Actividades

Laboratorio #1: Manejo de datos en el servidor e interacción con el cliente mediante una aplicación web

**DESARROLLO DE LA ACTIVIDAD**

**MEMORIA TÉCNICA.**

**Objetivo:** Diseñar y desarrollar un aplicativo web aplicando el patrón MVC (Modelo – Vista - Controlador). El aplicativo permitirá realizar acciones de consulta, creación, actualización y eliminación de un empleado.

**Introducción**

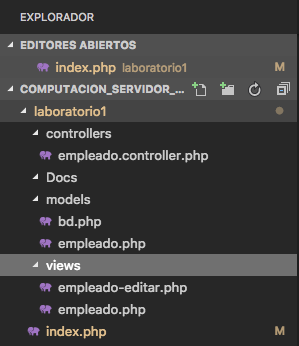
Componentes del patrón MVC (Modelo-Vista-Controlador).

**Modelo:** Se encarga de los datos, generalmente (pero no obligatoriamente). Realiza la conexión con la base de datos, realiza las consultas y actualizaciones de datos a que haya lugar.

**Controlador:** Se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

**Vistas:** Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

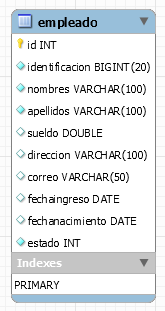
**Estructura de directorios usada para la aplicación.**



**Modelo**

Ruta: laboratorio1 / models

Como fuente de datos se usa la tabla “php\_mvc.empleado”



Como abstracción se crea la clase Empleado, esta clase tiene herencia con la clase BD la cual a su vez es la encargada de realizar la conexión a través de la librería PDO.

La implementación de la clase empleado del modelo es la siguiente:

class Empleado extends BD

Para el laboratorio se implementaron los siguientes métodos básicos para la administración de un empleado:

public function listarEmpleados()

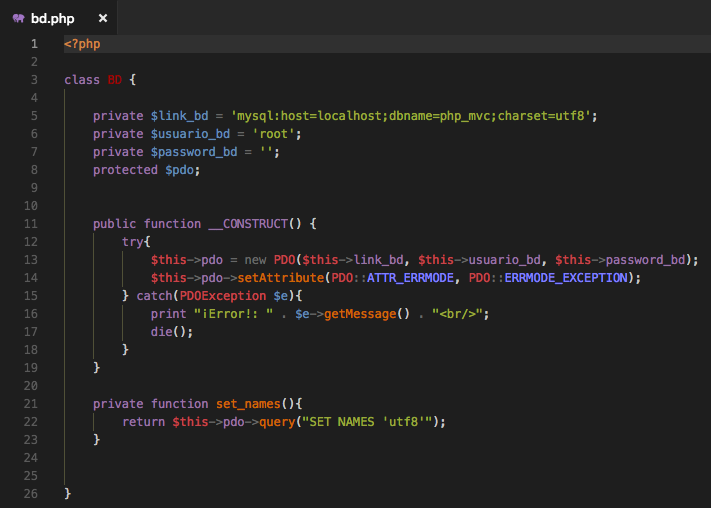
public function obtenerEmpleado($idEmpleado)

public function eliminarEmpleado($idEmpleado)

public function actualizarEmpleado($data)

public function registrarEmpleado(empleado $data)

La conexión a la base de datos se realiza por medio de la librería PDO, la cual está implementada en la clase BD del directorio de models.



Esta clase la deben heredar todas las clases del modelo que requieran conexión a la base de datos, principalmente esta clase se encarga de la conexión para extraer la información de los empleados almacenado en la base de datos.

Esta clase tiene un atributo $pdo declarado como protected que puede ser utilizado por todas las clases del modelo para realizar operaciones en la base de datos.

**Vista**

Ruta: laboratorio1 / views

Utiliza la clase Empleado, clase ubicada en el directorio correspondiente a la Vista. (Archivo “empleado.php”).

Permite maquetar la pantalla con la que interactúa el usuario. Formatea los títulos que aparecerán, pinta la grilla (tabla) con las columnas correspondientes a la tabla de empleado y consecuentemente pinta la información asociada a cada columna. Cuenta con los botones de “Nuevo Empleado”, “Editar” y “Eliminar”.

Así mismo utiliza la clase empleado-editar clase también ubicada en el directorio correspondiente a la Vista. (Archivo “empleado-editar.php”).

Para las operaciones de Creación y Edición permite **validar** el contenido de cada campo utilizando los recursos de que dispone html5.

**Controlador**

Ruta: laboratorio1 / controllers

Utiliza la clase empleadoController, clase ubicada en el directorio del Controlador anteriormente mencionado . (Archivo “empleadoController.php”).

Permite integrar tanto el modelo (models/empleado.php), como los componentes correspondientes a la vista (vista/empleado.php y vista/empleado-editar.php).

En el Modelo permite instanciar tanto la conexión, como todas las operaciones de persistencia con CRUD.

En la Vista permite desplegar el componente gráfico (GUI) de la aplicación.

Llama los métodos básicos para la administración de un empleado los cuales son:

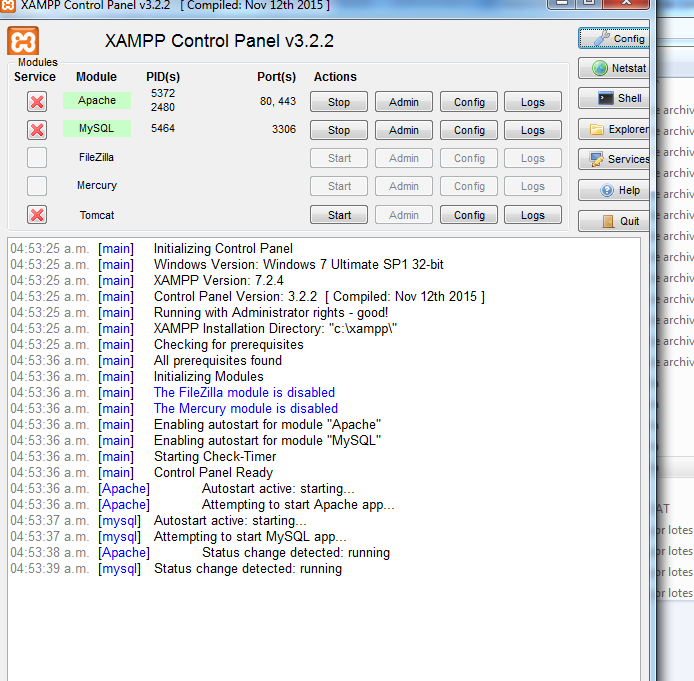
public function CRUD ()

public function guardarEmpleado ()

public function eliminarEmpleado ()

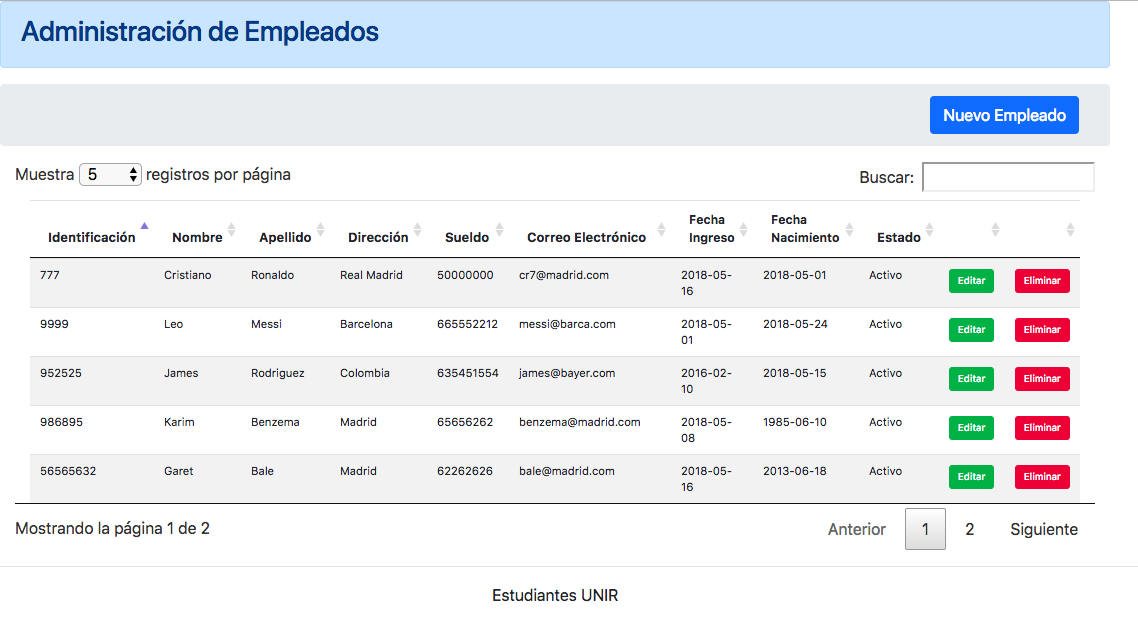
**Resultados del desarrollo**.

* A través de XAMPP levantamos tanto el servidor apache, como la base de Datos (MariaBD).

****

* Luego a través del uso de un navegador, invocamos el servicio- Cliente.

Pantalla principal

****

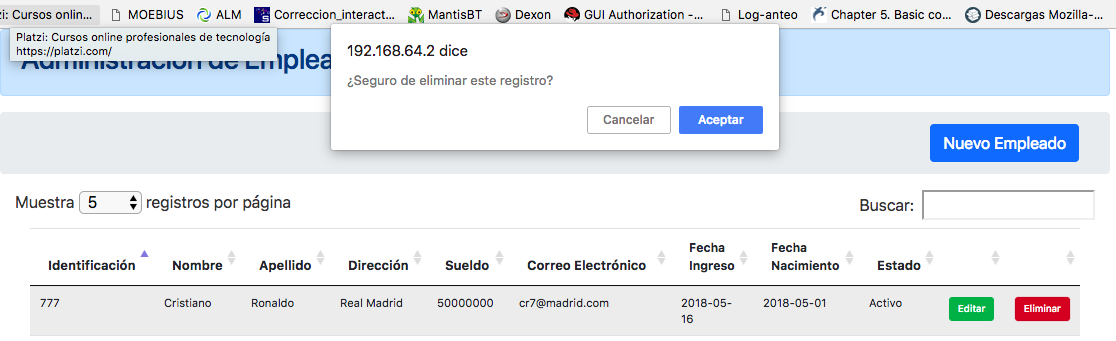
* Seleccionamos en la Interfaz la operación de “Nuevo Empleado”. Permitiendo desplegar la pantalla donde se registrarán los datos correspondientes al empleado así:

****

* El usuario puede seleccionar la opción de “Editar”, ubicada en el extremo derecho de cada registro. Esta acción despliega la información correspondiente al registro seleccionado, permitiendo la actualización de su información así:

****

* Por último el usuario puede seleccionar la opción de “Eliminar ” , ubicada en el extremo derecho de cada registro. Esta acción despliega un mensaje de confirmación donde se pregunta: “ Seguro de eliminar este registro” . Una vez el usuario le dé aceptar se ejecuta la operación de eliminación del registro en la tabla empleado así:



**Conclusiones:**

* En la ejecución del laboratorio se aplican los conocimientos adquiridos, asociados al patrón MVC. Como resultado final se tiene un aplicativo web funcional con las operaciones básicas CRUD de un modelo transaccional.
* El patrón MVC permite desacoplar el trabajo de los ingenieros que desarrollan la aplicación. La distribución de cargas de trabajo asociadas directamente al patrón MVC facilita y mejora el rendimiento de la fase de desarrollo.