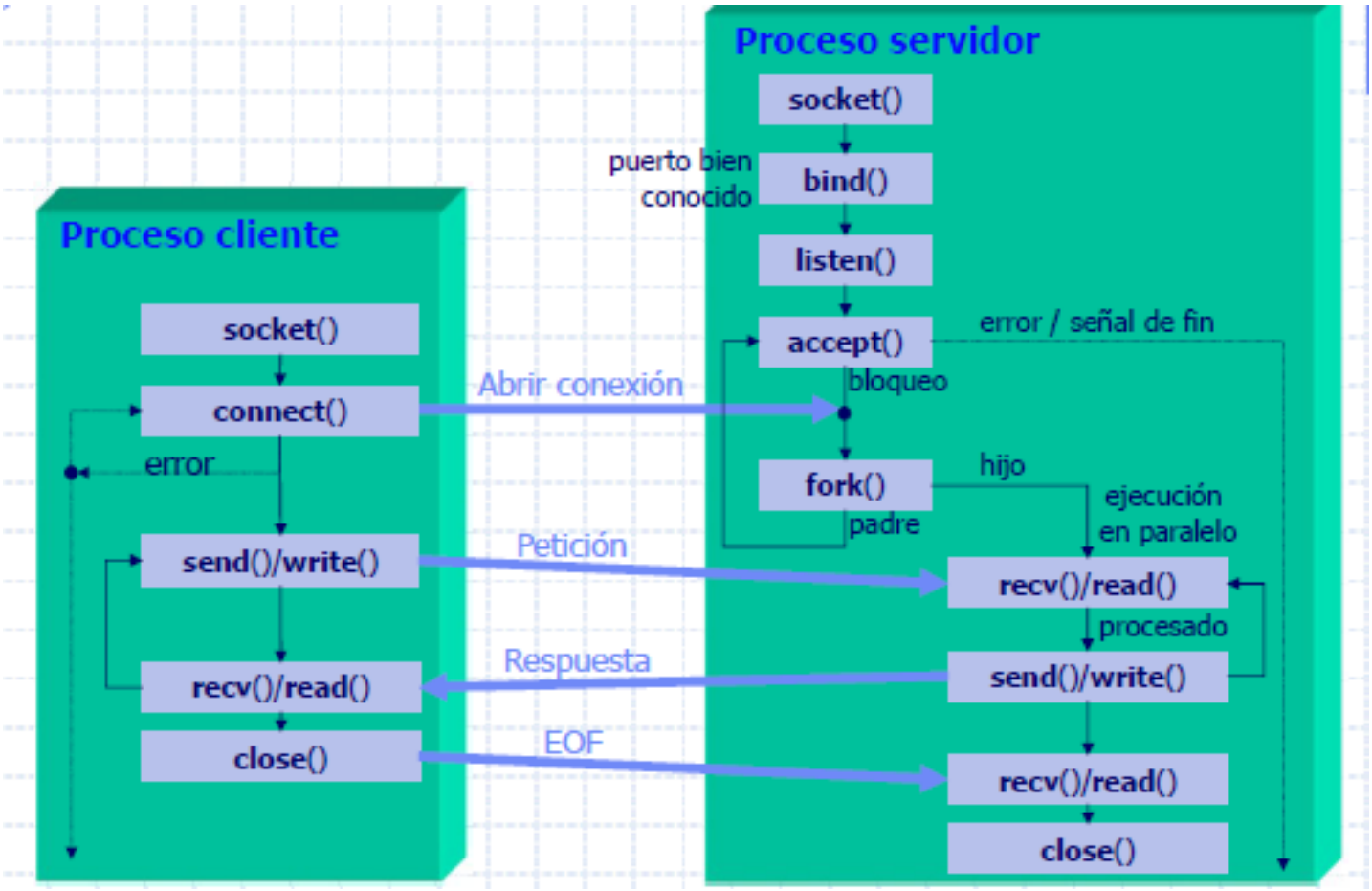
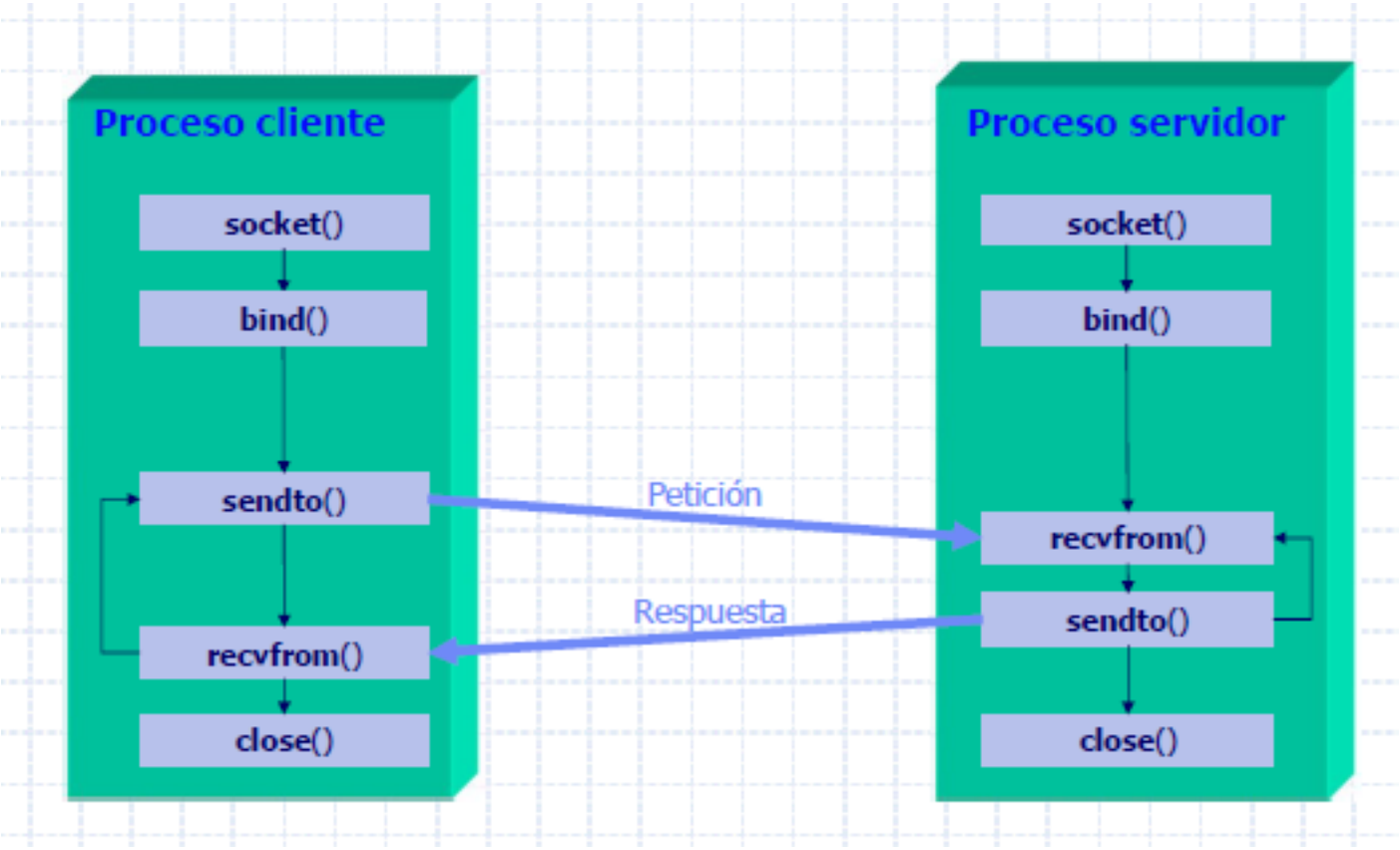
T5.2: Sockets

Socket -> enchufe
Representa el extremo cable
Mi socket está conectado con otro socket que está en otro lugar (en otro procesos)
Sockets son abstracciones que pertenecen a los procesos

pg 8: Escenario de uso de sockets streams



pg 9: Escenario de uso de sockets datagrama



Si hacemos muchas peticiones y respuestas tenemos que ordenarlas (se puede hacer con timestamps). Las aplicaciones más complejas suelen hacerse con streams y las más sencillas con datagramas.

Ambos (streams y dastaagramas) son bidireccionales, a diferencia de las practicas 2 y 3. Es decir, podemos escribir y leer a la vez, ya que tiene dos sentidos, uno para leer y otro para escribir.

pg 11: Dominios de comunicación

- Un dominio representa una familia de protocolos. (De la Capa de transporte)

¿Qué es una familia de protocolos? ¿Cuándo 2 protocolos son familia? Dos protocolos son familia cuando están relacionados y pueden hablar entre sí. Tienen que usar el mismo formato de direcciones. (Esto es lo que realmente hace familia a dos protocolos)

¿TCP y UDP son de la misma familia? Si son de la misma familia porque ambos utilizan direcciones IPv4.

- Un socket está asociado a un dominio desde su creación.
- Cada dominio tiene su propio formato de direcciones

- Los servicios de sockets son independientes del dominio. El segundo parámetro que se le da a un socket es el tipo de servicio que utiliza (los más comunes son stream y datagrama). Sólo se puede comunicar sockets del mismo dominio y de la misma familia.

- Primer dominio: PF_UNIX (para sockets locales, es decir, comunican procesos dentro de la máquina. Son como super tuberías) Practica 4 (rehacer p2 pero con sockets unix)

- Dominio PF_INET utiliza la pila de protoclos TCP/IP (IP + puerto)

¿IPv4 e IPv6 son de la misma familia? No porque no tienen el mismo formato. (Practicas 5,6 y 7)

-Otros dominios: Dominio PF_NS y Dominio PF_APPLETALK

pg 12: Tipos o estilos de sockets

Una vez marcado el dominio y el formato de direcciones, falta el tipo o estilo del socket. Los tipos más populares son stream o datagrama.

El tipo o estilo de un socket recoge el conjunto de propiedades del servicio que se desean.

pg 13: Stream

Representa un circuito virtual u orientado a conexión: al conectar se realiza una búsqueda de un camino libre entre origen y destino y se mantiene el camino en toda la conexión. (Es como si tuviera un cable punto a punto bidireccional, este funciona perfectamente hasta que se cierra la comunicación)

- Propiedades:
  - › Orientado a conexión; da una conexión full-duplex .
  - › Fiable, asegura que no se pierden ni se duplican datos.

› Secuencial, asegura el orden de entrega de los datos.
› No mantiene separación entre mensajes (byte stream).
› Permite el envío de mensajes fuera de banda (out of band). (Mensajes fuera de banda: es un flag. Ese paquete ignora la secuencialidad.)

En el dominio PF_INET se corresponden con el protocolo TCP.
En el dominio PF_UNIX son como una FIFO full-duplex

pg 14: Sockets tipo datargrama

Representa una red basada en datagramas (no orientada a conexión): no se fija un camino; cada paquete podrá ir por cualquier sitio. No existe una reserva de camino. Cada paquete va por dónde quiere.

- Propiedades:
› Sin conexión (posible "pseudoconexión" gracias al uso de direcciones implícitas).
› No fiable, no se asegura el orden en la entrega.
› No se garantiza la recepción secuencial (el orden) de los datos.
› Mantiene la separación entre mensajes.

En el dominio PF_INET se corresponden con el protocolo UDP.

pg 15: Otros tipos de Sockets

Raw: (WireShark)
- Permiten el acceso a los protocolos de niveles más bajos. (IP, Ethernet...)
- Requieren ser root para su utilización.
- Pueden usarse para implementar nuevos protocolos de transporte (RDP).

Otros tipos:
- Sequenced packet sockets
- RDM sockets

pg 16: Direcciones de sockets

Un socket debe tener asignada una dirección única. Las direcciones son dependientes del dominio. Cada socket tiene unicamente una dirección. Usos:
› Asignar una dirección local a un socket (bind()).
› Especificar una dirección remota (connect(), sendto()).

Tenemos una interfaz socket con varias llamadas, ¿cómo puedo pasar direcciones teniendo en cuenta que estas son diferentes y no se parecen en nada? Usando herencia y polimorfismo.

¿Cómo lo hacemos a nivel de llamadas de sistemas, donde no hay herencia ni polimorfismo? Con punteros void y un tamaño. (un buffer)

Formatos de direcciones: cada dominio usa una estructura específica:
Direcciones en PF_INET: dirección IP + puerto
› formato: socket.AF_INET
› ejemplo: ('172.25.3.33', 4567)
Direcciones en PF_UNIX: path de un fichero
› formato: socket.AF_UNIX
› ejemplo: "/tmp/socket"

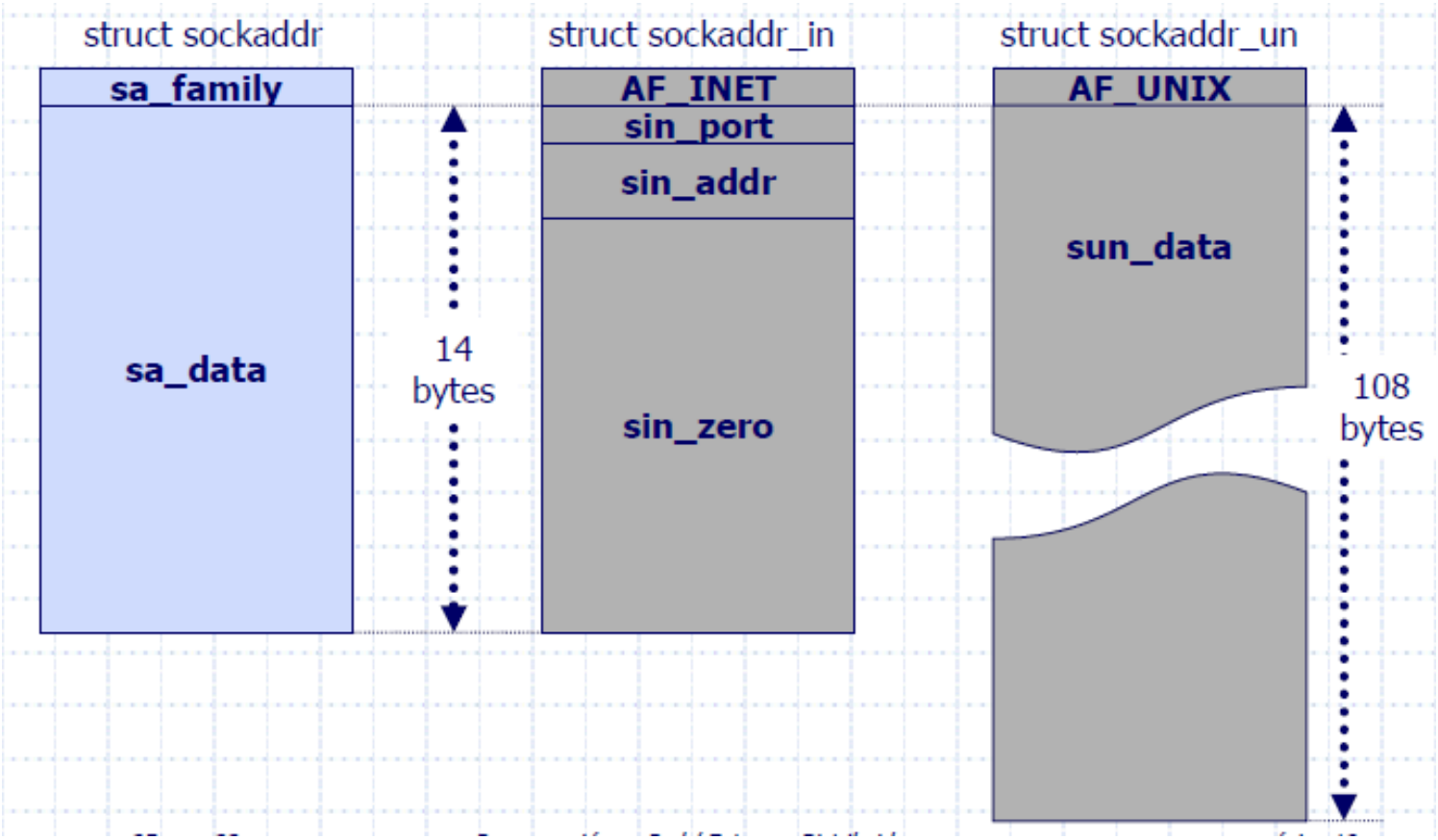
pg 17: Direcciones de Sockets en PF_INET

Una dirección de Internet viene determinada por: una dir ip, puerto de servicio.

Una transmisión se caracteriza por 5 parámetro únicos: (Dir host + puerto) origen, (Dir host + puerto) destino, protocolo de transporte (UDP o TCP)

CONSTANTES QUE HAY QUE SABERSE:
› INADDR_LOOPBACK ('127.0.0.1') – interfaz loopback
› INADDR_ANY ('','0.0.0.0') - cualquier dirección entrante. Cualquier direccion local mía que esté acrivada.
› INADDR_BROADCAST ('<broadcast>') – envío de broadcast
› INADDR_NONE ('255.255.255.255') - indica un error

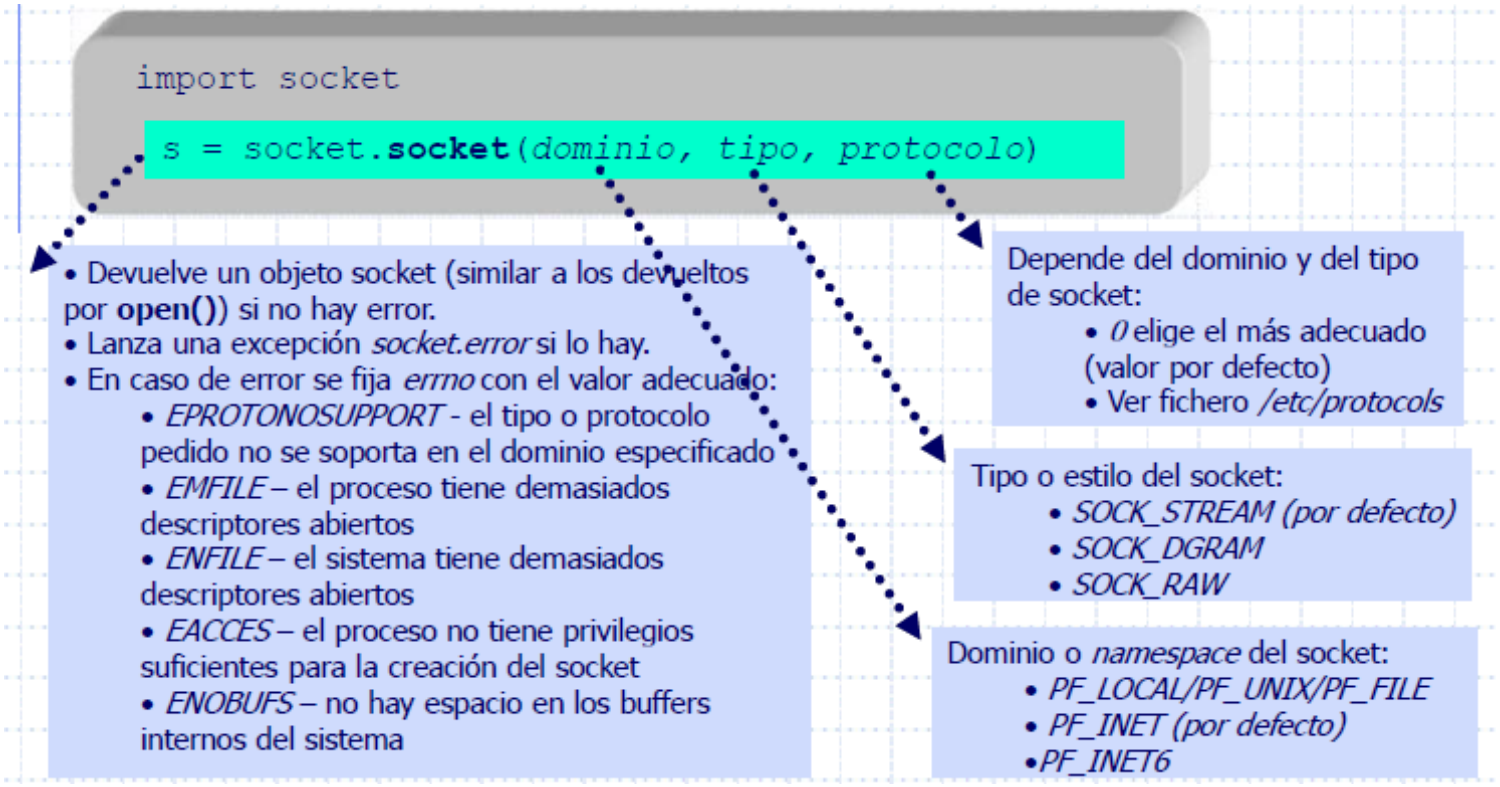
pg 18:



struct sockaddr
struct sockaddr_in: protocolos de internet
struct sockaddr_un: paths
¿pq se pone el campo family al principio? Type tag o algo así

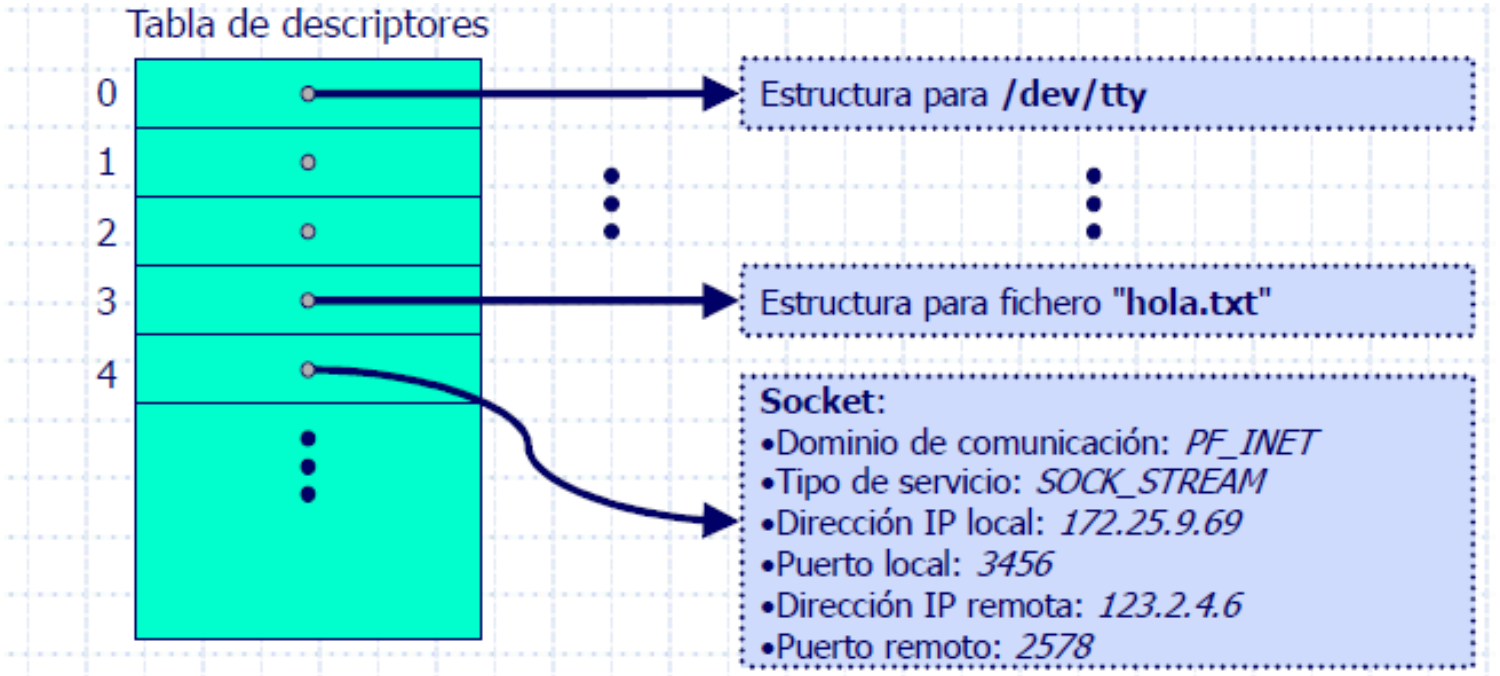
pg 19: Mapa de la API de sockets
Creación:
socket(): crear un socket
bind(): asignar una dirección a un socket

pg 20: Creación de un socket
La llamada socket() crea un socket. El socket creado no tiene dirección asignada. Parámetros obligatorios para crearlo: dominio, tipo y protocolo (Estos parámetros no cambian, una vez que el socket está creado sus valores no cambian).



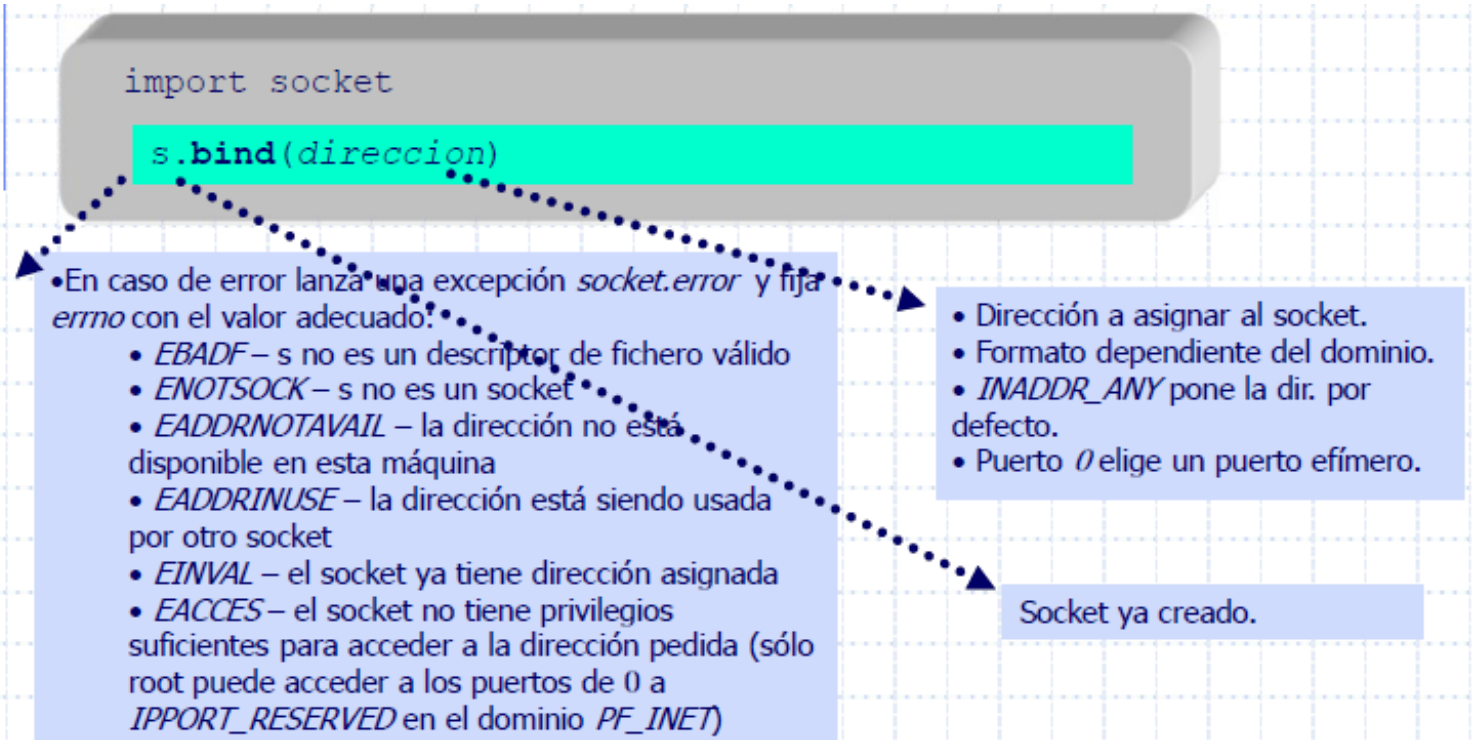
pg 21: Socket como descriptor de fichero

Un socket contiene un descriptor de ficheros. Un objeto socket, una vez creado, contiene un socket (abierto) del sistema operativo. Dicho socket no es otra cosa que un descriptor de fichero de Unix:



pg 22: Asignación de direcciones a un socket ya creado.

Si no se asigna una dirección o puerto, se le dará automáticamente en su primer uso.



pg 23: Asignación de direcciones en PF_INET