

Arquitecturas Web	
Modelo Cliente-Servidor	Aplicaciones web pequeñas, clientes realizan petición, servidor devuelve resultado
Modelo Cliente-Servidor con servidores encadenados	Tareas más complejas, distribuir sub tareas en varios servidores
Modelo Peer-to-Peer (P2P)	Todas las computadoras actúan simultáneamente como clientes y servidores

Servidores Web y de Aplicaciones	
Servidores Web	Servidores de Aplicaciones
Servidor que permite el acceso a recursos a través del protocolo HTTP, permiten consultar, cargar y eliminar recursos del servidor.	Servidor que ofrece a los clientes un servicio de ejecución de aplicaciones, es un software que controla la ejecución de programas.
<i>Ejemplos: Apache o Nginx (Linux), Microsoft Internet Information Server (Windows)</i>	<i>Ejemplos: GlassFish (servidor Java EE, Oracle), Tomcat o Microsoft Internet Information Server (servidor .NET)</i>

Servidores de Archivos y Directorios	
Servidores de Archivos	Servidores de Directorios
Se utilizan principalmente para implementar las aplicaciones en el servidor donde se ejecutarán	Permite gestionar información administrativa sobre el entorno de una aplicación web, facilitan la gestión de información relacionada con la explotación de aplicaciones web.
<i>Ejemplos: ProFTPD o vsftpd (Linux), Microsoft Internet Information Server (Windows).</i>	<i>Ejemplos: OpenLDAP (Linux), Active Directory (Windows).</i>

Plataformas Web libres y propietarias	
Plataformas Web Libres	Plataformas Web Propietarias
LAMP: Linux, Apache, MySQL, PHP	WISA: Windows, Internet Information Services (IIS), SQL Server, ASP
<i>Otras plataformas:</i>	
WAMP (Windows-Apache-MySQL-PHP)	
WIMP (Windows-IIS-MySQL-PHP)	
LNMP (Linux + Nginx + MySQL o MariaDB + PHP)	
WNMP (Windows + Nginx + MySQL o MariaDB + PHP)	

Protocolos HTTP y HTTPS	
Protocolo HTTP	Protocolo HTTPS

Protocolos HTTP y HTTPS (cont)	
El protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol) es el motor que da vida a Internet, ya que es la base para la web	Protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es la versión segura de HTTP.
Funcionamiento:	Funcionamiento
1. Un usuario accede a una URL	1. Cliente envía petición de sesión segura
2. El cliente Web descodifica la URL	2. Servidor envía certificado X.509
3. Se abre una conexión TCP/IP con el servidor	3. Cliente autentica certificado
4. El servidor devuelve la respuesta al cliente	4. Cliente genera clave simétrica aleatoria
5. Se cierra la conexión TCP	5. Cliente y servidor conocen y utilizan clave asimétrica.

Instalación y configuración de Nginx
Actualizar repositorios: <code>sudo apt update</code>
Actualizar sistema: <code>sudo apt upgrade</code>
Instalar Nginx: <code>sudo apt install nginx</code>
Comprobar instalación: <code>systemctl status nginx</code>
Ruta registros correctos del servidor: <code>/var/log/nginx/access.log</code>
Ruta registros erróneos del servidor: <code>/var/log/nginx/error.log</code>
Reiniciar servidor: <code>sudo systemctl restart nginx</code>



Crear sitio web en Nginx

Crear carpeta para el sitio: `sudo mkdir -p /var/www/ nombre /html`

Carpeta sea de usuario web: `sudo chown -R www-data : www -data /var/www/ nombre /html`

Conceder permisos para evitar error: `sudo chmod -R 755 /var/www/ nombre /html`

Editar archivo de configuración: `sudo nano /etc/nginx/sites-available/ nombre`

Crear archivo simbólico para alta automática: `sudo ln -s /etc/nginx/sites-available/ nombre /etc/nginx/sites-enabled/`

Reiniciar servidor: `sudo systemctl restart nginx`

Autenticación en Nginx

Instalar herramienta htpasswd para generar contraseña: `sudo apt install apache2-utils`

Generar usuario y contraseña: `sudo htpasswd /etc/nginx/.htpasswd usuario`

Editar configuración nginx: `sudo nano /etc/nginx/sites-available/ nombre`

Activación HTTPS en Nginx

Crear carpeta para los certificados: `sudo mkdir /etc/nginx/ssl`

Crear certificados y claves: `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/ servidor.crt`

