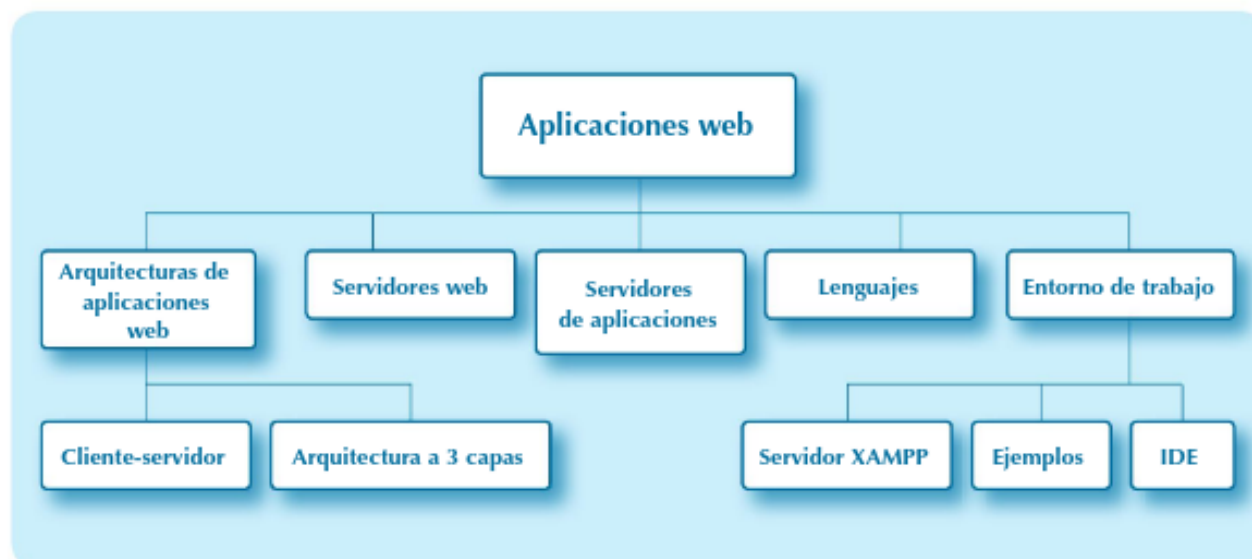


Introducción al desarrollo de aplicaciones web

Objetivos

- ✓ Conocer la arquitectura cliente-servidor.
- ✓ Presentar los modelos de programación en entornos cliente-servidor.
- ✓ Identificar los lenguajes más importantes para el desarrollo de aplicaciones web.
- ✓ Introducir los principales lenguajes y *frameworks* para el desarrollo en el lado del servidor.
- ✓ Entender cómo se integran los lenguajes de programación con los lenguajes de marcas y los servidores.
- ✓ Instalar el entorno de trabajo para ejecutar los ejemplos del libro.

Mapa conceptual



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 Environment*, 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.

1.1. Modelos de programación en entornos cliente-servidor

Las aplicaciones 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 proveen 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.



Figura 1.1
Modelo cliente-servidor.

La comunicación entre clientes y servidores web se realiza mediante protocolo HTTP o su versión segura, HTTPS, que tienen reservados los puertos 80 y ⁴⁴³553 respectivamente en la lista de puertos bien conocidos. Las características del modelo cliente-servidor y del protocolo HTTP condicionan el desarrollo de aplicaciones web.

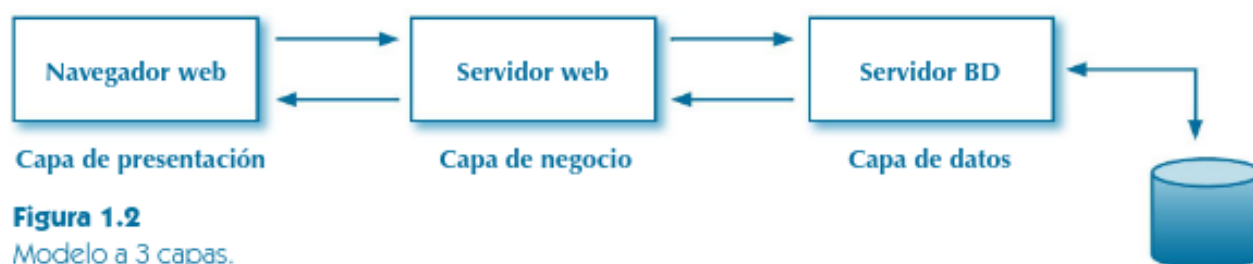
La arquitectura a tres capas es una ampliación del modelo cliente-servidor. La lógica de la aplicación se separa en:

- 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 dos capas.
- Capa de datos.* Para gestionar la base de datos.

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 de presentación y, si es necesario, se comunica con la capa de datos para consultar o modificar la base de datos. De la misma manera, pasa 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 propuesta 1.1



Visita la página del W3C (www.w3.org) para ver una lista de sus estándares. A lo largo del libro se utilizarán varios.

Investiga qué es un Estándar Web (*Web Standard*) y qué es una Recomendación (*Recommendation*).

1.2. Generación dinámica de páginas web

El lenguaje básico para la web es el HTML. Una página que esté escrita usando solo HTML será estática, es decir, mostrará siempre el mismo contenido. Cuando se utiliza un lenguaje de programación en el servidor se pueden generar páginas en función de la solicitud del cliente.

Por ejemplo, muchas páginas muestran anuncios diferentes según dónde esté el usuario. Es decir, no se ofrece el mismo contenido a todos los usuarios, sino que parte del contenido se genera dinámicamente. En este caso se habla de *páginas web dinámicas*.

Para ejecutar código en un servidor web hay varias opciones. Las más habituales son:

1. *Common Gateway Interface (CGI)*. Las peticiones de los clientes se pasan a un ejecutable en el servidor. Este programa genera la salida y el servidor se la pasa al cliente.
2. *Como módulo del servidor web*. En lugar de utilizar un programa externo, el propio servidor cuenta con un módulo, generalmente un intérprete, para ejecutar código.
3. *Servlets*. Los *servlets* de Java son objetos que reciben una petición y devuelven una respuesta en función de la petición. Para utilizar *servlets* hace falta un contenedor web, que es el que interactúa con ellos.

1.2.1. Lenguajes de programación en entornos servidor

Hay muchos lenguajes para el lado del servidor. Los más habituales son:

- a) PHP. Sin duda, el lenguaje más extendido en el lado del servidor. Es el lenguaje del que trata este libro. Normalmente se ejecuta como un módulo del servidor.
- b) JSP. La versión Java de PHP. Para utilizarlo hace falta un contenedor web.
- c) ASP.NET. La alternativa a PHP de Microsoft, integrada en la plataforma.NET.
- d) PERL. Muy utilizado para CGI. Es un lenguaje especialmente pensado para el procesamiento de expresiones regulares.
- e) Ruby. Es un lenguaje orientado a objetos muy apreciado por desarrolladores web.



Actividad propuesta 1.2

Busca información sobre otros lenguajes del lado del servidor y explica para qué se recomiendan.

También hay muchos *frameworks* para desarrollo de aplicaciones web (cuadro 1.1).

CUADRO 1.1
Frameworks para desarrollo web

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

1.2.2. Integración con los lenguajes de marcas

Las páginas dinámicas se componen de una parte estática en HTML y una parte dinámica en algún lenguaje de programación. Por eso se habla de lenguajes de plantillas.

Cuando se solicita una página, el servidor busca en ella bloques de código. Si los encuentra, los ejecuta y los sustituye por su salida. En PHP los bloques se delimitan por `<?php" y" ?>`.

Por ejemplo, en el fichero hay un bloque PHP en las líneas 7-9:

```
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>
```

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:

```
<!DOCTYPE html>
<html>
    <head>
        <title> Número aleatorio </title>
    </head>
    <body>
        8
    </body>
</html>
```

Actividad propuesta 1.3



Averigua qué caracteres se utilizan para insertar código JSP y ASP dentro de HTML.

1.2.3. Servidores de aplicaciones

Los servidores de aplicaciones ofrecen un entorno de ejecución para las aplicaciones. Su objetivo es liberar al programador de algunas tareas relacionadas con la infraestructura de la aplicación, como seguridad o balanceo de carga. Normalmente incluyen un servidor web.

Posiblemente, los más conocidos sean los servidores de aplicaciones para JEE, como JBoss o Websphere. Estos servidores tienen contenedores para ejecutar los componentes JEE, como contenedores de *servlets* y Java Enterprise Beans o persistencia. No todos los servidores dan soporte a todos los componentes.

1.3. Instalación del entorno de trabajo

Para poder probar los ejemplos del libro hay que:

1. Instalar un servidor web.
2. Instalar un servidor de bases de datos.
3. Copiar los ejemplos en el servidor e importar las bases de datos.

1.3.1. Servidores web y de base de datos

La instalación se puede realizar sencillamente con XAMPP. El paquete XAMPP incluye, entre otras cosas, un servidor web Apache con PHP y un servidor de bases de datos MySQL (ahora, MariaDB).

Recurso web

www

Puedes descargar XAMPP desde www.apachefriends.org/es/index.html.

Para instalarlo solo hay que ejecutar el instalador con las opciones por defecto.

Una vez instalado hay que arrancar los servidores. XAMPP también incluye un panel de control para arrancar y parar los servidores (figura 1.3).

Después de arrancar el servidor web se puede comprobar si la instalación ha sido correcta accediendo desde el navegador a “localhost” (figura 1.4).

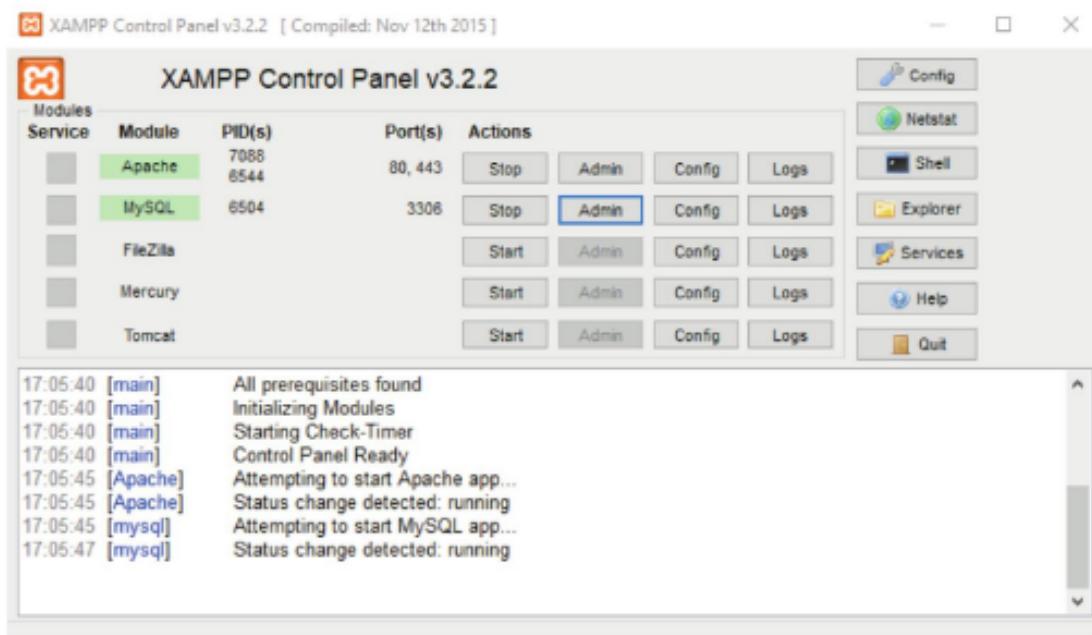


Figura 1.3
Panel
de control
de XAMPP.

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, pero con menos bases de datos en la sección de la izquierda.

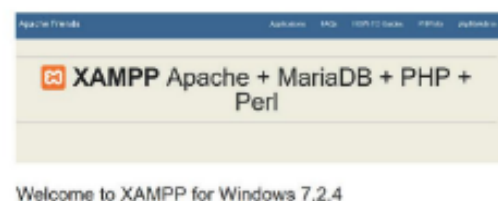


Figura 1.4
Instalación correcta.

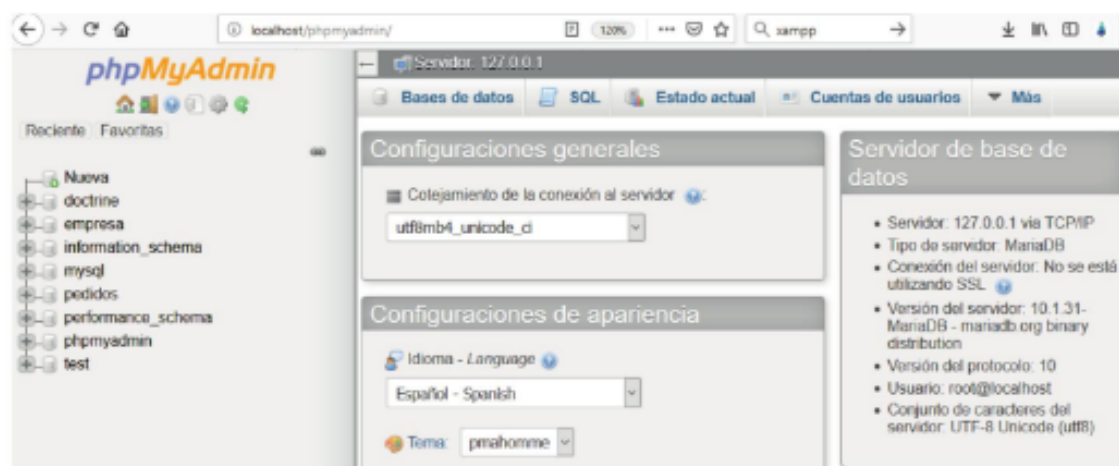


Figura 1.5
La aplicación phpMyAdmin.

Actividad propuesta 1.4



Investiga cómo se utiliza phpMyAdmin. Prueba a crear bases de datos, tablas y restricciones.

1.3.2. Ejemplos y bases de datos

Para probar los ejemplos del libro, descomprime el fichero **libro_servidor.rar** disponible en la web de Editorial Síntesis (www.sintesis.com), que puedes descargar con el código señalado en la primera página del libro.

Contiene 11 directorios, uno por cada capítulo del libro y uno llamado **vendor**. También hay un fichero denominado **crear.sql**. Los directorios hay que copiarlos en la carpeta raíz del servidor (C:\xampp\htdocs en Windows).

El fichero sirve para crear las bases de datos. Puedes importarlo desde phpMyAdmin, en la pestaña de importar. Si se ejecuta correctamente, debería crear tres bases de datos nuevas: empresa, pedidos y doctrine.

Actividad propuesta 1.5



Accede a la página web <https://getcomposer.org/> e instala Composer, un gestor de dependencias de librerías de PHP. Una vez instalado, se pueden añadir nuevas librerías con un simple comando.

1.3.3. Entorno de desarrollo

Como IDE es posible utilizar un entorno ligero como Notepad++. Si se prefiere un IDE más completo, algunas opciones son:

1. *Eclipse PDT*. La versión de Eclipse para PHP.
2. *Aptana*. Puede ser buena opción si se tiene interés en integrar el desarrollo en la parte del cliente. Está basado en Eclipse y está disponible como *plugin* y como aplicación independiente.
3. *PHPStorm*. Un entorno muy completo, pero es de pago. Se puede probar gratis durante un mes.

Aunque puedan parecer excesivos para los primeros ejemplos, a medida que los ejercicios se hacen más complejos conviene usar un buen IDE.

Recursos web

www

Puedes descargar estos IDEs y algunos tutoriales básicos en los siguientes enlaces:

Eclipse PDT: <https://www.eclipse.org/pdt/>

Aptana: <http://www.aptana.com/>

PHPStorm: <https://www.jetbrains.com/phpstorm/>

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.

Ejercicios propuestos



1. Instala XAMPP siguiendo las indicaciones del apartado 1.3.1.

2. Copia los directorios del fichero libro_servidor.rar al servidor web e importa el script crear.sql al servidor de bases de datos.
3. Busca información sobre IDEs para PHP. Descarga e instala algunos para ir probándolos con los ejemplos.
4. Analiza las ventajas del modelo a 3 capas respecto al enfoque cliente-servidor básico.
5. Piensa ejemplos de aplicaciones que utilices o conozcas que sigan el modelo.
6. Busca información sobre el modelo a N capas, una generalización del modelo a 3 capas.

ACTIVIDADES DE AUTOEVALUACIÓN

1. El W3C:
 - ☐ a) Elabora servidores de aplicaciones.
 - ☐ b) Elabora y mantiene estándares de tecnologías web.
 - ☐ c) Elabora regulación de obligado cumplimiento para los desarrolladores.
2. El lenguaje básico para la elaboración de páginas web es:
 - ☐ a) HTTP.
 - ☐ b) HTML.
 - ☐ c) HTTPS.
3. En el modelo a tres capas, la lógica de la interfaz gráfica se sitúa en la capa de:
 - ☐ a) Presentación.
 - ☐ b) Datos.
 - ☐ c) Negocio.
4. En el modelo a tres capas no hay comunicación directa entre:
 - ☐ a) La capa de datos y la de negocio.
 - ☐ b) La capa de presentación y la de negocio.
 - ☐ c) La capa de datos y la de presentación.
5. En el modelo cliente-servidor, la comunicación la inicia:
 - ☐ a) El cliente.
 - ☐ b) El servidor.
 - ☐ c) Cualquiera de los dos.
6. ¿Cuál es el puerto asignado al protocolo HTTP?
 - ☐ a) 80.
 - ☐ b) 21.
 - ☐ c) 8000.

7. ¿Cuál es el puerto asignado al protocolo HTTPS?
- ☐ a) 80.
 - ☐ b) 553.
 - ☐ c) 8000.
8. XAMPP incluye:
- ☐ a) Servidor web y PHP.
 - ☐ b) Servidor web, PHP y base datos.
 - ☐ c) Servidor de bases de datos y PHP.
9. ¿Cuál de los siguientes no es un estándar del W3C?
- ☐ a) HTML.
 - ☐ b) CGI.
 - ☐ c) HTTP.
10. Si se utiliza CGI, las peticiones se responden:
- ☐ a) Usando un módulo del servidor web.
 - ☐ b) Usando un ejecutable en el servidor web.
 - ☐ c) Usando un *servlet* en el servidor web.

SOLUCIONES:

1. ☐ a ☒ b ☐ c
2. ☐ a ☒ b ☐ c
3. ☒ a ☐ b ☐ c
4. ☐ a ☐ b ☒ c

5. ☒ a ☐ b ☐ c
6. ☒ a ☐ b ☐ c
7. ☐ a ☒ b ☐ c
8. ☐ a ☒ b ☐ c

9. ☐ a ☒ b ☐ c
10. ☐ a ☒ b ☐ c