

UNIVERSIDAD LUTERANA SALVADOREÑA.
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA.
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN.



Materia: Algoritmo II.

Docente: Licenciado Pedro Antonio Treja Noble

Tema Desarrollo: Que tan adicto eres al internet

Alumn@s:

N°	Apellido y Nombres	Carné
1	Rivas Iraheta Yasmin Lorena	RI01134648
2	Martinez Maravilla Emerson Elenilson	MM01134838
3	Molina Servellón Karina Vanessa	MS01134837
4	Ivan Steven Funes Ventura	FV01135561

Fecha de Entrega: Mayo de 2018

Contenido

Introducción	3
Avance II	4
Base de datos a utilizar	4
Capturas del sistema hasta el momento su diseño y funcionamiento con breve explicación	6
Avance III	20
Directorio de la carpeta que contiene el sistema y Breve explicación del código y su funcionamiento.	20

Introducción

Test “¿Qué tan adicto eres al Internet?”, mini sistema que permite guardar preguntas con respuesta posibles y por supuesto la respuesta correcta, también guardar lo que el usuario va respondiendo según la pregunta, para posteriormente hacer una comparación de las respuestas que ha dado el usuario con la respuesta correcta que ya se ha guardado por defecto; Diseño, desarrollo y base de datos utilizada para la realización del test, del funcionamiento de cada uno y el para que se está utilizando en el código según los campos que tiene cada tabla; El diseño que se tiene hasta el momento, que posiblemente este cambie más adelante, a llegado el tiempo de entrega del proyecto.

Avance II

Base de datos a utilizar

```
db_test.sql x
1 -- MySQL Workbench Forward Engineering
2
3 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
6
7 -----
8 -- Schema db_test
9 -----
10
11 -----
12 -- Schema db_test
13 -----
14 CREATE SCHEMA IF NOT EXISTS `db_test` DEFAULT CHARACTER SET utf8 ;
15 USE `db_test` ;
16
17 -----
18 -- Table `db_test`.`usuario`
19 -----
20 CREATE TABLE IF NOT EXISTS `db_test`.`usuario` (
21   `id` INT NOT NULL AUTO_INCREMENT,
22   `nombre` VARCHAR(45) NULL,
23   `edad` VARCHAR(45) NULL,
24   `genero` VARCHAR(45) NULL,
25   `usuario` VARCHAR(45) NULL,
26   `contra` VARCHAR(45) NULL,
27   `tipo` VARCHAR(45) NULL,
28   PRIMARY KEY (`id`))
29 ENGINE = InnoDB;
30
```

Se alojan todos los usuarios operarios con sus datos personales, ocupan el sistema test.

```
31
32 -----
33 -- Table `db_test`.`pregunta`
34 -----
35 CREATE TABLE IF NOT EXISTS `db_test`.`pregunta` (
36   `id` INT NOT NULL AUTO_INCREMENT,
37   `pregunta` VARCHAR(45) NULL,
38   `res_true` VARCHAR(45) NULL,
39   `res_false1` VARCHAR(45) NULL,
40   `res_false2` VARCHAR(45) NULL,
41   `res_false3` VARCHAR(45) NULL,
42   PRIMARY KEY (`id`))
43 ENGINE = InnoDB;
44
45 -----
46 -- Table `db_test`.`test`
47 -----
48
49 CREATE TABLE IF NOT EXISTS `db_test`.`test` (
50   `id` INT NOT NULL AUTO_INCREMENT,
51   `fecha` DATE NULL,
52   `hora` VARCHAR(45) NULL,
53   `id_usuario` INT NOT NULL,
54   PRIMARY KEY (`id`, `id_usuario`),
55   INDEX `fk_test_usuario_idx` (`id_usuario` ASC),
56   CONSTRAINT `fk_test_usuario`
57     FOREIGN KEY (`id_usuario`)
58     REFERENCES `db_test`.`usuario` (`id`)
59     ON DELETE NO ACTION
60     ON UPDATE NO ACTION)
61 ENGINE = InnoDB;
62
```

En tabla pregunta se guardan las preguntas del test con su respuestas: correcta y todas las incorrectas. También la test, esta guardando la hora y fecha, mandando a llamar al usuario que hizo dicho test.

```

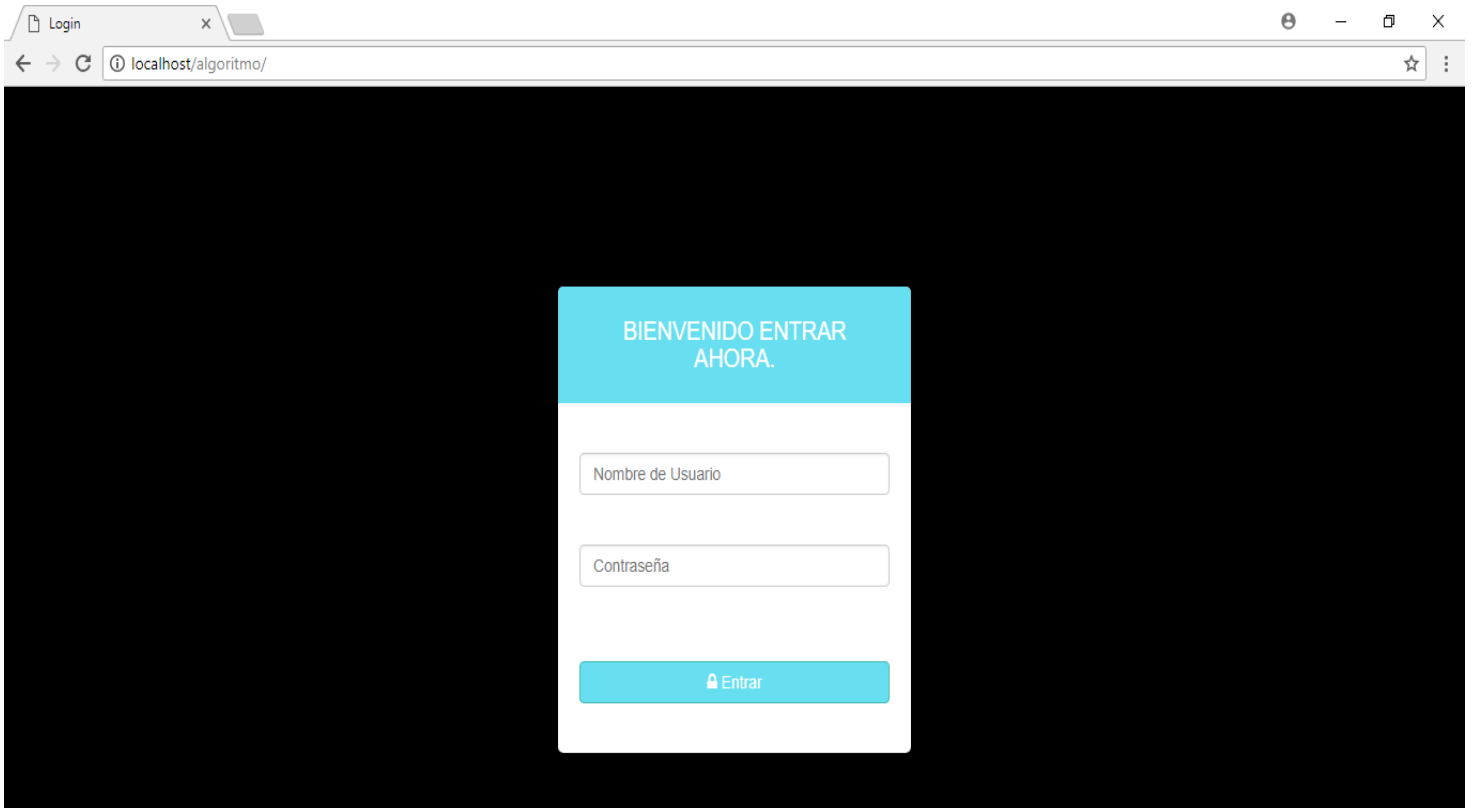
63
64 -----
65 -- Table `db_test`.`respuesta`
66 -----
67 CREATE TABLE IF NOT EXISTS `db_test`.`respuesta` (
68   `id` INT NOT NULL,
69   `respuesta` VARCHAR(45) NULL,
70   `true/false` VARCHAR(45) NULL,
71   `id_pregunta` INT NOT NULL,
72   `id_test` INT NOT NULL,
73   PRIMARY KEY (`id`, `id_pregunta`, `id_test`),
74   INDEX `fk_respuesta_preguntal_idx` (`id_pregunta` ASC),
75   INDEX `fk_respuesta_test1_idx` (`id_test` ASC),
76   CONSTRAINT `fk_respuesta_preguntal`
77     FOREIGN KEY (`id_pregunta`)
78     REFERENCES `db_test`.`pregunta` (`id`)
79     ON DELETE NO ACTION
80     ON UPDATE NO ACTION,
81   CONSTRAINT `fk_respuesta_test1`
82     FOREIGN KEY (`id_test`)
83     REFERENCES `db_test`.`test` (`id`)
84     ON DELETE NO ACTION
85     ON UPDATE NO ACTION)
86 ENGINE = InnoDB;
87
88
89 SET SQL_MODE=@OLD_SQL_MODE;
90 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
91 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
92

```

Y la ultima tabla de la db, la tabla respuesta, verifica si la respuesta dada es falsa o verdadera, mandando a llamar al test guardado y a la preguntas que fueron guardadas con respuesta correcta y las falsas tambien. Entonces lo verifica con el campo respuesta, para asi decirle al usuario si es o no es correcta.

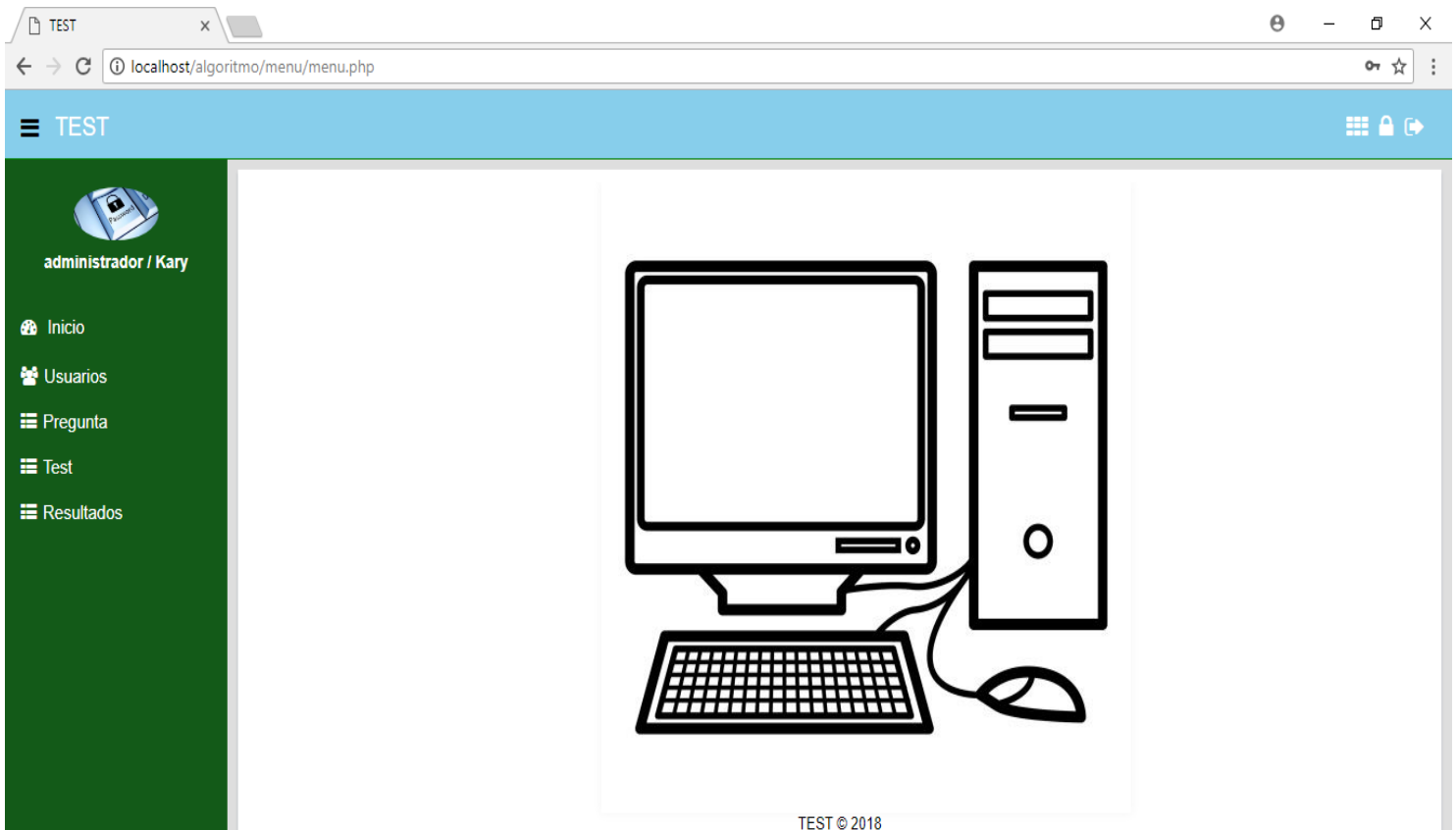
Capturas del sistema hasta el momento su diseño y funcionamiento con breve explicación

Inicio de sesión



Es la página inicial que un usuario ve.

Menú del administrador



En este menú, es cuando un administrador inicia sesión, podrá observar todas las opciones del sistema, como creación de nuevos registros, modificación y eliminación de los mismos.

Visualizar todo tipo de registro que estén en cada una de las opciones del menú.

Abajo del logo, se visualiza el usuario administrador que ha iniciado sesión.

Menú usuario



Este es el menú de usuario que ha iniciado sesión para realizar un test, y ver sus resultados.

Listado de usuarios registrados (Administradores)

The screenshot shows a web browser window with the URL `localhost/algorithmo/usuario/mostrar.php`. The page has a light blue header with the word "TEST" and a dark green sidebar on the left. The sidebar contains a user profile icon for "administrador / Kary" and a menu with items: Inicio, Usuarios, Pregunta, Test, and Resultados. The main content area has a blue navigation bar with "Administrador" and "Cliente" tabs. Below this is a section titled "Listado de Administradores" with a search bar labeled "Buscar cliente" and a "+ Agregar" button. A table displays one record with the following data:

N°	Nombre	Edad	Genero	Usuario	tipo	Opciones
1	Karina Servellon	22	femenino	Kary	administrador	Acciones

Below the table, it says "Mostrando 1 registros de 1" and there are navigation buttons: "Anterior", "1", and "Siguiente".

El administrador podrá listar los demás usuarios administradores, realizar acciones con estos registros.

Listado de usuarios registrados (Clientes)

The screenshot shows a web browser window with the URL `localhost/algorithmo/usuario/mostrar1.php`. The application has a blue header with the word "TEST" and a green sidebar on the left. The sidebar contains a user profile for "administrador / Kary" and a menu with items: Inicio, Usuarios, Pregunta, Test, and Resultados. The main content area has two tabs: "Administrador" and "Cliente", with "Cliente" selected. Below the tabs is a search bar labeled "Buscar cliente" with a "Buscar" button and an "+ Agregar" button. A table titled "Listado de clientes" displays one record:

Nº	Nombre	Edad	Genero	Usuario	Tipo	Opciones
2	Emerson Martinez	22	masculino	Emer	cliente	Acciones

Below the table, it says "Mostrando 1 registros de 1". A context menu is open over the "Acciones" column, showing three options: "Editar" (with a pencil icon), "Desactivar" (with a trash icon), and "Contraseña" (with a key icon).

El administrador podrá listar los demás usuarios no administradores, realizar acciones con estos registros.

Formulario para registro de un nuevo usuario

Nuevo Usuario. X

Ingrese Datos

Nombre:

Edad:

Genero:

Usuario:

Contraseña:

Tipo de usuario:

Esta vista es un formulario de creación de nuevo usuario, con sus datos necesarios para el registro del mismo.

Formulario para modificar usuario

Editar ×

Modifique Datos

Nombre:

Edad:

Genero:

Usuario:

Tipo de usuario:

Vista de formulario para modificar a un usuario.

Formulario para modificar contraseña del usuario

Editar contraseña ×






Modifique su contraseña

Nueva Contraseña:

Esta vista es exclusivamente especial para el cambio de contraseña del usuario, en donde podrá cambiarla cuantas veces desee.

Listado de preguntas (compartido en 2 paginas)

Listado de Preguntas

N°	Pregunta	Correcta	Falsa 1	Falsa 2	Falsa 3	Opciones
1	$2+2+1+9$	14	1	2	3	
2	$3*9$	27	8	9	8	
3	$2+1$	3	5	8	9	
4	$1+1$	2	6	9	3	
5	$10+15$	35	40	25	30	

Mostrando 5 registros de 10

[« Anterior](#) [1](#) [2](#) [Siguiente »](#)

Vista para listar las preguntas guardadas del test. (ojo, registros de prueba)

Formulario para modificar las preguntas

Editar ×

Modifique Datos

Pregunta:

Respuesta correcta:

Respuesta false #1:

Respuesta falsa #2:

Respuesta falsa #3:

Vista para modificar aquellos registros de preguntas.

Inicio de serie de preguntas del TEST

Listado de Preguntas

2+2+1+9 ×

Seleccione la respuesta correcta

14 :

1 :

2 :

3 :

Este es el test, el cual donde das click en iniciar, este empieza una serie de preguntas del test “¿Qué tan Adicto eres al Internet?”. Este es iterativo para que después de una pregunta se muestra la siguiente, y la siguiente y así el usuario podrá ir contestando. (ojo, son registros de prueba).

Muestra resultados una vez terminado el TEST

Detalle de respuestas

Fecha	Hora	Nombre	Calificación
2018-05-20	05:45:47	Karina Servellon	7
N°	Pregunta	Respuesta	
1	$2+2+1+9$	1	
2	$3*9$	27	
3	$2+1$	3	
4	$1+1$	2	
5	$10+15$	40	
6	$9-1$	8	
7	$3*3$	9	
8	$2*5$	10	
9	$12+3$	17	
10	$9*2$	18	

Esta vista permite al usuario ver todo lo que ha ido contestando, y su calificación obtenida.

**Muestra informe de los TEST realizados (sin detalles de respuestas).
Solo disponible para administrador.**

Listado de TEST

Buscar cliente

N°	Fecha	Hora	Usuario	Calificación	Opciones
3	2018-05-20	05:45:47	Karina Servellon	7	Detalles
2	2018-05-16	06:15:18	Emerson Martinez	6	Detalles
1	2018-05-16	06:13:48	Karina Servellon	7	Detalles

Mostrando 3 registros de 3

[« Anterior](#) [1](#) [Siguiente »](#)

Muestra de registros de test a administrador, visualizando la fecha y hora, calificaciones de "x" usuario.

Detalles de las respuestas del TEST. Solo disponible para administrador.

Detalle de respuestas

N°	Pregunta	Respuesta	Resultado
1	$2+2+1+9$	1	incorrecta
2	$3*9$	27	correcta
3	$2+1$	3	correcta
4	$1+1$	2	correcta
5	$10+15$	40	incorrecta
6	$9-1$	8	correcta
7	$3*3$	9	correcta
8	$2*5$	10	correcta
9	$12+3$	17	incorrecta
10	$9*2$	18	correcta

Cuando el administrador da click en ver detalles de cierto test, podrá visualizar con especificación si la respuesta que el usuario dio es correcta o incorrecta.

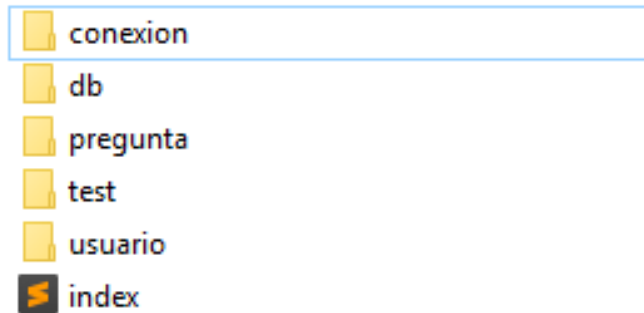
Opción de bloquear el sistema sin necesidad de cerrar sesión



Esta opción permite dar un tiempo determinado de bloquear pantalla de la sesión iniciada, después de un tiempo de inactividad.

Avance III

Directorio de la carpeta que contiene el sistema y Breve explicación del código y su funcionamiento.

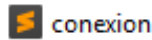


index.php

```
index.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Inicio</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <a href="usuario/mostrar.php">Usuario</a><br><br>
9 <a href="Pregunta/mostrar.php">Pregunta</a><br><br>
10 <a href="test/guardar_test.php">Test</a><br><br>
11 <?php
12   date_default_timezone_set('America/El_Salvador');
13   echo date('d/m/Y')."<br>";
14   echo date('h:i:s a')."<br>";
15   ?>
16   <!--div style="text-align:center;padding:1em 0;"> <h3><a style="te
/country/sv"><span style="color:gray;">Hora actual en</span><br />
www.zeitverschiebung.net/clock-widget-iframe-v2?language=es&size=m
height="115" frameborder="0" seamless></iframe> </div-->
17 </body>
18 </html>
```

El archivo index.php es lo primero que vemos despues de haber iniciado sesión.

conexion

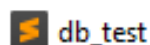


```
1 <?php
2 //Funcion que contiene el enlace a la base de datos.
3 function conexion(){
4     $conn = null;
5     $host = 'localhost';
6     $db = 'db_test';
7     $user = 'root';
8     $pwd = '';
9
10    try{
11        $conn = new PDO('mysql:host='.$host.'; dbname='.$db,$user,$pwd);
12        //echo 'Conexion satisfactoria.<br>';
13    }
14    }catch(PDOException $e){
15        echo "<style='background-color:lightgrey'>
16
17        ";
18        echo <hr>;
19        echo '<br><center><h1 style="font-size:300%"><p><font color="red">;¡No se puede conectar con la base de datos!!</font></p></
20        h1>';
21        echo "<embed src='../img/cad.png' heighth='50' width='50'></embed><br><br><progress id='p' max='70'> <span>0</span>%</progress>
22        <br>";
23        echo "<hr width='80%' color='black' size='8' /></center>";
24        echo "<p>Posibles causas:</p>
25        <ol>
26            <li>Conexión con el servidor pérdida. </li>
27            <li>Base de datos no encontrada. </li>
28            <li>Conexión expirada. </li>
29            <li>Clave o nombres incorrectos. </li>
30            <li>La base de datos fue removida. </li>
31        </ol>"
32        ;
33
34        echo "<center><h2 style='color:green'>Si el problema persiste consulte al Técnico más cercano att: el creador</h2></center>";
35    }
36    exit();
37    return $conn;
38 }
39 //conexion();
40 ?>
```

En conexion.php se hace la conexión con el servidor y la base de datos.








Si dicha conexión no se realiza, podrá ver un pequeña vista con estilo diciendole que “no se puede conectar con la base de datos”.

db



Este es el archivo mostrado anteriormente a detalle en la sesión primera de avance II de este documento.

pregunta

-  editar
-  editar_formu
-  eliminar
-  guardar
-  guardar_formu
-  mostrar
-  pregunta

editar.php

```
1 <?php
2
3 if(!empty($_POST)){
4
5
6     $id = $_POST["id"];
7     $pregunta = $_POST["pregunta"];
8     $res_true = $_POST["res_true"];
9     $res_false1 = $_POST["res_false1"];
10    $res_false2 = $_POST["res_false2"];
11    $res_false3 = $_POST["res_false3"];
12
13        include("pregunta.php");
14        $send = new pregunta();
15
16        $save = $send->editar($id, $pregunta,$res_true,$res_false1,$res_false2,$res_false3);
17        header("location:mostrar.php");
18
19
20
21    }
22
23
24 ?>
```

Es el archivo .php que permite editar lo que es la pregunta y sus respuesta correcta y todas las falsas que ahora quiere cambiar y en su lugar modificar cualquier campo.

editar.formu.php

```
editar_formu.php x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Editar Pregunta</title>
5 <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9     if(!empty($_POST)){
10         $id=$_POST["id"];
11     }
12     include_once('../conexion/conexion.php');
13     $connection = conexion();
14
15     $sql = "SELECT * FROM pregunta WHERE id=".$id." ";
16     $query = $connection->prepare($sql);
17
18     $query->execute();
19     $rowcount = $query->rowcount();
20
21     $model = array();
22     while($rows = $query->fetch())
23     {
24         $model[] = $rows;
25     }
26     ?>
27 <?php
28
29 foreach($model as $row)
30 {
31 ?>
32 <form action="editar.php" method="post">
33     <input type="hidden" name="id" value="<?php echo $row['id']; ?>">
34     Pregunta: <input type="text" name="pregunta" value="<?php echo $row['pregunta']; ?>"><br>
35     Respuesta Correcta: <input type="text" name="res_true" value="<?php echo $row['res_true']; ?>"><br>
36     Respuesta Incorrecta: <input type="text" name="res_false1" value="<?php echo $row['res_false1']; ?>" ><br>
37     Respuesta Incorrecta: <input type="text" name="res_false2" value="<?php echo $row['res_false2']; ?>" ><br>
38     Respuesta Incorrecta: <input type="text" name="res_false3" value="<?php echo $row['res_false3']; ?>" ><br>
39     <input type="submit" name="enviar" value="Editar">
40 </form>
41 <?php
42 }
```

Aquí se encuentra el html con php en donde hace uso de la conexión a la db y selecciona la tabla pregunta para finalmente mostrar con un value en los input y recordar el valor que ya está guardado antes, para que el usuario modifique el dato que desee.

eliminar.php

```
eliminar.php x
1 <?php
2 if (!empty($_POST['id'])) {
3     $id=$_POST['id'];
4
5     include 'pregunta.php';
6
7     $enviar = new pregunta();
8
9     $enviar->desactivar($id);
10    header("location:mostrar.php");
11 }
12 ?>
```

A travez del id se puede eliminar cualquier registro que se mostro y que no se desee en la tabla pregunta.

guardar.php

```
guardar.php
1 |<?php
2
3 | if(!empty($_POST)){
4
5 |     include_once('../conexion/conexion.php');
6
7 |     $conn = conexion();
8
9 |     $pregunta = $_POST["pregunta"];
10 |     $res_true = $_POST["res_true"];
11 |     $res_false1 = $_POST["res_false1"];
12 |     $res_false2 = $_POST["res_false2"];
13 |     $res_false3 = $_POST["res_false3"];
14
15 |     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
16 | $consult = $conn->prepare("SELECT * FROM pregunta where pregunta = '$pregunta'");
17 |     $consult->execute();
18
19 |     $data = $consult->fetch(PDO::FETCH_ASSOC);
20 |     $preguntas = $data['pregunta'];
21 |     if(($preguntas == "") or ($pregunta != $preguntas)){
22
23 |         include("pregunta.php");
24 |         $send = new pregunta();
25
26 |         $save = $send->guardar($pregunta,$res_true,$res_false1,$res_false2,$res_false3);
27 |         echo $save;
28 |         header("location:mostrar.php");
29 |     }elseif ($preguntas == $pregunta) {
30 |         echo "<script> alert('Ya existe la pregunta: $pregunta. No puede repetir datos')</script>";
31 |         echo "<meta http-equiv='refresh' content='0'; url=http://localhost/proyecto_algoritmo/pregunta/mostrar.php/ >";
32 |     }
33 | }
34 | }
35
36 | ?>
```

guardar.php permite conectar con la db, y seleecionar la tabla pregunta, para hacer la insercion del registro hecho en el formulario.

guardar_formu.php

```
guardar_formu.php
1 |<!DOCTYPE html>
2 |<html>
3 |<head>
4 |    <title>Formulario de registro</title>
5 |    <meta charset="utf-8">
6 |</head>
7 |<body>
8 |    <form action="guardar.php" method="post">
9 |        Pregunta: <input type="text" name="pregunta"><br>
10 |        Respuesta Correcta: <input type="text" name="res_true"><br>
11 |        Respuesta Incorrecta: <input type="text" name="res_false1"><br>
12 |        Respuesta Incorrecta: <input type="text" name="res_false2"><br>
13 |        Respuesta Incorrecta: <input type="text" name="res_false3"><br>
14 |        <input type="submit" name="enviar" value="Guardar">
15 |    </form>
16 |</body>
17 |</html>
```

Este es el formulario donde el usuario coloca los datos de la pregunta a guardar.

mostrar.php

```
1 <?php
2 include_once('../conexion/conexion.php');
3 $connection = conexion();
4
5 $sql = "SELECT * FROM pregunta ORDER BY id ASC ";
6 $query = $connection->prepare($sql);
7
8 $query->execute();
9 $rowcount = $query->rowCount();
10
11 $model = array();
12 while($rows = $query->fetch())
13 {
14     $model[] = $rows;
15 }
16
17 ?>
18 <div class="form-panel"><hr>
19 <form action="guardar_formu.php" method="post">
20     <a href=" ../index.php">Menú principal</a><br><br>
21     <input type="submit" name="nuevo" value="Nuevo">
22 </form>
23 <hr>
24 <center>
25 <div class="table table-responsive">
26 <table class="table table-striped table-bordered" border="1">
27 <tr>
28     <td width='5%' align="center" rowspan="2" >Nº</td>
29     <td align="center" rowspan="2" >Pregunta</td>
30     <td align="center" colspan="4" >Respuestas</td>
31     <td align="center" colspan="2" rowspan="2" >Opciones </td>
32 </tr>
33 <tr>
34 <td align="center">Correcta</td>
35 <td align="center">Incorrecta</td>
36 <td align="center">Incorrecta</td>
37 <td align="center">Incorrecta</td>
38 </tr>
39 <?php
```

```
40
41 foreach($model as $row)
42 {
43     echo "<tr align='center'>";
44     echo "<td>" . $row['id'] . "</td>";
45     echo "<td align='center' width='10%'>" . $row['pregunta'] . "</td>";
46     echo "<td align='center' width='10%'>" . $row['res_true'] . "</td>";
47     echo "<td align='center' width='10%'>" . $row['res_false1'] . "</td>";
48     echo "<td align='center' width='10%'>" . $row['res_false2'] . "</td>";
49     echo "<td align='center' width='10%'>" . $row['res_false3'] . "</td>";
50 }
51 <td align="center" width="10%"><form action="editar_formu.php" method="post">
52     <input type="hidden" name="id" value="<?php echo $row['id']; ?>"><input type="submit" name="editar" value="Editar"></form></td>
53 <td align="center" width="10%"><form action="eliminar.php" method="post"><input type="hidden" name="id" value="<?php echo $row['id']; ?>"><input type="submit" name="eliminar" value="Eliminar"></form></td>
54 <?php
55
56     }
57 ?>
58 </table>
59 </div>
60 </center>
61 </body>
62 </html>
```

Se hace conexión con db para seleccionar la tabla pregunta en donde consultamos para mostrar y a la misma vez en esta misma vista mostrar la acción que acabamos de hacer, que es ver los registros, listarlos.









pregunta.php

```
pregunta.php x
1 |<?php
2
3 class pregunta{
4     //definir variables para la clase usuario segun la tabla usuario en la base de datos
5     var $id;
6     var $pregunta;
7     var $res_true;
8     var $res_false1;
9     var $res_false2;
10    var $res_false3;
11
12    //funcion que guardar los datos del usuario
13
14    function guardar($pregunta, $res_true, $res_false1, $res_false2, $res_false3)
15    {
16        try{
17            include_once('../conexion/conexion.php');
18
19            $conn = conexion();
20
21
22            //prepare el sql and bind parameters
23            $stmt = $conn->prepare('INSERT INTO pregunta(pregunta,res_true,res_false1,res_false2,res_false3) VALUES(:a,:b,:c,:d,:e)');
24
25
26            $stmt->bindParam(':a',$a);
27            $stmt->bindParam(':b',$b);
28            $stmt->bindParam(':c',$c);
29            $stmt->bindParam(':d',$d);
30            $stmt->bindParam(':e',$e);
31
32            //insert a row
33            $a = $pregunta;
34            $b = $res_true;
35            $c = $res_false1;
36            $d = $res_false2;
37            $e = $res_false3;
38
39            $stmt->execute();
40            echo "<script> alert('Registro Almacenado')</script>";
41
```

```
45    }catch(PDOException $e){
46        echo "Error:". $e->getMessage();
47
48    }
49 }
50
51 function editar($id, $pregunta, $res_true, $res_false1, $res_false2, $res_false3)
52 {
53     require_once('../conexion/conexion.php');
54     $conn = conexion();
55     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
56
57
58     $sql = "UPDATE pregunta SET pregunta = '$pregunta', res_true = '$res_true', res_false1 = '$res_false1', res_false2 = '$res_false2', res_false3 = '$res_false3' WHERE id = '$id'";
59
60     $stmt = $conn->prepare($sql);
61     $stmt->execute();
62
63     header("location:mostrar.php");
64
65 }
66
67 //funcion que sirve para eliminar los datos de platillo
68 function desactivar($id){
69
70     require_once('../conexion/conexion.php');
71     $conn = conexion();
72
73     try{
74
75         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
76         $sql = "DELETE FROM pregunta WHERE id = '$id'";
77
78         $conn->exec($sql);
79
80
81     }catch(PDOException $e){
82         echo "Error:". $e->getMessage();
83
84     }
85
```

La clase donde se guarda el modelo que se va ocupar para cada una de las acciones hechas anteriormente, se podría decir que es como el patron que debe seguir.

test

-  editar
-  editar_formu
-  eliminar
-  guardar_formu
-  guardar_test
-  mostrar
-  respuesta
-  test

editar.php

```
1 |<?php
2
3 | if(!empty($_POST)){
4
5
6 |     $idPersona = $_POST["idPersona"];
7 |     $nombre = $_POST["nombre"];
8 |     $edad = $_POST["edad"];
9 |     $genero = $_POST["genero"];
10
11 |     include("persona.php");
12 |     $send = new persona();
13
14 |     $save = $send->editar($idPersona, $nombre, $edad, $genero);
15 |     header("location:mostrar.php");
16
17
18
19 | }
20
21
22 | ?>
```

El código php que permite actualizar el registro con sus nuevos datos volverlos a guardar.

editar_formu.php

```
editar_formu.php x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Editar Persona</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <?php
9     if(!empty($_POST)){
10      $idPersona=$_POST["idPersona"];
11    }
12    include_once('../conexion/conexion.php');
13    $connection = conexion();
14
15    $sql = "SELECT * FROM persona WHERE idPersona=".$idPersona." ";
16    $query = $connection->prepare($sql);
17
18    $query->execute();
19    $rowcount = $query->rowcount();
20
21    $model = array();
22    while($rows = $query->fetch())
23    {
24      $model[] = $rows;
25    }
26    ?>
27 <?php
28
29 foreach($model as $row)
30 {
31 ?>
32   <form action="editar.php" method="post">
33     <input type="hidden" name="idPersona" value="<?php echo $row['idPersona']; ?>">
34     Nombre: <input type="text" name="nombre" value="<?php echo $row['nombre']; ?>"><br>
35     Edad: <input type="text" name="edad" value="<?php echo $row['edad']; ?>"><br>
36     Genero: <select name="genero">
37       <?php
38         if ($row['genero']=="masculino"){
39           echo "<option value='masculino' slected>Masculino</option>";
40           echo "<option value='femenino'>Femenino</option>";
41         }else{
42           echo "<option value='masculino'>Masculino</option>";
43           echo "<option value='femenino' selected>Femenino</option>";
44         }
45       ?>
46     </select><br>
47     <input type="submit" name="enviar" value="Editar">
48   </form>
49 <?php
50 }
51 ?>
52 </body>
53 </html>
```

Es donde el usuario coloca los nuevos datos reemplazando los que quiere actualizar.

eliminar.php

```
eliminar.php x
1 <?php
2 if (!empty($_POST['idPersona'])) {
3     $id=$_POST['idPersona'];
4
5     include 'persona.php';
6
7     $enviar = new persona();
8
9     $enviar->desactivar($id);
10    header("location:mostrar.php");
11 }
12 ?>
```

Eliminar a travez del id.

guardar_formu.php

```
guardar_formu.php x
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Formulario de registro</title>
5     <meta charset="utf-8">
6 </head>
7 <body>
8     <form action="guardar.php" method="post">
9         Nombre: <input type="text" name="nombre"><br>
10        Edad: <input type="text" name="edad"><br>
11        Genero: <select name="genero">
12            <option value="masculino">Masculino</option>
13            <option value="femenino">Femenino</option>
14        </select><br>
15        <input type="submit" name="enviar" value="Guardar">
16    </form>
17 </body>
18 </html>
```

El formulario donde escribiremos los datos.

guardar_test.php

```
guardar_test.php x
1 <?php
2
3 if(empty($_POST)){
4
5     include_once('../conexion/conexion.php');
6     $conn = conexion();
7
8     /*$nombre = $_POST["nombre"];
9     $edad = $_POST["edad"];
10    $genero = $_POST["genero"];*/
11
12
13    /*$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14    $consult = $conn->prepare("SELECT * FROM persona where nombre = '$nombre'");
15    $consult->execute();
16
17    $data = $consult->fetch(PDO::FETCH_ASSOC);
18    $nombres = $data['nombre'];
19    if(($nombres == "" ) or ($nombre != $nombres)){ */
20    date_default_timezone_set('America/El_Salvador');
21    $fecha = date('d/m/Y');
22    $hora = date('h:i:s');
23    $id_usuario = "1";
24    include("test.php");
25    $send = new test();
26
27    $save = $send->guardar($fecha,$hora,$id_usuario);
28
29    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
30    $sql = "SELECT max(id) FROM test";
31    $q = $conn->query($sql);
32    $rows = $q->fetchAll();
33
34    echo $idext = $rows[0][0];
35    echo " <a href='../index.php'>Menú principal</a><br><br>";
36    include('respuesta.php');
37    $ped = new respuesta();
38
39    //$ped->guardar($idext,$consumo_id,$cantidad,$total);
40
41    //header("location:mostrar.php");
42
```

Permite guardar test insertandolo en la tabla test de la db.

mostrar.php

```
mostrar.php x
1 <?php
2 include_once('../conexion/conexion.php');
3 $connection = conexion();
4
5 $sql = "SELECT * FROM persona ORDER BY idPersona ASC ";
6 $query = $connection->prepare($sql);
7
8 $query->execute();
9 $rowcount = $query->rowCount();
10
11 $model = array();
12 while($rows = $query->fetch())
13 {
14     $model[] = $rows;
15 }
16
17 ?>
18 <div class="form-panel"><hr>
19 <form action="guardar_formu.php" method="post">
20 | <input type="submit" name="nuevo" value="Nuevo">
21 </form>
22 <hr>
23 <center>
24 <div class="table table-responsive">
25 <table class="table table-striped table-bordered">
26 <tr>
27 <td width='5%' align="center">Nº</td>
28 <td align="center">Nombre</td>
29 <td align="center">Edad</td>
30 <td align="center" colspan="3">Genero</td>
31 </tr>
32 <?php
33
34 foreach($model as $row)
35 {
36     echo "<tr align='center'>";
37     echo "<td>".$row['idPersona']. "</td>";
38     echo "<td align='center' width='10%'>".$row['nombre']. "</td>";
39     echo "<td align='center' width='10%'>".$row['edad']. "</td>";
40     echo "<td align='center' width='10%'>".$row['genero']. "</td>";
41     ?>
```

Opera y muestra los datos.

respuesta.php

```
1 <?php
2
3 class respuesta{
4     //definir variables para la clase usuario segun la tabla usuario en la base de datos
5     var $id;
6     var $nombre;
7     var $edad;
8     var $genero;
9
10    //funcion que guardar los datos del usuario
11
12    function guardar($nombre, $edad, $genero)
13    {
14        try{
15            include_once('../conexion/conexion.php');
16
17            $conn = conexion();
18
19
20            //prepare el sql and bind parameters
21            $stmt = $conn->prepare('INSERT INTO persona(nombre,edad,genero) VALUES(:a,:b,:c)');
22
23
24            $stmt->bindParam(':a',$a);
25            $stmt->bindParam(':b',$b);
26            $stmt->bindParam(':c',$c);
27
28            //insert a row
29            $a = $nombre;
30            $b = $edad;
31            $c = $genero;
32
33            $stmt->execute();
34            echo "<script> alert('Registro Almacenado')</script>";
35
36
37
38
```

```
39
40    }catch(PDOException $e){
41        echo "Error:".$e->getMessage();
42    }
43 }
44
45 function editar($id,$nombre,$edad,$genero)
46 {
47     require_once('../conexion/conexion.php');
48     $conn = conexion();
49     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
50
51
52     $sql = "UPDATE persona SET nombre = '$nombre', edad = '$edad', genero = '$genero' WHERE idPersona = '$id'";
53
54     $stmt = $conn->prepare($sql);
55     $stmt->execute();
56
57     header("location:mostrar.php");
58
59 }
60
61 //funcion que sirve para eliminar los datos de platillo
62 function desactivar($id){
63
64     require_once('../conexion/conexion.php');
65     $conn = conexion();
66
67     try{
68
69         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
70         $sql = "DELETE FROM persona WHERE idPersona = '$id'";
71
72         $conn->exec($sql);
73
74
75     }catch(PDOException $e){
76         echo "Error:".$e->getMessage();
77
78     }
```

Es el modelo a seguir para tabla respuesta.

test.php

```
1 <?php
2
3 class test{
4     //definir variables para la clase usuario segun la tabla usuario en la base de datos
5     var $id;
6     var $fecha;
7     var $hora;
8     var $id_usuario;
9
10    //funcion que guardar los datos del usuario
11
12    function guardar($fecha, $hora, $id_usuario)
13    {
14        try{
15            include_once('../conexion/conexion.php');
16
17            $conn = conexion();
18
19
20            //prepare el sql and bind parameters
21            $stmt = $conn->prepare('INSERT INTO test(fecha,hora,id_usuario) VALUES(:a,:b,:c)');
22
23
24            $stmt->bindParam(':a',$a);
25            $stmt->bindParam(':b',$b);
26            $stmt->bindParam(':c',$c);
27
28            //insert a row
29            $a = $fecha;
30            $b = $hora;
31            $c = $id_usuario;
32
33            $stmt->execute();
34            echo "<script> alert('Registro Almacenado')</script>";
35
36
37
38
39        }catch(PDOException $e){
40            echo "Error:".$e->getMessage();
41
```

```
41
42    }
43    }
44
45    function editar($id,$nombre,$edad,$genero)
46    {
47        require_once('../conexion/conexion.php');
48        $conn = conexion();
49        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
50
51
52        $sql = "UPDATE persona SET nombre = '$nombre', edad = '$edad', genero = '$genero' WHERE idPersona = '$id'";
53
54        $stmt = $conn->prepare($sql);
55        $stmt->execute();
56
57        header("location:mostrar.php");
58    }
59
60
61    //funcion que sirve para eliminar los datos de platillo
62    function desactivar($id){
63
64        require_once('../conexion/conexion.php');
65        $conn = conexion();
66
67        try{
68
69            $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
70            $sql = "DELETE FROM persona WHERE idPersona = '$id'";
71
72            $conn->exec($sql);
73
74
75            }catch(PDOException $e){
76            echo "Error:".$e->getMessage();
77
78        }
79    }
80
81
```

Modelo a seguir para tabla test.

usuario

- editar
- editar_formu
- eliminar
- guardar
- guardar_formu
- mostrar
- usuario

editar.php

```
editar.php x
1 <?php
2
3 if(!empty($_POST)){
4
5
6     $id = $_POST["id"];
7     $nombre = $_POST["nombre"];
8     $edad = $_POST["edad"];
9     $genero = $_POST["genero"];
10    $usuario = $_POST["usuario"];
11    $contra = $_POST["contra"];
12    $tipo = $_POST["tipo"];
13
14        include("usuario.php");
15        $send = new usuario();
16
17        $save = $send->editar($id, $nombre, $edad, $genero, $usuario, $contra, $tipo);
18        header("location:mostrar.php");
19
20
21    }
22 }
23
24
25 ?>
```

Editar al usuario del cual necesitamos cambiarle un campo o todos.

editar_formu.php

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Editar Persona</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <?php
9     if(!empty($_POST)){
10       $id=$_POST["id"];
11     }
12     include_once('../conexion/conexion.php');
13     $connection = conexion();
14
15     $sql = "SELECT * FROM usuario WHERE id=".$id." ";
16     $query = $connection->prepare($sql);
17
18     $query->execute();
19     $rowcount = $query->rowCount();
20
21     $model = array();
22     while($rows = $query->fetch())
23     {
24       $model[] = $rows;
25     }
26   ?>
27 <?php
28
29 foreach($model as $row)
30 {
31 ?>
32   <form action="editar.php" method="post">
33     <input type="hidden" name="id" value="<?php echo $row['id']; ?>">
34     Nombre: <input type="text" name="nombre" value="<?php echo $row['nombre']; ?>"><br>
35     Edad: <input type="text" name="edad" value="<?php echo $row['edad']; ?>"><br>
36     Genero: <select name="genero">
37       <?php
38         if ($row['genero']=="masculino"){
39           echo "<option value='masculino' slected>Masculino</option>";
40           <option value='femenino'>Femenino</option>;
41         }else{
42           echo "<option value='masculino'>Masculino</option>
```

```
44     }
45   ?>
46
47   </select><br>
48
49   Usuario: <input type="text" name="usuario" value="<?php echo $row['usuario']; ?>"><br>
50   Contraseña: <input type="text" name="contra" value="<?php echo $row['contra']; ?>"><br>
51   Tipo de usuario: <select name="tipo">
52     <?php
53       if ($row['tipo']=="administrador"){
54         echo "<option value='administrador' slected>Administrador</option>";
55         <option value='cliente'>Cliente</option>;
56       }else{
57         echo "<option value='administrador'>Administrador</option>";
58         <option value='cliente' selected>Cliente</option>;
59       }
60     ?>
61
62     </select><br>
63     <input type="submit" name="enviar" value="Editar">
64   </form>
65 <?php
66 }
67 ?>
68 </body>
69 </html>
```

Se escriben los datos en el formulario, pero primero borrando los datos que ya no nos sirven para reemplazarlo.

eliminar.php

```
eliminar.php x
1 <?php
2 if (!empty($_POST['id'])) {
3     $id=$_POST['id'];
4
5     include 'usuario.php';
6
7     $enviar = new usuario();
8
9     $enviar->desactivar($id);
10    header("location:mostrar.php");
11 }
12 ?>
```

Eliminar al usuario a travez del id.

guardar.php

```
guardar.php x
1 <?php
2
3 if(!empty($_POST)){
4
5     include_once('../conexion/conexion.php');
6
7     $conn = conexion();
8
9     $nombre = $_POST["nombre"];
10    $edad = $_POST["edad"];
11    $genero = $_POST["genero"];
12    $usuario = $_POST["usuario"];
13    $contra = $_POST["contra"];
14    $tipo = $_POST["tipo"];
15
16    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
17    $consult = $conn->prepare("SELECT * FROM usuario where nombre = '$nombre'");
18    $consult->execute();
19
20    $data = $consult->fetch(PDO::FETCH_ASSOC);
21    $nombres = $data['nombre'];
22    if(($nombres == "") or ($nombre != $nombres)){
23
24
25        include("usuario.php");
26        $send = new usuario();
27
28        $save = $send->guardar($nombre,$edad,$genero,$usuario,$contra,$tipo);
29        header("location:mostrar.php");
30    }elseif ($nombres == $nombre) {
31        echo "<script> alert('Ya existe la persona $nombre. No puede repetir datos')</script>";
32        echo "<meta http-equiv='refresh' content='0; url=http://localhost/proyecto_algoritmo/usuario/mostrar.php/' >";
33    }
34 }
35
36 ?>
37
```

Codigo php para insertar al usuario.

guardar_formu.php

```
guardar_formu.php x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formulario de registro</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <form action="guardar.php" method="post">
9     Nombre: <input type="text" name="nombre"><br><br>
10    Edad: <input type="text" name="edad"><br><br>
11    Genero: <select name="genero">
12      <option value="masculino">Masculino</option>
13      <option value="femenino">Femenino</option>
14    </select><br><br>
15
16    Usuario: <input type="text" name="usuario"><br><br>
17    Contraseña: <input type="password" name="contra"><br><br>
18    Tipo de usuario: <select name="tipo">
19      <option value="administrador">Administrador</option>
20      <option value="cliente">Cliente</option>
21    </select><br><br>
22    <input type="submit" name="enviar" value="Guardar">
23  </form>
24 </body>
25 </html>
```

Formulario para colocar los datos pertenecientes del usuario.

mostrar.php

```
mostrar.php x
1 <?php
2 include_once('../conexion/conexion.php');
3 $connection = conexion();
4
5 $sql = "SELECT * FROM usuario ORDER BY id ASC ";
6 $query = $connection->prepare($sql);
7
8 $query->execute();
9 $rowcount = $query->rowcount();
10
11 $model = array();
12 while($rows = $query->fetch())
13 {
14     $model[] = $rows;
15 }
16
17 ?>
18 <div class="form-panel"><hr>
19 <form action="guardar_formu.php" method="post">
20     <a href=" ../index.php">Menú principal</a><br><br>
21     <input type="submit" name="nuevo" value="Nuevo">
22 </form>
23 <hr>
24 <center>
25 <div class="table table-responsive">
26 <table class="table table-striped table-bordered" border="1">
27 <tr>
28 <td width='5%' align="center">Nº</td>
29 <td align="center">Nombre</td>
30 <td align="center">Edad</td>
31 <td align="center">Genero</td>
32 <td align="center">Usuario</td>
33 <td align="center">Tipo</td>
34 <td align="center" colspan="2">Opciones</td>
35 </tr>
36 <?php
```

```
28 <td width='5%' align="center">Nº</td>
29 <td align="center">Nombre</td>
30 <td align="center">Edad</td>
31 <td align="center">Genero</td>
32 <td align="center">Usuario</td>
33 <td align="center">Tipo</td>
34 <td align="center" colspan="2">Opciones</td>
35 </tr>
36 <?php
37
38 foreach($model as $row)
39 {
40     echo "<tr align='center'>";
41     echo "<td>". $row['id']. "</td>";
42     echo "<td align='center' width='10%'>". $row['nombre']. "</td>";
43     echo "<td align='center' width='10%'>". $row['edad']. "</td>";
44     echo "<td align='center' width='10%'>". $row['genero']. "</td>";
45     echo "<td align='center' width='10%'>". $row['usuario']. "</td>";
46     echo "<td align='center' width='10%'>". $row['tipo']. "</td>";
47 }
48 <td align="center" width="10%"><form action="editar_formu.php" method="post">
49     <input type="hidden" name="id" value="<?php echo $row['id']; ?>"><input type="submit" name="editar" value="Editar"></form></td>
50 <td align="center" width="10%"><form action="eliminar.php" method="post"><input type="hidden" name="id" value="<?php echo $row['id']; ?>"><input type="submit" name="eliminar" value="Eliminar"></form></td>
51 <?php
52     }
53 }
54 ?>
55 </table>
56 </div>
57 </center>
58 </body>
59 </html>
```

Muestra los datos del usuario.

usuario.php

```
1 <?php
2
3 class usuario{
4     //definir variables para la clase usuario segun la tabla usuario en la base de datos
5     var $id;
6     var $nombre;
7     var $edad;
8     var $genero;
9     var $usuario;
10    var $contra;
11    var $tipo;
12
13    //funcion que guardar los datos del usuario
14
15    function guardar($nombre, $edad, $genero,$usuario,$contra,$tipo)
16    {
17        try{
18            include_once('../conexion/conexion.php');
19
20            $conn = conexion();
21
22
23            //prepare el sql and bind parameters
24            $stmt = $conn->prepare('INSERT INTO usuario(nombre,edad,genero,usuario,contra,tipo) VALUES(:a,:b,:c,:d,:e,:f)');
25
26
27            $stmt->bindParam(':a',$a);
28            $stmt->bindParam(':b',$b);
29            $stmt->bindParam(':c',$c);
30            $stmt->bindParam(':d',$d);
31            $stmt->bindParam(':e',$e);
32            $stmt->bindParam(':f',$f);
33
34            //insert a row
35            $a = $nombre;
36            $b = $edad;
37            $c = $genero;
38            $d = $usuario;
39            $e = $contra;
40            $f = $tipo;
41
42            $stmt->execute();
43            echo "<script> alert('Registro Almacenado')</script>";
44
45
46
47
48        }catch(PDOException $e){
49            echo "Error:". $e->getMessage();
50
51        }
52    }
53
54    function editar($id,$nombre,$edad,$genero,$usuario,$contra,$tipo)
55    {
56        require_once('../conexion/conexion.php');
57        $conn = conexion();
58        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
59
60
61        $sql = "UPDATE usuario SET nombre = '$nombre', edad = '$edad', genero = '$genero', usuario = '$usuario', contra = '$contra',
62            tipo = '$tipo' WHERE id = '$id'";
63
64        $stmt = $conn->prepare($sql);
65        $stmt->execute();
66
67        header("location:mostrar.php");
68    }
69
70    //funcion que sirve para eliminar los datos de platillo
71    function desactivar($id){
72
73        require_once('../conexion/conexion.php');
74        $conn = conexion();
75
76        try{
77
78            $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
79            $sql = "DELETE FROM usuario WHERE id = '$id'";
80
```

```
80
81     $conn->exec($sql);
82
83
84     }catch(PDOException $e){
85     echo "Error:". $e->getMessage();
86
87     }
88
89     }
90
91 }
92
93
94 ?>
```

El modelo clase del usuario.