

Tema 1: Introducción a PHP y Entorno de Desarrollo



Contenido

1. Introducción a PHP

- Objetivos del Tema

2. Instalación y Configuración

- Descarga de XAMPP
- Requisitos del Sistema
- Personalización de la Instalación

3. Docker para Desarrollo Web

- Configuración de Contenedores
- Despliegue de Aplicaciones PHP
- Pruebas en Servidor Local

4. Creación de Proyectos Web

- Estructura de Proyectos PHP
- Archivos de Configuración
- Organización de Recursos

5. Estructura del Proyecto Web

- Carpeta de Assets y Recursos
- Archivos de Inclusión (includes)
- Configuración PHP y .htaccess

6. Diseño de Interfaz Web Básica

- Integración de HTML con PHP
- Uso de CSS y Frameworks
- Organización de Assets y Recursos
- Bootstrap y Responsive Design
- Separación de Lógica y Presentación

7. Conceptos Básicos de PHP

- Variables y Tipos de Datos
- Inclusión de Archivos (include/require)
- Funciones Básicas del Sistema

8. Ejercicios Prácticos

- Creación de un Proyecto Nuevo
- Desarrollo de una Página PHP Básica
- Implementación de Estructura Modular

9. Recursos Adicionales

- Documentación Oficial de PHP
- Comunidad y Frameworks PHP

1. Introducción a PHP






Objetivos del Tema:



En este primer tema, se introducirá a los estudiantes en el entorno de desarrollo web con PHP y la configuración de servidores locales.

Los participantes aprenderán a:

- Configurar XAMPP y Docker en Windows/Linux/macOS
 - Comprender la estructura básica de un proyecto web PHP
 - Explorar la integración de HTML, CSS y PHP
 - Familiarizarse con la creación de páginas dinámicas
 - Implementar estructura modular con includes
 - Desarrollar ejercicios prácticos fundamentales
- 

2. Instalación y Configuración del Entorno

+

•

○

+

○

•



1. Descarga de XAMPP:

Dirígete al sitio web oficial de XAMPP en <https://www.apachefriends.org/> y descarga el instalador adecuado para tu sistema operativo.

2. Ejecución del Instalador:

Una vez completada la descarga, ejecuta el instalador como administrador. Esto iniciará el proceso de instalación de XAMPP.

3. Selección de Componentes:

- Apache: Servidor web que ejecutará PHP
- MySQL: Sistema de gestión de base de datos
- PHP: Lenguaje de programación principal
- phpMyAdmin: Interfaz web para gestión de MySQL
- FileZilla FTP Server: Para transferencias de archivos (opcional)

```
// Configuración de errores para desarrollo
```

```
error_reporting( error_level: E_ALL);
```

```
ini_set( option: 'display_errors', value: 1);
```

```
// Variables con información personal
```

```
$nombre = "Juan Pérez";
```

```
$edad = 22;
```

```
$ciudad = "Madrid";
```

```
$profesion = "Estudiante";
```

```
$hobbies = ["Programar",
```

```
// Variables de fecha y
```

```
$fecha_actual = date( form
```

```
$hora_actual = date( form
```

```
$dia_semana = date( format
```

```
// Traducir día de la se
```

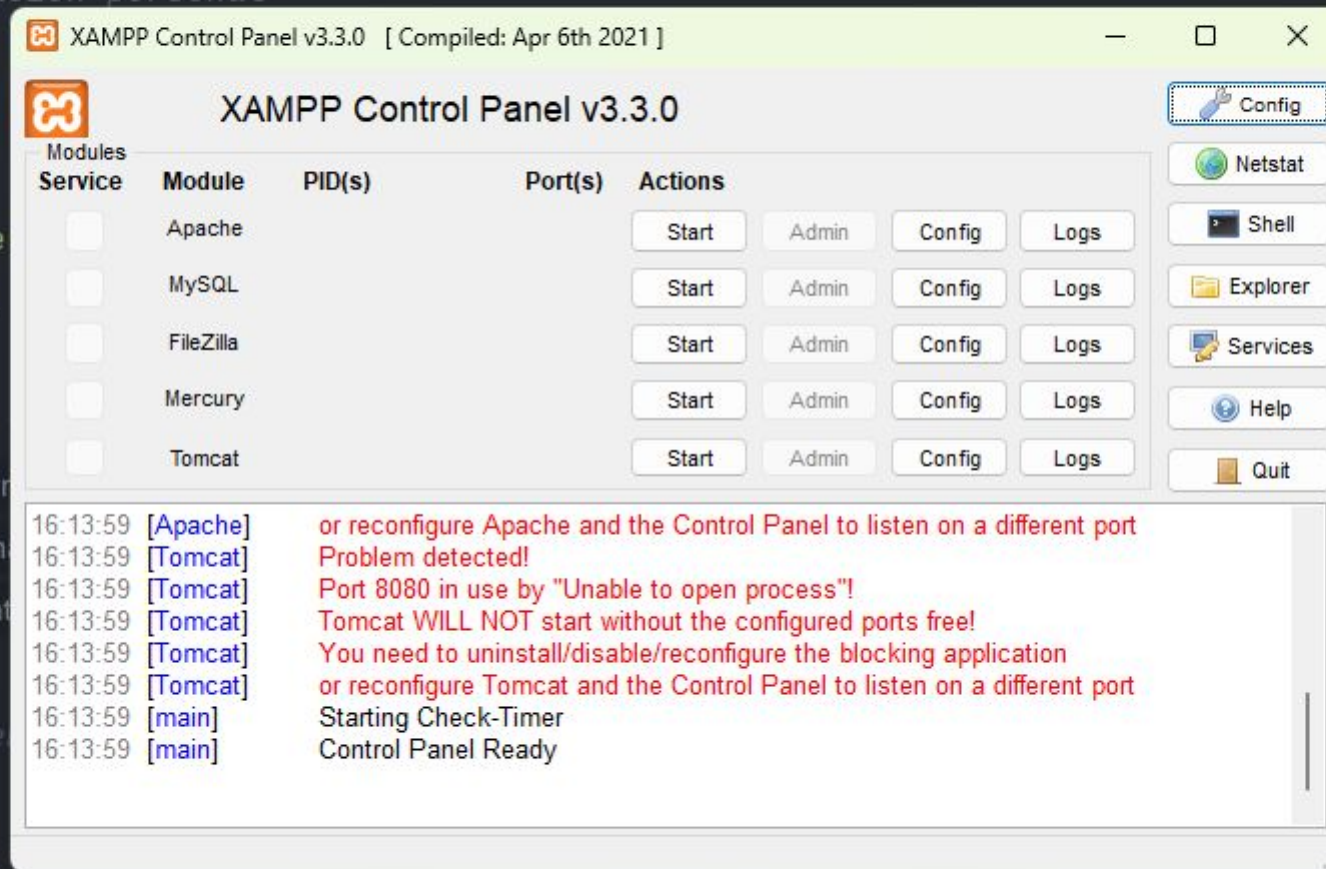
```
$dias_espanol = [
```

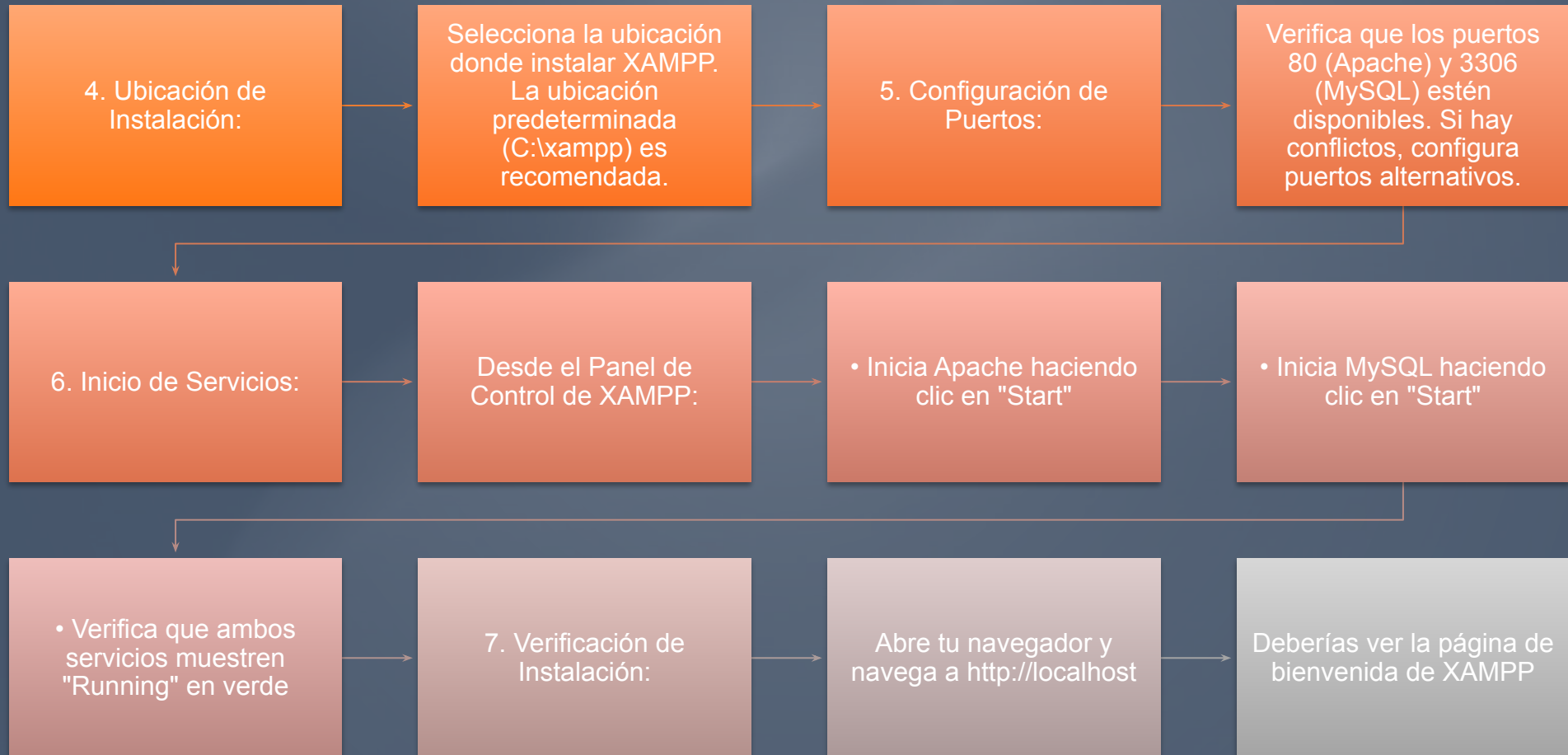
```
'Monday' => 'Lunes',
```

```
'Tuesday' => 'Martes',
```

```
'Wednesday' => 'Miércoles',
```

```
'Thursday' => 'Jueves',
```





8. Configuración de PHP:

XAMPP incluye PHP preconfigurado, pero puedes personalizar la configuración editando php.ini

Ubicación típica: C:\xampp\php\php.ini

9. Acceso a phpMyAdmin:

Navega a <http://localhost/phpmyadmin>

Interfaz web para gestionar bases de datos MySQL

Usuario por defecto: root (sin contraseña)

10. Configuración del Editor de Código:

Instala Visual Studio Code o tu editor preferido

Configura extensiones para PHP: PHP Intelephense, PHP Debug

Configura el workspace en la carpeta htdocs de XAMPP



3. Docker para Desarrollo Web PHP



Configuración de Contenedores Docker

Docker para PHP: Introducción

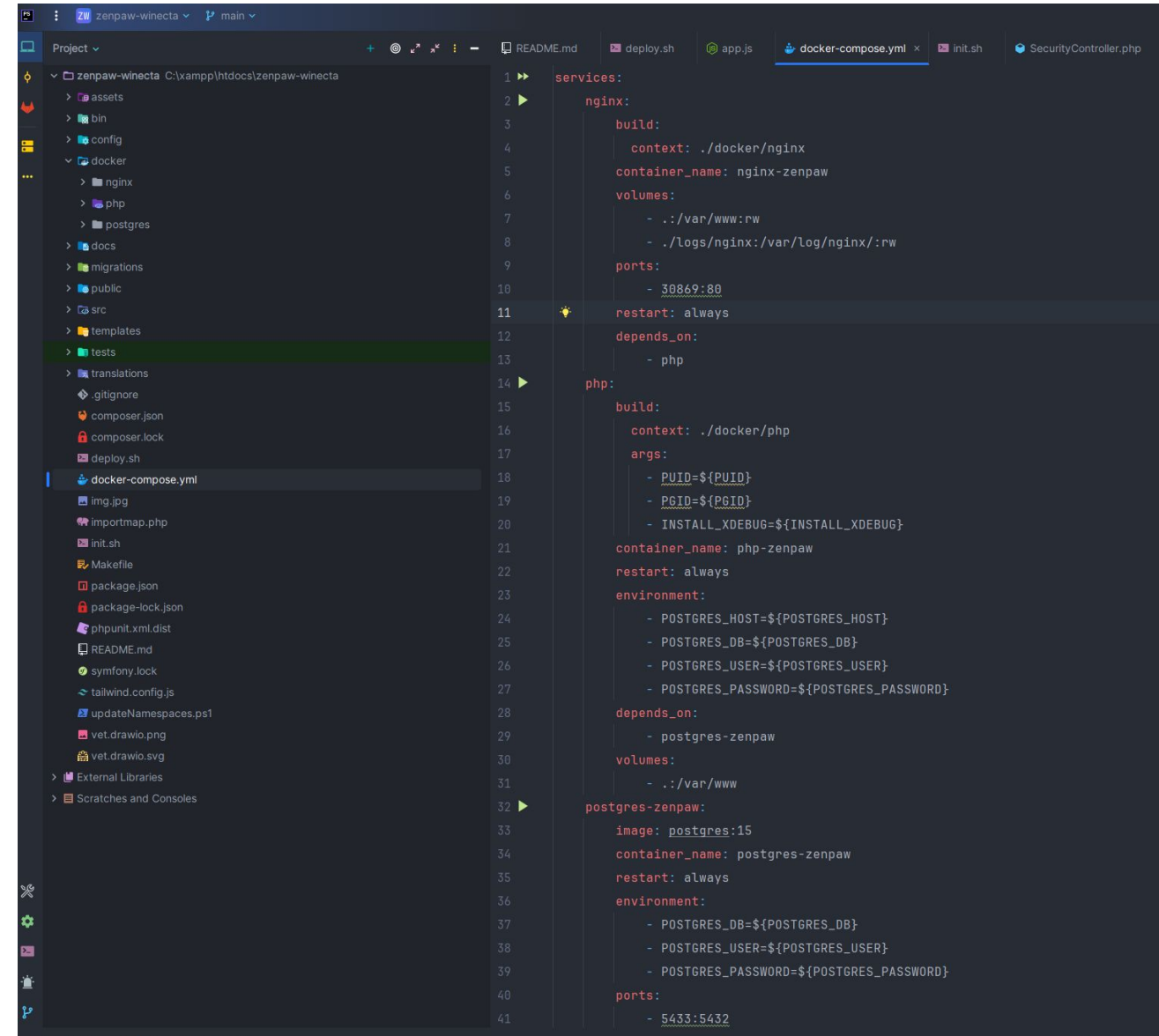
- Docker es una herramienta esencial para desarrollo web moderno.
- Permite crear entornos de desarrollo consistentes y reproducibles.
- Evita problemas de compatibilidad entre diferentes sistemas.

Creación de un Entorno Docker PHP

- Instala Docker Desktop desde <https://www.docker.com/products/docker-desktop>
- Crea un archivo docker-compose.yml para tu proyecto
- Define servicios para PHP, Apache/Nginx, MySQL
- Configura volúmenes para desarrollo en tiempo real

Configuración del Entorno Docker

- Define la imagen PHP adecuada (php:8.3-apache)
- Configura el mapeo de puertos (80:80 para web, 3306:3306 para MySQL)
- Establece variables de entorno para la base de datos
- Monta tu código fuente como volumen para desarrollo



```
services:
  nginx:
    build:
      context: ./docker/nginx
    container_name: nginx-zenpaw
    volumes:
      - ../var/www:rw
      - ../logs/nginx:/var/log/nginx/:rw
    ports:
      - 30869:80
    restart: always
    depends_on:
      - php
  php:
    build:
      context: ./docker/php
    args:
      - PUID=${PUID}
      - PGID=${PGID}
      - INSTALL_XDEBUG=${INSTALL_XDEBUG}
    container_name: php-zenpaw
    restart: always
    environment:
      - POSTGRES_HOST=${POSTGRES_HOST}
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
    depends_on:
      - postgres-zenpaw
    volumes:
      - ../var/www
  postgres-zenpaw:
    image: postgres:15
    container_name: postgres-zenpaw
    restart: always
    environment:
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
    ports:
      - 5433:5432
```

Instale la versión más reciente de PowerShell para obtener nu





```
~\PhpstormProjects\Curso git:[master]  
docker compose up -d
```

Uso del Entorno Docker

Comandos Básicos de Docker

- En tu directorio de proyecto, ejecuta: `docker-compose up -d`
- Accede a tu aplicación en `http://localhost`
- Para detener los contenedores: `docker-compose down`

Ventajas del Desarrollo con Docker

-  Entorno consistente entre desarrollo y producción
-  Fácil compartir configuración con el equipo
-  Aislamiento de dependencias y versiones
-  Escalabilidad para proyectos complejos


4. Creación de Proyectos Web



Estructura Básica de Proyectos PHP



Utilizaremos principalmente estas estructuras:

- Proyecto Simple: Una sola página PHP
 - Proyecto Modular: Separación de componentes con includes
- 

Creando un Proyecto PHP Paso a Paso:

1. ****Crear Carpeta del Proyecto****

- Navegar a C:\xampp\htdocs\
- Crear nueva carpeta: "mi_proyecto_web"

2. ****Configurar Estructura Base****

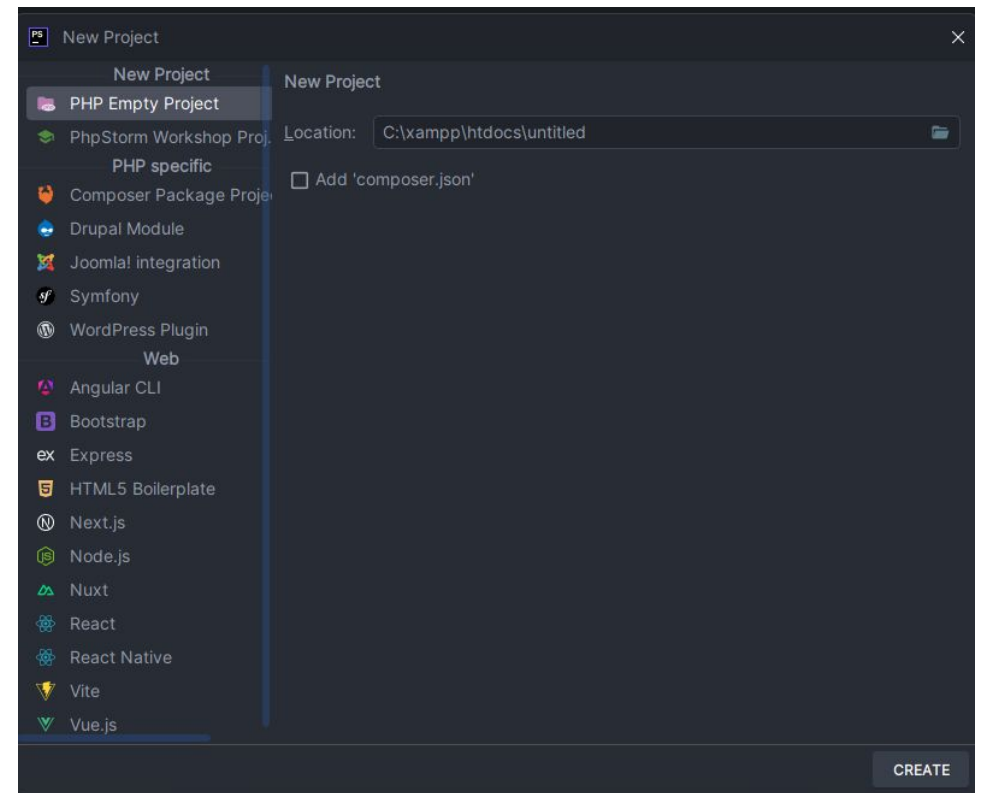
- Crear archivo index.php principal
- Crear carpetas: config/, assets/, includes/, pages/

3. ****Establecer Archivos de Configuración****

- config.php: Variables globales y configuración DB
- .htaccess: Configuración de servidor Apache

4. ****Probar el Proyecto****

- Abrir navegador en http://localhost/mi_proyecto_web/
- Verificar funcionamiento correcto



5. Estructura del Proyecto Web

Mi Proyecto Web/

```
|— index.php      # Página principal
|— config/       # Configuración
|   |— config.php
|— assets/       # Recursos estáticos
|   |— css/
|       |— style.css
|   |— js/
|       |— main.js
|   |— images/
|— includes/     # Archivos incluibles
|   |— header.php
|   |— footer.php
|— pages/        # Páginas del sitio
|   |— about.php
|   |— contact.php
|   |— services.php
```



Esta estructura permite:

- ✓ Código organizado y mantenible
- ✓ Reutilización de componentes
- ✓ Separación de responsabilidades
- ✓ Escalabilidad del proyecto

6. Diseño de Interfaz Web Básica

Integración HTML + PHP:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title><?php echo $titulo; ?></title>
  <link rel="stylesheet"href="assets/css/style.css">
</head>
<body>
  <h1><?php echo "¡Hola desde PHP!"; ?></h1>
</body>
</html>
```



CSS Frameworks Recomendados:

- ****Bootstrap 5****: Diseño responsive rápido
- ****Tailwind CSS****: Utilidades CSS modernas
- ****CSS Personalizado****: Control total del diseño



Responsive Design:

- Mobile First approach
- Media queries para diferentes dispositivos
- Flexbox y CSS Grid para layouts

7. Conceptos Básicos de PHP

Variables y Tipos de Datos:

```
<?php
```

```
// Strings
```

```
$nombre = "Juan Pérez";
```

```
$apellido = 'García';
```

```
// Números
```

```
$edad = 25;
```

```
$precio = 19.99;
```

```
// Booleanos y Arrays
```

```
$activo = true;
```

```
$colores = ["rojo", "verde", "azul"];
```

```
?>
```

Inclusión de Archivos:

- ****include****: Incluye archivo, continúa si falla
- ****require****: Incluye archivo, detiene si falla
- ****include_once / require_once****: Solo una vez

Funciones Básicas:

- **`date()`**: Fecha y hora actual
- **`echo / print`**: Mostrar contenido
- **`var_dump()`**: Información de variables

Ejecutando el servidor modo “vanilla”

```
php -S localhost:8000 -t public
```

```
manu@manu-Katana-A15-AI-B8VF:~/PhpstormProjects/flup$ php -S localhost:8000 -t public
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54202 Closing
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54218 Closing
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54230 Accepted
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54230 [200]: GET /build/runtime.544b3ba4.js
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54230 Closing
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54246 Accepted
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54246 [200]: GET /build/363.4d186148.js
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54246 Closing
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54258 Accepted
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54258 [200]: GET /build/58.f1d911bf.js
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54258 Closing
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54272 Accepted
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54272 [200]: GET /build/app.9ce15923.js
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54272 Closing
[Mon Sep 29 20:09:50 2025] 127.0.0.1:54280 Accepted
[Mon Sep 29 20:09:51 2025] 127.0.0.1:54280 [200]: GET /_wdt/f01df9?XDEBUG_IGNORE=1
[Mon Sep 29 20:09:51 2025] 127.0.0.1:54280 Closing
```

-t es solo para especificar el directorio donde esta el index.

Si lo tengo en “/public/” habrá que poner esa flag

Ejecutando php “a pelo”

- Podemos ejecutar php desde terminal como java pero sin compilar 😊

```
~\PhpstormProjects\Curso\Web 2025\Teoria\Tema_1\Ejercicios_Ejemplo\ejercicio_2 git:[master]
php index.php
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Página de inicio del sitio web de ejemplo DWES con PHP y Bootst
  <meta name="keywords" content="PHP, desarrollo web, DWES, ejemplo, tutorial">
  <meta name="author" content="Estudiante DWES">
```



8. Ejercicios Prácticos



Ejercicio 1: Mi Primera Página PHP

- Crear página personal con información dinámica
- Usar variables PHP (\$nombre, \$edad, \$ciudad)
- Mostrar fecha y hora actual con date()
- Aplicar CSS básico para diseño

Ejercicio 2: Sitio Web Modular

- Separar código con include/require
- Crear header.php y footer.php reutilizables
- Configuración centralizada en config.php
- Múltiples páginas (inicio, acerca, contacto, servicios)

Ejercicio 3: Sistema de Navegación Dinámico (opcional)


- Sistema de rutas con \$_GET
- Validación de páginas existentes
- Bootstrap 5 para diseño responsive
- Navegación activa según página actual


**  Todos los ejercicios están en /Ejercicios_Ejemplo/**




Recursos Adicionales



 Documentación Oficial de PHP:
<https://www.php.net/manual/es/>

 Frameworks y Herramientas:

- Bootstrap: <https://getbootstrap.com/>
- Composer: <https://getcomposer.org/>
- Docker: <https://docs.docker.com/>

 Comunidades:

- Stack Overflow - PHP
- Reddit r/PHP
- PHP.net Community

