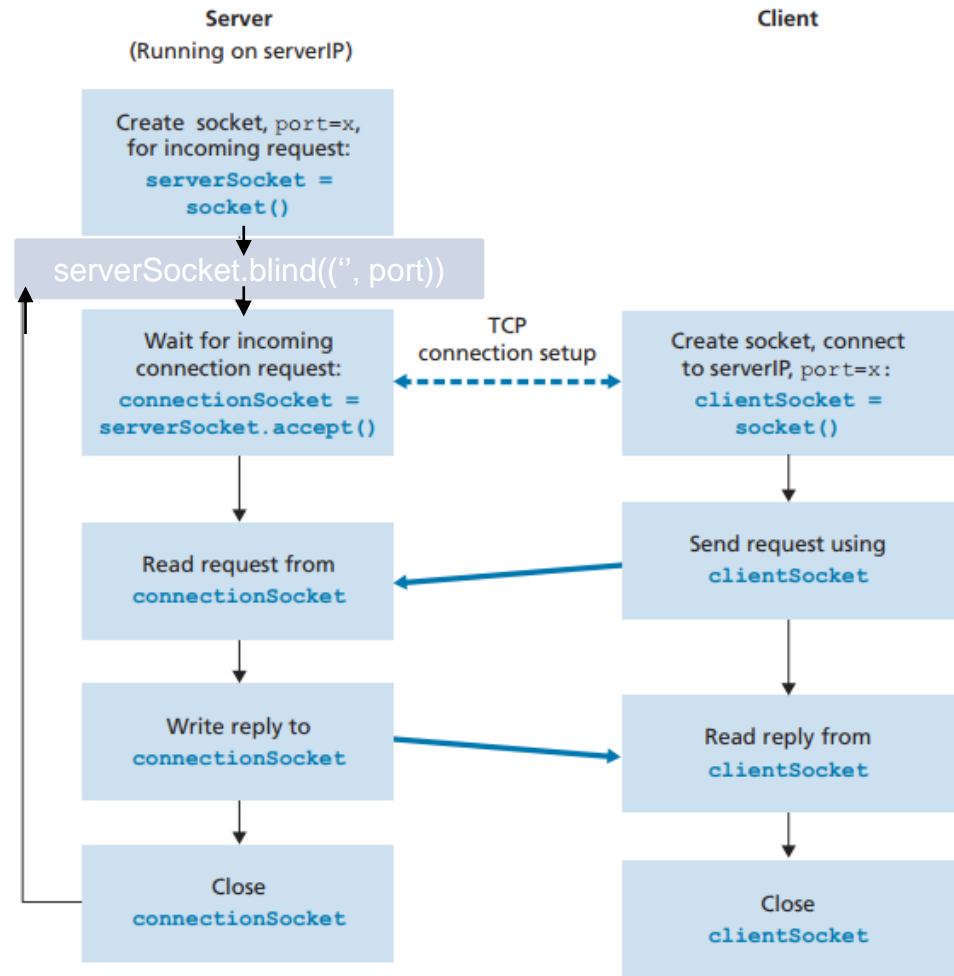
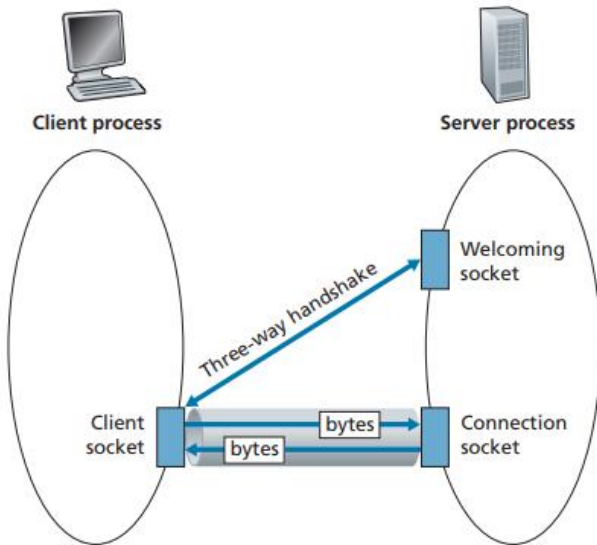
Sockets UDP

- ⦿ No orientado a conexión y no fiable



Socket TCP

- ⦿ Orientado a conexión y fiable (control de congestión)



Retos

- ◎ Rellenar los huecos del fichero *servidorWeb.py* para que funcione como un servidor web y se pueda acceder desde cualquier navegador con la url: `http://localhost:8080/` .
- ◎ Montar un servidor que convierta en minúscula todos los mensajes que recibe del cliente y se reenvíe la conversión al cliente.
- ◎ Montar un cliente/servidor resistente a fallos en la transmisión a nivel de aplicación.
 - El cliente envía un mensaje
 - El servidor al recibirlo lo elimina con un 33% de probabilidad y si es así pide al cliente que lo reenvíe, si no, le dice que se ha recibido correctamente
 - El cliente reenvía el mensaje hasta que el servidor confirma que se ha recibido correctamente
- ◎ **Superextra!** Montar un cliente/servidor en TCP para jugar al ahorcado.