

TEMA 1

IMPLANTACIÓN DE ARQUITECTURAS WEB

Despliegue de Aplicaciones Web +



The background is a solid blue color. It features several stylized white clouds of various shapes and sizes. Interspersed among the clouds are white plus signs (+) and small white circles (o).

¿Arquitectura web?

Definición

Definición de arquitectura web

La arquitectura web, también conocida como arquitectura de la información, es la manera en como se estructura toda la información de una página web para que sea fácilmente rastreable, además de ser amigable para los usuarios.



Nota: Cuanta mayor importancia, relevancia y tamaño tenga una página web o sitio web, mayor es la importancia que se le debe dar a la arquitectura web.

Definición de arquitectura web

Cuando se tiene una buena arquitectura web, se cumplen o consiguen dos objetivos fundamentales:

- Agradar a los usuarios ✓
- También agrada a las máquinas ✓

Nota: La ordenación de los contenidos + la interacción entre las webs = localizable y destacable



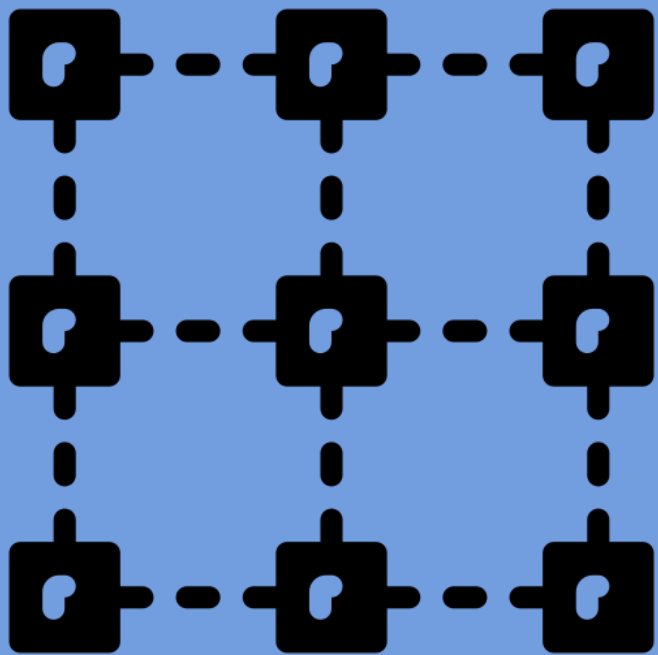
Definición de arquitectura web

Requerimientos para implantar un servicio web:

- Hardware
- Software
- Dependencias

Nota: Depende del contexto y del target de nuestro servicio





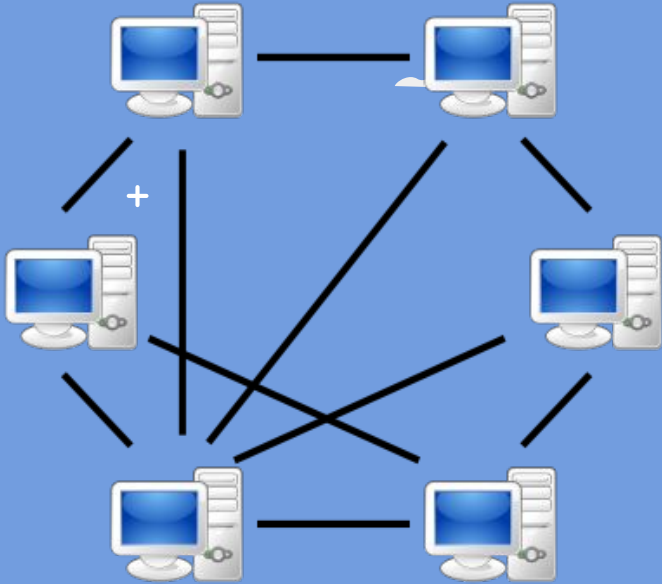
Modelos de arquitecturas web

Relación entre los distintos elementos que componen el funcionamiento de páginas y aplicaciones web. Y existen de varios tipos:

- Modelo Punto a Punto
- Modelo Cliente-Servidor
- Modelo con servidor de aplicaciones
- Modelo con varios servidores de aplicaciones

Modelo Punto a Punto

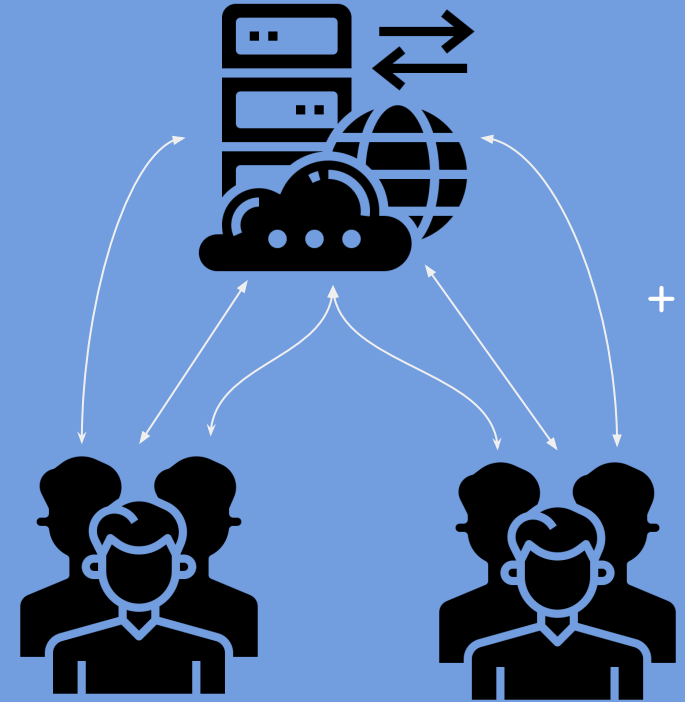
- Compuesta por distintos nodos, no solo uno (descentralizada)
- El peso de la red recae sobre todos esos nodos (distribuida)
- Reparto de recursos
- ¿Ejemplos...?



Modelo Cliente-Servidor

+

- Compuesta por proveedores (Servidores) de recursos o servicios y los solicitantes (Clientes) del servicio/recurso
- El peso de la red recae sobre el proveedor (centralizado)
- No tiene por qué ser una sola máquina ni un solo programa



Modelo Cliente-Servidor



Funcionamiento básico

1. Desde el navegador web (cliente) el usuario solicita la carga de una página web indicando su URL.
2. El servidor recibe la petición de la página web.
3. Busca en su sistema de almacenamiento la página solicitada.
4. Envía el contenido de la página web (código fuente) por el mismo medio por el que recibió la petición.
5. El navegador web recibe el código fuente de la página y lo interpreta mostrando al usuario la página web.



Nota: El servidor web envía al cliente el recurso sin hacer ningún tratamiento en él

Modelo con Servidor de Aplicaciones

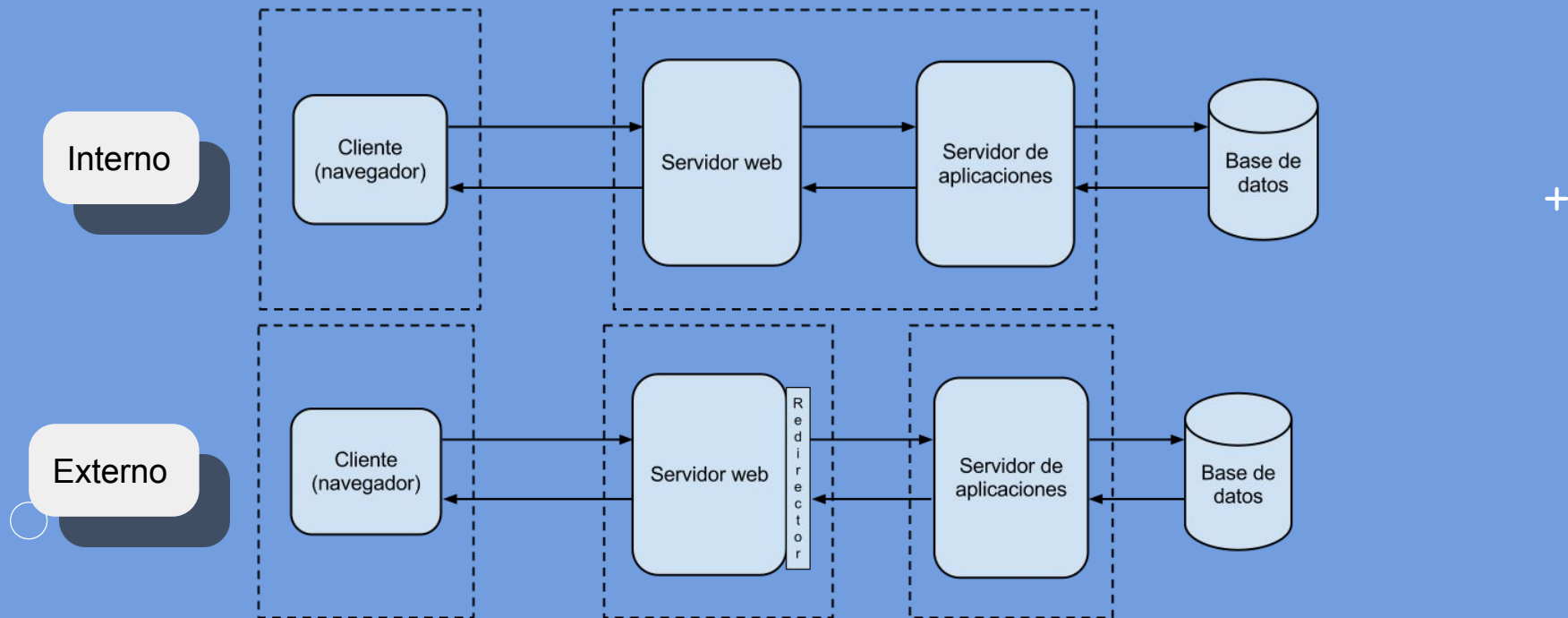
+

- Recursos dinámicos
- El servidor web trabaja como intermediario
- El servidor de aplicaciones accede a los recursos que sean necesarios para servir el recurso dinámico

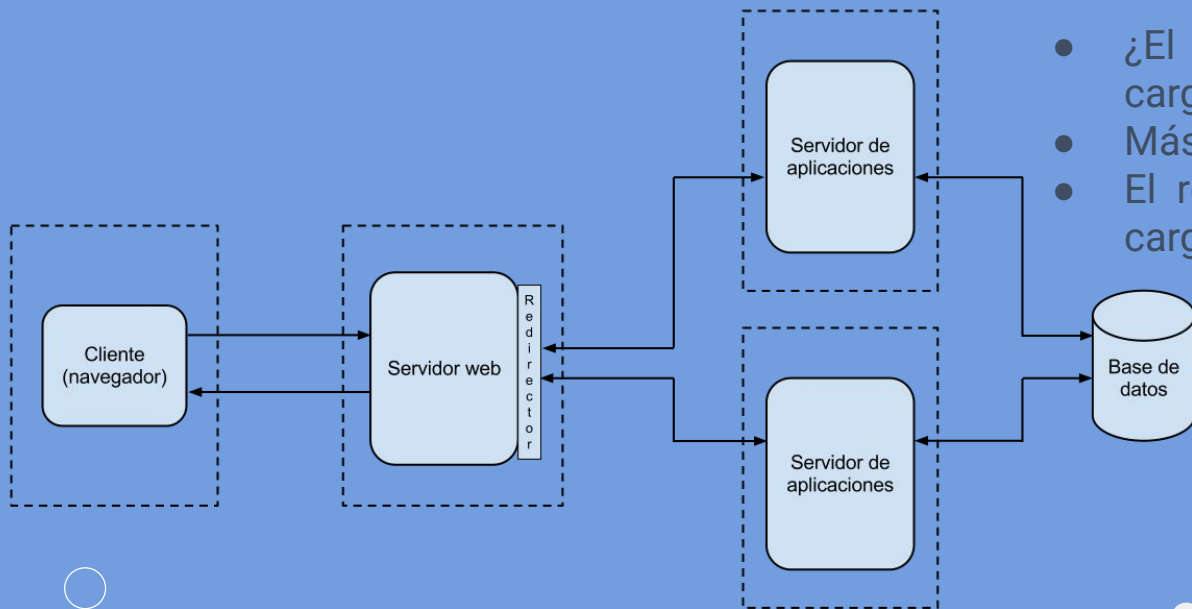


Modelo con Servidor de Aplicaciones

- El servidor de aplicaciones suele tener menos carga
- El servidor web tiene más recursos para él
- El redirector se encarga de transferir los elementos para su ejecución



Modelo con varios Servidores de Aplicaciones



- ¿El servidor de aplicaciones tiene más carga?
- Más servidores que trabajan en paralelo
- El redirector se encarga de balancear la carga de trabajo (SAs deben ser iguales)

Configuración básica

- `sudo apt install apache2`
- `sudo apt install nginx`



Configuración básica

Nota: AMBOS POR DEFECTO
OCUPAN EL PUERTO 80.
¡CUIDADO!

```
josej@jose-virtualbox: ~  
Tasks: 55 (limit: 2255)  
Memory: 5.1M (peak: 5.3M)  
CPU: 42ms  
CGroup: /system.slice/apache2.service  
├─3874 /usr/sbin/apache2 -k start  
├─3876 /usr/sbin/apache2 -k start  
└─3878 /usr/sbin/apache2 -k start  
sep 17 18:35:54 jose-virtualbox systemd[1]: Starting apache2.service  
sep 17 18:35:55 jose-virtualbox apachectl[3873]: AH00558: apache2: Co  
sep 17 18:35:55 jose-virtualbox systemd[1]: Started apache2.service  
lines 1-16/16 (END)...skipping...  
● apache2.service - The Apache HTTP Server  
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)  
  Active: active (running) since Wed 2025-09-17 18:35:55 CEST; 26s ago  
    Docs: https://httpd.apache.org/docs/2.4/  
  Main PID: 3874 (apache2)  
    Tasks: 55 (limit: 2255)  
  Memory: 5.1M (peak: 5.3M)  
    CPU: 42ms  
  CGroup: /system.slice/apache2.service  
          └─3874 /usr/sbin/apache2 -k start  
            └─3876 /usr/sbin/apache2 -k start  
              └─3878 /usr/sbin/apache2 -k start  
sep 17 18:35:54 jose-virtualbox systemd[1]: Starting apache2.service - The Apache HTTP Server...  
sep 17 18:35:55 jose-virtualbox apachectl[3873]: AH00558: apache2: Could not reliably determine the server  
sep 17 18:35:55 jose-virtualbox systemd[1]: Started apache2.service - The Apache HTTP Server.
```

```
josej@jose-virtualbox:~$ systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy  
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)  
  Active: active (running) since Wed 2025-09-17 19:01:30 CEST; 1min 28s ago  
    Docs: man:nginx(8)  
  Process: 8948 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; -p /run/nginx.pid  
 Main PID: 8951 (nginx)  
    Tasks: 2 (limit: 2255)  
  Memory: 3.0M (peak: 3.1M)  
    CPU: 22ms  
  CGroup: /system.slice/nginx.service  
          └─8951 "nginx: master process /usr/sbin/nginx -t -q -g daemon on; -p /run/nginx.pid"  
            └─8953 "nginx: worker process"
```

Configuración básica



Configuración básica



```
josej@jose-virtualbox:~$ curl -I http://localhost
```

```
HTTP/1.1 200 OK
```

```
Date: Thu, 18 Sep 2025 05:33:07 GMT
```

```
Server: Apache/2.4.58 (Ubuntu)
```

```
Last-Modified: Wed, 17 Sep 2025 16:35:51 GMT
```

```
ETag: "29af-63f01d463b8a3"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 10671
```

```
Vary: Accept-Encoding
```

```
Content-Type: text/html
```

```
josej@jose-virtualbox:~$ curl -I http://localhost
```

```
HTTP/1.1 200 OK
```

```
Server: nginx/1.24.0 (Ubuntu)
```

```
Date: Thu, 18 Sep 2025 05:34:11 GMT
```

```
Content-Type: text/html
```


```
Content-Length: 10671
```

```
Last-Modified: Wed, 17 Sep 2025 16:35:51 GMT
```

```
Connection: keep-alive
```

```
ETag: "68cae367-29af"
```

```
Accept-Ranges: bytes
```



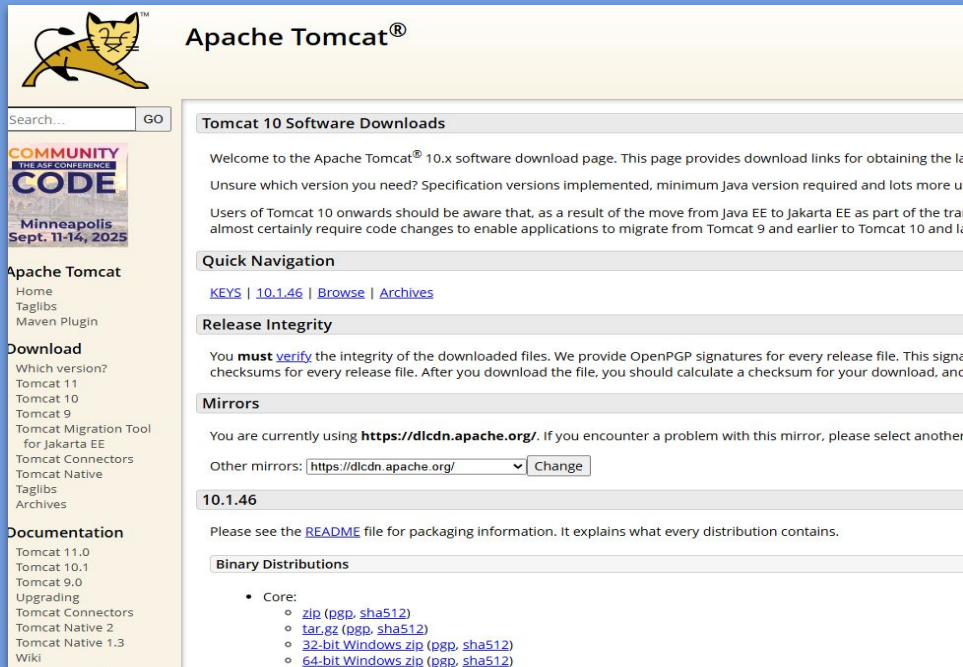
Configuración básica

```
server {  
    listen 80;  
    server_name dominio.local;  
  
    root /var/www/dominio;  
    index index.html;  
  
    access_log /var/log/nginx/dominio_access.log;  
    error_log /var/log/nginx/dominio_error.log;  
}
```

```
<VirtualHost *:80>  
    ServerName dominio.local  
    DocumentRoot /var/www/dominio  
    ErrorLog ${APACHE_LOG_DIR}/dominio_error.log  
    CustomLog ${APACHE_LOG_DIR}/dominio_access.log combined  
</VirtualHost>
```

- Puerto de escucha
- Nombre
- Directorio raíz
- web
- Logs de acceso y error

Configuración básica



The screenshot shows the Apache Tomcat 10 Software Downloads page. The header features the Apache Tomcat logo and a search bar. The main content area is titled "Tomcat 10 Software Downloads" and includes a welcome message, a "Quick Navigation" section with links to KEYS, 10.1.46, Browse, and Archives, and a "Release Integrity" section. A "Mirrors" section shows the current mirror as https://dlcdn.apache.org/ and a "Binary Distributions" section with a list of core distributions: zip (pgp, sha512), tar.gz (pgp, sha512), 32-bit Windows zip (pgp, sha512), and 64-bit Windows zip (pgp, sha512).

Apache Tomcat®

Search... GO

COMMUNITY
THE ASP CONFERENCE
CODE
Minneapolis
Sept. 11-14, 2025

Apache Tomcat

- Home
- Taglibs
- Maven Plugin

Download

Which version?

- Tomcat 11
- Tomcat 10
- Tomcat 9
- Tomcat Migration Tool for Jakarta EE
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation

- Tomcat 11.0
- Tomcat 10.1
- Tomcat 9.0
- Upgrading
- Tomcat Connectors
- Tomcat Native 2
- Tomcat Native 1.3
- Wiki

Tomcat 10 Software Downloads

Welcome to the Apache Tomcat® 10.x software download page. This page provides download links for obtaining the latest version of Tomcat. If you are unsure which version you need? Specification versions implemented, minimum Java version required and lots more useful information.

Users of Tomcat 10 onwards should be aware that, as a result of the move from Java EE to Jakarta EE as part of the transition to the new Eclipse Foundation, almost certainly require code changes to enable applications to migrate from Tomcat 9 and earlier to Tomcat 10 and later.

Quick Navigation

[KEYS](#) | [10.1.46](#) | [Browse](#) | [Archives](#)

Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signa checksums for every release file. After you download the file, you should calculate a checksum for your download, and compare it to the one provided.

Mirrors

You are currently using <https://dlcdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror.

Other mirrors: [Change](#)

10.1.46

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

Configuración básica



Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

To prove that they work, you can execute either of the following links:

- To a [JSP page](#).
- To a [servlet](#).

Configuración básica

Descriptor de despliegue

- Archivo de configuración que indica al **servidor de aplicaciones** como debe desplegar y gestionar una aplicación
- Este archivo normalmente se llama **web.xml**
 - Permite indicar la **página de inicio**
 - Mapeo de URLs
 - Filtros de peticiones
 - Roles de usuario y restricciones
 - Páginas de error personalizadas

Configuración básica



Descriptor de despliegue

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">

  <!-- Declaración de un servlet -->
  <servlet>
    <servlet-name>HolaMundoServlet</servlet-name>
    <servlet-class>com.ejemplo.HolaMundoServlet</servlet-class>
  </servlet>

  <!-- Asociación del servlet a una URL -->
  <servlet-mapping>
    <servlet-name>HolaMundoServlet</servlet-name>
    <url-pattern>/hola</url-pattern>
  </servlet-mapping>

  <!-- Página de inicio por defecto -->
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>

</web-app>
```

+

```
@WebServlet("/hola")
public class HolaMundoServlet extends HttpServlet { ... }
```



Buenas prácticas

- No ejecutar como root
- Configurar logs y revisarlos periódicamente
- Mantener el software actualizado
- Restringir permisos de directorios
- Usar HTTPS siempre que sea posible



Módulos/Extensiones



- Componente adicional que se integra en el servidor para ampliar o modificar sus funcionalidades sin cambiar el núcleo del software.
- Librerías que aportan funcionalidades extra para la ejecución y el despliegue de aplicaciones

+



Módulos/Extensiones

Ventajas

- Flexibilidad
- Rendimiento
- Seguridad
- Compatibilidad

VS

- Sobrecarga del servidor
- Riesgo en seguridad
- Mala gestión

+



Host virtual

- Configuración del servidor
- En un mismo servidor físico o MV
- Aloja múltiples sitios webs independientes

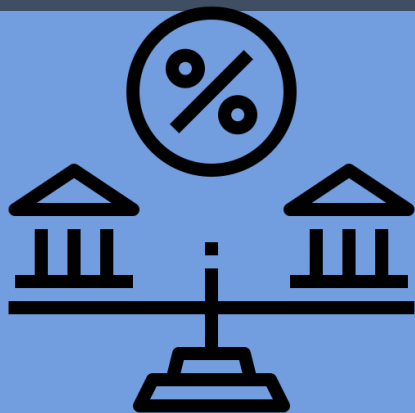
Nota: Cada uno con su dominio, su directorio de archivos y su configuración específica



Host virtual

Ventajas

- Host compartido: te ofrecen varios sitios web con distintos dominios.
- Entorno de desarrollo: puedes trabajar en varios sitios sin necesidad de tener más de un servidor.
- Optimizas recursos: reduces costes y facilitas la administración.



Host virtual

```
sudo chown -R $USER:$USER /var/www/ejemplo1.com/public_html  
sudo chown -R $USER:$USER /var/www/ejemplo2.com/public_html
```

```
echo "<h1>Bienvenido a Ejemplo1.com</h1>" >  
/var/www/ejemplo1.com/public_html/index.html  
echo "<h1>Bienvenido a Ejemplo2.com</h1>" >  
/var/www/ejemplo2.com/public_html/index.html
```

+

Host virtual

```
<VirtualHost *:80>
    ServerName ejemplo1.com
    DocumentRoot /var/www/ejemplo1.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/ejemplo1_error.log
    CustomLog ${APACHE_LOG_DIR}/ejemplo1_access.log combined
</VirtualHost>
```

```
server {
    listen 80;
    server_name ejemplo2.com;
    root /var/www/ejemplo2.com/public_html;
    index index.html;
}
```

```
sudo a2ensite ejemplo1.com.conf
sudo a2ensite ejemplo2.com.conf
sudo systemctl reload apache2
```

Certificados

- SSL o TLS, permite autenticar un sitio web, habilitando la transmisión cifrada de datos entre cliente/servidor.
- Autofirmado
- Emitidos por la CA(Autoridad certificadora)

```
openssl genrsa -out dominio.com.key 2048  
openssl req -new -key dominio.com.key -out dominio.com.csr
```

Creación de la clave privada + solicitud al CA

Certificados

- Una vez aprobada la solicitud tendremos estos archivos
 - **.crt** → identificador de tu dominio, firmado por la CA, cifra la comu.
 - **.key** → clave para descifrar las comunicaciones
 - **.pem** → certificado intermedio para que otros sitios te reconozcan al enlazarte con la CA

```
server {  
    listen 443 ssl;  
    server_name dominio.com;  
  
    ssl_certificate      /etc/ssl/certs/dominio.com.crt;  
    ssl_certificate_key  /etc/ssl/private/dominio.com.key;  
    ssl_trusted_certificate /etc/ssl/certs/ca-bundle.pem;  
  
    root /var/www/dominio.com;  
    index index.html;  
}
```

Indicamos el path en el archivo .conf

Protocolos

HTTP

Protocolo de comunicación para el envío de páginas web (hipertexto).

HTTPS

Protocolo que añade una capa de seguridad (SSL/TLS) al protocolo HTTP

SSH

Protocolo seguro de comunicación utilizado, para la gestión remota de sistemas, debido a que permite la ejecución remota de comandos.

FTP

Protocolo que se utiliza para la transferencia de archivos entre un equipo local y otro remoto.

SFTP

Protocolo que añade SSH al protocolo FTP para así cifrar los usuarios y los archivos enviados.

Nota: Existe una gran variedad de protocolos. Aquí solo se han mostrado los que se consideran más relevantes y que más se usan hoy en día y que, además, serán los que más utilizaremos a lo largo del curso, en los distintos temas.



¿POR QUÉ SALE EL CANDADO CUANDO
NAVEGAMOS POR UNA WEB?

¿Cómo hago que aparezca en mi web?