CGI

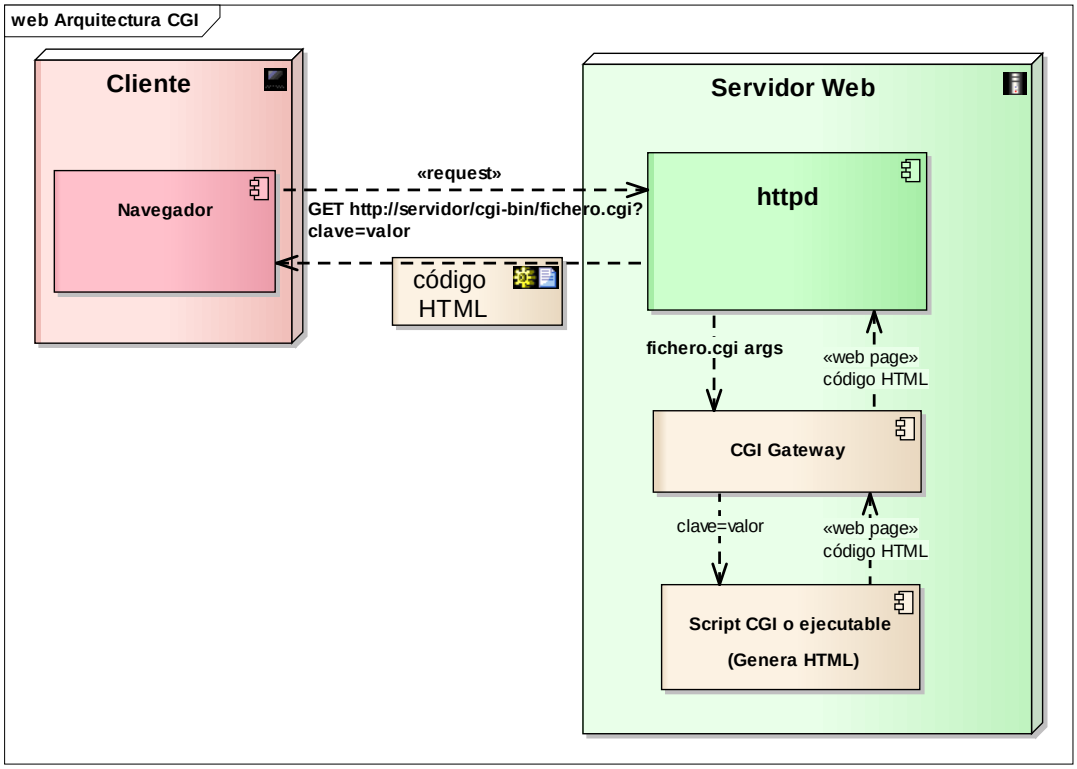
Aplicaciones Web / Sistemas Web

Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

CGI

- **CGI** (*Common Gateway Interface*) es un modo sencillo de crear contenido dinámico en un sitio Web
  - CGI define la forma en que un servidor Web puede interactuar con programas externos que generen contenido (páginas HTML)
    - Estándar CGI 1.1 especificado por IETF en RFC 3875
    - CGI define los parámetros (variables de entorno) que describen la petición del cliente
  - El script se invoca como un proceso del sistema
    - Se ejecuta como un hijo del proceso servidor Web
  - Programas o scripts CGI (simplemente *CGIs*): un script escrito en bash, MSDOS, Perl, php, Ruby, etc. o un ejecutable
    - Generan el contenido web que se enviará al cliente
    - El contenido se identifica con un tipo MIME

CGI



CGI en Apache

- Configuración de Apache para permitir la ejecución de CGIs
  - Tiene que estar cargado el módulo de CGI y el de alias:
    - *LoadModule cgi_module modules/mod_cgi.so*
    - *LoadModule cgi_module modules/mod_alias.so*
  - Tiene que estar definido el directorio donde estarán los CGIs:
    - *ScriptAlias /cgi-bin/ "C:/xampp/cgi-bin/"* (windows)
    - *ScriptAlias /cgi-bin/ "/opt/lampp/cgi-bin/"* (linux)
  - Para mejor controlar la seguridad, NO se recomienda ejecutar scripts fuera del directorio *cgi-bin*

- Para depurar, se puede ver lo que ha ocurrido en los logs
  - Están descritos en *http://httpd.apache.org/docs/2.4/logs.html*
  - Los principales:
    - error.log – información de diagnóstico
    - access.log – registra todas las peticiones procesadas por el servidor

Mi primer CGI (Linux)

- Crear un script básico (`script.sh`) en la carpeta `/opt/lampp/cgi-bin/`
- Probarlo: `http://localhost/cgi-bin/script.sh`
  - La salida normalmente será el conjunto de variables de entorno del script (que se le han pasado por la interfaz CGI)
  - NOTA: Si el fichero no tiene permisos de ejecución apache dará un internal server error

```
#!/bin/sh

echo "Content-Type: text/html"
echo

echo "<pre>"
env
echo "</pre>"
```


Mi primer CGI (Windows)

- Crear un script básico (`script.bat`) en la carpeta `xampp/cgi-bin`
- Probarlo: `http://localhost/cgi-bin/script.bat`
  - La salida normalmente será el conjunto de variables de entorno del script (que se le han pasado por la interfaz CGI)

```
@echo off

echo Content-type: text/html
echo.

echo ^<pre^>
set
echo ^</pre^>
```

Mi primer CGI (Perl)

- Perl suele ser el lenguaje más habitual para scripts CGI
- Hay ejemplos en la instalación de XAMPP
  - Para ver el entorno se puede probar el siguiente: `http://localhost/cgi-bin/printenv.pl`
  - La salida normalmente será el conjunto de variables de entorno del script (que se le han pasado por la interfaz CGI)

```
#!"C:\xampp\perl\bin\perl.exe"
##
##  printenv -- demo CGI program which just prints its environment
##

print "Content-type: text/plain; charset=iso-8859-1\n\n";
foreach $var (sort(keys(%ENV))) {
   $val = $ENV{$var};
   $val =~ s|\n|\\n|g;
   $val =~ s|"|\\"|g;
   print "${var}=\"${val}\"\n";
}
```


Mi primer CGI (C)

- Con un lenguaje de programación, como C, también se puede crear un programa cuyo ejecutable genere la página HTML

```
// MiPrimerCGI.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void main(void)
{
    printf("Content-type: text/html\n\n");
    printf("<HTML><HEAD><TITLE>Nombre</TITLE></HEAD>");
    printf("<BODY BGCOLOR=\"#FFFFFF\"><P ALIGN=CENTER>");

    printf("<BR>SERVER_NAME = %s", getenv("SERVER_NAME"));
    printf("<BR>SERVER_SOFTWARE = %s", getenv("SERVER_SOFTWARE"));
    printf("<BR>REQUEST_METHOD = %s", getenv("REQUEST_METHOD"));
    printf("<BR>HTTP_REFERED = %s", getenv("HTTP_REFERED"));
    printf("<BR>SCRIPT_NAME = %s", getenv("SCRIPT_NAME"));
    printf("<BR>QUERY_STRING = %s", getenv("QUERY_STRING"));
    printf("<BR>REMOTE_HOST = %s", getenv("REMOTE_HOST"));

    printf("</P></BODY></HTML>");

}
```

Variables de entorno de un CGI

- Específicas del servidor:
  - SERVER_SOFTWARE — nombre/versión del servidor HTTP
  - SERVER_NAME — nombre (o IP) del host del servidor
  - GATEWAY_INTERFACE — Versión de CGI
- Específicas de la petición:
  - SERVER_PROTOCOL — Versión de HTTP
  - SERVER_PORT — Puerto TCP usado
  - REQUEST_METHOD — método HTTP invocado
  - SCRIPT_NAME — programa invocado: /cgi-bin/script.cgi
  - QUERY_STRING — parte del URL tras el carácter ?
    - pares nombre=valor separados por & (var1=val1&var2=val2…)
  - REMOTE_HOST — nombre del host del cliente
  - REMOTE_ADDR — dirección IP del cliente
  - CONTENT_TYPE — tipo de los datos de entrada
  - CONTENT_LENGTH — tamaño de los datos de entrada
  - Variables pasadas por el agente de usuario (HTTP_ACCEPT, HTTP_ACCEPT_LANGUAGE, HTTP_USER_AGENT, HTTP_COOKIE, y otras)

Variables de entorno de un CGI

```
HTTP_HOST=localhost
HTTP_USER_AGENT=Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:11.0) Gecko/20120313
    Firefox/11.0
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE=en-us,en;q=0.5
HTTP_ACCEPT_ENCODING=gzip, deflate
HTTP_CONNECTION=keep-alive

SERVER_SIGNATURE=<address>Apache/2.4.3 (Win32) OpenSSL/1.0.1c PHP/5.4.7 Server at
    localhost Port 80</address>

SERVER_SOFTWARE=Apache/2.4.3 (Win32) OpenSSL/1.0.1c PHP/5.4.7
SERVER_NAME=localhost
SERVER_ADDR=::1
SERVER_PORT=80
REMOTE_ADDR=::1
DOCUMENT_ROOT=C:/xampp/htdocs
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=C:/xampp/cgi-bin/
SERVER_ADMIN=postmaster@localhost
SCRIPT_FILENAME=C:/xampp/cgi-bin/entorno.bat
REMOTE_PORT=50200
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/entorno.bat
SCRIPT_NAME=/cgi-bin/entorno.bat
```

Salida de CGI

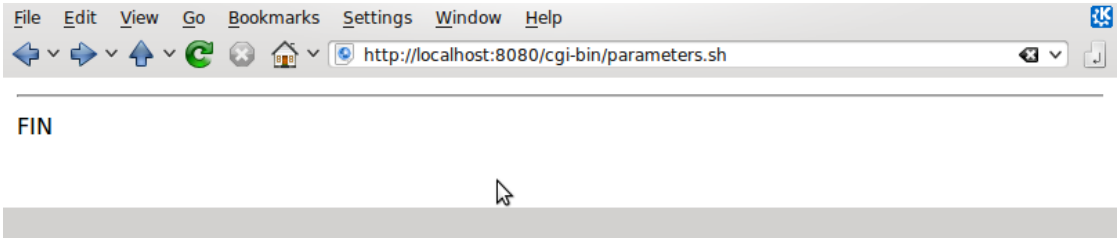
- El resultado del CGI es la salida estándar del programa, que es normalmente lo que el servidor envía al cliente
  - Puede ser variada
    - Documento HTML, texto normal, clip de audio, etc.
  - Empieza con una cabecera y una línea en blanco
    - La cabecera similar a una cabecera HTML, debe especificar el tipo de contenido (*MIME type*)
      - Los tipos **MIME** se especifican como tipo/subtipo:
        - text/html
        - text/plain
        - application/pdf
        - application/xml
        - audio/mpeg
        - image/jpeg
        - video/mpeg
        - multipart/form-data
  - El contenido va a continuación
    - Es posible referenciar a un fichero donde estaría el contenido
      - **Location: URL**

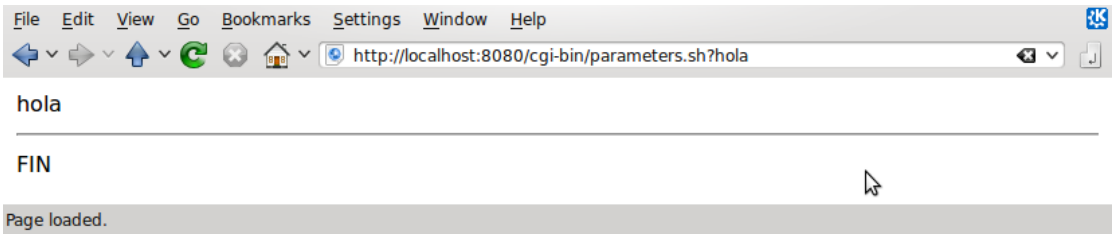
Generación de un fichero word con un CGI

```
@echo off

echo Content-Type: application/vnd.ms-word
echo Content-Disposition: attachment; filename=fichero.doc
echo.

REM Contenido del fichero word
echo Hola Mundo.
```

Uso de parámetros en CGI

```
#!/bin/sh
echo  "Content-Type: text/html"
echo

echo "$QUERY_STRING"
echo "<hr>"
echo "FIN"
```

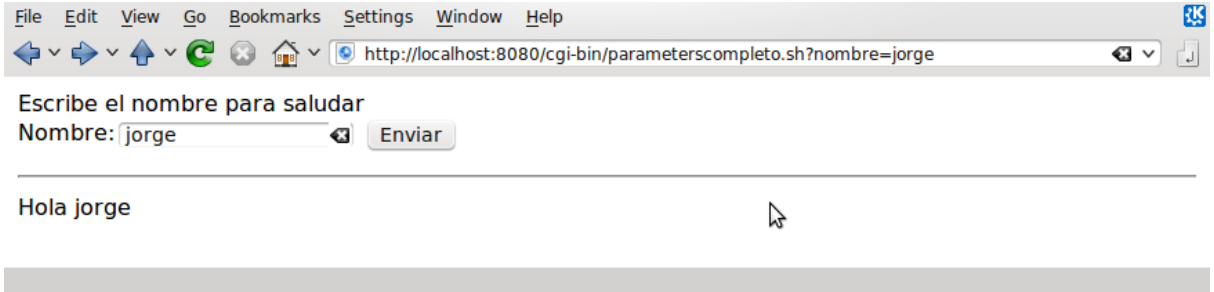




Asociación de CGIs a formularios

parametros.sh

```
#!/bin/sh
echo  "Content-Type: text/html"
echo
NOMBRE=`echo "$QUERY_STRING"| tr "&" "\n"| grep "nombre="|
                                    cut -f 2 -d "=" |head -n1`

echo "Escribe el nombre para saludar"
echo '<form method="get" action="/cgi-bin/parametros.sh ">'
echo "Nombre:<input type=\"text\" name=\"nombre\" value=\"\">"
echo '<input type="submit" value="Enviar"/>'
echo '</form>'
echo "<hr>"
echo "Hola $NOMBRE"
```



Ejercicios CGI

- Generar una página con un CGI que indique toda la información posible que se pueda conseguir sobre el cliente a partir de las variables del entorno
  - Impleméntalo con la shell de tu sistema operativo
  - Crea un programa C y configura el servidor Apache para que pueda invocar el ejecutable (.exe)

    • Consultar el Tutorial de Apache: Dynamic Content with CGI. http://httpd.apache.org/docs/2.4/howto/cgi.html

Valoración de CGI

- Ventajas
  - Independiente del lenguaje
  - Interfaz sencilla: no hace falta utilizar una librería o API específicos
    - Basta con utilizar entrada/salida estándar y variables de entorno
- Desventajas
  - Por cada petición HTTP se crea un nuevo proceso en el servidor
    - Ineficiente
      - Recursos requeridos para crear un nuevo proceso
      - Además un nuevo proceso implica establecer una nueva conexión a la base de datos
    - Se han considerado alternativas como FastCGI
  - Sobrecarga de la interpretación del script
    - El código interpretado normalmente consume más tiempo que el compilado
    - Pero se puede invocar ejecutables directamente
  - Las aplicaciones CGI son dependientes de la plataforma

Alternativas a CGI para contenido dinámico

- Uso de plantillas (templates)
  - Páginas SSI (Server-side include), Adobe ColdFusion
- Aplicaciones J2EE
  - Servlets Java
- Código de scripts embebido en las páginas
  - Hypertext Preprocessor (PHP)
  - Java Server Pages (JSP)
  - Active Server Pages (ASP)

- Todos estos métodos plantean una cuestión: ¿Quién es el dueño (y puede modificar) una página? ¿Los desarrolladores o los diseñadores?
  - Algunos frameworks, como Struts ,pretenden tratar este tema separando contenido de presentación

Alternativas a CGI: SSI

- **Server-side includes (SSI)**
  - Técnica más antigua de página activa o plantilla
  - Normalmente las páginas con SSI tienen la extensión **.shtml**
  - Incluye instrucciones como comentarios HTML:
    <!--#comando arg1="valor1" arg2="valor2"...-->
  - Estos comandos son ejecutados por el servidor al cargar la página
    - Antes incluso de ejecutar el código de scripts embebidos en el HTML
- Usos:
  - Mostrar información del sistema:
    - `La hora: <!--#echo var="DATE_LOCAL" -->`
    - `El último cambio de la página principal: <!--#flastmod virtual="/index.html"-->`
    - `Su dirección IP: <!--#echo var="REMOTE_ADDR" -->`
  - Incluir ficheros en el HTML:
    ```
    <!--#include file="externo.html"-->
    ```
  - Ejecutar CGIs:
    ```
    <!--#exec cgi="/cgi-local/unScript.pl"-->
    ```
- Las capacidades de SSI son limitadas y su uso está en declive

Alternativas a CGI: Java Servlets

- **Java Servlets**
- El mismo código se ejecuta igual en todas partes
  - Solo requiere que el servidor tenga una máquina virtual Java
- El código se ejecuta en el propio servidor web en vez de como un proceso aparte
- Los servlets pueden acceder a todas las características de Java (seguridad, conectividad con bases de datos, etc.)
- Permite estructurar mejor las aplicaciones
  - Arquitectura software, más modular
  - Aplicaciones más complejas

Alternativas a CGI: Scripts embebidos en el HTML

- **JSP**
  - Código Java embebido en la página HTML
    - <% *código Java* %>
    - Un preprocesador convierte las JSPs en servlets
      - Requiere un servidor que soporte servlets
- **ASP**
  - Propuesta de Microsoft, funciona en el Internet Information Server (IIS)
  - Código VBScript o JavaScript embebido en la página HTML
    - <% *código VBScript o JavaScript* %>
- **PHP**
  - Libre, mantenido por The PHP Group
  - Código embebido en la página HTML
    - <?php *código PHP* ?>

Bibliografía

- http://www.cgi101.com/book/
- http://www.ffnn.nl/pages/articles/linux/cgi-scripting-tips-for-bash-or-sh.php
- http://www.yolinux.com/TUTORIALS/BashShellCgi.html

Colaborar para completar la bibliografía en el campus virtual