Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

CGI

Programación en Internet
Curso 2007-2008

Índice

2

Introducción

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

3

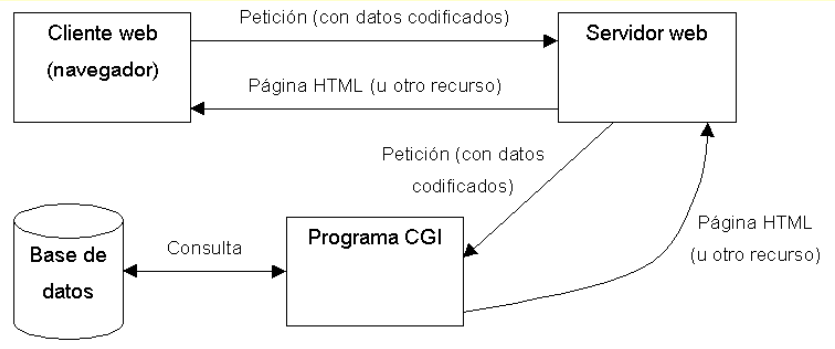
Introducción

- ¿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

Alternativas

- FastCGI:
  - Resuelve problemas de escalabilidad con muchos usuarios
  - Permite emplear un único proceso persistente que responde a múltiples peticiones
- SCGI (Simple CGI):
  - Protocolo más sencillo de implementar que CGI

7

Qué necesito para programar

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

8

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

9

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

10

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

11

Razones para no emplear CGI

- Tecnología obsoleta
  - No está orientado a “pagína web”, sino a programa
- No mantiene el estado (sesión)
- Integración débil CGI y servidor
- Instancia nueva de un programa en memoria → Ocupa muchos recursos

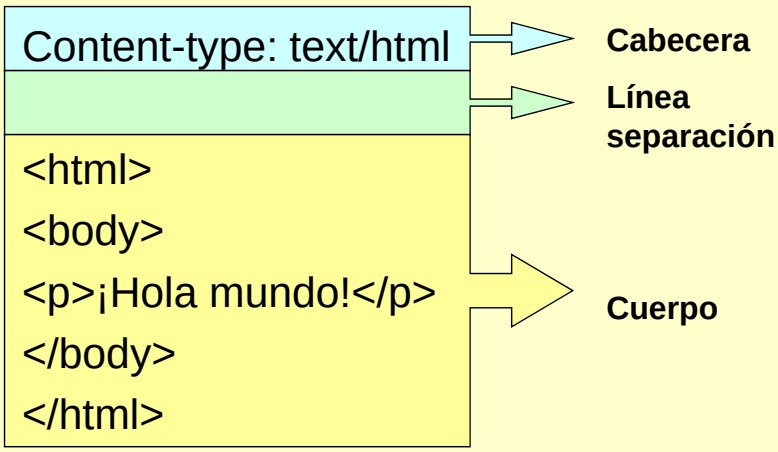
12

El primer CGI

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

13

El primer CGI



14

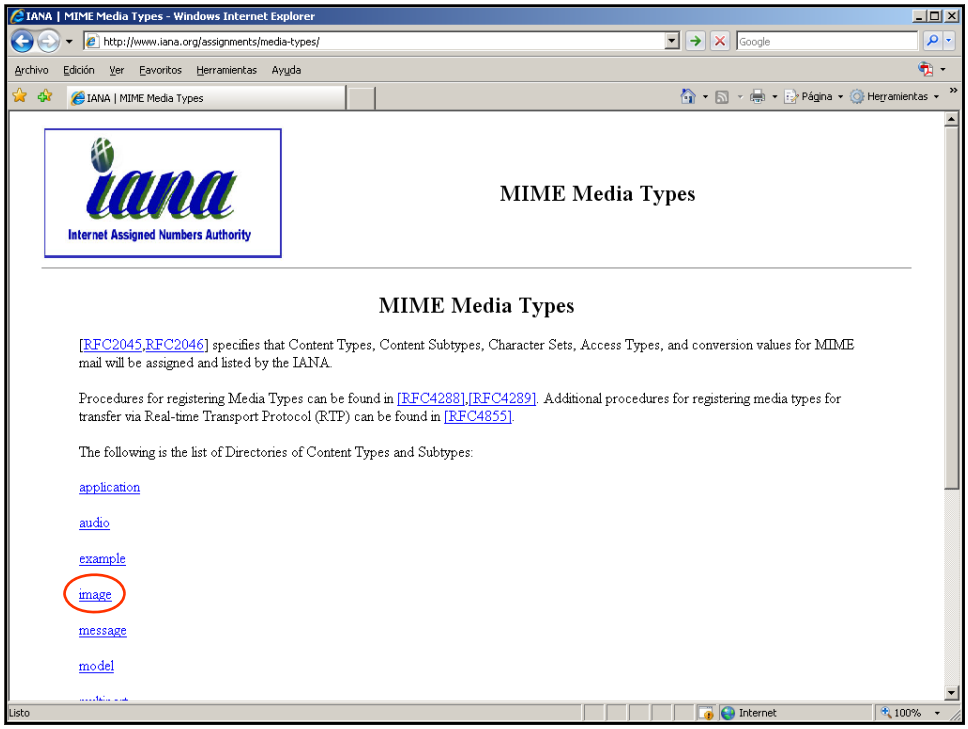
Programación en Internet – Curso 2007-2008

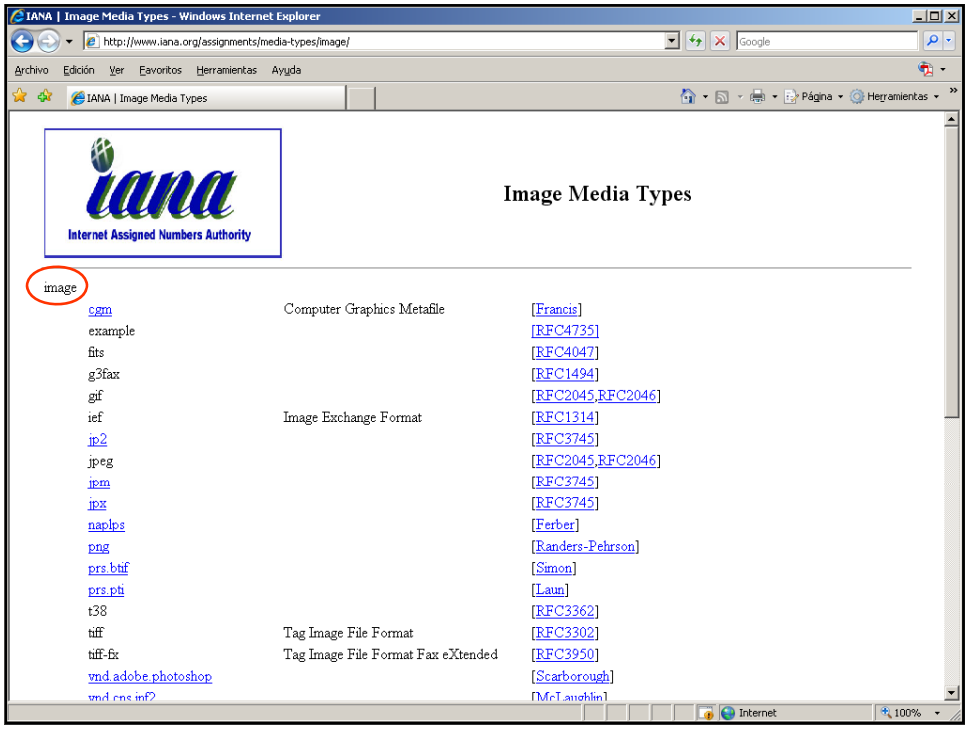
El primer CGI

```
#include <stdio.h>

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

15





Programación en Internet – Curso 2007-2008

El primer CGI

Tipo	Extensión
application/msword	doc
application/pdf	pdf
application/rtf	rtf
image/gif	gif
image/jpeg	jpeg jpe jpg
image/png	png
text/html	html htm
text/plain	txt
text/xml	xml
video/mpeg	mpeg mpg mpe

18

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;
}
```

19

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>
<p>Esto es un mensaje HTTP correcto</p>
</body></html>
```

20

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)

21

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

22

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>
```

23

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=`

24

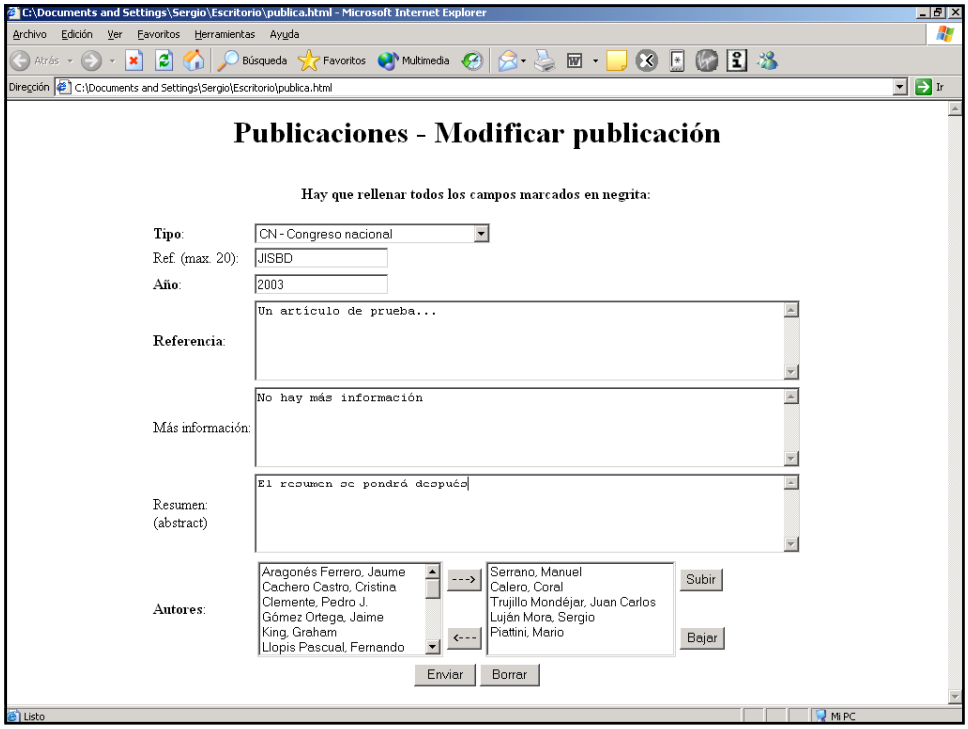
Programación en Internet – Curso 2007-2008

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
  - Descodificar valor

25



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 control `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)

27

A través de la URL (I)

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

28

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>
```

29

A través de la entrada estándar

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

30

A través de la información de ruta

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

¿GET o POST?

- Existe un límite en la cantidad de información que se puede enviar mediante GET
- GET no es seguro para claves o números de tarjeta de crédito:
  - Los datos se envían en la URL
  - La URL se almacena en el log del servidor
  - La URL se almacena en el historial de navegación del navegador

Variables de entorno CGI

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

33

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

34

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)

35

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

36

Cómo acceder desde C

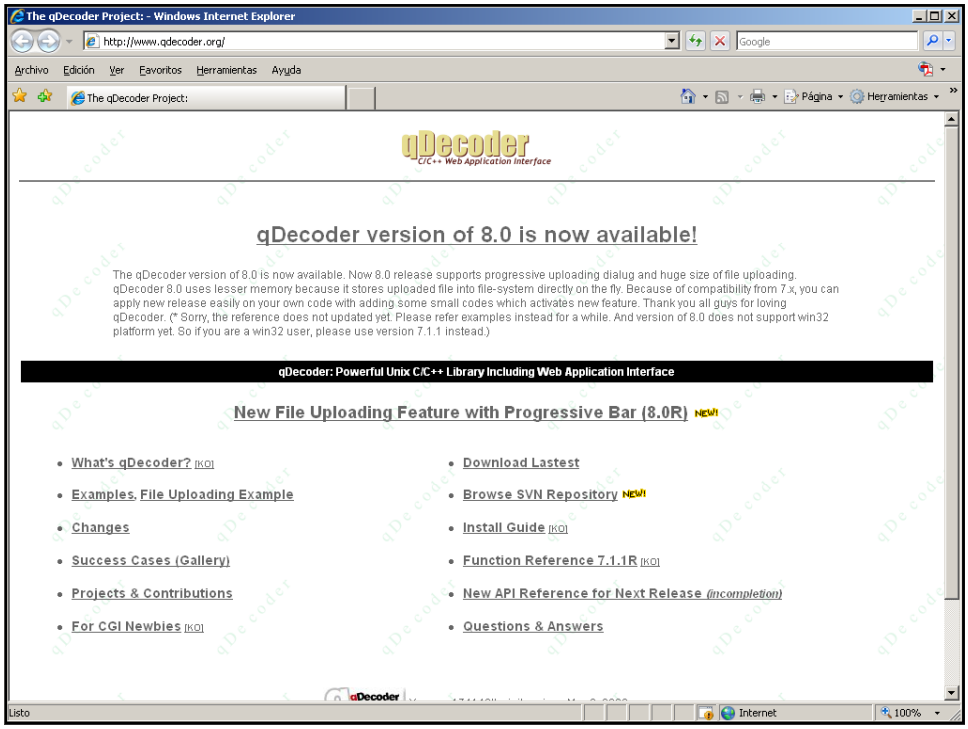
```
#include <stdlib.h>


char *variable;

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


Cómo acceder desde C

- Empleo de alguna librería
- qDecoder C/C++ Web Application Interface:
  - Acceso a GET/POST/COOKIE
  - Subida de ficheros
  - Manejo de sesiones
  - Multiplataforma (sistema operativo, servidor web y compilador)



Programación en Internet – Curso 2007-2008

Cómo acceder desde Perl

- cgi-lib.pl y CGI.pm son las librerías estándar *de facto* para crear CGIs en Perl

40

Cómo acceder desde Perl

- Ejemplo básico:

```
#!/usr/bin/perl -wT
print "Content-type: text/html\n\n";
print "<html><head><title>Prueba de
  Perl</title></head>\n";
print "<body>\n";
print "<p>¡Hola mundo!</p>\n";
print "</body></html>\n";
```

41

Cómo acceder desde Perl

- Formulario:

```
<html>
<head><title>Prueba de Perl</title></head>
<body>
<form action="get.cgi" method="GET">
Nombre: <input type="text" name="nombre" size=30>
<br>
Apellidos: <input type="text" name="apellido" size=30>
<br>
<input type="submit">
</form>
</body></html>
```

42

Cómo acceder desde Perl

- Script:

```
#!/usr/bin/perl -wT
use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);
use strict;

print header;
print start_html("Prueba de Perl");
my %form;
foreach my $p (param()) {
    $form{$p} = param($p);
    print "$p = $form{$p}<br>\n";
}
print end_html;
```


Seguridad

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

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

45

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

46

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

47