

POO

Streams en red

Concepto básico de redes – dirección IP

- Similar, desde el punto de vista lógico, al tradicional correo tradicional
 - Cada dirección identifica de manera única un objeto particular
- Cada computador conectado a Internet tiene una dirección IP (Internet Protocol)
- Se usa un número de 32 bits para identificar de manera única cada computador conectado a Internet
 - 193.144.189.63
 - macc.unican.es

DNS

LAN

'''

LP1

PC3

PC2

PC1

SWITCH

R

Protocolo

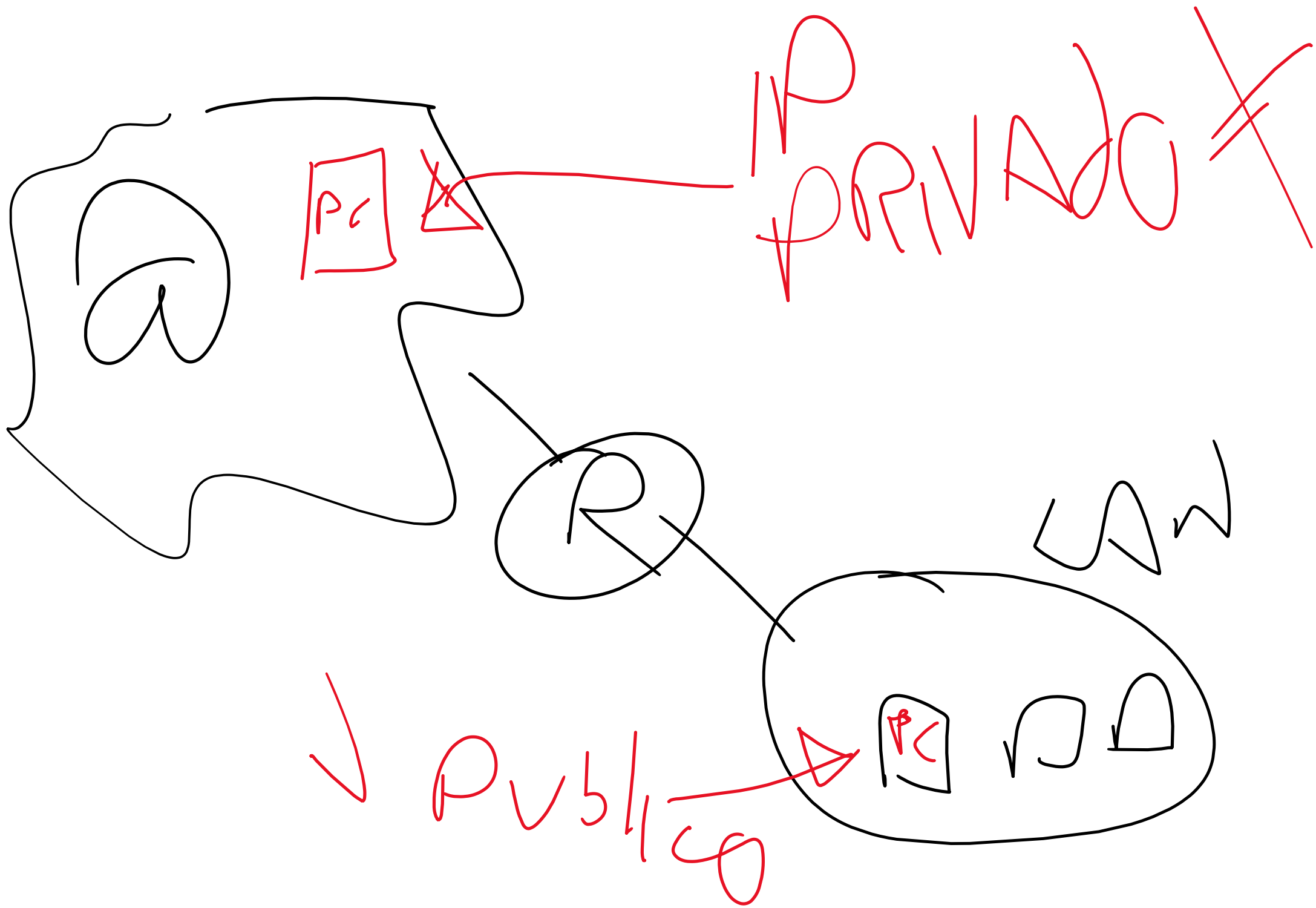
- Los protocolos son necesarios porque hay diferentes tipos de comunicación sobre Internet
- Cada tipo de comunicación requiere un protocolo específico y único
- Un protocolo se puede definir como el conjunto de reglas y estándares que definen un cierto tipo de comunicación Internet
- El protocolo describe el formato de dato enviado a Internet (cómo y cuando es enviado)
 - Los protocolos los usamos habitualmente (protocolo social)

102, 168, 3, 45

INTERNET

TIPO de DIR?

PRIVADA



Protocolo

- Algunos protocolos importantes usados en Internet son:
 - File Transfer Protocol (FTP), que permite transferir ficheros sobre Internet
 - Hypertext Transfer Protocol (HTTP), usada para transferir documentos HTML sobre la Web
- Ambos protocolos tienen su propio conjunto de reglas y estándares en la forma en que los datos se transfieren
- Java proporciona el soporte para ambos protocolos

44/3

Puertos

HTTPS

- Los protocolos tienen sentido cuando se usan en el contexto de un servicio

FTP

21

20

- El protocolo HTTP es usado cuando se proporciona contenido Web a través del servicio HTTP

- Cada computador en Internet puede proporcionar una variedad de servicios

- Para establecer una línea de comunicación a través de un protocolo y usar un servicio particular específico requiere la conexión al puerto apropiado

HTTP

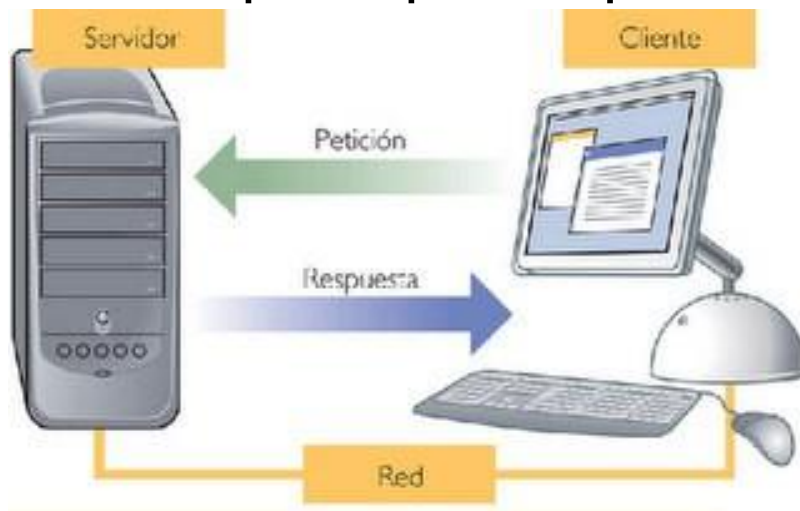
80

Puertos

- Un puerto es un número de 16 bits que identifica cada servicio ofrecido por el servidor de red
- Puertos estándar
 - Números asociados a un tipo de servicio particular.
Ejemplos: el puerto del servicio FTP es 21, el puerto del servicio HTTPS es 443, el puerto del servicio TELNET es 23
 - Los valores de puerto son < 1024
- Valores de puertos ≥ 1024
 - Disponibles para una comunicación personalizada
 - Si ya está en uso, debe buscarse otros valores no usados

El paradigma cliente/servidor

- Básico en la plataforma de red Java
- Involucra dos elementos principales:
 - Cliente, máquina que necesita de algún tipo de información
 - Servidor, máquina que almacena información y que espera por responder a una petición



Escenario:

- El cliente se conecta al servidor y solicita cierta información
- El servidor recibe la petición y responde al cliente con la información

http

64.1.3.81:80

socket

80

31,115

host

64.1.3.81

190.11.11.10

Sockets - definiciones

- Abstracción software para un medio de comunicación de entrada o salida
- Canales de comunicación que permiten transferir datos a través de un puerto en particular
- Punto de comunicación o de conexión lógica entre dos máquinas
- Tipo particular de comunicación en red usada por Java en la programación de redes
- Java realiza toda la comunicación de bajo nivel a través de sockets

Sockets - tipos

- Hay dos tipos de sockets:
 - Servidores
 - Clientes
- Un socket servidor espera las solicitudes de conexión de clientes.
- Un socket cliente se puede usar para enviar y recibir datos.

Package de Networking Java

- Java proporciona el paquete `java.net` que proporciona clases útiles para el desarrollo de aplicaciones de red
- Algunas clases en el package son:
 - `ServerSocket`
 - `Socket`
 - `MulticastSocket`
 - `DatagramSocket`

La clase ServerSocket

- Proporciona la funcionalidad básica de un servidor
- Escuchan un *puerto* específico, consistente en un número único
 - El número del puerto es necesario para distinguir diferentes servidores ejecutándose en una misma máquina
- Deben ejecutarse antes que los clientes inicien la comunicación

La clase ServerSocket

Resumen de Constructores

`ServerSocket()`

Crea un server socket sin asociarse a un puerto.

`ServerSocket(int port)`

Crea un server socket, asociado a un Puerto especifico.

`ServerSocket(int port, int backlog)`

Crea un server socket, asociado a un Puerto especifico, con registro de conexiones pendientes.

`ServerSocket(int port, int backlog, InetAddress bindAddr)`

Crea un server socket, asociado a un Puerto especifico, con registro de conexiones pendientes, y direcciones IP locales para asociar.

La clase `ServerSocket` - creación

- En su forma más simple se crean instancias de la clase `ServerSocket` mediante cualquiera de los siguientes constructores:

`ServerSocket(int port)`

`ServerSocket(int port, int backlog)`

- *port* es el número de puerto que atiende el socket.
- Cuando varios clientes contactan con el servidor al mismo tiempo, se colocan en una cola y se atienden en el orden de recepción.

La clase ServerSocket - métodos

- Métodos usados comúnmente:
accept() Espera por una conexión. El thread que ejecuta el método se bloqueará hasta que se recibe una solicitud, devolviendo un socket cliente.
close() para recibir solicitudes de clientes.
- La comunicación con sockets se gestiona con input / output streams.

La clase Socket (Cliente)

- Implementa un cliente socket. Se consigue de dos formas:

lado cliente: `Socket(String host, int port)` lado
servidor: metodo `accept()` de `ServerSocket`

- Cada cliente socket tiene un objeto `InputStream` para recibir datos y un objeto `OutputStream` para enviar datos.
- Métodos usados:
`getInputStream()` Devuelve un objeto `InputStream`
`getOutputStream()` Devuelve un objeto `OutputStream`
`close()` Cerrar la conexión

Actividad

- Investigar que protocolos son soportados por Java???
 - Ejem, http y ftp
 - Indique algunos clases , métodos y métodos constructores.