

# PROGRAMACIÓN CGI

La Common Gateway Interface, o CGI, es un conjunto de estándares que definen cómo se intercambia la información entre el servidor web y un script personalizado. Las especificaciones CGI las mantiene actualmente la NCSA.

## ¿Qué es CGI?

- La Interfaz de puerta de enlace común, o CGI, es un estándar para que los programas de puerta de enlace externos interactúen con servidores de información como los servidores HTTP.
- La versión actual es CGI / 1.1 y CGI / 1.2 está en proceso.

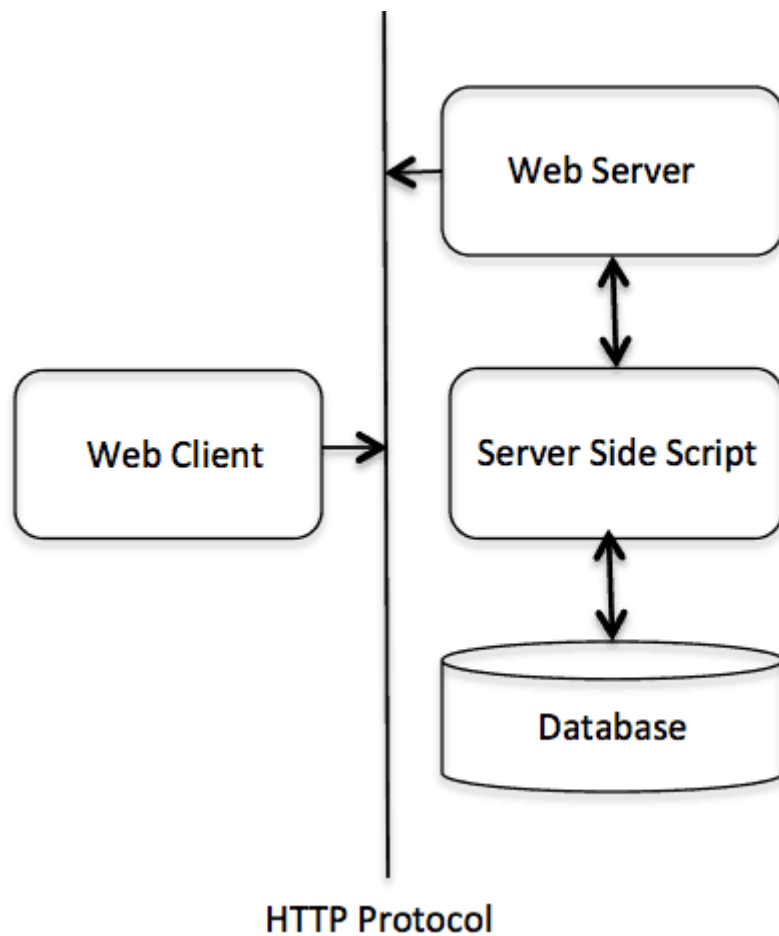
## Buscando en la web

Para comprender el concepto de CGI, veamos qué sucede cuando hacemos clic en un hipervínculo para navegar por una página web o URL en particular.

- Su navegador se pone en contacto con el servidor web HTTP y solicita la URL, es decir, el nombre del archivo.
- Web Server analiza la URL y busca el nombre del archivo. Si encuentra ese archivo, lo devuelve al navegador; de lo contrario, envía un mensaje de error que indica que solicitó un archivo incorrecto.
- El navegador web recibe la respuesta del servidor web y muestra el archivo recibido o el mensaje de error.

Sin embargo, es posible configurar el servidor HTTP para que cada vez que se solicite un archivo en un directorio determinado, ese archivo no se devuelva; en su lugar, se ejecuta como un programa, y todo lo que genera ese programa se envía de vuelta para que lo muestre su navegador. Esta función se llama Common Gateway Interface o CGI y los programas se denominan scripts CGI. Estos programas CGI pueden ser un script de Python, un script PERL, un script de shell, un programa C o C ++, etc.

## Diagrama de arquitectura CGI



## Forma de actuación de CGI

A continuación se describe la forma de actuación de un CGI de forma esquemática:

1. En primera instancia, el servidor recibe una petición (el cliente ha activado un URL que contiene el CGI), y comprueba si se trata de una invocación de un CGI.
2. Posteriormente, el servidor prepara el entorno para ejecutar la aplicación. Esta información procede mayoritariamente del cliente.
3. Seguidamente, el servidor ejecuta la aplicación, capturando su salida estándar.
4. A continuación, la aplicación realiza su función: como consecuencia de su actividad se va generando un objeto MIME que la aplicación escribe en su salida estándar.

5. Finalmente, cuando la aplicación finaliza, el servidor envía la información producida, junto con información propia, al cliente, que se encontraba en estado de espera. Es responsabilidad de la aplicación anunciar el tipo de objeto MIME que se genera (campo `CONTENT_TYPE`).

## Programación de un CGI

Un programa CGI puede ser escrito en cualquier lenguaje de programación que produzca un archivo ejecutable. Entre los lenguajes más habituales se encuentran: C, C++, Perl, Java, Python... No obstante, debido a que el CGI recibe los parámetros en forma de texto será útil un lenguaje que permita realizar manipulaciones de las cadenas de caracteres de una forma sencilla, como por ejemplo Python. Python es un lenguaje interpretado que permite manipulaciones sencillas de ficheros y textos, así como la extracción y manipulación de cadenas de caracteres, unidas a unas búsquedas rápidas y fáciles.

## Intercambio de información: Variables de entorno

Los siguientes datos son las variables de entorno que pasan por los programas CGI:

Variables específicas del servidor:

- `SERVER_SOFTWARE` — nombre y versión del servidor web.
- `SERVER_NAME` — nombre de equipo del servidor, puede ser una dirección IP.
- `GATEWAY_INTERFACE` — versión CGI.

Variables específicas de la petición:

- `SERVER_PROTOCOL` — versión HTTP.
- `SERVER_PORT` — Puerto TCP.
- `REQUEST_METHOD` — nombre del método HTTP.
- `PATH_INFO` — sufijo de la ruta.
- `PATH_TRANSLATED` — corresponde a la ruta completa del servidor, si `PATH_INFO` está presente.
- `SCRIPT_NAME` — ruta relativa al programa. Ejemplo: `/cgi-bin/script.cgi`.
- `QUERY_STRING` — la parte del URL después del carácter `?`. la cadena de consulta puede estar compuesta de `*nombre=valores` separados por el carácter `&` (ejemplo: `var1=val1&var2=val2...`) que se utiliza para enviar datos de un formulario web usando el método GET.
- `REMOTE_HOST` — nombre del host del cliente.
- `REMOTE_ADDR` — dirección IP del cliente.
- `AUTH_TYPE` — tipo de identificación.
- `REMOTE_USER` variable usada para ciertos tipos de autenticación (`AUTH_TYPE`).

- REMOTE\_IDENT — variable para identificar algunas conexiones TCP.
- CONTENT\_TYPE — tipo de contenido de la aplicación.
- CONTENT\_LENGTH — tamaño de los datos de entrada.
- Las variables que pasan por el agente de usuario del cliente (HTTP\_ACCEPT, HTTP\_ACCEPT\_LANGUAGE, HTTP\_USER\_AGENT, HTTP\_COOKIE etc.) contienen valores de sus correspondientes cabeceras HTTP por lo que sus datos no cambian.

## Tipos habituales de CGI

**Contador de accesos:** Cuenta el número de veces que se ha solicitado una página determinada. Se guarda el valor en un fichero. Cada vez que se invoca se incrementa, para su posterior visualización.

**Buscador:** Localiza páginas que contengan los términos especificados. Utiliza una tabla que enumera las palabras y para cada una especifica las páginas donde se encuentra.

**Correo:** Obtiene información estructurada del usuario.

**Contribuciones:** Permite añadir enlaces o anotaciones a una página, indicando la procedencia de la adición.

**Estadísticas de uso:** Presenta información sobre los acontecimientos producidos en el servidor de WWW. El servidor mantiene un registro (log) de los acontecimientos que se han producido.

**Administración remota del servidor:** Permite interactuar con el servidor desde WWW. Invoca los programas que controlan o modifican el comportamiento del servidor.

## Escenario de activación de un CGI

**1 - Situación inicial:** El cliente solicita la invocación de un CGI, bien de manera involuntaria (se envía únicamente información de cabecera) o bien de forma explícita (formulario). En el formulario hay parejas del tipo variable=valor. El método de HTTP especificado en el formulario puede ser GET o POST.

En el servidor en cambio, el archivo de configuración especifica un directorio cgi-bin con capacidad para ejecutar programas. Puede haber otros ficheros y otros programas a los que puede acceder tanto el servidor como sus CGIs.

**2 - El cliente pulsa el botón de tipo SUBMIT en el formulario:** Dependiendo del método se construye un mensaje que contiene la información del formulario en la cabecera (para GET) o en el cuerpo del mensaje (para POST). El mensaje se envía al servidor, añadiendo información propia del cliente que el

propio navegador conoce. El cliente queda a la espera de recibir un objeto MIME como respuesta del servidor.

**3 - El servidor recibe el mensaje de petición o pone en marcha el programa CGI:** El servidor compara la información del mensaje con la que conoce de su fichero de configuración, determinando así la validez de la petición. En realidad el servidor se pregunta: ¿Existe esta URL? ¿Se tienen todos los permisos?

Prepara el entorno añadiendo información propia a la comunicada por el navegador del cliente. Si es GET, la información procedente del formulario (parejas variables=valor) se definen en QUERY\_STRING. El servidor posteriormente pone en funcionamiento el CGI. Si se trata de POST, la información se coloca en la entrada estándar del CGI. Finalmente se inicia la ejecución del CGI y el servidor espera a que ésta acabe.

**4 - Ejecución del CGI:** El CGI accede a las variables de entorno. Comprueba o adapta el funcionamiento según el método GET o POST establecido en REQUEST\_METHOD: si se tratara de GET, la información estará en QUERY\_STRING, mientras que si se trata de POST, se tomará la entrada estándar.

Se construye un objeto MIME que se enviará al cliente. La primera escritura deberá anunciar el tipo de objeto: CONTENT\_TYPE: tipo/subtipo.

**1 - El servidor vuelve al trabajo:** El servidor añade a su respuesta del CGI una cabecera indicando su tamaño (CONTENT\_LENGTH).

**2 - El cliente recibe la respuesta:** Interpretación de la respuesta. Visualización con el navegador.

## Soporte y configuración del servidor web

Antes de continuar con la programación CGI, asegúrese de que su servidor web sea compatible con CGI y esté configurado para manejar programas CGI. Todos los programas CGI que ejecutará el servidor HTTP se guardan en un directorio preconfigurado. Este directorio se llama CGI Directory y por convención se denomina / var / www / cgi-bin. Por convención, los archivos CGI tienen la extensión. cgi, pero también puede mantener sus archivos con la extensión python .py .

De forma predeterminada, el servidor Linux está configurado para ejecutar solo los scripts en el directorio cgi-bin en / var / www. Si desea especificar cualquier otro directorio para ejecutar sus scripts CGI, comente las siguientes líneas en el archivo httpd.conf (Apache2):

```
<Directory "/var/www/cgi-bin">
```

```
    AllowOverride None
```

```
    Options ExecCGI
```

```
    Order allow,deny
```

```
    Allow from all
```

```
</Directory>
```

```
<Directory "/var/www/cgi-bin">
```

```
Options All
```

```
</Directory>
```

Aquí, asumimos que tiene Web Server funcionando correctamente y que puede ejecutar cualquier otro programa CGI como Perl o Shell, etc.

## Tutorial Ubuntu - Python CGI on Nginx

Install the Python package.

- `apt-get update`
- `apt-get install python3 python3-pip`

Install the Nginx server and the Fcgiwrap package.

- `apt-get install nginx fcgiwrap`

Create a configuration file for the CGI gateway.

- `cp /usr/share/doc/fcgiwrap/examples/nginx.conf /etc/nginx/fcgiwrap.conf`

Here is the file content:

```
location /cgi-bin/ {
    gzip off;
    root /usr/lib;
    fastcgi_pass unix:/var/run/fcgiwrap.socket;
    include /etc/nginx/fastcgi_params;
    fastcgi_param SCRIPT_FILENAME /usr/lib$fastcgi_script_name;
}
```

Create a directory to store the CGI files.

- `mkdir /usr/lib/cgi-bin -p`

Edit the Nginx configuration file for the default website.

- `nano /etc/nginx/sites-available/default`

Insert the following line in the area named SERVER.

`include fcgiwrap.conf;`

Here is the file, before our configuration:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /var/www/html;  
    index index.html index.htm index.nginx-debian.html;  
    server_name _;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Here is the file, after our configuration:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /var/www/html;  
    index index.html index.htm index.nginx-debian.html;  
    server_name _;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
include fcgiwrap.conf;  
}
```

Restart the Nginx service.

- `service nginx restart`

As an example, let's create a Python CGI script.

Use the PIP application to install a Python library named ART.

```
pip3 install art
```

Access the Nginx's CGI directory.

```
cd /usr/lib/cgi-bin
```

Create a test page using Python.

```
nano /usr/lib/cgi-bin/test.py
```

Here is the file content.

```
#!/usr/bin/python3
from art import *
Art=text2art("TEST",font='block',chr_ignore=True)
print('Content-Type: text/plain')
print('')
print('This is my test!')
print(Art)
```

Change the file permission.

```
chmod 755 /usr/lib/cgi-bin/test.py
```

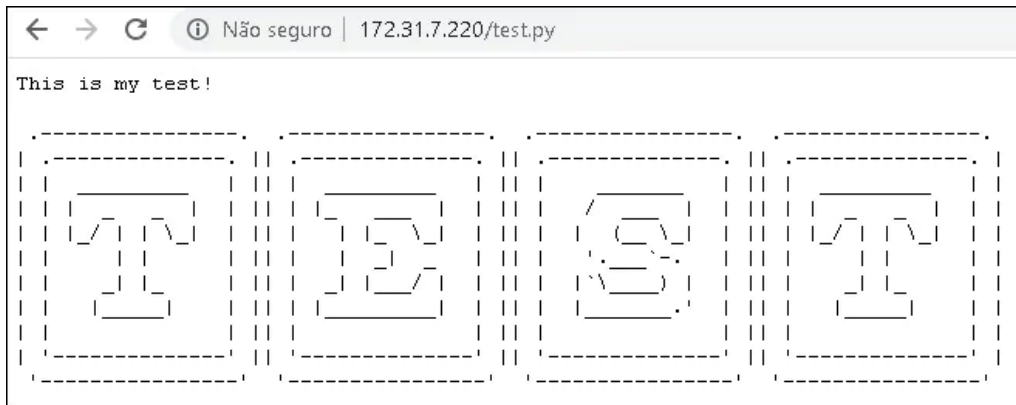
Open your browser and enter the IP address of your web server plus /cgi-bin/test.py.

In our example, the following URL was entered in the Browser:

- `http://192.168.2.70/cgi-bin/test.py`

The Python page should display the following message.





Congratulations! Nginx is now able to use Python scripts as CGI.

## Uso de cookies en CGI

El protocolo HTTP es un protocolo sin estado. Para un sitio web comercial, se requiere mantener la información de la sesión entre diferentes páginas. Por ejemplo, el registro de un usuario finaliza después de completar muchas páginas. ¿Cómo mantener la información de la sesión del usuario en todas las páginas web?

En muchas situaciones, el uso de cookies es el método más eficiente para recordar y rastrear preferencias, compras, comisiones y otra información requerida para una mejor experiencia del visitante o estadísticas del sitio.

## ¿Cómo funciona?

Su servidor envía algunos datos al navegador del visitante en forma de cookie. El navegador puede aceptar la cookie. Si es así, se almacena como un registro de texto sin formato en el disco duro del visitante. Ahora, cuando el visitante llega a otra página de su sitio, la cookie está disponible para su recuperación. Una vez recuperado, su servidor sabe / recuerda lo que se almacenó.

Las cookies son un registro de datos de texto sin formato de 5 campos de longitud variable:

- **Caduca** : la fecha en que caducará la cookie. Si está en blanco, la cookie caducará cuando el visitante salga del navegador.
- **Dominio** : el nombre de dominio de su sitio.
- **Ruta** : la ruta al directorio o página web que establece la cookie. Esto puede estar en blanco si desea recuperar la cookie de cualquier directorio o página.

- **Seguro** : si este campo contiene la palabra "seguro", la cookie solo se puede recuperar con un servidor seguro. Si este campo está en blanco, no existe tal restricción.
- **Nombre = valor** : las cookies se configuran y recuperan en forma de pares de clave y valor.

## Configuración de cookies

Es muy fácil enviar cookies al navegador. Estas cookies se envían junto con el encabezado HTTP antes del campo de tipo de contenido. Suponiendo que desea establecer el ID de usuario y la contraseña como cookies. La configuración de las cookies se realiza de la siguiente manera: