



---

# MEJORA HITO INDIVIDUAL PROGRAMACIÓN

---

Daniel Manzano Nuñez



# ÍNDICE

## FASE 1:

1. ¿QUÉ ES UN ALGORITMO?
2. CASOS DE USO
3. EXPLICAR COMO PASAR EL ALGORITMO A LA PRÁCTICA

## FASE 2:

1. IMPLEMENTACIÓN DEL CÓDIGO

## FASE 3:

1. JUSTIFICA LAS TECNOLOGÍAS, HERRAMIENTAS Y SOFTWARE ELEGIDOS PARA EL DESARROLLO DE LA WEB
2. ENUMERA LOS PROBLEMAS QUE HAS ENCONTRADO Y CÓMO LOS HAS RESUELTO
3. EVALÚA Y RECOMIENDA COSAS PARA SU MEJORA

# FASE 1

## 1. ¿QUÉ ES UN ALGORITMO?

Un algoritmo es una secuencia de pasos o instrucciones ordenadas y finitas para resolver un problema. En cuanto a la tarea propuesta se definen unos pasos concretos para la resolución del problema.

- El primer paso es leer y comprender lo que pide el enunciado. Antes de comenzar una web o cualquier aplicación tienes que hacerte una idea de que vas a utilizar, como lo vas a realizar y cómo va a quedar al final.
- Hay que pensar en los datos de entrada y salida en la aplicación. El cliente se va a registrar en la página web y va a introducir sus datos en la página para que esta le muestre otros datos de salida.

En este caso los algoritmos son los requisitos que exige la página web para su correcta puesta en marcha.

## 2. CASOS DE USO

Un caso de uso, son acciones que resultan o dan lugar a un resultado. En este caso el primer caso de uso sería que la página web se lance en un servidor Apache. Después, la página deberá tener un index en el cual nos encontramos los requisitos teóricos que piden.

Luego otro de los factores a tener en cuenta es el formulario en el cual el cliente escriba el blog correspondiente y que dicho blog se guarde en una base de datos anteriormente creada.

Llegamos al momento en el que se muestran las entradas a la página web en una tabla en la cual vamos a actualizar y eliminar los datos de dicha tabla.

Al finalizar, si todos esos requisitos se dan estaremos ante un caso de uso que funciona a la perfección después de haber pasado todos los requisitos.

## 3.PASAR DEL ALGORITMO A LA PRÁCTICA

Una vez tenemos definidos los algoritmos pasamos a la práctica. Si hemos analizado los algoritmos y los casos de uso tendremos una mejor idea sobre lo que hacer en el hito.

Tenemos que establecer el paso a paso e ir viendo qué problemas van surgiendo durante el transcurso del proyecto. En el caso del hito, es un ejercicio largo que puede desembocar en muchos problemas.

Si hemos hecho un buen algoritmo y durante los casos de usos hemos previsto problemas que nos puedan surgir tendremos una mejor respuesta ante dichos problemas.

## FASE 2

### 1.IMPLEMENTACIÓN DEL CÓDIGO

Para comenzar con la página web siempre hay que empezar por el index.

```
<> index.html > html > header > div.wrapper > nav > a
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 Individual Programación</title>
8      <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, mini
9      <script src="http://code.jquery.com/jquery-latest.js"></script>
10     <link rel="stylesheet" type="text/css" href="estilos.css" media="screen" />
11 </head>
12 <header>
13     <div class="wrapper">
14         <nav>
15             <a href="formularioblog.html">Formulario Blog</a>
16             <a href="mostrarentradas.php">Mostrar las entradas a la página</a>
17             <a href="actualizarcliete.php">Actualizar Clientes</a>
18             <a href="eliminarcliente.php">Eliminar Clientes</a>
19         </nav>
20     </div>
21 </header>
```

En esta imagen nos encontramos con un comienzo HTML 5 y a continuación con un header que nos muestra las páginas utilizadas para el desarrollo del proyecto.

```

22 <body>
23   <h1>Diferencias entre lenguajes de programación</h1>
24   <h2>Lenguajes de programación procedimentales</h2>
25   <p>Los lenguajes de programación procedimentales se basan en la idea de una secuencia de instrucciones que se ejecutan en un orden específico
26
27   <ul>
28     <li>La programación procedimental se centra en la lógica y los procesos, y no tanto en los datos.</li>
29     <li>Los programas procedimentales suelen ser más fáciles de entender y mantener que los programas orientados a objetos o a eventos.</li>
30     <li>Algunos ejemplos de lenguajes de programación procedimentales son C, Pascal y Fortran.</li>
31   </ul>
32
33   <h2>Lenguajes de programación orientada a objetos</h2>
34   <p>Los lenguajes de programación orientada a objetos se basan en el concepto de objetos, que pueden tener atributos y métodos. Los objetos se
35
36   <ul>
37     <li>La programación orientada a objetos se centra en los datos y la interacción entre objetos.</li>
38     <li>Los programas orientados a objetos son más fáciles de extender y reutilizar que los programas procedimentales.</li>
39     <li>Algunos ejemplos de lenguajes de programación orientada a objetos son Java, C++, Python y Ruby.</li>
40   </ul>
41
42   <h2>Lenguajes de programación orientada a eventos</h2>
43   <p>Los lenguajes de programación orientada a eventos se basan en la idea de que un programa responde a eventos que ocurren en el sistema, como
44
45   <ul>
46     <li>La programación orientada a eventos se centra en la gestión de eventos y la respuesta a ellos.</li>
47     <li>Los programas orientados a eventos son útiles para aplicaciones de interfaz de usuario y sistemas que interactúan con el mundo exterior
48     <li>Algunos ejemplos de lenguajes de programación orientada a eventos son JavaScript y Visual Basic.</li>
49   </ul>
50
51   <p>En resumen, los lenguajes de programación procedimentales se centran en la secuencia de instrucciones, los lenguajes de programación orient
52
53 </body>
54 </html>

```

A continuación dentro del index nos encontramos ante el primer requisito el cual nos pedía una parte teórica mostrada en el index.html. Podemos observar que los títulos están puestos en una etiqueta h2 típica de este tipo de documentos. A continuación vemos etiquetas p las cuales nos muestran la explicación en cadenas de texto. Finalmente para las listas de ejemplos nos encontramos con etiquetas ul y li también muy comunes en este tipo de documentos.

Una vez puesto esto nos quedaría de este estilo:

[Formulario Blog](#)
[Mostrar las entradas a la página](#)
[Actualizar Clientes](#)
[Eliminar Clientes](#)

## Diferencias entre lenguajes de programación

### Lenguajes de programación procedimentales

Los lenguajes de programación procedimentales se basan en la idea de una secuencia de instrucciones que se ejecutan en un orden específico para lograr un objetivo. En estos lenguajes, las funciones son llamadas en un orden particular para llevar a cabo una tarea.

La programación procedimental se centra en la lógica y los procesos, y no tanto en los datos. Los programas procedimentales suelen ser más fáciles de entender y mantener que los programas orientados a objetos o a eventos. Algunos ejemplos de lenguajes de programación procedimentales son C, Pascal y Fortran.

### Lenguajes de programación orientada a objetos

Los lenguajes de programación orientada a objetos se basan en el concepto de objetos, que pueden tener atributos y métodos. Los objetos se utilizan para representar entidades en el mundo real, y se pueden combinar para crear sistemas más complejos.

La programación orientada a objetos se centra en los datos y la interacción entre objetos. Los programas orientados a objetos son más fáciles de extender y reutilizar que los programas procedimentales. Algunos ejemplos de lenguajes de programación orientada a objetos son Java, C++, Python y Ruby.

### Lenguajes de programación orientada a eventos

Los lenguajes de programación orientada a eventos se basan en la idea de que un programa responde a eventos que ocurren en el sistema, como clics de mouse, pulsaciones de teclas o mensajes recibidos de otros programas.

La programación orientada a eventos se centra en la gestión de eventos y la respuesta a ellos. Los programas orientados a eventos son útiles para aplicaciones de interfaz de usuario y sistemas que interactúan con el mundo exterior. Algunos ejemplos de lenguajes de programación orientada a eventos son JavaScript y Visual Basic.

En resumen, los lenguajes de programación procedimentales se centran en la secuencia de instrucciones, los lenguajes de programación orientada a objetos se centran en los objetos y su interacción.

Prosiguiendo con el trabajo nos encontramos con un documento HTML que nos muestra el formulario que nos piden de requisito. Utiliza la etiqueta form para crear el formulario seguido del archivo donde queremos mandar los datos y con la etiqueta post. Quedaría de la siguiente manera:

---

## Escribir un post del blog

Email del autor:

Título:

Contenido:

Fecha de publicacion:  

Imagen:  Ninguno archivo selec.

Este formulario manda los datos a un archivo PHP el cual mediante una conexión manda los datos a una base de datos.

Prosiguiendo con el Ítem 4 nos encontramos con un PHP que nos muestra en una tabla los datos de las personas que han entrado en la página web.

```
mostrarentradas.php
1  <?php
2  // Conexión a la base de datos
3  $servername = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "clientes";
7
8  $conn = new mysqli($servername, $username, $password, $dbname);
9
10 // Verificar la conexión
11 if ($conn->connect_error) {
12     die("Connection failed: " . $conn->connect_error);
13 }
14
15 // Selección de todas las entradas
16 $sql = "SELECT * FROM `login`";
17 $result = $conn->query($sql);
18
```

```

13 }
14
15 // Selección de todas las entradas
16 $sql = "SELECT * FROM `login`";
17 $result = $conn->query($sql);
18
19 // Mostrar las entradas
20 if ($result->num_rows > 0) {
21     echo "<table border=1>";
22     echo "<tr><th>id</th><th>nombre</th><th>email</th></tr>";
23     while($row = $result->fetch_assoc()) {
24         echo "<tr>";
25         echo "<td>" . $row["id"] . "</td>";
26         echo "<td>" . $row["nombre"] . "</td>";
27         echo "<td>" . $row["email"] . "</td>";
28         echo "</tr>";
29     }
30     echo "</table>";
31 } else {
32     echo "No se encontraron entradas";
33 }
34
35 // Cerrar la conexión
36 $conn->close();
37 ?>
38

```

Por último el ítem 5 está muy relacionado con el 4 puesto que es la misma tabla solo que esta vez hay que eliminar un cliente del registro.

---

id	nombre	email
1	Luis	luis@gmail.com
3	Lucas	lucas@gmail.com

En este caso, se ha eliminado el cliente con el id número 2.

## FASE 3

### **1. Justifica las tecnologías, servicios, herramientas y software elegidos para la realización del sitio web.**

En primer lugar, he utilizado Xampp con el servidor apache en el puerto 80,443. En cuanto a el código he utilizado Visual Studio Code el cual es un editor de código encargado de ejecutar el código. En este caso el código programado lo he lanzado a localhost mediante el servidor apache anterior.

En segundo lugar, para la conexión a la base de datos he usado MySQL de Xampp y además el phpmyadmin. Esto funciona haciendo una conexión en el código de visual studio code y permite que en formularios en la web se manden a la base de datos de mySQL.

### **2. Enumera los problemas y dificultades que se ha encontrado en la actividad y cómo los ha resuelto.**

Durante el transcurso del hito nos hemos encontrado con diversos problemas y dificultades en el código.

Uno de los problemas o dificultades que nos hemos encontrado es la conexión a la base de datos. La conexión a una base de datos es complicada debido a que debes de poner bien el usuario, servidor y contraseña dependiendo de donde utilices la base de datos. Finalmente mirando los ejercicios de clase he conseguido hacer la conexión a la base de datos.

Otro de los problemas que me he encontrado es que en el formulario para publicar un blog nos muestra todos los datos excepto la imagen que añades en el formulario. Este problema es uno de los más complicados a los que me he enfrentado y he intentado buscar una solución a este problema pero no he podido solucionarlo.

En cuanto a actualizar clientes es el problema que más dificultad he tenido puesto que me ha costado esta parte del hito. He mirado e investigado soluciones para este problema pero no he podido solucionar del todo este problema, en cambio eliminar clientes no me ha planteado tantos problemas.

### **3. Evalúa los resultados del Test Plan y el éxito completo del sitio web con recomendaciones para su mejora.**

Llegados a este punto toca analizar el trabajo realizado y recomendar posibles mejoras para mejorar el próximo.



En primer lugar, encontramos el index el cual está completo con todo lo pedido pero podría mejorarse. Para mejorar este index podría haber puesto un header de alguna página de manera que quede más bonito y visible.

[Formulario Blog](#)[Mostrar las entradas a la página](#)[Actualizar Clientes](#)[Eliminar Clientes](#)

## Diferencias entre lenguajes de programación

### Lenguajes de programación procedimentales

Los lenguajes de programación procedimentales se basan en la idea de una secuencia de instrucciones que se ejecutan en un orden específico para lograr un objetivo. En estos lenguajes, las funciones son llamadas en un orden particular para llevar a cabo una tarea.

La programación procedimental se centra en la lógica y los procesos, y no tanto en los datos. Los programas procedimentales suelen ser más fáciles de entender y mantener que los programas orientados a objetos o a eventos. Algunos ejemplos de lenguajes de programación procedimentales son C, Pascal y Fortran.

### Lenguajes de programación orientada a objetos

Los lenguajes de programación orientada a objetos se basan en el concepto de objetos, que pueden tener atributos y métodos. Los objetos se utilizan para representar entidades en el mundo real, y se pueden combinar para crear sistemas más complejos.

La programación orientada a objetos se centra en los datos y la interacción entre objetos. Los programas orientados a objetos son más fáciles de extender y reutilizar que los programas procedimentales. Algunos ejemplos de lenguajes de programación orientada a objetos son Java, C++, Python y Ruby.

### Lenguajes de programación orientada a eventos

Los lenguajes de programación orientada a eventos se basan en la idea de que un programa responde a eventos que ocurren en el sistema, como clics de mouse, pulsaciones de teclas o mensajes recibidos de otros programas.

La programación orientada a eventos se centra en la gestión de eventos y la respuesta a ellos. Los programas orientados a eventos son útiles para aplicaciones de interfaz de usuario y sistemas que interactúan con el mundo exterior. Algunos ejemplos de lenguajes de programación orientada a eventos son JavaScript y Visual Basic.

En resumen, los lenguajes de programación procedimentales se centran en la secuencia de instrucciones, los lenguajes de programación orientada a objetos se centran en los objetos y su interacción.

En segundo lugar, nos encontramos ante un formulario para publicar un blog. Dicho formulario está bien organizado y las mejoras podrían ir en el sentido de el archivo php el cual podría haber hecho que los datos que se muestran del formulario se mostrarán en una tabla en vez de como están.

### Escribir un post del blog

Email del autor:

Título:

Contenido:

Fecha de publicación:

Imagen:  Ninguno archivo selec.

los datos del formulario son los siguientes

luis.garcia@gmail.com

Imagen fondo pagina web

imagen fondo

2023-04-10

Luego en cuanto a mostrar las entradas a la página está bastante bien ya que hacemos la conexión a la base de datos y esas entradas las mostramos en una tabla.

---

<b>id</b>	<b>nombre</b>	<b>email</b>
1	Luis	luis@gmail.com
3	Lucas	lucas@gmail.com
4	Dani	dani@gmail.com