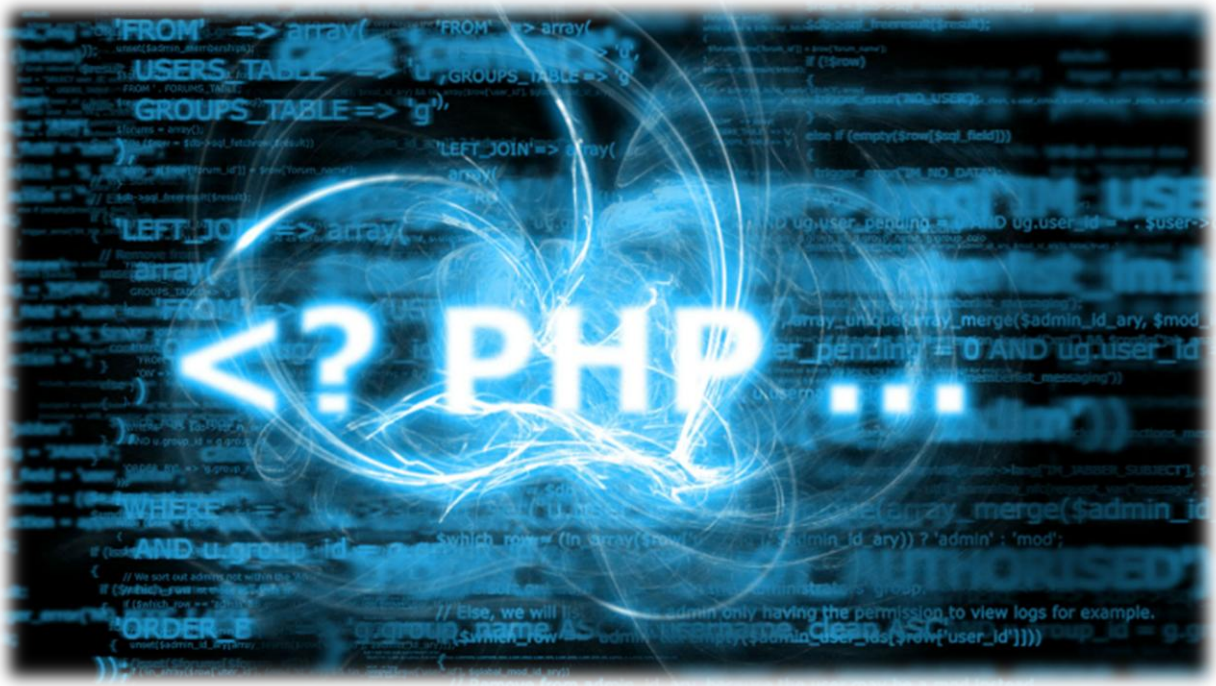
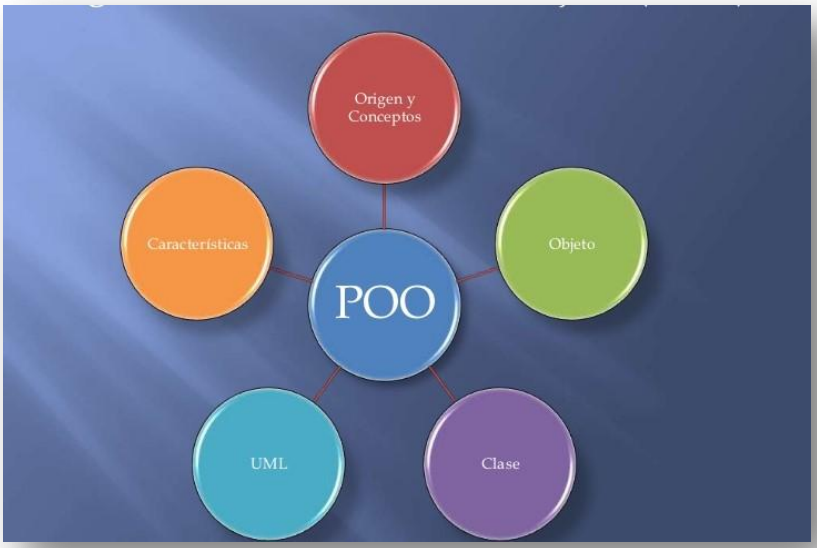
DESARROLLO DE UN FRAMEWORK DESDE CERO CON MVC, POO y PDO



ÍNDICE

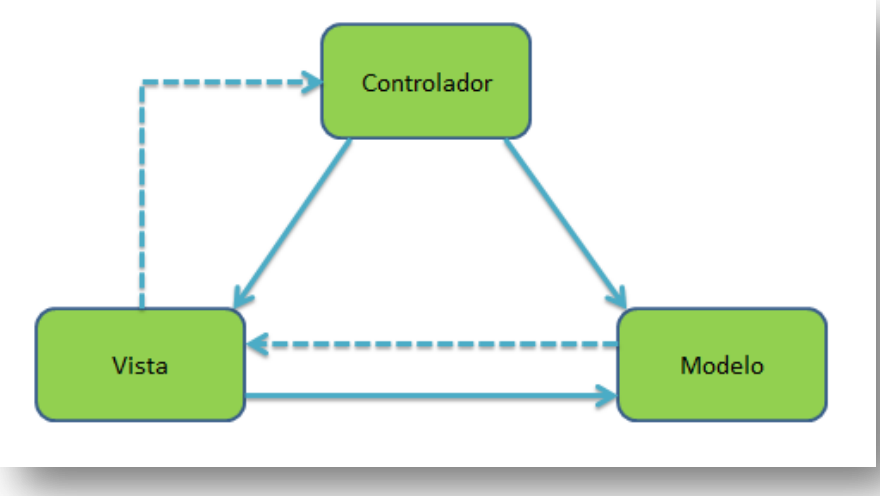
¿Qué es POO?

La Programación Orientada a Objetos es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento.



¿Qué es MVC?

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello **MVC** propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.



¿Qué es PDO?

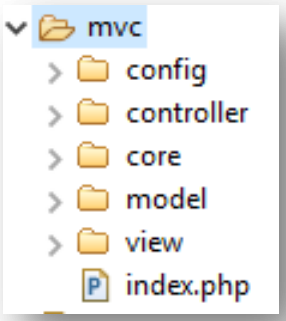
PHP Data Objects (o PDO-) es una extensión que provee una capa de abstracción de acceso a datos para PHP, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos gestores de bases de datos.

Características de la aplicación

- Estructura de nuestra aplicación MVC.
- Aplicación trabajando con POO.
- Uso de interfaces para ejemplo de crud.
- Conexión con base de datos utilizando PDO
- Control de acceso a directorios/archivos no permitidos.

Estructura de la aplicación MVC

Podemos ver en la siguiente imagen como sería nuestra estructura según el MVC para nuestra aplicación:



El framework tendrá varios directorios:

- **config**: aquí irán los ficheros de configuración de la base de datos, globales, etc.
- **controller**: este directorio contiene los controladores que se encargarán de recibir y filtrar datos que le llegan de las vistas, llamar a los modelos y pasar los datos de estos a las vistas.
- **core**: aquí colocaremos las clases base de las que heredarán por ejemplo controladores y modelos, esto sería el núcleo del framework.
- **model**: aquí irán los modelos, para ser fieles al paradigma orientado objetos tenemos que tener una clase por cada tabla o entidad de la base de datos (excepto para las tablas pivote) y estas clases servirán para crear objetos de ese tipo de entidad (por ejemplo crear un objeto usuario para crear un usuario en la BD). También tendremos modelos de consulta a la BD que contendrán consultas más complejas que estén relacionadas con una o varias entidades.
- **view**: aquí irán las vistas, es decir, donde se imprimirán los datos y lo que verá el usuario.
- index.php será el controlador frontal por el que pasará absolutamente todo en la aplicación.

1. Diseño e implementación:

Este proyecto ha sido diseñado conforme al modelo vista controlador. En cuanto al diseño de la aplicación final se ha realizado usando bootstrap. Para la conexión con la base de datos se ha usado PDO.

Base de datos:

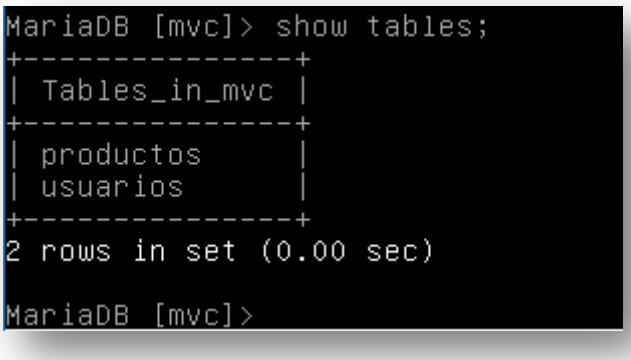
Usamos MariaDB como gestor de base de datos. Está formada por dos tablas que no se relacionan. Usuarios y productos. Cada una registra los siguientes valores:

Usuarios:

- **id**: clave primaria de tipo entero y auto incremental.
- **Email**: tipo varchar.
- **Password**: tipo varchar.
- **Nombre**: tipo varchar.
- **Apellidos**: tipo varchar.

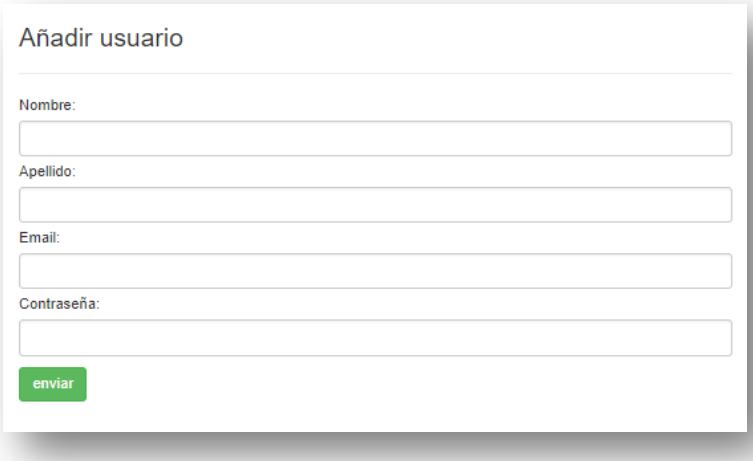
Productos:

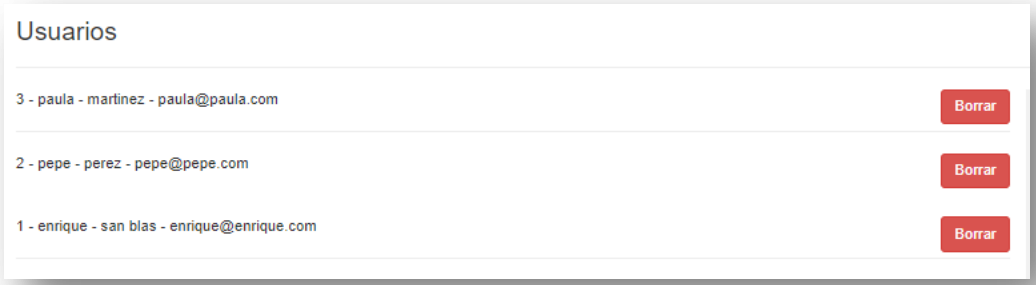
- **Id**: clave primaria de tipo entero y auto incremental.
- **Nombre**: tipo varchar.
- **Precio**: tipo varchar.
- **Marca**: tipo varchar.

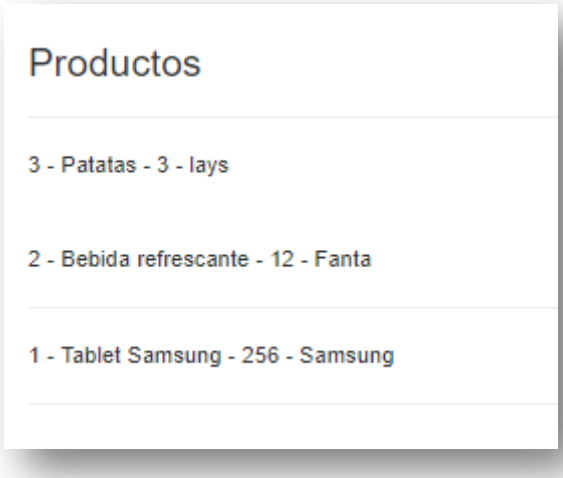


La configuración para la conexión de la base de datos la tenemos en el directorio config de nuestra estructura MVC.

Vista de la aplicación:







2. Problemas encontrados:

En la realización de este proyecto, nos hemos encontrado con los siguientes problemas:

- Errores a la hora de realizar enrutamientos para las acciones de los botones o el autoload.
- Consultas a la base de datos sin resultados.

3. Trabajo futuro y conclusiones:

Actualmente la aplicación no tiene funciones de inserción, modificación y borrado. Es meramente visual y solo recoge valores de la base de datos. Se prevé completar el proyecto con el resto de funciones CRUD además de hacer una ampliación de la base de datos pudiéndose adaptar a la de un programa de gestión de ventas.

4. Documentación:

Para poder acceder a la visualización de la aplicación, debemos pinchar en el siguiente enlace:

https://htmlpreview.github.io/?https://github.com/kike094c/ProyectoDSW/blob/master/view/indexView.php

Para visualizar y descargar el código debemos pinchar en el siguiente enlace:

https://github.com/kike094c/ProyectoDSW

En el repositorio se encuentra el archivo **mvc.sql**, es la base de datos de la aplicación que puede ser importada en cualquier gestor de bases de datos y modificando los parámetros del archivo databases.php dentro del directorio config, podemos ver el resultado. No obstante en el presente documento hay una serie de capturas de cómo sería la visualización.