

PRUEBA FONSECANTERO

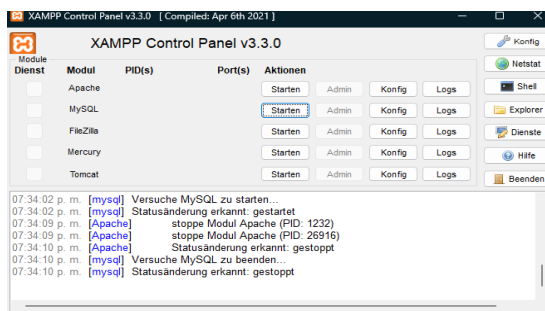
REQUISITOS:

-PHP 8.2.12 (usar [XAMPP](#) de preferencia)

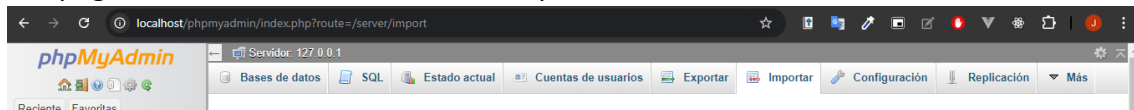
-Mysql

INSTRUCCIONES PARA EJECUTAR Y CONFIGURAR:

-Lo único que se configura es la importación de la base de datos para ello abrimos xampp, e iniciamos apache y Mysql



-Ahí mismo precionamos en Admin de Mysql lo cual nos abre una pagina y dentro de esa pagina seleccionamos el botón de importar



-ahí seleccionamos nuestro archivo db.sql que se encuentra dentro de la carpeta database, una vez seleccionado nos vamos hasta debajo de la pagina y le damos en importar



-ahora podemos cerrar esa pagina web y continuamos.

-Copiar la carpeta del proyecto en C:\xampp\htdocs

-En nuestro navegador accedemos a la ruta <http://localhost/prueba-fonsecantero/> (Si no le cambiamos el nombre al proyecto si no se tiene que cambiar prueba-fonsecantero por el nombre nuevo)

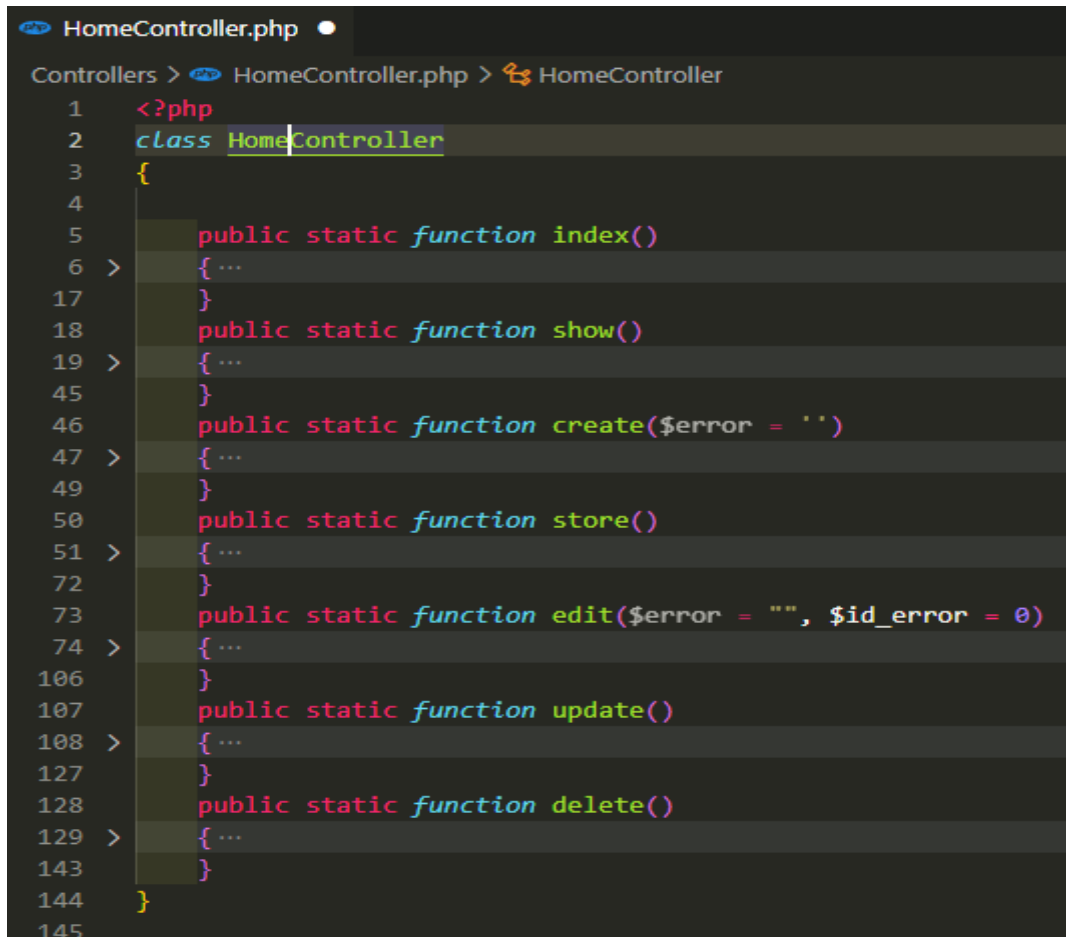
-Listo ya tenemos todo configurado para poder usar la pagina

-NOTA:Si se quiere modificar los parámetros de la DB (user,host,dbname) se puede hacer en el archivo database/config.php

ESTRUCTURA GENERAL:

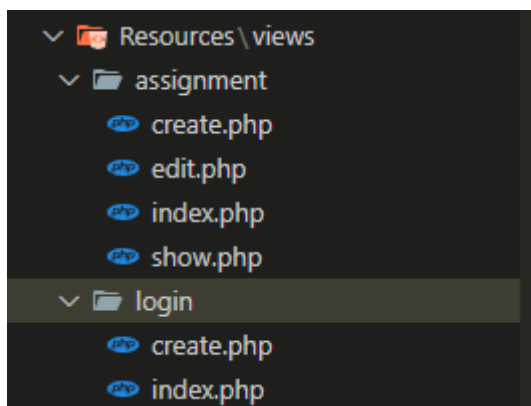
-Se creo un modelo modelo-vista-controlador

-Los controladores están dentro de la carpeta Controllers y tiene la estructura de NombreController.php los cuales son objetos que después nosotros llamamos en el routes.php



```
1  <?php
2  class HomeController
3  {
4
5      public static function index()
6  >  { ...
17  }
18  public static function show()
19  >  { ...
45  }
46  public static function create($error = '')
47  >  { ...
49  }
50  public static function store()
51  >  { ...
72  }
73  public static function edit($error = "", $id_error = 0)
74  >  { ...
106  }
107  public static function update()
108  >  { ...
127  }
128  public static function delete()
129  >  { ...
143  }
144  }
145
```

-Las vistas se crean en la carpeta resources/views y se ponen dentro de la carpeta con nombre igual al controlador al que pertenecen



```

v Resources \ views
v  folder assignment
    create.php
    edit.php
    index.php
    show.php
v  folder login
    create.php
    index.php
```

-Los modelos llaman al config.php de la carpeta database y ellos son clases que tienen la estructura de las tablas de la DB y estos pueden ser llamados por los controladores para hacer consultas simulando un ORM

```
<?php
class Progress
{
    private $id;
    public $name;
    public $color;
    private $status;
    private $created_at;
    private $updated_at;
    public static function All()
    {
        require (__DIR__ . '/../database/config.php');
        try {

            $sql = "SELECT * FROM progress";

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

            $stmt->execute();

            $resultados = $stmt->fetchAll(PDO::FETCH_ASSOC);
            $pdo = null;

            return $resultados;

        } catch (PDOException $e) {
            echo "Error: " . $e->getMessage();
        }

        $pdo = null;
    }
}
```

-Por ultimo y la llave principal es el archivo routes el cual se encarga dependiendo de la ruta de elegir que método y que controlador llamar controlando los accesos dependiendo de la sesión

```
<?php
require_once(__DIR__ . '/Controllers/SessionController.php');
require_once(__DIR__ . '/Controllers/AssignmentController.php');
//Aquí tenemos todas las rutas tanto para usuarios activos y no activos
//esta tiene una función que se llamara dependiendo de la ruta que se visite
$routesActive = [
];
$routesInactive = [
];
// Esto llama al url correspondiente si existe y si no existe la nos manda a la raíz
session_start();

if (isset($_SESSION['name'])) {
    if (array_key_exists($_SERVER['REQUEST_URI'], $routesActive)) {
        $routesActive[$_SERVER['REQUEST_URI']]();
    } else {
        header("Location: " . baseurl('/'));
        exit();
    }
} else {
    if (array_key_exists($_SERVER['REQUEST_URI'], $routesInactive)) {
        $routesInactive[$_SERVER['REQUEST_URI']]();
    } else {
        header("Location: " . baseurl('/'));
        exit();
    }
}
```