

INTRODUCCIÓN

Este libro complementa los trabajos que he dedicado a la programación de sitios web. En los dos anteriores (*“Domine XHTML 1.0 y CSS 2”* y *“Domine JavaScript, 2ª. Edición”*, publicados por esta misma editorial) me explayé sobre la programación en el lado del cliente. Ahora que usted, lector, ha llegado a dominar esa línea de trabajo es el momento de dar un gran salto en la creación de sitios web: la programación en el lado del servidor.

A lo largo de la segunda edición de esta obra aprenderá todo lo que necesita para acometer los proyectos más ambiciosos que pueblan la Red de redes. Atrás en el tiempo quedan los días oscuros en que las páginas de Internet se limitaban a cargarse y ejecutarse en el lado del cliente, con unos comportamientos inamovibles y una rigidez que hoy es inaceptable. Aprenderá a crear proyectos como una tienda virtual, foros, libros de visitas, tableros de anuncios, etc. Las posibilidades que abre esta vía de trabajo son ilimitadas.

En los dos primeros capítulos aprenderemos los conceptos básicos de esta forma de trabajar; desde el significado de “lado del cliente” y “lado del servidor” hasta la forma de comprender la arquitectura cliente-servidor, los conceptos básicos sobre redes, el software necesario para trabajar con un servidor, etc.

En los capítulos del 3 al 14 aprenderemos los conceptos básicos de la programación en el lado del servidor. Conceptos que, para el lector que ya conozca las técnicas de programación en el lado del cliente (JavaScript) son familiares, tales como variables, matrices, objetos, etc., serán elevados ahora a su máxima expresión. Para ello emplearemos PHP. Se ha escogido este lenguaje para la programación dinámica de sitios web por varias razones, entre las que cabe destacar las siguientes:

- Es un lenguaje que combina potencia, versatilidad y sencillez de aprendizaje.
- Permite gestionar eficientemente cualquier elemento de una página web.
- Se comunica perfectamente con HTML (XHTML) y JavaScript, permitiendo alcanzar un elevado nivel de integración con el navegador.
- Gestiona con eficacia bases de datos, tan necesarias para gran cantidad de proyectos modernos.
- Es un lenguaje en constante evolución. En el momento de escribir este libro se halla disponible la versión 5 (PHP 5), que presenta grandes mejoras con respecto a las anteriores.
- Es un lenguaje de programación totalmente libre, lo que en el ámbito informático se conoce como *open source*. Esto quiere decir que usted podrá disponer de la última versión completamente gratis. Pero no sólo eso. Además, puede disponer libremente del código fuente original del lenguaje, por si decide modificarlo para adaptarlo a sus necesidades específicas.
- Ningún otro lenguaje de programación está tan extendido en Internet. Los sitios de mayor éxito están basados en el uso de este lenguaje. Por algo será.

Del capítulo 15 al 17 acometeremos el conocimiento de MySQL, el gestor de bases de datos en Internet por excelencia. Entre otras, cuenta con las siguientes ventajas:

- Al igual que PHP, se trata de una herramienta open source. Aunque existen licencias de pago, destinadas a grandes usuarios que necesitan un soporte técnico especializado, éste no será su caso. En este libro encontrará toda la ayuda necesaria para manejar MySQL sin necesidad de soporte técnico. No obstante, existe en Internet gran cantidad de documentación, tanto en inglés como en español.
- Es un gestor de bases de datos sensiblemente más rápido y eficiente que la mayoría de los que hay en el mercado, muchos de los cuales, además, son de pago.
- Permite una gran versatilidad a la hora de almacenar y gestionar todo tipo de datos.
- Se gestiona mediante un lenguaje específico conocido como SQL (cuyo estudio también abordaremos, lógicamente, en el libro) que se integra perfectamente con PHP.

Desde el capítulo 18 seguiremos profundizando en la programación de scripts en el lado del servidor, usando los conceptos que hayamos aprendido en las

partes anteriores, y descubriendo otros nuevos. Además, abordaremos el desarrollo y la puesta en marcha completa de una aplicación web muy extendida hoy en día: un foro virtual. También aprenderemos algo que no todos los programadores de Internet conocen: el desarrollo de películas de Flash... SIN FLASH. Usando, exclusivamente PHP, podremos desarrollar las películas que necesitamos sin comprar ni usar ninguna herramienta adicional, y de un modo muy fácil.

Por último, cierra la obra una serie de apéndices en los que el lector encontrará información muy útil: desde una relación de páginas web de visita obligada hasta la descripción de herramientas complementarias que facilitan enormemente el trabajo del webmaster.

Deseo que usted encuentre estas disciplinas tan fascinantes como yo. Con este objetivo he redactado este libro, poniendo a su servicio toda mi experiencia, tanto como webmaster como a nivel docente. Si usted disfruta tanto leyendo este libro como yo he disfrutado escribiéndolo, y si le resulta tan útil como deseo, me sentiré satisfecho.

Para poder sacarle partido al presente texto es necesario que conozca perfectamente las técnicas de programación de sitios web en el lado del cliente (XHTML y JavaScript). Si no, no podrá seguir adelante. Si no las conoce, le recomiendo que lea previamente los dos títulos anteriores de esta trilogía.

Cuando complete el estudio de este volumen habrá alcanzado el nivel necesario para empezar a desarrollar sitios auténticamente profesionales por usted mismo. El desarrollo real de sitios dinámicos le dará luego la experiencia necesaria para crear su propio estilo de trabajo.

Capítulo 1

LA ARQUITECTURA CLIENTE-SERVIDOR

Cuando se habla de Internet el ciudadano común sabe que enciende su ordenador, abre su navegador y tiene una barra de direcciones donde teclea el nombre de la página que quiere visitar. Si todo va bien, se accede a la página solicitada y, si no, aparece un mensaje informando de que se ha producido un error. Lo que hay “detrás de la pared” (al otro lado del hilo telefónico o de la conexión ADSL, cable-módem, etc.) no se sabe lo que es. El usuario común no tiene por qué preocuparse de ello. Pero nosotros sí. En algún lugar del mundo (en realidad en distintos lugares) existen unos ordenadores conocidos genéricamente como *servidores*. Se les llama así porque son los que tienen almacenadas las páginas que visita el usuario, y se las sirven cuando éste las solicita. Por extensión, los ordenadores de los usuarios se conocen, genéricamente, como *clientes*. Así pues, cuando nos conectamos a Internet y pedimos una página tenemos dos ordenadores en juego: el nuestro propio (desde el que nos conectamos) que es el cliente y el que nos envía por cable (u otro medio) aquello que hemos pedido (el servidor).

En realidad, la arquitectura de Internet es bastante más compleja que esto, pero lo que hemos mencionado aquí constituye la espina dorsal de todo el sistema. En este capítulo vamos a analizar esta arquitectura en la medida necesaria para comprender qué es lo que vamos a aprender a lo largo del libro. Los conceptos que vamos a exponer a continuación les parecerán muy elementales a las personas con experiencia previa en redes, pero son necesarios como introducción para aquellos lectores que no se hayan asomado nunca a este aspecto de la informática. Ruego a los conocedores del tema que me disculpen por la simplificación que hago de estos conceptos, pero no pretendo que éste sea un tratado acerca de redes sino, simplemente, una introducción necesaria para luego poder ahondar en la programación en el lado del servidor.

1.1 LAS DIRECCIONES IP Y EL SERVICIO DNS

En realidad, cuando un usuario se conecta a Internet y escribe en la barra de direcciones de su navegador el nombre de la página que desea visitar tiende a pensar que es mediante este nombre como se localiza dicha página en La Red. Pero esto no es así. Cuando un ordenador está conectado a Internet (y un servidor es un ordenador) tiene asignado un número de identificación que permite localizarlo de forma inequívoca: es la **dirección IP**. Este número es un grupo de cuatro valores, cada uno de los cuales está en un rango entre 0 y 255. Así pues, una dirección IP tiene un aspecto como 192.168.1.1.

Las direcciones IP se dividen, atendiendo a su ámbito, en dos categorías: *internas* y *externas*. Las direcciones internas se emplean para identificar y localizar ordenadores dentro de nuestra propia red local (en el caso de que tengamos una). Las direcciones externas se emplean para acceder a Internet, es decir, para localizar e identificar ordenadores remotos. Dicho de otra manera: la diferencia entre una red local e Internet es el ámbito de trabajo. Mientras que en una red local nos conectamos con ordenadores que están en el mismo edificio que el nuestro (en ocasiones incluso en la misma habitación), en Internet nos conectamos a equipos que pueden estar a 300 metros o en el otro lado del mundo. Los rangos de direcciones IP internas aparecen en la tabla de la figura 1.1.

RANGO
De 10.0.0.0 a 10.255.255.255
De 172.16.0.0 a 172.31.255.255
De 192.168.0.0 a 192.168.255.255

Figura 1.1

Cuando usted se conecta a un ordenador de su red local (en casa o en la oficina) realmente se está conectando a un equipo que tiene asignada un IP de las que aparecen en la tabla. Usted no tiene por qué saber esto. Sus ordenadores, o los de sus compañeros de trabajo, tienen un nombre y un icono que los identifica para usted en el escritorio. Sin embargo, cada vez que usted “llama” a uno de esos ordenadores (a través del nombre o el icono), el sistema operativo de su ordenador “sabe” a qué IP tiene que dirigirse.

Existe una dirección IP especial que usaremos mucho en los ejercicios de este libro: la 127.0.0.1. Esta dirección se conoce como *de bucle local* o *localhost*. Identifica siempre al ordenador con el que estamos trabajando. Cualquier plataforma del mundo (cualquier sistema operativo) “sabe” que, si solicitamos un documento, página web, o cualquier otro recurso a esta dirección, en realidad lo

estamos solicitando a nuestro propio ordenador, que lo busca en una carpeta determinada (aquella que se ha definido como localhost). No se preocupe ahora de qué carpeta es esa, cómo se define, ni cómo funciona todo esto en detalle. En el próximo capítulo hablaremos en profundidad de esto.

Cuando un usuario se conecta a Internet y solicita una página web a través de su barra de direcciones, lo que hace es establecer contacto con el ordenador que tiene almacenada esa página web. En este ordenador está funcionando un programa que se conoce, genéricamente, como **servidor web**. Este programa busca la página solicitada en el disco duro del servidor y se la envía al cliente. Así pues, podemos decir que un servidor es un ordenador que almacena páginas web y también un programa que se encarga de entregar dichos documentos a los clientes que lo soliciten. Por analogía, el cliente es un ordenador (aquél desde el que nos conectamos a Internet) y un programa cliente (el navegador con el que nos conectamos y donde se muestra la página solicitada). Como vemos, cada elemento de una arquitectura cliente-servidor se compone, por lo tanto, de un hardware y, por supuesto, un software.

En este momento, el lector avisado me dice: “Pero cuando yo tecleo una dirección de Internet, escribo el nombre de la página que quiero, no el número que se llama dirección IP del que usted hablaba hace un momento”. Pues sí. Tiene usted razón. Los creadores de Internet comprendieron muy pronto que para los usuarios es mucho más fácil recordar un nombre con un significado específico que un número de IP. Además, las páginas de Internet pueden cambiar de servidor físico, con lo que cambiaría su dirección IP. Ahí es donde entra en juego el servicio **DNS** (*Domain Name Server*, Servidor de Nombre de Dominio) que completa Internet de una forma muy útil. Cuando usted se conecta a Internet lo hace a través de un proveedor de servicios (ISP, *Internet Service Provider*, Proveedor de Servicios de Internet). Este ISP le asigna dos direcciones DNS que usted especifica en la configuración de su conexión a Internet. Si su ISP le entrega un CD de instalación, es muy posible que el programa instalador establezca estas dos direcciones en la configuración de su navegador; si no, su ISP le dice cuáles son estas direcciones y cómo debe configurar su conexión.

Cuando usted abre su navegador y solicita una página web, su ordenador no se conecta directamente al servidor de dicha página. Se conecta a uno de los servidores DNS establecidos. El servidor DNS contiene un directorio con todas las páginas web del mundo y las IPs de los correspondientes servidores. Así pues, el DNS busca en la lista la página que usted ha solicitado y el número de IP correspondiente. Esto es lo que se llama **resolver el nombre**. El DNS le devuelve el número de IP a su navegador y éste se conecta entonces al servidor web que corresponde a esa IP y que contiene la página solicitada. En resumen, la conexión a una página web se establece así:

- El navegador envía el nombre de la página web deseada al DNS.
- El DNS resuelve el nombre y le devuelve al cliente la IP del servidor que contiene la página solicitada.
- El navegador se conecta al servidor web correspondiente mediante la IP recibida del DNS y solicita la página web.
- El servidor web entrega la página al cliente.

El esquema de funcionamiento aparece en la figura 1.2. Como usted puede ver, se aprecian gráficamente los cuatro pasos mencionados. Estos pasos son siempre los mismos, salvo que usted teclee en la barra de direcciones la IP de la página que desea ver. Pero claro, ningún usuario se aprende o se anota las IPs de las páginas. Todos trabajamos mejor con nombres. Para eso están los DNS, que se actualizan cada vez que aparece una nueva página en Internet.

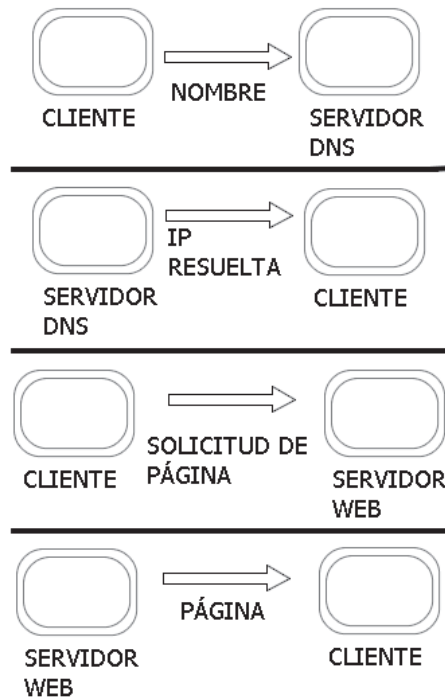


Figura 1.2

¿Se va haciendo una idea de la estructura de la Red de Redes? En realidad esto es un poco más complejo que lo que hemos visto hasta ahora. Por ejemplo, los ISPs suelen “intercalar” unos equipos entre su ordenador e Internet. Estos equipos se conocen con el nombre de *proxys*. Un proxy es un ordenador que almacena una copia de las páginas que usted visita con más frecuencia. De este modo, cuando

usted solicita de nuevo esas páginas, es el proxy quien se las sirve, sin que se busquen de nuevo en Internet. Este sistema tiene la ventaja de que le sirve las páginas más rápido que si hubiera que acceder a Internet, resolver el nombre y descargarlas desde el servidor correspondiente. Como contrapartida, puede que la página que le sirve el proxy no esté actualizada. En ese caso conviene que el navegador ignore lo que el proxy le ofrece y se conecte a Internet a buscar la página. Usted ya conoce los recursos que emplean los programadores en HTML para que la página se cargue desde el servidor, de modo que el cliente tenga la última versión disponible. Programando en PHP también existen recursos al efecto que usted aprenderá. Lo cierto es que con la implementación masiva de líneas de banda ancha, y los ISP compitiendo diariamente por ofrecer mayor velocidad de conexión, el uso de servidores proxy está dejando de tener sentido para el fin que hemos descrito. Sin embargo, los ISP los siguen empleando a fin de optimizar otros aspectos de la conexión entre el cliente e Internet.

1.2 LOS PROTOCOLOS TCP/IP

Conectar entre sí dos o más ordenadores para que puedan compartir información y recursos no es una tarea simple. Entre otras cosas, es necesario que la forma en que transmiten y reciben información los distintos equipos esté normalizada. Para este fin existen unos **protocolos** de comunicación. El concepto de protocolo implica que la transferencia de información se hace en la misma forma en cada ordenador, de tal modo que la información que uno envía, el otro la puede interpretar cuando la recibe. Esto es lógico. Si yo sólo hablo español y usted sólo habla inglés no vamos a entendernos nunca. Hace falta una normalización para que nos entendamos. El conjunto de normas establecidas para la transferencia de información es un protocolo. Todos los ordenadores actuales se comunican entre sí siguiendo lo que se conoce como **pila de protocolos TCP/IP**. Esta pila abarca diversos protocolos que los equipos informáticos emplean según el tipo de información que transfieran entre ellos. Por ejemplo, no se emplea el mismo protocolo para transferir una página web que para transferir un fichero que el usuario quiere descargarse, o para transmitir o recibir un mensaje de correo electrónico. Cada uno de los distintos servicios que uno puede encontrar en una Red funciona mediante un protocolo específico para dicho servicio.

A fin de que todos los ordenadores del mundo “conozcan” e implementen correctamente dichos protocolos se creó lo que hoy día se conoce como **el modelo OSI de la ISO**. La ISO (*International Standard Organization*, Organización Internacional de Normalizaciones) es una agencia dedicada a normalizar todo aquello que pueda normalizarse. Desde el tamaño de las clavijas de los enchufes eléctricos hasta, lógicamente, los protocolos de comunicaciones. El modelo OSI

(*Open Standard Interconnectivity*, Interconexión Normalizada Abierta) reúne todos los protocolos de los distintos servicios de que se puede disponer en una red de ordenadores. En realidad incluye más normalizaciones como, por ejemplo, los conectores de las tarjetas de red y otros aspectos de la arquitectura física de una red. La explicación que damos aquí está, por fuerza, muy simplificada, ya que para obtener un conocimiento profundo de redes necesitaríamos varios volúmenes y, desde luego, es algo que está fuera del objetivo de este libro. Lo que sí tenemos que tener claro es que cualquier ordenador del mundo, independientemente del sistema operativo que monte, respeta estas normalizaciones. Los fabricantes de hardware y de sistemas operativos ya se ocupan de que esto sea así. De hecho, un ordenador o un sistema operativo que no cumpliera estos protocolos no podría conectarse a ninguna red y no tendría mucho uso. Esta pila de protocolos es, por tanto, universal. Su sistema operativo la tiene implementada y gracias a ello puede usted conectarse a Internet. En posteriores capítulos veremos algunos aspectos de estos protocolos especialmente interesantes, a efectos de programación de sitios web.

1.2.1 Los paquetes

Como he dicho, no nos vamos a meter en detalle en el funcionamiento de la pila de protocolos TCP/IP ni del modelo OSI. Sin embargo, sí necesitamos conocer algunas líneas generales. Por ejemplo, debemos saber que la información que se transmite entre un servidor y un cliente no viaja “de golpe”, sino que antes de ser enviada es dividida en pequeños fragmentos que se conocen con el nombre de **paquetes**. Cuando estos paquetes llegan al ordenador de destino, son reagrupados para reconstruir la información original. Esto funciona así para cualquier transferencia de información de cualquier tipo. Este modo de transferir información tiene una doble razón de ser:

- Por una parte, dada la compleja estructura de nodos de La Red, cada paquete puede viajar por un camino diferente, con lo cual puede haber varios paquetes viajando del servidor al cliente simultáneamente.
- Por otra parte, si algún paquete se pierde o deteriora durante el trayecto, no es necesario reenviar toda la información, sino que basta con volver a mandar ese paquete.

Los paquetes están constituidos, básicamente, por dos partes: la **cabecera** y el **cuerpo**. En el cuerpo es donde se encuentra el fragmento de información que se transmite. En la cabecera encontramos algunas informaciones interesantes:

- El protocolo empleado, dependiendo del servicio del que se trate (un documento web, una descarga de archivos, etc).

- Tamaño del cuerpo del paquete y *checksum*. El checksum es un método de comprobación que se emplea para verificar que el paquete haya llegado correctamente. Si no es así, lo pide de nuevo.
- Las direcciones IP de los dos ordenadores, el de origen y el de destino. El lenguaje PHP permite identificar la IP del cliente. Esto se emplea en sitios de comercio electrónico o donde el cliente puede enviar o solicitar información sensible. Esto nos lleva de nuevo al uso de proxys por parte de los ISP. Algunos proxys interponen su propia IP entre la del cliente y la del servidor, con lo que la IP del cliente queda “oculta”. PHP 5 implementa un sistema muy efectivo de identificación de la IP del cliente.
- El puerto de comunicaciones empleado para la transferencia de información. En el próximo apartado comentaremos algo más al respecto.
- Información adicional, tal como la relativa al camino que sigue el paquete para llegar desde el ordenador de origen al de destino.

Como ve, la información contenida en la cabecera es de una importancia tan vital como la que hay en el cuerpo. Volveremos sobre esto en su momento.

1.2.2 Los puertos

Cada uno de los servicios que se pueden usar en Internet implica el uso de uno o más *puertos*. Los puertos son canales numerados que se usan para transferir los distintos tipos de información. Los sistemas operativos modernos reconocen más de 65.000 puertos diferentes. De éstos, los más significativos son los primeros 1.024, que se conocen con el nombre genérico de *puertos bien conocidos*. Éstos se emplean para comunicaciones de distintos tipos.

En la tabla de la figura 1.3 se citan los más habituales. Los puertos del 1025 en adelante no tienen, a priori, un uso tipificado, por lo que pueden usarse para lo que se quiera. Este elevado número de puertos tiene una ventaja. Se pueden diseñar sistemas que se comuniquen por cualquiera de los puertos altos para transferencias específicas que un programador implemente en una aplicación. Pero también hay un problema: cuando un puerto no se usa, el Administrador del Sistema debería asegurarse de que está *cerrado*, es decir, que no puede entrar ni salir información a través de él. Pero este estado ideal de cosas sólo se ve en los sueños. En la realidad, cualquier sistema comercial tiene muchos puertos abiertos, aunque no se usen. Esto implica un riesgo de intrusión por parte de personas malintencionadas, que pueden modificar nuestro equipo, robarnos información, etc. Afortunadamente existen en el mercado unos programas conocidos como *cortafuegos (firewalls)* que palían bastante esta situación.

Nº DE PUERTO	SERVICIO	DESCRIPCIÓN
20 y 21	FTP	Transferencia de archivos (descarga o envío de archivos al servidor)
25	SMTP	Envío de correo electrónico
80	HTTP	Transferencia de páginas web
110	POP	Recepción de correo electrónico
43	HTTPS	Transferencia de documentos web bajo una capa segura
8080	HTTP	Transferencia de páginas web (puerto alternativo empleado por algunos sistemas)

Figura 1.3

Estos puertos son los más habituales, aunque, como ya se ha dicho, existen muchos más puertos bien conocidos y otros de uso programable. Cuando un ordenador está enviando datos por un puerto se dice que *habla* por ese puerto. Cuando está recibiendo datos, o pendiente a la espera de recibirlos, por un puerto, se dice que *escucha* por ese puerto. Para que se produzca transferencia de datos, debe haber un ordenador hablando y otro escuchando.

Una buena técnica de seguridad, cuando use programas de transferencia de datos, es cambiar los puertos “normales” por otros más altos, por encima de 1025, en los dos equipos participantes en la transferencia.

1.2.3 Sockets

Cuando nos conectamos a una página web podemos teclear su dirección IP. Por ejemplo, así: **http://127.0.0.1**. Esto sería una llamada a la página principal (index.htm o index.php) del sitio web alojado en la máquina que tiene la dirección IP indicada. Como lo que buscamos es una página web (servicio HTTP, *Hyper Text Transfer Protocol*, Protocolo de Transferencia de Hiper Texto), el sistema asume, por defecto, que el servidor escucha por el puerto 80. Si el servidor está configurado de tal modo que escucha por otro puerto distinto, es necesario

especificarlo al teclear la dirección. Por ejemplo: si el servidor escucha por el puerto 8080, la dirección anterior quedaría así: **http://127.0.0.1:8080**. El puerto se separa de la IP mediante el signo de dos puntos. Esta notación completa (IP:puerto) se conoce con el nombre de *socket*.

1.3 SITIOS DINÁMICOS

Cuando tenemos una página escrita que utiliza únicamente las tecnologías del lado del cliente (HTML y JavaScript) se dice que es estática. Esto significa que sus contenidos son siempre los mismos. Si queremos que cambien, debemos cambiarlos nosotros mismos, editando la página, para que los usuarios vean los nuevos contenidos. A menudo esto es imposible, debido a que hay muchas ocasiones en las que los contenidos deben ser dependientes de una solicitud por parte del usuario. Suponga que tiene usted que escribir una página con un listado de empresas de su ciudad. Las habrá de todos los gremios. Ahora un usuario que consulta su página quiere ver una lista de empresas de, digamos, floristería. El usuario sólo quiere ver las floristerías, que es lo que necesita. No quiere ver una página enorme, con todas las empresas, y tener que andar buscando las floristerías. Quiere un resultado claro y útil. Las páginas dinámicas emplean lo que se conoce como programación en el lado del servidor. Desde el punto de vista técnico, la principal diferencia es que una página estática, cuando es llamada por un cliente, baja al navegador tal como está, y se muestra como es. Cuando el cliente llama a una página dinámica, existe un proceso en el lado del servidor que crea en ese momento la página, incorporando los contenidos que el cliente ha solicitado, y sirve la página recién creada al cliente. Otro cliente que solicite la misma página con otros contenidos recibirá un resultado diferente. Por eso se habla de páginas dinámicas, porque su contenido puede variar y se genera en el servidor en el momento de la solicitud.