**Taller práctico.**

**Actividad N 25.**

**MVC\_CRUD parte 4.**

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

**Paso 1.**

Continuando con el ejercicio, se especifican los controladores del proyecto, este código debe ubicarse en la carpeta Controller del proyecto. Creamos producto.controller.php, que contiene una lógica similar al apartado anterior. (recuerda que es un solo codigo)

|  |
| --- |
| producto.controller.php |
| <?php  require\_once 'model/producto.php';  class ProductoController{  private $model;  public function \_\_CONSTRUCT(){  $this->model = new producto();  }  //Llamado plantilla principal  public function Index(){  require\_once 'view/header.php';  require\_once 'view/producto/producto.php';  require\_once 'view/footer.php';  } |

**Paso 2.**

Creamos el CRUD, recuerda que debes antes haber creado la tabla producto en la BD.

|  |
| --- |
| producto.controller.php (continuación código paso 1) |
| public function Crud(){  $prod = new producto();  if(isset($\_REQUEST['idProducto'])){  $prod = $this->model->Obtener($\_REQUEST['idProducto']);  }  require\_once 'view/header.php';  require\_once 'view/producto/producto-editar.php';  require\_once 'view/footer.php'; |

**Paso 3.**

Se crea la función nuevo producto de igual manera se hace el requerimiento a las vistas.

|  |
| --- |
| producto.controller.php (continuación código paso 2) |
| public function Nuevo(){  $prod = new producto();  require\_once 'view/header.php';  require\_once 'view/producto/producto-nuevo.php';  require\_once 'view/footer.php';  } |

**Paso 4.**

Se crea la funcion guardar del CRUD.

|  |
| --- |
| producto.controller.php (continuación código paso 3) |
| public function Guardar(){  $prod = new producto();  $prod->idProducto = $\_REQUEST['idProducto'];  $prod->nit = $\_REQUEST['nit'];  $prod->nomprod = $\_REQUEST['nomprod'];  $prod->precioU = $\_REQUEST['precioU'];  $prod->descrip = $\_REQUEST['descrip'];  $this->model->Registrar($prod);  header('Location: index.php?c=producto');  } |

**Paso 5.**

Se crea la función editar del CRUD.

|  |
| --- |
| producto.controller.php (continuación código paso 4) |
| public function Editar(){  $prod = new producto();  $prod->idProducto = $\_REQUEST['idProducto'];  $prod->nit = $\_REQUEST['nit'];  $prod->nomprod = $\_REQUEST['nomprod'];  $prod->precioU = $\_REQUEST['precioU'];  $prod->descrip = $\_REQUEST['descrip'];  $this->model->Actualizar($prod);  header('Location: index.php?c=producto');  } |

**Paso 6.**

Se crea la función elimiar del CRUD.

|  |
| --- |
| producto.controller.php (continuación código paso 5) |
| public function Eliminar(){  $this->model->Eliminar($\_REQUEST['idProducto']);  header('Location: index.php');  }  }//no eliminar esta llave ya que cierra el class. |

Como actividad debes:

1. Crear los elementos descritos desde el paso 1, hasta el paso 6. Has captura de pantalla según la siguiente tabla: (recuerda hacer la descripción de cada paso empleado una buena redacción)

|  |
| --- |
| Captura de pantalla paso1 |
|  |
| Captura de pantalla paso2 |
|  |
| Captura de pantalla paso3 |
|  |
| Captura de pantalla paso4 |
|  |
| Captura de pantalla paso5 |
|  |
| Captura de pantalla paso6 |
|  |

1. ¿para que se usa class ProductoController? Explica en un párrafo de 57 palabras.
2. ¿Qué es el larabel? Explica en un párrafo de 500 palabras según sea el caso. Emplea un gráfico para entender el mismo.