**Taller práctico.**

**Actividad N 25.**

**MVC\_CRUD parte 1.**

**Objetivo de la actividad:** Codificar las interfaces graficas y módulos usando PHP y el servidor apache aplicando el concepto de MVC\_CRUD

**Paso 1.**

Crear la carpeta del proyecto llamada primermvc, en la ruta: c:/xampp/htdocs/

|  |
| --- |
| RUTA: c:/xampp/htdocs/primermvc |
|  |

**Paso 2.**

Una ves creada la carpeta, dentro de la misma, se crean las subcarpetas denominadas:

* Controller. Contendrá los controladores. Compuesto por acciones que en código son métodos / funciones.
* Model. Contendrá las y entidades de negocio.
* View. Contendrá las vistas (código html)

|  |
| --- |
| RUTA: C:\xampp\htdocs\primermvc |
|  |

**Paso 3.**

En las dentro de la carpera primermvc, se crean la subcarpeta.

* Assets. Contendrá el diseño css, código JavaScript (js) y fuentes del proyecto.

Ahora se cera el scrip llamado index.php. este será el FrontController

|  |
| --- |
| RUTA: C:\xampp\htdocs\primermvc |
|  |

Paso 4.

Contrayendo el FrontController (index.php)

El proyecto comienza ejecutando el archivo index.php, que va a hacer el papel de un FrontController, encargado de validar que controlador y acción se está ejecutando mientras que el usuario navega. La lógica dice que mediante la queryString habrá 2 (dos) parámetros esenciales para saber que controlador y acción ejecutar el cual es “c” de controlador y “a” de acción

|  |
| --- |
| RUTA: C:\xampp\htdocs\primermvc |
|  |
| index.php |
| <?php  //Se incluye la configuración de conexión a datos en el  //SGBD: MariaDB.  require\_once 'model/database.php';  //Para registrar productos es necesario iniciar los proveedores  //de los mismos, por ello la variable controller para este  //ejercicio se inicia con el ‘proveedor’.  $controller = 'proveedor';  // Todo esta lógica hará el papel de un FrontController  if(!isset($\_REQUEST['c']))  {  //Llamado de la página principal  require\_once "controller/$controller.controller.php";  $controller = ucwords($controller) . 'Controller';  $controller = new $controller;  $controller->Index();  }  else  {  // Obtiene el controlador a cargar  $controller = strtolower($\_REQUEST['c']);  $accion = isset($\_REQUEST['a']) ? $\_REQUEST['a'] : 'Index';  // Instancia el controlador  require\_once "controller/$controller.controller.php";  $controller = ucwords($controller) . 'Controller';  $controller = new $controller;  // Llama la acción  call\_user\_func( array( $controller, $accion ) );  } |

**Paso 5.**

A continuación, se crean los modelos, recuerda que los mismos deben ubicarse dentro de la carpeta model. Fíjate bien el nombre de la base de datos, la cual se llama para el ejemplo *mvc\_php,* igualmente fíjate las comillas que están basias. Estas corresponden a la contraseña del servidor.

|  |
| --- |
| database.php: Se define la conexión a la base de datos. |
| <?php  class Database  {  //Esta función permite la conexión al SGBD: MariaDB.  //host = tipo de conexión local - ‘localhost’.  //dbname = nombre de la base de datos.  //charset = utf8, indica la codificación de caracteres utilizada.  //root = nombre de usuario (solo para fines académicos=root).  //’’ = contraseña del root (solo para fines académicos).  public static function Conectar()  {  $pdo = new PDO('mysql:host=localhost;dbname=mvc\_php;charset=utf8', 'root', ' ');  //Filtrando posibles errores de conexión.  $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  return $pdo;  }  } |

Como actividad debes:

1. Crear los elementos descritos desde el paso 1, hasta el paso 5. Has captura de pantalla según la siguiente tabla: (recuerda hacer la descripción de cada paso empleado una buena redacción)

|  |
| --- |
| Captura de pantalla paso 1 |
|  |
| Captura de pantalla paso 2 |
|  |
| Captura de pantalla paso 3 |
|  |
| Captura de pantalla paso 4 |
| Index.php |
| Captura de pantalla paso 5 |
| Conexión.php |

1. ¿para que se usa require\_once? Explica en un párrafo de 57 palabras.

La palabra reservada del lenguaje PHP para requerir un archivo una sola vez es require\_once. Esta palabra clave se utiliza para incluir y evaluar un archivo en el código fuente de un programa PHP. A diferencia de la palabra clave require, require\_once garantiza que el archivo solo se incluya una vez, incluso si se llama a la función varias veces. Esto es útil cuando se trabaja con archivos que contienen definiciones de clases, funciones o variables globales, ya que evita errores de Re declaración.

1. ¿para que se usa PDO? Explica en un párrafo de 57 palabras.

es una extensión de PHP que proporciona una interfaz consistente para acceder a bases de datos desde PHP. Se utiliza para conectarse y trabajar con una amplia variedad de bases de datos, como MySQL, SQLite, Oracle y más. PDO ofrece una capa de abstracción que facilita la migración entre diferentes sistemas de bases de datos sin tener que reescribir el código.

1. ¿Qué es el MVC (modelo, vista, controlador)? Explica en un párrafo de 500 palabras según sea el caso. Emplea un gráfico para entender el mismo.
   1. El modelo vista controlador es un sistema para el desarrollo de aplicaciones web que permite estructurar nuestro código, separándolo en tres instancias principales: modelo, vista y controlador.

La lógica con la cual funciona una el MVC es la siguiente:

* + 1. El Usuario ingresa al sitio y se le es mostrada una vista.
    2. El usuario ejecuta una acción.
    3. La acción ejecutada es recibida por el controlador y envía la información a los modelos.
    4. Según la acción requerida y especificada por el controlador habrá un modelo que dará respuesta y reenviara la información ya procesada nuevamente al controlador.
    5. El controlador evalúa la información recibida y llama una nueva vista.
    6. Esta vista será mostrada al usuario.