



CGI

Programación en Internet
Curso 2004-2005

Programación en Internet – Curso 2004-2005

Índice

- Introducción
- Características
 - ¿Qué necesito?
 - Lenguajes
 - Razones a favor y en contra
- El primer CGI
- Comunicación con el cliente, métodos
- Variables de entorno: servidor, cliente, petición HTTP

Introducción

- *Common Gateway Interface*
- Estándar
- ¿Por qué es necesario?
- Aumento complejidad sitios web:
 - Conocimientos de programación
 - Conocimientos de administración

3

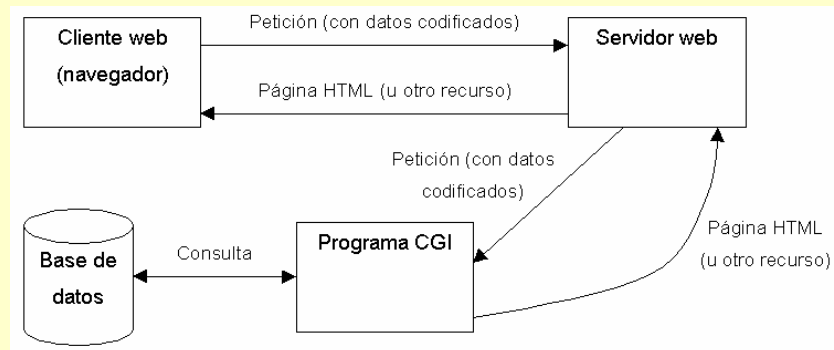
Introducción

- ¿Qué se puede hacer? Prácticamente todo

Cualquier cosa que haga un CGI, lo tiene que hacer rápidamente y empleando la menor cantidad posible de recursos

4

Cómo funciona



- Procesamiento
 - Directo, el propio CGI lo hace todo
 - Indirecto, el CGI hace de puente de otro programa diseñado inicialmente para no ejecutarse en la Web

5

Aplicaciones

- Gestión de libros de visitas (*guestbook*)
- Gestión de anuncios (*banners*)
- Gestión de contadores (*hit counters*)
- Imágenes sensibles procesadas en el servidor
- Acceso a bases de datos
- ...

6

Qué necesito para programar

- Editor ASCII estándar
- Compilador o intérprete
- Servidor web que acepte CGI
- Navegador

7

Lenguaje de programación

- Requisitos:
 - Leer datos de la entrada estándar
 - Acceder a las variables de entorno
 - Escribir en la salida estándar
- Compilado ↔ Interpretado
- Programas CGI → *Scripts* (porque los primeros se programaban como *shell scripts*)
- Lenguajes más comunes: C y Perl

8

Independencia

- De la plataforma:
 - Hardware
 - Software (sistema operativo)
- Del servidor web:
 - No asumir directorio (rutas)
 - No asumir IP
 - No asumir permisos
 - Solución: comprobar antes → Usar llamadas al sistema o ficheros de configuración

9

Razones para emplear CGI

- Es el método más rápido cuando se ejecuta mucho código
- Es un estándar, compatible con 99% plataformas, SOs y servidores web
- Es compatible con todos los clientes
- Se puede emplear prácticamente cualquier lenguaje
- Tecnología antigua: probada y estable

10

Razones para no emplear CGI

- Tecnología obsoleta
- No mantiene el estado (sesión)
- Integración débil CGI y servidor
- Instancia nueva de un programa en memoria → Ocupa muchos recursos

11

El primer CGI

- Salida estándar (stdout)
- Se puede generar cualquier tipo de documento (HTML, imagen, PDF, ...)
- Documento devuelto:
 - Cabecera → Tipos MIME
 - Cuerpo

12

Programación en Internet – Curso 2004-2005

El primer CGI

The diagram illustrates the structure of an HTTP response. It is divided into three sections:

- Cabecera:** A light blue box containing the text `Content-type: text/html`.
- Línea separación:** A light green box representing a blank line that separates the header from the body.
- Cuerpo:** A yellow box containing the HTML content:

```
<HTML>
<BODY>
¡Hola mundo!
</BODY>
</HTML>
```

13

Programación en Internet – Curso 2004-2005

El primer CGI

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Content-type: text/html\n");
    printf("\n");
    printf("<HTML>\n<BODY>\n");
    printf("¡Hola mundo!\n");
    printf("</BODY>\n</HTML>\n");
    return 0;
}
```

14

El primer CGI

Tipo	Extensión
image/gif	gif
image/jpeg	jpeg jpe jpg
text/html	html htm
text/plain	txt
video/mpeg	mpeg mpg mpe

15

El primer CGI

Para saltar a otra dirección:

Location: <http://www.ua.es>

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    /* Procesamiento del CGI */
    printf("Location: http://www.ua.es\n");
    printf("\n");
    return 0;
}
```

16

Cómo comunicarse directamente con el cliente

- Servidor web añade a la respuesta del CGI más datos, para formar mensaje HTTP correcto
- Si el servidor web no hace de intermediario, se ha de renombrar el CGI con el prefijo “nph-” (*no parse headers*) → El servidor web no procesa la respuesta del CGI
 - En este caso hemos de escribir el paquete HTTP completo
- Ejemplo:

```
HTTP/1.0 200 OK
Server: IIS/4.0
Content-type: text/html

<HTML><BODY>
Esto es un mensaje HTTP correcto
</BODY></HTML>
```

17

Cómo envía el servidor web información a un CGI

- A través de la línea de comandos (*command line*)
- A través de la URL (QUERY_STRING)
- A través de la entrada estándar (stdin)
- A través de la información de ruta directorios (PATH_INFO)

18

A través de la línea de comandos (I)

- Envío:
 - Búsquedas con <ISINDEX>
 - En la URL: cgi?termino1+termino2+... (no puede aparecer =)
- Recepción:
 - Cada palabra es un parámetro que se pasa al CGI (*argv*)
 - En QUERY_STRING

19

A través de la línea de comandos (y II)

- Ejemplo: listado de provincias y al pulsar sobre una de ellas se muestra una página nueva con las ciudades de la provincia

Seleccione una provincia:

```
<A HREF="ciudades.exe?alicante">Alicante</A>
```

```
<A HREF="ciudades.exe?castellon">Castellón</A>
```

```
<A HREF="ciudades.exe?valencia">Valencia</A>
```

20

Cómo tratar los formularios (I)

- Navegador envía los datos como:

```
control1=valor1&control2=valor2&...  
    &  
    controln=valorn
```

- Ejemplo:

```
nombre=Jose&universidad=UA&carrera=  
    Derecho
```

- Si un campo está vacío (excepto SELECT): `c1=&c2=`

21

Cómo tratar los formularios (II)

- Los datos se codifican: `%xx` y `+` (caracteres escapados para evitar conflictos con HTTP y el servidor)

- Ejemplo:

```
&%$ñ → %26%25%24%F1
```

- Tareas CGI:

- Separar entrada en parejas `control=valor`
- Separar `control = valor`
- Decodificar valor

22

Publicaciones - Modificar publicación

Hay que rellenar todos los campos marcados en negrita:

Tipo:

Ref (max. 20):

Año:

Referencia:

Más información:

Resumen (abstract):

Autores:

Subir Bajar

Enviar Borrar

Programación en Internet – Curso 2004-2005

Cómo tratar los formularios (y III)

```
tipo=5&ref=JISBD&anyo=2003&referencia=Un+
art%EDculo+de+prueba...&mas=No+hay+m%E1
s+informaci%F3n&resumen=El+resumen+se+p
ondr%E1+despu%E9s
```

1. &losautores=21+22+3+1+23

losautores es un con hidden y mediante JS se guardan los autores seleccionados en este control

2. &autores=21&autores=22&autores=3

&autores=1&autores=23

Otro método es antes del envío del formulario seleccionar mediante JS las opciones del desplegable (así se provoca que los datos se envíen)

A través de la URL (I)

- Envío:
 - Formulario con GET
 - En la URL: `cgi?c1=a&c2=b`
- Recepción:
 - En `QUERY_STRING`

25

A través de la URL (y II)

- Ejemplo: listado de productos y al seleccionar uno de ellos se muestra una página nueva con las características del producto seleccionado

Seleccione un producto:

```
<A HREF="ficha.exe?fam=3&prod=1">D.D. 40 GB</A>
```

```
<A HREF="ficha.exe?fam=3&prod=2">D.D. 80 GB</A>
```

```
<A HREF="ficha.exe?fam=6&prod=7">TFT 15''</A>
```

26

A través de la entrada estándar

- Envío:
 - Formulario con POST
- Recepción:
 - En la entrada estándar (stdin)
- Variables:
 - CONTENT_LENGTH
 - CONTENT_TYPE

27

A través de la información de ruta

- Envío:
 - En la URL: cgi/fich/prueba.txt
- Recepción:
 - En PATH_INFO

28

Variables de entorno CGI

- Específicas del servidor
- Específicas del cliente
- Específicas de la petición

29

Específicas del servidor

- Características sobre el servidor web:
 - GATEWAY_INTERFACE: CGI/1.1
 - SERVER_NAME: www.ua.es
 - SERVER_PORT: 80
 - SERVER_PROTOCOL: HTTP/1.1
 - SERVER_SOFTWARE: Microsoft-IIS/4.0

30

Específicas del cliente

- Información sobre el cliente (navegador):
 - HTTP_ACCEPT: image/gif, image/jpeg
 - HTTP_ACCEPT_ENCODING: gzip, deflate
 - HTTP_ACCEPT_LANGUAGE: es-ES,en,pdf
 - HTTP_REFERER: <http://www.ua.es/index.html>
 - HTTP_USER_AGENT: Mozilla/4.7 [en] (Win98; I)

31

Específicas de la petición

- Información sobre la petición recibida:
 - CONTENT_LENGTH
 - CONTENT_TYPE
 - PATH_INFO
 - QUERY_STRING
 - REMOTE_ADDR
 - REMOTE_HOST
 - REQUEST_METHOD
 - SCRIPT_NAME

32

Cómo acceder desde C

```
#include <stdlib.h>
```

```
char *variable;
```

```
variable = getenv("SERVER_NAME");
```

33

Seguridad

- Permisos de ejecución (/cgibin, /cgi-bin o /scripts)
- Examina el código → Caballos de Troya o puertas traseras
- Versiones estable de los programas empleados

34

Seguridad

- Las presunciones son peligrosas
 - Suponer que los datos provienen de nuestro formulario
 - Datos incorrectos
 - Exceso de datos (*buffer overrun*)
 - Inconveniente: código aumenta, más difícil de mantener
- Programa defensivamente

35

Seguridad

- Limpia los datos antes de usarlos
- Limpia los datos antes de pasarlos a otro programa
- Cuidado con HTML enviado desde el cliente
- Nivel de privilegio → Usuario específico con pocos privilegios
- Nivel de prioridad → Media o baja
- Usa un ordenador específico para ejecutar los CGI

36

Seguridad

- Consulta listas de correos y grupos de noticias
 - Actualiza conocimientos y avances de la tecnología
- Nunca olvides el código fuente en un directorio público → No se puede evitar con lenguajes interpretados