

Generales:

Licenciatura en:	Administración de Tecnologías de la Información		
Asignatura:	Arquitectura de la Información		
Asignatura antecedente:	N/A	Asignatura consecuente:	N/A
Objetivo de la asignatura:	Distinguir los fundamentos teóricos de la información, así como el funcionamiento y aplicación de los sistemas de información con el enfoque de negocio dentro de las organizaciones, entendiendo las necesidades actuales de los sistemas y el uso de Big Data.		
Área EGEL:		Sub área EGEL:	
Dirección académica responsable:	Ingenierías		
Nombre del profesor:			
Nombre del alumno:			

Datos de la Práctica 1

Práctica 1 de 5:	Estructura de contenido web con HTML
Objetivo de la práctica:	Estructurar la información de un <i>curriculum vitae</i> con tecnología HTML para representar los datos en formato de párrafos, encabezados, imágenes y listas en el diseño de páginas web.
Temas y subtemas asociados:	1. Fundamentos de la información 1.1 Dato/Información 1.3 Tiempo de vida de la información
Fecha:	
Duración (horas):	2 horas
Laboratorio:	Laboratorio de cómputo
Equipo de seguridad para ingresar al laboratorio (indispensable):	N/A
Software requerido:	<i>Netbeans IDE</i> con <i>plugin HTML 5 Kit</i> o acceso a la <i>IDE</i> basada en la nube <i>Replit</i> (Replit, 2022)
Equipo necesario en laboratorio:	Computadora con acceso a Internet
Material/Sustancias/Reactivos disponible en laboratorio:	N/A
Material/Sustancias/Reactivos aportado por el alumno:	N/A

Desarrollo práctica 1:

Indicaciones de la práctica:

1. Abre la *IDE* disponible en el laboratorio.
2. Crea un nuevo proyecto en la *IDE* a utilizar del tipo *HTML, CSS, Javascript*.
3. Nombra tu proyecto: *curriculum*
4. Una vez creado el proyecto, crea o abre un archivo llamado *index.html*
5. Copia la siguiente estructura básica de un documento HTML (ver figura 1):

Figura 1. Estructura básica de documento HTML

```
index.html x
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width">
7   <title>Curriculum Vitae</title>
8   <link href="style.css" rel="stylesheet" type="text/css" />
9 </head>
10
11 <body>
12
13   <script src="script.js">
14
15   </script>
16 </body>
17
18 </html>
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

6. Con ayuda de elementos HTML como: encabezados, párrafos, imagen, listas ordenada y no ordenadas; desarrolla la estructura de tu *curriculum vitae* dentro del elemento *<body></body>* que contenga lo siguiente (ver figura 2):
 - a) Nombre completo, estructurado con un encabezado
 - b) Fotografía, estructurado con una imagen
 - c) Perfil profesional, estructurado con un encabezado y un párrafo.
 - d) Experiencia laboral, estructurado con un encabezado y una lista no ordenada.
 - e) Educación, estructurado con un encabezado y una lista no ordenada.
 - f) Habilidades, estructurado con un encabezado y una lista ordenada.

Desarrollo práctica 1:

Figura 2. Esquema de página web solicitada



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Nota: Si tienes dudas de cómo estructurar los elementos solicitados, puedes apoyarte de la documentación de MDN Web Docs (Mozilla, 2022).

7. Una vez estructurada tu página completa, valúa tu código y revisa los resultados en un navegador.

Resultados obtenidos:

Desarrollar una página web con extensión .html que describa la estructura de un *curriculum vitae*, debe ser accesible y visible por medio de un navegador web.

Evidencia de la práctica: (fotografías, videos, archivo, etc.)

Integrar el proyecto que contiene el archivo index.html del que forma parte la página web.

Conclusiones:

Describe los beneficios del uso de la tecnología HMTL para estructurar información a partir de elementos de contenido HTML y redacta las competencias adquiridas durante el desarrollo de la práctica.

Bibliografía:

De la web:

- Mozilla. (20 de Febrero de 2022). MDN Web Docs. Obtenido de Referencia de Elementos HTML: <https://developer.mozilla.org/es/docs/Web/HTML/Element>
- Replit. (20 de Febrero de 2022). Replit. Obtenido de Crea una cuenta en Replit: <https://replit.com/signup>

Criterios de evaluación:

- Valuar la página index.html, probándola en un navegador web y comprobar que cumpla con la estructura de un *curriculum vitae*.
- Código libre de errores y documentado.

Datos de la Práctica 2

Práctica 2 de 5:	Definición de la estructura de la información para un sitio web de comercio electrónico de venta de ropa de moda
Objetivo de la práctica:	Construir un sitio web para definir la estructura básica de un sitio web con sus partes principales de encabezado, contenido principal y pie de página apoyado de la tecnología HTML para demostrar las funciones de un arquitecto de la información.
Temas y subtemas asociados:	2. Bases de los sistemas de Información 2.1 Conceptos básicos 2.2 Las funciones del arquitecto de la información 2.3 Arquitecturas de Información
Fecha:	
Duración (horas):	2 horas
Laboratorio:	Laboratorio de cómputo
Equipo de seguridad para ingresar al laboratorio (indispensable):	N/A
Software requerido:	<i>Netbeans IDE con plugin HTML 5 Kit</i> o acceso a la <i>IDE</i> basada en la nube <i>Replit</i> (Replit, 2022)
Equipo necesario en laboratorio:	Computadora con acceso a Internet
Material/Sustancias/Reactivos disponible en laboratorio:	N/A
Material/Sustancias/Reactivos aportado por el alumno:	N/A

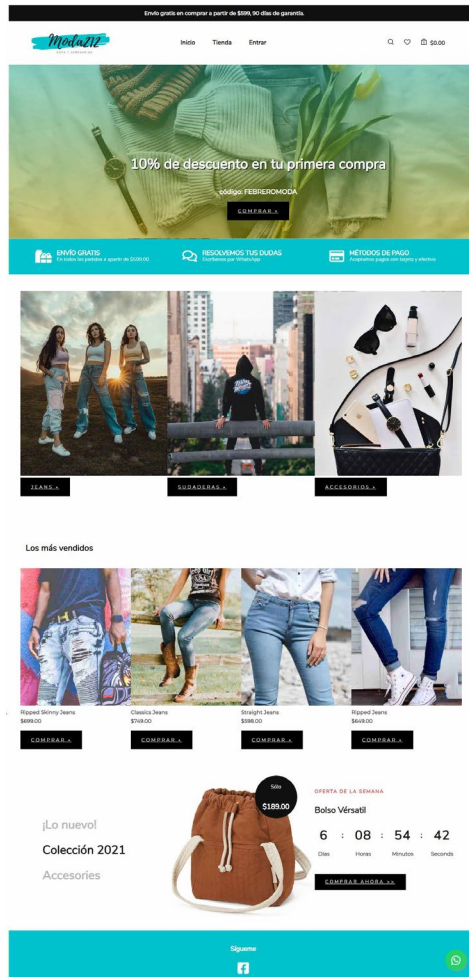
Desarrollo práctica 2:

Consideraciones previas al desarrollo de la práctica:

Durante el desarrollo de la práctica 2 y 3 se solicita la siguiente estructura de un sitio web de venta de ropa de moda (ver figura 3).

Desarrollo práctica 2:

Figura 3. Resultado esperado durante el desarrollo de práctica 2 y 3



Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información. En la práctica 2, su objetivo es desarrollar la siguiente estructura (ver figura 4) utilizando los elementos HTML de estructura (Mozilla, 2022).

Desarrollo práctica 2:

Figura 4. Propuesta de estructura de contenido para un sitio web de moda

BARRA SUPERIOR			
LOGO	MENU PRINCIPAL		LIGAS DE ICONOS
BANNER PRINCIPAL			
ICONO 1	ICONO 2		ICONO 3
CATEGORIA PRODUCTOS 1	CATEGORIA PRODUCTOS 2		CATEGORIA PRODUCTOS 3
PRODUCTO 1	PRODUCTO 2	PRODUCTO 3	PRODUCTO 4
BANNER PROMOCIONAL			
PIE DE PÁGINA			

Donde:

- La fila 1 y 2 corresponde al encabezado y se desarrollará con el elemento `<header></header>`
- La fila 3-7 corresponde al contenido principal y se desarrollará con el elemento `<main></main>`
- La fila 8 corresponde al pie de página y se desarrollará con el elemento `<footer></footer>`
- Cada fila se definirá con un elemento de estructura `<section></section>`
- Cada columna de cada fila puede ser definido con elementos `<div></div>` o `<article></article>`

Antes de comenzar tu práctica obtén nueve imágenes de licencia libre: tres imágenes para representar las categorías de los productos, cinco para representar cada producto, un logotipo, como recomendación, procura que sean de buen formato y mismo tamaño.

Indicaciones de la práctica:

1. Abre la *IDE* disponible en el laboratorio.
2. Crea un nuevo proyecto en la *IDE* a utilizar del tipo *HTML, CSS, Javascript*.
3. Nombra tu proyecto: *sitiomoda*
4. Una vez creado el proyecto, crea o abre un archivo llamado *index.html*
5. Copia la estructura principal con el siguiente código (ver figura 5) en el archivo *index.html*.

Desarrollo práctica 2:

Figura 5. Estructura inicial del documento index.html

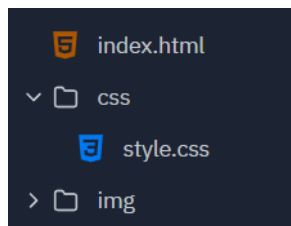
```
index.html x
1 <!DOCTYPE html>
2 <html lang="es">
3 <head><!--Inicia metadatos-->
4 <!-- Elementos requeridos -->
5 <meta charset="utf-8"> <!-- Especifica la codificación del documento -->
6 <meta name="viewport" content="width=device-width, initial-scale=1"> <!-- Prepara el documento para hacerlo responsivo -->
7 <meta name="keywords" content="moda, venta de ropa, fast fashion"> <!-- Relaciona las palabras clave, útil para SEO -->
8 <meta name="description" content="Venta de Ropa Fast Fashion en Línea"> <!-- Describe el documento, útil para SEO -->
9 <title>Venta de Ropa Fast Fashion - Moda 212</title><!-- Título de la página, útil para SEO -->
10 <!--Referencia a hoja de estilo-->
11 <link rel="stylesheet" type="text/css" href="css/style.css">
12 <!--Agrega un favicon-->
13 <link rel="icon" href="img/favicon.ico" type="image/x-icon">
14
15 </head><!--Termina metadatos-->
16 <body>
17 <!--Código visible para el usuario-->
18 </body>
19 </html>
```

Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información

Se integran etiquetas *meta* útiles para que sea encontrado el sitio web por buscadores como Google, mediante SEO (*Search Engine Optimization*, del español, Optimización para Máquinas de Búsqueda), además, se incluyen otros metadatos para referencias externas de estilos y *favicon*.

6. Dentro de tu proyecto, agrega una carpeta llamada *img*, en esta carpeta almacena las nueve imágenes solicitadas en las consideraciones previas de la práctica.
7. Agrega una carpeta llamada *css* y dentro de esta carpeta crea un archivo llamado *style.css*. La estructura de archivos debe quedar como en la figura 6.

Figura 6. Estructura de archivos del proyecto

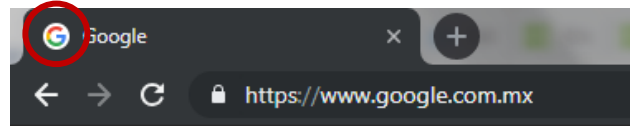


Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información

8. Regresa al archivo *index.html*, ahora guarda en tu proyecto (en la carpeta *img*) un archivo con extensión *.ico* para que se muestre una imagen en la pestaña del explorador, este elemento se llamada *favicon*, y se muestra en la pestaña del explorador de la página web (ver figura 7).

Desarrollo práctica 2:

Figura 7. Ejemplo de favicon en la página web de Google



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Para obtener el archivo .ico, puedes dirigirte a sitios web que generan el *favicon* a partir de una imagen, como *Favicon.ico & App Icon Generator* (Favicon.ico & App Icon Generator, 2022). Obtenido tu archivo con extensión .ico, almacénala en la carpeta de imágenes *img* e identifica la siguiente línea de código en tu archivo *index.html* (ver figura 8), si el nombre de tu archivo es diferente a *favicon.ico*, actualiza el nombre del archivo en tu código en la propiedad *href="img/favicon.ico"*, debe corresponder el nombre de tu archivo con el nombre al que se hace referencia en esta línea de código.

Figura 8. Elemento HTML para agregar un favicon a una página web

```

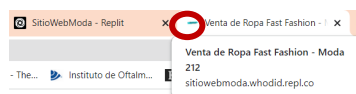
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <!-- Elementos requeridos -->
5   <meta charset="utf-8"> <!-- Especifica la codificación del documento -->
6   <meta name="viewport" content="width=device-width, initial-scale=1"> <!-- Prepara el documento para hacerlo responsive -->
7   <meta name="keywords" content="moda, venta de ropa, fast fashion"> <!-- Relaciona las palabras clave, útil para SEO -->
8   <meta name="description" content="Venta de Ropa Fast Fashion en Línea"> <!-- Describe el documento, útil para SEO -->
9   <title>Venta de Ropa Fast Fashion - Moda 212</title> <!-- Título de la página, útil para SEO -->
10  <!-- Elementos opcionales -->
11  <script src="script.js" type="text/javascript"></script>
12  <link rel="icon" href="img/favicon.ico" type="image/x-icon">

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

9. Evalúa el proyecto y prueba que la imagen de *favicon* sea visible en la pestaña del explorador donde revisas la página *index.html* (ver figura 9).

Figura 9. Resultado esperado al agregar un favicon a la página web.



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

10. Regresa al archivo *index.html* y dentro del elemento de *<body>* *</body>* se desarrolla la estructura de la barra superior (ver figura 3) con el siguiente código (ver figura 10):

Desarrollo práctica 2:

Figura 10. Código que estructura la barra superior.

```

10 <!-- Termina barra superior -->
11 <link rel="stylesheet" type="text/css" href="css/style.css">
12 <!-- Agrega un favicon -->
13 <link rel="icon" href="img/favicon.ico" type="image/x-icon">
14
15 </head><!-- Termina metadatos -->
16 <body>
17 <!-- Código visible para el usuario -->
18 <!-- Aquí inicia el código visible para el usuario -->
19 <header class="header"><!-- Inicia encabezado -->
20 <section class="header-top"><!-- Inicia barra superior -->
21 <div class="container">
22 <div class="row">
23 <div class="one-column">
24 <div class="header-top-center">
25 <p>Envío gratis en comprar a partir de $599, 90 días de garantía.</p>
26 </div>
27 </div>
28 </div>
29 </div>
30 </section><!-- Termina barra superior -->
31 </header><!-- Termina encabezado -->
32 </body>
33 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Integra cuidadosamente los atributos *class*, ya que haremos referencia a ellos en la siguiente práctica, personaliza el texto del elemento `<p></p>`.

11. Donde termina `</section>` y antes de `</header>`, copia el siguiente código (ver figura 11) que estructura la fila que incluye el logotipo, menú y ligas a iconos que pertenece al encabezado (ver figura 3).

Figura 11. Código que estructura sección de logotipo, menú y liga a iconos.

```

39 </div>
40 </section><!-- Termina barra superior -->
41 <section><!-- Inicia barra logo - Menu -->
42 <div class="container"><!-- Estructura el contenido de tabla -->
43 <div class="row"><!-- Estructura el contenido de fila -->
44 <article class="quarter-column"><!-- Inicia logo, columna 1 -->
45 <div class="header-logo">
46 <a href="index.html"></a>
47 </div>
48 </article><!-- Termina logo -->
49 <article class="half-column"><!-- Inicia menu, columna 2 -->
50 <nav class="header-menu">
51 <ul>
52 <li><a href="index.html">Inicio</a></li>
53 <li><a href="shop.html">Tienda</a></li>
54 <li><a href="login.html">Entrar</a></li>
55 </ul>
56 </nav>
57 </article><!-- Termina menu, columna 3 -->
58 <article class="quarter-column"><!-- Inicia liga de iconos, columna 4 -->
59 <div class="header-nav-option">
60 <a href="#"><span><i class="fas fa-search"></i></span></a>
61 <a href="#"><span><i class="fas fa-shopping-bag"></i></span></a>
62 <a href="#"><span><i class="far fa-heart"></i></span></a>
63 </div>
64 </article><!-- Termina liga de iconos, columna 5 -->
65 </div><!-- Termina estructura de fila -->
66 </div><!-- Termina estructura de tabla -->
67 </section><!-- Termina barra logo - Menu -->
68 </header><!-- Termina encabezado -->

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Reemplaza la ruta de la imagen "img/logo.jpg" por el nombre de tu logotipo, revisa que tu imagen de logotipo este en la carpeta /img de tu proyecto.

Desarrollo práctica 2:

12. En la sección `<head></head>`, justo antes del cierre de `</head>` (ver figura 12), copia la siguiente línea de código:

```
<!--Referencia de framework FontAwesome-->
<script src="https://kit.fontawesome.com/d6098394de.js" crossorigin="anonymous"></script>
```

El *script* anterior, agrega el *framework FontAwesome* (FontAwesome, 2022) a tu página web, el cual permite agregar iconos y resolverlos en tu página web, los iconos son estructurados con los elementos `<i></i>`

Figura 12. Se muestra en dónde debe agregarse la referencia a FontAwesome

```
8 <meta name="keywords" content="moda, venta de ropa, fast fashion"> <!-- Descripción de documento, útil para SEO -->
9 <meta name="description" content="Venta de Ropa Fast Fashion en línea"> <!-- Descripción del documento, útil para SEO -->
10 <title>Venta de Ropa Fast Fashion - Moda 212</title> <!-- Título de la página, útil para SEO -->
11 <link rel="stylesheet" type="text/css" href="css/style2.css">
12 <!-- Icono de favicon -->
13 <link rel="icon" href="img/favicon.ico" type="image/x-icon">
14 <!-- Referencia de framework FontAwesome -->
15 <script src="https://kit.fontawesome.com/d6098394de.js" crossorigin="anonymous"></script>
16
17 </head> <!-- Fin de metadatos -->
18 <body>
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

13. Después de `</header>` se desarrolla el *contenido principal* (Ver figura 3), integra el elemento `<main></main>` antes del cierre de `</body>` (ver figura 13).

Figura 13. Estructura que define el contenido principal

```
59 </div> <!-- Termina estructura de tabla -->
60 </section> <!-- Termina Barra Logo - Menu -->
61 <main> <!-- Inicia contenido principal -->
62
63 </main> <!-- Termina contenido principal -->
64 </body>
65 </html>
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

14. Dentro de `<main></main>` copia el siguiente código (ver figura 14) que estructura el *banner* principal (ver figura 3) se utiliza el elemento `<section>`

Figura 14. Código que estructura el banner principal

```
59 </div> <!-- Termina estructura de tabla -->
60 </section> <!-- Termina Barra Logo - Menu -->
61 <main> <!-- Inicia contenido principal -->
62 <section class="bgimage"> <!-- Banner -->
63 <div class="container-fluid">
64 <div class="row">
65 <div class="background">
66 <h3>10% de descuento en tu primera compra</h3>
67 <p>código: FEBREROMODA</p>
68 <p><a href="shop.html" class="primary-btn">Comprar </a></p>
69 </div>
70 </div>
71 </div>
72 </section> <!-- Termina Banner -->
73 </main> <!-- Termina contenido principal -->
74 </body>
75 </html>
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Desarrollo práctica 2:

15. Después de `</section>` y antes de `</main>`, copia el siguiente código (ver figura 15) que estructura la fila con 3 iconos (ver figura 3), personaliza los textos de los elementos `<p></p>`.

Figura 15. Código que estructura una fila con tres iconos

```

71 </div>
72 </section> <!-- Fin de Banner -->
73 <section class="icon-bg"> <!-- Inicio Iconos -->
74 <div class="container"> <!-- Estructura el contenido de la fila -->
75 <div class="row">
76 <article class="onethird-column"> <!-- Estructura la columna 1 -->
77 <div class="icon">
78 <i class="fas fa-gifts"></i> <!-- Elemento de Icono -->
79 </div>
80 <div class="icon-content">
81 <span class="icon-caption"> ENVÍO GRATIS </span>
82 <span class="icon-subcaption"> En todos los pedidos a partir de $599 </span>
83 </div>
84 </article> <!-- Fin de Estructura la columna 1 -->
85 <article class="onethird-column"> <!-- Estructura la columna 2 -->
86 <div class="icon">
87 <i class="far fa-comments"></i> <!-- Elemento de Icono -->
88 </div>
89 <div class="icon-content">
90 <span class="icon-caption"> RESOLVEMOS TUS DUDAS </span>
91 <span class="icon-subcaption"> Escríbenos por WhatsApp </span>
92 </div>
93 </article> <!-- Fin de Estructura la columna 2 -->
94 <article class="onethird-column"> <!-- Estructura la columna 3 -->
95 <div class="icon">
96 <i class="far fa-credit-card"></i> <!-- Elemento de Icono -->
97 </div>
98 <div class="icon-content">
99 <span class="icon-caption"> MÉTODOS DE PAGO </span>
100 <span class="icon-subcaption"> Aceptamos pagos con tarjeta y efectivo </span>
101 </div>
102 </article> <!-- Fin de Estructura la columna 3 -->
103 </div> <!-- Estructura el contenido de la fila -->
104 </div> <!-- Fin de Estructura el contenido de la fila -->
105 </section>
106 </main> <!-- Fin de contenido principal -->
107 </body>
108 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

16. Después de `</section>` y antes de `</main>`, copia el siguiente código (ver figura 16) que estructura la fila de las 3 categorías de productos (ver figura 3).

Figura 16. Código que estructura la fila de categorías

```

109 </div>
110 </section> <!-- Fin de Banner -->
111 <section class="categorias"> <!-- Inicio de la categoría -->
112 <div class="container"> <!-- Estructura el contenido de la fila -->
113 <div class="row">
114 <article class="onethird-column"> <!-- Estructura la columna 1 -->
115 <div class="container">
116 
117 <a href="shop.html" class="primary-btn">Jeans </a></p>
118 </div>
119 </article> <!-- Fin de Estructura la columna 1 -->
120 <article class="onethird-column"> <!-- Estructura la columna 2 -->
121 <div class="container">
122 
123 <a href="shop.html" class="primary-btn">Sudaderas </a></p>
124 </div>
125 </article> <!-- Fin de Estructura la columna 2 -->
126 <article class="onethird-column"> <!-- Estructura la columna 3 -->
127 <div class="container">
128 
129 <a href="shop.html" class="primary-btn">Accesorios </a></p>
130 </div>
131 </article> <!-- Fin de Estructura la columna 3 -->
132 </div> <!-- Estructura el contenido de la fila -->
133 </div> <!-- Fin de Estructura el contenido de la fila -->
134 </section>
135 </main> <!-- Fin de contenido principal -->
136 </body>
137 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Reemplaza la ruta de las imágenes para que corresponda a la de tus fotos y cambia el contenido de los párrafos `<p></p>` para que coincida con las categorías que se quieren mostrar.

Desarrollo práctica 2:

17. Después de `</section>` y antes de `</main>`, copia el siguiente código (ver figura 17) que estructura una fila con 4 productos (ver figura 3).

Figura 17. Código que estructura una fila con 4 productos

```

128 </div> <!-- Fin de estructura de contenido de tabla -->
129 </section> <!-- Fin de sección de categorías -->
130 <section class="categorias"> <!-- Inicio sección de categorías -->
131 <h2>Los más vendidos</h2><br>
132 <div class="container"> <!-- Estructura de contenido de tabla -->
133 <div class="row"> <!-- Estructura de fila -->
134 <article class="quarter-column"> <!-- Estructura la columna 1 -->
135 <div class="container">
136 
137 <p>Ripped Skinny Jeans</p><b>$699.00</b></p>
138 <a href="shop.html" class="primary-btn">Comprar </a></p>
139 </div>
140 </article> <!-- Fin de estructura de columna 1 -->
141 <article class="quarter-column"> <!-- Estructura la columna 2 -->
142 <div class="container">
143 
144 <p>Classic Jeans</p><b>$749.00</b></p>
145 <a href="shop.html" class="primary-btn">Comprar </a></p>
146 </div>
147 </article> <!-- Fin de estructura de columna 3 -->
148 <article class="quarter-column"> <!-- Estructura la columna 4 -->
149 <div class="container">
150 
151 <p>Straight Jeans</p><b>$590.00</b></p>
152 <a href="shop.html" class="primary-btn">Comprar </a></p>
153 </div>
154 </article> <!-- Fin de estructura de columna 4 -->
155 </div> <!-- Fin de estructura de fila -->
156 </div> <!-- Fin de estructura de contenido de tabla -->
157 </div> <!-- Fin de estructura de contenido de fila -->
158 </div> <!-- Fin de estructura de contenido de tabla -->
159 </div> <!-- Fin de estructura de contenido de fila -->
160 </div> <!-- Fin de estructura de contenido de tabla -->
161 </div> <!-- Fin de estructura de contenido de fila -->
162 </div> <!-- Fin de estructura de contenido de tabla -->
163 </div> <!-- Fin de estructura de contenido de fila -->
164 </div> <!-- Fin de estructura de contenido de tabla -->
165 </main> <!-- Fin de contenido principal -->
166 </body>
167 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Reemplaza la ruta de las imágenes para que corresponda a la de las fotos y cambia el contenido de `<p></p>` para que coincida con el nombre del producto y precio.

18. Después de `</section>` y antes de `</main>`, copia el siguiente código (ver figura 18), que estructura el banner promocional (ver figura 3).

Desarrollo práctica 2:

Figura 18. Código que estructura el banner promocional

```

165 </section> <!-- Fin de la sección de categorías -->
166 <section class="categorias"> <!-- Fin de la sección de categorías -->
167 <div class="container"> <!-- Estructura el contenido de la fila -->
168 <div class="row"> <!-- Estructura el contenido de la fila -->
169 <article class="onethird-column"> <!-- Estructura la columna 1 -->
170 <div class="offer-counter">
171 <h2>¡Lo nuevo! </h2> <span>Colección 2022</span> <hr> Accesorios</h2>
172 </div>
173 </article> <!-- Fin de la columna 1 -->
174 <article class="onethird-column"> <!-- Estructura la columna 2 -->
175 <div class="offer-price">
176 
177 <span>Sólo</span>
178 <h2>$199.00</h2>
179 </div>
180 </article> <!-- Fin de la columna 2 -->
181 <article class="onethird-column"> <!-- Estructura la columna 3 -->
182 <div class="offer-counter">
183 <span>Oferta de la Semana</span>
184 <h2>80% Vértigo</h2>
185 <div class="offer-counter-clock" id="countdown">
186 <div class="ed-item"><span>6</span></div>
187 <div class="ed-item"><span>06</span></div>
188 <div class="ed-item"><span>54</span></div>
189 <div class="ed-item"><span>42</span></div>
190 </div>
191 <a href="shop.html" class="primary-btn">Comprar Ahora >></a>
192 </div>
193 </article> <!-- Fin de la columna 3 -->
194 </div> <!-- Fin de la sección de categorías -->
195 </section> <!-- Fin de la sección de categorías -->
196 </div> <!-- Fin de la sección de categorías -->
197 </main> <!-- Fin de la sección principal -->
198 </body>
199 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Reemplaza la ruta de la imagen del producto promocional (*img/product-sale.png*), los textos que lo describen y su precio.

- Después de `</section>` y antes de `</main>`, copia el siguiente código (ver figura 19), que estructura la liga que llevará a una conversación a la aplicación de *WhatsApp* (ver figura 3).

Figura 19. Código que estructura un botón que liga a la aplicación de WhatsApp

```

195 </div> <!-- Estructura el contenido de la fila -->
196 </div> <!-- Fin de la sección de categorías -->
197 </section> <!-- Fin de la sección de categorías -->
198 <section>
199 <a href="#" class="whatsapp" target="_blank">
200 <i class="fab fa-whatsapp my-whatsapp"></i>
201 </a>
202 </section>
203 </main> <!-- Fin de la sección principal -->
204 </body>
205 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Reemplaza el carácter `#` del atributo *href* por la ruta a un teléfono de *WhatsApp* con un mensaje, investiga en la documentación de la aplicación de *WhatsApp* como realizarlo (WhatsApp, 2022).

- Después de `</section>` y antes de `</main>`, copia el siguiente código (ver figura 20), que estructura la sección del pie de página (ver figura 3).

Desarrollo práctica 2:

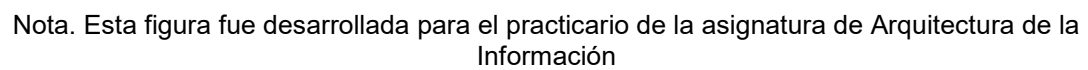
Figura 20. Código que estructura el pie de página

```
282 </section></div></div></div>
283 </body></html>
284 </div></div>
285 <div>
286 <a href="https://www.facebook.com/"><i class="fab fa-facebook-square"></i></a>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

- 21.** Una vez que se tiene el código completo, valúa la estructura ejecutando el código y prueba el resultado en una página del explorador; revisa la sección de Resultados obtenidos para validar tu resultado, en caso contrario, analiza tu código para corregir los errores.

Figura 21. Estructura de contenido esperada en práctica 2



Resultados obtenidos:

Nota: En las imágenes las imágenes fueron reducidas para efectos visuales, en la práctica, cada imagen debe ocupar el ancho de toda la pantalla.

Evidencia de la práctica: (fotografías, videos, archivo, etc.)

Integrar la estructura que conforman el proyecto de la página web (archivo index.html, carpeta css con archivo style.css en blanco, carpeta img con imágenes)

Conclusiones:

Describe los retos que te presentaste al estructurar una página web y cómo puedes aplicar los conocimientos adquiridos a proyectos reales.

Bibliografía:

De la web:

- Favicon.ico & App Icon Generator. (27 de Febrero de 2022). Favicon.ico & App Icon Generator. Obtenido de Generate your favicon.ico: <https://www.favicon-generator.org/>
- FontAwesome. (27 de Febrero de 2022). FontAwesome. Obtenido de FontAwesome: <https://fontawesome.com/>
- Google. (27 de Febrero de 2022). Google Font. Obtenido de Google Font: <https://fonts.google.com/>
- Google. (27 de Febrero de 2022). Google Font. Obtenido de Get Started with the Google Fonts API: https://developers.google.com/fonts/docs/getting_started
- Mozilla. (27 de 02 de 2022). MDN Web Docs. Obtenido de Estructura web y documentación: https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure
- Replit. (20 de Febrero de 2022). Replit. Obtenido de Crea una cuenta en Replit: <https://replit.com/signup>
- WhatsApp. (27 de Febrero de 2022). WhatsApp. Obtenido de Cómo usar el clic para chatear: <https://faq.whatsapp.com/general/chats/how-to-use-click-to-chat>

Criterios de evaluación:

- Valuar la página index.html, probándola en un navegador web y comprobar que la estructura cumpla con los solicitado en la práctica.
- Código libre de errores y documentado.

Datos de la Práctica 3

Práctica 3 de 5:	Aplicar diseño a una página web de comercio electrónico de venta de ropa de moda
Objetivo de la práctica:	Usar la tecnología de hojas de estilo en cascada, mejor conocida como CSS, para aplicar diseño a una página web con la finalidad de demostrar las funciones de un arquitecto de la información.
Temas y subtemas asociados:	2. Bases de los sistemas de Información 2.1 Conceptos básicos 2.2 Las funciones del arquitecto de la información 2.3 Arquitecturas de Información
Fecha:	
Duración (horas):	2 horas
Laboratorio:	Laboratorio de cómputo
Equipo de seguridad para ingresar al laboratorio (indispensable):	N/A
Software requerido:	<i>Netbeans IDE con plugin HTML 5 Kit</i> o acceso a la <i>IDE</i> basada en la nube <i>Replit</i> (Replit, 2022)
Equipo necesario en laboratorio:	Computadora con acceso a Internet
Material/Sustancias/Reactivos disponible en laboratorio:	N/A
Material/Sustancias/Reactivos aportado por el alumno:	N/A

Desarrollo práctica 3:

Consideraciones previas al desarrollo de la práctica:

- Tener completa y sin errores la **Práctica 2. Definición de la estructura de la información para un sitio web de comercio electrónico de venta de ropa de moda**.
- A lo largo de la práctica se utilizan propiedades CSS de la Tabla 1, para más detalle y ejemplos puedes leer de la documentación MDN Web Docs (Mozilla, 2022)

Tabla 1. Propiedades CSS básicas

Propiedad	Descripción	Ejemplo
<i>color</i>	Establece el contenido de un elemento a un color dado, aplica principalmente a textos.	<i>color: red;</i>
<i>background</i>	Establece el color de fondo de un elemento También puede personalizar el fondo con una imagen.	<i>background: #ffffff;</i> También se puede usar <i>background-color:#ffffff;</i> Para imágenes la definición es:

Desarrollo práctica 3:

		<i>background:</i> <i>url(../img/home.jpg);</i>
<i>border-style</i>	Define el borde de un elemento, puede ser sólido, punteado, sin borde, etc.	<i>border-style: solid;</i>
<i>border-width</i>	Define el ancho del borde de un elemento, dado en <i>píxeles</i> .	<i>border-width: 3px;</i>
<i>border-color</i>	Define el color de un borde de un elemento.	<i>border-color: white;</i>
<i>margin</i>	Define el margen de un elemento de los cuatro lados. Normalmente dado en <i>píxeles</i> .	<i>margin: 10px;</i>
<i>margin-top</i>	Define solo el margen superior del elemento.	<i>margin-top: 3px;</i>
<i>margin-right</i>	Define solo el margen derecho del elemento.	<i>margin-right: 5px;</i>
<i>padding</i>	Define un espacio invisible de relleno alrededor del elemento.	<i>padding: 5px;</i>
<i>padding-top</i>	Define un espacio invisible de relleno sólo arriba del elemento.	<i>padding-top: 2px;</i>
<i>padding-bottom</i>	Define un espacio invisible de relleno sólo abajo del elemento.	<i>padding-bottom: 2px;</i>
<i>padding-left</i>	Define un espacio invisible de relleno sólo de lado izquierdo del elemento.	<i>padding-left: 1px;</i>
<i>float</i>	Define si el elemento debe salir de su flujo normal y aparecer a la izquierda o derecha de su contenedor.	<i>float: left;</i> <i>float: none;</i>
<i>display</i>	Especifica el tipo de cuadro de representación utilizado para un elemento.	<i>display: block;</i> <i>display: none;</i> <i>display: inline-block;</i>
<i>list-style</i>	Permite cambiar el diseño del <i>bullet</i> de una lista para que se vea como círculo, cuadrado o sin <i>bullet</i>	<i>list-style: circle;</i> <i>list-style: square inside;</i> <i>list-style: none;</i>
<i>text-align</i>	Permite alinear texto dentro de un elemento; a la izquierda, derecho, justificado, etc.	<i>text-align: left;</i> <i>text-align: justify;</i>
<i>text-decoration</i>	Define apariencia de líneas decorativas en el texto, útil en ligas para visualizar u ocultar el subrayado.	<i>text-decoration: none;</i>
<i>text-transform</i>	Define el formato del texto; mayúsculas, minúsculas, etc.	<i>text-transform: uppercase;</i>
<i>font-family</i>	Define el tipo de fuente en que se mostrará el texto.	<i>font-family: Helvetica;</i> <i>font-family: 'Tangerine ';</i>

Desarrollo práctica 3:

		Las tipografías de Google van entre comilla simple.
<i>font-style</i>	Permite definir el aspecto de una familia tipográfica de un texto: itálica, normal, etc.	<i>font-style: normal;</i> <i>font-style: italic;</i>
<i>font-size</i>	Define el tamaño de la letra en <i>píxeles</i> .	<i>font-size: 12px;</i>
<i>font-weight</i>	Especifica el grueso de la letra, algunos tipos de letra solo tiene disponible normal y <i>bold</i> .	<i>font-weight: bold;</i> <i>font-weight: 200;</i>
<i>height</i>	Especifica la altura del elemento HTML, normalmente dado en <i>píxeles</i> o porcentaje.	<i>height: 10px;</i> <i>height: 100%;</i>
<i>width</i>	Especifica el ancho del elemento HTML, normalmente dado en <i>píxeles</i> o porcentaje.	<i>width: 300px;</i> <i>width: 20%;</i>
<i>line-height</i>	Establece la altura entre líneas de texto.	<i>line-height: 25px;</i>
<i>letter-spacing</i>	Espacio horizontal entre letras.	<i>letter-spacing: 4px;</i>
<i>max-width</i>	Ancho máximo del elemento.	<i>max-width: 1044px;</i>
<i>box-sizing</i>	Calcula el ancho y alto de un elemento. Sus valores posibles son <i>border-box</i> y <i>content-box</i>	<i>box-sizing: border-box;</i>

Indicaciones de la práctica:

1. Abre la *IDE* disponible en el laboratorio.
2. Abre el proyecto desarrollado en la práctica 2.
3. Comprueba que tengas la carpeta *css* y dentro el archivo *style.css* (ver figura 6).
4. Abre tu archivo *index.html* y comprueba que en el `<head></head>` tengas disponible la referencia `<link rel="stylesheet" type="text/css" href="css/style.css">`
5. Dentro del elemento `<head>` y antes del elemento de cierre `</head>` copia la siguiente referencia (ver figura 22), que corresponde a la referencia del *API* de *Google Font* (Google, 2022), que sirve para personalizar la tipografía de un sitio web.

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Nunito+Sans&family=Montserrat">
```

Donde:

Nunito+Sans y *Montserrat* corresponde al nombre de la tipografía a personalizar, visita la página web de la *API* de *Google Font* (Google, 2022) y obtén dos nombres de tipografías con las que quieras personalizar tu página web y sustituye los nombres en la referencia anterior, considera que cuando el

Desarrollo práctica 3:

nombre de una tipografía consta de más de dos palabras, escribe en la referencia cada palabra separada con el símbolo +, para más información puedes visitar la documentación de la *API de Google Font* (Google, 2022)

Nota: Más adelante haremos referencia a los nombres de las tipografías elegidas en el archivo *style.css*

Figura 22. Referencia a la API de Google Font

```
13 <link rel="icon" href="img/favicon.ico" type="image/x-icon">
14 <script src="https://kit.fontawesome.com/d699394de.js" crossorigin="anonymous"></script>
15
16 <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Nunito+Sans&family=Montserrat">
17
18 </head>
19 <body>
20
```

Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información

- Ahora, abre el archivo *style.css* y se copia el siguiente código (ver figura 23).

Figura 23. Código inicial de reglas CSS para las configuraciones generales

```
1 /*
2 Configuraciones generales
3 */
4 html,
5 body {
6   /*Se establece la altura del área del contenido HTML*/
7   height: 100%;
8   /*Se establece el tipo de fuente, si se utilizan fuentes de Google,
9   se especifica el nombre de la fuente con comillas simples ' */
10  font-family: 'Nunito Sans';
11 }
12
13 /*Regla que aplica a todos los párrafos*/
14 p {
15   /*Nombre de la fuente*/
16   font-size: 15px;
17   /*Tipo de fuente de Google*/
18   font-family: 'Montserrat';
19   color: #3d3d3d;
20   font-weight: bold;
21   /*Alinea sobre líneas de texto*/
22   line-height: 25px;
23   /*Se especifica el tipo de letra de la fuente de Google*/
24   margin: 0 0 15px 0;
25 }
26
27
```

Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información

Donde:

- El texto encerrado en los caracteres `/* */` son comentarios que explican lo que hace la regla
- `html, body` son selectores de elemento HTML y las reglas contenidas entre `{ }` son las propiedades que aplican a estos elementos del documento `index.html`, como `height: 100%;` y `font-family: 'Nunito Sans';`
- `p` es un selector de elemento HTML y las reglas contenidas entre `{ }` son las que aplican a todos los párrafos del documento `index.html`
- Cada regla incluye un comentario que indica el diseño que aplica, puedes apoyarte de la Tabla 1 para más detalle.

- Utiliza la propiedad *font-family* que define el tipo de tipografía, siendo el valor el nombre de una de las tipografías de Google elegidas en el paso 5, el nombre va entre comilla simple `'`. Valúa el estilo ejecutando el proyecto y probando la página `index.html` en el navegador, la tipografía elegida debe ser aplicada.

Desarrollo práctica 3:

8. En el archivo *style.css* y enseguida del código agregado en el paso 7, copia el siguiente código (ver figura 24), las reglas CSS aplican diseño a todas las ligas para que se muestren como un botón:

Figura 24. Código que aplica diseño a ligas.

```

28
29 /*Selector de clase que aplica un estilo de
30 texto a las ligas caso/ase/
31 .primary-btn {
32   /* Selector el tipo de cuadro de representación utilizado/
33   display: inline-block;
34   /* tamaño de letras/
35   font-size: 13px;
36   /* color de texto/
37   font-weight: 500;
38   /*Selector el texto en mayúsculas/
39   text-transform: uppercase;
40   /*Selector aplica ancho primer valor. Espaciado derecho segundo valor/
41   padding: 14px 30px;
42
43   color: #ffffff;
44   /*Selector de fondo/
45   background: #000000;
46   /*Selector horizontal para letras/
47   letter-spacing: 4px;
48 }
49 /*Selector de clase que aplica un estilo de
50 botón cuando el mouse está sobre la liga/
51 .primary-btn:hover {
52   /*Selector el subrayado de la liga/
53   text-decoration: none;
54   /*Selector de texto/
55   color: #000000;
56
57   background: #00C2CB;
58 }

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Reemplaza los tamaños de las fuentes, y los colores para lograr un diseño personalizado. Existen varias formas de personalizar los colores, revisa la documentación de *W3Schools* (W3Schools, 2022).

9. A continuación, se definirán las reglas de diseño para los elementos de estructura, las siguientes reglas están basadas en el Módulo de Caja Flexible, del inglés, *FlexBox* (Coyier, 2022). Copia el siguiente código (ver figura 25), abajo del código previo en el archivo *style.css*.

Desarrollo práctica 3:

Figura 25. Código que define la estructura del sitio.

```

60 //-----
61 Reglas que definen la estructura de filas y columnas
62 se hace a través del atributo de diseño flex box -----
63
64 //Contenedor principal/
65 .container
66 {
67     //diseño máximo de la fila,
68     //reglas > depende al valor de 1540px, dependerá de la resolución de tu pantalla/
69     max-width: 1540px;
70 }
71 //Contenedor de Filas/
72 .row
73 {
74     //Reglas de diseño flexibles/
75     display: flex;
76     flex-wrap: wrap;
77     flex-wrap: wrap;
78     //padding-left: 10px; //padding-right: 10px; //padding: 10px; //padding: 10px; //padding: 10px;
79     margin-right: auto;
80     margin-left: auto;
81 }
82
83 //Diseño de una fila con una columna al 100% del ancho/
84 .one-column
85 {
86     //Columna definida al 100%
87     flex: 0 0 100%;
88     max-width: 100%;
89 }
90
91 //Diseño 1/4 columnas/
92 .quarter-column
93 {
94     //Columna definida al 25%
95     flex: 0 0 25%;
96     max-width: 25%;
97 }
98 //Diseño 1/2 columnas/
99 .half-column
100 {
101     //Columna definida al 50%
102     flex: 0 0 50%;
103     max-width: 50%;
104 }
105 //Diseño 1/3 columnas/
106 .onethird-column
107 {
108     //Columna definida al 33.33%
109     flex: 0 0 33.33%;
110     max-width: 33.33%;
111 }
112
113 //Diseño de una fila con ancho de la página/
114 .container-fluid
115 {
116     //Ancho al 100%
117     width: 100%;
118     //padding-left: 10px; //padding-right: 10px; //padding: 10px; //padding: 10px; //padding: 10px;
119     padding: 0px 15px;
120     //padding-left: 10px; //padding-right: 10px; //padding: 10px; //padding: 10px; //padding: 10px;
121     margin-right: auto;
122     margin-left: auto;
123 }
124

```

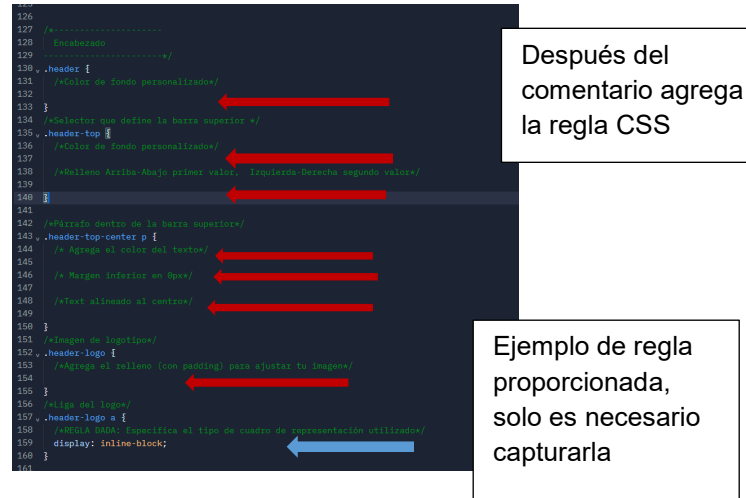
Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Nota: Las reglas anteriores no requieren personalización, sólo definen los comportamientos de las filas y sus columnas de forma flexible, es decir, para que se comparta el diseño como una tabla. Valúa el estilo ejecutando el proyecto y probando la página index.html en el navegador.

10. Copia el siguiente código en tu archivo `style.css` (ver figuras 26, 27 y 28) que define el diseño del encabezado de la página, a partir de ahora, sólo se te proporciona los comentarios de las reglas css que se espera, debes proponer la regla de estilo de acuerdo a los solicitado, usa las reglas que ya tienes en tu código y las definidas en la Tabla 1. Habrá algunas reglas más complejas que te serán proporcionadas y el comentario te indicará que hace la regla; tendrá la leyenda REGLA DADA y sólo tendrás que copiarla.

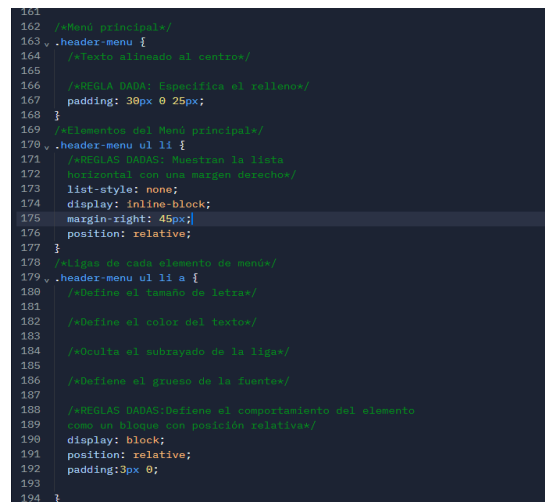
Desarrollo práctica 3:

Figura 26. Selectores que definen el diseño del encabezado



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Figura 27. Selectores que definen el diseño del menú



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Desarrollo práctica 3:

Figura 28. Continuación de código que define el menú

```

195 /* Añade un icono a cada elemento del menú para mejorar el aspecto */
196 .header-menu ul li a:after {
197     /* Define el valor del texto */
198 }
199
200 /* Menu de opciones */
201 .header-nav-option {
202     /* Define la alineación del texto a la derecha */
203     /* Define la alineación del texto a la derecha */
204     /* Define la alineación del texto a la derecha */
205 }
206
207
208 .header-nav-option a {
209     /* Define el comportamiento del elemento
210     como un bloque con posición relativa */
211     display: inline-block;
212     margin-right: 40px;
213     position: relative;
214 }
215
216 .header-nav-option a span {
217     /* Define el valor del texto */
218     /* Define el tamaño del texto */
219 }
220
221 /* Define el tamaño del texto */
222 /* Define la posición, puedes ajustar los valores
223 si no está contenido el valor dentro del texto */
224 position: absolute;
225 left: 5px;
226 top: 5px;
227 }

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Valuar el diseño, probando tu proyecto y revisando los cambios de la página index.html en el navegador, en caso de errores, revisar tu código.

11. Copia las siguientes reglas y propón las faltantes (ver figura 29 y 30), se diseña el banner principal:

Figura 29. Selector que define la imagen del banner principal

```

229
230 /* Selector de banner */
231 .banner {
232     /* Selector de banner */
233     /* Selector de banner */
234 }
235
236 .bgimage {
237     /* Define el tamaño, imagen, color de gradiente y comportamiento */
238     width: 100%;
239     height: 500px;
240     /* Define el color de gradiente con un tono RGB y la ruta de la imagen */
241     background-image: linear-gradient(to top right, rgba(0, 194, 203, 0.5), rgba(190, 175, 0, 0.5)), url(../img/home2.jpg);
242     background-position: center;
243     background-size: cover;
244     background-attachment: fixed;
245 }

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Prueba la regla *background-image*, contiene la propiedad *linear-gradient* que permite sobreponer en la imagen una transición entre dos colores con valores RGB, el cuarto valor de RGB (X, X, X, 0.5) determina la opacidad y toma valores entre 0 y 1, siendo 0 el más transparente y 1 el más intenso, cambia los valores para lograr un diseño en el que sobresalga tu texto sin perder la imagen de fondo. Reemplaza el nombre de tu imagen, asegúrate que sea de buen tamaño y esté contenida en tu carpeta img.

Desarrollo práctica 3:

Figura 30. Selectores que definen los elementos de texto sobre el banner

```

244
245
246 .bgimage .background
247 {
248
249   /*REGLA DATA: Permite controlar los elementos respecto del
250   banner, pueden hacer ajustes en valores de ser necesarios/
251   margin:auto;
252   padding: 10px;
253   -ms-transform: translate(50%);
254   transform: translate(50%);
255   text-align:center;
256 }
257
258 /*texto principal sobre imagen*/
259 .bgimage .background h3 {
260   /*tamaño de fuentes*/
261
262   /*relleno superior 50px*/
263
264   /*color del texto*/
265
266   /*REGLA DATA: Aplica una sombra al texto en tamaño y color*/
267   text-shadow:2px 2px #000;
268 }
269
270 /*texto secundario del banner*/
271 .bgimage .background p {
272   /*color del texto*/
273
274   /*tamaño del texto*/
275 }

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Valuar el diseño, probando tu proyecto y revisando los cambios de la página index.html en el navegador, en caso de errores, revisar tu código.

12. Copia las siguientes reglas y propón las faltantes (ver figura 31), se diseña la fila de iconos:

Figura 31. Selectores que define el diseño de la fila de iconos

```

276
277 /*-----*/
278 /*Fila de Iconos*/
279
280 /*Fondo de la fila*/
281 .icon-bg
282 {
283   /*Color de fondo*/
284
285   /*relleno de la fila*/
286 }
287
288 /*tamaño de iconos*/
289 .icon
290 {
291   /*Define el tamaño en 37px, es la misma regla para el tamaño de textos*/
292
293   /*Define el color del icono, es la misma regla para el color de textos*/
294
295   /*Define la altura entre líneas de texto en 40px*/
296   ;
297   /*Define un relleno izquierdo de 60px, se agrega al final*/
298
299   /*REGLA DATA: Reorganiza la posición del icono*/
300   display: inline-block;
301   position: relative;
302   float: left;
303 }
304
305 /*Sección de texto junto al icono*/
306 .icon-content {
307   /*REGLA DATA: Acomoda los texto del lado izquierdo*/
308   float: left;
309   /*Define un margen izquierdo de 10px*/
310
311   /*Define el color del texto*/
312 }

```

Desarrollo práctica 3:

```

313
314
315 /*Define prioridad al lado del icono*/
316 .icon-caption {
317   /*Define el tamaño del texto en 18px*/
318
319   /*Define la altura entre líneas de texto en 18px*/
320
321   /*Define el grueso de la fuente en 700*/
322
323   /*Define el color del texto*/
324
325   /*Media 100: Define la posición del texto*/
326   display: block;
327   position: relative;
328 }
329 /*Inicio secundario*/
330 .icon-subcaption {
331   /*Define el tamaño del texto en 18px*/
332
333   /*Media 100: Define la posición del texto*/
334   display: block;
335   position: relative;
336 }
337

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Valuar el diseño, probando tu proyecto y revisando los cambios de la página index.html en el navegador, en caso de errores, revisar tu código.

13. Copia las siguientes reglas y propón las faltantes (ver figura 32), se diseña la fila de categorías:

Figura 32. Selector que define la fila de categorías

```

337
338 /*Inicio secundario*/
339 .categories
340 /*Define el tamaño del texto en 18px*/
341 .categories
342 {
343   /*Define el padding en 3em, es una medida relativa*/
344   padding: 3em;
345   /*Define el ancho en 100%, es una medida relativa*/
346   padding-left: 100%;
347 }
348

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Nota: La regla anterior introduce las unidades de medida relativa (em), puedes usar la documentación para conocer más sobre las unidades de medida CSS. (W3 Schools, 2022)

14. Copia las siguientes reglas y propón las faltantes (ver figura 33), se diseña el banner promocional:

Figura 33. Selectores que definen el banner promocional

```

349
350 /*Inicio Banner promocional*/
351 .banner-promocional
352 /*Define el tamaño del texto en 18px*/
353 .offer {
354   /*Define el padding en 40px 15% 0px*/
355   padding: 40px 15% 0px;
356 }
357 /*Inicio de primera columna*/
358 .offer h2 {
359   /*Define el color del texto*/
360
361   /*Define la altura entre líneas de texto en 18px*/
362
363   /*Define el tamaño de la fuente en 18px*/
364 }
365 /*Inicio de segunda fila en la primera columna*/
366 .offer h2 span {
367   /*Define el grueso de la letra en 700*/
368
369   /*Define el color del texto*/
370 }
371
372 }
373 /*Inicio sección que contiene la imagen del producto promocional*/
374 .offer-center {
375   /*Define la altura entre los elementos*/
376   position: relative;
377   z-index: 5;
378 }
379

```

Desarrollo práctica 3:

```

388
389 
390 .offer-center img {
391     width: 100%; /*Define el tamaño, personaliza el valor
392     para ajustar la imagen en caso de ser necesario*/
393     min-width: 100%;
394 }
395
396 
398 .offer-price {
399     /*MOLDA DADA: Define el tamaño en alto y ancho del
400     elemento y la forma de círculo, ajusta los valores en caso de ser necesario*/
401     height: 100px;
402     width: 120px;
403     border-radius: 50%;
404     /*Personaliza el color de fondo del círculo*/
405 }
406
407 /*Agrega un relleno superior de 22px*/
408
409 /*Define el texto al centro*/
410
411 /*MOLDA DADA: Define la posición del círculo sobre la imagen,
412 para los valores en ser necesario*/
413 position: absolute;
414 right: 50px;
415 top: -36px;
416 }
417
418 /*Define el text 'Día' en el círculo del precio*/
419 .offer-price span {
420     /*MOLDA DADA: Define como muestra el elemento*/
421     display: block;
422     /*Define el tamaño de letra a 14px*/
423     /*Define el color del texto*/
424     /*Define un margen inferior a 4px*/
425 }
426
427 /*Define el valor del precio*/
428 .offer-price h5 {
429     /*Define el valor del texto*/
430     /*Define el tamaño del texto a 20px*/
431     /*Define el grupo del texto a 700*/
432 }
433
434 /*Define el elemento que contiene el reloj*/
435 .offer-counter span {
436     /*Define el color del texto 'Día de la semana'*/
437     /*Define el tamaño de letra a 14px*/
438     /*Define el grupo del texto a 700*/
439     /*Define el texto en mayúscula, la regla necesaria es
440     con text-transform*/
441     /*Define el espaciado horizontal del texto en 2px,
442     es con la regla letter-spacing*/
443     /*Define un margen inferior de 15px*/
444     /*MOLDA DADA: Define como muestra el elemento*/
445     display: block;
446 }
447
448 /*Define el nombre del producto en promoción*/
449 .offer-counter h2 {
450     /*Define el color del texto que nombra al producto*/
451     color: #111111;
452     /*Define el grupo del texto a 700*/
453     font-weight: 700;
454     /*Define el espacio entre líneas de texto en 40px*/
455     line-height: 40px;
456     /*Define un margen inferior de 25px*/
457     margin-bottom: 25px;
458 }
459
460 /*Define el bloque de los elementos del reloj*/
461 .offer-counter .offer-counter-clock {
462     /*Define un margen inferior de 20px*/
463     /*MOLDA DADA: Define como muestra el elemento*/
464     overflow: hidden;
465     margin-left: -30px;
466 }
467
468 /*Define los elementos del reloj: Día, Hora, Minutos, Segundos*/
469 .offer-counter .offer-counter-clock .cd-item {
470     /*MOLDA DADA: Define que cada elemento se muestra de forma horizontal,
471     con un fila donde cada elemento ocupa un 25% de columna*/
472     width: 25%;
473     float: left;
474     position: relative;
475     /*Define un margen inferior de 20px*/
476     /*Define el texto centrado*/
477 }

```

Desarrollo práctica 3:

```

478 /*define al elemento de : en el reloj*/
479 .offer-counter .offer-counter-clock .cd-item:after {
480   position: absolute;
481   position: absolute;
482   right: 0;
483   top: 7px;
484   content: ":";
485   /*define el tamaño de letra en 20px*/
486   /*define el grosor de letra en 300%*/
487   /*define el color del texto */
488 }
489
490 }
491
492 /*define se muestra el último elemento ":" dado que la regla
493 agrega ":" al elemento después del segundo*/
494 .offer-counter .offer-counter-clock .cd-item:last-child:after {
495   display: none;
496 }
497
498 /*define el valor numerico del reloj contador*/
499 .offer-counter .offer-counter-clock .cd-item span {
500   /*define el color del texto*/
501   /*define el grosor de la fuente en 300%*/
502   /*define el tamaño de la fuente en 20px*/
503   /*define el tamaño de la fuente en 20px*/
504   /*define el tamaño de la fuente en 20px*/
505   /*define el tamaño de la fuente en 20px*/
506   /*define el tamaño de la fuente en 20px*/
507   display: block;
508 }
509
510 /*define el texto de Dias, Horas, Minutos y Segundos
511 del reloj contador*/
512 .offer-counter .offer-counter-clock .cd-item p {
513   /*define el tamaño de letra en 20px*/
514 }
515 }
516 }
517
518

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Valuar el diseño, probando tu proyecto y revisando los cambios de la página index.html en el navegador, en caso de errores, revisar tu código.

15. Copia las siguientes reglas y propón las faltantes (ver figura 34), se diseña el icono de WhatsApp, para que se muestre de forma flotante en la parte inferior derecha.

Figura 34. Selectores que diseñan un botón flotante.

```

519 /*define el icono de whatsapp*/
520 .icono-whatsapp {
521   width: 40px;
522   height: 40px;
523   position: fixed;
524   bottom: 20px;
525   right: 20px;
526   z-index: 100;
527   background-color: #25d366;
528   color: white;
529   border-radius: 50%;
530   text-align: center;
531   font-size: 30px;
532   box-shadow: 2px 2px 3px #999;
533 }
534
535 /*define se muestra abajo del icono*/
536 .my-whatsapp {
537   margin-top: 10px;
538 }
539
540

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Valuar el diseño, probando tu proyecto y revisando los cambios de la página index.html en el navegador, en caso de errores, revisar tu código.

16. Copia las siguientes reglas y propón las faltantes (ver la figura 35), se diseña el pie de página:

Figura 35. Selectores que diseñan el pie de página

Desarrollo práctica 3:

```

547 /<----->
548 /<----->
549 /<----->
550 /<----->
551 footer {
552 /<----->
553 /<----->
554 /<----->
555 /<----->
556 /<----->
557 /<----->
558 /<----->
559 /<----->
560 /<----->
561 /<----->
562 }
563
564 /<----->
565 footer span a
566 {
567 /<----->
568 /<----->
569 /<----->
570 /<----->
571 /<----->
572 /<----->
573 /<----->
574 /<----->
575 /<----->
576 /<----->
577 /<----->
578 }

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Valuar el diseño, probando tu proyecto y revisando los cambios de la página index.html en el navegador, en caso de errores, revisar tu código.

Resultados obtenidos:

Al término de la práctica habrás desarrollado una página web completa, el resultado esperado debe ser similar en estructura a la figura 3, los colores, textos e imágenes deben ser propuestos.

Evidencia de la práctica: (fotografías, videos, archivo, etc.)

Integrar la estructura que conforman el proyecto de la página web (archivo index.html, carpeta css con archivo style.css con la propuesta de tus reglas CSS, carpeta img con imágenes)

Conclusiones:

Describe los retos que te presentaste al trabajar con la tecnología CSS y cómo puedes aplicar los conocimientos adquiridos a proyectos reales.

Bibliografía:

De la web

- Coyier, C. (27 de Febrero de 2022). CSS Tricks. Obtenido de A Complete Guide to Flexbox: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- Google. (27 de Febrero de 2022). Google Font. Obtenido de Google Font: <https://fonts.google.com/>
- Google. (27 de Febrero de 2022). Google Font. Obtenido de Get Started with the Google Fonts API: https://developers.google.com/fonts/docs/getting_started
- Mozilla. (27 de Febrero de 2022). MDN Web Docs. Obtenido de Referencia CSS: <https://developer.mozilla.org/es/docs/Web/CSS/Reference>
- Replit. (20 de Febrero de 2022). Replit. Obtenido de Crea una cuenta en Replit: <https://replit.com/signup>
- W3 Schools. (02 de Marzo de 2022). W3 Schools. Obtenido de CSS Units: https://www.w3schools.com/cssref/css_units.asp
- W3Schools. (27 de Febrero de 2022). W3Schools. Obtenido de CSS Legal Color Values: https://www.w3schools.com/cssref/css_colors_legal.asp
- WhatsApp. (27 de Febrero de 2022). WhatsApp. Obtenido de Cómo usar el clic para chatear: <https://faq.whatsapp.com/general/chats/how-to-use-click-to-chat>

Criterios de evaluación:

- Valuar la página index.html, probándola en un navegador web y comprobar que la estructura cumpla con lo solicitado en la práctica.
- Código libre de errores y documentado.

Datos de la Práctica 4

Práctica 4 de 5:	Desarrollo de contenido dinámico en página web de comercio electrónico de venta de ropa de moda.
Objetivo de la práctica:	Desarrollar contenido dinámico en una página web con el lenguaje de programación <i>Javascript</i> para poder aplicar los conocimientos de arquitectura de la información enfocados a necesidades de comercio electrónico.
Temas y subtemas asociados:	3. Aplicaciones de Negocio 3.2 Sistemas de comercio electrónico
Fecha:	
Duración (horas):	2 horas
Laboratorio:	Laboratorio de cómputo
Equipo de seguridad para ingresar al laboratorio (indispensable):	N/A
Software requerido:	<i>Netbeans IDE</i> con <i>plugin HTML 5 Kit</i> o acceso a la <i>IDE</i> basada en la nube <i>Replit</i> (Replit, 2022)
Equipo necesario en laboratorio:	Computadora con acceso a Internet
Material/Sustancias/Reactivos disponible en laboratorio:	N/A
Material/Sustancias/Reactivos aportado por el alumno:	N/A

Desarrollo práctica 4:

Consideraciones previas al desarrollo de la práctica:

Tener completa y sin errores la **Práctica 3. Aplicar diseño a una página web de comercio electrónico de venta de ropa de moda.**

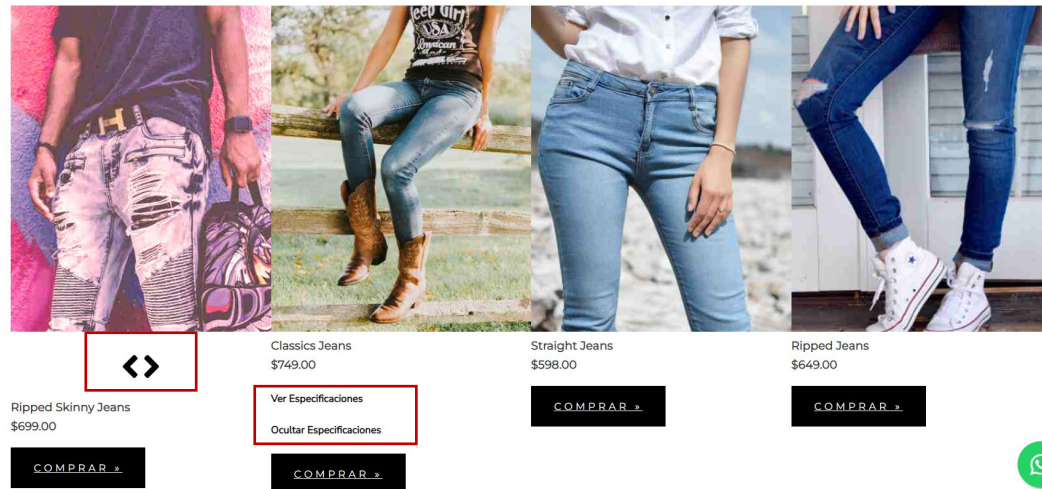
Utilizando Javascript, se modificará el código disponible para agregar interacción con el usuario. En la sección de productos (ver figura 36) se agregarán dos botones al primer producto, cuando el usuario da clic en el botón siguiente (representado por el icono >), las imágenes puedan cambiar con cada clic; cuando el usuario de clic en el botón atrás (representado por el icono <), las imágenes cambiarán en sentido contrario con cada clic.

En el segundo producto se agregará un texto con la leyenda “Ver Especificaciones” y otro con la leyenda “Ocultar Especificaciones”, cuando el usuario da clic al texto “Ver Especificaciones”, dinámicamente se muestra información adicional del producto; cuando da clic en el texto “Ocultar Especificaciones”, la información adicional se elimina.

Para lograr el contenido dinámico descrito usaremos Javascript manipulando el DOM HTML, consulta la documentación W3 Schools (W3 Schools, 2022).

Desarrollo práctica 4:

Figura 36. Se muestra en rojo los elementos que se afectarán para lograr el contenido dinámico.



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Indicaciones de la práctica:

1. Abre la *IDE* disponible en el laboratorio.
2. Abre el proyecto desarrollado en la práctica 3.
3. Abre el archivo *index.html* e identifica el código que estructura la sección de productos (ver figura 36)
4. Identifica el código que estructura el primer producto y después de la imagen (elemento ``), copia el siguiente código (ver figura 37):

Figura 37. El código agrega una lista de iconos.

```

140 <section class="categorias">...<!-- Inicia sección de productos -->
141 <h2>Los más vendidos</h2><br>
142 <div class="container">...<!-- Estructura el contenido de tabla -->
143 <div class="row">...<!-- Estructura el contenido de fila -->
144 <article class="quarter-column">...<!-- Estructura la columna 1 -->
145 <div class="container">
146 
147 <div class="list-decor">...<!-- Lista que genera estructuras internas de información -->
148 <ul class="list-item">
149 <li class="list-item">
150 <div class="list-item">
151 <div class="list-item">
152 <div class="list-item">
153 <div class="list-item">
154 <div class="list-item">
155 <div class="list-item">
156 <div class="list-item">
157 <div class="list-item">
158 <div class="list-item">
159 </div>...<!-- Termina código necesario para generar información -->
160 <p>Ripped Skinny Jeans<br><b>$699.00</b></p>
161 <p><a href="shop.html" class="primary-btn">Comprar </a></p>
162 </div>
163 </article>...<!-- Termina estructura la columna 1 -->
164 <article class="quarter-column">...<!-- Estructura la columna 2 -->
165 <div class="container">
166 

```

Desarrollo práctica 4:

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

5. Al elemento de imagen del primer producto, agrega el atributo `id="producto"`, este atributo permitirá manipular la imagen por medio de *Javascript* más adelante.
6. Abre el archivo `style.css` y al final de tu código, copia las siguientes reglas de estilo (ver figura 38) que aportan diseño a los iconos agregados previamente.

Figura 38. Reglas de estilo para lista de iconos

```

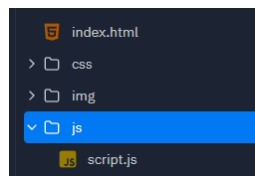
577
578 /* Aplica estilo a lista que contiene los iconos
579    agregados al primer producto */
580 .list-decor
581 {
582     padding-left: 0;
583     list-style: none;
584     text-align: center;
585 }
586 /*Permite mostrar los iconos de la lista de forma horizontal */
587 .list-item
588 {
589     display: inline-block;
590 }
591
592
593

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

7. A la estructura de tu proyecto, agrega una carpeta llamada `js` y dentro crea un archivo llamado `script.js` (ver figura 39), en este archivo se agregará el código de *Javascript*.

Figura 39. Estructura de carpetas y archivos del proyecto



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

8. En el archivo `index.html`, ubica el `<head></head>` y antes del cierre del elemento `</head>`, agrega la siguiente referencia (ver figura 40), la cual permite cargar el archivo `script.js` que contiene el código de *Javascript* y puede ser ejecutado.

Desarrollo práctica 4:

Figura 40. Referencia a archivo Javascript

```
16
17 <!-- Referencia de fuentes de Google -->
18 <!--<link rel="stylesheet"
19 href="https://fonts.googleapis.com/css2?family=CrimsonPro">-->
20 <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Nunito+Sans&family=Montserrat">
21
22 <!-- Referencia de Javascript -->
23 <script type="text/javascript" src="js/script.js"></script>
24
25 </head><!-- Termina metadatos -->
26 <body>
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

9. Dentro de la carpeta donde se almacenan las imágenes (carpeta *img*), agrega dos imágenes más, se utilizarán para que cambien en sustitución de la imagen del producto 1 cuando el usuario da clic en los iconos adelante y atrás (representados por los iconos < >).
10. Abre el archivo script.js y copia el siguiente código (ver imagen 41), se define el evento *onload()* que permite ejecutar código de Javascript después de que haya cargado la página index.html. Todo el código que se agregue en adelante debe ir entre las llaves.

Figura 41. Código inicial de Javascript

```
js/script.js <
1 //Permite que cargue el HTML primero, antes de hacer referencia
2 // a cualquier elemento evitando valores nulos
3 window.onload=function(){}
4
5
6 }
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

11. Copia el siguiente código (ver figura 42) dentro del evento *onload()*, permite obtener en una variable el icono adelante (representado en el HTML por el icono >). Después se registra el evento clic para que cuando el usuario interactué con el icono se ejecute la función llamada *verSiguiente()* y permita cambiar la imagen de forma dinámica.

Figura 42. Código para manipular el icono adelante

```
js/script.js <
1 //Permite que cargue el HTML primero, antes de hacer referencia
2 // a cualquier elemento evitando valores nulos
3 window.onload=function(){}
4
5 //Permite que cargue el HTML primero, antes de hacer referencia
6 var siguiente = document.getElementById("adelante");
7 siguiente.addEventListener("click", verSiguiente);
8
9
10 }
11
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

12. Antes del cierre de la llave copia la función *verSiguiente()* y sus instrucciones (ver figura 43).

Desarrollo práctica 4:

Figura 43. Función VerSiguiente()

```
js/script.js
1 //Permite que cargue el HTML primero, antes de hacer referencia
2 // a cualquier elemento evitando valores nulos
3 window.onload=function(){
4
5 //Se obtiene el icono siguiente >
6 var siguiente = document.getElementById("adelante");
7 //Se registra el evento
8 siguiente.addEventListener("click", verSiguiente);
9
10
11 //Función que manipula el cambio de imágenes
12 // cuando el usuario da clic en el icono siguiente
13 function verSiguiente() {
14     var aux = document.getElementById("producto")
15     var ruta= aux.src;
16     if (ruta.includes('img/jeans2.jpg'))
17         aux.src="img/jeans-green.jpg";
18     else if(ruta.includes("jeans-green.jpg"))
19         aux.src="img/jeans-red.jpg";
20     else if(ruta.includes('img/jeans-red.jpg'))
21         aux.src="img/jeans2.jpg";
22     else
23         aux.src="img/jeans2.jpg";
24 }
25
26 }
27
28
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Del código anterior, sustituye el nombre de las imágenes (en el código tienen terminación de archivo .jpg) para que correspondan con el nombre de tus imágenes, considera que deben estar almacenadas en la carpeta img y tengan relación con el producto uno, para que cuando el usuario de clic al icono adelante (representado por el icono >) cambie la imagen.

13. Antes del cierre de la llave, copia el siguiente código (ver figura 44) que manipula el icono atrás (representado por el icono <) para que ahora cambie las imágenes en sentido contrario. Nuevamente, sustituye el nombre de tus imágenes para que correspondan a la del producto 1.

Desarrollo práctica 4:

Figura 44. Código para manipular el control atrás

```
js/script.js x
25
26 //Se obtiene el icono atras
27 var vista_atras = document.getElementById("atras");
28 //Se registra el evento
29 vista_atras.addEventListener("click", verAtras);
30
31 //Función que maneja el cambio de imagenes
32 //Cuando el usuario da clic en el icono atras
33 function verAtras() {
34
35     var aux = document.getElementById("producto")
36     var ruta= aux.src;
37     if (ruta.includes('img/jeans2.jpg'))
38         aux.src="img/jeans-red.jpg";
39     else if(ruta.includes('img/jeans-red.jpg'))
40         aux.src="img/jeans-green.jpg";
41     else if(ruta.includes('img/jeans-green.jpg'))
42         aux.src="img/jeans2.jpg";
43     else
44         aux.src="img/jeans2.jpg";
45 }
46
47 }
48
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

14. Valuar el código, probando el proyecto y revisando los cambios de la página index.html en el navegador, da clic en el icono > y debe cambiar la imagen del producto uno en cada clic. Después prueba da clic en el icono <, ahora debe cambiar la imagen del producto uno en sentido contrario en cada clic. En caso de errores, revisa tu código.
15. En la página index.html, identifica el código que estructura el segundo producto y copia el siguiente código después del elemento <p></p> (ver figura 45).

Figura 45. Código para agregar interacción al producto dos

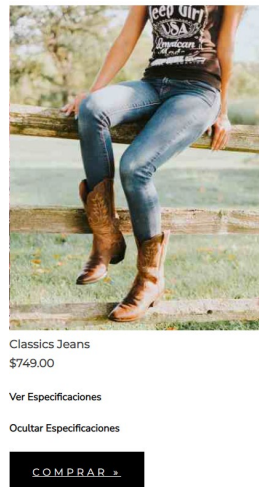
```
index.html x
156 <i id="adelante" class="fas fa-angle-right fa-3x"></i>
157 </li>
158 </ul> <!-- Termina código necesario para agregar interacción -->
159 <p>Ripped Skinny Jeans<br><b>$699.00</b></p>
160 <p><a href="shop.html" class="primary-btn">Comprar </a></p>
161 </div>
162 </article> <!-- Termina estructura la columna 1 -->
163 <article class="quarter-column"> <!-- Estructura la columna 2 -->
164 <div class="container">
165 
166 <p>Classics Jeans<br><b>$749.00</b></p>
167 <!-- Código necesario para agregar interacción a producto 2 -->
168 <h5 id="ver">Ver Especificaciones</h5>
169 <div>
170 <p id="datos"></p>
171 </div>
172 <h5 id="ocultar">Ocultar Especificaciones</h5>
173 <!-- Código necesario para agregar interacción -->
174 <p><a href="shop.html" class="primary-btn">Comprar </a></p>
175 </div>
176 </article> <!-- Termina estructura la columna 2 -->
177 <article class="quarter-column"> <!-- Estructura la columna 3 -->
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Desarrollo práctica 4:

Al validar el código anterior en el navegador se debe ver similar a la figura 46.

Figura 46. Vista del producto dos en el navegador



Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información

16. En el archivo script.js antes del cierre de la última llave copia el siguiente código (ver figura 47).

Figura 47. Código para crear interacción al producto dos.

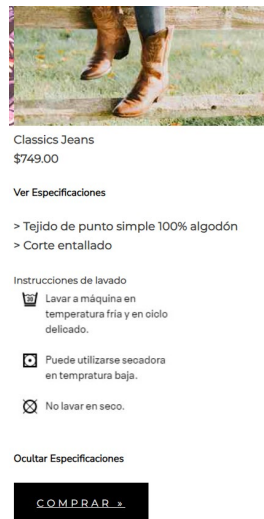
```
js/script.js x
46
47 //Se obtiene el elemento h5 Ver Especificaciones
48 var btn= document.getElementById("ver");
49 btn.addEventListener("click", mostrarEspecificaciones)
50 //Función que crea y añade elementos html al DOM
51 function mostrarEspecificaciones()
52 {
53     var text= document.getElementById("datos");
54     text.innerHTML=" Tejido de punto simple 100% algodón <br> > Corte entallado";
55     var instrucciones= document.createElement("h5");
56     instrucciones.innerHTML="Instrucciones de lavado";
57     text.appendChild(instrucciones);
58     var img_instrucciones= document.createElement("img");
59     img_instrucciones.src="img/cuidados.jpg";
60     instrucciones.appendChild(img_instrucciones);
61 }
62
63 //Se obtiene el elemento h5 Ocultar Especificaciones
64 var btn2= document.getElementById("ocultar");
65 btn2.addEventListener("click", ocultarEspecificaciones)
66
67 //Función elimina las especificaciones
68 function ocultarEspecificaciones()
69 {
70     var elemento= document.getElementById("datos");
71     while (elemento.firstChild) {
72         elemento.removeChild(elemento.firstChild);
73     }
74 }
75
76
77 }
```

Nota. Esta figura fue desarrollada para el practinario de la asignatura de Arquitectura de la Información

Desarrollo práctica 4:

El código anterior se ejecuta cuando el usuario da clic en el texto “Ver Especificaciones” (a través de ejecutarse la función *mostrarEspecificaciones()* cuando sucede el evento clic). Se muestra información adicional del producto de forma dinámica (ver figura 48), la cual se estructura con textos y una imagen. Personaliza los textos que se agregan a la propiedad innerHTML y sustituye la imagen cuidados.jpg por una imagen que muestre información adicional del producto, considera que la imagen debe estar en tu carpeta de imágenes img.

Figura 48. Estado del producto dos cuando el usuario da clic en Ver Especificaciones



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

La función *ocultarEspecificaciones()* se ejecuta cuando sucede el evento clic sobre el texto Ocultar Especificaciones, el cual elimina dinámicamente el contenido adicional del producto.

17. Valuar el código, probando tu proyecto y revisando los cambios de la página index.html en el navegador; sobre el segundo producto da clic en el texto Ver Especificaciones, se debe mostrar información adicional del producto, al dar clic en el texto Ocultar Especificaciones se deben eliminar la información adicional. En caso de errores o de no lograr el resultado esperado, revisar tu código.

Resultados obtenidos:

Al término de la práctica se habrá realizado un sitio web con contenido dinámico a través del lenguaje de programación de Javascript manipulando el DOM HTML.

Evidencia de la práctica: (fotografías, videos, archivo, etc.)

Integrar la estructura que conforma el proyecto de la página web (archivo index.html, carpeta css con archivo style.css, carpeta img con imágenes, carpeta js con archivo script.js, el cual contiene el código desarrollado)

Conclusiones:

Describe en qué funcionalidades puedes aplicar los conocimientos de Javascript manipulando el DOM HTML en otros sitios web de comercio electrónico.

Bibliografía:

De la web:

- W3 Schools. (7 de Marzo de 2022). W3 Schools. Obtenido de JavaScript HTML DOM: https://www.w3schools.com/js/js_htmlDOM.asp
- Replit. (20 de Febrero de 2022). Replit. Obtenido de Crea una cuenta en Replit: <https://replit.com/signup>

Criterios de evaluación:

- Valuar la página index.html, revisándola en un navegador web y comprobar que cumpla con la función dinámica de interacción con el usuario.
- Código libre de errores y documentado.

Datos de la Práctica 5

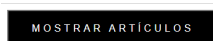
Práctica 5 de 5:	Desarrollo de contenido dinámico para una página web de tipo <i>blog</i> consumiendo datos semi estructurados.
Objetivo de la práctica:	Desarrollar una página web de tipo <i>Blog</i> usando el lenguaje de <i>Javascript</i> a través de <i>AJAX</i> para generar dinámicamente el contenido, para aplicaciones con datos semiestructurados en formato JSON.
Temas y subtemas asociados:	5. Tipos de información 5.2 Datos semi estructurados
Fecha:	
Duración (horas):	2 horas
Laboratorio:	Laboratorio de cómputo
Equipo de seguridad para ingresar al laboratorio (indispensable):	N/A
Software requerido:	<i>Netbeans IDE</i> con <i>plugin HTML 5 Kit</i> o acceso a la <i>IDE</i> basada en la nube <i>Replit</i> (Replit, 2022)
Equipo necesario en laboratorio:	Computadora con acceso a Internet
Material/Sustancias/Reactivos disponible en laboratorio:	N/A
Material/Sustancias/Reactivos aportado por el alumno:	N/A

Desarrollo práctica 5:

Consideraciones previas a la práctica:

La página inicialmente mostrará un botón con la leyenda “Mostrar Artículos” (ver figura 49). Cuando el usuario da clic sobre el botón, se agregan dinámicamente cuatro artículos de un blog con información de imagen alusiva, título, autor, descripción corta del artículo y botón (ver imagen 50). La información de los artículos se consume de una fuente de datos semiestructurada con el formato JSON, el desarrollo se realiza con Javascript a través de AJAX, para más información sobre AJAX puedes revisar previamente la documentación de MDN Web Docs (Mozilla , 2022).

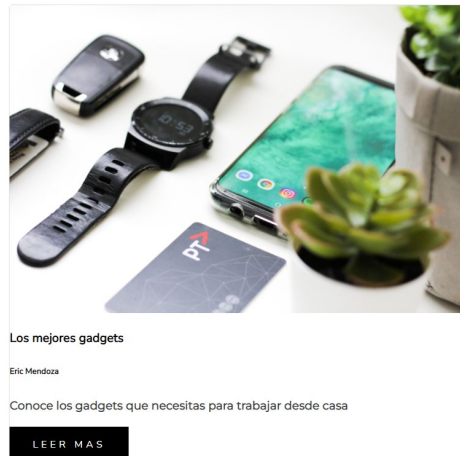
Figura 49. Botón inicial



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Desarrollo práctica 5:

Figura 50. Estructura de un artículo del blog



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Indicaciones de la práctica:

1. Abre la *IDE* disponible en el laboratorio.
2. Crea un nuevo proyecto en la *IDE* a utilizar del tipo *HTML, CSS, Javascript*.
3. Nombra tu proyecto: *blog*
4. Una vez creado el proyecto, crea o abre un archivo llamado *index.html*
5. Copia la siguiente estructura de código (ver figura 51) en el archivo *index.html*.

Figura 51. Estructura de código HTML

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width">
6 <title>Blog </title>
7 <!-- Referencia a la hoja de estilo -->
8 <link href="css/style.css" rel="stylesheet" type="text/css" />
9 <!-- Referencia al archivo con código de Javascript -->
10 <script src="js/script.js"></script>
11 <!-- Referencia a la fuente de la tipografía -->
12 <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Nunito&family=Montserrat">
13
14 </head>
15 <body>
16 <!-- Estructura de un botón -->
17 <p><input class="primary-btn" type="button" value="Mostrar Artículos" id="load"></p>
18 <!-- Esta estructura de código carga dinámicamente una estructura de blog -->
19 <div id="article"></div>
20
21 </body>
22 </html>

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

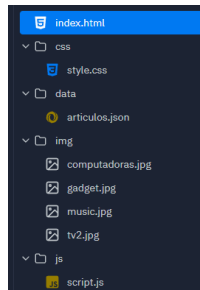
El código anterior crea la estructura de un botón que desencadena el resto de la acción dinámica.

6. Añade a tu proyecto la carpeta *css* y dentro crea un archivo llamado *style.css*.
7. Añade a tu proyecto la carpeta *js* y dentro crea un archivo llamado *script.js*.

Desarrollo práctica 5:

8. Añade a tu proyecto la carpeta *img*, agrega cuatro imágenes, cada imagen estructurará la portada de un artículo del blog.
9. Añade a tu proyecto la carpeta *data*, y dentro crea un archivo llamado *articulos.json*. La estructura debe quedar similar a la figura 52.

Figura 52. Estructura de carpetas del proyecto



Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

10. Abre el archivo *articulos.json* y copia el siguiente código (ver figura 53) que representa datos semi estructurados en el formato JSON (Mozilla, 2022). Los datos representan un arreglo de cuatro objetos, cada objeto representa un artículo del blog con información de título, autor, contenido de resumen e imagen de portada.

Figura 53. Formato de datos semi estructurados con JSON

```
1, {
2,   "articulos": [
3,     {
4,       "titulo": "Los mejores gadgets",
5,       "autor": "Eric Mendoza",
6,       "contenido": "Conoce los gadgets que necesitas para trabajar desde casa",
7,       "imagen": "img/gadget.jpg"
8,     },
9,     {
10,      "titulo": "Elige la bocina adecuada para ti",
11,      "autor": "Victor Iapia",
12,      "contenido": "Las mejores bocinas inalámbricas bluetooth",
13,      "imagen": "img/music.jpg"
14,    },
15,    {
16,      "titulo": "Las mejores televisiones en 2022",
17,      "autor": "Isene Hernandez",
18,      "contenido": "No te pierdas tu serie favorita",
19,      "imagen": "img/tv2.jpg"
20,    },
21,    {
22,      "titulo": "Mantén actualizada tu computadora con estos consejos",
23,      "autor": "Maria Sanchez",
24,      "contenido": "Obtén el mayor provecho de tu computadora",
25,      "imagen": "img/computadoras.jpg"
26,    }
27,  ]
28, }
29, }
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Edita y personaliza los textos en color anaranjado de acuerdo al contenido que quieres que tenga tu página de blog, los textos estructuran el título del artículo, el autor y la descripción corta del artículo. El valor de "imagen" corresponde a la ruta absoluta donde esta almacenada la imagen para cada artículo, edita los nombres de acuerdo a los nombres de tus imágenes.

Desarrollo práctica 5:

11. Abre el archivo *script.js* y copia el siguiente código (ver figura 54), los comentarios te ayudarán a comprender la tarea de cada instrucción.

Desarrollo práctica 5:

Figura 54. Código de Javascript

```

1. window.onload=function(){
2. //Obtiene en una variable al Botón Mostrar Articulos
3. var btn = document.getElementById("load")
4. //Se registra un evento al dar clic
5. btn.addEventListener("click", CargarArticulos)
6. //Se obtiene en una variable al elemento donde se agregarán
7. // los articulos
8. var div_art= document.getElementById("article")
9.
10. //Cuando se da clic en botón se ejecuta el método CargarArticulos
11. function CargarArticulos()
12. {
13.     //Se crea el Objeto XMLHttpRequest
14.     //El uso de este objeto es los que conocemos como AJAX
15.     // También conocido como javascript asincrono
16.     var xhttp = new XMLHttpRequest()
17.     //Indicamos que la conexión esta lista
18.     xhttp.onreadystatechange = function() {
19.         if (this.readyState == 4 && this.status == 200)
20.         {
21.             //Cuando esta lista la conexión se ejecuta la función leerJSON
22.             leerJSON(this)
23.         }
24.     };
25.
26.     //Cuando se establece la conexión con el servidor se obtiene el archivo articulos.json
27.     xhttp.open("GET", "data/articulos.json", true)
28.     //Se envia la solicitud de forma asincrona
29.     xhttp.send()
30. }
31.
32. //Función que se ejecuta cuando se establece conexión con el servidor
33. //En esta línea para recibir datos
34. function leerJSON(respuestaJSON)
35. {
36.     //Se obtiene los datos del servidor y se convierte a objeto Javascript
37.     var objJson = JSON.parse(respuestaJSON.responseText)
38.     //Variables de apoyo para recorrer el objeto JSON
39.     var i
40.     //Recorremos la variable
41.     div_art.innerHTML=""
42.
43.     //Se obtiene cada objeto del archivo JSON
44.     //Este ciclo se ejecuta para cada artículo
45.     for ( i in objJson.articulos) {
46.         //Se crea un elemento div para estructurar el artículo
47.         var divCard = document.createElement("div")
48.         //Se le asigna una clase para identificar artículo
49.         divCard.className="article"
50.
51.         //Se crea una imagen
52.         var imagenArticulo = document.createElement("img")
53.         //Imagen se asigna la ruta almacenada en el objeto JSON
54.         imagenArticulo.src= objJson.articulos[i].imagen
55.         //Se le asigna una clase para identificar imagen
56.         imagenArticulo.className="image-article"
57.         //Se agrega la imagen al div
58.         divCard.appendChild(imagenArticulo)
59.
60.         //Se crea elemento div que forma el contenido del artículo
61.         var divBody = document.createElement("div")
62.         //Se agrega el div al artículo
63.         divCard.appendChild(divBody)
64.
65.         //Se crea elemento h4 que forma el título del artículo
66.         var encabezadoh4 = document.createElement("h4")
67.         //Se asigna el título almacenado en el objeto JSON
68.         encabezadoh4.innerHTML=objJson.articulos[i].titulo
69.         //Se agrega al div el título del artículo
70.         divBody.appendChild(encabezadoh4)
71.
72.         //Se crea elemento h6 que agrega el autor al contenido del artículo
73.         var encabezadoh6 = document.createElement("h6")
74.         //Se asigna el autor almacenado en el objeto JSON
75.         encabezadoh6.innerHTML=objJson.articulos[i].autor
76.         //Se agrega al div el autor del artículo
77.         divBody.appendChild(encabezadoh6)
78.
79.         //Se crea elemento p que forma contenido del artículo
80.         var parrafoContenido = document.createElement("p")
81.         //Se asigna el contenido almacenado en el objeto JSON
82.         parrafoContenido.innerHTML=objJson.articulos[i].contenido
83.         //Se agrega al div el contenido del artículo
84.         divBody.appendChild(parrafoContenido)
85.
86.         //Se crea elemento a que forma botón del artículo
87.         var botonArticulo = document.createElement("a")
88.         //Se le asigna el estilo del botón
89.         botonArticulo.className="primary-btn"
90.         //Se le asigna texto al botón
91.         botonArticulo.innerHTML="Leer Más"
92.         //Se agrega al div el contenido del artículo
93.         divBody.appendChild(botonArticulo)
94.         //Se clona toda la estructura del artículo al código HTML
95.         div_art.appendChild(divCard)
96.     }
97. }

```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Desarrollo práctica 5:

12. Abre el archivo *style.css* y copia el siguiente código (ver figura 55), las reglas de estilo permiten aplicar diseño al botón y a la estructura de cada artículo.

Nota: Parte de este código se reutiliza del archivo *style.css* realizado en la práctica 3, copia las reglas definidas en la práctica 3 que se utilizan en esta práctica.

Figura 55. Código CSS

```
css/style.css
1  /* Configuraciones generales */
2  Configuraciones generales
3  Configuraciones generales
4  html,
5  body {
6  /*Especifica la altura del area del contenido HTML*/
7  height: 100%;
8  /*Define el tipo de fuente, si se utilizan fuentes de Google,
9  como se especifica el nombre de la fuente con comillas simple ' */
10 font-family: 'Nunito Sans';
11 }
12 /* Regla que aplica a todos los parrafos*/
13 p {
14 /*Tamaño de fuente*/
15 font-size: 15px;
16 /*Tipo de fuente de Google*/
17 font-family: 'Montserrat';
18 /*Color de texto por defecto*/
19 color: #3d3d3d;
20 /*Decoración de texto*/
21 font-weight: bold;
22 /*Altura entre líneas de textos*/
23 line-height: 25px;
24 /*Margen definido en los 4 lados en el orden
25 Arriba | Derecha | Abajo | Izquierda*/
26 margin: 0 0 15px 0;
27 }
28
29 /*Selector de clase que aplica un estilo de
30 botones a las ligas <a>*/
31 .primary-btn {
32 /*Especifica el tipo de cuadro de representación utilizado*/
33 display: inline-block;
34 /*Tamaño de letra*/
35 font-size: 13px;
36 /*Grosor de la fuente*/
37 font-weight: 500;
38 /*Carácter de texto en mayúsculas*/
39 text-transform: uppercase;
40 /*Bordes Arriba-Abajo primer valor, Izquierda-Derecha segundo valor*/
41 padding: 14px 30px;
42 /*Color de texto*/
43 color: #ffffff;
44 /*Color de fondo*/
45 background: #000000;
46 /*Espacio horizontal entre letras*/
47 letter-spacing: 4px;
48 }
49
50 /*Selector de clase que aplica un estilo de
51 botones cuando el mouse está sobre la ligas*/
52 .primary-btn:hover {
53 /*Elimina el subrayado de la ligas*/
54 text-decoration: none;
55 /*Color de texto*/
56 color: #000000;
57 /*Color de fondo*/
58 background: #00C2C8;
59 }
60
61 /*Aporta estilo a la estructura de cada artículo del blog */
62 .article
63 {
64 /*Definición de la estructura que se comporte como tabla
65 con el modelo flexbox */
66 position: relative;
67 display: flex;
68 flex-direction: column;
69 /*Agrega un borde a toda la estructura del artículo*/
70 border: 1px solid rgba(0,0,0,.125);
71 border-radius: .25rem;
72 /*Define el ancho de la estructura del artículo*/
73 width: 40%;
74 }
75
76 /*Define el ancho de la imagen del artículo */
77 .image-article
78 {
79 /*Se estructura al 100% del ancho de la columna*/
80 width: 100%;
81 }
```

Nota. Esta figura fue desarrollada para el practicario de la asignatura de Arquitectura de la Información

Desarrollo práctica 5:

13. Valuar el código, ejecuta el proyecto abriendo la página *index.html* en el navegador; da clic sobre el botón Mostrar Artículos y se debe poblar la página con cuatro artículos generados dinámicamente. En caso de errores o de no lograr el resultado esperado, revisar el código.

Resultados obtenidos:

Al término de la práctica se habrá realizado un sitio web con contenido dinámico, que consume datos semi estructurados en formato *JSON* con ayuda de *Javascript* a través de *AJAX*.

Evidencia de la práctica: (fotografías, videos, archivo, etc.)

Integrar la estructura que conforman el proyecto de la página web (archivo *index.html*, carpeta *css* con archivo *style.css*, carpeta *img* con imágenes, carpeta *js* con archivo *script.js*, y carpeta *data* con archivo *articulos.json*.

Conclusiones:

Describe en qué funcionalidades puedes aplicar el uso de datos semiestructurados y qué otros formatos de datos semi estructurados existen.

Bibliografía:

De la web:

- Mozilla . (17 de Marzo de 2022). MDN Web Docs. Obtenido de AJAX: <https://developer.mozilla.org/es/docs/Web/Guide/AJAX>
- Mozilla. (17 de Marzo de 2022). MDN Web Docs . Obtenido de Trabajando con JSON: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>
- Replit. (20 de Febrero de 2022). Replit. Obtenido de Crea una cuenta en Replit: <https://replit.com/signup>

Criterios de evaluación:

- Valuar la página *index.html* comprobando su funcionamiento basado en lo solicitado en la práctica.
- Código libre de errores y documentado.

Campus: Atizapán

Elaboró: Judith Miguel Hernández (Docente)

Revisor: Miguel Coss Guerrero (Director Académico) /DTEIP

Vigente a partir del ciclo: 22-3

Última actualización: 18/03/2022