



## Capítulo 1

# Arquitecturas y tecnologías de programación sobre clientes web

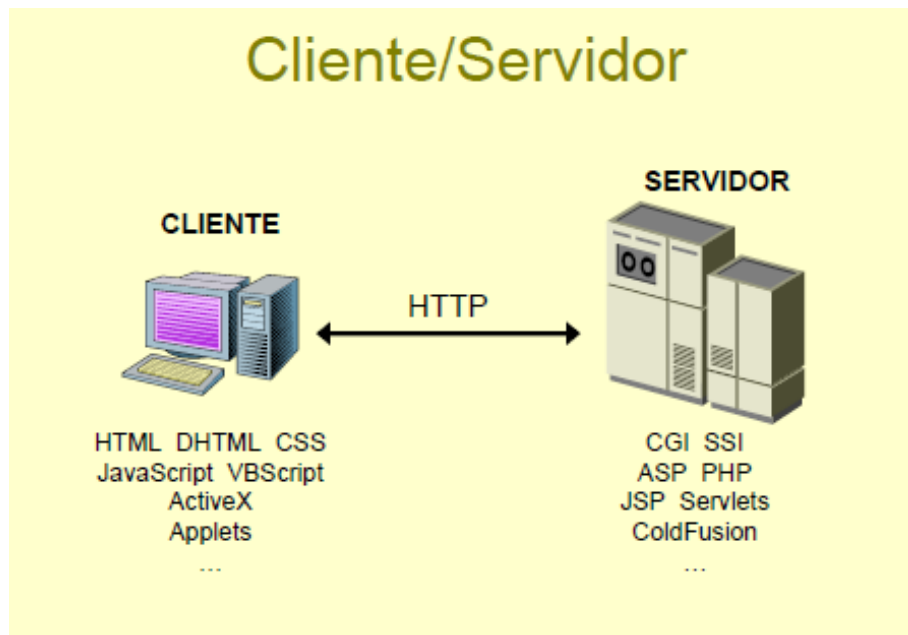
### 1.1. ¿ Qué es una aplicación web ?

A continuación se presentan distintas definiciones del concepto de aplicaciones web extraídas de distintas direcciones de Internet:

- En ingeniería de software se denomina aplicación web a las aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

Las aplicaciones web son muy populares debido a lo práctico del navegador como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea y la propia Wikipedia que son ejemplos bien conocidos de aplicaciones web.

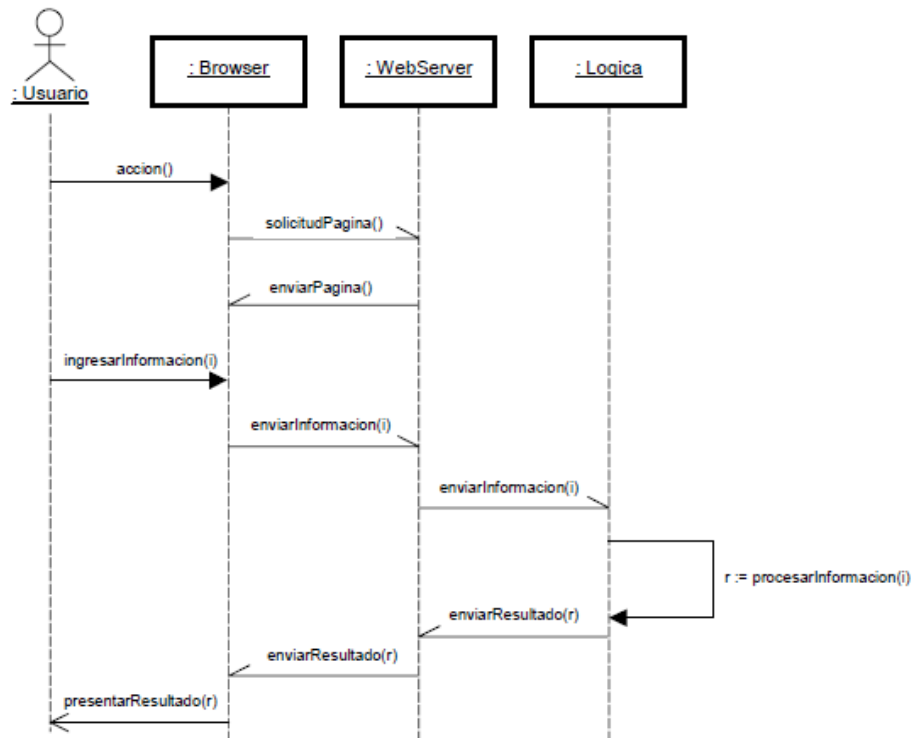
- Aplicación informática que se ejecuta en entorno web.
- Aplicación cliente/servidor: el cliente (navegador), el servidor (servidor web) y la comunicación (protocolo HTTP).



- Aplicación web = Página web + dinámica
  - HTML o XHTML
  - Imágenes, sonidos, etc...
  - Programación para ejecución en el cliente (javascript, VBScript, Applets)
  - Programación para ejecución en el servidor(CGI, Servlets, JSP, ASP, PHP).
- Las aplicaciones web ofrecen servicios a los usuarios de Internet que acceden utilizando un navegador web como Internet Explorer, Firefox, Chrome o Safari entre otros, dirigiéndose a una dirección de Internet donde obtendrán los servicios que buscan.

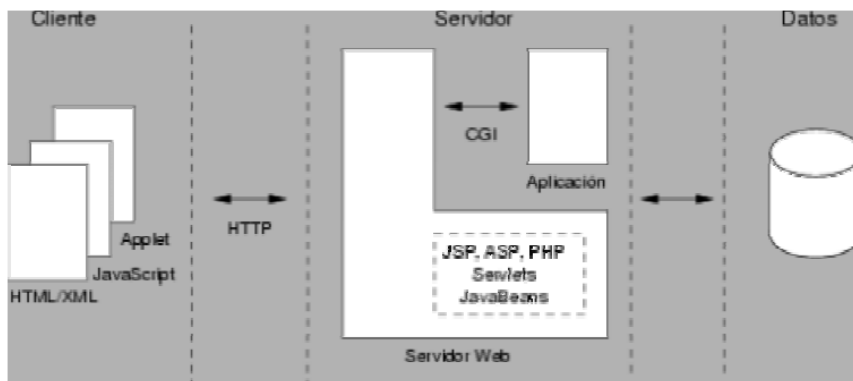
Las aplicaciones web pueden ser de acceso público como tiendas virtuales, diarios digitales, portales de Internet,... o de acceso restringido como son las intranets para mejorar las gestiones internas de su empresa, o el uso de extranets para aumentar y mejorar el servicio con sus distribuidores, clientes, proveedores, comerciales y colaboradores externos.

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía como interfaz de



usuario que tenía que ser instalado por separado en cada ordenador personal de cada usuario. El cliente realizaba peticiones a otro programa -el servidor- que le daba respuesta.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.



## 1.2. Arquitecturas

<https://www.youtube.com/watch?v=Tn4Kp720Hkk>

Las aplicaciones web son un tipo especial de aplicaciones cliente/servidor.

Cliente/servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. Normalmente, los servidores son ordenadores potentes dedicados a gestionar unidades de disco (servidor de ficheros), impresoras (servidor de impresoras), datos (servidor de bases de datos) o incluso aplicaciones (servidor de aplicaciones), mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores.

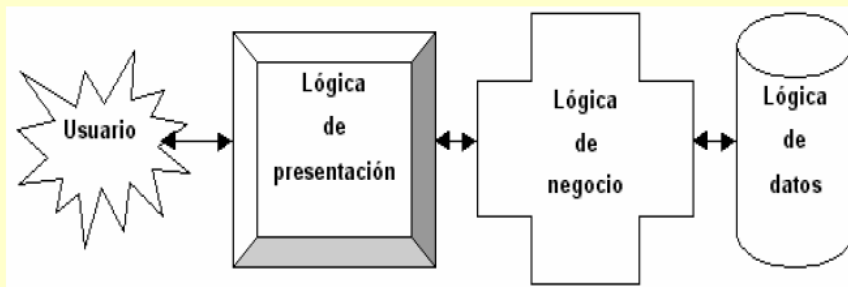
Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos. La arquitectura cliente/servidor permite la creación de **aplicaciones distribuidas**. La principal ventaja de esta arquitectura es que facilita la **separación de las funciones** según su servicio, permitiendo situar cada función en la plataforma más adecuada para su ejecución.

La arquitectura cliente/servidor nos permite la **separación de funciones** en tres niveles:

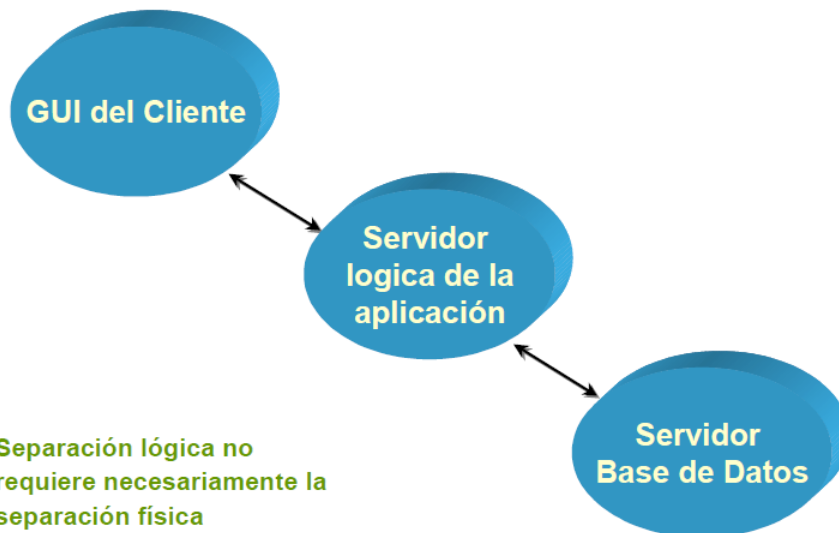
1. **Lógica de presentación.** Se encarga de la entrada y salida de la aplicación con el usuario. Sus principales tareas son: obtener información del usuario, enviar la información del usuario a la lógica de negocio para su procesamiento, recibir los resultados del procesamiento de la lógica de negocio y presentar estos resultados al usuario.
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 (business rules) que tiene que cumplir la aplicación (facturación, cálculo de nóminas, control de inventario, etc.) y enviar el resultado del procesamiento al nivel de presentación.
3. **Lógica de datos.** Se encarga de gestionar los datos a nivel de almacenamiento. Sus principales tareas son: almacenar los datos, recuperar los datos, mantener los datos y asegurar la integridad de los datos.

## Separación de funciones (I)

- Lógica de presentación
- Lógica de negocio (aplicación)
- Lógica de datos



## Cliente-Servidor de Tres-Capas



Si un sistema distribuido se diseña correctamente, los tres niveles anteriores pueden distribuirse y redistribuirse independientemente sin afectar al funcionamiento de la aplicación.

Según como se distribuyan las tres funciones básicas de una aplicación (presentación, negocio y datos) entre el cliente y el servidor, podemos contemplar tres modelos: presentación distribuida, aplicación distribuida y datos distribuidos.

- **Presentación distribuida** El cliente sólo mantiene la presentación, el resto de la aplicación se ejecuta remotamente. La presentación distribuida, en su forma más simple, es una interfaz gráfica de usuario a la que se le pueden acoplar controles de validación de datos, para evitar la validación de los mismos en el servidor.
- **Aplicación distribuida** Es el modelo que proporciona máxima flexibilidad, puesto que permite tanto a servidor como a cliente mantener la lógica de negocio realizando cada uno las funciones que le sean más propias, bien por organización, o bien por mejora en el rendimiento del sistema.
- **Datos distribuidos** Los datos son los que se distribuyen, por lo que la lógica de datos es lo que queda separada del resto de la aplicación. Se puede dar de dos formas: ficheros distribuidos o bases de datos distribuidas.

Un sistema cliente/servidor suele presentar las siguientes características:

1. Una combinación de la parte cliente (también llamada **front-end**) que interactúa con el usuario (hace de interfaz entre el usuario y el resto de la aplicación) y la parte servidor (o **back-end**) que interactúa con los recursos compartidos (bases de datos, impresoras, módems).
2. La parte cliente y servidor tienen diferentes necesidades de recursos a la hora de ejecutarse: velocidad de procesador, memoria, velocidad y capacidad de los discos duros, dispositivos de entrada/salida, etc.
3. El entorno suele ser heterogéneo y multivendedor. El hardware y sistema operativo del cliente y el servidor suelen diferir.
4. Normalmente la parte cliente se implementa haciendo uso de una interfaz gráfica de usuario.

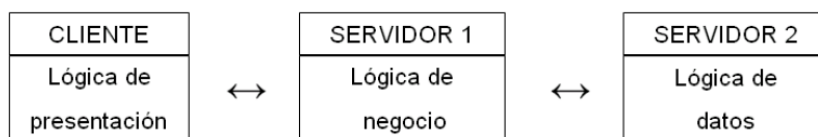
### 1.3. Arquitecturas de dos y tres niveles

La diferencia entre las arquitecturas de dos y tres niveles (o capas) estriba en la forma de distribución de la aplicación entre el cliente y el servidor.

Aunque todos los modelos de distribución en aplicaciones cliente/servidor que se han comentado antes se basan en arquitecturas de dos capas, normalmente cuando se habla de aplicaciones de dos niveles se está haciendo referencia a una aplicación donde el cliente mantiene la lógica de presentación, de negocio, y de acceso a los datos, y el servidor únicamente gestiona los datos. Suelen ser aplicaciones cerradas que supeditan la lógica de los procesos cliente al gestor de bases de datos que se está usando.

En las arquitecturas de tres niveles, la lógica de presentación, la lógica de negocio y la lógica de datos están separadas, de tal forma que mientras la lógica de presentación se ejecutará normalmente en la estación cliente, la lógica de negocio y la de datos pueden estar repartidas entre distintos procesadores. En este tipo de aplicaciones suelen existir dos servidores: uno contiene la lógica de negocio y otro la lógica de datos.

El objetivo de aumentar el número de niveles en una aplicación distribuida es lograr una mayor independencia entre un nivel y otro, lo que facilita la portabilidad en entornos heterogéneos y la escalabilidad en caso de incorporación de nuevos clientes.





## 1.4. Navegador web o Browser

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Para poder cumplir con todas estas funciones, los navegadores tienen la posibilidad de ejecución de programas de tipo script, con modelos de objetos que permiten manipular los contenidos de los documentos. Estos lenguajes de programación son VBScript, JScript y JavaScript, y proporcionan las soluciones llamadas del lado del cliente y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor proporcionando un alto grado de interacción con el usuario dentro del documento.

Otras de las posibilidades de los navegadores es la gestión del llamado HTML dinámico (DHTML). Éste está compuesto de HTML, hojas de estilo en cascada, (Cascade Style Sheets, CSS), modelo de objetos y scripts de programación que permiten formatear y posicionar correctamente los distintos elementos HTML de las páginas web, permitiendo un mayor control sobre la visualización de las páginas.

En esta línea, los navegadores han ido un poco más allá y permiten la visualización de documentos XML después de haber sido transformado adecuadamente a HTML por las hojas de estilo extensibles XSL. De esta manera se puede elegir visualizar ciertos elementos y otros no, dependiendo de las circunstancias. Además, los navegadores permiten la ejecución de aplicaciones dentro de los documentos mostrados. Las dos posibilidades más populares son la tecnología ActiveX y los applets Java. Los applets Java son pequeños programas que se descargan del servidor web y se ejecutan en la JVM (Java Virtual Machine) del navegador.

## 1.5. Servidor web

El servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.



*En el módulo de Desarrollo Web en Entorno Servidor se desarrollará la parte del servidor web*

## 1.6. Tecnologías de ejecución en el cliente

### ■ JavaScript

Principales características:

- Sintaxis semejante a C y Java.
- Ejecución interpretada por el navegador.
- Sometido a restricciones de seguridad.
- Facilidad para modificar dinámicamente el documento HTML mediante DOM.



**Limitaciones:** Lenguaje de programación no estructurado, no apto para aplicaciones demasiado grandes, comportamiento dependiente del navegador, algunos navegadores no lo ejecutan, código fuente disponible para el cliente. Estas limitaciones se están superando con las nuevas versiones.

### ■ Applets Java

Programas Java que se ejecutan en el navegador: El cliente descarga las clases (bytecodes), el cliente ejecuta el código en una JVM. Se ejecuta con restricciones de seguridad. Incrustado en navegador o en ventana propia.

**Limitaciones:** Navegadores sin JVM o con distintas versiones, menor interacción con el documento HTML que JavaScript.

### ■ Plugins

Programas en código nativo que extienden la funcionalidad del navegador.

**Usos habituales:** Visualizar o interaccionar con nuevos tipos de datos, acceder a funcionalidades básicas del navegador (carga de URLs, etc.), implementar nuevos protocolos de comunicación (por ejemplo, streaming multimedia). Audio, vídeo, juegos, animaciones, 3D, etc.

### ■ AJAX

Acrónimo de Asynchronous JavaScript and XML.

Aplicaciones Web ejecutadas en el cliente con comunicación asíncrona con el servidor en segundo plano, basada en XML. Aumenta significativamente la interactividad y velocidad de la aplicación. Puede reducir la usabilidad y accesibilidad.

El **Document Object Model** o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

Combinación de tecnologías existentes previamente: XHTML y CSS para presentación. JavaScript para ejecución en cliente. Modifica el documento XHTML mediante DOM. Comunicación asíncrona con el servidor mediante XMLHttpRequest. La comunicación asíncrona suele basarse en intercambio de documentos XML.

- **ActiveX (Microsoft)**

Controles incrustables en páginas Web. Requiere instalación. Riesgos de seguridad elevados. Sólo para MS-Windows.

- **Adobe Flash:**

Multimedia, animaciones, juegos, 3D, etc. Lenguaje de programación ActionScript.