

Instalación de Apache2

HTTP y HTTPS

apachectl

Host Virtual

Índice

Apache2.....	2
Instalación.....	2
Configuración básica.....	3
Tráfico cifrado y no cifrado.....	3
Comandos que ayudan.....	4
apachectl status.....	4
apachectl restart.....	5
apachectl graceful.....	5
apachectl configtest.....	5
Crear un host virtual.....	6
HTTP.....	6
HTTPS.....	6
Activación.....	7
Página de ejemplo.....	7
Fuentes.....	9
Complementarias.....	9

Apache2

En esta práctica vamos a documentar cómo instalar y configurar un **servidor HTTP**. Específicamente, vamos a utilizar Apache2.

Instalación

Para instalar **Apache2**, podemos utilizar el **administrador de paquetes** de nuestro sistema.

- **Debian / Ubuntu**¹

```
sudo apt install apache2
```

- **CentOS8 / RHEL8 / Fedora**²

```
sudo dnf install HTTPd
```

- **FreeBSD**³

```
sudo pkg install apache24
```

- **Alpine Linux**⁴

```
sudo apk add apache2
```

- **Windows**⁵

Por favor, vaya a la referencia oficial acerca de [cómo instalar Apache2 en Windows](#).



Esta práctica está elaborada bajo el sistema operativo **Ubuntu 22.04 LTS**. Por ello se asume que será responsabilidad del lector adaptar las utilidades y rutas empleadas al sistema propio.



Esta práctica utiliza **terminología técnica especializada**. Por ello, es recomendable consultar primero el siguiente **documento preliminar**. Se asume la lectura de dicho documento.

[GS-Práctica 2.1-Apache2-Stoyanov_Angel_Krasimirov.pdf](#)

Configuración básica

Vamos a ver ciertas configuraciones que debemos realizar conforme esta práctica.

Tráfico cifrado y no cifrado

Primero, debemos asegurar que Apache2 es capaz de manejar **tráfico** tanto **cifrado** como **no cifrado**. Esto hace referencia a los protocolos **HTTP** y **HTTPS**.

Primero, vamos a asegurarnos que Apache2 está utilizando los puertos **80** para el tráfico no cifrado, y el **8080** para el cifrado, mediante el módulo **ssl_module**.⁶

```
root@vmi2144575: /etc/apache2
Listen 80

<IfModule ssl_module>
    Listen 8080
</IfModule>
-- INSERT --
```

Para **permitir este tráfico**, debemos admitirlo en nuestro **cortafuegos**. En este caso, utilizamos **UFW**, por lo que podemos ejecutar los siguientes comandos.

```
sudo ufw allow 80
sudo ufw allow 8080
```

Es importante asegurarse de **activar los módulos** necesarios para el uso de **HTTPS**. En este caso, **ssl** para poder soportar **HTTPS**, y **rewrite** para poder redirigir de **HTTP** a **HTTPS**

```
sudo a2enmod ssl
sudo a2enmod rewrite
```

Para aplicar los cambios, podemos reiniciar Apache2 mediante **apachectl**.

```
sudo apachectl restart
```

Podemos comprobar que esto funciona con el comando **netstat -putln**

```
sudo netstat -putln
```

```
root@vmi2144575: /etc/apache2/sites-available
root@vmi2144575:/etc/apache2/sites-available# netstat -putln | grep apache2
tcp        0      0 0.0.0.0:8080        0.0.0.0:*        LISTEN    520/apache2
tcp        0      0 0.0.0.0:80         0.0.0.0:*        LISTEN    520/apache2
root@vmi2144575:/etc/apache2/sites-available#
```


`apachectl restart`

Este comando reinicia el servidor **HTTP**, lo cual permite **aplicar cambios** en las configuraciones, o simplemente iniciar nuevamente el servidor, en caso de encontrarse en un estado de error.

```
root@vmi2144575: /etc/apache2/sites-available
root@vmi2144575:/etc/apache2/sites-available# apachectl restart
root@vmi2144575:/etc/apache2/sites-available#
```

`apachectl graceful`

Parecido al anterior, solo que reinicia el servidor **HTTP** de forma más controlada, **gradual**.

```
root@vmi2144575: /etc/apache2/sites-available
root@vmi2144575:/etc/apache2/sites-available# apachectl graceful
root@vmi2144575:/etc/apache2/sites-available#
```

`apachectl configtest`

Comprueba la sintaxis de la configuración antes de ser aplicada. Nos ayuda a evitar errores antes de tiempo.

```
root@vmi2144575: /etc/apache2/sites-available
root@vmi2144575:/etc/apache2/sites-available# apachectl configtest
Syntax OK
root@vmi2144575:/etc/apache2/sites-available#
```

Crear un host virtual

Vamos a configurar y personalizar el **host virtual** de Apache2.

HTTP

La configuración que queremos lograr forzará las conexiones a utilizar **HTTPS**, puesto que **HTTP** es inseguro. Para ello, configuramos el archivo **000-default.conf**.

Tenemos como administrador **webmaster@dpl2daw.es**. También, el directorio raíz del host virtual será **public**, ubicado en **/var/www/html**. Las siguientes líneas definen la redirección a **HTTPS**.

```
root@vmi2144575: /etc/apache2/sites-available
<VirtualHost *:80>
    ServerAdmin webmaster@dpl2daw.es
    DocumentRoot /var/www/html/public

    RewriteEngine on
    RewriteCond %{SERVER_NAME} =dpl2daw.es
    RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
-- INSERT --
```

HTTPS

En el archivo de configuración **default-ssl.conf** o también, **000-default-le-ssl.conf**.

Al igual que en **HTTP**, configuramos el **administrador** y el **directorio raíz**. También, definimos el **nombre del host virtual**, que será el nombre de dominio **dpl2daw.es**. En las siguientes líneas, incluimos los **certificados SSL** necesarios para tener una correcta y segura conexión **HTTPS**. Nosotros hemos utilizado la utilidad de **letsencrypt** para generar estos certificados.⁸

```
root@vmi2144575: /etc/apache2/sites-available
<IfModule mod_ssl.c>
<VirtualHost *:8080>

    ServerAdmin webmaster@dpl2daw.es
    DocumentRoot /var/www/html/public

    ServerName dpl2daw.es
    SSLCertificateFile /etc/letsencrypt/live/dpl2daw.es/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/dpl2daw.es/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
</IfModule>
-- INSERT --
```



Para poder generar **certificados SSL** reales y **válidos**, es necesario **poseer legalmente el dominio** en cuestión. En este caso, nosotros **no lo poseemos**, sino que es una recreación educativa.

Activación

Aunque normalmente están activas por defecto, hay que activar los hosts virtuales que hemos creado.

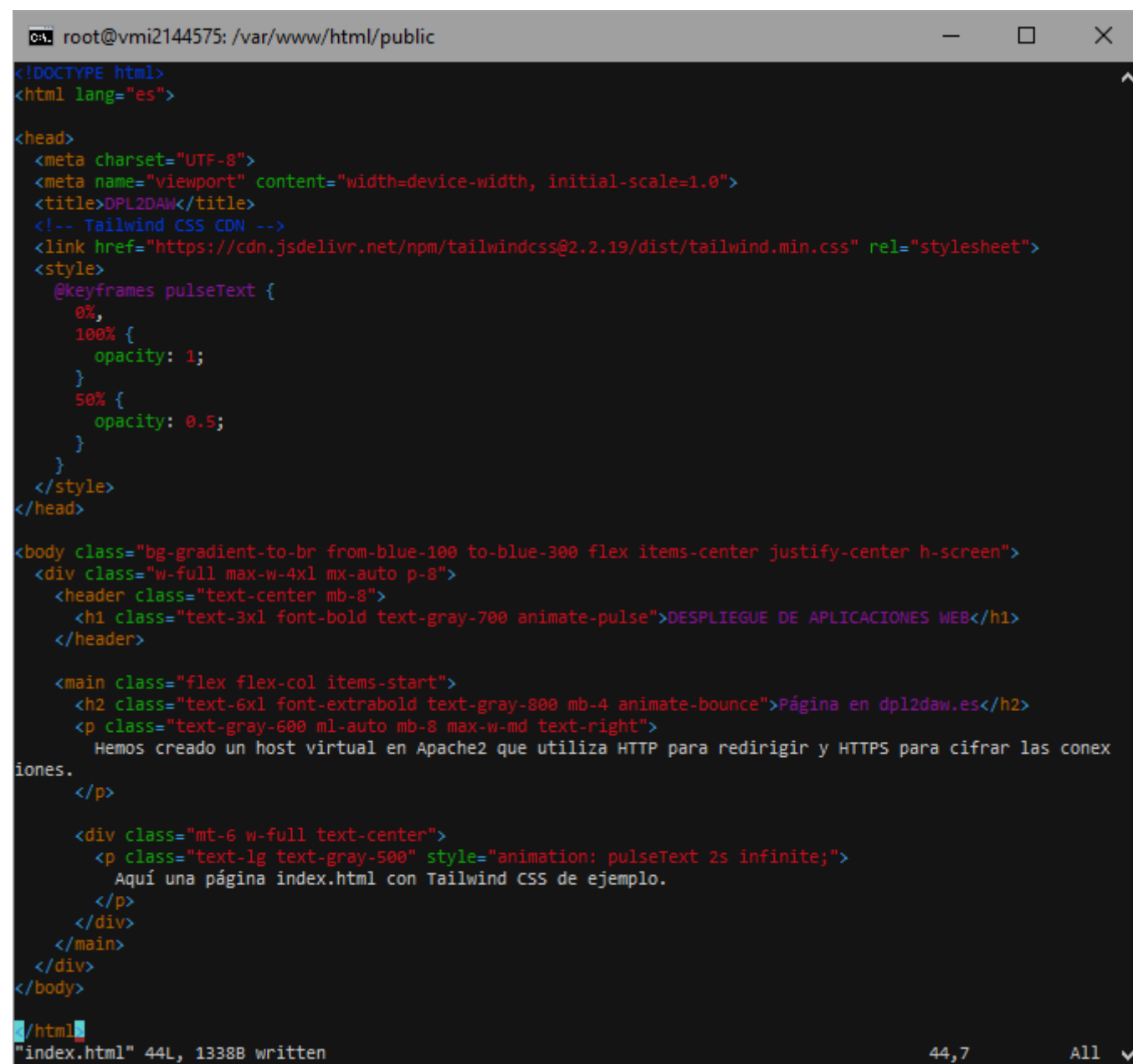
```
sudo a2ensite 000-default.conf
sudo a2ensite default-ssl.conf
```

Recordemos reiniciar el servicio si queremos ver los cambios aplicados.

```
sudo apachectl restart
```

Página de ejemplo

Una vez tenemos configurado el servidor, podemos crear una página `index.html` de ejemplo para mostrar en nuestro host virtual. Para ello, vamos al directorio `/var/www/html/public`.



```
root@vmi2144575: /var/www/html/public
<!DOCTYPE html>
<html lang="es">

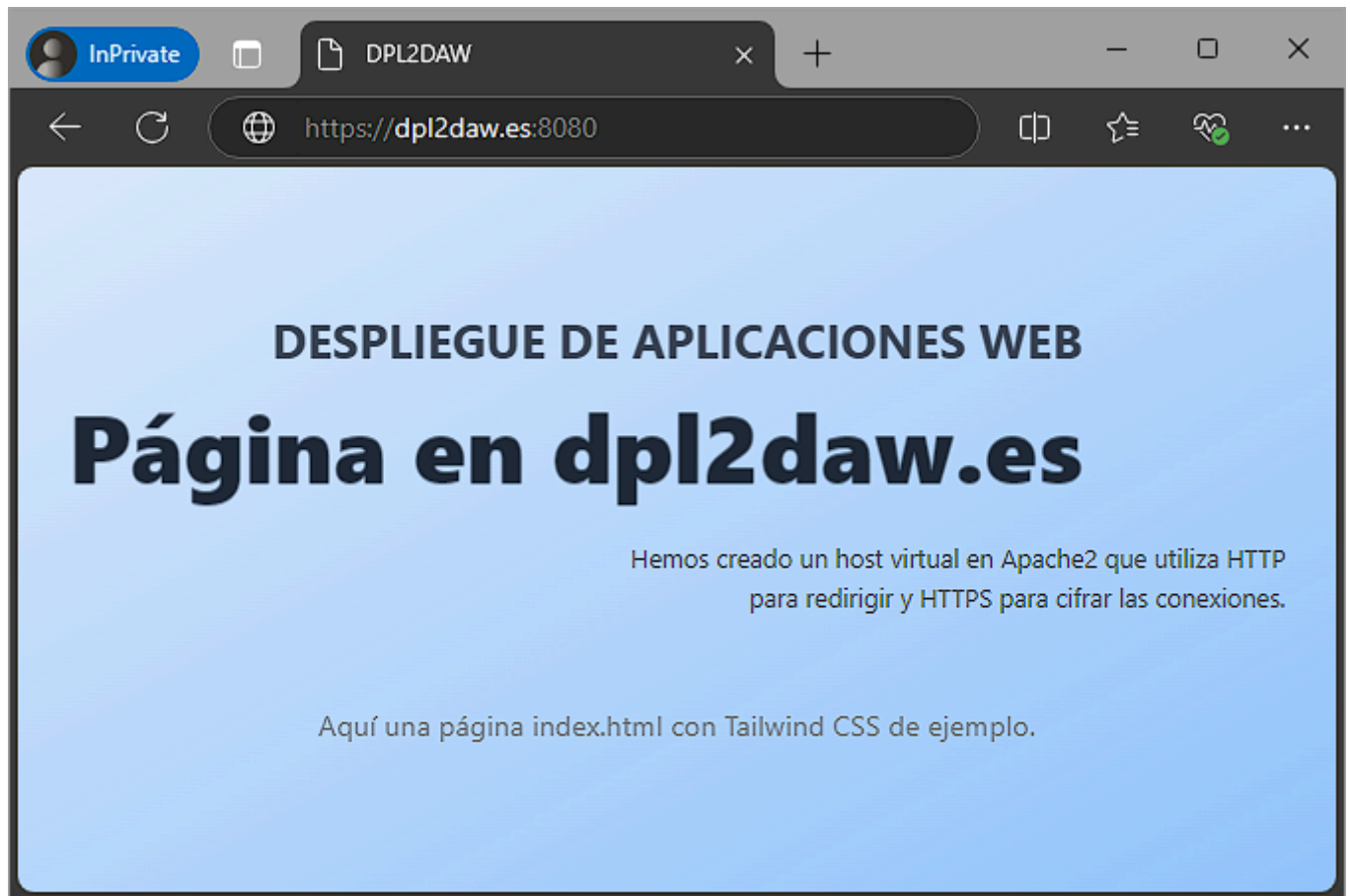
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DPL2DAW</title>
  <!-- Tailwind CSS CDN -->
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
  <style>
    @keyframes pulseText {
      0%,
      100% {
        opacity: 1;
      }
      50% {
        opacity: 0.5;
      }
    }
  </style>
</head>

<body class="bg-gradient-to-br from-blue-100 to-blue-300 flex items-center justify-center h-screen">
  <div class="w-full max-w-4xl mx-auto p-8">
    <header class="text-center mb-8">
      <h1 class="text-3xl font-bold text-gray-700 animate-pulse">DESPLIEGUE DE APLICACIONES WEB</h1>
    </header>

    <main class="flex flex-col items-start">
      <h2 class="text-6xl font-extrabold text-gray-800 mb-4 animate-bounce">Página en dpl2daw.es</h2>
      <p class="text-gray-600 ml-auto mb-8 max-w-md text-right">
        Hemos creado un host virtual en Apache2 que utiliza HTTP para redirigir y HTTPS para cifrar las conex
iones.
      </p>

      <div class="mt-6 w-full text-center">
        <p class="text-lg text-gray-500 style="animation: pulseText 2s infinite;">
          Aquí una página index.html con Tailwind CSS de ejemplo.
        </p>
      </div>
    </main>
  </div>
</body>

/html
"index.html" 44L, 1338B written 44,7 All
```



Como podemos ver, la página `index.html` se **carga por defecto**, pues Apache2, de forma predeterminada, busca los archivos `index.html` o `index.php` para cargar.

Fuentes

1. [Instalación y configuración Apache2 en Ubuntu](#) (en inglés). Guía oficial.
2. [Instalación y configuración Apache2 en CentOS8 / RHEL8 / Fedora](#) (en inglés).
3. [Instalación y configuración Apache2 en FreeBSD](#) (en inglés).
4. [Instalación Apache2 en Alpine](#) (en inglés).
5. [Instalación Apache2 en Windows](#) (en inglés). Documentación oficial.
6. [Apache2 configuración de puertos](#) (en inglés).
7. [Instalación de Lynx en Ubuntu](#) (en inglés).
8. [Apache2 y letsencrypt](#) (en inglés). Documentación de como trabajar con Apache2 usando letsencrypt.

Complementarias

Fuentes complementarias usadas para contrastar información, pero no incluidas en la elaboración de este documento.

- [GS-Práctica 2.1-Apache2-Stoyanov Angel Krasimirov.pdf](#). Documento preliminar.