HITO INDIVIDUAL PROGRAMACIÓN

Daniel Manzano Nuñez

28 DE FEBRERO DE 2023
CAMPUS FP
1ºDAM

ÍNDICE

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

En este proyecto he llevado a cabo una página web con lenguajes como HTML, CSS y PHP vinculado con bases de datos. El entorno de desarrollo elegido para el proyecto ha sido Visual Studio Code, las bases de datos en PHPmyAdmin y localhost con apache para lanzar la página.

FASE 3

1.JUSTIFICA EL TRABAJO

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

index.html › html › header › div.wrapper › nav › a

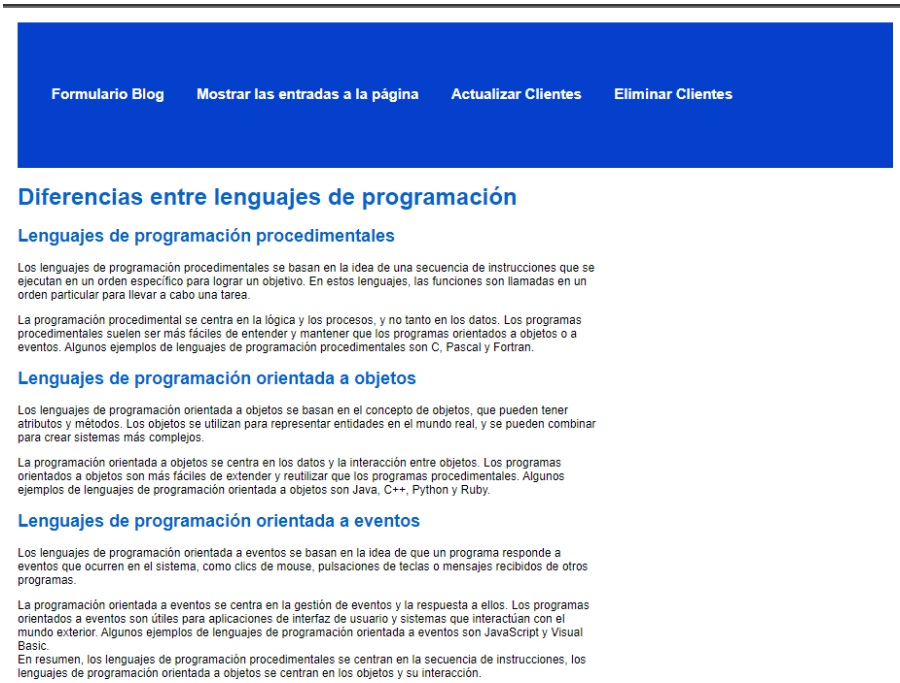
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hito Individual Programación</title>
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, mini
    <script src="http://code.jquery.com/jquery-latest.js"></script>
    <link rel="stylesheet" type="text/css" href="estilos.css" media="screen" />
</head>
<header>
    <div class="wrapper">
        <nav>
            <a href="formularioblog.html">Formulario Blog</a>
            <a href="mostrarentradas.php">Mostrar las entradas a la página</a>
            <a href="actualizarcliete.php">Actualizar Clientes</a>
            <a href="eliminarcliente.php">Eliminar Clientes</a>
        </nav>
    </div>
</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.

```html
<body>
    <h1>Diferencias entre lenguajes de programación</h1>
    <h2>Lenguajes de programación procedimentales</h2>
    <p>Los lenguajes de programación procedimentales se basan en la idea de una secuencia de instrucciones que se ejecutan en un orden específico

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

    <h2>Lenguajes de programación orientada a objetos</h2>
    <p>Los lenguajes de programación orientada a objetos se basan en el concepto de objetos, que pueden tener atributos y métodos. Los objetos se

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

    <h2>Lenguajes de programación orientada a eventos</h2>
    <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

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

    <p>En resumen, los lenguajes de programación procedimentales se centran en la secuencia de instrucciones, los lenguajes de programación orient

</body>
</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:



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:

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

```php
<?php
// Conexión a la base de datos
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "clientes";

$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar la conexión
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Selección de todas las entradas
$sql = "SELECT * FROM `login`";
$result = $conn->query($sql);

```

```php
}

// Selección de todas las entradas
$sql = "SELECT * FROM `login`";
$result = $conn->query($sql);

// Mostrar las entradas
if ($result->num_rows > 0) {
    echo "<table border=1>";
    echo "<tr><th>id</th><th>nombre</th><th>email</th></tr>";
    while($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["id"] . "</td>";
        echo "<td>" . $row["nombre"] . "</td>";
        echo "<td>" . $row["email"] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "No se encontraron entradas";
}

// Cerrar la conexión
$conn->close();
?>
```

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.