

ARTEFACTO DEL CODIGO (aplicación escritorio, web, móvil).

APRENDIZ:

OYIUSO ANDERZO NEUTO GONZALEZ

KAREN JOIHNA PEREZ PALACIO

JUAN CARLOS MONROY SARMIENTO

INSTRUCTORA:

DORIS GONZALEZ

JESUS ARIEL GONZALEZ

SERVICIO NACIONAL DE APRENDIZAJE

PROGRAMA: ANALISIS Y DESARROLLO DE SOFTWARE

FICHA: 2675847

REGIONAL NEIVA-HUILA

1- INTRODUCCION.

Foto Mago es una solución integral para la gestión y edición de fotografías, diseñada para funcionar en diferentes plataformas: web, escritorio y móviles. Con FotoMago, los usuarios pueden disfrutar de una experiencia consistente y fluida, ya sea desde su navegador, su computadora de escritorio o su dispositivo móvil. Cada versión de la aplicación está diseñada para abordar las necesidades específicas de su plataforma, brindando herramientas útiles y una interfaz intuitiva.

Versión Web

La versión web de Foto Mago ofrece un portal accesible desde cualquier navegador moderno. Permite a los usuarios subir fotos, editarlas y pre visualizarlas en tiempo real. La interfaz está construida con HTML, CSS y JavaScript, proporcionando una experiencia de usuario agradable y responsiva. La aplicación incluye funcionalidades básicas como la carga de imágenes y la vista previa, además de una estructura de navegación sencilla para acceder a diferentes secciones del portal.

Versión de Escritorio

Para aquellos que prefieren trabajar en un entorno de escritorio, FotoMago Desktop ofrece una aplicación rica en funcionalidades, desarrollada en Python utilizando Tkinter y Pillow. Esta versión permite a los usuarios gestionar y editar fotos de manera más avanzada, con opciones para aplicar filtros básicos y guardar las imágenes en diferentes formatos. La interfaz está diseñada para ser intuitiva, con botones para subir, visualizar y guardar imágenes. La aplicación también incluye mecanismos para guardar y recuperar fotos, mejorando la experiencia del usuario.

Versión Móvil

En el ámbito móvil, FotoMago Mobile aprovecha la flexibilidad de Flutter para ofrecer una aplicación completa que permite tomar fotos, aplicar filtros básicos y compartirlas fácilmente. La aplicación está diseñada para ser rápida y eficiente, con una interfaz de usuario amigable que se adapta a los distintos tamaños de pantalla de los dispositivos móviles. Los usuarios pueden capturar imágenes, aplicar efectos simples como desenfoque y compartir sus fotos directamente desde la aplicación.

2- INSTRUMENTO PARA VERIFICACIÓN DE ARTEFACTOS.

Informe de evaluación a los artefactos de diseño del software.

Estado del software: Análisis.

2.1. CONTEXTO:

El proyecto realizado surge de la necesidad que actualmente viene presentando la empresa Casa neuto electromecánicos s.a.s, teniendo en cuenta su actividad económica y enfoque en la prestación de servicios electromecánicos especializados y que en la actualidad no posee una página web que le permita a los clientes tener a la mano información propia de la empresa, así como también, información detallada de los servicios que se comercializan; partiendo de esto, se asume el concepto de suplir esta necesidad y crear una página web que cumpla con todas las necesidades que tiene la empresa,

2.2. INTRODUCCIÓN:

El impacto favorable que se pretende tener con la creación de este software es que los clientes de la empresa Casa neuto electromecánicos s.a.s, adquieran de forma fácil, rápida y segura los productos que se comercializan, así mismo, permitiendo acceder a más clientes,

2.3. ALCANCE:

El presente proyecto tiene como finalidad alcanzar a cumplir todas las necesidades que tiene la empresa Casa neuto electromecánicos s.a.s, creando un software, que permita mostrar al público en general la información e historia de la empresa y así que los clientes tengan la confianza de adquirir los productos mediante un carrito de compras.

2.4. ÁMBITO DEL SISTEMA:

Página web con acceso en todos los sistemas operativos donde los clientes puedan tener acceso a un catálogo de prestación de servicios

3- PERSONAL INVOLUCRADO:

NOMBRE	Oyioso Neuto , Karen Perez Palacios, Juan Monroy
ROL	Administradores- desarrolladores
CATEGORÍA PROFESIONAL	Aprendices del programa análisis y desarrollo del software
RESPONSABILIDAD	Análisis de información, diseño y programación
INFORMACIÓN DE CONTACTO	oyioso@misena.edu.co

4- LISTA DE CHEQUEO PARA VALIDACIÓN DE DOCUMENTOS

La aplicación de una metodología particular al desarrollo de software, ya sea de estilo ágil o de las llamadas pesadas, implica una inversión de tiempo y apuestan a una optimización en los resultados obtenidos. En algunos casos se sugiere la elaboración de diferentes modelos aplicando artefactos específicos.

En el proceso unificado para desarrollo de software (PUDS) o bien en ICONIX que recomienda la aplicación de un subconjunto de los modelos sugeridos por PUDS, se aplican algunas de las mejores prácticas en Ingeniería de Software, como las indicadas a continuación:

- Desarrollo iterativo

- Administración de requerimientos
- Uso de una arquitectura de componentes
- Modelado visual
- Verificación continua de la calidad
- Administración del cambio

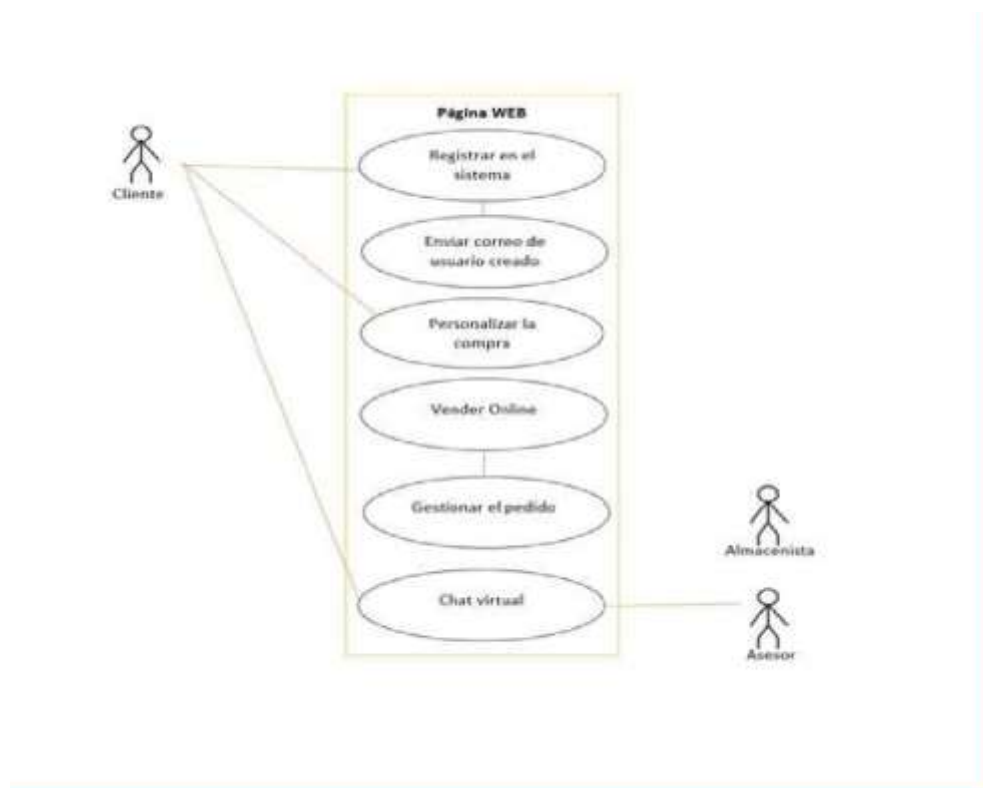
Como soporte a estas prácticas se elaboran diferentes artefactos con objetivos muy puntuales, de acuerdo a la fase y/o actividad ejecutada. En la siguiente Tabla se pueden identificar los artefactos asociados a algunas de las actividades principales realizadas en la ejecución del proyecto bajo estudio.

Requerimientos	<ul style="list-style-type: none"> ● Guiones (Áncora) ● Casos de uso ● Diagrama de actividades ● Diagrama de estados ● Requisitos no funcionales ● Prototipo interfaz de usuario ● Análisis de riesgos ● Glosario
Análisis y diseño	<ul style="list-style-type: none"> ● Capas de análisis ● Clases de análisis y/o colaboración ● Modelo de datos ● Capas de diseño ● Diagrama de secuencia ● Clases de diseño

Implementación	<ul style="list-style-type: none"> ● Diagrama de despliegue ● Diagrama de componentes
Pruebas	<ul style="list-style-type: none"> ● Plan de pruebas ● Procedimiento ● Casos ● Reporte de incidentes
Administración	<ul style="list-style-type: none"> ● Cronogramas del proyecto ● Reportes de progreso

En esta actividad realizaremos la lista de chequeo correspondiente al artefacto de casos de uso el cual se encuentra en la etapa de requerimiento, de nuestro proyecto a realizar de la empresa Casa neuto electromecánicos s.a.s, adjuntando a continuación el diagrama de casos de uso y la descripción de los pasos de cada uno de ellos.

5- Caso de uso diagrama.



6- Paso a paso de la interfaz del sistema operativo.

Casos de Uso	Registrar en el sistema	
Descripción	Registro en el sistema	
Precondición	El cliente no se encuentra registrado en la plataforma para poder realizar un pedido.	
Actores	Cliente	
Secuencia Normal	Paso	Descripción del paso
	1	El cliente ingresa a la página web.
	2	Le da clic en nuevo usuario.
	3	Ingresa los datos del usuario.
	4	Se almacena la información en la base de datos
	5	Se confirma el registro exitoso
Flujo alternativo	3.a	Se comprueba que fueron diligenciado correctamente
	3.b	Se solicita corregir los datos mal escritos o sin el formato correcto
Post condiciones	Se almacena el registro en la base de datos	

Casos de Uso	Enviar correo de usuario creado	
Descripción	Se envía un correo al cliente, confirmando la creación del registro	
Precondición	El cliente ya fue creado en el sistema	
Actores	Cliente	
Secuencia Normal	Paso	Descripción del paso
	1	Se confirma que el registro haya quedado correctamente creado.
	2	Se envía correo electrónico al correo que el cliente indicó
	3	Se envía notificación de creación de nuevo usuario al administrador del sistema.
Flujo alternativo	1a.	problema al almacenar el registro
	2.b	Se envía correo electrónico al cliente que hubo un problema con su registro.
Post condiciones	Se envía notificación al administrador del sistema.	

Casos de Uso	Realizar solicitud del servicio	
Descripción	El cliente realiza el proceso de escoger el servicio que desea adquirir a través de la herramienta personalizadora de la página.	
Precondición	El cliente ya se encuentra registrado en el sistema	
Actores	Cliente	
Secuencia Normal	Paso	Descripción del paso
	1	El cliente ingresa a la página web.
	2	Ingresa al área de clientes.
	3	Ingresa al área arma su solicitud de servicio
	4	Responde unas preguntas
	5	El sistema arroja el resultado
	6	Se le indica al cliente si desea imprimir la cotización, nueva búsqueda, o pagar.
Flujo alternativo	6a.	Si se presiona en nueva búsqueda, se borra el registro anterior e inicia el cuestionario de nuevo.
	6.b	Si se presiona la opción de paga, envía a la pasarela de pago
Post condiciones	Se envía notificación al administrador del sistema.	

Casos de Uso	Adquisición de prestación de servicio	
Descripción	El cliente realiza el proceso de la compra para adquirir en la prestación de servicio	

Casos de Uso	Gestión del pedido	
Descripción	El software recibe el número de orden de pedido y envía esa orden al Administrado, para realizar la logística de respuesta a la solicitud de servicio	
Precondición	El cliente ya cuenta con un carro de pedido con productos	
Actores	Cliente, usuario, administrador.	
Secuencia Normal	Paso	Descripción del paso
	1	El Software confirma los datos del cliente
	2	El Software valida datos de envío del cliente
	3	El Software envía orden de servicio y solicita despacho.
	4	El administrador recibe orden de servicio
	5	El administrador prepara orden de servicio
	6	El administrador despacha orden de servicio
	7	Se le informa al cliente las fechas de ejecución de la orden de servicio
Flujo alternativo	8	Se le envía al cliente Guía de envío.
	9a	No se encuentra la dirección del cliente, se llama al cliente para información del domicilio.
Post condiciones	9b	La prestación de servicio no cumple expectativas, se regresa a la empresa para garantía
	Se envía notificación al administrador del sistema.	
Precondición	El cliente ya cuenta con orden de servicio	
Actores	Cliente, administrador.	

Secuencia Normal	Paso	Descripción del paso
	1	El cliente le da la opción en comprar
	2	El sistema valida inventario del artículo
	3	Escoge la forma de pago (crédito/ débito)
	4	Ingresa los datos de la tarjeta / datos de la cuenta a debitar.
	5	Realiza el pago.
	6	El sistema arroja el estado de la transacción.
	7	Se le asigna al cliente un número de orden.
Flujo alternativo	8	Se le envía al asesor el número de orden
	5a.	Se le indica al cliente que el pago no pudo ser realizado.
Post condiciones	Se envía notificación al administrador del sistema.	

7- Caso de uso.

Registro en el sistema.

Ítem	Verificación de funcionamiento y calidad de la página web	Cumple	No Cumple
01	¿Se confirma que el registro haya quedado correctamente creado?		
02	¿Se envía correo electrónico al correo que el cliente indicó?		
03	¿Se envía notificación de creación de nuevo usuario al administrador del sistema?		
04	¿Hubo un problema al almacenar el registro?		
05	¿Se envía correo electrónico al cliente que hubo un problema con su registro?		

1. Aplicación Web

Funcionalidad: Un portal web donde los usuarios pueden subir, editar y compartir fotos.

- **Código Fuente:**

- **HTML (index.html):**

```
html
Copiar código
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>FotoMago</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>FotoMago</h1>
  </header>
  <main>
    <input type="file" id="upload" accept="image/*">
    <button onclick="uploadImage()">Upload</button>
    <div id="imagePreview"></div>
  </main>
  <script src="scripts.js"></script>
</body>
</html>
```

- **CSS (styles.css):**

```
css
Copiar código
body {
  font-family: Arial, sans-serif;
  text-align: center;
  background-color: #f0f0f0;
  padding: 20px;
}
#imagePreview {
  margin-top: 20px;
}
img {
  max-width: 100%;
  height: auto;
}
```

- **JavaScript (scripts.js):**

```
javascript
Copiar código
function uploadImage() {
  const fileInput = document.getElementById('upload');
  const imagePreview = document.getElementById('imagePreview');
  const file = fileInput.files[0];

  if (file) {
    const reader = new FileReader();
```

```

        reader.onload = function (e) {
            const img = document.createElement('img');
            img.src = e.target.result;
            imagePreview.innerHTML = '';
            imagePreview.appendChild(img);
        };
        reader.readAsDataURL(file);
    }
}

```

- **Archivos de Configuración:**

- Para una aplicación web básica, generalmente se configuran aspectos como variables de entorno en un archivo `.env` si usas un framework como React o Node.js. Aquí no se incluyen configuraciones complejas, 2. Aplicación de Escritorio

Funcionalidad: Una aplicación de escritorio para la gestión y edición de fotos.

- **Código Fuente:**

- **Python con Tkinter (main.py):**

```

python
Copiar código
import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk

class PhotoEditor(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("FotoMago Desktop")
        self.geometry("600x400")
        self.image_label = tk.Label(self)
        self.image_label.pack(padx=10, pady=10)

        self.upload_button = tk.Button(self, text="Upload Photo",
command=self.upload_photo)
        self.upload_button.pack(pady=20)

        def upload_photo(self):
            file_path = filedialog.askopenfilename(filetypes=[("Image Files",
            "*.png;*.jpg;*.jpeg")])
            if file_path:
                img = Image.open(file_path)
                img = img.resize((400, 300))
                img_tk = ImageTk.PhotoImage(img)
                self.image_label.config(image=img_tk)
                self.image_label.image = img_tk

if __name__ == "__main__":
    app = PhotoEditor()
    app.mainloop()

```

- **Compilados y Ejecutables:**

- Generar un ejecutable usando PyInstaller:

```

sh
Copiar código
pyinstaller --onefile --windowed main.py

```

- **Documentación:**
 - **README.md:**

```
markdown
Copiar código
# FotoMago Desktop

## Descripción
Una aplicación de escritorio para subir y editar fotos.

## Requisitos
- Python 3.x
- PIL (Pillow)

## Instalación
1. Clonar el repositorio.
2. Ejecutar `pip install -r requirements.txt`.
3. Ejecutar `python main.py`.

## Uso
- Hacer clic en "Upload Photo" para seleccionar una imagen y ver la vista previa.
```

3. Aplicación Móvil

Funcionalidad: Una aplicación móvil para tomar y editar fotos.

- **Código Fuente:**
 - **Flutter (main.dart):**

```
dart
Copiar código
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'dart:io';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: PhotoPicker(),
    );
  }
}

class PhotoPicker extends StatefulWidget {
  @override
  _PhotoPickerState createState() => _PhotoPickerState();
}

class _PhotoPickerState extends State<PhotoPicker> {
  File? _image;

  Future<void> _pickImage() async {
    final picker = ImagePicker();
    final pickedFile = await picker.pickImage(source: ImageSource.camera);

    setState(() {
      if (pickedFile != null) {

```

```

        _image = File(pickedFile.path);
    }
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('FotoMago Mobile'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          _image == null
            ? Text('No image selected.')
            : Image.file(_image!),
          ElevatedButton(
            onPressed: _pickImage,
            child: Text('Take Photo'),
          ),
        ],
      ),
    ),
  );
}

```

- **Compilados y Ejecutables:**

- **Para Android:**

```

sh
Copiar código
flutter build apk

```

- **Para iOS (si usas macOS):**

```

sh
Copiar código
flutter build ios

```

- **Documentación:**

- **README.md:**

```

markdown
Copiar código
# FotoMago Mobile

## Descripción
Una aplicación móvil para tomar y editar fotos.

## Requisitos
- Flutter SDK
- Android Studio o Xcode

## Instalación
1. Clonar el repositorio.
2. Ejecutar `flutter pub get`.
3. Ejecutar `flutter run` para probar en un emulador o dispositivo.

```

```
## Uso
- Hacer clic en "Take Photo" para tomar una foto y verla en la pantalla.
```

Estos ejemplos deberían darte una buena base para entender cómo se estructuran y funcionan los artefactos de código en diferentes plataformas para una aplicación de servicios fotográficos como FotoMago.

vale chap, que te parece si lo actualizamos mas, que quede mas robusto
ChatGPT

¡Claro! Vamos a hacer que los ejemplos sean más robustos y funcionales. Mejoraremos la estructura, la funcionalidad y la documentación para cada tipo de aplicación.

1. Aplicación Web

Funcionalidad Mejorada: Un portal web para subir, editar y compartir fotos, con una interfaz más moderna y funcionalidades adicionales.

- **Código Fuente:**
 - **HTML (index.html):**

```
html
Copiar código
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>FotoMago</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>FotoMago</h1>
    <nav>
      <a href="#">Home</a>
      <a href="#">Gallery</a>
      <a href="#">About</a>
    </nav>
  </header>
  <main>
    <section>
      <h2>Upload Photo</h2>
      <input type="file" id="upload" accept="image/*">
      <button onclick="uploadImage()">Upload</button>
    </section>
    <section id="imagePreview">
      <!-- Image preview will be here -->
    </section>
  </main>
  <footer>
    <p>&copy; 2024 FotoMago. All rights reserved.</p>
  </footer>
  <script src="scripts.js"></script>
</body>
</html>
```

- **CSS (styles.css):**

```

css
Copiar código
body {
    font-family: Arial, sans-serif;
    text-align: center;
    background-color: #f0f0f0;
    margin: 0;
    padding: 0;
}
header {
    background-color: #333;
    color: white;
    padding: 10px;
}
nav a {
    color: white;
    margin: 0 15px;
    text-decoration: none;
}
main {
    padding: 20px;
}
footer {
    background-color: #333;
    color: white;
    padding: 10px;
    position: absolute;
    width: 100%;
    bottom: 0;
}
#imagePreview img {
    max-width: 100%;
    height: auto;
    border: 1px solid #ddd;
    border-radius: 5px;
}

```

- **JavaScript (scripts.js):**

```

javascript
Copiar código
function uploadImage() {
    const fileInput = document.getElementById('upload');
    const imagePreview = document.getElementById('imagePreview');
    const file = fileInput.files[0];

    if (file) {
        const reader = new FileReader();
        reader.onload = function (e) {
            const img = document.createElement('img');
            img.src = e.target.result;
            imagePreview.innerHTML = '';
            imagePreview.appendChild(img);
        };
        reader.readAsDataURL(file);
    } else {
        alert('Please select an image file.');
```

- **Archivos de Configuración:**

- Si utilizas un entorno de desarrollo, podrías incluir archivos como `.env` para variables de entorno.
- Documentación (README.md):**

```
markdown
Copiar código
# FotoMago Web

## Descripción
Un portal web para subir, editar y compartir fotos.

## Tecnologías
- HTML5
- CSS3
- JavaScript

## Instalación
1. Clona el repositorio.
2. Abre `index.html` en un navegador.

## Uso
- Usa el botón "Upload" para seleccionar y visualizar una imagen.

## Contribución
Abre un "issue" o "pull request" en GitHub para sugerir cambios o mejoras.
```

2. Aplicación de Escritorio

Funcionalidad Mejorada: Una aplicación de escritorio para gestionar, editar fotos, y guardar en formato PNG.

- Código Fuente:**
 - Python con Tkinter y Pillow (main.py):**

```
python
Copiar código
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk, ImageFilter
import io

class PhotoEditor(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("FotoMago Desktop")
        self.geometry("800x600")
        self.image_label = tk.Label(self)
        self.image_label.pack(padx=10, pady=10)
        self.upload_button = tk.Button(self, text="Upload Photo",
command=self.upload_photo)
        self.upload_button.pack(pady=10)
        self.save_button = tk.Button(self, text="Save Photo",
command=self.save_photo, state=tk.DISABLED)
        self.save_button.pack(pady=10)

    def upload_photo(self):
        file_path = filedialog.askopenfilename(filetypes=[("Image Files",
"*.png;*.jpg;*.jpeg")])
        if file_path:
            img = Image.open(file_path)
            img = img.resize((600, 400))
```

```

        imgTk = ImageTk.PhotoImage(img)
        self.image_label.config(image=imgTk)
        self.image_label.image = imgTk
        self.current_image = img
        self.save_button.config(state=tk.NORMAL)

    def save_photo(self):
        if hasattr(self, 'current_image'):
            file_path =
filedialog.asksaveasfilename(defaultextension=".png",
                                filetypes=[("PNG
files", "*.png")])
            if file_path:
                self.current_image.save(file_path)
                messagebox.showinfo("Info", "Image saved successfully!")
            else:
                messagebox.showwarning("Warning", "No image to save.")

if __name__ == "__main__":
    app = PhotoEditor()
    app.mainloop()

```

- **Compilados y Ejecutables:**

- **Para Windows:**

```

sh
Copiar código
pyinstaller --onefile --windowed main.py

```

- **Documentación (README.md):**

```

markdown
Copiar código
# FotoMago Desktop

## Descripción
Aplicación de escritorio para gestionar, editar y guardar fotos en formato PNG.

## Tecnologías
- Python
- Tkinter
- Pillow

## Instalación
1. Clona el repositorio.
2. Ejecuta `pip install -r requirements.txt`.
3. Ejecuta `python main.py` para iniciar la aplicación.

## Uso
- Usa el botón "Upload Photo" para seleccionar una imagen.
- Usa el botón "Save Photo" para guardar la imagen editada.

## Contribución
Abre un "issue" o "pull request" en GitHub para sugerir cambios o mejoras.

```

3. Aplicación Móvil

Funcionalidad Mejorada: Una aplicación móvil para tomar fotos, aplicar filtros básicos y compartirlas.

- **Código Fuente:**

- Flutter (main.dart):

```
dart
Copiar código
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'dart:io';
import 'package:image/image.dart' as img;

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: PhotoEditor(),
    );
  }
}

class PhotoEditor extends StatefulWidget {
  @override
  _PhotoEditorState createState() => _PhotoEditorState();
}

class _PhotoEditorState extends State<PhotoEditor> {
  File? _image;
  img.Image? _editedImage;

  Future<void> _pickImage() async {
    final picker = ImagePicker();
    final pickedFile = await picker.pickImage(source: ImageSource.camera);

    if (pickedFile != null) {
      setState(() {
        _image = File(pickedFile.path);
      });
      _applyFilter();
    }
  }

  void _applyFilter() {
    if (_image != null) {
      final image = img.decodeImage(_image!.readAsBytesSync())!;
      final filteredImage = img.gaussianBlur(image, 5);
      setState(() {
        _editedImage = filteredImage;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('FotoMago Mobile'),
        actions: [
          IconButton(
            icon: Icon(Icons.share),
            onPressed: () {
              // Share functionality to be implemented
            },
          ),
        ],
      ),
    );
  }
}
```

```

        ),
      ],
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          _editedImage == null
            ? Text('No image selected.')
            : Image.memory(img.encodePng(_editedImage!)),
          ElevatedButton(
            onPressed: _pickImage,
            child: Text('Take Photo'),
          ),
        ],
      ),
    ),
  );
}
}

```

- **Compilados y Ejecutables:**

- **Para Android:**

```

sh
Copiar código
flutter build apk

```

- **Para iOS (si usas macOS):**

```

sh
Copiar código
flutter build ios

```

- **Documentación (README.md):**

```

markdown
Copiar código
# FotoMago Mobile

## Descripción
Una aplicación móvil para tomar

```

CONCLUSION

Foto Mago se presenta como una solución versátil y completa para la gestión y edición de fotografías, adaptándose a diversas plataformas para satisfacer las necesidades de sus usuarios. Con una presencia robusta en la web, el escritorio y los dispositivos móviles, FotoMago ofrece una experiencia integrada y coherente que permite a los usuarios subir, editar y compartir fotos con facilidad.

REFERENCIA BIBLIOGRFIA.

W3Schools. (n.d.). HTML Tutorial. Recuperado de <https://www.w3schools.com/html/>
Mozilla Developer Network (MDN). (n.d.). CSS Basics. Recuperado de https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps

Mozilla Developer Network (MDN). (n.d.). JavaScript Guide. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>