

# PRÁCTICA 1 – Arquitectura cliente / servidor

1. Indica la secuencia de pasos para recuperar información de un sitio web, indicando lo que hace el cliente y los pasos que se realizan en el servidor.
2. Explica las diferencias fundamentales entre páginas web estáticas y dinámicas.
3. Instala Xampp en tu PC y muestra una captura de un fichero holamundo.html funcionando en tu navegador a través del servidor web
4. Crea un fichero info.php con el siguiente código y muestra captura de pantalla funcionando a través del servidor web:

```
<?php
    phpinfo();
?>
```

5. **Creación de un proyecto inicial y configuración de alias.**

- Crea un espacio de trabajo en "C:\workspace-web"
- Configura un **alias** en httpd.conf (/servidor → C:\software\workspace-web)
- Crea una carpeta dentro de workspace-web llamada "tema1"
- Crea un proyecto en tu IDE con el nombre "tema1" asociado a esa carpeta
- Crea un fichero "**index.html**":

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Ejercicio 1</title>
    <meta name="viewport" content="width=device-width" />
  </head>
  <body>
    <p>PÁGINA PRINCIPAL DEL TEMA1</p>
  </body>
</html>
```

- Crea un fichero llamado "**index.php**" que tenga el siguiente código:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
```

```

        echo "Bienvenido al maravilloso mundo de PHP</br>";
        echo "<a href='test.php'>Test</a>";
        ?>
    </body>
</html>

```

- Dime qué fichero se cargará cuando entres en <http://localhost/servidor>
  - Cambia la configuración de Apache para que cargue primero los ficheros PHP (DirectoryIndex)
- Ejecuta el archivo index.php en el navegador y compruebe que funciona

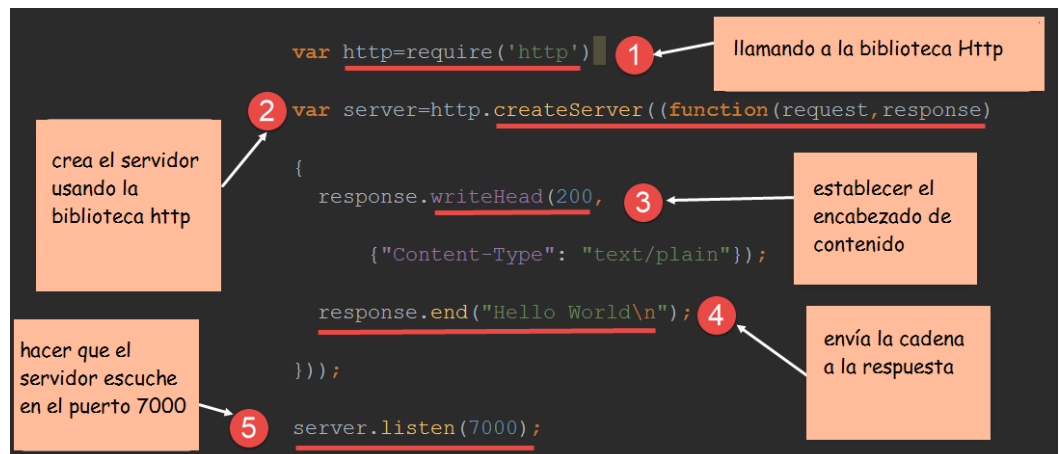
## 6. Instalación de LAMP en Ubuntu.

Puedes realizar este ejercicio en máquina virtual o en el subsistema de Linux para Windows 10.

- Actualiza repositorios
- Instala Apache → `sudo apt install apache2`
- Arranca Apache → `sudo service apache2 start`
- [Opcional] Cambia el puerto al 81 → `sudo nano /etc/apache2/ports.conf`
- Instalar Base de Datos (Mysql Server o MariaDB)
  - `sudo apt install mariadb-server`
  - `sudo /etc/init.d/mysql start` (o `sudo systemctl start mariadb`)
  - `sudo mysql_secure_installation` (no establecemos pass para root)
  - `sudo mariadb`
  - `GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;`
  - `FLUSH PRIVILEGES;` (exit)
  - Lo probamos con: `mysql -u admin -p`
- Instala soporte PHP → `sudo apt-get install php libapache2-mod-php php-mysql`
- Reinicia Apache → `sudo service apache2 restart` (o `sudo systemctl restart apache2`)
- Pruébalo → crea un fichero en `/var/www/html` llamado `test.php` y escribe el código que muestra en php la información ( `phpinfo()` ).
- Instala phpmyadmin → `sudo apt install phpmyadmin`
- Enlace para probar phpmyadmin → `sudo ln -s /usr/share/phpmyadmin /var/www/html`  
(puede que tengas que cambiar el fichero `config.inc.php` para poner el usuario `admin`)

## 7. Node.js. Javascript en el servidor.

- Dime qué es Node.js y para qué se utiliza
- Instala Node.js
- Comprueba que funciona escribiendo en un terminal:
  - `node --version`
  - `npm --version`
- Dime qué es TypeScript y para qué se utiliza
- Instala TypeScript y dime cómo lo has hecho
- Creación de un servidor web con Node.js. Escribe el siguiente código en un fichero



- Por último, lo ejecutamos con node. Para ello escribimos desde la consola "node servidor.js"
- Ya podemos consultar en localhost en el puerto 7000 y nos debería mostrar el mensaje de Hola mundo.
- Añade captura del código y del servidor funcionando en el navegador.