**Documentación Técnica**

**FRONTEND**  
Nuestro Programa está realizado en PHP junto con código HTML y CSS.

En la parte superior tenemos nuestro menú el cual contiene 2 botones “Registros” y “Registrar”.

Registrar cómo su nombre indica sirve para crear un nuevo registro que posteriormente será visualizado en la ventana de registros y también se avisa por medio de una ventana modal si el registro fue exitoso o no.

Nuestra página principal(index.php) muestra todos los registros almacenados en la BD mediante un select de todos los registros activos y se hace un loop para almacenar cada registro en una card individual en caso de no haber registros se muestra el mensaje “no se encontraron resultados.

Cada contiene 2 botones que son “editar” y “eliminar”. En el caso de eliminar, estamos haciendo una baja lógica, es decir, nuestro información como tal no se borra, si no que simplemente su status se actualiza a 0 en la base de datos lo cual nos permite seguir teniendo este registro en caso de ser necesario, pero para el usuario no será más visible. Editar coloca la información de la card en el encabezado y nos lleva a otra ventana muy similar a la de registro donde ya nos vendrán los datos del elemento elegido en las casillas, una vez realizado el cambio nos redirige de nuevo al index.php. Todo lo anterior mencionado se confirmará con una ventana modal para saber si salió bien o mal.

**BACKEND:**

Nuestra Base de datos es muy simple y la tenemos justo aquí:  
CREATE TABLE records(

idRecords MEDIUMINT NOT NULL AUTO\_INCREMENT,

title CHAR(30) NOT NULL,

info CHAR(100) NOT NULL,

status TINYINT DEFAULT 1,

PRIMARY KEY(idRecords)

)  
cómo podemos ver tenemos un id que se autoincrementa y se llena en automático, así evitamos posibles conflictos de una llave repetida y a su vez tenemos el status que mencionamos anteriormente, lo cual nos permitirá tener la baja lógica. Por último tenemos nuestros atributos “title” e “info” que es la información que ingresa nuestro usuario.  
  
En el backend hay distintos archivos, están separados por las funciones CRUD para una más fácil comprensión. En todos los casos se usó un “REQUEST METHOD POST” ya que en este ámbito es mucho más seguro enviar nuestros archivos con este método. Por medio del “header” una vez realizada la tarea requerida se nos es enviado de regreso a una ventana en específico y utilizando la sesión de PHP enviamos información para saber si la tarea fue exitosa y mostrar las debidas ventanas modales.

Nuestro archivo config.php nos asigna los datos para conexión de la base de datos. Hay que enfatizar que en este caso estoy usando Xampp junto con apache y mysql.

También tenemos nuestro archivo “validation.php” el cual nos ayuda a comprobar nuestra información antes de hacer el registro en la base de datos. El formato aceptado de los datos es el siguiente:

* title: Solo acepta letras del abecedario ya se acentuadas o no y la “ñ” además debe tener un largo máximo de 30 caracteres.
* info: Permite cualquier tipo de carácter, pero no debe se superar los 100 caracteres de largo.

Por último, tenemos nuestros archivos javascript los cuales son dropdownMenu.js y validation.js.

El primero nos sirve para soltar el menú cuando estamos en móviles. Ya que nuestra página es responsiva al estar por debajo de 601 pixeles se activa nuestro menú desplegable para tener una mejor vista y le sea más fácil al usuario navegar.  
  
El segundo archivo es de igual manera un archivo muy similar a nuestro archivo del backend que nos permite dar seguridad que se están siguiendo las reglas preestablecidas de nuestra información.