

# TEMARIO: ARQUITECTURA EN LA NUBE - PRIMERA SUBIDA DE UN SERVICIO EN LA NUBE

---

## CONCEPTOS BÁSICOS PARA LA CLASE

### ¿Qué es WSL?

WSL (*Windows Subsystem for Linux*) es como tener dos ordenadores en uno. Permite que Windows pueda simular un ordenador Linux sin necesidad de instalarlo por separado. Es como una ventana dentro de Windows donde puedes usar programas y comandos propios de Linux.

### ¿Por qué es útil?

Porque muchas herramientas de programación y servidores web funcionan mejor en Linux, pero no queremos abandonar Windows. WSL nos permite usar ambos al mismo tiempo sin complicaciones.

---

### ¿Qué es una Máquina Virtual?

Una máquina virtual es un programa que simula ser un ordenador completo dentro de tu ordenador real. Tiene su propio sistema operativo y funciona como un ordenador independiente.

- **VirtualBox** es el programa que utilizamos para crear máquinas virtuales.
- **Ubuntu** es el sistema operativo Linux que instalaremos dentro de esa máquina virtual.
- Ambos son gratuitos y fáciles de usar.

### Conceptos Clave que Vamos a Usar

- **Apache y Nginx:** Programas que convierten tu ordenador en un servidor web, como los que utilizan las páginas de Internet. Es como tener tu propia mini-Internet en casa.
  - **PHP:** Lenguaje de programación que permite crear páginas web dinámicas. A diferencia de HTML, que solo muestra información fija, PHP puede mostrar contenido diferente según el usuario.
- 

### ¿Por qué hacemos esta práctica?

Estamos aprendiendo los primeros pasos para entender cómo funciona Internet por dentro. Cada página web que visitas está alojada en un servidor similar al que vamos a configurar.

Es como aprender a cocinar empezando por hervir agua: una tarea simple pero fundamental.

Al final de la clase comprenderás cómo tu ordenador puede convertirse en un pequeño servidor web, parecido a los que usan Google o Facebook, pero en versión casera.

---

## CONCEPTOS CLAVE QUE VAMOS A USAR

### PARTE 1: COMANDOS APACHE + PHP

#### 1. Actualizar el sistema

Comando:

```
sudo apt update && sudo apt upgrade -y
```

Descripción: Actualiza los paquetes del sistema.

---

#### 2. Instalar Apache2

Comando:

```
sudo apt install apache2 -y
```

Descripción: Instala el servidor web Apache.

---

#### 3. Instalar PHP

Comando:

```
sudo apt install php libapache2-mod-php -y
```

Descripción: Instala PHP y su módulo para Apache.

---

#### 4. Iniciar Apache

Comando:

```
sudo service apache2 start
```

Descripción: Inicia el servicio de Apache.

---

#### 5. Verificar estado de Apache

Comando:

```
sudo systemctl status apache2
```

Descripción: Comprueba que Apache está funcionando.

---

## 6. Crear archivo PHP info

Comando:

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

Descripción: Crea un archivo que muestra información de PHP.

---

## 7. Verificar en navegador

Comando:

```
curl http://localhost/info.php
```

Descripción: Prueba el archivo PHP desde terminal.

---

## **PARTE 2: COMANDOS NGINX + HTML**

### 1. Instalar Nginx

Comando:

```
sudo apt install nginx -y
```

Descripción: Instala el servidor web Nginx.

---

### 2. Iniciar Nginx

Comando:

```
sudo service nginx start
```

Descripción: Inicia el servicio de Nginx.

---

### 3. Verificar estado de Nginx

Comando:

```
sudo systemctl status nginx
```

Descripción: Comprueba que Nginx está funcionando.

---

#### 4. Crear archivo HTML

Comando:

```
echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>" |  
sudo tee /var/www/html/index.html
```

Descripción: Crea una página HTML simple.

---

#### 5. Verificar en navegador

Comando:

```
curl http://localhost
```

Descripción: Prueba la página HTML desde terminal.

---

#### 6. Ver IP de WSL (si aplica)

Comando:

```
ip addr show eth0 | grep inet
```

Descripción: Muestra la IP para acceso desde Windows.

---

# PROPUESTA DE EJERCICIO PRÁCTICO

## Laboratorio: Configuración de Servidores Web Apache y Nginx

### Objetivos:

- Configurar un servidor Apache con PHP.
  - Configurar un servidor Nginx con contenido HTML.
  - Verificar el funcionamiento de ambos servicios.
  - Documentar el proceso mediante capturas de pantalla.
- 

### Parte 1: Servidor Apache con PHP

#### Tareas a realizar:

1. Actualizar el sistema y tomar captura del comando y resultado.
2. Instalar Apache2 y mostrar captura de la instalación exitosa.
3. Instalar PHP y capturar la confirmación de instalación.
4. Iniciar Apache y tomar captura del estado del servicio.
5. Crear archivo info.php y mostrar captura del comando.
6. Acceder a `http://localhost/info.php` y capturar la página de información de PHP.

#### Entregables Parte 1:

- Captura del comando `sudo systemctl status apache2` mostrando el servicio activo.
  - Captura de la página `info.php` funcionando en el navegador mostrando la versión de PHP.
- 

### Parte 2: Servidor Nginx con HTML

#### Tareas a realizar:

1. Detener Apache con `sudo service apache2 stop` y capturar confirmación.
2. Instalar Nginx y mostrar captura de instalación.
3. Iniciar Nginx y capturar el estado del servicio.
4. Crear página HTML personalizada y mostrar el comando utilizado.
5. Acceder a `http://localhost` y capturar la página HTML funcionando.

### Entregables Parte 2:

- Captura del comando `sudo systemctl status nginx` mostrando el servicio activo.
  - Captura de la página HTML personalizada en el navegador.
- 

### Parte 3: Verificación y Comparación

#### Tareas finales:

1. Cambiar entre servicios (*Apache/Nginx*) y documentar el proceso.
  2. Capturar diferencias entre las páginas servidas por cada servidor.
- Ejecutar comandos de verificación desde terminal con `curl`.
- 

### Criterios de Evaluación:

1. Funcionamiento correcto (60%)
    - Apache sirve correctamente la página PHP con información del sistema.
    - Nginx sirve correctamente la página HTML personalizada.
  2. Documentación (40%)
    - Capturas claras de todos los comandos ejecutados.
    - Capturas del resultado funcional en navegador.
    - Capturas de los estados de los servicios.
- 

### Formato de Entrega:

El documento (*PDF o Word*) debe incluir:

1. Portada con nombre del alumno y fecha.
2. Parte 1: Apache + PHP
  - Capturas de instalación y configuración.
  - Captura de `systemctl status apache2`.
  - Captura de página `info.php` funcionando.
3. Parte 2: Nginx + HTML
  - Capturas de instalación y configuración.
  - Captura de `systemctl status nginx`.
  - Captura de página HTML funcionando.