



Diplomado virtual en
PROGRAMACIÓN EN PHP
Guía didáctica 4: Desarrollo web I



Formación Virtual

.....educación sin límites



Competencia específica

Se espera que, con los temas abordados en la guía didáctica del módulo 4: Desarrollo web I, el estudiante logre la siguiente competencia específica:

- Conocer los conceptos básicos de la programación web en PHP.



Contenidos temáticos

Los contenidos temáticos, para desarrollar en la guía didáctica del módulo 4: Desarrollo web I, son:

Desarrollo web

HTML

HTML - PHP

Bases de datos - MySQL (parte I)

Ilustración 1: caracterización de la guía didáctica.

Fuente: autor¹.

¹ Todas las ilustraciones y tablas de esta guía son autoría propia del docente y tienen como función mostrar la aplicación y práctica del contenido que se desarrolla, por ello solo se enumerarán.

Tema 1: Desarrollo Web

PHP es el lenguaje de programación 100 % para el desarrollo del web del lado del *backend* más popular, pero ¿qué es el *backend*?

Backend

Backend es un término que se usa para referirse a un ámbito de la programación de aplicaciones y *software*, es la parte que se encarga del acceso y manipulación a los datos y la seguridad de la información. Aunque se usa mucho para referirnos a la programación web que se realiza del lado del servidor, es válido hablar de *backend* para el desarrollo de todo tipo de aplicaciones.

Con PHP puedes desarrollar sitios y aplicaciones de todo tipo. PHP es el motor además de los [CMS](#) más populares, como WordPress, Joomla, Drupal o Magento, entre muchos otros.

Una de las ventajas de PHP es su facilidad de uso. Cualquier persona con muy pocos conocimientos puede comenzar a usar PHP para el desarrollo de páginas web. Sin embargo, esa misma ventaja también puede ser un inconveniente, pues si no se dispone de la debida formación es fácil caer en malas prácticas.

Otra gran ventaja es su disponibilidad, ya que lo encontramos en la mayoría de los alojamientos web de todo tipo.

Al hablar de desarrollo web con PHP debemos hablar inmediatamente de HTML y la estrecha relación que existe entre estos.

Todo el mundo de la programación web se sienta sobre las bases de lo que es HTML, toda página, componente o estructura web debe estar ceñida sobre HTML. PHP nos facilita la vida gracias a que el vínculo programacional que existe entre ambos permite crear un dinamismo muy potente en el desarrollo web bajo PHP y haciendo uso de HTML.

PHP se integra con mucha facilidad en páginas HTML y no por nada es uno de los lenguajes más utilizado y soportados en el desarrollo web, más adelante en

esta guía entenderá y conocerá la forma en que PHP y HTML se dan la mano en el desarrollo.

Los datos también juegan un papel clave en el desarrollo de *software* web. Cada lenguaje de programación debe estar diseñado para soportar los motores de bases de datos más populares en la actualidad, SQL, MySQL PostgreSQL, etc.; PHP en este caso tiene una estrecha relación directa con MySQL, dado que viene integrado dentro del motor de ejecución de este (XAMPP) y permite administrar una base de datos desde su creación hasta sus copias de seguridad con increíble facilidad.

Ahora, ya con la estructura de un sitio web (HTML), un lenguaje de programación potente (PHP) y un motor de bases de datos funcional e integrador con PHP (MySQL) simplemente hay que darle rinda suelta a la imaginación, al código y al futuro.

Tema 2: HTML

En pocas palabras, HTML, por sus siglas en inglés de *hypertext markup language*, hace referencia al lenguaje de marcado para la elaboración de páginas web.

- HTML significa lenguaje de marcado de hipertexto.
- HTML describe la estructura de una página web.
- HTML consiste en una serie de elementos.
- Los elementos HTML le dicen al navegador cómo mostrar el contenido.
- Los elementos HTML están representados por etiquetas.
- Las etiquetas HTML etiquetan piezas de contenido como «encabezado», «párrafo», «tabla», etc.
- Los navegadores no muestran las etiquetas HTML, pero las usan para representar el contenido de la página.

Normalmente HTML se presenta por archivo con la extensión .html y al ser un lenguaje de marcado independiente de PHP no hace falta XAMPP o Apache (mientras no se escriba código PHP), un ejemplo puede ser el siguiente: crear un nuevo folder dentro del proyecto diplomado llamado: «html», en este caso trabajaremos con un solo archivo de extensión .html y se llamará «pagina.html», de la siguiente forma:

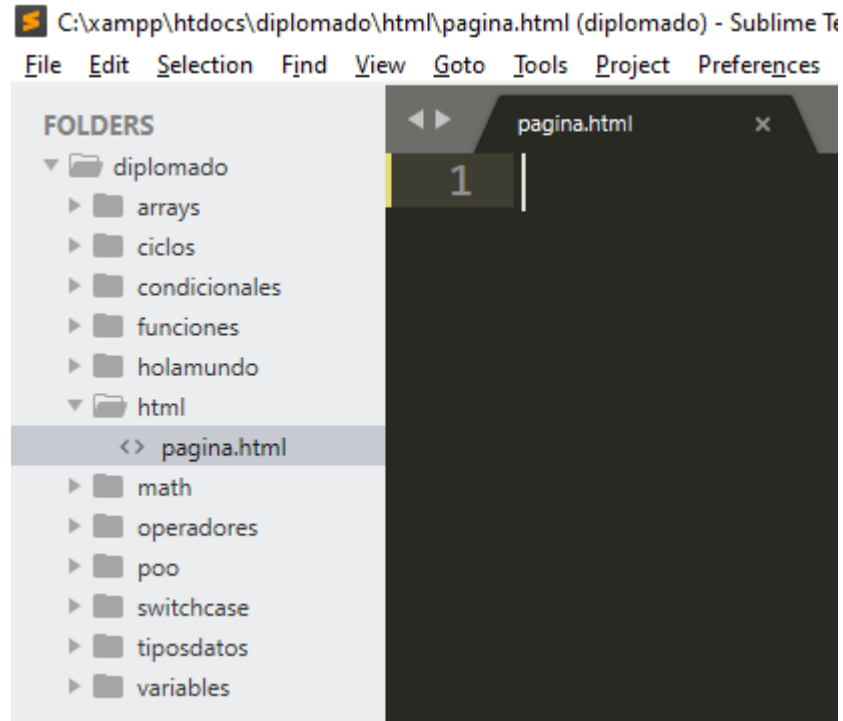


Ilustración 2.

HTML es un poco extenso de explicar, pero nos centraremos en los conceptos y características que utilizaremos en el diplomado, como lo son:

- Elementos básicos.
- Tablas.
- Formularios.
- Botones.
- Hipervínculos.
- Imágenes.
- Atributos y etiquetas.

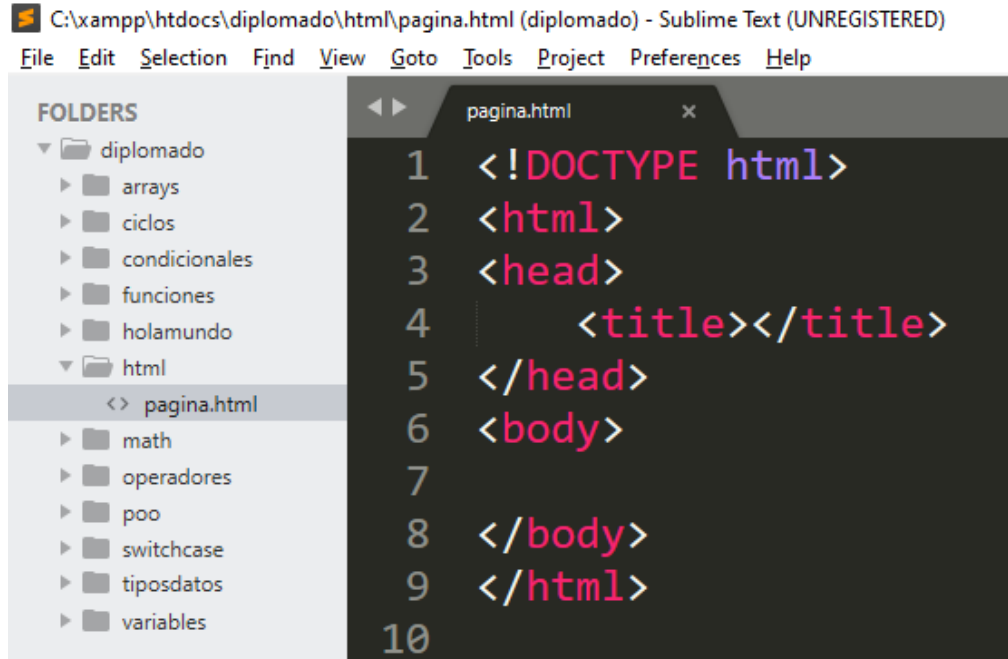
Iniciemos creando nuestra primera página y mostremos algo básico, el «Hola Mundo»:

Toda página en HTML tiene unas etiquetas base que se encargan de dividir el contenido de esta:

- `<!DOCTYPE html>`.
- `<html>`.
- `<head>`.

- <title>.
- <body>.

En el código se vería de la siguiente forma:



C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8  </body>
9  </html>
10
  
```

Ilustración 3.

Existe algo particular con HTML y es que toda etiqueta tiene una etiqueta de cierre: <> </>. Esto permite delimitar el alcance de un bloque de código. Veamos cómo se ve la página en el navegador simplemente con este formato.

Para observar el contenido de esta, sin usar XAMMP o Apache, debemos usar la dirección exacta del archivo o, más fácil aún, buscando el mismo en nuestro computador:

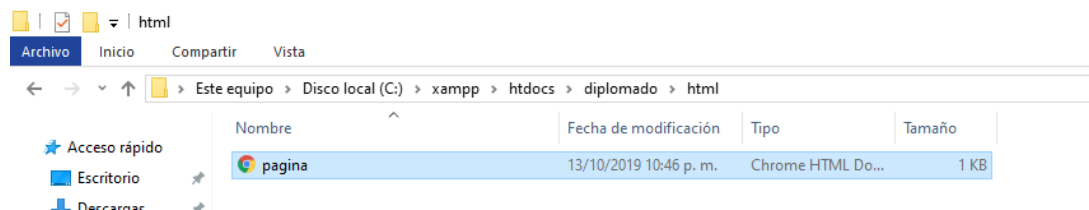


Ilustración 4.

Y simplemente abrirlo con doble clic o yendo directamente a la URL en el navegador: <file:///C:/xampp/htdocs/diplomado/html/pagina.html>.

Recuerda que no estamos usando XAMMP; si desea hacerlo utilizando XAMMP, simplemente realiza el proceso al igual que en los módulos anteriores, la URL sería: <http://localhost/diplomado/html/pagina.html>.

Eres libre de escoger como trabajar, el resultado será el mismo.

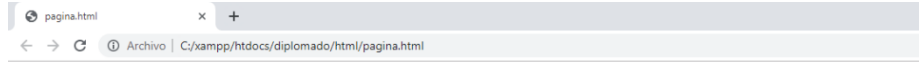


Ilustración 5.

El resultado es la pantalla completamente en blanco, esto es debido a que el navegador no muestra las etiquetas de HTML, simplemente muestra el contenido dentro de estas.

Ahora démosle contenido al nuestro documento, el «Hola Mundo». En este caso usaremos una nueva etiqueta muy particular.

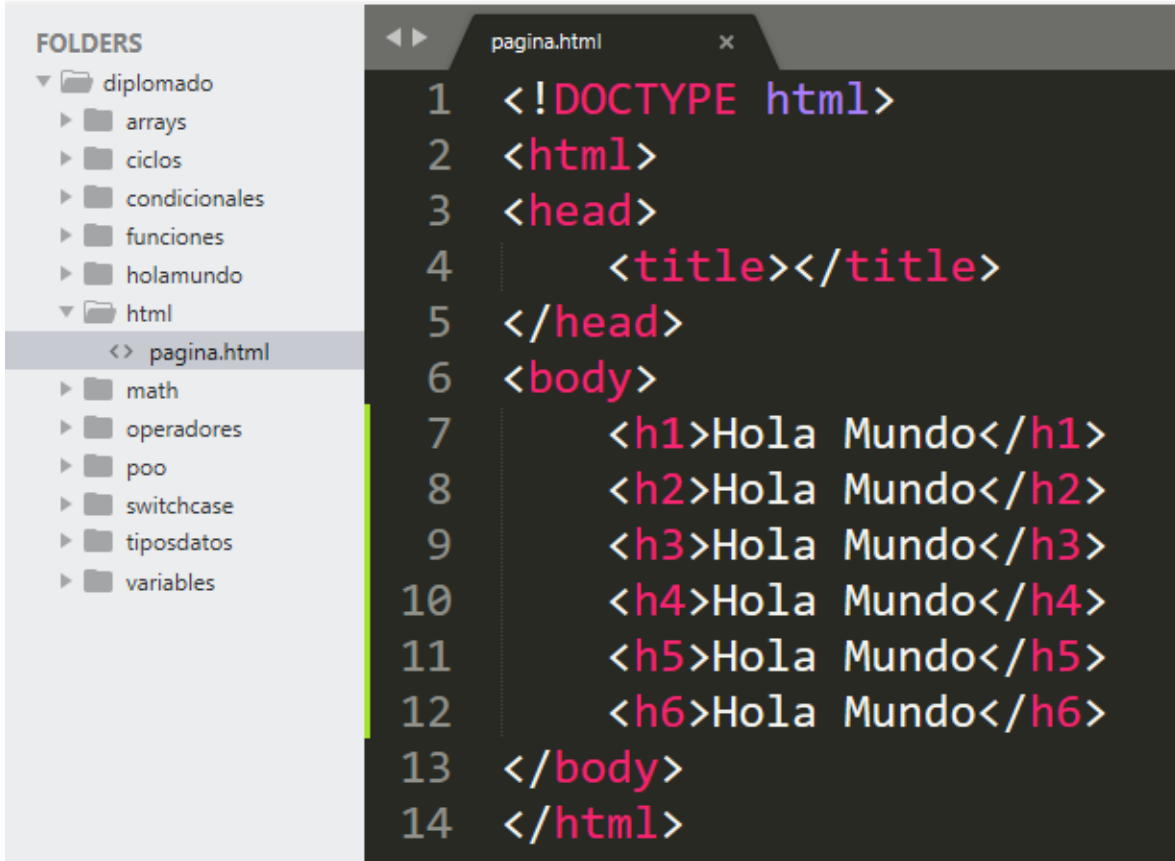
Encabezados

Los encabezados se definen con las etiquetas **<h1>** a **<h6>**.

<h1> define el encabezado más importante. **<h6>** define el encabezado menos importante.

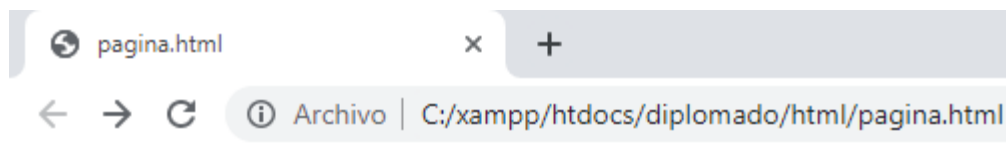
C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help



```
FOLDERS
└─ diplomado
  └─ arrays
  └─ ciclos
  └─ condicionales
  └─ funciones
  └─ holamundo
  └─ html
    └─ <> pagina.html
      └─ math
      └─ operadores
      └─ poo
      └─ switchcase
      └─ tiposdatos
      └─ variables

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title></title>
5 </head>
6 <body>
7     <h1>Hola Mundo</h1>
8     <h2>Hola Mundo</h2>
9     <h3>Hola Mundo</h3>
10    <h4>Hola Mundo</h4>
11    <h5>Hola Mundo</h5>
12    <h6>Hola Mundo</h6>
13 </body>
14 </html>
```



Hola Mundo

Hola Mundo

Hola Mundo

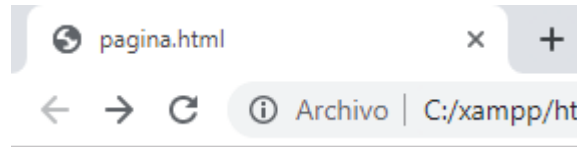
Hola Mundo

Hola Mundo

Hola Mundo

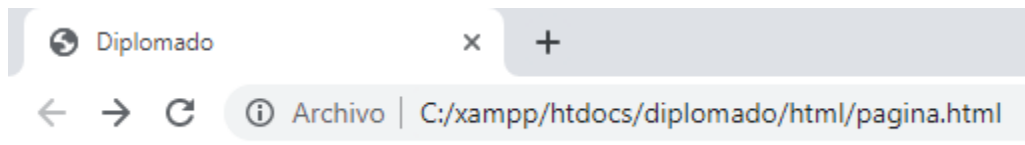
Ilustraciones 6 y 7.

Este ha sido el resultado de nuestra primera página web. Realicemos otro cambio. Dentro de sus etiquetas de HTML se encuentra: `<title></title>`. Title permite darle un título o nombre a nuestra página, si se fijan en la parte superior aparece el nombre del archivo.



Si se realiza el cambio en la etiqueta Title, el resultado sería el siguiente:

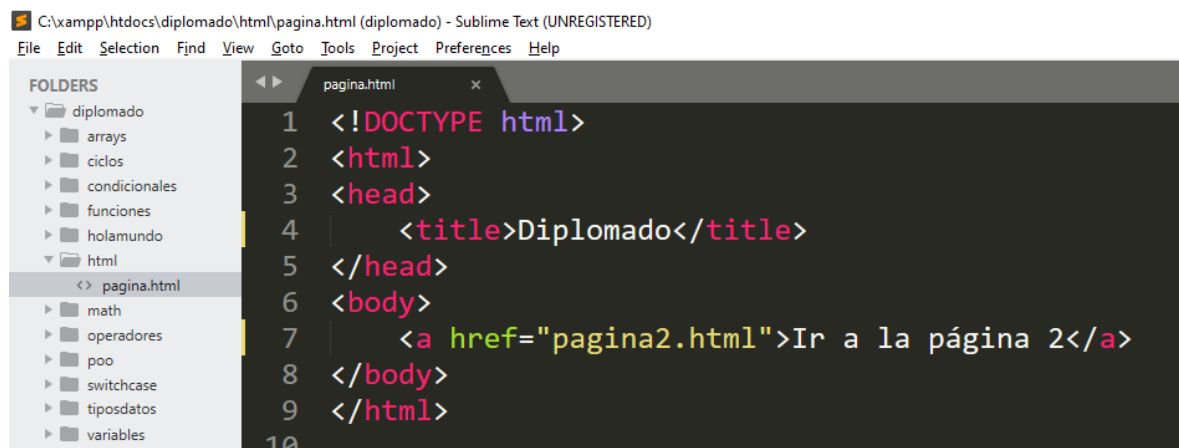
```
<head>
  <title>Diplomado</title>
</head>
```

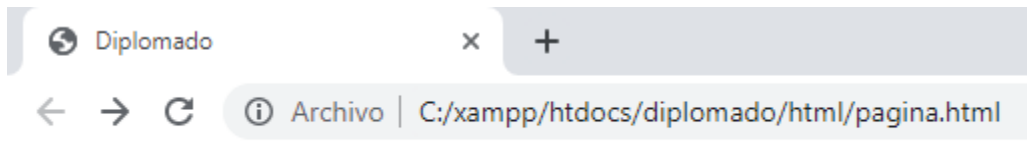


Ilustraciones 8, 9 y 10.

Enlaces - hipervínculos

Los enlaces permiten a los usuarios hacer clic de una página a otra y navegar en ese sentido. En HTML los hipervínculos se definen con la `<a>` y ``:





[Ir a la página 2](#)

Ilustraciones 11 y 12.

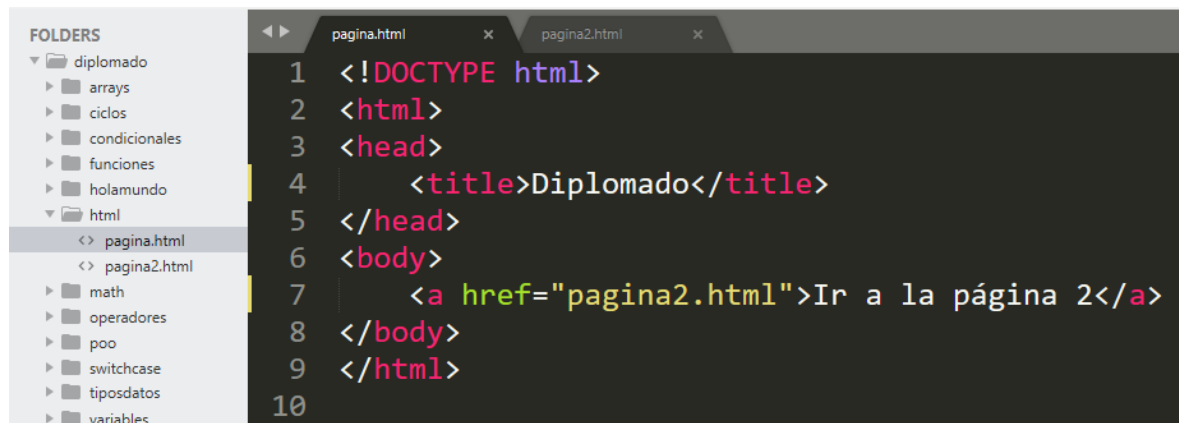
La etiqueta <a> implementa un nuevo concepto en HTML, que son los atributos. Los atributos son bloques de código que se encuentran en las etiquetas de HTML.

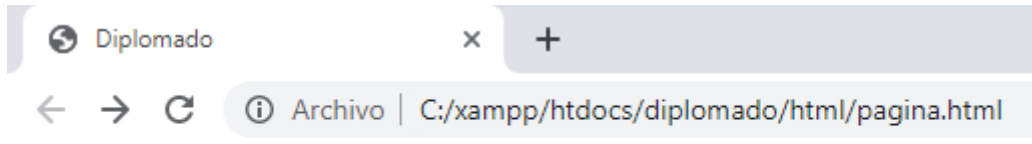
Atributos. Todos los elementos HTML pueden tener atributos.

- Los atributos proporcionan información adicional sobre un elemento.
- Los atributos siempre se especifican en la etiqueta de inicio.
- Los atributos generalmente vienen en pares nombre / valor como: nombre = "valor".

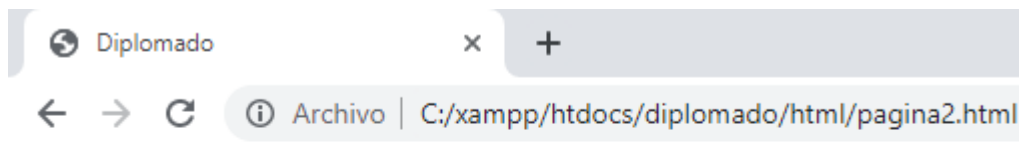
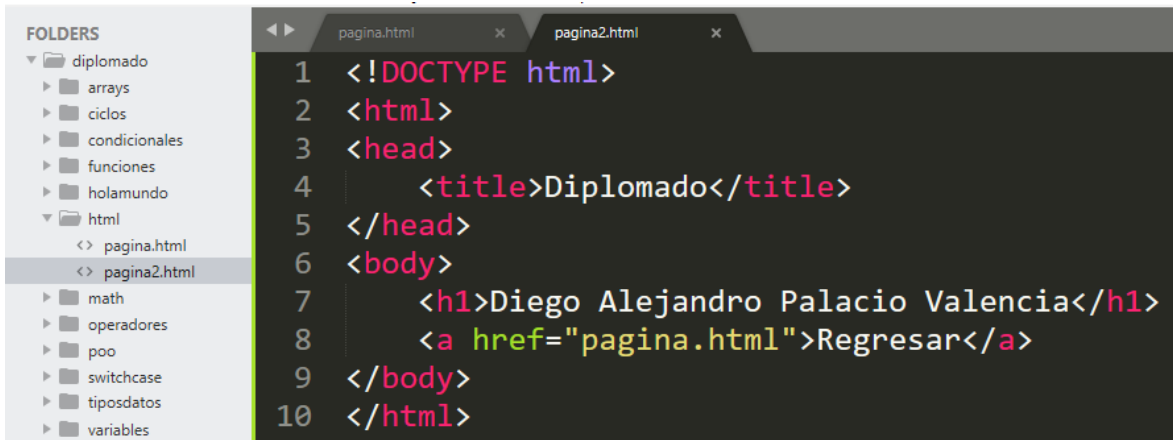
En el caso de la etiqueta <a>, el nombre del atributo es «href» y el valor que recibe es una URL.

Creemos una segunda página, con el nombre asignado a la respectiva etiqueta: "pagina2.html" con un encabezado que diga su respectivo nombre y otro enlace para regresar. El resultado sería el siguiente:





[Ir a la página 2](#)



Diego Alejandro Palacio Valencia

[Regresar](#)

Ilustraciones 13, 14, 15 y 16.

Imágenes. En HTML, las imágenes se definen con la ``. `` está vacía, contiene solo atributos y no tiene una etiqueta de cierre. El atributo `src` especifica la URL de la imagen:

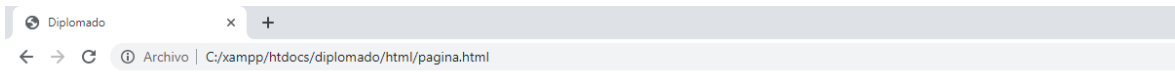
C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - logo-php.png
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Diplomado</title>
5 </head>
6 <body>
7     <h1>Imágenes</h1>
8     
9 </body>
10 </html>
```



Imágenes



Ilustraciones 17 y 18.

Para descargar esta imagen ve al siguiente enlace y guárdalo en folder html y del proyecto diplomado: <http://politecnicodecolombia.edu.co/paquetes/php/logo-php.png>.

 contiene otros atributos muy importantes que permiten trabajar con el tamaño de las imágenes.

```
<h1>Imagenes</h1>

```

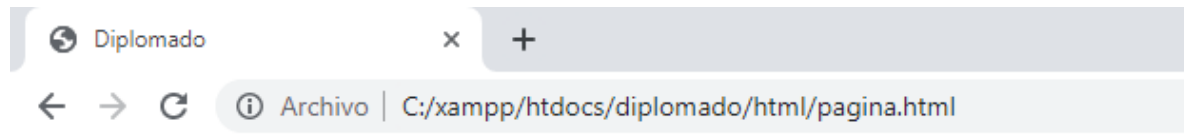
Ilustración 19.

- **Width:** define el ancho de la imagen.
- **Height:** define el alto de la imagen.

Los valores que reciben estos atributos son pixeles o porcentajes, en este caso manejemos los valores con pixeles. Veamos.

```
<h1>Imagenes</h1>

```



Imagenes



Ilustraciones 20 y 21.

Jugando un poco con las etiquetas podemos utilizar enlaces e imágenes en un solo bloque de código, digamos que deseamos que cuando demos clic en la imagen del logo de PHP vayamos a la página oficial de PHP. Este sería el código:



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7
8   <h1>Imagenes</h1>
9   <a href="https://www.php.net/"></a>
10
11 </body>
12 </html>
  
```

Ilustración 22.

Tablas

Una tabla HTML se define con la etiqueta **<table> </table>**.

Cada fila de la tabla se define con la etiqueta **<tr> </tr>**. Un encabezado de tabla se define con la etiqueta **<th> </th>**. Por defecto, los encabezados de las tablas están en negrita y centrados. Una tabla de datos / celda se define con la etiqueta **<td> </td>**.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

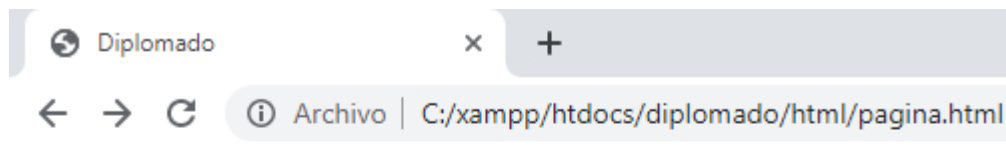
FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - logo-php.png
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <h1>Tablas</h1>
8      <table>
9          <tr>
10             <th>Nombres</th>
11             <th>Apellidos</th>
12             <th>Edad</th>
13         </tr>
14         <tr>
15             <td>Juan</td>
16             <td>Rich</td>
17             <td>29</td>
18         </tr>
19         <tr>
20             <td>Evelin</td>
21             <td>Londoño</td>
22             <td>42</td>
23         </tr>
24     </table>
25 </body>
26 </html>

```



Tablas

Nombres Apellidos Edad

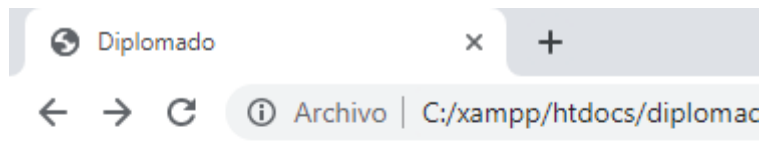
Juan Rich 29

Evelin Londoño 42

Ilustraciones 23 y 24.

A simple vista parece una tabla, hagamos uso de uno de los atributos de las tablas, en este caso «border».

```
<body>
  <h1>Tablas</h1>
  <table border="1">
    <tr>
      <th>Nombres</th>
      <th>Apellidos</th>
      <th>Edad</th>
```



Tablas

Nombres	Apellidos	Edad
Juan	Rich	29
Evelin	Londoño	42

Ilustraciones 25 y 26.

Cambia un poco más visualmente. La estructura de una tabla la podemos definir en tres partes.

- Encabezado.
- Fila.
- Columna.

El encabezado hace referencia directamente a los campos que representará la tabla y las etiquetas que lo definen son: **<tr>** y **<th>**, de la siguiente forma:

```
<tr>
  <th>Nombres</th>
  <th>Apellidos</th>
  <th>Edad</th>
</tr>
```

Las filas hacen referencia a cada uno de registros que representa la tabla frente a sus encabezados, cada fila representa un registro en la tabla y las etiquetas son: **<tr>**.

```
<tr>
  <td>Juan</td>
  <td>Rich</td>
  <td>29</td>
</tr>
```

Las columnas son datos específicos dentro de una fila en relación con un encabezado de la tabla, la etiqueta que la representa es: **<td>**

```
<tr>
  <td>Juan</td>
  <td>Rich</td>
  <td>29</td>
</tr>
```

Ilustraciones 27, 28 y 29.

Existen dos atributos muy importantes cuando se trabaja con filas, permite cambiar el estilo, orden y tamaño:

- Rowspan.
- Colspan.

Rowspan. Permite que una celda ocupe más de una fila.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

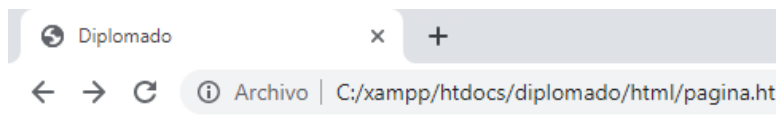
FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - logo-php.png
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <h1>Tablas</h1>
8      <table border="1">
9          <tr>
10             <th>Nombre:</th>
11             <td>Diego Palacio</td>
12          </tr>
13          <tr>
14             <th rowspan="2">Celular:</th>
15             <td>3108743692</td>
16          </tr>
17          <tr>
18             <td>3225874136</td>
19          </tr>
20          <tr>
21             <th>Nombre:</th>
22             <td>Carlos Palacio</td>
23          </tr>
24          <tr>
25             <th rowspan="2">Celular:</th>
26             <td>0345974130</td>
27          </tr>
28          <tr>
29             <td>3014782369</td>
30          </tr>
31      </table>
32  </body>
33  </html>

```

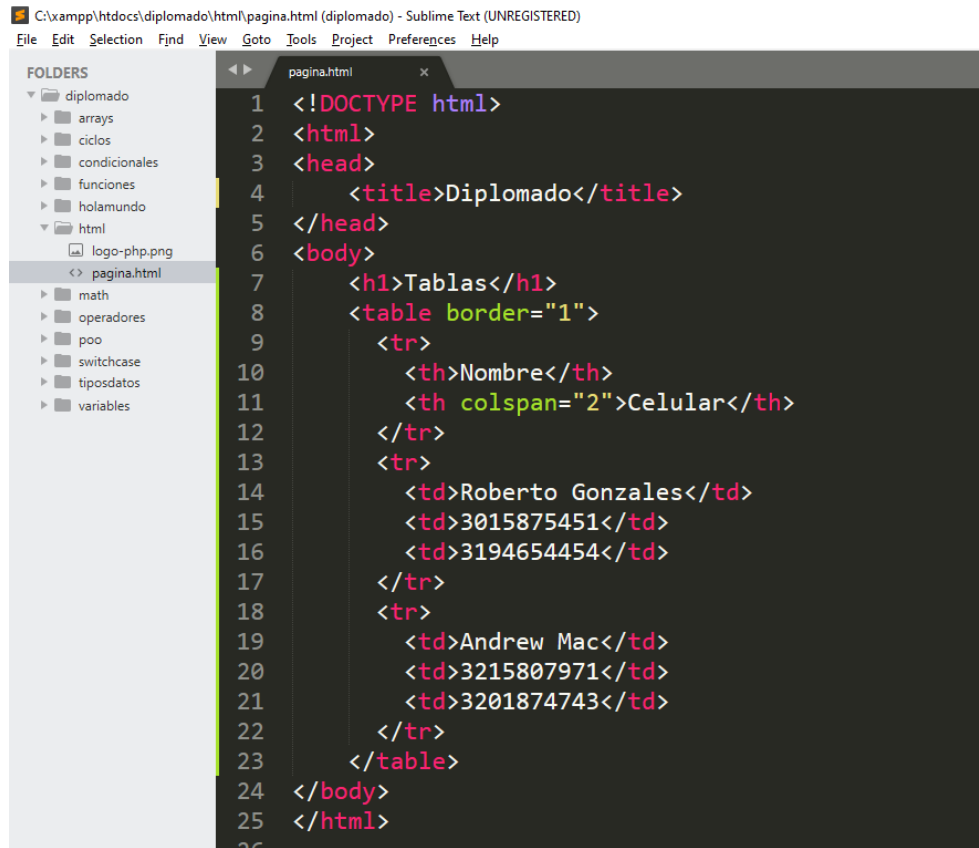


Tablas

Nombre:	Diego Palacio
Celular:	3108743692
	3225874136
Nombre:	Carlos Palacio
Celular:	0345974130
	3014782369

Ilustraciones 30 y 31.

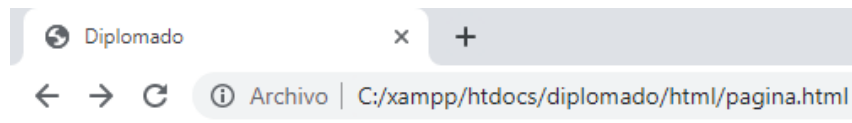
Colspan. Permite que una celda ocupe más de una columna.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <h1>Tablas</h1>
8   <table border="1">
9     <tr>
10      <th>Nombre</th>
11      <th colspan="2">Celular</th>
12    </tr>
13    <tr>
14      <td>Roberto Gonzales</td>
15      <td>3015875451</td>
16      <td>3194654454</td>
17    </tr>
18    <tr>
19      <td>Andrew Mac</td>
20      <td>3215807971</td>
21      <td>3201874743</td>
22    </tr>
23  </table>
24 </body>
25 </html>

```



Tablas

Nombre	Celular	
Roberto Gonzales	3015875451	3194654454
Andrew Mac	3215807971	3201874743

Ilustraciones 32 y 33.

Formularios

Con `<form></form>` HTML define un formulario que se utiliza para recopilar la entrada del usuario.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

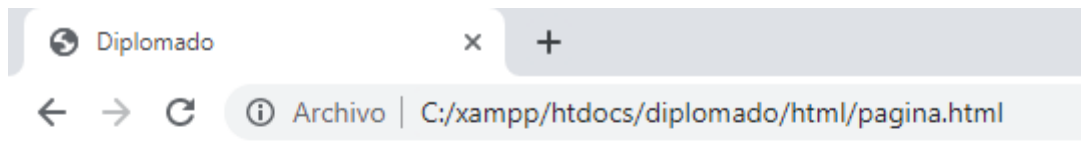
FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <h1>Formularios</h1>
8      <form>
9          Nombre:<br>
10         <input type="text" name="nombre">
11         <br>
12         Email:<br>
13         <input type="email" name="email">
14         <br><br>
15         <input type="submit" value="Enviar">
16     </form>
17 </body>
18 </html>

```



Formularios

Nombre:

Email:

Enviar

Ilustraciones 34 y 35.

Un formulario HTML contiene elementos de formulario, valga la redundancia.

Los elementos de formulario son diferentes tipos de elementos de entrada de datos, como campos de texto, casillas de verificación, botones de opción, botones de envío y más, pero todos deben de ir dentro de las etiquetas de **<form> </form>**.

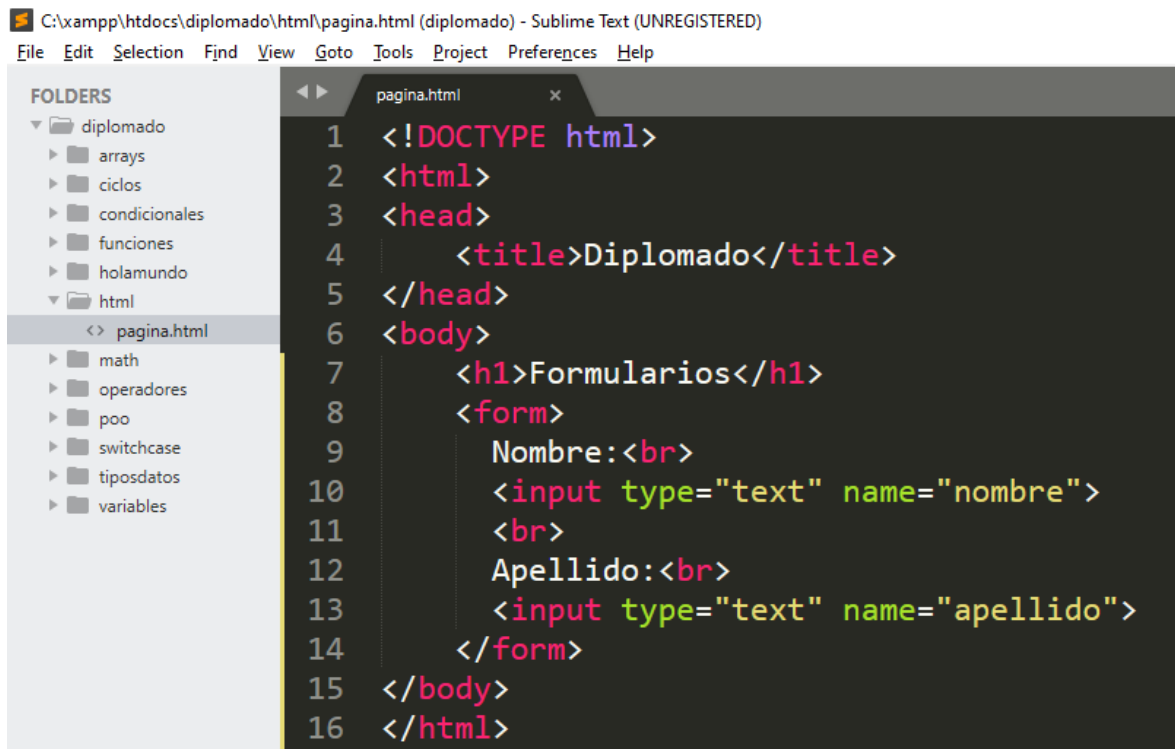
Los elementos de entrada son: *input*, *select*, *textarea*, *button*.

El elemento <input>. Es el elemento de forma más importante. El elemento **<input>** se puede mostrar de varias maneras, dependiendo del atributo de tipo. Aquí hay unos ejemplos:

Tipo	Descripción
<code><input type = "text"></code>	Define un campo de entrada de texto de una línea.
<code><input type = "radio"></code>	Define un botón de radio (para seleccionar una de muchas opciones)
<code><input type = "submit"></code>	Define un botón de envío (para enviar el formulario)

Ilustración 36.

Entrada de texto. `<input type="text">` define un campo de entrada de una línea para la entrada de texto:

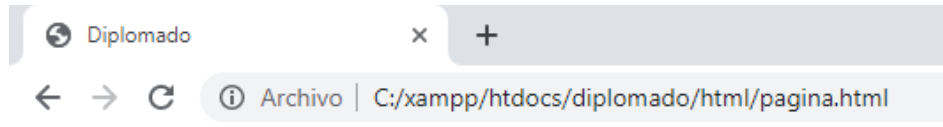


```

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ diplomado
  └─ html
    └─ pagina.html
      └─ math
      └─ operadores
      └─ poo
      └─ switchcase
      └─ tiposdatos
      └─ variables

pagina.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <h1>Formularios</h1>
8      <form>
9          Nombre:<br>
10         <input type="text" name="nombre">
11         <br>
12         Apellido:<br>
13         <input type="text" name="apellido">
14     </form>
15 </body>
16 </html>
  
```

Formularios

Nombre:

Apellido:

Ilustraciones 37 y 38.

Atributo type <input>

El atributo de la etiqueta **<input>** es sumamente funcional por todas las características técnicas en las que se puede usar en relación con los formularios. Esta se puede especificar para todo tipo de datos, desde solo números hasta una fecha, incluso importar imágenes; estos son algunos tipos:

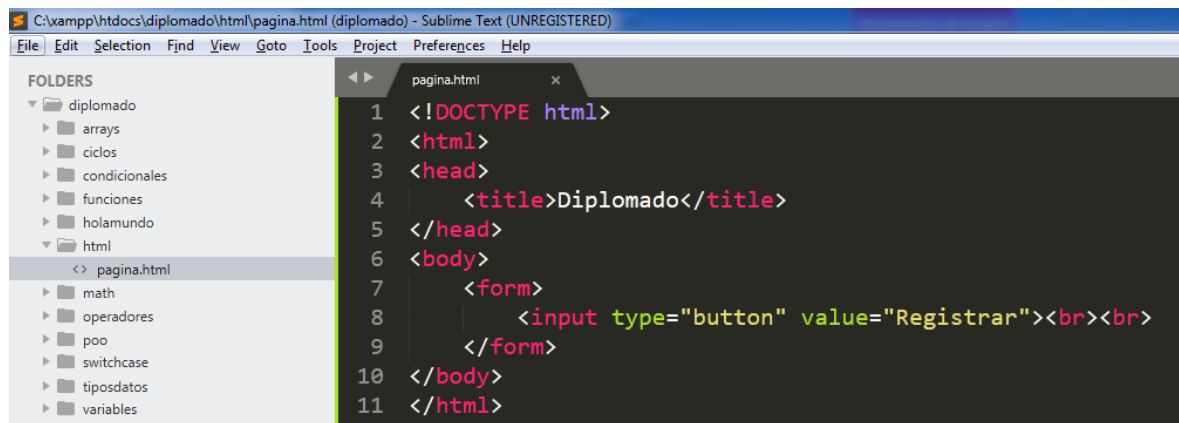
Tipo
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text">
<input type="time">

<code><input type="url"></code>
<code><input type="week"></code>

Tabla 1.

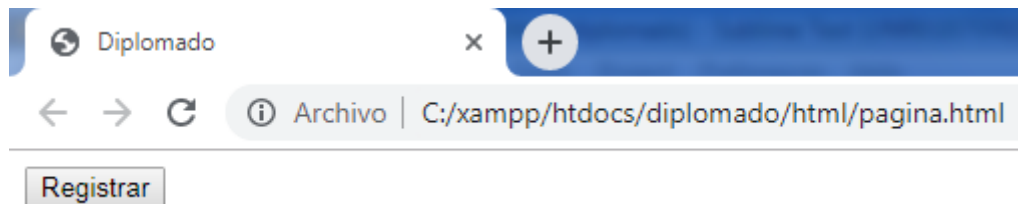
Ahora veremos algunos de los principales *types* en código y la forma en que operan en los formularios.

Type button. `<input type="button">` define un botón:



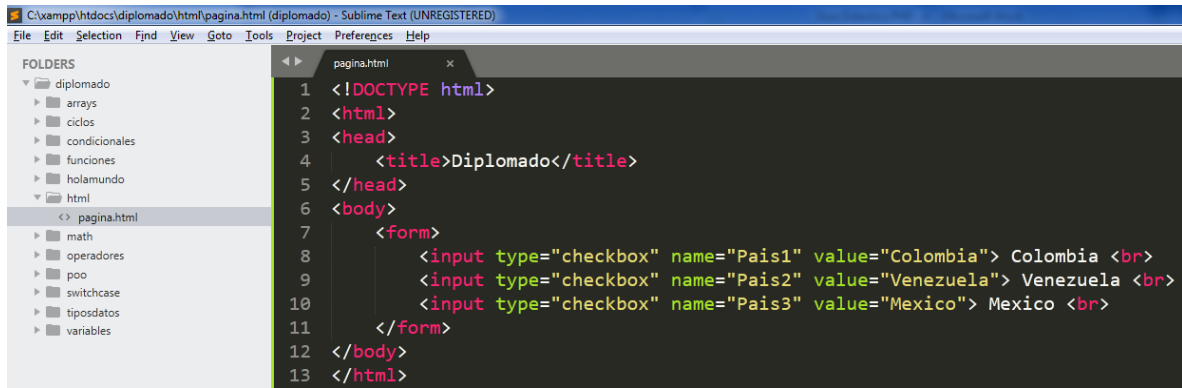
```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form>
8     <input type="button" value="Registrar"><br><br>
9   </form>
10 </body>
11 </html>
  
```



Ilustraciones 39 y 40.

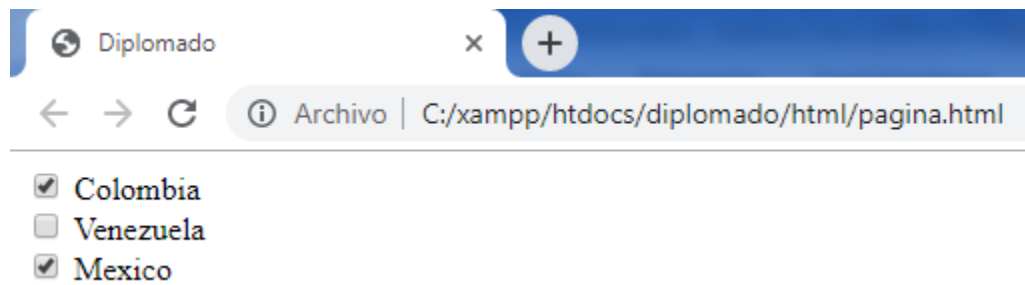
Type checkbox. `<input type="checkbox">` define una **casilla de verificación**. Las casillas de verificación le permiten al usuario seleccionar CERO o MÁS opciones de un número limitado de opciones.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form>
8     <input type="checkbox" name="Pais1" value="Colombia"> Colombia <br>
9     <input type="checkbox" name="Pais2" value="Venezuela"> Venezuela <br>
10    <input type="checkbox" name="Pais3" value="Mexico"> Mexico <br>
11  </form>
12 </body>
13 </html>

```



Ilustraciones 41 y 42.

Type color. `<input type="color">` utiliza para campos de entrada que deben contener un color.

Dependiendo de la compatibilidad del navegador puede aparecer un selector de color en el campo de entrada.

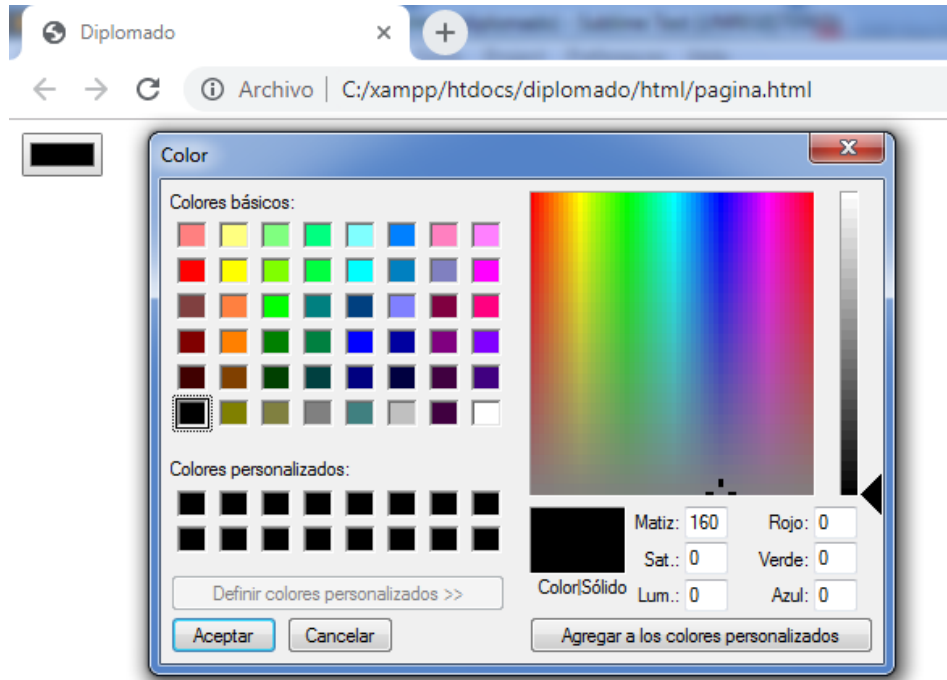


Ilustración 43.

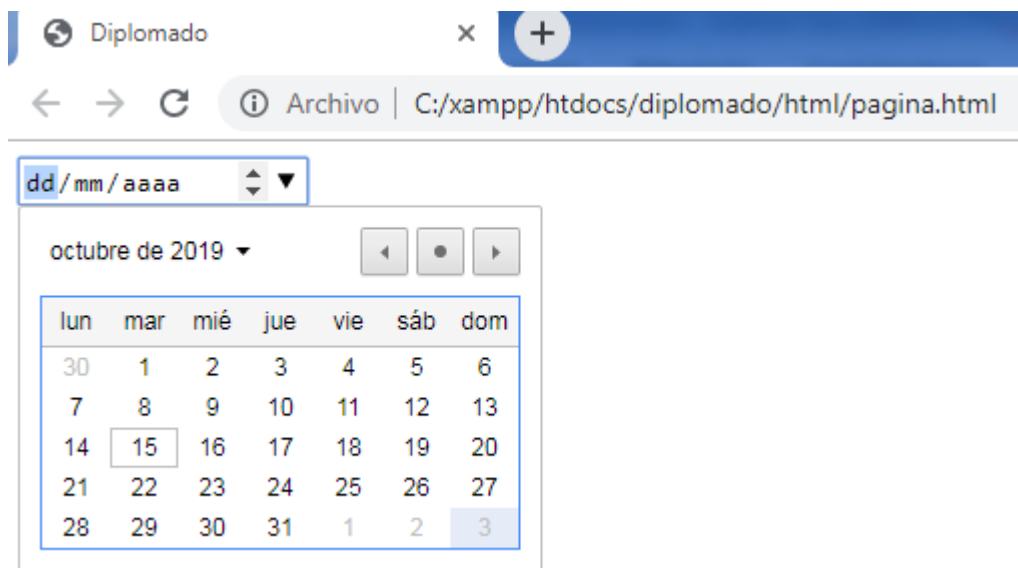
Type date. `<input type="date">` se utiliza para los campos de entrada que deben contener una fecha.

Dependiendo del soporte del navegador, un selector de fecha puede aparecer en el campo de entrada.



```

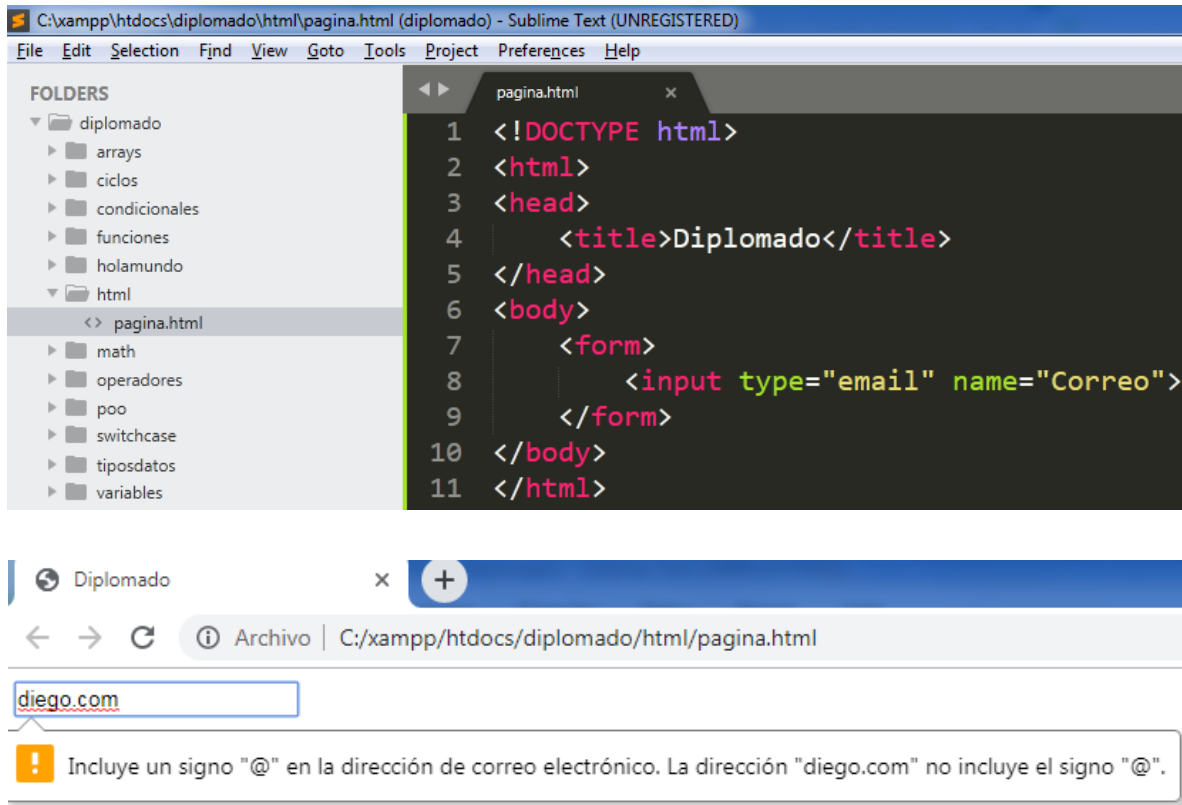
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form>
8     <input type="date" name="FechaNacimiento">
9   </form>
10 </body>
11 </html>
  
```



Ilustraciones 44 y 45.

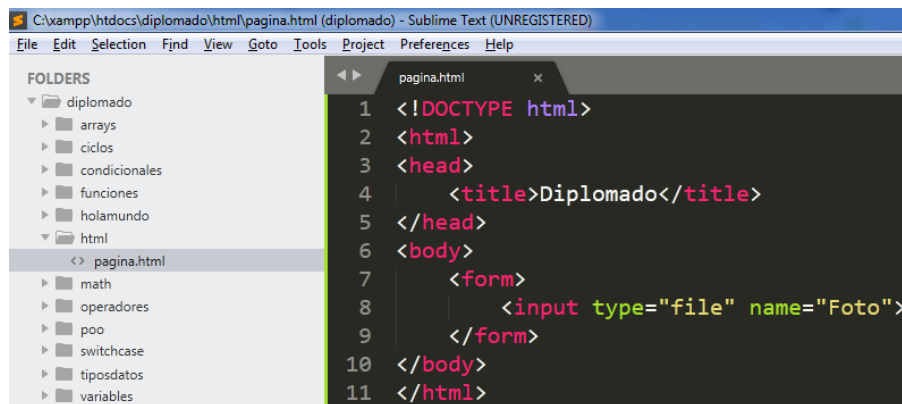
Type email. `<input type="email">` se utiliza para campos de entrada que deben contener una dirección de correo electrónico.

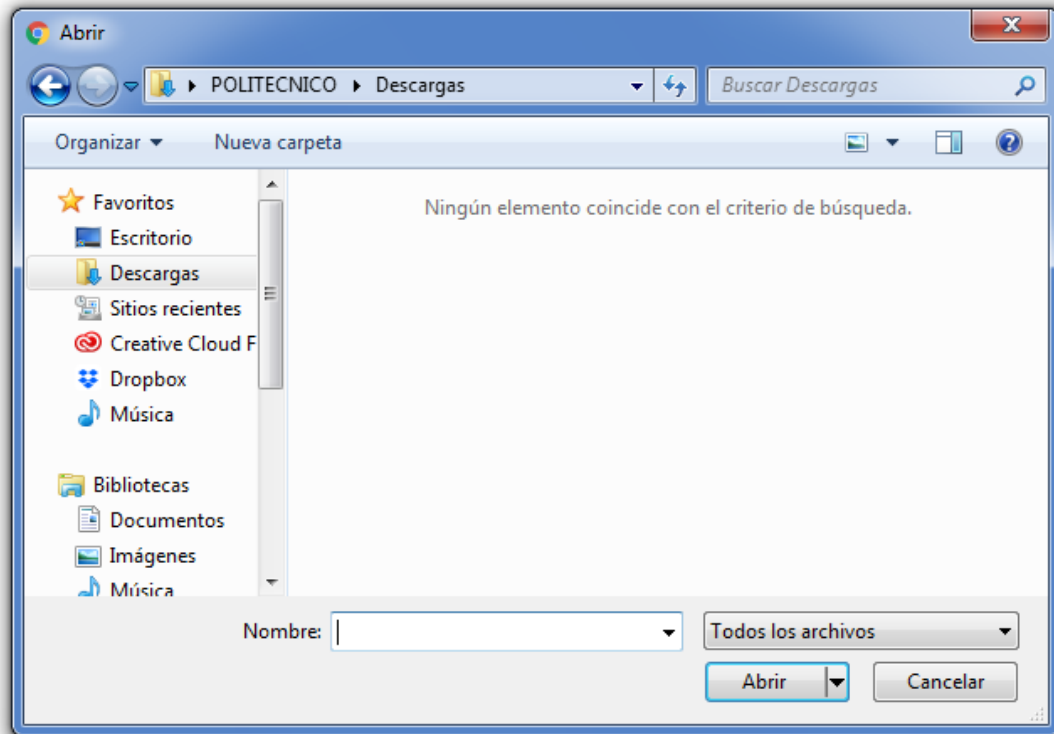
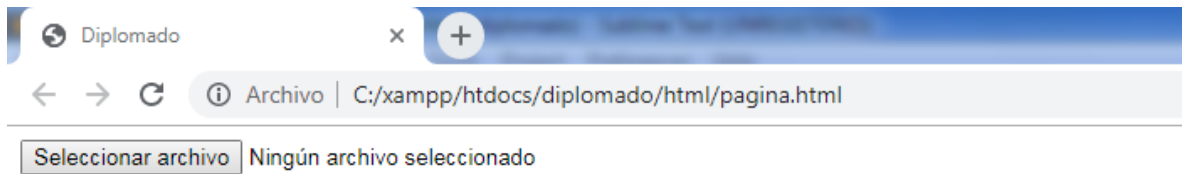
Dependiendo del soporte del navegador, la dirección de correo electrónico se puede validar automáticamente cuando se envía.



Ilustraciones 46 y 47.

Type file. `<input type="file">` define un campo de selección de archivo y un botón «Examinar» para la carga de archivos.





Ilustraciones 48 y 49.

Type number. `<input type="number">` define un campo de entrada numérico.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <form method="POST" action="guardar.php">
8          Edad <br>
9          <input type="number" name="Edad">
10         <br><br>
11         <input type="submit" value="Enviar">
12     </form>
13 </body>
14 </html>
  
```

Diplomado x +

← → ↻ ⓘ Archivo | C:/xampp/htdocs/diplomado/html/pagina.html

Edad

19

Enviar

Ilustraciones 50 y 51.

Type password. `<input type="password">` define un campo de contraseña, con la particularidad de que los caracteres se ven representados en asteriscos:

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form method="POST" action="guardar.php">
8     <input type="text" name="Usuario">
9     <input type="password" name="Password">
10    <input type="submit" value="Enviar">
11  </form>
12 </body>
13 </html>
```

Diplomado x +

← → ↻ ⓘ Archivo | C:/xampp/htdocs/diplomado/html/pagina.html

Ilustraciones 52 y 53.

Type submit. `<input type="submit">` define un botón para enviar los datos del formulario a un controlador de formulario.

El controlador de formulario suele ser una página de servidor con un *script* para procesar datos de entrada (en este caso PHP, pero más adelante veremos su uso).

El controlador de formulario se especifica en el atributo de acción del formulario:

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <h1>Formularios</h1>
8      <form action="guardar.php">
9          Nombre:<br>
10         <input type="text" name="nombre">
11         <br>
12         Genero:<br>
13         <input type="radio" name="genero"> Masculino <br>
14         <input type="radio" name="genero"> Femenino <br>
15         <br>
16         <input type="submit" value="Enviar">
17     </form>
18 </body>
19 </html>
  
```

Diplomado

← → ↻ Archivo | C:\xampp\htdocs\diplomado\html\pagina.html

Formularios

Nombre:

Genero:

☐ Masculino

☐ Femenino

Enviar

Ilustraciones 54 y 55.

Atributo action. Define la acción a realizar cuando se envía el formulario.

Normalmente, los datos del formulario se envían a una página web en el servidor cuando el usuario hace clic en el botón «Enviar».

En el ejemplo anterior, los datos del formulario se envían a una página en el servidor llamada "guardar.php". Esta página contiene un *script* del lado del servidor que maneja los datos del formulario:

```
<form action="guardar.php">
```

Ilustración 56.

Nota: si en action se omite el atributo, la acción se establece en la página actual.

Atributo method. Especifica el método HTTP (GET o POST) que se utilizará al enviar los datos del formulario:

Method GET:

```
<form action="guardar.php" method="GET">
```

Ilustración 57.

El método predeterminado al enviar datos del formulario es GET.

Sin embargo, cuando se utiliza GET los datos del formulario enviado serán visibles en el campo de dirección de la página:

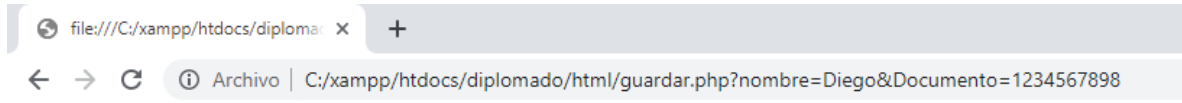


Ilustración 58.

Notas sobre GET:

Agrega datos de formulario a la URL en pares de nombre / valor.


La longitud de una URL es limitada (aproximadamente 3000 caracteres).



¡Nunca use GET para enviar datos confidenciales! (será visible en la URL).

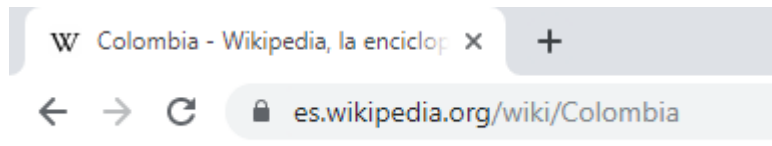
Útil para envíos de formularios donde un usuario desea marcar el resultado.

GET es mejor para datos no seguros, como cadenas de consulta en Wikipedia.

No has accedido [Discusión](#) [Contribuciones](#) [Crear una cuenta](#) [Acceder](#)

Leer [Ver código](#) [Ver historial](#) 

Coordenadas:  4°35'56"N 74°04'51"O ([mapa](#)) 



Ilustraciones 59 y 60.

Method POST:

```
<form action="guardar.php" method="POST">
```

Ilustración 61.

Utilice siempre POST si los datos del formulario contienen información confidencial o personal. El método POST no muestra los datos del formulario enviado en el campo de dirección de página.

Notas sobre POST:

POST no tiene limitaciones de tamaño y puede usarse para enviar grandes cantidades de datos.

Los envíos de formularios con POST no pueden marcarse.

Atributo name. Cada campo de entrada debe tener un atributo name para ser enviado.

Si se omite el atributo, los datos de ese campo de entrada no se enviarán en absoluto.

Este ejemplo solo enviará el campo de entrada "Apellido". Dado que «nombre» no cuenta con el atributo name.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

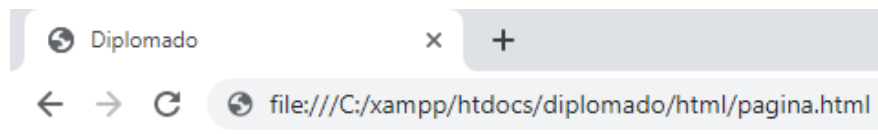
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

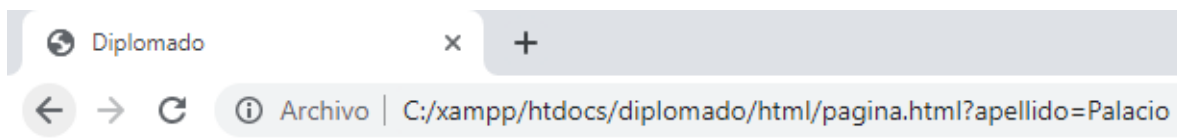
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <h1>Formularios</h1>
8   <form action="pagina.html" method="GET">
9     Nombre:<br>
10    <input type="text"><br>
11    Apellido:<br>
12    <input type="text" name="apellido">
13    <br><br>
14    <input type="submit" value="Enviar">
15  </form>
16 </body>
17 </html>
  
```



Formularios

Nombre:

Apellido:



Ilustraciones 62, 63 y 64.

Entrada de selección de lista desplegable

El `<select>` `</select>` elemento define una lista desplegable:

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

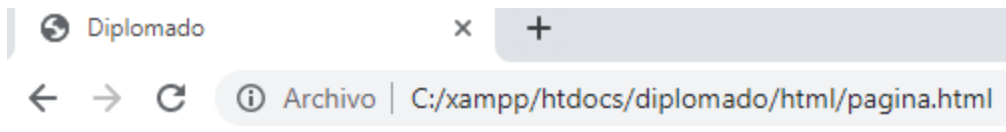
- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <h1>Formularios</h1>
8      <form>
9          <select name="Carros">
10             <option value="Volvo">Volvo</option>
11             <option value="Chevrolet">Chevrolet</option>
12             <option value="Fiat">Fiat</option>
13             <option value="Kia">Kia</option>
14          </select>
15      </form>
16 </body>
17 </html>

```

Ilustración 65.



Formularios

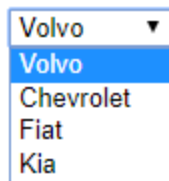


Ilustración 66.

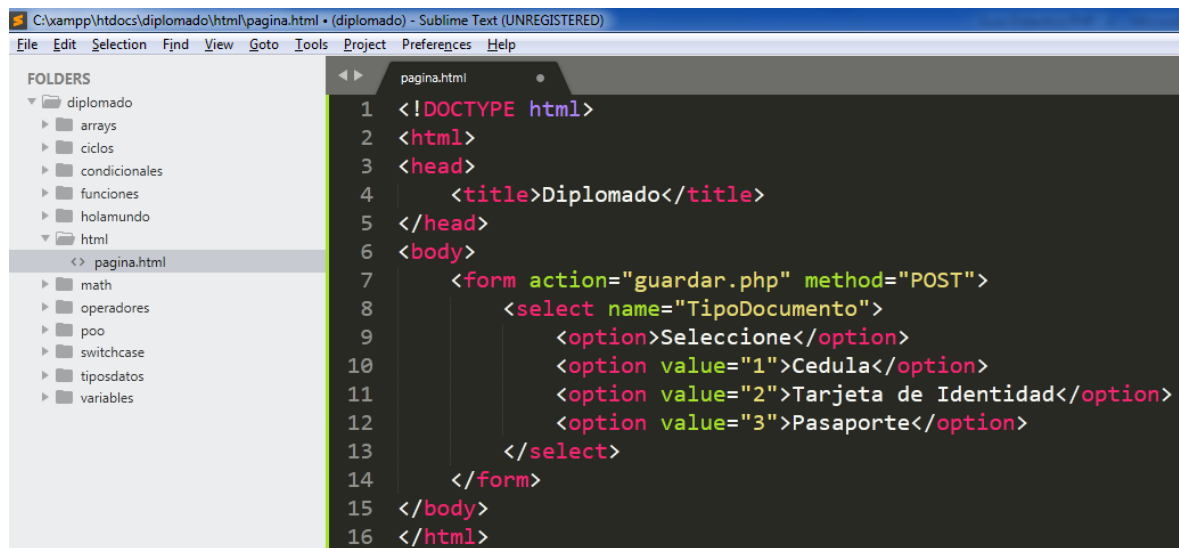
Dentro de la etiqueta **<option>** **</option>** van los valores a seleccionar dentro de la lista desplegable. En ese orden de ideas, el atributo **value** contiene el mismo valor u otro que se desea almacenar.

Por ejemplo, representemos la siguiente información en una lista desplegable:

Código	Valor
1	Cédula
2	Tarjeta de identidad
3	Pasaporte

Tabla 2.

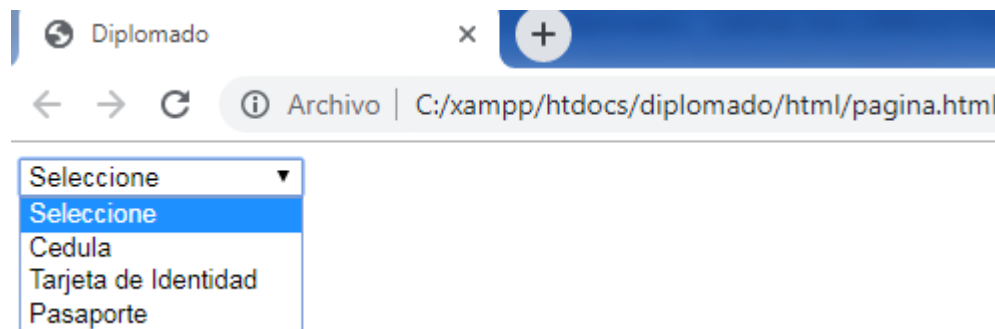
El código estará contenido en el atributo **value** y el valor en la etiqueta **<option></option>**, de la siguiente forma:



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form action="guardar.php" method="POST">
8     <select name="TipoDocumento">
9       <option>Seleccione</option>
10      <option value="1">Cedula</option>
11      <option value="2">Tarjeta de Identidad</option>
12      <option value="3">Pasaporte</option>
13    </select>
14  </form>
15 </body>
16 </html>

```



Ilustraciones 67 y 68.

Textarea. **<textarea>** define un campo de entrada de varias líneas (un área de texto):

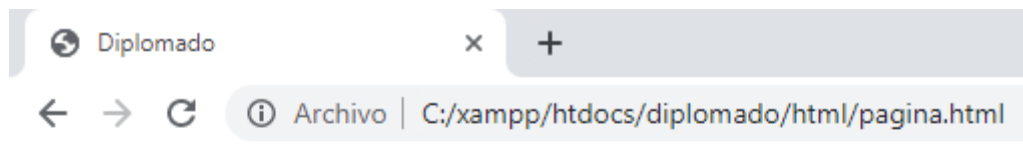
C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

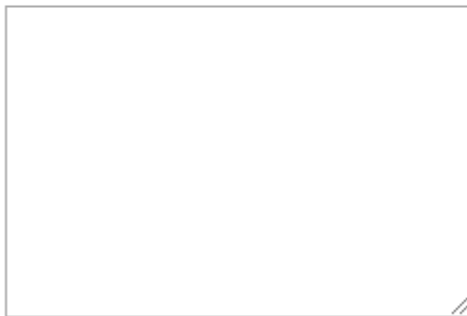
FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <h1>Formularios</h1>
8   <form>
9     <textarea name="Presentacion" rows="10" cols="30"></textarea>
10  </form>
11 </body>
12 </html>
```



Formularios



Ilustraciones 69 y 70.

Rows. Permite definir el número de filas del campo de texto.

Cols. Permite definir el número de columnas del campo de texto.

Las etiquetas de entradas cuentan con algunos atributos y restricciones muy especiales. Aquí la lista de algunas y su uso.

Atributo	Descripción
checked	Especifica que se debe preseleccionar un campo de entrada cuando se carga la página (para type = "checkbox" o type = "radio")
disabled	Especifica que un campo de entrada debe estar deshabilitado
max	Especifica el valor máximo para un campo de entrada
maxlength	Especifica el número máximo de caracteres para un campo de entrada
min	Especifica el valor mínimo para un campo de entrada
pattern	Especifica una expresión regular para verificar el valor de entrada
readonly	Especifica que un campo de entrada es de solo lectura (no se puede cambiar)
required	Especifica que se requiere un campo de entrada (debe completarse)
size	Especifica el ancho (en caracteres) de un campo de entrada
step	Especifica los intervalos de números legales para un campo de entrada
value	Especifica el valor predeterminado para un campo de entrada

Ilustración 71.

Checked. Preselecciona un *checkbox* automáticamente desde que la página es abierta.

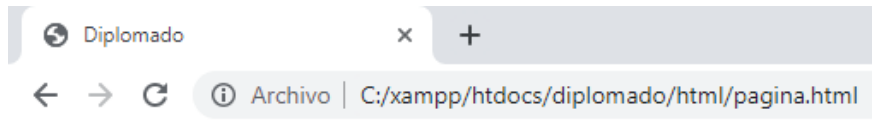


```

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ diplomado
  └─ arrays
  └─ ciclos
  └─ condicionales
  └─ funciones
  └─ holamundo
  └─ html
    └─ <─ pagina.html
      └─ math
      └─ operadores
      └─ poo
      └─ switchcase
      └─ tiposdatos
      └─ variables

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <h1>Formularios</h1>
8   <form>
9     <input type="checkbox" value="Colombia" name="Colombia" checked=""> Colombia <br><br>
10    <input type="checkbox" value="Argentina" name="Argentina"> Argentina <br><br>
11    <input type="submit" value="Enviar">
12  </form>
13 </body>
14 </html>
  
```



Formularios

☒ Colombia

☐ Argentina

Enviar

Ilustraciones 72 y 73.

Disable. Deshabilita uno o varios campos de entrada.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

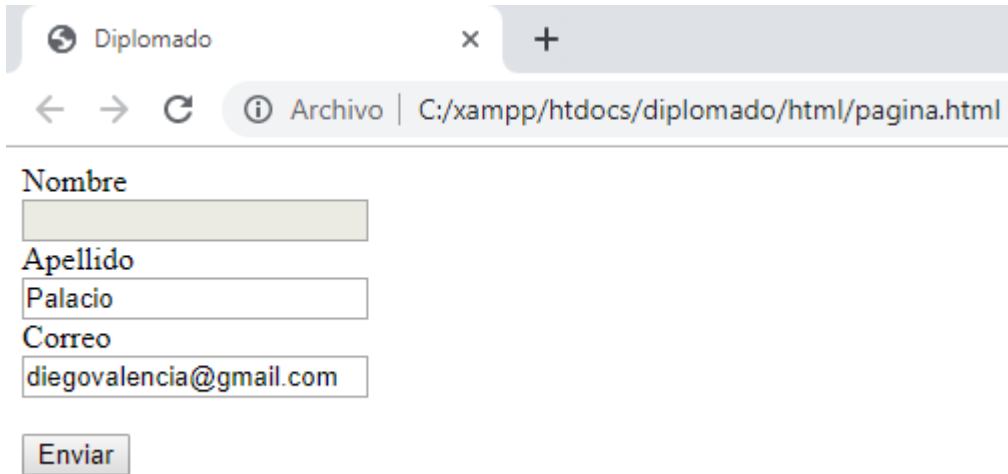
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <form method="POST" action="guardar.php">
8          Nombre <br>
9          <input type="text" name="Nombre" disabled=""><br>
10         Apellido <br>
11         <input type="text" name="Apellido"><br>
12         Correo <br>
13         <input type="email" name="Correo">
14         <br><br>
15         <input type="submit" value="Enviar">
16     </form>
17 </body>
18 </html>
  
```



Diplomado

← → ↻ ⓘ Archivo | C:/xampp/htdocs/diplomado/html/pagina.html

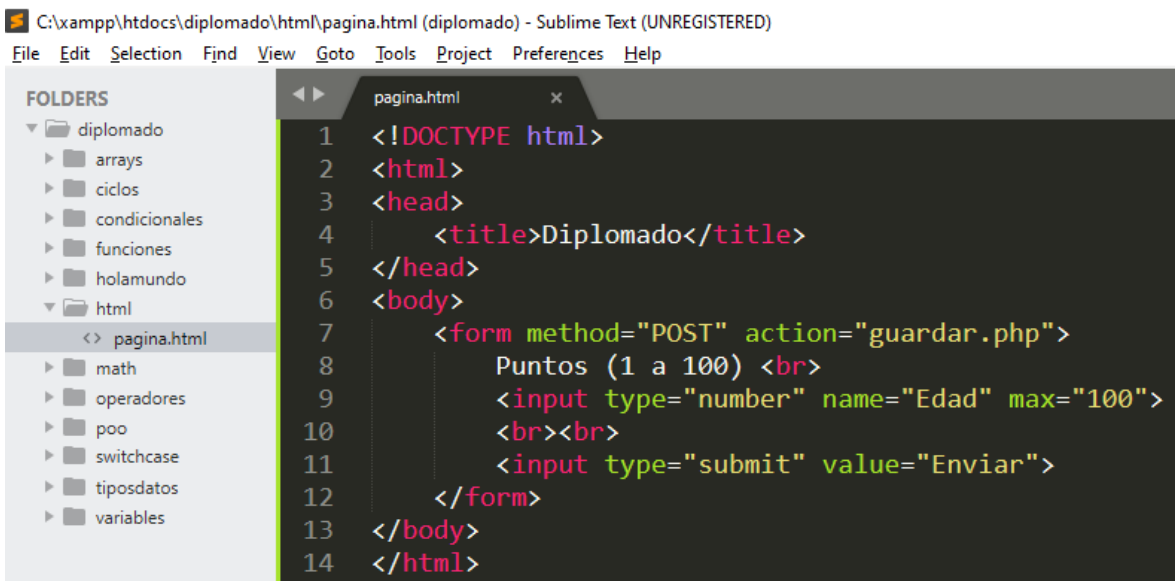
Nombre

Apellido

Correo

Ilustraciones 74 y 75.

Max. Especifica el valor máximo para un campo de entrada.



C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

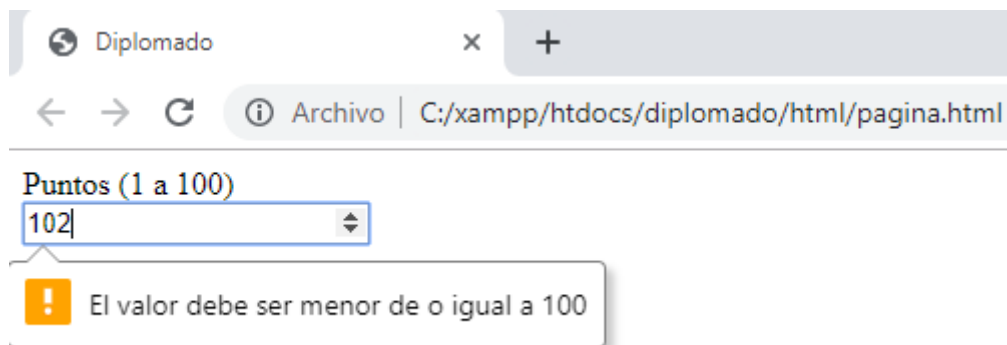
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```


1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form method="POST" action="guardar.php">
8     Puntos (1 a 100) <br>
9     <input type="number" name="Edad" max="100">
10    <br><br>
11    <input type="submit" value="Enviar">
12  </form>
13 </body>
14 </html>
  
```



Diplomado

← → ↻ ⓘ Archivo | C:/xampp/htdocs/diplomado/html/pagina.html

Puntos (1 a 100)

 El valor debe ser menor de o igual a 100

Ilustraciones 76 y 77.

Min. Especifica el valor mínimo para un campo de entrada.

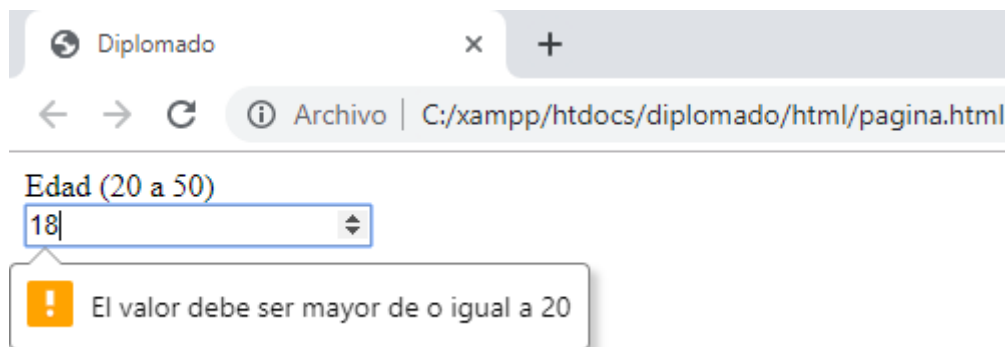
C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form method="POST" action="guardar.php">
8     Edad (20 a 50) <br>
9     <input type="number" name="Edad" min="20">
10    <br><br>
11    <input type="submit" value="Enviar">
12  </form>
13 </body>
14 </html>
```



Ilustraciones 78 y 79

Maxlength. Especifica el número máximo de caracteres de un campo.

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <form method="POST" action="guardar.php">
8          Pais <br>
9          <input type="text" name="Pais" maxlength="5">
10         <br><br>
11         <input type="submit" value="Enviar">
12     </form>
13 </body>
14 </html>
  
```

C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Diplomado</title>
5  </head>
6  <body>
7      <form method="POST" action="guardar.php">
8          Pais <br>
9          <input type="text" name="Pais" required="">
10         <br><br>
11         <input type="submit" value="Enviar">
12     </form>
13 </body>
14 </html>
  
```

Ilustraciones 80 y 81.

Required. Especifica que se debe completar el campo de entrada.

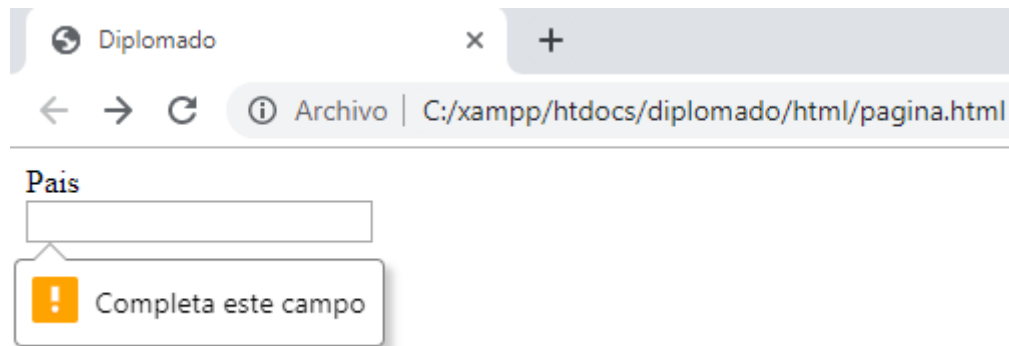
C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - pagina.html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <form method="POST" action="guardar.php">
8     Pais <br>
9     <input type="text" name="Pais" required="">
10    <br><br>
11    <input type="submit" value="Enviar">
12  </form>
13 </body>
14 </html>
```



Ilustraciones 82 y 83.

Tema 3: HTML - PHP

PHP es claramente un lenguaje orientado al 100 % al desarrollo web, por lo cual se lleva de la mano de gran forma con el lenguaje de marcado HTML.

HTML y PHP se permiten integrar mutuamente en documentos donde el código de estos lenguajes es completamente legible el uno con el otro, esta es la ventaja principal de esta relación: la posibilidad de combinar código sin excepción alguna. Un ejemplo de esto podría ser el siguiente.

Generar números del 1 a 5 por medio de un ciclo y mostrar estos en encabezados `<h3></h3>`.

Para esto, si vienes trabajando con el archivo `pagina.html`, habrá que hacer un pequeño cambio: cuando estamos trabajando con PHP y HTML en un mismo documento, la extensión que debe definir el archivo es PHP, el único caso en que se puede omitir esto es cuando en un archivo `.html` se tenga simplemente un formulario que envía los datos a un `.php`, más adelante veremos los ejemplos.

Lo ideal será continuar, de ahora en adelante, con todos los documentos en formato `.php`. Por lo que debes crear ahora un `index.php`.

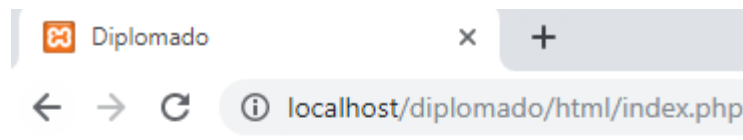
C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - index.php
 - html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

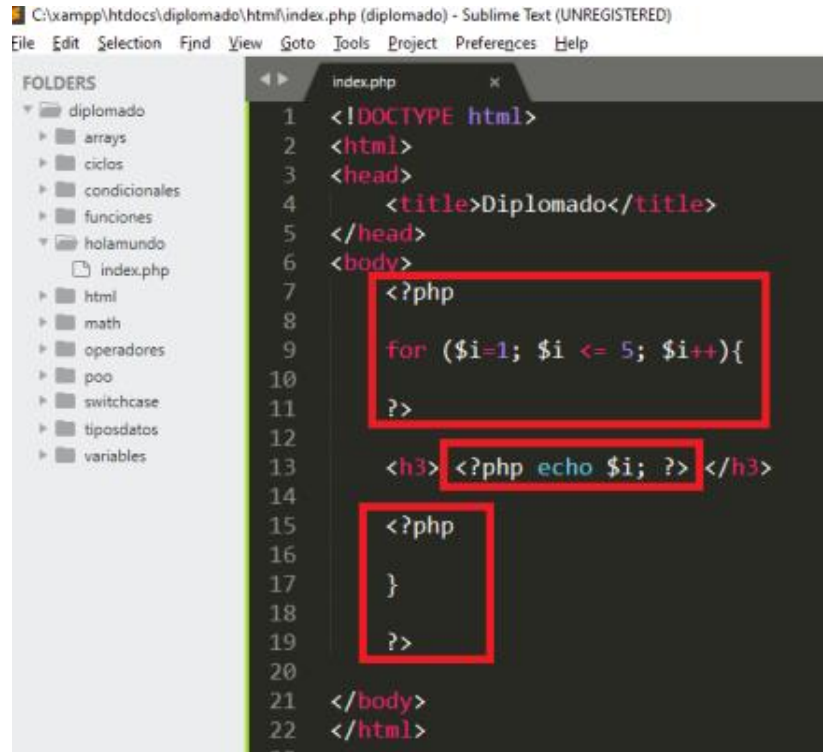
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Diplomado</title>
5 </head>
6 <body>
7     <?php
8         for ($i=1; $i <= 5; $i++){
9             ?>
10            <h3><?php echo $i; ?></h3>
11            <?php
12                }
13            ?>
14 </body>
15 </html>
```



Ilustraciones 84 y 85.

Este es el primer acercamiento entre PHP y HTML. El contenido y etiquetas de HTML se ve perfectamente e igual en nuestro index.php. Hay que detallar cómo funciona PHP y cómo se integra en el HTML.

Hasta el momento, en los documentos y ejercicios realizados en PHP solo contábamos con una etiqueta de apertura y otra de cierre (`<?php ¿>`), pero tener múltiples es válido y, más aún, necesario cuando se trabaja con HTML, dado que se está incrustando código en varias sesiones del documento. Veamos:



```
C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ diplomado
  └─ arrays
  └─ ciclos
  └─ condicionales
  └─ funciones
  └─ holamundo
    └─ index.php
  └─ html
  └─ math
  └─ operadores
  └─ poo
  └─ switchcase
  └─ tiposdatos
  └─ variables

index.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <?php
8     for ($i=1; $i <= 5; $i++){
9
10    }
11  <?>
12
13  <h3> <?php echo $i; ?> </h3>
14
15  <?php
16    }
17  <?>
18
19
20
21 </body>
22 </html>
```

Ilustración 86.

En el ejemplo anterior hay tres casos en los que incrustamos código PHP en HTML.

- Declaración del ciclo *for*.
- Impresión encabezado.
- Cierre del ciclo *for*.

Esta es la forma en la que debe operar PHP frente HTML, por ejemplo, en caso de una sola declaración de etiquetas PHP, y volcando todo el código dentro de estas, el resultado sería el siguiente:

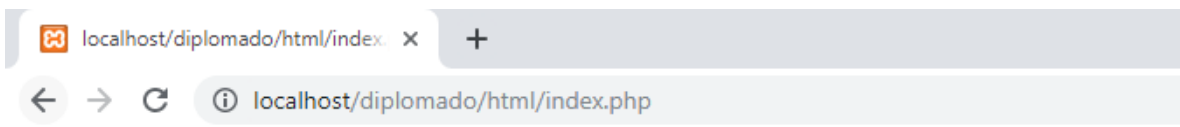
C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - index.php
 - html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Diplomado</title>
5 </head>
6 <body>
7     <?php
8
9     for ($i=1; $i <= 5; $i++){
10         <h3> echo $i; </h3>
11     }
12
13     ?>
14
15 </body>
16 </html>
```

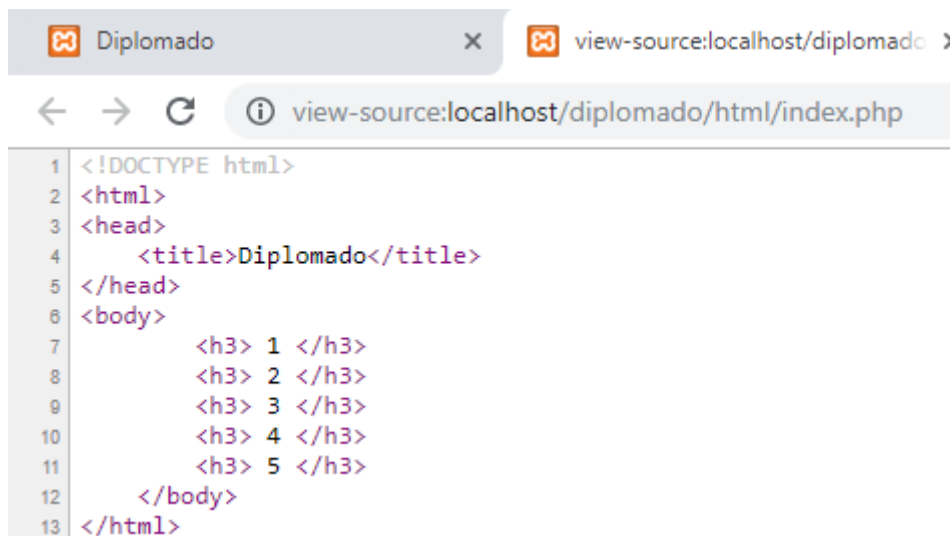
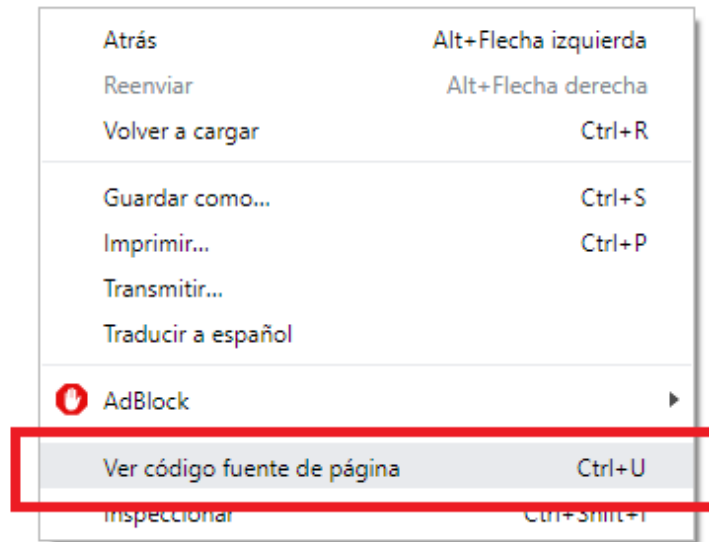


Parse error: syntax error, unexpected '<' in C:\xampp\htdocs\diplomado\html\index.php on line 10

Ilustraciones 87 y 88.

Error claramente, por eso es importante reconocer en qué lugares incrustar PHP, para eso es importante tener muy claro las etiquetas de HTML y cómo opera PHP.

Para ver en el navegador el resultado de este ejercicio (codificado de forma correcta), simplemente se debe ir al navegador y por medio de clic derecho seleccionar «Ver código fuente de página».

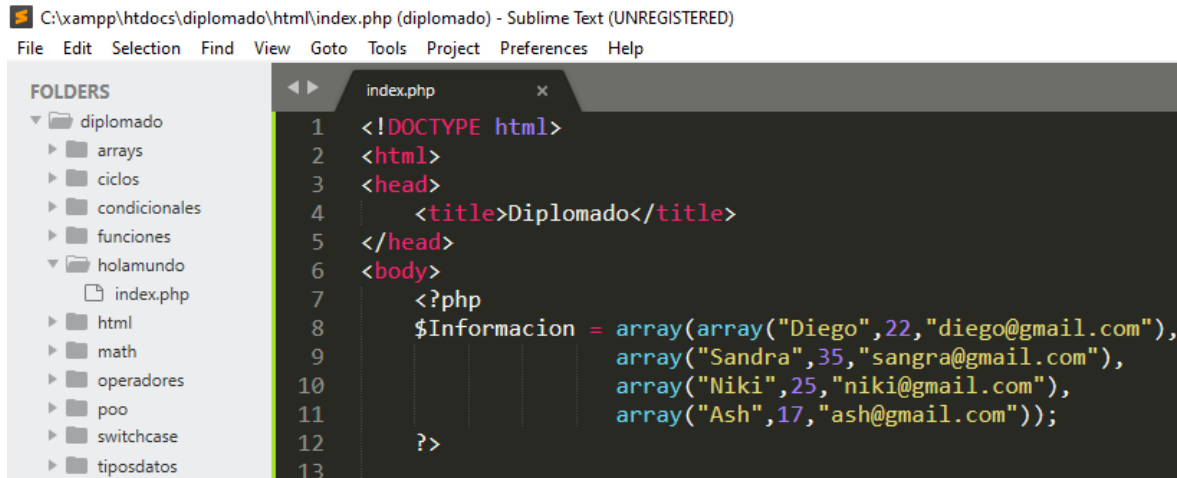


Ilustraciones 89 y 90.

En el navegador vemos claramente el resultado en código HTML, porque esta es otra cualidad de trabajar con PHP y HTML, que, en el navegador, únicamente se visualiza el código HTML y oculta por completo cualquier pieza de código de PHP, lo cual lo hace muy seguro al ocultar el lenguaje de programación con el que se está trabajando.

Veamos más ejemplos de código HTML en relación con PHP.

En este caso, generaremos una tabla a partir de un *array* con la siguiente estructura.



```

C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ diplomado
  └─ arrays
  └─ ciclos
  └─ condicionales
  └─ funciones
  └─ holamundo
    └─ index.php
  └─ html
  └─ math
  └─ operadores
  └─ poo
  └─ switchcase
  └─ tiposdatos

index.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <?php
8     $Informacion = array(array("Diego",22,"diego@gmail.com"),
9                           array("Sandra",35,"sangra@gmail.com"),
10                          array("Niki",25,"niki@gmail.com"),
11                          array("Ash",17,"ash@gmail.com"));
12   ?>
13
  
```

Ilustración 91.

En este primer bloque del archivo vamos a declarar el arreglo con la siguiente información:

Nombre	Edad	Correo
Diego	22	diego@gmail.com
Sandra	35	sandra@gmail.com
Niki	25	niki@gmail.com
Ash	17	ash@gmail.com

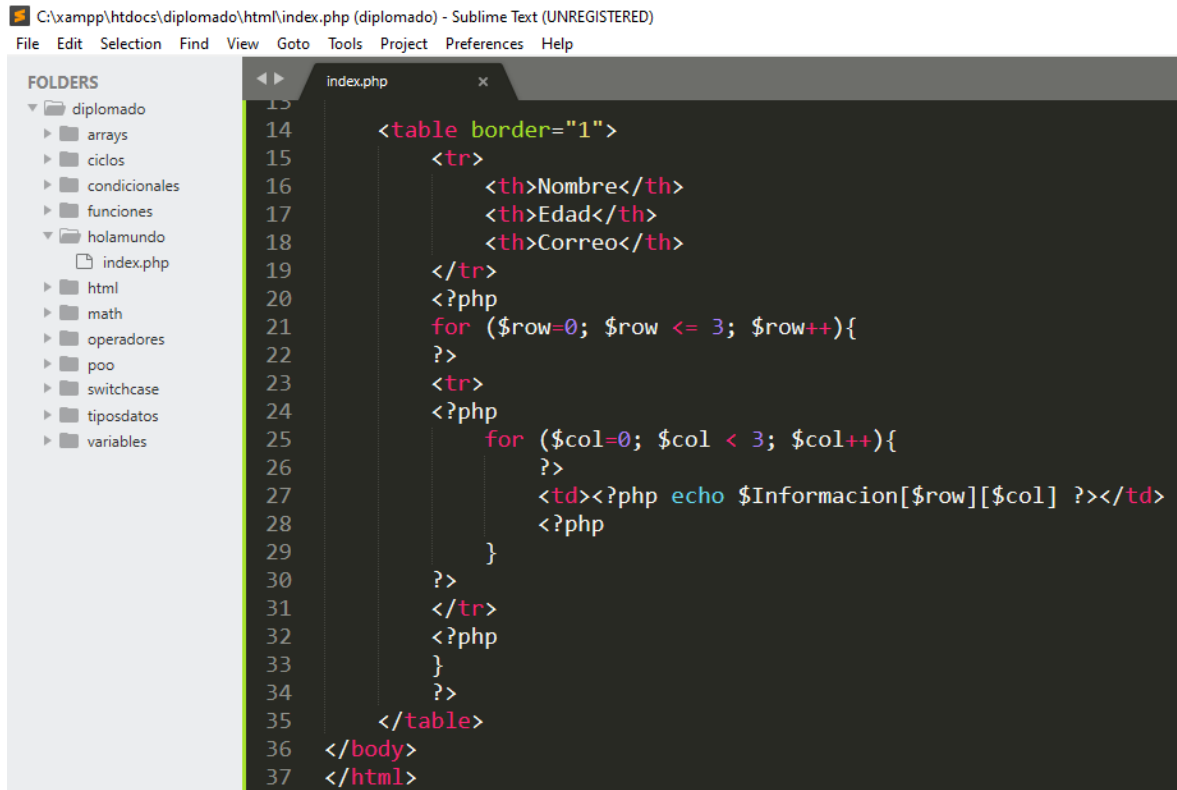
Tabla 3.

Es un arreglo multidimensional con cuatro *arrays* internos de datos. Deseamos mostrar toda la información en una tabla, recuerda que ya aprendimos como declarar una tabla y para recorrer el *array* se ocupan dos ciclos *for*: uno para filas y otro para columnas.

Recuerda que la etiqueta `<tr>` se encarga de las filas y la etiqueta `<td>` de las columnas, es decir, hay un ciclo para cada etiqueta.

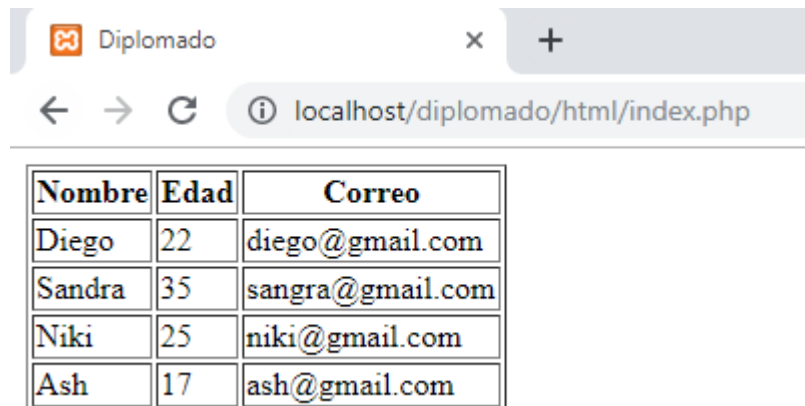
La forma de acceder al *array* es por medio de dos posiciones, fila y columna, de la siguiente forma: `$datos[0][2]`. En este caso retornaría el tercer dato de la primera fila (recuerda que los índices empiezan en 0).

El resultado sería siguiente:



```

13
14 <table border="1">
15     <tr>
16         <th>Nombre</th>
17         <th>Edad</th>
18         <th>Correo</th>
19     </tr>
20     <?php
21     for ($row=0; $row <= 3; $row++){
22     ?>
23     <tr>
24     <?php
25         for ($col=0; $col < 3; $col++){
26         ?>
27         <td><?php echo $Informacion[$row][$col] ?></td>
28         <?php
29         }
30     ?>
31     </tr>
32     <?php
33     }
34     ?>
35 </table>
36 </body>
37 </html>
    
```

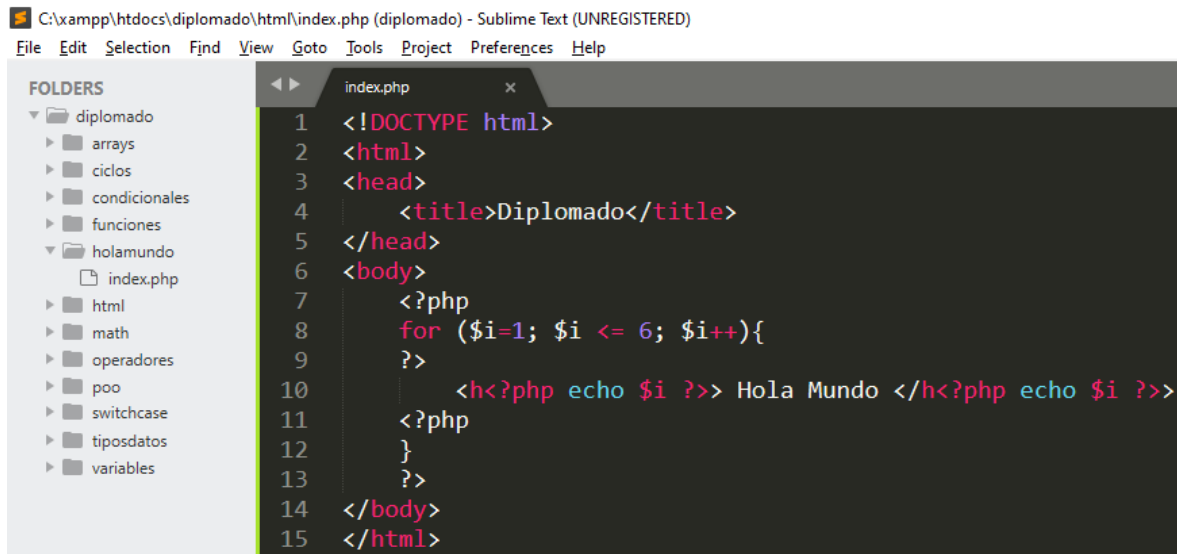


Nombre	Edad	Correo
Diego	22	diego@gmail.com
Sandra	35	sangra@gmail.com
Niki	25	niki@gmail.com
Ash	17	ash@gmail.com

Ilustraciones 92 y 93.

Con cada recorrido del primer ciclo de filas creamos todos los `<tr>` para contener los `<td>` de las columnas, esto debe quedar muy claro. Lo más difícil puede ser entender cómo integrar PHP en HTML, pero con práctica será muy fácil.

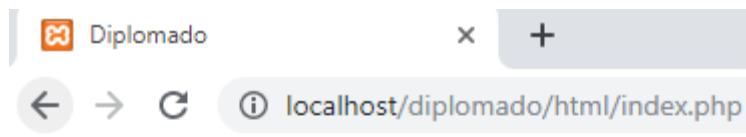
El código PHP puede ir incrustado directamente de las declaraciones de las etiquetas, por ejemplo, en los encabezados. `<h1>`, el uno podría remplazarse por una variable de PHP, como por ejemplo con un ciclo, e imprimir todos los encabezados, veamos:



```
C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ diplomado
  └─ arrays
  └─ ciclos
  └─ condicionales
  └─ funciones
  └─ holamundo
    └─ index.php
  └─ html
  └─ math
  └─ operadores
  └─ poo
  └─ switchcase
  └─ tiposdatos
  └─ variables

index.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <?php
8     for ($i=1; $i <= 6; $i++){
9       ?>
10      <h<?php echo $i ?>> Hola Mundo </h<?php echo $i ?>>
11      <?php
12      }
13      ?>
14 </body>
15 </html>
```



Hola Mundo

Hola Mundo

Hola Mundo

Hola Mundo

Hola Mundo

Hola Mundo

Ilustraciones 94 y 95.

Otro ejemplo puede ser una operación aritmética dentro de HTML e imprimiendo el resultado en un encabezado.

C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)

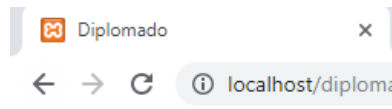
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - index.php
 - html
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <?php
8     $numero1 = 5;
9     $numero2 = 7;
10    $resultado = $numero1 * $numero2;
11    ?>
12    <h3> El resultado es: <?php echo $resultado; ?> </h3>
13  </body>
14 </html>
  
```



El resultado es: 35

Ilustraciones 96 y 97.

Pasemos ahora a conceptos mucho más increíbles en el desarrollo con PHP y HTML. Para el envío de datos por medio de formularios, vamos a usar los métodos `_GET` y `_POST`.

Para realizar el primer ejercicio, supongamos que debemos realizar una calculadora con dos números y un operador. Los dos números deben ser campos de entrada de número, es decir, `type="number"` y con la restricción de requeridos, además, el operador será una lista desplegable con los cuatro (+, -, *, /). Los datos deberán ser enviados a un archivo en el folder «html» con nombre: «calculadora.php» y mostrar el resultado en este.

Primero utilizaremos el método `POST`:

C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

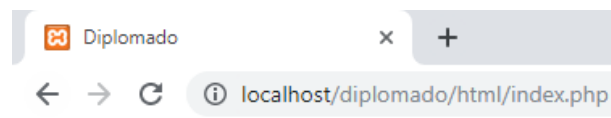
FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - calculadora.php
 - index.php
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <h1>Calculadora</h1>
8   <form method="POST" action="calculadora.php">
9     Numero 1 <br>
10    <input type="number" name="Numero1" required=""><br><br>
11    Operador <br>
12    <select name="Operator" required="">
13      <option disabled="">Selecione</option>
14      <option value="+>+</option>
15      <option value="->-</option>
16      <option value="/">/</option>
17      <option value="*>*</option>
18    </select><br><br>
19    Numero 2 <br>
20    <input type="number" name="Numero2" required=""><br><br>
21    <input type="submit" value="Calcular">
22  </form>
23 </body>
24 </html>

```



Calculadora

Numero 1
9

Operador
/

Numero 2
3

Calcular

C:\xampp\htdocs\diplomado\html\calculadora.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

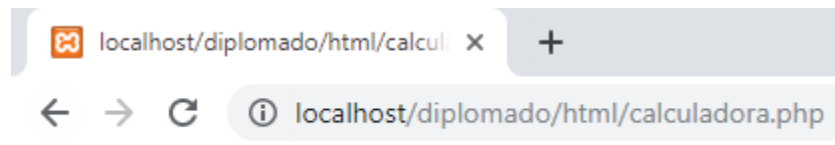
FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - calculadora.php
 - index.php
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <?php
2
3  $Numero1 = $_POST['Numero1'];
4  $Numero2 = $_POST['Numero2'];
5  $Operador = $_POST['Operador'];
6
7  switch ($Operador) {
8      case '+':
9          $Resultado = $Numero1 + $Numero2;
10         break;
11     case '-':
12         $Resultado = $Numero1 - $Numero2;
13         break;
14     case '/':
15         $Resultado = $Numero1 / $Numero2;
16         break;
17     case '*':
18         $Resultado = $Numero1 * $Numero2;
19         break;
20     default:
21         $Resultado = "Error";
22         break;
23 }
24
25 ?>
26
27 <h1>El resultado es: <?php echo $Resultado; ?></h1>
28 <a href="index.php">Regresar a la calculadora</a>
29

```



El resultado es: 3

[Regresar a la calculadora](#)

Ilustraciones 98, 99, 100 y 101.

Con pocas líneas de código construimos una pequeña calculadora que funciona de la siguiente forma:

El formulario de captura de datos cuenta con tres campos:

- Campo número 1: campo de type “number” para recibir únicamente números, el *name* es “Numero1” para recibir dos datos después, además es requerido.

```
<input type="number" name="Numero1" required=""><br><br>
```

- Campo número 2: campo de type “number” para recibir únicamente números, el *name* es “Numero2” para recibir dos datos después, además es requerido.

```
<input type="number" name="Numero2" required=""><br><br>
```

- Lista desplegable: lista desplegable con los operadores (+, -, *, /), el *name* es “Operador” para recibir los datos después y además es requerido.

```
<select name="Operador" required="">  
  <option disabled="">Selecione</option>  
  <option value="+">+</option>  
  <option value="-">-</option>  
  <option value="/">/</option>  
  <option value="*">*</option>  
</select><br><br>
```

- La etiqueta <form> deberá contener dos atributos muy importantes, el método y el action.

```
<form method="POST" action="calculadora.php">
```

Ilustraciones 102, 103, 104 y 105.

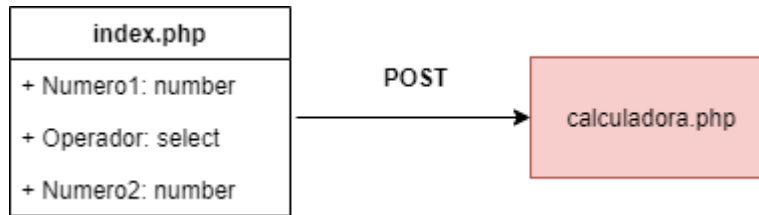
Recordemos que el método será la forma de enviar la información, que puede ser vía POST o GET.

- **POST**: envía los datos ocultos y seguros.
- **GET**: manda los datos visibles.

POST se suele usar principalmente para los formularios, en la mayoría de los casos es la mejor opción para los formularios por mantener la información oculta y segura, evitando la sensibilidad de la información enviada. Más adelante en el diplomado verás el uso y aplicación de cada uno de mejor forma.

GET se suele usar principalmente para enviar datos por la URL, es usado puntualmente para casos muy específicos de envío de información, por ejemplo, en buscadores, etc.

Action, por otra parte, especifica el lugar donde deseamos enviar la información recolectada en los campos del formulario, el lugar en la mayoría de los casos es un archivo .php y se debe especificar su ruta.



C:\xampp\htdocs\diplomado\html\calculadora.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - calculadora.php
 - index.php
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <?php
2
3  $Numero1 = $_POST['Numero1'];
4  $Numero2 = $_POST['Numero2'];
5  $Operador = $_POST['Operador'];
6
7  switch ($Operador) {
8      case '+':
9          $Resultado = $Numero1 + $Numero2;
10         break;
11     case '-':
12         $Resultado = $Numero1 - $Numero2;
13         break;
14     case '/':
15         $Resultado = $Numero1 / $Numero2;
16         break;
17     case '*':
18         $Resultado = $Numero1 * $Numero2;
19         break;
20     default:
21         $Resultado = "Error";
22         break;
23 }
24
25 ?>
26
27 <h1>El resultado es: <?php echo $Resultado; ?></h1>
28 <a href="index.php">Regresar a la calculadora</a>
  
```

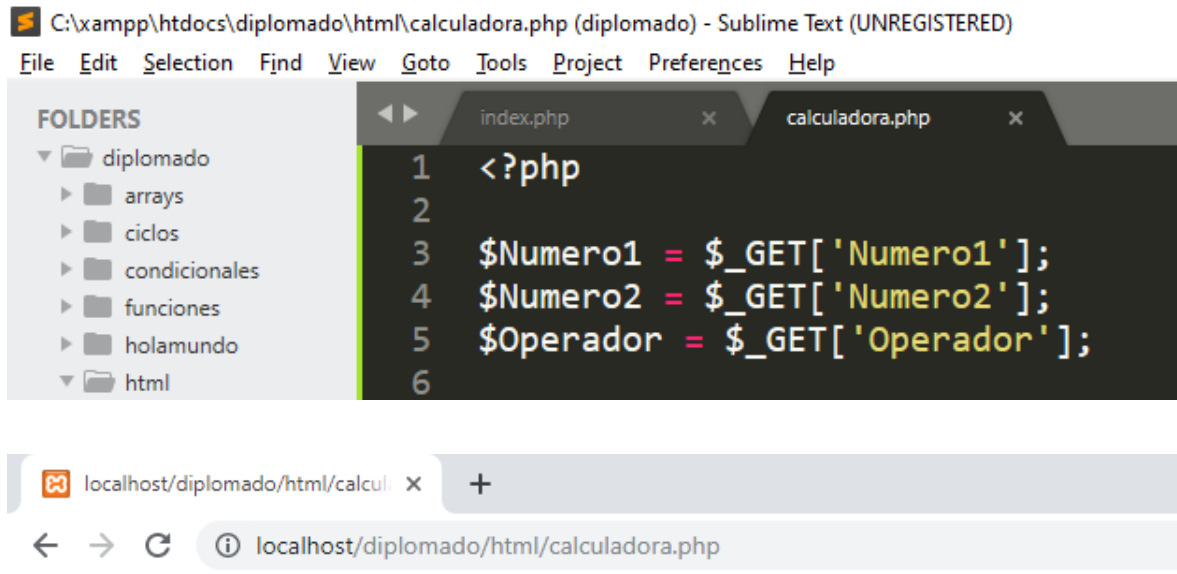
Ilustraciones 106 y 107.

El archivo calculador .php se encargará en primera instancia en recibir los datos enviados por desde el formulario, de la siguiente forma:

Cuando se define el método POST en el formulario en sus atributos, ese será el método sobre el que se va a enviar la información:

- Si se envía por POST, se recibe por POST.
- Si se envía por GET, se recibe por GET.

Una prueba sencilla de esto es cambiar en calculadora.php la forma de recibir la información, de POST a GET, y veremos el error al enviar una operación.



The screenshot shows a web browser window with the address bar displaying 'localhost/diplomado/html/calculadora.php'. Below the address bar, there are three error messages: 'Notice: Undefined index: Numero1 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 3', 'Notice: Undefined index: Numero2 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 4', and 'Notice: Undefined index: Operator in C:\xampp\htdocs\diplomado\html\calculadora.php on line 5'. To the right of the error messages, a code editor window is open, showing the PHP code for 'calculadora.php'. The code is as follows:

```
1 <?php
2
3 $Numero1 = $_GET['Numero1'];
4 $Numero2 = $_GET['Numero2'];
5 $Operator = $_GET['Operator'];
6
```

Notice: Undefined index: Numero1 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 3

Notice: Undefined index: Numero2 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 4

Notice: Undefined index: Operator in C:\xampp\htdocs\diplomado\html\calculadora.php on line 5

El resultado es: Error

[Regresar a la calculadora](#)

Ilustraciones 108 y 109.

El error radica en que PHP no encuentra las tres variables (\$Numero1, \$Numero2, \$Operator) enviadas con un método \$_GET.

Lo mismo ocurre si envío por GET y recibo por POST. Veamos.

```
5 </head>
6 <body>
7   <h1>Calculadora</h1>
8   <form action="calculadora.php" method="GET">
9     Número 1 <br>
```

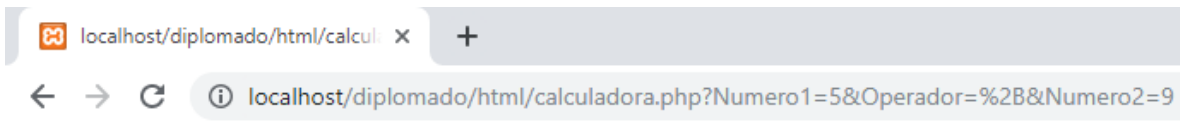
C:\xampp\htdocs\diplomado\html\calculadora.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html

```
1 <?php
2
3 $Numero1 = $_POST['Numero1'];
4 $Numero2 = $_POST['Numero2'];
5 $Operador = $_POST['Operador'];
6
```



Notice: Undefined index: Numero1 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 3

Notice: Undefined index: Numero2 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 4

Notice: Undefined index: Operador in C:\xampp\htdocs\diplomado\html\calculadora.php on line 5

El resultado es: Error

[Regresar a la calculadora](#)

Ilustraciones 110, 111 y 112.

El mismo error. Por eso es muy importante la consistencia de la información en relación con los métodos de envío y recepción de los datos. Continuemos con la explicación de la calculadora.

Otro error muy común es la inconsistencia entre los name="" y los métodos al recibir la información, por ejemplo:

C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - calculadora.php
 - index.php
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Diplomado</title>
5 </head>
6 <body>
7   <h1>Calculadora</h1>
8   <form action="calculadora.php" method="POST">
9     Número 1 <br>
10    <input type="number" name="numero1" required=""><br><br>
11    Operador <br>
12    <select name="operador" required="">
13      <option disabled="">Selecione</option>
14      <option value="+>+</option>
15      <option value="->-</option>
16      <option value="/">/</option>
17      <option value="*>*</option>
18    </select><br><br>
19    Número 2 <br>
20    <input type="number" name="numero2" required=""><br><br>
21    <input type="submit" value="Calcular">
22  </form>
23 </body>
24 </html>

```

Ilustración 113.

Todos los atributos *name* de los campos de entrada han sido cambiados, ahora todos tienen el mismo estilo, todo en minúscula. Veamos ahora el archivo *calculadora.php*:

C:\xampp\htdocs\diplomado\html\calculadora.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo

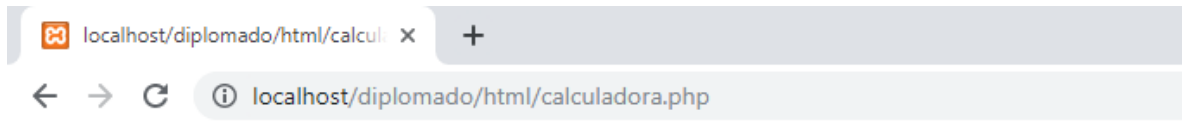
```

1 <?php
2
3 $Numero1 = $_POST['Numero1'];
4 $Numero2 = $_POST['Numero2'];
5 $Operador = $_POST['Operador'];

```

Ilustración 114.

Los métodos de envío coinciden en ambos casos, envío por POST, recibo por POST. Pero no hay consistencia en el nombre de los atributos, veamos que ocurre cuando enviamos información.



Notice: Undefined index: Numero1 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 3

Notice: Undefined index: Numero2 in C:\xampp\htdocs\diplomado\html\calculadora.php on line 4

Notice: Undefined index: Operador in C:\xampp\htdocs\diplomado\html\calculadora.php on line 5

El resultado es: Error

[Regresar a la calculadora](#)

Ilustración 115.

Exactamente el mismo error, en conclusión, los dos principales aspectos en uso de los formularios radican en:

- Métodos de envío y recepción de información.
- Nombre de los campos de entradas.

Teniendo la información encapsulada en nuestras variables, después de cada envío del formulario, se puede operar y determinar el resultado, una de las formas más fáciles de hacerlo es por medio de la estructura *switch*. Evaluando cada caso a raíz de la operación seleccionada.

```
switch ($Operador) {  
    case '+':  
        $Resultado = $Numero1 + $Numero2;  
        break;  
    case '-':  
        $Resultado = $Numero1 - $Numero2;  
        break;  
    case '/':  
        $Resultado = $Numero1 / $Numero2;  
        break;  
    case '*':  
        $Resultado = $Numero1 * $Numero2;  
        break;  
    default:  
        $Resultado = "Error";  
        break;  
}
```

Ilustración 116.

Recuerda el módulo número dos y la explicación del mismo concepto.

En la variable resultado vamos a contener el resultado de la operación según la seleccionada por el usuario y los números ingresados.

Finalmente haremos uso de dos conceptos directamente de HTML: encabezados e hipervínculos.

```
26  
27 <h1>El resultado es: <?php echo $Resultado; ?> </h1>  
28 <a href="index.php">Regresar a la calculadora</a>  
29
```

Ilustración 117.

El resultado estará en un <h1> y tendremos un hipervínculo para regresar a la calculadora. Ejercicio finalizado.

Otro ejercicio será simular un formulario de registro como Facebook. Veamos el formulario.

Formulario de Registro:



Registrarte ×

Es rápido y fácil.

Nombre

Apellido

Número de celular o correo electrónico

Contraseña nueva

Fecha de nacimiento ?

17 oct 1994

Sexo ?

Mujer ☐ Hombre ☐ Personalizado ☐

Al hacer clic en "Registrarte", aceptas nuestras [Condiciones](#), la [Política de datos](#) y la [Política de cookies](#). Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.

Registrarte

Ilustración 118.

En este formulario existen 11 campos de entrada de información (contando el botón), divididos de la siguiente forma:

- Campo de entrada de texto para el nombre:
<input type="text">
- Campo de entrada de texto para el apellido:
<input type="text">
- Campo de entrada de email para el correo electrónico:
<input type="email">
- Campo de entrada de *password* para la contraseña:
<input type="password">

- Lista desplegable para el día del nacimiento:
<select>
- Lista desplegable para el mes de nacimiento:
<select>
- Lista desplegable con el año de nacimiento:
<select>
- Campo de entrada de selección para el sexo “Mujer”:
<input type=“radio”>
- Campo de entrada de selección para el sexo “Hombre”:
<input type=“radio”>
- Campo de entrada de selección para el sexo “Personalizado”:
<input type=“radio”>
- Botón de envío de información:
<input type=“submit”>

Adicionalmente, replicaremos el párrafo de información adicional del formulario.

Al hacer clic en "Registrarte", aceptas nuestras [Condiciones](#), la [Política de datos](#) y la [Política de cookies](#). Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.

Ilustración 119.

Por otra parte, todos los atributos *name* tienen que ser claros:

- Nombres.
- Apellidos.
- Correo.
- Contraseña (recuerda que la ñ no es válida).
- DiaNacimiento.
- MesNacimiento.
- AnoNacimiento (recuerda que la ñ no es válida).
- Sexo (recuerda que los tres type=“radio” deben contar con el mismo nombre, dado que la selección de sexo se da gracias a estas tres opciones).

Veamos ahora el formulario codificado y con sus respectivos campos de entrada declarados, nombrados y listos para enviar información:

C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
 - arrays
 - ciclos
 - condicionales
 - funciones
 - holamundo
 - html
 - index.php
 - registrar.php
 - math
 - operadores
 - poo
 - switchcase
 - tiposdatos
 - variables

```

1  <?php
2
3  $Meses = array("Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto",
4              "Septiembre","Octubre","Noviembre","Diciembre");
5  ?>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <title>Diplomado</title>
10 </head>
11 <body>
12     <h1>Registro Facebook</h1>
13
14     <form method="POST" action="registrar.php">
15         Nombres <br>
16         <input type="text" name="Nombres" required="" placeholder="Nombres"><br><br>
17         Apellidos <br>
18         <input type="text" name="Apellidos" required="" placeholder="Apellidos"><br><br>
19         Correo <br>
20         <input type="email" name="Correo" required="" placeholder="Correo"><br><br>
21         Contraseña <br>
22         <input type="password" name="Contraseña" required="" placeholder="Contraseña"><br><br>
23         <select name="DiaNacimiento" required="">
24             <option>Día</option>
25             <?php
26             for ($i=1; $i <= 31; $i++) {
27                 ?>
28                 <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
29             }
30             ?>
31         </select>
32
33     <select name="MesNacimiento" required="">
34         <option>Mes</option>
35         <?php
36         foreach ($Meses as $Mes) {
37             ?>
38             <option value="<?php echo $Mes; ?>"><?php echo $Mes; ?></option>
39         }
40         ?>
41     </select>
42     <select name="AñoNacimiento" required="">
43         <option>Año</option>
44         <?php
45         for ($i=2019; $i >= 1950; $i--) {
46             ?>
47             <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
48         }
49         ?>
50     </select><br><br>
51     Sexo <br>
52     <input type="radio" name="Sexo" value="Mujer"> Mujer
53     <input type="radio" name="Sexo" value="Hombre"> Hombre
54     <input type="radio" name="Sexo" value="Personalizado"> Personalizado
55     <p>Al hacer clic en "Registrarte", aceptas nuestras Condiciones, la Política de datos y la Política
56     de cookies.<br>Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.</p>
57     <input type="submit" value="Registrarte">
58 </form>
59
60 </body>
61 </html>

```

Ilustraciones 120 y 121.

- Para ver el resultado da clic aquí: [Formulario de Registro](#)

- Para descargar el archivo da clic aquí: [Descargar](#)

Este formulario cuenta con un gran volumen de datos, por lo que hay que ser cuidadosos en manejar el diseño y funcionalidad de este.

La forma en la que opera es la siguiente:

```
<form method="POST" action="registrar.php">
```

- El método de envío es POST.
- El archivo al enviar la información será «registrar.php».

```
Nombres <br>
<input type="text" name="Nombres" required="" placeholder="Nombres"><br><br>
Apellidos <br>
<input type="text" name="Apellidos" required="" placeholder="Apellidos"><br><br>
Correo <br>
<input type="email" name="Correo" required="" placeholder="Correo"><br><br>
Contraseña <br>
<input type="password" name="Contraseña" required="" placeholder="Contraseña"><br><br>
```

Ilustraciones 122 y 123.

- Los campos de entrada, cada uno cuenta con las siguientes características:
 1. Atributo *type* especificado.
 2. Atributo *name* claro.
 3. Atributo *required*.
 4. Atributo *placeholder* con la descripción del campo.

El atributo *placeholder* es nuevo, este permite definir un texto predeterminado en los campos de entrada. El texto es simplemente visual, no se encuentra habilitado como un valor dentro del campo.

La forma óptima de mostrar los días del mes en la lista desplegable es por medio de un ciclo *for* para recorrer todos los días de un mes. Comprendiendo los días del mes desde 1 hasta 31, de la siguiente forma:

```
<select name="DiaNacimiento" required="">
  <option>Día</option>
  <?php
    for ($i=1; $i <= 31; $i++) {
      ?>
      <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
      <?php
    }
    ?>
</select>
```

Ilustración 124.

Para escribir los doce meses del año, y evitar escribir uno por uno en la lista desplegable (también es válido hacerlo), cree un *array* con los doce meses y en la lista desplegable se ejecuta un *foreach* para mostrar los mismo.

```
<?php

$Meses = array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto",
               "Septiembre", "Octubre", "Noviembre", "Diciembre");
?>
```

```
<select name="MesNacimiento" required="">
  <option>Mes</option>
  <?php
    foreach ($Meses as $Mes) {
      ?>
      <option value="<?php echo $Mes; ?>"><?php echo $Mes; ?></option>
      <?php
    }
    ?>
</select>
```

Ilustraciones 125 y 126.

La forma que se implementó para obtener los años en la lista desplegable es una muy recursiva, un ciclo *for* que se decrementa a razón de uno (un año a la vez) hasta llegar a un límite, en este caso hasta 1950 (2019-1950).

```
<select name="AñoNacimiento" required="">
  <option>Año</option>
  <?php
    for ($i=2019; $i >= 1950; $i--) {
      ?>
      <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
      <?php
    }
    ?>
</select><br><br>
```

Ilustración 127.

Los tipos radio son muy particulares, dado que se crean X a partir de un solo *name*, es decir, a partir del atributo *name* pueden crear los siguientes tipos, por ejemplo, para un examen, las posibles respuestas podrían ser:

- Sí.
- No.
- Casi nunca.
- Nunca.

Y todos contener el mismo nombre, por decir uno, *name="respuesta"*, lo que permite esto es la selección unitaria de uno de estos campos, así como ocurre en el caso del formulario de ejemplo con el sexo.

```
Sexo <br>
<input type="radio" name="Sexo" value="Mujer"> Mujer
<input type="radio" name="Sexo" value="Hombre"> Hombre
<input type="radio" name="Sexo" value="Personalizado"> Personalizado
```

Ilustración 128.

Tres diferentes campos, diferentes opciones, pero el mismo atributo *name*.

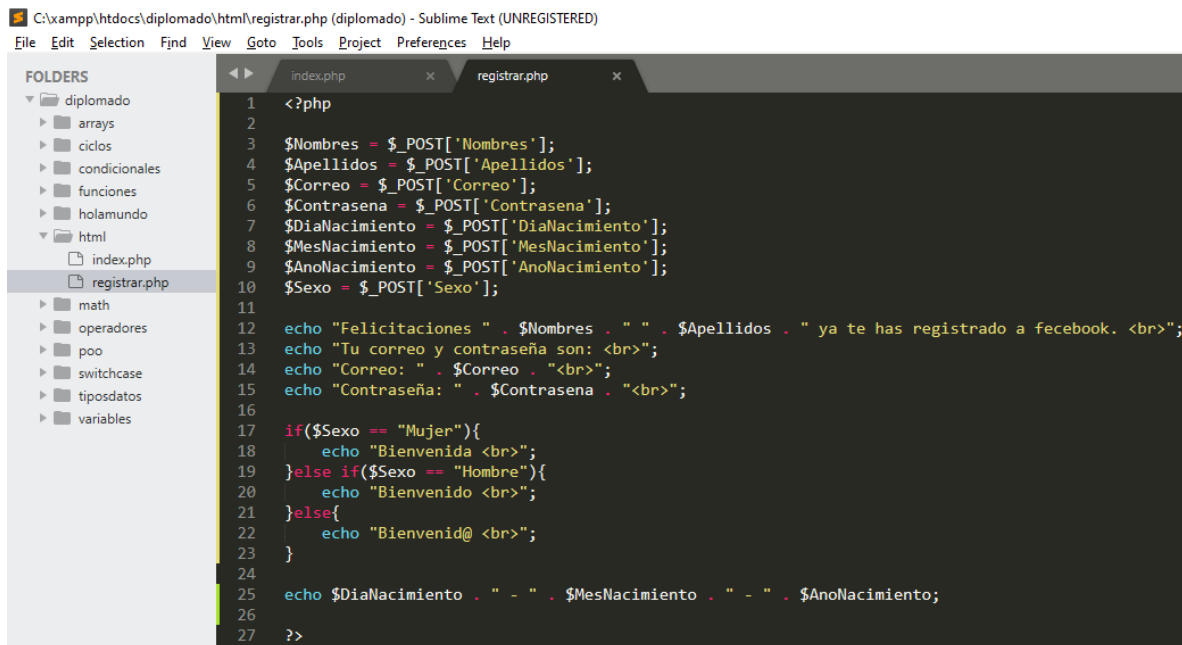
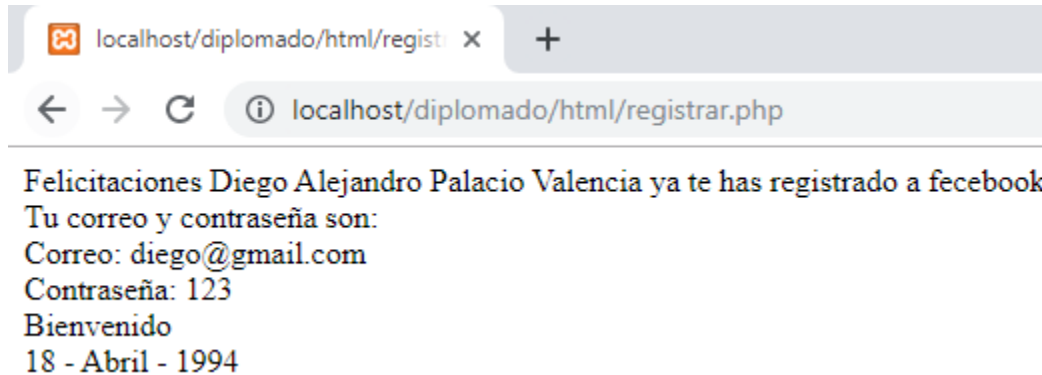
En este ejercicio conocimos una nueva etiqueta, *<p>*, que es muy simple y funcional, y permite definir un párrafo:

```
<p>Al hacer clic en "Registrarte", aceptas nuestras Condiciones, la Política de datos y la Política de cookies.<br>Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.</p>
```

Ilustración 129.

Con toda la información de los campos completada, simplemente resta realizar el envío al *action* descrito, en este caso, registrar.php.

Recibiremos todos los campos enviados por POST para procesarlos debidamente y mostrar la siguiente estructura:



Ilustraciones 130 y 131.

Lo primero que se debe hacer en este caso es siempre recibir la información enviada por el método definido, sea GET o POST:

```
$Nombres = $_POST['Nombres'];  
$Apellidos = $_POST['Apellidos'];  
$Correo = $_POST['Correo'];  
$Contrasena = $_POST['Contrasena'];  
$DiaNacimiento = $_POST['DiaNacimiento'];  
$MesNacimiento = $_POST['MesNacimiento'];  
$AñoNacimiento = $_POST['AñoNacimiento'];  
$Sexo = $_POST['Sexo'];
```

Ilustración 132.

Recuerda siempre que los atributos *name* deben coincidir con las variables recibidas en `$_POST[]`, deben ser claros y dicientes.

Los primeros *echos* se encargan de hacer uso de las variables recibidas, este proceso es completamente similar al visto en los módulos anteriores, cuando realizábamos impresión de variables con `echo` y texto plano (no te olvides de la etiqueta de salto de línea `
`).

```
echo "Felicitaciones " . $Nombres . " " . $Apellidos . " ya te has registrado a fecebook. <br>";  
echo "Tu correo y contraseña son: <br>";  
echo "Correo: " . $Correo . "<br>";  
echo "Contraseña: " . $Contrasena . "<br>";
```

Ilustración 133.

De nuevo haremos uso de una estructura condicional, *if - else*. Queremos mostrar un mensaje a partir del sexo seleccionado en el formulario.

- Bienvenida - Para las mujeres.
- Bienvenido - Para los hombres.
- Bienvenid@ - Para los personalizados.

```
if($Sexo == "Mujer"){  
    echo "Bienvenida <br>";  
}else if($Sexo == "Hombre"){  
    echo "Bienvenido <br>";  
}else{  
    echo "Bienvenid@ <br>";  
}
```

Ilustración 134.

Y para finalizar, debemos imprimir la fecha de nacimiento concatenando día (Dia), mes (Mes) y año (Año), de la siguiente forma:

```
echo $DiaNacimiento . " - " . $MesNacimiento . " - " . $AñoNacimiento;
```

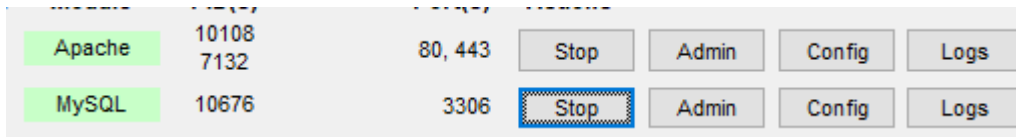
Ilustración 135.

Ya hemos realizado un formulario con buena cantidad de campos y está completamente funcional. Recuerda muy bien cada uno de estos conceptos, dado que son los que siempre estarán presentes en el desarrollo de HTML y PHP. Puedes practicar creando un formulario de inicio de sesión, un formulario de contacto, etc, las posibilidades son infinitas.

Con el proyecto que desarrollaremos, realizaremos varios formularios con funcionalidades diferentes, utilizaremos tablas y bases de datos, será básico, pero con mucho contenido práctico para entender cómo funciona el desarrollo web con PHP. Más adelante en el módulo tendrás todos los requisitos y características del proyecto.

Tema 4: Bases de Datos - MySQL (Parte I)

Al momento de descargar XAMPP, este ofrecía dos características fundamentales en el desarrollo bajo PHP. El primero fue Apache, el servidor que se ha utilizado a lo largo del diplomado para ejecutar nuestro código PHP en el navegador, y ahora el otro, MySQL o, en otras palabras, phpMyAdmin:



Apache	10108 7132	80, 443	Stop	Admin	Config	Logs
MySQL	10676	3306	Stop	Admin	Config	Logs

Ilustración 136.

phpMyAdmin es una herramienta gratuita que permite de una manera muy completa acceder a todas las funciones de la base de datos MySQL a través de PHP, mediante una interfaz web muy intuitiva.

¿Qué ofrece phpMyAdmin?

Esta herramienta es muy completa y nos ofrece una gran cantidad de usos y características, algunas de ellas son:

- Esta aplicación nos permitirá realizar las operaciones básicas en base de datos MySQL, como son: crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves de campos, administrar privilegios y exportar datos en varios formatos. La función de exportar datos se emplea muchas veces para realizar *backups* de la base de datos y poder restaurar esta copia de seguridad en el futuro a través de phpMyAdmin mediante la opción «importar».
- phpMyAdmin es el administrador de bases de datos por defecto en muchos paneles de control web comerciales, como son cPanel, Plesk o DirectAdmin.
- Los usuarios no deberían tener problemas a la hora de manejar esta herramienta, ya que es fácil de usar.

- Otra de las funciones más importantes que nos ofrece es que permite optimizar y reparar tablas, las cuales son dos tareas de mantenimiento fundamentales.
- Nos da la posibilidad de realizar búsquedas en las bases de datos, además de poder escribir nuestras propias consultas SQL de manera directa y ejecutarlas.
- Esta herramienta también es de gran ayuda para desarrolladores de aplicaciones que empleen MySQL, ya que permite depurar consultas y hacer test de forma rápida y sencilla.

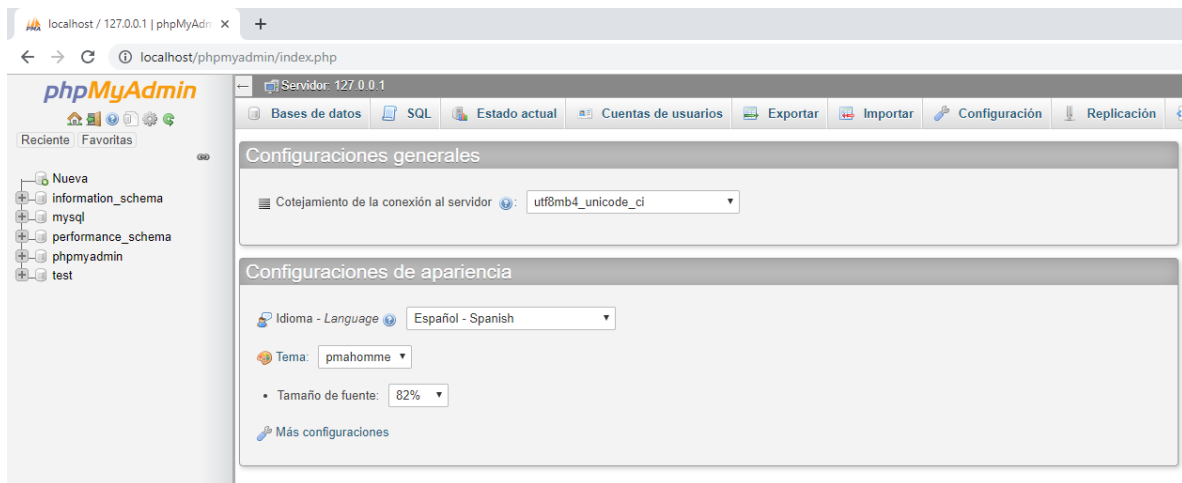


Ilustración 137.

Las formas de acceder a phpMyAdmin son muy sencillas, la primera de estas es directamente por medio de XAMPP, dando clic en la opción «admin»:

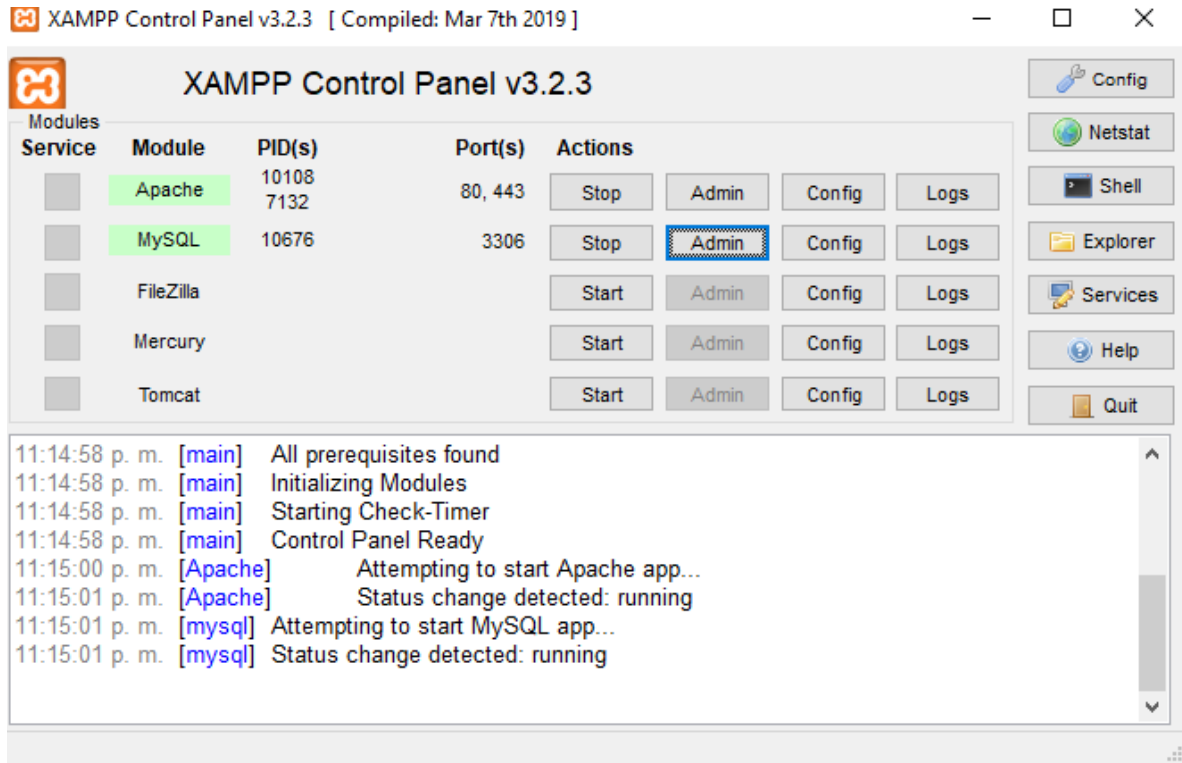


Ilustración 138.

También es válido ir directamente a la ruta donde se ejecuta phpMyAdmin en: <http://localhost/phpmyadmin/index.php>.

Esta herramienta provista por XAMPP tiene una gran cantidad de características y funciones esenciales descritas anteriormente, pero el alcance del diplomado estará relacionado con los siguientes temas:

- Creación de bases de datos.
- Creación de tablas.
- Especificación de campos de tabla.
- Tipos de datos.
- Claves primarias.
- Inserción de datos.
- Consulta de datos.
- Actualización de datos.
- Eliminación de datos.
- Exportación de datos y tablas.

- Copias de seguridad.
- Modelo Entidad Relación.

Para crear nuestra primera base de datos, que será la utilizada en el proyecto final, el proceso que debemos realizar es el siguiente:

- Abrir phpMyAdmin.
- En el apartado superior izquierdo se encuentran las bases de datos establecidas por defecto de MySQL y las que el desarrollador vaya ocupando en su labor. Ir a la opción de «Nueva».

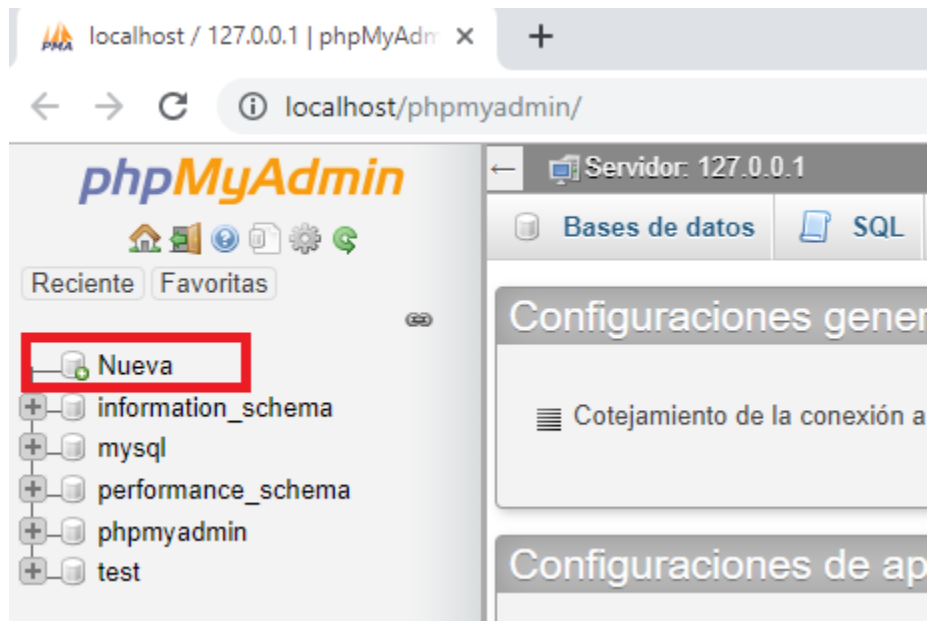
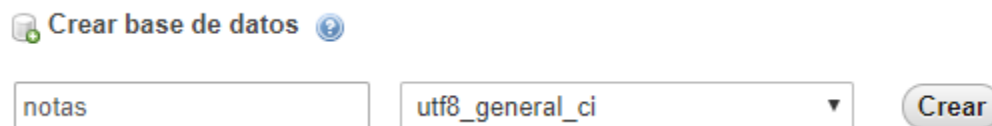
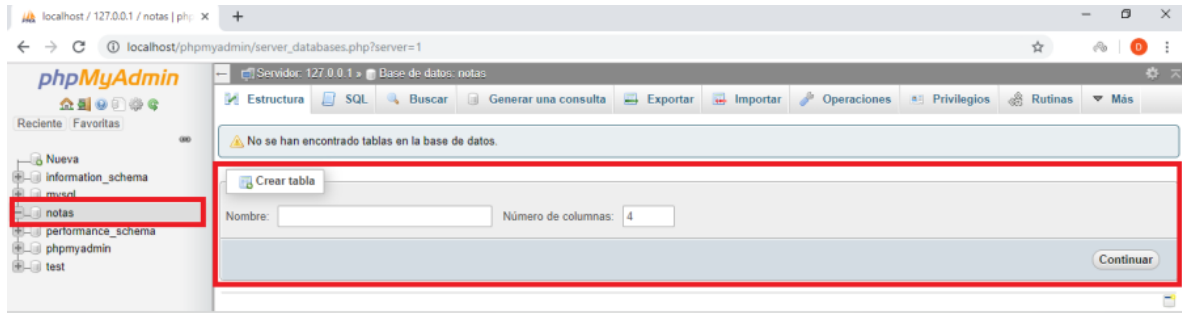


Ilustración 139.


- En esta nueva ventana tendremos dos opciones para continuar, el nombre de la base de datos y el cotejamiento (codificación de datos). Por defecto serán los siguientes:





Ilustraciones 140 y 141.

Este es el resultado final del proceso de crear la base de datos. Es muy sencillo, como viste. En el próximo módulo realizamos la creación de las tablas de la base de datos y todos los temas que girarán en torno al proyecto, al PHP y MySQL.



Esta guía fue elaborada para ser utilizada con fines didácticos como material de consulta de los participantes en el diplomado virtual en PROGRAMACIÓN EN PHP del Politécnico de Colombia, y solo podrá ser reproducida con esos fines. Por lo tanto, se agradece a los usuarios referirla en los escritos donde se utilice la información que aquí se presenta.

GUÍA DIDÁCTICA 4

M2-DV59-GU04

MÓDULO 4: DESARROLLO WEB I

© DERECHOS RESERVADOS - POLITÉCNICO DE COLOMBIA, 2023
Medellín, Colombia

Proceso: Gestión Académica Virtual
Realización del texto: Diego Palacio, docente
Revisión del texto: Comité de Revisión
Diseño: Comunicaciones

Editado por el Politécnico de Colombia.