HITO DEL 1º TRIMESTRE DE LENGUAJE DE MARCAS

Alejandro Cortés Díaz

DISEÑO DE APLICACIONES MULTIPLATAFORMA – LENGUAJE DE MARCAS

Índice

DESARROLLO

Antes de crear el CV, lo primero que te solicitan es un trabajo de investigación académico por escrito sobre los puntos que se exponen a continuación y plasmarlo de tal forma que seas capaz de identificar fácilmente los temas y los contenidos asociados a esos temas. Se

recomienda tomar referencias de otras páginas web con temática similar. Se valorará el uso de elementos multimedia. Debéis consultar varias fuentes cuya veracidad sea demostrable y explicar con vuestras propias palabras cada uno de los apartados, incluyendo información adicional y complementaria sobre el enunciado del mismo. Debéis determinar una correcta estructura de las categorías más importantes en el apartado clasificación.

Debe contener:

1. Explica los orígenes de los lenguajes de marcas.

El concepto de lenguaje de marcas fue expuesto por primera vez en 1967 por William W. Tunnicliffe. La mayor novedad consistía en la separación entre la presentación y la estructura del texto.

Sin embargo, Charles Goldfarb, perteneciente en ese momento a la famosa empresa IBM, y considerado padre de los lenguajes de marcas, participó en el lenguaje GML, y posteriormente dirigió el equipo responsable de elaborar el estándar SGML (Standard Generalized Markup Language), una pieza clave de los lenguajes de marcas.

Tanto, que en los 80 fue desarrollado para estructurar y almacenar documentos en un formato digital, sentando las bases para el HTML (HyperText Markup Language), diseñado en el 91, tiempo después, por Tim Berners-Lee, con la intención de estructurar páginas web.

2. Explicar los diferentes estándares y organismos oficiales en el desarrollo y diseño de lenguajes de marcas.

Los estándares web se introdujeron para proteger el ecosistema web, mantenerlo abierto, gratuito, y accesible. Cuando estos fueron creados se alentó a los creadores de navegadores a adherirse a una forma estandarizada de construir las páginas. Lo que supuso que no hubiera necesidad de crear múltiples versiones del mismo sitio web.

Algunos de los estándares más conocidos y ampliamente utilizados son:

- **HTML** (HyperText Markup Language), para definir la estructura de los documentos.
- **XML** (eXtensible Markup Language), que sirve de base para un gran número de tecnologías.
- **CSS** (Cascading Style Sheets), que permite asignar estilos para la representación de los documentos.
- **Javascript**, que permite otorgar dinamismo y funcionalidad.

En este contexto de velar por una manera estandarizada de crear páginas web, tiene alta relevancia la World Wide Web, o W3C.

Esta consiste en una comunidad internacional donde las organizaciones miembros trabajan conjuntamente para desarrollar estándares web.

Este promueve el uso de estándares para reducir el coste y la complejidad del desarrollo, así como para incrementar la accesibilidad y viabilidad de cualquier documento publicado en la web.

De hecho, los navegadores actuales (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, etc.) tienen por tanto un serio compromiso con el cumplimiento de estos estándares.

El trabajo de la organización W3C hacia una web semántica está actualmente enfocado por publicaciones relacionadas al Marco de

Descripción de Recursos (RDF), Gleaning Resource Descriptions from Dialects of Languages (GRDDL) y Web Ontology Language (OWL).

Otras organizaciones son colaboradoras de la W3C en el desarrollo de html y otros estándares, como es el caso de WHATWG (Web Hypertext Application Technology Working Group). O ECMA International, responsable de estandarizar JavaScript a través del ECMAScript.

3. Explica los orígenes y las diferencias entre CSS y SCSS.

CSS (Cascading Style Sheets):

CSS fue introducido en 1996 como un lenguaje de estilo en cascada para diferenciar la estructura HTML del aspecto visual. CSS posibilita la aplicación de estilos tales como colores, tipografías y márgenes.

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".

CSS es el lenguaje de estilos que todos los navegadores comprenden para crear páginas web.

SCSS (Sassy CSS):

Para discutir el origen de SCSS, es fundamental mencionar SASS (Syntactically Awesome Stylesheets).

Es un lenguaje de hojas de estilo en cascada que fue concebido por Hampton Catlin y luego perfeccionado por Natalie Weizenbaum. Tras sus primeras versiones, Nathan Weizenbaum y Chris Eppstein han seguido ampliando Sass con SassScript, un lenguaje de guion sencillo, que se utiliza en los archivos Sass.

Sass se compone de dos tipos de sintaxis. La sintaxis original, conocida como indented syntax («sintaxis con sangrado»), emplea una estructura parecida a Haml. Esta utiliza la indentación para dividir bloques de código y el carácter de nueva línea para distinguir reglas.

La sintaxis más moderna, SCSS, emplea el formato por bloques similar a CSS. Este emplea llaves para indicar bloques de código y punto y coma (;) para dividir las líneas dentro de un bloque. La sintaxis de indentación y los archivos SCSS poseen las extensiones. sass y .scss respectivamente.

Es una variante de CSS que añade elementos de un lenguaje de

programación, como variables, funciones y anidación de selectores, lo que mejora la legibilidad del código y facilita su mantenimiento.

Se trata de un archivo particular diseñado para SASS, un software desarrollado en Ruby que compila hojas de estilo CSS para navegadores. SASS añade numerosas funciones extras a las variables CSS, facilitando y acelerando la escritura de CSS.

Los archivos SCSS son procesados por el servidor que corre una aplicación web para crear un CSS convencional que su navegador puede interpretar.

4. Principales etiquetas HTML5, así como una breve descripción de que son y cómo funcionan las etiquetas semánticas.

HTML5 incluye etiquetas semánticas que ayudan a estructurar el contenido, mejorando la accesibilidad y optimización SEO de las páginas. Las etiquetas semánticas describen claramente el propósito del contenido.

Para comprender mejor lo que es una etiqueta html semántica tengamos en cuenta también el por qué algunas etiquetas no lo son;

Por ejemplo, etiquetas como <header>, <article> y <footer> son etiquetas HTML semánticas. Indican claramente la funcionalidad de su contenido.

En cambio, etiquetas como y son ejemplos típicos de elementos HTML no semánticos. Aunque albergan contenido, no indican qué tipo de contenido contienen ni qué función desempeña esa pieza en la página.

Algunas de las principales etiquetas de HTML5 son;

<body>: para el contenido.

<head>: para información sobre el documento.

: división dentro del contenido.

<a>: para enlaces.

: para poner el texto en negrita (también puede emplearse

<b>: con la misma función)

: para saltos de línea.

<H1>…<H6>: para títulos dentro del contenido, siendo el H1 el mayor de ellos, y el H6 el menor.

<img>: para añadir imágenes al documento.

: para listas ordenadas, <ul> para listas desordenadas, <li> para elementos dentro de la lista.

<p>: para parágrafos.

: para estilos de una parte del texto.

<footer>: define el pie de página.

<section>: representa secciones temáticas del contenido.

<article>: representa contenido independiente y autocontenido, como artículos de un blog o noticias.

<nav>: define un conjunto de enlaces de navegación, como el menú de una página.

<aside>: representa contenido adicional, como barras laterales o contenido relacionado.

<figure>: agrupa contenido ilustrativo, como imágenes, gráficos o diagramas, junto con su leyenda.

<figcaption>: define una leyenda para el elemento <figure>.

<main>: define el contenido principal de un documento, excluyendo encabezados, pie de página y navegación.

<audio>: inserta contenido de audio en la página, con atributos para controles y opciones de reproducción.

<video>: inserta contenido de video en la página, también con controles y opciones.

<time>: representa una fecha, hora o ambos, que puede ser procesada de forma automática.

<mark>: resalta o marca texto importante o relevante.

<progress>: representa el progreso de una tarea, ideal para barras de progreso.

<details>: define contenido que el usuario puede mostrar u ocultar bajo demanda.

<summary>: se usa junto con <details> para crear un encabezado visible en el que se puede hacer clic para mostrar/ocultar contenido.

<meter>: representa una medición dentro de un rango conocido, como la capacidad de almacenamiento.

<meta>:se coloca dentro de la etiqueta <head>, proporciona metadatos sobre el documento HTML, como el conjunto de caracteres, la descripción de la página, las palabras clave, el autor, etc.

<link>: define una relación entre el documento actual y un recurso externo se utiliza principalmente para enlazar hojas de estilo CSS.

<script>: inserta o enlaza scripts de JavaScript dentro de un documento HTML, puede ir en el <head> o antes del cierre del <body>

<button>: representa un botón que se puede utilizar para realizar alguna acción cuando se hace clic.

5. Principales elementos CSS y su aplicación.

Selectores: Permiten apuntar a elementos específicos en el HTML.

Propiedades de color y fondo: Permiten definir colores de texto y fondo, y aplicar imágenes.

Modelo de caja (Box Model): Define márgenes, bordes, padding, y tamaño de los elementos.

Flexbox y Grid: Para diseño de layouts responsivos.

Font-size:

Define el tamaño de la fuente y el valor se puede escribir en pixels o en ems.

Color:

Define el color de la tipografía.

Width:

Define el ancho de un elemento, el valor se puede escribir en pixels, ems o porcentaje.

Max-width o min-width:

Definen el ancho máximo o mínimo de un elemento. Muy importante en sitios adaptables.

Padding:

Es la distancia desde el borde de un elemento hasta su contenido.

Margin:

Es la distancia entre un elemento y otro (desde el borde de un elemento hacia afuera).

Border:

Define el borde de un elemento, su color, su estilo y grosor.

Background:

Define los fondos de un objeto. El fondo puede ser una imagen o un color.

6. Guía de creación de un sitio web con el paso a paso del desarrollo.

Planificación y estructura del contenido: Definir el objetivo y la estructura de la web, también es importante tener una idea sobre el diseño que querrías tener sobre tu propia página, puedes ayudarte con dibujos, o con cualquier herramienta que te permita tener una imagen lo más clara posible sobre el aspecto que deseas de tu sitio web, de esta forma, sabrás cómo aplicar el posterior desarrollo.

También tener en mente un dominio web donde alojar tu página es esencial, esto supondrá el nombre de tu sitio web, debería ser corto, simple, claro, y tener algo que incite a los usuarios a clickar en ella. De esta forma tendrás más posibilidades de que tenga más visitas

Adquirir un plan de hosting es lo que te permitirá que se encuentre en línea tu propio sitio web, de tal forma que sea accesible para cualquier persona.

Creación de HTML básico: Esqueleto del HTML con etiquetas semánticas. Aquí estructurarás los apartados que tendrá tu página, cómo será su menú de qué forma se distribuirá el contenido de tu página web, etc.

Estilización con CSS: Aplicación de estilos para mejorar la apariencia. Este paso, junto con el próximo, son esenciales para que la interacción con el usuario resulte lo más cómoda posible, un diseño no demasiado complejo, pero atractivo, no ha de ser una montaña rusa de elementos danzando en la página, pero con ciertos movimientos, usando JavaScript, hará de la experiencia de usuario, algo mucho más gratificante.

Esta es una parte muy personal y en la que se emplean una importante cantidad de capacidades creativas únicas.

Interactividad con JavaScript: Añadir scripts para mejorar la experiencia de usuario.

Pruebas y ajustes finales: Revisión de errores y optimización en varios dispositivos.

Pues así te asegurarías de que tu página proporciona un estado en igualdad de condiciones para todo tipo de usuarios,

independientemente de desde qué dispositivos se esté accediendo a tu página. Al fin y al cabo, si los usuarios de móvil no tendrán un contexto cómo para acceder a tu web, eso se traducirá en una página menos atractiva para ese grupo de posibles navegantes.

7. Guia y uso de librerías de CSS para el desarrollo web

Las librerías de CSS son una opción muy popular entre los desarrolladores Fronted de todos los niveles, debido a que, por lo general, la curva de aprendizaje es casi nula. Te permiten crear interfaces completas en minutos, copiando y pegando bloques de código y simplemente hacer cambios en algunas clases para llegar al resultado que buscas.

Gracias a que están basadas en solamente HTML, CSS y en ocasiones algo de JS, son compatibles con todas las herramientas como React, Angular, Vue o cualquier otra que decidas usar.

Hay una serie de librerías realmente interesantes y funcionales que han marcado precedente en la experiencia creadora de espacios interactivos con los usuarios, por medio de la proporción de estilos, unos ejemplos son los siguientes:

1: Bootstrap

Bootstrap comenzó como un proyecto secundario, hecho y compartido por desarrolladores de Twitter. Hoy, es el framework de CSS más popular para diseño web receptivo y móvil. Es simple, limpio y fácil de usar

Características clave:

- Sistema de cuadrícula adaptable
- Componentes preconstruidos de IU
- Temas personalizables y extensibles
- Documentación extensa

2: Tailwind CSS

Quizá, Tailwind CSS es un movimiento en la misma medida que es un framework.

Su creador, Adam Wathan, escribió un manifiesto, sobre el pensamiento tras bambalinas de Tailwind CSS. Para él, en esencia, el CSS no debería ser descriptivo y semántico (Como la clase “header”), sino funcional (Como “center-flex-3”).

Él lo llama un framework CSS de utilidad primero.

Características clave:

- Clases de utilidad para facilitar el estilo.
- Diseño adaptable.
- Configuración personalizable.
- Enfoque compatible con componentes.

3: Materialize

El framework de CSS Materialize, se basa en los principios del Diseño Material de Google.

Se enfoca en darte un diseño visual audaz, con animaciones centradas en UX (motion).

Características clave:

- Componentes y estilos inspirados en el Diseño Material
- Sistema de cuadrícula adaptable
- Personalización potenciada por Sass
- Plugins de JavaScript incorporados

4: Bulma

Bulma es un framework de CSS ligero, basado en Flexbox.

La sintaxis de este framework es en lenguaje sencillo, así que se basa en gran medida en clases utilitarias, o modificadores descriptivos, como ".button" y como ".is-large".

Características clave:

- Sistema de cuadrícula, basado en Flexbox

- Arquitectura modular
- Potenciado por Sass para brindar una personalización sencilla
- Código y diseño minimalistas

Ejemplos de sitios web conocidos que utilizan Bulma: CSS Ninja y Signal.

Enlace a GitHub

https://github.com/Cortes-cmd/LenguajeDeMarcas.git

Bibliografía

ChatGPT. (s/f). Chatgpt.com. Recuperado el 11 de noviembre de 2024, de https://chatgpt.com/c/67320241-fe88-8001-8566-c8d404dbf742

¿Conoces cuáles son las propiedades mas utilizadas en CSS? (s/f). Mgpanel.org. Recuperado el 11 de noviembre de 2024, de https://blog.mgpanel.org/post/-conoces-cuales-son-las-propiedades-mas-utilizadas-en-css-

CSS básico. (s/f). MDN Web Docs. Recuperado el 11 de noviembre de 2024, de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics

Desarrollo de Aplicaciones WEB. Introducción a HTML y CSS. Página Rafael Barzanallana. Universidad de Murcia. (s/f). Www.um.es. Recuperado el 11 de noviembre de 2024, de https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-1-introduccion-html-css.html

Hernandez, I. (2023, junio 22). *16 Frameworks Populares de CSS, Útiles Para Ahorrar Tiempo (Con Estilo)*. Guías para Sitios Web, Tips & Conocimiento; DreamHost. https://www.dreamhost.com/blog/es/frameworks-css-populares/

Islas, D. S. (2023, julio 10). Cómo crear una página web: Guía paso a paso. *Blog de Wix*. https://es.wix.com/blog/guia-para-crear-paginas-web

¡Ohh, la UI!, ¿librerías de CSS o librerías de componentes, cuál usar? (s/f). Fixtergeek.com. Recuperado el 11 de noviembre de 2024, de https://fixtergeek.com/blog/ohh-la-ui-librerias-de-css-o-librerias-de-componentes-cual-usar

Pavlik, V. (2023, junio 14). *HTML Semántico: Qué Es y Cómo Usarlo Correctamente*. Semrush Blog; Semrush. https://es.semrush.com/blog/html-semantico/

(S/f). Iebschool.com. Recuperado el 11 de noviembre de 2024, de https://www.iebschool.com/blog/que-es-etiqueta-html-analitica-usabilidad/