INTRODUCCIÓN

• Ventajas de las aplicaciones web:

– Las aplicaciones web tiene varias ventajas sobre las aplicaciones tradicionales:

• Compatibilidad.

• Las aplicaciones web utilizan el navegador del cliente como interfaz de usuario.

• El lenguaje HTML garantiza la compatibilidad en distintas plataformas. – Accesibilidad

• Acceso remoto es un pre-requisito de diseño.

• Los dispositivos móviles están generalmente soportados.

• Hay muchas soluciones a nivel de navegador para personas con discapacidad:

– Lectores automáticos de texto.

– Gran variedad de dispositivos de entrada (teclados, ratones, micrófonos…) – Tamaños y colores de texto ajustables.

INTRODUCCIÓN

• Inconvenientes del HTML:

– Las arquitecturas y técnicas de

programación web buscan lo siguiente: • Máxima compatibilidad con los navegadores (potenciales clientes). • Eficiencia del lado del servidor (soportar el máximo número de conexiones concurrentes).

• Creación de contenidos llamativos y dinámicos sin sacrificar las dos anteriores.

**¡HTML falla en este último punto!**

(Si quieres conocer cómo era el WWW del pasado, visita https://archive.org/web/)

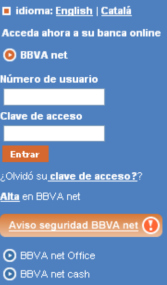
INTRODUCCIÓN

• Inconvenientes del HTML: 

– Se necesitan lenguajes de 

programación más potentes

que permitan “mini”

aplicaciones dentro de una 

web, como pueden ser:

• Carro de la compra y

comercio electrónico

• Contador de visitas

• Web Mail

• Registro y entrada a un

sistema

• Transferencias y

operaciones bancarias

• Etc.

DESARROLLO WEB EN ENTORNO SERVIDOR

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

• Modelos de programación en entornos cliente/servidor.

• Generación dinámica de páginas web. CGI (Common Gateway Interface).

• Lenguajes de programación en entorno servidor.

• Integración con los lenguajes de marcas. • Integración con los servidores web. • Herramientas de programación.

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

**MODELOS DE PROGRAMACIÓN CLIENTE-SERVIDOR**

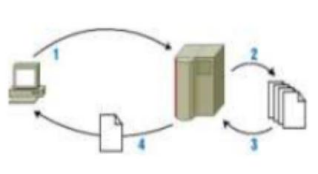
MODELOS DE PROGRAMACIÓN CLIENTE-SERVIDOR

• Introducción: ¿qué sucede

cuando visitamos una web?

– Página web (HTML) + Estilo

(CSS) + Funciones dinámicas

(JS) 

– Ejemplo de comunicación

cliente-servidor

1. **Un ordenador solicita a un**

**servidor web una página** con

extensión htm/html/xhtml

2. **El servidor busca esa página**

en el almacenamiento

(típicamente asociada a un

fichero)

3. **El servidor recupera la página**

**y los recursos** asociados a ella.

4. **Se envía al navegador** para

que pueda mostrar su contenido.

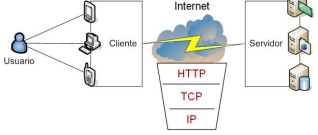
MODELOS DE PROGRAMACIÓN CLIENTE-SERVIDOR

• World Wide Web:

– Conjunto de recursos

interconectados que

conforman el

conocimiento humano 

actual.

• Hubs, repetidores, puentes

(o bridges), gateways (o

pasarelas), routers (o

encaminadores).

• Protocolos de

comunicaciones: TCP, IP,

HTTP, FTP, SMTP.

• Sistema de nombres de

dominio (DNS).

MODELOS DE PROGRAMACIÓN CLIENTE-SERVIDOR

• Configuración de

arquitectura más habitual es Cliente/Servidor (2- tier):

– El **cliente** es un

componente **consumidor**

de servicios.

– El **servidor** es un proceso **proveedor** de servicios.

• No es la única

arquitectura: n-tier o P2P.

• Configuración 1-tier: MS-Access

• Configuración 2-tier: Cajero

• Configuración 3-tier: web de venta de

billetes

• Configuración n-tier: raras excepciones. En la práctica son modelos 3- tier.

MODELOS DE PROGRAMACIÓN CLIENTE-SERVIDOR

• Funcionalidades en los entornos

cliente/servidor (capas):

– **Capa de presentación**: es la que ve el usuario. – **Capa de negocio**: es la capa que conoce y gestiona las funcionalidades que esperamos del sistema o aplicación web.

– **Capa de persistencia o de datos**: es la capa donde residen los datos y la encargada de acceder a los mismos.

MODELOS DE PROGRAMACIÓN CLIENTE-SERVIDOR

• Clasificación de los modelos de programación Web: – **Según el tamaño de los componentes**:

• Fat Client (Thin Server), el mayor peso de la aplicación se ejecuta en el cliente.

• Thin Client (Fat Server), funcionalidad asociada al cliente limitada a la presentación de la información enviada por el servidor.

– **Según la naturaleza del servicio ofrecido**: en función de las capacidades ofrecidas por el servidor.

• Servidores de ficheros, servidores de bases de datos, servidores de transacciones, servidores de objetos, servidores web…

– **Reparto de funciones entre cliente y servidor**: gestionar y distribuir las responsabilidades de cada una de las prestaciones funcionales entre el cliente y el servidor:

• Modelo en dos capas.

• Modelo en tres capas.

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

**GENERACIÓN DINÁMICA DE PÁGINAS WEB. CGI (COMMON GATEWAY INTERFACE)**

GENERACIÓN DINÁMICA DE PÁGINAS WEB

• Clasificación de las aplicaciones Web: – **Aplicaciones web estáticas**: el usuario recibe una página web cuya interacción no conlleva ningún tipo de acción.

– **Aplicaciones web dinámicas**: la interacción del cliente con el recurso recibido por parte del servidor (página web) produce algún tipo de cambio en la visualización o el contenido del mismo.

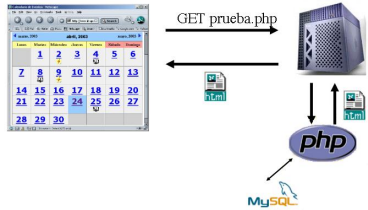
– **Aplicaciones web interactivas**: la interacción del usuario hace que se genere un diálogo entre el cliente y el servidor.

GENERACIÓN DINÁMICA DE PÁGINAS WEB

• Código en el lado del cliente



• Código en el lado del servidor



GENERACIÓN DINÁMICA DE PÁGINAS WEB

• Web estática: ventajas – Simplicidad.

– Fácil construcción,

económicas (HTML +

CSS).

– Rapidez. Procesos de back-end mínimos o

nulos.

– Rendimiento. La falta de variación de contenido favorecen su

almacenamiento en

caché.

• Web estática:

inconvenientes

– Ausencia de

personalización.

– Falta de escalabilidad – Difícil mantenimiento (un cambio debe

reproducirse múltiples veces).

– Cada nueva página

requiere un nuevo

fichero HTML.

GENERACIÓN DINÁMICA DE PÁGINAS WEB

• Web dinámica: ventajas – Mayores funcionalidades que habilitan la

interacción del usuario.

– Personalización del

contenido en función de las necesidades del

usuario (múltiples).

– Mayor flexibilidad

– Posibilidad de conectar con un gestor de

contenidos.

• Web dinámica:

inconvenientes

– Creación más compleja (funciones, conectores, bases de datos, etc.) y más cara.

– Menos seguras

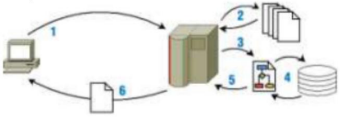
GENERACIÓN DINÁMICA DE PÁGINAS WEB

• Funcionamiento básico de

una página web dinámica:

– 1. El cliente web (navegador)

solicita a un servidor web una

página. 

– 2. El servidor web busca esa

página y la recupera.

– 3. Si se trata de una página

**web dinámica**, el servidor

web contacta con el módulo

responsable de ejecutar el

código y se lo enviará.

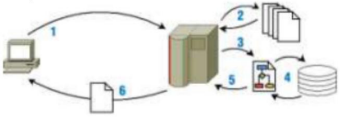
GENERACIÓN DINÁMICA DE PÁGINAS WEB

• Funcionamiento básico de

una página web dinámica:

– 4. Como parte del proceso de

ejecución, puede ser necesario

recuperar parte de la 

información de un repositorio

externo (por ejemplo de una

BD).

– 5. El resultado de la ejecución

será una página en formato

HTML, similar a cualquier otra

página web estática.

– 6. El servidor web envía el

resultado obtenido al

navegador, que lo procesa y lo

muestra en pantalla.

CGI (COMMON GATEWAY INTERFACE)

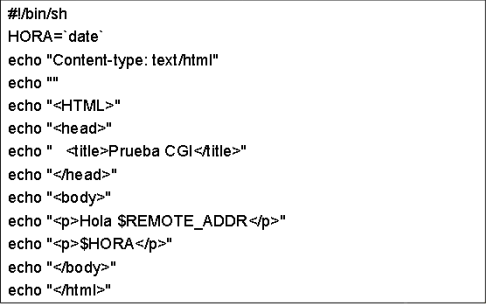
• Creado inicialmente para gestionar formularios. Estándar CGI (RFC3875) define su comportamiento. • No es un lenguaje de programación. Es un mecanismo para que el servidor web pueda llamar a un programa externo.

• El programa CGI tiene acceso a cierta **información sobre la conexión** y a los **valores del formulario**. • El programa CGI puede estar desarrollado en cualquier lenguaje: C/C++, python, perl, sh...

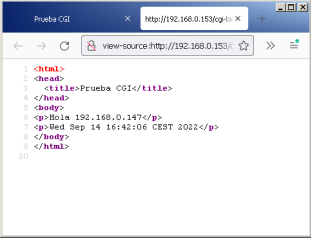
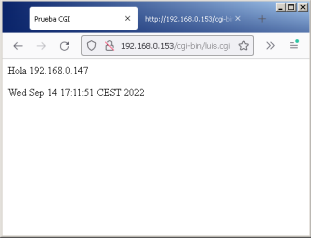
CGI (COMMON GATEWAY INTERFACE)

• Ejemplo de CGI

*http://x.x.x.x/cgi-bin/luis.cgi*

**

• Ejemplo de CGI



SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE

PROGRAMACIÓN

**LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR**

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguaje de programación correspondiente a un entorno servidor:

– “Aquel cuyo código, bien sea como objeto precompilado o bien como código interpretado, es ejecutado por un software específico en el componente que actúa como servidor”.

• Tipos:

– Aplicaciones CGI y derivados.

– Lenguajes de scripting.

– Aplicaciones híbridas de código repartido.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Aplicaciones CGI y derivados:

– Delegan la creación de páginas dinámicas a un programa externo, que recibe ciertos parámetros de entrada y devuelve como resultado el contenido que debe visualizar el cliente.

– El programa externo no depende del código a generar.

– Desventaja: escaso rendimiento a la hora de responder a múltiples peticiones CGI

simultáneamente. Alternativas:

• FastCGI.

• Servlets.

• EJB / JavaBeans.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Aplicaciones CGI y derivados:

– Servlets:

• Programas escritos en Java que corren en un servidor de aplicaciones con JVM.

• Al ejecutarse en el servidor son aplicaciones sin GUI. • Similar a CGI pero con los beneficios de Java.

• Capaces de mantener la conexión abierta y, por tanto, actualizar el contenido del navegador del cliente.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

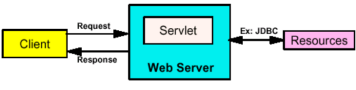
• Aplicaciones CGI y

derivados:

– Servlets:

• El cliente hace una

petición.

• El servidor envía la 

información de la petición

al Servlet.

• El servlet con está

información crea un

contenido dinámico que

entrega al servidor.

• El contenido es devuelto

al cliente.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting:

– PHP (PHP Hypertext Processor).

– ASP (Active Server Pages).

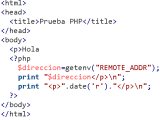
– Perl.

– Python.

– JSP (Java Server Pages).

– Muchos otros: Ruby, LUA…

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting: 

– PHP (PHP Hypertext

Processor):

• Más extendido; gratuito

y abierto. 

• Tipo imperativo. Soporte

OO.

• Código embebido en

HTML.

• El cliente no ve el código

PHP, sino el código 

HTML puro resultante de

la ejecución.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting:



– PHP (PHP Hypertext

Processor):

• Existen módulos para

acceder fácilmente a BD.

• Delimitadores **<?php** y **?>**

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting:

– PHP (PHP Hypertext

Processor): 

• Integración con el

servidor web (módulo

mod\_php).

• Directivas:

– **short\_open\_tags:** Indica si se pueden utilizar en PHP los delimitadores cortos <? y ?>. **Es preferible no usarlos**, pues puede causar problemas si utilizamos páginas con XML. Para restringir la utilización de estos delimitadores con PHP le asignamos a esta directiva el valor Off.

– **max\_execution\_time:** Permite ajustar el número máximo de segundos que podrá durar la ejecución de un script PHP. Evita que el servidor se bloquee si se produce algún error en un script.

– **error\_reporting:** Indica qué tipo de errores se mostrarán en el caso de que se produzcan. Por ejemplo, si haces **error\_reporting = E\_ALL**, te mostrará todos los tipos de errores. Si no quieres que te muestre los avisos pero sí otros tipos de errores, puedes hacer

**error\_reporting = E\_ALL & ~E\_NOTICE**.

– **file\_uploads:** Indica si se pueden o no subir ficheros al servidor por HTTP.

– **upload\_max\_filesize:** En caso de que se puedan subir ficheros por HTTP, puedes indicar el límite máximo permitido para el tamaño de cada archivo. Por ejemplo,

upload\_max\_filesize = 1M.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting: 

– ASP (Active Server Pages):

• Propietario; código

cerrado (parte servidor);

Sustituido por ASP.Net

• ASP permite integrar

componentes ActiveX (del 

lado del servidor), como

acceso a bases de datos,

scripts (VBScript, Jscript)

lo que le dota a la lógica

de negocios de una

potencia y flexibilidad

enorme.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting:

– Perl:

• Antiguo; gratuito y

abierto.

• Empleado para el

desarrollo de

aplicaciones CGI.

LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting: 

– Python:

• Muy famoso

actualmente;

interpretado y OO.

• Gran número de librerías 

disponibles.

• Válido para el desarrollo

de aplicaciones

empresariales.



LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Lenguajes de scripting: 

– JSP (Java Server Pages):

• Funcionamiento similar

a PHP/ASP 🡪 servlet.

• Mejora el rendimiento

en sucesivas

ejecuciones.

• JSP vs Servlet





LENGUAJES DE PROGRAMACIÓN DE ENTORNO SERVIDOR

• Aplicaciones híbridas:

– Tecnología intermedia alternativa a los lenguajes de scripting (interpretados) y a las aplicaciones CGI y derivadas.

– Solución más representativa: plataforma Microsoft .Net – Framework a través de ASP.Net:

• Los formularios web están contenidos en archivos con una extensión ASPX que son los que el cliente solicita a través de una URL al servidor.

• Estos ficheros ASPX contienen código HTML o estático y también etiquetas propias de la plataforma .Net.

• Estas etiquetas definen Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web.

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE

PROGRAMACIÓN

**INTEGRACIÓN CON LOS LENGUAJES DE MARCAS**

INTEGRACIÓN CON LOS LENGUAJES DE MARCAS

• Evolución de la páginas web dinámicas.

– Integración de las etiquetas HTML en el código de los programas. 

– Un enfoque distinto: **integrar el código del programa en medio de las etiquetas HTML de la página web**.



• Esta arquitectura es la que se emplea en lenguajes como PHP, ASP y las páginas JSP de Java EE.

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE

PROGRAMACIÓN

**INTEGRACIÓN CON LOS SERVIORES WEB**

INTEGRACIÓN CON LOS SERVIDORES WEB

• Lenguaje de programación correspondiente a un entorno servidor:

– “Aquel cuyo código, bien sea como objeto precompilado o bien como código interpretado, es ejecutado por un software específico en el

componente que actúa como servidor”.

• Tipos:

– Lenguajes de scripting.

– Aplicaciones CGI y derivados.

– Aplicaciones híbridas de código repartido.

HERRAMIENTAS DE PROGRAMACIÓN

• El proceso de desarrollo no tiene por qué realizarse en el mismo equipo en el que finalmente se despliegue y ejecute la aplicación web que se está desarrollando.

• Instrumentos involucrados en el desarrollo Web: – Navegadores.

– Editores de documentos.

– Entornos de programación.

– Herramientas de tratamiento de imágenes.

– Herramientas para la creación y administración de bases de datos.

• Entornos de programación: programas, aplicaciones o simples utilidades destinadas a la programación web.