\*Sistema Desarrollo11\*

El presente documento expone el manual técnico del \*sistema\* con la finalidad de contar con una explicación clara y precisa del código.

A continuación, se enlistarán las tecnologías ocupadas, así como el patrón de programación utilizado en el \*sistema\*

Tecnologías

* PHP 7.1.20
* MySQL Community Server – GPL
* MaterializeCSS 1.0.0-beta
* XAMPP
* HTML 5
* CSS 3
* JavaScript
* jQuery 3.3.1

Patrón

* MVC
* Orientado a objetos.

Es importante remarcar que no se usó ningún framework para el desarrollo de este proyecto, por lo que es necesario que quien lo manipule cuente con los conocimientos adecuados de PHP nativo, así como JavaScript.

**Estructura del proyecto**

Debido al uso del patrón MVC el sistema cuenta con tres carpetas principales, así como carpetas de recursos, en la siguiente figura se muestra el árbol principal del proyecto.

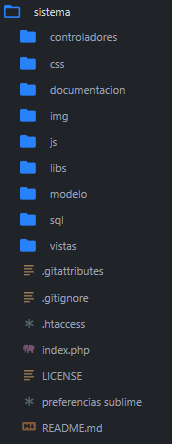


Figura 1 Estructura del proyecto.

Sistema: es la carpeta raíz del proyecto.

Controladores (Figura 2) contiene la lógica.

CSS (Figura 3) Hojas de estilo.

Documentacion (Figura 4) Manuales.

Img (Figura 5) imágenes del proyecto.

Js (Figura 6) Scripts de Javascript.

Libs (Figura 7) librerías nativas del proyecto.

Modelo (Figura 8) contiene las consultas.

Sql (Figura 9) script y modelo de la BD.

Vistas (Figura 10) HTML para la interfaz.

Clases de github.

.htaccess (Figura 11) especificaciones.

Index.php (Figura 12) HTML inicial.

Archivos pertenecientes a materialize.

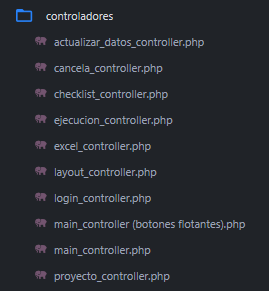
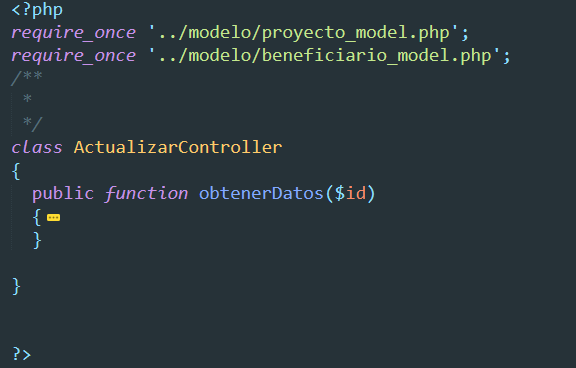


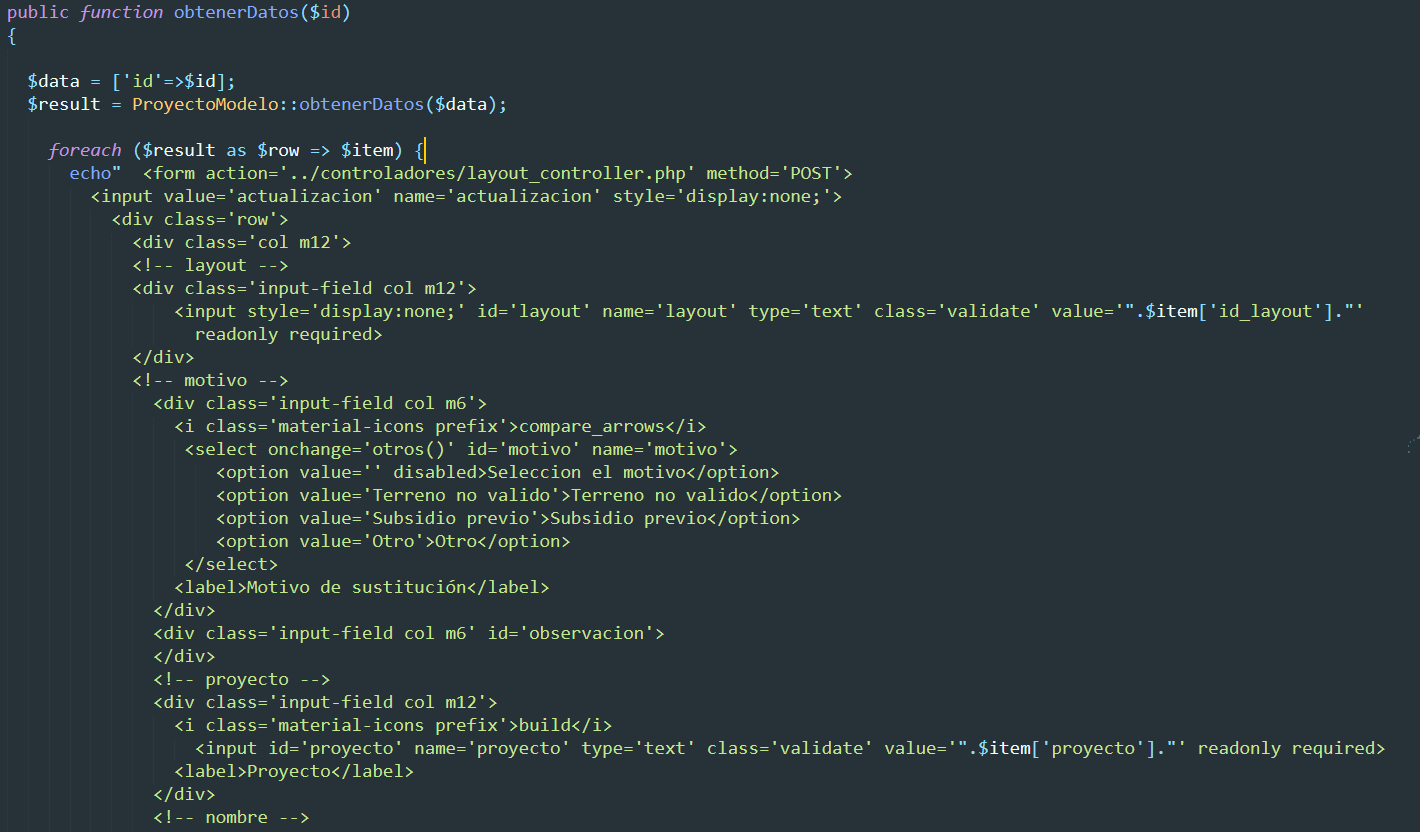
Figura 2 Controladores

Cada controlador se liga una vista, con la finalidad de mantener un orden por lo que en esta sección se explicarán paso a paso cada controlador y sus funciones, empezando por orden alfabético:

**Actualizar\_datos\_controller.php**:



Este archivo únicamente cuenta con una clase llamada ActualizarController, la cual contiene una sola función “ObtenerDatos” que recibe un id el cual pertenece al layout del usuario seleccionado. A grandes rasgos la función busca la información existente el usuario y la convierte en un formulario de actualización de la siguiente forma:



El id del layout se almacena en el arreglo ‘$data’ con la clave ‘id’, dicho arreglo es enviado a la función “obtenerDatos” de la clase “ProyectoModelo”