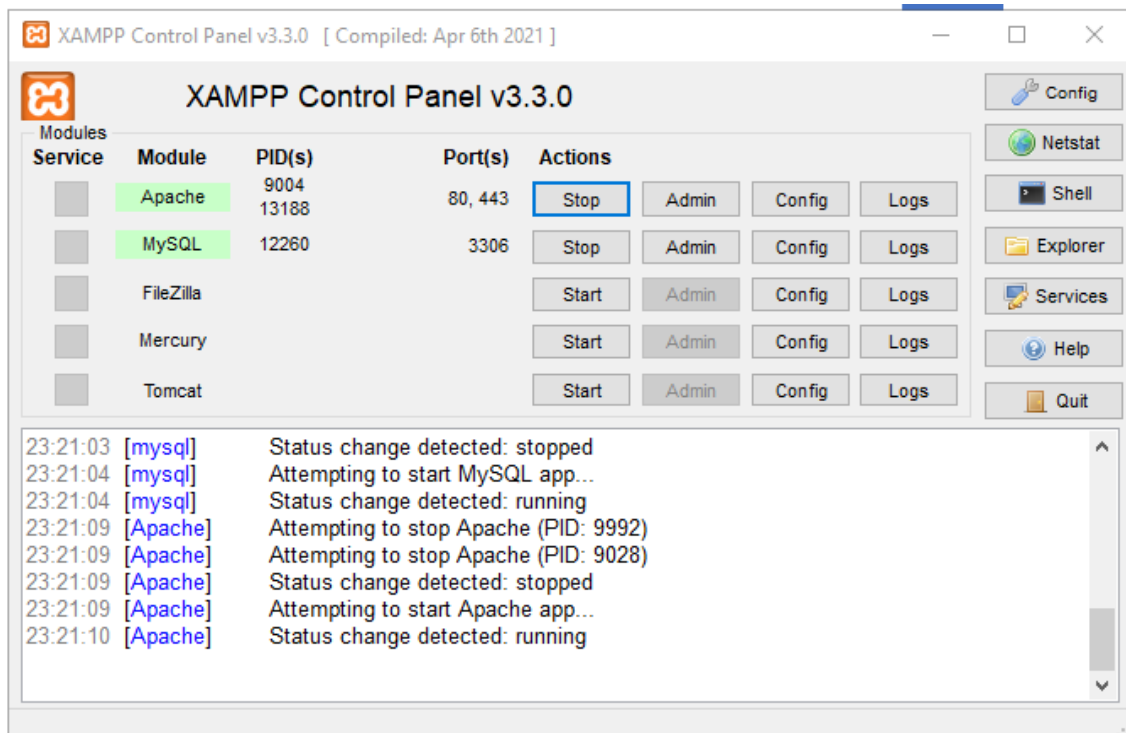


Hito Individual Programación

LARA LOPEZ-POZUELO CEBADERA

Desplegada en un entorno de pruebas consistente en un servidor Apache local.



XAMPP es un paquete de software libre que incluye un conjunto de herramientas que permiten crear y gestionar un servidor web en una computadora personal. La sigla XAMPP proviene de las primeras letras de los programas que incluye: Apache, MySQL, PHP y Perl.

XAMPP es utilizado principalmente por desarrolladores web que necesitan configurar y probar aplicaciones web en un ambiente local antes de subirlas a un servidor en línea. Al instalar XAMPP, se pueden crear sitios web y aplicaciones en el computador personal sin necesidad de tener una conexión a Internet.

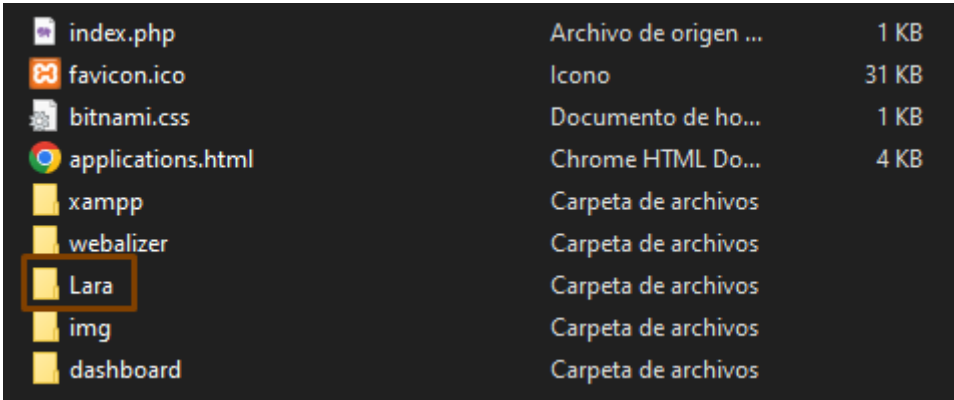
Además de los programas mencionados, XAMPP incluye otros componentes como phpMyAdmin, un gestor de bases de datos MySQL; FileZilla FTP Server, un servidor FTP; Mercury Mail Server, un servidor de correo electrónico; y Tomcat, un servidor de aplicaciones Java.

XAMPP es una herramienta útil para cualquier persona que necesite trabajar con servidores web en su computadora personal, ya sea para el desarrollo de sitios web, aplicaciones o para la realización de pruebas y experimentos. Su facilidad de instalación y uso lo convierte en una opción popular entre los desarrolladores web.

Como Guardar un archivo y verlo usando XAMPP

1. Abre XAMPP y asegúrate de que Apache y MySQL estén activados.
2. Abre tu editor de texto preferido (como Notepad o Sublime Text) y crea un archivo HTML o PHP.
3. Guarda el archivo con el nombre y la extensión correspondientes (por ejemplo, "index.php" o "about.html") en la carpeta "htdocs" dentro de la carpeta de instalación de XAMPP. Esta carpeta es la que se utiliza para alojar los archivos que se mostrarán en el navegador web.
4. Abre un navegador web y escribe "localhost/nombre-del-archivo" en la barra de direcciones. Asegúrate de que el nombre del archivo coincide exactamente con el que guardaste en la carpeta "htdocs".
5. Si todo está configurado correctamente, deberías ver el contenido del archivo en el navegador web.

Recuerda que para que el archivo se muestre correctamente en el navegador, debe tener el formato correcto (HTML, PHP, etc.) y estar diseñado y escrito correctamente. Además, si estás utilizando archivos PHP, es posible que debas configurar la base de datos y la conexión a la base de datos para que los scripts se ejecuten correctamente.



index.php	Archivo de origen ...	1 KB
favicon.ico	Icono	31 KB
bitnami.css	Documento de ho...	1 KB
applications.html	Chrome HTML Do...	4 KB
xampp	Carpeta de archivos	
webalizer	Carpeta de archivos	
Lara	Carpeta de archivos	
img	Carpeta de archivos	
dashboard	Carpeta de archivos	

En la página de arranque / inicio, el cliente nos pide que hagamos una explicación de las diferencias entre lenguajes de programación orientada a objetos, a eventos y lenguajes procedimentales. Esta explicación será texto con marcado HTML y diseño CSS.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Lenguajes de programación</title>
    <link rel="stylesheet" href="index.css">
  </head>
  <body>
    <header>
      <h1>Lenguajes de programación</h1>
      <ul class="menu">
        <li><a href="index.html">Home</a></li>
        <li><a href="post.html">Formulario</a></li>
        <li><a href="login.html">Login</a></li>
        <li><a href="comentarios.php">Comentarios</a></li>
        <li><a href="borrar_delete.html">Modificar</a></li>
      </ul>
    </header>
    <section id="oop">
      <h2>Programación orientada a objetos</h2>
      <p>La programación orientada a objetos (POO) se centra en los objetos y las interacciones entre ellos. En POO, un objeto es una instancia de una clase que tiene propiedades y métodos que pueden ser heredados y compartidos con otros objetos. Los lenguajes de programación orientada a objetos más populares son Java, C++, Python y Ruby.</p>
    </section>
    <section id="event">
      <h2>Programación por eventos</h2>
      <p>La programación por eventos se enfoca en la interacción de los usuarios con una aplicación. Los eventos son acciones que realiza un usuario, como hacer clic en un botón o escribir texto en un campo de entrada. Los lenguajes de programación por eventos más populares son JavaScript y PHP.</p>
    </section>
    <section id="proc">
      <h2>Programación procedimental</h2>
      <p>La programación procedimental se basa en la ejecución de una serie de instrucciones secuenciales. Las instrucciones pueden ser agrupadas en funciones o procedimientos, que pueden ser reutilizados en otras partes del programa. Los lenguajes de programación procedimentales más populares son C, Pascal y Fortran.</p>
    </section>
  </body>
</html>
```

Este es un archivo HTML que contiene una página web que muestra información sobre tres tipos de programación: programación orientada a objetos, programación por eventos y programación procedimental. La página incluye un encabezado con un menú de navegación que permite al usuario moverse entre diferentes páginas web relacionadas, así como tres secciones que describen cada tipo de programación. Cada sección incluye un título y un párrafo que describe brevemente cada tipo de programación. El archivo también enlaza a un archivo CSS externo para aplicar estilos visuales a la página web.

Con su respectivo CSS para darle una apariencia mas llamativa y intuitiva para el usuario

```

/* Estilos generales */
body {
  font-family: 'Open Sans', sans-serif;
  margin: 0;
  background-color: #f9f9f9;
  color: #333;
}

header {
  background-color: #4CAF50;
  color: #fff;
  padding: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

h1 {
  margin: 0;
  font-size: 2.5rem;
}

section {
  padding: 50px;
  text-align: justify;
}

p {
  line-height: 1.5;
}

/* Estilos del menú */
.menu {
  list-style: none;
  margin: 0;
  padding: 0;
  background-color: #333;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.menu li {
  margin: 0 10px;
}

.menu li a {
  display: block;
  color: #fff;
  text-align: center;
  padding: 10px 20px;
  text-decoration: none;
  transition: background-color 0.3s ease;
}

.menu li a:hover {
  background-color: #4CAF50;
}

.menu .active a {
  background-color: #fff;
  color: #333;
  font-weight: bold;
}

/* Estilos específicos de cada sección */
#oop {
  background-color: #ffc107;
}

#event {
  background-color: #00bcd4;
}

```

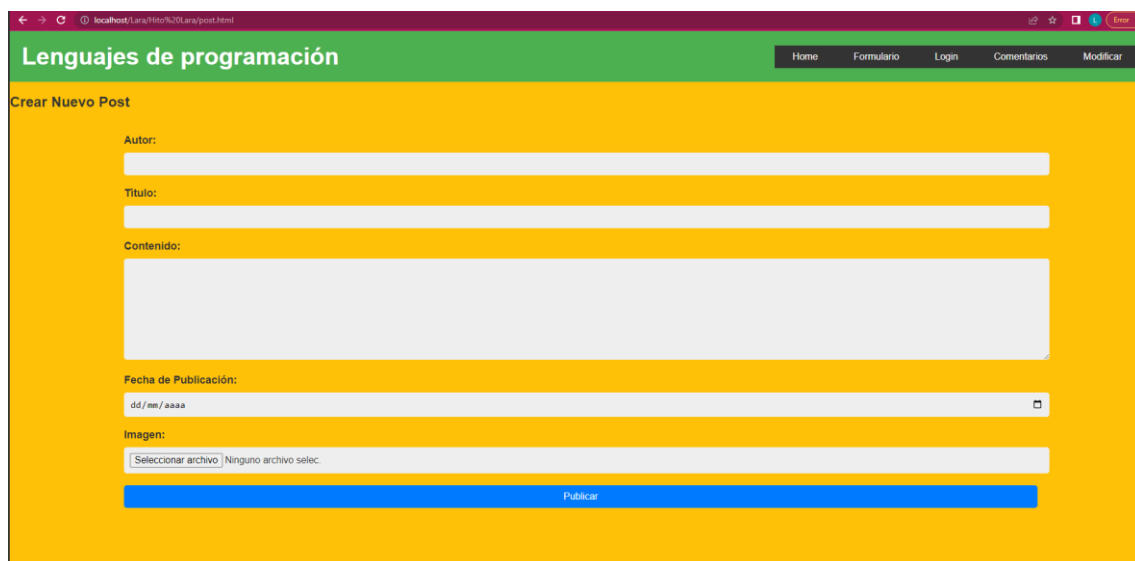
Se utiliza la fuente 'Open Sans' para el texto de la página y se establece un color de fondo gris claro para el cuerpo del documento. El encabezado se establece con un color de fondo verde (#4CAF50), letra blanca y un espacio de relleno de 20 píxeles. Se utiliza un display flex para alinear el contenido del encabezado, y se establece la fuente a un tamaño de 2.5 rem.

Las secciones tienen un espacio de relleno de 50 píxeles y se alinean justificadamente. El color de fondo para cada sección se establece en amarillo (#ffc107) para la sección de programación orientada a objetos, azul claro (#00bcd4) para la programación por eventos y morado (#673ab7) para la programación procedimental. Los títulos de cada sección se establecen en mayúsculas, se les aplica un espaciado de letra de 2 píxeles y un tamaño de fuente de 2 rem.

El menú se establece con una lista sin viñetas y se ajusta la margen y el relleno a cero. El color de fondo se establece en negro (#333) y se utiliza un display flex para alinear el contenido del menú. Los elementos de la lista tienen un margen de 10 píxeles y se establece un tamaño de letra de 10 píxeles. Al pasar el cursor sobre los enlaces del menú, se aplica un cambio de color de fondo (#4CAF50). La opción del menú activa se establece con un color de fondo blanco (#fff), letra negra (#333) y un peso de fuente negrita.

En general, estos estilos hacen que la página se vea organizada, atractiva y fácil de leer.

Este ítem consiste en diseñar una página que permita al usuario escribir un post del blog de cliente. Se realizará un formulario que nos pida, email del autor, título, contenido, fecha de publicación e imagen. La imagen puede ser considerada para almacenar en la base de datos como texto o como objeto.



The screenshot shows a web browser window with the address bar displaying 'localhost/Lenguajes/CrearPost.html'. The page has a green header with the title 'Lenguajes de programación' and a navigation menu with links: Home, Formulario, Login, Comentarios, and Modificar. The main content area has a yellow background and is titled 'Crear Nuevo Post'. It contains a form with the following fields: 'Autor:' with a text input, 'Título:' with a text input, 'Contenido:' with a large text area, 'Fecha de Publicación:' with a date picker showing 'dd/mm/aaaa', and 'Imagen:' with a file upload button labeled 'Seleccionar archivo' and the text 'Ninguno archivo selec.'. At the bottom of the form is a blue button labeled 'Publicar'.

Este es un archivo HTML que contiene un formulario para crear un nuevo post en un sitio web sobre lenguajes de programación. El encabezado incluye un título y un menú de navegación con enlaces a otras páginas. El cuerpo contiene el formulario que tiene campos para el autor, título, contenido, fecha de publicación e imagen. Al final del formulario hay un botón para enviar y publicar el nuevo post. El estilo visual del formulario se especifica en un archivo de hoja de estilo CSS llamado "post.css".

Los estilos CSS aplicados en el código son para dar formato visual a una página web que tiene un encabezado con un menú de navegación, una sección de contenido y un formulario para crear un nuevo post. Los estilos generales incluyen la selección de fuente, color de fondo y color de texto para todo el cuerpo de la página. Los estilos

para el menú de navegación incluyen la selección de fuente, color de fondo y color de texto para los elementos de la lista del menú, así como el efecto de cambio de color al pasar el cursor sobre ellos y la selección del elemento activo. Los estilos para el formulario incluyen la selección de fuente, tamaño y formato de los campos de entrada, así como el formato y color del botón de envío.

```
1 <?php
2 require_once('registrar.php');
3
4 if(isset($_POST["enviar"])) {
5
6     echo "El formulario se ha realizado y enviado con éxito";
7
8 } else {
9     echo "No se ha podido enviar el formulario";
10 }
```

Este código PHP verifica si el formulario ha sido enviado al servidor después de hacer clic en el botón de "enviar". Si el formulario ha sido enviado, se mostrará el mensaje "El formulario se ha realizado y enviado con éxito". De lo contrario, se mostrará el mensaje "No se ha podido enviar el formulario". En este caso, se está utilizando un condicional if para verificar si la variable `$_POST["enviar"]` está establecida, lo que indica que se ha enviado el formulario. La función `isset()` se utiliza para comprobar si la variable está definida y no es nula. La línea `require_once('registrar.php')` se utiliza para incluir un archivo PHP que probablemente contenga el código para procesar el formulario enviado y almacenar los datos en una base de datos, por ejemplo.

```
1 <?php
2 $Autor = $_POST['Autor'];
3 $Titulo = $_POST['Titulo'];
4 $Contenido = $_POST['Contenido'];
5 $Fecha = $_POST['Fecha'];
6
7
8 $conexion = new mysqli('localhost', 'root','','test' );
9 $consulta = "INSERT INTO `comentarios` (`Autor`, `Titulo`, `Contenido`, `Fecha`) VALUES (?, ?, ?, ?)";
10 $insertar = $conexion->prepare($consulta);
11 $resultado = $insertar->execute([$Autor, $Titulo, $Contenido, $Fecha]);
```

Este fragmento de código PHP recibe los valores de las variables Autor, Titulo, Contenido y Fecha a través del método POST, que se enviaron desde un formulario HTML. Luego, crea una conexión a una base de datos MySQL utilizando la clase `mysqli`.

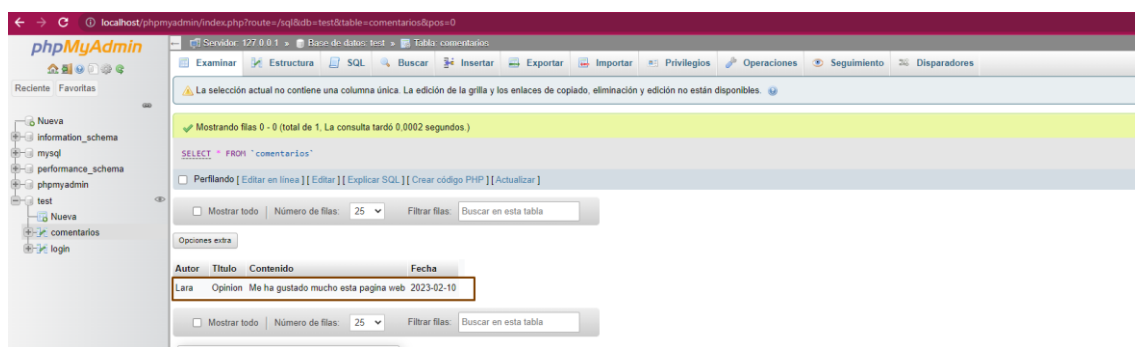
A continuación, prepara una consulta SQL que inserta los valores de las variables mencionadas en la tabla comentarios de la base de datos. La consulta preparada

contiene parámetros de marcador de posición (?) en lugar de los valores reales, lo que hace que la consulta sea más segura y previene ataques de inyección de SQL.

Luego, se ejecuta la consulta preparada mediante el método `execute()`, pasando un array con los valores de las variables para reemplazar los marcadores de posición en la consulta.

Si la consulta se ejecuta correctamente, la variable `$resultado` contendrá `true`, lo que indica que se ha insertado un nuevo registro en la tabla comentarios.

En esta sección, se muestra una página con todas las entradas publicadas por todos los usuarios. Debemos utilizar un diseño atractivo y usable.



Aquí se puede apreciar como se ha almacenado el comentario relleno posteriormente en el formulario

```
1 <?php
2 $consulta = "select * from comentarios";
3 $conexion = new mysqli('localhost', 'root', '', 'test');
4
5 $insertar = $conexion->prepare($consulta);
6 $resultado = $conexion->query($consulta);
7
8 echo("<table border='1'>");
9     echo("<tr>");
10         echo("<th>Autor</th>");
11         echo("<th>Título</th>");
12         echo("<th>Contenido</th>");
13         echo("<th>Fecha</th>");
14     echo("</tr>");
15     while($row=$resultado->fetch_assoc()){
16         echo("<tr>");
17             echo("<td>".$row["Autor"]."</td>");
18             echo("<td>".$row["Título"]."</td>");
19             echo("<td>".$row["Contenido"]."</td>");
20             echo("<td>".$row["Fecha"]."</td>");
21         echo("</tr>");
22     }
23 echo("</table>");
```


Este código PHP se utiliza para recuperar y mostrar en una tabla los comentarios almacenados en la base de datos MySQL.

Primero, se establece una conexión a la base de datos utilizando el objeto mysqli. Luego, se define la consulta SQL que se desea realizar para seleccionar todos los comentarios de la tabla "comentarios".

A continuación, se prepara la consulta con el método prepare() y se ejecuta con el método query(). El resultado de la consulta se almacena en la variable \$resultado.

Luego, se imprime el inicio de una tabla HTML con las etiquetas <table>, <tr> y <th> para las columnas de la tabla. Se utiliza un bucle while para iterar a través de cada fila de resultado devuelta por la consulta.

Dentro del bucle while, se utiliza la función fetch_assoc() para obtener la siguiente fila de resultados como un arreglo asociativo. A continuación, se imprimen los valores de los campos "Autor", "Titulo", "Contenido" y "Fecha" en celdas de tabla utilizando las etiquetas <td>.

Finalmente, se cierra la tabla HTML con la etiqueta </table> y se muestra en el navegador.

Tendremos una sección para eliminar / actualizar entradas. No es necesario implementar esta funcionalidad, pero sí que necesitamos que el usuario deba autenticarse para poder acceder.



HTML

Este es un archivo HTML que contiene un encabezado con el título "Eliminar/Actualizar". En la sección <head>, se establece la codificación de caracteres como UTF-8 y se establece el título de la página. También hay un enlace a un archivo CSS para aplicar estilos a la página.

En el cuerpo de la página, hay un encabezado de nivel 1 con el título "Lenguajes de programación" y una lista de navegación con enlaces a otras páginas del sitio web, como "Home", "Formulario", "Login", "Comentarios" y "Modificar".

Dentro del elemento <div class="container">, hay otro encabezado de nivel 1 con el título "Eliminar o Actualizar". Debajo de este encabezado hay dos botones con las clases "delete" y "update" respectivamente, que se utilizarán para eliminar o actualizar elementos en la base de datos.

CSS

Este código contiene los estilos CSS para la página HTML que muestra los botones de "Eliminar" y "Actualizar" en la sección de "Modificar" de la aplicación web.

En primer lugar, se establecen los estilos generales de la página: se usa la fuente 'Open Sans', se define un margen de 0, un color de fondo de #f5f5f5 y un color de texto de #333.

A continuación, se definen los estilos del encabezado (header) que contiene el título de la página y el menú de navegación. El fondo del encabezado es de color #4CAF50 y el texto es de color blanco. Se establece un padding de 20px y se alinea el contenido al centro con la propiedad display:flex y justify-content:space-between.

Luego, se definen los estilos para la sección principal de la página (section). Se le da un padding de 50px y se justifica el texto.

Se establecen estilos para los elementos p: se define un line-height de 1.5.

A continuación, se definen los estilos del menú de navegación (menu). El fondo es de color #333, el contenido está alineado al centro con justify-content: space-between, los elementos de la lista li tienen un margen de 0 10px y los enlaces a tienen un padding de 10px 20px, un color de texto de blanco y un transition: background-color 0.3s ease.

Se define un hover para los enlaces del menú que cambia el fondo a #4CAF50.

Luego, se establecen los estilos para el título (h1) de la sección "Eliminar o Actualizar". Se define un tamaño de fuente de 3rem y un margen inferior de 2rem.

Los botones (button) tienen un fondo de color #4CAF50, un color de texto de blanco, un padding de 1rem 2rem, un borde sin estilo, un borde redondeado de 5px, un cursor de puntero y un tamaño de fuente de 1.2rem. Además, tienen un transition: background-color 0.3s ease.

Cuando el cursor se posa sobre el botón, cambia el fondo a #3e8e41 en el caso del botón "Eliminar" (delete) y a #006f8e en el caso del botón "Actualizar" (update).

Esta primera fase de diseño y análisis nos debe permitir abordar la implementación de la aplicación web con garantías. Para ello, proponemos explicar con un algoritmo la tarea de publicar una entrada.

Un algoritmo es un conjunto ordenado de instrucciones que permiten resolver un problema o llevar a cabo una tarea específica. En el contexto de la informática, los algoritmos son utilizados para describir procesos que pueden ser llevados a cabo por una computadora. Los algoritmos se pueden expresar en diversos lenguajes de programación y se utilizan en una amplia variedad de aplicaciones, desde la resolución de problemas matemáticos hasta la búsqueda y ordenamiento de datos en grandes bases de datos. Un buen algoritmo debe ser claro, preciso y eficiente para que pueda ser implementado y ejecutado de manera efectiva.

Algoritmo de la aplicación:

1. Realizar una reunión con el equipo de desarrollo para determinar las herramientas y librerías que se utilizarán para desarrollar la aplicación web.
2. Crear un entorno de pruebas con un servidor Apache local para desplegar la aplicación.
3. Crear una página de inicio que contenga una explicación de las diferencias entre lenguajes de programación orientada a objetos, a eventos y lenguajes procedimentales.
4. Al acceder a la página de inicio, almacenar en una cookie la IP del equipo que está accediendo y la fecha de acceso utilizando PHP.
5. Crear un formulario que permita al usuario escribir un post del blog del cliente, incluyendo email del autor, título, contenido, fecha de publicación e imagen.
6. Al enviar el formulario, confirmar al autor el correcto funcionamiento de la tarea.
7. Crear una página que muestre todas las entradas publicadas por todos los usuarios utilizando un diseño atractivo y usable.
8. Crear una sección para eliminar / actualizar entradas, pero requerir que el usuario se autentique para poder acceder.

Caso de uso

Flujo principal:

1. El usuario accede a la página de publicación de entradas en el blog.
2. El sistema muestra un formulario de publicación de entradas con los campos requeridos: email del autor, título, contenido, fecha de publicación e imagen.
3. El usuario rellena los campos y envía el formulario.
4. El sistema valida los campos del formulario y, si son correctos, almacena la entrada en la base de datos.
5. El sistema muestra una confirmación al usuario de que la entrada ha sido publicada correctamente.

Flujo alternativo:

- En el paso 4, si los campos del formulario no son correctos, el sistema muestra un mensaje de error al usuario.
- Postcondiciones: La entrada es almacenada en la base de datos y es mostrada en la sección de todas las entradas publicadas.

Excepciones:

- Si el usuario no está registrado en el sistema, el sistema muestra un mensaje de error y no permite el acceso a la página de publicación de entradas.
- Si el usuario intenta acceder a la sección de eliminación/actualización de entradas sin autenticarse previamente, el sistema redirige al usuario a la página de inicio de sesión.

Algoritmo, caso de uso y desarrollo de código sus diferencias:

- **Algoritmo:** Es una secuencia de pasos o instrucciones que se llevan a cabo para resolver un problema específico. Un algoritmo describe cómo se deben realizar ciertas tareas en un orden lógico y claro. Es importante mencionar que un algoritmo no está vinculado a un lenguaje de programación en particular. Puede ser descrito de manera informal o formal, y puede ser implementado en diferentes lenguajes de programación.
- **Caso de uso:** Es una técnica que se utiliza para describir la funcionalidad de un sistema desde el punto de vista del usuario. Un caso de uso describe una serie de pasos que el usuario sigue para realizar una tarea específica en el sistema. Esta técnica se utiliza para definir los requisitos de un sistema de software, y es una forma de especificar las necesidades del usuario en términos de interacción con el sistema.
- **Desarrollo de código:** Es el proceso de escribir código para implementar un algoritmo o una funcionalidad específica. El código se escribe en un lenguaje de programación específico, siguiendo las reglas y convenciones del mismo. El código debe ser claro, fácil de entender y mantener, y debe ser eficiente en términos de tiempo y recursos utilizados.

En resumen, un algoritmo es una descripción de la solución a un problema, un caso de uso describe cómo el usuario interactúa con el sistema para realizar una tarea, y el desarrollo de código es la implementación de una solución en un lenguaje de programación específico. Todos estos conceptos son importantes en el desarrollo de software y están interrelacionados entre sí.