U.T. 1: Arquitecturas y herramientas de programación en entorno cliente/servidor.

Contenidos

- Modelos de programación en entornos cliente / servidor.
- Generación dinámica de páginas web.
- Lenguajes de programación en entorno servidor.
- Integración con los servidores web.
- Herramientas de programación.

La web

- ***World Wide Web***:
  - Universo de información interconectada, accesible a través de internet.
- Propuesta por Tim Berners-Lee (1989).
- Ha tenido mayor difusión que otros servicios contemporáneos (Archie, Gopher, WAIS), gracias sobre todo a uno de sus elementos: el HTML.
  - Hipertexto.
  - Hipermedia.

Arquitectura cliente/servidor

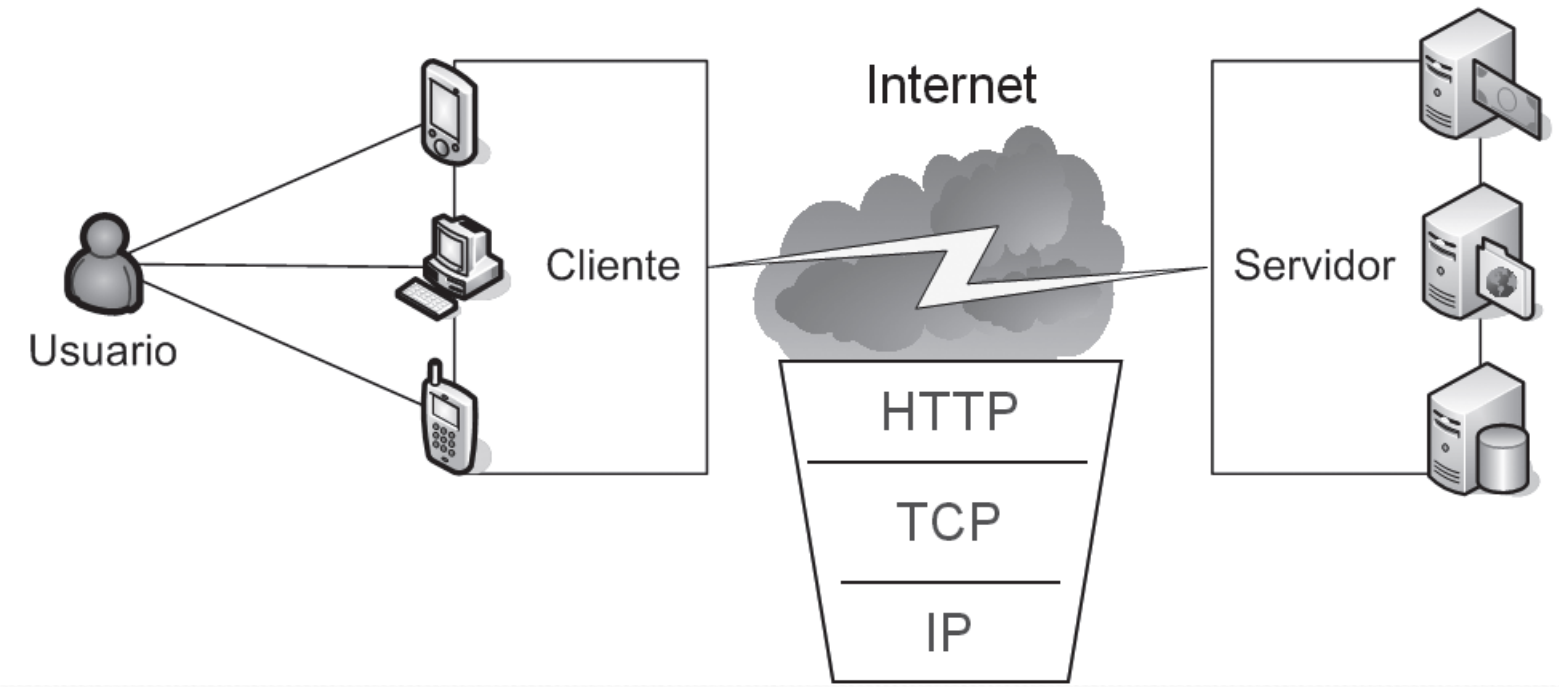
El funcionamiento de la Web es posible gracias a la coexistencia de una serie de componentes hardware y software:

- hubs, repetidores, puentes, pasarelas, enrutadores, etc.
- protocolos de comunicaciones (TCP, IP, **HTTP**, FTP, SMTP, etc.)
- sistema de nombres de dominio (DNS) para la búsqueda y utilización de recursos => **URL** del recurso solicitado.
- software específico para consumir los recursos

Arquitectura cliente/servidor

- La arquitectura cliente/servidor está basada en la idea de servicio.
- El ***cliente*** es un componente consumidor de servicios.
- El ***servidor*** es un proceso proveedor de dichos servicios.
- La comunicación entre ambos componentes se lleva a cabo a través del intercambio de mensajes.
- Normalmente el cliente, a través de un navegador, inicia el intercambio de información, solicitando datos al servidor.
- El servidor responde enviando uno o más flujos de datos al cliente.

Arquitectura cliente/servidor



Arquitectura cliente/servidor: Clientes

- Originan el tráfico web.
  - Envían las peticiones y reciben las respuestas.
- Dos clases de clientes web: navegadores y robots.
- Los navegadores (Netscape, IE, Chrome, Opera, FireFox, Safari).
  - Las peticiones están dirigidas por el usuario.
  - Repiten peticiones al mismo objeto cuando navegan por un site.
  - Utilizan caches de memoria y disco.
- Robots (Motores de búsqueda)
  - Las peticiones son automatizadas.

Arquitectura cliente/servidor:

Funciones de los navegadores

- Construyen y envían la petición HTTP.
- Reciben, interpretan y presentan la respuesta.
- Proporcionan el interfaz para conectarse y utilizar otros servicios: mail, news, ftp, etc.
  - El protocolo por defecto es https.
- Caché local.
  - Sirve recursos guardados en la caché sin conectarse al servidor.
- Manejo de las Cookies.

Arquitectura cliente/servidor: Servidores

Su cometido básico es proveer de contenido estático a un cliente que ha realizado una petición a través de un navegador.

El servidor es un lugar donde almacenar y administrar los recursos HTML (DHTML).

Arquitectura cliente/servidor: Servidores

Las peticiones al servidor contienen una dirección de tipo URL (Localizador Uniforme de Recurso) formada por:

- la referencia a un cierto protocolo (HTTP, FTP, etc.)
- la dirección IP o nombre de dominio del servidor
- la descripción del recurso en forma de ***ruta*** al objeto que queremos descargar.
- opcionalmente se puede incluir el puerto por el que el servidor escucha las peticiones del cliente.

Arquitectura cliente/servidor: Servidores

- Programa que contesta y genera la respuesta HTTP a las peticiones de recursos web por parte del cliente.
  - Involucra múltiples servidores, scripts, bases de datos, ..
- Trabajo básico:
  1. Se conecta con el cliente.
  2. Recibe el mensaje HTTP de la petición (GET, HEAD, POST)
  3. Procesa el mensaje HTTP.
  4. Localiza y envía el resultado (en forma de mensaje HTTP)
- Los servidores de altas prestaciones, además:
  - Tratan múltiples peticiones:
    - hilos para manejar cada conexión.
  - Generan dinámicamente contenido: ASP, PHP, JSP

Arquitectura cliente/servidor: Servidores

- Los más populares son :
  - Apache HTTP Server
  - Microsoft IIS (Internet Information Services)
  - NGINX
  - Lighttpd
  - Sun Java System Web Server

Ejercicio: Investiga las características y diferencias de cada uno de éstos servidores web: plataforma en la que se ejecutan, lenguajes de programación que interpretan, propietario, etc.

Arquitectura cliente/servidor: Servidores

El intercambio de información cliente/servidor puede requerir también:

- Controlar el acceso a recursos restringidos:
  - Autenticación.
    - Piden al usuario que se identifique (login y password)
    - La información se incluye en la cabecera del mensaje.
  - Autorización.
    - Comprobar en la lista de acceso si el usuario está autorizado.

Aplicaciones web

- Una aplicación web es proporcionada por un servidor web y utilizada por usuarios que se conectan desde cualquier punto vía clientes web (navegadores).
- Definición:
  - Son aplicaciones basadas en el modelo Cliente/Servidor gestionadas por servidores web, y que utilizan como interfaz páginas web.
  - La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario.
  - El acceso a las páginas está agrupado también en el tiempo (sesión).
  - Ejemplos: venta de libros, reserva de billetes, etc.

Arquitectura de las aplicaciones web

- Las funcionalidades en la arquitectura cliente/servidor de la web se agrupan en diferentes capas.
- Cada capa se centra en la gestión de una aspecto determinado del sistema web.
- El modelo más extendido divide la arquitectura cliente/servidor en tres capas:
  - Capa de presentación
  - Capa de negocio
  - Capa de persistencia o de datos

Arquitectura de las aplicaciones web

- Capa de presentación
  - Es la capa que ve el usuario
  - Presenta la interfaz gráfica del recurso solicitado
  - Sirve para recoger la interacción con el usuario
  - Suele estar situada en el cliente

Arquitectura de las aplicaciones web

- Capa de negocio
  - Gestiona y conoce las funcionalidades que esperamos del sistema web.
  - Recibe las peticiones del usuario, las procesa y envía las respuestas apropiadas.
  - Suele estar programada tanto en el cliente como en el servidor.

Arquitectura de las aplicaciones web

- Capa de persistencia o de datos
  - Encargada del almacenamiento y acceso a los datos
  - Está formada por uno o más gestores de bases de datos que administran y resuelven las solicitudes de almacenamiento y recuperación de información desde la capa de negocio.

Generación de páginas web

- Al observar la capacidad de las aplicaciones web de comunicarse con los usuarios, las podemos clasificar en:
  - Aplicaciones web estáticas => Páginas web estáticas
  - Aplicaciones web dinámicas => Páginas web dinámicas
  - Aplicaciones web interactivas

Generación de páginas web

- Si lo combinamos con el concepto de página web:

Arquitectura	Cliente	Servidor
Página web estática	Estático. HTML y CSS	Estático. Recursos en disco duro
Página web interactiva	Dinámico. JavaScript	Estático. Recursos en disco duro
Aplicación web con cliente estático	Estático. HTML y CSS	Dinámico. Ejecución código
Aplicación web interactiva	Dinámico. JavaScript	Dinámico. Ejecución código
Aplicación web con AJAX	Dinámico. JavaScript	Dinámico. Ejecución código
Aplicación web SPA	Dinámico. JavaScript	Dinámico. Ejecución código

Generación de páginas web

- Páginas web estáticas
  - El usuario recibe una página web desde el servidor, que no conlleva ningún tipo de acción, ni en la propia página, ni genera ninguna respuesta por parte del servidor.
  - Utilizan HTML para la organización visual de la información.

Generación de páginas web

- Páginas web estáticas
  - El navegador hace petición al servidor mediante HTTP
  - El servidor transforma URL a ruta en disco
  - El servidor devuelve el fichero de disco al navegador
  - El navegador visualiza (renderiza) la página HTML con estilos CSS e imágenes (sin JavaScript).
  - Cuando el usuario hace clic en un enlace, el navegador repite el proceso con la URL del link y recarga por completo la página web

Generación de páginas web

- Páginas web estáticas
  - Con esta arquitectura el servidor siempre devuelve los mismos recursos
  - Desde el punto de vista del servidor, la web es estática
  - La web está formada por HTML, CSS, imágenes, PDF, etc… (pero no incluye JavaScript)
  - La Web (WWW) se diseñó con esta arquitectura
  - Al principio todas las páginas web eran así (no existía el concepto de *aplicaciones web*)

Generación de páginas web

- Páginas web estáticas
  - Actualmente esta arquitectura se usa principalmente para:
    - Páginas personales
    - Páginas de proyectos software
    - Documentación técnica (JavaDoc en Java, Maven site, etc…)

Generación de páginas web

- Páginas web estáticas

```
001  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
002  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-ES" lang="es-ES">
003     <head>
004        <title>T&iacute;tulo de la p&aacute;gina</title>
005        <meta http-equiv="content-type" content="text/html; charset=utf-8">
006     </head>
007     <body>
008        <p>Ha accedido a las 9:20</p>
009     </body>
010  </html>
```

Generación de páginas web

- Páginas web interactivas (dinámicas)
  - El contenido de la página web está alojado en el disco duro del servidor (estático)
  - El cliente es dinámico porque las páginas incluyen código JavaScript que se ejecuta en el navegador
  - Este JavaScript se usa para incluir efectos gráficos:
    - Efectos gráficos que no se pueden implementar con CSS
    - Mostrar u ocultar información en función de los elementos que se seleccionan (para documentos largos)
    - Menús desplegables
    - Páginas adaptables para móviles (responsive)

Generación de páginas web

- Páginas web dinámicas
  - La interacción del cliente con la página web recibida desde el servidor produce algún cambio en la visualización de la misma (cambio de formato, ocultación de partes de la página, comienzo de animaciones, aparición de elementos nuevos, …).
  - Incluyen DHTML, Flash, CSS, JavaScript, …

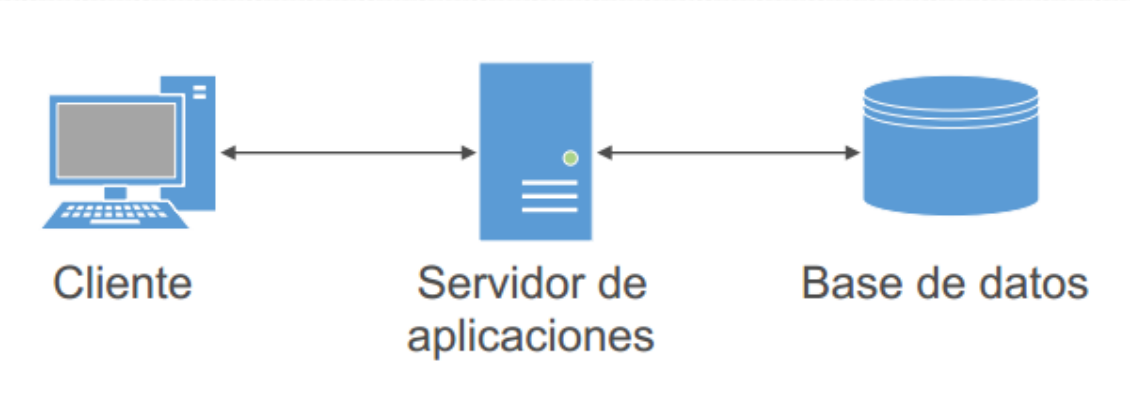
Generación dinámica de páginas web

- Páginas web dinámicas

```
001  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
002  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-ES" lang="es-ES">
003          <head>
004                  <title>T&iacute;tulo de la p&aacute;gina</title>
005                  <meta http-equiv="content-type" content="text/html; charset=utf-8">
006          </head>
007          <body>
008                  <p>Ha accedido a las <script type='text/javascript'>var hora=new
     Date();document.write(hora.getHours()+':'+hora.getMinutes());</script></p>
009          </body>
010  </html>
```

Generación de páginas web

- Aplicación web con cliente estático



- Es un ejemplo de arquitectura de 3 capas:
  - Navegador: Capa de presentación
  - Servidor web: Capa de aplicación (lógica de negocio)
  - Base de datos: Capa de datos

Generación de páginas web

- Aplicación web con cliente estático
  - Cuando el servidor web recibe una petición, dependiendo de la URL, puede:
    - Devolver contenido del disco
    - Ejecutar código para generar el recurso dinámicamente
  - Cuando se ejecuta código, normalmente se hacen consultas a una base de datos para recuperar la información
  - Lo más habitual es que se genere la página HTML de forma dinámica (pero también puede generar imágenes, PDFs, etc…)
  - Si el usuario pulsa un link, se recarga la página al completo

Generación de páginas web

- Aplicación web con cliente estático
  - Es la arquitectura de las primeras aplicaciones web
  - Todavía sigue habiendo muchas webs con esta arquitectura
  - El contenido es dinámico, porque se ejecuta código en el servidor para generar dicho contenido
  - La experiencia de usuario antes no era muy buena:
    - Conexiones lentas implican tiempos de carga apreciables en cada click
    - La recarga completa de la página ofrece una mala experiencia de usuario (página en blanco)
  - Pero ha mejorado:
    - Mayor velocidad de Internet (menos tiempo de espera)
    - Navegadores muestran la nueva página una vez cargada (sin pasar por la página en blanco)

Generación dinámica de páginas web

- Aplicación web con cliente estático

```
001  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
002  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-ES" lang="es-ES">
003          <head>
004                  <title>T&iacute;tulo de la p&aacute;gina</title>
005                  <meta http-equiv="content-type" content="text/html; charset=utf-8">
006          </head>
007          <body>
008                  <p>Ha accedido a las <?php echo date('G:i');?></p>
009          </body>

010  </html>
```

Generación de páginas web

- La mayoría de las aplicaciones web actuales son dinámicas tanto en cliente como en servidor
- Dependiendo de cómo se use el JavaScript en el cliente se diferencian tres arquitecturas:
  - Aplicación web interactiva
  - Aplicación web con AJAX
  - Aplicación web SPA
  - Se muestra la nueva página una vez cargada (sin pasar por la página en blanco)

Generación dinámica de páginas web

- Aplicaciones web interactivas
  - La interacción del cliente con la página web recibida desde el servidor hace que se genere un diálogo entre ambos.
  - Son las aplicaciones que más se utilizan en Internet actualmente.
  - En el lado cliente encontramos HTML, controles ActiveX, Flash, applets, AJAX, etc.
  - En el lado servidor se utilizan lenguajes embebidos en código HTML como PHP, ASP, JSP, enlaces a ejecutables CGI, servlets, ASP.net.

Generación dinámica de páginas web

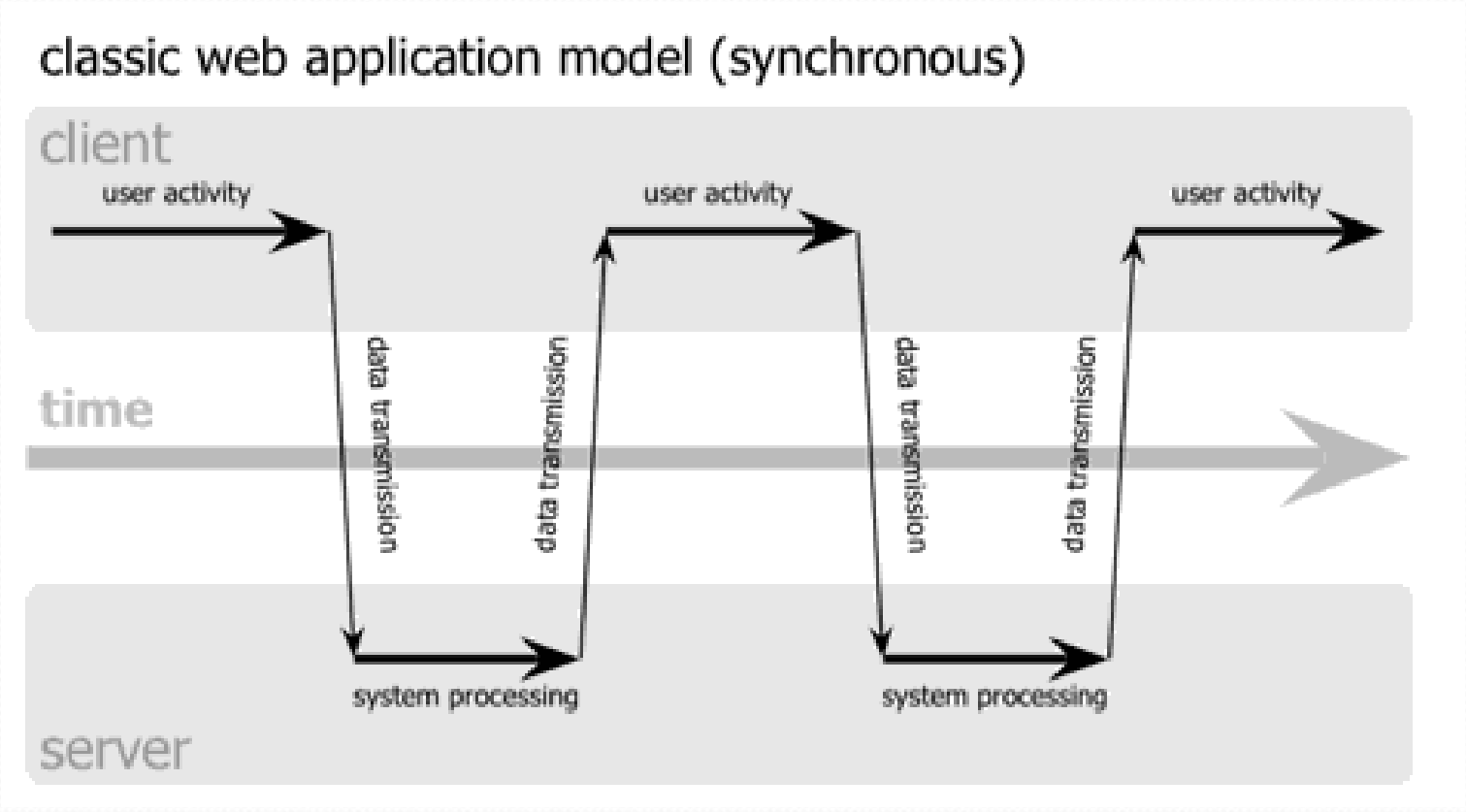
- Aplicaciones web interactivas
  - JavaScript se utiliza para crear efectos gráficos
  - El dinamismo en el cliente se utiliza exactamente igual que en las páginas web interactivas.
  - JavaScript se diseñó, entre otras cosas, para añadir efectos gráficos básicos a las páginas cuando el CSS era muy limitado
  - La gran mayoría de las aplicaciones web que existen en Internet siguen esta arquitectura

Generación dinámica de páginas web

- Aplicaciones web con AJAX
  - JavaScript se usa para no tener que **recargar completamente** la página al pulsar un link
  - Permite hacer petición al servidor web en **segundo plano** (oculta al usuario)
  - Cuando llega al navegador el resultado de la petición, el código JavaScript actualiza aquellas partes de la página necesarias
  - A esta técnica se la conoce como **AJAX (*Asynchronous JavaScript And XML*)**

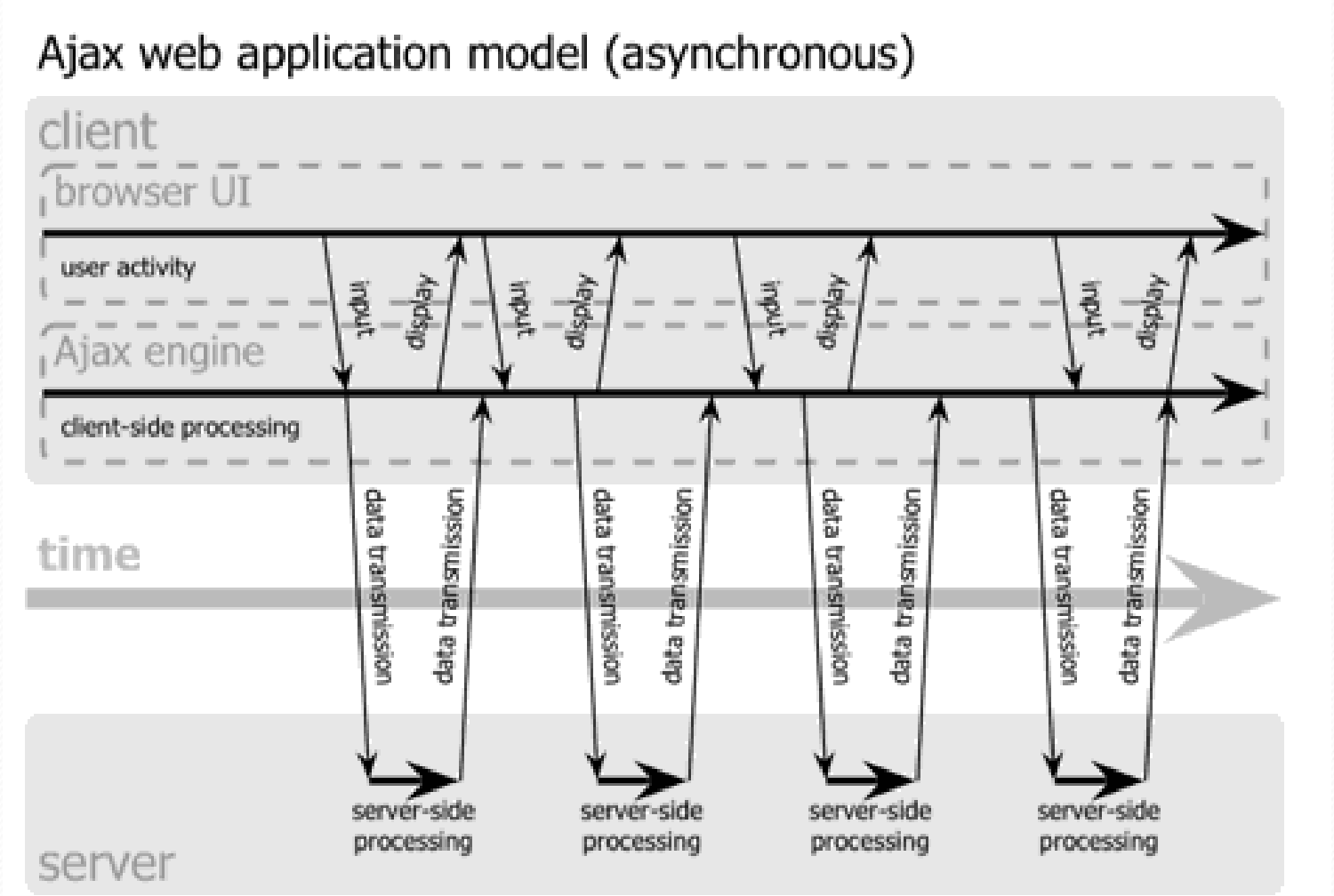
Generación dinámica de páginas web

- Aplicaciones web con AJAX



Generación dinámica de páginas web

- Aplicaciones web con AJAX



Generación dinámica de páginas web

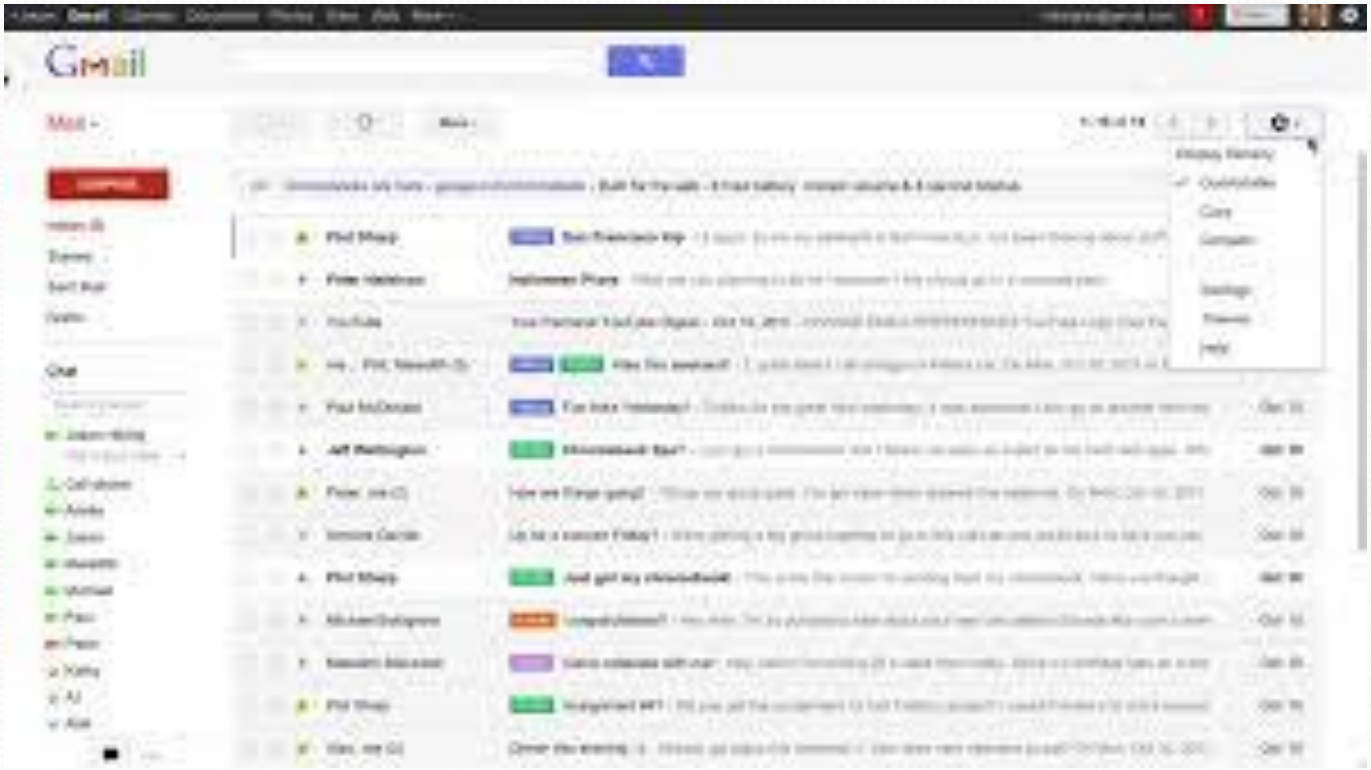
- Aplicaciones web con AJAX
  - Usar **AJAX** en una página mejora mucho la **experiencia de usuario**
  - No es necesario recargar la página al completo, sólo aquellas partes que cambian (p.e. se puede dejar el menú fijo).
  - La página se puede cargar por partes, primero la información importante y en segundo plano otros elementos complementarios (p.e. los botones de compartir, los comentarios en un blog…)
  - Se puede dar **realimentación** al usuario de formas más adecuadas (cuadro de diálogo, error de validación en un formulario, quitar el icono de carga de un recurso, etc…)

Generación dinámica de páginas web

- Aplicaciones web con SPA
  - SPA (Single Page Application)
  - La técnica AJAX se puede llevar al extremo y que todo el contenido dinámico se cargue con JavaScript en segundo plano.
  - Existe una única página cuyo contenido va cambiando según el usuario interactúa con botones, pestañas, etc.
  - El botón de atrás del navegador funciona porque se “emula” una navegación por páginas cuando se evoluciona por los estados de la aplicación

Generación dinámica de páginas web

- Aplicaciones web con SPA
  - Google popularizó AJAX y SPA con Gmail y Maps



Lenguajes de programación en entorno servidor

Son aquellos cuyo código, precompilado o interpretado, es ejecutado en el servidor por un software específico para dicho código.

Existen múltiples alternativas a la hora de ejecutar código en el servidor:

- Lenguajes de scripting (PHP, ASP, JSP, …)
- Enlaces a código ejecutable (CGI, JSP, EJB, …)
- Estrategias híbridas que utilizan tanto enlaces a código ejecutable como código embebido en la propia página web (Web Forms de ASP.net).

Lenguajes de scripting

- Código que se intercala con el código HTML de una aplicación web.
- El código está formado por instrucciones escritas en multitud de lenguajes de programación que son ejecutadas por un servidor para proporcionar dinamismo a la página web.
- El servidor web debe tener instalado un módulo o programa que le permita interpretar el lenguaje de programación del código embebido en la página web.

Lenguajes de scripting

- PHP (Hypertext Processor)
  - gratuidad, código abierto, portable en diferentes plataformas
  - lenguaje interpretado
  - construcciones orientadas a objetos
  - lo soportan la mayoría de los servidores web
- ASP (Active Server Pages)
  - tecnología propietaria de código cerrado de Microsoft.
  - diseñado para ejecutarse especialmente sobre IIS (Internet Information Server)
  - actualmente este lenguaje ha dejado paso a la versión ASP.net

Lenguajes de scripting

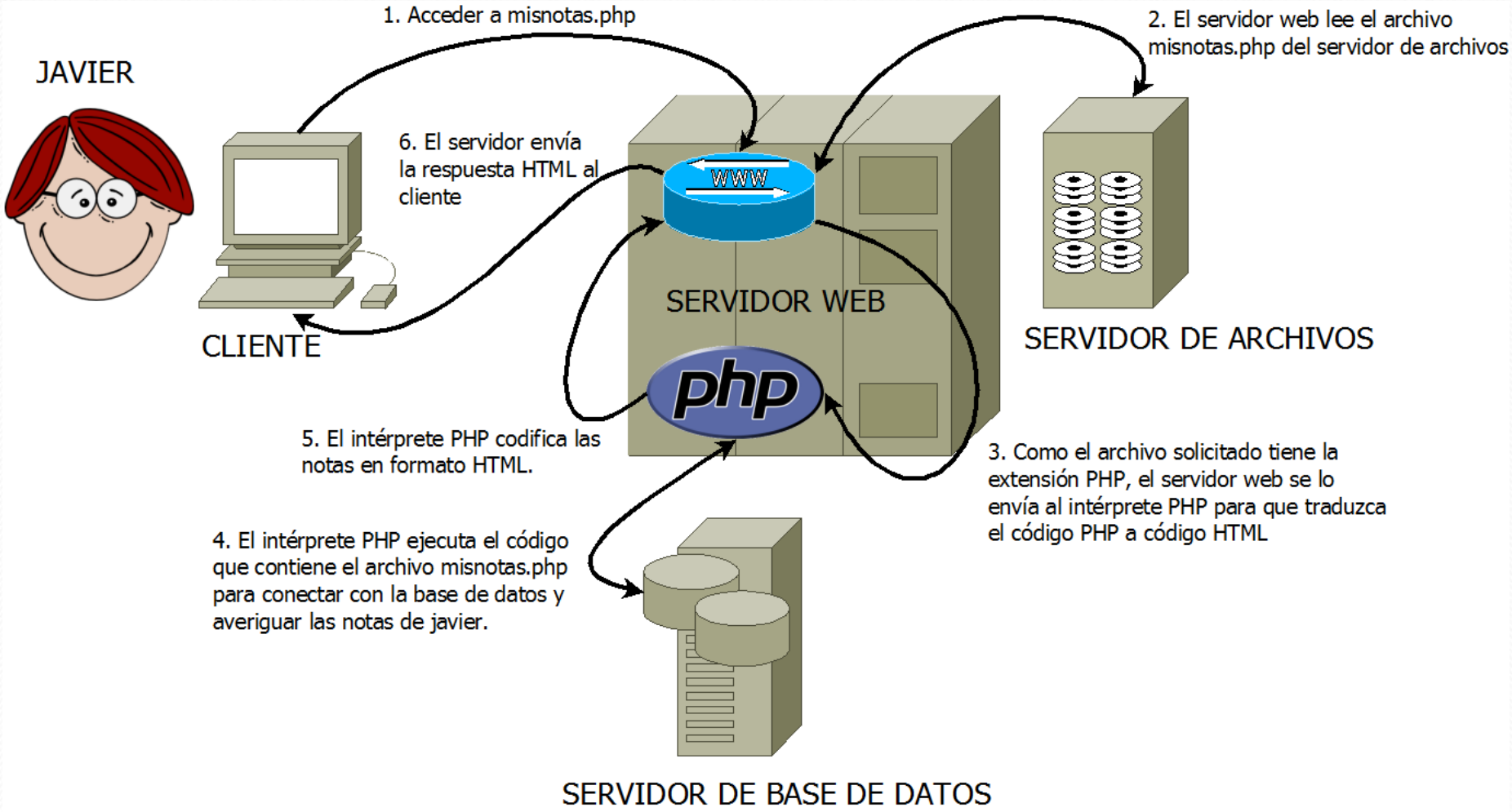
- Perl
  - inicialmente se utilizó para manipular cadenas de caracteres.
  - primeros lenguajes para la programación web.
  - código abierto
  - basado en programación estructurada como C
- Python
  - portable en diferentes plataformas y gratuito
  - orientado a objetos
  - lenguaje interpretado

Lenguajes de scripting

- JSP (Java Server Pages)
  - porciones de código java intercalado con HTML estático
  - el código java embebido se denomina servlet
  - el servlet se carga en la memoria del servidor web al ser ejecutado la primera vez, para mejorar su ejecución en posteriores llamadas a dicha página web.

Lenguajes de scripting

El proceso que se realiza a la hora de visitar una página PHP sería:



Aplicaciones CGI y derivados

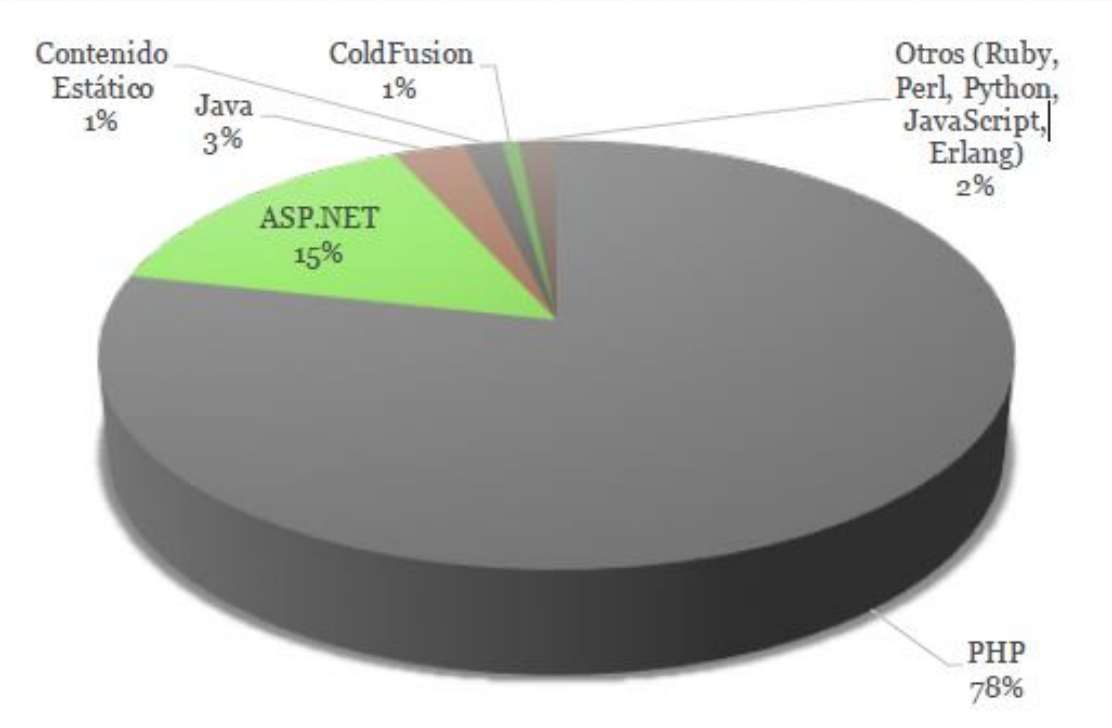
- CGI (Common Gateway Interface) es la forma más simple de crear páginas web dinámicas.
- Un programa independiente al servidor web devuelve como resultado el contenido que debe visualizar el cliente (la página web resultante) a partir de ciertos parámetros de entrada.
- La ubicación del programa a ejecutar se indica en la URL que forma la petición HTTP del cliente.

Aplicaciones híbridas

- Tecnología intermedia entre los lenguajes de scripting y los programas CGI.
- Plataforma de Microsoft.Net Framework con el lenguaje ASP.Net.
- ASP.Net es una tecnología totalmente orientada a objetos, que permite la creación de páginas web dinámicas utilizando lenguajes como VB.Net, C#, Jscript.Net.
- Las páginas ASP.Net están contenidas en ficheros.ASPX que son los que el cliente solicita a través de una URL al servidor.

Tecnologías del servidor

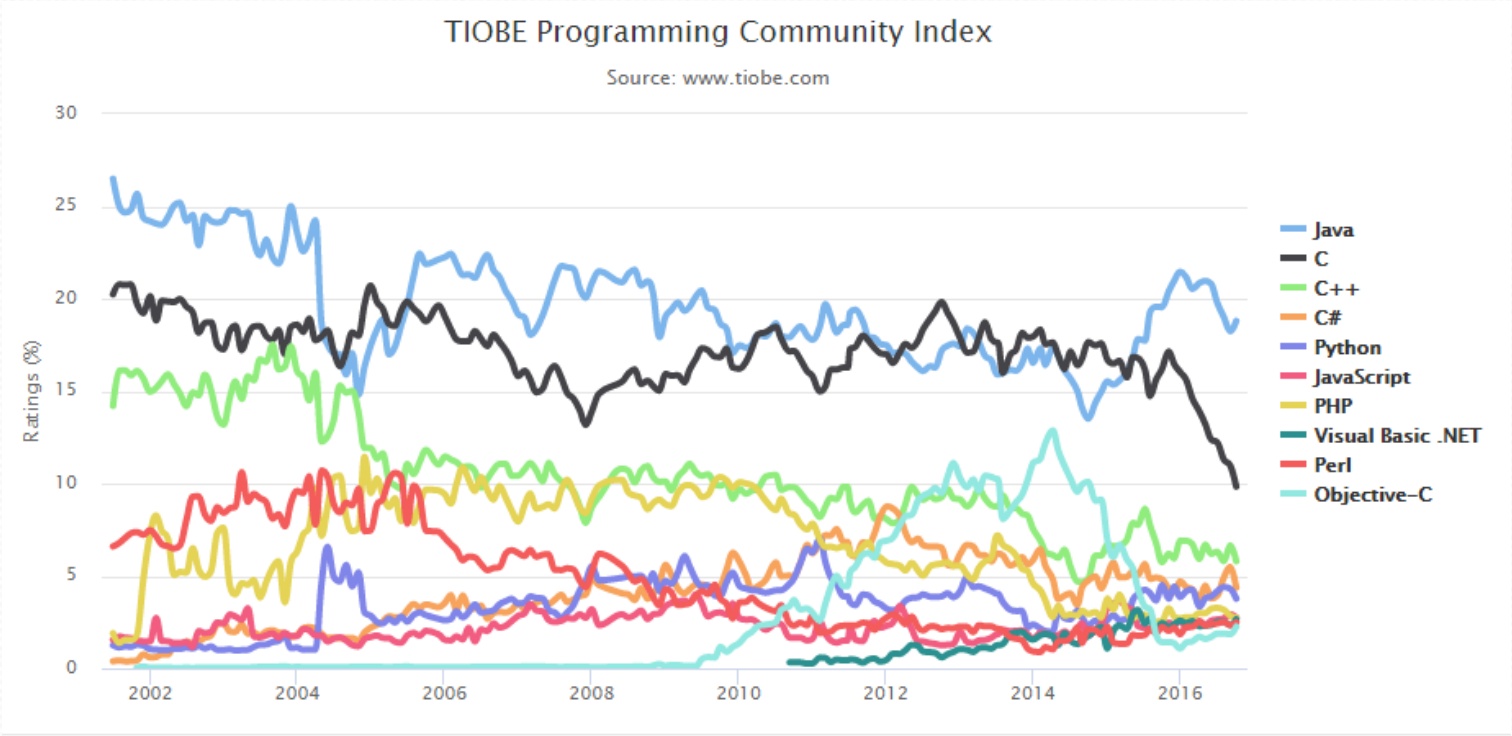
- Cuota de uso tecnologías del servidor (octubre 2015)



- http://w3techs.com/technologies/overview/programming_language/all

Tecnologías del servidor

- Índice TIOBE (septiembre 2016):



- http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Herramientas de programación

Los instrumentos involucrados en el desarrollo web se pueden clasificar en:

- Navegadores: Internet Explorer, Google Chrome, Mozilla firefox, etc.
- Editores de documentos
- Entornos de programación: Eclipse, NetBeans, CodeIgniter, Dreamweaver, FrontPage,…
- Herramientas para la creación y administración de bases de datos
- Herramientas para el tratamiento de imágenes.

Herramientas de programación

El servidor web debe contar con el software necesario para responder a las peticiones del cliente y ejecutar el código embebido dentro del recurso solicitado.

Existen paquetes de libre distribución que incluyen todo el software necesario para el desarrollo web:

- appServer : www.appservnetwork.com
- xampp: www.apachefriends.org
- wamp: www.wampserver.com
- lamp: www.apachefriends.org

Herramientas de programación

Todos los paquetes de instalación de un entorno para el desarrollo de aplicaciones web suelen contener el siguiente software:

- Apache HTTP Server
- MySQL
- PHP
- PhpMyAdmin

Ejercicio: Investiga qué versión de cada una de estas herramientas se instalarán en el paquete LAMP de Ubuntu 14.04 LTS.

Programa de prueba

Almacenar el siguiente fichero en el directorio:

C:\xampp\htdocs\prueba.php ó

/var/www/html/prueba.php

```
<?php
Phpinfo();
?>
```

Solicitarlo a través del navegador desde la URL:

http://localhost/prueba.php