

UNIDAD 1: Plataformas de programación web en entornos servidor. Aplicaciones LAMP.

Contenidos:

1. Características de la programación web.

1.1 Páginas web estáticas y dinámicas.

1.2 Ejecución de código en el servidor web y en el cliente.

2. Tecnologías para programación web del lado del servidor

2.1 Arquitecturas y plataformas.

2.2 Integración con el servidor web.

3. Lenguajes.

3.1 Código embebido en el lenguaje de marcas.

3.2 Herramientas de programación

3.3 Programación web con Java.

3.4 Programación web con PHP

1. Características de la programación web.

El contenido que ves cuando descargas una página web en tu ordenador está programado

en lenguaje marcado, formado por etiquetas HTML o XHTML.

La información que le indica al navegador el estilo con el que se muestra cada parte de la página estará almacenado en otro fichero, hoja de estilos o CSS.

Pasos de comunicación cliente-servidor:

1. Tu ordenador solicita a un servidor web una página
- 2.El servidor busca esa página
- 3.Si el servidor encuentra esa página la recupera.
4. Se la envía al navegador y este muestra su contenido.

1.1 Páginas web estáticas y dinámicas.

Página web estática:

Se encuentran almacenadas de forma definitiva.

Su contenido no varía.

Se usan para mostrar información en concreto.

Solo se puede modificar su contenido si un programador modifica y actualiza este.

Páginas web dinámicas:

Su contenido cambia en función de diversas variables(El navegador que uses, el usuario con el que te registras, o lo que hayas realizado anteriormente)

Hay dos tipos de páginas web dinámicas.

Las que incluyen código que ejecuta el navegador:

El código ejecutable, en JavaScript, se incluye dentro del HTML o XHTML y se descarga junto con la página.

Las que el HTML de esas páginas se forma como resultado de la ejecución de un programa.

Una página web dinámica funciona así:

1. El cliente solicita a un servidor web una página web.
- 2- El servidor busca la página
3. 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.
5. El resultado de la ejecución será una página en formato HTML
6. El servidor web envía el resultado al navegador.

1.1.1 Ventajas de las páginas web estáticas.

- No es necesario saber programar para crear un sitio de solo páginas estáticas. (Como mucho HTML XHTML y CSS pero también puedes usar programas de diseño web)
- Su contenido nunca varía.
- Siempre está indexada a diferencia de las dinámicas, que su indexación se puede estropear.
- Para que un servidor pueda procesar una página web dinámica, necesita ejecutar un programa. Esto puede requerir recursos del servidor.
- Las páginas web estáticas solo necesitan un servidor web que se comunique con tu navegador.

Las páginas estáticas tienen limitaciones.

La desventaja más importante es la actualización de su contenido, la cual debe hacerse de forma manual.

Las primeras páginas web fueron estáticas.

1.1.1.1 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.

Ventajas de las aplicaciones web:

- No es necesario instalarlas en aquellos equipos en que se vayan a usar.(Solo se instalan en el servidor. Se utiliza de forma simultánea desde muchos equipos)
- Se pueden utilizar en todos los sistemas que dispongan de un navegador web independientemente de sus características.
- Se pueden utilizar desde cualquier lugar en el que dispongamos conexión con el servidor.

Desventajas de las aplicaciones web.

- El interfaz de usuario de las apps web es la página que se muestra en el navegador(Restringe características del interfaz a aquellas de una página web)
- Dependemos de una conexión con el servidor para poder utilizarlas
- La información que se muestra en el navegador debe transmitirse desde el servidor.(Algunas apps no son adecuadas para su implementación como app web por culpa de esto)

El contenido de una app web se almacena en una base de datos(Drupal, Joomla!)

Parte externa o front-end:

Conjunto de páginas que ven la gran mayoría de los usuarios que las usan.

Parte interna o back-end:

Conjunto de páginas dinámicas que usan las personas que producen el contenido y las que administran al aplicación web.

1.2 Ejecución de código en el servidor y en el cliente.

En el caso de que el navegador solicite al servidor web una página y ésta haya tenido que ejecutar un programa para obtenerla, **el código se ejecuta en el entorno del servidor web.**

Cuando una página llega a tu navegador, puede incluir algún programa o fragmento de código ejecutable, normalmente en JavaScript.

Este se ejecutará en tu navegador y puede modificar el contenido de tu página(también puede animar textos u objetos de la página o comprobar los datos de un formulario.)

Estas dos tecnologías se complementan una con otra.

Esto es así porque el código que se ejecuta en el cliente web no tenía acceso a los datos que se almacenan en el servidor.

Existe una técnica de desarrollo web conocida como AJAX, que nos posibilita realizar programas en los que el código JavaScript se pueda comunicar con un servidor de Internet para poder modificar la página web.

Muchas de las aplicaciones web actuales utilizan estas dos tecnologías:

La ejecución de código en el servidor y en el cliente.

2. Tecnologías para programación web del lado del servidor.

Los componentes principales para ejecutar aplicaciones web en un servidor son:

1. Un servidor web para recibir las peticiones de los clientes web y enviarles la página que solicitan.
2. El módulo encargado de ejecutar el código o programa y generar la página web resultante.
3. Una aplicación de base de datos, que normalmente también será un servidor.
4. El lenguaje de programación que utilizarás para desarrollar las aplicaciones.

Hay que decidir como se organizar el código

Muchas arquitecturas estructuran el código en capas o niveles.

Dividimos en capas el diseño de una app para separar las funciones lógicas de la misma, de forma que sea posible ejecutar cada una en un servidor distinto.

En una app se distingue:

Funciones de presentación:

Da formato a los datos para presentárselo al usuario.

Lógica:

Utiliza los datos para ejecutar un proceso y obtener un resultado.

Persistencia:

Mantiene los datos ordenados.

Acceso:

Obtiene e introduce datos en el espacio de almacenamiento.

En las aplicaciones de 3 capas nos podemos encontrar con:

-Capa cliente: Dónde se programa todo lo relacionado con el interfaz de usuario.

-Capa intermedia: Dónde se programa la funcionalidad de la aplicación.

-Capa de acceso a datos: Se encarga de almacenar la información de la aplicación en una base de datos y recuperarla cuando sea necesario.

2.1 Arquitecturas y plataformas:

-Java EE(Enterprise edition) o (J2EE):

Plataforma orientada a la programación de aplicaciones en lenguaje Java.

Funciona con distintos gestores de bases de datos e incluye varias librerías y especificaciones para el desarrollo de aplicaciones de forma modular.

Apoyada por empresas como Sun , Oracle o IBM.

Buena solución para aplicaciones de tamaño medio o grande.

Una de sus principales ventajas es la multitud de librerías que existen en ese lenguaje y la gran base de programadores que lo conocen.

-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 corresponde con el lenguaje de programación utilizado.

Dependiendo del SO que se use para el servidor se utilizan las siglas LAMP(Linux), WAMP(Windows) o MAMP(Mac)

Todos los componentes de esta arquitectura son de código libre(open source)

Permite desarrollar aplicaciones de tamaño pequeño o mediano con aprendizaje sencillo.

Su gran ventaja es la gran comunidad que la soporta y la multitud de aplicaciones de código libre disponibles

CGI/Perl:

Combina Perl(potente lenguaje de código libre creado para la administración de servidores) y CGI(estándar para permitir al servidor web ejecutar programas genéricos, escritos en cualquier lenguaje).

Devuelven páginas web(HTML) como resultado de su ejecución.

Es la más primitiva de las 4 arquitecturas.

Su principal desventaja es que CGI es lento, aunque se puede acelerar.

Por otra parte, Perl es un lenguaje muy potente con una amplia comunidad de usuarios y mucho código libre disponible.

ASP.Net:

Es la arquitectura comercial propuesta por Microsoft para el desarrollo de aplicaciones.

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, ISS y puede obtener información de varios gestores de base de datos.

Una de las mayores ventajas de la arquitectura .Net es que incluye todo lo necesario para el desarrollo y el despliegue de apps web.

Por ejemplo tiene su propio IDE, Visual Studio.

La mayor desventaja es que se trata de una plataforma comercial de código propietario

2.1.1 Selección de una arquitectura de programación web.

Página 20 del libro, no encuentro nada para el examen si alguien le saca algo que lo diga.

2.2 Integración con el servidor web.

La comunicación entre un cliente web o navegador y un servidor web se lleva cabo gracias al protocolo HTTP.

En el lado de la app web, HTTP es el vínculo que une al usuario y a la aplicación.

En el lado del servidor, estas peticiones las procesa el servidor HTTP(servidor web).

Es el que se encarga de decidir de como procesar las peticiones que recibe.

Guión CGI: Un programa externo que utiliza CGI a la hora de procesar una página web.

Cada vez que se ejecuta un guión CGI, el SO debe de crear un nuevo proceso.(Mayor consumo de recurso y menor velocidad de ejecución).

Hay otros métodos para ejecutar guiones como el módulo mod_perl(Ejecuta en Apache guiones programados en Perl), mod_php(Ejecuta guiones en PHP utilizando plataformas AMP) y mod_python(el equivalente del mod_php pero en Python)

3. Lenguajes.

Hay 3 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.

Necesita de un intérprete para que procese las líneas del programa y genere una página web.

A este grupo pertenecen Perl, Python y ASP.

Lenguajes compilados a código nativo:

Aquellos a los que el código fuente se traduce a código binario antes de ser ejecutado.

El servidor web almacena los programas en modo binario y los ejecuta directamente cuando los invoca.

El método principal para ejecutar programas binarios desde un servidor web es CGI.(Usando CGI podemos ejecutar el código programado en cualquier lenguaje de propósito general, como C)

Lenguajes compilados a código intermedio:

Lenguajes en los que el código fuente original se traduce a un código intermedio, antes de ser ejecutado(no depende del procesador).

Es la forma en las que se ejecutan las aplicaciones programadas en Java.

En la programación web, operan así los lenguajes de las arquitecturas Java EE(servlets y páginas JSP) y ASP.Net

Procedimiento de compilación JIT: Forma en la que se convierte el código intermedio a código binario para ser ejecutado por el procesador.

Ventajas e inconvenientes de los lenguajes:

- Los lenguajes de guiones tienen la ventaja de que no es necesario traducir el código fuente original para ejecutarse.
- Los lenguajes de guiones tienen mas portabilidad.
- El proceso de interpretación de los lenguajes de guiones ofrecen un peor rendimiento que las otras alternativas.
- Los lenguajes compilados a código nativo son los de mayor velocidad de ejecución, pero tienen problemas en su integración con el servidor web.
- No se reutilizan procesos para atender a varias peticiones sino que va de uno a uno.
- Los lenguajes compilados a código nativo no son portables entre distintas plataformas.
- Los lenguajes compilados a código intermedio ofrecen un equilibrio.
- Su rendimiento es muy bueno y es portable mientras la plataforma tenga una implementación de su arquitectura.
(Contenedor de servlets o aplicaciones Java EE)

Otro enfoque consiste en integrar el código del programa en medio de las etiquetas HTML de la página web.

Así el contenido que no varía de la página se puede introducir directamente en HTML, y el lenguaje de programación se usará para todo lo que sea dinámico.

Los servlets de Java EE son programas compilados y almacenados en el contenedor de servlets mientras que las páginas JSP contienen código Java embebido en lenguaje HTML y se almacenan de forma individual en el servidor web.

En la arquitectura ASP.Net cada página se divide en dos ficheros, uno contiene las etiquetas HTML y otro el código en el lenguaje de programación usado.

Con esto se logra una independencia entre el aspecto de la app y la gestión del contenido dinámico.

A partir de esto se obtiene un código intermedio que almacena el servidor.

3.2 Herramientas de programación

Debemos de tener en cuenta con que herramientas contamos para poder ayudarnos de una forma u otra.

Para eso tenemos los IDE que nos ayudan en muchas cosas.

Entre otras cosas, los IDE tienen las siguientes características que nos facilitan nuestro trabajo:

Resaltado de texto:

Muestra con distinto color o tipo de letra los diferentes elementos del lenguaje: sentencias, variables, comentarios, etc.

Completado automático:

Detecta que estás escribiendo y te muestra distintas opciones para completar el texto.

Navegación en el código:

Permite buscar de forma sencilla elementos dentro del texto.

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:

Se encarga de crear estructuras básicas, para que solo la rellenemos.

Ejecución y depuración:

El IDE puede encargarse de ejecutar un programa para poder probar su funcionamiento.

Además, si algo no funciona, puedes depurarlo.

Gestión de versiones:

Te permite ayudar a guardar copias del estado del proyecto a lo largo del tiempo, para que si es necesario puedas revertir los cambios.

Los IDE de código abierto más utilizados son Eclipse y NetBeans.

3.2.1 Instalación de NetBeans para PHP en Linux.

Java- version

En caso de no tener Java, deberemos instalarlo antes del IDE.

Actualizamos los repositorios de java

```
sudo add-apt-repository ppa:webupd8team/java
```

Instalamos java

```
sudo apt-get update
```

```
sudo apt-get install oracle-java8-installer
```

Nos descargamos netbeans desde su pagina oficial.

3.2.2 Instalación de LAMP

En la terminal, ponemos el siguiente comando:

```
sudo apt install lamp-server^
```

3.3 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 estos que daban errores.

Hay dos formas que te permiten programar usando Java EE:

servlets(clases Java compiladas que contienen instrucciones de salida para generar las etiquetas HTML de las páginas)

JSP(páginas web que contienen instrucciones para añadir contenido de forma dinámica).

3.4 Programación web con PHP

PHP es un lenguaje de guiones de propósito general, pero diseñado para el desarrollo de páginas web dinámicas utilizando código embebido dentro del lenguaje de marcas.

Su sintaxis está basada en la de C/ C++ y por ello es muy similar a Java.

Los delimitadores recomendados para incluir código PHP dentro de una página web son `<?php` y `?>`

El código se ejecuta por un entorno de ejecución con el que está integrado el servidor web.

La configuración del server se realiza por ficheros de configuración

El de Apache `httpd.conf` o `apache2.conf`

El de PHP es `php.ini`

`phpinfo()` te informa el lugar en el que `php.ini` se encuentra en tu ordenador

Algunas de las directivas más utilizadas que figuran en el fichero php.ini son:

short_open_tag: Indica si se pueden utilizar en PHP los delimitadores cortos <? y ?>.(Mejor no usarlos, porque puede causar problemas si usamos páginas con XML)

max_execution time: Permite que puedes ajustar el número máximo de segundos que podrá durar la ejecución de un script PHP.(También evita que el servidor se bloquee si se produce algún error en un script)

error_reporting: Indica qué tipo de errores se mostrarán en el caso de que se produzcan.(error_reporting = E_ALL para mostrar todos los tipos de errores)

file_uploads: Indica si se pueden o no subir ficheros al server HTTP

upload_max_filesize: Si se pueden subir ficheros por HTTP, indica el límite máximo permitido para el tamaño de cada archivo.

(Copiado de la diapositiva 38)

A medida que vayas escribiendo código en PHP, será útil que introduzcas en el mismo algunos comentarios que ayuden a revisarlo cuando lo necesites. En una página web los comentarios al HTML van entre los delimitadores `<!--` y `-->`. Dentro del código PHP, hay tres formas de poner comentarios:

Comentarios de una línea utilizando `//`. Son comentarios al estilo del lenguaje C. Cuando una línea comienza por los símbolos `//`, toda ella se considera que contiene un comentario, hasta la siguiente línea.

Comentarios de una línea utilizando `#`. Son similares a los anteriores, pero utilizando la sintaxis de los scripts de Linux.

Comentarios de varias líneas. También iguales a los del lenguaje C. Cuando en una línea aparezcan los caracteres `/*`, se considera que ahí comienza un comentario. El comentario puede extenderse varias líneas, y acabará cuando escribas la combinación de caracteres opuesta: `*/`.

Recuerda que cuando pongas comentarios al código PHP, éstos no figurarán en ningún caso en la página web que se envía al navegador (justo al contrario de lo que sucede con los comentarios a las etiquetas HTML).

3.4.1 Variables y tipos de datos PHP

Para crear una variable en php:

```
$mi_variable= 7;
```

Dependiendo del tipo de valor que le asignemos cambiará su contenido.

/ Si le cambiamos el contenido

```
$mi_variable = “siete”;
```

// La variable puede cambiar de tipo

// En este caso pasa a ser de tipo “cadena”

Los tipos de datos simples en PHP son:

booleano(boolean): true o false (podemos poner cualquier número entero y se considera true excepto el 0 que es false)

entero(integer): Cualquier número sin decimales

real(float): Cualquier número con decimales

cadena (string). Conjuntos de caracteres delimitados por comillas simples o dobles.

null. Es un tipo de datos especial, que se usa para indicar que la variable no tiene valor.

La conversión automática que realiza PHP no siempre es lo que queremos.

PHP permite otras conversiones implícitas de tipos :

- (int) : Fuerza la conversión a entero
- (real), (double), (float): Fuerza la conversión a coma flotante.
- (string): Fuerza la conversión a cadena de caracteres.
- (array): Fuerza la conversión a matriz
- (object): Fuerza la conversión a un objeto.

3.4.2 Expresiones y operadores.

Como en muchos otros lenguajes, en PHP se utilizan las expresiones para realizar acciones dentro de un programa

Hay tipos de expresiones como:

++(sumar 1 a la variable)

Además del operador =, existen operadores con los que realizar operaciones y asignaciones en un único paso (+=, -=,...).

Comparar operandos. Además de los que nos podemos encontrar en otros lenguajes (>, >=,...), en PHP tenemos dos operadores para comprobar igualdad (==, ===) y cuatro para comprobar diferencia (<>, !=, !== y <=>).

Los operadores <> y != son equivalentes. Comparan los valores de los operandos.

El operador === devuelve verdadero (true) sólo si los operandos son del mismo tipo y además tienen el mismo valor.

El operador !== devuelve verdadero (true) si los valores de los operandos son distintos o bien si éstos no son del mismo tipo

Comparar expresiones booleanas. Tratan a los operandos como variables booleanas (true o false).

Existen operadores para realizar un Y lógico (operadores and o &&), O lógico (operadores or o ||), No lógico (operador !) y O lógico exclusivo (operador xor)