

**UNIVERSIDAD TECNOLÓGICA DE TIJUANA**



**TSU EN TECNOLOGÍAS DE LA INFORMACIÓN**

**ÁREA**

**DESARROLLO DE SOFTWARE**

**MULTIPLATAFORMA**

**MATERIA:**

**Aplicaciones WEB**

**PROFESOR:**

**Ray Brunett Parra Galaviz**

**ESTUDIANTE:**

**Suastegui Leyva Yerlan Axel**

**GRUPO:**

**3D BIS**

**ACTIVIDAD:**

**Portafolio**

**FECHA:**

**31 de octubre DEI 2024**

## ¿Que son las Variables Super Globales?

Este tipo de Variables a diferencia de las variables normales, son variables integradas o preestablecidas en el mismo Lenguaje de Programación PHP para ser usadas desde cualquier parte de tu código de PHP, es decir son accesibles independientemente del alcance que tenga una línea de código.

Puedes acceder a estas Variables Super Globales, desde cualquier función, clase o archivo PHP, sin tener que hacer algo especial, son sencillas de usarse y muy potentes.

### **\$\_GET:**

- Se usa para capturar datos enviados desde un formulario HTML utilizando el método **GET** o a través de parámetros en la URL.
- Los datos enviados a través de \$\_GET se muestran en la barra de direcciones del navegador, lo que hace que sean visibles para cualquiera, lo que no es ideal para información sensible.

- Los valores pasados tienen un límite en la cantidad de caracteres (generalmente 2048), por lo que no es adecuado para grandes volúmenes de datos o archivos binarios.

#### Ejemplo #1 Ejemplo de `$_GET`

```
<?php
echo '¡Hola ' . htmlspecialchars($_GET["nombre"]) . '!';
?>
```

Asumiendo que el usuario introdujo `http://example.com/?nombre=Hannes`

El resultado del ejemplo sería algo similar a:

```
¡Hola Hannes!
```

## **\$\_POST:**

- Se utiliza para manejar datos enviados a través del método **POST**, donde los valores no se muestran en la URL, haciéndolos más seguros para transmitir información confidencial, como contraseñas.
- A diferencia de \$\_GET, no tiene límite de tamaño en los datos enviados, lo que lo hace adecuado para el envío de formularios más grandes o archivos.

### Ejemplo #1 Ejemplo de \$\_POST

```
<?php
echo '¡Hola ' . htmlspecialchars($_POST["nombre"]) . '!';
?>
```

Asumiendo que el usuario envió por el método POST nombre=Juan

El resultado del ejemplo sería algo similar a:

```
¡Hola Juan!
```

## **BIBLIOGRAFÍAS:**

Colectiva, N. (2020, 20 enero). Que son las Variables Super Globales y cuales son en PHP. *Blog de Programación y Desarrollo - Nube Colectiva*.

<https://blog.nubecolectiva.com/que-son-las-variables-super-globales-y-cuales-son-en-php/>

Gabil, D. (2024, 5 marzo). *PHP: SuperGlobals \$\_GET, \$\_POST, \$\_REQUEST, \$\_COOKIE, \$\_SESSION*. ScriptVerse. <https://www.scriptverse.academy/tutorials/php-superglobals.html>

*PHP: Hypertext Preprocessor*. (s. f.).

<https://www.php.net/manual/es/reserved.variables.post.php>

## Qué implica hacer una conexión en PHP a MySQL

Conectar PHP a MySQL es una tarea fundamental en el desarrollo web, ya que permite la interacción entre el código de nuestra aplicación y la base de datos donde se almacenan los datos. Este proceso implica varios conceptos clave y una comprensión profunda de cómo funcionan juntos PHP y MySQL.

PHP, acrónimo de «Hypertext Preprocessor», es un lenguaje de programación del lado del servidor ampliamente utilizado para el desarrollo web, especialmente popular debido a su facilidad de uso, flexibilidad y capacidad para integrarse con diversas bases de datos, incluidos MySQL, PostgreSQL, SQLite, y otros. Se utiliza principalmente para crear páginas web dinámicas que pueden interactuar con bases de datos para almacenar y recuperar información en tiempo real.

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que utiliza el lenguaje SQL (Structured Query Language) para gestionar y manipular datos, conocido por su rendimiento, fiabilidad y facilidad de uso. Es una elección popular para aplicaciones web debido a su capacidad para manejar grandes volúmenes de datos y su compatibilidad con numerosas plataformas y lenguajes de programación, incluido PHP.

## Por Qué PHP y MySQL Van de la Mano

PHP y MySQL son una combinación potente y eficiente para el desarrollo web por varias razones:

- 1. Compatibilidad:** PHP tiene soporte integrado para MySQL, lo que facilita la conexión y manipulación de bases de datos desde el código PHP sin necesidad de librerías adicionales.
- 2. Facilidad de Uso:** Ambos son fáciles de aprender y usar, lo que los hace accesibles para desarrolladores de todos los niveles de experiencia.
- 3. Rendimiento:** Juntos, PHP y MySQL pueden manejar grandes volúmenes de tráfico y datos, lo que los hace ideales para aplicaciones web de alto rendimiento.
- 4. Comunidad y Soporte:** Ambos tienen grandes comunidades de usuarios y una amplia documentación, lo que facilita encontrar ayuda y recursos.

## Funcionamiento de la Conexión PHP-MySQL

El proceso de conexión y comunicación entre PHP y MySQL sigue estos pasos:

- 1. Inicialización:** PHP utiliza las credenciales proporcionadas para establecer una conexión con el servidor MySQL.
- 2. Autenticación:** MySQL verifica las credenciales del usuario y, si son correctas, permite el acceso.
- 3. Comunicación:** Una vez autenticado, PHP puede enviar consultas SQL a MySQL. MySQL ejecuta las consultas y devuelve los resultados a PHP.
- 4. Procesamiento:** PHP recibe los resultados y puede procesarlos, formatearlos y presentarlos en la página web.
- 5. Terminación:** Finalmente, PHP cierra la conexión cuando ya no es necesaria.

### Conectar PHP a MySQL

Dentro de tu archivo PHP, puedes conectar MySQL usando **MySQLi** o **PDO** (PHP Data Objects). Aquí hay un ejemplo usando **MySQLi**:

```
<?php
$host = 'localhost';
$db = 'ejemplo';
$user = 'usuario';
$pass = 'password';

// Crear la conexión
$conn = new mysqli($host, $user, $pass, $db);

// Verificar la conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
echo "Conexión exitosa a la base de datos";
?>
```

### BIBLIOGRAFÍAS:

Tecnológico, S. (2024, 16 julio). *Cómo hacer una conexión PHP a MySQL - Sendero tecnológico*. Sendero Tecnológico. <https://senderotecnologico.com/como-hacer-una-conexion-en-php-a-mysql-tanto-en-sitios-wordpress-como-nativos/>

## Proyecto de bienes raíces

### HEADER

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>BIENES RAICES</title>
</head>
<body>
  <section>
    <h1>Bienes Raices</h1>
    <header>
      <nav>
        <a href="index.php">Home</a>
        <a href="nosotros.php">Nosotros</a>
        <a href="">Anuncios</a>
        <a href="contacto.php">Contacto</a>
        <a href="blog.php">Blog</a>
      </nav>
    </header>
  </section>
</body>
```

En la cabecera de la página, se define la codificación de caracteres como UTF-8 y se ajusta la visualización para dispositivos móviles con la etiqueta `<meta name="viewport" content="width=device-width, initial-scale=1.0">`. Dentro de la etiqueta `<body>`, hay una sección principal con el encabezado `<h1>` que muestra el título "Bienes Raíces". También se incluye una barra de navegación dentro de `<nav>`, con enlaces (mediante `<a>` tags) que permiten a los usuarios acceder a diferentes secciones: "Home" (página principal), "Nosotros" (información sobre la empresa), "Anuncios" (listado de propiedades), "Contacto" y "Blog".

## FOOTER

```
<Section>
  <div>
    <nav>
      <a href="index.php">Home</a>
      <a href="nosotros.php">Nosotros</a>
      <a href="">Anuncios</a>
      <a href="contacto.php">Contacto</a>
      <a href="blog.php">BLOG</a>
    </nav>
  </div>
  <p>Todos los derechos reservados by: ME</p>
</Section>
</body>
</html>
```

Este código HTML define una sección de pie de página para una página web. Utiliza la etiqueta `<section>` (aunque está mal escrita en mayúsculas como `<Section>`, lo que no es estándar en HTML). Dentro de esta sección, hay un `<div>` que contiene un elemento de navegación (`<nav>`) con varios enlaces (`<a>`). Estos enlaces permiten al usuario navegar entre diferentes secciones del sitio, como "Home", "Nosotros", "Anuncios", "Contacto" y "BLOG".

Fuera del `<div>`, hay un párrafo (`<p>`) con el texto "Todos los derechos reservados by: ME", que indica los derechos de autor del contenido. El archivo termina cerrando las etiquetas `<section>`, `<body>`, y `<html>`.



# ÍNDEX

[illegible]

Este código es una estructura HTML para una página de bienes raíces, donde se muestran propiedades en venta, información sobre la empresa y un blog. La página comienza con un encabezado que contiene una navegación básica con enlaces a secciones como "Nosotros", "Anuncios", "Contacto" y "Blog", seguido de un subtítulo promocionando la venta de propiedades exclusivas. Luego, se presenta una sección titulada "Más sobre nosotros" que incluye varios íconos SVG con descripciones de los servicios o características de la empresa. El cuerpo principal, o "main", contiene un listado de propiedades en venta. Cada propiedad muestra un título, el precio, una breve descripción, y una serie de íconos que representan la cantidad de baños, garajes y habitaciones, además de un enlace para ver más detalles de cada propiedad. Posteriormente, hay una sección que invita a los usuarios a "Encontrar la casa de sus sueños", seguida de un enlace para ponerse en contacto. Se incluye también una sección de blog que contiene artículos con títulos y breves descripciones sobre temas de decoración y mejoras del hogar. Al final, se añade una sección de testimonios y un pie de página con navegación adicional, derechos de autor y créditos.

## NOSOTROS

```
<?php include "includes/header.php" ?>
<section>
  <h2>Conoces sobre nosotros</h2>
  <!-- imagen -->
  <p>25 years de experiencia</p>
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nam aliquam, corporis vitae obcaecati vel debitis qui, ex atque esse, natus itaque sed asperiores rep
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Possimus velit saepe est dolore nemo asperiores at queraat esse</p>
</section>

<section>
  <h2> Mas sobre nosotros</h2>
  <div>
    <svg xmlns="http://www.w3.org/2000/svg" class="icon icon-tabler icon-tabler-lock-cancel" width="80" height="80" viewBox="0 0 24 24" stroke-width="1.5" stroke=
    <path stroke="none" d="M0 0h24v24H0z" fill="none"/>
    <path d="M12.5 21h-5.5a2 2 0 0 1 -2 -2v-6a2 2 0 0 1 2 -2h10a2 2 0 0 1 1.749 1.028" />
    <path d="M11 16a1 1 0 1 0 2 0a1 1 0 0 0 -2 0" />
    <path d="M8 11v-4a4 4 0 1 1 8 0v4" />
    <path d="M19 19m-3 0a3 3 0 1 0 6 0a3 3 0 1 0 -6 0" />
    <path d="M17 21l4 -4" />
  </svg>
  <p>Lorem aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</p>
</div>
  <div>
    <svg xmlns="http://www.w3.org/2000/svg" class="icon icon-tabler icon-tabler-brand-cashapp" width="80" height="80" viewBox="0 0 24 24" stroke-width="1.5" strok
    <path stroke="none" d="M0 0h24v24H0z" fill="none"/>
    <path d="M17.1 8.648a.568 .568 0 0 1 -.761 .011a5.682 5.682 0 0 0 -3.659 -1.34c-1.102 0 -2.205 .363 -2.205 1.374c0 1.023 1.182 1.364 2.546 1.875c2.386 .79
  </svg>
  <p>Lorem aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</p>
</div>
  <div>
    <svg xmlns="http://www.w3.org/2000/svg" class="icon icon-tabler icon-tabler-clock-hour-1" width="80" height="80" viewBox="0 0 24 24" stroke-width="1.5" stroke
    <path stroke="none" d="M0 0h24v24H0z" fill="none"/>
    <path d="M12 12m-9 0a9 9 0 1 0 18 0a9 9 0 1 0 -18 0" />
    <path d="M12 7v5" />
    <path d="M12 12l2 -3" />
  </svg>
  <p>Lorem aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</p>
</div>
```

Incluyen información general sobre sus 25 años de experiencia, una breve descripción y una sección adicional de "Más sobre nosotros". En esta segunda sección, se utilizan iconos SVG para resaltar características o servicios clave de la empresa, con una breve descripción junto a cada icono. El código también incluye un encabezado y un pie de página mediante archivos externos, integrados con include, permitiendo una estructura modular y fácil mantenimiento de los componentes de la página.

## CONTACTO

```
<?php include "includes/header.php" ?>
<section>
  <h2>Contacto</h2>
  <!-- imagen -->
</section>
<section>
  <h2>Llena el formulario de contacto</h2>
  <div>
    <form action="">
      <fieldset>
        <legend>INFORMACION PERSONAL</legend>
        <div>
          <label for="nombre">Nombre:</label>
          <input type="text" name="nombre" id="nombre" placeholder="Your name">
        </div>
        <div>
          <label for="email">email:</label>
          <input type="text" name="email" id="email" placeholder="YourEmail@gmail.com">
        </div>
        <div>
          <label for="Phone">Phone:</label>
          <input type="text" name="Phone" id="Phone" placeholder="555 5 5555 55">
        </div>
        <div>
          <label for="msg">Message:</label>
          <input type="text" name="msg" id="msg" placeholder="Your Message">
        </div>
      </fieldset>
      <fieldset>
        <legend>INFORMACION DE LA PROPIEDAD</legend>
        <div>
          <label for="vencom">Vende o compra:</label>
          <input type="select" name="vencom" id="vencom">
        </div>
      </fieldset>
    </form>
  </div>
</section>
```

Formulario para que los usuarios envíen su información personal y detalles de una propiedad que desean comprar o vender. La página, estructurada en secciones, presenta un encabezado y un pie de página que se cargan mediante include. El formulario de contacto está organizado en tres fieldset principales: el primero recopila datos personales como nombre, correo electrónico y mensaje; el segundo solicita información sobre la propiedad, como si el usuario quiere comprar o vender y la cantidad; y el tercero permite seleccionar el método de contacto preferido (teléfono o correo electrónico) e incluye campos para definir la fecha y hora de contacto. Al final del formulario, hay un botón para enviar la información.

## BLOG

```
<?php include "includes/header.php" ?>

<section>
    <h2>Nuestro blog</h2>
    <div>
        <h2>Terraza en el techo de tu casa</h2>
        <!-- imagen -->
        <p>Escrito "fecha" by: "autor"</p>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure quos reiciendis</p>
    </div>
    <div>
        <h2>GUIA para decorar tu hogar</h2>
        <!-- imagen -->
        <p>Escrito "fecha" by: "autor"</p>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure quos reiciendis</p>
    </div>
    <div>
        <h2>GUIA PARA DECORAR TU HABITACION</h2>
        <!-- imagen -->
        <p>Escrito "fecha" by: "autor"</p>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure quos reiciendis</p>
    </div>
</section>
<php
<?php include "includes/footer.php" ?>
```

Al iniciar, utiliza include para cargar un encabezado desde otro archivo PHP (header.php). La sección central muestra tres artículos sobre temas de decoración y mejoras para el hogar, cada uno con un título, un marcador de fecha y autor (actualmente como texto de marcador), una breve descripción y un espacio comentado para insertar una imagen. Finalmente, se incluye un pie de página desde footer.php, concluyendo la estructura de la página.

## CONEXIÓN A BASE DE DATOS

```
<?php

function conectar(): mysqli {
    $db = mysqli_connect("localhost","root","","Bienesraces");

    if ($db) {
        echo 'conectado';
    }else{
        echo 'no conectado';
    }
    return $db;
}

?>
```

Este código define una función llamada conectar, que establece una conexión a una base de datos MySQL llamada Bienesraciones en un servidor local (localhost) usando las credenciales root como usuario y una contraseña vacía. Si la conexión es exitosa, muestra el mensaje "conectado"; de lo contrario, muestra "no conectado". La función devuelve el objeto de conexión a la base de datos (\$db), permitiendo su reutilización en otras partes del código para interactuar con la base de datos.

## CREAR SELLER

```
<section>
  <h2>Seller Form</h2>
  <div>
    <form action="crearSeller.php" method="post">
      <fieldset>
        <legend>Fill all fillds to create a new seller</legend>
        <div>
          <label for="name">Name:</label>
          <input type="text" id="name" name="name" placeholder="Your name">
        </div>
        <div>
          <label for="email">Email:</label>
          <input type="email" id="email" name="email" placeholder="Your@email.com">
        </div>
        <div>
          <label for="phone">Phone:</label>
          <input type="text" id="phone" name="phone" placeholder="555 5 5555 55">
        </div>
        <div>
          <button type="submit">CREATE A NEW SELLER</button>
        </div>
      </fieldset>
    </form>
  </div>
</section>
```

formulario para registrar un nuevo vendedor en la base de datos. Comienza incluyendo un archivo header.php y un archivo de configuración de la conexión (connectDB.php). Luego, establece una conexión llamando a la función conectar(). Después de recibir los datos enviados desde el formulario (name, email, phone) mediante el método POST, se asignan a variables y se utiliza una consulta SQL de inserción para añadir estos valores a la tabla seller en la base de datos. Si la consulta se ejecuta correctamente, muestra el mensaje "Seller creado"; de lo contrario, muestra "Seller no creado". Al final, se encuentra un formulario HTML con campos de entrada para el nombre, correo electrónico y teléfono del vendedor, y un botón para enviar los datos y crear un nuevo vendedor. La página concluye con la inclusión de footer.php.

## CREAR PROPIERTY

```
<section>
  <h2>Propierties Form</h2>
  <div>
    <form action="crearPropiedad.php" method="post">
      <fieldset>
        <legend>Fill all fillds to create a new Propierty</legend>
        <div>
          <label for="title">Title:</label>
          <input required type="text" id="title" name="title" placeholder="Title for Propierty">
        </div>
        <div>
          <label for="price">Price:</label>
          <input required type="number" id="price" name="price" placeholder="1111111111">
        </div>
        <div>
          <label for="image">Image:</label>
          <input type="image" id="image" name="image">
        </div>
        <div>
          <label for="description">Description:</label>
          <textarea id="description" name="description" placeholder="Description for Propierty"></textarea>
        </div>
        <div>
          <label for="room">Rooms:</label>
          <input type="number" id="room" name="room">
        </div>
        <div>
          <label for="wc">WC:</label>
          <input type="number" id="wc" name="wc">
        </div>
        <div>
          <label for="garage">Garage:</label>
          <input type="number" id="garage" name="garage" >
        </div>
      </fieldset>
    </form>
  </div>
</section>
```

formulario para crear una nueva propiedad en una base de datos. Al inicio, incluye archivos externos para la cabecera y la configuración de conexión a la base de datos. Luego, recibe los datos del formulario enviados mediante el método POST, asignando cada campo (title, price, image, description, room, wc, garage, timeStamp, seller) a variables y ejecuta una consulta SQL de inserción para añadir estos valores a la tabla property. Si la inserción es exitosa, muestra el mensaje "Propierty creada"; de lo contrario, indica que no se creó. Posteriormente, ejecuta una consulta para recuperar los nombres de los vendedores desde la tabla seller, mostrando cada nombre en el formulario como opciones dentro de un campo select. La estructura HTML incluye campos de entrada para cada atributo de la propiedad, un campo de selección para elegir un vendedor, y un botón de envío. Finalmente, se cierra la página con un pie de página (footer.php).