

Desarrollo Web en entorno servidor

Unidad 1:

Introducción a la programación web en entorno servidor.

Tabla de contenido

1.	Modelos de programación en entornos cliente-servidor	2
1.1.	Páginas web.....	2
1.2.	Aplicaciones web.....	4
1.2.1.	Ventajas de las aplicaciones web:	4
1.2.2.	Desventajas de las aplicaciones web:.....	4
2.	Tecnologías para programación web del lado del servidor.	5
2.1.	Tipos de arquitecturas y plataformas.	6
2.1.1.	Selección de una arquitectura de programación web.	7
3.	Lenguajes	8
3.1.	Código embebido en el lenguaje de marcas.	9
4.	Nuestro entorno de trabajo.....	10
5.	Resumen	12
	Glosario.....	12

1. Modelos de programación en entornos cliente-servidor

Las aplicaciones y páginas web se basan en el modelo cliente-servidor. Es un modelo distribuido en el que hay dos tipos de elementos con funciones diferenciadas; clientes y servidores. Los servidores provén servicios, como información o funcionalidad, a los clientes. Es el cliente el que inicia el proceso al enviar una solicitud al servidor que, a su vez, envía un mensaje de respuesta.

Cliente y servidor se comunican mediante un protocolo que define el formato de solicitudes y respuestas y, en general, se ejecutan en máquinas diferentes conectadas por red, aunque también pueden ejecutarse en el mismo ordenador.

En el caso de los servidores web, los clientes son, la mayoría de las veces, los navegadores que realizan peticiones de páginas al servidor. Este responde con la página web solicitada o con un mensaje de error si no la encuentra o el acceso no está autorizado.

La comunicación entre clientes y servidores web se realiza mediante protocolo HTTP su versión segura, HTTPS. Las características del modelo cliente-servidor y del protocolo HTTP condicionan el desarrollo de aplicaciones web.

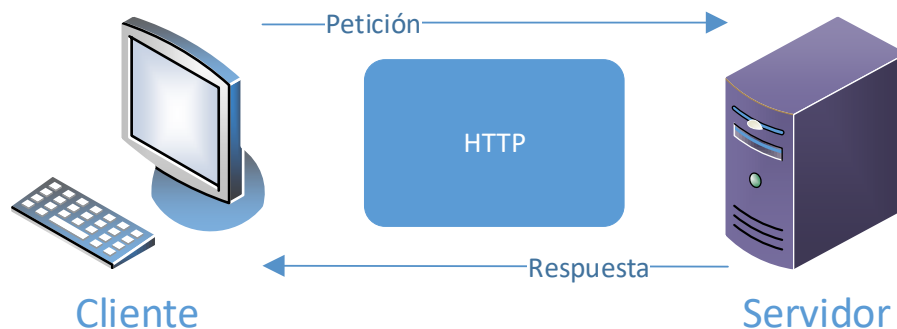


Figura 1. - Modelo cliente-servidor

1.1. Páginas web

Cuando una página web se descarga en el cliente, su contenido define qué se debe mostrar en pantalla. Este contenido está programado en un lenguaje de marcado, formado por etiquetas, que puede ser HTML o XHTML. Las etiquetas que componen la página indican el objetivo de cada una de las partes que la componen. Así, dentro de estos lenguajes hay etiquetas para indicar que un texto es un encabezado, que forma parte de una tabla, o que simplemente es un párrafo de texto.

Además, si la página está bien estructurada, la información que le indica al navegador el estilo con que se debe mostrar cada parte de la página estará almacenado en otro fichero, una hoja de estilos o CSS¹. La hoja de estilos se encuentra indicada en la página web y el navegador la descarga junto a ésta. En ella nos podemos encontrar, por ejemplo, estilos que indican que el encabezado debe ir con tipo de letra Arial y en color rojo, o que los párrafos deben ir alineados a la izquierda.

¹ Abreviatura de "Hoja de estilos en cascada", del inglés Cascading Style Sheet (CSS). Es un lenguaje utilizado para definir las características de presentación de un documento escrito en lenguaje HTML, XHTML o XML

Bajo el modelo Cliente-Servidor, cuando visitamos una página web poniendo su dirección en el navegador ocurren los siguientes pasos

1. El ordenador solicita a un servidor web una página con extensión .htm, .html o .xhtml.
2. El servidor busca esa página en un almacén de páginas.
3. Si el servidor encuentra esa página, dependiendo del tipo de página ocurre:
 - 3.1 Si es estática, sigue el siguiente esquema:

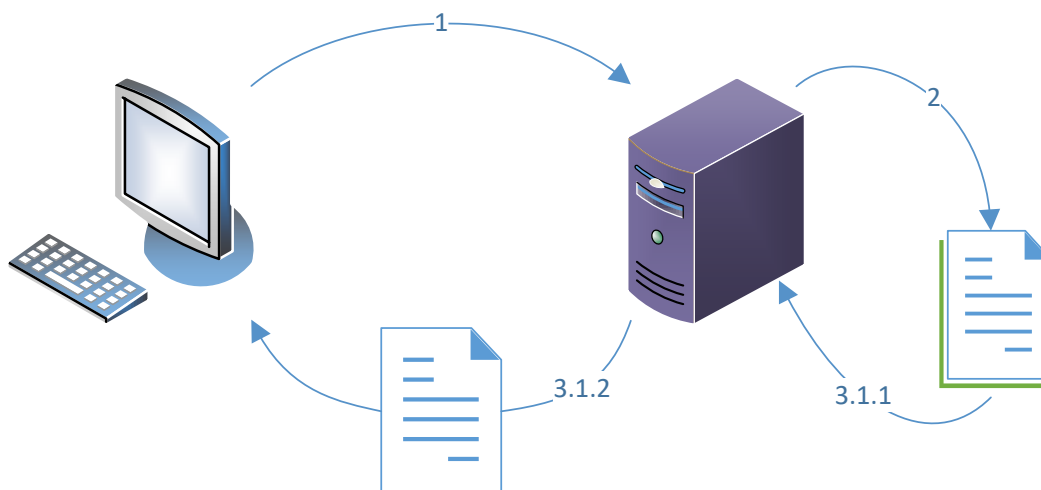


Figura 2. – Página web estática

3.1.1. La recupera.

3.1.2. La envía al navegador para que éste pueda mostrar su contenido.

3.2. Si es dinámica, sigue el siguiente esquema:

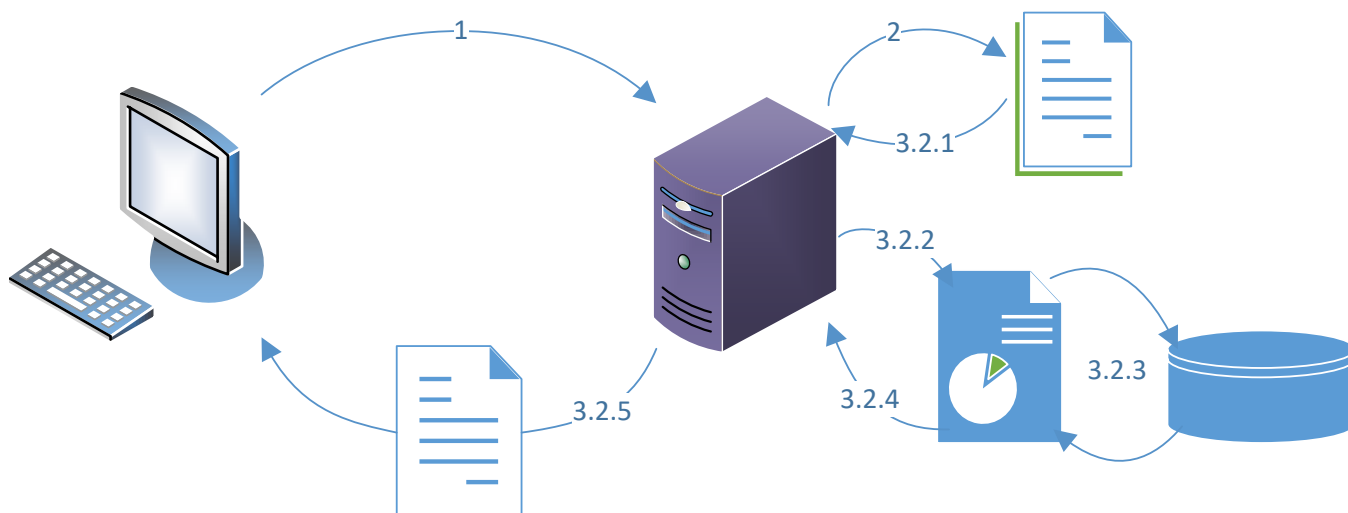


Figura 3 – Página web dinámica

3.2.1. La recupera.

3.2.2. El servidor web contacta con el módulo responsable de ejecutar el código y se lo envía.

3.2.3. 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.

3.2.4. El resultado de la ejecución será una página en formato HTML, similar a cualquier otra página web no dinámica.

3.2.5. El servidor web envía el resultado obtenido al navegador, que la procesa y muestra en pantalla.

1.2. 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. Podemos encontrar aplicaciones web para realizar múltiples tareas. Unas de las primeras en aparecer fueron los clientes de correo, que 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.

1.2.1. 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 de forma 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, etc.
- Se pueden utilizar en todos aquellos sistemas que dispongan de un navegador web, independientemente de sus características 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.

1.2.2. Desventajas de las aplicaciones web:

- La interface de usuario de las aplicaciones web es la página que se muestra en el navegador. Esto restringe las características de la 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).

2. 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.

Además de los componentes a utilizar, también es importante decidir cómo vamos a organizar el código de la aplicación. Muchas de las arquitecturas² que se usan en la programación de aplicaciones web te ayudan a estructurar el código de las aplicaciones en capas o niveles.

La arquitectura en tres capas es una ampliación del modelo cliente-servidor. La lógica de aplicación se separa en:

- Capa de presentación. Para la interfaz de usuario. Muestra información y permite interactuar con el sistema al usuario.
- Capa de negocio. Para la lógica propia de la aplicación. Se comunica con las otras capas.
- Capa de datos. Para gestionar la base de datos.



Figura 4. –Arquitectura en tres capas

² Estructura en la que se organiza y relacionan cada uno de los componentes necesarios para ejecutar un sistema informático o parte del mismo.

La capa de presentación no se comunica directamente con la de datos, sino que se hace a través de la capa de negocio. La capa de negocio recibe las acciones del usuario de la capa presentación y, si es necesario, se comunica con la capa de datos para consultar o modificar la base de datos. De la misma manera se produce la salida de la capa de datos a la de presentación.

En una aplicación web, la capa de presentación se ejecuta en un navegador, en el ordenador del usuario; la capa de negocio, en el servidor web, y la de datos, en el servidor de bases de datos. Estas dos pueden estar en equipos diferentes o en el mismo.

El objetivo es desacoplar la lógica de negocio de la de presentación para conseguir código más reutilizable. Además, mediante la separación entre componentes se obtienen aplicaciones más fáciles de mantener y ampliar.

Por ejemplo, al hacer *login* en una aplicación web

- Desde el cliente (navegador) se envía un formulario. Es la capa de presentación.
- En servidor, capa de negocio, se consulta con la base datos (capa de datos) si los datos son correctos.
- En función de la respuesta, se pasan las instrucciones correspondientes a la capa de presentación (se permite el acceso o se muestra un mensaje de error).

Actividad 1

Consulta en la página del W3C (www.w3.org) los estándares referidos al diseño de páginas y aplicaciones web. En particular, elabora un documento en formato pdf de no más de 5 hojas en el que recojas ejemplos de situaciones o estándares de accesibilidad en la web.

2.1. Tipos de arquitecturas y plataformas.

La primera elección antes de comenzar a programar una aplicación web es la arquitectura que vamos a utilizar. Hoy en día, puedes elegir entre:

- 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). Las siglas XAMPP hacen referencia a cualquier

sistema operativo, incluyendo el lenguaje Perl, a mayores del lenguaje PHP. 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. 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.

- CGI/Perl. Es la combinación de dos componentes: Perl, un potente lenguaje de código libre creado originalmente para la administración de servidores, y CGI, un estándar para permitir al servidor web ejecutar programas genéricos, escritos en cualquier lenguaje (también se pueden utilizar por ejemplo C o Python), que devuelven páginas web (HTML) como resultado de su ejecución. Es la más primitiva de las arquitecturas que comparamos aquí.

El principal inconveniente de esta combinación es que CGI es lento, aunque existen métodos para acelerarlo. 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. 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.

2.1.1. Selección de una arquitectura de programación web.

Como hemos visto, hay muchas decisiones que debemos tomar antes aún de comenzar el desarrollo de una aplicación web. La arquitectura que utilizaremos, el lenguaje de programación, el entorno de desarrollo, el gestor de bases de datos, el servidor web, incluso cómo se estructurará la aplicación.

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

- ¿Qué tamaño tiene el proyecto?
- ¿Qué lenguajes de programación conocemos? ¿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?
- ¿Vamos a programar la aplicación solos o formaré parte de un grupo de programadores?
- ¿Contamos con algún servidor web o gestor de base de datos disponible o puedo decidir libremente utilizar el que crea necesario?
- ¿Qué tipo de licencia vamos a aplicar a la aplicación que desarrollemos?

Estudiando las respuestas a éstas preguntas, podemos ver qué arquitecturas se adaptan mejor a la aplicación y cuáles no son viables.

3. Lenguajes

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. Distinguimos tres grandes grupos:

- **Lenguajes de guiones.** Son aquellos en los que los programas se ejecutan directamente a partir de su código fuente³ 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. Pertenecen a este grupo Perl, Python y PHP.

- **Lenguajes compilados a código nativo**⁴. 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 es CGI. 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. En la programación web, operan de esta forma los lenguajes de las arquitecturas Java EE (servlets y páginas JSP) y ASP.Net.

En la plataforma ASP.Net y en muchas implementaciones de Java EE, se utiliza un procedimiento de compilación JIT. Este término hace referencia a la forma en que se convierte el código intermedio a código binario para ser ejecutado por el procesador. Para acelerar la ejecución, el compilador puede traducir todo o parte del código intermedio a código nativo cuando se invoca a un programa. El código nativo obtenido suele almacenarse para ser utilizado de nuevo cuando sea necesario.

Cada una de estas formas de ejecución del código por el servidor web tiene sus 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.

³ Conjunto de instrucciones que componen un programa, y que no son ejecutables directamente, sino que deben traducirse utilizando un compilador, intérprete o similar antes de que pueda ser ejecutado por la máquina

⁴ Denominación habitual del lenguaje máquina. Código que puede ser ejecutado directamente por el procesador

- **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).

3.1. Código embebido en el lenguaje de marcas.

Los lenguajes de programación web, a la hora de generar el código HTML, siguen dos estrategias distintas: integrar las etiquetas HTML en el código de los programas o integrar el código de los programas en medio de las etiquetas HTML.

Cuando **integramos las etiquetas HTML en el código de los programas**, estos incluyen dentro de su código sentencias de salida que son las que incluyen el código HTML de la página web que se obtendrá cuando se ejecuten. De esta forma se programan, por ejemplo, los guiones CGI y los servlets.

Por otro lado, cuando **integramos el código del programa en medio de las etiquetas HTML de la página web**, 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. Esta metodología de programación es la que se emplea en los lenguajes ASP, PHP y con las páginas JSP de Java EE.

Veamos un ejemplo.

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Número aleatorio</title>
5:   </head>
6:   <body>
7:     <?php
8:       echo rand(0, 100);
9:     ?>
10:  </body>
11: </html>
```

El bloque de líneas anterior contiene, dentro de una página HTML, un pequeño código en lenguaje PHP, el cual genera un número aleatorio entre 0 y 100, como vemos en las líneas 7, 8 y 9. Al solicitar la página, el servidor ejecuta el bloque y lo sustituye por su salida, en este caso un número aleatorio entre 0 y 100. Al solicitar la página, el servidor ejecuta el bloque y lo sustituye por su salida, en este caso un número aleatorio entre 0 y 100. Si sale un 8, enviará al cliente:

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Número aleatorio</title>
5:   </head>
6:   <body>
7:     8
8:   </body>
9: </html>
```

4. Nuestro entorno de trabajo

En nuestro entorno de trabajo vamos a utilizar XAMPP en el sistema operativo Windows 10, el lenguaje PHP y con el editor de código Visual Studio Code.

La descarga del paquete XAMPP se lleva a cabo desde la web <https://www.apachefriends.org/>. Para instalarlo solo hay que ejecutar el instalador con las opciones por defecto. Una vez instalado hay que arrancar los servidores desde el panel de control.

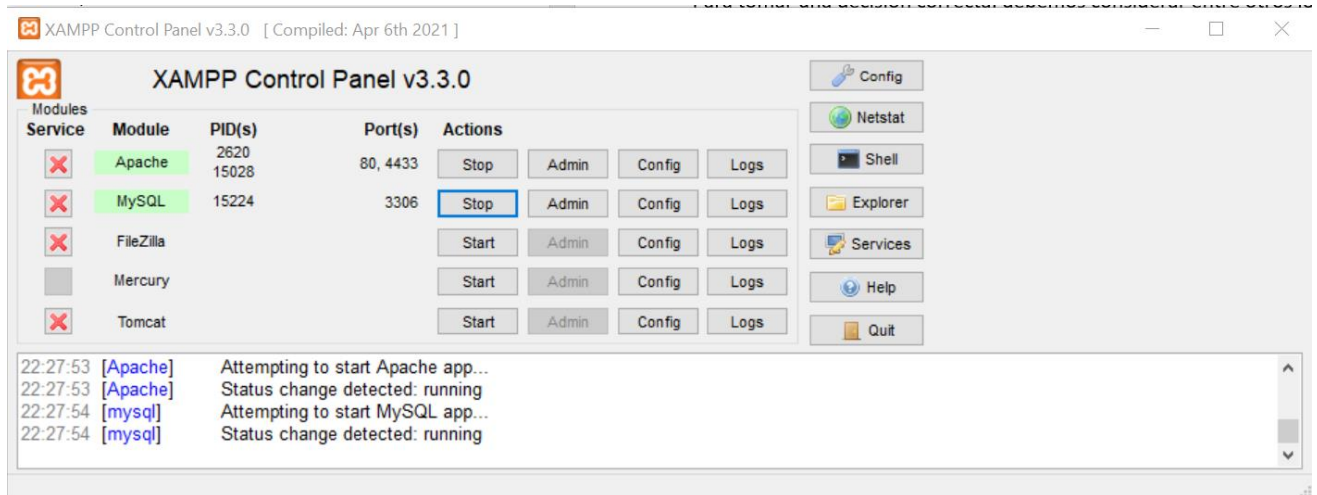


Figura 5. – Panel de control de XAMPP

Después de arrancar el servidor web se puede comprobar si la instalación ha sido correcta accediendo desde el navegador a "localhost"

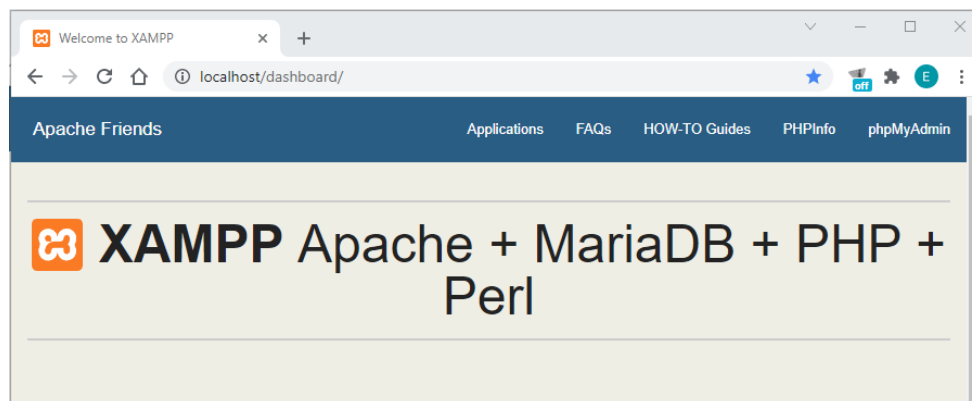


Figura 6. – Navegador visitando localhost

Para comprobar si se ha instalado bien la base de datos, se arranca desde el panel de control y se accede a <http://localhost/phpmyadmin/>. Es una aplicación web para manejar la base de datos incluida en XAMPP. Si no ha habido ningún problema, se mostrará una página como la recogida en la figura 1.5.



Figura 7. – Panel pdpmyadmin

Los archivos que queramos sean accesibles por cualquier navegador web hemos de alojarlos en la ruta C:\xampp\htdocs.

Para ello, accedemos a dicha ruta y creamos una nueva carpeta con las siglas de la asignatura: DWES, en esta carpeta es donde guardaremos los documentos *.php que trabajaremos durante el curso.

Si accedemos desde el navegador a la dirección: <http://localhost/DWES/> podremos ver el contenido de la carpeta siempre y cuando el servidor se encuentre encendido.

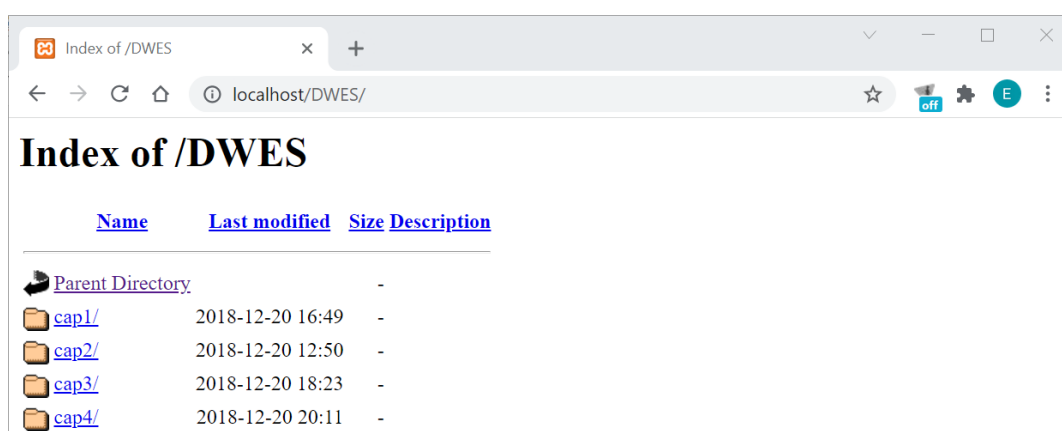


Figura 8. – Carpeta DWES del directorio localhost

Para instalar Visual Studio Code, accedemos a <https://code.visualstudio.com/>, descargamos el instalador y lo ejecutamos.

Nuestra área de trabajo estará ubicada en la ruta C:\xampp\htdocs\DWES. Por tanto, buscamos dicha carpeta en la opción **Abrir Carpeta**, en el menú **Archivo**.

5. Resumen

- Las aplicaciones web siguen la arquitectura cliente-servidor.
- El modelo a 3 capas es una evolución del modelo cliente-servidor.
- El modelo a 3 capas separa la lógica de las aplicaciones en tres partes para conseguir código más reusable.
- Las capas son: presentación, negocio y datos.
- El W3C mantiene varios de los estándares más importantes para el desarrollo web.
- La comunicación entre clientes y servidores se hace con el protocolo HTTP.
- Se pueden generar páginas dinámicas usando lenguajes de programación.
- El lenguaje de programación más extendido en el lado del servidor es PHP.
- El código del lenguaje se inserta dentro del HTML. El servidor se encarga de ejecutarlo antes de enviar la respuesta.
- XAMPP instala Apache con PHP y MySQL (MariaDB)

Glosario

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

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 W3C.

HTTPS: Versión segura del HTTP.

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

PHP: *Hypertext Preprocessor*, lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Servidor: Computadora o conjunto de computadoras que **almacenan, distribuyen y suministran información**. Atienden las peticiones de un cliente devolviendo la respuesta correspondiente. El cliente puede ser tanto un ordenador como una aplicación que requiere información del servidor para funcionar.

W3C: *World Wide Web Consortium*, organismo que elabora y mantiene varios de los estándares más importantes en Internet, como el HTTP o el HTML. <https://www.w3.org/Protocols/>