



**Nombre: Germán Darío Sánchez Zagal**

**Institución: Universidad Tecnológica de Tijuana**

**Nombre de trabajo: Portafolio de evidencias**

**Grupo: 3:D**

**Docente: Ray Brunett Parra Galaviz**

**Materia: Desarrollo de Aplicaciones Web**

**Carrera: Desarrollo y gestión de software**

**Fecha de entrega: 31/10/2024**

## Contents

PHP (Hypertext Preprocessor): .....	3
Súper Globales \$GET Y \$POST .....	5
Box Model CSS .....	6
Lenguaje Marcado y de Presentación .....	7
Mapa Mental Tecnologías y Herramientas para el desarrollo WEB .....	9
Configuración del ambiente para desarrollo web .....	10
Herramientas para el desarrollo web .....	12
Tecnologías para la creación de apps web .....	13

## **PHP (Hypertext Preprocessor):**

- Descripción: PHP es un lenguaje de programación de código abierto diseñado principalmente para el desarrollo de aplicaciones web del lado del servidor. Se integra fácilmente con HTML, lo que lo hace ideal para generar contenido dinámico en sitios web. PHP es ampliamente utilizado por su simplicidad, flexibilidad y capacidad para interactuar con bases de datos, crear formularios, gestionar sesiones, y realizar tareas del servidor como el manejo de archivos.

- Características clave:

- o Código abierto: Es gratuito y de código abierto, lo que permite que sea accesible para cualquier desarrollador.

- o Lado del servidor: PHP se ejecuta en el servidor, lo que significa que el código PHP no es visible para los usuarios finales; sólo ven el HTML generado.

- o Interacción con bases de datos: PHP se conecta fácilmente con bases de datos como MySQL, PostgreSQL, SQLite, entre otros.

- o Amplia compatibilidad: Funciona en varios servidores web (como Apache, Nginx) y sistemas operativos (Linux, Windows, macOS).

- o Simplicidad: Tiene una sintaxis sencilla que es fácil de aprender para principiantes, pero lo suficientemente poderosa para proyectos más avanzados.

## **2. MySQL:**

- Descripción: MySQL es un sistema de gestión de bases de datos relacional (RDBMS) que utiliza el lenguaje SQL (Structured Query Language) para gestionar y organizar los datos. Almacena datos en tablas que pueden estar relacionadas entre sí, facilitando la organización y recuperación eficiente de información. MySQL es de código abierto, lo que lo hace popular en proyectos tanto comerciales como no comerciales, y es una de las bases de datos más

utilizadas en aplicaciones web.

- **Características clave:**

- o Sistema relacional: Los datos se almacenan en tablas, y las relaciones entre ellas se manejan mediante claves primarias y foráneas.

- o Rápido y eficiente: MySQL está diseñado para ser rápido y eficiente, lo que lo convierte en una opción ideal para aplicaciones web que manejan grandes cantidades de datos.

- o Soporte para múltiples usuarios: MySQL permite que varios usuarios accedan y gestionen la base de datos simultáneamente.

- o Escalabilidad: Es capaz de manejar desde pequeñas aplicaciones hasta grandes sistemas con millones de registros.

- o Seguridad: Ofrece un sistema robusto de control de acceso y autenticación de usuarios para proteger los datos.

## Súper Globales \$GET Y \$POST

Las superglobales en PHP son variables predefinidas que están disponibles en todos los ámbitos de un

script. Estas variables permiten acceder a datos de forma rápida sin la necesidad de definir su alcance, ya

que son accesibles desde cualquier parte del código.

\$GET y \$POST son dos de estas superglobales:

### 1. \$GET:

o Se utiliza para recopilar datos enviados a través de la URL o de un formulario con el **método GET**.

o Los datos enviados por este método se anexan a la URL, lo que significa que son visibles

para los usuarios y tienen una limitación en cuanto a la cantidad de datos que pueden enviarse.

### 2. \$POST:

- Se utiliza para recopilar datos enviados a través de un formulario con el método POST.

- Los datos se envían de manera oculta en el cuerpo de la solicitud HTTP, lo que significa

que no se anexan a la URL y no tienen una limitación estricta de tama

## Box Model CSS

El box model de CSS es un concepto fundamental para entender cómo se renderizan los elementos en una página web. Es una forma de describir cómo los elementos HTML

son estructurados y visualizados en la pantalla. Cada elemento se puede imaginar como una caja rectangular que está compuesta por varias partes. Aquí están los componentes del box model:

1. Contenido (Content): Esta es el área donde se muestra el texto o las imágenes del elemento. Es el núcleo del box model y define el tamaño del contenido del elemento.
2. Relleno (Padding): El área de relleno es el espacio entre el contenido del elemento y su borde. El padding es transparente y se puede ajustar en todos los lados (arriba, derecha, abajo, izquierda) de forma independiente o conjunta.
3. Borde (Border): El borde rodea el área de relleno y el contenido. Puedes definir el grosor, el estilo (sólido, punteado, etc.) y el color del borde.
4. Margen (Margin): El margen es el espacio entre el borde del elemento y los elementos adyacentes. Al igual que el padding, el margen puede ser ajustado en todos los lados de forma independiente o conjunta.

### Cómo se define en CSS

Puedes ajustar cada una de estas partes del box model usando propiedades CSS:

- Contenido: El tamaño del contenido se define con las propiedades width y height.
- Relleno: Controlado con padding. Ejemplo: padding: 10px;
- Borde: Controlado con border. Ejemplo: border: 1px solid black;
- Margen: Controlado con margin. Ejemplo: margin: 20px;

## Modelos de Box

En CSS, puedes cambiar cómo se calculan las dimensiones de la caja usando la propiedad `box-sizing`. Por defecto, el modelo de caja se basa en el contenido, pero puedes cambiarlo a `border-box` para que el padding y el borde se incluyan en las dimensiones definidas del elemento. Esto asegura que el ancho y alto especificado para un elemento incluye el padding y el borde, haciendo más sencillo el cálculo de tamaños.

## Lenguaje Marcado y de Presentación

Los lenguajes de marcado y de presentación son esenciales en el desarrollo web y en la

creación de documentos estructurados. Aquí te presento un resumen sobre ambos conceptos, basado en información de diversas fuentes:

### Lenguajes de Marcado

Los lenguajes de marcado son sistemas de notación utilizados para estructurar y etiquetar contenido en un documento. Estos lenguajes definen la estructura y el significado del contenido, permitiendo que los elementos se identifiquen y se manipulen de manera específica. Algunos ejemplos comunes incluyen:

**1. HTML (Hypertext Markup Language):** Es el lenguaje de marcado estándar utilizado para crear páginas web. HTML define la estructura básica de una página, permitiendo incluir textos, imágenes, enlaces, y otros elementos multimedia. Los navegadores web interpretan HTML para mostrar el contenido al usuario.

**2. XML (Extensible Markup Language):** A diferencia de HTML, XML no tiene etiquetas predefinidas. Se utiliza para definir reglas para la codificación de documentos en un formato que sea tanto legible para humanos como para

máquinas. XML es ampliamente utilizado en aplicaciones donde la portabilidad y la interoperabilidad son cruciales.

**3. Markdown:** Es un lenguaje de marcado ligero que se usa principalmente para formatear texto en documentos y en plataformas como GitHub. Es fácil de leer en su forma sin procesar, lo que lo hace popular para la documentación y las notas rápidas.

### Lenguajes de Presentación

Mientras que los lenguajes de marcado se centran en la estructura y el contenido, los lenguajes de presentación se enfocan en cómo ese contenido se presenta visualmente a los usuarios. Estos lenguajes permiten definir estilos, colores, y diseños específicos para el contenido marcado. Algunos ejemplos incluyen:

**1. CSS (Cascading Style Sheets):** CSS es el lenguaje de presentación más comúnmente utilizado en la web. Permite a los desarrolladores aplicar estilos a los elementos HTML, como colores, fuentes, márgenes, y posiciones. CSS trabaja en conjunto con HTML para separar la estructura del contenido de su presentación.

**2. XSL (Extensible Stylesheet Language):** XSL es una familia de lenguajes usados para transformar y renderear documentos XML. Uno de sus componentes, XSLT (XSL Transformations), se utiliza para transformar documentos XML en otros formatos, como HTML, para la presentación.

**3. Sass/SCSS:** Son lenguajes de presentación que extienden las capacidades de CSS, permitiendo el uso de variables, anidación de reglas, y otras características avanzadas que hacen más eficiente la escritura de estilos complejos.

### Relación entre Lenguajes de Marcado y Presentación

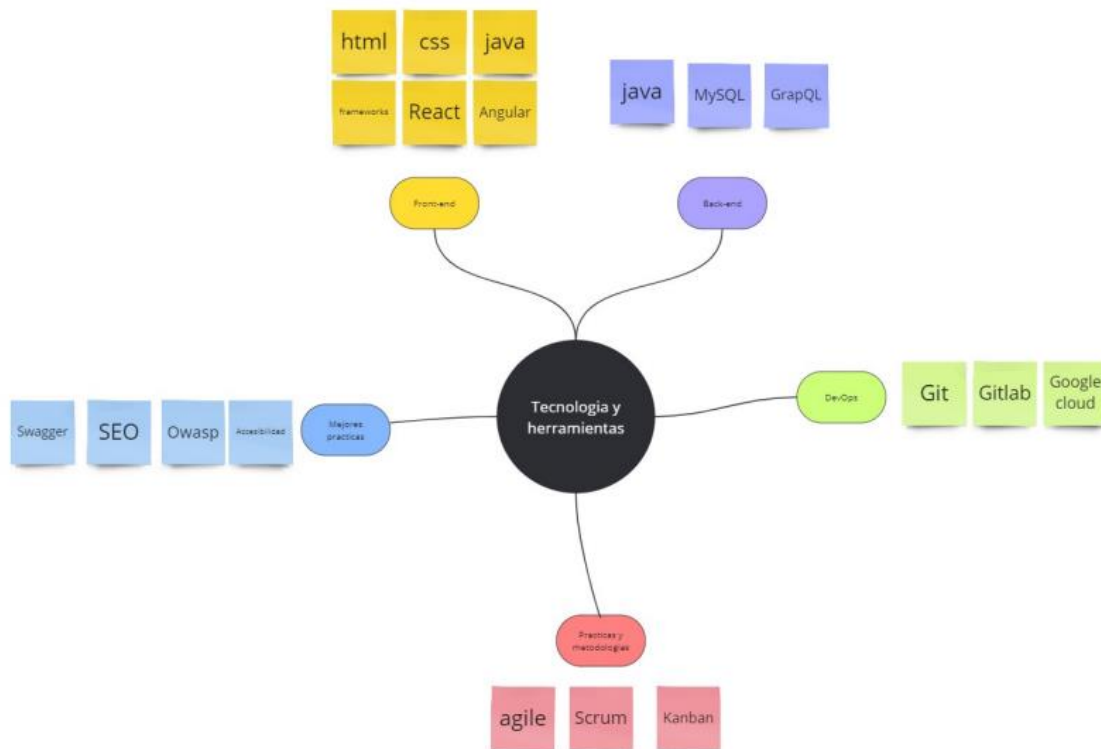
En el desarrollo web, los lenguajes de marcado y de presentación trabajan juntos para



crear experiencias web completas. Por ejemplo, HTML define la estructura y el contenido

de una página web, mientras que CSS se usa para estilizar ese contenido, asegurando que se vea de la manera deseada en diferentes dispositivos y tamaños de pantalla.

## Mapa Mental Tecnologías y Herramientas para el desarrollo WEB



## Configuración del ambiente para desarrollo web

### ¿Qué es un ambiente de para desarrollo web?

En lo que refiere a desarrollo web, la definición más sencilla y clara que podríamos aportar de un entorno de desarrollo, es que es igual a un servidor web, con ciertos recursos asignados, software y librerías instalados, y una configuración determinada. Entonces podríamos decir que un servidor es una aplicación que es capaz de atender las peticiones generadas por un cliente (en la mayoría de los casos se puede entender esto como un navegador web como Firefox, Google Chrome, Safari, u otro) y devolverle una respuesta en concordancia.

Dicho claramente, un servidor es el entorno de desarrollo que un programador de aplicaciones web requiere de forma obligatoria y que le permite escribir su código, diseñar y maquetar una interface, generar y consultar resultados de diversa índole, y servir o visualizar los cambios.

En nuestro caso, un entorno de desarrollo en su versión más básica, suele conformarse de:

Un servidor web, que usualmente es Apache o Nginx.

Librerías de PHP y módulos necesarios, que es el lenguaje de programación utilizado para ejecutar Wordpress y OpenCart.

Una base de datos, que suele ser MySQL pero que podría ser cualquier otra.

Más allá de estas características básicas, los procesos que las definen, sus características técnicas y los softwares, librerías y versiones de estas, que utilizamos para estructurar un entorno, pueden variar en función de las características de cada proyecto.

Esto nos permite desarrollar aplicaciones web de una forma segura, libre de errores, con entornos diferenciados para realizar la programación, realizar pruebas, compartir resultados con los clientes y permitirles realizar pruebas y prácticas, y finalmente publicar una aplicación al alcance de cualquier usuario de internet, libre de errores, robusta, estable.

## Niveles de un entorno de desarrollo

Es útil y adecuado definir diferentes “niveles de entorno” o servidores utilizados en el desarrollo de una aplicación web. Cada uno de ellos son utilizados con un fin específico, y presentan ciertas ventajas sobre los otros en determinado momento del proceso de trabajo. Podemos decir entonces que es conveniente distinguir de: a) entorno de desarrollo, b) entorno de pruebas, c) entorno de pre-producción y d) entorno de producción.

## Movimientos de la aplicación entre los niveles de entornos

Lo habitual para desplazar el código de las aplicaciones, entre los diferentes niveles de entornos de desarrollo, es utilizar sistemas de integración continua basados en GIT. Git es un sistema de subversiones de código. Una vez configurado e implementado en el entorno de desarrollo, permite rastrear todos y cada uno de los cambios que se realizan en el código, y definir puntos de anclaje en esos cambios, los cuales se denominan “commits”. Por otro lado, un sistema de integración continua, permite entre otras cosas, facilitar el despliegue de los cambios en cada entorno, desde un repositorio GIT, de forma ordenada y predecible.

Lo habitual es que los desarrolladores de aplicaciones web escriban código en el entorno de desarrollo y desplacen ese código mediante “commits” a entornos de pruebas (o de pre-producción, cuando el entorno de pruebas sea innecesario o excesivo para la aplicación que estamos trabajando). El entorno de pruebas permite realizar pruebas avanzadas en la aplicación y también a otros miembros del equipo, realizar pruebas. Contar con un equipo QA o de testers, formados por miembros internos del staff y revisores externos (por ejemplo, usuarios clave), permite probar la aplicación en profundidad. Los reportes QA se envían de vuelta a los desarrolladores, quienes se encargan de resolver los errores (“bugs”). Luego de que todos los errores son resueltos, se entrega una nueva versión en testing.

Este proceso continúa hasta que el equipo QA declara que la versión en testing está “lista para ser liberada a producción”. El administrador de entornos (en ocasiones llamado “release manager”) eleva la versión que se encuentra a los servidores de pre-producción y en ausencia de errores en el despliegue, finalmente a los servidores en producción para ser consumidos por la audiencia.

Lo usual, es que a medida que avanza el tiempo, se hacen pedidos de nuevas características, o cambios en las funcionalidades de la aplicación, con lo cual los desarrolladores escriben nuevo código y comienza nuevamente el ciclo de desarrollo-pruebas-preproducción-producción.

## Herramientas para el desarrollo web

### Recursos para el Desarrollo Web de Investigación

A continuación, se presentan algunas herramientas relevantes para el desarrollo web de la investigación:

**Editores de código:** Visual Studio Code, conocido por su velocidad y flexibilidad, es una opción popular entre los desarrolladores web.

**Control de versiones:** Git es una herramienta fundamental para el control de versiones en el desarrollo web.

**Diseño y prototipado:** Bootstrap es una biblioteca de código abierto para el diseño y prototipado de sitios web.

**Desarrollo eficiente:** PhpStorm es un entorno de desarrollo integrado (IDE) que ofrece características avanzadas para el desarrollo web.

### Herramientas para la recopilación, interpretación y difusión de datos

**Portales de información:** El portal web del Ministerio de Inclusión, Seguridad Social y Migraciones (Gobierno de España) ofrece información sobre trámites y procedimientos en materia de extranjería, normativa y programas de integración.

**Ranking de revistas científicas:** Scimago Journal & Country Rank (SJR) es un portal académico que proporciona información sobre revistas científicas, incluyendo su impacto y clasificación por área temática y país.

**Bases de datos de investigación:** La base de datos Scopus es una fuente importante para la investigación en ciencias sociales y humanas, y puede ser utilizada para buscar y analizar artículos científicos.

### Otras herramientas

**Repositorio de recursos educativos:** La base de datos de recursos educativos de la Universidad de Barcelona contiene información sobre investigación, innovación, recursos didácticos y revistas iberoamericanas.

**Guía de publicación científica:** IEEE Publishing offers a step-by-step guide for scientists to publish their work in English-language journals and conferences.

Estas herramientas pueden ser útiles para investigadores y desarrolladores web para realizar tareas de investigación, desarrollo y publicación de resultados.

## Tecnologías para la creación de apps web

Para crear aplicaciones web, se utilizan diversas tecnologías que abarcan tanto el frontend (la parte visible para el usuario) como el backend (la parte que maneja la lógica del servidor). Aquí tienes un resumen de las tecnologías más comunes:

### 1. Frontend

**HTML** (HyperText Markup Language): Es el lenguaje de marcado que estructura el contenido de la web.

**CSS** (Cascading Style Sheets): Se utiliza para diseñar y dar estilo a los elementos creados con HTML.

**JavaScript**: Es un lenguaje de programación que permite añadir interactividad a la web.

#### **Frameworks y librerías de JavaScript:**

**React**: Librería desarrollada por Facebook para construir interfaces de usuario.

**Angular**: Framework desarrollado por Google para crear aplicaciones web dinámicas.

**Vue.js**: Un framework progresivo para construir interfaces de usuario.

### 2. Backend

#### **Lenguajes de programación:**

**JavaScript (Node.js)**: Permite ejecutar JavaScript en el servidor.

**Python (Django, Flask)**: Python es conocido por su simplicidad y potencia. Django es un framework completo, mientras que Flask es más ligero.

**Ruby (Ruby on Rails)**: Framework que facilita el desarrollo rápido de aplicaciones.

**PHP**: Un lenguaje ampliamente utilizado en el desarrollo web, conocido por ser fácil de aprender y muy compatible con servidores web.

**Java (Spring)**: Java es un lenguaje robusto, y Spring es un framework muy utilizado para aplicaciones empresariales.

#### **Bases de datos:**

**SQL (MySQL, PostgreSQL)**: Son bases de datos relacionales que utilizan el lenguaje SQL.

Sanchez Zagal German Dario

NoSQL (MongoDB, Firebase): Bases de datos no relacionales, ideales para manejar grandes volúmenes de datos no estructurados.

Servidores web:

Apache: Uno de los servidores web más utilizados.

Nginx: Conocido por su eficiencia y capacidad para manejar grandes cantidades de tráfico.

### 3. Full Stack Frameworks

Meteor: Un framework que permite desarrollar tanto el frontend como el backend con JavaScript.

MEAN/MERN Stack:

MEAN: MongoDB, Express.js, Angular, Node.js.

MERN: MongoDB, Express.js, React, Node.js.

### 4. Herramientas de desarrollo y control de versiones

Git: Sistema de control de versiones para rastrear cambios en el código.

GitHub/GitLab/Bitbucket: Plataformas para alojar y colaborar en proyectos de software.

Webpack: Un módulo que agrupa archivos y dependencias de un proyecto.

Docker: Permite la creación de contenedores que encapsulan aplicaciones y sus dependencias.

### 5. Servicios y APIs

RESTful APIs: Arquitectura que permite la comunicación entre sistemas usando HTTP.

GraphQL: Un lenguaje de consulta para APIs, que permite solicitar exactamente los datos necesarios.

Servicios en la nube:

AWS (Amazon Web Services): Ofrece infraestructura y servicios en la nube. Google

Cloud Platform (GCP): Servicios en la nube ofrecidos por Google. Microsoft Azure:

Servicios en la nube de Microsoft.

Estas tecnologías forman la base de muchas aplicaciones web modernas, permitiendo desde la creación de simples sitios web hasta complejas aplicaciones empresariales