

UT1

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

Conceptos básicos de Internet

2

- ¿Qué es y para qué sirve Internet?
 - Es una gran red de ordenadores y como todas ellas sirve para compartir recursos e intercambiar información.
 - Ordenadores unidos a través de conexiones de varios tipos
 - Se utiliza el protocolo TCP/IP
 - Se ofrecen diferentes servicios a los usuarios: WWW, FTP, DNS, e-mail, SSH, etc...

Introducción

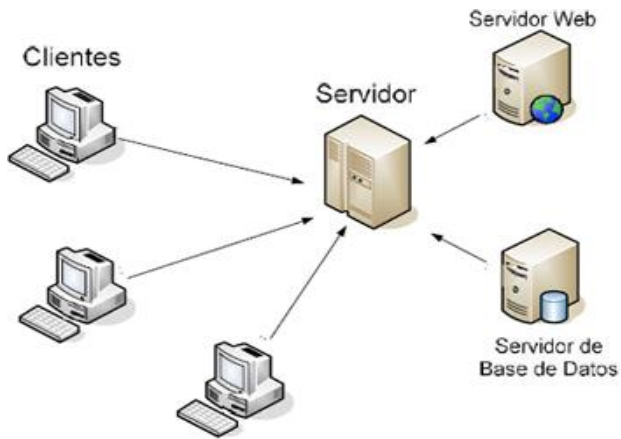
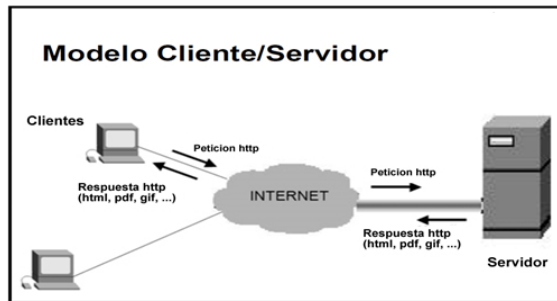
3

- Una web es un sistema formado por volúmenes de información organizados mediante recursos web (enlaces, vínculos que permiten relacionarse entre ellos...).
- Surge como idea original para la búsqueda de un sistema que permita acceder rápidamente a documentos electrónicos mediante enlaces.
- Se utiliza un lenguaje denominado HTML (**HyperText Markup Language**)
- HTML es el lenguaje con el que se "escriben" la mayoría de páginas Web
- El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium (<http://www.w3.org>) más conocido como **W3C**.
- El software para comprender e interpretar la información en un documento HTML se denomina navegador.

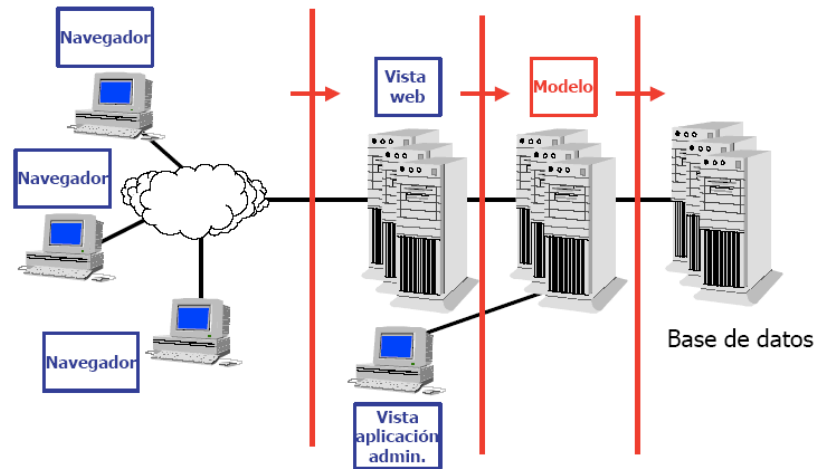
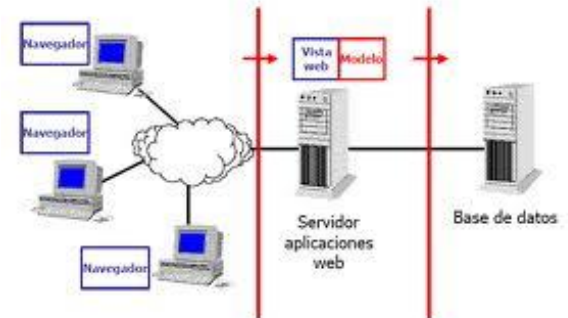


Modelos de programación en entornos cliente/servidor

4



Arquitectura Cliente Servidor



Capas: Presentación | Negocio | Datos

Generación dinámica de páginas web

5

- Podemos dividir las **aplicaciones web** en:
 - **Estáticas:** el usuario recibe una página web cuya interacción no conlleva ningún tipo de acción, ni en la propia página, ni genera respuesta alguna por parte del servidor. Usan lenguaje HTML exclusivamente.
 - **Dinámicas:** la interacción del cliente con el recurso recibido por parte del servidor (página web) produce algún tipo de cambio en la visualización del mismo (cambios de formato, ocultación de partes del documento, creación de elementos nuevos, etc.). Los lenguajes involucrados en este tipo de aplicaciones incluyen, entre otros, HTML, CSS o JavaScript.
 - **Interactivas:** la interacción hace que se genere un diálogo entre el cliente y el servidor. Desde el punto de vista del modelo de programación, la lógica asociada al inicio y gestión de dicho diálogo puede ser ejecutada tanto en el cliente como en el servidor (e incluso en ambos).
- Las más demandadas actualmente son las interactivas. Las tecnologías implicadas varían mucho en función de si son ejecutadas en el lado del cliente o en el lado del servidor

Lenguajes de programación utilizados.

Lenguajes de scripting

6

- Hay que diferenciar entre:
 - ▣ Lenguajes de programación del lado del cliente.
 - ▣ Lenguajes de programación del lado del servidor o lenguajes scripting.

Del lado del cliente

7

- **HTML:** solo presenta texto en una página, además de definir algunos aspectos visuales en el documento mediante el uso de estilos.
- **Hojas de estilo (CSS):** definen reglas de representación del contenido en colaboración con HTML.
- **JavaScript:** lenguaje de programación interpretado, permite el uso de efectos visuales y nuevas interacciones con el usuario. Se usa habitualmente a través de bibliotecas como jQuery, Angular o React.
- **XML:** lenguaje de programación auto descriptivo ampliamente utilizado para el intercambio de datos entre aplicaciones multiplataforma.
- **JSON:** surge como el gran rival del XML ya que es más ligero y legible.
- **Applets:** programas desarrollados en Java (no recomendado)
- **ActiveX:** controles que tienen un comportamiento parecido a los applets pero son desarrollados por Microsoft (no recomendado)
- **Adobe Flash** (problemas de usabilidad)

Lenguajes del lado del servidor

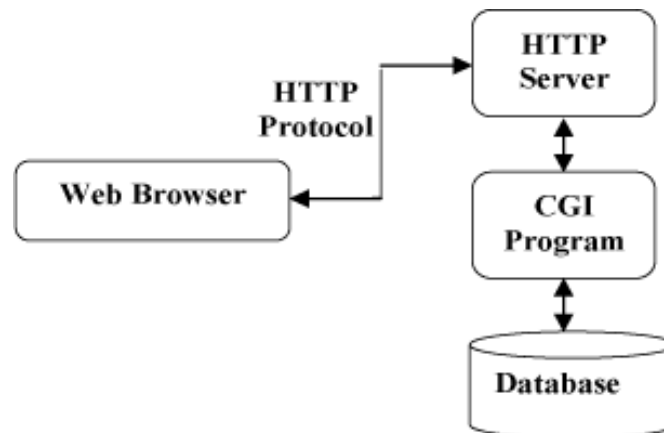
8

- El código de los lenguajes del lado del servidor es ejecutado por un software específico en el componente que actúa como servidor.
- Existen múltiples alternativas:
 - CGIs
 - Lenguajes específicos para web: PHP
 - Lenguajes dinámicos: Perl, Python, Ruby,...
 - JavaScript en servidor: NodeJS
 - Lenguajes generados compilados:
 - JVM: Java (JSP, Servlets, ...), Scala, Groovy
 - CLR: C# (ASP.NET), ...
 - Otros: frameworks, CMS, etc.

Lenguajes del lado del servidor

9

- **CGI** (Common Gateway Interface), 1.1 (2004)
 - Método estándar para transmitir parámetros entre servidor y programas ejecutables
 - Los programas CGIs son ejecutados por el servidor
 - Devuelve la respuesta de la ejecución
 - Problemas de escalabilidad y seguridad



Lenguajes del lado del servidor

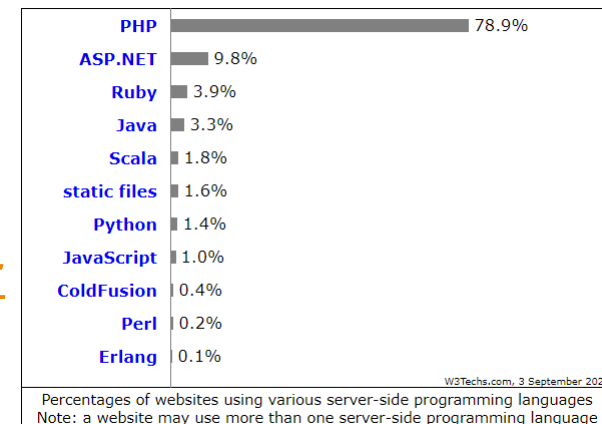
10

□ PHP

- Lenguaje interpretado por el servidor
- El código se incrusta en HTML mediante marcas especiales
- Cuando el servidor reconoce código PHP:
 - Llama al intérprete
 - Ejecuta el código
 - Devuelve el resultado

■ Según w3techs

(https://w3techs.com/technologies/overview/programming_language/all), el 79% de los sitios web utiliza PHP



Lenguajes del lado del servidor

11

□ Lenguajes dinámicos

- Python, Ruby
- Lenguajes interpretados de propósito general
- Buenos frameworks y librerías para Web
 - Ruby: Ruby on Rails
 - Python: Django



□ JavaScript del lado del servidor

- Librería Node.js
 - Basado V8, el motor JavaScript de Google
 - Permite utilizar JavaScript en el servidor
 - Entrada/salida basada en eventos
 - Creciendo en popularidad



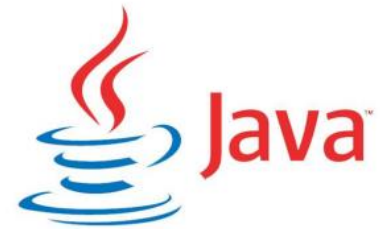
Lenguajes del lado del servidor

12

□ Lenguajes generales compilados

▣ Java (JVM)

- Máquina virtual de Java
- Empotrado: JSP
- Servlets y Contenedores de aplicaciones: Tomcat
- Otros lenguajes sobre JVM: Scala, Groovy, ...



▣ ASP .NET

- Basado en CLR
 - Máquina virtual de C#
- ASP Permite empotrar lenguaje en HTML
 - Extensión aspx <% código %>
- Diversos frameworks: ASP.Net MVC
- Otros lenguajes:
 - VB.Net, ...



Lenguajes del lado del servidor

13

□ Frameworks

- Normalmente basados en el patrón MVC
 - Separación de modelo, vista, controlador
- Numerosos frameworks
 - **PHP:** **Laravel**, Symfony, CakePHP, Zend
 - Ruby: Ruby on Rails
 - Python: Django, Zope
 - Java: Spring MVC, Play
 - Scala: Play, Lift
 - Groovy: Grails
 - Haskell: Yesod

Integración con los servidores web

14

- Los clientes hacen peticiones a los servidor.
 - En ellas tienen que constar de unos elementos concretos y especificados en un orden determinado.
 - Las direcciones de las peticiones suelen ser de tipo **URL** (Localizador Uniforme de Recurso):
 - Cadena de texto que se utiliza para identificar un recurso y además nos da información sobre como acceder a él, como localizarlo.
 - Formato URL:

Parte de la URL	Descripción	Ejemplo
Servicio:	Indica el servicio o protocolo a usar: http, https, ftp, telnet, ...	http:
//	Separador	//
Servidor	Indica la IP o el nombre del servidor	www.ubuntu.com
Ruta	Indica el directorio o subdirectorios donde reside en recurso.	/desktop
Recurso	Recurso al que se quiere acceder.	/index.html

Integración con los servidores web

15

- Existen distintos modos o métodos para intercambiar información entre cliente y servidor:
 - ▣ Método **GET**: petición de información
 - GET consultatelefono.php?cliente=empresa1
 - Los parámetros se pasan como pares nombre=valor
 - Se pueden pasar varios parámetros seguidos con &
 - ▣ Método **POST**:
 - Se usa habitualmente para enviar información a un servidor web.
 - Enviar el contenido de un **formulario** de autenticación
 - Entradas de datos
 - Especificar parámetros para algún tipo de componente ejecutado en el servidor.

Herramientas de programación

16

- **Navegadores**
- **Editores de documentos:** este grupo está formado por editores de texto que permiten escribir código HTML directamente, sin ninguna ayuda ni facilidad adicional.
- **Entornos de programación:** entornos integrados que nos permiten editar, compilar y ejecutar los programas generados a partir de diferentes lenguajes usados en el desarrollo de las aplicaciones web.
- **Herramientas de tratamiento de imágenes:** la mayoría de las páginas web muestran contenido gráfico de una u otra manera. Es necesario el uso de este tipo de herramientas para adecuar las características de las imágenes a su transmisión.
- **Herramientas para la creación y administración de bases de datos:** herramientas para la carga de datos y el mantenimiento posterior de los datos almacenados.