



## Curso de Java

“Java – Redes”

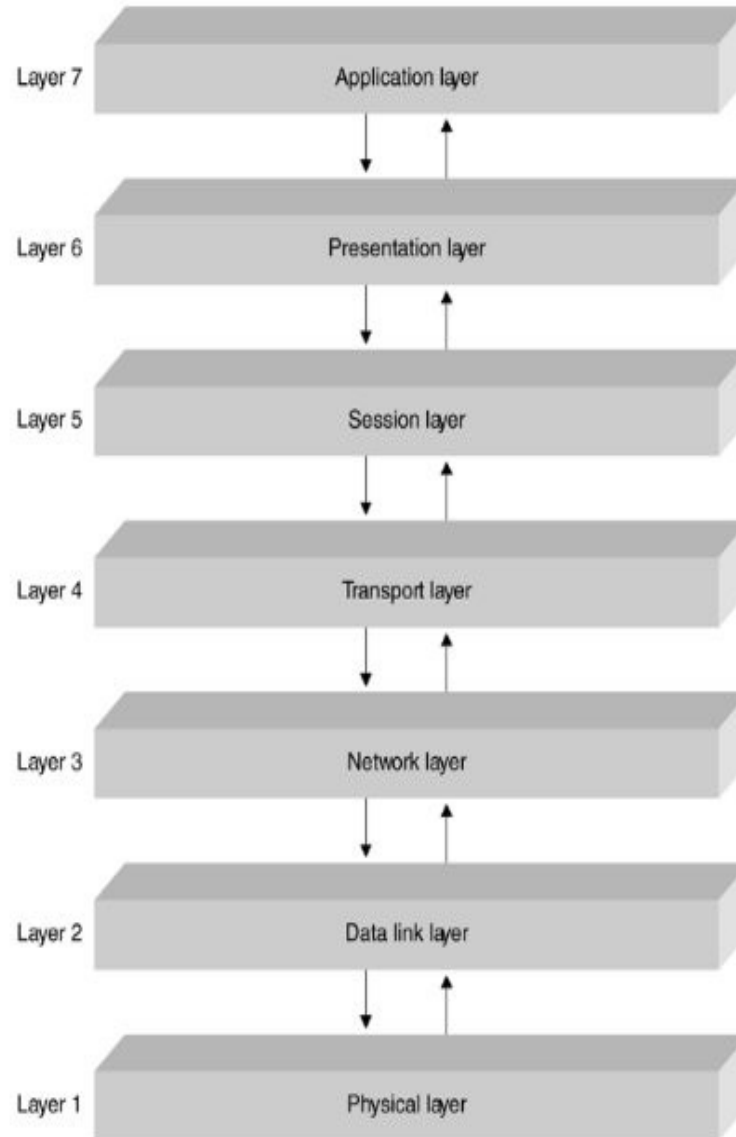
Rogelio Ferreira Escutia



# Conceptos de Redes

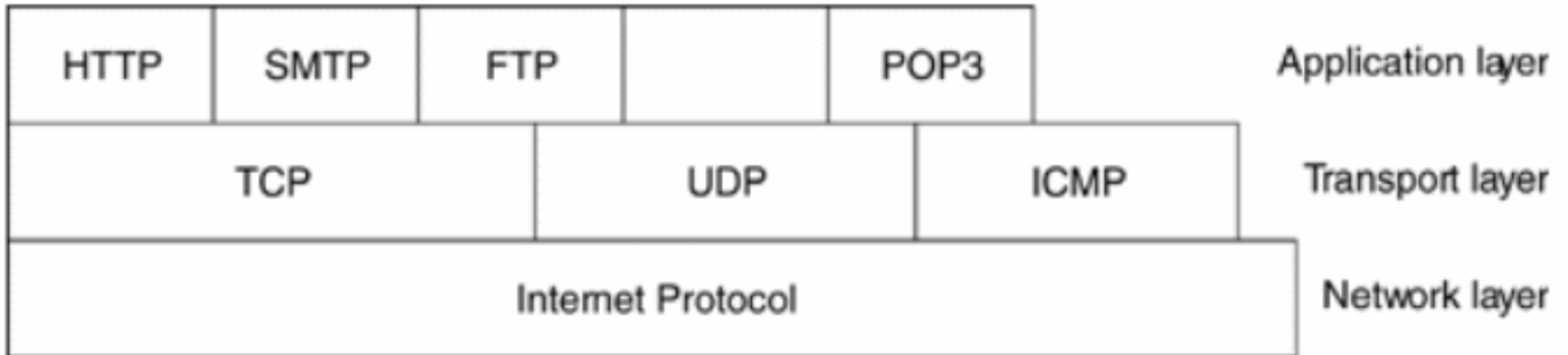


# Modelo OSI



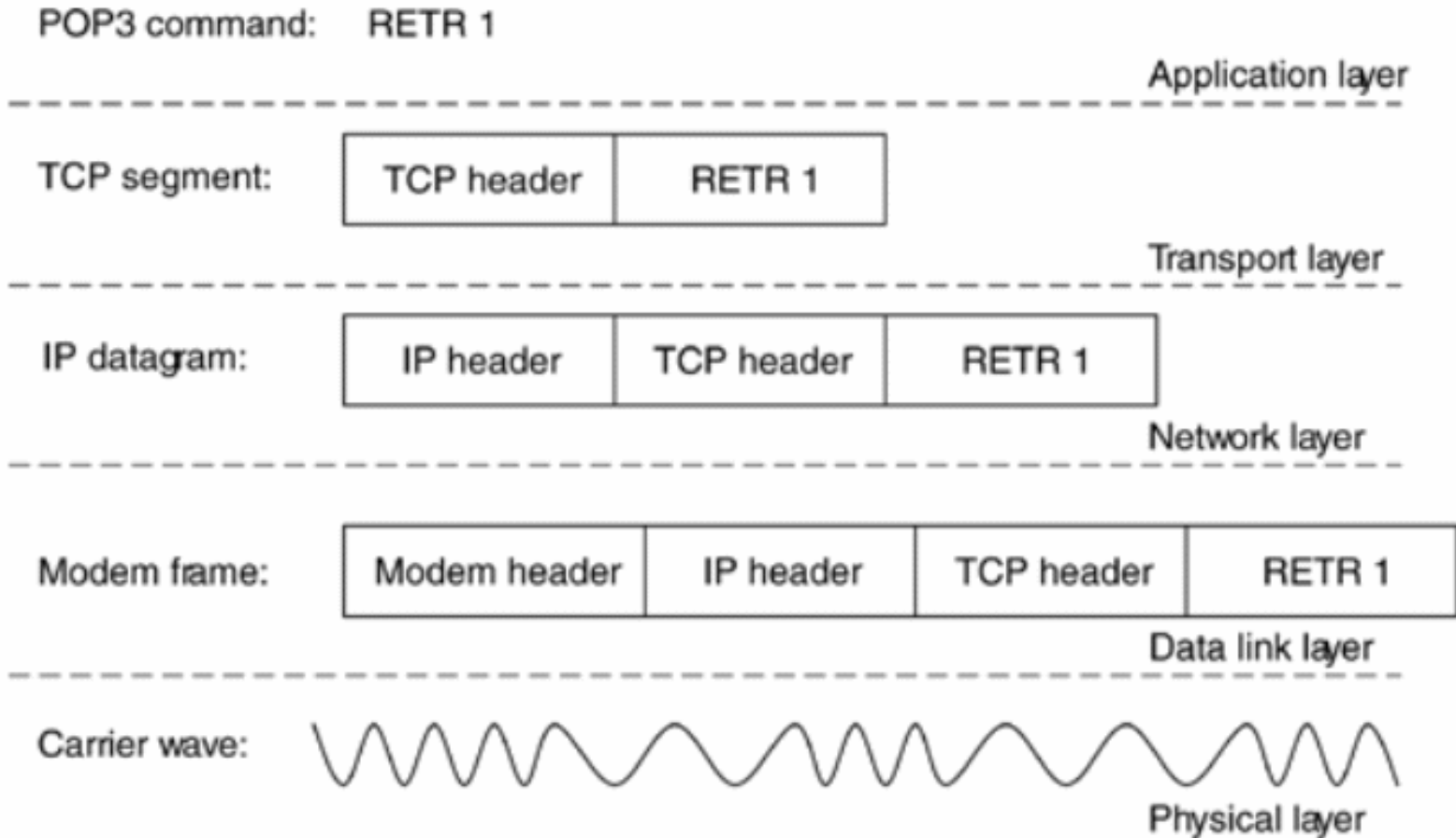


# Capas TCP/IP



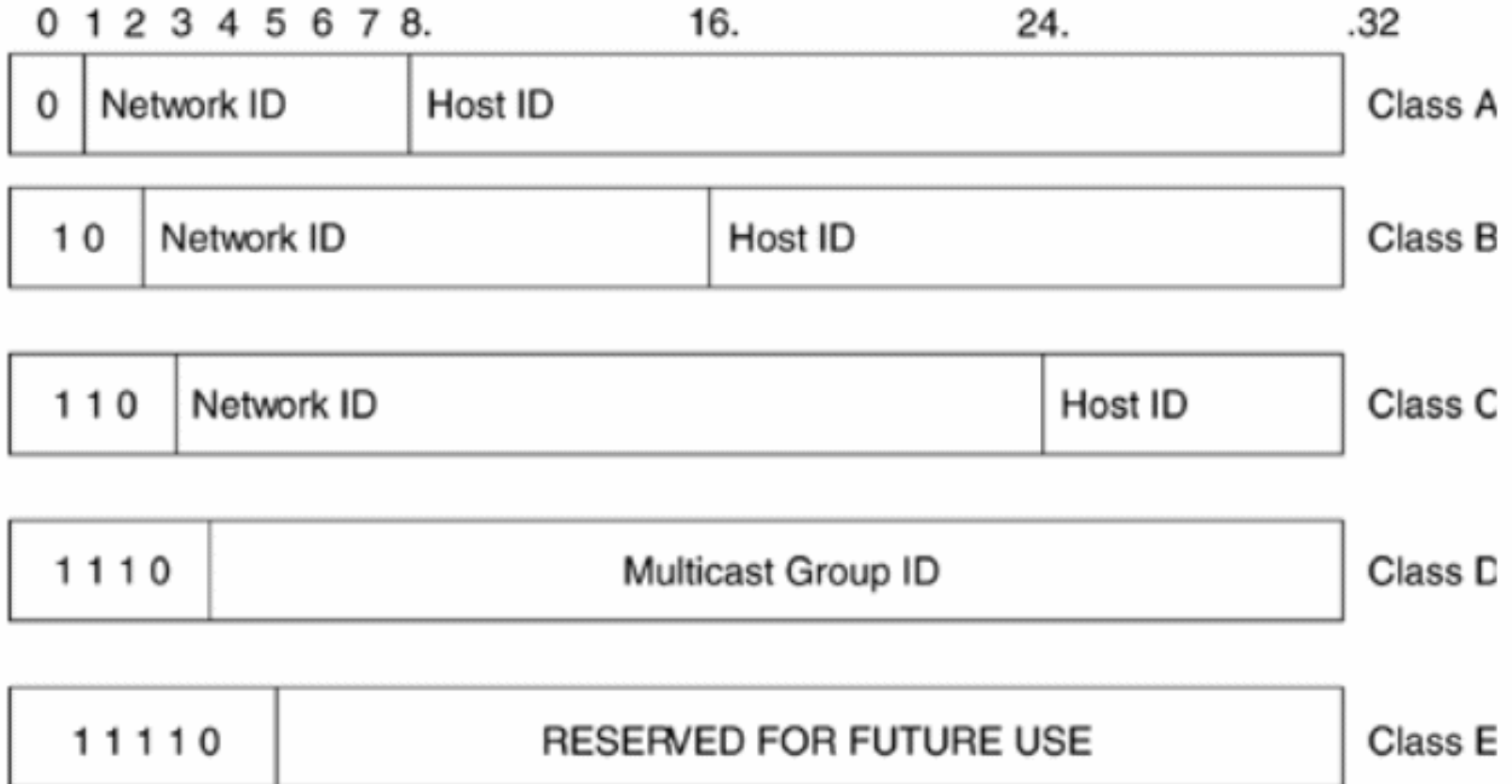


# Encapsulación de Capas



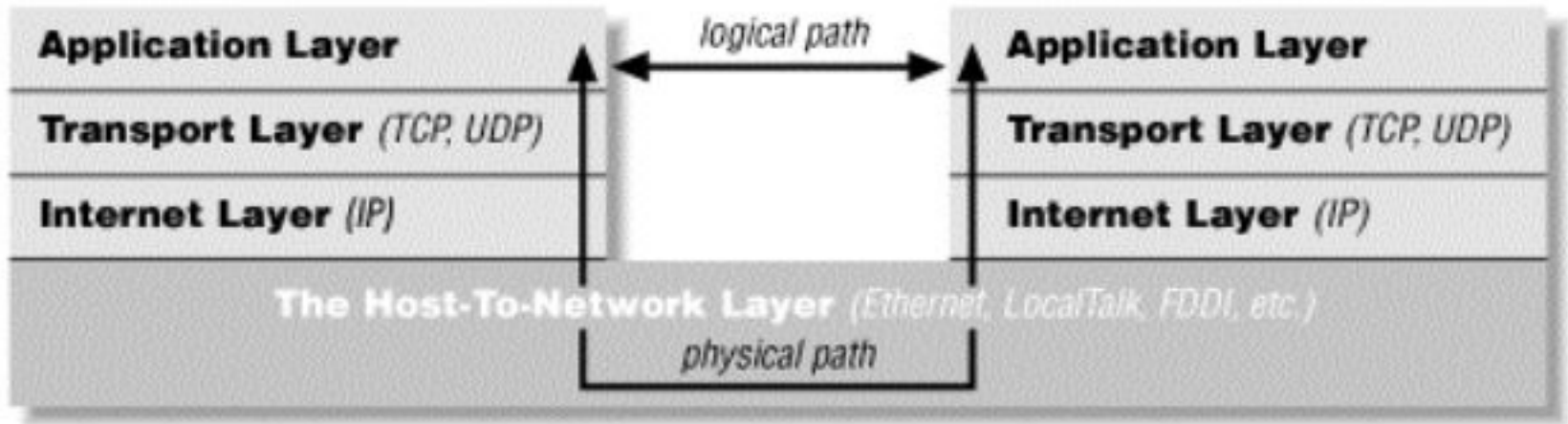


# Clases de Redes



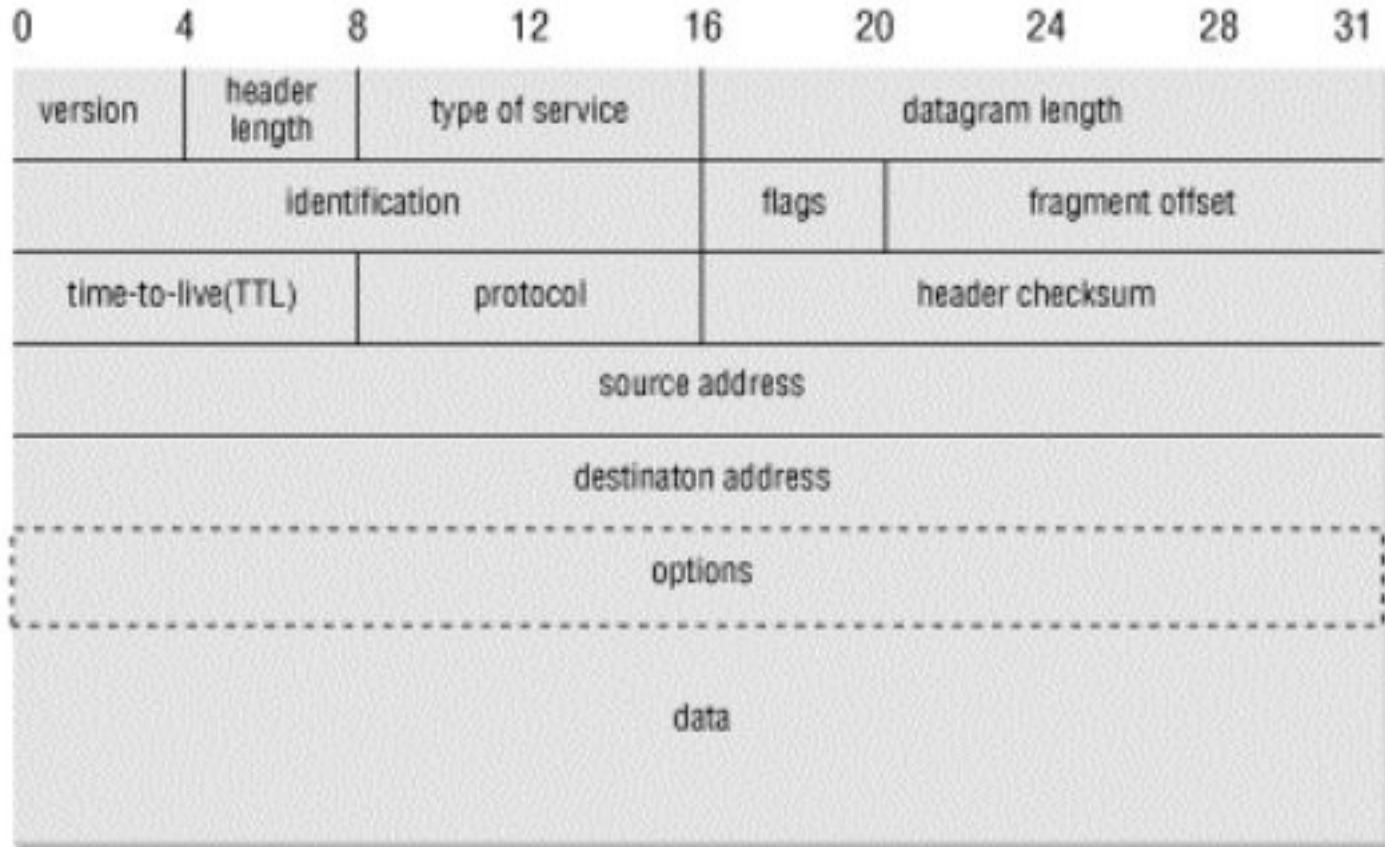


# Capas de Red



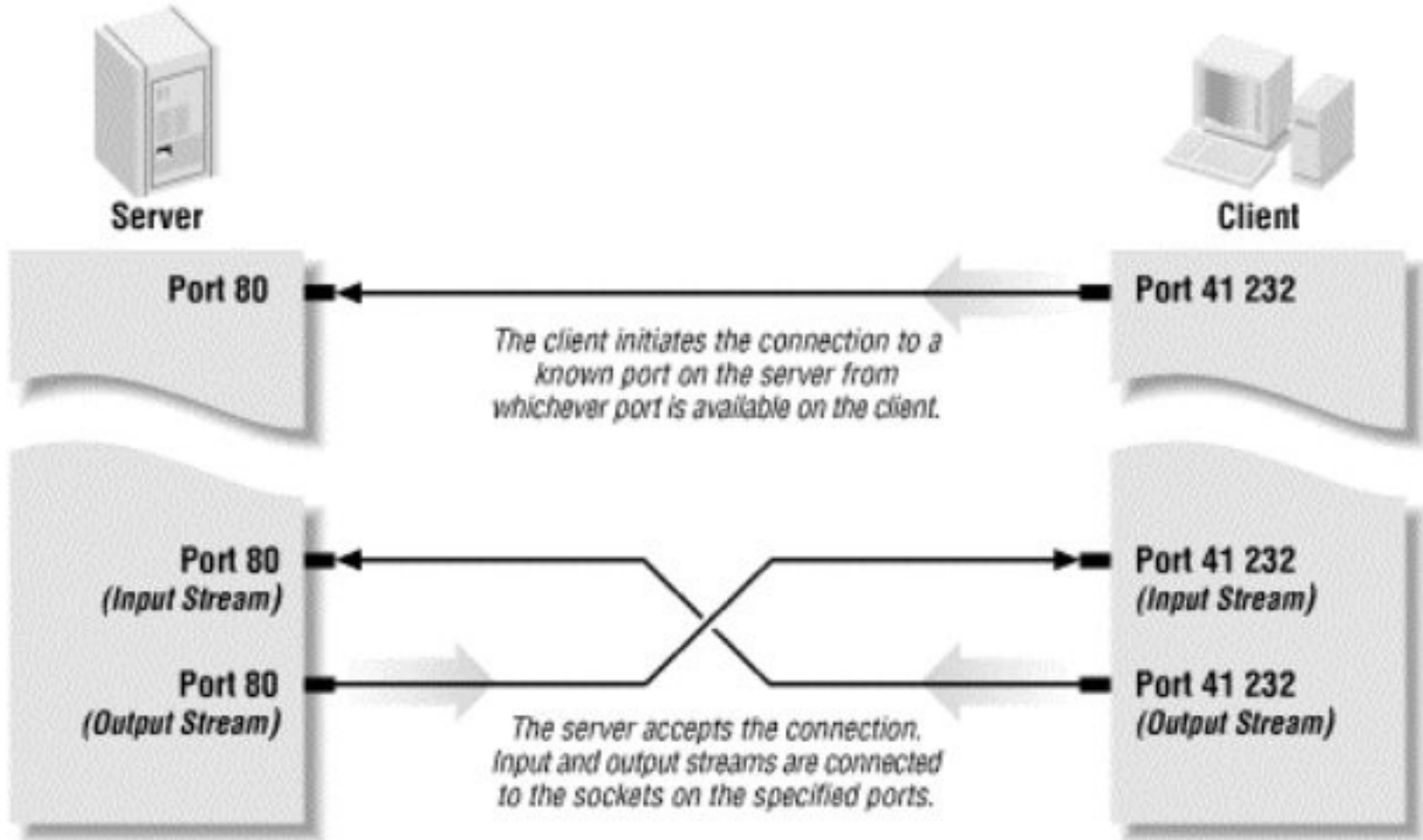


# Datagrama IPV4





# Cliente Servidor



**Entrada / Salida**

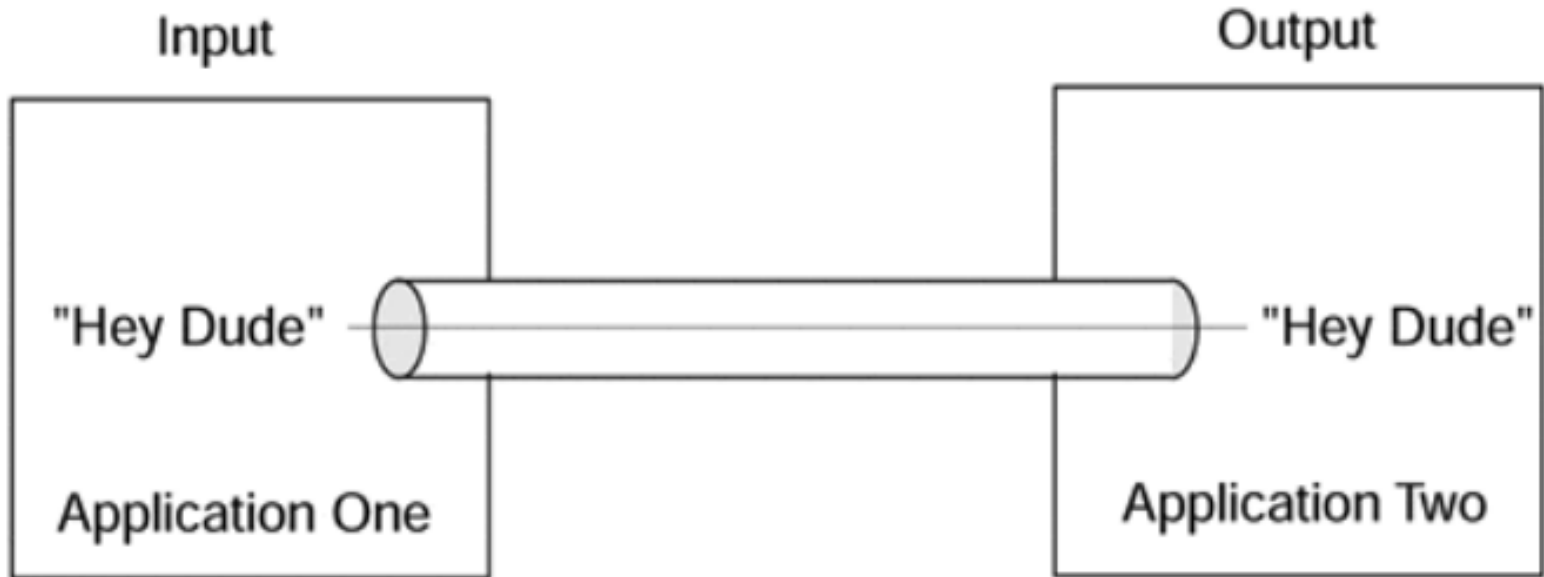


# Stream



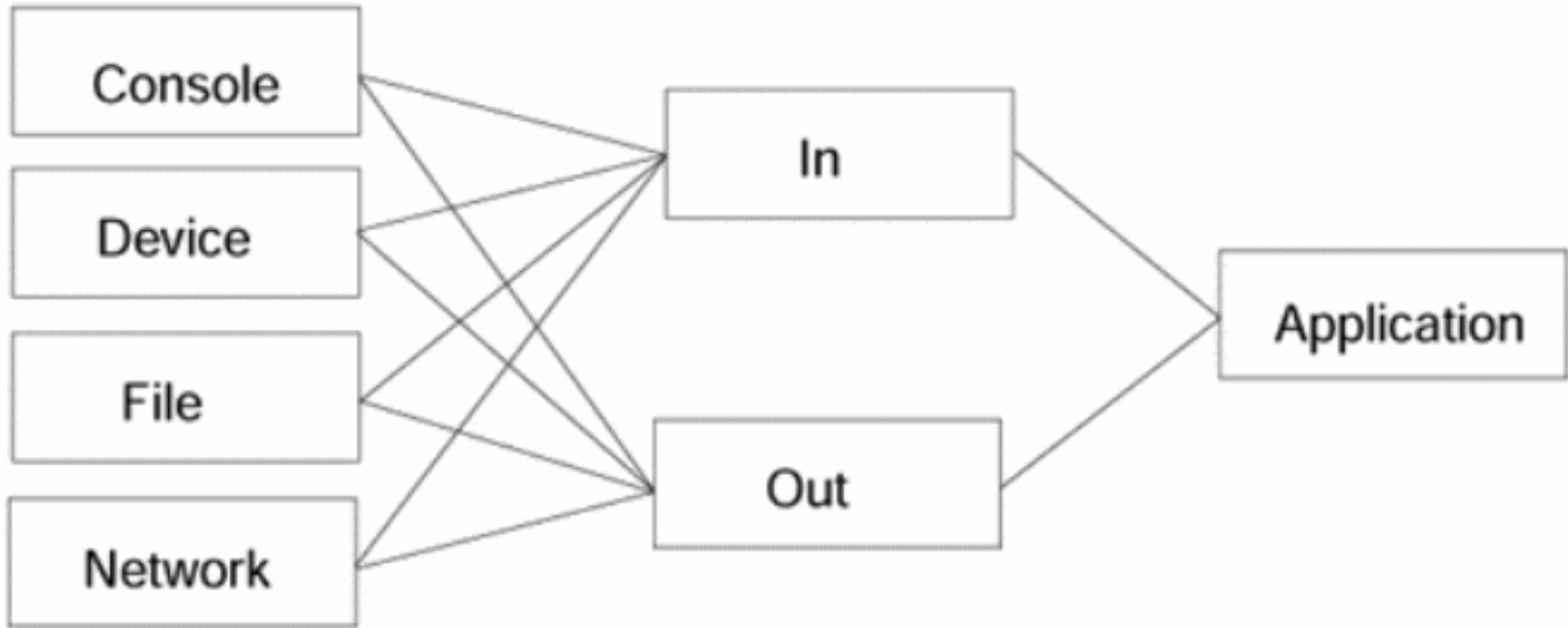


# Stream





# Entrada / Salida



# Sockets

# Socket

- **Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.**





- **Para que dos programas puedan comunicarse entre sí es necesario que se cumplan ciertos requisitos:**
  - Que un programa sea capaz de localizar al otro.
  - Que ambos programas sean capaces de intercambiarse cualquier secuencia de octetos, es decir, datos relevantes a su finalidad.
- **Para ello son necesarios los tres recursos que originan el concepto de socket:**
  - Un protocolo de comunicaciones, que permite el intercambio de octetos.
  - Una dirección del Protocolo de Red (Dirección IP, si se utiliza el Protocolo TCP/IP), que identifica una computadora.
  - Un número de puerto, que identifica a un programa dentro de una computadora.





- **Un puerto de red es una interfaz para comunicarse con un programa a través de una red.**
- **Un puerto suele estar numerado.**
- **La implementación del protocolo en el destino utilizará ese número para decidir a qué programa entregará los datos recibidos.**
- **Esta asignación de puertos permite a una máquina establecer simultáneamente diversas conexiones con máquinas distintas, ya que todos los paquetes que se reciben tienen la misma dirección, pero van dirigidos a puertos diferentes.**



# Direcciones de puertos

- Los números de puerto se indican mediante una palabra, 2 bytes (16 bits), por lo que existen 65535. Aunque podemos usar cualquiera de ellos para cualquier protocolo, existe una entidad, la IANA, encargada de su asignación, la cual creó tres categorías:
  - 1. Los puertos inferiores al 1024 son puertos reservados para el sistema operativo y usados por "protocolos bien conocidos". Si queremos usar uno de estos puertos tendremos que arrancar el servicio que los use teniendo permisos de administrador.



# Direcciones de puertos

- 2. Los comprendidos entre 1024 (0400 en hexadecimal) y 49151 (BFFF en hexadecimal) son denominados "registrados" y pueden ser usados por cualquier aplicación. Existe una lista publica en la web del IANA donde se puede ver qué protocolo usa cada uno de ellos.
- 3. Los comprendidos entre los números 49152 (C000 en hexadecimal) y 65535 (FFFF en hexadecimal) son denominados dinámicos o privados, porque son los usados por el sistema operativo cuando una aplicación tiene que conectarse a un servidor y por tanto necesita un puerto por donde salir.



## Direcciones de Puertos

No. de Puerto	Descripción
Números 255 e inferiores	Se asignan como aplicaciones públicas (como SMTP).
Números 256 a 1023	Se asignan a organizaciones para identificar productos de aplicación de red.
Números 1024 y superiores	Los asigna de manera dinámica la aplicación de usuario final usando la aplicación de red.

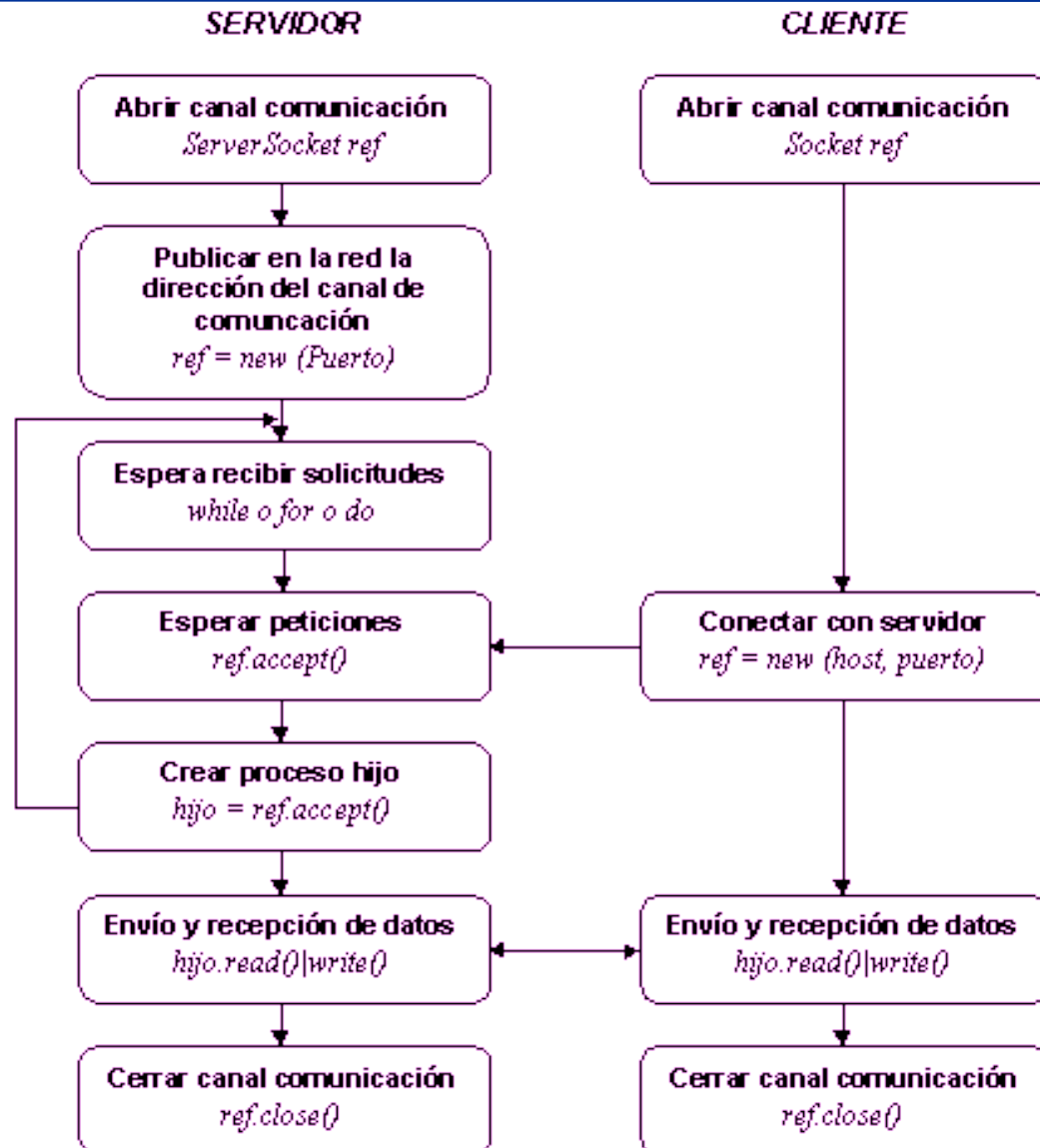
# Sockets

- Los sockets permiten implementar una arquitectura cliente-servidor. La comunicación ha de ser iniciada por uno de los programas que se denomina programa cliente. El segundo programa espera a que otro inicie la comunicación, por este motivo se denomina programa servidor.
- Un socket es un fichero existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

# Propiedades de sockets

- Las propiedades de un socket dependen de las características del protocolo en el que se implementan. El protocolo más utilizado es TCP, aunque también es posible utilizar UDP o IPX. Gracias al protocolo TCP, los sockets tienen las siguientes propiedades:
  - Orientado a conexión.
  - Se garantiza la transmisión de todos los octetos sin errores ni omisiones.
  - Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido.
- Estas propiedades son muy importantes para garantizar la corrección de los programas que tratan la información.
- El protocolo UDP es un protocolo no orientado a la conexión. Sólo se garantiza que si un mensaje llega, llegue bien. En ningún caso se garantiza que llegue o que lleguen todos los mensajes en el mismo orden que se mandaron. Esto lo hace adecuado para el envío de mensajes frecuentes pero no demasiado importantes, como por ejemplo, mensajes para los refrescos (actualizaciones) de un gráfico.

# Cliente Servidor

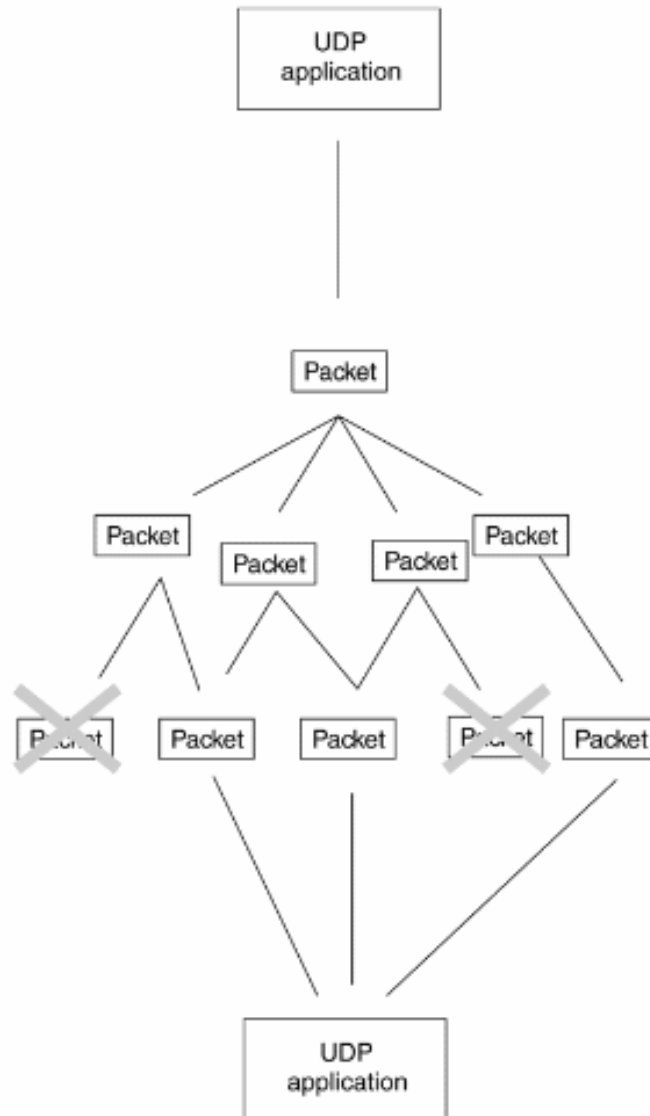


**UDP**



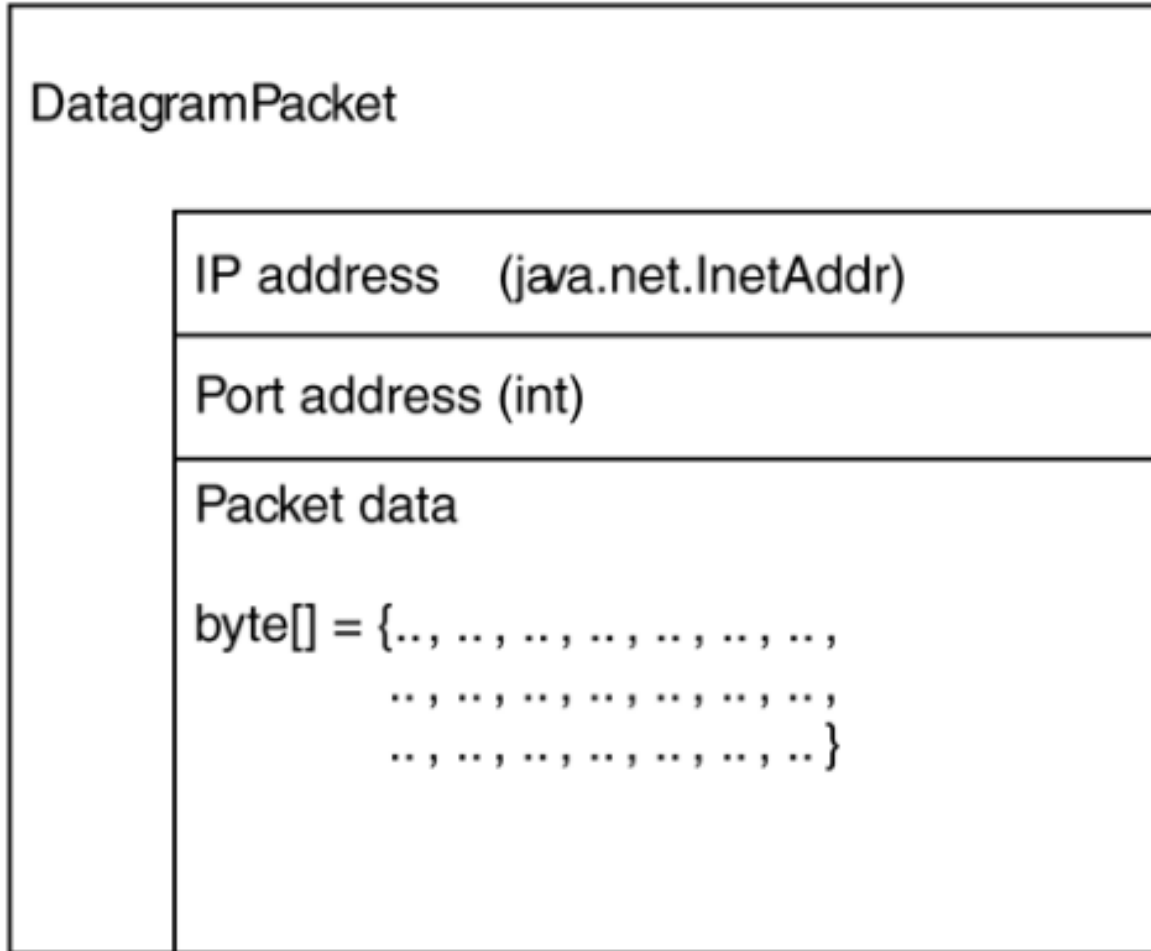


# Aplicación UDP



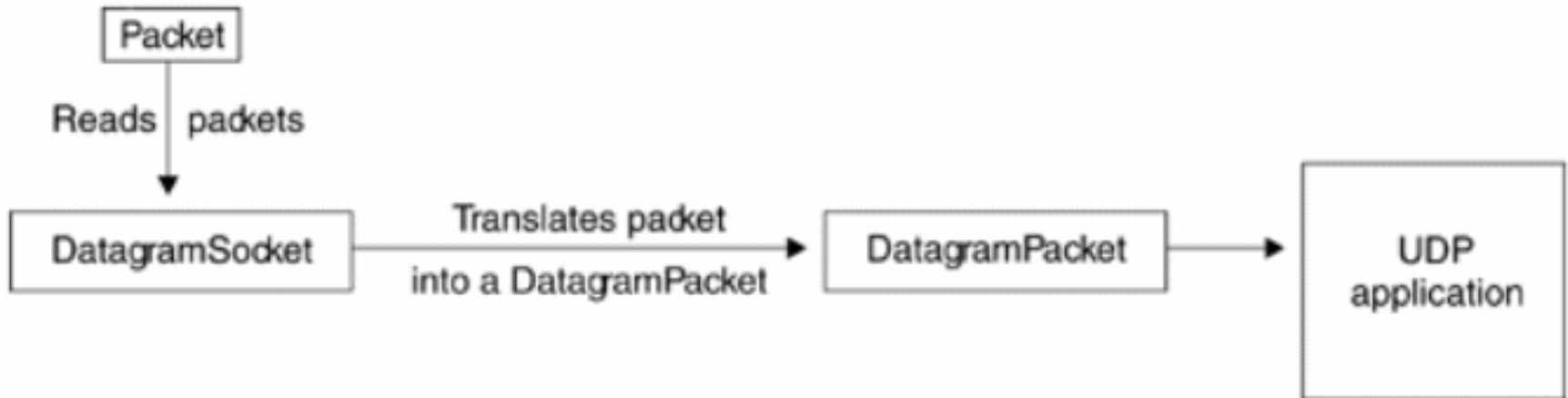


# Paquete UDP



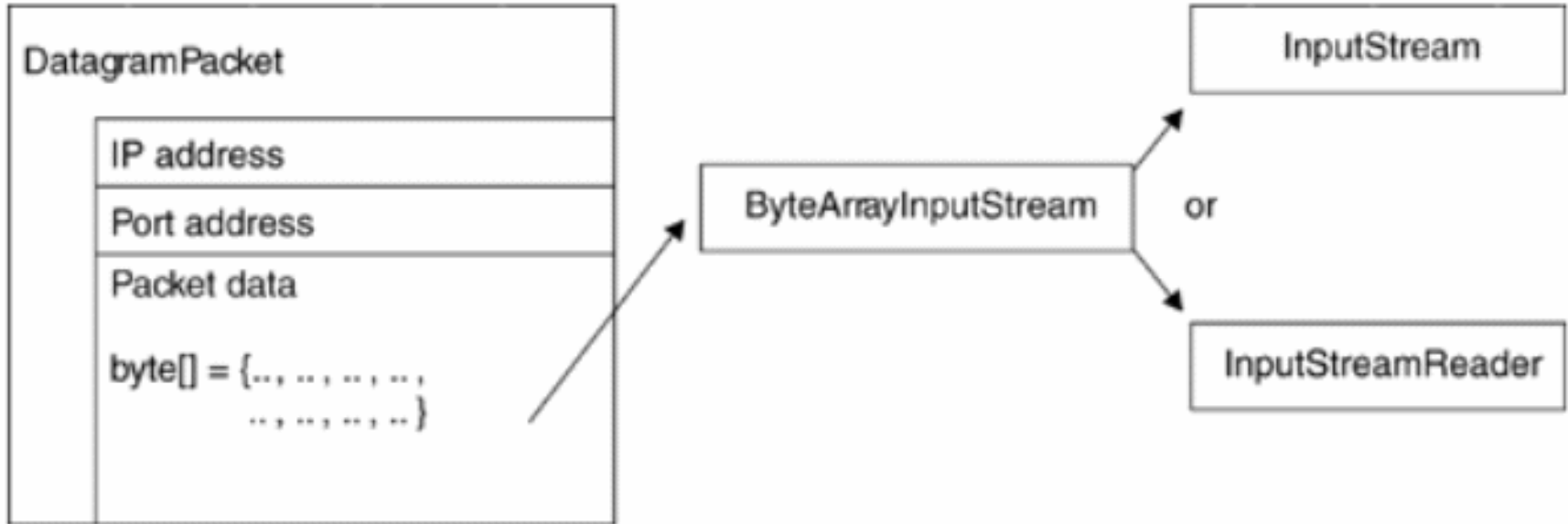


# Escuchar UDP

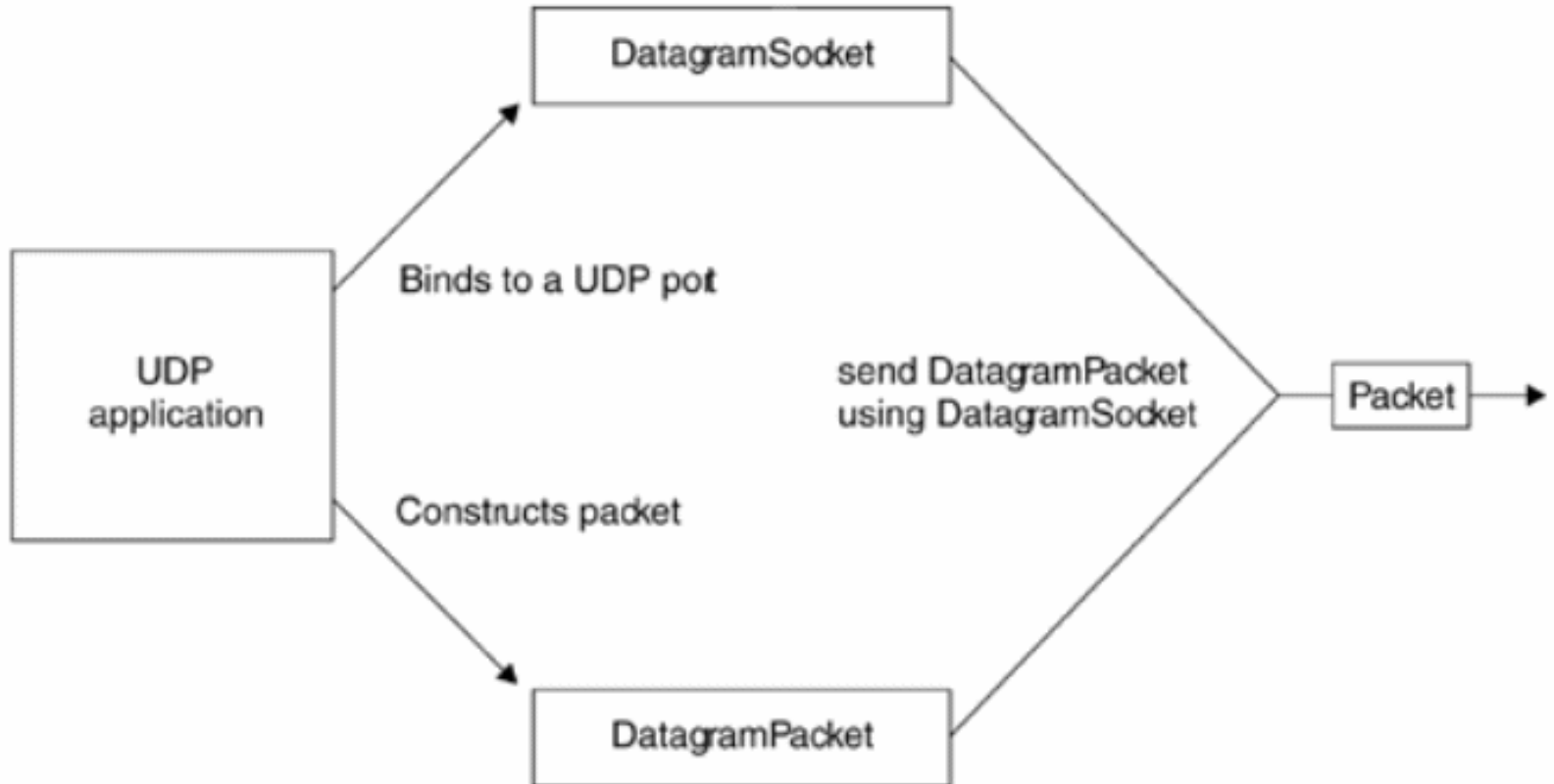




# Leer UDP

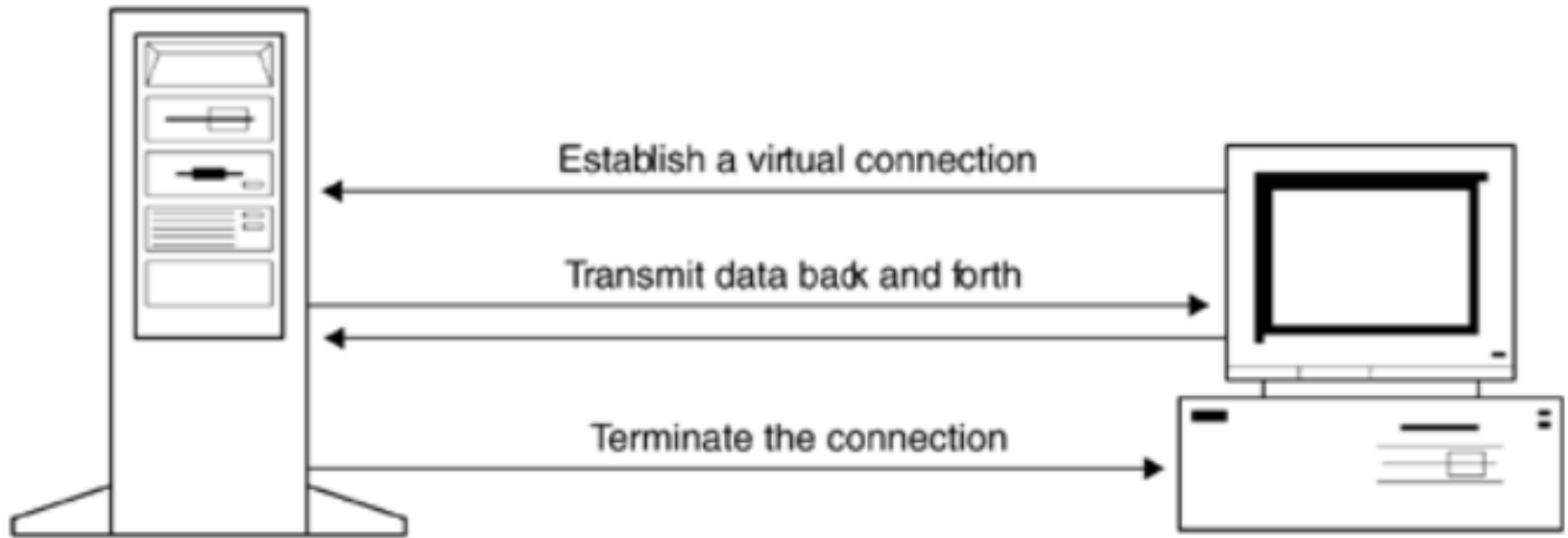


# Enviar UDP

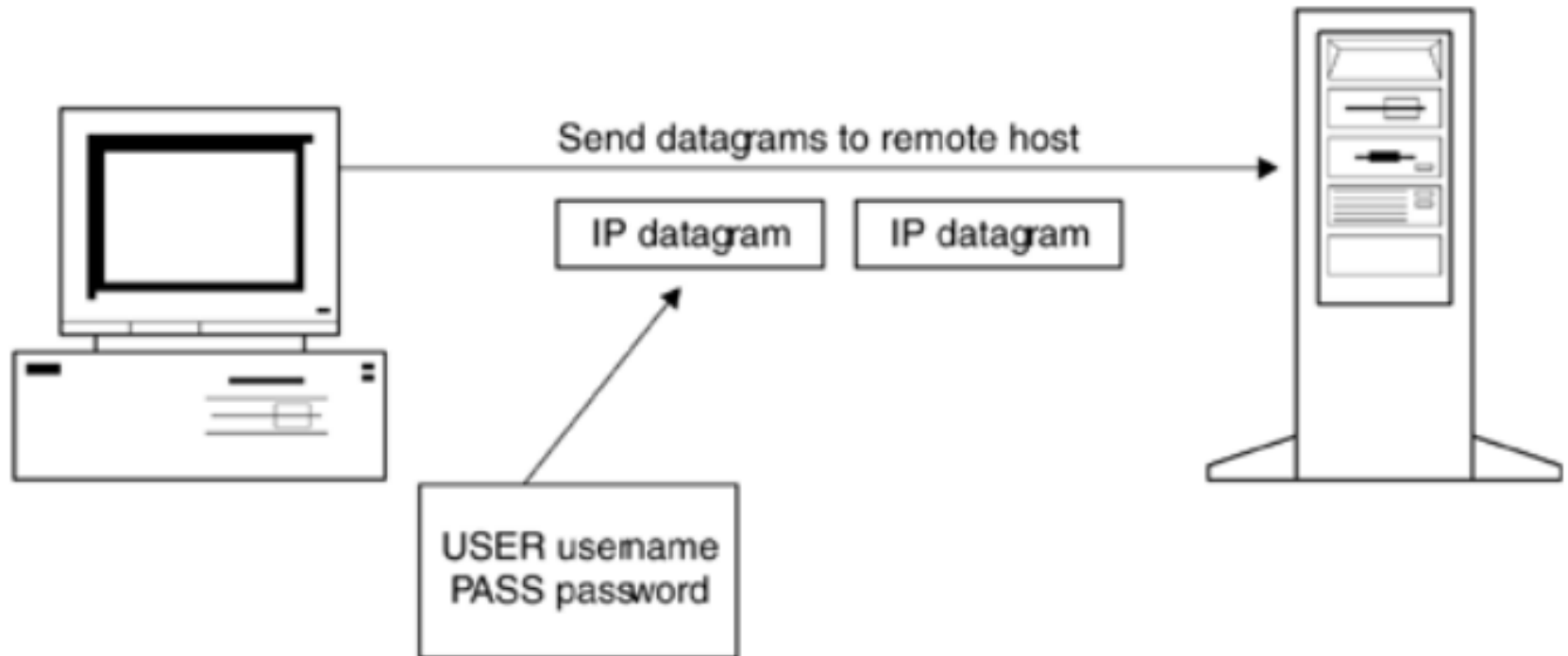


**TCP**

# Conexión TCP



# Conexión TCP







# Conexión TCP

