

## Tema 02. Inserción de Código en Páginas WEb

Desarrollo Web Entorno Servidor. 2º DAW 23/24

# Introducción

---

- ▶ El desarrollo de aplicaciones web del lado del servidor se fundamenta en la utilización de una serie de lenguajes y tecnología, que son ejecutadas por software especializado en uno o más servidores
- ▶ En función del lenguaje que queramos utilizar:
  - ▶ la distribución de la lógica de la aplicación
  - ▶ la forma de interactuar con la información que proviene del cliente
  - ▶ la gestión del flujo de trabajo de la aplicación
  - ▶ la configuración del entorno del servidor será diferente
- ▶ Presentamos:
  - ▶ tipos de servidores web
  - ▶ características principales de algunos lenguajes de servidor
  - ▶ manejar el flujo de información para tratar la información enviada por un cliente a través de una página web

# Objetivos

---

- ▶ Reconocer la arquitectura de las aplicaciones web del lado del servidor dependiendo del lenguaje utilizado.
- ▶ Aprender a generar código de forma dinámica para ser mostrado por el cliente web.
- ▶ Conocer la sintaxis y las etiquetas propias de cada lenguaje de servidor que permite insertar código en páginas web ejecutadas en el servidor.
- ▶ Dominar la declaración de variables, tipos de datos simples, así como la conversión entre cada uno de ellos en función del lenguaje utilizado.
- ▶ Comprender la importancia de controlar el ámbito de declaración de cada variable y su influencia en el desarrollo de aplicaciones web.

# Lenguajes y Tecnologías de Servidor

---

- ▶ Lo primero es entender que el *servidor web* es un programa cuya misión última es servir datos en forma de documentos HTML (HyperText Markup Language) codificados en este lenguaje
- ▶ El objetivo es proveer al cliente con textos complejos con enlaces, figuras, formularios, botones y objetos incrustados, tales como animaciones o reproductores de sonidos.
- ▶ El intercambio de datos cliente/servidor se hace por medio un protocolo determinado, generalmente mediante el protocolo HTTP. Por defecto, un servidor web se mantiene a la espera de peticiones HTTP realizadas por parte de un cliente (a través de un navegador web) "escuchando" un puerto de comunicaciones (normalmente el 80).

# Secuencia Comunicación Cliente/Servidor

---

- ▶ En un primer paso, el navegador solicita, como cliente DNS la traducción de una URL (por ejemplo *https://www.ieslosremedios.org*) a una IP.
- ▶ Una vez que ha recibido la traducción del servidor DNS se realiza la petición HTTP al servidor web que tenga la IP obtenida (si ponemos directamente la dirección IP en vez de la URL en el navegador el resultado sería equivalente).
- ▶ Puesto que HTTP es un protocolo sin estado, cada petición de un cliente a un servidor no está influenciada por las transacciones anteriores. Esto quiere decir que el servidor tratará cada petición como una operación totalmente independiente del resto.
- ▶ Una novedad de este protocolo a partir de su versión 1.1 es que se pueden habilitar conexiones persistentes, lo cual afecta directamente a la eficiencia de las transacciones puesto que se permite enviar más objetos con un número menor de conexiones.
- ▶ En el siguiente paso, el servidor procesa la solicitud realizada por el cliente y, tras ejecutar el código asociado al recurso solicitado en la URL, responde al cliente enviando el código HTML de la página.
- ▶ Finalmente, el navegador del cliente, cuando recibe el código, lo interpreta y lo muestra en pantalla.

# Protocolo

---

- ▶ Conjunto de reglas y estándares que gobiernan el intercambio de información entre entidades dentro de una red.
- ▶ El protocolo de comunicación de internet es el protocolo TCP/IP, que está formado por un conjunto de sub protocolos, dependiendo del servicio de internet que queramos usar: HTTP, FTP, POP3, SMTP
- ▶ Protocolo HTTP- Protocolo de la capa de aplicación usado en cada transacción web. Cada petición realizada mediante HTTP implica una conexión con el servidor que es liberada al término de la misma, es decir, no tiene estado.

# Clasificación Servidores Web

---

- ▶ En función de cómo se procesen las peticiones realizadas por un cliente web y de las estrategias de optimización de la ejecución y de diseño del servidor dan lugar a diferentes tipos de servidores web:
  - ▶ Servidores basados en procesos
  - ▶ Servidores basados en hilos
  - ▶ Servidores dirigidos por eventos
  - ▶ Servidores implementados en el núcleo del sistema

# Servidores Web Basados en Procesos

---

- ▶ Se puede considerar como la estrategia predecesora de todas las demás
- ▶ Su funcionamiento se basa en la obtención de un paralelismo de ejecución mediante la duplicación del proceso de ejecución
- ▶ El proceso principal espera la llegada de una nueva conexión (petición HTTP) y en ese momento, se duplica creando una copia exacta (fork) que atenderá esta conexión
- ▶ De esta forma un proceso se dedica al tratamiento de la petición mientras que otro puede seguir escuchando nuevas peticiones.
- ▶ Técnica *pre-fork*
  - ▶ En la memoria del servidor web lanzan varias copias del proceso que escucha peticiones
  - ▶ Entre las ventajas de esta técnica destaca la facilidad para ser implementada y la seguridad para aislar el procesamiento de una petición frente a otras
  - ▶ Por el contrario, el mayor inconveniente de esta estrategia está en un mayor consumo de recursos de memoria (dedicada a cada proceso) y un menor rendimiento debido a que la gestión de los procesos es una tarea del sistema operativo



# Servidores Basados en hilos

---

- ▶ Alternativa más económica en cuanto consumo de recursos
- ▶ Al funcionamiento es básicamente idéntico al de los servidores basados en procesos
- ▶ La mayor diferencia se encuentra en la distinción entre el concepto de hilo (thread) y el de proceso
- ▶ La creación de un hilo por parte de un proceso no es tan costosa como la duplicación de un proceso completo. Los hilos de ejecución creados por un proceso comparten el mismo espacio de memoria reduciendo el consumo de memoria del servidor web de forma drástica
- ▶ Desventaja. Puede constituir un problema de seguridad a la hora de acceder al espacio de memoria. Todos los hilos creados por un servidor comparten la misma zona de memoria, si un hilo modifica una variable, el resto de hilos del mismo proceso verían esa variable con el valor modificado

# Servidores Dirigidos por Eventos

---

- ▶ La novedad de este tipo de servidores es la utilización de **sockets** (espacios de memoria para la comunicación entre dos aplicaciones que permiten que un proceso intercambie información con otro proceso estando los dos en distintas máquinas)
- ▶ Las lecturas y escrituras sobre sockets son realizadas de forma asíncrona y bidireccional
- ▶ De esta forma, aunque un cliente realizará peticiones de forma independiente (sin estado), el servidor podrá acceder a la información intercambiada con dicho cliente con solo acceder al socket creado para tal cliente.
- ▶ La ventaja de este diseño radica principalmente en su velocidad
- ▶ el mayor inconveniente se encuentra en que la concurrencia de procesamiento es simulada; es decir, existe un solo proceso y un solo hilo, desde el cual se atienden todas las conexiones gestionadas por un conjunto de sockets

# Servidores implementados en el núcleo del sistema

---

- ▶ La principal idea de esta estrategia de diseño es situar el procesamiento de cada una de las ejecuciones del servidor web en un espacio de trabajo perteneciente al sistema operativo (kernel) y no en un nivel de usuario (sobre el sistema operativo)
- ▶ Si bien es cierto que aunque esta alternativa permite acelerar el procesamiento de las peticiones al servidor, conlleva bastantes peligros
- ▶ Cualquier incidencia (de seguridad o rendimiento) en este tipo de servidores implica un problema a nivel del sistema operativo que lo puede llegar a dejarlo completamente imperativo

# Servidores Web más utilizados

---

- ▶ Soportan diferentes lenguajes PHP, Java, Perl, ...
- ▶ Conjunto extensiones capaz de responder a cualquier petición de servicio
- ▶ Soporte tanto para lenguajes interpretados como compilados.
- ▶ Es necesario por tanto la configuración adecuada y el uso de las extensiones necesarias para obtener mayor rendimiento del servidor web

# Apache Server

---

- ▶ Servidor HTTP admite múltiples plataformas y sistemas operativos
- ▶ Servidor robusto (seguro) que implementa los últimos estándares y protocolos de la red, con gran capacidad de personalización y modularización
- ▶ Se distribuye bajo licencia GPL. Es gratuito de código abierto. Gran comunidad de desarrolladores.
- ▶ Puede configurarse tanto como servidor basado en hilos como basado en procesos.
- ▶ Su diseño altamente flexible y configurable permite a los administradores de sitios web elegir qué características van a ser soportadas por el servidor,
- ▶ Soporte para acceso a bases de datos, establecer páginas protegidas por contraseña, personalizar las páginas de error devueltas por el servidor, generar registros de actividad en múltiples formatos

# Microsoft IIS

---

- ▶ Servidor web de Microsoft IIS (Internet Information Server)
- ▶ Es un motor de páginas web flexible, seguro y fácil de gestionar que permite alojar cualquier tipo de contenido on-line.
- ▶ Destaca principalmente por dar soporte nativo a ASP (Active Server Pages) y a las diferentes tecnologías de la plataforma .Net
- ▶ permite añadir ciertos módulos para la ejecución de otros lenguajes como PHP
- ▶ Este tipo de servidor lo llevan solo los sistemas basados en Windows, y suele instalarse como un complemento adicional al sistema operativo
- ▶ Servicios de despliegue de aplicaciones multimedia, acceso a bases de datos o administración remota del servidor

# Sun Java Systems Web Server.

---

- ▶ Se trata de un servidor web de alto rendimiento, de escalabilidad masiva y seguro, que ofrece contenido dinámico y estático
- ▶ Actualmente forma parte del software que Oracle tiene disponible para su descarga de forma gratuita
- ▶ Destaca por sus características de vitalización de dominios, versatilidad de configuración y seguridad robusta.
- ▶ Está optimizado para la integración y ejecución de aplicaciones Java (JSP, Servlets, NSAPI y DGI)
- ▶ Como servidor web puede instalarse en prácticamente cualquier sistema operativo
- ▶ admite la configuración de módulos y extensiones para la ejecución de código PHP, Ruby on Rails, Perl, Python y más lenguajes de servidor

# Nginx

---

- ▶ Es un servidor HTTP que ha ganado cuota de mercado en los últimos años (de  $< 1\%$  en 2007 a casi el 15% a finales de 2016)
- ▶ Es de código abierto y cuenta con una arquitectura modular de alto rendimiento
- ▶ Permitir funcionar como servidor proxy para otros protocolos de Internet como IMAP o POP3
- ▶ Nginx es un servidor asíncrono basado en hilos
- ▶ Conocido por su estabilidad, contener un amplio conjunto de filtros, ser fácilmente configurable y capaz de ser ejecutado en múltiples plataformas consumiendo relativamente pocos recursos
- ▶ El sistema es usado por una larga lista de sitios web conocidos, como: WordPress, Netflix, Hulu, GitHub, Ohloh, SourceForge, TorrentReactor y partes de Facebook



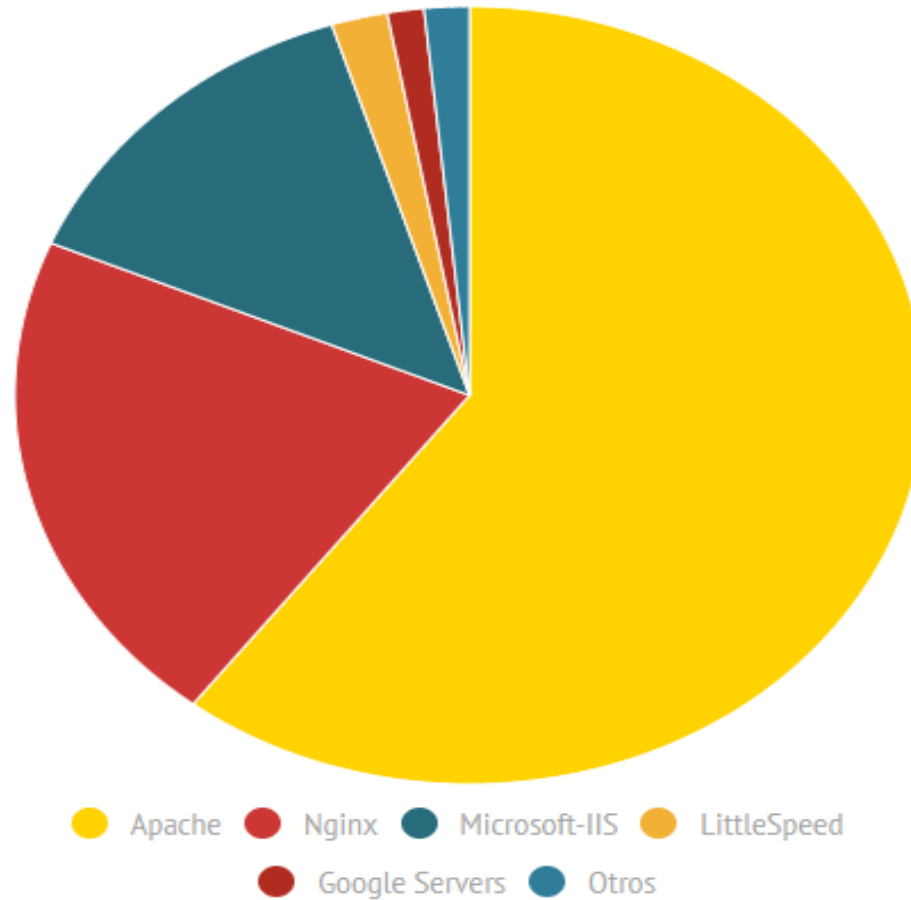
# Lighttpd

---

- ▶ Es un servidor web especializado para entornos en los que se requieren respuestas rápidas.
- ▶ Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM que otros servidores
- ▶ Por todo lo que ofrece, lighttpd es apropiado para cualquier servidor que tenga problemas de carga.
- ▶ Se distribuye bajo licencia BSD. Licencia especial para sistemas operativos.
- ▶ Otra de las características de este servidor es que utiliza un solo proceso con varios hilos de ejecución pero sin tener la capacidad de crear nuevos hilos de ejecución en función de la demanda.
- ▶ Funciona también bajo Windows con su versión específica.

# Cuota de mercado de Servidores Web

---



Datos de W3Techs.com

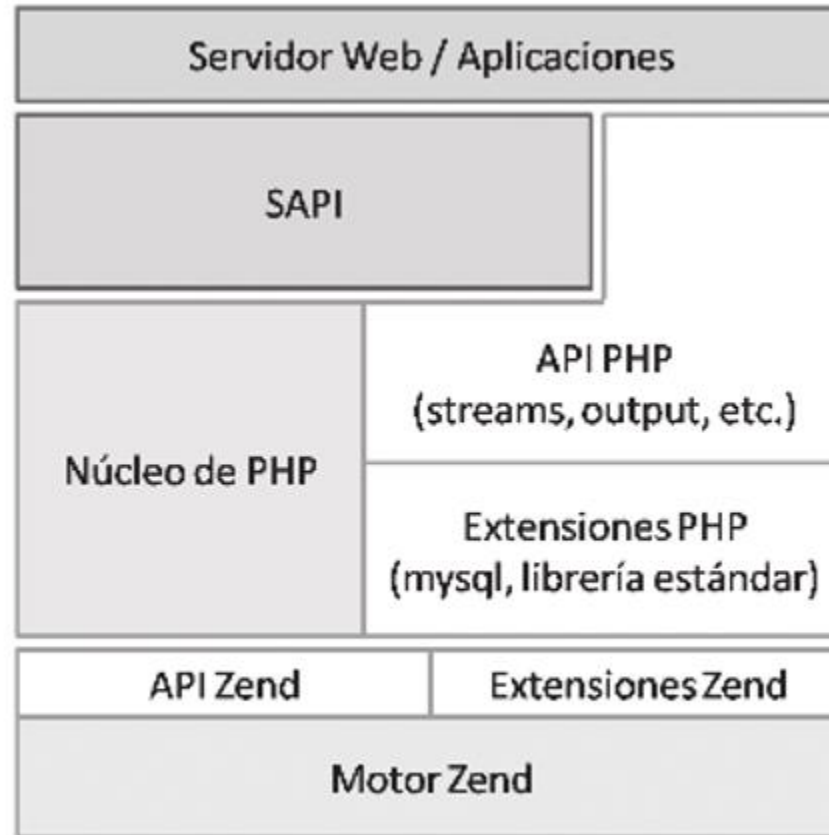
# Obtención Código Enviado Cliente

---

- ▶ En este capítulo nos centraremos en analizar las características fundamentales de PHP como ejemplo de lenguaje de programación web del lado del servidor.
- ▶ PHP (Hypertext Preprocessor) es un lenguaje de scripting de propósito general y de código abierto que ha sido especialmente diseñado para el desarrollo de aplicaciones web y que puede ser embebido (intercalado) en código HTML
- ▶ PHP nació en 1994 a partir de un conjunto de ficheros binarios que se utilizaban para recoger datos de tráfico de datos en sitios web
- ▶ Desde su versión 4, (publicada en el año 2000), hasta su versión más actual, PHP se ha convertido en un lenguaje de gran popularidad entre la comunidad de desarrolladores web
- ▶ Entre las razones de su éxito destaca una relativa facilidad para su aprendizaje.

# Las capas de una arquitectura genérica de un servidor que soporte la ejecución de PHP

---



*Figura 2.1. Arquitectura genérica de PHP con Zend*

# Capas

---

- ▶ La capa más externa de esta arquitectura la conforma el propio servidor web. Recibe las peticiones que el cliente hace a una dirección determinada
- ▶ La capa inmediatamente inferior es la capa SAPI (Server Abstraction API) donde PHP interactúa con el servidor u otras aplicaciones:
  - ▶ gestiona, parcialmente, la inicialización y cierre de PHP como parte integrante de un servidor;
  - ▶ contiene diferentes mecanismos para acceder a datos del cliente tales como las cookies o la información enviada por el método POST
- ▶ **Núcleo PHP:**
  - ▶ maneja la configuración del entorno de ejecución (asignando valores a variables globales y estableciendo las directivas de inicio del fichero de configuración php.ini)
  - ▶ ofreciendo interfaces de programación (API PHP) tales como la interfaz de entrada/salida estándar, transformación de datos (data parsing) y, quizás lo más importante, ofreciendo un interfaz para cargar extensiones, tanto aquellas compiladas estáticamente como las cargadas dinámicamente
- ▶ **Motor zend**
  - ▶ encargado del análisis y ejecución de las porciones de código PHP (scripts)
  - ▶ ha sido diseñado con el fin de admitir extensiones que permiten modificar completamente su funcionalidad

# Pasos ejecución script PHP en Motor Zend

---

1. El script es analizado por un analizador Léxico (lexer) que transforma el código escrito por el desarrollador en un conjunto de piezas (tokens) entendibles por la máquina.
  2. Estos tokens son pasadas después al analizador sintáctico (parser).
  3. El parser toma el conjunto de tokens y genera un conjunto de instrucciones (o código intermedio) que es ejecutado por el motor Zend. Este proceso se conoce con el nombre de compilación. Muchos parsers generan un árbol sintáctico que puede ser manipulado u optimizado antes de ser enviado al generador de código
  4. Fase es la ejecución donde interviene el componente ejecutor de Zend. Su función es ejecutar una por una las instrucciones indicadas en el código intermedio
- ▶ El motor Zend funciona como una máquina virtual, es decir, se trata de un programa que simula una computadora física
  - ▶ El motor Zend implementa alrededor de 150 instrucciones específicas y optimizadas para la ejecución de código PHP (como inclusión de librerías o imprimir un string)