

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped graphic points to the right, containing the date. Below the arrow, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

17-3-2023

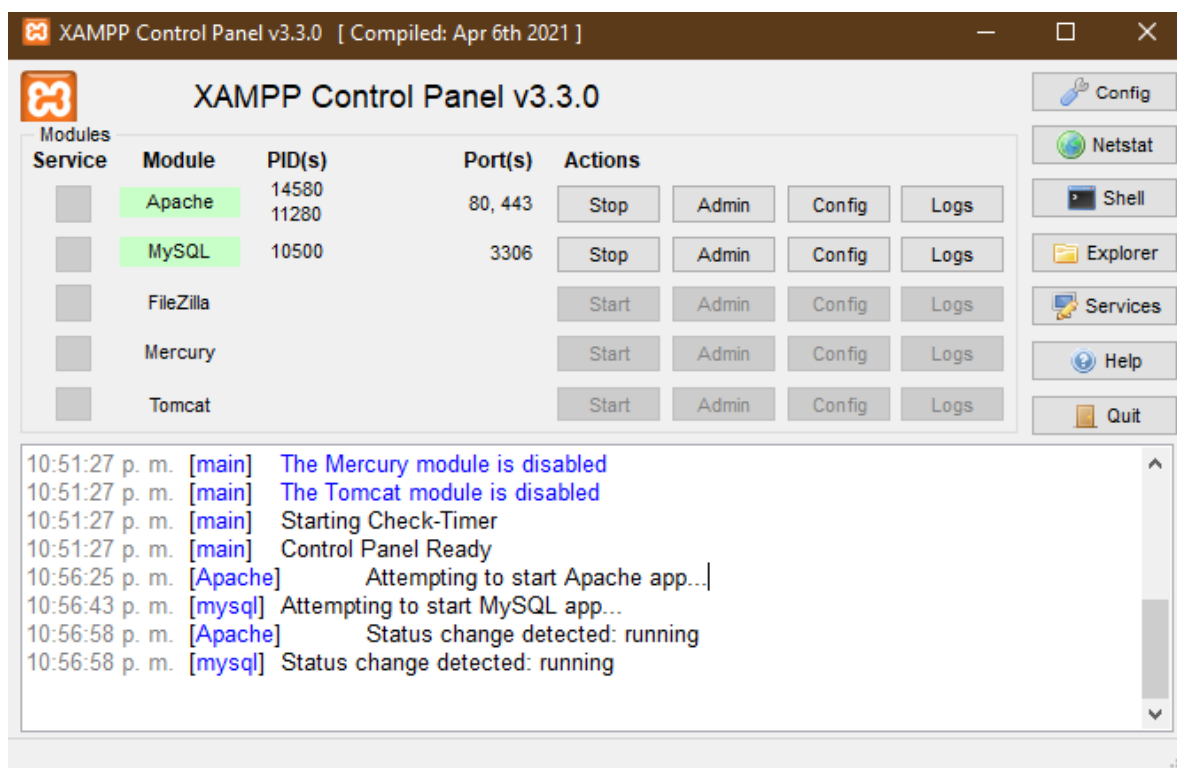
# Documentación de proyecto

Servicio Social

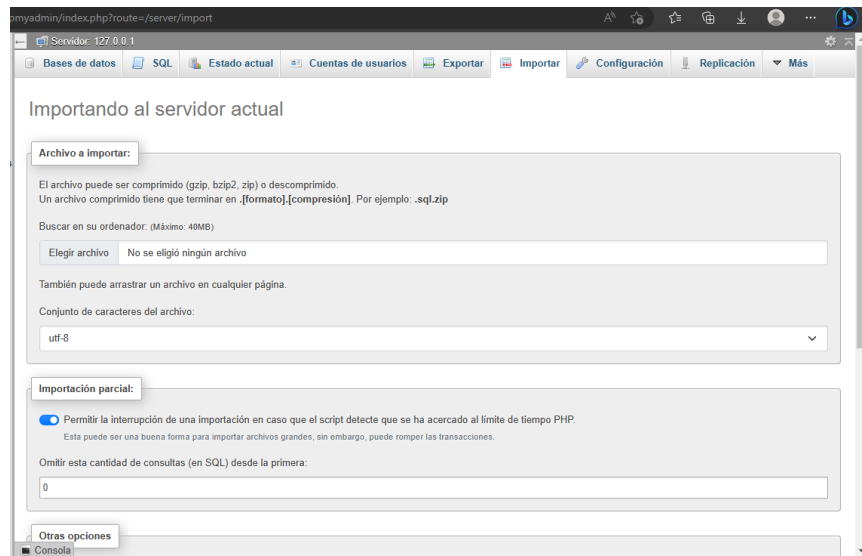
El sistema de la **Matriz De Administración De Riesgos Institucional** esta desarrollado bajo el lenguaje de PHP, usando como complemento phpMyadmin como el “motor” que contiene la base de datos.

Primeramente, para poder ejecutar y acceder al sistema es importante realizar la instalación del programa XAMPP, el cual contiene lo que es Apache para la ejecución del proyecto, MySQL en su modalidad de phpMyAdmin y el lenguaje de PHP, entonces durante el proceso de instalación solo es necesario seleccionar el servicio de Apache y de MySQL en dado caso de ya contar con MySQL, no será necesario instalarlo.

Una vez que este instalado XAMPP, en la ruta que se haya especificado, en la carpeta de htdocs que se encuentra en la carpeta principal del programa se debe incluir la carpeta con el proyecto de la Matriz. Ya hecho todo esto, se debe ejecutar el programa de XAMP control panel para iniciar los servicios:



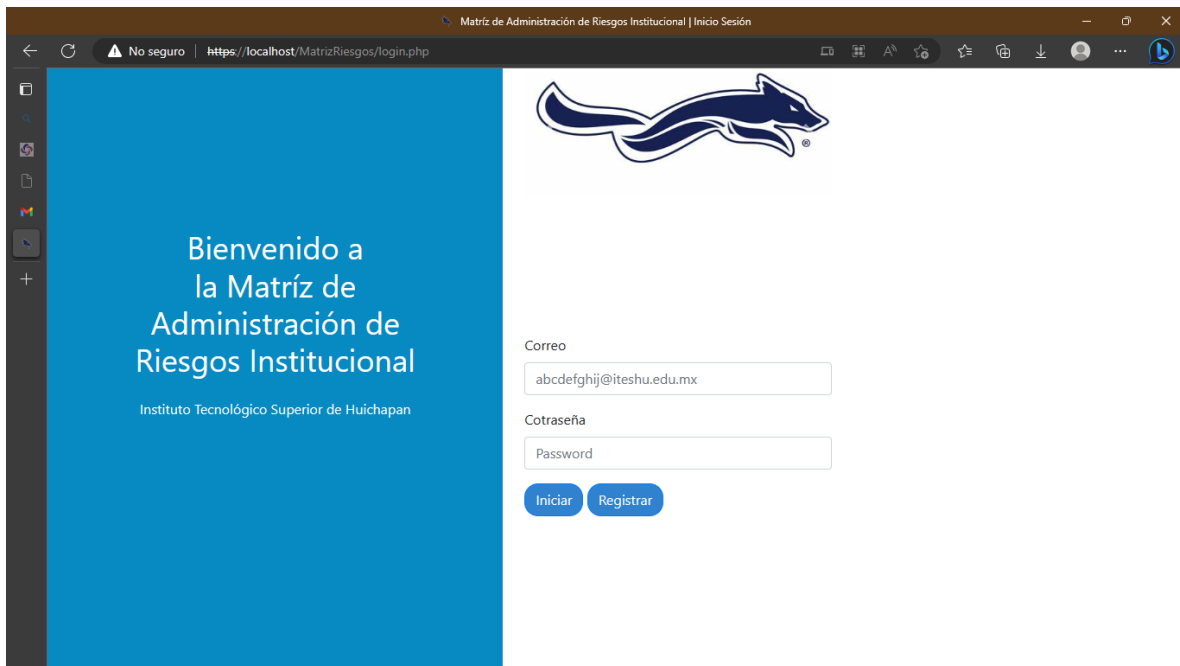
Importante mencionar que el sistema ya cuenta con la conexión a la base de datos por lo que será necesario ejecutar el script de la misma dentro de phpMyAdmin para poder acceder al sistema sin ningún problema, esto se hace en el apartado de importar desde la página principal de la herramienta antes mencionada:



Posterior a la ejecución del script de la base de datos en el archivo de “DataBaseConnect.php” que forma parte del proyecto, se deben cambiar las credenciales en caso de ser necesario:

```
1 <?php /* Etiqueta de apertura de código PHP */
2 $servername = "127.0.0.1";
3 $database = "matrizadminriesgositeshu";
4 $username = "root";
5 $password = "";
6 // Create connection
7 $conn = mysqli_connect($servername, $username, $password, $database);
8 // Check connection
9 if (!$conn) {
10     die("Connection failed: " . mysqli_connect_error());
11 }
12 echo "<script type='text/javascript'>console.log('Connected successfully'); </script>"; //Comando para imprimir mensajes.
13
14 <!-- Etiqueta de cierre de código PHP -->
```

Teniendo todo eso modificado ya se podrá acceder al proyecto desde el navegador con la siguiente ruta: <https://localhost/MatrizRiesgos/login.php>



Matriz de Administración de Riesgos Institucional | Inicio Sesión

No seguro | <https://localhost/MatrizRiesgos/login.php>

Bienvenido a  
la Matriz de  
Administración de  
Riesgos Institucional

Instituto Tecnológico Superior de Huichapan

Correo  
abcdefghijkl@iteshu.edu.mx

Contraseña  
Password

Iniciar Registrar

Se debe generar un usuario para poder acceder al sistema, esto se puede lograr desde el módulo de registro o usar el usuario que se ha definido como administrador: admin@admin con la contraseña de 123, para poder ver la siguiente interfaz:



Información General

No seguro | <https://localhost/MatrizRiesgos/Informacion.php>

ESTADOS UNIDOS MEXICANOS  
**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA

Formato de Administración de  
Riesgos Institucional

RAMO / SECTOR: 11 Educación Pública

INSTITUCIÓN: Registrar el nombre de la Institución

AÑO: Ingresar el año (4 dígitos)

Titular de la Institución Nombre y Apellidos de la titular

Coordinador de Control Interno Nombre y Apellidos del Coordinador

Enlace de Administración de Riesgos Nombre y Apellidos

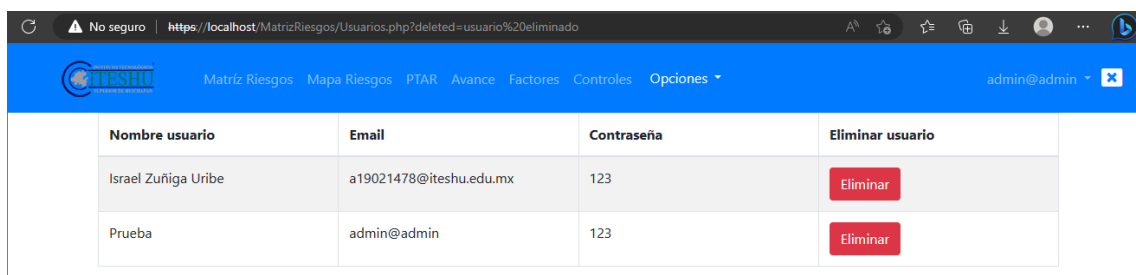
Siguiente

En esta solo faltaría hacer que los cambios que se de información que haga el administrador solo sea visible y no editable por los demás usuarios.



El diseño de estos módulos es con HTML, CSS y un poco de JavaScript, las conexiones entre los módulos ya están realizadas, además de ciertas alertas que permiten la validar ciertas funciones del sistema.

Las pestañas importantes son desde matriz hasta avance, incluyendo la parte del menú de opciones, el cual solo le aparece al usuario administrador, en este se pueden visualizar los usuarios existentes en la base de datos, así como las áreas y servidores dados de alta en el sistema, tal y como se muestra en las siguientes capturas, ambas interfaces cuentan con el botón para poder eliminar algún registro, en el caso de los usuarios se debe tener cuidado con lo de las llaves foráneas debido a que si un usuario está relacionado a algún registro de otra tabla no será posible borrarlo, dentro del módulo de áreas y encargados, **faltaría completar la parte de poder actualizar los campos**, mi propuesta fue mediante alertas así como se muestra en la última captura de pantalla.



Nombre usuario	Email	Contraseña	Eliminar usuario
Israel Zuñiga Uribe	a19021478@iteshu.edu.mx	123	Eliminar
Prueba	admin@admin	123	Eliminar



Número	Área Administrativa	Servidor	Eliminar	Editar Área	Editar Servidor
1	Dirección General	Mtra. Imelda Pérez Espinoza	Eliminar	Editar	Editar
2	Dirección Académica	Mtro. Juan José Molina Montufar	Eliminar	Editar	Editar
3	Subdirección Académica	Dr. Octavio Guerrero Andrade	Eliminar	Editar	Editar
4	Dirección de Planeación y Vinculación	Mtra. Ada Lucía Prieto Pérez	Eliminar	Editar	Editar
5	Subdirección de Investigación y Posgrado	Dr. Miguel Ángel Espíndola Lugo	Eliminar	Editar	Editar
6	Subdirección de Vinculación y Servicios Externos	L.C.C. Francisco Uriel Camacho Aranda	Eliminar	Editar	Editar



### Editar Área

Cancel
OK

En el módulo de áreas encargados, se encuentra un botón hasta la parte final de la página que permite ir a una nueva interfaz para dar de alta una nueva área con su respectivo encargado.



## Registro

Módulo para Agregar Áreas y Servidores

Área

Área Institucional

Servidor

Titular del Área

Enviar

Los registros a la base de datos se logran de la siguiente manera:

```
InsertarAreaEncargado.php M X
InsertarAreaEncargado.php > html > body > div.alert.alert-success > h4.alert-heading
17 </head>
18 <body>
19 <?php
20     require_once("DataBaseConnect.php");
21     if(isset($_POST['registrar'])){
22         $area = $_POST['area'];
23         $servidor = $_POST['encargado'];
24
25         $query = "INSERT INTO areas_encargados VALUES ('','$area','$servidor')";
26         if(mysqli_query($conn, $query)){
27             ?>
28             <script type="text/javascript">
29                 swal({
30                     title: "Inserción completada!",
31                     text: "Los datos han sido registrados correctamente!",
32                     icon: "success",
33                 });
34             setTimeout("location.href='AreasEncargados.php'", 400);
35             </script>
36         <?php
37         } else{
38             ?><div class="alert alert-success" role="alert">
39                 <h4 class="alert-heading">Error al insertar los datos</h4>
40             </div>
41             <?php
42         }
43     }
44 }
45 ?>
```

Se utilizó un tipo de programación embebida, es decir incrustando código PHP en medio del código HTML, entonces el anterior fragmento de código se encuentra

dentro del archivo del archivo que se encarga de mostrar el formulario para insertar el área y el servidor, primero se manda a llamar el archivo que realiza la conexión a la base de datos, para poder insertarle información, las líneas de código 22 y 23 se encargan de meter en variables lo que contenga la caja de texto (input) tanto del área como del servidor, la variable \$query es la sentencia SQL en la que se escribe lo que se desea hacer en la base de datos, en este caso es una inserción, pero bien podría ser alguna otra operación propias de una base de datos. Lo que se encuentra en el if es para ejecutar lo que contiene la variable \$query usando el método de mysqli\_query el cual requiere de dos parámetros, la cadena de conexión que sale del archivo de conexión a la base de datos y como segundo parámetro se ocupa la variable \$query que contiene la sentencia SQL, si la ejecución es correcta se mostrara una alerta de confirmación y después se redireccionará a la ventana donde están listadas todas las áreas y servidores incluyendo el nuevo registro que se haya realizado.

Este fragmento de código sirve de base para poder entender algunos otros similares dentro del sistema, lo único diferente serían las sentencias SQL, las alertas usadas y el nombre de las variables que se deseen usar, además de que ciertas partes del código cuentan con comentarios para ayudar a su comprensión.

Dentro del sistema hay módulos que requieren solo ciertos cambios para lograr un correcto funcionamiento, por ejemplo, en el de "MatrizRiesgos.php" falta corregir la parte del diseño como lo mencionaba anteriormente, así como arreglar la parte de añadir una nueva fila en la tabla HTML para poder hacer el registro de varios riesgos, una vez corregido eso, se debe de continuar con la parte de ingresar a la base de datos todos esos registros de manera estructurada, es decir, que cada riesgo tenga sus respectivos factores y que cada factor vaya con sus controles, es decir, en las siguientes líneas de código:

```
179 $insertar = "INSERT INTO riesgos(num_riesgo,unidad_administrativa, alineacion_seleccion, alineacion_descripcion, riesgo, nivel, clasificacion_
180 // $insertar = "INSERT INTO factores VALUES ('$factor1','$descripcionF1','$clasFac','$tipoFac','$noRiesgo')";
181
182 // $insertar = "INSERT INTO controles VALUES ('$numControl','$desControl','$tipoControl','$documentado','$formalizado','$seAplica','$seEfectivo'
183 if(mysqli_query($conn,$insertar)){
184     /* header("Location: MatrizRiesgos.php"); */
185 }else{
186     echo "<div class='form'><h3>incorrect.</h3><br/>Haga clic aquí para <a href='MatrizRiesgos.php'>volver</a></div>";
187 }
188
```

Ahora bien, en dado caso de que esto sea complicado de hacer, en los archivos que conforman el sistema se encuentran los módulos de factores y controles que se pensaron como alternativa para añadir información a la base de datos en base a los riesgos existentes, por ejemplo en la pestaña de factores primero se debe de seleccionar el riesgo al cual se le quieren asignar esos factores y después se rellenan los datos en la tabla HTML y se da clic en guardar, este modulo funciona correctamente para hacer el proceso mencionado, faltaría agregar la función de poder agregar más filas a la tabla HTML y después guardar todas esas filas en la tabla de la base de datos:




Matríz Riesgos Mapa Riesgos PTAR Avance Factores Controles Opciones
admin@admin

## Inserción de factores a los riesgos

2022\_1

No. de Factor	Descripción	Clasificación	Tipo
No.	Descripción	Humano	Interno

Guardar Factores

El elemento HTML de tipo select se rellena en base a los riesgos ingresados en la base de datos, usando el siguiente código, en donde primero se solicita el archivo de la conexión a la base de datos, después se asigna la sentencia SQL a una variable, para posteriormente ejecutar esa sentencia, ya dentro de las opciones del select se leen los datos de la consulta mediante un ciclo WHILE, este ejemplo de código se utiliza en algunos otros SELECTS que se encuentran en el sistema, es decir, son alimentados en base a los valores que existen en la base de datos:

```


<body>
<?php include("NavBar.php");
require_once("DataBaseConnect.php");

$select = "SELECT num_riesgo, riesgo FROM riesgos";
$query = mysqli_query($conn,$select);

?>
<form method="post" action="Factores.php">
<div class="container">
<h1 class="mt-6 mb-4 text-3xl xl:text-4xl font-bold text-gray-800 text-center">Inserción de factores a los riesgos</h1>
<div class="row clearfix">
<div class="col-md-12 table-responsive">
<div class="row">
<div class="col">
<select class="form-control" name="no_riesgo">
<option value="0" data-riesgo="" selected disabled>-- Seleccione un Riesgo --</option>
<?php
$i = 0;
while($row = mysqli_fetch_array($query)){
// $data = "data-riesgo =" $row[1]";
}
<option name="no_riesgo" data-riesgo = ""><?php echo $row[$i]></option>
<?php $i + 1;>
</select>
</div>

```

Para la interfaz de controles se intento implementar algo similar, solo que en este caso primero se debe elegir el riesgo, después en base a la elección del riesgo se debería poder elegir el factor perteneciente a ese riesgo para así tener un llenado de la base de manera más ordenada y correcta:


Matríz Riesgos Mapa Riesgos PTAR Avance Factores Controles Opciones
admin@admin

## Inserción de Controles a los Factores

2022\_2
-- Seleccione un Factor --

Los módulos de mapa de riesgos, PTAR y avance primeramente mandan a llamar información de la base de datos, esa información tiene que ver con los riesgos, factores y controles insertados en la parte de la matriz, además de que también cuentan con campos en los que el usuario debe de ingresar información y después esta información debe de ser guardada en la base de datos, para que el usuario administrador después pueda consultar toda esa información, debido a que el diseño de esta interfaz no esta del todo bien la información consultada no aparece de la mejor manera, este sería otro punto a revisar.

Programa de Trabajo Anual Realizado											
No. Riesgo	Descripción del Riesgo	Clasificación del Riesgo	Valor del impacto	Valor de Probabilidad	Cuadrante	Estrategia	No. Factor de Riesgo	Factor de Riesgo	Descripción de la Acción de control	Unidad Administrativa	
		1	1	1		1	1 2 3 10			Dirección General	
							2022_1	Error			

La parte de la consulta de la información se logro usando este fragmento de código:

```

<?php
require_once("DataBaseConnect.php");
//Carga de responsables de areas
$responsables = "SELECT servidor FROM areas_encargados";
$traerDatos = mysqli_query($conn, $responsables);

//$usuario = strtoupper($conn, $_SESSION["userName"], $usuario);
$consultaPtar = "SELECT num_riesgo, riesgo, clasificacion_riesgo_seleccion, valoracion_inicial_grado_impacto, valoracion_inicial_grado_probabilidad FROM riesgos WHERE riesgos.num_riesgo = $usuario";
$run = mysqli_query($conn, $consultaPtar);
$factores = "SELECT num_factor, descripcion FROM factores, riesgos WHERE factores.num_riesgo = riesgos.num_riesgo";
$conFac = mysqli_query($conn, $factores);
while($fila=mysqli_fetch_array($run)){
    $noRiesgo = $fila[0];
    $desRiesgo = $fila[1];
    $claRiesgo = $fila[2];
    $valorImpacto = $fila[3];
    $valorProb = $fila[4];
    $cuadrante = $fila[5];
    $estrategiaUsar = $fila[6];
    /* $noFactor = $fila[7];
    $desFactor = $fila[8]; */
    $accionControl = $fila[7];
    $unidadAdmin = $fila[8];
    $i = 0;

    <tr rowspan="5">
        <td rowspan="5">
            <?php echo $noRiesgo?>
        </td>
        <td rowspan="5">
            <?php echo $desRiesgo?>
        </td>
        <td rowspan="5">
            <?php echo $claRiesgo?>
        </td>
        <td rowspan="5">
            <?php echo $valorImpacto?>
        </td>
        <td rowspan="5">
            <?php echo $valorProb?>
        </td>
        <td rowspan="5">
            <?php echo $cuadrante?>
        </td>
        <td rowspan="5">
            <?php echo $estrategiaUsar?>
        </td>
        <td rowspan="5">
            <?php echo $noFactor?>
        </td>
        <td rowspan="5">
            <?php echo $desFactor?>
        </td>
        <td rowspan="5">
            <?php echo $accionControl?>
        </td>
        <td rowspan="5">
            <?php echo $unidadAdmin?>
        </td>
    </tr>
    <tr></tr>
    <tr></tr>
    <tr></tr>
    <tr></tr>
}

```

Se usa código similar al explicado anteriormente en el cual primero se hace la conexión a la base de datos, después se escriben las sentencias SQL para luego ejecutarlas y guardarlas en lo que es un arreglo que después se desarma para ir mostrando los datos en la tabla HTML que pertenece a esta ventana, algo similar se debe de usar en las otras ventanas de mapa de riesgos y avance.