Tema 1. Introducción

Desarrollo Web en Entorno Servidor

Desarrollo de aplicaciones web

Tecnología cliente / servidor

Concepto

- Procesamiento cooperativo de la información por medio de un conjunto de PCs.
- Los clientes y los servidores pueden estar distribuidos geográficamente.
- Se pueden solicitar servicios a más de un servidor, sin que por ello se vea disminuido el rendimiento.

Arquitectura Distribuida que permite a usuarios finales obtener acceso a la información de forma transparente.

Características del modelo Cliente / Servidor

Servicio

- Es la unidad principal del modelo.
- El servidor los proporciona.
- El cliente los solicita.
- Incluyen cualquier elemento que forme parte de la empresa: impresora, archivo, escáner, DNS, procesamiento, datos, etc.

Recurso compartido

- Se considera recurso compartido todo elemento físico o lógico que pertenece a la lógica de la empresa.
- Ya que muchos clientes los utilizan, los servidores deberán gestionarlos.

Protocolos asimétricos

- Se consideran protocolos asimétricos porque no cualquier nodo de la red puede comenzar una comunicación.
- Será el cliente el que comience la comunicación realizando una petición cualquiera.
- En ese momento se crea un canal bilateral único.

Transparencia de localización

- Ni el cliente sabe dónde está físicamente el servidor ni el servidor sabe de dónde le vienen físicamente las peticiones.
- En caso de que los datos estén repartidos, el cliente tampoco sabrá exactamente dónde se localizan los datos.

Encapsulamiento de servicios

- Los detalles de implementación de un servicio son transparentes al cliente.
- No es necesario saber de qué forma está programado el servidor para interactuar con él.
- No es necesario saber sobre qué plataforma se ejecuta el servidor para trabajar con él.

Funcionamiento de un sistema C/S

Funcionamiento

- El servidor (o demonio por la traducción del inglés *daemon*) se activa y espera solicitudes.
- Tanto los clientes como los servidores son normalmente parte de un programa mayor.
- El proceso es:

1. Cliente solicita recurso a un servidor
2. Servidor recibe la petición
3. Servidor procesa solicitud
4. Servidor envía resultado obtenido
5. Cliente recibe resultado y procesa

Componentes de la arquitectura Cliente Servidor

Componentes

- Es un modelo basado en los servicios.

Componentes

- Según el esquema anterior tenemos dos elementos fundamentales:

1. Proceso cliente.
2. Proceso servidor.

Elementos principales

El cliente

El cliente

- Todo proceso que reclama servicios a otro proceso del sistema.
- Se le conoce como: FRONT-END
- El cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos.
- Suelen estar desarrolladas sobre sistemas que permiten las interfaces gráficas de usuario.

Funciones

- Administrar el interfaz de usuario
- Interactuar con el usuario.
- Procesar la lógica de la aplicación.
- Hacer validaciones locales.
- Generar requerimientos de bases de datos (Consultas)
- Recibir resultados del servidor.
- Dar formato comprensible de los resultados.

Elementos principales

El servidor

El Servidor

- Todo proceso que proporciona un servicio a otros procesos del sistema.
- Debe ser capaz de atender a múltiples clientes que hacen diferentes peticiones.
- Se le conoce como: BACK-END
- Normalmente maneja todas las funciones relacionadas con las reglas de la empresa.

Funciones

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar dichos requerimientos.
- Dar formato a los datos para transmitirlos por la red.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.
  - NOTA: puede darse el caso de que un servidor actúe como cliente de otro.

Tipos de Arquitecturas Cliente/Servidor

Clasificación por tamaño de los componentes

FAT CLIENT

- El peso de la aplicación se ejecuta en el cliente.
- En el cliente están el nivel de presentación y el de aplicación.
- El servidor sólo realiza las funciones de administrador de BD.
- La potencia del sistema es equilibrada: tanto el cliente como el servidor deben ser potentes.

FAT SERVER

- Es el caso opuesto al anterior.
- En el cliente sólo se gestiona el nivel de presentación.
- Todo el peso de la aplicación por tanto recae sobre el servidor.
- En este caso la potencia del sistema **NO** es equilibrada: el servidor deberá ser muy potente mientras que el cliente no necesita grandes cantidades de recursos.

Tipos de Arquitecturas Cliente/Servidor

Clasificación por la naturaleza de los servicios

Servidores de Archivos

- Es una forma muy primitiva de servicio de datos.
- El servidor tiene todos los archivos en su disco duro (o en varios)
- El cliente solicita un archivo que necesita.
- Se necesita intercambio de muchos mensajes sobre la red para encontrar el dato requerido.

Servidores de Bases de Datos

- El servidor debe proveer un acceso compartido a los datos.
- Debe incluir mecanismos:
  - De protección.
  - De concurrencia.
  - De seguridad
  - De consistencia.

Servidores de Bases de Datos

- El servidor no devuelve al cliente todos los datos de la BD, sólo aquellos que son resultados de una o varias consultas.
- Todo ello a través de elementos de SGBD:
  - Procedimientos almacenados: al que el cliente llama y recibe un resultado.
  - Disparadores: que se ejecutarán como respuesta a un evento.
  - Restricciones: orientadas a llevar a cabo validaciones simples de datos.

Servidores de Objetos

- Las aplicaciones son escritas como un conjunto de objetos que se comunican.
- Los objetos cliente se comunican con los objetos servidor por medio del ORB (Object Request Broker)
- Los servidores de objetos deben soportar una gran cantidad e concurrencia.

Servidores WEB

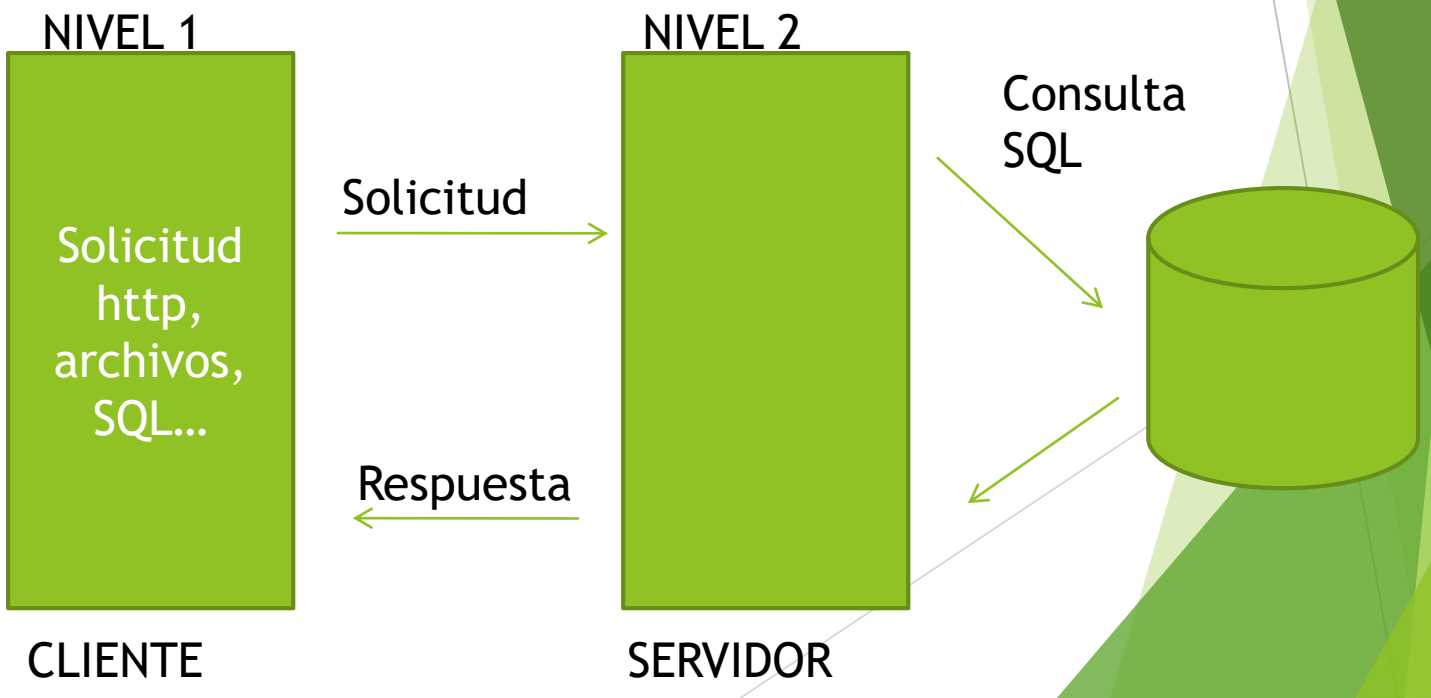
- La mayor aplicación Cliente / Servidor que existe cubre todo el planeta: WWW.
- El sistema son clientes simples que hablan con servidores Web.
- El servidor Web devuelve documentos que solicita el cliente.
- El cliente interpreta dichos documentos.
- El protocolo usado en este sistema es HTTP

Tipos de Arquitecturas Cliente/Servidor

Clasificación por capas de software

Modelo C/S en 2 capas

- Son sistemas en los que el cliente solicita recursos y el servidor responde directamente.
- Existe conexión directa entre el cliente y el administrador de la base de datos.

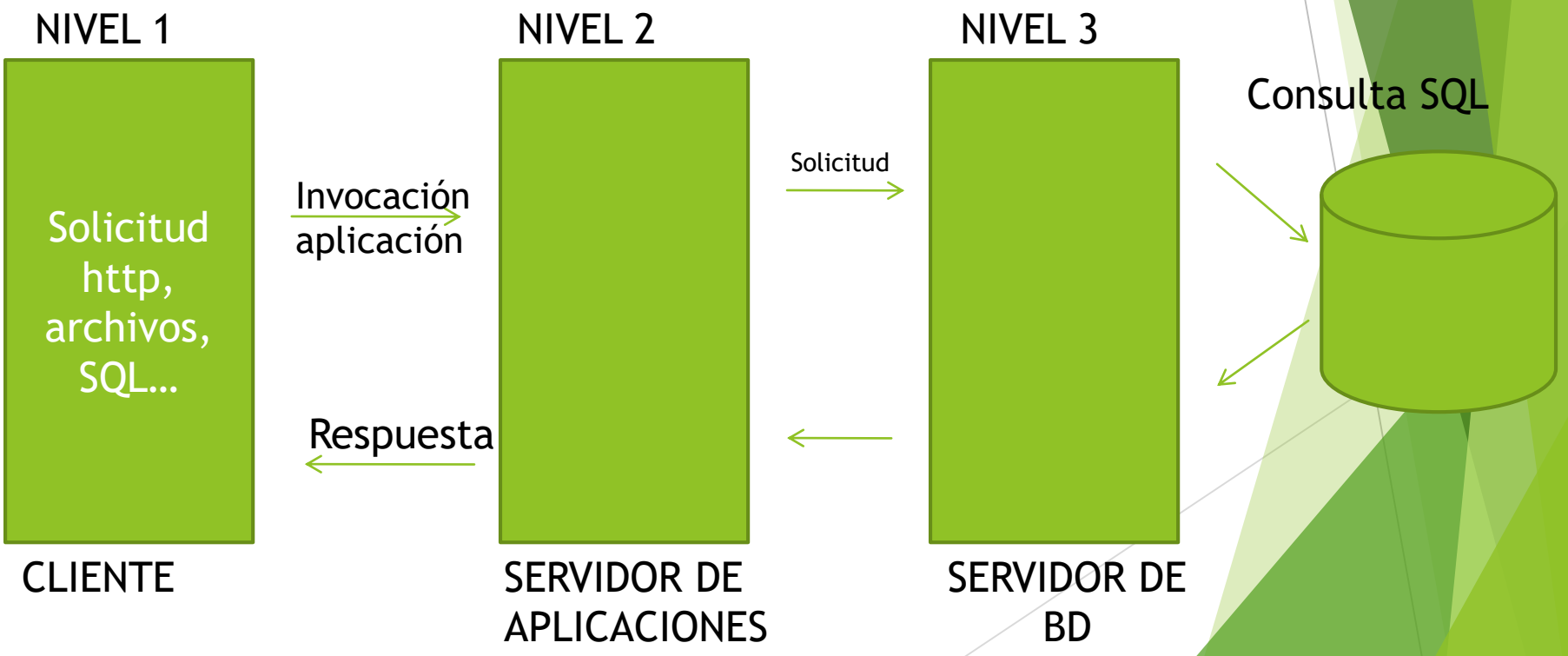


Modelo C/S en 2 capas

- Implementado con SQL remoto
  - El cliente envía la solicitud SQL al servidor.
  - El servidor devuelve el resultado de cada instrucción SQL recibida.
  - Ventajas:
    - Simplicidad.
  - Inconvenientes
    - Mucha información viajando por la red
    - Bajo rendimiento.

Modelo C/S en 3 capas

- Añade al nivel en 2 capas un nivel intermedio.
- El nivel intermedio: servidor de aplicaciones.



Modelo C/S en 3 capas

- Ventajas:
  - Reduce tráfico de información.
  - Mayor flexibilidad de desarrollo + escalabilidad.
  - Mantiene independencia entre aplicaciones y datos.
  - Usa lenguajes estándares.
  - Menos clientes conectados a la base de datos.
- Inconvenientes:
  - Puede presentar mayor complejidad de programación.
  - Existen pocos proveedores y de alto costo.

Modelo C/S múltiples niveles

- Entre el cliente y la Base de datos se colocan tantos servidores como se consideren necesarios.
- Cada servidor realiza una tarea especializada (servicio)
- Cada servidor puede utilizar los servicios del resto de servidores para completar el suyo propio.

Lenguajes de programación Web

Web estática
VS
Web dinámica

Web Estática

- Son sitios que se constituyen en HTML.
- No permiten grandes florituras.
- La mayor funcionalidad que incluyen son los enlaces.
- Sencillas de crear.
- Sólo presentan textos planos con imágenes.
- Máximo contenidos multimedia: sonido y vídeo.

Web Dinámica (Interactiva)

- Una página es dinámica si incluye cualquier efecto especial o funcionalidad.
- Se puede interactuar con ellas.
- El contenido cambiará dependiendo de la interacción con el usuario.
- Es necesario utilizar lenguajes de programación para gestionar la interacción con el usuario.

Programación en Entorno Cliente y en Entorno Servidor

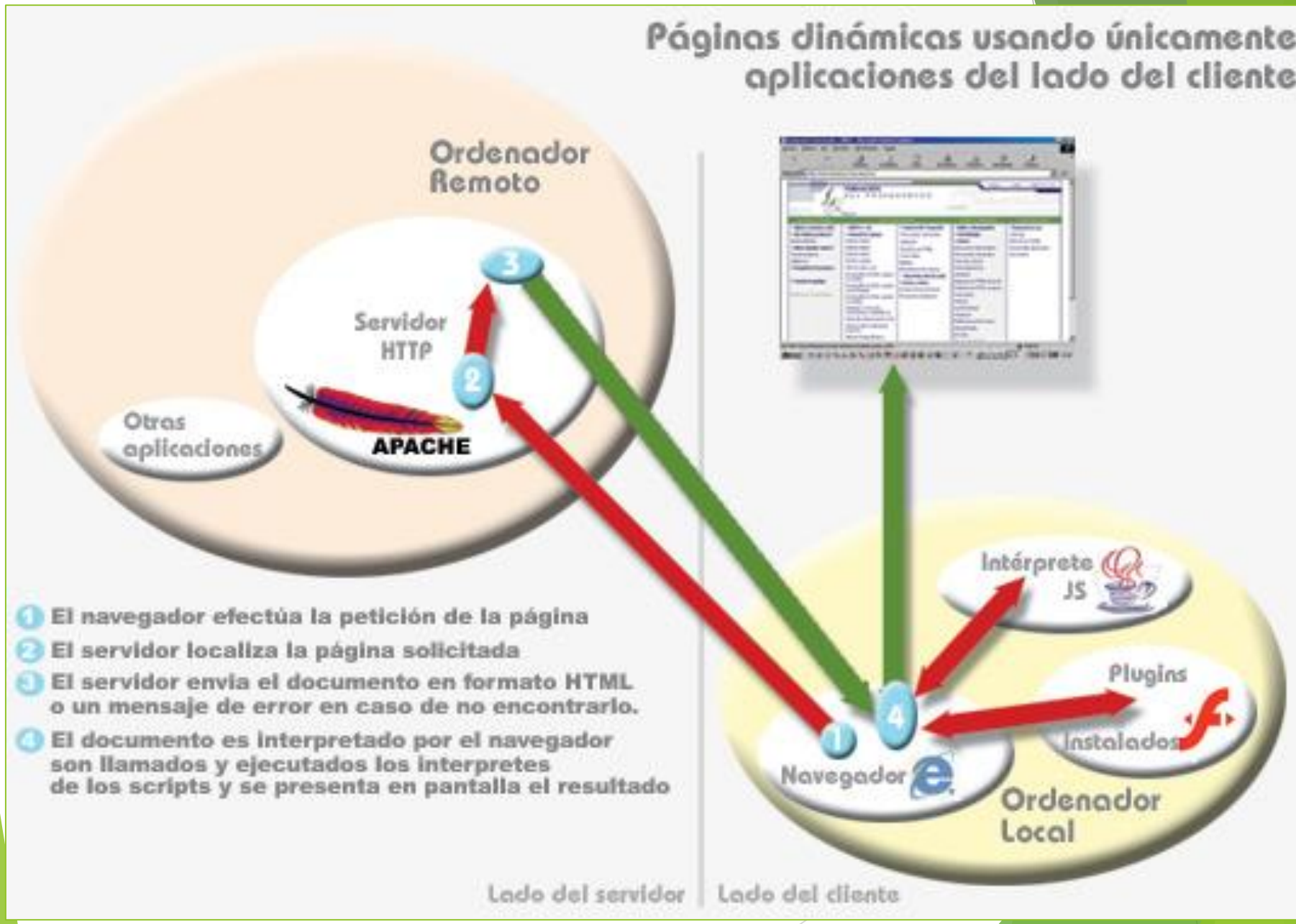
El navegador Web

- El navegador sólo interpreta ordenes escritas en código HTML.
- Convierte las instrucciones HTML en páginas web.
- Documento HTML = Página Web
- Clic sobre un enlace = petición de otro documento HTML

Lenguajes del lado del Cliente

- Son lenguajes que son interpretados directamente por el cliente.
- El servidor Web envía los documentos al cliente sin realizar ningún tipo de transformación o pretratamiento.
- Entre ellos se encuentran:
  - HTML
  - Java
  - JavaScript

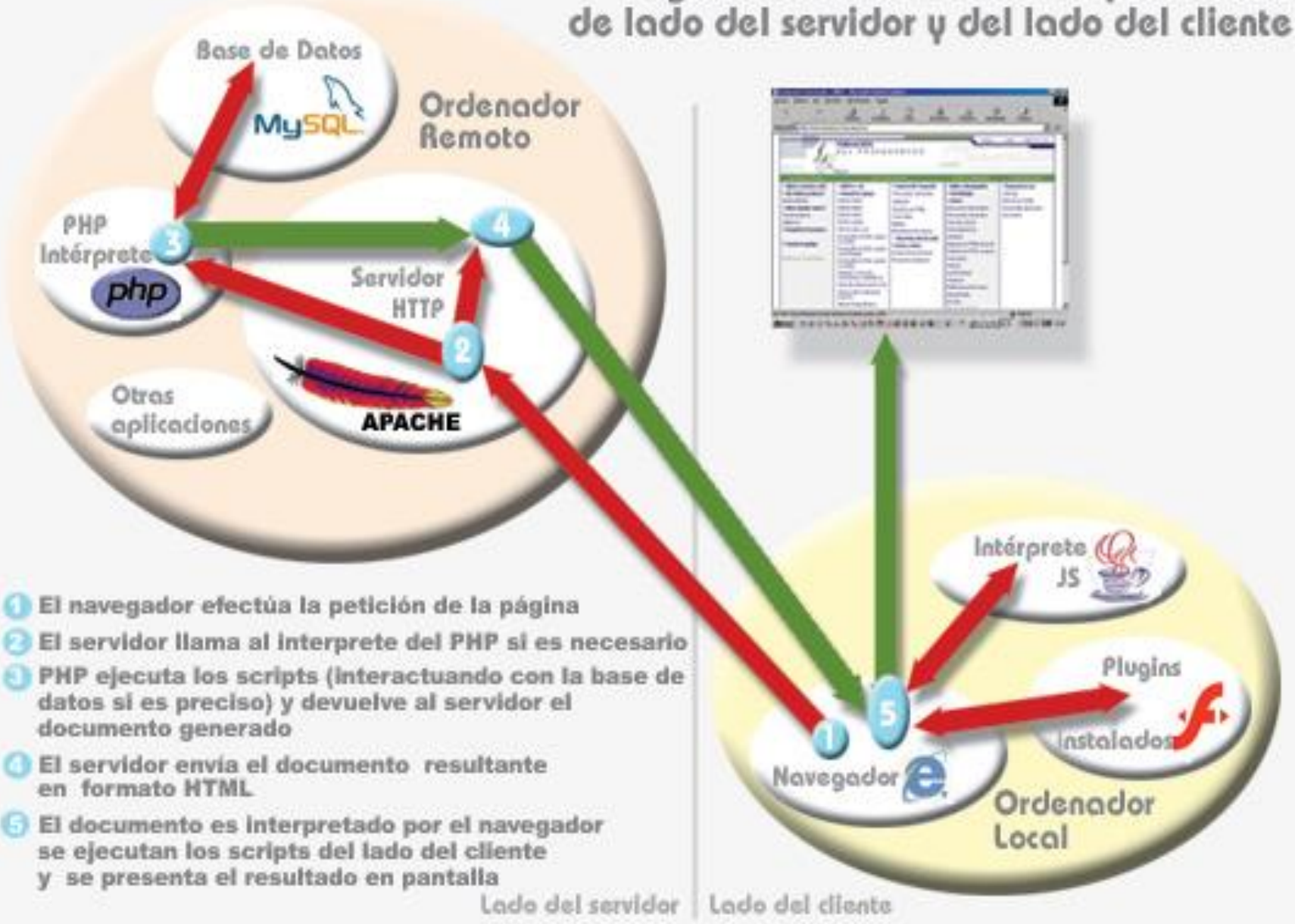
Páginas dinámicas usando únicamente aplicaciones del lado del cliente



Lenguajes del lado del servidor

- Son reconocidos, ejecutados e interpretados por el propio servidor.
- El servidor transforma el código escrito en un lenguaje de programación en sentencias HTML.
- El servidor envía al cliente un documento HTML comprensible para él.
- Incluyen:
  - PHP
  - ASP
  - Otros

Páginas dinámicas usando aplicaciones de lado del servidor y del lado del cliente



Introducción

Desarrollo Web en Entorno Servidor

Desarrollo de aplicaciones web