Arquitectura y Diseño de Sistemas Web y C/S

Logotipo, nombre de la empresa

Descripción generada automáticamente

Práctica 1:

Instalación y Configuración de Servidores de Internet y Visualización de cabeceras del protocolo HTTP

David Bachiller Vela 03221211S

Alejandro Rodríguez Blanco 03206933C

Víctor Sanavia Valdeolivas

Contenido

[Introducción 3](#_Toc147412194)

[Instalación y Configuración de Servidores de Internet: Apache-Tomcat 3](#_Toc147412195)

[Instalación 3](#_Toc147412196)

[Configuración Básica 6](#_Toc147412197)

[Host Virtuales 8](#_Toc147412198)

[Introducción cabeceras HTTP 10](#_Toc147412199)

[Visualización de Cabeceras HTTP 12](#_Toc147412200)

[Ejercicio 13](#_Toc147412201)

# Introducción

En esta práctica vamos a familiarizarnos con la herramienta Apache Tomcat que nos servirá para conocer más detalladamente como es la comunicación con un servidor WEB a través de un cliente, que será nuestro navegador. Aprenderemos a configurarlo, gestionarlo y a trabajar con la herramienta

Además, finalizaremos está analizando trazas HTTP y visualizando sus cabeceras, para adquirir una visión más profunda al respecto.

# Instalación y Configuración de Servidores de Internet: Apache-Tomcat

## Instalación

El primer paso y el más importante para el correcto funcionamiento durante el resto de la práctica es la instalación. Vamos a ir paso a paso:

1. Descargar el archivo de instalación.

Nos tenemos que dirigir a <https://tomcat.apache.org/> desde la página de inicio podemos seleccionar la versión que se nos ajuste más a nuestras necesidades y compatibilidad de versiones con java. En mi caso hemos descargado la versión 9.0.80 que se puede encontrar bien deslizando la pagina hacia abajo hasta encontrar la siguiente información.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza media

O bien en el menú de la izquierda.

Una vez en la página de instalación seleccionamos **Windows Service Installer** para reducir posibles errores ya que obtenemos un ejecutable válido para el sistema operativo Windows.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteUna vez descargado el instalador basta con ejecutarlo para comenzar el proceso de instalación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El usuario y contraseña son importantes para un paso posterior en la práctica. En nuestro caso hemos escogido un usuario y contraseña simples como es “admin”. El puerto al que nos vamos a conectar inicialmente es el puerto 8080.

Algo muy importante y previo al proceso de instalación de este servidor de aplicaciones es la instalación de la máquina virtual java ya que servidor web que está diseñado principalmente para ejecutar aplicaciones web desarrolladas en Java. Java proporciona la plataforma en la que se ejecutan las aplicaciones web de Tomcat.

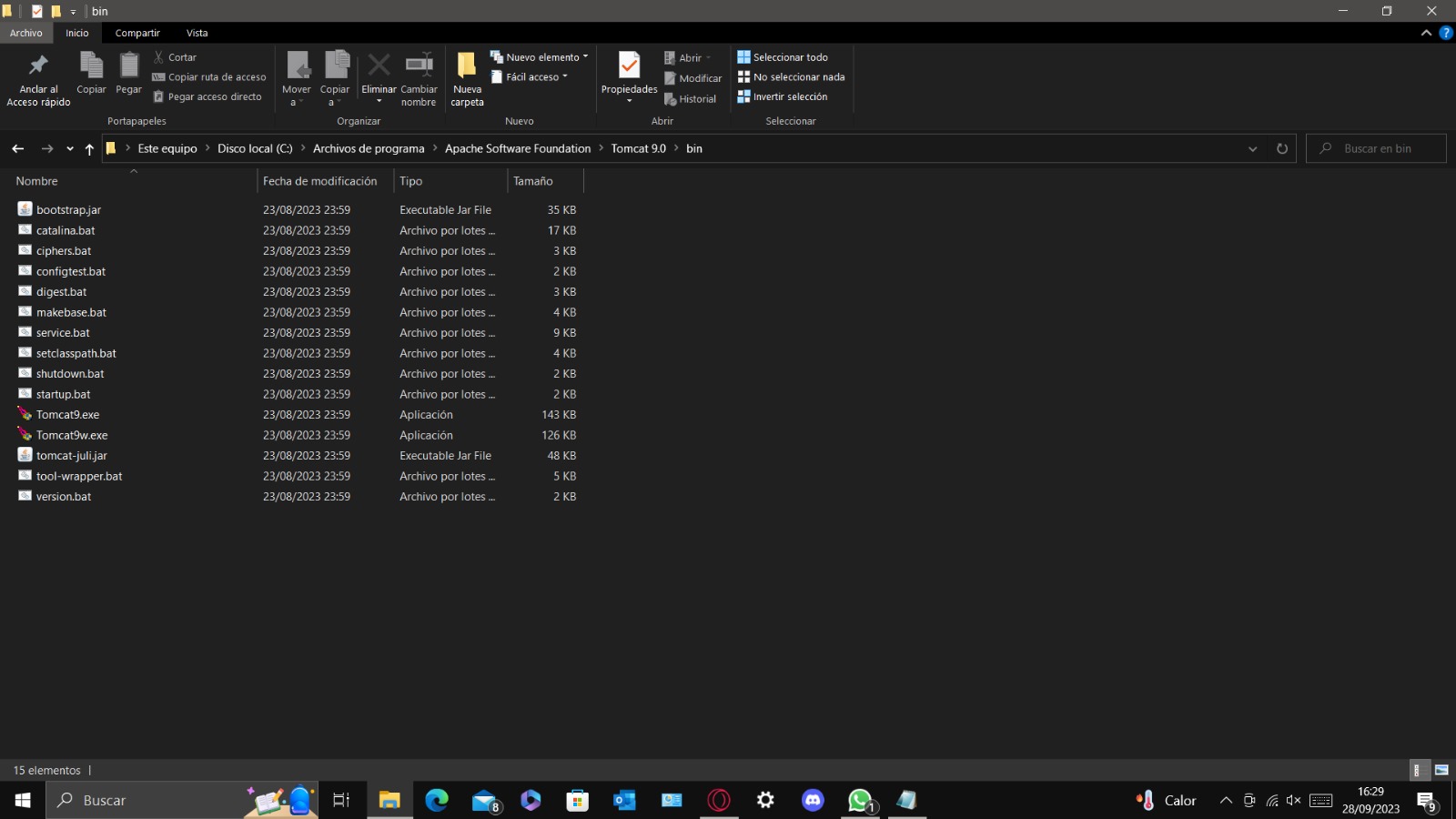
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Sitio web

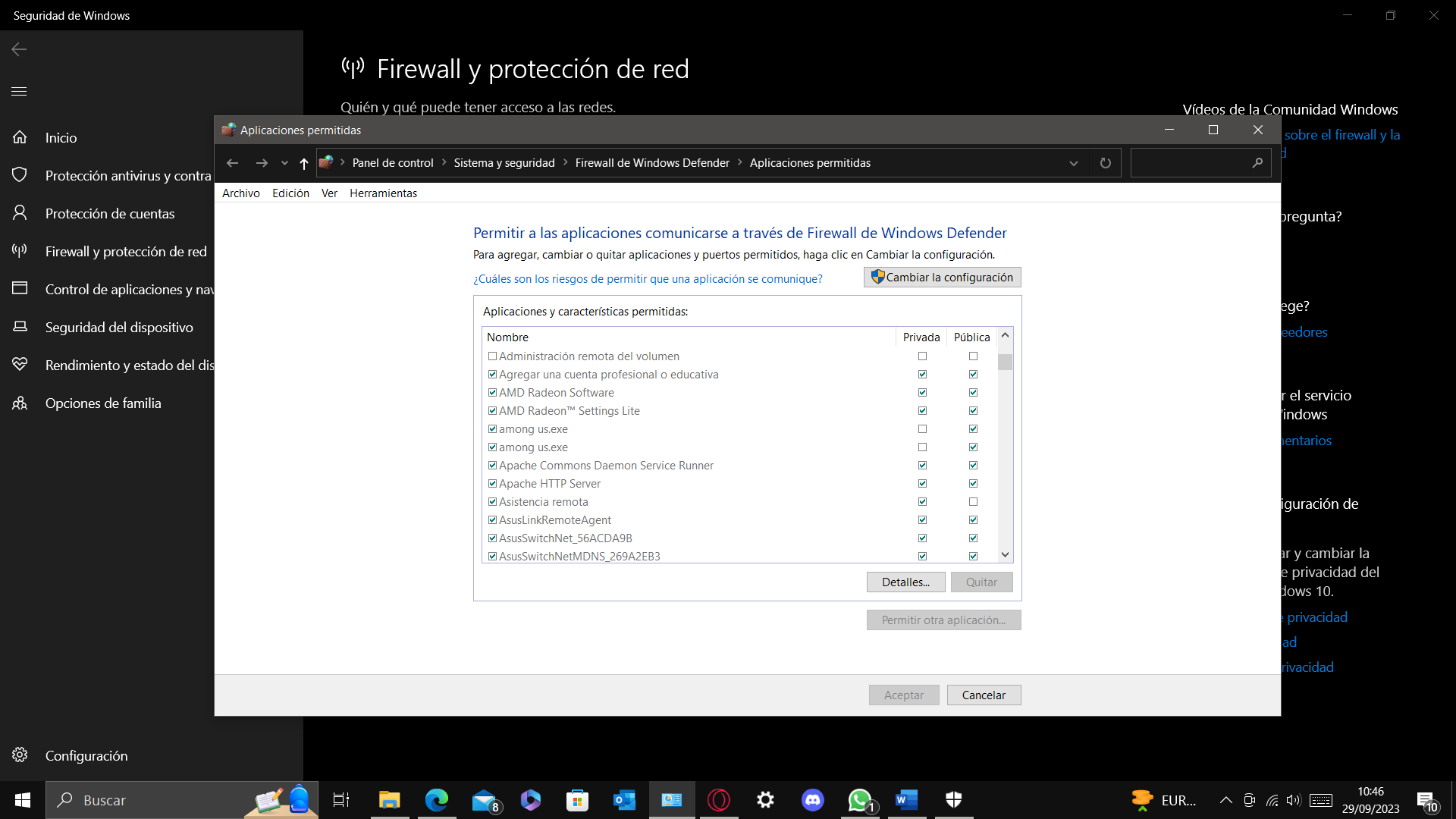
Descripción generada automáticamente

Posteriormente se elige la localización donde se va a instalar Apache Tomcat, el instalador por defecto crea la carpeta **Apache Software Foundation**.

Una vez instalado todo, nos va a pedir permisos de administrador para poder acceder a la carpeta y vamos a tener que darle permisos a Windows Defender Firewall para poder ejecutar el servidor y poder acceder a los puertos.



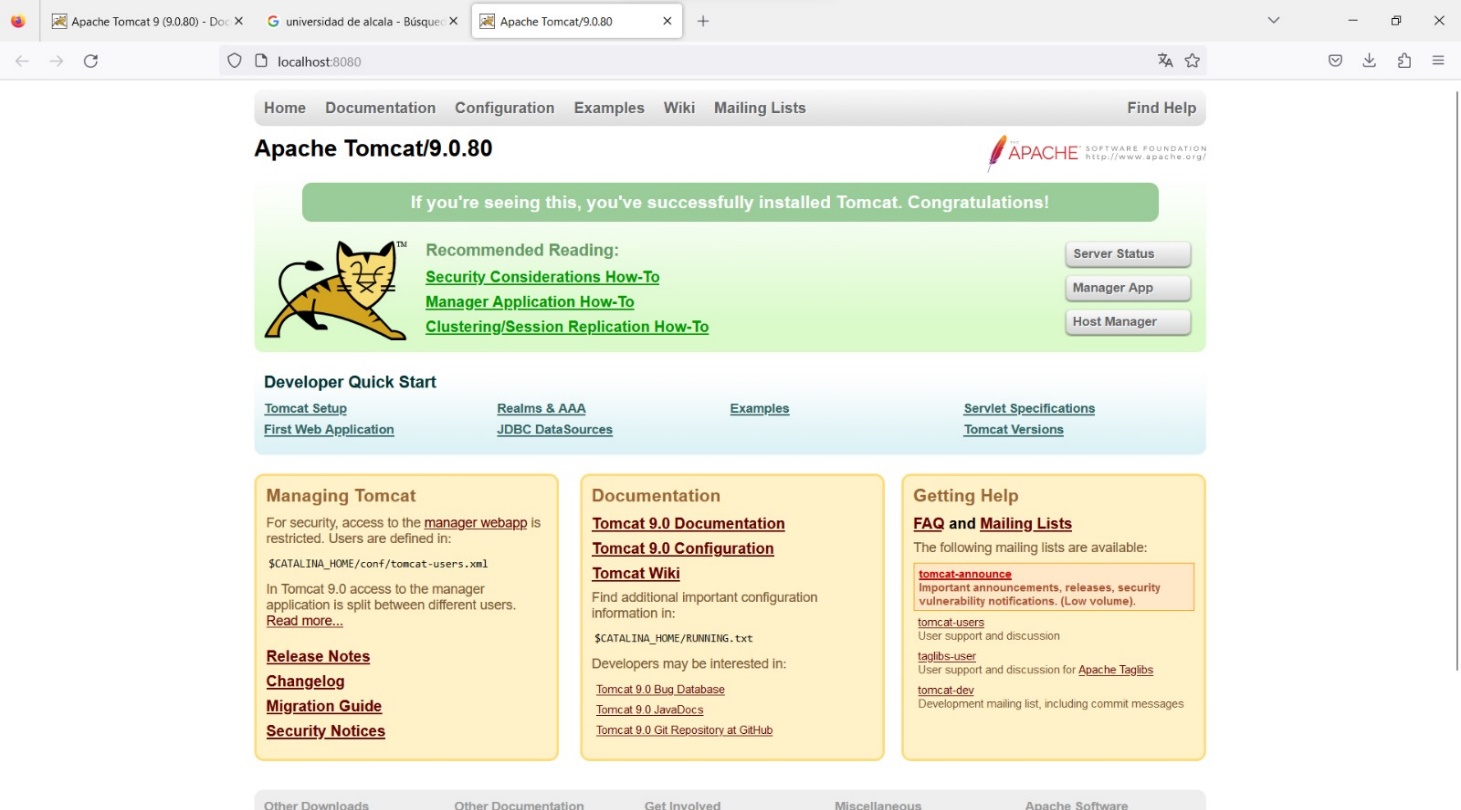
En el caso de que no nos solicite estos permisos lo vamos a tener que configurar nosotros de forma manual, entrando a **Firewall y protección de la red**, dentro del buscador de Windows, y una vez dentro seleccionamos la opción de permitir una aplicación a través de firewall. Y marcamos la opción de privada en este caso en Apache HTTP Server.





Dentro de la carpeta bin tenemos **Tomcat9.exe**, que es desde donde vamos a arrancar el servidor y por otro lado tenemos **Tomcat9w.exe**, que es un panel de configuración de tomcat donde vamos a poder parar y reanudar el servidor y realizar diferentes configuraciones. Una vez arrancado el servidor ya podemos acceder desde un navegador al servidor.

<http://localhost:8080/>



## Configuración Básica

Dentro de la ruta **Apache Software Foundation\Tomcat 9.0\bin\Tomcat9w.exe** tenemos el panel de propiedades donde podemos realizar diferentes funciones. En este caso tenemos la pestaña general que nos permite parar y levantar el servicio del servidor, mediante los botones **Start** y **Stop**. Además, se puede elegir como queremos que arranque el servidor: de manera automática (cada vez que se encienda la máquina) o manual (cuando nosotros queramos).

Captura de pantalla de computadora

Descripción generada automáticamenteCaptura de pantalla de computadora

Descripción generada automáticamente

Además de esta herramienta gráfica, el Apache-Tomcat dispone de una serie de ficheros de configuración ubicados en el directorio **conf/**, donde tenemos el fichero de configuración **server.xml**.

Captura de pantalla de computadora

Descripción generada automáticamente

Donde podemos crear hosts virtuales como veremos más adelante o cambiar el puerto, en este caso como prueba vamos a ver como cambiar del puerto 8080 al 8081.

En primero lugar accedemos al fichero server.xml y dentro buscamos este fragmento de código:

Captura de pantalla de computadora

Descripción generada automáticamente

Y en este caso cambiamos al puerto 8081, arrancamos el servidor y accedemos desde el navegador

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

## Host Virtuales

El fichero donde se configuran los hosts virtuales del servidor Apache Tomcat se encuentra en la ruta **$ conf/server.xml**

En esta parte añadiremos dos elementos service con nombre **Servicio1** y **Servicio2.** En Servicio1 se han añadido dos hosts, uno con el nombre **localhost**, y otro con el nombre **127.0.0.1,** asociados a las carpetas vhost1 y vhost2 respectivamente.

Captura de pantalla de computadora

Descripción generada automáticamente

De igual forma, en Servicio2 se han creado otros dos hosts con los mismos nombres, asociados a las carpetas vhost3 y vhost4.

Captura de pantalla de computadora

Descripción generada automáticamente

Una vez configurados los dos servicios guardamos el fichero y se crean las respectivas carpetas en la ruta **$ Apache Software Foundation\Tomcat 9.0**

Captura de pantalla de computadora

Descripción generada automáticamente

Posteriormente, creamos una carpeta con nombre ROOT en los directorios raíz de cada uno de ellos, y dentro añadiremos los ficheros, como ejemplo vamos a crear un fichero html básico que tiene como contenido el nombre del host

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Una vez todo creado de forma correcta obtenemos acceso a las carpetas y podemos ver su contenido.

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

# Introducción cabeceras HTTP

Para realizar este apartado nos hemos servido de información encontrada en la página web [RFC 2616](https://datatracker.ietf.org/doc/html/rfc2616#section-4.2) y de la información facilitada en la guida de la práctica.

Lo más importante que debemos saber de una cabecera HTTP es que podemos encontrarnos dos tipos distintos: de solicitud (request), de respuesta (response) que siguen un formato genérico similar. Cada campo de encabezado consta de un nombre seguido de dos puntos (":") y un valor de campo.

La estructura de un campo de encabezado se define:

* message-header = field-name ":" [field-value]
* field-name = token
* field-value = \*(field-content | LWS)
* field-content = <los Octetos que componen el valor del campo y consisten en \**TEXTO* o combinaciones de token, separadores y cadena entre comillas>

Sabiendo esto podemos añadir la tabla de cabecera:valor que podemos encontrarnos en mensajes HTTP generados automáticamente por el navegador.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la cabecera** | **Descripción** | **Ejemplo** | **Estado** |
| Accept | Content-Types (tipos de contenido) que se aceptan. | Accept: text/plain | Permanente **\*1** |
| Accept-Charset | Conjunto de caracteres que se aceptan. | Accept-Charset: utf-8 | Permanente |
| Accept-Encoding | Lista de codificaciones que se aceptan. | Accept-Encoding: gzip, deflate | Permanente |
| Accept-Language | Idiomas que se aceptan. | Accept-Language: en-US | Permanente |
| Accept-Datetime | Versión de la hora y fecha que se aceptan. | Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT | Provisional |
| Authorization | Credenciales de autorización. | Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ== | Permanente |
| Caché-Control | Se controla las políticas de caché. | Cache-Control: no-cache | Permanente |
| Connection | Se controla el tipo de conexión. | Connection: keep-alive  Connection: Upgrade | Permanente |
| Cookie | Una cookie enviada previamente por el servidor usando Set-Cookie | Cookie: $Version=1; Skin=new; | Permanente: Estándar |
| Content-Length | El tamaño del contenido de la petición en bytes | Content-Length: 348 | Permanente |
| Content-MD5 | Un checksum en MD5 sobre el contenido | Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ== | Obsoleto |
| Content-Type | El tipo de contenido de la petición en POST o PUT | Content-Type: application/x-www-form-urlencoded | Permanente |
| Date | La fecha y la hora de la petición | Date: Tue, 15 Nov 1994 08:12:31 GMT | Permanente |
| Forwarded | Indica la información original del cliente en caso de conexión por proxy. | Forwarded: for=192.0.2.60;proto=http;by=203.0.113.43 Forwarded: for=192.0.2.43, for=198.51.100.17 | Permanente |
| From | La dirección de correo electrónico de la petición. | From: user@example.com | Permanente |
| Host | El nombre de dominio o dirección IP (puede incluir número de puerto). El uso de la cabecera es obligatorio a partir de HTTP 1.1 | Host: en.wikipedia.org:8080  Host: en.wikipedia.org | Permanente |
| Max-Forwards | Limita el número de veces que un mensaje viaja a través de los proxies. | Max-Forwards: 10 | Permanente |
| Origin | Inicia una petición para servidores con respuesta a Access-Control-Allow-Origin. | Origin: http://www.example-social-network.com | Permanente: Estándar |
| Pragma | Implementa cabeceras en donde múltiples efectos se aplica a todo. | Pragma: no-cache | Permanente |
| Proxy-Authorization | Credenciales de autorización para conectarse a un proxy. | Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ== | Permanente |
| Range | Pide sólo una parte del contenido | Range: bytes=500-999 | Permanente |
| [Referer](https://es.wikipedia.org/wiki/Referer) [[*sic*](https://es.wikipedia.org/wiki/Sic)] | Indica la dirección URL de donde proviene, en otras palabras, es la dirección web del botón Atrás. | Referer: http://en.wikipedia.org/wiki/Main\_Page | Permanente |
| User-Agent | Contiene la información de la petición, como el navegador, el sistema operativo, etc. | User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:12.0) Gecko/20100101 Firefox/21.0 | Permanente |
| [Upgrade](https://es.wikipedia.org/w/index.php?title=Upgrade_header&action=edit&redlink=1) | Pide al servidor que se actualice la versión de HTTP para funcionar. | Upgrade: HTTP/2.0, HTTPS/1.3, IRC/6.9, RTA/x11, websocket | Permanente |
| Warning | Una advertencia general sobre problemas de la entidad. | Warning: 199 Miscellaneous warning | Permanente |

**\*1:** significa que la información contenida en esa cabecera se considera estable y duradera en el tiempo, a menos que se indique lo contrario mediante otras cabeceras o directivas específicas.

# Visualización de Cabeceras HTTP

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para poder entender mejor lo descrito anteriormente vamos a dirigirnos a nuestro navegador, en mi caso voy a usar Firefox porque me parece más fácil de entender que por ejemplo Chrome o Edge.

Lo primero que tenemos que hacer es abrir nuestro navegador y abrir la herramienta de Desarrollador Web desde el navegador o bien mediante la combinación Ctrl + Mayus + I.

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteUna vez hecho esto nos tiene que aparecer algo así en el navegador:

Ahora mismo lo único que podemos ver es ver codigo HTML que emplea la pagina web. Pero nosotros queremos ver las cabeceras HTTP, para ello tenemos que dirigirnos en esa pestaña al botón ***RED o Network.*** Apareciendo en pantalla (Si no saliera nada bastaría con recargar la página pulsando F5):

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Todavía no hemos accedido a las cabeceras de la petición. Con hacer clic sobre una de las peticiones (NO CONFUNDIR CON RESPUESTA) obtenemos algo tal que así:

|  |  |
| --- | --- |
| Procesado: | Sin procesar: |
| Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico  Descripción generada automáticamente |  |

Si lo vemos sin procesar observamos que está escrito como habíamos descrito previamente.

## Ejercicio

Una vez comprobado el acceso a la información de red del protocolo HTTP en ambos navegadores, se propone el acceso a servidores/páginas de internet con formularios o sin formularios para realizar distintas llamadas al servidor: **de tipo GET y de tipo POST** y así poder visualizar sus cabeceras. Se pretende que se capture distintas páginas y se visualice su cabecera http con la petición tipo GET y tipo POST. (Por ejemplo, **tres páginas de cada tipo, como las presentadas en la siguiente tabla aunque pueden ser otras.**).

A continuación, vamos a hacer unos ejercicios para mostrar los contenidos aprendidos. Son ejercicios simples donde realizaremos distintas llamadas al servidor de tipo GET y de tipo POST para así visualizar sus cabeceras.

|  |  |
| --- | --- |
| **GET** | **POST** |
| http://www.abc.es | Correo de la Universidad |
| http://www.cocacola.es | Crunchyroll |
| <https://www.astonmartin.com/en/> | UAH |

**ABC**

Si aplicamos los pasos explicados en el apartado anterior y usando el navegador Firefox, buscamos en el explorador la página web: [www.abc.es](http://www.abc.es) . Obteniendo algo similar a esto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Nos centraremos en las cabeceras de la petición GET. <https://developer.mozilla.org/es/docs/Web/HTTP/Headers/>

GET / HTTP/2

**Host:** [www.abc.es](http://www.abc.es)

* Este parámetro nos indica que el nombre del dominio de la página web es www.abc.es.

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0

* Contiene información de nuestro navegador, el sistema operativo de la maquina donde lo ejecutamos.

**Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8

* Indica los tipos de contenido que se pueden obtener: texto html, aplicación xhtml+xml, aplicación xml con ancho de 0.9, imágenes avif, imágenes webp con un ancho de 0.8.
* Añadir que están organizados como “tipo/subtipo”: es decir el tipo es texto que indica que es contenido legible por un humano y subtipo que indica que el texto es html.

**Accept-Language**: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3

* Idiomas que se aceptan: español e inglés con ancho 0.5 y 0.3 respectivamente.

**Accept-Encoding**: gzip, deflate, br

**Connection:** keep-alive

**Upgrade-Insecure-Requests:** 1

**Sec-Fetch-Dest:** document

**Sec-Fetch-Mode**: navigate

**Sec-Fetch-Site:** none

**Sec-Fetch-User:** ?1