

## **CONCEPTOS**

### **PÁGINA WEB**

Una página web es un documento electrónico que forma parte de un sitio web y se encuentra alojado en un servidor web accesible a través de Internet. Es una representación digital de contenido que puede contener texto, imágenes, videos, enlaces, y otros elementos multimedia, organizados en un diseño visual y estructurado.

Las páginas web son accesibles a través de un navegador web, como Google Chrome, Mozilla Firefox, Safari, o Microsoft Edge, y se pueden visualizar en dispositivos como computadoras, tabletas, teléfonos inteligentes y otros dispositivos con conexión a Internet.

Las páginas web se crean utilizando lenguajes de marcado, como HTML (Hypertext Markup Language), que define la estructura y el contenido de la página, y CSS (Cascading Style Sheets), que define la apariencia y el estilo de la página. También pueden incluir código JavaScript para agregar interactividad y funcionalidades dinámicas a la página.

Las páginas web se utilizan para diversos fines, como proporcionar información, presentar productos o servicios, compartir contenido multimedia, realizar transacciones comerciales, ofrecer servicios en línea, interactuar con usuarios, y mucho más. Las páginas web son la base de Internet y permiten a las personas acceder a una amplia gama de información y servicios en línea.

### **APLICACIÓN WEB**

Una aplicación web es un tipo de software que se ejecuta en un servidor web y se accede a través de un navegador web en lugar de instalarse en un dispositivo local, como un ordenador o un dispositivo móvil. Las aplicaciones web utilizan tecnologías web, como HTML, CSS y JavaScript, para ofrecer funcionalidades interactivas y dinámicas a los usuarios a través de una interfaz de usuario basada en un navegador.

A diferencia de las aplicaciones de escritorio que se instalan en un dispositivo local y funcionan de forma independiente, las aplicaciones web se ejecutan en un servidor remoto y se acceden a través de Internet utilizando un navegador web. Los usuarios interactúan con la aplicación web a través de la interfaz de usuario proporcionada por el navegador, y todas las operaciones y procedimientos se realizan en el servidor web.

Las aplicaciones web pueden ser de diversos tipos y tamaños, desde aplicaciones empresariales y de productividad, hasta aplicaciones de comercio electrónico, redes sociales, herramientas de colaboración, servicios en línea, y muchas otras. Pueden ofrecer funcionalidades como registro y autenticación de usuarios, gestión de bases de datos, procesamiento de formularios, generación de contenido dinámico, integración con APIs, y más.

## **DIFERENCIA ENTRE PAGINA WEB Y APLICACIÓN WEB**

La diferencia principal entre una página web y una aplicación web radica en su funcionalidad y complejidad.

Una página web generalmente se refiere a un solo documento electrónico, que forma parte de un sitio web más grande y que contiene contenido estático o semi estático, como texto, imágenes, enlaces y otros elementos multimedia.

Por otro lado, una aplicación web es un tipo de software interactivo y dinámico que se ejecuta en un servidor web y se accede a través de un navegador web.

## **FRAMEWORK**

Un framework, en el contexto del desarrollo de software, es un conjunto de herramientas, librerías, convenciones y patrones de diseño que proporcionan una estructura y un conjunto de funcionalidades predefinidas para facilitar la creación de aplicaciones y sistemas de software. Un framework proporciona una base sobre la cual los desarrolladores pueden construir su propio software, aprovechando las funcionalidades y abstracciones proporcionadas por el framework, lo que les permite desarrollar aplicaciones de manera más rápida y eficiente.

Un framework generalmente define una arquitectura y una estructura de directorios para organizar el código fuente, así como una serie de reglas y convenciones para el diseño de código y la interacción entre los diferentes componentes del sistema. También puede incluir librerías y utilidades que facilitan tareas comunes, como manejo de bases de datos, manipulación de interfaces de usuario, gestión de sesiones, manejo de errores y más. Los frameworks suelen estar escritos en un lenguaje de programación específico y se utilizan para desarrollar aplicaciones en ese lenguaje.

Los frameworks son utilizados por los desarrolladores para acelerar el proceso de desarrollo, ya que proporcionan una base sólida y probada sobre la cual se puede construir un software personalizado. Al utilizar un framework, los desarrolladores pueden aprovechar el trabajo previo realizado por otros, evitando tener que reinventar la rueda y escribir código desde cero. Los frameworks también suelen seguir mejores prácticas y patrones de diseño establecidos, lo que puede ayudar a mejorar la calidad y la mantenibilidad del código desarrollado.

## **LIBRERÍA**

En el contexto del desarrollo de software, una librería (también conocida como biblioteca) es un conjunto de código preescrito y precompilado que contiene una colección de funciones, clases o rutinas que pueden ser utilizadas por los desarrolladores para realizar tareas específicas en sus programas. Una librería es un conjunto de código que ha sido creado y probado previamente por otros desarrolladores y se distribuye para su reutilización, lo que

permite a los desarrolladores evitar tener que escribir código desde cero y acelerar el proceso de desarrollo.

Las librerías se utilizan para proporcionar funcionalidades específicas a un programa sin tener que implementarlas desde cero. Las librerías suelen estar escritas en un lenguaje de programación específico y pueden abarcar una amplia variedad de áreas, como manejo de bases de datos, manipulación de archivos, manipulación de imágenes, procesamiento de datos, interfaz de usuario, criptografía, y muchos otros.

## **DIFERENCIA ENTRE FRAMEWORK Y LIBRERÍA**

La diferencia principal entre una librería y un framework radica en cómo son utilizados y cómo interactúan con el código de una aplicación.

**Funcionalidad:** Una librería es un conjunto de código preescrito que proporciona funcionalidades específicas que pueden ser utilizadas por los desarrolladores para realizar tareas específicas en sus programas. Las librerías son piezas independientes de código que se pueden usar según las necesidades del desarrollador. En contraste, un framework es un conjunto de herramientas, convenciones y patrones de diseño que proporciona una estructura y una funcionalidad más amplia para el desarrollo de aplicaciones. Un framework incluye una arquitectura y una estructura predefinida, y los desarrolladores deben seguir las convenciones y patrones de diseño del framework para desarrollar una aplicación en particular.

**Control del flujo de ejecución:** En una librería, el desarrollador tiene un mayor control sobre el flujo de ejecución de su aplicación. El desarrollador decide cómo y cuándo utilizar las funciones o clases proporcionadas por la librería en su código. En cambio, en un framework, el flujo de ejecución de la aplicación es controlado por el propio framework. El desarrollador debe seguir las reglas y convenciones del framework y adaptar su código para que funcione dentro de la estructura proporcionada por el framework.

**Nivel de abstracción:** Una librería proporciona una funcionalidad específica y concreta que se puede utilizar en cualquier parte del código de una aplicación. En contraste, un framework proporciona una estructura y un conjunto de funcionalidades más amplias que definen cómo se desarrolla una aplicación en particular. Un framework suele tener un mayor nivel de abstracción, lo que significa que proporciona una visión más general y abstracta de cómo se desarrolla una aplicación, mientras que una librería es más específica y concreta en términos de la funcionalidad que ofrece.

**Flexibilidad y personalización:** Las librerías suelen ser más flexibles y permiten a los desarrolladores personalizar su uso en mayor medida. Los desarrolladores pueden elegir qué funciones o clases de la librería utilizar y cómo integrarlas en su código. En cambio, los

frameworks suelen ser más rígidos en términos de su estructura y convenciones, lo que puede limitar la flexibilidad y personalización en comparación con las librerías.

En resumen, una librería es un conjunto de código preescrito que proporciona funcionalidades específicas para ser utilizadas por los desarrolladores, mientras que un framework es un conjunto de herramientas, convenciones y patrones de diseño que proporciona una estructura y una funcionalidad más amplia para el desarrollo de aplicaciones. Las librerías ofrecen más flexibilidad y control sobre el flujo de ejecución, mientras que los frameworks proporcionan una estructura y un conjunto de funcionalidades más amplias, pero pueden ser más rígidos en términos de convenciones y estructura. Ambos son importantes en el desarrollo de software y se utilizan en diferentes contextos según las necesidades del proyecto y las preferencias del desarrollador.

## **HTML**

HTML (Hypertext Markup Language) es el lenguaje de marcado estándar utilizado para crear páginas web. Es el componente básico y fundamental de la web, utilizado para describir la estructura y el contenido de una página web. HTML utiliza una serie de etiquetas (también conocidas como elementos) para marcar el contenido de una página web, definiendo la estructura del documento, la presentación y los elementos multimedia que se muestran en un navegador web.

## **CSS**

CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para describir la presentación visual de una página web escrita en HTML u otros lenguajes de marcado. CSS permite a los desarrolladores web aplicar estilos, diseños y formatos a las páginas web, lo que les permite tener un control preciso sobre la apariencia y el diseño de un sitio web.

Con CSS, los desarrolladores pueden definir cómo se deben mostrar los elementos HTML en una página web, como colores, fuentes, márgenes, tamaños, alineaciones y efectos visuales. CSS utiliza una sintaxis de reglas de estilo que se aplican a elementos HTML específicos o a grupos de elementos, lo que permite una separación clara entre el contenido y la presentación visual de una página web.

## **JAVASCRIPT**

JavaScript es un lenguaje de programación ampliamente utilizado que se utiliza principalmente para agregar interacción y funcionalidad dinámica a las páginas web. Es un lenguaje de programación del lado del cliente, lo que significa que se ejecuta en el navegador web del usuario final y permite la manipulación de contenido y la interacción del usuario en tiempo real en una página web.

JavaScript se utiliza para crear efectos visuales, interactuar con el usuario, manipular y modificar el contenido de una página web en tiempo real, validar formularios, realizar

solicitudes de red para obtener datos, crear aplicaciones web y mucho más. Con JavaScript, los desarrolladores web pueden agregar interactividad, funcionalidad y dinamismo a una página web, lo que permite una experiencia de usuario más rica y personalizada.

## **PHP**

PHP es un lenguaje de programación de código abierto ampliamente utilizado y especialmente diseñado para el desarrollo de aplicaciones web del lado del servidor. PHP es un acrónimo recursivo que significa "PHP: Hypertext Preprocessor", lo que indica su enfoque original en la generación de contenido dinámico para la web.

PHP se ejecuta en el servidor y se utiliza para procesar la lógica del lado del servidor en una aplicación web, lo que significa que el código PHP se ejecuta en el servidor antes de que se envíe el contenido HTML al navegador del usuario. PHP permite a los desarrolladores web crear aplicaciones web dinámicas, interactivas y personalizadas, como sistemas de gestión de contenidos (CMS), foros, blogs, aplicaciones de comercio electrónico, y mucho más.

## **FLIGHT PHP**

"Flight PHP" es un microframework de código abierto para el desarrollo de aplicaciones web en PHP. Un microframework es una versión minimalista de un framework, que se centra en proporcionar solo lo esencial para construir aplicaciones web, sin las características y funcionalidades adicionales que se encuentran en frameworks más completos.

Flight PHP se caracteriza por ser ligero y fácil de usar, lo que lo convierte en una opción popular para desarrolladores que desean una solución minimalista y rápida para el desarrollo de aplicaciones web en PHP. Fue creado por Mike Cao en 2010 y está diseñado para ser simple y eficiente, lo que lo hace adecuado para proyectos pequeños o medianos que no requieren la complejidad y la abstracción de un framework completo.

Algunas de las características y beneficios de Flight PHP incluyen:

Estructura simple: Flight PHP utiliza una estructura simple y minimalista que permite a los desarrolladores tener un mayor control sobre su código y su flujo de ejecución.

Enrutamiento: Flight PHP ofrece un enrutamiento simple y eficiente, lo que permite definir fácilmente las rutas y las acciones que se ejecutan en respuesta a las URL solicitadas.

Manejo de peticiones y respuestas: Flight PHP proporciona métodos sencillos para manejar peticiones HTTP, así como para generar respuestas en diferentes formatos, como HTML, JSON, XML, entre otros.

**Flexibilidad:** Al ser un microframework, Flight PHP no impone una estructura o arquitectura específica para la organización del código, lo que permite a los desarrolladores adaptarlo a sus necesidades específicas.

**Extensibilidad:** Aunque es un microframework, Flight PHP permite agregar funcionalidades adicionales mediante la adición de extensiones o librerías externas.

En resumen, Flight PHP es un microframework de código abierto para el desarrollo de aplicaciones web en PHP, que se caracteriza por ser ligero, simple y fácil de usar. Es una opción popular para proyectos pequeños o medianos que buscan una solución minimalista y eficiente para el desarrollo de aplicaciones web en PHP.

## **ANGULAR**

Angular es un framework de código abierto desarrollado por Google para la construcción de aplicaciones web de una sola página (SPA, por sus siglas en inglés) y aplicaciones web dinámicas. Angular permite a los desarrolladores crear aplicaciones web complejas y ricas en funcionalidades con una arquitectura modular y escalable.

Angular utiliza un enfoque basado en componentes para la construcción de aplicaciones web, donde los componentes son bloques de construcción reutilizables que encapsulan la lógica y la interfaz de usuario de una parte específica de la aplicación. Los componentes de Angular se comunican entre sí a través de propiedades y eventos, lo que permite una separación clara de la lógica de presentación y la lógica de negocio.

Algunas de las características principales de Angular incluyen:

**Enrutamiento:** Angular ofrece un sistema de enrutamiento integrado que permite crear rutas y manejar la navegación entre las diferentes vistas de una SPA de forma eficiente.

**Inyección de dependencias:** Angular tiene un poderoso sistema de inyección de dependencias que permite manejar la gestión de dependencias de manera fácil y organizada.

**Binding de datos:** Angular permite la vinculación bidireccional de datos entre la vista y el modelo de datos, lo que facilita la actualización automática de la interfaz de usuario en función de los cambios en los datos.

**Directivas:** Angular proporciona un sistema de directivas que permite extender y personalizar la sintaxis de HTML para crear componentes personalizados y reutilizables.

Testing: Angular tiene un enfoque fuerte en la calidad del código y ofrece herramientas integradas para realizar pruebas unitarias y de integración en las aplicaciones desarrolladas con Angular.

Integración con RESTful APIs: Angular ofrece una integración nativa con APIs RESTful, lo que facilita la comunicación con servicios web y la manipulación de datos en una aplicación web.