

03

Introducción a la Programación Web

¿Qué es una "página web"?
¿Y una "aplicación web"?



¿Qué es una página web?

Una web, o página web, está formada por un conjunto de documentos de diferentes tipos **ubicados en una carpeta de un servidor http**.

Los documentos que describen el **contenido** y **aspecto** de una página son, respectivamente, los HTML y CSS.

Puede haber otros tipos de documentos enlazados, tales como imágenes, contenidos multimedia o scripts de programación.

HTML: HyperText Markup Language

- Describe la estructura y contenido de una página
- Se guarda como un archivo de texto plano con extensión **.html** (o **.htm**)

CSS: Cascading Style Sheets

- Describe el aspecto de la página
- Documentos con extensión **.css**, también de texto plano
- Las directrices CSS pueden estar también dentro del mismo documento html.



¿Qué es una aplicación web?

Los contenidos que muestra una página web pueden ser estáticos o dinámicos:

- **Estáticos:** el contenido de la página es el mismo para cualquier usuario. Los documentos que forman la página están guardados en el disco duro del servidor y son "fijos".
- **Dinámicos:** el contenido que se muestra se crea en el momento de la consulta, en función del cliente, el momento, la ubicación, etc.

Cuando estos contenidos dinámicos responden a una necesidad de proceso de información, nos encontramos ante una aplicación web.

Una app web utiliza las mismas tecnologías HTML y CSS, pero además requiere un lenguaje de programación, bien sea en el "front-end" (javascript), en el back-end (php, javascript, java, c#, python, etc), o en ambos (lo más habitual).

Actualmente la mayor parte de la programación se desarrolla en el ámbito de las aplicaciones web.



Resumen: Página vs. Aplicación

Página

Contenidos **estáticos**.

HTML, CSS y Javascript

Ejemplos: páginas personales, landing pages que anuncian un producto, catálogos online...

Aplicación

Contenidos **dinámicos**, se adaptan al usuario.

Además de HTML, CSS y JS requieren programación en el servidor: **NodeJS, Java, C#, PHP...**

Ejemplos: WordPress, Facebook, Google Maps, Wallapop...



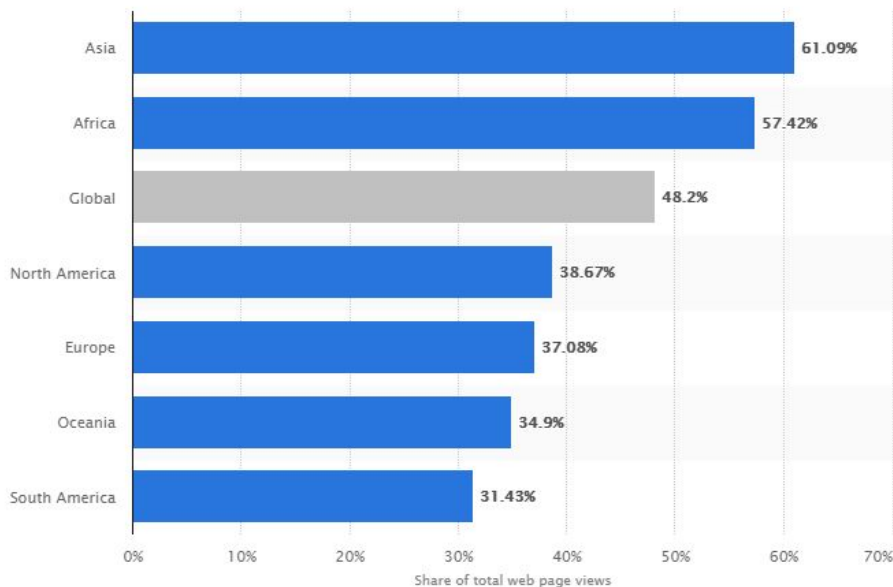
Páginas web / apps responsivas

Globalmente alrededor del 50% de las visualizaciones de páginas web se realizan desde teléfonos móviles (tablets excluidas).

La diversidad de tamaños de pantalla es un reto para el diseño web. Las páginas que se adaptan al tamaño de las pantallas se denominan **"responsivas"**.

En general SIEMPRE crearemos páginas pensando en esta característica. Una opción muy recomendable es utilizar un framework CSS que ya sea responsivo, tal como Bootstrap o Materialize.

Tráfico de internet móvil como porcentaje del tráfico web total en noviembre de 2018, por región



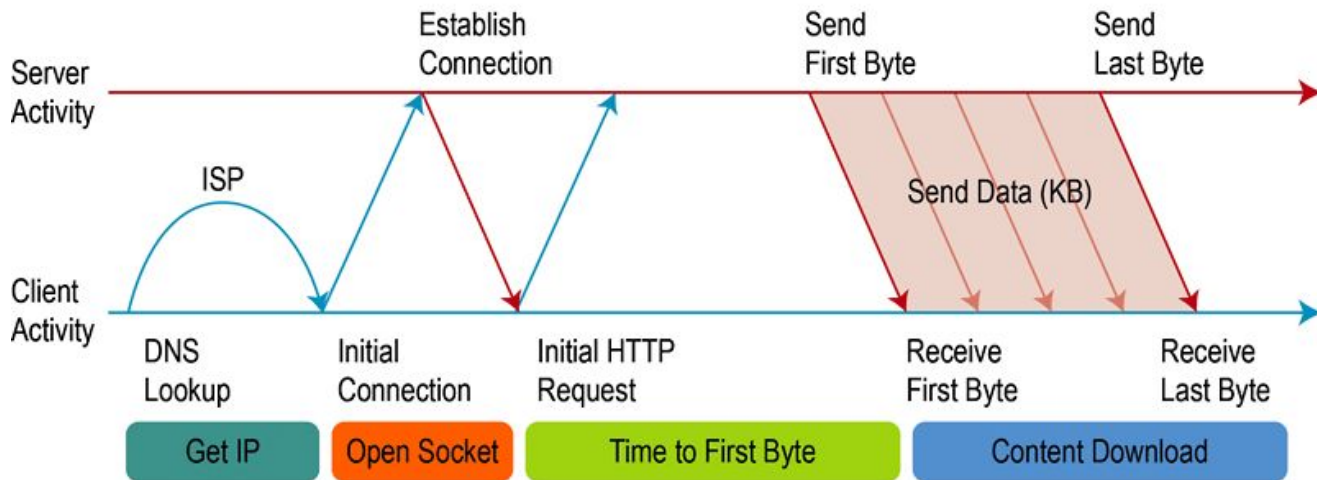
¿Cómo "funciona" una web?



Proceso de consulta de una página web

El emisor, usualmente un navegador web, realiza una petición HTTP (**HTTP Request**) a un servidor a partir de una dirección URI (Uniform Resource Identifier, también denominada URL o Universal Resource Locator). El servidor atiende esta petición y genera la respuesta (**Response**) que el navegador descarga.

The HTTP Request





Proceso de consulta de una página web

La petición (request) puede realizarse mediante comandos GET, POST, PUT, OPTIONS y DELETE. Los más habituales son GET y POST. Veamos un caso de uso dentro de un formulario (<form>):

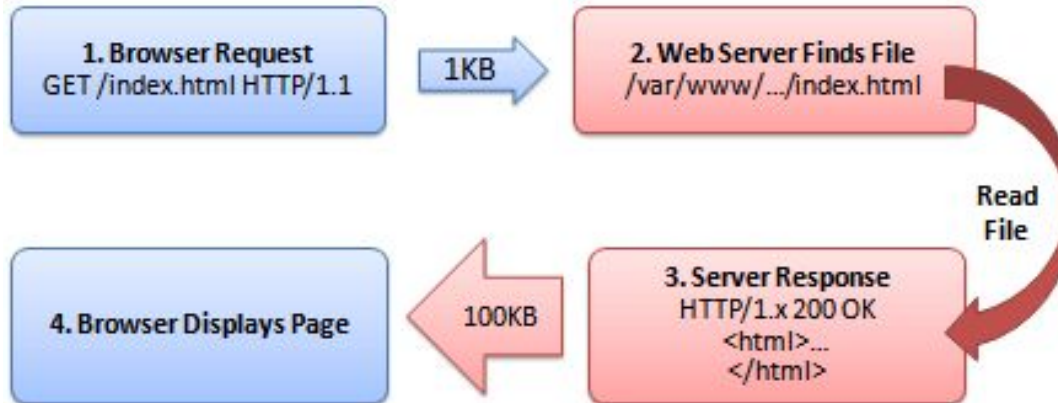
```
formulario.jsp
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2      pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4      "http://www.w3.org/TR/html4/loose.dtd">
5  <html>
6  <head>
7  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
8  <title>Formulario Login</title>
9  </head>
10 <body>
11     <form action="login" method="post">
12         <label>Nombre:</label>
13         <input type="text" name="nombre">
14         <input type="submit" value="Enviar">
15     </form>
16 </body>
17 </html>
```



Proceso de consulta de una página web

Normalmente GET se usa para realizar consultas (pedir datos) al servidor mientras que POST se usa para enviar datos al servidor, por ejemplo rellenar un formulario, realizar una acción de “login”, “subir” una foto, etc.

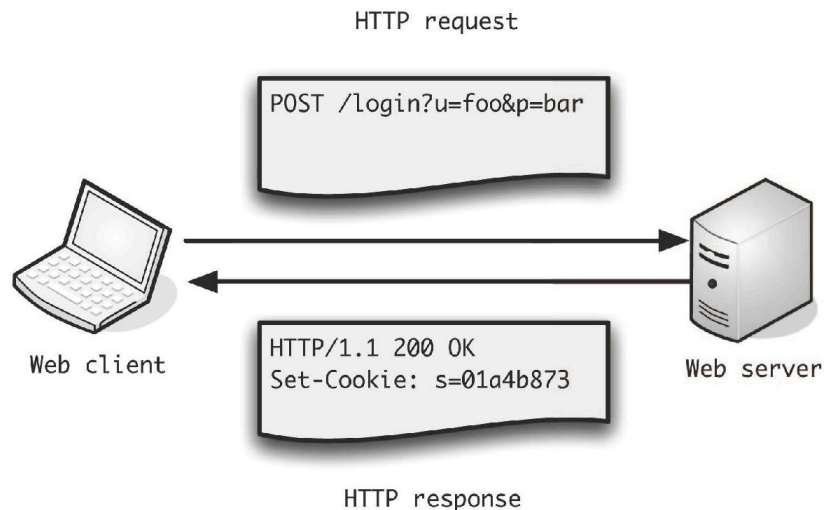
HTTP Request and Response





Envío de parámetros

Tanto en GET como en POST podemos enviar datos al servidor: en la petición GET los datos se envían en la misma dirección uri/url, tras el signo ? y separados por "&", siendo fáciles de visualizar/rastrear. En la petición POST se suelen enviar "escondidos" en el "body" de la petición (request).

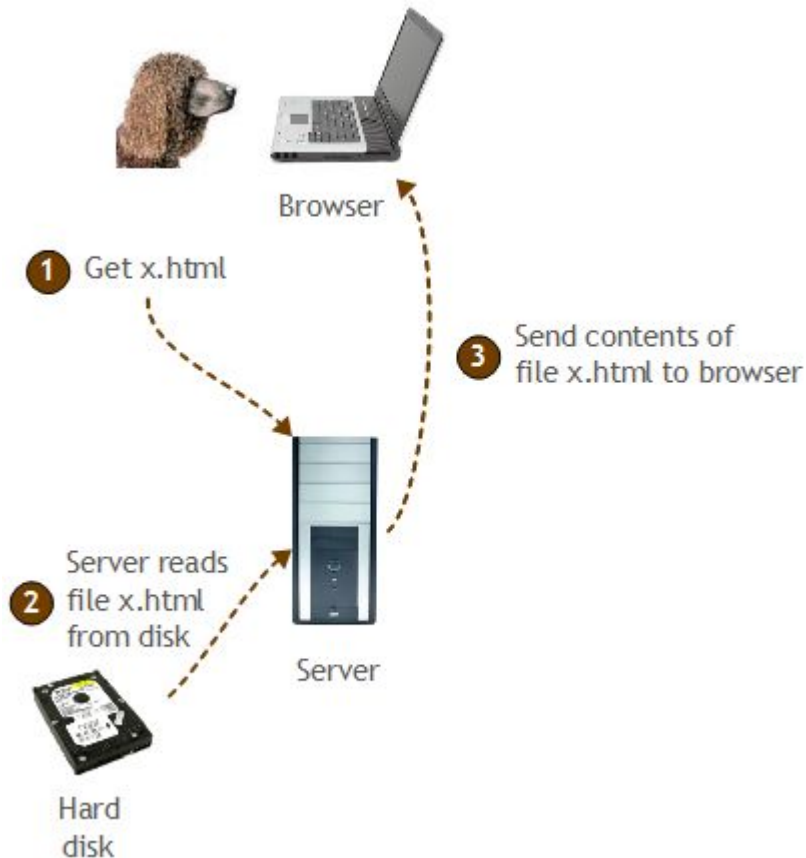




Consulta de página estática

La consulta de la página estática "x.html" sigue el esquema adjunto.

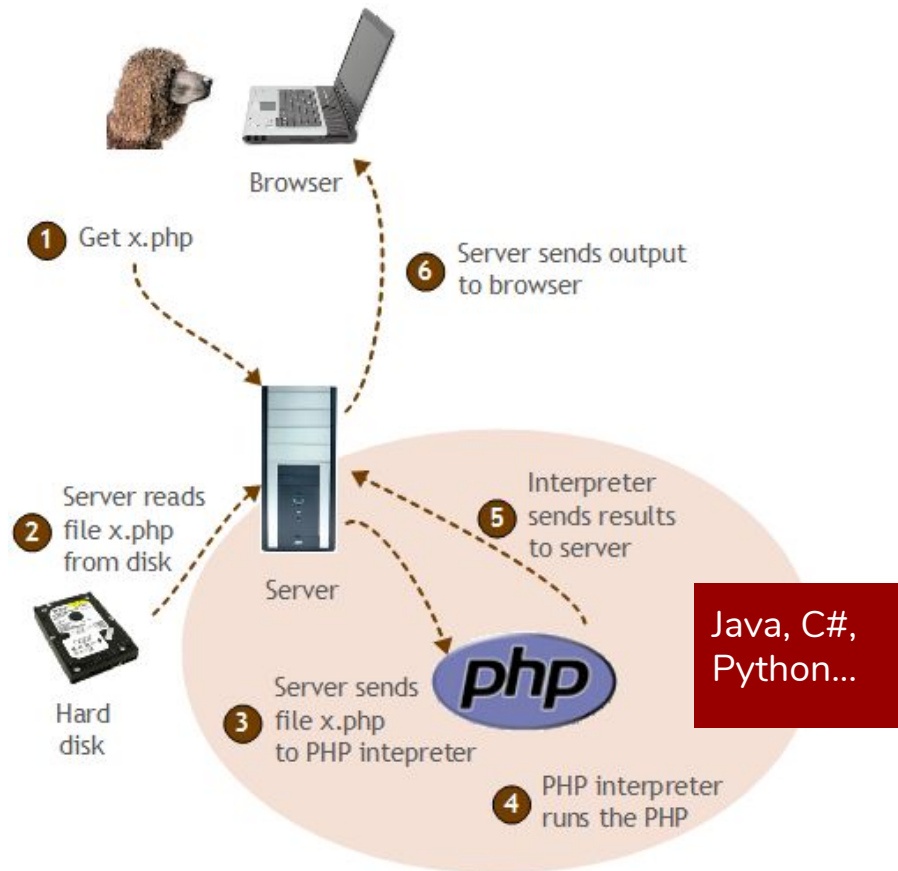
El servidor recibe la petición, busca la página y la devuelve sin ninguna otra intervención.

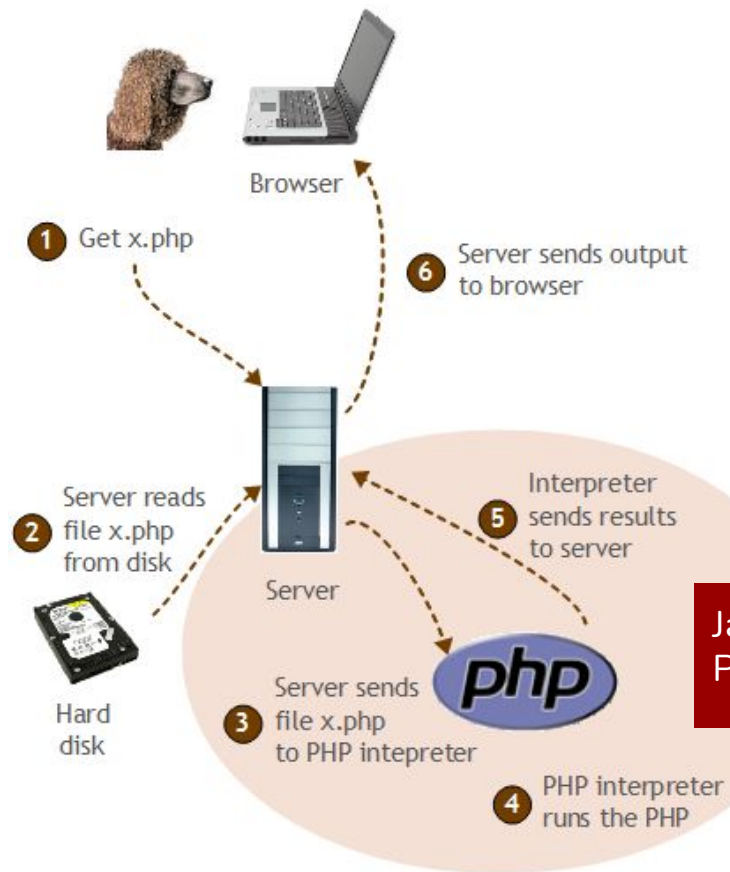
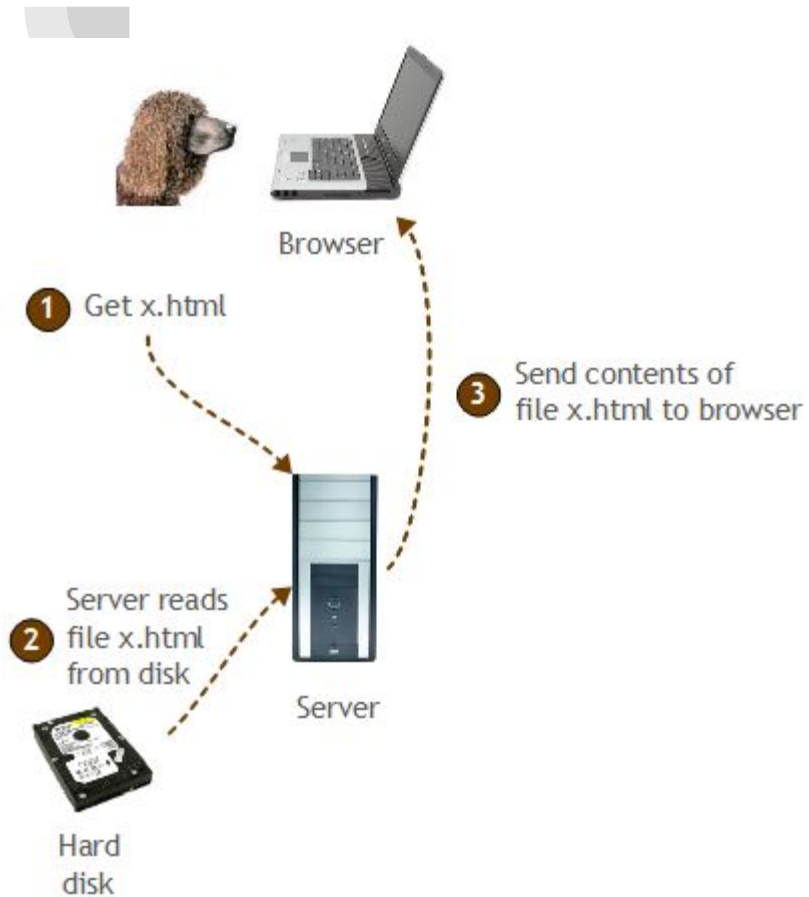


Consulta de página dinámica (app web)

En caso de una página dinámica interviene un proceso adicional: la página se crea en el servidor mediante un lenguaje de programación, en este caso PHP.

Otros lenguajes habituales en el lado del servidor: Java, C#, Python, Javascript mediante Node JS, Ruby.

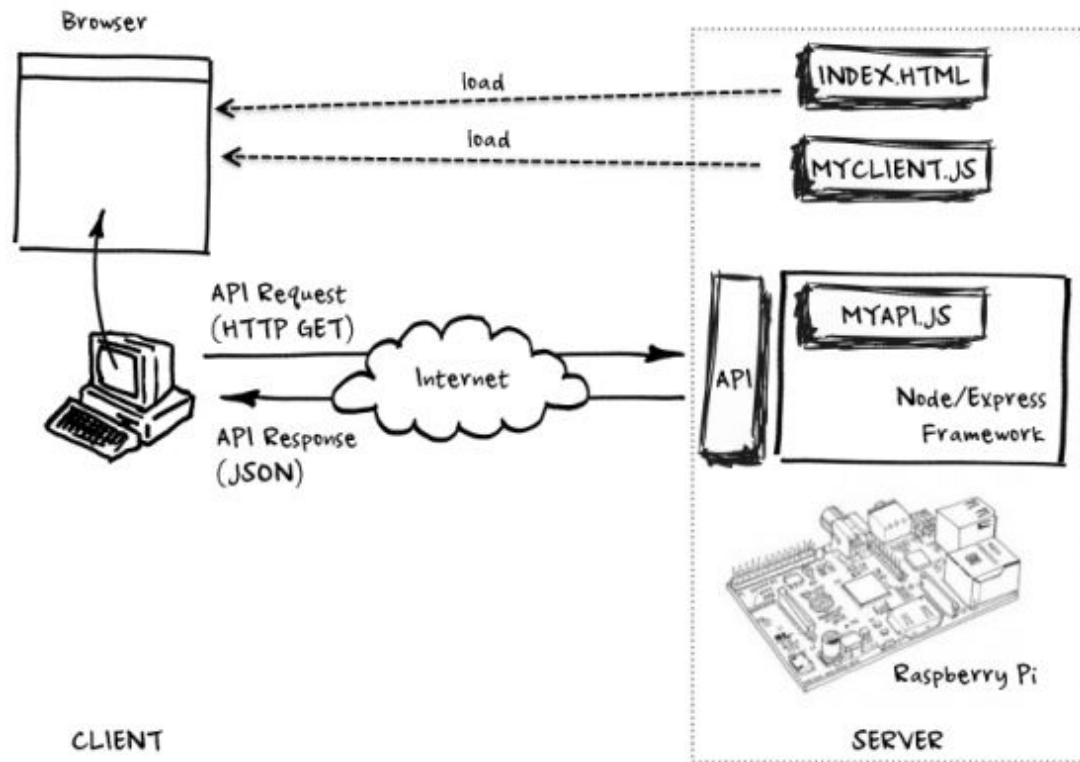




Esquema de aplicación JS / API REST

En el esquema, el navegador carga la aplicación javascript "myclient.js" (posiblemente React o Angular), que controlará toda creación de páginas en el lado del cliente (navegador).

Los datos que requiera del servidor se realizarán mediante peticiones "get" o "post" en segundo plano, sin que la página deba recargarse.



Programación web: tecnologías, lenguajes...



Tecnologías web, entornos

En programación web distinguimos dos ámbitos o entornos:

- **Front-end:** programación de la interfaz de usuario. HTML, CSS y Javascript.
- **Back-end:** programación de la aplicación de gestión de datos y lógica de negocio, pudiendo ser en múltiples lenguajes.

Normalmente los programadores se especializan en uno de estos ámbitos.

Un programador que domine tanto el front como el back se denomina **full stack developer**

Hay múltiples combinaciones de tecnologías para desarrollo front+back. Una combinación concreta se denomina "stack".

STACK





"Stacks" más habituales para desarrollo web

LAMP

- Linux (S.O. de servidor)
- Apache (servidor HTTP)
- MySQL (base de datos)
- PHP (Lenguaje backend)

MEAN/MERN (Full JS)

- MongoDB (base de datos)
- Express JS + Node JS (lenguaje de backend y servidor http)
- Angular (front-end)
- Alternativa: ReactJS en lugar de Angular

ROR

- Ruby, Ruby-on-Rails

CMS

- Drupal, Wordpress, Magento

Frameworks (principalmente back-end)

- Laravel, Symfony (PHP)
- Spring MVC, Struts (Java)
- .NET MVC (C#)
- Django (Python)



Publicación de una web

Publicar una web significa ponerla en "red" a disposición del público ("colgarla en internet"). Hoy en día publicar una web está al alcance de cualquiera. Una vez hemos creado los contenidos basta con:

1. **Registrar** un nombre de dominio
ej) botifarresdepages.com
2. Disponer de un **servidor** conectado a Internet con una **dirección IP fija**.
ej) 215.128.130.266
3. Crear un **registro en un servidor DNS** que vincule el dominio registrado con la IP de nuestro servidor.



Registrar un dominio

Hay que encontrar una combinación de nombre y dominio genérico que no existan (whois.com). Ej:

- botifarradepages.com

Se registra a través de una empresa autorizada por la ICANN. El precio depende del vendedor y del tipo de dominio genérico, y se renovará anualmente.

- .com aprox 12 eur / año
- .cat depende ... de 5 a 35 eur / año
- .es sobre 15 eur / año

Cualquier particular o empresa puede registrar un dominio en Internet, pero determinados dominios genéricos pueden requerir una verificación. Ex) .cat es para páginas con contenido en catalán.

Hace unos años registrar un buen dominio podía ser un negocio muy rentable...

- LasVegas.com - \$90 milion
- CarlInsurance.com - \$49,7 milion
- PrivateJet.com - \$30 milion
- Internet.com - \$18 million
- Sex.com - \$13 million

Disponer de un servidor...

Tenemos el nombre de dominio contratado, y los contenidos que queremos enseñar al mundo. ¿Qué podemos hacer?

Necesitamos poner estos contenidos en un servidor y decirle a todos donde encontrarnos.

- servidor privado/propio
- servidores dedicados
- servidores virtuales
- alojamiento compartido





Tipos de hosting

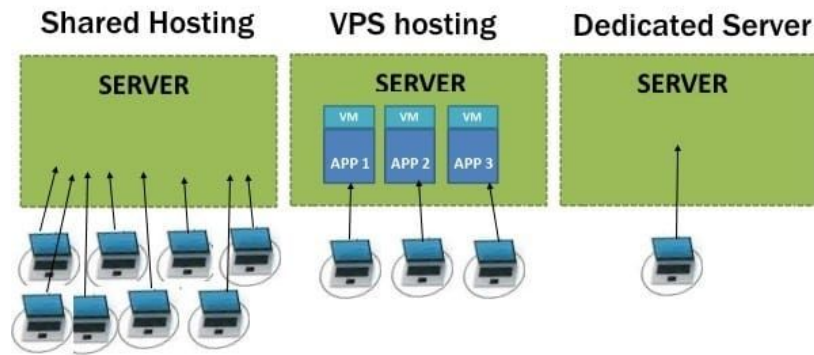
(1) Servidor propio: Compramos una máquina y la conectamos. condiciones:

- IP Fija, conexión de alta velocidad (no recomendable adsl)
- Servidor o daemon HTTP (Apache, Nginx ...)
- Si queremos email: servidor SMTP

(2) Servidor dedicado: Alquilamos una máquina a una empresa especializada en "hosting", ellos la configuran por nosotros y la conectan a internet a través de su canal.

(3) Servidor virtual o VPS: máquina virtual sólo para nosotros en un servidor de gran potencia, también alojada en una empresa de hosting.

(4) Alojamiento compartido: Panel de control para gestionar una instancia privada de servidor http, mysql y email, en un servidor de shared hosting. La opción más económica...





Registro DNS

Si tenemos el espacio web y el dominio contratado con el mismo proveedor (lo más recomendable al principio), la configuración de los registros DNS suele ser automática o muy simple.

Si son empresas diferentes tendremos que hacerlo manualmente, pero no suele ser complicado.

Sin embargo la gestión del hosting puede ser una tarea compleja. Renovar un dominio caducado, por ejemplo, puede ser un lío.

Como programadores, lo mejor es trabajar con empresas de hosting locales, que ofrezcan un servicio cercano y ágil. Y nosotros concentrarnos en hacer las mejores aplicaciones...



Alternativas de bajo coste para publicar

Hay muchas y cambian continuamente, algunas opciones:

Generadores de páginas web:

ej. 1and1.com

Generador de páginas basado en plantillas (sólo hay que poner textos y fotos)

Creación de la web y alojamiento por 1 eur / mes

Blogs:

Páginas web en un servidor **CMS**: Blogger, Wordpress ...

Tenemos nuestro espacio con subdominio, plantillas, panel de control ...

Gratuito, o con mayores prestaciones i dominio propio a cambio de cuota mensual.

Entornos específicos para desarrollo:

Espacio de trabajo y hosting para pruebas c9.io, codeanywhere.com, codesandbox...



Recursos para la programación web

Actualmente la programación, y en especial la programación web, consiste tanto o más en saber cómo combinar tecnologías y recursos existentes, como en crear nuevos contenidos.

Por tanto "no reinventar la rueda" debería ser uno de nuestros lemas como programadores. No re-escribir lo que ya esté escrito. Por dos motivos:

- Para ser más productivos, centrándonos en aquello que es importante y diferencial en nuestra aplicación.
- Para crear un código más sólido, menos falible y más fácil de mantener (al utilizar recursos muy probados y evolucionados).

Por otra parte, a medida que evolucionamos como programadores, iremos creando nuestra propia biblioteca de recursos, y eventualmente la podemos ofrecer a la comunidad.



Algunos recursos (gratuitos)

Recursos gráficos o de presentación:

- Frameworks o superestructuras CSS:
Bootstrap
- “Temas” o plantillas de diseño:
Themeforest...
- Bibliotecas de iconos o imágenes:
Fontawesome...
- Tipografías: Google fonts

Recursos de programación:

- Editores de código / IDEs
 - VS Code, Visual Studio, Eclipse, NetBeans
- Bibliotecas y frameworks front-end: jQuery (JS), React, Angular...
- Back-end: Laravel, Symfony, NodeJS...
- Máquinas virtuales: VMWare, Virtual Box...

Y nos falta la herramienta más importante para la programación web...



El navegador!

Es la principal herramienta del programador web. Los más recomendables actualmente son Chrome o Firefox. Además de la utilidad obvia de visualizar las páginas, incluyen unas herramientas específicas para el programador. Estas son algunas utilidades de las **herramientas de desarrollador de Google Chrome (Inspector)**:

- Visualizar y cambiar los estilos CSS de una página
- Acceder a todos los documentos de la página (html, css, js, tipos de letra...)
- Simular distintos tamaños de pantalla para ver el comportamiento responsivo
- "Debugar" javascript, visualizar mensajes y ejecutar funciones en la consola
- Analizar las prestaciones de la página (velocidad de carga, posibles cuellos de botella, simular conexiones lentas...)
- **Copiar y aprender!!!**



Anexo: el “posicionamiento”

El **posicionamiento** es el orden en que aparece la página en los buscadores, especialmente en Google. ¿Cómo calcula google el orden en que muestra los resultados? Esta no es la pregunta del millón... es la piedra filosofal y el santo grial juntos!

¿Por qué es tan importante? Fácil: para una empresa, la diferencia entre estar en la primera página de resultados cuando alguien busca "botifarres", o estar en la segunda, puede significar la diferencia entre hacerse rico o arruinarse.

Por ello, las empresas gastan fortunas en SEM, SEO y marketing digital. Para obtener un buen "posicionamiento".

SEM: Search Engine Marketing

SEO: Search Engine Optimization

SMM: Social Media Marketing



Ejercicio: Propuesta de dominio y hosting para página web

Por equipos de 3 ó 4.

Recibimos el encargo de crear una página web para una tienda de reparación de teléfonos móviles. Debemos inventar un nombre fácil de recordar, comprobar que el dominio no exista y proponer al menos dos alternativas de hosting al cliente, incluyendo el precio de compra del dominio y contratación del espacio.

Además: el cliente quiere también una cuenta en twitter y otra en facebook con el mismo nombre...