

**SISTEMA IDENTIFICADOR DE LA SALUD DE FRUTOS DE ÁRBOLES DE CACAO  
Y CAFÉ MEDIANTE ALGORITMOS CLASIFICADORES DE MACHINE LEARNING  
Y PROCESAMIENTO DE IMÁGENES**

**JHON MICHAEL DIAZ ARANGO**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SOFTWARE  
MEDELLÍN  
2024**

**SISTEMA IDENTIFICADOR DE LA SALUD DE FRUTOS DE ÁRBOLES DE CACAO  
Y CAFÉ MEDIANTE ALGORITMOS CLASIFICADORES DE MACHINE LEARNING  
Y PROCESAMIENTO DE IMÁGENES**

**JHON MICHAEL DIAZ ARANGO**

**Trabajo de grado para optar al título de Ingeniero de Software**

**Asesor Técnico**

**Juan Carlos Briñez de León**

**Ph.D. en Ingeniería – Sistemas e Informática**

**Asesor Metodológico**

**Diego Hernando Orozco Gómez**

**M.Sc. en Ingeniería – Automatización Industrial**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SOFTWARE**

**MEDELLÍN**

**2024**

## Dedicatoria

A mi familia

## **Agradecimientos**

Agradezco sinceramente a mis asesores Juan Carlos Briñez de León y Diego Hernando Orozco Gómez por su valiosa guía y sus observaciones críticas, que fueron fundamentales para la correcta orientación y finalización de este proyecto.

Finalmente, agradezco profundamente a Dios y mi familia, especialmente a mi madre Viviana Paola Diaz Arango, cuyo constante apoyo y motivación hicieron posible la finalización de este proyecto en el tiempo previsto.

## Contenido

	Pág.
Introducción .....	13
1. Planteamiento del problema.....	14
1.1 Descripción.....	14
1.2 Formulación .....	15
2. Justificación .....	16
3. Objetivos.....	17
3.1 Objetivo general .....	17
3.2 Objetivos específicos.....	17
4. Marco teórico.....	18
4.1 Árbol del cacao y fruto de cacao .....	18
4.2 Café y granos de café .....	19
4.3 Defectos del café .....	21
4.4 Enfermedades en los frutos del cacao .....	22
4.5 Antecedentes en estudios de detección de enfermedades en plantas utilizando procesamiento de imágenes y visión computacional .....	27
5. Metodología.....	33
5.1 Tipo de proyecto.....	33
5.2 Método .....	33
5.3 Instrumentos de recolección de información .....	35
5.3.1 Fuentes primarias.....	35
5.3.2 Fuentes secundarias. ....	36
6. Resultados del proyecto.....	37
6.1 Creación de un banco de imágenes de frutos de árboles de cacao y granos de café.....	37
6.2 Entrenamiento del modelo de machine learning .....	41
6.3 Evaluación del desempeño de los algoritmos clasificadores de machine learning .....	60
6.4 Diseño de la aplicación .....	64
7. Conclusiones.....	82
8. Recomendaciones .....	83

9. Referencias bibliográficas.....	85
10. Bibliografía .....	89

## Lista de figuras

	Pág.
<i>Figura 1.</i> Fruto de cacao .....	19
<i>Figura 2.</i> Imagen de referencia grano de café de tipo arábica y robusta.....	20
<i>Figura 3.</i> Imagen de muestras de varios defectos de café .....	21
<i>Figura 4.</i> Imagen de referencia grano de café con defecto secundario inmaduro o verde .....	22
<i>Figura 5.</i> Fruto de cacao con una mancha blanca indicando etapa de esporulación .....	24
<i>Figura 6.</i> Frutos de cacao con enfermedad pudrición parda o mazorca negra .....	25
<i>Figura 7.</i> Imagen de referencia de la mosquilla del cacao .....	26
<i>Figura 8.</i> Imagen del chinche Hoplophorion pertusa .....	26
<i>Figura 9.</i> Imagen de referencia del insecto Selenothrips rubrocinctus .....	27
<i>Figura 10.</i> Imagen de referencia de Cenicilla .....	28
<i>Figura 11.</i> Collage de imágenes utilizadas para cacao CCN51 saludable en el dataset.....	38
<i>Figura 12.</i> Collage de imágenes utilizadas para cacao CCN51 con enfermedad mazorca negra en el dataset.....	38
<i>Figura 13.</i> Collage de imágenes utilizadas para cacao CCN51 con enfermedad monilia en el dataset .....	39
<i>Figura 14.</i> Collage de imágenes utilizadas en el dataset para grano de café saludable .....	40
<i>Figura 15.</i> Collage de imágenes utilizadas en el dataset para grano de café con defecto negro ..	40
<i>Figura 16.</i> Collage de imágenes utilizadas en el dataset para grano de café con defecto verde ..	41
<i>Figura 17.</i> Histograma de clases en el conjunto de entrenamiento del modelo knn para el dataset de frutos de cacao.....	59
<i>Figura 18.</i> Histograma de clases en el conjunto de prueba del modelo knn para el dataset de frutos de cacao .....	59
<i>Figura 19.</i> Matriz de confusión del modelo knn para el dataset de frutos de cacao .....	60
<i>Figura 20.</i> Porcentaje de precisión modelo knn para el dataset de frutos de cacao .....	61
<i>Figura 21.</i> Tiempo de ejecución reconociendo una imagen modelo random forest dataset granos de café .....	64
<i>Figura 22.</i> Tiempo de ejecución reconociendo una imagen modelo red neuronal dataset granos de café .....	64

<i>Figura 23.</i> Interfaz principal de usuario para la aplicación .....	67
<i>Figura 24.</i> Interfaz selección de imagen de fruto del cacao en la aplicación.....	69
<i>Figura 25.</i> Interfaz selección de imagen de grano de café en la aplicación .....	71
<i>Figura 26.</i> Respuesta de la interfaz de la aplicación cuando no es seleccionada una fotografía .	78
<i>Figura 27.</i> Clasificación que da el modelo en la aplicación para un grano de café con defecto verde.....	78
<i>Figura 28.</i> Clasificación que da el modelo en la aplicación para un fruto de cacao saludable ....	79
<i>Figura 29.</i> Clasificación incorrecta que da el modelo en la aplicación para granos de café de tipo robusta se clasifica como defecto verde.....	80
<i>Figura 30.</i> Clasificación incorrecta que da el modelo en la aplicación para varios granos de café se clasifica como defecto verde .....	80
<i>Figura 31.</i> Clasificación incorrecta que da el modelo en la aplicación para un fruto de cacao donde se observa plantas y nubes en el fondo además se encuentra en contra luz.....	81

## Lista de tablas

	Pág.
Tabla 1. <i>Información de recolección de imágenes de frutos del cacao CCN51</i> .....	37
Tabla 2. <i>Agrupación de imágenes de frutos del cacao y granos de café</i> .....	41
Tabla 3. <i>Reporte de clasificación modelo knn dataset frutos del cacao</i> .....	60
Tabla 4. <i>Reporte de precisión por modelo y dataset</i> .....	63
Tabla 5. <i>Respuestas que da la interfaz al usuario según el modelo seleccionado y la clasificación de la imagen.</i> .....	77

## Resumen

# **SISTEMA IDENTIFICADOR DE LA SALUD DE FRUTOS DE ÁRBOLES DE CACAO Y CAFÉ MEDIANTE ALGORITMOS CLASIFICADORES DE MACHINE LEARNING Y PROCESAMIENTO DE IMÁGENES**

**JHON MICHAEL DIAZ ARANGO**

En el cultivo de cacao y café, las enfermedades como la moniliasis y la mazorca negra en los frutos de cacao, así como los defectos en los granos de café, representan un desafío significativo para los agricultores, afectando la calidad y la producción. Este trabajo presenta una intervención a través del diseño de una aplicación informática para el sistema operativo Microsoft Windows 10, que emplea dos algoritmos clasificadores de machine learning. La aplicación permite a los usuarios cargar imágenes de frutos de cacao y granos de café para identificar su estado de salud, facilitando así la toma de decisiones para el manejo adecuado de los cultivos.

La implementación de esta herramienta ofrece varias ventajas, incluyendo la capacidad de diagnóstico rápido y preciso de las condiciones de los frutos, lo que puede contribuir a la mejora en las prácticas agrícolas y a la optimización de los rendimientos. Además, el desarrollo de un dataset cuidadosamente curado, con más de 1200 imágenes clasificadas de manera rigurosa, garantiza que los modelos entrenados sean robustos y efectivos.

Los resultados obtenidos demuestran altos niveles de precisión en la clasificación, con un 100% para los granos de café y un 94% para los frutos de cacao utilizando el algoritmo Random Forest y KNN, respectivamente. Estos hallazgos indican que la aplicación tiene un gran potencial para ser utilizada en el campo, proporcionando una herramienta valiosa para los agricultores en la detección temprana de enfermedades.

*Palabras claves:* algoritmo, café, cacao, machine learning, enfermedades.

## **Abstract**

# **IDENTIFIER SYSTEM FOR THE HEALTH OF CACAO AND COFFEE TREE FRUITS USING MACHINE LEARNING CLASSIFIER ALGORITHMS AND IMAGE PROCESSING**

**JHON MICHAEL DIAZ ARANGO**

In cacao and coffee cultivation, diseases like moniliasis and black pod in cacao fruits, as well as defects in coffee beans, show a significant challenge for farmers, impacting quality and production. This work presents an intervention through the design of a software application for the Microsoft Windows 10 operating system, employing two machine learning classifier algorithms. The application allows users to upload images of cacao fruits and coffee beans to identify their health status, thus facilitating informed decision-making for proper crop management.

The implementation of this tool offers several advantages, including the capability for quick and accurate diagnosis of the fruits conditions, which can contribute to improvements in agricultural practices and yield optimization. Additionally, the development of a carefully curated dataset with over 1200 rigorously classified images ensures that the trained models are robust and effective.

The results obtained demonstrate high levels of accuracy in classification, with 100% accuracy for coffee beans and 94% for cacao fruits using the Random Forest and KNN algorithms, respectively. These findings indicate that the application has great potential for field use, providing a valuable tool for farmers in the early detection of diseases.

*Keywords:* algorithm, cacao, coffee, diseases, machine learning.

## Glosario

**Cacao (*Theobroma cacao*):** árbol tropical cultivado por sus semillas, las cuales son utilizadas en la producción de chocolate, cacao en polvo y manteca de cacao.

**Condiciones morfológicas:** características físicas y estructurales de los frutos de cacao, como su forma, tamaño, color y textura, que pueden variar dependiendo del estado de salud del fruto.

**Dataset:** conjunto de datos, en este contexto, una colección de imágenes de frutos de cacao en distintas condiciones, utilizadas para entrenar un modelo de clasificación.

**Esporas:** estructuras reproductoras del hongo que permiten su propagación. En el caso de la monilia, las esporas suelen aparecer como una capa blanca o amarillenta en los frutos afectados.

**Fungicida:** producto químico utilizado para prevenir o controlar las enfermedades fúngicas en plantas. En el control de la monilia, los fungicidas son una herramienta importante junto con otras prácticas de manejo.

**Machine learning (aprendizaje automático):** rama de la inteligencia artificial que permite a las máquinas aprender de los datos y hacer predicciones sin ser programadas explícitamente para cada tarea.

**Mazorca Negra (*Phytophthora palmivora*):** enfermedad causada por un hongo que afecta los frutos de cacao, provocando manchas oscuras en la superficie de las mazorcas, que eventualmente se vuelven negras y pueden pudrirse.

**Monilia:** es una enfermedad fúngica que afecta principalmente a los frutos del cacao (*Theobroma cacao*).

## Introducción

El presente trabajo de grado se centra en el desarrollo de una aplicación informática diseñada para identificar el estado de salud de frutos de cacao y granos de café, empleando algoritmos de machine learning y técnicas de procesamiento de imágenes. Este proyecto responde a una necesidad creciente en el sector agrícola, donde las enfermedades como la moniliasis y la mazorca negra en el cacao y los defectos como el grano verde y negro en el café representan serios desafíos para la producción y calidad. Estas afecciones no solo disminuyen el rendimiento, sino que también incrementan los costos de cultivo, afectando directamente a los agricultores y al mercado en general. Así, surge la importancia de implementar una herramienta que facilite la detección temprana y precisa de dichas enfermedades, contribuyendo al manejo eficiente y sostenible de los cultivos.

La estructura del trabajo está organizada para ofrecer una solución práctica y accesible para los agricultores mediante una aplicación compatible con el sistema operativo Microsoft Windows 10. A través de esta aplicación, los usuarios pueden cargar imágenes de frutos de cacao o granos de café y recibir un diagnóstico automatizado sobre su estado de salud. Este diagnóstico es generado por dos modelos de clasificación basados en machine learning, optimizados y adaptados específicamente para cada tipo de cultivo. Los modelos utilizados fueron seleccionados tras un proceso de evaluación en el que se identificaron aquellos con los mayores niveles de precisión y velocidad de procesamiento, de acuerdo con las características de las imágenes.

## 1. Planteamiento del problema

### 1.1 Descripción

La producción de cacao colombiano es reconocida en el mundo por sus características de sabor y aroma, el sector es motor de generación de empleo desarrollo sostenible y económico en muchas economías agrícolas del país (Procolombia, s.f.). El cacao es una referencia dentro del país pues se siembra en 29 de los 32 departamentos (Procolombia, 2020). Así mismo en cifras de 2021 comparando con cifras de 2006 la producción de cacao nacional mostro un crecimiento de 127% pasando de 30.357 a 69.040 toneladas (Fedecacao, 2022). Además del cacao el café es una de las principales fuentes de ingresos de Colombia y desempeña un papel fundamental en la economía nacional.

En términos de empleo, se estima que la producción de café proporciona sustento a más de 500,000 familias en el país, generando oportunidades de trabajo en las zonas rurales donde se cultivan los cafetales (Concafe, 2023). Sin embargo el incremento de las enfermedades y plagas representan una amenaza significativa para la productividad y la calidad de las plantas contribuyendo con un alto porcentaje de las pérdidas de producción, las enfermedades en el cultivo de cacao causan más pérdidas al agricultor que los insectos, llegando a causar pérdidas de hasta el 80% de la producción del cacao, algunas de las enfermedades pueden destruir las mazorcas de una plantación como lo son la monilia y la mazorca negra (Progresia Caribe, 2022). La mazorca negra es reconocida mundialmente como una de las enfermedades más agresivas que afectan al cacao, pudiendo ocasionar hasta un 30% de pérdida en la producción y causar la muerte anual del 10% de las plantas debido a la aparición de cánceres en el tallo (INIAP, s.f.). La detección de los frutos enfermos puede facilitar intervenciones preventivas y mejorar la producción agrícola reduciendo la propagación (Progresia Caribe, 2022).

Este proyecto tiene como objetivo desarrollar un sistema de software de computador para la plataforma Microsoft Windows 10, diseñada para identificar, a través de imágenes que el usuario tiene de los frutos de su cultivo, si un fruto de cacao presenta alguna de dos enfermedades: monilia o mazorca negra y para los granos de café defectos verde o negro, o si se encuentra en

buen estado. Para determinar el estado de salud de los frutos de cacao o granos de café, se emplean algoritmos clasificadores de machine learning y procesamiento de imágenes, lo que permite informar al productor o usuario de la aplicación sobre la condición sanitaria del fruto en la imagen.

## **1.2 Formulación**

¿Es posible que la aplicación identifique las dos principales enfermedades monilia y mazorca negra en los frutos de plantas de cacao y defectos verde o negro en granos de café utilizando los resultados de dos algoritmos computacionales de tipo clasificador basados en machine learning y procesamiento de imágenes?

## 2. Justificación

Este proyecto tiene como objetivo desarrollar una aplicación informática para identificar dos enfermedades, monilia y mazorca negra, en los frutos de cacao, y dos defectos, verde y negro, en los granos de café, utilizando técnicas de procesamiento de imágenes y machine learning. La aplicación permitirá determinar tres estados: el primero, enfermedad o defecto uno; el segundo, enfermedad o defecto dos; y el tercero, fruto o grano saludable, indicando la ausencia de ambas enfermedades o defectos. Este sistema busca facilitar intervenciones preventivas con el fin de mejorar la producción agrícola mejorando directamente la identificación del estado de salud de los frutos de cacao o granos de café.

A nivel social, el proyecto contribuirá a la seguridad de los cultivos, asegurando una producción continua y de alta calidad. Esto permitirá a las comunidades mantener ingresos estables mediante las plantaciones de cacao y café, ya que los frutos de cacao enfermos y los granos de café con defectos podrán ser retirados oportunamente. De esta forma, se reducirán las pérdidas y se evitará la propagación a los frutos o granos que están en buen estado. Desde un punto de vista tecnológico, representa un avance significativo en la aplicación de inteligencia artificial en la agricultura, promoviendo innovaciones que pueden ser adaptadas a otras áreas de cultivo.

Ambientalmente, permitirá una gestión más sostenible de las plantaciones, reduciendo el uso excesivo de pesticidas para manejo de hongos como la moniliasis esto porque el eliminar un fruto infectado con moniliasis evita que otros 20 frutos se contagien en promedio (Agrobanco, s.f.) lo que no solo disminuye los costos de mantenimiento y control de plagas al aplicar únicamente los pesticidas necesarios, sino que también reduce el impacto ambiental.

### **3. Objetivos**

#### **3.1 Objetivo general**

Diseñar una aplicación informática para el sistema operativo Microsoft Windows 10 que identifique a partir del análisis visual de cuatro enfermedades la salud de frutos de árboles de cacao y café mediante el uso de procesamiento de imágenes y dos algoritmos clasificadores de machine learning.

#### **3.2 Objetivos específicos**

Crear un banco de imágenes de frutos de árboles de cacao y granos de café saludables y con las enfermedades moniliasis y mazorca negra en el cacao y defectos de granos de café verde y negro con diversas etapas de madurez garantizando una calidad de imagen aceptable y condiciones de iluminación adecuadas.

Entrenar con el banco de imágenes creado dos modelos de machine learning a partir de la extracción de características relacionadas a morfología, color y textura que permitan la clasificación de imágenes de frutos de cacao y granos de café saludables y con enfermedades.

Evaluar el desempeño de los algoritmos clasificadores de machine learning determinando el porcentaje de precisión de cada modelo en la identificación de los estados de salud de frutos de cacao y granos de café.

## 4. Marco teórico

### 4.1 Árbol del cacao y fruto de cacao

*Theobroma cacao* L. es el nombre científico que recibe el árbol del cacao o cacaotero. El cacao es un árbol americano de origen amazónico, conocido también como cacaotero, ya que por cacao se refiere normalmente al fruto que da dicho árbol, o incluso al producto del secado y la fermentación de las semillas de dicho fruto, el cacaotero es un árbol que necesita de humedad y de calor. Es de hoja perenne y siempre se encuentra en floración, Suele medir alrededor de 7 metros si es cultivado y por encima de 20 en la naturaleza (Equipo editorial, Etecé, 2021).

Requiere sombra (crecen a la sombra de otros árboles más grandes como platanero). El terreno debe ser rico en nitrógeno, magnesio y en potasio, y el clima húmedo, con una temperatura entre los 20 °C y los 30 °C. El cacao es natural de América y se da comúnmente en los territorios de Colombia, Venezuela, Brasil, Ecuador, Perú, Bolivia y Trinidad y Tobago, pero también en las zonas calientes de México y en el continente africano, en Costa de Marfil, Camerún, Nigeria, Togo, República de Congo y Ghana, así como en la selva asiática de Malasia e Indonesia. Actualmente se conocen tres variedades principales del cacao: Criollo (caribeño y centroamericano principalmente), Forastero (amazónico, pero cultivado en África mayormente) y Trinitario (híbrido). En torno a estos tres nombres se agrupan al menos diez familias modernas del cacao (Equipo editorial, Etecé, 2021).

Es un Árbol caulífloro (flores y frutos nacen directamente del tallo y ramas). Sus pequeñas flores de color rosa y sus frutos crecen directamente del tronco y de las ramas más antiguas. Las flores son polinizadas. Un árbol comienza a rendir cuando tiene 4 ó 5 años. En un año, cuando madura, puede tener 6000 flores, pero sólo 20 frutos. A pesar de que sus frutos maduran durante todo el año, normalmente se realizan dos cosechas: la principal (que empieza hacia el final de la estación lluviosa y continúa hasta el inicio de la estación seca) y la intermedia (al principio del siguiente periodo de lluvias), y son necesarios de cinco a seis meses entre su fertilización y su recolección (Jardín Tecina, s.f.).



*Figura 1.* Fruto de cacao

Fuente: extraído de <https://www.eurococoa.com/es/historia-del-cacao-el-cacao-como-materia-prima/>

El fruto del cacao mide entre 10 y 30 centímetros de largo, de ancho puede ser 7 a 9 centímetros, tiene una piel rugosa y coriácea de entre 2 y 3 centímetros de grosor (dependiendo del origen y la variedad del fruto) y está rellena de una pulpa dulce y mucilaginosa, que contiene entre 20 y 50 semillas, bastante blandas y de color entre lavanda pálido y morado oscuro (Lucía Graziani de Fariñas, 2000). La recolección del fruto de cacao no se limita a un periodo del año, sino que suele durar varios meses. De hecho, en muchos países, el cacao puede cosecharse en cualquier época del año. Las vainas cosechadas se abren, normalmente con un machete, para exponer los granos, se retiran la pulpa y las semillas del cacao y se desecha la cáscara. A continuación, la pulpa y las semillas se amontonan, se colocan en cajas de madera o se extienden sobre rejillas durante varios días para fermentarse, luego de aproximadamente una semana se ponen las semillas bajo el sol para el secado final en una semana las semillas estarán casi completamente secas y listas para la venta o usos del cacao (Ecotact, 2022).

## **4.2 Café y granos de café**

El café arábico fue la primera variedad de café en ser cultivada, además de ser la más importante a nivel mundial, ocupando entre el 70 y el 80% de la producción mundial de café. El café de la variedad arábica tiene gran sabor, toques aromáticos y su entrada en el paladar es más suave. Esto se debe a la cantidad de cafeína que contienen, entre un 1 y 1,5%. Los granos de café arábica son grandes, con una forma aplastada y alargada y la línea divisoria del grano forma una característica S (Euskovazza, s.f.).

El café no es el fruto de una sola especie de árbol, de hecho, existen varios tipos de cafés con diferencias notables entre ellos. Se han catalogado más de 100 especies diferentes de cafetos, pero solo dos tipos copan la práctica totalidad de la producción mundial: el café arábica y el robusta. El arábica está considerado el mejor café en grano, ya que aunque tiene menos cuerpo que el robusta, es más equilibrado, aromático y tiene una acidez muy agradable. Además, tiene menos cafeína que otras variedades.

Se encuentra café arábica en casi todos los países productores de café con resultados muy diferentes según la variedad, clima y geografía. El cafeto de arábica crece entre los 500 y los 2400 metros de altura, en zonas húmedas y de clima cálido y la altura en que se cultiva influye en su calidad. El café 100% arábica que es cultivado por encima de los 1.000 (e incluso 2.000 metros de altura) se considera el de mayor calidad (Bonka, s.f.).

Los frutos del cafeto llamados cerezas de café, se cosechan manualmente por los recolectores quienes seleccionan los frutos maduros. Las cerezas se empacan y transportan hasta el recibo y el beneficiadero para su proceso por beneficio húmedo, que consiste en separar del fruto de café, la pulpa (epicarpio) por medio del despulpado, y el mucilago por fermentación natural o con desmucilaginosos mecánicos. Los granos obtenidos se lavan y se secan para su conservación (Cenicafé, s.f.).



*Figura 2.* Imagen de referencia grano de café de tipo arábica y robusta

Fuente: extraído de <https://www.bonka.es/amor-por-el-cafe/mejor-con-cafe-arabica#:~:text=E1%20grano%20de%20ar%C3%A1bica%20puede,S%2C%20su%20caracter%C3%ADstica%20m%C3%A1s%20visible.>

### 4.3 Defectos del café

Un defecto del café es cualquier característica que no cumple con los estándares de control de calidad y puede ser el resultado de las condiciones de la finca, el procesamiento o incluso el tueste. Afortunadamente, los productores pueden eliminar los granos defectuosos clasificándolos. Pero si un defecto pasa desapercibido, se hará evidente durante el tueste y en la bebida.

Existen diferentes tipos de defectos: primarios (los peores) y secundarios (que siguen siendo malos, pero no tanto). Para el café de especialidad “specialty grade”, una muestra de 300 gramos debe tener cero defectos primarios y cinco o menos secundarios. Es de esperar que el café comercial tenga más defectos por muestra: el café de grado de intercambio “Exchange grade”, por ejemplo, puede tener hasta 23 defectos. Para los productores, seguir un riguroso control de calidad para eliminar defectos puede ser costoso; sin embargo, esta es también una de las razones por las cuales el café especial tiene un valor mayor que los granos de calidad comercial.

Uno de los defectos es los granos de café negros y parcialmente negros se verán negros o marrones y marchitos, además, la grieta en el centro será demasiado ancha. A veces es el resultado por la falta de desarrollo, también las enfermedades por hongos y las deficiencias nutricionales pueden causar este defecto. La falta de agua durante el período de crecimiento puede ser otra de las causas, la sobre fermentación o las cerezas demasiado maduras que se recogen del suelo, en lugar del árbol, pueden conducir a granos total o parcialmente negros. (Perfect Daily Grind, 2018).



*Figura 3.* Imagen de muestras de varios defectos de café

Fuente: extraído de <https://perfectdailygrind.com/es/2018/07/19/guia-del-productor-defectos-del-cafe-como-evitarlos/>

Otro defecto es el grano inmaduro con las características del grano color verdoso o gris claro, su cutícula no se desprende y su tamaño es menor que el normal, puede presentar también una deformación cóncava y bordes afilados. Proviene de una recolección de granos verdes o inmaduros y aporta tazas con astringentes con sabor verdoso a hierba o paja (Tueste Café, 2022).



*Figura 4.* Imagen de referencia grano de café con defecto secundario inmaduro o verde  
Fuente: extraído de <https://www.yoamoelcafedecolombia.com/2016/03/16/defectos-del-cafe/>

#### **4.4 Enfermedades en los frutos del cacao**

La enfermedad ocasionada por el hongo *Moniliophthera roleri* es conocida como Moniliasis, el hongo afecta a nivel de frutos, en todos sus estados de desarrollo. Los síntomas son variables de acuerdo a la edad del fruto. Van desde maduración prematura anormal, deformación de frutos y presencia de manchas aceitosas. Conforme la infección avanza va apareciendo, sobre la mancha, un tejido blanco algodonoso, este tejido algodonoso se vuelve gris debido a la aparición de esporas o semillas, que se van a diseminar con el viento. El daño termina con la momificación del fruto. Después de la momificación hay esporulación hasta por 9 meses. Tener en cuenta que existe un periodo de incubación, de 1 mes, antes que aparezcan los primeros síntomas.

Diseminación: a partir de frutos infestados, por medio del viento, durante la etapa de esporulación. Una mayor temperatura, precipitación (cuyo impacto favorece la liberación de esporas) y humedad relativa incrementan la infección. Un exceso de sombreado, alta densidad de plantación sin poda oportuna y falta de control de malezas, crea un medio con mayor humedad, por ende incrementa el ataque del hongo.

Las esporas pueden permanecer viables hasta por nueve meses, tiempo suficiente para conservarse de una campaña a otra. En el tiempo donde no hay fruta permanecen refugiados en ramas, frutos caídos y hojas.

Control: recojo y entierro de frutos afectados (desde los primeros indicios) para reducir al mínimo la fuente de inóculo. La evaluación de la enfermedad es fundamental, al menos una vez por semana. Tener presente que un fruto infectado no retirado del campo puede contagiar a otros 20 en promedio. Luego de la cosecha no incorporar la cáscara directamente al campo pues puede diseminar el mal, de preferencia debe compostarse, pues durante este proceso (fase de calentamiento) se elimina todo resto del hongo presente en los tejidos del fruto. Hacer canales de infiltración para facilitar el drenaje y así evitar encharcamiento, que al evaporarse origina exceso de humedad en el ambiente. Selección de ejemplares tolerantes para su propagación (si se va a trabajar con variedades criollas elegir dentro de la población a los más tolerantes).

Las plantas madres deben tener un registro de trazabilidad, es decir tener un registro de su tolerancia a la enfermedad de por lo menos 3 años. 16 Regular el exceso de sombra mediante poda de los árboles plantados para este fin. Control oportuno de malezas. Manejar la densidad de la plantación mediante podas y empleando un número adecuado de plantas por hectárea (3x3 metros). Control de altura (máximo 3 metros) para facilitar la ubicación de frutos infectados. Control químico con oxiclورو de cobre 1 mes después de la floración principal. No aplicar en floración y cuajado pues puede causar quemaduras. Este producto es de contacto, por lo que hay que asegurar una buena cobertura. Control químico con Triadimefon, Tebuconazol y Prochloraz pueden ayudar a curar y prevenir la enfermedad. Para aplicaciones de productos químicos lo más recomendable es hacerlo cuando el fruto esté entre los 30 y 90 días, etapa de máxima sensibilidad del cultivo a este hongo.

Otra enfermedad de los frutos de cacao se conoce como podrición parda de la mazorca o mazorca negra, el causante de esta enfermedad es el falso hongo *Phytophthora* sp, el hongo ataca sobre todo a nivel de frutos, en cualquier estado que se encuentre éste. 21 El síntoma inicial es una mancha circular color pardo, que poco a poco va agrandándose, pudiendo abarcar todo el fruto. A diferencia del mal ocasionado por la escoba de bruja en este caso la mancha es de

consistencia acuosa. A nivel del tronco produce chancros circulares, que en un estado avanzado exudan un fluido color rojizo, a través de las cuarteaduras de la corteza. Causa problemas a nivel de viveros, causando chupaderas, muerte regresiva de brotes y muerte de hojas nuevas.



*Figura 5.* Fruto de cacao con una mancha blanca indicando etapa de esporulación

Fuente: extraído de <https://www.agrobanco.com.pe/data/uploads/ctecnica/010-e-cacao.pdf>

Diseminación: por salpicadura de la gota de lluvia, que impacta el suelo y levanta tierra con esporas del hongo del suelo a la parte superior. Una vez en la planta el hongo puede ser arrastrado por el agua de lluvia hacia otras partes del árbol y por salpicadura hacia otros árboles. También el viento juega un papel importante para la diseminación. Los insectos también propagan la enfermedad. Por ejemplo, el Monalonia. Le favorece un microclima húmedo, producto de un ambiente emboscado. De una campaña a otra sobreviven en frutos afectados que se encuentran en el árbol.

Control: es necesario identificar la época de mayor concentración de frutos susceptibles (menores de 3 meses) en el año. Una vez identificado hay que hacer un plan de aplicaciones. Para las aplicaciones se utilizan fungicidas hechos a base de cobre como el oxiclورو de cobre o caldo bordalés. Siempre considerar el uso de un adherente agrícola para un mejor humedecimiento a la hora de hacer las aplicaciones. Cuando se tienen chancros se procede a hacer la cirugía, que consiste en ir removiendo la parte podrida hasta encontrar tejido sano, el cual se sella con una pasta hecha con: 1 litro de pintura + 25 gr de oxiclورو de cobre. Los árboles muertos se deben sacar de raíz y desinfectar el suelo. Para esta labor se puede aplicar guano fresco, oxiclورو de cobre, cal agrícola o aliette o metalaxil dirigido al suelo.



*Figura 6.* Frutos de cacao con enfermedad pudrición parda o mazorca negra

Fuente: extraído de <https://www.agrobanco.com.pe/data/uploads/ctecnica/010-e-cacao.pdf>

La enfermedad Mosquilla del cacao es otra enfermedad que afecta los árboles de cacao es causada por el chinche llamado *Monalonion dissimulatum*. En su biología, no se le conoce hospedantes alternantes. Es decir, ataca exclusivamente al cacao. Los insectos en su etapa de juventud se alimentan de los brotes, pero una vez adultos se alimentan de las mazorcas en proceso de maduración. Este insecto posee un estilete en la boca, con lo cual succiona los jugos de la planta, por lo que es llamado comúnmente picador chupador. La oviposición la hacen sobre todo a nivel de brotes tiernos. Un ambiente húmedo permite tener una mayor cantidad de huevos viables. El ambiente es más húmedo si se tienen altos niveles de precipitaciones aunado a un ambiente emboscado.

Los daños se localizan a nivel de brotes y frutos. En los frutos se dan pústulas o heridas circulares en la mitad apical del fruto. Si el daño se da a nivel de frutos tiernos, estos pueden sufrir pudrición y caída prematura. Si el ataque se da en frutos en crecimiento puede haber de formación. Y en casos severos la picadura puede comprometer los granos de la mazorca. A nivel de brotes, se da quemado y marchitamiento. El ataque de esta plaga favorece la contaminación por *Monilia* y *Phytophthora*. Pues las heridas abiertas le sirven como puerta de entrada al hongo.

Para el control de la mosquilla del cacao tiene mayor agresividad si la plantación se desarrolla en ambiente húmedo. Entonces se debe evitar el emboscamiento de la parcela, para disminuir la acumulación de humedad en el ambiente. Las labores de podas y desmalezado son importantes, pues permiten una mayor ventilación e iluminación. También se debe tener en cuenta un adecuado distanciamiento de plantación. Como plaguicidas se puede usar dimetoato, endosulfan o imidacloprid. Evitar aplicar estos productos en la época de floración y cuajado pues se afecta la población del polinizante *Forcipomyia*. Si se tienen algunas flores durante el ataque de este

insecto, aplicar de preferencia en la noche el insecticida metomil. La mosquilla, dentro de un plan de manejo orgánico puede controlarse con insecticidas permitidos para este fin como la rotenona a razón de 300gr. Por cilindro de 200 litros. También puede usarse el sulfato de nicotina a razón de 100 gr. Por cilindro de 200 litros. Como parte del control biológico se está trabajando actualmente con cepas nativas de *Beauveria bassiana*. También dentro del control biológico se está estudiando la crianza y liberación del predator *Podisus* sp. A.



*Figura 7.* Imagen de referencia de la mosquilla del cacao

Fuente: extraído de <https://www.agrobanco.com.pe/data/uploads/ctecnica/010-e-cacao.pdf>

La enfermedad causada por el chinche *Hoplophorion pertusa* es conocida como Asta de torito. Al igual que *Monalonion*, este insecto también es un picador chupador. Los adultos se alimentan de la savia de los brotes y ramas jóvenes, succionando estos jugos con su estilete. Las ramas infestadas presentan puntos y manchas negras. Las hojas que se encuentran sobre ellas se secan y terminan cayéndose causando defoliación. El exceso de sombra en el cacaotal, predispone a éste a un mayor ataque de la plaga.

Control: control químico con productos permitidos para agricultura orgánica como sulfato de nicotina o rotenona. Evitar el emboscamiento de la plantación.



*Figura 8.* Imagen del chinche *Hoplophorion pertusa*

Fuente: extraído de <https://www.agrobanco.com.pe/data/uploads/ctecnica/010-e-cacao.pdf>

La enfermedad causada por el insecto *Selenothrips rubrocinctus* se conoce como Trips del cacao, Biología y daño: Las hembras realizan la oviposición en el envés de las hojas, a partir del cual salen los juveniles, que también se asientan en esta zona. Tanto juveniles como adultos se alimenta del envés de las hojas, la que raspan para luego succionar sus jugos. Si el ataque se da sobre frutos este insecto prefiere alimentarse de aquellos que están cercanos a la madurez. El ataque de trips ocasiona esquelitación de hojas. Ante daños intensos la planta puede presentar muerte regresiva. Amarillamiento y defoliación de hojas atacadas. Control: aplicación de insecticidas como el clorpirifos o el metomil. El manejo de malezas, pues estas son hospedantes alternantes de esta plaga (Agrobanco, s.f.).



*Figura 9.* Imagen de referencia del insecto *Selenothrips rubrocinctus*

Fuente: extraído de <https://www.agrobanco.com.pe/data/uploads/ctecnica/010-e-cacao.pdf>

#### **4.5 Antecedentes en estudios de detección de enfermedades en plantas utilizando procesamiento de imágenes y visión computacional**

Detección de cenicilla en rosa usando procesamiento de imágenes por computadora: en esta investigación se desarrolló un sistema de detección de la cenicilla (*Sphaerotheca pannosa*) en rosa con Open CV. Open CV es una librería de fuente abierta de visión por computadora, la cual está escrita en lenguaje C y C++. La detección fue hecha de acuerdo con el espacio de color HSV. La imagen fuente fue convertida del espacio de color RGB al espacio HSV y las partes de la enfermedad y la planta fueron extraídas de acuerdo con la información de H, S y V. Posteriormente se removieron los objetos blancos (ruido). Finalmente, la precisión de la

detección fue evaluada. El sistema de detección desarrollado es capaz de detectar la cenicilla a través del espacio de color HSV con Open CV. Mejores diagnósticos fueron obtenidos usando fotografías cercanas (10 cm).

De acuerdo con el desarrollo de este proyecto para la detección de cenicilla en rosa se descubrió que el algoritmo logra la detección de la enfermedad con más precisión cuando las fotografías que muestran la enfermedad son tomadas cerca por lo que para lograr más precisión en la detección de las 2 enfermedades de los frutos de cacao las imágenes se tomaran con una distancia más cercana a los frutos (Noé Velázquez-López, 2009).



FIGURA 1. Signos y síntomas de cenicilla en hojas de rosa.  
FIGURE 1. Signs and symptoms of powdery mildew on leaves of roses.

*Figura 10.* Imagen de referencia de Cenicilla

Fuente: extraído de [https://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S1027-152X2011000200008](https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1027-152X2011000200008)

Machine learning en la detección de enfermedades en plantas: este artículo, busca mostrar la relevancia a nivel tecnológico del uso del machine learning en la identificación de enfermedades en plantas. Mediante la revisión literaria de varios trabajos y las diversas técnicas de reconocimiento, que permitirán identificar y diagnosticar las enfermedades en las plantas y cultivos, lo que permitirá tomar decisiones y acciones para evitar la proliferación de la enfermedad. Las conclusiones que proporciona este artículo son que sin lugar a dudas el machine learning a través de la inteligencia artificial ha tomado una gran relevancia en varios campos de la ciencia gracias al reconocimiento de imágenes que permiten dar un diagnóstico en un porcentaje muy elevado puntualmente en enfermedades de plantas. Al mismo tiempo esta herramienta a su vez proporciona métodos más rápidos de diagnóstico gracias a que las redes neuronales que se entrenan con un gran conjunto y variedad de datos permite casi al instante hacer un análisis y clasificación de la imagen que son muy difíciles de manejar para los seres

humanos inclusive para especialistas en cada uno de los sectores en donde se utilice el reconocimiento de imágenes con machine learning (Andrés Calderón Romero, 2019).

El proyecto de grado sistema de visión por computador para detectar hierba no deseada en prototipo de cultivo de frijol usando ambiente controlado, se fundamenta en la búsqueda de herramientas tecnológicas que mejoren el trabajo de los campesinos es algo necesario, ya que se ven afectados por hierbas no deseadas, las cuales afectan la calidad y retrasan la producción del cultivo. Los campesinos necesitan nuevas tecnologías que les garanticen acelerar y mejorar la producción de sus cultivos. Esto es lo que se busca lograr en este proyecto, proponiendo la visión artificial como una herramienta útil que a futuro mejorará la competencia del agro colombiano frente a otras entidades agrarias internacionales y las conclusiones que deja este trabajo de grado es que logró implementar un algoritmo con una efectividad del 92.5% en detección de malezas que rodean una planta de frijol mediante el procesamiento de imágenes y minería de datos, el programa realizado permite la detección de elementos bajo una fase de entrenamiento previa basada en sus características principales, siguiendo el estándar general de un sistema de visión artificial el cual está compuesto de: adquisición de imágenes, preprocesamiento, segmentación, descripción y reconocimiento de patrones (Gualteros, 2014).

La tesis Detección de enfermedades en plantas de crisantemo, por medio de visión artificial aplicada a imágenes multiespectrales se fundamenta en que el sector floricultor en Colombia representa un mercado significativo, siendo el principal exportador de flores hacia Estados Unidos, donde alrededor del 80% de las flores importadas son colombianas. Para asegurar la estabilidad y el crecimiento de esta industria, es crucial implementar rigurosos sistemas de control fitosanitarios, que eviten la dispersión de plagas y cumplan con estándares de calidad internacional. Dada la falta de investigaciones locales y la necesidad de integrar nuevas tecnologías en la floricultura en Antioquia, este trabajo se enfoca en optimizar un sistema de visión artificial para identificar plagas y enfermedades en hojas de crisantemo.

Mediante el desarrollo de esta tesis se optimizaron parámetros del equipo de captura de imágenes multiespectrales, como tiempos de adquisición y control del ruido, seleccionando canales espectrales basados en áreas de interés fitosanitario. Se analizaron muestras de hojas de

crisantemo con síntomas de enfermedades como trips, el virus del marchitamiento moteado del tomate (TSWV) y ácaros (*Tetranychus* sp). Tras calibrar el sistema de captura, se generaron imágenes multiespectrales y se desarrolló una metodología que combina procesamiento de imágenes, análisis multiespectral y aprendizaje automático para identificar muestras sanas o afectadas por estas enfermedades.

Las conclusiones que deja esta tesis son que los sistemas de procesamiento óptico son herramientas valiosas para abordar problemáticas ingenieriles, y en este trabajo se utilizan dos métodos distintos para capturar información óptica con el fin de mejorar el control fitosanitario en la industria floricultora local.

En primer lugar, se aplicó la espectroscopía para analizar el comportamiento espectral de enfermedades en hojas de crisantemo, como ácaros, trips y virus. Esta técnica permitió identificar problemas en la industria y caracterizar algunas enfermedades, ofreciendo alternativas a los métodos convencionales de control de plagas. A través de la espectroscopía, se determinaron regiones espectrales asociadas con cambios biológicos en las hojas, aunque no fue eficiente como un diagnóstico definitivo debido a la posibilidad de confundir tejidos sanos con infectados en etapas tempranas.

Para abordar estas limitaciones, se mejoró un sistema de captura de imágenes multiespectrales, que proporciona información tanto espectral como espacial. Se desarrolló desde cero una cámara monocromática y una interfaz de control, facilitando la captura de imágenes representativas. Esto, en colaboración con el trabajo de un ingeniero agrónomo, permitió desarrollar una metodología para identificar enfermedades, utilizando índices vegetales que ayudaron a diferenciar regiones con síntomas.

Se encontró que la identificación de ácaros era particularmente desafiante debido a su leve sintomatología, lo que requiere la intervención de expertos en campo para una detección temprana. Sin embargo, se logró clasificar tejidos sanos e infectados en el caso de virus y trips mediante algoritmos de clasificación, aunque se necesitó evaluarlas por separado debido a sus

manifestaciones distintas. La efectividad de la investigación fue validada por un experto en la materia.

En conclusión, combinar información espacial de imágenes con datos espectrales de cambios biológicos en las hojas de crisantemo puede ser una herramienta útil para mejorar los procesos de control de plagas, ofreciendo una alternativa a los prolongados tiempos de capacitación de revisores expertos.

Finalmente, este trabajo representa un primer acercamiento académico a la industria floricultora en un entorno de laboratorio controlado. Se subraya la necesidad de realizar investigaciones futuras para llevar estas herramientas a un contexto de campo abierto, lo que permitiría mejorar significativamente la eficiencia en los procesos de control fitosanitario en los cultivos. Para lograr esto, sería necesario desarrollar un sistema más compacto que sea fácil y práctico de implementar en el campo, superando las dificultades asociadas a los invernaderos y las variaciones geológicas en su ubicación (Rios).

El artículo *Classification of Cocoa Beans by Analyzing Spectral Measurements Using Machine Learning and Genetic Algorithm* examinó el rendimiento de diferentes algoritmos de clasificación en datos de medición espectral utilizando un algoritmo genético para la optimización de parámetros para los granos de frutos de plantas de cacao. Los resultados revelan variaciones significativas en la precisión del algoritmo entre diferentes generaciones, lo que destaca la importancia de la optimización de parámetros para lograr el mejor rendimiento. Los algoritmos de regresión logística, SVM y bosque aleatorio demostraron estabilidad del rendimiento, SVM mantiene una exactitud del 73,91% y una precisión del 75,36%. El F-score alcanzó el 74,38%, el recall se mantuvo en el 73,91% y el MCC se situó en el 60,85%. Aunque este rendimiento ha disminuido ligeramente en comparación con la generación anterior, sigue siendo sólido. Decision Tree mantiene una exactitud del 65,21%, con una precisión del 65,28%. El F-score se sitúa en el 65,09%, el recall alcanza el 65,21% y el MCC se sitúa en el 47,86%. El rendimiento se mantiene similar a la generación anterior. Random forest también presenta un rendimiento estable, con una exactitud del 73,91%, una precisión del 74,12%, un F-score del 73,82%, un recall del 73,91% y un MCC del 60,96%. XGBoost alcanza una precisión del

69,56%, con una precisión del 70,14%. El F-score alcanza el 69,15%, el recall se mantiene en el 69,56% y el MCC es del 54,88%. Aunque el rendimiento ha disminuido ligeramente, XGBoost sigue siendo una opción competente. KNeighbors mantiene un rendimiento similar a la generación anterior, con una precisión del 65,21%, una precisión del 67,14%, un F-score del 65,94%, un recall del 65,21% y un MCC del 47,71%. La regresión logística mantiene un rendimiento estable con una precisión del 69,56%, una precisión del 71,49%, un F-score del 70,29%, un recall del 69,56% y un MCC del 54,28% (Kacoutchy Jean Ayikpa, 2023).

## 5. Metodología

### 5.1 Tipo de proyecto

El proyecto se clasifica como exploratorio y descriptivo. Su enfoque exploratorio se centra en investigar nuevas aplicaciones de tecnologías avanzadas, específicamente en el procesamiento de imágenes y el aprendizaje automático (machine learning), dentro del ámbito de la agricultura. Este enfoque permite identificar y analizar innovaciones que pueden optimizar los procesos agrícolas y mejorar la productividad.

Además, el proyecto tiene un componente descriptivo que se enfoca en la evaluación del estado de salud de los frutos de los árboles de cacao y granos de café. Esto implica no solo la recolección de datos sobre las condiciones de los frutos o granos, sino también la interpretación de estos datos para determinar la presencia de enfermedades o plagas, así como la identificación de frutos o granos saludables. Al integrar ambos enfoques, el proyecto busca proporcionar un panorama integral sobre la salud de los cultivos de cacao y contribuir al desarrollo de estrategias de manejo más efectivas basadas en la tecnología.

### 5.2 Método

El método a seguir en este proyecto se basa en una combinación de enfoques inductivos y deductivos, con el objetivo de desarrollar y validar una aplicación móvil para la identificación de enfermedades en frutos de cacao mediante algoritmos de machine learning. A continuación, se describen los procedimientos específicos que se implementaron.

Se llevará a cabo la recolección de datos, que consiste en realizar una captura sistemática de imágenes de frutos de cacao utilizando cámaras de alta resolución. Se asegurarán condiciones de iluminación adecuadas para garantizar la calidad de las imágenes. Se recolectarán un mínimo de 600 imágenes, distribuidas en tres categorías: 200 imágenes de frutos con moniliasis, 200 de frutos con mazorca negra y 200 de frutos saludables. Además, las imágenes también pueden extraerse de bases de datos públicas disponibles en Internet.

El análisis de los datos incluirá el procesamiento de imágenes, donde las imágenes recolectadas serán procesadas para mejorar su calidad y extraer características relevantes que ayudarán en la identificación de patrones asociados con las enfermedades. Este proceso incluirá técnicas de segmentación y filtrado. A partir de estas observaciones, se desarrollarán generalizaciones y modelos predictivos utilizando un enfoque inductivo.

En cuanto al entrenamiento de algoritmos, se aplicará machine learning para entrenar dos algoritmos de clasificación utilizando el conjunto de datos especializado. Se implementarán técnicas de validación cruzada para evaluar el desempeño de los modelos y ajustar sus parámetros de manera efectiva.

La evaluación y validación de los modelos se llevará a cabo mediante simulaciones que utilizarán datos reales para validar el desempeño de los algoritmos clasificadores. Se evaluará la precisión y la capacidad de los modelos para diferenciar entre frutos saludables y aquellos con enfermedades.

Además, se diseñará y desarrollará una aplicación móvil para el sistema Android que permitirá a los usuarios utilizar imágenes propias de frutos de cacao. Esta aplicación integrará los algoritmos de machine learning para clasificar el estado de los frutos, informando al usuario si el fruto es saludable, si presenta moniliasis o si tiene la enfermedad de mazorca negra. El desarrollo de la aplicación incluirá la creación de una interfaz amigable, el procesamiento de imágenes y la integración de las herramientas de machine learning, garantizando una experiencia de usuario eficiente y efectiva.

Finalmente, se sintetizarán los resultados sobre la eficacia de los algoritmos en la implementación del procesamiento de imágenes y el reconocimiento de enfermedades en los frutos de cultivos de cacao, generando alertas a los usuarios de la aplicación que puedan contribuir a mejorar los procesos de control fitosanitario en esta industria.

### 5.3 Instrumentos de recolección de información

Para la recolección de información en este proyecto, se utilizará el análisis documental como principal instrumento. Esta técnica permitirá examinar una variedad de documentos relevantes para entender en profundidad las características visuales de las enfermedades en los frutos de cacao (moniliasis y mazorca negra) y en los granos de café (defectos verdes o negros), así como las metodologías utilizadas en otros estudios de clasificación de imágenes mediante algoritmos de machine learning.

El análisis documental se enfocará en revisar estudios previos, informes técnicos, bases de datos públicas de imágenes, y publicaciones científicas. También se consultarán documentaciones de librerías de Python especializadas en procesamiento de imágenes y entrenamiento de clasificadores (como OpenCV, scikit-image y TensorFlow), lo cual brindará una base sólida para el desarrollo de los modelos clasificadores de machine learning.

**5.3.1 Fuentes primarias.** Las fuentes primarias incluyen documentos y materiales originales que contienen información directa y sin intermediarios sobre las características de las enfermedades en frutos de cacao, defectos en granos de café y técnicas de machine learning.

**Artículos científicos y técnicos:** publicaciones que presenten estudios originales sobre la identificación y clasificación de enfermedades en plantas y defectos en granos de café mediante machine learning, los cuales proporcionarán un marco teórico y metodológico para el desarrollo del modelo.

**Informes de investigación:** documentos elaborados por instituciones académicas o gubernamentales sobre enfermedades del cacao, defectos en café y su impacto en la agricultura.

**Imágenes originales:** un conjunto de datos que incluye al menos 600 imágenes de frutos de cacao en distintas condiciones (saludable, con moniliasis y con mazorca negra) y otro de granos de café con características de defectos verdes o negros. Algunos de estos datos provendrán de bases de datos públicas y serán empleados específicamente para entrenar y evaluar el modelo.

**5.3.2 Fuentes secundarias.** Libros y artículos de revisión: Fuentes que examinan y resumen avances en machine learning aplicado a la agricultura, así como en técnicas avanzadas de procesamiento de imágenes.

Enciclopedias, manuales técnicos y documentaciones de librerías de Python: Publicaciones que proporcionan definiciones, conceptos y contexto general sobre machine learning, redes neuronales, y procesamiento de imágenes mediante Python (librerías como OpenCV y scikit-image).

Libros y artículos de investigación: fuentes que examinan y resumen avances en machine learning aplicado a la agricultura, así como en técnicas avanzadas de procesamiento de imágenes.

## 6. Resultados del proyecto

A continuación, se describen los procedimientos, técnicas, actividades y demás estrategias metodológicas utilizadas durante la investigación e igualmente los resultados correspondientes a la evaluación de técnicas de clasificación para el reconocimiento de enfermedades en frutos de cacao.

### 6.1 Creación de un banco de imágenes de frutos de árboles de cacao y granos de café

Inicialmente se requiere la creación de un dataset de mínimo 600 imágenes de frutos de cacao con la manifestación de síntomas de las enfermedades de interés. Para la toma de imágenes se utilizó frutos de cultivos de la variedad o clon del cacao CCN51, las plantas de cacao están ubicadas en la vereda las Camelias, Mariquita, Tolima. Los cultivos del cacao están a una altura aproximada de 700 metros sobre el nivel del mar, las condiciones climáticas son tropicales con una humedad promedio de 80% una temperatura mínima promedio de 18 grados centígrados y una temperatura máxima promedio de 28 grados centígrados. Las imágenes se adquirieron con un celular Infinix Note 30 Pro cuenta con una cámara triple compuesta por un sensor principal Samsung HM6 de 108 MP (megapíxeles) con apertura f/1.75, un segundo sensor de 2.0 MP (megapíxeles) con apertura f/2.4 para profundidad en fotografías de retrato, y un tercer sensor de 0.3 MP (megapíxeles). Las fotografías se toman en condiciones de luz natural para asegurar una iluminación precisa y mantener la fidelidad de los colores. Estas imágenes se capturan en días soleados, entre las 13:00 y las 16:00, cuando la luz es más estable y uniforme.

Tabla 1.

*Información de recolección de imágenes de frutos del cacao CCN51*

<b>Fecha</b>	<b>Cantidad</b>	<b>Tipo</b>
9/25/2024	209	Frutos del cacao CCN51 saludable
9/26/2024	245	Frutos del cacao CCN51 con enfermedad monilia
9/27/2024	233	Frutos del cacao CCN51 con enfermedad mazorca negra

Fuente: diseño propio



*Figura 11.* Collage de imágenes utilizadas para cacao CCN51 saludable en el dataset  
Fuente: fotografías propias



*Figura 12.* Collage de imágenes utilizadas para cacao CCN51 con enfermedad mazorca negra en el dataset  
Fuente: fotografías propias

Para el dataset de imágenes sobre granos de café se utilizó un dataset publico extraído de la web Kaggle la descripción de la toma de las fotografías se realizó con las siguientes características: Las fotos de los granos de café se capturan con un iPhone 12 Mini, equipado con una cámara trasera de 12 megapíxeles, Ultra Ancha y Ancha. La cámara se coloca en una posición paralela al plano del objeto al capturar las fotografías. Las imágenes de los granos de café tostado se obtienen en una variedad de configuraciones para validar una amplia gama de

entradas de imágenes de granos de café tostado. Este experimento utiliza tanto luz LED de una caja de luz como luz natural para capturar el conjunto de datos; luego, el ruido de la imagen se incrementa colocando cada variedad de granos de café en un contenedor. Las imágenes se recopilan automáticamente y se guardan en formato PNG (Ontoum, 2022).



*Figura 13.* Collage de imágenes utilizadas para cacao CCN51 con enfermedad monilia en el dataset

Fuente: fotografías propias

En total se encuentran 900 fotografías en el dataset de granos de café, 300 para la característica de grano de café saludable, 300 fotografías para la característica de grano de café con defecto negro y 300 fotografías para la característica de grano de café con defecto verde.

Ahora con los dataset de imágenes para los frutos de cacao y los granos de café se agrupa todas las imágenes según las características en carpetas diferentes, cada carpeta tiene las imágenes que coincida al mismo tipo de características como se observa en la tabla 2



*Figura 14.* Collage de imágenes utilizadas en el dataset para grano de café saludable  
Fuente: extraído de <https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224/data>



*Figura 15.* Collage de imágenes utilizadas en el dataset para grano de café con defecto negro  
Fuente: extraído de <https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224/data>



Figura 16. Collage de imágenes utilizadas en el dataset para grano de café con defecto verde  
 Fuente: extraído de <https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224/data>

Tabla 2.

*Agrupación de imágenes de frutos del cacao y granos de café*

<b>Carpeta</b>	<b>Cantidad</b>	<b>Tipo</b>
0 Cacao saludable	209	Frutos del cacao CCN51 saludable
1 Mazorca negra	245	Frutos del cacao CCN51 con enfermedad monilia
2 Moniliasis	233	Frutos del cacao CCN51 con enfermedad mazorca negra
0 granos de cafe normal	300	Granos de café saludables
1 granos de cafe verde	300	Granos de café con defecto verde
2 granos de cafe negros	300	Granos de café con defecto negro

Fuente: diseño propio

## 6.2 Entrenamiento del modelo de machine learning

A continuación, se desarrolla el algoritmo para el entrenamiento de los modelos de clasificación, utilizando el dataset de imágenes, el lenguaje de programación es Python versión 3.13, como editor de código Google colab y visual studio code, la descripción del código se encuentra en pseudocódigo.

De acuerdo al artículo *Classification of Cocoa Beans by Analyzing Spectral Measurements Using Machine Learning and Genetic Algorithm* aquí se concluyó que los modelos de clasificación son los más estables comparados con otros modelos para la clasificación de granos de frutos de cacao (Kacoutchy Jean Ayikpa, 2023). por esta razón el enfoque para el desarrollo de los modelos será con modelos de tipo clasificador, como KNN, random forest o redes neuronales.

Bloque 1: se importa las bibliotecas necesarias para el procesamiento de imágenes, manejo de datos, y modelos de machine learning.

## INICIO

```
# Importar librerías necesarias
IMPORTAR matplotlib.pyplot COMO plt
IMPORTAR cv2
IMPORTAR output DESDE google.colab
IMPORTAR numpy COMO np
IMPORTAR os
IMPORTAR time
IMPORTAR graycomatrix COMO Co_ocurrencia DESDE skimage.feature
IMPORTAR graycoprops DESDE skimage.feature

# Definir la ruta para acceder a las imágenes del conjunto de datos
DEFINIR Ruta COMO "/content/drive/MyDrive/semestre 10/Trabajo de grado/dataset3"

# Crear listas vacías para almacenar imágenes y etiquetas
CREAR LISTA imagenes
CREAR LISTA etiquetas

# Crear listas para almacenar imágenes en formato BGR y sus etiquetas
CREAR LISTA imagenesBGR
```

CREAR LISTA etiquetasBGR

FIN

Bloque 2: se carga las imágenes desde una carpeta específica, utilizando os para acceder a los archivos y cv2 para leer cada imagen en el formato adecuado. Se crea una lista para almacenar las imágenes y las respectivas etiquetas.

INICIO

# Ordenar las carpetas en la ruta especificada para obtener las clases de imágenes

CLASES = LISTAR Y ORDENAR elementos en el directorio Ruta

# Para cada clase de imagen (etiqueta) en CLASES

PARA i, label EN CLASES HACER

carpeta\_clase = RUTA DE la clase (Ruta, label)

# Para cada archivo de imagen en la carpeta de la clase

PARA archivo EN LISTAR ARCHIVOS DE carpeta\_clase HACER

Ruta\_img = RUTA DEL ARCHIVO DE IMAGEN (carpeta\_clase, archivo)

LIMPIAR la salida de la consola

# Leer la imagen en dos formatos

Img = LEER IMAGEN DESDE Ruta\_img

ImgBGRformatrix = LEER IMAGEN DESDE Ruta\_img

baseImgHLS = LEER IMAGEN DESDE Ruta\_img

# Convertir la imagen al espacio de color HLS

img\_HLS = CONVERTIR baseImgHLS A HLS

# Dividir la imagen en canales H, L y S

```

hue, lightness, saturation = DIVIDIR img_HLS EN CANALES

# Aumentar la luminosidad del canal L para hacer la imagen más brillante
lightness = INCREMENTAR lightness EN 1

# Combinar los canales H, L y S en una imagen brillante
brighter_img_HLS = COMBINAR hue, lightness, saturation

# Convertir la imagen brillante de HLS a BGR
brighter_img_BGR = CONVERTIR brighter_img_HLS A BGR

# Redimensionar la imagen para mostrarla
Bin = REDIMENSIONAR brighter_img_BGR A (100,100)

# Crear bordes en la imagen aplicando umbral binario
APLICAR UMBRAL a Bin CON VALOR 100 Y MAXIMO 255

# Dividir la imagen binarizada en canales H, L y S
hue, lightness, saturation = DIVIDIR Bin EN CANALES

# Convertir la imagen a escala de grises
Bin = CONVERTIR Bin A ESCALA DE GRISES

# Agregar la imagen binaria y la etiqueta a las listas correspondientes
AGREGAR Bin A imagenes COMO MATRIZ
AGREGAR i A etiquetas

# Agregar la imagen en formato BGR y su etiqueta a las listas correspondientes
AGREGAR ImgBGRformatrix A imagenesBGR COMO MATRIZ
AGREGAR i A etiquetasBGR

```

FIN PARA

FIN PARA

FIN

Bloque 3: se procesan las imágenes cargadas, ajustando su tamaño y aplicando técnicas de preprocesamiento como la conversión a escala de grises, normalización, o extracción de características específicas.

INICIO

```
# Importar librerías necesarias para medir características y visualización
IMPORTAR time PARA medir tiempos de ejecución
IMPORTAR matplotlib.pyplot COMO plt PARA visualización
IMPORTAR cv2 PARA procesamiento de imágenes
IMPORTAR output DESDE google.colab PARA manipulación de salida
IMPORTAR numpy COMO np PARA operaciones numéricas

# Iniciar el proceso de medición de características usando momentos de Hu
# Recorriendo el directorio para medir características de área
```

FIN

Bloque 4: Se organiza el conjunto de datos en variables (X para las características y Y para las etiquetas) para facilitar su uso en el entrenamiento de los modelos de machine learning. Se extraen 78 características por cada imagen y se almacenan en la matriz X.

INICIO

```
# Inicializar los vectores de características y etiquetas
```

```

X = np.zeros((número_de_imagenes, 79)) # Vector de características geométricas
Y = np.zeros((número_de_imagenes, 1)) # Vector de etiquetas

# Recorrer todas las imágenes
PARA k EN rango(len(imagenes)) HACER
    output.clear()

    # Asignar la etiqueta correspondiente
    Y[k] = etiquetas[k]
    Bin = imagenes[k]
    bindefaultimage = imagenesBGR[k]
    bindefaultimage_rgb = cv2.cvtColor(bindefaultimage, cv2.COLOR_BGR2RGB)

    # Conversión a diversos canales de color para extraer características
    baseImgLuv = bindefaultimage_rgb
    baseImgBGR = bindefaultimage_rgb
    baseImggray = bindefaultimage_rgb
    baseImgLAB = bindefaultimage_rgb
    baseImg_YCbCr = bindefaultimage_rgb
    baseImgHLS = bindefaultimage_rgb

    # Procesando la imagen

    # ---- Luv Channels ----
    img_Luv = cv2.cvtColor(baseImgLuv, cv2.COLOR_BGR2Luv)
    L_Luv, u, v = cv2.split(img_Luv)

    # ---- BGR channels ----
    blue, green, red = cv2.split(baseImgBGR)

    # ---- GRAY ----

```

```

img_gray = cv2.cvtColor(baseImggray, cv2.COLOR_BGR2GRAY)

# ---- LAB ----
img_LAB = cv2.cvtColor(baseImgLAB, cv2.COLOR_BGR2LAB)
L_channel, a_channel, b_channel = cv2.split(img_LAB)

# ---- img_YCbCr ----
img_YCbCr = cv2.cvtColor(baseImg_YCbCr, cv2.COLOR_BGR2YCrCb)
Y_YCbCr, Cr, Cb = cv2.split(img_YCbCr)

# ---- HLS ----
img_HLS = cv2.cvtColor(baseImgHLS, cv2.COLOR_BGR2HLS)
hue, lightness, saturation = cv2.split(img_HLS)
lightness = cv2.add(lightness, 1) # Aumentar brillo
brighter_img_BGR = cv2.cvtColor(brighter_img_HLS, cv2.COLOR_HLS2BGR)

# Encontrar los contornos de la imagen
contours, _ = cv2.findContours(np.uint8(Bin), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
glcm = Co_ocurrencia(np.uint8(Bin), [1, 2, 3], [0, np.pi/20, np.pi/2, 200*np.pi/4],
levels=256, symmetric=True, normed=True)

SI (len(contours) > 0) HACER
# Tomar solo el contorno más grande
largest_contour = max(contours, key=cv2.contourArea)

# Calcular área y otras características
area = cv2.contourArea(largest_contour)
contrast = graycoprops(glcm, 'contrast')
contrast = np.mean(contrast)
Homogeneidad = graycoprops(glcm, 'homogeneity')

```

```
ASM = graycoprops(glcm, 'ASM')
M = cv2.moments(largest_contour)
Homogeneidad = np.mean(Homogeneidad)
ASM = np.mean(ASM)
Energy = graycoprops(glcm, 'energy')
Energy = np.mean(Energy)
Dissimilaridad = graycoprops(glcm, 'dissimilarity')
Dissimilaridad = np.mean(Dissimilaridad)

# Calcular momentos de Hu
Hu_moments = np.transpose(cv2.HuMoments(M)) # Siete valores

# Agregar alto y ancho
P1, P2, rugosity, Alto = cv2.boundingRect(np.uint8(Bin))
rugosity = np.std(glcm)

# Extraer características del canal de saturación
saturation_mean = np.mean(saturation)
saturation_std = np.std(saturation)
saturation_25th = np.percentile(saturation, 25)
saturation_75th = np.percentile(saturation, 75)

# Extraer características de HLS
hue_mean = np.mean(hue)
hue_std = np.std(hue)
hue_25th = np.percentile(hue, 25)
hue_75th = np.percentile(hue, 75)
lightness_mean = np.mean(lightness)
lightness_std = np.std(lightness)
lightness_25th = np.percentile(lightness, 25)
lightness_75th = np.percentile(lightness, 75)
```

```
# Extraer características del canal YCbCr
Y_mean = np.mean(Y_YCbCr)
Y_std = np.std(Y_YCbCr)
Y_25th = np.percentile(Y_YCbCr, 25)
Y_75th = np.percentile(Y_YCbCr, 75)
Cb_mean = np.mean(Cb)
Cb_std = np.std(Cb)
Cb_25th = np.percentile(Cb, 25)
Cb_75th = np.percentile(Cb, 75)
Cr_mean = np.mean(Cr)
Cr_std = np.std(Cr)
Cr_25th = np.percentile(Cr, 25)
Cr_75th = np.percentile(Cr, 75)

# Extraer características de LAB
L_mean = np.mean(L_channel)
L_std = np.std(L_channel)
L_25th = np.percentile(L_channel, 25)
L_75th = np.percentile(L_channel, 75)
a_mean = np.mean(a_channel)
a_std = np.std(a_channel)
a_25th = np.percentile(a_channel, 25)
a_75th = np.percentile(a_channel, 75)
b_mean = np.mean(b_channel)
b_std = np.std(b_channel)
b_25th = np.percentile(b_channel, 25)
b_75th = np.percentile(b_channel, 75)

# Extraer características de la imagen en escala de grises
gray_mean = np.mean(img_gray)
gray_std = np.std(img_gray)
```

```
gray_25th = np.percentile(img_gray, 25)
gray_75th = np.percentile(img_gray, 75)

# Extraer características de los canales BGR
blue_mean = np.mean(blue)
blue_std = np.std(blue)
blue_25th = np.percentile(blue, 25)
blue_75th = np.percentile(blue, 75)
green_mean = np.mean(green)
green_std = np.std(green)
green_25th = np.percentile(green, 25)
green_75th = np.percentile(green, 75)
red_mean = np.mean(red)
red_std = np.std(red)
red_25th = np.percentile(red, 25)
red_75th = np.percentile(red, 75)

# Extraer características de LUV
L_Luv_mean = np.mean(L_Luv)
L_Luv_std = np.std(L_Luv)
L_Luv_25th = np.percentile(L_Luv, 25)
L_Luv_75th = np.percentile(L_Luv, 75)
u_mean = np.mean(u)
u_std = np.std(u)
u_25th = np.percentile(u, 25)
u_75th = np.percentile(u, 75)
v_mean = np.mean(v)
v_std = np.std(v)
v_25th = np.percentile(v, 25)
v_75th = np.percentile(v, 75)
```

```
# Almacenar resultados en la matriz de características
```

```
X[k, 0] = area
```

```
X[k, 1] = contrast
```

```
X[k, 2] = Dissimilaridad
```

```
X[k, 3] = Homogeneidad
```

```
X[k, 4] = ASM
```

```
X[k, 5] = Energy
```

```
X[k, 6:13] = Hu_moments
```

```
X[k, 13] = Alto
```

```
X[k, 14] = rugosity
```

```
X[k, 15] = saturation_mean
```

```
X[k, 16] = saturation_std
```

```
X[k, 17] = saturation_25th
```

```
X[k, 18] = saturation_75th
```

```
X[k, 19] = hue_mean
```

```
X[k, 20] = hue_std
```

```
X[k, 21] = hue_25th
```

```
X[k, 22] = hue_75th
```

```
X[k, 23] = lightness_mean
```

```
X[k, 24] = lightness_std
```

```
X[k, 25] = lightness_25th
```

```
X[k, 26] = lightness_75th
```

```
X[k, 27] = Y_mean
```

```
X[k, 28] = Y_std
```

```
X[k, 29] = Y_25th
```

```
X[k, 30] = Y_75th
```

```
X[k, 31] = Cb_mean
```

```
X[k, 32] = Cb_std
```

```
X[k, 33] = Cb_25th
```

```
X[k, 34] = Cb_75th
```

```
X[k, 35] = Cr_mean
```

X[k, 36] = Cr\_std  
X[k, 37] = Cr\_25th  
X[k, 38] = Cr\_75th  
X[k, 39] = L\_mean  
X[k, 40] = L\_std  
X[k, 41] = L\_25th  
X[k, 42] = L\_75th  
X[k, 43] = a\_mean  
X[k, 44] = a\_std  
X[k, 45] = a\_25th  
X[k, 46] = a\_75th  
X[k, 47] = b\_mean  
X[k, 48] = b\_std  
X[k, 49] = b\_25th  
X[k, 50] = b\_75th  
X[k, 51] = gray\_mean  
X[k, 52] = gray\_std  
X[k, 53] = gray\_25th  
X[k, 54] = gray\_75th  
X[k, 55] = blue\_mean  
X[k, 56] = blue\_std  
X[k, 57] = blue\_25th  
X[k, 58] = blue\_75th  
X[k, 59] = green\_mean  
X[k, 60] = green\_std  
X[k, 61] = green\_25th  
X[k, 62] = green\_75th  
X[k, 63] = red\_mean  
X[k, 64] = red\_std  
X[k, 65] = red\_25th  
X[k, 66] = red\_75th

```
X[k, 67] = L_Luv_mean
X[k, 68] = L_Luv_std
X[k, 69] = L_Luv_25th
X[k, 70] = L_Luv_75th
X[k, 71] = u_mean
X[k, 72] = u_std
X[k, 73] = u_25th
X[k, 74] = u_75th
X[k, 75] = v_mean
X[k, 76] = v_std
X[k, 77] = v_25th
X[k, 78] = v_75th
```

FIN SI

FIN PARA

FIN

Bloque 5: se utiliza StratifiedShuffleSplit para dividir el conjunto de datos en entrenamiento y prueba, asegurando que la proporción de clases se mantenga en ambas particiones.

INICIO

```
# Importar el método StratifiedShuffleSplit de sklearn
IMPORTAR StratifiedShuffleSplit DE sklearn.model_selection

# Inicializar StratifiedShuffleSplit
Indices = StratifiedShuffleSplit(n_splits=1, test_size=0.12, random_state=7)

# Dividir los datos en conjuntos de entrenamiento y prueba
```

PARA train\_index, test\_index EN Indices.split(X, Y) HACER

```
X_train = X[train_index, :] # Datos de entrenamiento
```

```
X_test = X[test_index, :] # Datos de prueba
```

```
Y_train = Y[train_index] # Etiquetas de entrenamiento
```

```
Y_test = Y[test_index] # Etiquetas de prueba
```

FIN PARA

FIN

Bloque 6: se dibujan histogramas que representan la distribución de clases en el conjunto de entrenamiento y el conjunto de prueba, ayudando a visualizar el número de imágenes por clase en cada conjunto para verificar que la división sea proporcional.

INICIO

```
# Dibujar el histograma del vector de entrenamiento
```

```
hist_train, bins = np.histogram(Y_train, bins=[0, 1, 2, 3, 4]) # Calcular el histograma
```

```
plt.figure(figsize=(8, 6)) # Crear una nueva figura
```

```
plt.bar(bins[:-1], hist_train, width=0.5, color='skyblue', edgecolor='black') # Dibujar el
```

histograma

```
plt.xlabel('Clase') # Etiqueta del eje X
```

```
plt.ylabel('Número de muestras') # Etiqueta del eje Y
```

```
plt.title('Histograma de clases en el conjunto de entrenamiento') # Título del gráfico
```

```
plt.xticks(bins[:-1]) # Marcas en el eje X
```

```
plt.grid(True) # Activar la cuadrícula
```

```
plt.show() # Mostrar el gráfico
```

```
# Dibujar el histograma del vector de prueba
```

```
hist_test, bins = np.histogram(Y_test, bins=[0, 1, 2, 3, 4]) # Calcular el histograma
```

```
plt.figure(figsize=(8, 6)) # Crear una nueva figura
```

```
plt.bar(bins[:-1], hist_test, width=0.5, color='skyblue', edgecolor='black') # Dibujar el
histograma
plt.xlabel('Clase') # Etiqueta del eje X
plt.ylabel('Número de muestras') # Etiqueta del eje Y
plt.title('Histograma de clases en el conjunto de prueba') # Título del gráfico
plt.xticks(bins[:-1]) # Marcas en el eje X
plt.grid(True) # Activar la cuadrícula
plt.show() # Mostrar el gráfico
```

FIN

Bloque 7: se normalizan los datos de entrada utilizando MinMaxScaler, para que todas las características presenten el mismo escalado antes de entrenar el modelo. Se entrena un clasificador KNN (K-Nearest Neighbors) inicializándolo con un número de 3 vecinos y ajustándolo al conjunto de entrenamiento normalizado.

INICIO

```
# Escalando los datos
IMPORTAR MinMaxScaler DE sklearn.preprocessing # Importar el escalador
scaler = MinMaxScaler() # Crear una instancia del escalador
X_train_normalized = scaler.fit_transform(X_train) # Ajustar y transformar los datos de
entrenamiento
X_test_normalized = scaler.transform(X_test) # Transformar los datos de prueba

# Entrenando el clasificador
IMPORTAR KNeighborsClassifier COMO KNN DE sklearn.neighbors # Importar el
clasificador KNN

# Inicializar el clasificador KNN
```

```
Modelo_knn = KNN(n_neighbors=3) # Crear una instancia del clasificador KNN con 3
vecinos
```

```
# Entrenar el clasificador KNN con el conjunto de entrenamiento
```

```
Modelo_knn.fit(X_train_normalized, Y_train) # Ajustar el modelo a los datos de
entrenamiento
```

FIN

Bloque 8: se evalúa el desempeño del modelo KNN calculando su precisión sobre el conjunto de prueba e imprimiendo el resultado.

INICIO

```
# Evaluando el desempeño
```

```
IMPORTAR accuracy_score DE sklearn.metrics # Importar la función para calcular la
precisión
```

```
Y_pred = Modelo_knn.predict(X_test_normalized) # Realizar predicciones sobre el
conjunto de prueba
```

```
# Calcular la precisión del modelo
```

```
accuracy = accuracy_score(Y_test, Y_pred) # Calcular la precisión comparando
predicciones y etiquetas verdaderas
```

```
IMPRIMIR "Precisión del modelo KNN:", accuracy # Mostrar la precisión del modelo
```

FIN

Bloque 9: se evalúa el modelo KNN calculando su precisión generando una matriz de confusión y un informe de clasificación. Se importan las bibliotecas necesarias, se muestra los resultados en la consola, y se visualiza la matriz de confusión en un gráfico. Esto permite analizar el desempeño del modelo en las predicciones.

## INICIO

```

    IMPORTAR graycoprops DE skimage.feature # Importar función para calcular
propiedades de textura
    IMPORTAR matplotlib.pyplot COMO plt # Importar biblioteca para graficar
    IMPORTAR confusion_matrix, classification_report, ConfusionMatrixDisplay DE
sklearn.metrics # Importar métricas de evaluación
    IMPORTAR seaborn COMO sns # Importar Seaborn para visualización (opcional)

# Calcular métricas
accuracy = Modelo_knn.score(X_test_normalized, Y_pred) # Calcular precisión del
modelo
cm = confusion_matrix(Y_test, Y_pred) # Calcular matriz de confusión
report = classification_report(Y_test, Y_pred) # Generar informe de clasificación

# Mostrar resultados
IMPRIMIR "Accuracy:", accuracy # Imprimir precisión
IMPRIMIR "======" # Separador

# Visualizar la matriz de confusión
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=Modelo_knn.classes_) # Crear objeto de visualización
disp.plot(cmap=plt.cm.Blues) # Graficar la matriz de confusión
plt.title('Matriz de Confusión') # Título del gráfico
plt.show() # Mostrar gráfico
IMPRIMIR "======" # Separador

IMPRIMIR " " # Espacio
IMPRIMIR "Classification Report:" # Imprimir título del informe
IMPRIMIR report # Imprimir el informe de clasificación

```

```
IMPRIMIR "===== " # Separador
```

FIN

Bloque 10: se guarda el modelo KNN entrenado y el escalador utilizado para normalizar los datos en archivos utilizando joblib, permitiendo el uso en futuras predicciones sin necesidad de volver a entrenar.

INICIO

```
# Guardar el modelo entrenado en un archivo
```

```
IMPORTAR joblib # Importar la biblioteca para guardar modelos
```

```
joblib.dump(Modelo_knn, '/content/drive/MyDrive/semestre 10/Trabajo de
grado/Modelo_knn.pkl') # Guardar el modelo KNN en un archivo
```

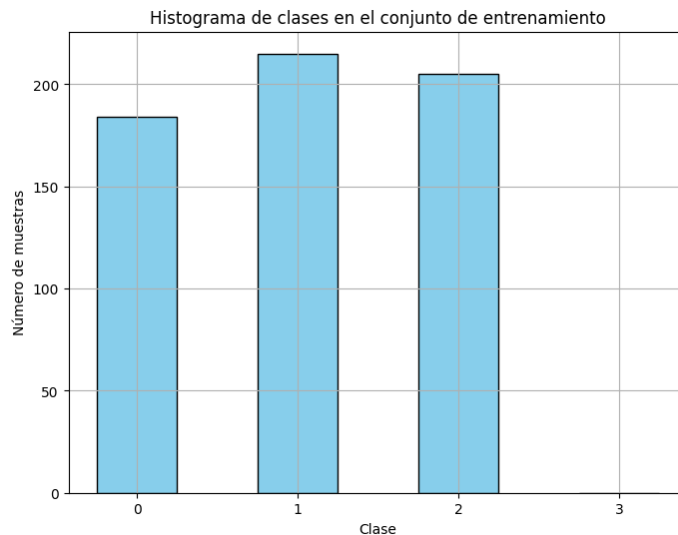
```
# Guardar el escalador
```

```
joblib.dump(scaler, '/content/drive/MyDrive/semestre 10/Trabajo de
grado/escaladordedatosparaelmodelo.pkl') # Guardar el escalador en un archivo
```

FIN

Los resultados de este algoritmo son dos histogramas una matriz de confusión y una tabla con los resultados de precisión en la predicción/clasificación de las tres clases además del modelo entrenado y el escalador de datos para este mismo modelo.

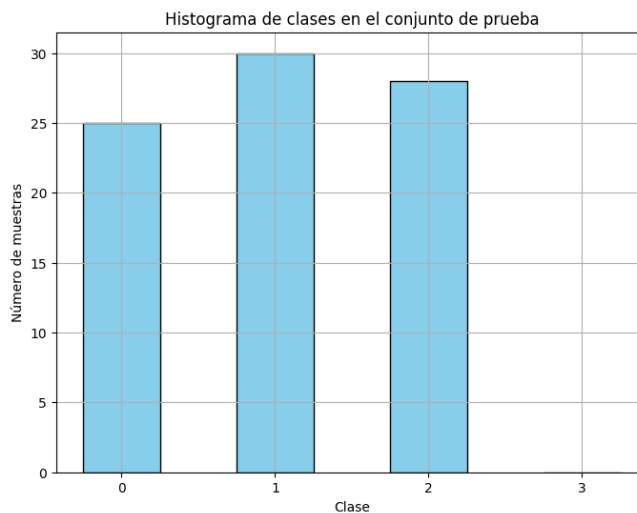
El histograma uno consiste en el número de imágenes que se utilizaron por cada clase en el conjunto de entrenamiento figura 17.



*Figura 17.* Histograma de clases en el conjunto de entrenamiento del modelo knn para el dataset de frutos de cacao

Fuente: fotografía propia

El histograma dos consiste en el número de imágenes que se utilizaron por cada clase en el conjunto de prueba figura 18



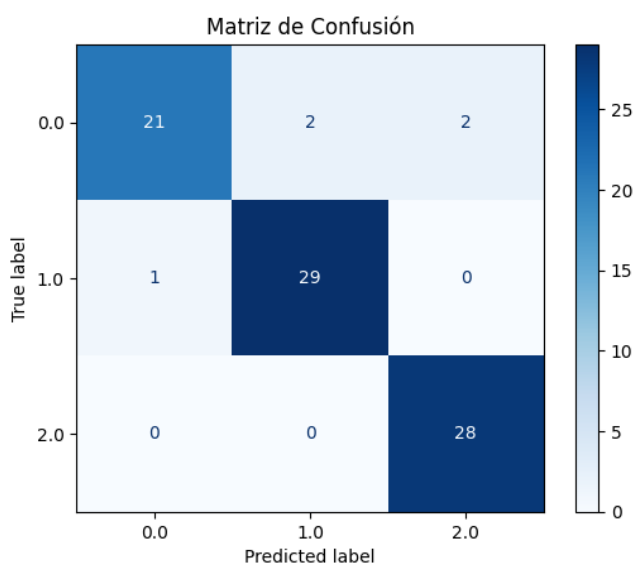
*Figura 18.* Histograma de clases en el conjunto de prueba del modelo knn para el dataset de frutos de cacao

Fuente: fotografía propia

### 6.3 Evaluación del desempeño de los algoritmos clasificadores de machine learning

La evaluación se realizó haciendo uso de la matriz de confusión, la cual muestra los resultados de las predicciones en cada clase, dando una visual de las respuestas correctas e incorrectas que tuvo el modelo. Para el modelo KNN en el dataset de los frutos del cacao se tuvo las clasificaciones: clase 0 con 21 respuestas correctas y 4 incorrectas, clase 1 con 29 respuestas correctas y 1 incorrecta, clase 2 con 28 respuestas correctas y 0 incorrectas, tal como se observa en la figura 19.

La tabla 3 permite conocer el porcentaje de precisión para cada clase que tuvo el modelo KNN entrenado con el dataset de frutos del cacao.



*Figura 19.* Matriz de confusión del modelo knn para el dataset de frutos de cacao

Fuente: fotografía propia

Tabla 3.

*Reporte de clasificación modelo knn dataset frutos del cacao*

Clase	Precisión	Recall	F1Score	Support
0	0.95	0.84	0.89	25
1	0.94	0.97	0.95	30
2	0.93	1.00	0.97	28

Fuente: diseño propio

accuracy			0.94	83
macro avg	0.94	0.94	0.94	83
weighted avg	0.94	0.94	0.94	83

*Figura 20.* Porcentaje de precisión modelo knn para el dataset de frutos de cacao  
Fuente: fotografía propia

A continuación, en busca de cuál es el modelo con más precisión se desarrolla los algoritmos para el entrenamiento de otros 2 modelos de clasificación, se toma como base el algoritmo del modelo ya entrenado KNN para frutos de cacao las características y solo se modifica en el entrenamiento sustituyendo por los modelos que se desean utilizar, los modelos a utilizar son Random Forest, y redes neuronales la descripción del código para entrenamiento se realiza en pseudocódigo.

Descripción en pseudocódigo para implementar el modelo Random Forest para dataset frutos del cacao en el algoritmo.

```
# Importar la clase RandomForestClassifier de sklearn
Importar RandomForestClassifier de sklearn.ensemble como RF

# Inicializar el clasificador Random Forest
Definir Modelo_rf como RF con n_estimators = 100 y random_state = 42

# Entrenar el clasificador Random Forest
Llamar a Modelo_rf.fit con los parámetros X_train_normalized y Y_train

# Evaluar el desempeño del modelo
Importar accuracy_score de sklearn.metrics
Definir Y_pred como Modelo_rf.predict con el parámetro X_test_normalized

# Calcular la precisión del modelo
Definir accuracy como accuracy_score con los parámetros Y_test y Y_pred
```

```
# Imprimir la precisión del modelo
Imprimir "Precisión del modelo Random Forest:", accuracy
```

Descripción en pseudocódigo para implementar el modelo de Redes neuronales para dataset frutos del cacao en el algoritmo.

```
# Importar la clase MLPClassifier de sklearn
Importar MLPClassifier de sklearn.neural_network

# Inicializar el clasificador MLP (red neuronal)
Definir Modelo_mlp como MLPClassifier con hidden_layer_sizes = (100,), max_iter = 500 y
random_state = 42

# Entrenar el clasificador MLP
Llamar a Modelo_mlp.fit con los parámetros X_train_normalized y Y_train

# Evaluar el desempeño del modelo
Importar accuracy_score de sklearn.metrics
Definir Y_pred como Modelo_mlp.predict con el parámetro X_test_normalized

# Calcular la precisión del modelo
Definir accuracy como accuracy_score con los parámetros Y_test y Y_pred

# Imprimir la precisión del modelo
Imprimir "Precisión del modelo MLP:", accuracy
```

Para el desarrollo de los modelos para el dataset de granos de café, se utiliza como base los algoritmos para el entrenamiento de los modelos del dataset de frutos del cacao, se modifica la línea de código que corresponde a la ruta de extracción y almacenamiento de imágenes, se cambia por la ruta o carpeta donde se encuentra el dataset de granos de café y se corre el código para cada uno de los tres modelos KNN, Random Forest y Redes neuronales.

En total se entrenan 6 modelos distintos, 3 modelos son para el dataset con imágenes del cacao y 3 modelos son para el dataset con imágenes de granos de café, los resultados de la precisión para cada modelo se encuentran en la tabla 4.

Tabla 4.  
*Reporte de precisión por modelo y dataset*

<b>Dataset</b>	<b>Modelo</b>	<b>Precisión</b>
Frutos del cacao	KNN	0.94
Frutos del cacao	Random Forest	0.93
Frutos del cacao	Redes neuronales	0.83
Granos de café	KNN	0.99
Granos de café	Random Forest	1.00
Granos de café	Redes neuronales	1.00

Fuente: diseño propio

Para problemas de cacao, proyectos similares alcanzaron un 84.67% de precisión utilizando un modelo VGG19 (Kouassi Simeon Kouassi, 2024) y un 73.91% de precisión utilizando un modelo SVM (Kacoutchy Jean Ayikpa, 2023), Comparando estos resultados con el modelo KNN entrenado, se observa una mejora de 9.33% en precisión respecto a VGG19 y de 20.09% en relación con SVM. Asimismo, los resultados del modelo Random Forest indican una mejora del 8.33% en comparación con VGG19 y del 19.09% frente a SVM. Finalmente, el modelo de red neuronal muestra una disminución de 1.67% respecto a VGG19, pero una mejora de 9.09% en comparación con SVM.

Para problemas de café, proyectos similares alcanzaron un 100% de precisión con modelos de tipo clasificador SVM (Motta, 2024), Comparando estos resultados con el modelo Random Forest y el modelo de red neuronal empleados en este proyecto, ambos lograron también una precisión del 100%, confirmando la correcta selección de modelos. Por otro lado, el modelo clasificador adicional mostró una diferencia de apenas 1%, logrando un 99% de precisión en comparación con el SVM, manteniendo una precisión destacada al ser también un modelo clasificador.



pertenece según el tipo ya sea un fruto de cacao o grano de café, el usuario puede elegir cual tipo desea utilizar de acuerdo a la imagen, a continuación, se describe en pseudocódigo el algoritmo de creación y diseño de la aplicación.

Bloque 1: se importa las bibliotecas necesarias para la aplicación. Incluye tkinter para crear la interfaz gráfica, PIL para trabajar con imágenes, filedialog para permitir la selección de archivos, y time para gestionar o medir tiempos en el programa.

```
# Importar la biblioteca tkinter como tk para crear interfaces gráficas
```

```
Importar tkinter como tk
```

```
# Importar todas las funciones y clases de tkinter
```

```
Desde tkinter importar todas las clases y funciones
```

```
# Importar Image y ImageTk desde la biblioteca PIL para manipular y mostrar imágenes
```

```
Importar Image y ImageTk desde PIL
```

```
# Importar filedialog de tkinter para abrir un cuadro de diálogo de selección de archivos
```

```
Desde tkinter importar filedialog
```

```
# Importar la biblioteca time para medir o gestionar el tiempo
```

```
Importar time
```

Bloque 2: se configura la interfaz gráfica usando Tkinter. Se define la ventana principal, su tamaño, título y estructura de cuadrícula. Se cargan dos imágenes y se muestran en la ventana junto con descripciones de texto y botones para verificar el estado de frutos de cacao y granos de café. Finalmente, se inicia el bucle principal para que la ventana permanezca abierta y activa.

```
# Crear una instancia de Tk para la ventana principal
```

```
Definir ventana como tk.Tk()
```

```
# Configurar el título de la ventana principal
Establecer título de ventana a "Reconocimiento estado de salud de cacao o cafe"

# Definir el tamaño inicial de la ventana (690x500 píxeles)
Establecer ventana.geometry a "690x500"

# Crear y configurar el título en la parte superior de la ventana
Crear label_ventana como tk.Label con texto multilinea
Colocar label_ventana en la cuadrícula en la posición (0,0), con expansión a tres columnas

# Configurar el layout en una cuadrícula de tres columnas y cuatro filas
Configurar ventana.columnconfigure para tres columnas con peso igual (0, 1, 2)
Configurar ventana.rowconfigure para cuatro filas con peso igual (0, 1, 2, 3)

# Cargar y preparar imágenes para mostrar
Abrir imagen_izquierda y imagen_derecha desde rutas especificadas
Redimensionar ambas imágenes a 200x200 píxeles usando el filtro LANCZOS
Convertir imagenes a formato que pueda manejar Tkinter (imagen_izquierda_tk y
imagen_derecha_tk)

# Crear labels para mostrar las imágenes en la interfaz
Definir imagen_izquierda_label con imagen_izquierda_tk, y colocar en (1,0)
Definir imagen_derecha_label con imagen_derecha_tk, y colocar en (1,2)

# Agregar descripciones de texto debajo de cada imagen
Crear texto_izquierda con información sobre reconocimiento de cacao
Colocar texto_izquierda en la cuadrícula en la posición (2,0)
Crear texto_derecha con información sobre reconocimiento de café
Colocar texto_derecha en la cuadrícula en la posición (2,2)

# Crear botones para verificar el estado del fruto de cacao y el grano de café
```

Crear boton\_izquierda y colocar en la posición (3,0) con texto y comando para abrir una ventana de cacao

Crear boton\_derecha y colocar en la posición (3,2) con texto y comando para abrir una ventana de café

# Iniciar el bucle principal para mostrar la ventana

Llamar a ventana.mainloop() para iniciar la interfaz gráfica



*Figura 23.* Interfaz principal de usuario para la aplicación  
Fuente: fotografía propia

Bloque 3: se define una función que crea una nueva ventana donde el usuario puede cargar una fotografía de un fruto de cacao para verificar su estado de salud. La ventana incluye un título, instrucciones, y un botón para seleccionar la imagen. También se añade un botón para regresar a la ventana principal y una etiqueta para mostrar el resultado del modelo de reconocimiento.

# Definir la función abrir\_nueva\_ventana\_cacao para crear una nueva ventana de estado del fruto de cacao

Definir función abrir\_nueva\_ventana\_cacao()

# Inicializar variables globales para imagen, ruta de archivo y respuesta del modelo

Definir imagen\_global como None

Definir ruta\_archivo\_global como None

Definir respuesta\_modelo\_cacao\_global como None

# Declarar variables globales para etiquetas de respuesta y de imagen en la nueva ventana

Definir label\_respuestacacao y imagen\_cacao\_label como globales

# Ocultar la ventana principal y crear una nueva ventana secundaria

Ocultar ventana principal (ventana.withdraw())

Crear nueva\_ventana como una nueva ventana secundaria (Toplevel)

Establecer título de nueva\_ventana a "Estado de Fruto de Cacao"

Establecer tamaño de nueva\_ventana a 690x500 píxeles

# Configurar la cuadrícula en nueva\_ventana con tres columnas y cuatro filas, todas con peso igual

Configurar columna 0, 1, y 2 en nueva\_ventana con peso 1

Configurar fila 0, 1, 2 y 3 en nueva\_ventana con peso 1

# Crear y colocar etiqueta principal en la nueva ventana para el título

Crear label\_nueva\_ventana como etiqueta con texto "Estado del fruto de cacao"

Colocar label\_nueva\_ventana en la cuadrícula (fila 1, columna 0) con expansión a tres columnas y un margen vertical

# Crear botón para regresar a la ventana principal

Crear boton\_regresar en nueva\_ventana con texto "Regresar" y comando para ejecutar la función regresar

Colocar boton\_regresar en la posición superior izquierda (fila 0, columna 0)

# Crear segunda etiqueta en la nueva ventana con instrucciones

Crear label\_instruccion en nueva\_ventana con texto "Elige o sube una fotografía de un fruto de cacao para conocer su estado de salud"

Colocar `label_instruccion` en la cuadrícula (fila 2, columna 0) con expansión a tres columnas y un margen vertical

# Crear botón para seleccionar una imagen del cacao

Crear `boton_seleccionarimagen cacao` con texto "Seleccionar fotografía" y comando para ejecutar `seleccionarfotografia`

Colocar `boton_seleccionarimagen cacao` en la cuadrícula (fila 3, columna 1)

# Crear etiqueta para mostrar la imagen seleccionada en la nueva ventana

Crear `imagen_cacao_label` en `nueva_ventana`

Colocar `imagen_cacao_label` en la cuadrícula (fila 4, columna 1)

# Crear etiqueta para mostrar la respuesta del modelo

Crear `label_respuestacacao` en `nueva_ventana` con texto de `respuesta_modelo_cacao_global`

Colocar `label_respuestacacao` en la cuadrícula (fila 5, columna 1)



*Figura 24.* Interfaz selección de imagen de fruto del cacao en la aplicación  
Fuente: fotografía propia

Bloque 4: se define una función que crea una nueva ventana donde el usuario puede cargar una fotografía de granos de café para verificar su estado de salud. La ventana incluye un título, instrucciones, y un botón para seleccionar la imagen. También se añade un botón para regresar a la ventana principal y una etiqueta para mostrar el resultado del modelo de reconocimiento.

# Definir la función abrir\_nueva\_ventana\_cafe para crear una nueva ventana de estado del grano de café

Definir función abrir\_nueva\_ventana\_cafe()

# Inicializar variables globales para imagen, ruta de archivo y respuesta del modelo

Definir imagen\_globalcafe como None

Definir ruta\_archivo\_globalcafe como None

Definir respuesta\_modelo\_cafe\_global como None

# Declarar variables globales para etiquetas de respuesta y de imagen en la nueva ventana

Definir label\_respuestacafe y imagen\_cafe\_label como globales

# Ocultar la ventana principal y crear una nueva ventana secundaria

Ocultar ventana principal (ventana.withdraw())

Crear nueva\_ventana como una nueva ventana secundaria (Toplevel)

Establecer título de nueva\_ventana a "Estado de grano de Cafe"

Establecer tamaño de nueva\_ventana a 690x500 píxeles

# Configurar la cuadrícula en nueva\_ventana con tres columnas y cuatro filas, todas con peso igual

Configurar columna 0, 1, y 2 en nueva\_ventana con peso 1

Configurar fila 0, 1, 2 y 3 en nueva\_ventana con peso 1

# Crear y colocar etiqueta principal en la nueva ventana para el título

Crear label\_nueva\_ventana como etiqueta con texto "Estado del grano de cafe"

Colocar label\_nueva\_ventana en la cuadrícula (fila 1, columna 0) con expansión a tres columnas y un margen vertical

# Crear botón para regresar a la ventana principal

Crear boton\_regresar en nueva\_ventana con texto "Regresar" y comando para ejecutar la función regresar

Colocar boton\_regresar en la posición superior izquierda (fila 0, columna 0)

# Crear segunda etiqueta en la nueva ventana con instrucciones

Crear label\_instruccion en nueva\_ventana con texto "Elige o sube una fotografía de granos de cafe para conocer su estado"

Colocar label\_instruccion en la cuadrícula (fila 2, columna 0) con expansión a tres columnas y un margen vertical

# Crear botón para seleccionar una imagen del café

Crear boton\_seleccionarimagencafe con texto "Seleccionar fotografía" y comando para ejecutar seleccionarfotoagricafe

Colocar boton\_seleccionarimagencafe en la cuadrícula (fila 3, columna 1)

# Crear etiqueta para mostrar la imagen seleccionada en la nueva ventana

Crear imagen\_cafe\_label en nueva\_ventana

Colocar imagen\_cafe\_label en la cuadrícula (fila 4, columna 1)

# Crear etiqueta para mostrar la respuesta del modelo

Crear label\_respuestacafe en nueva\_ventana con texto de respuesta\_modelo\_cafe\_global

Colocar label\_respuestacafe en la cuadrícula (fila 5, columna 1)



*Figura 25.* Interfaz selección de imagen de grano de café en la aplicación

Fuente: fotografía propia

Bloque 5: se define una función que permite al usuario seleccionar una fotografía de un fruto de cacao desde el sistema. Si se selecciona una imagen, se carga y se llama a la función `predecircacao()` para realizar predicciones sobre la salud del fruto. Si no se selecciona ninguna imagen, se asigna un mensaje de error a la variable de respuesta correspondiente. Finalmente, se actualizan las etiquetas de respuesta y la imagen mostrada en la interfaz, este bloque se duplica con la misma estructura y nombre de variables diferente para utilizarse en la ventana de granos de café.

```
# Definir la función seleccionarfotografia para cargar una imagen seleccionada por el usuario
Definir función seleccionarfotografia()
```

```
# Referenciar variables globales para imagen, ruta de archivo y respuesta del modelo
Definir imagen_global y ruta_archivo_global como globales
Definir respuesta_modelo_cacao_global como global
```

```
# Abrir el cuadro de diálogo para seleccionar una imagen
ruta_archivo_global = abrir un cuadro de diálogo (filedialog.askopenfilename) con título
"Seleccionar imagen" y filtros para archivos de imagen
```

```
# Comprobar si se ha seleccionado un archivo
Si ruta_archivo_global existe entonces:
    Cargar imagen_global usando la ruta seleccionada
    Imprimir mensaje de éxito con la ruta del archivo cargado
    Llamar a la función predecircacao() para procesar la imagen cargada
Sino:
    Imprimir mensaje de error indicando que no se seleccionó ninguna imagen
    Asignar mensaje de error a respuesta_modelo_cacao_global
```

```
# Actualizar la etiqueta de respuesta y la imagen en la ventana
Llamar a la función actualizarlabelrespuestacacao()
Llamar a la función actualizar_imagen_cacao()
```

Bloque 6: se define funciones que permiten regresar a la ventana principal y cerrar la ventana actual. También contiene funciones para realizar predicciones sobre el estado de los frutos de cacao y café, utilizando las variables de imagen y ruta correspondientes. Además, se declaran varias variables globales para almacenar las imágenes y rutas utilizadas en el proceso.

```
# Definir la función regresar para cerrar la ventana actual y mostrar la principal
```

```
Definir función regresar(nueva_ventana):
```

```
    Cerrar la ventana nueva utilizando nueva_ventana.destroy()
```

```
    Mostrar la ventana principal nuevamente utilizando ventana.deiconify()
```

```
# Definir función predecircacao para procesar el estado del cacao
```

```
Definir función predecircacao():
```

```
    Llamar a la función reconocer_estado_cacao con la imagen y la ruta globales
```

```
# Definir función predecircafe para procesar el estado del café
```

```
Definir función predecircafe():
```

```
    Llamar a la función reconocer_estado_cafe con la imagen y la ruta globales para café
```

```
# Declarar variables globales para almacenar imágenes y rutas
```

```
Definir imagen_global, ruta_archivo_global y respuesta_modelo_cacao_global como None
```

```
Definir imagen_globalcafe, ruta_archivo_globalcafe y respuesta_modelo_cafe_global como
None
```

Bloque 7: se definen funciones para actualizar el texto en los labels que muestra las respuestas de los modelos de clasificación para cacao y café. También incluye funciones para actualizar las imágenes mostradas en los labels correspondientes, redimensionando las imágenes a un tamaño específico y asegurando que estén en un formato que Tkinter pueda manejar. Las imágenes se redimensionan utilizando el filtro LANCZOS para un suavizado adecuado.

```
# Actualizar texto en el label de respuesta del cacao
```

```
Definir función actualizarlabelrespuestacacao():
```

Hacer referencia a la variable global `respuesta_modelo_cacao_global` y `label_respuestacacao`

Configurar el texto del `label_respuestacacao` con la respuesta del modelo de cacao

# Actualizar texto en el label de respuesta del café

Definir función `actualizarlabelrespuestacafe()`:

Hacer referencia a la variable global `respuesta_modelo_cafe_global` y `label_respuestacafe`

Configurar el texto del `label_respuestacafe` con la respuesta del modelo de café

# Actualizar imagen en el label del cacao

Definir función `actualizar_imagen_cacao()`:

Hacer referencia a la variable global `imagen_cacao_label` y a la variable `imagen_global`

Cargar la imagen de cacao desde `imagen_global`

Redimensionar la imagen a (200, 200) píxeles utilizando LANCZOS para suavizado

Convertir la imagen a un formato manejable por Tkinter

Actualizar la imagen del label `imagen_cacao_label` con la imagen redimensionada

Guardar la referencia a la imagen en `imagen_cacao_label` para asegurar que no se elimine

# Actualizar imagen en el label del café

Definir función `actualizar_imagen_cafe()`:

Hacer referencia a la variable global `imagen_cafe_label` y a la variable `imagen_globalcafe`

Cargar la imagen de café desde `imagen_globalcafe`

Redimensionar la imagen a (200, 200) píxeles utilizando LANCZOS para suavizado

Convertir la imagen a un formato manejable por Tkinter

Actualizar la imagen del label `imagen_cafe_label` con la imagen redimensionada

Guardar la referencia a la imagen en `imagen_cafe_label` para asegurar que no se elimine

Bloque 8: en este bloque de código se define una función para reconocer el estado del cacao a partir de una imagen cargada. La función comienza importando las bibliotecas necesarias y cargando el modelo entrenado y un escalador previamente guardados. Luego, procesa la imagen en diferentes espacios de color (como HLS, LAB y YCbCr) y extrae características relevantes,

como estadísticas de color (media, desviación estándar, percentiles) y propiedades de textura mediante la matriz de coocurrencia de gris (GLCM). Estas características se almacenan en una matriz de características, que se utilizará posteriormente para clasificar el estado del cacao (saludable o afectado por enfermedades) usando un modelo de clasificación, como KNN. Finalmente, para implementar esta función con el modelo de los granos de café se debe duplicar la misma y modificar el modelo y escalador a usar que se encuentran previamente entrenados y modificar la variable que se toma para la imagen que se desea clasificar o predecir en base al modelo

```
FUNCION reconocer_estado_cacao(imagen_global, ruta_archivo_global)
```

```
    DEFINIR respuesta_modelo_cacao_global COMO VARIABLE GLOBAL
```

```
    IMPORTAR librerías necesarias (matplotlib, cv2, numpy, os, time, Co_ocurrencia,
    graycoprops, joblib, MinMaxScaler)
```

```
    CARGAR Modelo_entrenado DESDE 'Modelo_knn_cacao.pkl'
```

```
    CARGAR loaded_scaler DESDE 'escaladorcacaoknn.pkl'
```

```
    CARGAR imagen desde imagen_global
```

```
    Nombre_archivo = ruta_archivo_global
```

```
    INICIALIZAR listas imagenes y etiquetas
```

```
    Img = LEER imagen desde Nombre_archivo
```

```
    ImgBGRformatrix = LEER imagen desde Nombre_archivo
```

```
    baseImgHLS = LEER imagen desde Nombre_archivo
```

```
    img_HLS = CONVERTIR Img a espacio de color HLS
```

```
    EXTRAER canales H, L, S
```

```
    AUMENTAR brillo del canal L
```

COMBINAR canales a imagen brillante y CONVERTIR a BGR

Bin = REDIMENSIONAR imagen a (100, 100)

BINARIZAR Bin utilizando umbral

CONVERTIR Bin a escala de grises

AÑADIR Bin y ImgBGRformatrix a las listas imagenes y etiquetasBGR

INICIALIZAR matriz X con ceros

PARA cada imagen en imagenes HACER

Bin = imagen actual

bindefaultimage = imagen BGR correspondiente

CONVERTIR bindefaultimage a diferentes espacios de color (Luv, BGR, GRIS, LAB, YCbCr, HLS)

ENCONTRAR contornos en Bin

CALCULAR GLCM de Bin

SI hay contornos ENTONCES

LARGEST\_CONTOUR = contorno más grande

CALCULAR área y perímetro de LARGEST\_CONTOUR

CALCULAR características de textura usando GLCM (contrast, homogeneity, ASM, energy, dissimilarity)

CALCULAR momentos de Hu y dimensiones (alto, ancho)

EXTRAER características del canal de saturación

EXTRAER características del canal YCbCr

EXTRAER características del canal LAB

EXTRAER características del canal gris

EXTRAER características de los canales RGB (azul, verde, rojo)

EXTRAER características del espacio de color LUV

ALMACENAR todas las características en la matriz X

FIN SI

FIN PARA

PREDECIR utilizando el modelo entrenado y X

RETORNAR la respuesta del modelo

FIN FUNCION

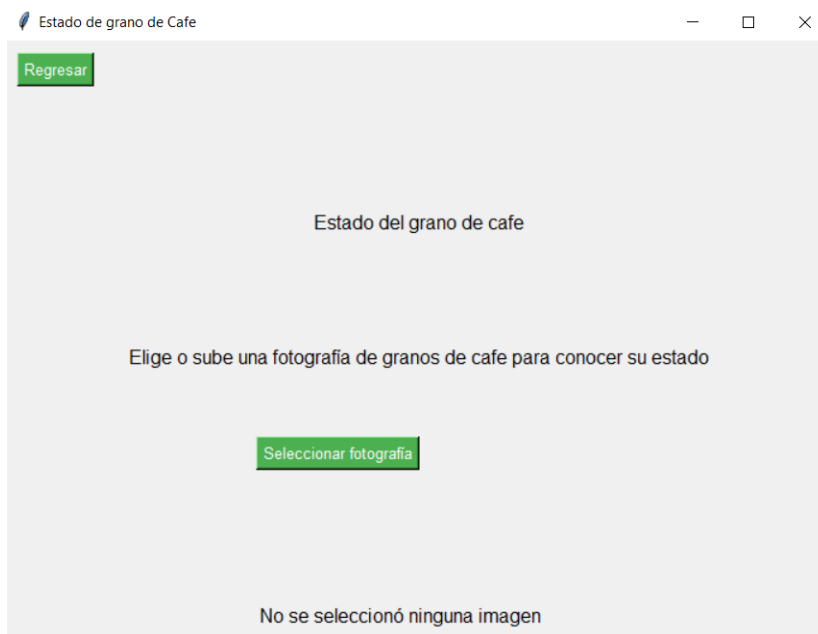
La aplicación permite al usuario ingresar una fotografía desde el equipo local según el modelo que desea utilizar para luego extraer las 78 características, es escalada y procesada por el modelo, esto brinda una respuesta que se imprime en la interfaz del usuario, al presionar el botón seleccionar una fotografía y no seleccionar ninguna la interfaz muestra un mensaje que dice No se seleccionó ninguna imagen

Tabla 5.

*Respuestas que da la interfaz al usuario según el modelo seleccionado y la clasificación de la imagen.*

<b>Dataset</b>	<b>Modelo</b>	<b>Respuesta</b>
Frutos del cacao	KNN	Estoy reconociendo cacao saludable
Frutos del cacao	KNN	Estoy reconociendo cacao con enfermedad mazorca negra
Frutos del cacao	KNN	Estoy reconociendo cacao con enfermedad moniliasis
Granos de café	Random Forest	Estoy reconociendo granos de café saludables
Granos de café	Random Forest	Estoy reconociendo granos de café con defecto verde
Granos de café	Random Forest	Estoy reconociendo granos de café con defecto negro

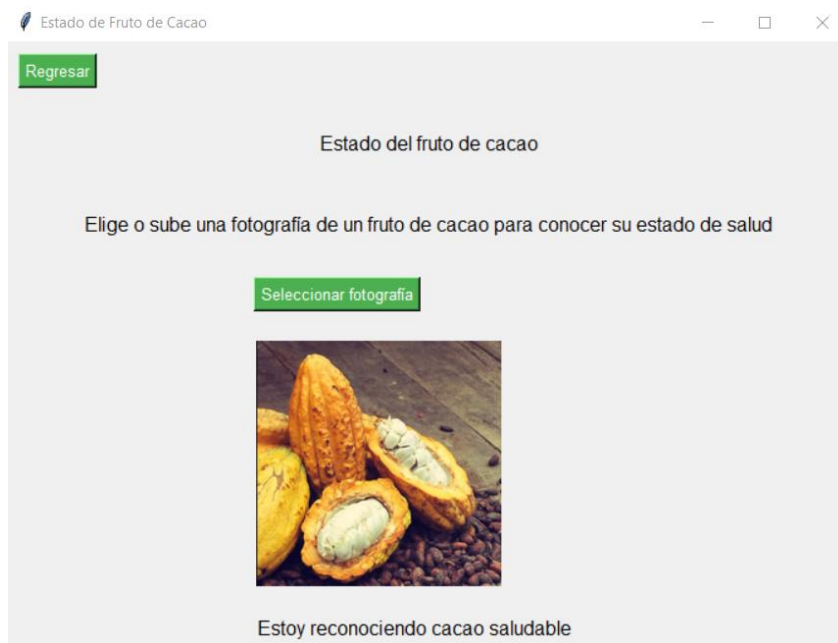
Fuente: diseño propio



*Figura 26.* Respuesta de la interfaz de la aplicación cuando no es seleccionada una fotografía  
Fuente: fotografía propia



*Figura 27.* Clasificación que da el modelo en la aplicación para un grano de café con defecto verde  
Fuente: fotografía propia



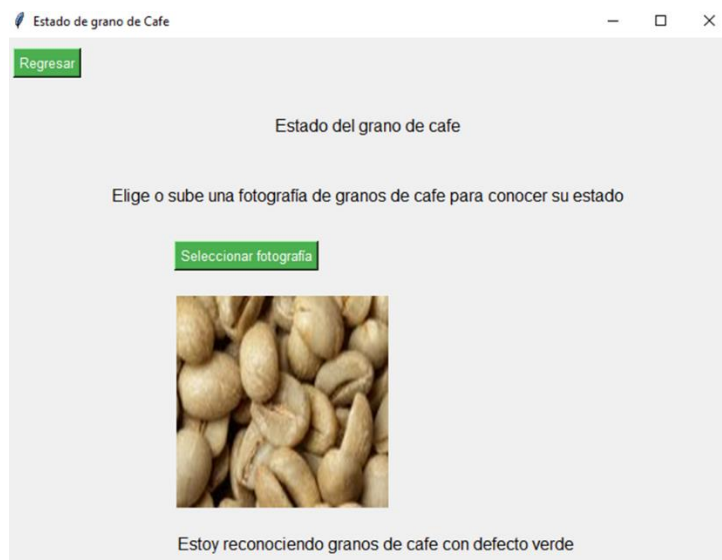
*Figura 28.* Clasificación que da el modelo en la aplicación para un fruto de cacao saludable  
Fuente: fotografía propia

Se realizó una validación de la aplicación y de los modelos mediante el uso de fotografías no vistas previamente por los modelos, subiendo imágenes con las que no fueron entrenados para identificar sus fortalezas y debilidades.

Para el caso del cacao, se observó que ciertas condiciones dificultan el reconocimiento correcto del tipo de fruto. En particular, cuando las fotografías se toman a mayor distancia del fruto, o cuando hay contraluz, el modelo tiene dificultades para clasificarlo correctamente. Además, la precisión disminuye si la imagen incluye el árbol de cacao junto al fruto, otras plantas de fondo, el cielo o nubes, múltiples frutos, o si el fondo es blanco o negro. En estos casos, el modelo tiende a clasificar frutos con fondo blanco como afectados por moniliasis y con fondo negro como afectados por mazorca negra. Para mejorar la precisión, es recomendable tomar las fotografías con buena iluminación y a una distancia de 20 a 30 centímetros del fruto, lo cual minimiza la interferencia de otros objetos en la imagen.

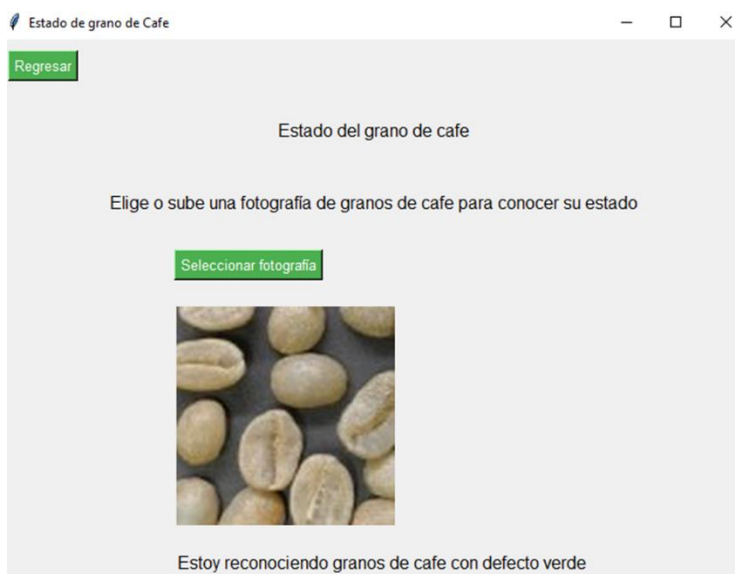
En el caso de los granos de café, se sugiere usar iluminación natural o luz blanca en entornos artificiales, y mantener un fondo neutro, preferiblemente blanco, ya que no afecta la clasificación si es blanco o negro. Es importante capturar un solo grano en cada imagen, ya que la presencia

de múltiples granos puede llevar a clasificaciones incorrectas. Además, el modelo es más preciso con granos de café de tipo arábica, ya que otros tipos pueden variar en color y forma, lo cual podría afectar negativamente la clasificación.



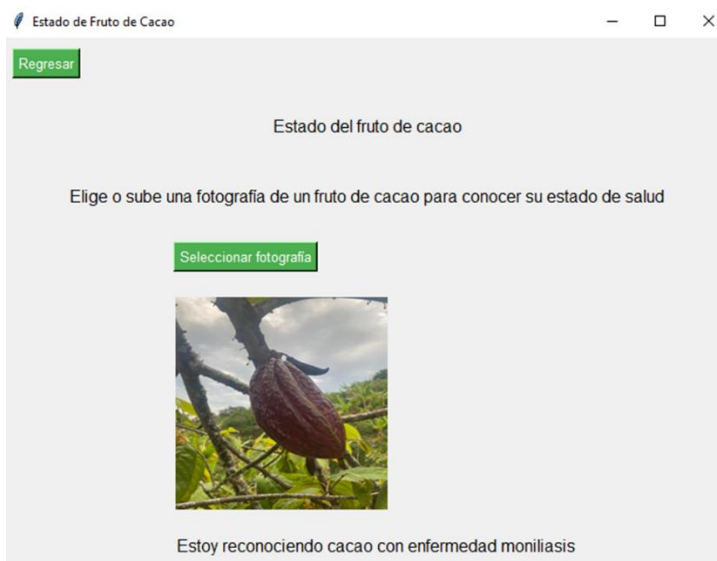
*Figura 29.* Clasificación incorrecta que da el modelo en la aplicación para granos de café de tipo robusta se clasifica como defecto verde

Fuente: fotografía propia



*Figura 30.* Clasificación incorrecta que da el modelo en la aplicación para varios granos de café se clasifica como defecto verde

Fuente: fotografía propia



*Figura 31.* Clasificación incorrecta que da el modelo en la aplicación para un fruto de cacao donde se observa plantas y nubes en el fondo además se encuentra en contra luz  
Fuente: fotografía propia

## 7. Conclusiones

La aplicación diseñada para clasificar imágenes de frutos de cacao y granos de café demuestra una alta efectividad en el uso de algoritmos de machine learning, destacando el KNN y el Random Forest, que alcanzaron precisiones del 94% y 100%, respectivamente. Esto da a entender que estos modelos son herramientas confiables para identificar la salud de los cultivos.

La recolección y creación de los datasets de imágenes de alta calidad, tanto para los frutos de cacao como para los granos de café, es importante para el rendimiento de los modelos. Las condiciones óptimas de toma de fotografías, como la iluminación natural y el uso de dispositivos con cámaras de alta resolución, contribuyen a la confiabilidad de las clasificaciones.

La selección del modelo KNN para el dataset de cacao y Random Forest para el de café se basó en sus respectivas precisiones y tiempos de ejecución. Este enfoque asegura que los modelos no solo sean precisos, sino también eficientes.

La implementación de la librería Tkinter en python para el desarrollo de la aplicación de escritorio ha permitido crear una interfaz intuitiva y accesible para los usuarios del sistema operativo Microsoft Windows 10. Esta interfaz facilita la interacción del usuario con el sistema, permitiendo la selección y carga de imágenes de frutos de cacao o granos de café para la clasificación por medio de los modelos.

Esta investigación destaca la importancia de integrar tecnologías de machine learning en la agricultura, particularmente en la identificación temprana de enfermedades y defectos en cultivos. Al integrar estas tecnologías pueden facilitar la toma de decisiones para los agricultores, mejorando la salud de los cultivos y potencialmente incrementando la producción.

## 8. Recomendaciones

Es importante que el dataset contenga imágenes de alta calidad que representen de manera precisa las características de los frutos de cacao y los granos de café. Las imágenes deben ser nítidas y bien iluminadas para evitar cualquier confusión durante el proceso de entrenamiento del modelo.

Para garantizar un entrenamiento efectivo del modelo, es recomendable que las imágenes incluyan únicamente los frutos o granos en cuestión, evitando la inclusión de elementos como las plantas de cacao o de café. La homogeneidad en las imágenes ayudará al modelo a aprender las características específicas de cada clase sin distracciones.

El fondo de las imágenes juega un papel importante en el proceso de clasificación. Se sugiere utilizar fondos neutros y uniformes que no interfieran con la identificación del objeto principal. Por ejemplo, un fruto de cacao saludable presentado sobre un fondo blanco puede resultar confuso para el modelo. Un fondo que contraste con el color del objeto ayudará a mejorar la precisión en la clasificación.

Para aumentar la capacidad de generalización del modelo, es importante incluir una amplia variedad de imágenes que representen diferentes condiciones, etapas de madurez y características morfológicas de los frutos y granos. Esto permitirá al modelo aprender a reconocer los patrones de manera más efectiva.

Antes de implementar el modelo en la aplicación, es recomendable realizar validaciones y pruebas exhaustivas con un conjunto de datos de prueba para evaluar su rendimiento y realizar ajustes necesarios. Esto asegurará que el modelo sea robusto y confiable en situaciones del mundo real.

Como trabajos futuros para este proyecto se podría realizar una automatización en el proceso de captura de imágenes con el desarrollo de un sistema automatizado, posiblemente mediante el uso de drones o cámaras instaladas en los campos de cultivo que registren imágenes de los frutos

de cacao y café de forma periódica. Automatizar el proceso de captura permitiría recolectar datos de manera más rápida y sistemática, además de mejorar la consistencia en la calidad de las imágenes. la automatización facilitaría la detección de problemas en tiempo real, lo que proporcionaría a los agricultores información inmediata para la intervención oportuna.

Segundo trabajo futuro para este proyecto, ampliación del dataset y diversificación de enfermedades, expandir el dataset para incluir imágenes de frutos de cacao y granos de café que presenten otras enfermedades y defectos no considerados en la fase inicial. Además, recopilar imágenes bajo diferentes condiciones ambientales y de calidad. Contar con un dataset más amplio y diverso permitirá entrenar modelos con una mayor capacidad de generalización, reduciendo los errores de clasificación en condiciones variadas. Un modelo más robusto contribuirá a la toma de decisiones tempranas y más precisas, lo cual es crucial para la gestión efectiva de los cultivos y la mejora de la producción.

Finalmente, como trabajo futuro es ideal la optimización de los modelos para dispositivos móviles, esto consiste en adaptar y optimizar los modelos de machine learning para que puedan ejecutarse eficientemente en dispositivos móviles, utilizando frameworks como TensorFlow Lite. Esto se debe a que la capacidad de realizar diagnósticos directamente desde un teléfono móvil facilitaría a los agricultores que no disponen de acceso regular a una computadora. Se aumentaría el impacto social y económico del proyecto, especialmente en áreas rurales, donde una herramienta móvil resulta mucho más accesible para los agricultores, especialmente cuando los cultivos están dispersos o abarcan grandes extensiones.

## 9. Referencias bibliográficas

Agrobanco. (s.f.). *Manejo Integrado de plagas y enfermedades en el cultivo de cacao*. Recuperado el 11 de Septiembre de 2024, de <https://www.agrobanco.com.pe/data/uploads/ctecnica/010-e-cacao.pdf>

Andrés Calderón Romero, H. H. (26 de Diciembre de 2019). Machine learning in plant disease detection. *TIA*. Recuperado el 12 de Septiembre de 2024, de <https://revistas.udistrital.edu.co/index.php/tia/article/download/15685/15932/88365>

Bonka. (s.f.). *Café arábica: el mejor café para saborear cada día*. Recuperado el 14 de Septiembre de 2024, de <https://www.bonka.es/amor-por-el-cafe/mejor-con-cafe-arabica#:~:text=El%20grano%20de%20ar%C3%A1bica%20puede,S%2C%20su%20caracter%C3%ADstica%20m%C3%A1s%20visible>.

Cenicafé. (s.f.). *Especificaciones de origen y buena calidad del café de Colombia*. Recuperado el 14 de Septiembre de 2024, de <https://biblioteca.cenicafe.org/bitstream/10778/388/1/avt0316.pdf>

Concafe. (18 de Julio de 2023). *El café de Colombia: Un tesoro cafetero que impulsa la economía y enamora los sentidos*. Recuperado el 10 de Septiembre de 2024, de [https://concafe.es/el-cafe-de-colombia-un-tesoro-cafetalero-que-impulsa-la-economia-y-enamora-los-sentidos/?srsltid=AfmBOorTHQZMHiUF\\_Slf3GWnS8\\_FpHpx3zs2rK7TrM3XMVkcF50xPTbe](https://concafe.es/el-cafe-de-colombia-un-tesoro-cafetalero-que-impulsa-la-economia-y-enamora-los-sentidos/?srsltid=AfmBOorTHQZMHiUF_Slf3GWnS8_FpHpx3zs2rK7TrM3XMVkcF50xPTbe)

Ecotact. (2 de Agosto de 2022). *How cocoa bean is harvested from trees*. Recuperado el 12 de Septiembre de 2024, de <https://www.ecotactbags.com/blog/cocoa-harvesting-how-cocoa-bean-is-harvested-from-trees>

Equipo editorial, Etecé. (5 de Agosto de 2021). *Cacao*. Recuperado el 11 de Septiembre de 2024, de <https://concepto.de/cacao/#ixzz8nFLNATnN>

Euskovazza. (s.f.). *El grano del café: tipos, variedades, términos y cómo diferenciarlos*. Recuperado el 14 de Septiembre de 2024, de <https://euskovazza.com/el-mundo-del-cafe/granos-del-cafe-tipos-y-variedades/>

Fedecacao. (11 de Febrero de 2022). *La producción cacaotera nacional sigue creciendo: en 2021 logra un nuevo récord histórico*. Recuperado el 10 de Septiembre de 2024, de <https://www.fedecacao.com.co/post/la-producci%C3%B3n-cacaotera-nacional-sigue-creciendo-en-2021-logra-un-nuevo-r%C3%A9cord-hist%C3%B3rico>

Gualteros, D. M. (12 de Noviembre de 2014). Sistema de visión por computador para detectar hierba. Recuperado el 13 de Septiembre de 2024, de <https://repository.ucatolica.edu.co/bitstream/10983/1690/1/TRABAJO%20DE%20GRADO.pdf>

INIAP. (s.f.). *LAS ENFERMEDADES DEL CACAO Y LAS BUENAS PRÁCTICAS AGRONÓMICAS PARA SU MANEJO*. Recuperado el 10 de Septiembre de 2024, de <https://repositorio.iniap.gob.ec/bitstream/41000/5747/6/iniapbeetpG178.pdf>

Jardín Tecina. (s.f.). *Árbol del Cacao*. Recuperado el 11 de Septiembre de 2024, de <https://www.jardin-tecina.com/es/experiencias-que-hacer/paseo-botanico/arbol-del-cacao/#:~:text=El%20cacaotero%20es%20un%20%C3%A1rbol,los%2010%20m%20de%20altura.>

Kacoutchy Jean Ayikpa, P. G. (25 de Noviembre de 2023). *Classification of Cocoa Beans by Analyzing Spectral Measurements Using Machine Learning and Genetic Algorithm*. Recuperado el 26 de Septiembre de 2024, de <https://www.mdpi.com/2313-433X/10/1/19>

Kouassi Simeon Kouassi, M. D.-C. (1 de Octubre de 2024). *Detection of Cocoa Leaf Diseases Using the CNN-Based Feature Extractor and XGBOOST Classifier*. Recuperado el 27 de Septiembre de 2024, de <https://www.scirp.org/journal/paperinformation?paperid=136903>

Lucía Graziani de Fariñas, L. O. (6 de Junio de 2000). Características físicas del fruto de cacao tipos criollo, forastero y trinitario de la localidad de cumboto, venezuela . Recuperado el

12 de Septiembre de 2024, de [https://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S0002-192X2002000300006](https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0002-192X2002000300006)

Motta, I. V. (18 de Junio de 2024). *Machine Learning Techniques for Coffee Classification: A Comprehensive Review of Scientific Research*. Recuperado el 27 de Septiembre de 2024, de <https://www.preprints.org/manuscript/202406.1462>

Noé Velázquez-López, Y. S.–M. (3 de Diciembre de 2009). *Detección de cenicilla en rosa usando procesamiento de imágenes por computadora*. Recuperado el 28 de Septiembre de 2024, de [https://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S1027-152X2011000200008](https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1027-152X2011000200008)

Ontoum, S. K. (2022). *Coffee Bean Dataset Resized (224 X 224)*. Recuperado el 16 de Septiembre de 2024, de <https://www.kaggle.com/datasets/gpiosenska/coffee-bean-dataset-resized-224-x-224/data>

Ontoum, S. K. (1 de July de 2022). *Kaggle*. Recuperado el 25 de Septiembre de 2024, de <https://www.kaggle.com/datasets/gpiosenska/coffee-bean-dataset-resized-224-x-224/data>

OpenCV. (s.f.). *OpenCV Documentation*. Recuperado el 21 de Septiembre de 2024, de <https://docs.opencv.org/4.x/index.html>

Perfect Daily Grind. (19 de Julio de 2018). *Guía del Productor: Defectos del Café & Cómo Evitarlos*. Recuperado el 15 de September de 2024, de <https://perfectdailygrind.com/es/2018/07/19/guia-del-productor-defectos-del-cafe-como-evitarlos/>

PerfectDailyGrind. (19 de Julio de 2018). *Guía del Productor: Defectos del Café & Cómo Evitarlos*. Recuperado el 11 de Septiembre de 2024, de <https://perfectdailygrind.com/es/2018/07/19/guia-del-productor-defectos-del-cafe-como-evitarlos/>

Procolombia. (25 de Junio de 2020). *El cacao en Colombia está presente en sus departamentos*. Recuperado el 10 de Septiembre de 2024, de

<https://procolombia.co/colombiatrade/exportador/articulos/el-cacao-en-colombia-esta-presente-en-sus-departamentos>

Procolombia. (s.f.). *Cacao, derivados y chocolate*. Recuperado el 10 de Septiembre de 2024, de <https://investincolombia.com.co/es/sectores/agroindustria-y-produccion-de-alimentos/cacao-derivados-y-chocolate#:~:text=Colombia%20se%20posiciona%20como%20uno,mercados%20m%C3%A1s%20sofisticados%20del%20mundo>

Progresia Caribe. (3 de Agosto de 2022). *Enfermedades en el cultivo del cacao*. Recuperado el 10 de Septiembre de 2024, de <https://progresacaribe.info/enfermedades-en-el-cultivo-del-cacao/>

Python. (s.f.). *Graphical User Interfaces with Tk*. Recuperado el 20 de Septiembre de 2024, de <https://docs.python.org/3/library/tk.html>

Python. (s.f.). *Python documentation*. Recuperado el 21 de Septiembre de 2024, de <https://docs.python.org/3/>

Rios, C. A. (s.f.). Detección de enfermedades en plantas de crisantemo, por medio de visión artificial aplicada a imágenes multiespectrales. Recuperado el 14 de Septiembre de 2024, de <https://repositorio.itm.edu.co/handle/20.500.12622/4531>

ScikitLearn. (s.f.). *User Guide*. Recuperado el 20 de Septiembre de 2024, de [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)

Tamayo, M. (2016). *Universidad CLEA*. Obtenido de <https://clea.edu.mx/biblioteca/Tamayo%20Mario%20-%20El%20Proceso%20De%20La%20Investigacion%20Cientifica.pdf>

Tueste Café. (22 de Junio de 2022). *Contaminaciones y defectos en el café*. Recuperado el 15 de Septiembre de 2024, de <https://tuestecafe.mx/blogs/blog-del-cafe/contaminaciones-y-defectos-en-el-cafe#:~:text=Grano%20Inmaduro,verdoso%20a%20hierba%20o%20paja.>

## 10. Bibliografía

OpenCV. (s.f.). *OpenCV Documentation*. Recuperado el 21 de Septiembre de 2024, de <https://docs.opencv.org/4.x/index.html>

Python. (s.f.). *Graphical User Interfaces with Tk*. Recuperado el 20 de Septiembre de 2024, de <https://docs.python.org/3/library/tk.html>

Python. (s.f.). *Python documentation*. Recuperado el 21 de Septiembre de 2024, de <https://docs.python.org/3/>

ScikitLearn. (n.d.). *User Guide*. Retrieved Septiembre 20, 2024, from [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)