

MVC: Modelo Vista Controlador

Ejercicio práctico

Nombre: EmDevApps

ÍNDICE

1. VISTA

1.1 index.php

2. CONTROLADOR

2.1 controlador_index.php

2.2 Función pintaNoticias()

3. MODELO

3.1 model_index.php

3.2 getNoticiasModel()

3.3 getUsuarioModel(\$email, \$contrasena)

3.4 agregarNoticiaModel(\$usuario_id, \$titulo, \$cuerpo, \$fecha)

3.5 editarNoticiaModel(\$id, \$titulo, \$cuerpo, \$fecha)

3.6 Función borrarNoticiaModel(\$id)

4. FUNCIONAMIENTO DE LA APLICACIÓN

5. CONCLUSIONES Y AUTOEVALUACIÓN

1.VISTA

Realmente tenemos 4 vistas en nuestra aplicación:

- vista_ver_noticia.php
- vista_registro.php
- vista_agregar_noticia.php
- index.php (**vista estudiada en esta memoria**)

Pero debido a que el modelo vista controlador (MVC) es un concepto, lo estudiaremos a fondo desglosando el documento **index.php**, acompañado de su controlador y su modelo.

A continuación se va a mostrar la estructura de nuestra aplicación. Comenzaremos por **index.php**

Login

[Iniciar sesión](#)

[Registrarse](#)

Cerrar sesión

Noticias

Titulo	Fecha	Autor o Usuario_ID
---	--	---

El archivo index.php, es enteramente una VISTA. Que realiza un **require_once** de su controlador. **controlador_index.php**

```
<?php
require_once "controlador_index.php"; // Incluir el controlador
iniciarSesion(); // Iniciar sesión
verificarEnvioFormulario(); //Verificar si se ha cerrado sesion
?>
```

El archivo **index.php**, por tratarse de una VISTA. Contendrá código html, y llamadas a funciones que se encuentran en el controlador. En este caso, **controlador_index.php**

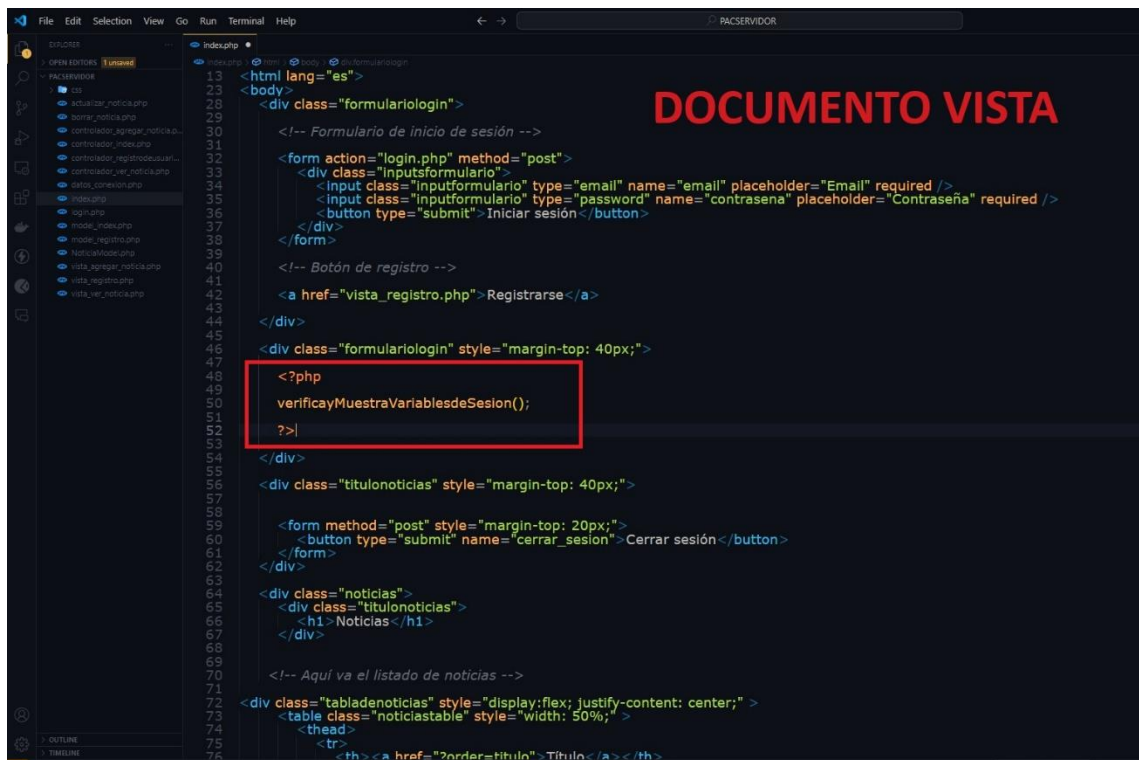
El documento **index.php**, por lo tanto, es enteramente un documento HTML, pero que requiere tener extensión **.php** porque en su interior van a existir llamadas a funciones php, que van a retornar contenido **HTML + datos integrados**.

A continuación vemos llamadas a funciones como por ejemplo:

```
<?php
```

```
verificayMuestraVariablesdeSesion();
```

```
?>
```



Otro ejemplo de llamada a función dentro de nuestra vista sería la siguiente:

```
<?php  
pintaNoticias();  
?>
```

```
<!-- Aquí va el listado de noticias -->  
<div class="tabladenoticias" style="display:flex; justify-content: center;" >  
  <table class="noticiastable" style="width: 50%;" >  
    <thead>  
      <tr>  
        <th><a href="?order=titulo">Título</a></th>  
        <th><a href="?order=fecha">Fecha</a></th>  
        <th><a href="?order=autor">Autor o Usuario_ID</a></th>  
        <?php columnaBotones(); ?>  
      </tr>  
    </thead>  
    <tbody>  
      <?php  
      pintaNoticias();  
      ?>  
    </tbody>  
  </table>  
</div>  
  
  <?php agregarNoticiaLink(); ?>  
</div>  
</body>  
</html>
```

2.CONTROLADOR

Todas estas funciones (**verificayMuestraVariablesdeSesion()**; **pintaNoticias()**; ...) se encuentran dentro del controlador. **controlador_index.php**

```
// Definir la función pintaNoticias
function pintaNoticias()
{
    // Obtener las noticias
    $noticias = obtenerNoticias();

    // Iterar sobre cada noticia
    foreach ($noticias as $noticia) {
        ?>
        <tr>
        <td>
        <!-- Enlace para ver la noticia -->
        <a href="vista_ver_noticia.php?id=<?= $noticia['id'] ?>">
        <?= $noticia['titulo'] ?>
        </a>
        </td>
        <td><?= $noticia['fecha'] ?></td>
        <!-- Verificar si la clave 'autor' está definida -->
        <td>
        <?= isset($noticia['id_autor']) ? $noticia['id_autor'] : 'Sin autor' ?>
        </td>
        <?php if (isset($_SESSION['usuario_id']) && isset($_SESSION['usuario_nombre']) && $noticia['id_autor'] == $_SESSION['usuario_id']) : ?>
        <td>
        <!-- Botón de editar -->
        <form action="vista_ver_noticia.php" method="get" style="display: inline;">
        <input type="hidden" name="id" value="<?= $noticia['id'] ?>">
        <button type="submit">Editar</button>
        </form>
        <!-- Botón de borrar -->
        <form action="borrar_noticia.php" method="post" style="display: inline;">
        <input type="hidden" name="id" value="<?= $noticia['id'] ?>">
        <button type="submit">Borrar</button>
        </form>
        </td>
        <?php endif; ?>
        </tr>
    </td>
    <?php
    }
}
```

ESTUDIO DE pintaNoticias()

pintaNoticias(); por ejemplo:

Es una función que necesita primero el array asociativo de **\$noticias**. El cual contiene aquello que retorne la función **obtenerNoticias()**. Esta última función también se encuentra dentro del controlador, pero trabaja con una función del modelo llamada **getNoticiasModel()**.

El modelo es aquel documento que recoge datos de la base de datos, y se los brinda al controlador para que este aplique la lógica, y mapee o “pinte” la información en formato HTML. Escribiendo con “**echo**” en el documento u otras formas que nos permita el lenguaje de programación empleado.

El controlador, debe de requerir o incluir, el documento modelo.

```
<?php
require_once "datos_conexion.php"; // Incluir el archivo de datos de conexión
require_once "model_index.php"; // Incluir el modelo

// Función para obtener todas las noticias
```

```
function obtenerNoticias()
{
    return getNoticiasModel();
}
```

3. MODELO

Vemos entonces la función **getNoticiasModel()**, de nuestro documento **model_index.php**

```
// A continuación veremos las funciones del MODELO.
// En modelo siempre tendremos SQL por definición.
// Las funciones del modelo nos devuelven (true, false) si se ha insertado/editado bien una noticia
// Las funciones del modelo pueden retornarnos datos. (diferentes tipos de datos almacenados en variables-> objetos, variables, arrays).

// La siguiente función getNoticias() nos retorna un array asociativo de TODAS las noticias Ejemplo: id=>{} id_autor=>{} titulo=>{}
// cuerpo=>{} fecha=>{}

function getNoticiasModel()
{
    global $conn;
    $sql = "SELECT * FROM noticia";
    $result = $conn->query($sql);
    $noticias = [];
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $noticias[] = $row;
        }
    }
    return $noticias;
}
```

Y en resumen, esa es la estructura del modelo vista controlador, que se aplicó al resto de vistas.

Resto de vistas:

- vista_ver_noticia.php
- vista_registro.php
- vista_agregar_noticia.php
- index.php (**vista estudiada en esta memoria**)

Para profundizar en las funciones del modelo, Veamos un resumen de funciones importantes del **model_index.php**.

Función getNoticiasModel()

Esta función devuelve un array asociativo que contiene todas las noticias de la base de datos. Cada noticia está representada por un conjunto de campos, incluyendo ID, ID del autor, título, cuerpo y fecha.

Función getUserModel(\$email, \$contrasena)

Esta función recupera un usuario de la base de datos utilizando su dirección de correo electrónico y contraseña como parámetros. Retorna un array asociativo con la información del usuario, incluyendo ID, nombre, correo electrónico y contraseña.

Función agregarNoticiaModel(\$usuario_id, \$titulo, \$cuerpo, \$fecha)

Esta función agrega una nueva noticia a la base de datos con los parámetros proporcionados, incluyendo el ID del autor, título, cuerpo y fecha de la noticia. Retorna true si la inserción fue exitosa y false si no lo fue.

Función editarNoticiaModel(\$id, \$titulo, \$cuerpo, \$fecha)

Esta función actualiza una noticia existente en la base de datos con los nuevos valores proporcionados para el título, cuerpo y fecha. Se identifica la noticia a actualizar mediante su ID. Retorna true si la actualización fue exitosa y false si no lo fue.

Función borrarNoticiaModel(\$id)

Esta función elimina una noticia de la base de datos utilizando su ID como parámetro. Retorna true si la eliminación fue exitosa y false si no lo fue.

Cada función del modelo está diseñada para interactuar con la base de datos y realizar operaciones específicas relacionadas con la gestión de noticias y usuarios.

4.FUNCIONAMIENTO DE LA APLICACIÓN

Login

Email

Contraseña

Iniciar sesión

[Registrarse](#)

[Cerrar sesión](#)

Noticias

Titulo	Fecha	Autor o Usuario_ID
--------	-------	--------------------

Registro

Nombre

Email

Contraseña

Registrarse

[Volver](#)

Registro

Registrarse

[Volver](#)

Login

Iniciar sesión

[Registrarse](#)

[Cerrar sesión](#)

Noticias

Titulo	Fecha	Autor o Usuario_ID
------------------------	-----------------------	------------------------------------

Login

Iniciar sesión

[Registrarse](#)

Usuario ID: 17
Usuario Nombre: Eduardo

Cerrar sesión

Noticias

Titulo	Fecha	Autor o Usuario_ID	
------------------------	-----------------------	------------------------------------	--

[Agregar noticia](#)

Agregar Noticia

Sube el precio de la gasolina

Sube la gasolina debido a cambios en las negociaciones con rusia y oriente medio...

12/04/2024



Usuario ID: 17

Usuario Nombre: Eduardo

Agregar Noticia

Volver

Login

Email

Contraseña

Iniciar sesión

[Registrarse](#)

Usuario ID: 17
Usuario Nombre: Eduardo

Cerrar sesión

Noticias

Titulo	Fecha	Autor o Usuario_ID	
Sube el precio de la gasolina	2024-04-12	17	<button>Editar</button> <button>Borrar</button>

[Agregar noticia](#)

Cerramos sesión

Login

Email

Contraseña

Iniciar sesión

[Registrarse](#)

Cerrar sesión

Noticias

Titulo	Fecha	Autor o Usuario_ID
Sube el precio de la gasolina	2024-04-12	17

Detalles de la Noticia

Sube el precio de la gasolina

Fecha: 2024-04-12

Autor o Usuario_ID: 17

Cuerpo de la noticia:

Sube la gasolina debido a cambios en las negociaciones con rusia y oriente medio...

Volver

5.CONCLUSIONES

La aplicación cumple con los requisitos especificados.