

# Documentación Apache2

---

Documentado por Andrés Ruslan Abadías Otaí | [Nisamov](#)

## Instalación

Para evitar posibles problemas durante la instalación, se recomienda actualizar los paquetes del sistema.

```
# Esto es unicamente opcional
sudo apt update && sudo apt upgrade -y
```

Instalación de apache2: Instalamos el servicio con los privilegios de administrador.

```
sudo apt install apache2
```

## Creación de la página

Para crear una página hay que dirigirse a la dirección `/var/www/`, donde veremos un fichero llamado `index.html`. Para crear la primera página, creamos un directorio dentro de la ubicación anterior con el nombre del dominio (por organización) que queramos, este tiene que acabar en un prefijo `.es`, `.org`, `.com`... Posteriormente accedemos al interior del directorio creado y ahí dentro creamos un fichero llamado `index.html`.

```
#Nos dirigimos a la ubicación mencionada
cd /var/www/
# - Ubicación actual: /var/www

#Creamos el directorio con el prefijo
mkdir mipagina.es
# - Ubicación actual: /var/www

#Accedemos al directorio
cd mipagina.es
# - Ubicación actual: /var/www/mipagina.es

#Creamos el fichero dentro del directorio creado
touch index.html
# - Ubicación actual: /var/www/mipagina.es
```

## Edición de la página

Para dar estructura a la página, hay que editar el fichero `index.html` creado anteriormente, para ello es necesario tener los conocimientos básicos de HTML5. Si se quiere agregar diseño y funcionalidad, es

requerido usar ficheros `.css` y `.js`.

```
#Editamos el fichero index.html
sudo nano index.html
# - Ubicación actual: /var/www/mipagina.es/index.html
```

Dentro del fichero `index.html` agregamos la estructura básica de HTML5:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MiPagina</title>
  <!--Elimina la siguiente linea si no quieres aplicar estilos-->
  <link href="styles/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <!--Agrega el contenido que veas necesario-->
  <p>Este es el contenido de tu pagina</p>
  <!--Elimina la siguiente linea si no quieres aplicar estilos-->
  <script src="scripts/script.js"></script>
</body>
</html>
```

En el código anterior vemos comentarios con la opción de dejar líneas que permiten agregar código CSS y JavaScript, esto es completamente opcional, se recomienda hacer eso lo último. Para poner un ejemplo de cómo sería la estructura con la implementación de estilos y scripts tenemos que tener en cuenta que estos ficheros se deberían agregar en subdirectorios dentro de la ruta de la página principal. Esto permitirá un acceso rápido, así como una buena organización, en el siguiente ejemplo se presenta la estructura ideal para este proceso.

## Estructura de directorio

```
var
├── www
│   └── mipagina.es
│       ├── index.html
│       ├── styles
│       │   └── style.css
│       ├── scripts
│       │   └── script.js
```

Para crear esta misma estructura hay que estar dentro de la ruta de la página principal `mipagina.es` y crear dos directorios, el primero sería `styles`, donde almacenaremos los ficheros de estilos `.css` y el segundo `scripts`, donde almacenaremos ficheros `.js`.

Para llevar a cabo este proceso opcional, hay que hacer lo siguiente:

```
#Nos posicionamos dentro de la ruta de la pagina
cd /var/www/mipagina.es
# - Ubicación actual: /var/www/mipagina.es

#Creamos dos directorios
mkdir styles
mkdir scripts
# - Ubicación actual: /var/www/mipagina.es
```

## Estilo y Funcionamiento de la pagina con CSS y JS

Tras la creación de los directorios, deberemos llenarlos con el código necesario, este estará almacenado dentro de ficheros, los cuales son llamados por la pagina principal `index.html`. Para crear estos ficheros es necesario acceder a cada directorio y crear un fichero con su correspondiente contenido:

```
#Nos posicionamos dentro de la ruta JavaScript (scripts)
cd /var/www/mipagina.es/scripts
# - Ubicación actual: /var/www/mipagina.es/scripts

#Creamos el primer fichero de scripts con touch, esto nos abrirá una ventana para
introducir texto
sudo touch script.js
# - Ubicación actual: /var/www/mipagina.es/scripts/script.js
```

Tras crear el fichero dentro de la ruta especificada, nos abrirá una ventana donde ingresar texto plano, en esta ventana ingresaremos código JavaScript, el cual estará vinculado a la pagina principal (debido al nombre y ruta usados, estos mismos están referenciados en la pagina web, por lo que si se quiere, se puede cambiar la ruta de la pagina, no obstante, para su correcto funcionamiento es recomendable usar una ruta clara y fija).

El código del fichero JavaScript puede ser el que se desee, en este caso se usará un código simple como muestra de correcto funcionamiento:

```
//Alerta que muestra el correcto funcionamiento del código
alert("El Código JavaScript Funciona correctamente");
//Log en consola indicando el correcto funcionamiento del mismo
console.log("Correcto funcionamiento");
```

Para poder aplicar estilos en la pagina, es necesario usar el mismo sistema usado durante la creación de scripts de `JavaScript`, para ello iremos a la ruta de los estilos creada anteriormente y crearemos un fichero donde almacenaremos el código de los estilos de la pagina principal, este fichero deberá contener el nombre referenciado en la pagina web, el cual es `style.css`, para que pueda funcionar, así como el código `.js` deberá estar en la misma ruta que la mencionada en la pagina web (`/var/www/mipagina.es/styles`).

```
#Nos posicionamos en la ruta de los estilos (styles)
cd /var/www/mipagina.es/styles
# - Ubicación actual: /var/www/mipagina.es/styles

#Creamos un fichero el cual almacenara los estilos de la pagina, esto nos abrira
una ventana para introducir texto, de la misma manera que sucedio con la creacion
del fichero de scripts de JavaScript
sudo touch style.css
# - Ubicación actual: /var/www/mipagina.es/scripts/style.css
```

Posteriormente agregamos contenido dentro del fichero, este ha de ser código **CSS**, esto permitira a la pagina obtener un estilo, bien sea un color de fondo, efectos, fuentes de texto...

Un ejemplo sencillo de código **CSS** es el siguiente:

```
/*Color de fondo en el cuerpo de la pagina*/
body {
    background-color: #fefbd8;
}

/*Color de fondo en los titulos h1*/
h1 {
    background-color: #80ced6;
}

/*Color de fondo en los divs*/
div {
    background-color: #d5f4e6;
}

/*Estilos cuerpo de texto*/
p {
    font-family: 'Open Sans';
    font-size: 14px;
    color: #ccc;
    line-height: 18px;
    margin-bottom: 20px;
}
```

Con esto hecho, tenemos la pagina web montada enlazada con los scripts y diseño, a continuación deberemos crear un certificado, para ello instalaremos **openssl**:

```
sudo apt install openssl
```

A continuación habilitaremos el modo **ssl**:

```
sudo a2enmod ssl
```

Finalmente reiniciaremos el servicio apache2:

```
sudo systemctl restart apache2
```

Con este comando habremos instalado **openssl** y habilitado el modo **ssl**, permitiéndonos continuar con la configuración y creación de certificados.

[!] En caso de usar VirtualBox, pondremos la máquina en adaptador puente.

Tras este previo paso hecho, accederemos a la localización de directorios y ficheros de configuración, para ello accederemos a la siguiente ubicación y copiaremos el fichero **000-default.conf**.

```
#Accedemos a la ruta donde se almacena la configuración de las páginas dentro de
apache
cd /etc/apache2/sites-available
# - Ubicación actual: /etc/apache2/sites-available

#Clonamos el fichero 000-default.conf y lo renombramos con el nombre de nuestra
pagina, seguido de un ".conf"
sudo cp 000-default.conf mipagina.es.conf
# - Ubicación actual: /etc/apache2/sites-available
```

Tras hacer una clonación, accedemos al interior de la copia creada por nosotros con el nombre de nuestra pagina, a la cual le aplicaremos atributos para hacer que la pagina funcione.

Para esto accedemos con permisos de root y agregamos el siguiente contenido sustituyendo todo lo que pueda haber en su interior:

```
#Editamos el contenido como root
sudo nano mipagina.es.conf
# - Ubicación actual: /etc/apache2/sites-available/mipagina.es.conf
```

Dentro del fichero agregamos el siguiente contenido:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/mipagina.es
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog /etc/apache2/sites-available/access.log
ServerName www.mipagina.es
</VirtualHost>
```

```
<VirtualHost *:443>
ServerName mipagina.es
Redirect / http://www.mipagina.es
</VirtualHost>

# Contenido para certificado SSL
<VirtualHost *:443>
ServerName www.mipagina.es
DocumentRoot /var/www/mipagina.es
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog /etc/apache2/sites-available/access.log
SSLEngine on
SSLCertificateFile /etc/apache2/certificate/apache-certificate/apache-
certificate.crt
SSLCertificateKeyFile /etc/apache2/certificate/apache.key
</VirtualHost>
```

Esto permitira enlazar la configuracion y redireccion mediante http a nuestra pagina creada previamente, en caso de haber utilizado un nombre diferente, habria que sutituir cada valor que hace referencia a **mipagina.es** con el nombre que sea requerido.

Para aplicar los cambios hay que usar el siguiente comando agregandole el nombre de nuestra página:

```
#Activamos nuestra carpeta, agregando a2ensite + nombreDeCarpeta
sudo a2ensite mipagina.es
# - Ubicación actual: /etc/apache2/sites-available
```

Tras seguir todos los pasos previos, hayq eu asignar una direccion a la pagina, para ello editaremos el fichero **/etc/hosts**:

```
#Editamos el fichero /etc/hosts
sudo nano /etc/hosts
# - Ubicación actual: /etc/hosts
```

En el interior de este asignaremos una ip al fichero index de la pagina. La direccion 127.0.1.1 está asociada a la propia maquina, mientras que la ip **40.0.0.2** es una direccion asociada a **mipagina.es**, es una direccion pública.

```
127.0.0.1      localhost
127.0.1.1      theritex
40.0.0.2       mipagina.es      www.mipagina.es
# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

Prosiguiendo con la configuración, procedemos a la configuración de los puertos dentro de `/etc/apache2/ports.conf`, para ello accedemos como root:

```
#Editamos el fichero de configuracion de puertos de apache2
sudo nano /etc/apache2/ports.conf
# - Ubicación actual: /etc/hosts
```

En el interior agregamos el siguiente contenido, el cual permite la escucha por los puertos 80 y 443:

```
#Hacemos que escuche el puerto 80
Listen 80
#Configuracion adicional para el certificado ssl con escucha por el puerto 443
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Habiendo guardado la configuración anterior, procedemos a reiniciar el servicio apache2:

```
sudo systemctl restart apache2
```

Tras estos pasos previos, accedemos a la configuración de apache2 agregando el siguiente contenido en las últimas líneas de la configuración, permitiendo sobrescribir las directivas anteriores, siendo este un paso necesario para el HTTPs y su funcionamiento:

```
sudo nano /etc/apache2/apache2.conf
```

```
<Directory /var/www/mipagina.es>
AllowOverride All
</Directory>
```

Siguiendo con el procedimiento, creamos varios directorios para almacenar los certificados y llaves para la pagina, situandonos previamente en `/etc/apache2/sites-available`:

```
#Accedemos a la ruta donde almacenaremos los certificados
cd /etc/apache2/sites-available
```

```
#Creamos el primer directorio de almacenamiento, con el nombre "certificado"
mkdir certificado
#Accedemos al interior del directorio creado previamente
cd certificado
#Creamos el certificado
#En el siguiente comando creamos un certificado con el nombre "apache-
certificado.crt"
#Asi mismo creamos una llave llamada "apache.key", los cuales usaremos en la
configuración previa, dentro de /etc/apache2/sites-available/mipagina.es.conf
sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-
certificado.crt -keyout apache.key
```

Tras la creación del certificado y la llave solicitará el ingreso de datos, siendo el único requerido **Common name**, donde deberemos escribir la dirección ip **fija**. Estas llaves y certificados son necesarios dentro de la configuración previa `/etc/apache2/sites-available/mipagina.es.conf`, haciendo referencia a estos mismos, los cuales usará para la conexión HTTPS.

Como últimos pasos hay que reiniciar el servicio apache2:

```
sudo service apache2 restart
```

Tras esto, si se intenta acceder a la página, nos mostrará un aviso "Advertencia: Riesgo potencial de seguridad a continuación", para evitar un aviso similar, accederemos a la configuración de nuestra página:

```
sudo nano /etc/apache2/sites-available/mipagina.es.conf
```

En esta ubicación agregaremos la siguiente línea: **Redirect permanent / http://www.mipagina.es**, redirigiendo todos los accesos http, a una conexión segura mediante https:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/mipagina.es
Redirect permanent / http://www.mipagina.es
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog /etc/apache2/sites-available/access.log
ServerName www.mipagina.es
</VirtualHost>

<VirtualHost *:443>
ServerName mipagina.es
Redirect / http://www.mipagina.es
</VirtualHost>

<VirtualHost *:443>
```



```
ServerName www.mipagina.es
DocumentRoot /var/www/mipagina.es
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog /etc/apache2/sites-available/access.log
SSLEngine on
SSLCertificateFile /etc/apache2/certificate/apache-certificate/apache-
certificate.crt
SSLCertificateKeyFile /etc/apache2/certificate/apache.key
</VirtualHost>
```

Finalmente hay que reiniciar el servicio apache2 una última vez:

```
sudo service apache2 restart
```

Si se ha seguido el procedimiento indicado, la página debería ser funcional, durante la aplicación de cambios y los reinicios de apache2 pueden surgir errores, los cuales son detallados durante la misma ejecución.