 Desarrollo de Interfaces WEB (DAWN 2024-2025)

UT2.2 -Repaso de HTML

**1.- Crea un archivo HTML para cada una de las propuestas que permita construir a los navegadores la siguientes interfaces:**

Las páginas deben ser validas según HTML5: <https://validator.w3.org/>

Utilizad los estilos que adjunto para poder

1.- Versión del Tapenade Provenzal. Utiliza las etiquetas para la gestión de párrafos.

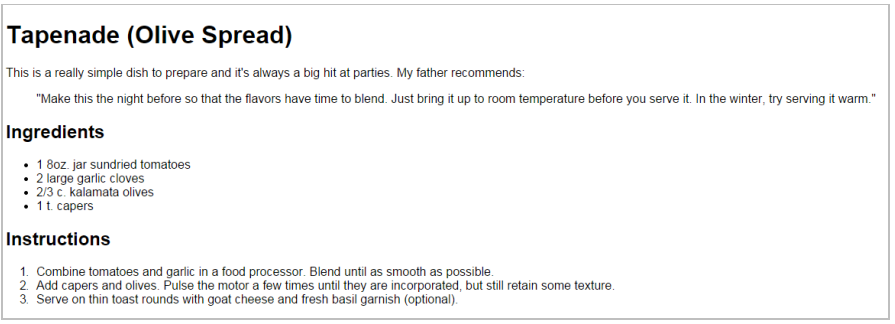
    <style>

        body{

            font-family:sans-serif;

        }

    </style>



2.- Glosario. Crea una página como la que siguiente.

• Haciendo clic en la letra B, el documento debe ir a la descripción de esa letra.

• En esta página se ha utilizado una lista del tipo dl.

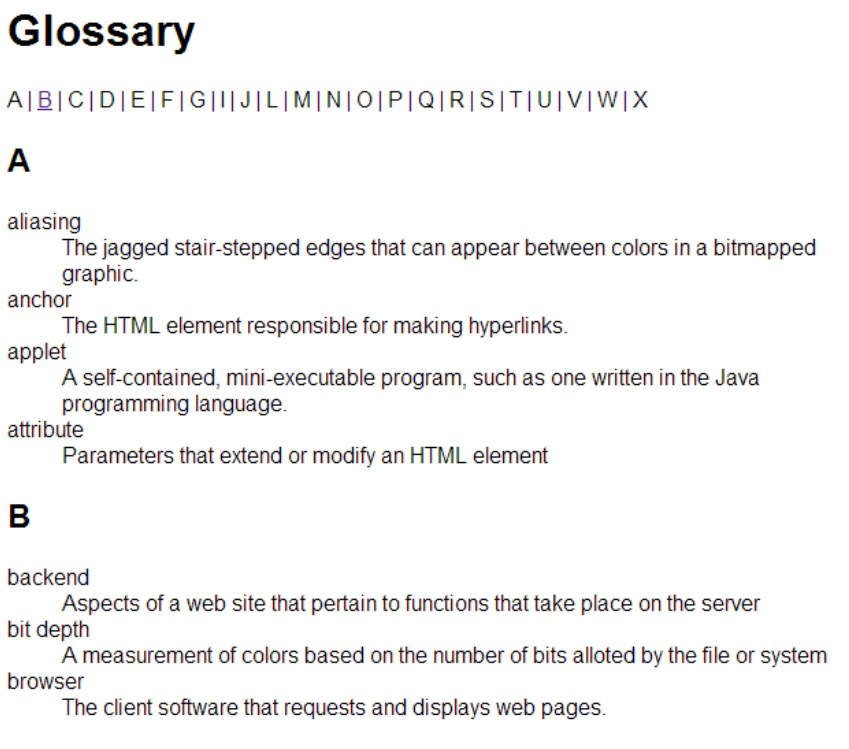
    <style>

        body{

            margin-bottom: 100%;

        }

    </style>



3.- Ejercicio de tablas,

Crea un documento con tablas como las que se ven a continuación:

    <style>

        td, th{

            border: 1px solid grey;

        }

        div{

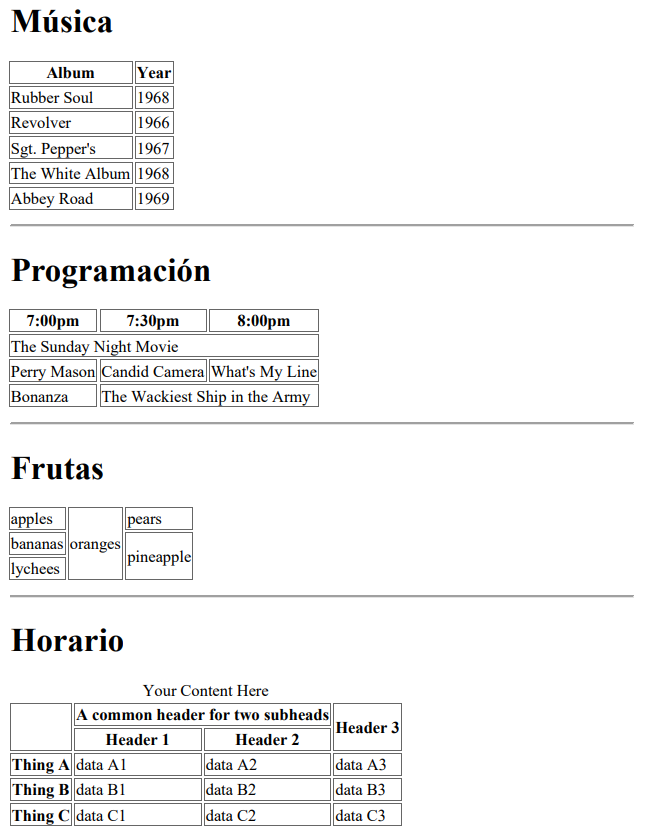
            margin-bottom: 30px;

            border-bottom: 2px solid grey;

            padding-bottom: 30px;

        }

    </style>



4.-Ejercicio de Listas:

Crea un documento utilizando listas para que aparezca como las que se ven a continuación:

    <style>

        .roman {

            list-style-type:upper-roman;

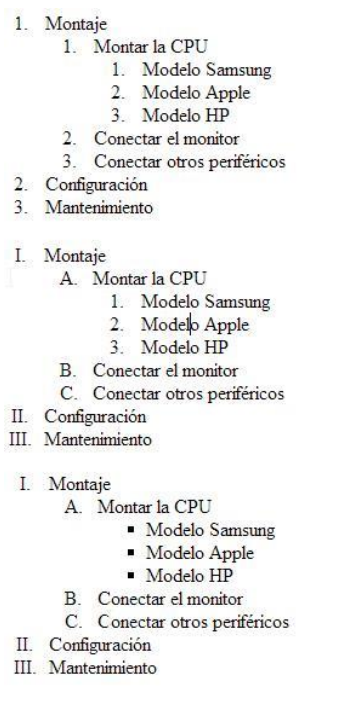
        }

        .mayus{

            list-style-type: upper-alpha;

        }

    </style>



5.- Etiquetas Semánticas

Dado el documento escritores.html que se adjunta en “UT2.2 Repaso HTML Documento Escritores.zip”,

Analiza en qué casos se deben utilizar cada una de las etiquetas semánticas usadas y justifica si se ha realizado correctamente.

Por ejemplo:

La etiqueta <aside> en HTML se utiliza para representar una sección de una página que contiene contenido relacionado indirectamente con el contenido principal

         <aside>

            <p><img src="img/becquer\_mini.jpg" alt="Gustavo Adolfo Bécquer" title="Gustavo Adolfo Bécquer" width="100" height="150"></p>

            <p>Aldolfo Bécquer</p>

          </aside>

En este contexto su uso es correcto porque se está utilizando para incluir una imagen al articulo de Bécquer, No obstante se podría haber utilizado <figure> para detallar más el contenido:

Listado de etiquetas semánticas:

<header>: Representa el encabezado de una sección o del documento completo.

<nav>: Define una sección de navegación.

<main>: Indica el contenido principal del documento.

<section>: Define una sección temática o lógica del documento.

<article>: Representa una composición autónoma en un documento.

<aside>: Contiene contenido relacionado indirectamente con el contenido principal.

<footer>: Representa el pie de una sección o del documento completo.

<figure>: Agrupa contenido ilustrativo, como imágenes, diagramas, etc.

<figcaption>: Proporciona una leyenda para el contenido del <figure>.

<details>: Contiene detalles adicionales que el usuario puede ver o esconder.

<summary>: Proporciona un resumen visible para el elemento <details>.

<time>: Representa una fecha y/o una hora.

<mark>: Resalta texto que es de especial interés o relevancia.

5.- Formularios

Los métodos GET y POST son dos formas de enviar datos desde un formulario HTML a un servidor. Aquí te explico las diferencias clave entre ellos:

**Método GET**

Los datos del formulario se envían como parámetros en la URL.

Por ejemplo, www.ejemplo.com/formulario?nombre=Juan&edad=30.

Visibilidad: Los datos son visibles en la barra de direcciones del navegador, lo que puede ser un problema de seguridad si se envía información sensible.

Limitaciones de tamaño: La cantidad de datos que se pueden enviar está limitada por la longitud máxima de la URL (alrededor de 2000 caracteres).

Se almacena en Caché: Las solicitudes GET pueden ser almacenadas en caché por el navegador, lo que permite volver a acceder a la misma página sin necesidad de reenviar los datos.

**Método POST**

Datos en el cuerpo de la solicitud: Los datos del formulario se envían en el cuerpo de la solicitud HTTP, no en la URL.

Mayor seguridad: Los datos no son visibles en la barra de direcciones, lo que es más seguro para enviar información sensible como contraseñas.

Sin limitaciones de tamaño: No hay un límite práctico en la cantidad de datos que se pueden enviar.

No se almacena en cache: Las solicitudes POST no se almacenan en caché, lo que significa que cada solicitud es única y no se puede reutilizar.

<!-- Formulario usando GET -->

<form action="/buscar" method="get">

  <input type="text" name="query" placeholder="Buscar...">

  <button type="submit">Enviar</button>

</form>

<!-- Formulario usando POST -->

<form action="/enviar" method="post">

  <input type="text" name="nombre" placeholder="Nombre">

  <input type="password" name="contrasena" placeholder="Contraseña">

  <button type="submit">Enviar</button>

</form>

Aquí tienes un ejemplo básico de cómo se usan en los estilos:

   <style type="text/css">

      ol, ul {

     list-style-type: none;

      }

    label {

    width: 50em;

   }

   </style>

5.- Formularios

