



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO



Innovación, Ciencia  
y Tecnología



**INSTITUTO TECNOLÓGICO**  
**JOSÉ MARIO MOLINA**  
**PASQUEL Y HENRÍQUEZ**

## **PROGRAMACION WEB**

### **Practica “Lagos de Moreno”**

**Profesor:** Pedro Espinosa Esparza

**Alumna:** Zelma Guadalupe Ruiz Ramos

05 de marzo de 2020

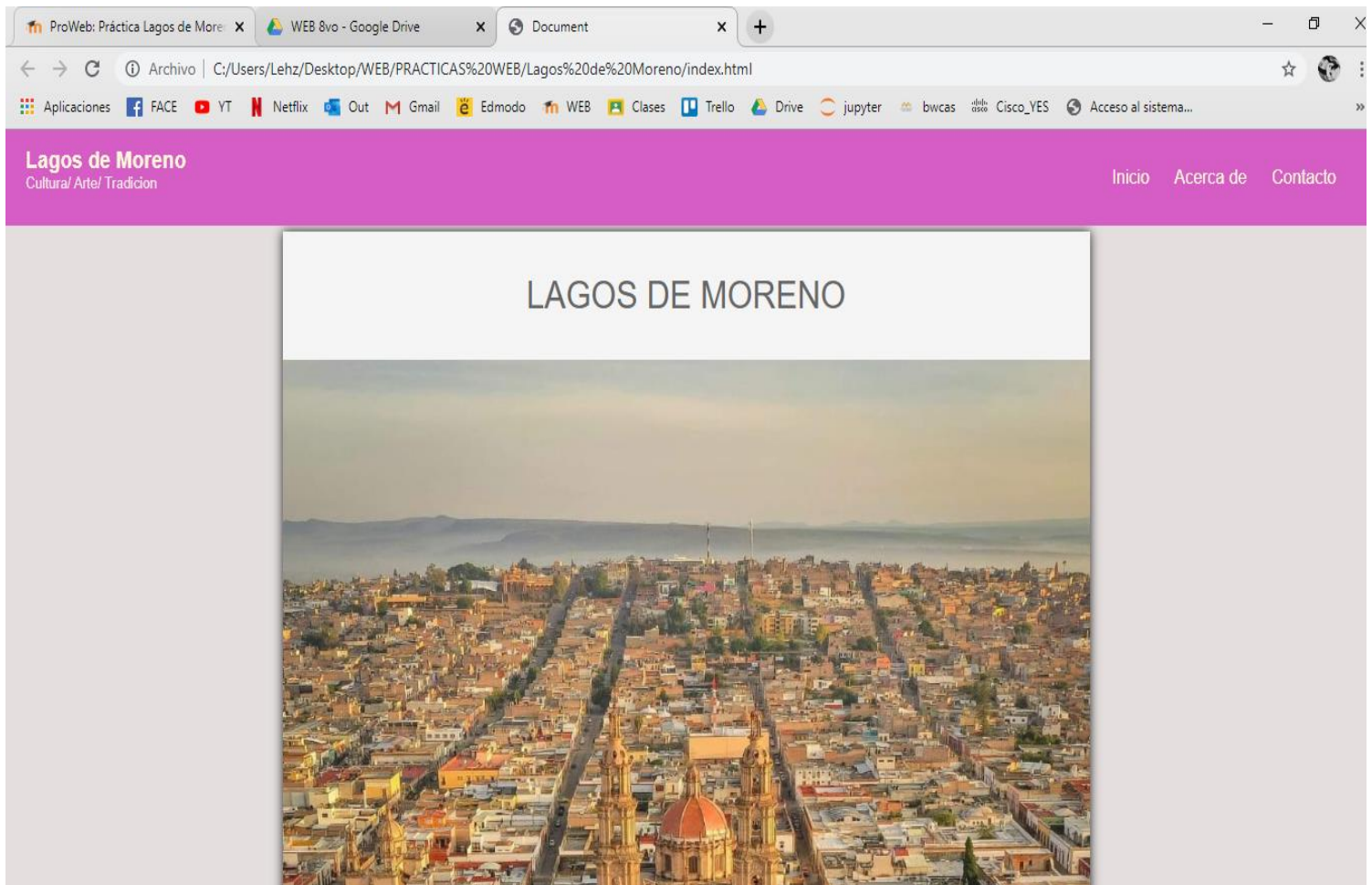


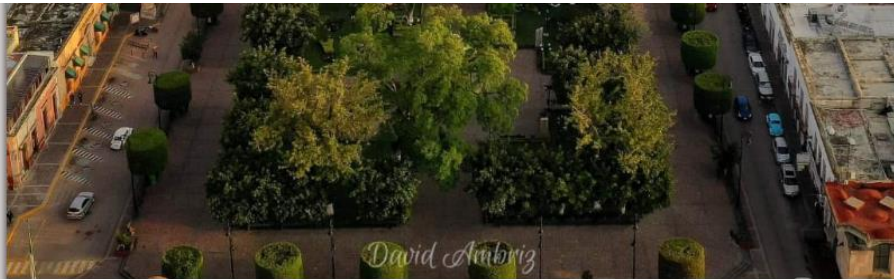
Esta práctica consiste en la creación de una pagina web de lagos de moreno en la cual buscamos generar un blog para el municipio con acceso a alguna reseña y contactos para la búsqueda de alguna información adicional.

Para la creación de esta pagina usamos como recurso las hojas de estilo CSS que nos ayudan a darle una mejor facilidad de creación en cuanto a colores y estilos.

Esta pagina esta compuesta de 3 vistas que son:

- ✓ **Índex:** En este apartado generamos lo que viene siendo el cuerpo o estructura de la vista, es decir nuestra cabecera y pie de página, donde en la cabecera mostramos el título de la página que es “Lagos de Moreno” con su lema y además de 3 apartados donde nos redirigirá a las demás vistas, esto que se agregó tendrá lugar en todas las vistas como un estilo general lo que cambia es el contenido ya que en el índex se cuenta con un contenido con margen donde al agregarle datos se convierte en un blog.





La composición de funciones nos permite simplificar las funciones grandes y complejas separando en pequeñas funciones atómicas mucho más simples de probar. Existen varias herramientas que nos ayudan con esta tarea. Ramda es una de estas herramientas, con su función compose, le podemos pasar un número indeterminado de funciones, y las concatena, devolviendo una función que recibe un parámetro, y devuelve un valor.

En la programación funcional, evitamos a funciones como splice que mutan datos. Lo que buscamos son funciones limpias y confiables que regresen el mismo resultado siempre, no funciones que dejen un desastre.

Autor: Zelma Ruiz

Ahora si se llegara agregar otro texto es decir otro apartado simplemente se reflejaría de manera que estuviera independiente del primer reporte.

Por ejemplo: Aquí observamos que muestra una división, es por lo mismo que el otro texto dentro del blog es independiente del primero.

ProWeb: Práctica Lagos de More... WEB 8vo - Google Drive Document

Archivo | C:/Users/Lehz/Desktop/WEB/PRACTICAS%20WEB/Lagos%20de%20Moreno/index.html

Aplicaciones FACE YT Netflix Out Gmail Edmodo WEB Clases Trello Drive jupyter bwcas Cisco\_YES Acceso al sistema...

## Lagos de Moreno

Cultura/ Arte/ Tradición

Inicio Acerca de Contacto

pequeñas funciones atómicas mucho más simples de probar. Existen varias herramientas que nos ayudan con esta tarea. Ramda es una de estas herramientas, con su función compose, le podemos pasar un número indeterminado de funciones, y las concatena, devolviendo una función que recibe un parámetro, y devuelve un valor.

En la programación funcional, evitamos a funciones como splice que mutan datos. Lo que buscamos son funciones limpias y confiables que regresen el mismo resultado siempre, no funciones que dejen un desastre.

Autor: Zelma Ruiz

## LAGOS DE MORENO



Para darle estos acabados al diseño utilizamos el siguiente código en CSS donde especificamos desde el cuerpo general hasta la cabecera y de página además de crear las clases para los textos, párrafos, imágenes, artículos y en el hove, entre otras indicaciones generadas para la creación de estos estilos en los cuales ordenamos de manera estructurada la pagina para obtener una buena vista:

```
/*General*/
body{
    margin: 0;
    padding: 8em;
    font-
family: Arial, Helvetica, sans-
serif;
    color: #666;
    background: rgb(228, 223, 223);
    font-size: 1em;
    padding-top: 80px;

    background-
image: url(.../IMG/fondo.jpg);
    background-repeat: no-repeat;
    background-attachment: fixed;
}
/*PARRAFOS*/
p{
    text-align: justify;
}

/*todos los h1*/
h1{
    font-size: 2em;
    margin: 15px 0;
    font-weight: 300;
    text-align: center;
}
/*imagenes*/
img{

    max-width: 100%;
    height: auto;
    /*max-width: auto;
    max-height: auto;
    display: block;
```

```
margin: auto;*/
}

/*CABECERA*/

#main-header{
    background-
color: rgb(214, 93, 198);
    color: cornsilk;
    height: 80px;
    width: 100%;
    left: 0;
    top: 0;
    position: fixed;
}

#main-header a{
    color: cornsilk;
}

#logo-header{
    float: left;
    padding: 15px 0 0 20px;
    text-
decoration: none; /*quita las linea
s*/
}

/*cuando pongas encima es hove*/
#logo-header:hover{
    color: rgb(114, 15, 76);
}
```



```
#logo-header .site-name{
  display: block;
  font-weight: 700;
  font-size: 1.2em;
}

#logo-header .site-desc{
  display: block;
  font-weight: 300;
  font-size: 0.8em;
  color: rgb(245, 229, 229);
}

/*NAVEGACION*/
nav{
  float: right;
}

nav ul{
  margin: 0;
  padding: 0;
  list-style: none;
  padding-right: 20px;
}

nav ul li{
  display: inline-block;
  line-height: 80px;
}

nav ul li a{
  display: block;
  padding: 0 10px;
  text-decoration: none;
}

nav ul li a:hover{
  background-
color: rgb(85, 21, 85, 0.2);
}
```

```
/*contenido*/

#main-content{
  background: whitesmoke;
  width: 100%;
  max-width: 800px;
  margin: 20px auto;
  box-
shadow: 0 0 10px rgb(0, 0, 0,0.9);
}

#main-content header,
#main-content .content{
  padding: 20px;
}

/*Pie de pagina*/
#main-footer{
  background:rgb(214, 93, 198) ;
  color: white;
  text-align: right;
  padding: 20px;
  margin-top: 40px;
}

#main-footer p{
  margin: 0;
}

/*articulo*/
.footerArticle{
  background: rgb(238, 168, 255);
  color: rgb(109, 31, 79);
  padding: 1px;
}

.footerArticle p{
  font-weight: 500;
  font-style: oblique;
}
```





**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

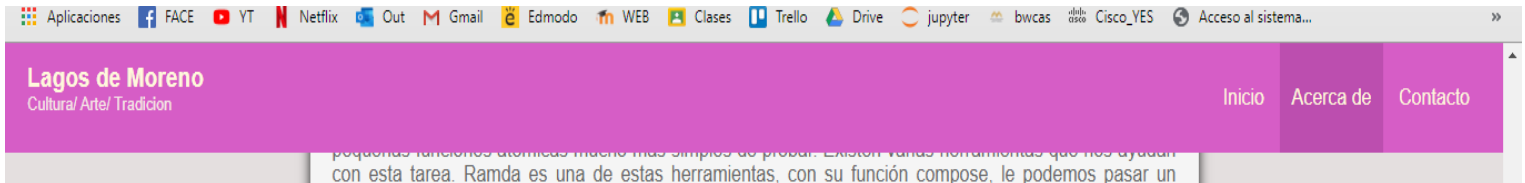


Innovación, Ciencia  
y Tecnología



Cuando agregamos el `hover` en la hoja de estilos además de las otras instrucciones, lo que genera es que al pasar el mouse por encima del apartado al que asignamos cambiara de color, esto hace que en el usuario sea una experiencia más atractiva visualmente.

Ejemplo: De esta manera se visualiza cuando el mouse pasa por donde están las casillas marcadas es decir que se encuentran previamente designadas.



Desde las hojas de estilos CSS se acomoda de la siguiente manera el código:

```
/*/cuando pongas encima es hove*/
#logo-header:hover{
    color: rgb(114, 15, 76);
}
```

Dentro del código de HTML para mandar llamar el `hover` es de la siguiente manera:

```
<header id="main-header">
  <a id="logo-header" href="#">
    <span class="site-name">Lagos de Moreno</span>
    <span class="site-desc">Cultura/ Arte/ Tradicion</span>
  </a>
  <nav>
    <ul>
      <li><a href="index.html">Inicio</a></li>
      <li><a href="acerca.html">Acerca de</a></li>
      <li><a href="contacto.html">Contacto</a></li>
    </ul>
  </nav>
</header>
```

Ahora para mostrar el código terminado de la primera pagina se muestra el código completo del HTML que es en el cual se crea toda la página del Índice.



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="CSS/estilos.css" >
</head>
<body>
  <header id="main-header">
    <a id="logo-header" href="#">
      <span class="site-name">Lagos de Moreno</span>
      <span class="site-desc">Cultura/ Arte/ Tradicion</span>
    </a>
    <nav>
      <ul>
        <li><a href="index.html">Inicio</a></li>
        <li><a href="acerca.html">Acerca de</a></li>
        <li><a href="contacto.html">Contacto</a></li>
      </ul>
    </nav>
  </header>

  <section id=main-content>
    <article>
      <header>
        <h1>LAGOS DE MORENO</h1>
      </header>
      

      <div class="content">
        <p>
          La composición de funciones nos permite simplificar las func
iones grandes y complejas
          separando en pequeñas funciones atómicas mucho más simples d
e probar.

          Existen varias herramientas que nos ayudan con esta tarea. R
amda es una de estas
          herramientas, con su función compose, le podemos pasar un nú
mero indeterminado de
          funciones, y las concatena, devolviendo una función que reci
be un parámetro,
          y devuelve un valor.
        </p>
      </div>
    </article>
  </section>
</body>
</html>
```



```

    </p>
    <p>
        En la programación funcional, evitamos a funciones como splice que mutan datos.
        Lo que buscamos son funciones limpias y confiables que regresen el mismo resultado
        siempre, no funciones que dejen un desastre.
    </p>
</div>
<footer class="footerArticle">
    <p>
        Autor: Zelma Ruiz
    </p>

</footer>
</article>
</section>

<!--DESDE SECTION PARA CORTAR Y SEPARAR-->
<section id=main-content>
    <article>
        <header>
            <h1>LAGOS DE MORENO</h1>
        </header>
        

        <div class="content">
            <p>
                La composición de funciones nos permite simplificar las funciones grandes y complejas
                separando en pequeñas funciones atómicas mucho más simples de probar.

                Existen varias herramientas que nos ayudan con esta tarea. Ramda es una de estas
                herramientas, con su función compose, le podemos pasar un número indeterminado de
                funciones, y las concatena, devolviendo una función que recibe un parámetro,
                y devuelve un valor.
            </p>
            <p>
                En la programación funcional, evitamos a funciones como splice que mutan datos.
            </p>
        </div>
    </article>
</section>

```





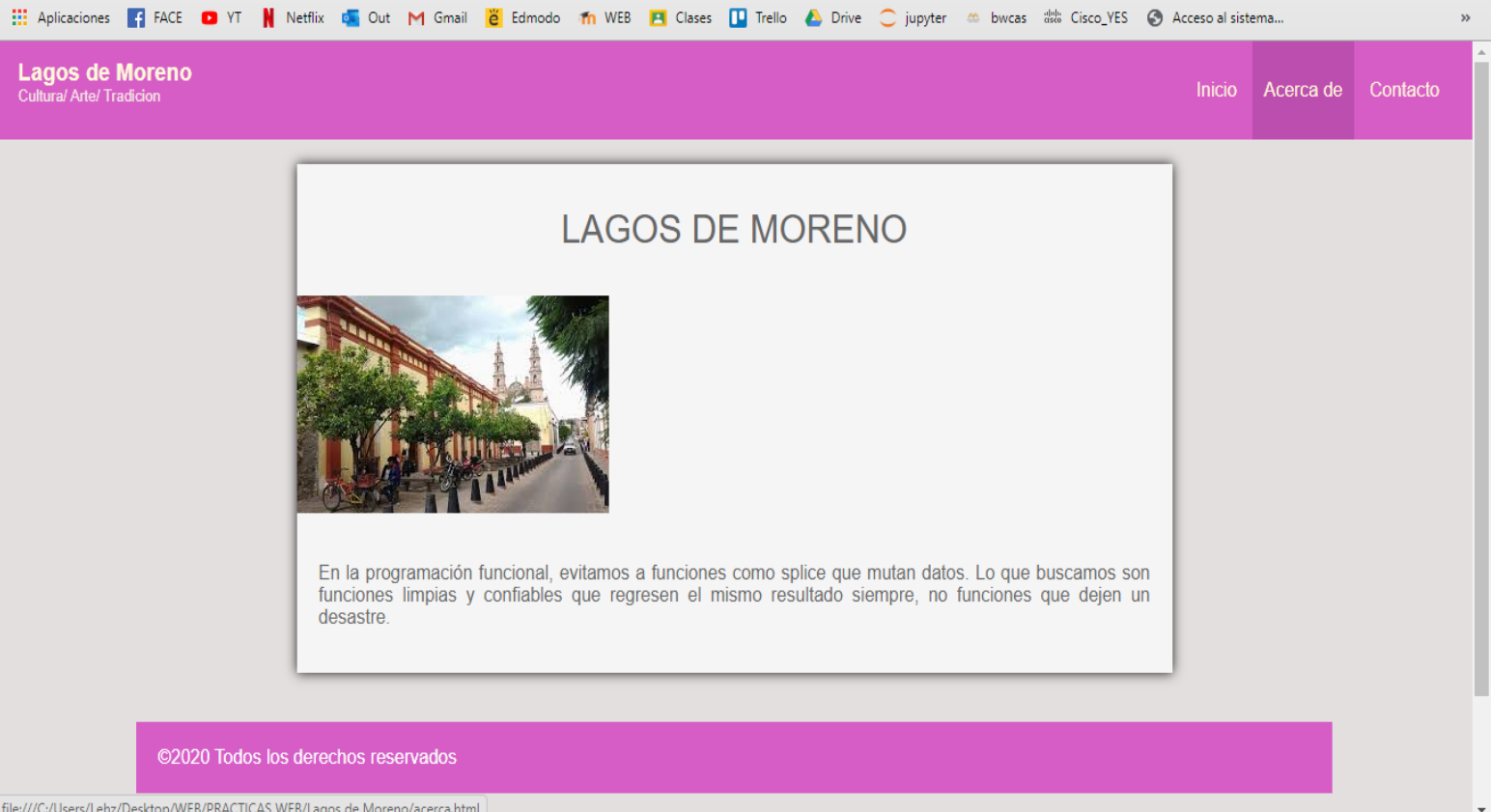
Lo que buscamos son funciones limpias y confiables que regresen el mismo resultado

siempre, no funciones que dejen un desastre.

```
</p>
</div>
<footer class="footerArticle">
  <p>
    Autor: Zelma Ruiz
  </p>
</footer>
</article>
</section>

<footer id="main-footer">
  <p>&copy;2020 Todos los derechos reservados</p>
</footer>
</body>
</html>
```

✓ **Acerca de:** La vista es la siguiente:





Su código en HTML es el siguiente:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="CSS/estilos.css" >
</head>
<body>
  <header id="main-header">
    <a id="logo-header" href="#">
      <span class="site-name">Lagos de Moreno</span>
      <span class="site-desc">Cultura/ Arte/ Tradicion</span>
    </a>
    <nav>
      <ul>
        <li><a href="index.html">Inicio</a></li>
        <li><a href="acerca.html">Acerca de</a></li>
        <li><a href="contacto.html">Contacto</a></li>
      </ul>
    </nav>
  </header>

  <section id="main-content">
    <article>
      <header>
        <h1>LAGOS DE MORENO</h1>
      </header>
      

      <div class="content">
        <p>
          En la programación funcional, evitamos a funciones como split
          que mutan datos.
          Lo que buscamos son funciones limpias y confiables que regresen
          el mismo resultado
          siempre, no funciones que dejen un desastre.
        </p>
      </div>
    </article>
  </section>
```



```
<footer id="main-footer">
  <p>&copy;2020 Todos los derechos reservados</p>
</footer>

</body>
</html>
```

Prácticamente el CSS es el mismo para todos ya que el mismo patrón se usa en las 3 vistas.

- ✓ **Contacto:** En esta vista creamos la interfaz con contenedores para lograr escribir y anexar texto además de agregar un botón creado desde el HTML.

Aplicaciones FACE YT Netflix Out Gmail Edmodo WEB Clases Trello Drive jupyter bwcas Cisco\_YES Acceso al sistema...

**Lagos de Moreno**  
Cultura / Arte / Tradición

Inicio Acerca de Contacto

## CONTACTO

Nombre:

Correo:

Contenido

El código de HTML es el siguiente para la creación de esta vista donde se muestra la creación de los botones y de cada contenedor de texto.

```
<!DOCTYPE html>
<html lang="en">
<head>
```



```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<link rel="stylesheet" href="CSS/estilos.css" >

</head>
<body>
  <header id="main-header">
    <a id="logo-header" href="#">
      <span class="site-name">Lagos de Moreno</span>
      <span class="site-desc">Cultura/ Arte/ Tradicion</span>
    </a>
    <nav>
      <ul>
        <li><a href="index.html">Inicio</a></li>
        <li><a href="acerca.html">Acerca de</a></li>
        <li><a href="contacto.html">Contacto</a></li>
      </ul>
    </nav>
  </header>

  <section id=main-content>
    <article>
      <header>
        <h1>CONTACTO</h1>

        <div class="content">
          <form>
            <label for="">Nombre:</label><br>
            <input type="text" name="" id=""><br>
            <label for="">Correo:</label><br>
            <input type="text" name="" id=""><br>
            <label for="">Contenido</label><br>
            <textarea name="" id="" cols="30" rows="10"></textarea><br>
            <button type="submit">Enviar</button>

          </form>
        </div>

      </article>
    </section>

  </body>
</html>
```

## Conclusión:

Para concluir la practica es importante recalcar que cada paso que se dio para la creación de la página lleva un proceso largo para estructurar correctamente las vistas ya que ir creando cada parte es un poco tedioso, pero cuando ya se cuenta con la hoja de estilos solo es mandar llamar y acortar el paso del tiempo en la estructura ya que le damos instrucciones de las cuales ya solo mandamos llamar en lo que se requiere dándole un color, una forma, una estructura ya definida para que la vista de la misma sea amigable y al gusto del usuario.