



# UNIVERSIDAD DE GRANADA

Tecnologías Web (2023-2024)  
Grado en Ingeniería Informática

---

## Practica 2

---

Realizado por:  
Claudia de la Vieja Lafuente

1. Ejecute las acciones indicadas e identifique y describa los mensajes HTTP así como sus diferentes partes (primera línea, encabezados, etc.).

```
claudia@claudia-laptop:~$ curl http://void.ugr.es/publico/noredir/hola.html
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Hola mundo</title>
</head>
<body>
  <h1>¡ Hola, mundo !</h1>
</body>
</html>
```

**<!DOCTYPE html>**: Esta línea declara el tipo de documento como HTML5.

**<html lang="es">**: Aquí comienza el elemento raíz **<html>**, que envuelve todo el contenido de la página. El atributo **lang="es"** especifica el idioma de la página, en este caso, español.

**<head>**: Este elemento contiene metadatos sobre el documento, como el título de la página, enlaces a hojas de estilo CSS, scripts JavaScript, y otra información que no se muestra directamente en la página.

**<meta charset="utf-8">**: Esta etiqueta define el estándar de codificación de caracteres que incluye la mayoría de los caracteres escritos en la mayoría de los idiomas del mundo.

**<title>Hola mundo</title>**: Aquí se define el título de la página que se muestra en la pestaña del navegador. En este caso, el título es "Hola mundo".

**</head>**: Cierra la sección de metadatos (**<head>**).

**<body>**: Este elemento contiene todo el contenido visible de la página, como texto, imágenes, enlaces, etc.

**<h1>¡ Hola, mundo !</h1>**: Este es un encabezado de nivel 1 (**<h1>**). Los encabezados se utilizan para estructurar el contenido y proporcionar jerarquía visual.

**</body>**: Cierra la sección del cuerpo (**<body>**).

**</html>**: Cierra el elemento raíz **<html>**, indicando el final del documento HTML.

```

claudia@claudia-laptop:~$ curl -v http://void.ugr.es/publico/noredir/hola.html
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 11:54:41 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4"
< Accept-Ranges: bytes
< Content-Length: 152
< Vary: Accept-Encoding
< Content-Type: text/html
<
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Hola mundo</title>
</head>
<body>
  <h1>¡ Hola, mundo !</h1>
</body>
</html>
* Connection #0 to host void.ugr.es left intact

```

**GET /publico/noredir/hola.html HTTP/1.1:** Esta línea indica que se está realizando una solicitud GET al recurso /publico/noredir/hola.html en el servidor.

**Host: void.ugr.es:** Aquí se especifica el nombre del servidor al que se está realizando la solicitud, en este caso, void.ugr.es.

**User-Agent: curl/7.81.0:** Este encabezado indica qué agente de usuario (navegador o herramienta) está realizando la solicitud. En este caso, se está utilizando cURL versión 7.81.0.

**Accept: \*/\*:** Este encabezado especifica qué tipos de contenido acepta el cliente. En este caso, \*/\* indica que acepta cualquier tipo de contenido.

**\* Mark bundle as not supporting multiuse:** Este mensaje indica que la sesión no soporta la reutilización de conexiones.

**HTTP/1.1 200 OK:** Esta línea indica que el servidor ha respondido con éxito a la solicitud (200 OK).

**Date: Thu, 14 Mar 2024 11:54:41 GMT:** Aquí se muestra la fecha y hora en que se generó la respuesta.

**Server: Apache/2.4.52 (Ubuntu):** Este encabezado indica el servidor web utilizado.

**Strict-Transport-Security: max-age=0:** Este encabezado especifica políticas de seguridad, en este caso, el servidor establece una política de seguridad de transporte estricta con un tiempo de vigencia de 0 segundos.

**Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT:** Indica la fecha y hora en que se modificó por última vez el recurso solicitado.

**ETag: "98-6125b0a4529c4":** Este encabezado proporciona una etiqueta de entidad única para el recurso solicitado.

**Accept-Ranges: bytes:** Indica que el servidor acepta solicitudes para rangos de bytes del recurso.

**Content-Length: 152:** Aquí se especifica la longitud del contenido en bytes.

**Vary: Accept-Encoding:** Este encabezado indica que la respuesta puede variar dependiendo del encabezado de codificación de aceptación del cliente.

**Content-Type: text/html:** Indica el tipo de contenido de la respuesta, en este caso, es HTML.

Este es el final de los encabezados y el comienzo del cuerpo de la respuesta, que como podemos observar es igual que en el apartado anterior.

2. Abra un navegador y escriba la URL <http://void.ugr.es/publico/hola.html> (observe que se ha escrito http y no https). Podrá ver la página del ejemplo anterior y si observa la URL verá que el esquema es HTTPS (aunque usted solicitó HTTP). Si habilita la herramienta de análisis de tráfico HTTP del navegador verá que se han hecho dos peticiones. Esto ocurre porque en void.ugr.es, si se recibe una petición con el esquema HTTP, es redirigida automáticamente para que use el esquema HTTPS. Esto es transparente al usuario del navegador que lo gestiona de forma automática. En caso de usar

cURL, podrá comprobar que la redirección no se hace de forma automática.

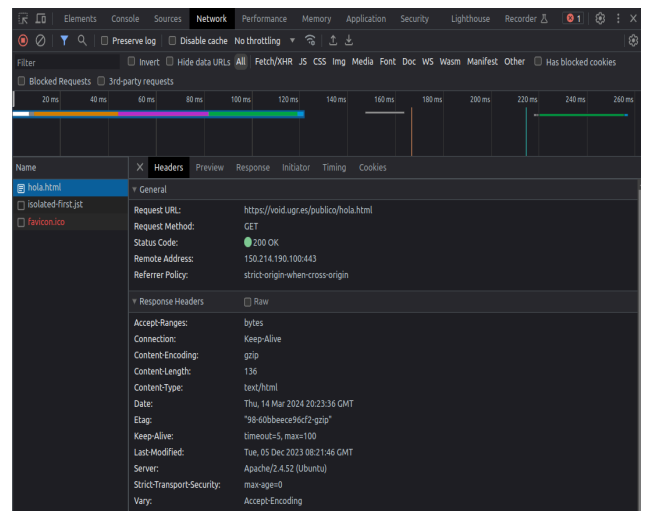
Ejecute: `curl -v http://void.ugr.es/publico/hola.html`

Para que cURL haga también las redirecciones si fuesen necesarias debe

usar el modificador -L: `curl -vL http://void.ugr.es/publico/hola.html`

Ejecute las acciones indicadas con cURL y muestre los resultados. A continuación, hágalo también con el navegador e incluya capturas de pantalla de la herramienta de análisis de tráfico de red.

```
claudia@claudia-laptop:~$ curl -v https://void.ugr.es/publico/hola.html
* Trying 150.214.190.100:443...
* Connected to void.ugr.es (150.214.190.100) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* CAfile: /etc/ssl/certs/ca-certificates.crt
* Capath: /etc/ssl/certs
* TLSv1.0 (OUT), TLS header, Certificate Status (22):
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS header, Finished (20):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.2 (OUT), TLS header, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, change cipher spec (1):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* ALPN, server accepted to use http/1.1
* Server certificate:
* subject: CN=void.ugr.es
* start date: Feb 16 16:33:20 2024 GMT
* expire date: May 16 16:33:19 2024 GMT
* subjectAltName: host "void.ugr.es" matched cert's "void.ugr.es"
* issuer: C=US; O=Let's Encrypt; CN=R3
* SSL certificate verify ok.
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
> GET /publico/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
*
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, NewSession Ticket (4):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, NewSession Ticket (4):
* old SSL session ID is stale, removing
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 20:19:30 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 05 Dec 2023 08:21:46 GMT
< ETag: "98-60bbece96cf2"
< Accept-Ranges: bytes
< Content-Length: 152
< Vary: Accept-Encoding
< Content-Type: text/html
<
<!DOCTYPE html>
<html lang="es">
```

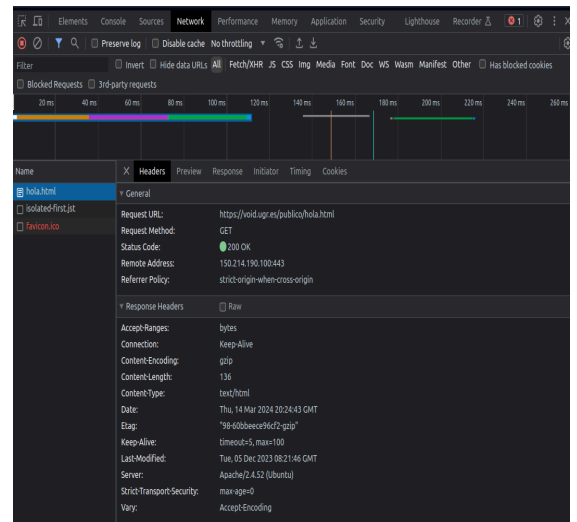


Ejecución de curl -v

```

claudia@claudia-laptop:~$ curl -vL https://void.ugr.es/publico/hola.html
* Trying 150.214.190.100:443...
* Connected to void.ugr.es (150.214.190.100) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* CAfile: /etc/ssl/certs/ca-certificates.crt
* Caphath: /etc/ssl/certs
* TLSv1.0 (OUT), TLS header, Certificate Status (22):
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS header, Finished (20):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.2 (OUT), TLS header, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* Server certificate:
* subject: CN=void.ugr.es
* start date: Feb 16 16:33:20 2024 GMT
* expire date: May 16 16:33:19 2024 GMT
* subjectAltName: host "void.ugr.es" matched cert's "void.ugr.es"
* issuer: C=US; O=Let's Encrypt; CN=R3
* SSL certificate verify ok.
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
> GET /publico/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
>
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 20:24:13 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 05 Dec 2023 08:21:46 GMT
< ETag: "98-60bbece96cf2"
< Accept-Ranges: bytes
< Content-Length: 152
< Vary: Accept-Encoding
< Content-Type: text/html
<

```



## Ejecución de curl -vL

### 3. Ejecute las acciones indicadas y muestre los resultados.

```

claudia@claudia-laptop:~$ curl -v -H "Accept-Encoding: gzip" http://void.ugr.es/publico/noredir/hola.html
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
> Accept-Encoding: gzip
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 20:40:07 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4-gzip"
< Accept-Ranges: bytes
< Vary: Accept-Encoding
< Content-Encoding: gzip
< Content-Length: 136
< Content-Type: text/html
<
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
* Failure writing output to destination
* Closing connection 0

```

El error que nos da puede ocurrir cuando la respuesta del servidor está en un formato que no es compatible con la terminal, como en este caso donde el contenido está comprimido con gzip.

```
claudia@claudia-laptop:~$ curl -v -H "Accept-Encoding: gzip" http://void.ugr.es/publico/noredir/hola.html --output -
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
> Accept-Encoding: gzip
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 20:42:18 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4-gzip"
< Accept-Ranges: bytes
< Vary: Accept-Encoding
< Content-Encoding: gzip
< Content-Length: 136
< Content-Type: text/html
<
* Connection #0 to host void.ugr.es left intact
9eyJ  ;.  D G [F4]
%DI fIN G~N Bni^J >D F & )? 4  B R Z E "C R  d E claudia@claudia-laptop:~$ |
```

El error que he experimentado se debe a que la respuesta del servidor está codificada en gzip, pero la salida en la terminal no se decodifica correctamente. Cuando se solicita contenido comprimido, cURL generalmente lo descomprime automáticamente antes de mostrarlo en la terminal. Sin embargo, en este caso, parece que algo salió mal en la descompresión de la respuesta.

```

claudia@claudia-laptop:~$ curl -v -H "Accept-Encoding: gzip" http://void.ugr.es/publico/noredir/hola.html --output fichero.html.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
  0     0    0     0     0     0      0  0 --:--:-- --:--:-- --:--:--    0*   Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
> Accept-Encoding: gzip
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 21:01:55 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4-gzip"
< Accept-Ranges: bytes
< Vary: Accept-Encoding
< Content-Encoding: gzip
< Content-Length: 136
< Content-Type: text/html
<
{ [136 bytes data]
100  136 100  136    0     0  1528    0 --:--:-- --:--:-- --:--:--  1545
* Connection #0 to host void.ugr.es left intact

```

Se realiza una solicitud GET al archivo hola.html en el servidor void.ugr.es, indicando que acepte la codificación gzip para la respuesta. El servidor responde con un código de estado 200 OK, confirmando que la solicitud fue exitosa. La respuesta contiene el contenido del archivo hola.html, comprimido en formato gzip. Después de recibir los datos, la conexión con el servidor se cierra correctamente.

```

claudia@claudia-laptop:~$ curl -v --compressed http://void.ugr.es/publico/noredir/hola.html
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
> Accept-Encoding: deflate, gzip, br, zstd
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 14 Mar 2024 21:06:07 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4-gzip"
< Accept-Ranges: bytes
< Vary: Accept-Encoding
< Content-Encoding: gzip
< Content-Length: 136
< Content-Type: text/html
<
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Hola mundo</title>
</head>
<body>
  <h1>¡ Hola, mundo !</h1>
</body>
</html>
* Connection #0 to host void.ugr.es left intact

```



4. Ejecute las acciones indicadas y muestre los resultados. Marque la información que aporta cURL en relación con el uso o no de una conexión persistente.

```
claudia@claudia-laptop:~$ sudo curl -v http://void.ugr.es/publico/noredir/hola.html
[sudo] contraseña para claudia:
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 12 Mar 2024 17:31:06 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4"
< Accept-Ranges: bytes
< Content-Length: 152
< Vary: Accept-Encoding
< Content-Type: text/html
<
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Hola mundo</title>
</head>
<body>
  <h1>¡ Hola, mundo !</h1>
</body>
</html>
* Connection #0 to host void.ugr.es left intact
```

```

claudia@claudia-laptop:~$ curl -v -H "Connection: Close" http://void.ugr.es/publico/noredir/hola.html
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
> Connection: Close
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 12 Mar 2024 17:33:57 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4"
< Accept-Ranges: bytes
< Content-Length: 152
< Vary: Accept-Encoding
< Connection: close
< Content-Type: text/html
<
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Hola mundo</title>
</head>
<body>
  <h1>¡ Hola, mundo !</h1>
</body>
</html>
* Closing connection 0

```

```

claudia@claudia-laptop:~$ curl -v -H "Connection: Keep-Alive" http://void.ugr.es/publico/noredir/hola.html
* Trying 150.214.190.100:80...
* Connected to void.ugr.es (150.214.190.100) port 80 (#0)
> GET /publico/noredir/hola.html HTTP/1.1
> Host: void.ugr.es
> User-Agent: curl/7.81.0
> Accept: */*
> Connection: Keep-Alive
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 12 Mar 2024 17:35:09 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Strict-Transport-Security: max-age=0
< Last-Modified: Tue, 27 Feb 2024 11:08:03 GMT
< ETag: "98-6125b0a4529c4"
< Accept-Ranges: bytes
< Content-Length: 152
< Vary: Accept-Encoding
< Keep-Alive: timeout=5, max=100
< Connection: Keep-Alive
< Content-Type: text/html
<
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Hola mundo</title>
</head>
<body>
  <h1>¡ Hola, mundo !</h1>
</body>
</html>
* Connection #0 to host void.ugr.es left intact

```

5. Averigüe qué servidor web utiliza cada uno de los sitios del siguiente listado. Para ello use el navegador y despliegue la ventana de herramientas de desarrollo.

void.ugr.es -> Apache/2.4.52 (Ubuntu)

www.ugr.es -> nginx

upm.es -> Apache

xkcd.com -> nginx

medium.com -> cloudflare

eldiario.es -> nginx

larazon.es -> Apache/2.4.54 () OpenSSL/1.0.2k-fips

elpais.es -> nginx/1.14.2

slack.com -> Apache

cni.es -> Apache

6. Cree un fichero llamado formulario.html en su carpeta public\_html del servidor void.ugr.es

### **Variables recibidas**

#### **Desde \$\_GET**

- nombre = claudiavieja2324
- password = viFH6BlartU76Oic
- enviar = Enviar

#### **Desde \$\_POST**

7. Modifique el formulario y cambie el método de envío a “POST”. A continuación repita el procedimiento del ejercicio anterior desde el navegador. Podrá comprobar que se ha modificado la forma en la que se envían los datos al servidor.

### **Variables recibidas**

#### **Desde \$\_GET**

#### **Desde \$\_POST**

- nombre = pepito
- password = laclave
- enviar = Enviar

8. En los dos ejercicios previos el script procesar.php está en una carpeta sin protección de void.ugr.es. Sin embargo, el documento formulario.html está en su home por lo que para poder acceder a él debe autenticarse. Copie el documento procesar.php en su carpeta public\_html y modifique el formulario para que el action apunte hacia él. Una vez autenticado en el navegador el funcionamiento será el mismo que antes. En cambio, si ahora intenta ejecutar el comando cURL para hacer el envío de datos obtendrá un error 401. Para incluir las credenciales en la ejecución de cURL debe usar el modificador `--user`

Lo único que necesitamos hacer es actualizar la ruta de destino en el archivo formulario.html, para indicar dónde se encuentra ahora el archivo procesar.php. Después de este cambio, cuando enviamos el formulario, todo funciona sin problemas.

Sin embargo, si ejecutamos el comando mencionado en el enunciado, no veremos errores. Esto se debe a que no estamos accediendo al archivo procesar.php en la carpeta principal. Pero si cambiamos la ubicación para que apunte a nuestro archivo procesar.php, obtendremos un error 401. Esto ocurre porque la solicitud no puede completarse ya que no tenemos las credenciales necesarias para acceder al recurso que estamos intentando solicitar.

9. Cree un nuevo documento llamado formulariofi.html con el siguiente contenido:

Desde \$\_GET

Desde \$\_POST

- nombre = Claudia
- password = hola
- enviar = Enviar

Ficheros recibidos:

- imagen =  
Array  
(  
[name] => ABD.png  
[full\_path] => ABD.png  
[type] => image/png  
[tmp\_name] => /tmp/php7XqxVH  
[error] => 0  
[size] => 86137  
)

Practica 1	ABD
<b>Índice</b>	
<b>1 Introducción</b>	<b>3</b>
<b>2 TDA Campo</b>	<b>3</b>
<b>3 TDA Registro</b>	<b>3</b>
3.1 Crear registro . . . . .	4
3.2 Modificar el valor de un campo del registro . . . . .	4
3.3 Obtener el valor de un campo del registro . . . . .	4
3.4 Obtener el estado de un registro . . . . .	5
3.5 Modificar el estado de un registro . . . . .	5
3.6 Obtener el número de atributos de un registro . . . . .	5
3.7 Obtener el tipo de campo de un registro . . . . .	6
3.8 Modificar el tipo de campo de un registro . . . . .	6
3.9 Obtener longitud de un campo . . . . .	6
<b>4 TDA Bloque</b>	<b>6</b>
4.1 Crear bloque . . . . .	7
4.2 Obtener identificación del bloque . . . . .	7
4.3 Obtener el número de registros . . . . .	7
4.4 Obtener el tamaño del bloque . . . . .	7
<b>5 TDA Fichero</b>	<b>8</b>
5.1 Creación de un fichero . . . . .	8

10. Revise la API de Open Library e indique dos formas diferentes de obtener un listado en formato JSON con la bibliografía del autor Antonio Garrido Carrillo (profesor de la UGR). Explique las peticiones HTTP que debe realizar en cada caso. Es posible que en algún caso deba proceder en varios pasos (hacer varias peticiones HTTP).

La primera forma de hacerlo sería haciendo la búsqueda directamente con el nombre del autor:

<https://openlibrary.org/search/authors.json?q=Antonio+Garrido+Carrillo>

La cual al abrirlo en el buscador nos devuelve lo siguiente:

```
{
  "numFound": 1,
  "start": 0,
  "numFoundExact": true,
  "docs": [
    {
      "key": "OL12536697A",
      "type": "author",
      "name": "Antonio Garrido Carrillo",
      "top work": "Estructuras de datos avanzadas con soluciones en C++",
      "work_count": 6,
      "_version_": 1767964953855131649
    }
  ]
}
```

Y la segunda forma seria filtrando por títulos de libros del autor, de la siguiente manera:

<https://openlibrary.org/search.json?author=Antonio+Garrido+Carrillo>

El cual nos devuelve lo siguiente:

```
{
  "numFound": 60,
  "start": 0,
  "numFoundExact": true,
  "docs": [
    {
      "key": "/works/OL34747513W",
      "type": "work",
      "seed": [
        "/books/OL47096624M",
        "/works/OL34747513W",
        "/authors/OL12536697A"
      ],
      "title": "Estructuras de datos avanzadas con soluciones en C++",
      "title_suggest": "Estructuras de datos avanzadas con soluciones en C++",
      "title_sort": "Estructuras de datos avanzadas con soluciones en C++",
      "edition_count": 1,
      "edition_key": [
        "OL47096624M"
      ],
      "publish_date": [
        "Oct_03, 2018"
      ],
      "publish_year": [
        2018
      ],
      "first_publish_year": 2018,
      "number_of_pages_median": 576,
      "isbn": [
        "8433863363",
        "9788433863362"
      ],
      "last_modified_i": 1678911939,
      "ebook_count_i": 0,
      "ebook_access": "no_ebook",
      "has_fulltext": false,
      "public_scan_b": false,
      "cover_edition_key": "OL47096624M",
      "cover_i": 13468530,
      "publisher": [
        "Editorial Universidad de Granada"
      ],
      "author_key": [
        "OL12536697A"
      ],
      "author_name": [
        "Antonio Garrido Carrillo"
      ],
      "publisher_facet": [
        "Editorial Universidad de Granada"
      ],
      "version": 1767932631110909952,
      "author_facet": [
        "OL12536697A Antonio Garrido Carrillo"
      ]
    }
  ],
}
```

11. Una de estas herramientas básicas es tcpdump. Mire la documentación de este software y averigüe cómo usarlo para capturar los mensajes HTTP entre su máquina y void.ugr.es. Para ello, desde un navegador Firefox, solicite las URL que se indican a continuación y use tcpdump para ver el tráfico HTTP: