



Ejecutar Insert

En el formulario que envía los datos...

- En el formulario que envía, se ha de indicar la ruta del documento **php** que recibirá los datos de sus campos, esto lo hacemos con el atributo ***action***.

```
<form role="form" id="login-form"
      method="post" class="form-signin"
      action="proveedor_guardar.php">
```

- Cada información que contienen los campos del formulario se obtendrá mediante su atributo ***name***

```
<div class="form-group">
  <label for="nombre_del_proveedor">Nombre del Proveedor (requerido)</label>
  <input type="text" class="form-control" id="nombre_del_proveedor" name="nombre_del_proveedor"
        placeholder="Ingresa nombre del proveedor" style="text-transform:uppercase;" required>
</div>
```

Creando una conexión

- Para conectar a nuestra BD utilizaremos un bloque de código **php** en el que se especifica: la URL del servidor, el nombre de usuario que se utiliza, el password, y el nombre de la BD correspondiente, de forma conveniente crearemos un documento **php** con dicho código y lo guardaremos en la carpeta *inc* de nuestro proyecto, para este ejemplo lo llamaremos *conexión.php*

```
<?php
$db_host = "localhost";
$db_user = "root";
$db_password = "";
$db_name = "tallerbd";

$con = new mysqli($db_host, $db_user, $db_password, $db_name);
?>
```

Insertando registros

- Existen múltiples formas de hacer que un documento **php** inserte registros en la BD, esto puede depender de la versión del lenguaje que se esté utilizando, algunas de estas formas son:
 - 1) Utilizando una consulta ensamblada de forma directa (no recomendado).
 - 2) Llamando a un procedimiento almacenado.
 - 3) Utilizando consultas preparadas.

Consulta ensamblada

- La razón por la que no se recomienda esta práctica es por ser susceptible a inyecciones SQL

```
<?php
include 'inc/conexion.php';
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nombre_del_proveedor_post = strtoupper($_POST['nombre_del_proveedor']);
    $direccion_del_proveedor_post = strtoupper($_POST['direccion_del_proveedor']);
    $telefono_1_post = strtoupper($_POST['telefono_1']);
    $telefono_2_post = strtoupper($_POST['telefono_2']);
    $correo_proveedor_post = strtoupper($_POST['correo_proveedor']);
    $ins = $con->query("INSERT INTO proveedor VALUES ("
        . "NULL, '$nombre_del_proveedor_post', '$direccion_del_proveedor_post', "
        . "'$telefono_1_post', "
        . "'$telefono_2_post', '$correo_proveedor_post')");
    if ($ins) {
        echo "Proveedor Registrado";
    } else {
        echo "NO SE HA EFECTUADO EL REGISTRO";
    }
    $con->close();
}
```

- Nota: se ha utilizado el función **strtoupper()** para forzar la conversión a mayúsculas


Insertar con consulta preparada

```
<?php
include 'inc/conexion.php';
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nombre_del_proveedor_post = strtoupper($_POST['nombre_del_proveedor']);
    $direccion_del_proveedor_post = strtoupper($_POST['direccion_del_proveedor']);
    $telefono_1_post = strtoupper($_POST['telefono_1']);
    $telefono_2_post = strtoupper($_POST['telefono_2']);
    $correo_proveedor_post = strtoupper($_POST['correo_proveedor']);
    $id_proveedor='';
    $ins=$con->prepare("INSERT INTO proveedor VALUES(?,?,?,?,?,?)");
    $ins->bind_param("isssss",$id,$nombre_del_proveedor_post,$direccion_del_proveedor_post,
    $ins->execute();
    $ins->close();
    $con->close();
}
```

Es de suma importancia cerrar las conexiones que de abren a la BD

Estos caracteres indican el tipo de variable que se está pasando: i (enteros), d (doubles), s (strings)

Cada símbolo representa cada uno de los campos que se van a insertar, incluyendo los de tipo auto-numérico



```
<?php
include 'inc/conexion.php';
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nombre_del_proveedor_post = strtoupper($_POST['nombre_del_proveedor']);
    $direccion_del_proveedor_post = strtoupper($_POST['direccion_del_proveedor']);
    $telefono_1_post = strtoupper($_POST['telefono_1']);
    $telefono_2_post = strtoupper($_POST['telefono_2']);
    $correo_proveedor_post = strtoupper($_POST['correo_proveedor']);
    $id_proveedor="";
    $ins=$con->prepare("INSERT INTO proveedor VALUES(?,?,?,?,?,?)");
    $ins->bind_param("isssss",$id,$nombre_del_proveedor_post,$direccion_del_proveedor_post,$telefono_1_post,$telefono_2_post,
$correo_proveedor_post);
    if($ins->execute()){
        echo "Se ha registrado el proveedor";
    }
    else{
        echo "Error al insertar Proveedor";
    }
    $ins->close();
    $con->close();
}
```

NOTA: Podríamos mejorar la salida de nuestro script, utilizando la plantilla que hemos definido para nuestro proyecto (menú y contenedor en bootstrap), sin embargo posteriormente lo haremos utilizando Sweet Alert.

Validación de controles HTML5

- Las validaciones nativas de HTML5 se han extendido en las últimas revisiones, sin embargo se presentan algunas utilizadas de forma habitual en los controles input, utilizando su atributo Type: *search, url, tel, email, password, date pickers, number, checkbox, radio y file*. Por ejemplo:
 - `<input title="Se requiere" type="text" name="nombre" required>`
 - `<input type="email" name="email" required>`
- Si queremos añadir un valor numérico con cierto formato en cuanto a sus incrementos o decimales podemos utilizar el atributo step
 - `<input type="number" class="form-control" id="precio" name="precio" step="0.01" placeholder="Precio actual" required>`

Otras funcionalidades recurrentes

- Llenar un input con la fecha del sistema:
 - `<input type="date" class="form-control" id="fecha_solicitud" name="fecha_solicitud" step="1" value="<?php echo date("Y-m-d"); ?>" required>`
- Convertir a mayúscula los caracteres del input
 - `<input type="text" class="form-control" id="nombre_del_proveedor" name="nombre_del_proveedor" placeholder="Ingresa nombre del proveedor" style="text-transform:uppercase;" required>`

Clase de validación en CSS

- invalid representa un campo invalido es decir que no haya pasado la verificación, el cual adoptará la regla que se haya declarado
 - `input:invalid { border: 1px solid red; }`
 - `/* Estilo por defecto */`
 - `input:valid { border: 1px solid green;}`
- Para un campo requerido
 - `/* Estilo por defecto */`
 - `input:required:invalid { border: 1px solid red; }`
 - `input:required:valid { border: 1px solid green; }`