

# Segundo Avance del Portafolio

## *Desarrollo de Aplicaciones Web*

Monarrez Barron, Polo Alejandro.

## Contenido

Superglobales \$GET y \$POST en PHP .....	3
\$GET .....	3
Características del método GET .....	3
\$POST .....	4
Características del método POST .....	4
Conclusión .....	4
Conectar MySQL y PHP .....	5
Pasos para conectar PHP y MySQL .....	5
Resumen .....	10
Documentación del código .....	11
Header.php .....	11
Footer.php .....	12
Index.php .....	13
Nosotros.php .....	14
Anuncios.php .....	16
Contacto.php .....	18
Blog.php .....	21
Sellers.php .....	23
Propiedades.php .....	25
Resumen .....	27

## Superglobales \$GET y \$POST en PHP

Los superglobales en PHP son variables predefinidas que están disponibles en todos los ámbitos (de manera global, local, dentro de funciones y métodos). Entre ellas, las superglobales \$GET y \$POST son muy utilizadas para manejar datos enviados desde el formulario HTML u otros medios, como parámetros de URL.

### ***\$GET***

La superglobal \$GET es una matriz asociativa que recoge los datos enviados mediante el método GET. Este método es usado principalmente cuando se desea enviar información a través de la URL, por ejemplo, al navegar a una página con parámetros como:

- <https://ejemplo.com/pagina.php?nombre=Polo&Edad=25>

Cada parámetro se envía en formato clave-valor. En el ejemplo, nombre y edad son las claves, y sus valores son Juan y 25.

En PHP, puedes acceder a estos datos de la siguiente forma:

- `$nombre = $_GET['nombre'];`
- `$edad = $GET['edad'];`

### ***Características del método GET***

- **Visible:** Los datos se envían a través de la URL, por lo que no son adecuados para información sensible.
- **Límite de datos:** Tiene una limitación en cuanto a la cantidad de datos que se pueden enviar (generalmente alrededor de 2048 caracteres).
- **Reutilizable:** Las URL generadas con GET pueden ser guardadas como marcadores o compartidas.

## ***\$POST***

La superglobal `$POST` también es una matriz asociativa que recoge los datos, pero enviados mediante el método POST. A diferencia de GET, estos datos no se envían a través de la URL, sino como parte del cuerpo de la solicitud HTTP, lo que los hace más adecuados para información confidencial. Por ejemplo, un formulario enviado mediante POST:

- `<form action="procesar.php" method="POST">`
- `<input type="text" name="usuario" />`
- `<input type="password" name="clave" />`
- `<input type="submit" value="Enviar" />`
- `</form>`

En PHP, se puede acceder a los datos de la siguiente manera:

- `$usuario = $_POST['usuario'];`
- `$clave = $_POST['clave'];`

## ***Características del método POST***

- **No visible:** Los datos no se muestran en la URL, lo que permite enviar información sensible como contraseñas.
- **Sin límite de datos:** No está limitado por el tamaño de la URL, por lo que es adecuado para enviar grandes cantidades de datos o archivos.
- **No reutilizable:** Al no ser visible en la URL, no se puede reutilizar directamente con un marcador.

## ***Conclusión***

Las superglobales `$GET` y `$POST` en PHP juegan un papel fundamental en la transmisión de datos entre el cliente (navegador) y el servidor. Cada una tiene su propósito específico, siendo `$GET` adecuada para consultas y `$POST` para operaciones más seguras que requieren el envío de información más sensible o voluminoso.

# Conectar MySQL y PHP

Conectar MySQL y PHP es una tarea común en el desarrollo web, especialmente cuando se trabaja con aplicaciones que requieren bases de datos. MySQL es un sistema de gestión de bases de datos relacional muy popular, mientras que PHP es un lenguaje de programación de propósito general que a menudo se usa para el desarrollo de aplicaciones web dinámicas. A continuación, te ofrezco un resumen detallado sobre cómo conectar PHP con MySQL.

## *Pasos para conectar PHP y MySQL*

### 1. Preparar el entorno

- **Servidor Web:** Se debe tener instalado un servidor web como Apache o Nginx.
- **Base de datos MySQL:** Debes tener MySQL instalado y configurado. También es necesario que tengas acceso a una base de datos.
- **PHP:** Asegúrate de que PHP está correctamente instalado y configurado para interactuar con MySQL.

### 2. Extensiones de PHP para MySQL

PHP ofrece varias extensiones para interactuar con MySQL. Las más comunes son:

- **MySQLi (MySQL Improved):** Una versión mejorada de la extensión original de MySQL, que permite una interacción más segura y eficiente con la base de datos. Es compatible con consultas preparadas y otras características modernas.
- **PDO (PHP Data Objects):** Una interfaz de acceso a bases de datos que soporta múltiples motores de bases de datos, incluyendo MySQL. Ofrece un enfoque orientado a objetos y es una opción versátil cuando se trabaja con varias bases de datos.

### 3. Conectar con MySQL usando MySQLi

#### Conexión básica:

```
<?php

$servername = "localhost";

$username = "usuario";

$password = "contraseña";

$dbname = "nombre_base_datos";

// Crear la conexión

$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar la conexión

if ($conn->connect_error) {

    die("Conexión fallida: " . $conn->connect_error);

}

echo "Conexión exitosa";

?>
```

En este ejemplo, se establece una conexión con MySQL utilizando la extensión MySQLi. Si la conexión falla, se mostrará un mensaje de error.

#### Cerrar la conexión:

```
$conn->close();
```

### Consultar la base de datos:

php

Copiar código

```
$sql = "SELECT * FROM usuarios";

$result = $conn->query($sql);

if ($result->num_rows > 0) {

    while($row = $result->fetch_assoc()) {

        echo "id: " . $row["id"]. " - Nombre: " . $row["nombre"]. "<br>"; }

    } else {

        echo "0 resultados";

    }
```

#### 4. Conectar con MySQL usando PDO

PDO permite trabajar de manera más versátil, especialmente si necesitas cambiar de base de datos en el futuro (por ejemplo, de MySQL a PostgreSQL). También soporta transacciones y consultas preparadas.

##### Conexión básica:

```
<?php

$servername = "localhost";

$username = "usuario";

$password = "contraseña";

$dbname = "nombre_base_datos";

try {

    $conn = new PDO("mysql:host=$servername;dbname=$dbname",
    $username, $password);

    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); echo
    "Conexión exitosa";

} catch(PDOException $e) {

    echo "Conexión fallida: " . $e->getMessage();

}

?>
```

Cerrar la conexión: PDO no requiere una función explícita para cerrar la conexión, ya que cuando el script finaliza, la conexión se cierra automáticamente.



### Consultar la base de datos con PDO:

php

Copiar código

```
$sql = "SELECT * FROM usuarios";

$stmt = $conn->prepare($sql);

$stmt->execute();

// Establecer el modo de obtención de resultados

$stmt->setFetchMode(PDO::FETCH_ASSOC);

foreach ($stmt->fetchAll() as $row) {

    echo "id: " . $row["id"] . " - Nombre: " . $row["nombre"] . "<br>";

}
```

## 5. Seguridad: Evitar la inyección SQL

Es fundamental proteger la aplicación contra inyecciones SQL, un ataque común en aplicaciones web que no sanitizan correctamente las entradas del usuario. Se recomienda el uso de consultas preparadas, las cuales separan los datos de las consultas SQL, previniendo ataques de inyección.

### Ejemplo con MySQLi (consultas preparadas):

```
$stmt = $conn->prepare("SELECT * FROM usuarios WHERE id = ?"); $stmt->bind_param("i", $id);  
  
$stmt->execute();  
  
$result = $stmt->get_result();
```

### Ejemplo con PDO (consultas preparadas):

```
$stmt = $conn->prepare("SELECT * FROM usuarios WHERE id = :id"); $stmt->bindParam(':id', $id);  
  
$stmt->execute();
```

## Resumen

MySQLi y PDO son las dos opciones más comunes para conectar PHP con MySQL. MySQLi es específico de MySQL, mientras que PDO es más flexible y compatible con otros motores de bases de datos.

La conexión básica implica proporcionar el nombre del servidor, el usuario, la contraseña y el nombre de la base de datos.

La seguridad es clave: siempre usa consultas preparadas para evitar la inyección SQL.

Cerrar la conexión es una buena práctica, aunque en PDO no es estrictamente necesario.

# Documentación del código

## Header.php

```
<section>
  <h1>BIENES RAICES</h1>
  <header>
    <nav>
      <a href="index.php">Home</a>
      <a href="nosotros.php">Nosotros</a>
      <a href="anuncios.php">Anuncios</a>
      <a href="contacto.php">Contacto</a>
      <a href="blog.php">Blog</a>
    </nav>
    <h2>Venta de Casas y Departamentos<br>Exclusivos de Lujo</h2>
  </header>
</section>
```

Este código es un componente fundamental de la página web de bienes raíces, diseñado para ser claro y accesible. Facilita la navegación del usuario y proporciona información relevante sobre la venta de propiedades exclusivas, mejorando la experiencia general del usuario en el sitio web.

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

**Descripción:** Esta línea usa PHP para incluir un archivo externo llamado **header.php**. Esto permite mantener la estructura modular del sitio web, haciendo que el encabezado se cargue automáticamente en todas las páginas que lo requieran, facilitando la gestión del código y cambios futuros.

## Footer.php

```
<section>
  <div>
    <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>
  </div>
  <p>Todos los Derechos Reservados by: Yo</p>
</section>
```

Este código es un componente básico de un sitio web, diseñado para facilitar la navegación y asegurar que se reconozcan los derechos de autor. Es esencial en la creación de una interfaz de usuario amigable y en la presentación de información de manera organizada y accesible. La inclusión de un menú de navegación mejora la experiencia del usuario, permitiendo un acceso rápido a las diferentes áreas del sitio.

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

**Descripción:** Este código PHP incluye el archivo de pie de página en el documento, permitiendo que el pie de página (footer) sea modular y esté disponible en varias páginas del sitio web sin tener que duplicar el código en cada archivo.

## Index.php

```
<?php include "includes/header.php"; ?>
<section>
  <h2>Encuentra la casa de tus sueños</h2>
  <p>
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. Repudiandae, quibusdam
    porro esse voluptatem sunt repellat laudantium cupiditate vel autem nobis doloremque
    necessitatibus nostrum ipsum harum dolorem, facere blanditiis neque excepturi!
  </p>
</section>
<?php include "includes/footer.php" ?>
```

Este código HTML representa una sección de una página web destinada a ayudar a los usuarios a encontrar la casa de sus sueños.

- **Elementos Estructurales:** La sección está definida por la etiqueta <section>, que se utiliza para agrupar contenido relacionado dentro del documento HTML, facilitando así la organización y la accesibilidad del contenido.
- **Título:** Dentro de la sección, se incluye un encabezado <h2>, que presenta el título "Encuentra la casa de tus sueños". Este encabezado se destaca como un elemento importante, informando al usuario sobre el propósito de la sección y mejorando la navegación en la página.
- **Descripción:** A continuación, hay un párrafo <p> que proporciona un texto descriptivo. El texto utiliza un marcador de posición de "Lorem ipsum", que es un texto estándar en la industria del diseño gráfico y desarrollo web. Este texto es utilizado para ilustrar cómo se verá el contenido final en esa área de la sección.

## Nosotros.php

```
<section>
  <h2>Conoce sobre nosotros</h2>
  <!--Imagen-->
  <p>25 years of experience</p>
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Voluptas suscipit
facilis aspernatur iusto quasi deleniti omnis excepturi tempore. Sit adipisci quis
tenetur consectetur culpa voluptatibus tempore recusandae voluptates consequatur
repellat?</p>
  <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Rerum cupiditate
laboriosam fuga </p>
</section>
```

Este código HTML presenta dos secciones distintas que brindan información sobre una empresa o entidad. La estructura está organizada en dos `<section>` para mejorar la legibilidad y la separación de contenido. A continuación, se detallan los elementos clave de cada sección:

### Primera Sección: "Conoce sobre nosotros"

- **Encabezado:** Se utiliza un `<h2>` para titular la sección, haciendo hincapié en la introducción de la entidad.
- **Imagen (placeholder):** Se menciona una imagen que podría representar la identidad de la empresa (aunque no está implementada en el código).
- **Texto Descriptivo:** Incluye párrafos que destacan la experiencia de la entidad (25 años) y ofrecen una breve descripción de su misión o filosofía. El texto está en formato de párrafo (`<p>`), proporcionando una lectura fluida.

```

<section>
  <h2>Sobre nosotros</h2>
  <div>
    <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="orange"
class="bi bi-lock" viewBox="0 0 16 16">...</svg>
    <h2>Seguridad</h2>
    <p>Lorem ipsum, dolor... sit amet consectetur adipisicing elit. Ducimus
laboriosam vitae, nulla suscipit doloremque voluptatum quasi fuga alias id magni,
asperiores dolorem beatae dolor officiis! Cum nobis fuga provident cupiditate?</p>
  </div>
  <div>
    <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="orange"
class="bi bi-currency-dollar" viewBox="0 0 16 16">...</svg>
    <h2>Precio</h2>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Ducimus laboriosam
vitae, nulla suscipit doloremque voluptatum quasi fuga alias id magni, asperiores dolorem
beatae dolor officiis! Cum nobis fuga provident cupiditate?</p>
  </div>
  <div>
    <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="orange"
class="bi bi-clock-history" viewBox="0 0 16 16">...</svg>
    <h2>Tiempo</h2>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Ducimus laboriosam
vitae, nulla suscipit doloremque voluptatum quasi fuga alias id magni, asperiores dolorem
beatae dolor officiis! Cum nobis fuga provident cupiditate?</p>
  </div>
</section>

```

## Segunda Sección: "Sobre nosotros"

- **Encabezado:** Otro <h2> indica el tema de esta sección.
- **Divisiones de Contenido:** Esta sección se divide en tres bloques (<div>), cada uno de los cuales presenta un aspecto relevante de la empresa, complementado por un icono SVG que refuerza visualmente el tema de cada bloque.
  - **Seguridad:** Incluye un ícono de candado, acompañado de un título y una descripción breve que resalta la seguridad como un valor central de la empresa.
  - **Precio:** Presenta un ícono de dólar y proporciona información sobre la política de precios de la empresa.
  - **Tiempo:** Utiliza un ícono de reloj y describe la importancia del tiempo en las operaciones de la entidad.

Cada bloque sigue un formato similar: un ícono que complementa el título y una breve descripción que proporciona detalles adicionales.

## Anuncios.php

```
<section>
  <h2>Casas y Departamentos en Venta</h2>
  <div>
    <h3>Casa de lujo en el Lago</h3>
    <p>...</p>
    $111,111,111
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-router" viewBox="0 0 16 16">...</svg>
    3
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-car-front" viewBox="0 0 16 16">...</svg>
    2
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-moon" viewBox="0 0 16 16">...</svg>
    3
    <a href="">Ver Propiedad</a>
  </div>
  <div>
    <h3>Casa Terminados de Lujo</h3>
    <p>...</p>
    $222,111,111
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-router" viewBox="0 0 16 16">...</svg>
    4
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-car-front" viewBox="0 0 16 16">...</svg>
    3
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-moon" viewBox="0 0 16 16">...</svg>
    5
    <a href="">Ver Propiedad</a>
  </div>
  <div>
    <h3>Casa con Alberca</h3>
    <p>...</p>
    $333,111,111
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-router" viewBox="0 0 16 16">...</svg>
    3
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-car-front" viewBox="0 0 16 16">...</svg>
    4
    <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" fill="orange"
class="bi bi-moon" viewBox="0 0 16 16">...</svg>
    10
    <a href="">Ver Propiedad</a>
  </div>
</section>
```



Este fragmento de código HTML presenta una sección de un portafolio que exhibe propiedades en venta, específicamente casas y departamentos de lujo. La estructura general está diseñada para ser clara y fácil de navegar, mostrando detalles relevantes sobre cada propiedad.

### Estructura del Documento:

- **Encabezado Principal (<h2>):** Indica el tema general de la sección: "Casas y Departamentos en Venta".
- **Divisiones de Propiedades (<div>):** Cada propiedad se encuentra dentro de su propia división, permitiendo que la información de cada una esté bien organizada y separada.

### Elementos de Cada Propiedad:

1. **Título de la Propiedad (<h3>):** Indica el nombre o tipo de la propiedad, como "Casa de lujo en el Lago" o "Casa con Alberca".
2. **Descripción (<p>):** Un párrafo que proporciona detalles generales sobre la propiedad. Actualmente contiene texto de marcador de posición (Lorem ipsum), lo que sugiere que se debe reemplazar con descripciones reales.
3. **Precio:** Se muestra el precio de la propiedad en formato monetario.
4. **Íconos de Características:** Se utilizan íconos SVG para ilustrar características específicas de la propiedad. Estos íconos incluyen:
  - **Router:** Indica la cantidad de habitaciones.
  - **Carro:** Representa la cantidad de estacionamientos disponibles.
  - **Luna:** Indica la cantidad de habitaciones o características adicionales.
5. **Enlace de Visualización:** Cada propiedad incluye un enlace para ver más detalles, con el texto "Ver Propiedad", que invita al usuario a acceder a una página con información más completa.

## Contacto.php

```

<section>
    <h2>Contacto</h2>
</section>
<section>
    <h2>Llene el formulario de contacto</h2>
    <div>
        <form action="">
            <fieldset>
                <legend>Informacion personal</legend>
                <div>
                    <label for="name">Name:</label>
                    <input type="text" name="name" id="name" placeholder="Your
Name">
                </div>
                <div>
                    <label for="email">e-mail:</label>
                    <input type="text" name="email" id="email"
placeholder="your@email.com">
                </div>
                <div>
                    <label for="phone">Phone:</label>
                    <input type="text" name="phone" id="phone" placeholder="555
555 5555">
                </div>
                <div>
                    <label for="msg">Message:</label>
                    <textarea type="text" name="msg" id="msg" placeholder="Your
message"></textarea>
                </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>
                <div>
                    <label for="quantity">Quantity</label>
                    <input type="number" name="quantity" id="quantity">
                </div>
            </fieldset>

            <fieldset>
                <legend>Contacto</legend>
                <div>
                    <label for="contactForm">Como desea ser contactado</label>
                    <label for="tel">Phone</label>

                    <input type="radio" name="tel" id="tel">
                    <label for="e-mail">E-mail</label>
                    <input type="radio" name="e-mail" id="e-mail">
                </div>
                <div>
                    <label for="date">Date</label>
                    <input type="date" name="date" id="date">
                </div>
            </fieldset>
        </form>
    </div>
</section>

```

```

        <div>
          <label for="time">Time</label>
          <input type="time" name="time" id="time">
        </div>
      </fieldset>
    <div>
      <button>Contactame</button>
    </div>
  </form>
</div>
</section>

```

Este código HTML define una sección de contacto que incluye un formulario para que los usuarios envíen su información personal y detalles relacionados con una propiedad. La estructura del formulario está organizada en secciones, cada una con su propio conjunto de campos, facilitando la recopilación de información relevante de manera clara y ordenada.

## Estructura del Documento

### 1. Sección de Contacto:

- **Título:** "Contacto", que sirve como encabezado para la sección.
- **Imagen:** Se menciona un lugar para una imagen (aunque no se incluye en el código), que podría ser un logo o una representación visual relacionada con la empresa o el servicio.

### 2. Sección de Formulario:

- **Título:** "Llene el formulario de contacto", que instruye al usuario sobre la acción que se debe tomar.
- **Formulario:** Contiene varios fieldset, que agrupan campos relacionados para mejorar la usabilidad.

## Campos del Formulario

### • Información Personal:

- **Nombre:** Campo de texto para que el usuario introduzca su nombre.
- **Correo Electrónico:** Campo de texto para la dirección de correo electrónico.

- **Teléfono:** Campo de texto para el número de teléfono.
- **Mensaje:** Área de texto donde el usuario puede escribir un mensaje.
- **Información de la Propiedad:**
  - **Vende o Compra:** Etiqueta que indica un campo para seleccionar si el usuario está interesado en vender o comprar una propiedad. Sin embargo, el tipo select debería cambiarse a un tipo adecuado, como radio o dropdown.
  - **Cantidad:** Campo numérico para que el usuario introduzca una cantidad relacionada con la propiedad.
- **Opciones de Contacto:**
  - **Método de Contacto:** Pregunta sobre cómo el usuario desea ser contactado, con opciones de teléfono y correo electrónico (aunque los botones de opción no están completamente definidos).
  - **Fecha:** Campo de fecha para seleccionar un día específico.
  - **Hora:** Campo de tiempo para seleccionar una hora específica.

### Botón de Envío

- **Botón "Contáctame":** Un botón que, al hacer clic, debería enviar el formulario. Actualmente, la acción del formulario está vacía (`action=""`), lo que significa que no se especifica dónde se enviarán los datos.

## Blog.php

```
<section>
  <h2>Nuestro Blog</h2>
  <div>
    <h3>Terraza en el techo de tu casa</h3>
    <p>Escrito el: "fecha", by: "user"</p>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Harum, nam.</p>
  </div>
  <div>
    <h3>Construye una alberca en tu hogar</h3>
    <p>Escrito el: "fecha", by: "user"</p>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Harum, nam.</p>
  </div>
  <div>
    <h3>Guia para decorar tu hogar</h3>
    <p>Escrito el: "fecha", by: "user"</p>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Harum, nam.</p>
  </div>
  <div>
    <h3>Guia para decorar tu habitacion</h3>
    <p>Escrito el: "fecha", by: "user"</p>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Harum, nam.</p>
  </div>
</section>
```

Este código HTML representa una sección de un sitio web dedicada a un blog, donde se presentan diferentes publicaciones relacionadas con mejoras y decoración del hogar. A continuación, se detallan los componentes y la estructura del código:

### 1. Estructura del Documento:

- El contenido está contenido dentro de una etiqueta `<section>`, que se utiliza para agrupar un conjunto de elementos relacionados, en este caso, las entradas del blog.
- El encabezado principal de la sección está marcado con un `<h2>`, titulado "Nuestro Blog", que indica el propósito de esta parte del contenido.

### 2. Entradas del Blog:

- Cada entrada del blog se organiza en un `<div>`, lo que permite una separación clara entre las diferentes publicaciones.
- Dentro de cada `<div>`, se utiliza un `<h3>` para el título de la publicación, proporcionando un formato claro y jerárquico que ayuda a los lectores a identificar rápidamente los temas.

- La línea que sigue al título incluye información sobre la fecha de publicación y el autor, usando elementos de párrafo `<p>`. Actualmente, los valores "fecha" y "user" son marcadores de posición que deberían ser reemplazados por datos dinámicos o estáticos en una implementación real.
- Finalmente, cada entrada incluye un segundo párrafo `<p>` que contiene un breve resumen o descripción de la publicación, utilizando texto de ejemplo "Lorem ipsum", que es comúnmente usado como texto de marcador.

### 3. Elementos Visuales:

- El código incluye comentarios que indican la ubicación de imágenes relacionadas con cada entrada, lo que sugiere que se espera que se incluyan imágenes visuales para mejorar la presentación y el atractivo de las publicaciones.

## Sellers.php

```

<?php
    require "../proyecto1/includes/config/connectdb.php";
    $db = connectdb();
    var_dump($_POST);
    $name = $_POST["name"];
    $email = $_POST["email"];
    $phone = $_POST["phone"];
    $query = "INSERT INTO sellers(name,email,phone) VALUES('$name','$email','$phone')";
    $response = mysqli_query($db, $query);
    if ($response) {
        echo "Seller created";
    }
    else{
        echo "Seller no created";
    }
}
?>

<section>
    <h2>Sellers Form</h2>
    <div>
        <form action="crearSellers.php" method="POST">
            <fieldset>
                <legend>Fill all Fields 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">E-Mail</label>
                    <input type="text" id="email" name="email"
placeholder="your@email.com">
                </div>
                <div>
                    <label for="phone">Phone</label>
                    <input type="text" id="phone" name="phone" placeholder="555-555-
5555">
                </div>
                <div>
                    <button type="submit">Create a New Seller</button>
                </div>
            </fieldset>
        </form>
    </div>
</section>

```

Este código PHP implementa un formulario para crear un nuevo vendedor en una base de datos. Primero, establece una conexión a la base de datos mediante el archivo de configuración `connectdb.php`. Luego, captura los datos del formulario (nombre, correo electrónico y teléfono) enviados a través del método POST. A continuación, construye una consulta SQL para insertar los datos en la tabla `sellers`. Si la ejecución de la consulta es exitosa, se muestra un mensaje de confirmación; de lo contrario, se notifica que la creación del vendedor ha fallado. Además, el código incluye un formulario HTML que permite al usuario ingresar la información necesaria para crear un nuevo vendedor, con campos para el nombre, el correo electrónico y el teléfono.



## Propiedades.php

```

<?php include "includes/header.php";
require "../proyecto1/includes/config/connectdb.php";
$db = connectdb();
echo "<br>";
$querySellers = "SELECT id, name FROM sellers;";
$responseSellers = mysqli_query($db, $querySellers);
if($_POST){
    $title = $_POST["title"];
    $price = $_POST["price"];
    $image = $_POST["image"];
    $description = $_POST["description"];
    $rooms = $_POST["rooms"];
    $wc = $_POST["wc"];
    $garage = $_POST["garage"];
    $timestamp = $_POST["timestamp"];
    $seller = $_POST["seller"];
    $query = "INSERT INTO
propiedades(title,price,image,description,rooms,wc,garage,timestamp,id_seller)
VALUES('$title','$price','$image','$description','$rooms','$wc','$garage','$timestamp','$
seller')";
    $response = mysqli_query($db, $query);
    if ($response) {
        echo "Propierty created";
    }
    else{
        echo "Propierty NO created";
    }
}
?>
<section>
    <h2>Propierties Form</h2>
    <div>
        <form action="crearPropiedades.php" method="POST">
            <fieldset>
                <legend>Fill all your form fields to create a new property</legend>
                <div>
                    <label for="title">Title</label>
                    <input type="text" id="title" name="title" placeholder="Propierty
Title" required>
                </div>
                <div>
                    <label for="price">Price</label>
                    <input type="number" id="price" name="price" placeholder="$1000000"
required min="0">
                </div>
                <div>
                    <label for="image">Image URL</label>
                    <input type="text" id="image" name="image"
placeholder="imageURL.com/image.png">
                </div>
                <div>
                    <label for="description">Description</label>
                    <textarea id="description" name="description" placeholder="Propierty
Title"></textarea>
                </div>
                <div>
                    <label for="rooms">Rooms</label>

```

```

        <input type="number" id="rooms" name="rooms" placeholder="5">
    </div>
    <div>
        <label for="wc">wc</label>
        <input type="number" id="wc" name="wc" placeholder="3">
    </div>
    <div>
        <label for="garage">Garage</label>
        <input type="number" id="garage" name="garage" placeholder="2">
    </div>
    <div>
        <label for="timestamp">TimeStamp</label>
        <input type="date" id="timestamp" name="timestamp">
    </div>
    <div>
        <label for="seller">Seller</label>
        <select name="seller" id="seller">
            <?php while ($sellerID = mysqli_fetch_assoc($responseSellers)) {
?>
                <option value="<?php echo $sellerID["id"]; ?>"> <?php echo
$sellerID["name"] ?> </option>
                <?php } ?>
            </select>
        </div>
    <div>
        <button type="submit">Create a New Propierty</button>
    </div>
</fieldset>
</form>
</div>
</section>

```

El siguiente código PHP implementa un formulario para la creación de nuevas propiedades en una base de datos. Primero, se establece una conexión a la base de datos mediante un archivo de configuración. Luego, se realiza una consulta para obtener la lista de vendedores disponibles, la cual se utiliza para llenar un menú desplegable en el formulario. Cuando se envía el formulario (mediante el método POST), se recogen los datos ingresados, como el título, precio, imagen, descripción, número de habitaciones, baños, garajes, fecha y vendedor. Estos datos se insertan en la tabla propiedades de la base de datos mediante una consulta SQL. Si la inserción es exitosa, se muestra un mensaje de confirmación; de lo contrario, se indica que la propiedad no fue creada. El formulario incluye validaciones básicas, como campos requeridos y tipos de datos adecuados para asegurar la integridad de los datos ingresados.

## ***Resumen***

En resumen, el código del sitio web no solo es funcional y accesible, sino que también refleja buenas prácticas en el desarrollo web, facilitando la creación de una plataforma efectiva para la venta de propiedades. Este enfoque modular, junto con una estructura clara y un diseño orientado al usuario, sienta las bases para un sitio web que puede adaptarse a futuras necesidades y cambios.