**Ejercicios básicos DWEC - Hoja 1.1 TERESA ESTEBAN**

1. ¿Qué es una aplicación Web?

Las aplicaciones web son programas informáticos que funcionan en un navegador web y se pueden usar desde cualquier lugar con acceso a Internet. Son accesibles tanto en redes internas como en la web pública y no requieren instalación en dispositivos locales. Esto las hace flexibles y fáciles de usar.

1. Explica brevemente el funcionamiento del protocolo HTTP

El usuario accede a una URL haciendo clic en un enlace o escribiéndola directamente en el navegador.

El navegador divide la URL en partes como el tipo de protocolo, la dirección IP del servidor, el puerto y el recurso solicitado.

Se establece una conexión con el servidor a través de una conexión TCP/IP en el puerto adecuado.

El navegador envía una solicitud al servidor, incluyendo el tipo de comando (GET, POST, HEAD), la URL y la versión de HTTP utilizada, así como otra información opcional.

El servidor responde con un código de estado, el tipo de dato MIME y la información solicitada.

La conexión TCP se cierra.

Este proceso se repite para cada acceso al servidor HTTP, por ejemplo, cuando se cargan imágenes en una página web.

Los tipos de mensajes principales que utiliza HTTP son GET (para obtener información), POST (para enviar información al servidor, como datos de un formulario) y HEAD (para obtener información sobre un objeto, como su tipo o tamaño)..

1. Funcionamiento básico de un navegador Web.

El navegador web solicita al servidor los recursos web que el usuario elige y los presenta en una ventana.

Estos recursos suelen ser documentos codificados en HTML, pero también pueden ser archivos en otros formatos como PDF, Word, audio, imágenes, entre otros.

El usuario indica la ubicación del recurso utilizando una dirección URI (Uniform Resource Identifier) o identificador.

1. Estructura de una URI.

Una URI, o Identificador Uniforme de Recursos, es una especie de "etiqueta" que se utiliza en internet para identificar de manera única un recurso, como una página web, un archivo, o cualquier cosa que se pueda acceder en línea. Piénsalo como una dirección única que te lleva a un lugar específico en la web.

La URI tiene una estructura que incluye varias partes importantes. Primero, está el "esquema", que indica cómo se debe acceder al recurso. Por ejemplo, cuando ves "http://" o "https://" en una URI, eso te dice que estás accediendo a través de un navegador web usando el protocolo HTTP o HTTPS.

Luego, viene la "parte jerárquica", que es la dirección o ubicación del recurso. Por ejemplo, cuando ves el nombre de dominio de un sitio web en la URI, como "[www.ejemplo.com](http://www.ejemplo.com/)", eso te dice dónde encontrar el recurso en la red.

Después, hay una "solicitud", que es opcional. Esta parte puede contener información adicional que el servidor necesita para entender lo que estás buscando. Por ejemplo, en una URI de búsqueda en un motor de búsqueda, la solicitud podría incluir las palabras clave que estás buscando.

Finalmente, tenemos el "fragmento", que también es opcional. Esto se utiliza para señalar una parte específica dentro del recurso. Por ejemplo, en una URI que te lleva a un artículo largo, el fragmento podría indicar la sección exacta que quieres ver.

1. Explica los subsistemas de la arquitectura de referencia de un navegador Web.

La arquitectura de un navegador web se divide en varias partes:

Interfaz de Usuario: Es la parte que interactúa con el usuario, como los botones y ventanas.

Motor de Búsqueda: Proporciona una forma de buscar y acceder a sitios web.

Renderizado: Crea una representación visual de las páginas web que visitas.

Comunicaciones: Se encarga de la transferencia de datos desde y hacia la web utilizando protocolos como HTTP y FTP.

En la segunda parte:

JavaScript: Lee y ejecuta código JavaScript en las páginas web.

Parser XML(dom): Carga y organiza la estructura de las páginas web.

Visualización: Muestra el contenido de las páginas web en tu pantalla.

Persistencia de Datos: Almacena información necesaria para el funcionamiento del navegador.

1. Diferencias básicas entre páginas Web estáticas y dinámicas.

Contenido:

Estáticas: El contenido de una página web estática es fijo y predefinido. No cambia a menos que se actualice manualmente el código HTML.

Dinámicas: El contenido de una página web dinámica se genera en tiempo real. Puede cambiar según las interacciones del usuario o la información almacenada en una base de datos.

Tecnología de Desarrollo:

Estáticas: Se crean principalmente utilizando HTML y CSS. El contenido y el diseño son estáticos en el archivo HTML.

Dinámicas: Requieren tecnologías adicionales como JavaScript, PHP, Python, Ruby, o sistemas de gestión de contenido (CMS) como WordPress, Drupal o Joomla. Estas tecnologías permiten la interacción y la generación de contenido en el servidor antes de enviarlo al navegador del usuario.

Carga de Página:

Estáticas: Cargan más rápidamente ya que el contenido está predefinido y no requiere procesamiento en el servidor antes de mostrarse.

Dinámicas: Pueden tener tiempos de carga más largos debido a la generación de contenido en tiempo real y a la necesidad de acceder a bases de datos u otros servicios.

Escalabilidad:

Estáticas: Son fáciles de escalar en términos de ancho de banda y servidores para manejar un mayor tráfico web, pero requieren actualizaciones manuales para cambiar el contenido.

Dinámicas: Son más escalables en términos de contenido y funcionalidad, ya que pueden gestionar grandes cantidades de datos y adaptarse a las necesidades cambiantes sin cambiar el código fuente.

Interactividad:

Estáticas: Tienen interacción limitada con el usuario. Por lo general, se utilizan para mostrar información estática o contenido informativo sin funciones interactivas complejas.

Dinámicas: Pueden ofrecer una amplia gama de interacciones, como formularios interactivos, comentarios de usuarios, tiendas en línea, y más, gracias a la ejecución de scripts en el servidor o en el navegador del usuario.

Mantenimiento:

Estáticas: Requieren menos mantenimiento continuo, ya que el contenido se actualiza manualmente modificando el código fuente.

Dinámicas: Pueden requerir más mantenimiento debido a la gestión de bases de datos, actualizaciones de software y seguridad, y cambios en la lógica de la aplicación web.

1. Diferencias básicas entre SPA y MPA.

Arquitectura:

SPA: En una SPA, toda la aplicación se carga en una sola página web. Las interacciones y la navegación entre diferentes secciones o vistas se gestionan mediante JavaScript, sin necesidad de recargar la página completa.

MPA: En una MPA, la aplicación web consta de múltiples páginas HTML independientes. Cada página representa una vista diferente y generalmente se carga completamente cuando se navega a ella.

Carga Inicial:

SPA: Una SPA carga una vez y, a partir de ahí, solo actualiza el contenido necesario sin recargar la página completa. Esto da como resultado tiempos de carga más rápidos después de la primera carga inicial.

MPA: Cada página de una MPA se carga por separado, lo que puede llevar a tiempos de carga más lentos al cambiar entre páginas, especialmente si son pesadas.

Interacción del Usuario:

SPA: Ofrece una experiencia de usuario más fluida y rápida, ya que las transiciones entre vistas son suaves y se pueden realizar acciones en la página sin recargarla.

MPA: Requiere recargar páginas completas al cambiar de una vista a otra, lo que puede resultar en una experiencia de usuario menos fluida.

SEO (Optimización para Motores de Búsqueda):

SPA: Puede presentar desafíos adicionales para el SEO debido a la carga dinámica de contenido. Se requiere un enfoque cuidadoso para garantizar la indexación adecuada por parte de los motores de búsqueda.

MPA: Las páginas individuales de una MPA son más fáciles de indexar para los motores de búsqueda, ya que cada página tiene su propia URL y contenido.

Desarrollo y Mantenimiento:

SPA: Requiere un enfoque más orientado al frontend y a menudo utiliza bibliotecas o frameworks como React, Angular o Vue.js. El desarrollo y la implementación pueden ser más rápidos, pero la gestión del estado y el SEO pueden ser más complejos.

MPA: Permite una estructura de desarrollo más tradicional, con páginas separadas que pueden ser desarrolladas de forma independiente. Es más fácil de entender y mantener, especialmente en aplicaciones grandes.

Requerimientos del Servidor:

SPA: Suelen requerir menos solicitudes al servidor en la navegación debido a la carga dinámica de contenido.

MPA: Generan más solicitudes al servidor, ya que cada página se carga por separado.

1. Explica las capas de una aplicación Web.

Estructura de la Web:

Esta capa se refiere a la organización y disposición general de una página web. Incluye la estructura HTML (HyperText Markup Language) de la página, que define los elementos básicos como encabezados, párrafos, listas y enlaces.

También abarca la creación de una jerarquía lógica de contenido, como la división en secciones, la organización de menús y la colocación de elementos como encabezados, pies de página y barras laterales.

En resumen, la estructura de la web se encarga de cómo se organiza y presenta el contenido en una página, proporcionando una base sólida para la presentación y el comportamiento.

Presentación de los Elementos de la Web:

Esta capa se relaciona con el diseño visual y la apariencia de una página web. Utiliza hojas de estilo CSS (Cascading Style Sheets) para definir el aspecto de los elementos HTML.

Aquí se controla la fuente, el tamaño, el color, el espaciado, el diseño de la página y otros aspectos visuales que hacen que la página sea atractiva y fácil de leer.

La presentación también incluye el uso de imágenes, gráficos y otros elementos multimedia para enriquecer la experiencia del usuario.

En resumen, la presentación de elementos se encarga de cómo se ven los elementos en la página web, mejorando la estética y la legibilidad.

Comportamiento de la Web:

Esta capa está relacionada con la interactividad y la funcionalidad de la página web. Se utiliza JavaScript principalmente para agregar comportamientos dinámicos.

Aquí se gestionan las acciones del usuario, como hacer clic en botones, enviar formularios, desplazarse por la página, etc.

También se encarga de las validaciones de entrada, la actualización de contenido en tiempo real, la animación y la interacción con el servidor para cargar datos o enviar información.

El comportamiento de la web garantiza que la página sea más que un simple documento estático y permita a los usuarios interactuar y realizar acciones específicas.

NOTA: Se debe subir un archivo con el formato “DWEC\_Apellido1\_Nombre\_Ejerc1.1.pdf”.