



HITO INDIVIDUAL PROGRAMACION

MIGUEL LOZANO GARCIA

INDICE

Fase1_____2-3

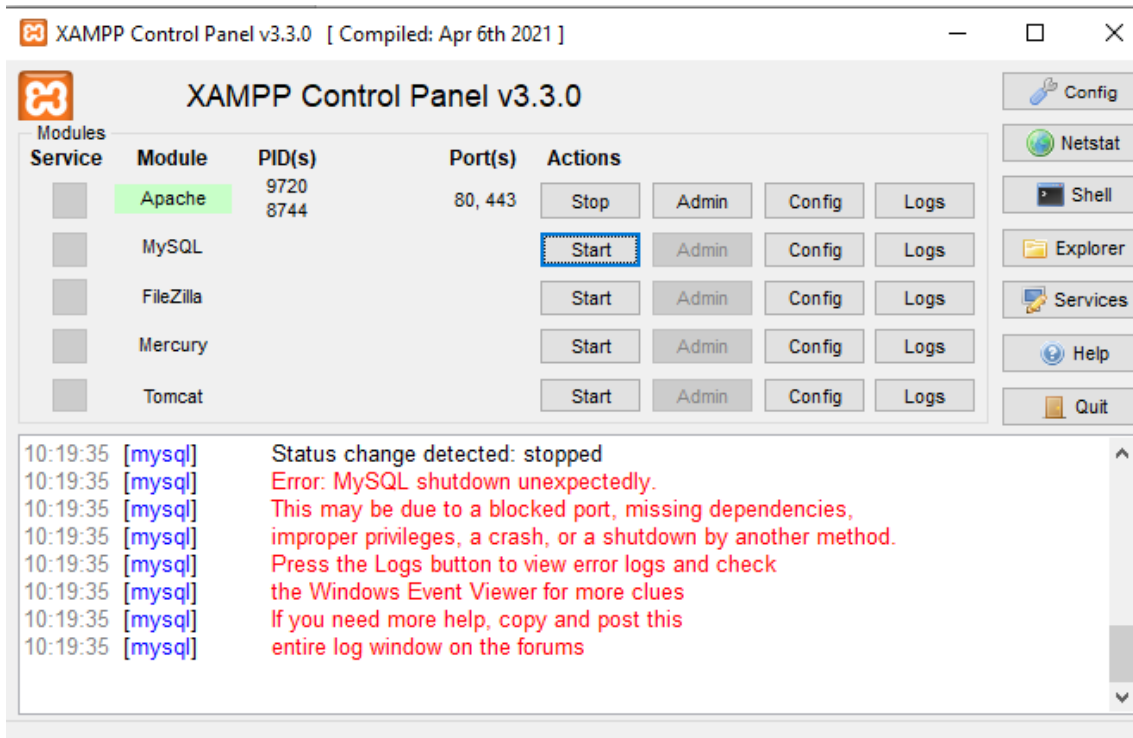
Fase2_____4-12

Fase3_____13

FASE 1

Algoritmia

Para realizar este proyecto voy a utilizar el servidor XXAMP, el cual proporciona un servidor como apache, una base de datos como MySQL y un lenguaje de programación como PHP.



Para empezar, hay que asegurarse de que la configuración del servidor sea adecuada para el entorno local y que se cumplan los requisitos de sistema necesarios para ejecutar el software.

A continuación, usando HTML y CSS se diseñará la página de inicio y proporcionar una descripción clara de los diferentes tipos de lenguajes de programación. Se almacenará la información utilizando las funciones de PHP.

El formulario se realizará con HTML para su diseño y PHP para poder consultar la base de datos de MySQL.

Por último para las sesiones se utilizarán las funciones de PHP de inicio de sesión.

Caso de uso

La página creada consiste en lo siguiente:

Una página principal en la cual el usuario podrá ver las diferencias entre lenguajes de programación orientada a objetos, a eventos y lenguajes procedimentales.

Existirá un apartado en el que el usuario pueda realizar un formulario, otro donde tenga que iniciar sesión y por último un apartado para poder actualizar los clientes por medio de la base de datos ya sea eliminando añadiendo o modificando.

Cómo pasar el algoritmo a la implementación de la aplicación

Lo primero es entender completamente el algoritmo que se va a implementar. Conociendo los pasos necesarios para resolver el problema, los datos de entrada y salida, y cualquier otra consideración importante.

Como la aplicación tiene una interfaz gráfica de usuario, hay que diseñarla antes de comenzar a escribir código. Teniendo en cuenta las necesidades de los usuarios y cómo se van a ingresar los datos de entrada y mostrar los resultados de salida.

Una vez terminado, se puede comenzar a escribir el código. Para ello hay que tener en cuenta el lenguaje de programación elegido previamente.

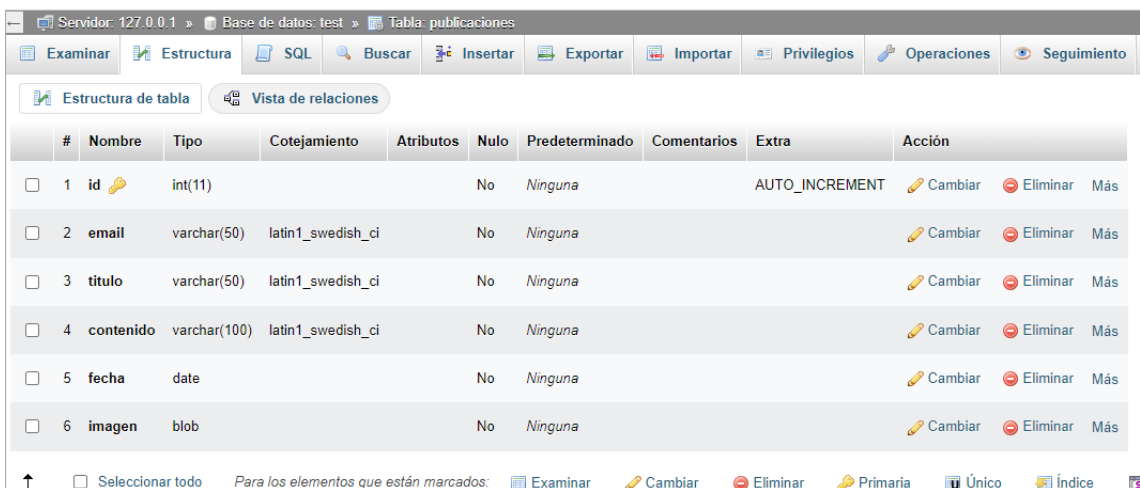
Cuando se haya completado el código nos queda comprobar que todo vaya según lo indicado.

Diferencias

Un algoritmo es una secuencia ordenada de pasos para resolver un problema, mientras que un caso de uso es un escenario que describe cómo un usuario interactúa con un sistema para lograr un objetivo, y el desarrollo de código es el proceso de escribir y depurar el código fuente de un programa de software para implementar el algoritmo y satisfacer los casos de uso.

FASE 2

Primeramente he creado una base de datos desde PHPMyAdmin con los siguientes elementos en la tabla .



The screenshot shows the PHPMyAdmin interface for a database named 'test'. The table 'publicaciones' is selected. The 'Estructura de tabla' (Table Structure) tab is active, displaying the table's schema. The table has 6 columns: 'id' (int(11), primary key, auto-increment), 'email' (varchar(50)), 'titulo' (varchar(50)), 'contenido' (varchar(100)), 'fecha' (date), and 'imagen' (blob). The table is using the 'latin1_swedish_ci' character set. The interface includes navigation tabs at the top (Examinar, Estructura, SQL, etc.) and a bottom toolbar with actions like 'Seleccionar todo', 'Examinar', 'Cambiar', 'Eliminar', etc.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	email	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
3	titulo	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
4	contenido	varchar(100)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
5	fecha	date			No	Ninguna			Cambiar Eliminar Más
6	imagen	blob			No	Ninguna			Cambiar Eliminar Más

Después usando visualStudio Code como entorno de desarrollo he creado un archivo index.htm el cual lleva información sobre la programación y un header que nos llevará a otros apartados.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Hito Programacion</title>
8   <link rel="stylesheet" href="style.css">
9 </head>
10 <body>
11   <div class="box-area">
12     <div class="header">
13       <div class="wrapper">
14         <div class="logo">
15           <a href="index.html">Hito Programacion</a>
16         </div>
17         <nav>
18           <a href="eliminarActualizar.html">Eliminar/Actualizar</a> <a href="publicar.html">Publicar</a> <a href="publicaciones.html">Publicaciones</a>
19         </nav>
20       </div>
21     </div>
22     <div class="banner-area">
23       <h2>PROGRAMACION</h2>
24     </div>
25     <div class="content-area">
26       <div class="wrapper">
27         <h2>Lenguajes De Programación Orientada A Objetos</h2>
28         <p>Son aquellos que están diseñados para permitir a los programadores modelar el mundo real en términos de objetos, los cuales tienen propiedades y comportamientos específicos. La programación orientada a objetos se divide en: Encapsulamiento, Herencia, Polimorfismo y Abstracción.</p>
29         <h2>Lenguajes De Programación Orientada A Eventos</h2>
30         <p>Son aquellos que se enfocan en la programación de eventos y la reacción a los mismos. Un evento es una acción que ocurre en un programa, como la pulsación de un botón, la recepción de datos de un sensor, etc. Estos lenguajes de programación a menudo incluyen bibliotecas y frameworks para facilitar la gestión de eventos, como la suscripción a eventos específicos y la definición de manejadores de eventos.</p>
31         <h2>Lenguajes Procedimentales</h2>
32         <p>Son aquellos que se enfocan en la secuencia de instrucciones o procedimientos que se deben seguir para resolver un problema. En la programación procedimental, los programas se estructuran en una serie de pasos o procedimientos que se ejecutan en orden. En la programación procedimental, los datos son tratados como variables y se manejan mediante la asignación de valores a estas variables. Los procedimientos o funciones se definen y se llaman desde otros puntos del programa. La programación procedimental es útil en situaciones donde el programa necesita seguir un conjunto de instrucciones bien definidas para resolver un problema. Estos lenguajes de programación incluyen: C, C++, Pascal, Fortran, etc.</p>
33       </div>
34     </div>
35   </div>
36 </body>
37 </html>
```

Esto se ha realizado mediante etiquetas nav y div para organizar la página y enlaces que redirigen a otros apartados.

En el apartado de publicar se divide en dos archivos un html en el cual consiste en un título y un formulario donde se encuentran etiquetas de botón y de texto para poder añadir las publicaciones nuevas.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Publicar</title>
8      <link rel="stylesheet" href="style.css">
9  </head>
10 <body>
11     <h1>Publicar en el Blog</h1>
12     <form action="publicar.php" method="post">
13         <label for="email">Email:</label>
14         <input type="email" id="email" name="email" required>
15
16         <br><br>
17
18         <label for="titulo">Título:</label>
19         <input type="text" id="titulo" name="titulo" required>
20
21         <br><br>
22
23         <label for="contenido">Contenido:</label>
24         <textarea id="contenido" name="contenido" rows="10" required></textarea>
25
26         <br><br>
27
28         <label for="fecha">Fecha de publicación:</label>
29         <input type="date" id="fecha" name="fecha" required>
30
31         <br><br>
32
33         <label for="imagen">Imagen:</label>
34         <input type="file" id="imagen" name="imagen">
35
36         <br><br>
37
38         <input type="submit" value="Publicar">
39     </form>
40 </body>
41 </html>
```

Y otro archivo php el cual hace que se conecte con la base de datos creada y de esta manera actualizar las publicaciones y dejarnos regresar al menú con un botón creado con un link que nos devuelve al index.

```

1 <?php
2 $conn = new PDO("mysql:host=localhost;port=3306;dbname=test", "root", "");
3 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
4     $email = $_POST['email'];
5     $titulo = $_POST['titulo'];
6     $contenido = $_POST['contenido'];
7     $fecha = $_POST['fecha'];
8     $imagen = $_POST['imagen'];
9
10    $sql = "INSERT INTO publicaciones (email, titulo, contenido, fecha, imagen) VALUES (:email, :titulo, :contenido, :fecha, :imagen)";
11    $statement = $conn->prepare($sql);
12    $statement->bindValue(':email', $email);
13    $statement->bindValue(':titulo', $titulo);
14    $statement->bindValue(':contenido', $contenido);
15    $statement->bindValue(':fecha', $fecha);
16    $statement->bindValue(':imagen', $imagen);
17    $ejecutar = $statement->execute();
18
19    if (!$ejecutar) {
20        echo "Error";
21    } else {
22        echo "Post Publicado";
23    }
24
25    echo "<a href='index.html'>Volver al menú</a>";
26 }
27 ?>
28

```

Para el apartado de publicaciones es donde podremos comprobar todas las publicaciones que se han subido nuevas o las que han sido modificadas para ello se han creado nuevamente dos archivos uno html con un título y un pequeño formulario formado por un botón que nos redirigirá a la tabla

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Publicaciones</title>
8     <link rel="stylesheet" href="style.css">
9 </head>
10 <body>
11     <h2>Todas las Publicaciones</h2>
12     <form action="publicaciones.php" method="post">
13         <br><br> <input type="submit" value="Ver Publicaciones"><br>
14     </form>
15 </body>
16 </html>

```

y un archivo php que está conectado a la base de datos y de esta manera poder ver las publicaciones mediante un bucle y con un enlace al final que redirija de nuevo al index.

```

1  <?php
2  // conexión - PDO
3  $conn=new PDO("mysql:host=localhost;port=3306;dbname=test","root","");
4
5  // mostrar lo haces con consulta normal y lo muestras en una tabla
6  $consulta = "SELECT * FROM publicaciones;";
7  $resultado = $conn->query($consulta);
8  //var_dump($resultado);
9
10 //tabla con los datos y bucle con fetch
11 echo "<table border='1'>";
12     echo "<tr>";
13         echo "<td>Id</td>";
14         echo "<td>email</td>";
15         echo "<td>titulo</td>";
16         echo "<td>contenido</td>";
17         echo "<td>fecha</td>";
18         echo "<td>imagen</td>";
19     echo "</tr>";
20 while ($row = $resultado->fetch(PDO::FETCH_OBJ)) {
21     echo "<tr>";
22         echo "<td>" . $row->id . "</td>";
23         echo "<td>" . $row->email . "</td>";
24         echo "<td>" . $row->titulo . "</td>";
25         echo "<td>" . $row->contenido . "</td>";
26         echo "<td>" . $row->fecha . "</td>";
27         echo "<td>" . $row->imagen . "</td>";
28     echo "</tr>";
29 }
30 echo "</table>";
31
32 // enlace al menu de inicio
33 echo "<a href='index.html'>Volver al menú</a>";

```

En la sección de eliminar/actualizar he creado una página en la que hay un menú con tres opciones una para eliminar, otra para actualizar y otra para regresar. Esto lo he hecho mediante una etiqueta nav para organizarlo y sus tres links correspondientes.dando con css un estilo para que sea más visual cambiando colores, alineación, fondos y bordes.


```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <link rel="stylesheet" href="styleEA.css">
9  </head>
10 <body>
11     <div class="container">
12         <h1>¿Qué deseas hacer?</h1>
13         <a href="eliminar.html">Eliminar Publicacion</a>
14         <a href="actualizar.html">Actualizar Publicacion</a>
15         <a href="index.html">Volver al Menu</a>
16     </div>
17 </body>
18 </html>
19
20

```

```

1  .container {
2      display: flex;
3      flex-direction: column;
4      align-items: center;
5      justify-content: center;
6      height: 100vh;
7  }
8  a {
9      display: inline-block;
10     padding: 12px 24px;
11     margin: 10px;
12     font-size: 20px;
13     text-decoration: none;
14     color: #ffffff;
15     background-color: #4CAF50;
16     border-radius: 4px;
17     box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
18     transition: background-color 0.3s ease;
19 }
20 a:hover {
21     background-color: #3e8e41;
22 }
23 body{
24     background-image: url(imagenes/programacion-de-diseno-web.jpg);
25 }
26 h1{
27     color: white;
28 }

```

El link de actualizar lleva a un archivo html el cual mediante un formulario nuevamente nos deja seleccionar el título de la publicación a actualizar y la fecha que deseamos cambiar.

```

actualizar.html > html > body > form > input
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Actualizar</title>
8      <link rel="stylesheet" href="stylef.css">
9  </head>
10 <body>
11     <h1>Actualizar Publicacion</h1>
12     <form action="actualizar.php" method="post">
13         <input type="text" name="titulo" placeholder="actualizar publicacion"><br>
14         <input type="date" name="fecha" placeholder="fecha"><br>
15         <input type="submit" value="Nuevo producto"><br>
16     </form>
17 </form>
18 </body>
19 </html>

```

En la parte del php lo he enlazado con la base de datos y he creado no he logrado el código para que se puedan cambiar los datos antes mencionados creando variables para título y fecha pero debido a que no he podido crear el procedimiento almacenado de la base de datos me da error.

```

1  <?php
2  //conexión - PDO
3  $conn = new PDO("mysql:host=localhost;port=3306;dbname=test", "root", "");
4  //insertar : procedimiento almacenado
5  //prepare
6
7  $sql = 'CALL publicaciones(?, ?)';
8  $stmt = $conn->prepare($sql);
9
10 $titulo = $_POST['titulo'];
11 $fecha = $_POST['fecha'];
12 //bindParam
13 $stmt->bindParam(1, $titulo);
14 $stmt->bindParam(2, $fecha);
15 //execute
16 $stmt->execute();
17 if (!$ejecutar) {
18     echo "Error";
19 } else {
20     echo "Post Publicado";
21 }
22 }
23 echo "<a href='index.html'>Volver al menú</a>";
24

```

Para la sección eliminar he creado un título y en este caso solo es necesario tener en el formulario el título para que de esta manera se elimine directamente.

```
<? eliminar.html > <? html>
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Eliminar</title>
8  </head>
9  <body>
10     <h2>Eliminar Publicacion</h2>
11     <form action="eliminar.php" method="post">
12         <input type="text" name="titulo" placeholder="Titulo de la Publicacion"><br>
13         <input type="submit" value="Eliminar"><br>
14     </form>
15 </body>
16 </html>
```

Por último en eliminar me ha pasado lo mismo que con el anterior método ya que en ambos necesitamos el procedimiento almacenado.

Para el css he creado tres distintos uno que explique anteriormente otro que es para los formularios en el cual doy estilo a las etiquetas de los formularios ya sea introduciéndolas en bloque o dando márgenes.

```
1  label {
2      display: block;
3      margin-bottom: 10px;
4      font-weight: bold;
5  }
6  input[type="submit"] {
7      margin-top: 20px;
8  }
```

Y el css de la página principal, es decir el index, en el cual he creado varias clases para poder organizarlo y dar formatos de manera más específica con los típicos comandos de cambiar el color, poner fondo, cambiar el tipo de letra, aplicar un float para mover de lado el bloque, etc.

```

1  √ * {
2      margin: 0;
3      padding: 0;
4  }
5  √ body {
6      text-align: center;
7  }
8  √ .wrapper {
9      width: 1170px;
10     margin: 0 auto;
11 }
12 √ header {
13     height: 100px;
14     background: #262626;
15     width: 100%;
16     z-index: 10;
17     position: fixed;
18 }
19 √ .logo {
20     width: 30%;
21     float: left;
22     line-height: 100px;
23 }
24 √ .logo a {
25     text-decoration: none;
26     font-size: 30px;
27     font-family: poppins;
28     color: #fff;
29     letter-spacing: 5px;
30 }
31 √ nav {
32     float: right;
33     line-height: 100px;
34 }
35 √ nav a {
36     text-decoration: none;
37     font-family: poppins;
38     letter-spacing: 4px;
39     font-size: 20px;
40     margin: 0 10px;
41     color: #fff;
42 }

```

```

43 .banner-area {
44     width: 100%;
45     height: 500px;
46     position: fixed;
47     top: 100px;
48     background-image: url(imagenes/programacion-de-diseno-web.jpg);
49     -webkit-background-size: cover;
50     background-size: cover;
51     background-position: center center;
52 }
53 .banner-area h2 {
54     padding-top: 8%;
55     font-size: 70px;
56     font-family: poppins;
57     text-transform: uppercase;
58     color: #fff;
59 }
60 .content-area {
61     width: 100%;
62     position: relative;
63     top: 450px;
64     background: #ebebeb;
65     height: 1500px;
66 }
67 .content-area h2 {
68     font-family: poppins;
69     letter-spacing: 4px;
70     padding-top: 30px;
71     font-size: 40px;
72     margin: 0;
73 }
74 .content-area p {
75     padding: 2% 0px;
76     font-family: poppins;
77     line-height: 30px;
78 }
79 h2{
80     font-family: 'Courier New', Courier, monospace;
81 }
82 p{
83     font-size: 20px;
84 }
85 #publicaciones{
86     font-size: 50px;
87     color: white;
88 }

```

FASE 3

Tecnologías, servicios, herramientas y software

Para el diseño de esta página he utilizado apache Netbeans el cual ha sido iniciado desde el programa xampp al igual que mysql el cual he tenido que deshabilitar previamente para poder iniciarlo desde aquí.

Para comprobar el progreso lo he ido mirando desde localhost desde internet y en el momento de crear la base de datos también lo realice desde una opción de localhost llamada phpMyadmin la cual luego se conectará a los archivos php creados para unir la página web html con la base de datos y algun css todo esto realizado en VisualStudioCode como entorno de desarrollo.

Problemas, dificultades y errores

En este proyecto me he topado con diferentes errores algunos los cuales he podido resolver como es el caso de no colocar las cosas como quería mediante el css o a la hora de la conexión con la base de datos ya que me faltaba el símbolo igual y algún error que no he conseguido solucionar como la creación del procedimiento almacenado por lo que no he podido completar el proceso de actualizar y eliminar.

Evaluación

La página web cuenta con una estructura intuitiva y fácil de entender de manera que el usuario no tendrá ningún tipo de pérdida al navegar por ella.

Los usuarios encontrarán lo requerido y lo pedido en esta para que puedan disfrutar de una página donde publicar lo que deseen.

En la página se podrán actualizar las publicaciones en caso de ser necesario por lo que siempre se estará al día de lo nuevo.