
PRACTICA 1: INSTALACIÓN Y CONFIGURACIÓN DE SERVIDORES DE INTERNET Y VISUALIZACIÓN DE CABECERAS DEL PROTOCOLO HTTP

1 Instalación y configuración de Tomcat

1.1 Instalación

Para instalar Tomcat, primero debemos actualizar la lista de paquetes de actualización, para ello accedemos a la terminal e introducimos:

sudo apt update && sudo apt upgrade -y

```
Filip@filip-IdeaPad-3-15ALC6:~$ sudo apt update && sudo apt upgrade -y
[sudo] lozinka za filip:
Pogodak:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Preuzimanje:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Preuzimanje:3 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease [18,1 kB]
Pogodak:4 http://hr.archive.ubuntu.com/ubuntu jammy InRelease
Preuzimanje:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43,1 kB]
Preuzimanje:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [40,1 kB]
Preuzimanje:7 http://hr.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Preuzimanje:8 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy/main i386 Packages [596 kB]
Preuzimanje:9 http://hr.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Preuzimanje:10 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy/main amd64 Packages [894 kB]
Preuzimanje:11 http://hr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.056 kB]
Preuzimanje:12 http://hr.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [503 kB]
```

Ya que Tomcat es un servidor Java, así que necesitamos tener instalado JDK en nuestro sistema. Para ello introducimos el siguiente comando en el terminal:

sudo apt install openjdk-11-jdk

```
filip@filip-IdeaPad-3-15ALC6:~$ sudo apt install openjdk-11-jdk
Čitanje popisa paketa... Završeno
Izgradnja stabla zavisnosti... Završeno
Čitanje informacija stanja... Završeno
openjdk-11-jdk je već najnovije inačice (11.0.20.1+1-0ubuntu1~22.04).
openjdk-11-jdk je postavljen za ručnu instalaciju.
Sljedeći paketi su automatski instalirani i više nisu potrebni:
  linux-headers-6.2.0-26-generic linux-hwe-6.2-headers-6.2.0-26
  linux-image-6.2.0-26-generic linux-modules-6.2.0-26-generic
  linux-modules-extra-6.2.0-26-generic
Koristi 'sudo apt autoremove' kako bi ih uklonili.
0 nadograđenih, 0 novo instaliranih, 0 za uklanjanje i 2 bez nadogradnje.
filip@filip-IdeaPad-3-15ALC6:~$
```

Podemos verificar la instalación con:

java -version

```
filip@filip-IdeaPad-3-15ALC6:~$ java -version
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
filip@filip-IdeaPad-3-15ALC6:~$
```

Ahora crearemos un nuevo usuario "Tomcat" tal que el directorio principal sea: /opt/tomcat.

```
filip@filip-IdeaPad-3-15ALC6:~$ sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
useradd: user 'tomcat' already exists
filip@filip-IdeaPad-3-15ALC6:~$
```

Descargamos el tar.gz con el comando:

wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.10/bin/apache-tomcat-10.1.10.tar.gz

```
filip@filip-IdeaPad-3-15ALC6:~$ wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.10/bin/apache-tomcat-10.1.10.tar.gz
--2023-10-05 16:37:48-- https://downloads.apache.org/tomcat/tomcat-10/v10.1.10/bin/apache-tomcat-10.1.10.tar.gz
Razrješavam downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.9.95.219, 2a01:4f8:10a:201a::2, ...
Povezivanje s downloads.apache.org (downloads.apache.org)[135.181.214.104]:443...
. povezano.
```

Una vez la descarga ha terminado lo descargamos en /opt/tomcat de la siguiente manera:

```
sudo tar xzf apache-tomcat-*.tar.gz -C /opt/tomcat
```

1.2 Configuración de Tomcat

Ahora que tenemos el servidor tomcat instalado debemos asegurarnos el acceso al administrador de páginas. Para ello editamos el fichero `conf/tomcat-users.xml` y pegamos el siguiente código entre los tags `<tomcat-users>` `</tomcat-users>`

```
<!-- user manager can access only the manager section -->
<role rolename="manager-gui" />
<user username="manager" password="_SECRET_PASSWORD_" roles="manager-gui" />

<!-- user admin can access manager and admin section both -->
<role rolename="admin-gui" />
<user username="admin" password="_SECRET_PASSWORD_" roles="manager-gui,admin-gui" />
```

Ahora creamos un archivo de servicio systemd para manejar el servicio de Tomcat. Abrimos un nuevo archivo de servicio con el siguiente comando:

```
nano /etc/systemd/system/tomcat.service
```

Copiamos la siguiente información:

```
[Unit]
Description=Apache Tomcat 10 Web Application Server
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Tras crear el archivo de servicio reactivamos el Daemon de systemd para que lea el nuevo archivo:

```
sudo systemctl daemon-reload
```

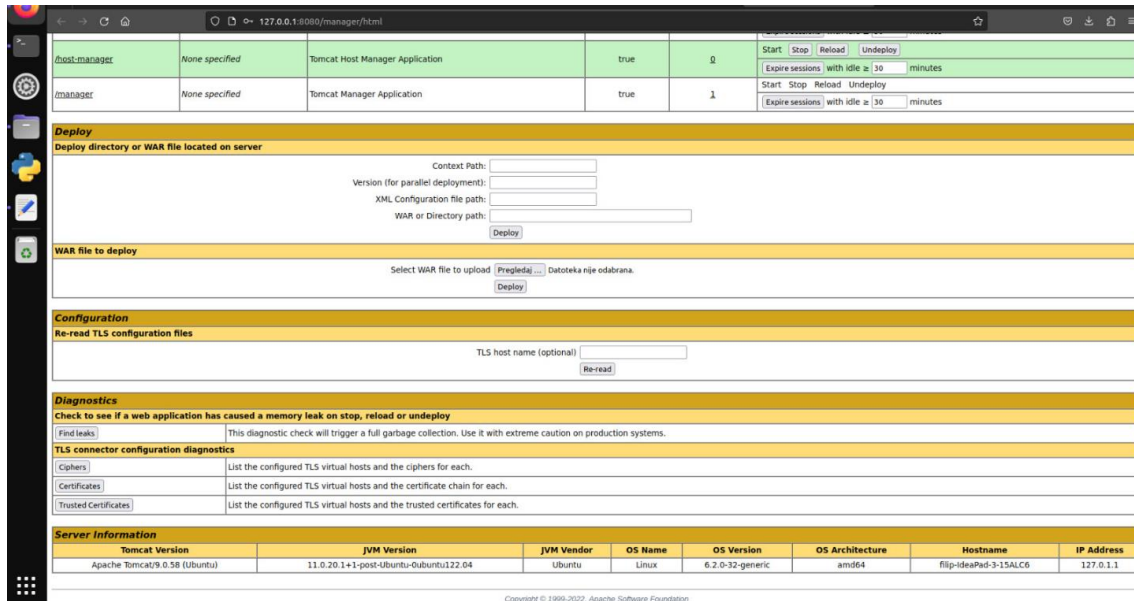
```
filip@filip-IdeaPad-3-15ALC6:~$ sudo systemctl daemon-reload
filip@filip-IdeaPad-3-15ALC6:~$
```

Activamos que Tomcat se habrá al encender:

sudo systemctl enable tomcat

```
filip@filip-IdeaPad-3-15ALC6:~$ sudo systemctl enable tomcat
filip@filip-IdeaPad-3-15ALC6:~$
```

Ahora podremos acceder a tomcat poniendo en el navegador: 127.0.0.1:8080/manager



1.3 Gestión de aplicaciones

Tomcat dispone de enlaces dedicados a la Administración: Manager app. Para ello debemos identificarse

Inicie sesión para obtener acceso a este sitio

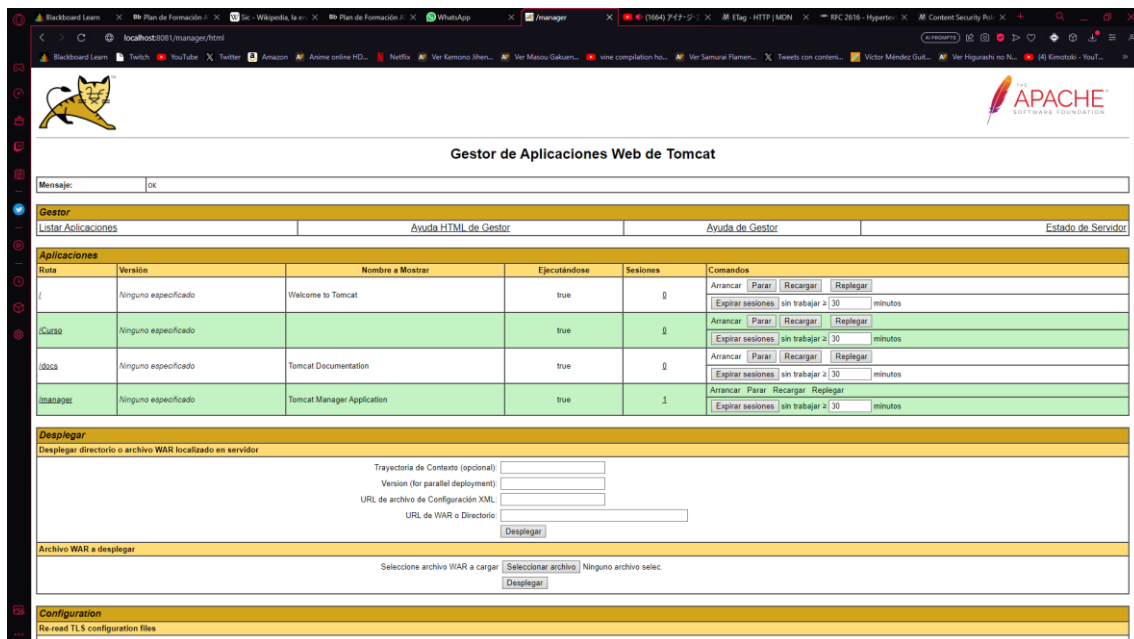
Autorización requerida por <http://localhost:8081>

Nombre de usuario

Contraseña

Iniciar sesiónCancelar

Esto nos brinda acceso a la siguiente página:



Además, nos permite subir archivos war para desplegar aplicaciones de forma automática:

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones"/> <input type="text" value="sin trabajar a 30"/> minutos
/Curso	Ninguno especificado		true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones"/> <input type="text" value="sin trabajar a 30"/> minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones"/> <input type="text" value="sin trabajar a 30"/> minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones"/> <input type="text" value="sin trabajar a 30"/> minutos



1.4 Host Virtuales

El objetivo es hacer 4 host virtuales que cuelguen de localhost y la ip 127.0.0.1 en los puertos :80 y :8085. Para ello accedemos al fichero de configuración y editarlo para añadir 2 servicios con el código que se muestra a continuación:

```

<Service name="Servicio1">
  <Connector port="80" protocol="HTTP/1.1" />
  <Engine name="Servicio1" defaultHost="localhost">
    <Host name="localhost" appBase="vhost1" ></Host>
    <Host name="127.0.0.1" appBase="vhost2" ></Host>
  </Engine>
</Service>

<Service name="Servicio2">
  <Connector port="8085" protocol="HTTP/1.1" />
  <Engine name="Servicio2" defaultHost="localhost">
    <Host name="localhost" appBase="vhost3" ></Host>
    <Host name="127.0.0.1" appBase="vhost4" ></Host>
  </Engine>
</Service>


```

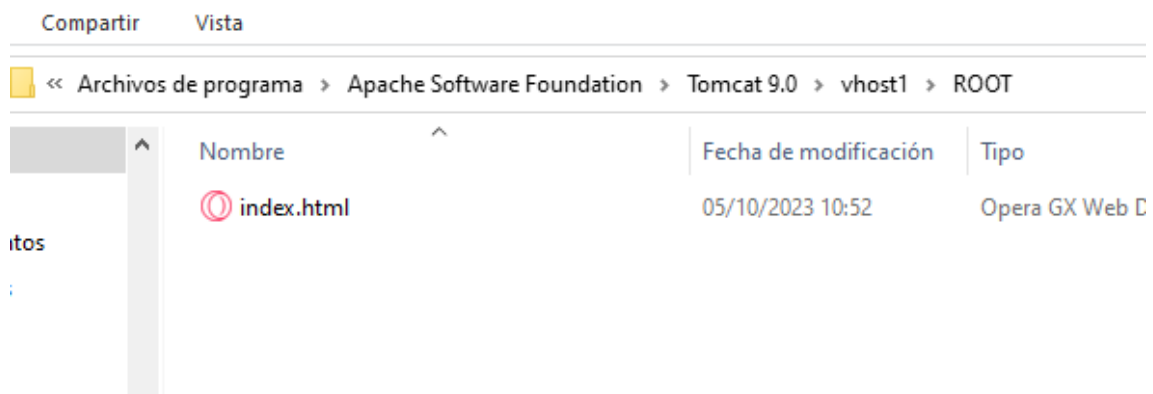
Una vez establecido los servicios con nuestros hosts virtuales debemos crear las carpetas contenedoras de los mismos de la siguiente manera:

logs	05/10/2023 10:28	Carpeta de archivos	
temp	28/09/2023 16:26	Carpeta de archivos	
vhost1	05/10/2023 10:56	Carpeta de archivos	
vhost2	05/10/2023 11:11	Carpeta de archivos	
vhost3	05/10/2023 11:11	Carpeta de archivos	
vhost4	05/10/2023 11:11	Carpeta de archivos	
webapps	05/10/2023 11:00	Carpeta de archivos	
work	05/10/2023 11:11	Carpeta de archivos	
LICENSE	23/08/2023 23:59	Archivo	57 KB
NOTICE	23/08/2023 23:59	Archivo	2 KB

De cada uno de los vhost debe colgar un directorio raíz (ROOT) en el que este nuestra página virtual, por tanto, nuestra dirección quedaría tal que así:

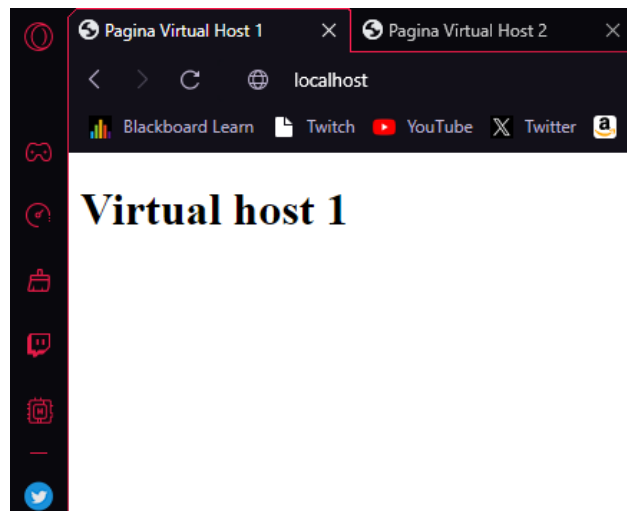
\$CATALINA_HOME/vhost1/ROOT/index.html

Vista			
> Archivos de programa > Apache Software Foundation > Tomcat 9.0 > vhost1 >			
Nombre	Fecha de modificación	Tipo	
 ROOT	05/10/2023 10:56	Carpeta de archivos	

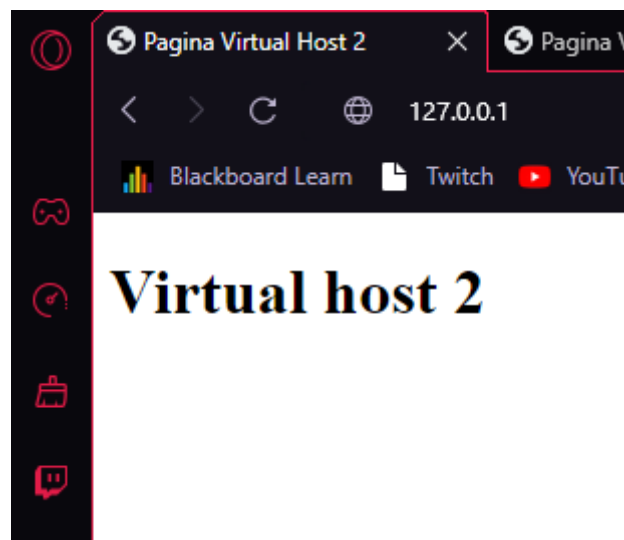


Una vez hecho todo esto, ha apagado y vuelto a encender el servido para aplicar los cambios seremos capaces de acceder a nuestros servicios desde las direcciones indicadas:

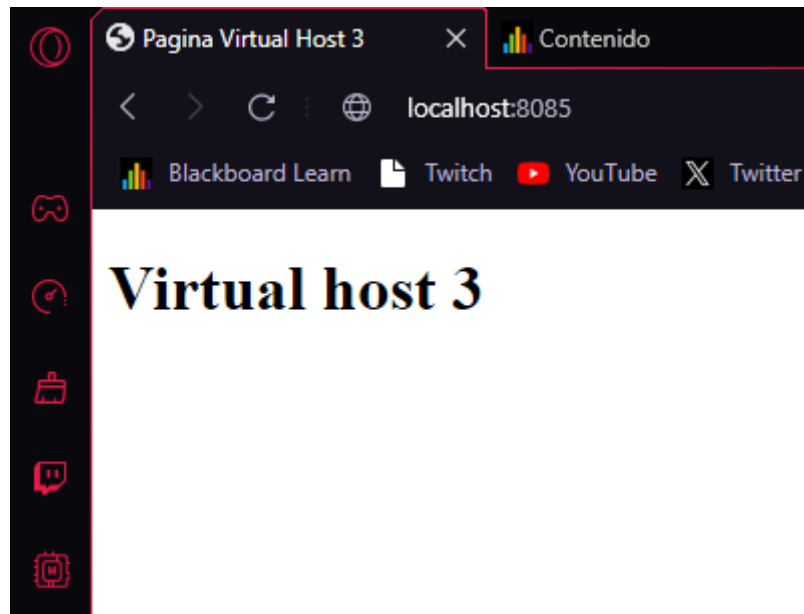
- localhost:80:



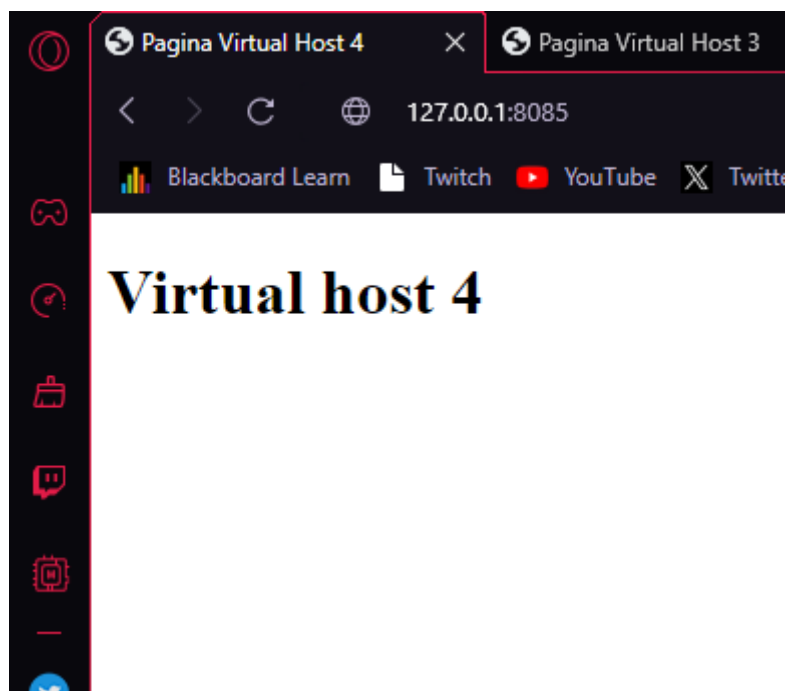
- 127.0.0.1:80:



- localhost:8085



- 127.0.0.1:8085



Todas estas paginas tienen el mismo código base, con las modificaciones necesarias:


```
index.html x server.xml x entrada.htm x
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Pagina Virtual Host 1</title>
  </head>
  <body>
    <h1>Virtual host 1</h1>
  </body>
</html>
```

3.Ejercicio Cabeceras HTTP

En este ejercicio se nos pide analizar las cabeceras HTTP de peticiones GET y POST de distintas páginas web.

3.1 GET

www.abc.es

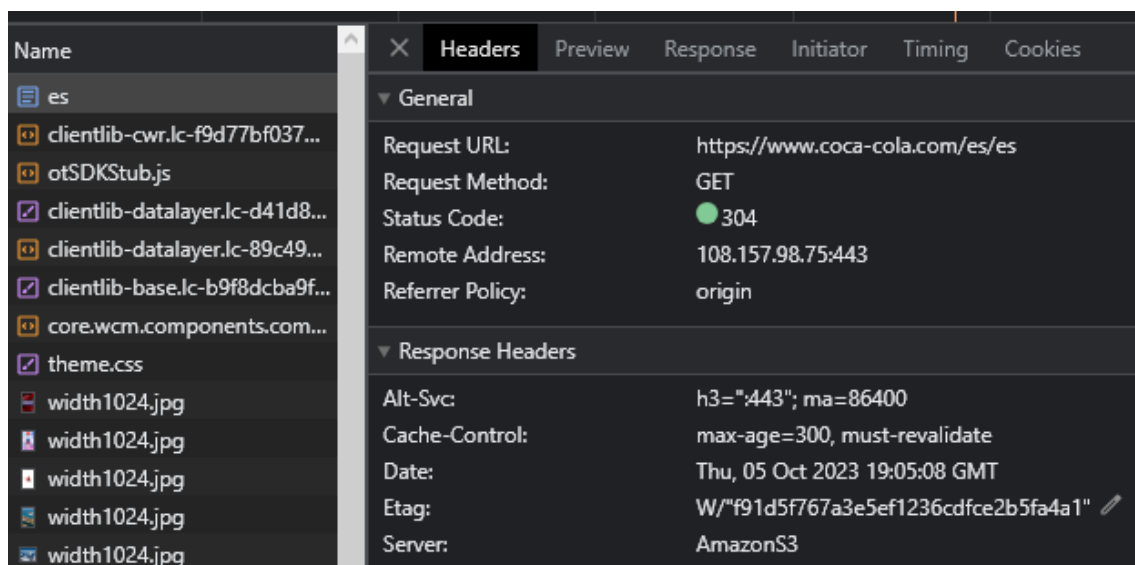
Request URL:	https://www.abc.es/
Request Method:	GET
Status Code:	● 200
Remote Address:	23.200.66.154:443
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers	
Alt-Svc:	h3=":443"; ma=93600
Cache-Control:	max-age=0, no-cache, no-store
Content-Encoding:	gzip
Content-Type:	text/html; charset=UTF-8
Date:	Thu, 05 Oct 2023 18:34:08 GMT
Expires:	Thu, 05 Oct 2023 18:34:08 GMT

- “Request URL”: Nos da la URL a la que le estamos pidiendo un contenido
- “Request Method”: El método utilizado, en este caso, un GET
- “Status Code”: El código de estado nos da información sobre la clase de respuesta. En este caso, 200 quiere decir que todo ha ido bien.
- Remote Address: La dirección IP del servidor, en este caso es 230.200.66.154:443
- Referrer Policy: Referer es un header que contiene la url del botón atrás del navegador. En este caso strict-origin-when-cross-origin evita que el referer desvelar información privada de la url
- Alt-Svc: Esta cabecera indica al servidor que otra ubicación de red puede ser autoritativa por otro origen cuando se hacen otras peticiones. En este caso h3=":443" indica la versión

3 de HTTP y que si hay otras conexiones de ese puerto pueden autorizar otras peticiones, y ma=93600 es el número de segundos en los que al recurso se le trata como nuevo.

- Cache Control: Directivas para el control del cache, en este caso no se guarda info en la cache.
- Content-Encoding: método de codificación, en este caso se ha usado gzip.
- Content-Type: tipo de contenido, en este caso es texto en formato html que usa el set de caracteres UTF-8.
- Date: Fecha y hora de la petición

www.coca-cola.com/es/es



- "Request URL": Nos da la URL a la que le estamos pidiendo un contenido
- "Request Method": El método utilizado, en este caso, un GET
- "Status Code": En este caso el código de estado es 304, que quiere decir que la pagina no ha sido modificada desde mi última petición
- Remote Address: La dirección IP del servidor, en este caso es 108.157.98.75:443
- Referrer Policy: Referer es un header que contiene la url del botón atrás del navegador. En este caso origin, nos referencia a la página origen del servidor
- Alt-Svc: Esta cabecera indica al servidor que otra ubicación de red puede ser autoritativa por otro origen cuando se hacen otras peticiones. En este caso h3=":443" indica la versión 3 de HTTP y que si hay otras conexiones de ese puerto pueden autorizar otras peticiones, y ma=86400 es el número de segundos en los que al recurso se le trata como nuevo.

- Cache Control: El tiempo máximo definido en max-age es de 300 segundos, y la directiva de must-revalidate indica que el cache no debe usar la entrada que venga después de la misma.
- Date: Fecha y hora de la petición.
- Etag: identificador de la versión de un recurso.
- Server: Servidor en el que se aloja la página Web.

www.uah.es

The screenshot shows the 'Headers' tab in a web browser's developer tools. On the left, a list of resources is shown, including 'www.uah.es', 'es/', and various CSS and JS files. The 'es/' resource is selected. The main panel displays the 'General' and 'Response Headers' sections. The 'General' section shows the Request URL as 'https://www.uah.es/es/', the Request Method as 'GET', the Status Code as '200 200', the Remote Address as '193.146.56.125:443', and the Referrer Policy as 'strict-origin-when-cross-origin'. The 'Response Headers' section shows the Cache-Control as 'public, max-age=0', the Connection as 'close', the Content-Encoding as 'gzip', the Content-Type as 'text/html; charset=UTF-8', and the Date as 'Thu, 05 Oct 2023 19:18:06 GMT'.

Header	Value
Request URL	https://www.uah.es/es/
Request Method	GET
Status Code	200 200
Remote Address	193.146.56.125:443
Referrer Policy	strict-origin-when-cross-origin
Cache-Control	public, max-age=0
Connection	close
Content-Encoding	gzip
Content-Type	text/html; charset=UTF-8
Date	Thu, 05 Oct 2023 19:18:06 GMT

- "Request URL": Nos da la URL a la que le estamos pidiendo un contenido
- "Request Method": El método utilizado, en este caso, un GET
- "Status Code": El código de estado nos da información sobre la clase de respuesta. En este caso, 200 quiere decir que todo ha ido bien.
- Remote Address: La dirección IP del servidor, en este caso es 193.146.56.125:443
- Referrer Policy: Referer es un header que contiene la url del botón atrás del navegador. En este caso strict-origin-when-cross-origin evita que el referer desvele información privada de la url
- Cache Control: Directivas para el control del cache, en este caso public indica que el cache puede ser cogido incluso si no debería serlo por una cache compartida.
- Connection: Indica si la conexión debe ser persistente o no, en este caso la directiva close quiere decir que no lo es
- Content-Encoding: método de codificación, en este caso se ha usado gzip.
- Content-Type: tipo de contenido, en este caso es texto en formato HTML que usa el set de caracteres UTF-8.
- Date: Fecha y hora de la petición.

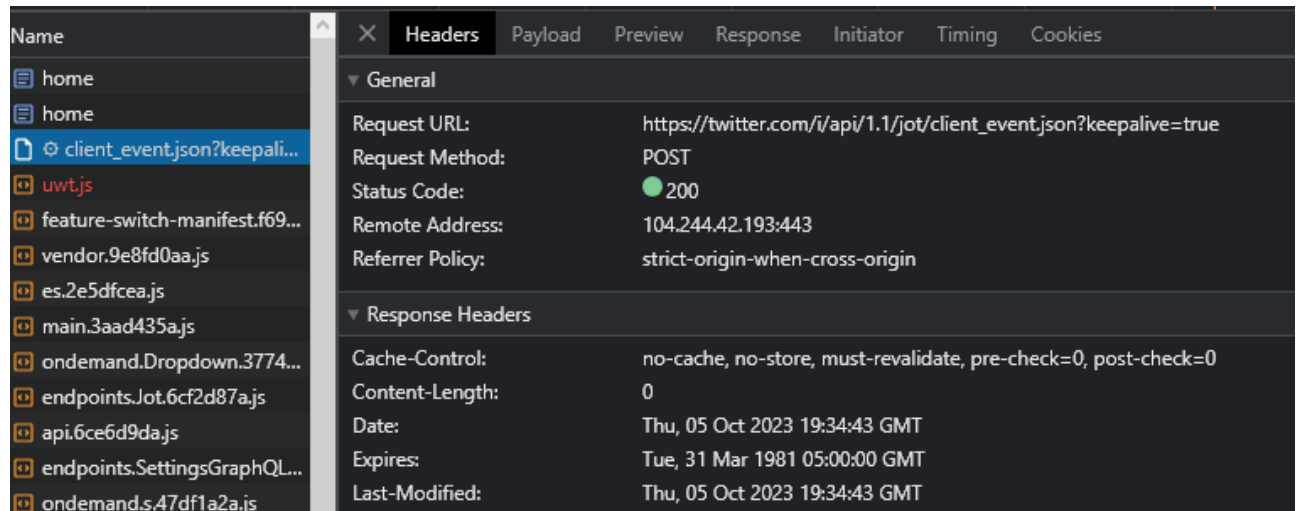
POST:

outlook.office365.com/mail/

Name	×	Headers	Payload	Preview	Response	Initiator	Timing	Cookie
login								
converged.v2.login.min_itjvsk...								
ConvergedKmsi_Core_IDY4eliE...								
ux.converged.kmsi.strings-en...								
Me.htm?v=3								
illustration?ts=636540499926...								
bannerlogo?ts=63654049994...								
convergedkmsi_customization...								
		▼ General						
		Request URL:	https://login.microsoftonline.com/common/login					
		Request Method:	POST					
		Status Code:	200 OK					
		Remote Address:	[2603:1027:1:28::a]:443					
		Referrer Policy:	strict-origin-when-cross-origin					
		▼ Response Headers	Raw					
		Cache-Control:	no-store, no-cache					
		Content-Encoding:	gzip					
		Content-Length:	10404					
		Content-Type:	text/html; charset=utf-8					
		Date:	Thu, 05 Oct 2023 19:28:37 GMT					

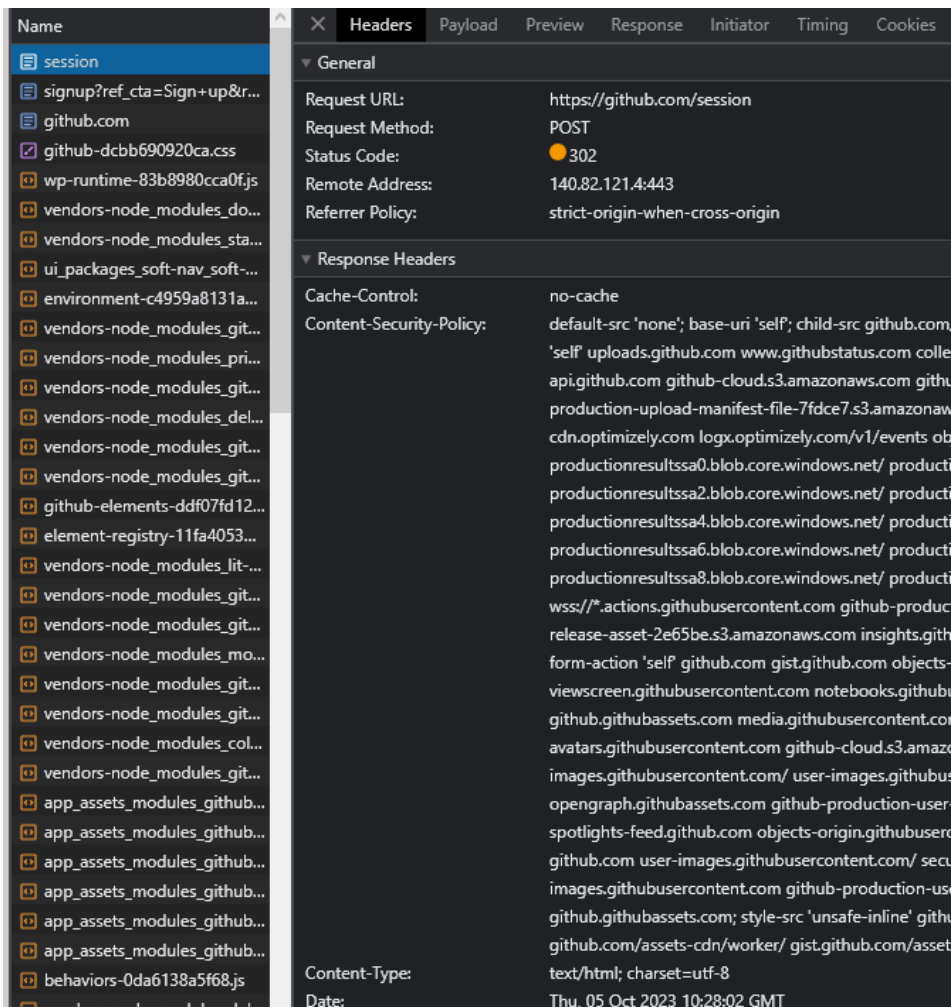
- “Request URL”: Nos da la URL a la que le estamos pidiendo un contenido
- “Request Method”: El método utilizado, en este caso, un POST, es decir, enviamos información
- “Status Code”: El código de estado nos da información sobre la clase de respuesta. En este caso, 200 quiere decir que todo ha ido bien.
- Remote Address: La dirección IP del servidor, en este caso es [2603:1027:1:28::a]:443
- Referrer Policy: Referer es un header que contiene la url del botón atrás del navegador. En este caso strict-origin-when-cross-origin evita que el referer desvelar información privada de la url
- Cache Control: Directivas para el control del cache, en este caso no se guarda información en la cache.
- Content-Encoding: método de codificación, en este caso se ha usado gzip.
- Content-Length: Indica el tamaño de la petición en octetos, en este caso 10404.
- Content-Type: tipo de contenido, en este caso es texto en formato html que usa el set de caracteres UTF-8.
- Date: Fecha y hora de la petición.

<https://twitter.com>



- “Request URL”: Nos da la URL a la que le estamos pidiendo un contenido
- “Request Method”: El método utilizado, en este caso, un POST, es decir, enviamos información
- “Status Code”: El código de estado nos da información sobre la clase de respuesta. En este caso, 200 quiere decir que todo ha ido bien.
- Remote Address: La dirección IP del servidor, en este caso es 104.244.42.193:443
- Referrer Policy: Referer es un header que contiene la url del botón atrás del navegador. En este caso strict-origin-when-cross-origin evita que el referer desvelar información privada de la url
- Cache Control: Directivas para el control del cache, en este caso no se guarda info en la cache, de must-revalidate indica que el cache no debe usar la entrada que venga después de la misma. Apparently pre-check=0, post-check=0 es un vestigio de como internet explorer usaba las caches y no tiene ninguna utilidad.
- Content-Length: Indica el tamaño de la petición en octetos, en este caso es 0, lo que indica que la petición no tiene contenido.
- Date: Fecha y hora de la petición.
- Expires: Es la fecha a partir de la cual el recurso se considera viejo. El hecho que sea una fecha pasada indica que este recurso es un vestigio de la internet antigua
- Last-Modified: Fecha en la que el servidor cree que la variante ha cambiado

<https://github.com>



- Request URL: Nos da la URL a la que le estamos pidiendo un contenido
- Request Method: El método utilizado, en este caso, un POST, es decir, enviamos información
- Status Code: El código de estado nos da información sobre la clase de respuesta. En este caso, 302 quiere decir que ha sido encontrado ("FOUND") indica que el usuario no debe ser redireccionado automáticamente.
- Remote Address: La dirección IP del servidor, en este caso es 140.82.121.4:443
- Referrer Policy: Referer es un header que contiene la url del botón atrás del navegador. En este caso strict-origin-when-cross-origin evita que el referer desvelar información privada de la url
- Cache Control: Directivas para el control del cache, no-cache sin nombre indica que el cache no debe usar la respuesta la satisfacer una petición que venga derivada de la primera.
- Content-Security-Policy: Una capa adicional de seguridad. Indica la url de la directiva que está siguiendo.
- Content-Type: tipo de contenido, en este caso es texto en formato html que usa el set de caracteres UTF-8.
- Date: Fecha y hora de la petición.

