

# *APLICACIONES WEB*

## **TEMA 1:** *INTRODUCCIÓN A LAS APLICACIONES WEB*



### **Objetivos**

- ✓ Hacer un repaso de las Redes de Computadoras e Internet
- ✓ Introducir el concepto de Aplicación Web
- ✓ Mostrar las diferentes arquitecturas para la implementación de una AW

# Índice: Tema 1

## **1.1 Internet**

1.1.1 Redes de computadoras

1.1.2 El modelo de capas TCP/IP

## **1.2 Aplicaciones web**

1.2.1 Páginas web dinámicas

1.2.2 Aplicaciones cliente-servidor

## **1.3 Arquitectura de una aplicación web**

1.3.1 Arquitecturas cliente-servidor

1.3.2 Modelos de distribución de una aplicación web

# Índice: Tema 1

## **1.1 Internet**

### **1.1.1 Redes de computadoras**

#### 1.1.2 El modelo de capas TCP/IP

## **1.2 Aplicaciones web**

### 1.2.1 Páginas web dinámicas

### 1.2.2 Aplicaciones cliente-servidor

## **1.3 Arquitectura de una aplicación web**

### 1.3.1 Arquitecturas cliente-servidor

### 1.3.2 Modelos de distribución de una aplicación web

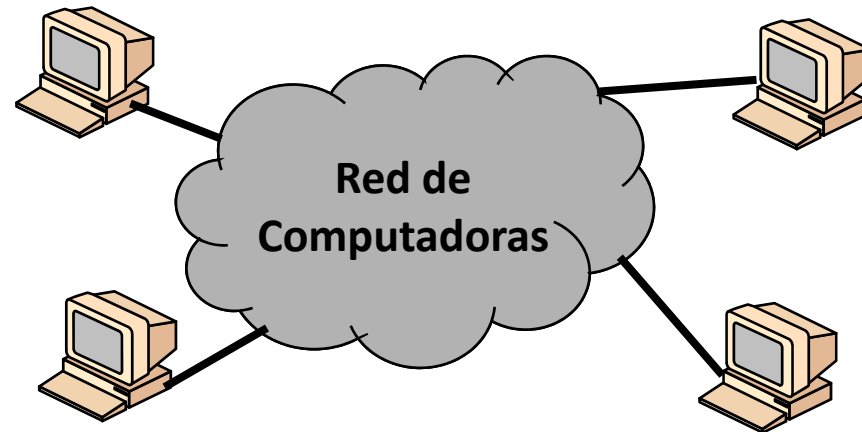


## 1.1 INTERNET

### 1.1.1 REDES DE COMPUTADORAS

#### ➤ Redes de computadoras

- ✓ Conjunto de Computadoras (**Hosts**) interconectadas entre sí.
  
- ✓ Elementos:
  - Hosts (Clientes/Servidores)
  - Hubs
  - Switches
  - Líneas de Transmisión
  
- ✓ Las computadoras pueden estar en una misma sala (**LAN**) o en cualquier parte del mundo (**Internet**).





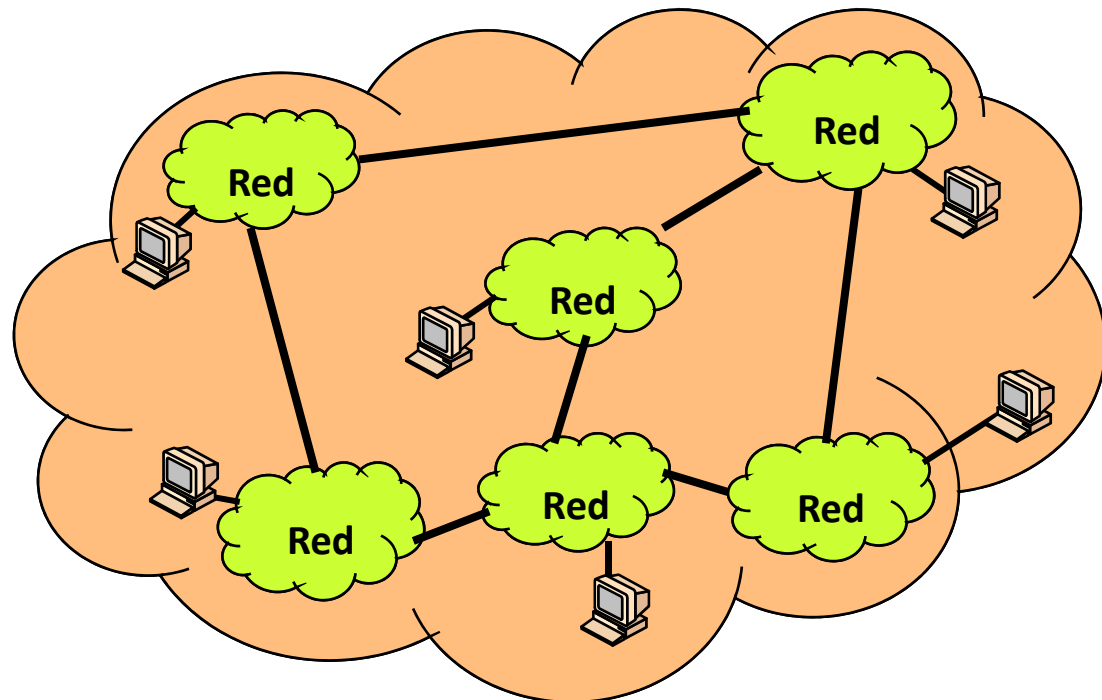
## 1.1 INTERNET

### 1.1.1 REDES DE COMPUTADORAS

➤ **Internet = Interconnected Networks**

✓ Elementos:

- Hosts (Clientes/Servidores)
- LAN's
- Redes Públicas
- Routers
- Líneas de Transmisión





## 1.1 INTERNET

### 1.1.1 REDES DE COMPUTADORAS



#### ➤ Direccionamiento IP

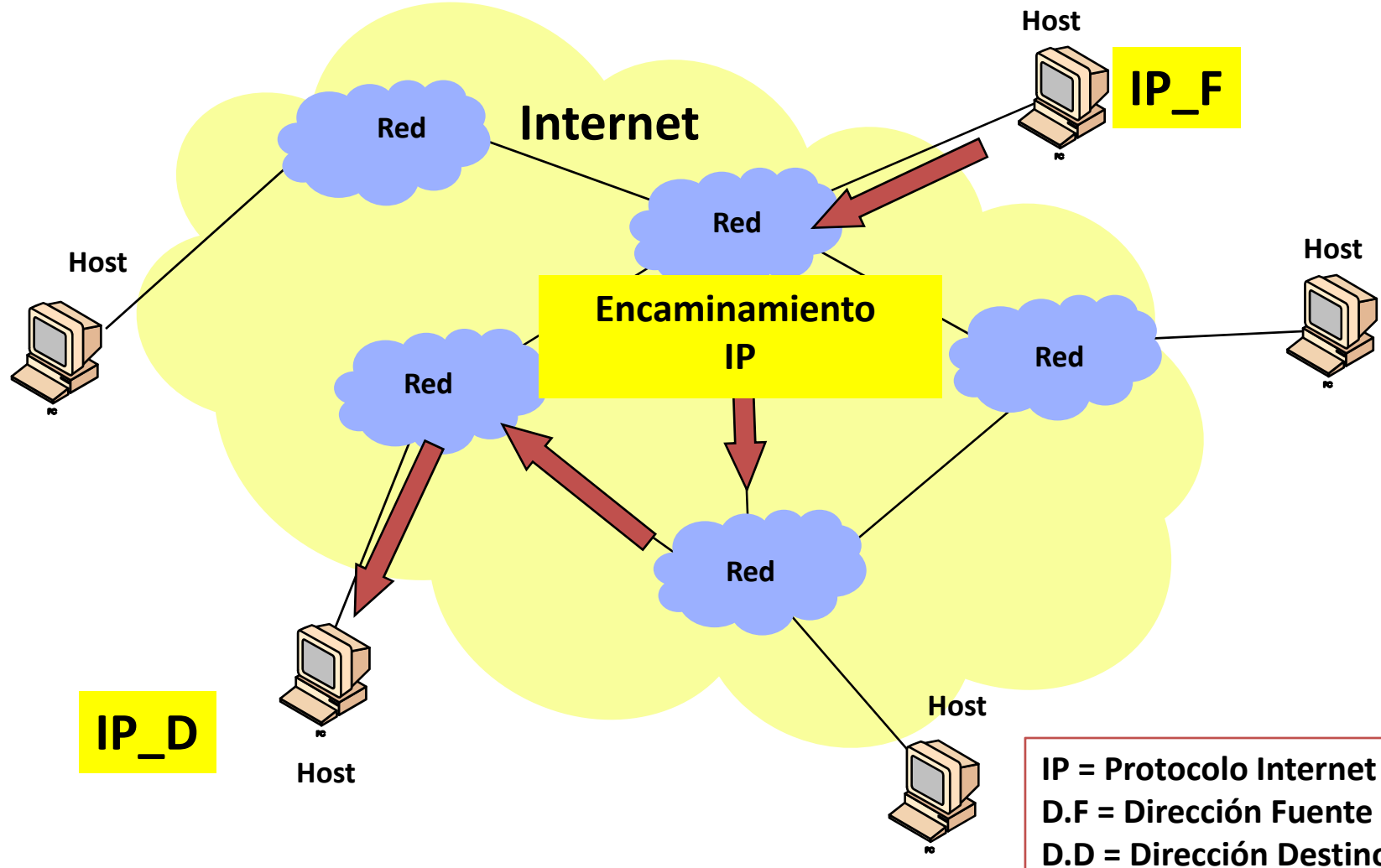
- ✓ Por tanto, Internet consiste en un conjunto de redes interconectadas entre si que conectan, a su vez, varios tipos de computadoras.
- ✓ Dichas computadoras se clasifican en clientes o servidores (**Host**) y encaminadores (**Router**).
- ✓ Los **servidores** están permanentemente conectados ofreciendo diferentes servicios que son habitualmente consultados por los **clientes**.
- ✓ Cada Host tiene una “única” **Dirección IP**.
- ✓ Para comunicarse entre si, los Hosts se intercambian/envían **Paquetes IP**.
- ✓ Cada paquete contiene una **Dirección de la Fuente (IP\_F)**, y una **Dirección del Destino (IP\_D)**.
- ✓ Los Routers permiten encaminar los paquetes a través de la red de forma que lleguen a su destino.



## 1.1 INTERNET

### 1.1.1 REDES DE COMPUTADORAS

#### ➤ Encaminamiento IP



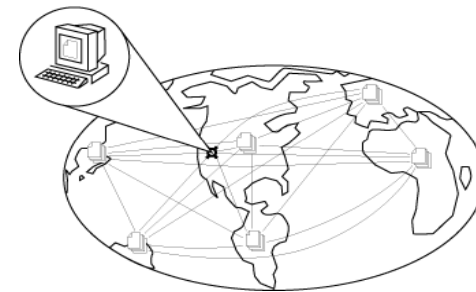


## 1.1 INTERNET

### 1.1.1 REDES DE COMPUTADORAS

#### ➤ Pero, ¿qué es lo que realmente se conoce como Internet?

- ✓ World Wide Web es un almacén arquitectónico que permite acceder a documentos distribuidos por todos los equipos conectados a Internet.
- ✓ Las WWW es uno de los múltiples servicios que nos ofrece la interconexión de múltiples redes de ordenadores, otros son:
  - Transferencia de ficheros
  - Correo electrónico
  - Acceso remoto, etc...
- ✓ Sin embargo, las WWW es la aplicación más exitosa que funciona en Internet, hasta el punto que se ha apropiado de dicho nombre.
- ✓ Se basa en la transferencia de “hipertexto” entre el cliente (el que solicita el documento) y el servidor (el que lo sirve) mediante el protocolo HTTP.







## 1.1 INTERNET

### 1.1.1 REDES DE COMPUTADORAS



#### ➤ Historia de las WWW

- ✓ Las WWW fue creada en 1989 en el CERN (Centro Europeo de Investigación Nuclear):
  - La propuesta inicial partió de Tim Berners-Lee.
  - El propósito original es que los científicos que investigaban en física de partículas en diferentes lugares pudieran intercambiar información de manera rápida.
- ✓ En 1991 ya había un prototipo en funcionamiento (tan sólo basado en texto).
- ✓ Su impactó llevo al desarrollo del primer navegador gráfico (Mosaic).
- ✓ A continuación se produjo la “guerra de los navegadores” entre los dos principales navegadores: Internet Explorer de Microsoft y Netscape.
- ✓ En 1994 el CERN y el MIT crearon el World Wide Web Consortium (W3C):
  - Dedicado a estandarizar protocolos y lenguajes y cuidar la interoperabilidad.

# Índice: Tema 1

## **1.1 Internet**

1.1.1 Redes de computadoras

**1.1.2 El modelo de capas TCP/IP**

## **1.2 Aplicaciones web**

1.2.1 Páginas web dinámicas

1.2.2 Aplicaciones cliente-servidor

## **1.3 Arquitectura de una aplicación web**

1.3.1 Arquitecturas cliente-servidor

1.3.2 Modelos de distribución de una aplicación web



## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP



#### ➤ Arquitectura de red

- ✓ Las tareas que debe resolver una red de computadores para que la comunicación pueda llevarse a cabo son muy numerosas y diversas:
  - Intentar resolver todas “de una vez” sería prácticamente inabarcable.
  - Solución típica de la informática → “divide y vencerás”.
- ✓ La idea es dividir la arquitectura en varias **capas**:
  - Cada una se ocupa de un grupo de tareas.
  - Tareas de mayor abstracción cuanto más alta sea la capa.
- ✓ La comunicación entre capas se lleva a cabo de acuerdo a diferentes **protocolos**:
  - Conjunto de reglas o convenios que gobiernan el intercambio de información entre entidades situadas en sistemas diferentes.

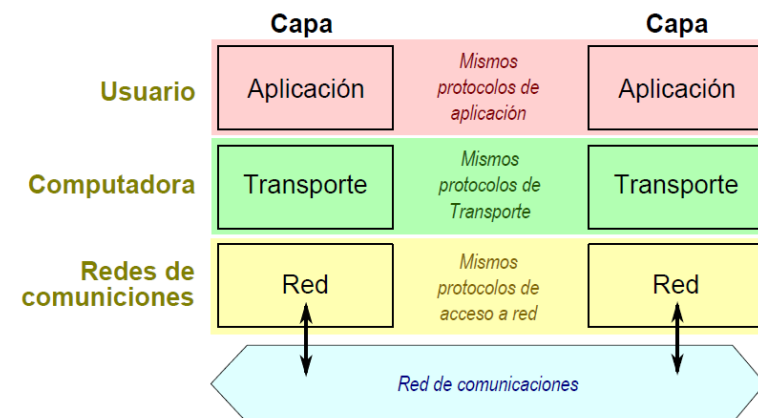


## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP

#### ➤ Comunicación entre capas

- ✓ **Usuario** que desea realizar la comunicación usando alguna aplicación.
  - **Ejemplo:** Transferir un fichero por FTP (usando Filezilla por ejemplo).
- ✓ **Computadora** en la que se encuentra instalada esa aplicación.
  - **Ejemplo:** El origen y destino deben estar preparados para la transmisión y recepción, garantizando seguridad y fiabilidad.
- ✓ **Red** a la que esta conectada la computadora.
  - **Ejemplo:** El origen debe activar un camino de salida e identificar al destino.
- ✓ En cada nivel hay varias tareas, organizadas en capas.
  - Cada capa se ocupa de una tarea.
  - Cada capa ofrece servicios a la capa superior





## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP



#### ➤ Capas del modelo TCP/IP

##### APLICACIÓN

Gestiona los detalles de cada **aplicación**.

##### TRANSPORTE

Proporciona al nivel de Aplicación un flujo de datos **sin errores entre máquinas**.  
Realiza tareas de control de flujo y gestión de errores.

##### RED

Se ocupa del **encaminamiento** de los paquetes por la red.

##### ACCESO A RED

La **capa de enlace** convierte el medio físico en una línea de transmisión sin errores.

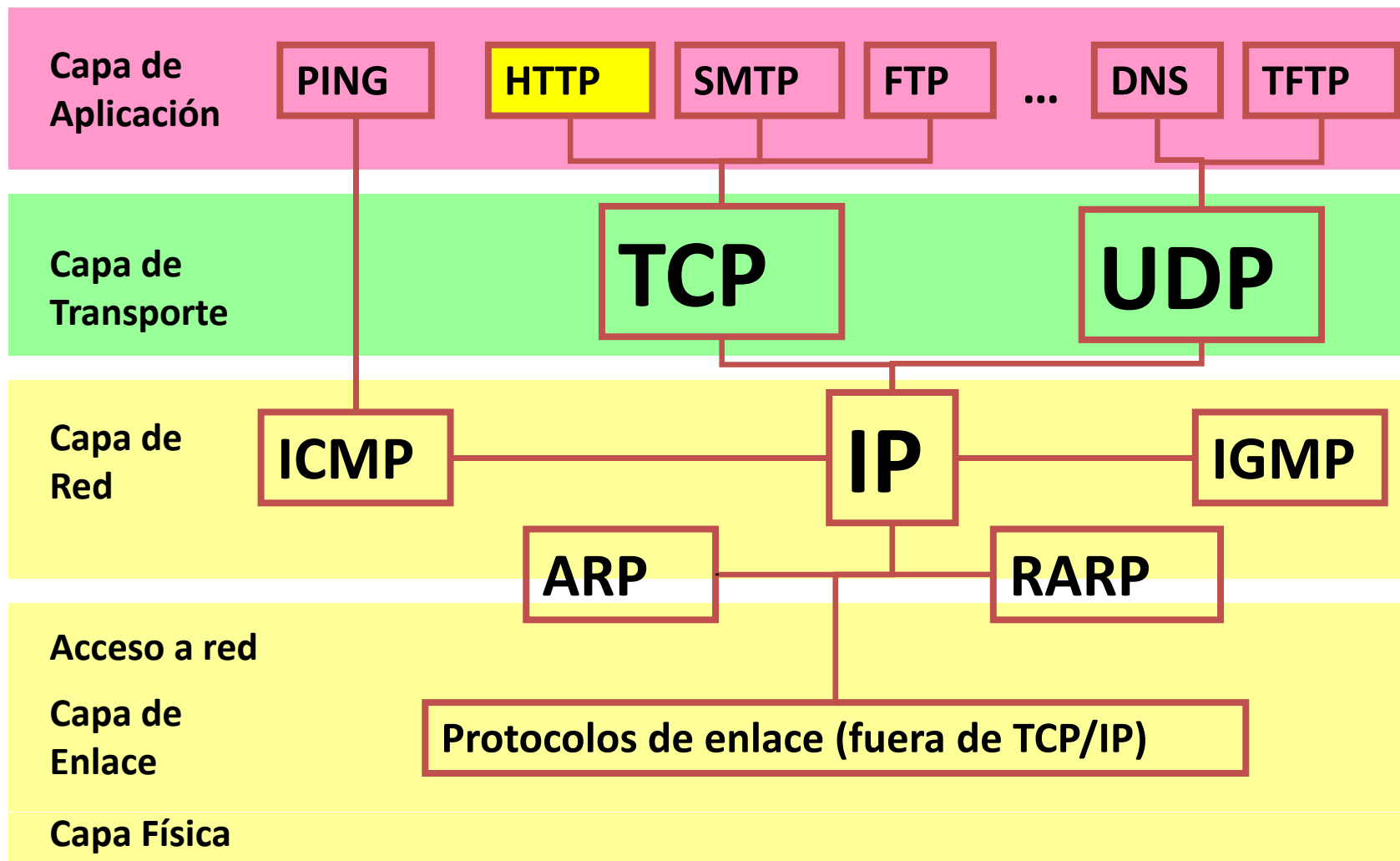
La **capa física** transmite/recibe una secuencia de bits por un canal de comunicación.



## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP

#### ➤ Protocolos de la familia TCP/IP asociados a cada capa





## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP



#### ➤ El protocolo HTTP

- ✓ HTTP es el protocolo a nivel de aplicación que usan las aplicaciones web para comunicar cliente y servidor.
- ✓ Funciona sobre el protocolo TCP/IP y define:
  - Las peticiones del cliente al servidor.
  - Las respuestas del servidor a las peticiones del cliente.
- ✓ Únicamente contempla petición y respuesta, no almacena estado:
  - Si queremos mantener una sesión, es necesario simularla.
- ✓ Las peticiones siempre las realiza el cliente:
  - El servidor no puede enviar información si no es como respuesta a una petición.
- ✓ HTTPS es la versión segura del protocolo HTTP:
  - Cifra el contenido de los mensajes mediante SSL.



## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP



#### ➤ Transferencia web (1)

1. El navegador determina la URL (seleccionada por el usuario).
2. El navegador solicita al DNS la dirección IP del host de destino contenido en la URL (*www.servidor.com*).
3. El DNS contesta con la dirección IP (198.64.191.11).
4. El navegador establece una conexión TCP al puerto 80 usando esa dirección IP.
5. El servidor está escuchando por ese puerto y reconoce la petición solicitada.
6. Se envía una solicitud para el envío del fichero (*directorio/documento.html*).
7. El servidor envía el fichero al cliente.
8. Se libera la conexión TCP.
9. El navegador interpreta el HTML y representa en la pantalla la página:
  - Si el recurso fuese de otro tipo el navegador necesitaría saber cómo presentarlo. A menudo los navegadores disponen de plug-ins que se encargan de presentar este tipo de recursos.

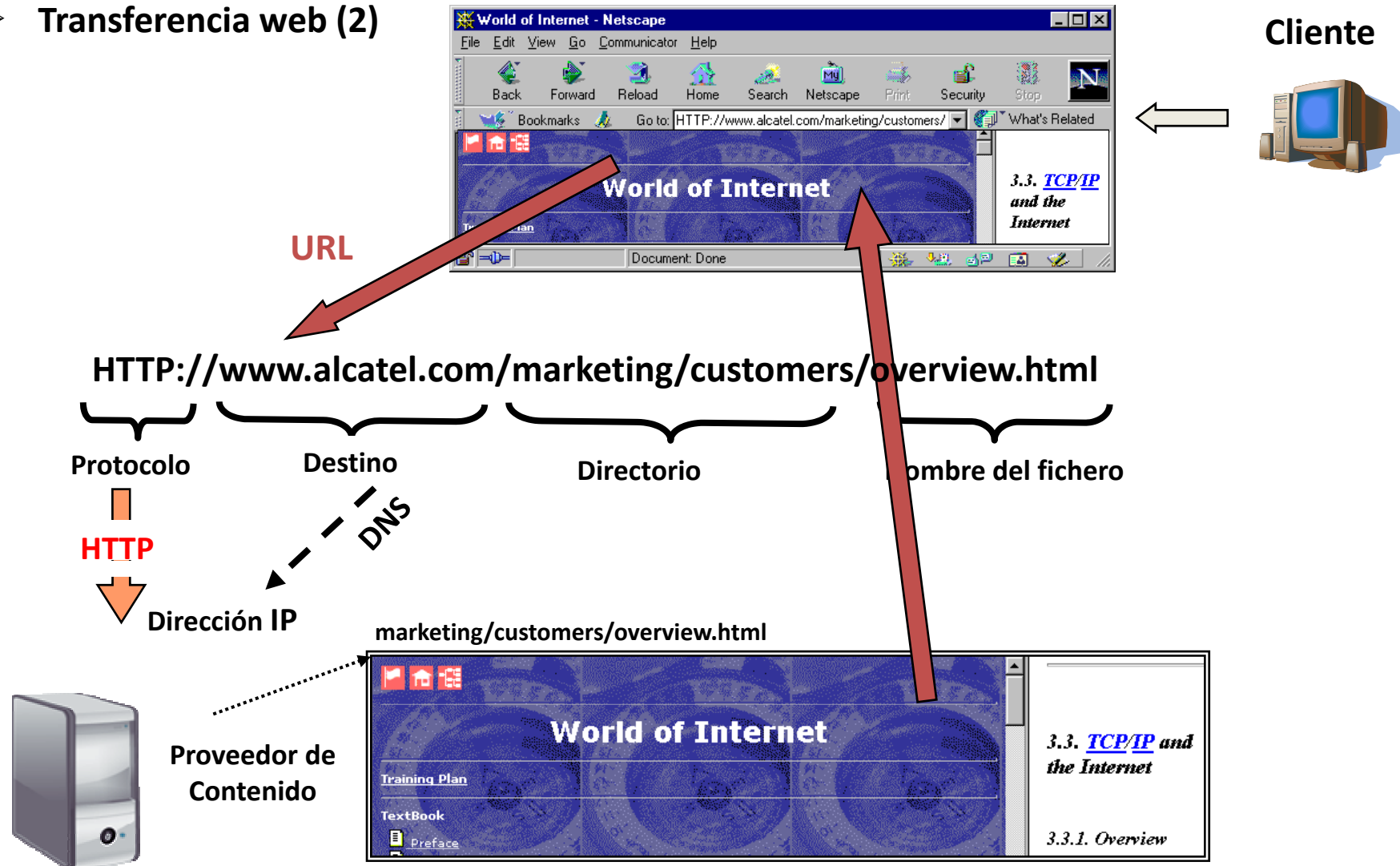




## 1.1 INTERNET

### 1.1.2 EL MODELO DE CAPAS TCP/IP

#### ➤ Transferencia web (2)



# Índice: Tema 1

## **1.1 Internet**

1.1.1 Redes de computadoras

1.1.2 El modelo de capas TCP/IP

## **1.2 Aplicaciones web**

**1.2.1 Páginas web dinámicas**

1.2.2 Aplicaciones cliente-servidor

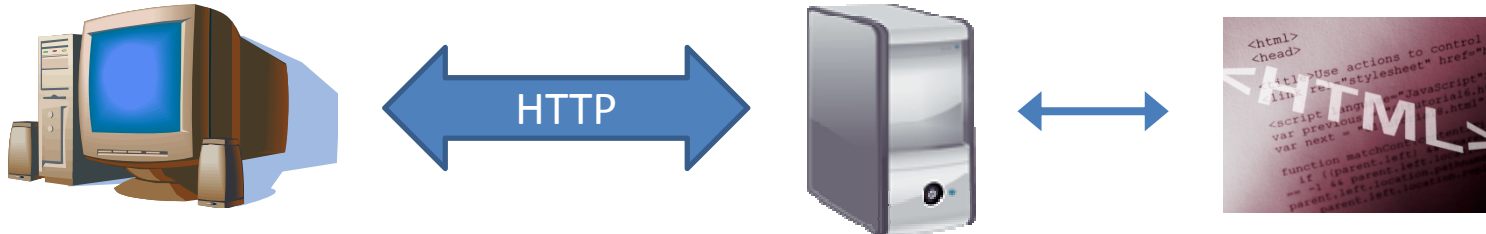
## **1.3 Arquitectura de una aplicación web**

1.3.1 Arquitecturas cliente-servidor

1.3.2 Modelos de distribución de una aplicación web

#### ➤ Modelo estático de web

- ✓ El servidor proporciona documentos estáticos al cliente, que los interpreta mediante un navegador:



- ✓ Los documentos suelen ser páginas HTML, aunque también puede haber otro tipo de recursos estáticos como imágenes, vídeos pdfs, etc...
- ✓ Tanto cliente como servidor realizan muy poco procesamiento (se habla, en este caso, de *thin client* o *thin server*).
- ✓ No ofrece demasiadas posibilidades más allá de la presentación.



## 1.2 APLICACIONES WEB

### 1.2.1 PÁGINAS WEB DINÁMICAS



#### ➤ Aplicaciones web

- ✓ Para conseguir mayor funcionalidad, se ha avanzado ampliando las capacidades tanto del cliente como del servidor.
- ✓ Así, las **Aplicaciones Web** son aplicaciones cliente-servidor accesibles en la red a través de un navegador web.
- ✓ También se definen como un conjunto de recursos en el servidor que permiten crear una aplicación interactiva.
- ✓ Los recursos pueden ser:
  - Documentos estáticos.
  - Recursos almacenados en el servidor que se ejecutan en el cliente.
  - Recursos que se ejecutan en el servidor.
  - Datos.



## 1.2 APLICACIONES WEB

### 1.2.1 PÁGINAS WEB DINÁMICAS



#### ➤ **Ventajas y desventajas**

- ✓ Las Aplicaciones Web ofrecen algunas ventajas sobre una aplicación convencional:
  - Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
  - Multiplataforma, sin problemas de compatibilidad y actualizaciones inmediatas.
  - No ocupan espacio en nuestro disco duro y el consumo de recursos es bajo.
  - Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
  - Permite la colaboración en línea.
- ✓ También tienen algunas desventajas:
  - Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio (aunque existen las aplicaciones web enriquecidas).
  - Aunque su disponibilidad sea alta, siempre depende de un tercero.

# Índice: Tema 1

## **1.1 Internet**

1.1.1 Redes de computadoras

1.1.2 El modelo de capas TCP/IP

## **1.2 Aplicaciones web**

1.2.1 Páginas web dinámicas

**1.2.2 Aplicaciones cliente-servidor**

## **1.3 Arquitectura de una aplicación web**

1.3.1 Arquitecturas cliente-servidor

1.3.2 Modelos de distribución de una aplicación web



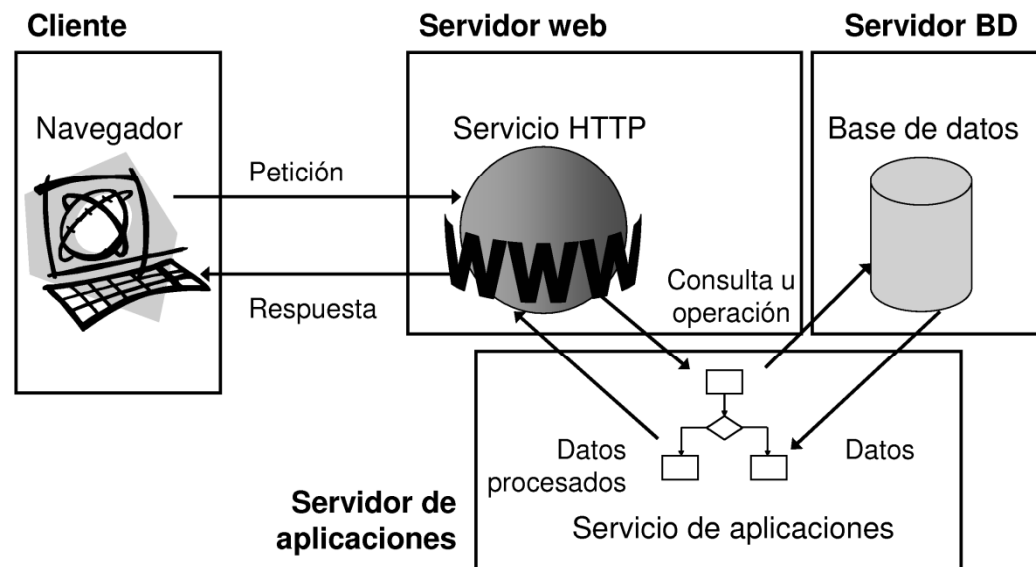
## 1.2 APLICACIONES WEB

### 1.2.2 APLICACIONES CLIENTE-SERVIDOR



#### ➤ Aplicaciones cliente-servidor

- ✓ El procesamiento está distribuido entre el cliente y el servidor.
- ✓ El cliente realiza peticiones a las que el servidor da respuesta.
- ✓ Según la cantidad de proceso que realice cada uno, se suele hablar *de thin/fat client y thin/fat server*.





#### ➤ Ejecución en el cliente

- ✓ El **cliente**, además de interpretar o mostrar documentos HTML y de otros tipos, es capaz de ejecutar código como:
  - Applets: Programas de Java que se ejecutan en el navegador.
  - Scripts: Lenguajes de secuencias de órdenes que interpreta el navegador (como JavaScript, VBScript, etc...).
  - Distintas tecnologías que necesitan de la existencia de un *plug-in* instalado en el en el navegador (Adobe Acrobat Reader, Macromedia Flash, etc...).
- ✓ El servidor devuelve al cliente tanto el contenido estático, como el código a ejecutar.
- ✓ Por tanto, dicho código es visible desde el cliente.





#### ➤ Ejecución en el servidor (1)

- ✓ El **servidor** no se limita a devolver un documento estático que tiene almacenado, sino que es capaz de ejecutar un programa y devolver al cliente una respuesta generada en ese momento por ese programa:
  - Páginas web dinámicas.
- ✓ La respuesta puede depender de varios factores:
  - Parámetros enviados por el cliente.
  - Datos almacenados por la aplicación en el servidor.
  - Interacción previa de ese cliente (sesión).
- ✓ El código de la aplicación no es visible para el cliente, con lo que no se pierde el control de la aplicación (frente a los scripts).



#### ➤ Ejecución en el servidor (2)

- ✓ Existen varias tecnologías que permiten generar páginas dinámicas
  - Interfaz de pasarela común (CGI, Common Gateway Interface).
  - **PHP** (Lenguaje script para procesar peticiones HTTP y generar respuestas HTML).
  - ASP (Solución propietaria de Microsoft, basada en Visual Basic).
  - Java EE (Extensión de Java SE mediante los componentes Servlets, JSP, EJB, etc...).
  - Plataforma .NET (Plataforma de Microsoft, bastante similar a Java en concepto).
  - Etc...

# Índice: Tema 1

## **1.1 Internet**

1.1.1 Redes de computadoras

1.1.2 El modelo de capas TCP/IP

## **1.2 Aplicaciones web**

1.2.1 Páginas web dinámicas

1.2.2 Aplicaciones cliente-servidor

## **1.3 Arquitectura de una aplicación web**

**1.3.1 Arquitecturas cliente-servidor**

1.3.2 Modelos de distribución de una aplicación web



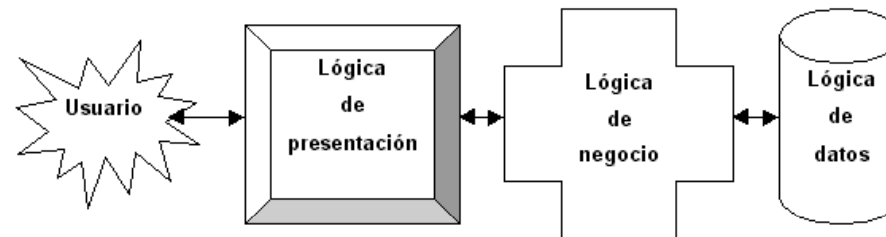
## 1.3 ARQUITECTURA DE UNA APLICACIÓN WEB

### 1.3.1 ARQUITECTURAS CLIENTE-SERVIDOR



#### ➤ Separación de funciones (1)

- ✓ La arquitectura cliente-servidor permite la separación de funciones en tres niveles, similares a los tres niveles del MVC de cualquier aplicación:



- ✓ **Lógica de presentación:** Se encarga de la entrada/salida de la aplicación o interfaz de usuario. Sus principales tareas son:
  - Obtener información del usuario.
  - Enviar la información del usuario a la lógica de negocio para su procesamiento.
  - Recibir los resultados del procesamiento de la lógica de negocio.
  - Presentar estos resultados al usuario.



## 1.3 ARQUITECTURA DE UNA APLICACIÓN WEB

### 1.3.1 ARQUITECTURAS CLIENTE-SERVIDOR



#### ➤ Separación de funciones (2)

- ✓ **Lógica de negocio (o aplicación):** Se encarga de gestionar los datos a nivel de procesamiento. Actúa de puente entre el usuario y los datos. Sus principales tareas son:
  - Recibir la entrada del nivel de presentación.
  - Interactuar con la lógica de datos para ejecutar las reglas de negocio que tiene que cumplir la aplicación.
  - Enviar el resultado del procesamiento al nivel de presentación.
- ✓ **Lógica de datos:** Se encarga de gestionar los datos a nivel de almacenamiento. Sus principales tareas son:
  - Almacenar y recuperar los datos.
  - Mantener los datos y asegurar la integridad de los mismos.

# Índice: Tema 1

## **1.1 Internet**

1.1.1 Redes de computadoras

1.1.2 El modelo de capas TCP/IP

## **1.2 Aplicaciones web**

1.2.1 Páginas web dinámicas

1.2.2 Aplicaciones cliente-servidor

## **1.3 Arquitectura de una aplicación web**

1.3.1 Arquitecturas cliente-servidor

**1.3.2 Modelos de distribución de una aplicación web**



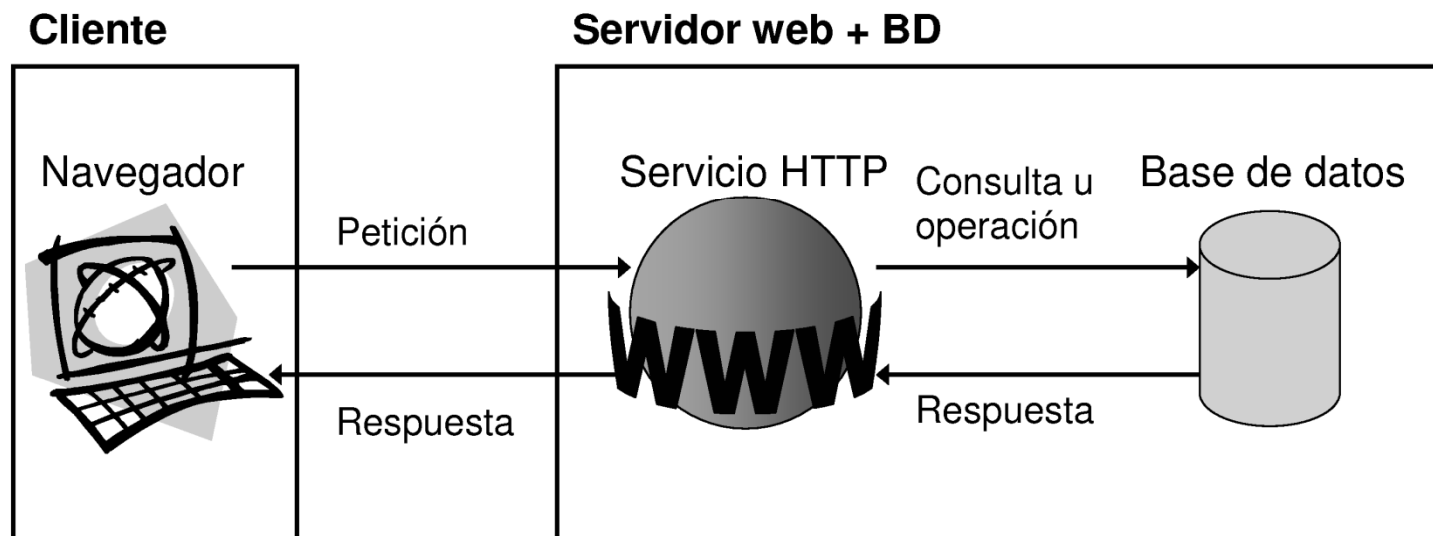
## 1.3 ARQUITECTURA DE UNA APLICACIÓN WEB

### 1.3.2 MODELOS DE DISTRIBUCIÓN DE UNA APLICACIÓN WEB



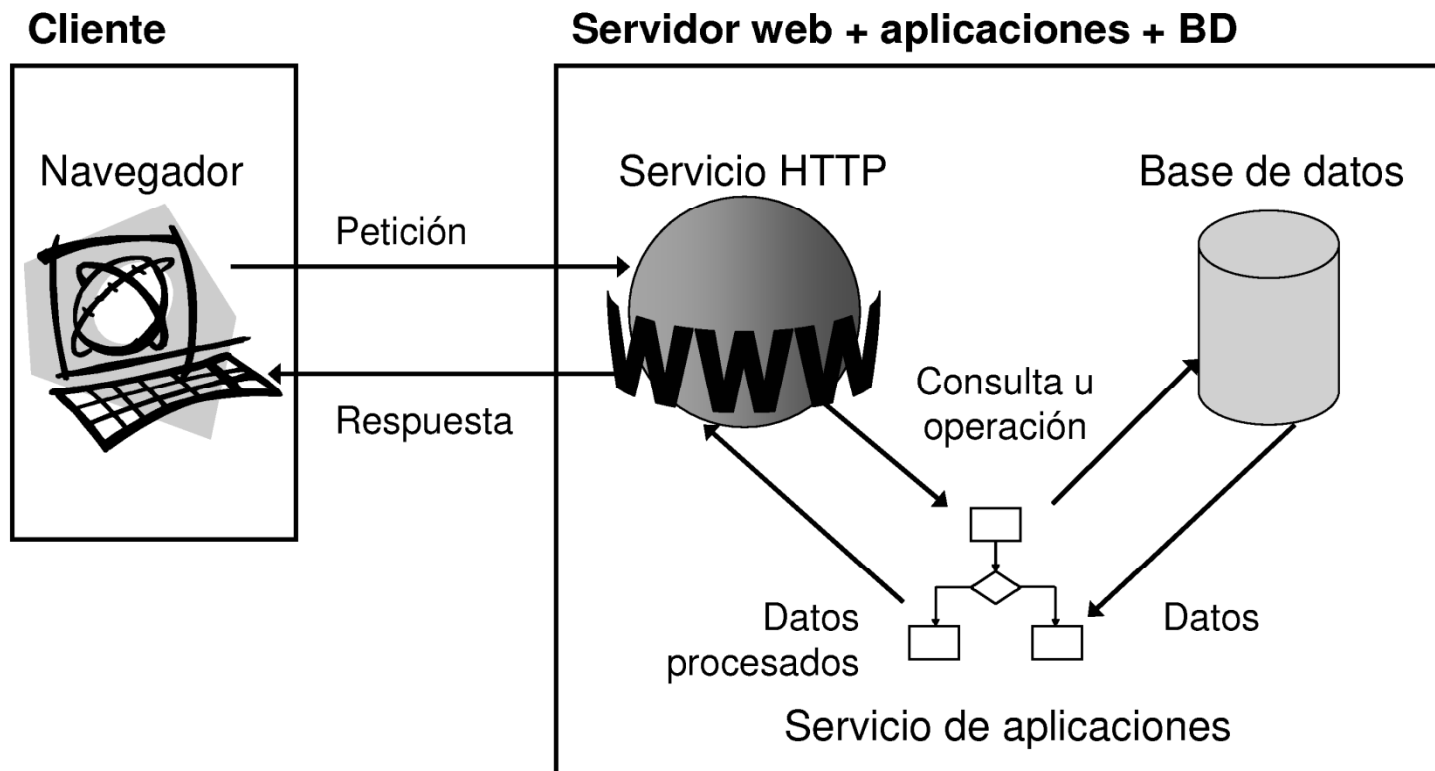
#### ➤ Distribución en dos niveles (1)

- ✓ Un único ordenador aloja el servicio de HTTP, la lógica de negocio y la lógica de datos y los datos:
  - Presentación distribuida.
  - Aplicación distribuida (opcional en función de si el cliente hace parte del trabajo, JS, PHP).



#### ➤ Distribución en dos niveles (2)

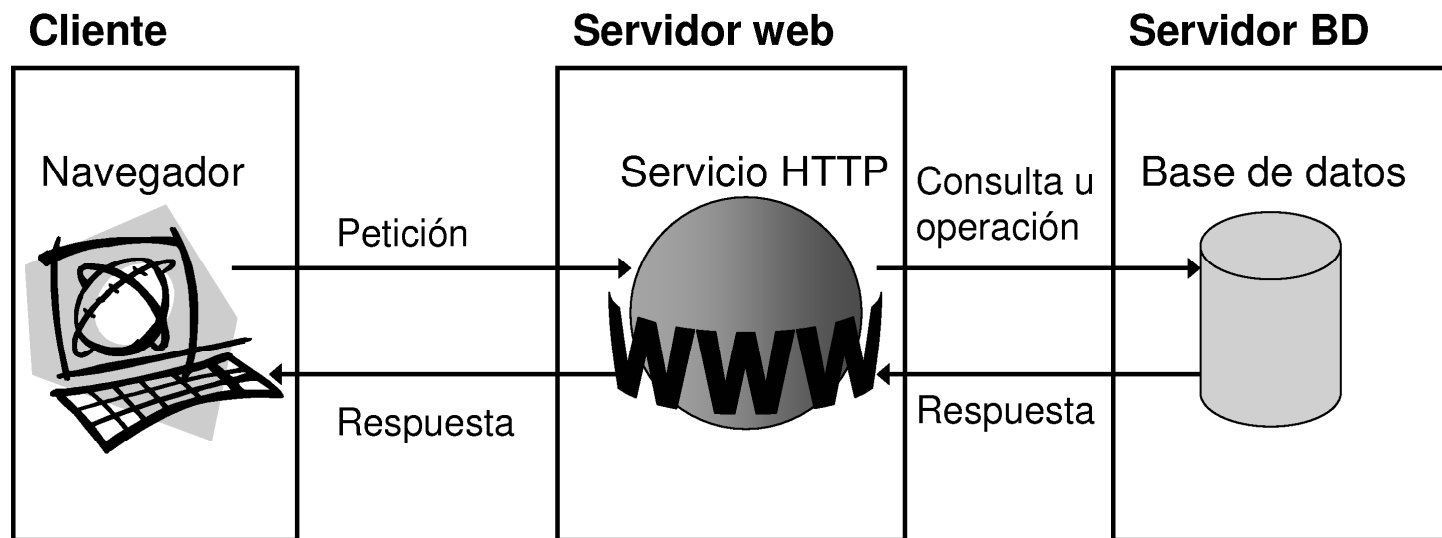
- ✓ Se separa la lógica de negocio del servicio de HTTP y se incluye el servicio de aplicaciones para gestionar los procesos que implementan la misma (JSP).





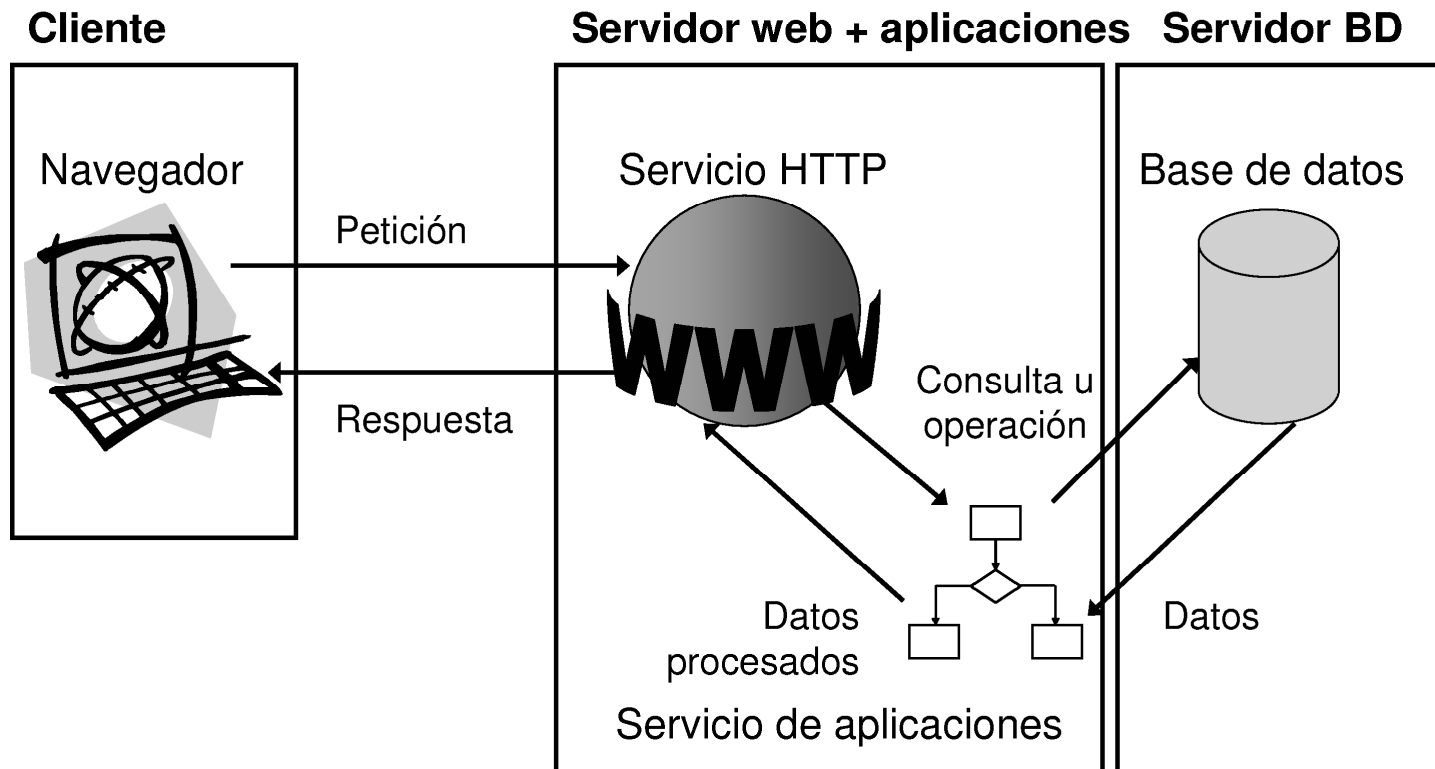
#### ➤ Distribución en tres niveles (1)

- ✓ Similar a la distribución en dos niveles (1) pero separando la lógica de datos y los datos a un servidor de bases de datos específico:
  - Datos distribuidos.



#### ➤ Distribución en tres niveles (2)

- ✓ Similar a la distribución en dos niveles (2) pero separando la lógica de datos y los datos a un servidor de bases de datos específico:



#### ➤ Distribución en tres niveles (3)

- ✓ Similar a la distribución en dos niveles (2) pero separando tanto la lógica de negocio como la lógica de datos y los datos a servidores específicos:

