

INTRODUCCIÓN A WWW



En esta primera lección, vamos a revisar una Introducción a la tecnología Web.

Un sitio Web contiene información estática, y que debemos cambiar manualmente si deseamos ver cambios en la misma. En cambio una aplicación Web la información se puede recuperar de manera dinámica.

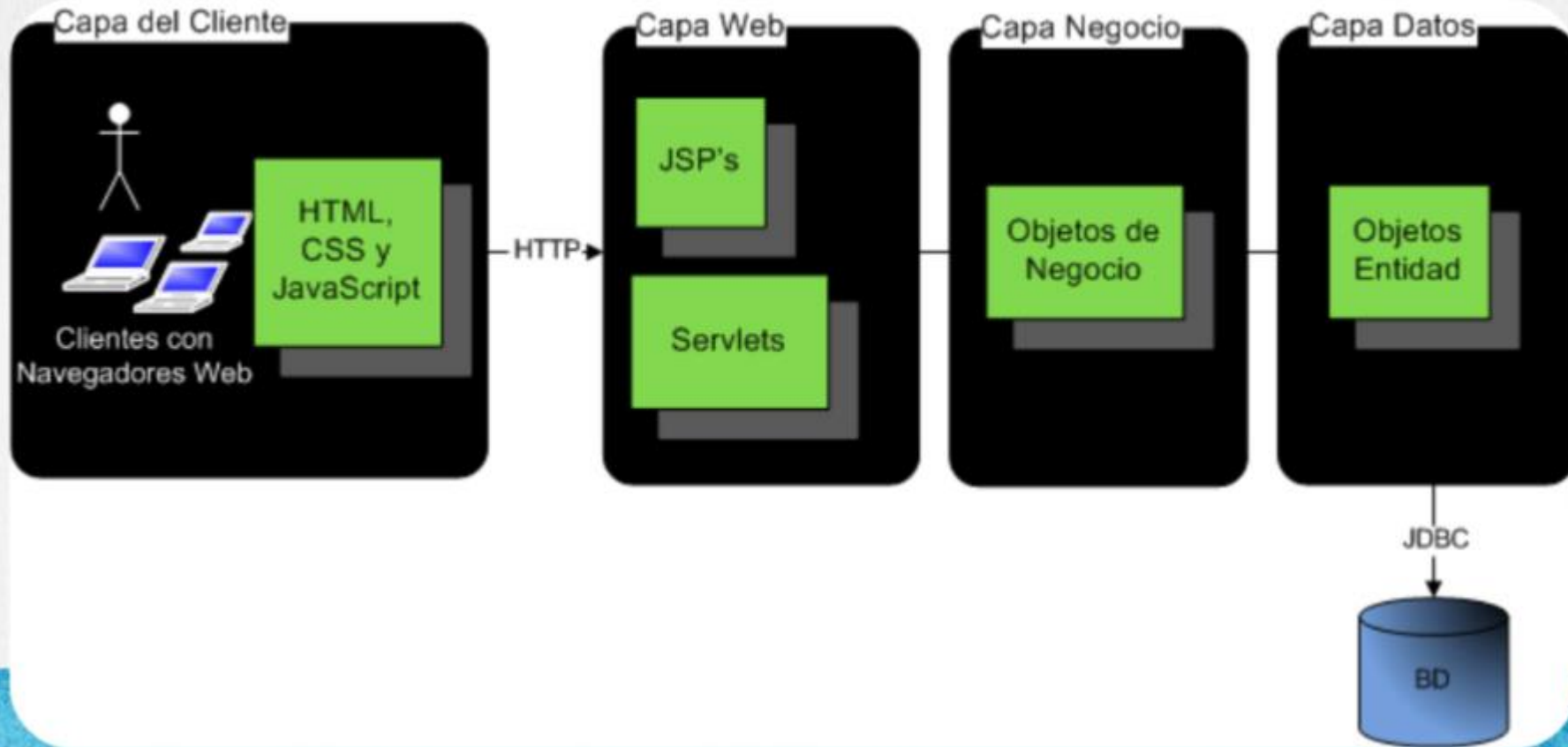
En un sitio web normalmente contamos con varios elementos como pueden ser páginas HTML, imágenes, hojas de estilo CSS, archivos con código JavaScript entre varios elementos más.

JavaScript nos permite agregar cierto dinamismo a nuestra página pero la información no puede provenir de base de datos ni puede provenir de archivos almacenados o de información que esté depositada en otras fuentes de información. En cambio, una aplicación Web maneja de manera dinámica esta información.

Así, el objetivo de una aplicación Web es procesar una petición de un cliente y una vez que sea procesada esta petición en un servidor web, el resultado se vuelve a mostrar al cliente que solicitó cierta información. Este es el flujo que comúnmente veremos en una aplicación Web.

Internet o lo que es lo mismo WWW (World Wide Web) permite crear sitios y aplicaciones web, y estos pueden ser accedidos desde internet o desde una intranet. Al día de hoy existen una gran variedad de aplicaciones que permiten cubrir diferentes giros de negocios como pueden ser tiendas virtuales, noticias en internet, redes sociales, entre otros tipos de aplicaciones web.

ARQUITECTURA DE UNA APLICACIÓN WEB CON JAVA



Vamos a revisar la arquitectura de una aplicación Web con Java, como podemos observar tenemos distintas **capas lógicas** que se encargan de cierta funcionalidad en nuestra aplicación Web.

Por un lado tenemos la capa del cliente. En esta capa es el navegador Web el encargado de realizar las peticiones al servidor Java, así como mostrar la información al cliente. En esta capa tenemos tecnologías como pueden ser HTML, hojas de estilo CSS y JavaScript.

Y del lado del servidor tenemos las siguientes capas utilizadas en una aplicación Web. Comúnmente tenemos la capa web, la capa de negocios y la capa de datos. Esta comunicación la establecemos por medio del protocolo HTTP (**Hypertext Transfer Protocol**), que es el protocolo utilizado en Internet.

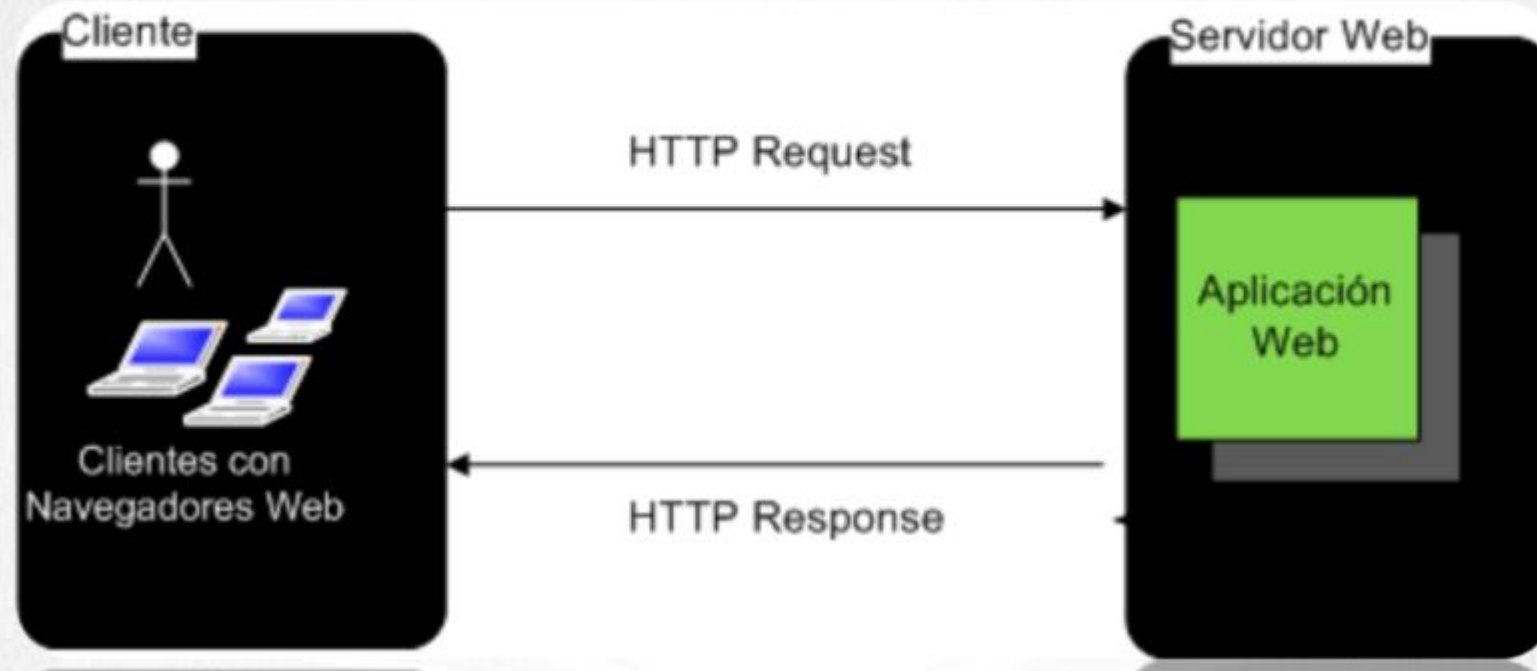
Del lado del servidor pueden haber uno o más servidores, ya que cada capa del lado del servidor podría estar en un solo servidor o en varios. Del lado del servidor Java podemos tener las siguientes capas:

Capa Web: En el caso de la capa web podemos tener tecnologías como pueden ser JSPs y Servlets y es precisamente el objetivo de este curso estudiar la tecnología que nos va a permitir procesar la capa web.

En las capas que siguen a la capa web podemos tener **capas de negocio** y **capas de acceso a datos**, esta capa de datos y sus patrones de diseño los analizamos en el curso de JDBC y también existen otros tipos de cursos en los que podemos aprender a manejar Frameworks como pueden ser Hibernate, JPA (Java Persistence API) entre otro tipo de Frameworks del manejo de la capa de datos. La capa de negocio hemos utilizando hasta el momento objetos de Java conocidos como Java Beans o POJOs y estos objetos nos van a permitir establecer las reglas de negocio para nuestra aplicación web. De igual manera existen tecnologías para la capa de negocios, como pueden ser los EJBs (Enterprise JavaBeans) o Frameworks como Spring, los cuales uno de sus principales objetivos es brindar las características que debe de poseer la capa de negocio, como seguridad, transaccionalidad, entre muchas características más.

En este curso vamos a enfocarnos a crear la capa web aplicando las mejores prácticas y patrones de diseño que nos permitirán crear aplicaciones web robustas, extensibles, escalables y más fáciles de dar mantenimiento, entre varias características más.

PROCESO DE UNA PETICIÓN HTTP



Vamos a revisar el proceso de una petición HTTP. En la imagen anterior habíamos comentado que tenemos un cliente (navegador Web) y tenemos un servidor web. En el caso de nuestra aplicación web, el servidor puede ser el servidor de Tomcat, Glassfish, Jboss, entre otros, y lo que sucede es que normalmente un usuario hace una petición hacia nuestro servidor web. Esto se conoce como HTTP Request, es decir la petición que realiza el usuario del cliente al servidor.

Una vez que el servidor obtiene la información que ha solicitado el cliente o guarda la información que el cliente ha proporcionado, entonces el servidor manda una respuesta (HTTP Response) a nuestro cliente. La respuesta se visualiza en los navegadores web como pueden ser Internet Explorer, Firefox, Google Chrome, entre otros. Este es el proceso de una petición HTTP en la creación de nuestra aplicación web.

Cabe resaltar la importancia del protocolo HTTP, ya que este protocolo nos va a permitir establecer la comunicación entre nuestro navegador web y nuestro servidor web y por medio de estas peticiones y respuestas es que vamos a establecer la comunicación entre cliente-servidor entre nuestras distintas aplicaciones web.

Una de las grandes ventajas del uso de este tipo de aplicaciones Web, es decir, cliente-servidor, es que es posible utilizar prácticamente cualquier computadora para emular este ambiente de cliente-servidor, o como hemos comentado, podemos separar tanto el cliente y cada una de las capas del servidor según sea necesario. De esta manera las aplicaciones que desarrollemos no necesitamos de equipos muy poderosos, sino que la gran mayoría de las aplicaciones las podremos emular en la comodidad de nuestro equipo de cómputo, laptop o computadora de Escritorio.

