Sistemas Distribuidos e Internet

Seminario 2a

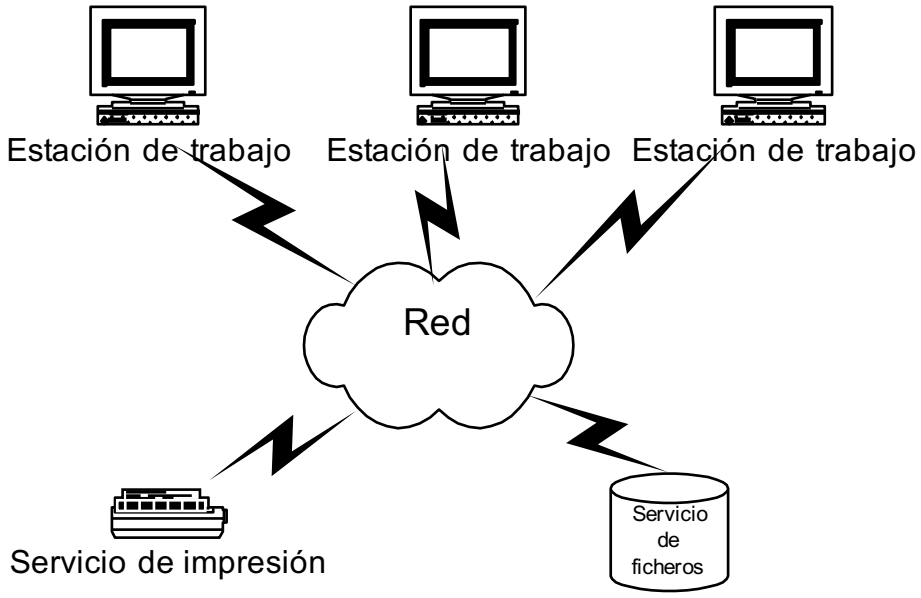
Repaso general de la web

Índice

- Introducción a los Sistemas Distribuidos
  - Desafíos
  - Paradigmas de computación distribuida
- WWW
  - HTTP
  - URL
  - HTML
  - Servidores web

Introducción a los SD

- "Un SD es aquel en el que los componentes hardware y software ubicados en los computadores en red se comunican y coordinan sus acciones intercambiando mensajes"



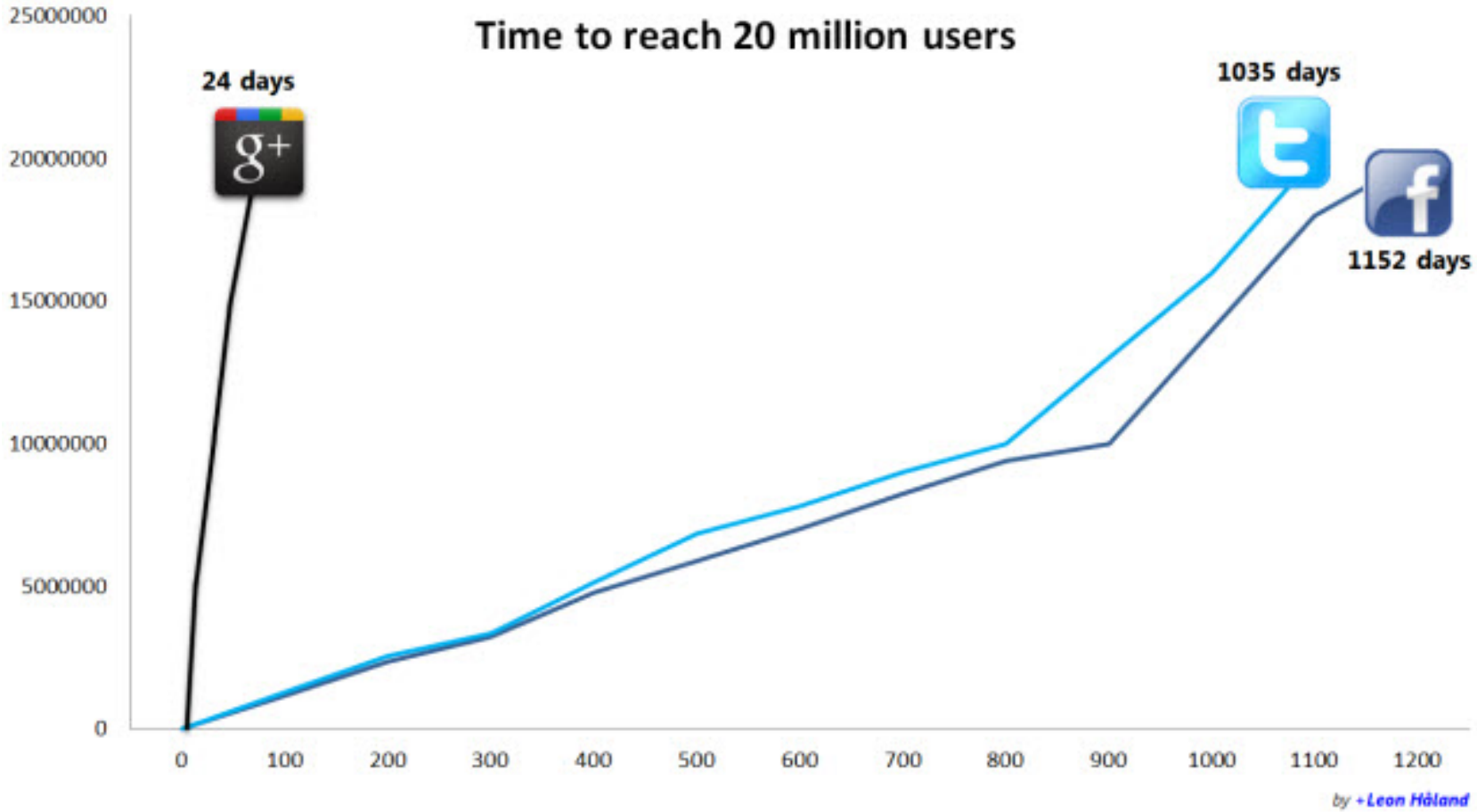
Introducción a los SD

- Consecuencias derivadas de la definición
  - Concurrencia
  - Inexistencia de reloj global
  - Fallos independientes
- ¿Por qué construir SD?
  - Para compartir recursos
  - Porque existen aplicaciones inherentemente distribuidas
- Internet es el mejor ejemplo de un SD

Introducción a los SD > Desafíos

- Heterogeneidad
  - Distintas redes, HW, SO, lenguajes de programación
- Extensibilidad
  - Especificaciones, estándares, APIs
- Seguridad
  - Confidencialidad, integridad, disponibilidad
- Escalabilidad
  - El software no debería cambiar al aumentar la escala del sistema

Introducción a los SD > Desafíos

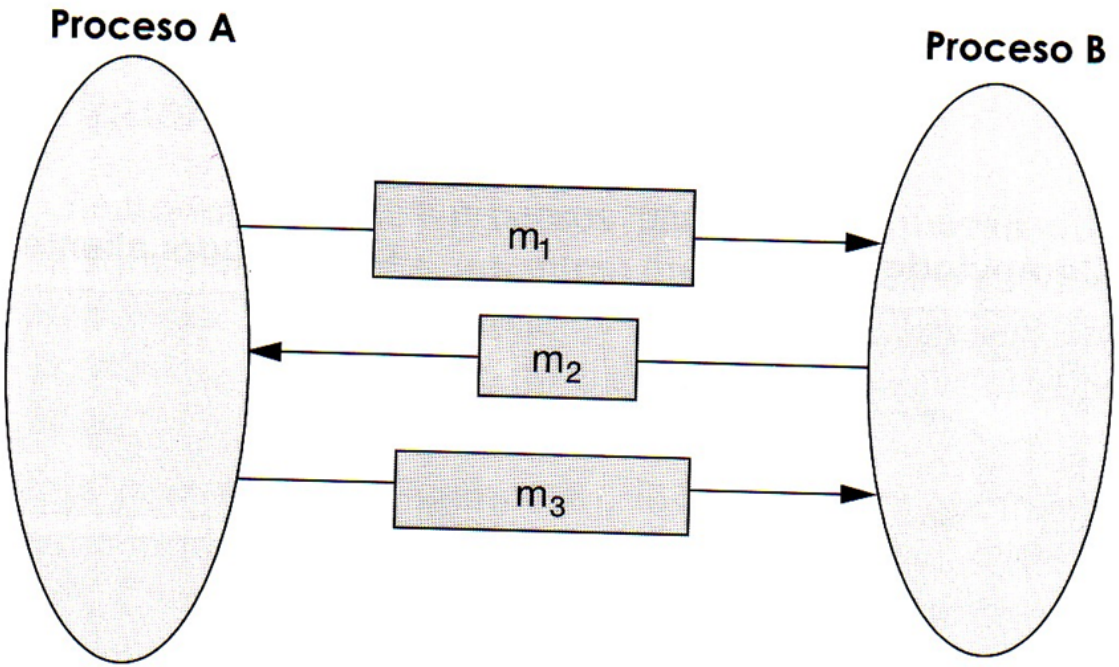


Introducción a los SD > Desafíos

- Gestión de fallos
  - Detectarlos, ocultarlos, tolerarlos, recuperación, redundancia
- Concurrencia
  - Es necesario garantizar el acceso sincronizado a recursos compartidos
- Transparencia
  - De acceso, de localización, de concurrencia, de replicación, de fallo, de migración, de rendimiento, de escalado

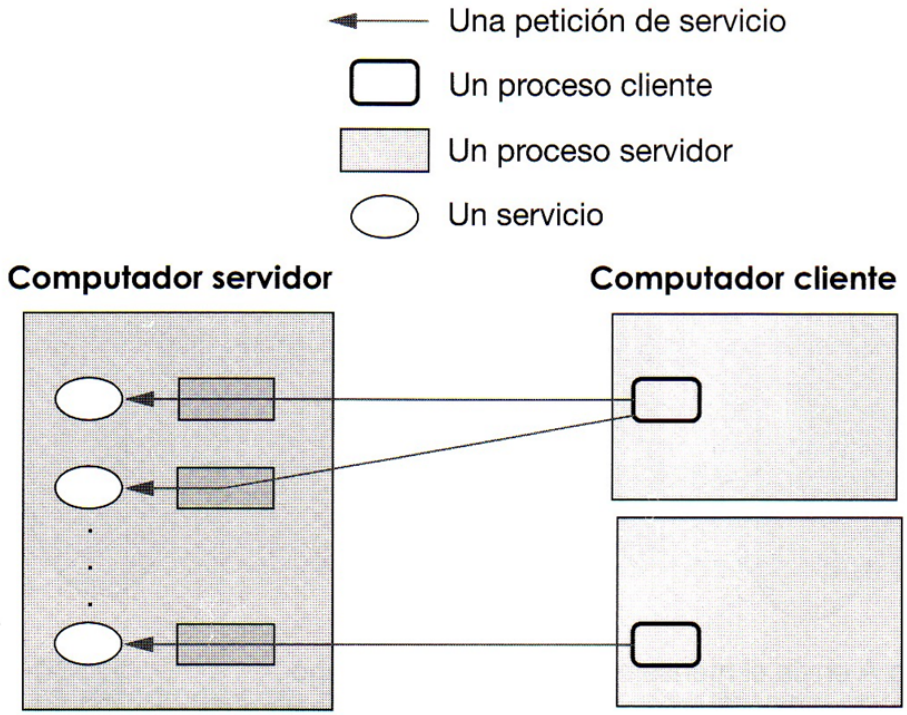
Intro. a los SD > Paradigmas de computación distribuida

- Paso de mensajes
  - Los datos, encapsulados en mensajes, se intercambian entre un emisor y un receptor
    - *sockets*



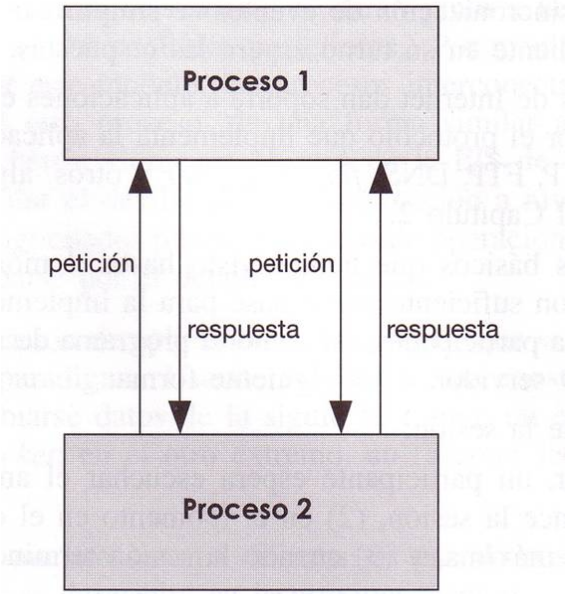
Intro. a los SD > Paradigmas de computación distribuida

- Cliente/servidor
  - Quizá, el más conocido (HTTP, FTP, etc.)
  - Asigna roles diferentes a 2 procesos que cooperan



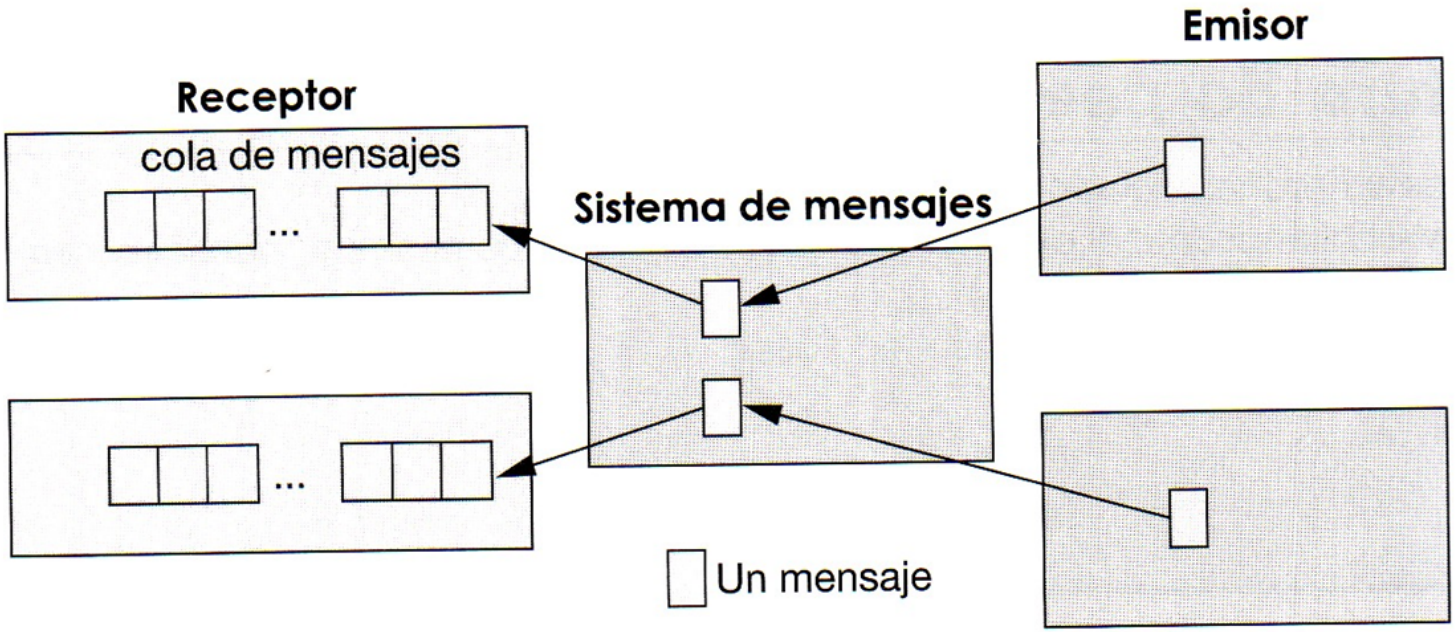
Intro. a los SD > Paradigmas de computación distribuida

- Igual a igual
  - Los procesos participantes interpretan los mismos papeles, con idénticas capacidades y responsabilidades
- Apropiado para aplicaciones como
  - Mensajería instantánea
  - Transferencia de ficheros
  - Vídeo-conferencia
  - Trabajo colaborativo



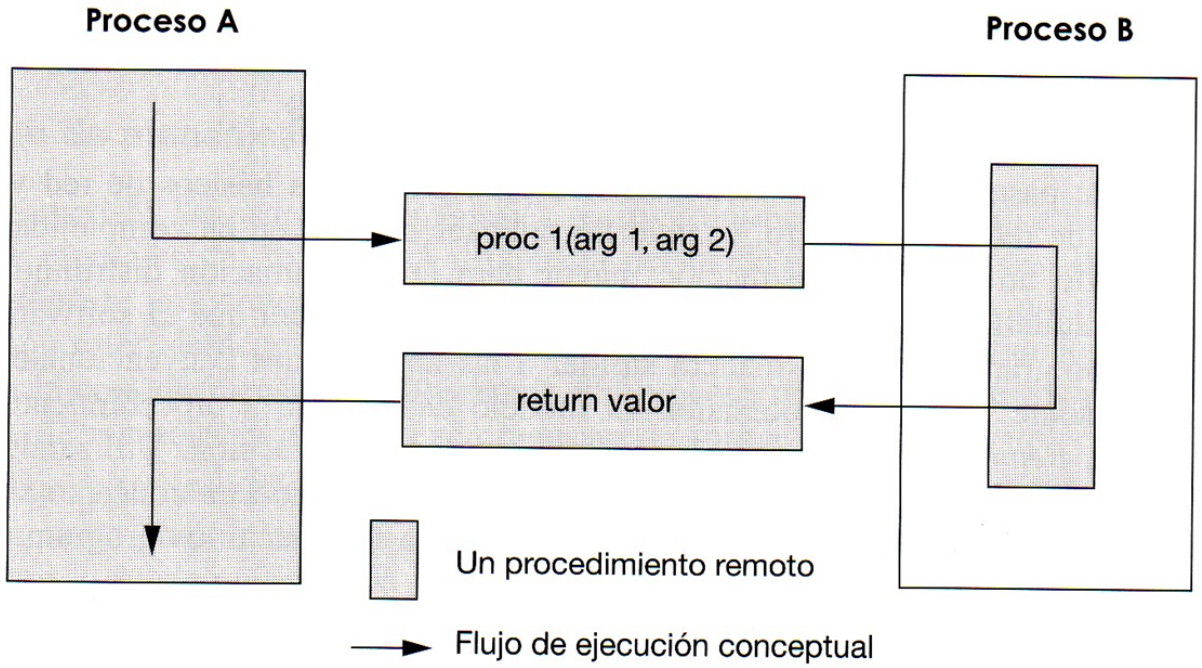
Intro. a los SD > Paradigmas de computación distribuida

- El MOM (*Message-Oriented Middleware*) es una evolución del paso de mensajes
  - JMS, MQS, MSMQ
  - Se introduce un intermediario



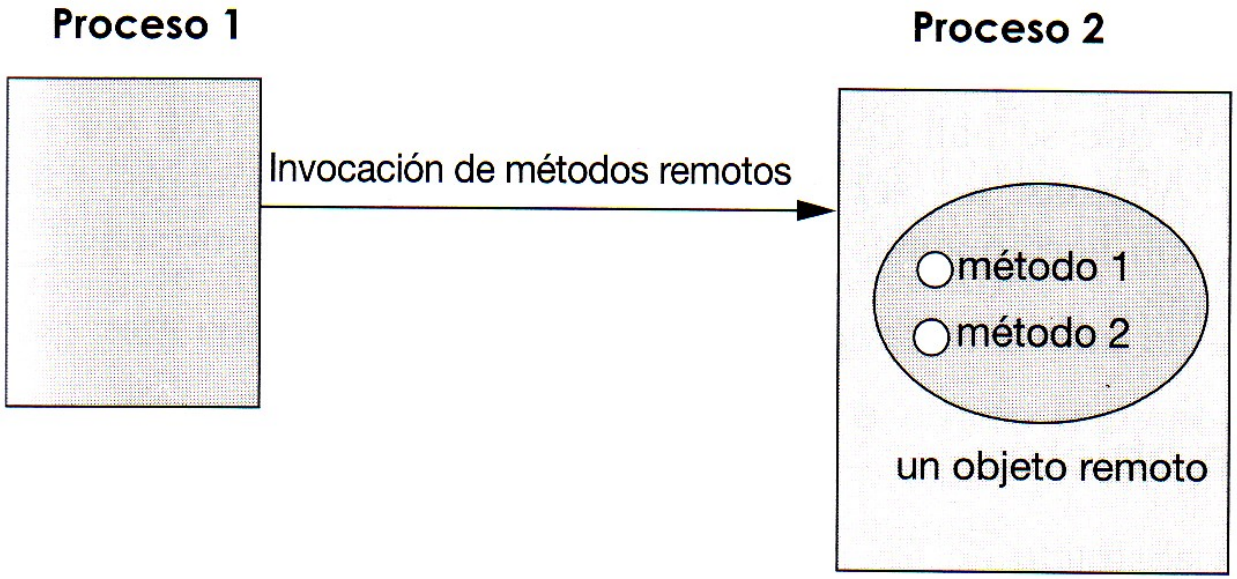
Intro. a los SD > Paradigmas de computación distribuida

- Llamadas a procedimientos remotos (RPC)
  - La comunicación entre dos procesos se realiza de manera similar a la invocación a un procedimiento local



Intro. a los SD > Paradigmas de computación distribuida

- La invocación de métodos remotos (*Remote Method Invocation*, RMI) es el equivalente en la orientación a objetos al RPC
  - CORBA, RMI



Intro. a los SD > Paradigmas de computación distribuida

- En general, todas las implementaciones de los paradigmas anteriores
  - Eran sistemas cerrados
    - Difícil añadir servicios por terceros por no existir especificaciones públicas de los sistemas
  - Utilizaban protocolos propietarios para el intercambio de información
  - Todo ello dificultaba su implantación en entornos heterogéneos
- Solución
  - Utilizar protocolos y especificaciones abiertas
    - Por ejemplo, HTTP

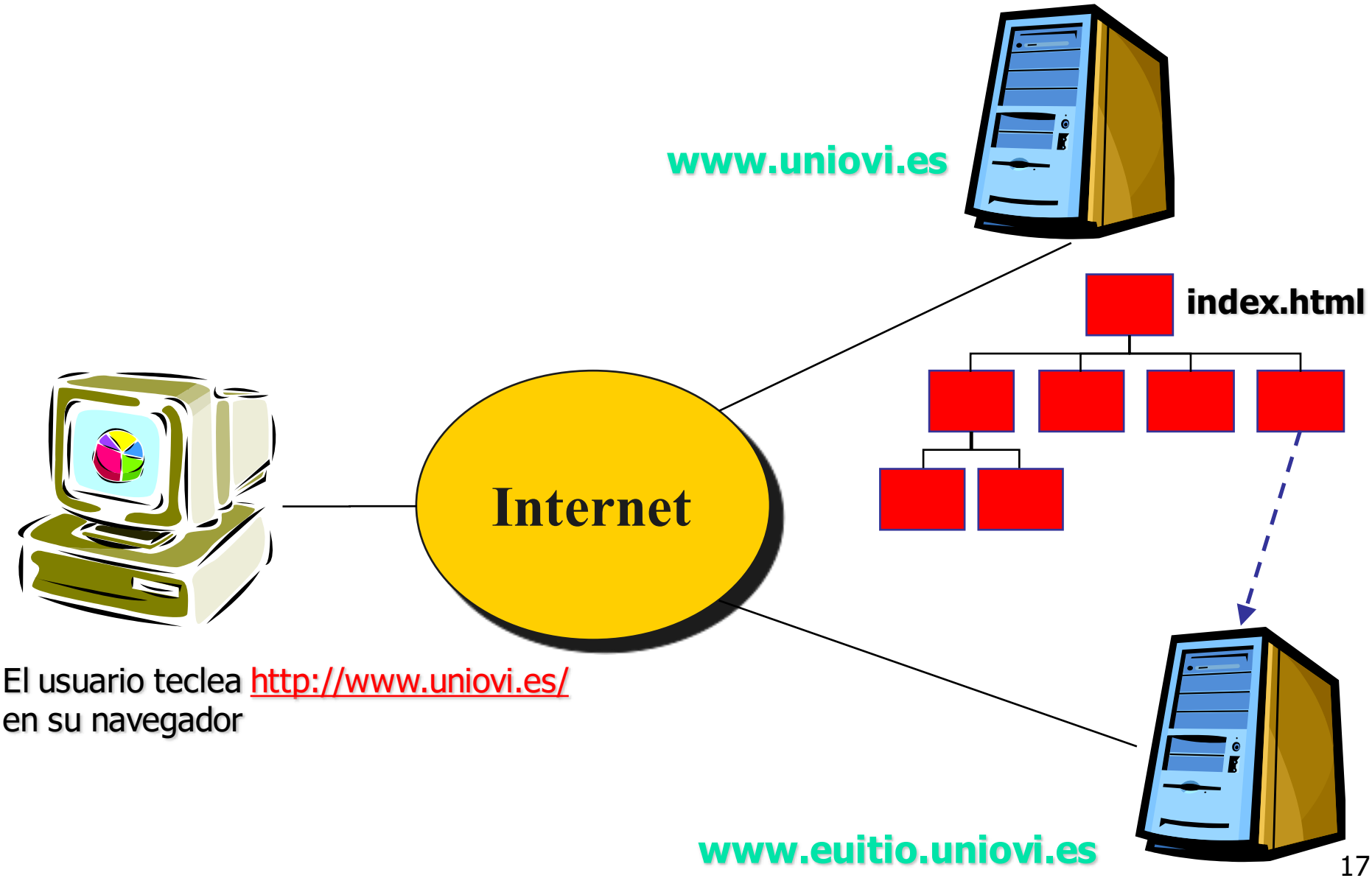
Índice

- Introducción a los Sistemas Distribuidos
  - Desafíos
  - Paradigmas de computación distribuida
- WWW
  - HTTP
  - URL
  - HTML
  - Servidores web
- Servidores de aplicaciones

WWW

- Sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet
  - Es, por tanto, un servicio que funciona sobre Internet
- Creada por Tim Berners-Lee en 1989
- Cuatro componentes básicos
  - HTTP, HTML, un servidor web, un navegador

WWW



El usuario teclea http://www.uniovi.es/
en su navegador

WWW > HTTP

- Es el protocolo utilizado en toda transacción en la WWW
  - Define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, etc.) para comunicarse
- A la información transferida se le denomina recurso y se identifica mediante un Localizador Uniforme de Recursos (URL)
- Es un **protocolo sin estado**
  - La **sesión** finaliza tan pronto como se devuelve el recurso solicitado

WWW > HTTP

- Ejemplo de un diálogo HTTP
  - Petición

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: nombre-cliente
[Línea en blanco]
```

  - Respuesta

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221
<html>
<body>
<h1>Página principal de tu host</h1>
(Contenido) . . .
</body>
</html>
```

WWW > HTTP

- Métodos de petición
  - HEAD, GET, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT
- Códigos de respuesta
  - 1xx: respuestas informativas
  - 2xx: petición correcta
  - 3xx: redirecciones
  - 4xx: errores del cliente
  - 5xx: errores del servidor

WWW > URL

- Es una secuencia de caracteres que sigue un formato estándar y se utiliza para nombrar recursos en Internet para su localización o identificación
  - Formato general
    - esquema://máquina:puerto/directorio/archivo
  - Ejemplos
    - `http://www.uniovi.es:80/estudiantes/`
    - `ftp://petra.euitio.uniovi.es/pub/notas.pdf`
    - `file:///c:/Users/USUARIO/Desktop/`

WWW > URL

- La URL puede incluir una cadena de consulta (*query string*)
  - Se utiliza para refinar la consulta sobre un recurso dado
  - Es una lista de parejas "parámetro=valor"
    - `http://directo.uniovi.es/catalogo/DetalleCentroDpto.asp?departamento=34&centro=66`

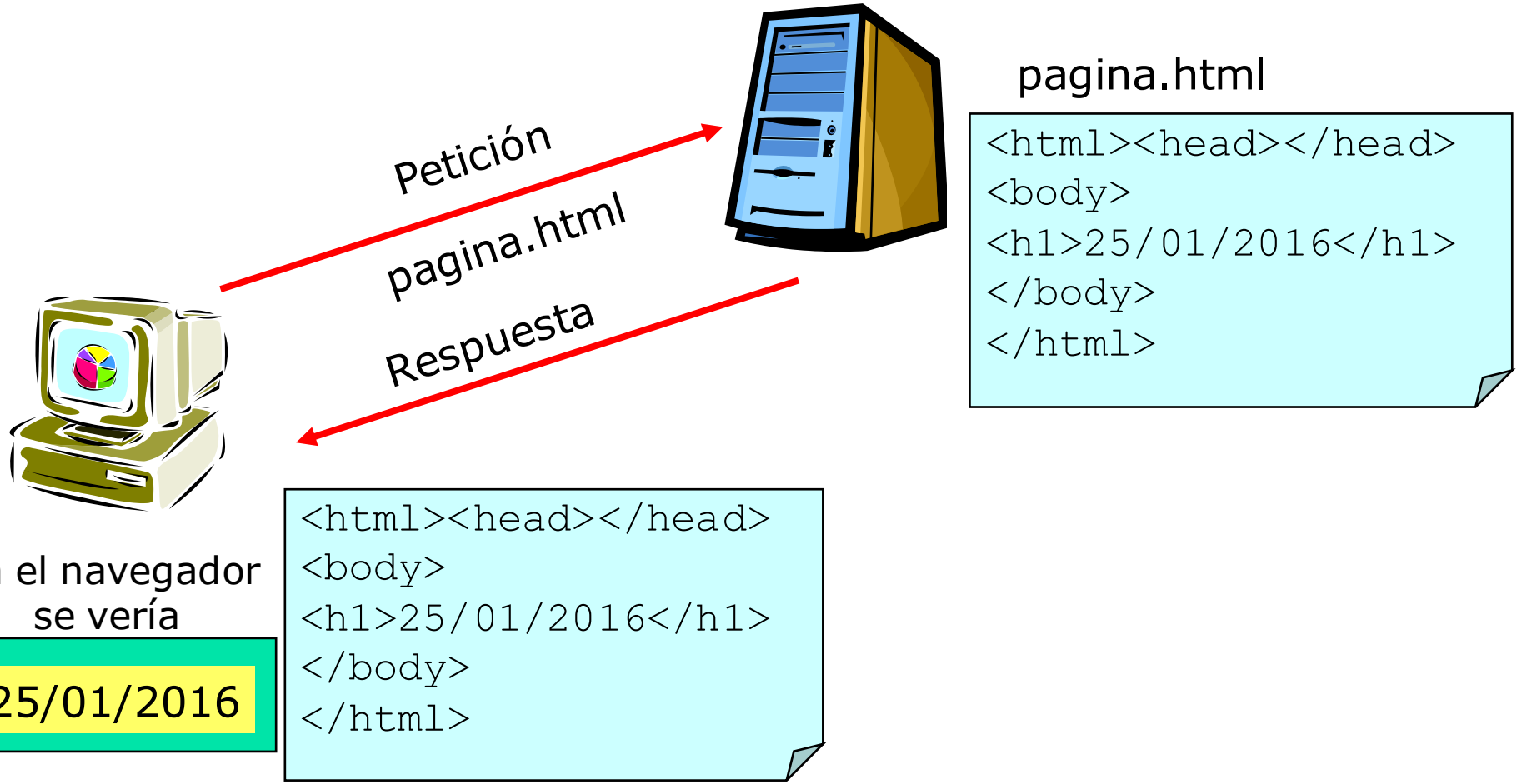
WWW > HTML

- Es un lenguaje de marcado utilizado para estructurar páginas web que van a ser mostradas en un navegador
  - Como tal lenguaje, está definido por una gramática
  - Permite incrustar imágenes, vídeo y otros objetos
  - Permite la creación de formularios interactivos
    - Posibilitan el envío de información de usuario del navegador al servidor

WWW > Servidores web

- Se refiere al hardware (ordenador) o software (programa) que ayuda a entregar contenido web que puede ser accedido en Internet
  - Apache, Internet Information Server (IIS), etc.
- En el origen de la web, sólo eran capaces de servir páginas estáticas
  - El servidor web localizaba el recurso solicitado a partir de la URL y lo devolvía al cliente

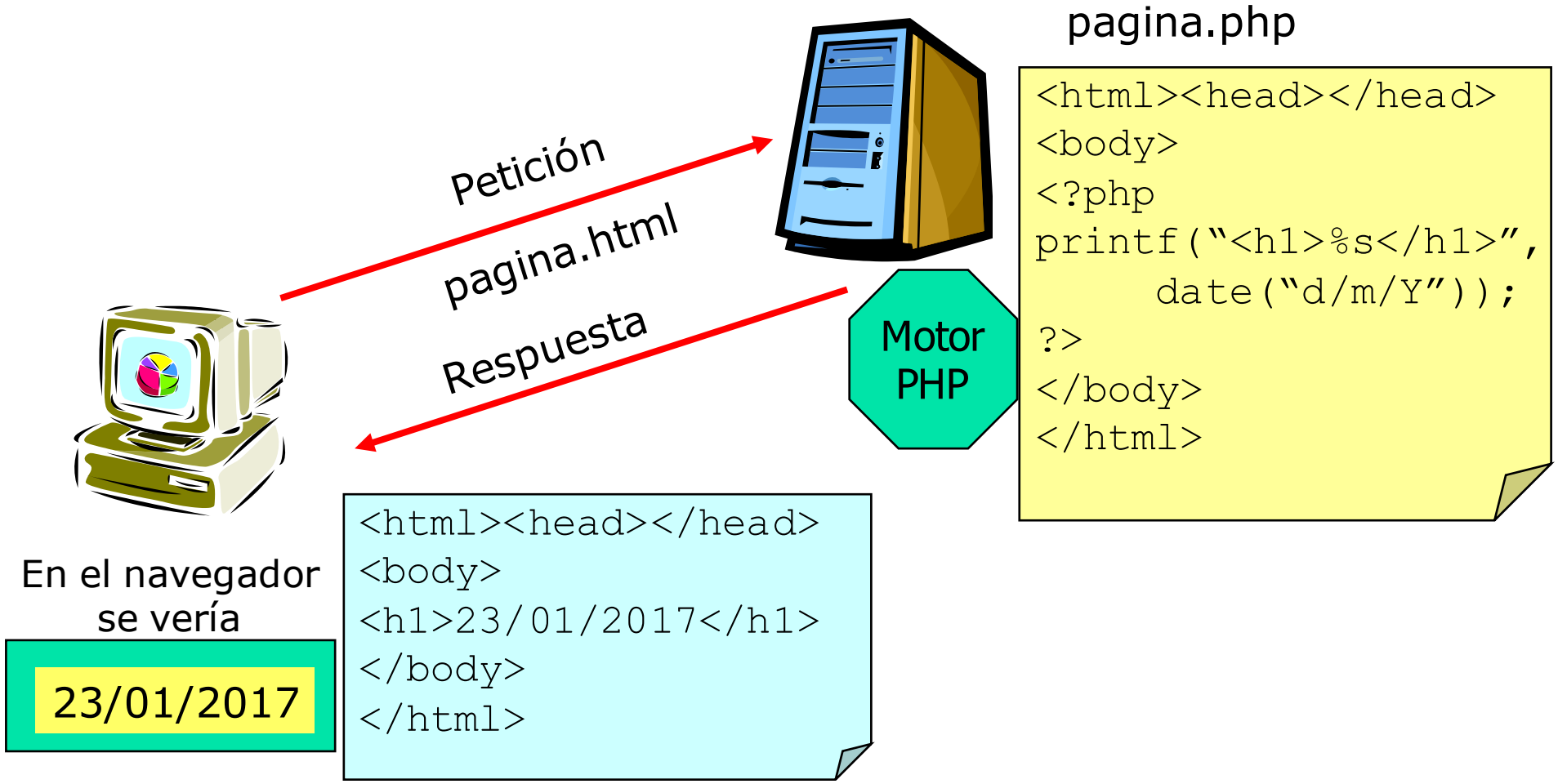
WWW > Servidores web



WWW > Servidores web

- Posteriormente apareció la posibilidad de generar **páginas web dinámicas**
  - Páginas cuyo contenido varía en función de parámetros proporcionados por el usuario (formularios), por el navegador (su tipo), por el paso del tiempo o por programas externos
  - Dichas páginas son creadas habitualmente con la ayuda de lenguajes de programación del lado servidor combinados con algún tipo de tecnología que posibilita su ejecución
    - CGI, ASP, PHP, Perl, JSP, servlets, etc.

WWW > Servidores web



WWW > Mantenimiento de la sesión

- HTTP es un **protocolo sin estado**
  - El servidor no mantiene información o estado sobre cada usuario a lo largo de múltiples peticiones
    - Denominado **sesión** (de usuario)
  - Pero, aplicaciones web típicas como las tiendas virtuales, necesitan conocer el estado de la interacción con el usuario
    - Usuario validado o no, estado del carrito, en qué punto del proceso de compra se encuentra, etc.
- Se necesitan, pues, alternativas software que permitan mantener el estado

WWW -> Mantenimiento de la sesión

- Alternativas
  - Almacenar toda la información de la sesión en una *cookie*
  - Utilizar una combinación de *cookie* (que guardaría un identificador de usuario) y una base de datos
  - La técnica denominada "*URL rewriting*"
    - Todas las URL enviadas al cliente contienen algún tipo de identificador de sesión
  - Utilización de un objeto *Session* (o similar) disponible en los entornos de programación como ASP o JEE (servlets, JSPs)
  - Etc.

WWW > Mantenimiento de la sesión

- *Cookies*
  - Pequeña pieza de información creada por un servidor web y enviada a un navegador, donde es almacenada
    - Como parte de un mensaje de respuesta

      ```
      Set-Cookie: name=value
      ```

    - Así el servidor podrá en el futuro consultar dicha información para reconocer la actividad previa del usuario
  - Si el navegador tiene habilitadas las *cookies*, estas podrán formar parte de mensajes de petición al servidor que previamente las envió

    ```
    Cookie: name=value; name2=value2
    ```

WWW > Mantenimiento de la sesión

- *Cookies*: detalles de implementación
  - Un navegador deberá ser capaz de manejar
    - *Cookies* de hasta 4096 bytes
    - 50 *cookies* por dominio
    - 3000 *cookies* en total
  - Por su parte, los servidores web deberán utilizar el menor número de ellas y del menor tamaño posible
- Ejemplo

```
Set-Cookie: HSID=AYQEVn….DKrdst;
Domain=.foo.com; Path=/; Expires=Wed,
13-Jan-2021 22:23:01 GMT; HttpOnly
```

WWW > Mantenimiento de la sesión

- *URL rewriting*
  - Consiste en incluir la información de estado en el URL

    `/…/comprar.asp?uid=9195&paso=3&producto1=01992CX&producto2=ZZ112230&producto3=HJ19X25…`
  - En aplicaciones importantes no es conveniente su uso
    - Introduce problemas de seguridad
    - Sólo se puede utilizar con peticiones GET
    - Habitualmente sólo se usa esta técnica si el uso de cookies es imposible
      - Desactivadas en el navegador