

# Formularios web y recursos a la base de datos

TECNOLOGÍAS WEB

# formularios

- La forma en que los usuarios pueden enviar datos al servidor mediante tecnologías Web es usando formularios
- El HTML5 incluye muchas nuevas características para hacer los formularios web más fáciles de crear, más poderosos y consistentes a través de la Web.

# <form>

- Es la etiqueta que delimita los elementos que permitirán al usuario introducir datos para ser enviados al servidor
- Atributos:
  - accept-charset
  - action
  - autocomplete
  - enctype
  - method
  - Name
  - novalidate

# <label>

- Nos permite escribir los elementos que se requieren para que el usuario los introduzca, es un elemento que puede se asociado a un control específico del formulario
  - El atributo id es necesario para ser asociado a un elemento del formulario

```
<label for="Nombre">Nombre del usuario </label>
```

```
<input type="text" id="Nombre" name="Nombre" size="10"  
maxlength="20" placeholder="Nombre" required><br>
```

# <input>

- Las entradas de datos pueden ser de diferentes tipos dependiendo de las necesidades del sistema, es por eso que las entradas se clasifican de acuerdo al tipo de dato que se soporta
  - HTML 4: text, password, checkbox, radio, submit, image, reset, button, hidden, file
  - HTML 5: search, tel, url, email, datetime, date, month, week, time, datetime-local, number, range, color.
  - Algunos atributos añadidos adicionalmente son: pattern (el valor del campo debe cumplir un patrón que se define mediante expresiones regulares), placeholder (para incluir un texto por defecto en el campo), required (campo obligatorio), autofocus (el campo tiene por defecto el foco)

# <button>

- Es un elemento que permite disparar una acción cuando se habilita, existen diferentes tipos de botones
  - submit button: Envía elementos del formulario al servidor
  - reset button: Borra los valores de todos los elementos del formulario
  - Button: Debemos definir el tipo de acción

# <select>

- Es un elemento que permite selección un valor de un conjunto de elementos previamente especificado, soporta una selección de múltiples elementos.
  - Cada elemento del select debe definirse con la etiqueta <option>
  - Cada etiqueta <option> debe tener atributos de value definidos

# <datalist>

- Es un elemento que permite definir valores de un conjunto que pueden ser elegibles predefinidos para otros controles
  - Se enlaza cada etiqueta <input> colocando como atributo list y en la definición del <datalist> se debe definir el identificador con el atributo id

```
<input type="email" id="email" name="email"
      size="30" list="listaformatos" required>
<datalist id="listaformatos">
  <option>.com</option>
  <option>.mx</option>
  <option>.com.mx</option>
  <option>.org</option>
</datalist>
```



# Recepción de datos en PHP

- Los archivos de PHP que del lado del servidor reciben datos, tienen la capacidad de almacenar de manera temporal los datos pasados por formulario
- Lo adecuado es recibir la información, validar y posterior a eso almacenar.
- Existen librerías especializadas en la limpieza de datos provenientes de formulario

# \$\_GET

- Es un arreglo asociativo que permite almacenar de manera temporal los datos pasados por formulario mediante el método GET
- Los datos se van enviando al componer la url con un signo de ?.
- Cada par se compone de nombre=valor
  - Se pueden recuperar los datos a partir del arreglo accediendo mediante las tuplas de “name” y “value”
    - `valor_de_la_variable =$_GET [“name_del_input_del_formulario”]`

# `$_POST`

- Es un arreglo asociativo que permite almacenar de manera temporal los datos pasados por formulario mediante el método POST
- Los datos se van a enviar como parte del paquete de datos cuando se emplea `application/x-www-form-urlencoded` o `multipart/form-data` como Content-Type del formulario.
- Cada par se compone de `nombre=valor`
  - Se pueden recuperar los datos a partir del arreglo accediendo mediante las tuplas de “name” y “value”
    - `valor_de_la_variable = $_POST [“name_del_input_del_formulario”]`

# \$\_FILES

- Es un arreglo asociativo que permite acceder a la ruta de los archivos copiados al servidor de manera temporal, los archivos fueron enviados mediante formulario con la etiqueta `<input>` y el tipo `file`.
  - Se requiere después de acceder a la ruta, copiar el archivo a la nueva dirección donde permanecerá de manera permanente, ya que la carpeta temporal del servidor se borra de manera frecuente.
  - `move_uploaded_file()` \*

# Conexión a la base de datos con php

- El manejo de solicitudes con PHP y MySQL se realiza haciendo una solicitud que requiere de la autorización del gestor de base de datos.
- Las conexiones pueden abrir y mantenerse en ese estado mientras se ejecutan las operaciones del script.
- Es una buena práctica cerrar las conexiones una vez que ya no se utilizan.

# mysqli\_connect

- Es la función que permite crear conexiones de PHP con gestores de bases de datos
- Se puede hacer de manera de lógica procedimental o de orientación a objetos
  - Ruta del servidor
  - Base de datos
  - Nombre de usuario (BD)
  - Clave de usuario (BD)

# mysqli\_query

- Permite realizar consultas a la base de datos
- Las consultas deben de construirse siguiendo la sintaxis del manejador de base de datos
- Se permite capturar el código de retorno de la operación de la función para conocer si se realiza de manera correcta una operación en la base de datos.

# Ejemplo de conexión a base de datos

```
$servername = "localhost";  
$database = "registro";  
$username = "alex";  
$password = "tecnologias";  
// Conectar a la base de datos  
$conection = mysqli_connect($servername, $username, $password, $database);  
// REvisar que se hace de manera correcta  
  
if (!$conection) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
echo ("Conexión realizada de manera correcta");
```



# Insertar datos en MySQL

```
// Insertar registro en la base de datos

$respuesta=mysqli_query($conexion, "INSERT INTO personales (nombre,edad)
VALUES ('Alejandro',35)");

//echo $respuesta;

if($respuesta){
|   echo "Se ingresaron los registros con exito";
}
else
{
|   echo "No se ingresaron los registros. <br>";
}
```

# Cerrar la conexión a base de datos

```
mysqli_close($conection);
```