

# **Configuración Avanzada de Apache**

Antonio Espín Herranz

# Contenidos

- Configuración de **hosts virtuales** y **múltiples sitios** en Apache.
- Ajustes de rendimiento con **KeepAlive**, **Timeout**, **MaxClients**.
- Administración de **logs** y monitoreo de **rendimiento**.

# Hosts Virtuales

- Host Virtuales son una funcionalidad que permite que un solo servidor web (una sola máquina con Apache instalado) **sirva múltiples sitios web diferentes.**
- Tendremos varios sitios independientes corriendo en el mismo servidor, cada uno con su propio dominio, configuración y contenido.

# Host Virtuales

- Pueden ser basados en nombre o en dirección IP
- **Name-based Virtual Hosts:** Se basan en el nombre del dominio: [www.ejemplo.es](http://www.ejemplo.es). Pueden estar en la misma dirección IP. Apache los distingue por el nombre del host que se envía la solicitud.
- **IP-based Virtual Hosts:** Se basan en direcciones IP. Cada sitio tiene su propia IP, apache responde según la dirección IP solicitada.

# Configuración

- Primero tenemos que ir al fichero de configuración de Apache:
- C:\Apache24\conf\**httpd.conf**
  - Localizar la línea:
  - **# Virtual hosts**
  - **#Include conf/extra/httpd-vhosts.conf**
- Descomentar y reiniciar.
- Ir al fichero (añadir un virtual host por cada proyecto)
- **C:\Apache24\conf\extra\httpd-vhosts.conf**

# Múltiples sitios en Apache

```
<VirtualHost *:80>  
    ServerName proyecto1.local  
    DocumentRoot "C:/websites/proyecto1"  
    <Directory "C:/websites/proyecto1">  
        AllowOverride All  
        Require all granted  
    </Directory>  
</VirtualHost>
```

```
<VirtualHost *:80>  
    ServerName proyecto2.local  
    DocumentRoot "C:/websites/proyecto2"  
    <Directory "C:/websites/proyecto2">  
        AllowOverride All  
        Require all granted  
    </Directory>  
</VirtualHost>
```

# Múltiples sitios en Apache

- Abrir como administrador el fichero (con el block de notas):
  - Seleccionar “todos los archivos”
  - **C:\Windows\System32\drivers\etc\hosts**
- Añadir:
  - ***127.0.0.1 proyecto1.local***
  - ***127.0.0.1 proyecto2.local***
- Tantas entradas como proyectos tengamos.

# Múltiples sitios en Apache

- Reiniciar Apache
- Crear un fichero index.php en cada carpeta de los dos sitios:
  - C:/websites/proyecto1/index.php → <?php phpinfo(); ?>
  - C:/websites/proyecto2/index.php → <?php phpinfo(); ?>
- Desde un navegador en dos pestañas distintas:
  - <http://proyecto1.local>
  - <http://proyecto2.local>



# Con páginas de error personalizadas

```
<VirtualHost *:80>
  ServerName www.sitio1.local
  DocumentRoot "C:/Apache24/htdocs/sitio1"

  <Directory "C:/Apache24/htdocs/sitio1">
    AllowOverride All
    Require all granted
  </Directory>

  ErrorDocument 404 /error/404_sitio1.html
  ErrorDocument 500 /error/500_sitio1.html
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerName www.sitio2.local
  DocumentRoot "C:/Apache24/htdocs/sitio2"

  <Directory "C:/Apache24/htdocs/sitio2">
    AllowOverride All
    Require all granted
  </Directory>

  ErrorDocument 404 /error/404_sitio2.html
  ErrorDocument 500 /error/500_sitio2.html
</VirtualHost>
```

# Con páginas de error personalizadas

```
C:/Apache24/htdocs/  
├── sitio1/  
│   └── error/  
│       ├── 404_sitio1.html  
│       └── 500_sitio1.html  
├── sitio2/  
│   └── error/  
│       ├── 404_sitio2.html  
│       └── 500_sitio2.html  
└──
```

# Múltiples sitios en Apache

- OJO que el orden en que se colocan los virtualhosts puede ser determinante.
- Si tenemos APIs de servicios (por ejemplo: con Slim), podemos tener problemas a la hora de que nuestro sitio responda.
- En este caso, podemos colocarlo el primero.

# Ajustes de rendimiento

# KeepAlive

# Timeout

# MaxClients