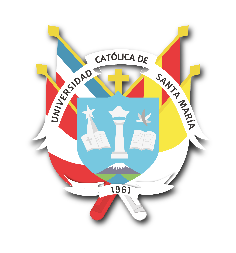
Informe de Prácticas

**DESARROLLO DE APLICACIONES**

**UCSM Esc. Prof. de Ingeniería de Sistemas**

**Agosto - 2023**



**Práctica N° 01: HTML 5**

Elaborado por:

Pezo Vizcarra Adriel Francesco

Roque Monterroso Geraldyne Beatriz

Mamani Salazar Ximena Rosario

Esquivel Carpio Raúl Antonio



**© IEEE 2013 The Institute of Electrical and Electronics Engineers, Inc.**

**GRUPO N° 2**

**PRÁCTICAS DE DESARROLLO DE APLICACIONES**

Presentado por:

2021810041 PEZO VIZCARRA ADRIEL FRANCESCO

**RECONOCIMIENTOS**

Los autores de este trabajo reconocen con gratitud a los creadores del HTML TIM .

**PALABRAS CLAVES**

HTML, formularios web, sitio WEB,

**ÍNDICE**

[1. RESÚMEN 1](#_Toc143454605)

[2. INTRODUCCIÓN 1](#_Toc143454606)

[3. MARCO TEÓRICO 1](#_Toc143454607)

[3.1 ¿Qué es HTML? 1](#_Toc143454608)

[3.2 Estructura básica HTML 1](#_Toc143454609)

[3.3 Estructura de una etiqueta HTML 2](#_Toc143454610)

[3.4 Etiquetas semánticas y no semánticas 2](#_Toc143454611)

[3.5 Categorización de etiquetas 3](#_Toc143454612)

[3.6 Github Pages 3](#_Toc143454613)

[3.7 Etiquetas de Formulario y Validaciones, Fundamentos de CSS 4](#_Toc143454614)

[3.8 Etiquetas de formulario y validación 4](#_Toc143454615)

[3.9 Tipos de Input 5](#_Toc143454616)

[3.10 Atributos de validaciones 5](#_Toc143454617)

[3.11 Patrones regulares (regex) 5](#_Toc143454618)

[3.12 Pseudo clase y css de validación 6](#_Toc143454619)

[3.13 ¿Qué es una tabla? 6](#_Toc143454620)

[3.14 ¿Cuándo usar tablas 6](#_Toc143454621)

[3.15 Atributos 7](#_Toc143454622)

[3.16 ¿Qué es SEO? 7](#_Toc143454623)

[3.17 Metadata 7](#_Toc143454624)

[3.18 Open Graph 8](#_Toc143454625)

[3.19 Twitter Card 8](#_Toc143454626)

[4. EXPERIENCIAS DE PRÁCTICA 1](#_Toc143454627)

[4.1 Fundamentos de HTML. 1](#_Toc143454628)

[4.2 Formulario Web 1](#_Toc143454629)

[4.3 TABLAS. 2](#_Toc143454630)

[4.4 Metadata. 2](#_Toc143454631)

[5. CONCLUSIONES DE LA PRÁCTICA: 3](#_Toc143454632)

[6. CUESTIONARIO 4](#_Toc143454633)

[7. BIBLIOGRAFÍA 7](#_Toc143454634)

**ÍNDICE DE TABLAS Y FIGURAS**

[Figura N° 1: Diagrama de Clases 3](#_Toc40685625)

[Figura N° 2: código de función 4](#_Toc40685626)

# RESÚMEN

HTML5 es un lenguaje de marcado esencial para el diseño web moderno. Comienza con una estructura básica que incluye etiquetas , y . Las etiquetas semánticas, como y , mejoran la estructura y accesibilidad del contenido. Las etiquetas de formulario, como e , permiten la interacción del usuario y la validación de datos mediante atributos como "required" y "pattern", que emplean patrones regulares (regex). Los tipos de entrada varían según las necesidades, como texto, número y fecha. Las tablas () organizan datos tabulares de manera ordenada, con atributos como "border" y "cellspacing" para personalización. Sin embargo, se deben utilizar con moderación, prefiriendo CSS para diseño. La optimización para motores de búsqueda (SEO) es esencial, utilizando metadata y etiquetas como Open Graph y Twitter Card para mejorar la visibilidad en redes sociales. En resumen, HTML5 ofrece la estructura, interacción y optimización necesarias para crear páginas web eficientes y atractivas en el entorno digital actual. IV (La práctica tiene una duración de 2 horas)

# INTRODUCCIÓN

La presentación de información ha sido un proceso que se ha realizado a través de los años por medios electrónicos, el desarrollo del internet se debe al trabajo de una persona que creó una aplicación para mostrar de forma más simple y conveniente los artículos científicos que se publicaban en esos años, a partir de este trabajo se agregan los hipervínculos a los documentos electrónicos, con la inclusión de los hipervínculos nace el hipertexto, el cual permite navegar entre documentos omitiendo la necesidad de los pies de página que antes se incluían en los artículos científicos, cuando se comienza a presentar la información de esta manera en internet esencialmente en un inicio de la información científica otras personas aprovechan las características y capacidades de este tipo de documentos para presentar información gráfica y multimedia, con lo que nace la hipermedia; de esta manera en poco tiempo la intención inicial de presentar de forma más completa y fidedigna los artículos científicos se transforma en el espacio web que contiene no solamente información textual sino gráfica y de otros tipos. Con posterioridad el lenguaje html evoluciona permitiendo en la actualidad el dinamismo pero no por sí mismo sino a través de su integración con otros tipos de lenguaje. el modelo de computación para internet está basado en qué cae html permite la construcción de la estructura de un documento web y a su vez dicha estructura es decorada con características gráficas qué se proveen a través del lenguaje css, al aspecto gráfico de interfaz sí le agrega el dinamismo que es provisto a través de lenguajes como javascript; inicialmente este modelo qué estructura de manera muy eficiente y dinámica la construcción de un documento web permite un manejo de información a través del punto de acceso del usuario o consumidor de la información, a lo que le damos el nombre de frontend.

# MARCO TEÓRICO

## ¿Qué es HTML?

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Acrónimo de HyperText Markup Language. Es un lenguaje de marcado que se utiliza para estructurar y diseñar el contenido de una página web [1], es decir, define la estructura básica de una página, como los encabezados, los párrafos, las imágenes, etc. HTML no es un lenguaje de programación [2]. | <!DOCTYPE html> | Para emplearlo, lo primero que necesitamos es la herramienta esencial para desarrollar, un editor de código. Como información adicional, también se puede codificar en bloc de notas. | Lo empleamos cuando queremos desarrollar una página web. En un archivo propio o insertado en archivos de otros lenguajes, como por ejemplo PHP. |

## Estructura básica HTML

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| La estructura básica de HTML consiste en los siguientes elementos: <!DOCTYPE html>, <html>, <head> y <body>. El <!DOCTYPE html> se utiliza para especificar la versión de HTML que se está utilizando, mientras que el elemento <html> se utiliza para encerrar todo el contenido de la página web [3]. | <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <title>Título de la página</title>  </head>  <body>      <!--Contenido-->  </body>  </html> | En algunos editores de código encontramos el programa Emmet (integrado o en forma de extensión) que entre las funciones que posee, encontramos las abreviaciones. Escribiendo html:5 (5 por la versión de HTML) y presionando Enter se genera la estructura básica de HTML | Cada vez que queremos crear un archivo HTML, dado a que es lo básico para empezar a codificar y define información relevante como el título de la página, el lenguaje en el que se encuentra, como se debe visualizar dependiendo del dispositivo, etcétera. |

## Estructura de una etiqueta HTML

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Generalmente las etiquetas se componen de 3 partes: La etiqueta de apertertura, el contenido y la etiqueta de cierre [2]. Claro hay excepciones a estas reglas como la etiqueta <br>, cuya única función es realizar un salto de llínea. Dentro de la etiqueta de apertura también puede poseer atributos | Etiqueta promedio  <p>Mi perro es muy grande</p>  Etiqueta con atributo  <p class="pr-style">Mi perro es muy grande</p> | Las etiquetas HTML se utilizan para crear y dar formato al contenido de una página web. Van entre corchetes angulares y tienen un nombre y un valor. El nombre de la etiqueta es la palabra clave que identifica el tipo de elemento que crea, y el valor es el contenido que encierra la etiqueta. | Son la base de nuestro código HTML, por ende siempre debemos emplearlo. Pese a que los navegadores casi siempre corrigen los problemas de sintaxis, debemos procurar escribir nuestro código de la manera correcta, |

## Etiquetas semánticas y no semánticas

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Las etiquetas semánticas en HTML son aquellas que describen su significado de manera que sea entendible para el ser humano [4], por ejemplo <header>. Las no semánticas son lo contrario, no expresan ningún significado. Etiquetas como <div> y <span> son los ejemplos más conocidos de estas etiquetas no semánticas [5]. | Etiqueta Semántica  <header>Título</header>  <footer>Enlaces</footer>  <section>Sección</section>  Etiqueta no Semántica  <div>Título</div>  <div>Enlaces</div>  <div>Sección</div> | Empleando las etiquetas correspondientes según sea el caso. Por ejemplo, emplear <nav> a la hora de crear una barra de navegación, <main> para identificar el contenido principal y <aside> para el contenido de menor importancia. | Siempre debemos crear código semántico, ya que además de proveer información a los navegadores, permite que la página sea más accesible para los usuarios que posean discapacidades visuales y empleen un lector de pantalla, y mejoran tu posicionamiento en los buscadores [5]. |

## Categorización de etiquetas

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Es una forma de agrupación de etiquetas que comparten características y nos ayuda a definir y describir cómo se comportan estas y las reglas en común que siguen [6]. | Contenido de metadatos:  <link/>  Flujo de Contenido:  <iframe></iframe>  Flujo de Contenido:  <article></article> | Al emplear categorías de contenido, es importante seguir las reglas de cada categoría. Por ejemplo, el contenido de encabezado solo puede colocarse dentro del contenido de sección. | Siempre que empleamos una etiqueta que forme parte de uno de estas categorías. |

## Github Pages

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Es un servicio de alojamiento de sitios web estáticos que permite a los usuarios de GitHub el poder crear y publicar un sitio web estático directamente desde un repositorio [7]. | Creamos el archivo de entrada para el sitio (puede ser index.html, index.md o README.md)    Hacemos clic en Configuración    En la sección “Código y automatización", hacemos clic en Páginas. Para ver el sitio hacemos clic en Visitar sitio [8]. | Debemos crear el repositorio donde almacenaremos todo el código. En cuanto envíes el index.html al repositorio remoto, se iniciará automáticamente un proceso que desplegará nuestro sitio web funcionando automáticamente [9]. | GitHub Pages es una excelente opción sobre otros hosting gratuitos, ya que nos permite mostrar nuestros proyectos a partir de repositorios de GitHub de manera ilimitada [9]. |

## Etiquetas de Formulario y Validaciones, Fundamentos de CSS

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
|  |  |  |  |

## Etiquetas de formulario y validación

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Son las etiquetas que se emplean para crear formularios y validar los datos introducidos por los usuarios que rellenen este [10]. | Ejemplo de formulario simple:  <form>    <label>Nombre</label>      <input name='nombre'  type='text'/>  <br>      <label>Sexo: </label>      <select name="select">      <option value="H">  Hombre</option>      <option value="M">  Mujer</option>      </select>      <label>Edad: </label>      <input type='checkbox'  name='edad' value='20-  39' />20-39      <input type='checkbox'  name='edad' value='40-  59' /> 40-59      <input type='checkbox'  name='edad' value='60-  79' /> 60-79      <input type="submit"  value="Enviar" />  </form> | Existen una variedad de etiquetas a emplear. Entre las más importantes tenemos:  <input>: Esta etiqueta se utiliza para crear una variedad de controles de entrada, como campos de texto, botones, casillas de verificación.  <fieldstep>: Esta etiqueta nos permite organizar en grupos los campos del formulario a desarrollar.  <select>: Esta etiqueta se utiliza para crear un menú desplegable.  <option>: Junto con select, lo empleamos para definir las opciones.  <textarea>: Se utiliza para crear un campo de texto multilínea. | Empleamos estas etiquetas dependiendo que tipo de formularios estemos realizando. Por ejemplo en un formulario de contacto necesitaremos 4 o 5 tipos de etiquetas, en cambio en formularios de pedido necesitarán una mayor cantidad dada la complejidad de este. |

## Tipos de Input

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Son elementos de entrada que se utilizan en los formularios web y se definen mediante el atributo "type". [11] | Tipos de input más empleados:  <input type="text">  <input type="number">  <input type="password">  <input type="submit">  <input type="date"> | Empleamos dependiendo lo que queramos hacer con el input. Si queremos que se ingresen números, texto, si queremos enviar el formulario, si queremos que se definan fechas, etc. | Siempre que empleemos formularios, usaremos inputs. Dependiendo la complejidad de este, emplearemos mas o menos tipos de este. |

## Atributos de validaciones

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Son elementos empleados en las etiquetas de formulario para validar los datos introducidos por los usuarios en este [12]. | required:  <input id="choose" name="i\_pref" required />  minlength/maxlength:  <input type="text" minlength="5" id="fruit" />  min/max:  <input type="number" min="0" step="5" max="100">  type: | Entre las etiquetas más importantes:  required: Especifica si se requiere completar el campo.  minlength y maxlength: Especifican la longitud mínima y máxima de las cadenas.  min y max: Especifican los valores mínimo y máximo de los tipos de entrada numéricos.  type: Especifica que tipo de dato debe ingresarse en dicho campo [13] | Siempre debemos emplear estos atributos para asegurarnos que los datos ingresados y que se enviarán posteriormente coinciden con los requisitos y diversos controles de formulario que exigen las organizaciones [13]. |

## Patrones regulares (regex)

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Son secuencias de caracteres que conforman un patrón de búsqueda, que se utiliza para encontrar una determinada combinación de caracteres dentro de un string [14]. | Un numero celular que empiece con 9:  <form method="post" action="/send/">      Número:      <input type="number" name="username" pattern="9.+" required>  </form> | Se debe agregar el atributo "pattern" en el elemento input que se desea validar. El atributo "pattern" acepta una expresión regular (regex) que se utilizará para validar el valor ingresado en el campo input [15]. | Se emplea en el input deseado, donde se especifica la expresión regular que se utilizará para validar el valor ingresado . Si el valor ingresado no coincide con el patrón especificado arrojará error. |

## Pseudo clase y CSS de validación

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Son las herramientas que se emplean en CSS para seleccionar elementos HTML que cumplen ciertas características de validación. Las pseudo-clases son palabras clave que se agregan a los selectores y que especifican un estado especial del elemento seleccionado [16]. | Input Válido:  input:valid {      background-color: green;  }  Input Inválido:  input:invalid {      background-color: red;  } | La pseudo-clase :valid representa cualquier elemento <input> u otro elemento <form> cuyo contenido se valide satisfactoriamente [17]. Por otro lado, la pseudo-clase :invalid de CSS representa cualquier elemento <input> u otro elemento <form> cuyos contenidos no se puedan validar [18]. | El emplear estas pseudo clases ayudan a la experiencia del usuario, ya que permite que los campos adopten una apariencia que ayuda al usuario a confirmar que sus datos han sido ingresados correctamente o no, y corregirlos si es el caso [17]. |

## ¿Qué es una tabla?

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Es un conjunto estructurado de datos distribuidos en filas y columnas , que permite organizar y presentar información de forma clara y fácil de entender para el usuario [19]. | <table class="default">    <tr>      <td>Celda 1</td>      <td>Celda 2</td>      <td>Celda 3</td>    </tr>    <tr>      <td>Celda 4</td>      <td>Celda 5</td>      <td>Celda 6</td>    </tr>  </table> | Se crean utilizando las etiquetas <table>, <tr> y <td> y se utilizan con mucha frecuencia en la web para mostrar datos de manera organizada [20]. | Las tablas en HTML se utilizan para mostrar datos de forma ordenada , organizada en filas y columnas [21]. Algunos casos de uso típicos incluyen horarios, calendarios y listados de elementos con información diversa que se pueden expresar en forma tabular. Sin embargo, se debe tener en cuenta que en la actualidad se recomienda su uso solo en los casos en los que sea necesario incluir información tabulada en una página web [22]. |

## ¿Cuándo usar tablas

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
| Cuando se necesita presentar datos estructurados en filas y columnas de manera organizada. | <table>      <tr>          <th>Nombre</th>          <th>Edad</th>          <th>Pais</th>      </tr>  </table> |  |  |

## Atributos

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
|  |  |  |  |

## ¿Qué es SEO?

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
|  |  |  |  |

## Metadata

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
|  |  |  |  |

## Open Graph

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
|  |  |  |  |

## Twitter Card

| **¿QUÉ ES?** | **¿CÓMO SE DECLARA?** | **¿CÓMO LO USO?** | **¿CUÁNDO LO USO?** |
| --- | --- | --- | --- |
|  |  |  |  |

¿cómo se implementa o usa?

Figura N° 2: código de función

main()

{

return 0;

}

Se usa para liberar recursos usados por el objeto ejemplo, memoria dinámica:

UnaClase::~UnaClase()

{

cout << "Destructor de Clase" << endl;

//delete ptrAtributo;

}

# EXPERIENCIAS DE PRÁCTICA

## Fundamentos de HTML.

a)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Primeros Pasos</title>

</head>

<body>

    <em>Primeros Pasos</em>

</body>

</html>

b)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Primeros Pasos</title>

</head>

<body>

    <a href="https://www.mozilla.org/" title="La página de inicio de Mozilla" target="\_blank">Click aquí</a>

</body>

</html>

## Formulario Web

a)

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Formulario de Contacto Simple</title>

    <style>

        form {

            margin: 0 auto;

            width: 400px;

            padding: 1em;

            border: 1px solid #CCC;

            border-radius: 1em;

        }

        ul {

            list-style: none;

            padding: 0;

            margin: 0;

        }

        form li + li {

            margin-top: 1em;

        }

        label {

            /\* Tamaño y alineación uniforme \*/

            display: inline-block;

            width: 90px;

            text-align: right;

        }

        input,

        textarea {

            font: 1em sans-serif;

            width: 300px;

            box-sizing: border-box;

            border: 1px solid #999;

        }

        input:focus,

        textarea:focus {

            border-color: #000;

        }

        textarea {

            vertical-align: top;

            height: 5em;

        }

        .button {

            padding-left: 90px;

        }

        button {

            margin-left: .5em;

        }

    </style>

<body>

        <form action="/my-handling-form-page" method="post">

            <ul>

                <li>

                    <label for="name">Nombre:</label>

                    <input type="text" id="name" name="user\_name" />

                </li>

                <li>

                    <label for="mail">E-mail:</label>

                    <input type="email" id="mail" name="user\_mail" />

                </li>

                <li>

                    <label for="msg">Mensaje:</label>

                    <textarea id="msg" name="user\_message"></textarea>

                </li>

                <li class="button">Ç

                    <button type="submit">Envie su mensaje</button>

                </li>

            </ul>

        </form>

    </body>

b)

<!DOCTYPE html>

<html lang="en-us">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width">

    <title>Animals table</title>

    <link href="minimal-table.css" rel="stylesheet" type="text/css">

  </head>

  <body>

    <h1>Animals table</h1>

    <table>

      <tr>

        <th colspan="2">Animals</th>

      </tr>

      <tr>

        <th colspan="2">Hippopotamus</th>

      </tr>

      <tr>

        <th rowspan="2">Horse</th>

        <td>Mare</td>

      </tr>

      <tr>

        <td>Stallion</td>

      </tr>

      <tr>

        <th colspan="2">Crocodile</th>

      </tr>

      <tr>

        <th rowspan="2">Chicken</th>

        <td>Hen</td>

      </tr>

      <tr>

        <td>Rooster</td>

      </tr>

    </table>

  </body>

</html>

## TABLAS.

a)

<!DOCTYPE html>

<html lang="en-us">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width">

    <title>Table template</title>

    <link href="minimal-table.css" rel="stylesheet" type="text/css">

  </head>

  <body>

    <h1>Table template</h1>

    <table>

      <tr>

          <td>Hola, soy tu primera celda.</td>

          <td>Soy tu segunda celda.</td>

          <td>Soy tu tercera celda.</td>

          <td>Soy tu cuarta celda.</td>

      </tr>

      <tr>

          <td>Segunda fila, primera celda.</td>

          <td>Segunda celda.</td>

          <td>Tercera celda.</td>

          <td>Cuarta celda.</td>

      </tr>

  </table>

  </body>

</html>

## 

b)

<!DOCTYPE html>

<html lang="en-us">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width">

    <title>Dogs table</title>

  <link href="minimal-table.css" rel="stylesheet" type="text/css">

  </head>

  <body>

    <h1>Dogs Table</h1>

    <table>

      <tr>

        <td>&nbsp;</td>

        <th>Knocky</th>

        <th>Flor</th>

        <th>Ella</th>

        <th>Juan</th>

      </tr>

      <tr>

        <th>Breed</th>

        <td>Jack Russell</td>

        <td>Poodle</td>

        <td>Streetdog</td>

        <td>Cocker Spaniel</td>

      </tr>

      <tr>

        <th>Age</th>

        <td>16</td>

        <td>9</td>

        <td>10</td>

        <td>5</td>

      </tr>

      <tr>

        <th>Owner</th>

        <td>Mother-in-law</td>

        <td>Me</td>

        <td>Me</td>

        <td>Sister-in-law</td>

      </tr>

      <tr>

        <th>Eating Habits</td>

        <td>Eats everyone's leftovers</td>

        <td>Nibbles at food</td>

        <td>Hearty eater</td>

        <td>Will eat till he explodes</td>

      </tr>

    </table>

  </body>

</html>

## Metadata.

a)

<!DOCTYPE html>

<html lang="en-us">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width">

<title>&lt;title&gt; element</title>

</head>

<body>

<h1>&lt;h1&gt; element</h1>

</body>

</html>

b)

<!DOCTYPE html>

<html lang="en-us-ja">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width">

<title>&lt;title&gt; element</title>

</head>

<body>

<h1>&lt;h1&gt; element</h1>

<p>Ejemplo en japonés: ご飯が熱い。</p>

</body>

</html>

c)

<!DOCTYPE html>

<html lang="en-us-ja">

  <head>

    <meta property=”og:title” content=”Donquixote Crime Family”/>

    <meta property=”og:type” content=”article” />

    <meta property=”og:url” content=”http://www.example.com” />

    <meta property=”og:image” content=

    ”http://www.example.com/DonquixoteFamily.jpeg”/>

   <meta name="twitter:card" content="summary"></meta>

    <meta charset="ISO-8859-1">

    <meta name="viewport" content="width=device-width">

    <title>&lt;title&gt; element</title>

  </head>

  <body>

    <h1>&lt;h1&gt; element</h1>

    <p>Ejemplo en japonés: ご飯が熱い。</p>

  </body>

</html>

<!DOCTYPE html>

<html lang="en-us-ja">

  <head>

    <meta property=”og:title” content=”Donquixote Crime Family”/>

    <meta property=”og:type” content=”article” />

    <meta property=”og:url” content=”http://www.example.com” />

    <meta property=”og:image” content=

    ”http://www.example.com/DonquixoteFamily.jpeg”/>

    <meta name="twitter:card" content="summary" />

    <meta name="twitter:site" content="@flickr" />

    <meta name="twitter:title" content="Donquixote Family”" />

    <meta charset="ISO-8859-1">

    <meta name="viewport" content="width=device-width">

    <title>&lt;title&gt; element</title>

  </head>

  <body>

    <h1>&lt;h1&gt; element</h1>

    <p>Ejemplo en japonés: ご飯が熱い。</p>

  </body>

</html>

# CONCLUSIONES DE LA PRÁCTICA:

1. La encapsulación de datos se refiere al control de acceso, y al uso de private, protected y public
2. Los diagramas UML nos ayudan a saber qué tipo de acceso tiene un miembro de clase y saber qué tipo de variable guarda, atributo, o retorna (o recibe) método.
3. TERCERA
4. CUARTA

# CUESTIONARIO

1. **¿Qué es HTML?**

HyperText Markup Language(Lenguaje de Marcas de Hipertexto) es un componente básico el cual es utilizado para estructurar el contenido y presentar información en la web. Define la estructura básica de una página web utilizando etiquetas y elementos lo cuales nos permiten organizar y formatear textos, párrafos, imágenes, enlaces, formularios, tablas de datos y otros tipos de contenido.

1. **¿Cuál es la estructura de un documento html?**

La estructura HTML de una página web se compone de las etiquetas <html>, estas indican el comienzo de la página; <head>, que contiene la información más descriptiva sobre el sitio; y <body>, que conforma todos los elementos que sean visibles de la página.

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Título de la página</title>

</head>

<body>

    <!--Contenido-->

</body>

</html>

1. **¿Qué significa la etiqueta DOCTYPE?**

La etiqueta <!DOCTYPE> define el tipo de documento e informa al navegador la versión de HTML que se está utilizando. Ayuda al navegador a interpretar el contenido correctamente.

1. **¿Qué es una etiqueta de apertura en html?**

Una etiqueta de apertura en HTML es la que marca el comienzo de un elemento y se denota mediante el uso del símbolo <. En el caso de <p> que marca el comienzo de un párrafo.

1. **¿Qué es el contenido de un elemento en html?**

El contenido de un elemento en HTML es la información que se encuentra entre las etiquetas de apertura y cierre del elemento. Un ejemplo sería <p>Esto es un párrafo</p>, "Esto es un párrafo" es el contenido del elemento <p>.

1. **¿Qué es una etiqueta de cierre en html?**

Una etiqueta de cierre en HTML marca el final de un elemento y se denota mediante el uso del símbolo </, un ejemplo seria </p> que marca el final de un párrafo.

1. **¿Qué son elementos en html?**

Los elementos en HTML son componentes básicos lo cuales son utilizados para poder estructurar y presentar contenido en una página web. Cada elemento consta de una etiqueta de apertura, contenido y una etiqueta de cierre, o también podría ser una etiqueta vacía.

1. **¿Qué son atributos en html?**

Los atributos en HTML son valores adicionales que se proporcionan en las etiquetas y de esta manera proporcionar información adicional sobre el elemento.

1. **¿Cómo se anidan los elementos en html?**

Los elementos se pueden anidar dentro de otros elementos en HTML siempre y cuando se coloque las etiquetas de apertura y cierre correspondientes dentro del contenido de otro elemento.

1. **¿Qué son elementos de bloque en html?**

Son elementos que ocupan todo el ancho disponible en su contenedor o elemento padre. Un ejemplo de estos son los encabezados, los párrafos, listas. Comienzan a partir de una nueva línea. Pueden contener elementos de línea así como otros elementos de bloque.

1. **¿Qué son elementos de línea en html?**

Los elementos de línea en HTML son aquellos elementos que ocupan solo el espacio necesario para mostrar su contenido, y no un bloque entero, y ahí la diferencia con respecto a los elementos de bloque. Algunos de los efectos más remarcables son enlaces (a), los botones (button), los elementos enfatizados (em), entre otros.

1. **¿Qué son elementos vacíos en html?**

Los elementos vacíos en HTML, también conocidos como elementos auto-cerrados o elementos sin contenido, no cuentan con una etiqueta de cierre. Se denotan agregando un símbolo de / antes del cierre de la etiqueta.

1. **¿Qué son atributos booleanos en html?**

Son aquellos que no necesitan un valor porque tienen solo una opción, que generalmente es el mismo que el nombre del atributo. Ejemplos de estos son: "required", que se utiliza en los campos de formulario y requiere que los usuarios completen ese campo antes de enviar el formulario , y "checked", que se utiliza en los elementos de entrada de tipo "checkbox".

1. **¿Qué son etiquetas semánticas?**

Las etiquetas semánticas en HTML son aquellas que brindan un significado adicional al contenido de esta manera facilitan la comprensión tanto para los desarrolladores como para los motores de búsqueda.

1. **¿Qué son etiquetas no semánticas?**

Son lo contrario a las etiquetas semánticas, no proporcionan ningún significado a los navegadores. Es una mala práctica ya que además de afectar al posicionamiento de la página, dificulta el entendimiento de nuestro código tanto para nosotros y nuestros colaboradores como para la máquina.

1. **¿Qué son etiquetas semánticas que son etiquetas no semánticas?**

Son semánticas cuando proporcionan un significado adicional tanto al navegador como a los desarrolladores. Por ejemplo, <nav> es una etiqueta semántica dado a que da a entender que es una sección de la página donde se proporcionará enlaces de navegación tanto a otras como dentro de la misma página.

1. **¿Qué es SEO?**

Search Engine Optimization se refiere a un grupo de prácticas y estrategias que tienen como objetivo mejorar la visibilidad y el posicionamiento de un sitio web en los resultados de búsqueda de los motores de búsqueda como Google, Yahoo!, YouTube, Bing, etc. El objetivo del SEO es aumentar el tráfico orgánico y la calidad de las visitas al sitio web, de esta manera optimizando diversos aspectos técnicos, de contenido y de enlaces.

1. **¿Qué reglas debemos seguir para un adecuado posicionamiento de nuestra página web en el buscador?**
2. Investigar que palabras clave son las más relevantes para el tipo de negocio o temática de la página.
3. Optimizar el contenido de la página con dichas palabras clave (en títulos, encabezados, partes del textos, enlaces, entre otros).
4. Mejorar la experiencia del usuario de la página: facilitando la navegación, carga rápida, información relevante y concisa.
5. Tener enlaces de calidad, que son los que proceden de sitios con un alto DA (Domain Authority, métrica de calificación que predice qué lugar ocupará tu sitio web en las páginas de resultados), lo que aumenta su autoridad y relevancia.
6. Mantener la página actualizada y relevante, agregando nuevo contenido de manera constante y mejorando y actualizando el existente.
7. **¿Qué función cumplen las etiquetas meta data?**

Empleamos estas etiquetas para proporcionar información adicional sobre una página web a los buscadores y otros clientes, y entender de qué trata la página y a mostrarla de manera más relevante. Ejemplo de estas son las *meta* *description*, *meta* *keywords*, *meta* *author*, entre muchos otros.

1. **¿Cómo agregamos etiquetas de metadata a nuestra página web?**

Siguiendo con los ejemplos de la pregunta anterior, agregamos las etiquetas *metadata3322w*:

* Para dar una descripción de la página

<meta name="description" content="Este es un sitio web sobre gatos">

* Para incluir palabras clave relevantes para el contenido de la página.

<meta name="keywords" content="gatos, juguetes para gato, adopción">

* La etiqueta meta name="author" para especificar el autor de la página.

<meta name="author" content="Armando Casas Velarde">

1. **¿Qué es Open Graph?**

Es un protocolo creado por Facebook que le permite añadir metadatos estructurados a las páginas de su sitio web. Estos metadatos pueden ser utilizados por las plataformas de redes sociales como el mismo Facebook, LinkedIn y Twitter (que implementa su propio estándar) para mostrar vistas previas más detalladas de nuestro contenido, a partir de las etiquetas para especificar el título del artículo, la imagen a mostrase y la descripción del producto.

1. **¿Cómo incluimos a open graph dentro de nuestra página web?**

En esencia, es similar a la declaración de etiquetas meta normales, solo que en vez de usar *name* se utiliza *property* como atributo, y este llevara un *og:* y el tipo de metadato a incluir.

<meta property="og:title" content="Amantes de los Gatos">

<meta property="og:description" content="Este es un sitio web sobre gatos">

<meta property="og:url" content="amantesdegatos.io">

<meta property="og:image" content="https://i.pinimg.com/1200x/23/e7/80/23e780c25eb0f70c3411b196df71e1fd.jpg">

1. **¿Qué perdemos y no incluimos a open graph dentro de nuestra página web?**

Perdemos la oportunidad de optimizar nuestra presencia en redes, ya que esta es una herramienta muy útil para llevar a cabo este propósito. También existe el problema de que vista previa poco atractiva o incompleta de nuestra página cuando se comparte en línea, como se puede visualizar en la siguiente imagen.



1. **¿Qué es Twitter Card?**

Es un tipo de contenido que está diseñado para mejorar la experiencia de los usuarios en Twitter (ahora X) al acompañar los tuits con información adicional como imágenes , videos y experiencias multimedia. Esto mediante las etiquetas de metadatos Open Graph necesarias

1. **¿Qué pierde nuestra página web si no incluye recursos Twitter Card?**

Perdemos la posibilidad de que los enlaces a su sitio web se muestren con una imagen y un resumen enriquecido en Twitter. Esto puede hacer que los enlaces a la página sean menos atractivos para los usuarios de Twitter y, por lo tanto, reducir el tráfico al sitio web.

1. **¿Cómo se agregan recursos Twitter Card dentro de nuestra página web?**

Similar a lo visto con Open Graph, para agregar los recursos declaramos los atributos en las etiquetas meta: name="twitter:card", name="twitter:title", name="twitter:description", y name="twitter:image". Al agregar estas etiquetas, se puede mejorar la apariencia de la página al compartir en Twitter y aumentar la visibilidad de la página en línea.

1. **¿Cómo se programan los controles en un formulario web?**

Se utilizan diferentes lenguajes de programación, como HTML, CSS y JavaScript. En HTML, se utilizan las etiquetas <form> para crear el formulario y <input> para crear los diferentes controles, como botones, casillas de verificación y campos de texto. Luego, en CSS se puede personalizar la apariencia de los controles y en JavaScript se pueden agregar interacciones y validaciones a los controles.

# BIBLIOGRAFÍA

1. G. Bustos. (2023, January 10). ¿Qué es HTML? Explicación de los fundamentos del lenguaje de marcado de hipertexto, Tutoriales Hostinger [Online]. Available: <https://www.hostinger.es/tutoriales/que-es-html>
2. M. D. Net. (2023, July 18). Conceptos Básicos de HTML [Online] Available: <https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics>
3. H. Delgado. (2022, August 22). Estructura básica de una página Web - html, head y body [Online]. Available: <https://disenowebakus.net/domine-html-y-dhtml-primeros-pasos.php>
4. E. Castellanos. (2023, January 6). Elementos semánticos HTML5 explicados [Online]. Available: <https://www.freecodecamp.org/espanol/news/elementos-semanticos-html5-explicados/>
5. R. Mendoza. (2023, June 14). HTML Semántico: Qué Es y Cómo Usarlo Correctamente [Online]. Available: <https://es.semrush.com/blog/html-semantico/>
6. M. D. Net. (2023, August 2). Categorías de contenido [Online]. Available: <https://developer.mozilla.org/es/docs/Web/HTML/Content_categories>
7. G. Pages. (2023). Acerca de GitHub Pages [Online]. Available: <https://docs.github.com/es/enterprise-cloud@latest/pages/getting-started-with-github-pages/about-github-pages>
8. G. Pages. (2023). Crear un sitio de Páginas de GitHub [Online]. Available: <https://docs.github.com/es/enterprise-cloud@latest/pages/getting-started-with-github-pages/creating-a-github-pages-site>
9. D. Diaz. (2023, March 1). Preparados, Listos, Lanzamiento: Creación de un Sitio Estático con GitHub Pages [Online]. Available: <https://kinsta.com/es/blog/github-pages/>
10. Eniun. (2023, August 4). Etiquetas para la creación de formularios HTML [Online]. Available: <https://www.eniun.com/etiquetas-formularios-html5/>
11. A. Web. (2023, July 4). Tipos de INPUT en HTML [Online]. Available: <https://www.adaweb.es/tipos-de-input-en-html/>
12. R. Anderson. (2023, August 3). Validación de modelos en ASP.NET Core MVC y Razor Pages [Online]. Available: <https://learn.microsoft.com/es-es/aspnet/core/mvc/models/validation?view=aspnetcore-7.0>
13. M. D. Net. (2023, August 2). Validación de formularios de datos [Online]. Available: <https://developer.mozilla.org/es/docs/Learn/Forms/Form_validation>
14. Wikipedia. (2023, May 22). Expresión regular [Online]. Available: <https://es.wikipedia.org/wiki/Expresión_regular>
15. T. Firdaus. (2021, April 21). HTML5 Validación de Formularios con el atributo “pattern” [Online]. Available: <https://webdesign.tutsplus.com/es/html5-form-validation-with-the-pattern-attribute--cms-25145t>
16. M. D. Net. (2022, November 25). Pseudo-classes [Online]. Available: <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>
17. M. D. Net. (2023, July 21). :valid [Online]. Available: <https://developer.mozilla.org/es/docs/Web/CSS/:valid>
18. M. D. Net. (2023, July 24). :invalid [Online]. Available: <https://developer.mozilla.org/es/docs/Web/CSS/:invalid>
19. H. Quick. (2023) Tablas en HTML [Online]. Available: <https://www.htmlquick.com/es/tutorials/tables.html>
20. M. D. Net. (2023, August 2). [Online]. Available: <https://developer.mozilla.org/es/docs/Learn/HTML/Tables/Basics>
21. K. (2022, December 15). ¿Cómo crear tablas en HTML? [Online]. Available: <https://keepcoding.io/blog/como-crear-tablas-en-html/>