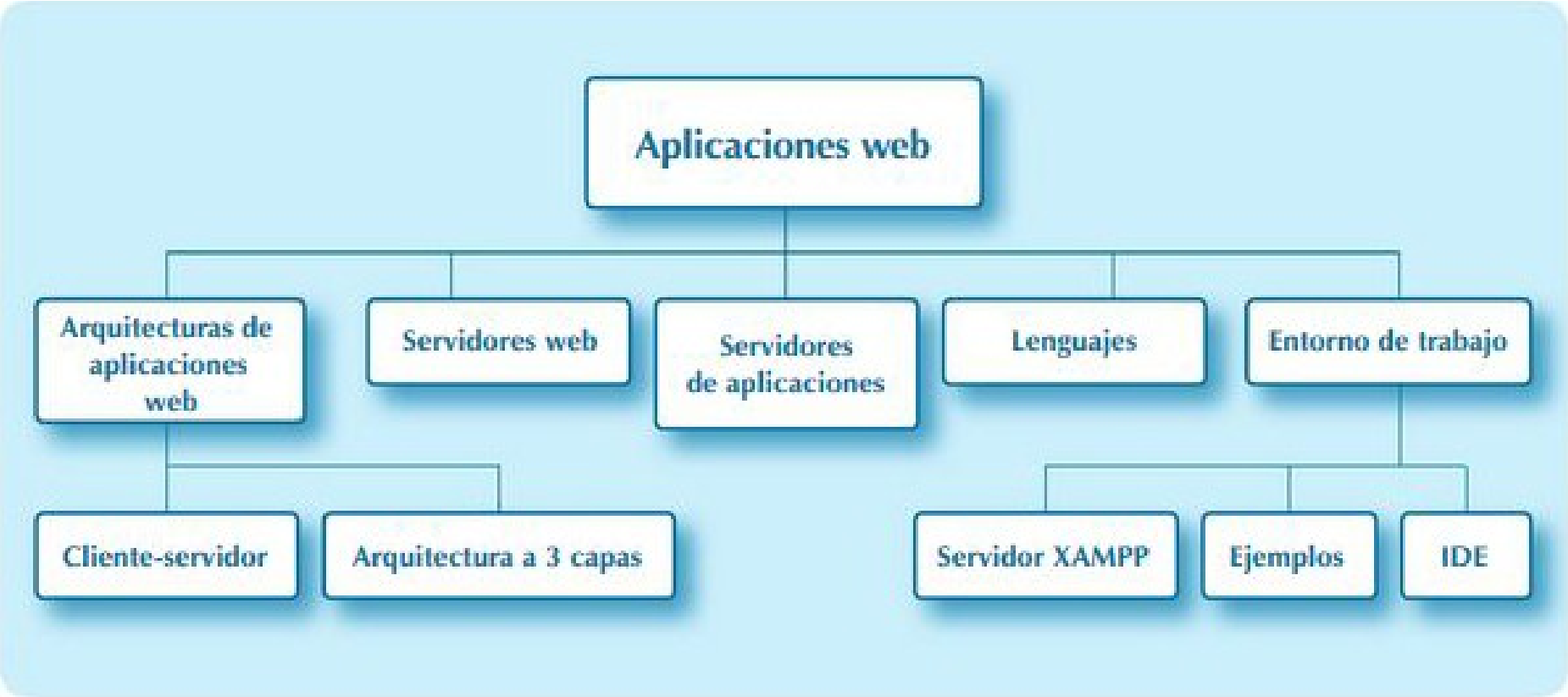
DESARROLLO WEB EN ENTORNO SERVIDOR

Dpto. Informática y comunicaciones.
Curso. 2024 / 2025
José Luís Casado Valero

Virrey Morcillo
Instituto de Educación Secundaria

1. Introducción al desarrollo web:



1.1 Glosario:

Aplicación web. Aplicación informática a la que se accede mediante una interfaz web utilizando un navegador.

Cliente. En el modelo cliente-servidor, los clientes solicitan funcionalidad a los servidores.

Framework. Un *framework* es una plataforma para el desarrollo de aplicaciones. Puede incluir librerías y metodologías.

HTML. *Hyper Text Markup Language*, el lenguaje básico para la creación de páginas web. Es un estándar del W3C.

HTTP. *Hyper Text Transfer Protocol*, protocolo de transferencia de hipertexto. Es el protocolo que utilizan clientes y servidores web para comunicarse. Es un estándar del W3C.

HTTPS. Versión segura del HTTP.

IDE. *Integrated Development Envoroment*, entorno de desarrollo integrado. Programa que integra herramientas útiles para programar, como editores, compiladores o control de versiones.

Protocolo. Según la RAE, conjunto de reglas que se establecen en el proceso de comunicación entre dos sistemas

Servidor. En el modelo cliente-servidor, los servidores proveen servicios a los clientes.

W3C. *World Wide Web Consortium*, organismo que elabora y mantiene varios de los estándares más importantes en Internet, como el HTTP o el HTML.

1.2 *Modelo Cliente-servidor:*

1. El cliente solicita a un servidor web una página con extensión .htm, .html o .xhtml.
2. El servidor busca esa página en un almacén de páginas (cada una suele ser un fichero).
3. Si el servidor encuentra esa página, la recupera.
4. Y por último se la envía al navegador para que éste pueda mostrar su contenido.

Figura 1.1
Modelo cliente-servidor.

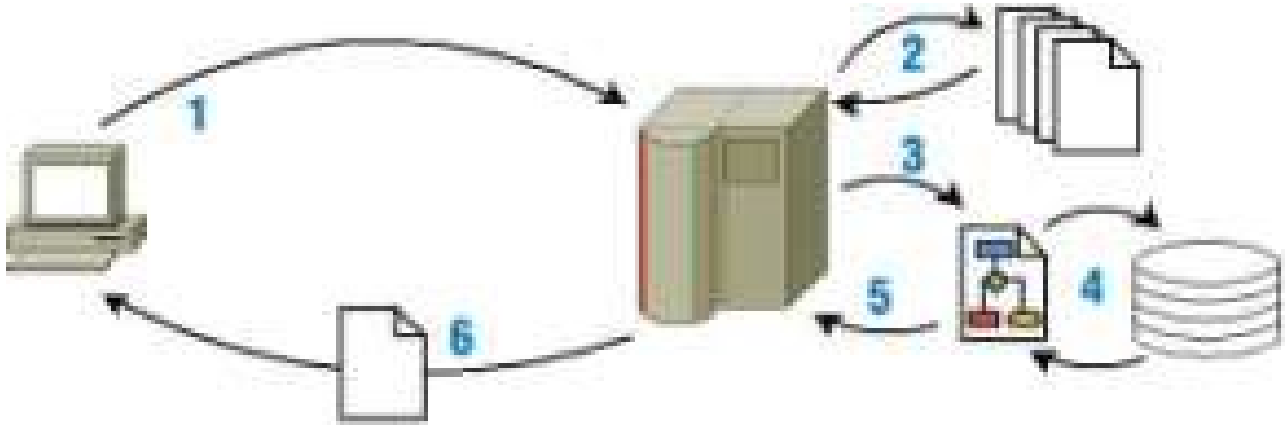
1.3 *Páginas web estáticas y dinámicas.*

- *Las páginas web estáticas se encuentran almacenadas en su forma definitiva, tal y como se crearon, y su contenido no varía. Son útiles para mostrar una información concreta, y mostrarán esa misma información cada vez que se carguen. La única forma en que pueden cambiar es si un programador la modifica y actualiza su contenido.*
- *En contraposición a las páginas web estáticas, como ya te imaginarás, existen las páginas web dinámicas. Estas páginas, como su nombre indica, se caracterizan porque su contenido cambia en función de diversas variables, como puede ser el navegador que estás usando, el usuario con el que te has identificado, o las acciones que has efectuado con anterioridad.*

Dentro de las páginas web dinámicas, es muy importante distinguir dos tipos:

- *Aquellas que incluyen código que ejecuta el navegador. En estas páginas el código ejecutable, normalmente en lenguaje JavaScript, se incluye dentro del HTML (o XHTML) y se descarga junto con la página. Cuando el navegador muestra la página en pantalla, ejecuta el código que la acompaña. Este código puede incorporar múltiples funcionalidades que pueden ir desde mostrar animaciones hasta cambiar totalmente la apariencia y el contenido de la página.*

- *Muchas páginas en Internet que no tienen extensión .htm, .html o .xhtml. Muchas de estas páginas tienen extensiones como .php, .asp, .jsp, .cgi o .aspx. En éstas, el contenido que se descarga al navegador es similar al de una página web estática: HTML (o XHTML). Lo que cambia es la forma en que se obtiene ese contenido. Al contrario de lo que vimos hasta ahora, esas páginas no están almacenadas en el servidor; más concretamente, el contenido que se almacena no es el mismo que después se envía al navegador. El HTML de estas páginas se forma como resultado de la ejecución de un programa, y esa ejecución tiene lugar en el servidor web (aunque no necesariamente por ese mismo servidor).*



El esquema de funcionamiento de una página web dinámica es el siguiente:

1. El cliente web (navegador) de tu ordenador solicita a un servidor web una página web.
2. El servidor busca esa página y la recupera.
3. En el caso de que se trate de una página web dinámica, es decir, que su contenido deba ejecutarse para obtener el HTML que se devolverá, el servidor web contacta con el módulo responsable de ejecutar el código y se lo envía.
4. Como parte del proceso de ejecución, puede ser necesario obtener información de algún repositorio (Cualquier almacén de información digital, normalmente una base de datos), como por ejemplo consultar registros almacenados en una base de datos.
5. El resultado de la ejecución será una página en formato HTML, similar a cualquier otra página web no dinámica.
6. El servidor web envía el resultado obtenido al navegador, que la procesa y muestra en pantalla.
Este procedimiento tiene lugar constantemente mientras consultamos páginas web.

***Ejemplo**: Correo web.*

1.4 *Ventajas y limitaciones.*

Aunque la utilización de páginas web dinámicas te parezca la mejor opción, las páginas web ***estáticas*** *tienen también algunas* ***ventajas****:*

- *No es necesario saber programar para crear un sitio que utilice únicamente páginas web estáticas. Simplemente habría que conocer HTML/XHTML y CSS, e incluso esto no sería indispensable: se podría utilizar algún programa de diseño web para generarlas.*
- *La característica diferenciadora de las páginas web estáticas es que su contenido nunca varía, y esto en algunos casos también puede suponer una ventaja. Sucede, por ejemplo, cuando quieres almacenar un enlace a un contenido concreto del sitio web: si la página es dinámica, al volver a visitarla utilizando el enlace su contenido puede variar con respecto a cómo estaba con anterioridad.*
  O cuando quieres dar de alta un sitio que has creado en un motor de búsqueda como Google. Para que Google muestre un sitio web en sus resultados de búsqueda, previamente tiene que indexar su contenido.

- *Menos recursos. Como ya sabes, para que un servidor web pueda procesar una página web dinámica, necesita ejecutar un programa. Esta ejecución la realiza un módulo concreto, que puede estar integrado en el servidor o ser independiente. Además, puede ser necesario consultar una base de datos como parte de la ejecución del programa. Es decir, la ejecución de una página web dinámica requiere una serie de recursos del lado del servidor. Estos recursos deben instalarse y mantenerse. Las páginas web estáticas sólo necesitan un servidor web que se comunique con tu navegador para enviártela. Y de hecho para ver una página estática almacenada en tu equipo no necesitas siquiera de un servidor web. Son archivos que pueden almacenarse en un soporte de almacenamiento como puede ser un disco óptico o una memoria USB y abrirse desde él directamente con un navegador web.*

Pero si decides hacer un sitio web utilizando páginas estáticas, ten en cuenta que tienen limitaciones:

- *La desventaja más importante ya la comentamos anteriormente: la actualización de su contenido debe hacerse de forma manual editando la página que almacena el servidor web. Esto implica un mantenimiento que puede ser prohibitivo en sitios web con gran cantidad de contenido.*

***Autoevaluación**:*

¿Qué tipo de tecnología emplearías para crear una página web personal? ¿Sería necesario utilizar páginas dinámicas? ¿Qué tareas de actualización y mantenimiento tendrías que realizar en cada caso?

1.5 Protocolos de comunicaciones.

La comunicación entre el cliente y el servidor web se realiza mediante los protocolos:

- *HTTP. Puerto bien conocido 80. (No es aceptado ya por muchos navegadores).*
- *HTTPS. Puerto bien conocido 553. (Version segura)*

El protocolo de transferencia de hipertextos se utiliza en cualquier transacción de Internet. En él ***se define la sintaxis y la semántica que utilizan los proxies, los servidores y todos los componentes de la arquitectura web*** *para poder establecer una comunicación entre ellos.*

Es un protocolo orientado a la transacción que ***se apoya en el esquema de petición y respuesta*** *habitual entre un cliente o agente del usuario, como un navegador web y un servidor. La información que se transmite en este proceso se denomina recurso y se identifica por URL.*

1.6 Evolución en el desarrollo web.

- *Las primeras páginas web que se crearon en Internet fueron páginas estáticas. A esta web compuesta por páginas estáticas se le considera la primera generación.*
- *La segunda generación de la web surgió gracias a las páginas web dinámicas.*
- *Tomando como base las web dinámicas, han ido surgiendo otras tecnologías que han hecho evolucionar Internet hasta llegar a lo que ahora conocemos.* ***Aplicaciones Web.***

1.7 Aplicaciones web.

Las aplicaciones web emplean páginas web dinámicas para crear aplicaciones que se ejecuten en un servidor web y se muestren en un navegador. Puedes encontrar aplicaciones web para realizar múltiples tareas. Unas de las primeras en aparecer fueron las que viste antes, los clientes de correo, que te permiten consultar los mensajes de correo recibidos y enviar los tuyos propios utilizando un navegador.

Hoy en día existen aplicaciones web para multitud de tareas como procesadores de texto, gestión de tareas, o edición y almacenamiento de imágenes. Estas aplicaciones tienen ciertas ventajas e inconvenientes si las comparas con las aplicaciones tradicionales que se ejecutan sobre el sistema operativo de la propia máquina.

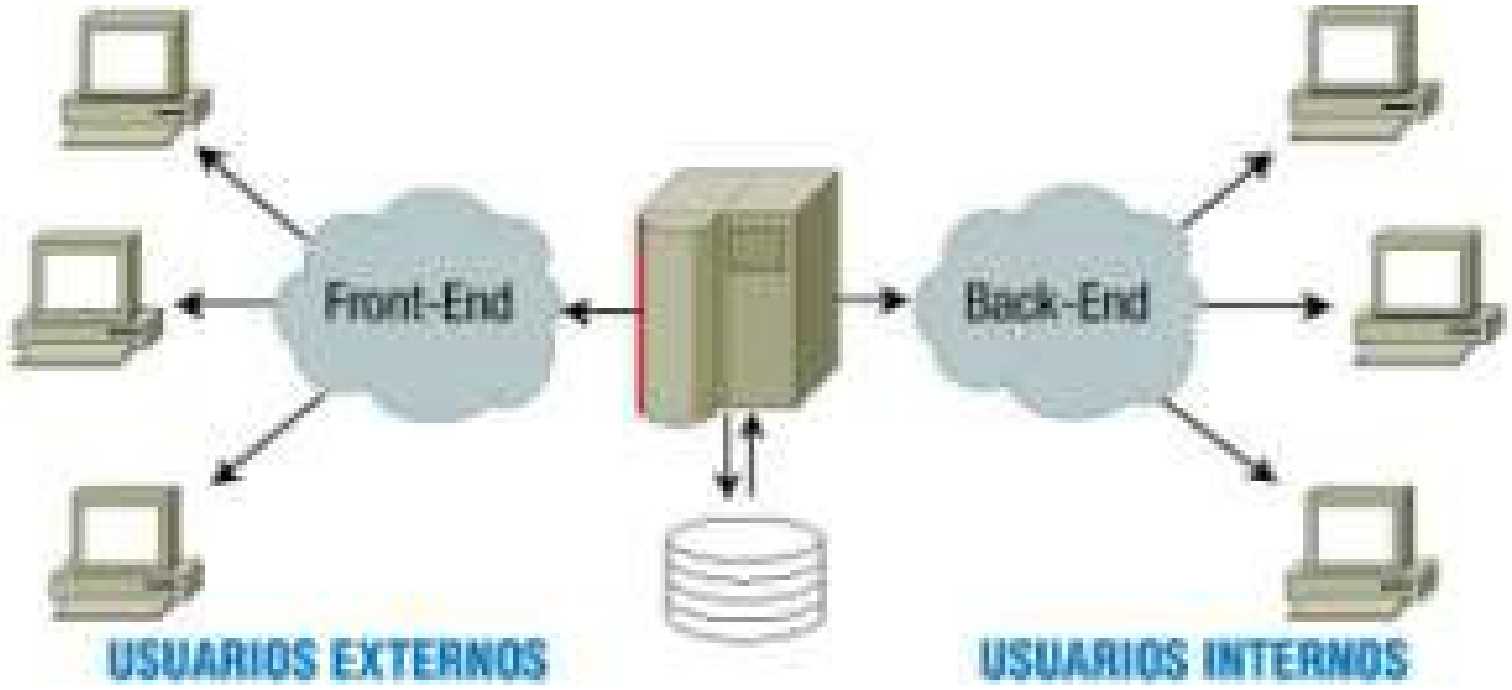
1.7.1 Ventajas de las aplicaciones web:

- *No es necesario instalarlas en aquellos equipos en que se vayan a utilizar. Se instalan y se ejecutan solamente en un equipo, en el servidor, y esto es suficiente para que se puedan utilizar deforma simultánea desde muchos equipos.*
- *Como solo se encuentran instaladas en un equipo, es muy sencillo gestionarlas (hacer copias de seguridad de sus datos, corregir errores, actualizarlas).*
- *Se pueden utilizar en todos aquellos sistemas que dispongan de un navegador web, independientemente de sus características (no es necesario un equipo potente) o de su sistema operativo.*
- *Se pueden utilizar desde cualquier lugar en el que dispongamos de conexión con el servidor. En muchos casos esto hace posible que se pueda acceder a las aplicaciones desde sistemas no convencionales, como por ejemplo teléfonos móviles.* ***(Multiplataforma)***

1.7.2 Inconvenientes de las aplicaciones web:

- *El interface de usuario de las aplicaciones web es la página que se muestra en el navegador. Esto restringe las características del interface a aquellas de una página web.*
- *Dependemos de una conexión con el servidor para poder utilizarlas. Si nos falla la conexión, no podremos acceder a la aplicación web.*
- *La información que se muestra en el navegador debe transmitirse desde el servidor. Esto hace que cierto tipo de aplicaciones no sean adecuadas para su implementación como aplicación web (por ejemplo, las aplicaciones que manejan contenido multimedia, como las de edición de vídeo).*

1.7.3 *Integración de Aplicaciones:*



1.7.4 *Ejecución de código en el servidor y en el cliente.*

- ***Servidor:*** *cuando tu navegador solicita a un servidor web una página, es posible que antes de enviártela haya tenido que ejecutar, por sí mismo o por delegación, algún programa para obtenerla. Ese programa es el que genera, en parte o en su totalidad, la página web que llega a tu equipo. En estos casos, el código se ejecuta en el entorno del servidor web.*
- ***Cliente:*** *además, cuando una página web llega a tu navegador, es también posible que incluya algún programa o fragmentos de código que se deban ejecutar. Ese código, normalmente en lenguaje JavaScript, se ejecutará en tu navegador y, además de poder modificar el contenido de la página.*

- ***AJAX:*** *nos posibilita realizar programas en los que el código JavaScript que se ejecuta en el navegador pueda comunicarse con un servidor de Internet para obtener información con la que, por ejemplo, modificar la página web actual.*
- ***Web-Socket:*** *nos permite una comunicación bidireccional entre el servidor y el cliente, esto se utiliza normalmente para la modificación de la información en tiempo real.*

1.7.5 Tecnologías para programación web del lado del servidor.

Los componentes principales con los que debes contar para ejecutar aplicaciones web en un servidor son los siguientes:

- *Un **servidor web** para recibir las peticiones de los clientes web (normalmente navegadores) y enviarles la página que solicitan (una vez generada puesto que hablamos de páginas web dinámicas). El servidor web debe conocer el procedimiento a seguir para generar la página web: qué módulo se encargará de la ejecución del código y cómo se debe comunicar con él.*
- *El **módulo encargado de ejecutar el código** o programa y generar la página web resultante. Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje y tecnología que utilicemos para programar la aplicación web.*
- *Una aplicación de **base de datos**, que normalmente también será un servidor. Este módulo no es estrictamente necesario pero en la práctica se utiliza en todas las aplicaciones web que utilizan grandes cantidades de datos para almacenarlos.*
- *El **lenguaje de programación** que utilizarás para desarrollar las aplicaciones.*

1.7.6 Organización de la aplicación, modelo / vista / control. (MVC)

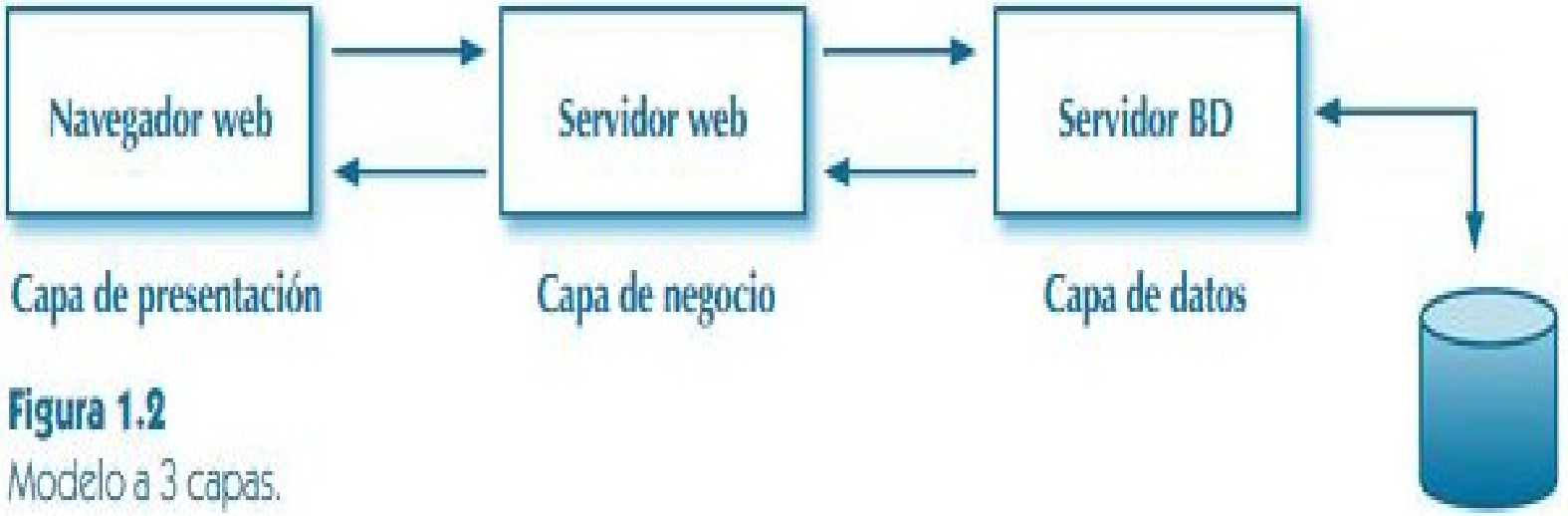


Figura 1.2
Modelo a 3 capas.

1.7.7 Arquitecturas y plataformas.

La primera elección que harás antes de comenzar a programar una aplicación web es la arquitectura que vas a utilizar. Podemos encontrar entre otras:

- ***Java EE** (Enterprise Edition), que antes también se conocía como J2EE. Es una plataforma orientada a la programación de aplicaciones en lenguaje Java. Puede funcionar con distintos gestores de bases de datos, e incluye varias librerías y especificaciones para el desarrollo de aplicaciones de forma modular.*

  Está apoyada por grandes empresas como Sun y Oracle, que mantienen Java, o IBM. Es una buena solución para el desarrollo de aplicaciones de tamaño mediano o grande. Una de sus principales ventajas es la multitud de librerías existentes en ese lenguaje y la gran base de programadores que lo conocen.

  Dentro de esta arquitectura existen distintas tecnologías como las páginas JSP y los servlets, ambos orientados a la generación dinámica de páginas web, o los EJB, componentes que normalmente aportan la lógica de la aplicación web.

- ***AMP.*** *Son las siglas de Apache, MySQL y PHP/Perl/Python. Las dos primeras siglas hacen referencia al servidor web (Apache) y al servidor de base de datos (MySQL). La última se corresponde con el lenguaje de programación utilizado, que puede ser PHP, Perl o Python, Dependiendo del sistema operativo que se utilice para el servidor, se utilizan las siglas LAMP (para Linux), WAMP (para Windows) o MAMP (para Mac). También es posible usar otros componentes, como el gestor de bases de datos PostgreSQL en lugar de MySQL.*

  Todos los componentes de esta arquitectura son de código libre (open source). Es una plataforma de programación que permite desarrollar aplicaciones de tamaño pequeño o mediano con un aprendizaje sencillo. Su gran ventaja es la gran comunidad que la soporta y la multitud de aplicaciones de código libre disponibles.

- ***ASP.Net*** *es la arquitectura comercial propuesta por Microsoft para el desarrollo de aplicaciones. Es la parte de la plataforma .Net destinada a la generación de páginas web dinámicas. Proviene de la evolución de la anterior tecnología de Microsoft, ASP.*

  El lenguaje de programación puede ser Visual Basic.Net o C#. La arquitectura utiliza el servidor web de Microsoft, IIS, y puede obtener información de varios gestores de bases de datos entre los que se incluye, como no, Microsoft SQL Server.

  Una de las mayores ventajas de la arquitectura .Net es que incluye todo lo necesario para el desarrollo y el despliegue de aplicaciones. Por ejemplo, tiene su propio entorno de desarrollo, Visual Studio, aunque hay otras opciones disponibles. La mayor desventaja es que se trata de una plataforma comercial de código propietario.

Para tomar una decisión correcta, deberás considerar entre otros los siguientes puntos:

- *¿Qué tamaño tiene el proyecto?*
- *¿Qué lenguajes de programación conozco? ¿Vale la pena el esfuerzo de aprender uno nuevo?*
- *¿Voy a usar herramientas de código abierto o herramientas propietarias? ¿Cuál es el coste de*
- *utilizar soluciones comerciales?*
- *¿Voy a programar la aplicación yo solo o formaré parte de un grupo de programadores?*
- *¿Cuento con algún servidor web o gestor de base de datos disponible o puedo decidir libremente*
- *utilizar el que crea necesario?*
- *¿Qué tipo de licencia voy a aplicar a la aplicación que desarrolle?*
- *Restricciones en los requisitos del cliente.*

1.7.8 Lenguajes de programación web:

Una de las diferencias más notables entre un lenguaje de programación web y otro es la manera en que se ejecutan en el servidor web. Debes distinguir tres grandes grupos:

- ***Lenguajes de guiones (scripting).*** *Son aquellos en los que los programas se ejecutan directamente a partir de su código fuente original. Se almacenan normalmente en un fichero de texto plano. Cuando el servidor web necesita ejecutar código programado en un lenguaje de guiones, le pasa la petición a un intérprete, que procesa las líneas del programa y genera como resultado una página web.*
  De los lenguajes que estudiaste anteriormente, pertenecen a este grupo Perl, Python, PHP y ASP (el precursor de ASP.Net).

- ***Lenguajes compilados a código nativo*** *(Denominación habitual del lenguaje máquina. Código que puede ser ejecutado directamente por el procesador). Son aquellos en los que el código fuente se traduce a código binario, dependiente del procesador, antes de ser ejecutado. El servidor web almacena los programas en su modo binario, que ejecuta directamente cuando se les invoca.*
  El método principal para ejecutar programas binarios desde un servidor web. Utilizando CGI podemos hacer que el servidor web ejecute código programado en cualquier lenguaje de propósito general como puede ser C.

- ***Lenguajes compilados a código intermedio**. Son lenguajes en los que el código fuente original se traduce a un código intermedio, independiente del procesador, antes de ser ejecutado. Es la forma en la que se ejecutan por ejemplo las aplicaciones programadas en Java, y lo que hace que puedan ejecutarse en varias plataformas distintas.*

Ventajas e inconvenientes:

- *Los lenguajes de guiones tienen la ventaja de que no es necesario traducir el código fuente original para ser ejecutados, lo que aumenta su portabilidad. Si se necesita realizar alguna modificación a un programa, se puede hacer en el momento. Por el contrario el proceso de interpretación ofrece un peor rendimiento que las otras alternativas.*
- *Los lenguajes compilados a código nativo son los de mayor velocidad de ejecución, pero tienen problemas en lo relativo a su integración con el servidor web. Son programas de propósito general que no están pensados para ejecutarse en el entorno de un servidor web. Por ejemplo, no se reutilizan los procesos para atender a varias peticiones: por cada petición que se haga al servidor web, se debe ejecutar un nuevo proceso. Además los programas no son portables entre distintas plataformas.*
- *Los lenguajes compilados a código intermedio ofrecen un equilibrio entre las dos opciones anteriores. Su rendimiento es muy bueno y pueden portarse entre distintas plataformas en las que exista una implementación de la arquitectura (como un contenedor de servlets o un servidor de aplicaciones Java EE).*

1.7.9 *Frameworks de desarrollo web:*

Un framework web es una estructura o un conjunto de herramientas predefinidas que proporciona una base para la construcción de aplicaciones web. Está diseñado para ayudar a los desarrolladores a crear sitios web de manera más eficiente y consistente, al proporcionar una serie de componentes y funciones comunes que se utilizan con frecuencia en el desarrollo web.

Framework	Descripción
Spring	El *framework* más extendido para JEE.
Ruby on Rails	Muy popular para el desarrollo MVC, ha influenciado otros muchos *frameworks* extendidos. En Ruby.
Django	Basado en Python, sigue el modelo MVT.
AngularJS	*Framework* de Google para el lado del cliente basado en JavaScript.
Symfony	*Framework* para el desarrollo MVC en PHP.

1.7.10 Código embebido en el lenguaje de marcas.

En la actualidad, una de las principales formas de realizar páginas web dinámicas, es integrar el código del programa en medio de las etiquetas HTML de la página web. De esta forma, el contenido que no varía de la página se puede introducir directamente en HTML, y el lenguaje de programación se utilizará para todo aquello que pueda variar de forma dinámica.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
 <head>
  <title>Prueba PHP</title>
  <meta http-equiv = "Content-type" content = "text/html; charset=utf-8">
 </head>
 <body>
 Prueba de página en PHP. </br>
 El nombre del servidor es: <?php echo $_SERVER['SERVER_NAME']; ?>
 </body>
</html>
```

Esta metodología de programación es la que se emplea en los lenguajes ASP, PHP y con las páginas JSP de Java EE.
Los servlets de Java EE se diferencian de las páginas JSP en que los primeros son programas Java compilados y almacenados en el contenedor de servlets. Sin embargo, las páginas JSP contienen código Java embebido en lenguaje HTML y se almacenan de forma individual en el servidor web. La primera vez que se necesita una página JSP, se convierte a un servlet y éste se guarda para ser utilizado en posteriores llamadas a la misma página.

1.7.11 Herramientas de programación.

Desde hace tiempo, existen entornos integrados de desarrollo (IDE) que agrupan en un único programa muchas de estas herramientas. Algunos de estos entornos de desarrollo son específicos de una plataforma o de un lenguaje, como sucede por ejemplo con ***Visual Studio****, el IDE de Microsoft para desarrollar aplicaciones en lenguaje C# o Visual Basic para la plataforma .Net. Otros como* ***Eclipse*** *o* ***NetBeans*** *te permiten personalizar el entorno para trabajar con diferentes lenguajes y plataformas, como Java EE o PHP.*

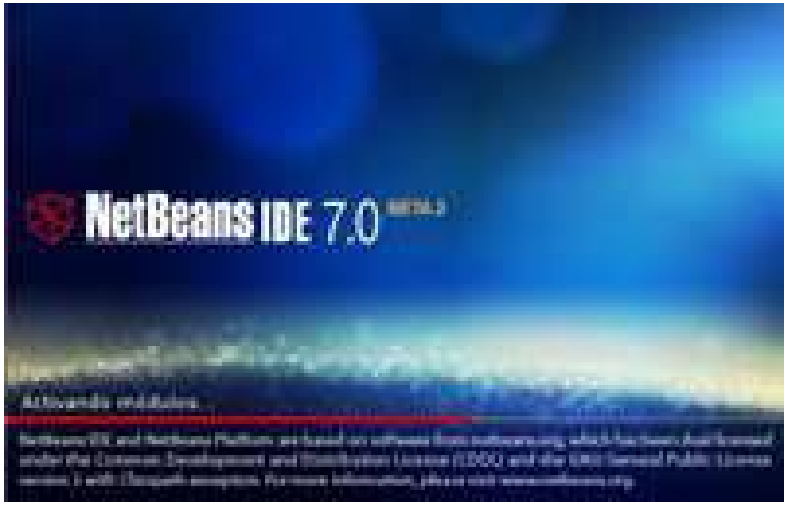
No es imprescindible utilizar un IDE para programar. En muchas ocasiones puedes echar mano de un simple editor de texto para editar el código que necesites. Sin embargo, si tu objetivo es desarrollar una aplicación web, las características que te aporta un entorno de desarrollo son muy convenientes.

Entre estas características se encuentran:

- ***Resaltado de texto****. Muestra con distinto color o tipo de letra los diferentes elementos del lenguaje: sentencias, variables, comentarios, etc. También genera indentado automático para diferenciar de forma clara los distintos bloques de un programa.*
- ***Completado automático****. Detecta qué estás escribiendo y cuando es posible te muestra distintas opciones para completar el texto.*
- ***Navegación en el código****. Permite buscar de forma sencilla elementos dentro del texto, por ejemplo, definiciones de variables.*
- ***Comprobación de errores al editar****. Reconoce la sintaxis del lenguaje y revisa el código en busca de errores mientras lo escribes.*

- ***Generación automática de código**. Ciertas estructuras, como la que se utiliza para las clases, se repiten varias veces en un programa. La generación automática de código puede encargarse de crear la estructura básica, para que sólo tengas que rellenarla.*
- ***Ejecución y depuración.** Esta característica es una de las más útiles. El IDE se puede encargar de ejecutar un programa para poder probar su funcionamiento. Además, cuando algo no funciona, te permite depurarlo con herramientas como la ejecución paso a paso, el establecimiento de puntos de ruptura o la inspección del valor que almacenan las variables.*
- ***Gestión de versiones.** En conjunción con un sistema de control de versiones, el entorno de desarrollo te puede ayudar a guardar copias del estado del proyecto a lo largo del tiempo, para que si es necesario puedas revertir los cambios realizados. Integración con sistemas de versiones.*

Nosotros usaremos:



1.7.12 Programación web con Java.

Java es el lenguaje de programación más utilizado hoy en día. Es un lenguaje orientado a objetos, basado en la sintaxis de C y C++ y eliminando algunas características de éstos que daban lugar a errores de programación, como los punteros. Todo el código que escribas en Java debe pertenecer a una clase.

El código fuente se escribe en archivos con extensión .java. El compilador genera por cada clase un archivo .class. Para ejecutar una aplicación programada en Java necesitamos tener instalado un entorno de ejecución (JRE). Para crear aplicaciones en Java necesitamos el kit de desarrollo de Java (JDK), que incluye el compilador.

Existen básicamente dos tecnologías que te permiten programar páginas web dinámicas utilizando Java EE: servlets (clases Java compiladas que contienen instrucciones de salida para generar las etiquetas HTML de las páginas) y JSP (páginas web que contienen instrucciones para añadir contenido de forma dinámica).

Aunque no es así en todos los casos, la mayoría de implementaciones disponibles para JSP compilan cada página y generan un servlet a partir de la misma la primera vez que se va a ejecutar. Este servlet se almacena para ser usado en futuras peticiones.

El problema de utilizar servlets directamente es que, aunque son muy eficientes, son muy tediosos de programar puesto que hay que generar la salida en código HTML con gran cantidad de funciones como println. Este problema se resuelve fácilmente utilizando JSP, puesto que aprovecha la eficiencia del código Java, para generar el contenido dinámico, y la lógica de presentación se realiza con HTML normal.

De esta forma estas dos tecnologías se suelen combinar para crear aplicaciones web. Los servlets se encargan de procesar la información y obtener resultados, y las páginas JSP se encargan del interface, incluyendo los resultados obtenidos por los servlets dentro de una página web.