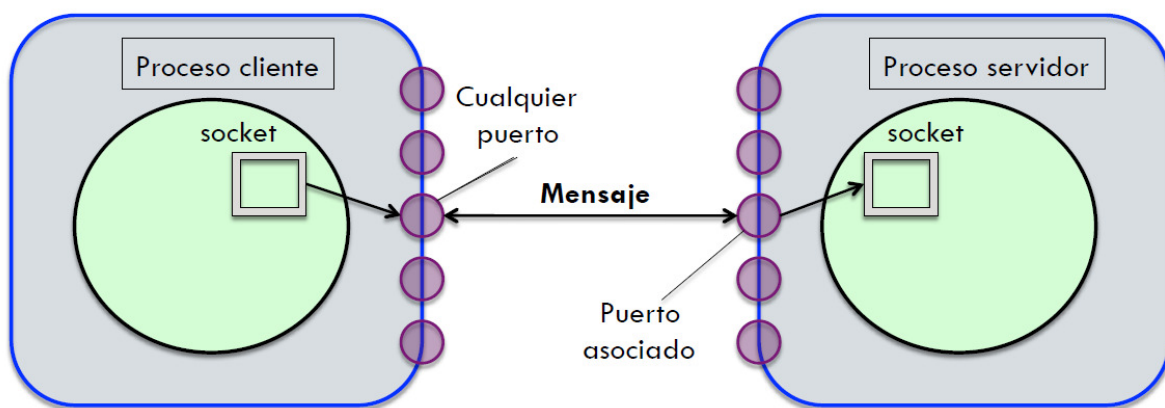


Tema 2: Paradigmas de la programación distribuida

Sockets

Es un mecanismo de comunicación entre procesos ejecutados en distintas máquinas que permite el envío y la recepción de mensajes entre ellos.

Funciona de tal forma que su proceso remoto asocia su socket a un puerto concreto y el IP del nodo donde se ejecuta.



Hay que recordar que los puertos 0 - 1024 son privados y >1024 públicos

RPC

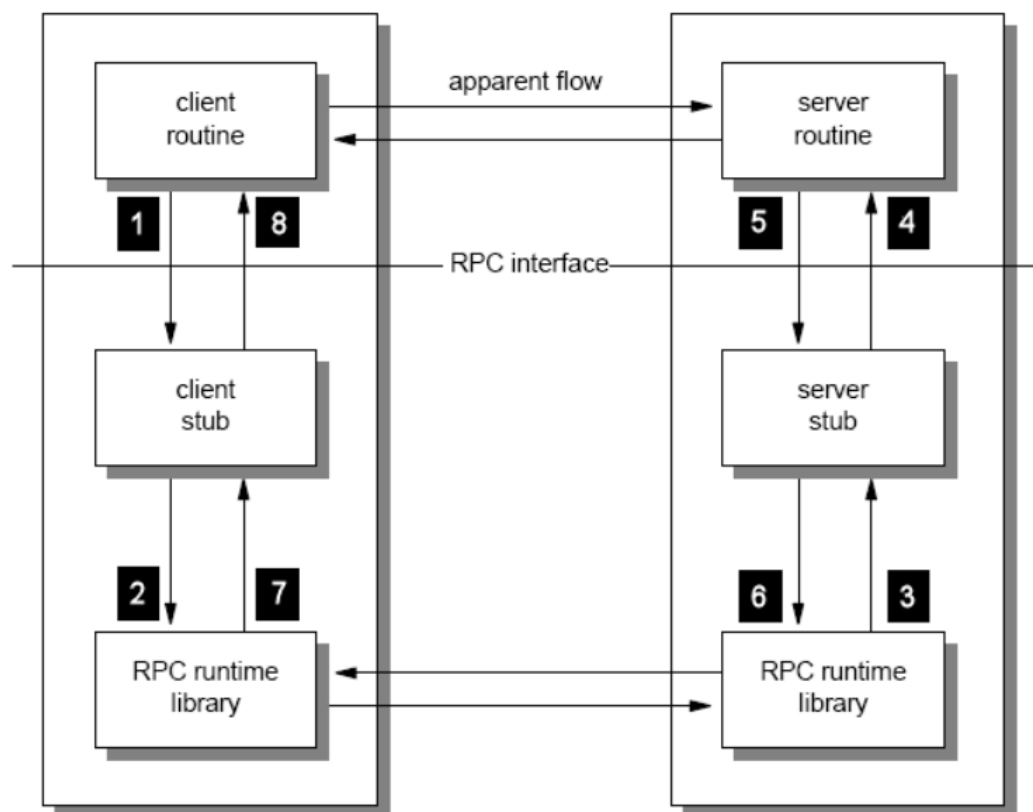
El objetivo es que la programación de aplicaciones de la sensación que se ejecutan en local, simulando una programación no distribuida. Podemos decir que es **una invocación a procedimientos** donde el cliente invoca a un stub que se va a generar automáticamente donde la comunicación se hace de forma oculta al programador.

Una característica importante que tenemos es:

- El envío de datos por la red → enviamos en un formato estandar:

- **Marshaling:** Es el proceso de transformación de los datos en un formato adecuado para su almacenamiento o transferencia.
- **Unmarshaling:** Transforma un flujo de bytes en una estructura de datos (porque en RPC sólo podemos enviar un dato → enviamos ésta estructura), es el proceso inverso al marshaling.

Un **stub** es un representante local del participante remoto, por ejemplo del servidor en el cliente nos permitiría hacer llamadas locales al procedimiento como si fuera el servidor real.



Fuente: IBM Redbook.TCP/IP Tutorial and Technical Overview

- La **desventaja** de sockets sobre RPC es que en RPC nosotros podemos en un sólo envío pasar múltiples datos mientras que en sockets tenemos que tener más detalle al sincronizar cada envío.
- La **ventaja** de sockets sobre RPC es la facilidad de detalle que podemos tener para realizar aplicaciones a medida (por ejemplo para tablas de bbdd), aunque perdamos un poco de rendimiento.

ONC - RPC

XDR - External Data Representatio: es un protocolo de presentación de datos que permite la transferencia de datos entre máquinas de diferentes arquitecturas y sistemas operativos.

Port Mapper o RPCbind: Es un servicio de nombres que incluye operaciones para localizar servicios, usa un puerto fijo (111), éste servicio se encarga de devolver un número de puerto donde hacer la conexión.



EXAMEN! ¿Si el cliente no indica el puerto en RPC, cómo sabe a quién se tiene que conectar? → Haciendo uso del servicio de nombres llamado rpcbind o port mapper en donde a partir de un número de programa y un programa el cliente pide al port mapper que le devuelva el puerto por el cual se tiene que conectar.

Comandos útiles:

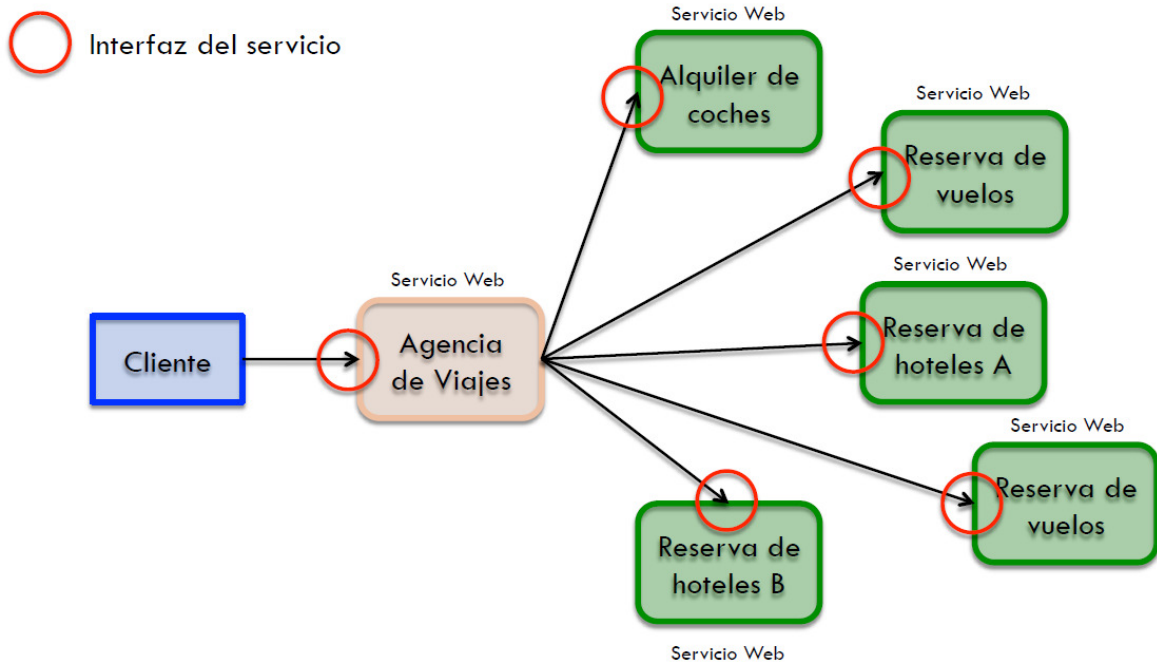
- `rpcbind`
- `rpcinfo`
- `rpcgen`
- `sudo service rpcbind stop` O `sudo service portmap stop`
- `sudo rpcbind -wi`
- en el server `./programa_server`
- en el cliente `./programa_client argumentos`

Servicios Web

Nace bajo la idea de que 2 servicios que pueden estar escritos en lenguajes de programación distintos como c y java puedan interactuar sin necesidad de un navegador, esto se consigue gracias a la serialización de los datos, todos se comparten usando XML y en general usando el protocolo SOAP.



La idea es poder concatenar web services para consultar lo que necesitamos.



EXAMEN! ¿Qué es un web service? → Los web services son una tecnología que utiliza un conjunto de protocolos estándares que sirven para intercambiar datos XML entre aplicaciones.

Ventajas

- Interoperabilidad
- Van sobre HTTP y pasan los filtros firewall
- Basado en estándares abiertos
- Pueden encadenar el uso de servicios

Desventaja

- Tienen mucho overhead ya que tienen que generar el documento XML y este lleva más información a nivel texto



EXAMEN! Protocolos estándares de web service → HTTP, XML, SOAP, WSDL, UDDI

XML es un lenguaje de marcado que permite definir nuestras propias etiquetas.

Hay que resaltar que en gSoap - Web service, a diferencia de RPC, si nos dejan enviar varios argumentos pero siempre tiene que ir la respuesta en el último de tal forma que tendríamos:

```
int funcion_primera(int a, int b, int c, int *res);
```

Para el examen hay que saber ejecutar, no tanto codificarlo todo.

SOAP - Simple Object Access Protocol

Este protocolo especifica cómo representar los mensajes de texto en XML, procesamiento de los elementos de los mensajes, cómo utilizar el protocolo de aplicación (HTML, SMTP, etc)

WSDL - Web Service Description Language

Es un lenguaje que se utiliza para definir la interfaz (IDL - Interface Definition Language) de los servicios web en XML, describe qué mensajes se van a enviar, el formato de éstos y la localización y mecanismos de acceso al servicio. Este documento está escrito también en XML, en el servidor nos indica cómo activar el servicio y en el cliente para generar el proxy.

UDDI - Universal Description Discovery Integration

Es una especie de directorio a los cuales los servicios web pueden consultar para encontrar otros servicios a los cuales consultar según su actividad, localización geográfica, tipo de servicio, etc.

