

## DESPLIEGUE TEMA 1

### ÍNDICE

1. Introducción
2. El protocolo HTTP
3. Arquitectura web
  1. Capas de la arquitectura
  2. Modelos
  3. Páginas
  4. Aplicaciones Web
4. Plataformas y servidores web
5. Despliegue en internet
6. Virtual Box
  1. Modos de conexión de red
  2. LAMPP
  3. NGINX

### 1 INTRODUCCIÓN

- **¿Qué es internet?:** Un conjunto de **nodos** conectados en forma de malla parcial, que se comunican a través de los **protocolos**. Cada protocolo permite ofrecer un **servicio** o parte de él.
- **Protocolos de internet:**
  - **HTTP:** World Wide Web
  - **FTP:** Transferencia de archivos
  - **DNS:** Resolución de nombres
  - **SMTP, POP, IMAP:** Correo Electrónico
  - **RIP, OSPF, BGP:** Enrutamiento de paquetes.
  - **TELNET, SSH:** Conexión remota por terminal
  - **VNC, RDP:** Conexión remota gráfica

### 2 EL PROTOCOLO HTTP

Permite el servicio WWW, desarrollado por el World Wide Consortium y la Internet Engineering Task Force, el servidor atiende peticiones en el puerto 80. La versión más usada actualmente es HTTP/1.1, también existe la versión HTTP/2 poco usada.

Existe la versión segura, que es HTTPS, en la que el servidor atiende el puerto 443. Es el protocolo sin estado, no guarda ninguna información sobre conexiones anteriores.

- **Métodos HTTP:** get, post, put, delete, head, options, connect, patch, trace.
- **Códigos de respuesta HTTP:** 4xx – Errores del cliente, 5xx – Errores del servidor.

## **3 ARQUITECTURA WEB**

### **3.1 CAPAS DE LA ARQUITECTURA**

- **Capa de presentación:** Se trata de la capa que se presenta al usuario.
- **Capa de negocio:** Recibe las peticiones del usuario y desde donde se le envían las respuestas, en esta capa se verifican que las reglas establecidas se cumplen.
- **Capa de acceso de datos:** es la formada por determinados gestores de datos que se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa de negocio.

### **3.2 MODELOS**

- **Modelo Punto a Punto:** Arquitectura de red descentralizada y distribuida en la que los nodos individuales de la red actúan tanto como suministradores como clientes.
  - Ventajas: Escalabilidad, Robustez, Descentralización, Distribución de costes.
  - Desventajas: Falta de fiabilidad de los recursos, Difícil mantenimiento.
- **Modelo Cliente-Servidor:** Es un modelo informático que actúa como una aplicación distribuida que particiona tareas o cargas de trabajo entre los servidores y clientes.
  - Ventajas: Centralización, Escalabilidad, Fácil Mantenimiento.
  - Desventajas: Congestión, No Robustez, Cliente no dispone de recursos del servidor.
- **Modelo Servidor de aplicaciones:** La función que realiza un servidor de aplicaciones es diferente, ya que los recursos que va a manipular no son archivos estáticos, sino que contienen el código que tiene que ejecutar.
- **Modelo con servidor de aplicaciones externo:** Separa el servidor web del servidor de aplicaciones.
- **Modelo con varios servidores de aplicaciones:** Si la carga del trabajo del servidor de aplicaciones se estima que va a ser elevado, se puede implantar un sistema con varios servidores de aplicaciones unidos a un mismo servidor web que requiere menos rendimiento.

### **3.3 PÁGINAS**

Las primeras páginas web que se crearon fueron estáticas, éstas se basan en el lenguaje **HTML**, el cual es interpretado por los navegadores web para mostrar la información al usuario. Las páginas web creadas exclusivamente con lenguaje HTML no variarán en su forma ni en su contenido mientras su desarrollador no modifique el código de la página.

El lenguaje de marcas **CSS** se puede utilizar junto al lenguaje HTML para facilitar el diseño del aspecto visual que ofrecen las páginas web.

### **3.4 APLICACIONES WEB**

Algunos lenguajes de programación permiten integrar código junto con el lenguaje HTML para ofrecer contenidos dinámicos en las páginas web. De esta manera en la web crece enormemente en posibilidades y da origen a las aplicaciones web.

- **Lado Cliente:** Los casos más sencillos de aplicaciones web son los lenguajes de programación que son interpretados directamente por los navegadores y que se encuentran embebidos dentro del código fuente HTML de la página web, estos se conocen como los **lenguajes embebidos**.
- **Lado Servidor:** No sólo el navegador puede crear dinamismo en las páginas web. El servidor web puede devolvernos una información u otra en una misma página web en función de la interacción del usuario o de cualquier otra circunstancia. En este tipo de aplicaciones es muy frecuente que el servidor web acceda a una **base de datos**.

### **4 DESPLIEGUE EN INTERNET**

- **Escalabilidad:**
  - **Vertical:** Servidores más potentes.
  - **Horizontal:** Más servidores.
- **Tipos de servidores:**
  - **Sin virtualización:**
    - **Servidor Dedicado**
    - **Servidor Compartido**
  - **Con virtualización:**
    - **Servidor Virtual**
    - **Nube**
- **La nube:**
  - **IaaS:** Infraestructura como Servicio.
  - **PaaS:** Plataforma como Servicio.
  - **SaaS:** Software como Servicio.

### **5 VIRTUAL BOX**

#### **5.1 MODOS DE CONEXIÓN DE RED**

- **Red Interna:** Podremos comunicar las máquinas virtuales entre ellas, pero no tendremos acceso ni al equipo real ni a Internet.
- **Red NAT:** El host facilita la dirección IP a la máquina virtual.
- **Adaptador Puente:** Simula una conexión física a la red en la que está la máquina real, en nuestro caso, el aula.

## 5.2 LAMPP

LAMPP es un paquete de software libre, que consiste, principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache, los intérpretes para lenguajes de script PHP y Perl.

## 5.2 NGINX

Es un servidor Open Source de alto rendimiento, que permite numerosas conexiones simultáneas, lo que proporciona más velocidad y escalabilidad. Entrega el contenido del sitio web rápidamente, es fácil de configurar y tiene un bajo consumo de recursos.

NGINX funciona con base en eventos. En lugar de hacer una solicitud directa al servidor, ejecuta un proceso maestro, llamado worker, y varios procesos de trabajo, llamados conexiones worker. Todo este proceso funciona de forma continua y asíncrona.

Diferencias con Apache:

- **Compatibilidad del sistema operativo:** Los dos servidores funcionan en ambiente basados en UNIX, como LINUX. Con respecto a la plataforma Windows, NGINX tiene un rendimiento inferior en este ambiente.
- **Configuraciones:** La configuración de Apache se realiza de forma descentralizada y la carga de sus módulos se realiza en tiempo de ejecución, y en NGINX, la configuración se centraliza en el archivo "nginx.conf" y sus módulos se cargan dinámicamente.
- **Rendimiento:** La gran ventaja de NGINX es su capacidad de operar con miles de conexiones simultáneas, al doble de la velocidad requerida en Apache y, aun así, consumir menos memoria para contenido estático.