



UNIVERSIDAD  
PRIVADA  
DEL NORTE

# FACULTAD DE INGENIERÍA

---

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

“DESARROLLO DE UNA APLICACIÓN MÓVIL QUE PERMITA LA DETECCIÓN Y SUGERENCIA DE TRATAMIENTOS PARA ENFERMEDADES EN CULTIVOS DE MAÍZ”

Tesis para optar el título profesional de:

**Ingeniero de Sistemas Computacionales**

**Autor(es):**

Br. Ruiz Ordoñez Katherine Liseth

Br. Zavala Bustamante Julio

**Asesor:**

Mg. Ing. José Gómez Ávila

**Trujillo – Perú**

**2018**

## APROBACIÓN DE LA TESIS

El asesor y los miembros del jurado evaluador asignados, **APRUEBAN** la tesis desarrollada por el (la) Bachiller **Julio Zavala Bustamante y Katherine Liseth Ruíz Ordóñez**, denominada:

**“DESARROLLO DE UNA APLICACIÓN MÓVIL QUE PERMITA LA DETECCIÓN Y SUGERENCIA DE TRATAMIENTOS PARA ENFERMEDADES EN CULTIVOS DE MAÍZ”**

---

Mg. Ing. José Gómez Ávila  
**ASESOR**

---

Mg. Ing. Rolando Berrú Beltrán  
**JURADO**  
**PRESIDENTE**

---

Mg. Ing. Luis Gutiérrez Magán  
**JURADO**

---

Mg. Ing. Víctor Dávila Rodríguez  
**JURADO**

## DEDICATORIA

Este proyecto está dedicado a nuestros padres, por ser el motor de nuestras vidas, y quienes día a día nos dieron su apoyo para llegar alcanzar esta meta tan importante, a nuestras abuelitas quienes recorrieron el camino con nosotros dándonos ánimos por salir adelante.

## AGRADECIMIENTO

A Dios por ser nuestro guía espiritual.

A nuestros padres y familia por el apoyo diario en nuestros estudios.

A nuestro asesor que con su dedicación, paciencia, esmero y profesionalismo nos dirigió durante todo este trayecto.

## ÍNDICE DE CONTENIDOS

<b>APROBACIÓN DE LA TESIS .....</b>	<b>ii</b>
<b>DEDICATORIA .....</b>	<b>iii</b>
<b>AGRADECIMIENTO .....</b>	<b>iv</b>
<b>ÍNDICE DE CONTENIDOS .....</b>	<b>v</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>vii</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>viii</b>
<b>RESUMEN .....</b>	<b>ix</b>
<b>ABSTRACT .....</b>	<b>x</b>
<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>11</b>
1.1. Realidad problemática .....	11
1.2. Formulación del problema .....	12
1.3. Justificación .....	12
1.4. Limitaciones .....	13
1.5. Objetivos .....	13
1.5.1. <i>Objetivo general</i> .....	13
1.5.2. <i>Objetivos específicos</i> .....	13
<b>CAPÍTULO 2. MARCO TEÓRICO.....</b>	<b>13</b>
2.1. Antecedentes .....	13
2.2. Bases teóricas .....	16
2.2.1. <i>Fundamentos Tecnológicos</i> .....	16
2.2.2. <i>Fundamentos Agrónomos</i> .....	24
2.2.3. <i>Visión Computacional y Procesamiento Digital de Imágenes</i> .....	30
<b>CAPÍTULO 3. HIPÓTESIS .....</b>	<b>50</b>
3.1. Formulación de la Hipótesis .....	50
3.2. Operacionalización de variables.....	51
<b>CAPÍTULO 4. DESARROLLO .....</b>	<b>51</b>
4.1. Fase de Análisis.....	53
4.1.1. <i>Requerimientos Funcionales</i> .....	53
4.1.2. <i>Requerimientos No Funcionales</i> .....	54
4.2. Fase de Diseño .....	54
4.2.1. <i>Diseño de Interfaces</i> .....	54
4.2.2. <i>Procesamiento Digital de Imágenes</i> .....	58
4.3. Fase de Desarrollo.....	63
4.3.1. <i>Implementación de Fragments</i> .....	63
4.3.2. <i>Implementación de Funciones</i> .....	66
4.4. Fase de Pruebas.....	68

<b>CAPÍTULO 5. METODOLOGÍA .....</b>	<b>69</b>
5.1. Diseño de investigación .....	69
5.2. Unidad de estudio .....	70
5.3. Población .....	70
5.4. Muestra .....	70
5.5. Técnicas, instrumentos y procedimientos de recolección de datos .....	70
5.6. Métodos, instrumentos y procedimientos de análisis de datos .....	71
<b>CAPÍTULO 6. RESULTADOS.....</b>	<b>72</b>
6.1. Análisis de Indicadores .....	72
6.1.1. <i>Índice de Sensibilidad</i> .....	73
6.1.2. <i>Índice de Especificidad</i> .....	73
6.1.3. <i>Precisión</i> .....	73
6.1.4. <i>Desempeño</i> .....	74
<b>CAPÍTULO 7. DISCUSIÓN.....</b>	<b>75</b>
<b>CONCLUSIONES .....</b>	<b>76</b>
<b>RECOMENDACIONES.....</b>	<b>76</b>
<b>REFERENCIAS.....</b>	<b>77</b>
<b>ANEXOS .....</b>	<b>81</b>

## ÍNDICE DE TABLAS

Tabla 1: Cuadro comparativo entre gestores de base de datos.....	19
Tabla 2: Cuadro Comparativo de los Ciclos de Vida del Software .....	22
Tabla 3: Fungicidas recomendados para el control de mancha de asfalto .....	30
Tabla 4: Cuadro Comparativo entre los dispositivos para la captura de imágenes .....	32
Tabla 5: Cuadro comparativo de algoritmos de reducción de ruido .....	38
Tabla 6: Cuadro comparativo de algoritmos de detección de bordes .....	41
Tabla 7: Cuadro comparativo de las técnicas en la etapa de Descripción.....	47
Tabla 8: Cuadro comparativo de las técnicas en la etapa de Reconocimiento. ....	50
Tabla 9: Operacionalización de variables. ....	51
Tabla 10: Comparación características de modelos de software. ....	51
Tabla 11: Requerimientos Funcionales. ....	53
Tabla 12: Escala de calificación para la comparación de técnicas. ....	58
Tabla 13: Cuadro comparativo de los dispositivos de captura de una imagen digital basado en puntajes. ....	59
Tabla 14: Cuadro comparativo de las técnicas de reducción de ruido basado en puntajes. ....	60
Tabla 15: Cuadro basado en características de algoritmos de detección de bordes. ....	60
Tabla 16: Cuadro comparativo de los algoritmos de extracción de características. ....	61
Tabla 17: Cuadro comparativo de los algoritmos de reconocimiento e interpretación. ....	62
Tabla 18: Cuadro de técnicas e instrumentos de recolección de datos. ....	70
Tabla 19: Cuadro de métodos y procedimientos de análisis de datos. ....	71
Tabla 20: Análisis Pre Experimental .....	82
Tabla 21: Cuadro de Distribución de Resultados de la Matriz de Confusión aplicada en la App Móvil .....	84

## ÍNDICE DE FIGURAS

Figura 1: Características de Calidad.....	23
Figura 2. Histograma Trimodal.....	42
Figura 3. Ecuación: Segundo pico máximo. ....	42
Figura 4. Ecuación: umbral optimo como el valor que maximiza la dispersión total. ....	43
Figura 5. Imagen segmentada por regiones.....	44
Figura 6. Ejemplo de segmentación por división agrupamiento en una imagen sintética. ....	46
Figura 7. Espacio paramétrico con 2 parámetros y 3 clases.....	47
Figura 8. Ejemplo de algoritmo de Hamming .....	48
Figura 9: Ciclo de Vida Iterativo .....	53
Figura 10: Diagrama de los resultados obtenidos en cada fase del PDI .....	68



## RESUMEN

La presente tesis tiene como objetivo principal el Desarrollo de una Aplicación Móvil que permita la detección y sugerencia de tratamientos para enfermedades en cultivos de maíz usando fundamentos, técnicas y algoritmos del procesamiento digital de imágenes, así como la librería Open CV 3.1.

Se pretende aportar una solución versátil a la detección de enfermedades en los cultivos de maíz a través de una app, siguiendo las fases del PDI, iniciando con la captura y adquisición de imágenes en la que se usó la cámara de un teléfono móvil, luego para el pre-procesamiento, se implementó la técnica de reducción de ruido: filtro gaussiano, esta fase nos facilitó la segmentación, en la que se aplicó la detección de bordes mediante el algoritmo de Canny, seguidamente se llevó a cabo la fase de descripción en la cual se usó la técnica de descripción de regiones según su área, finalmente en el reconocimiento se aplicó la técnica de Decisión teórica para el reconocimiento.

La Aplicación Móvil implementada se trabajó en un ambiente controlado y en un ambiente real, asimismo los cultivos de maíz que se analizaron fueron de la Hacienda Roque Ángulo en la provincia de Ascope.

## ABSTRACT

The main objective of this thesis is the Development of a Mobile Application that allows the detection and suggestion of treatments for diseases in corn crops using foundations, techniques and algorithms of digital image processing, as well as the Open CV 3.1 library.

It is intended to provide a versatile solution to the detection of diseases in corn crops through an app, following the PDI phases, starting with the capture and acquisition of images in which the camera of a mobile phone was used, then the pre-processing, the noise reduction technique was implemented: Gaussian filter, this phase facilitated the segmentation, in which the detection of edges was applied by the Canny algorithm, followed by the description phase in the which was used the technique of description of regions according to their area, finally in the recognition was applied the technique of theoretical decision for recognition.

The Mobile Application implemented was worked in a controlled environment and in a real environment, also the corn crops that were analyzed were from the Hacienda Roque Angulo in the province of Ascope.

## CAPÍTULO 1. INTRODUCCIÓN

### 1.1. Realidad problemática

Hoy en día vivimos en una época donde uno de los problemas que enfrentan los agricultores en diferentes lugares del mundo son las distintas plagas que afectan la producción de los cultivos, por lo que la búsqueda de alternativas de solución frente a este problema se ha convertido en uno de los objetivos centrales en el ámbito de la agricultura, siendo el agricultor quien tiene que identificarlas y tomar decisiones para poder combatirlos. (Merino, 2010).

Las plagas y enfermedades transfronterizas de las plantas pueden propagarse fácilmente a varios países y alcanzar dimensiones de epidemia. Los brotes y los recrudescimientos pueden provocar pérdidas enormes de cultivos y pastos, poniendo en peligro los medios de vida de los agricultores vulnerables y la seguridad alimentaria y nutricional de millones de personas cada vez. (FAO, 2018).

Ante la necesidad de detectar las plagas de los cultivos, se han implementado sistemas de visión computacional para la identificación de sembradíos, identificación de plagas y enfermedades de plantas, sistemas de control de crecimiento de plantas, sistemas de visión para identificación automática de tipo de suelo, sistemas de control de temperatura en invernaderos, identificación de tipo de planta en base a imágenes de su hoja, etc. (Domínguez, 2015).

En los últimos 5 años aproximadamente, se han desarrollado diversos sistemas de identificación de plantas en base a imágenes, explotando diversas características que contienen las hojas como textura, forma geométrica, color y venación. (Domínguez, 2015).

En el ámbito nacional, las Instituciones del MINAGRI y SENASA han intensificado sus acciones de vigilancia para detectar, controlar y contener posibles brotes de plagas o enfermedades en catorce regiones del país en las que se ha sentido los efectos climáticos. Las Instituciones han realizado un trabajo permanente para la detección de plagas y enfermedades que han podido afectar la agricultura. (Senasa, 2015).

Asimismo, en los últimos 10 años en el Perú, diversas instituciones han aplicado programas de manejo integrado de plagas (MIP) en comunidades de pequeños agricultores, como una alternativa al uso de plaguicidas. Mediante estos programas se ha capacitado a miles de pequeños agricultores en el uso de estrategias de manejo de plagas. Sin embargo, 2 años después de concluido el proyecto, el agricultor ha regresado al uso convencional de plaguicidas. En pocos lugares se ha logrado la adopción sostenida del MIP. (REDESA, 2006).

En la costa Peruana, en las plantaciones de maíz realizadas mayormente en los meses de setiembre a marzo, existe una enfermedad que los agricultores denominan la mancha de asfalto. La enfermedad está causando cuantiosas pérdidas debido a que afecta totalmente las hojas en la época de floración y formación de las mazorcas, y reduce el 50% de la producción. (Meneses, 2016).

En el ámbito local, no existe ningún sistema de detección y sugerencia de tratamientos de plagas en las plantas o cultivos. Según MINAGRI (2006) afirma que, la limitada transferencia de tecnología agraria que realizan las diversas instituciones públicas y privadas, así como la poca apropiación de tecnología por parte de los agricultores, ha traído como consecuencia la baja productividad de los cultivos, la poca disponibilidad de los productores a pagar por transferencia tecnológica y la descapitalización de sector, lo que no ha permitido consolidar un mercado de tecnología y servicios agrarios haciendo poco efectivo el sistema de transferencia tecnológica promovido por el Estado a través del INIA.

## **1.2. Formulación del problema**

¿De qué manera la implementación de una aplicación móvil permite la detección y sugerencias de tratamientos para enfermedades en cultivos de maíz?

## **1.3. Justificación**

La investigación se realiza porque existe la necesidad de detectar las plagas a tiempo en cultivos de maíz de pequeños y medianos agricultores buscando así, minimizar los costos que ocasiona el análisis de plagas en un laboratorio, asimismo, haciendo uso de la tecnología móvil se permitirá identificar la enfermedad o plaga correctamente.

De igual manera, con la precisión y confiabilidad de la aplicación se podrán tomar decisiones certeras sobre el uso de pesticidas u otros productos químicos, los cuales muchas veces son usados indiscriminadamente; logrando así, disminuir el impacto en el ecosistema.

El desarrollo de esta investigación ofrece una aplicación móvil para detectar y sugerir un tratamiento a las enfermedades en cultivos de maíz, beneficiando a pequeños y medianos agricultores como también a aquellas empresas u organizaciones del estado, en donde ellos podrán realizar el resto de sus actividades fitosanitarias sin ningún problema y evaluar el cultivo del maíz cuando ellos crean conveniente.

Asimismo, la presente investigación, permite poner en práctica los conocimientos adquiridos en la formación profesional y podrá ser tomado como un antecedente para futuros trabajos enriqueciendo la información sobre aplicaciones móviles que detecten y traten enfermedades en cultivos de maíz, para la comunidad estudiantil.

#### 1.4. Limitaciones

El desarrollo de este estudio requiere conocimientos sobre las enfermedades en las plantas de maíz; por ello, los investigadores hemos establecido contar con el especialista Omar Colán Garay, experto en la rama de Biología, para obtener la información necesaria para el desarrollo de la aplicación propuesta para la detección de enfermedades del maíz.

Asimismo, al no conocer los algoritmos a utilizar en OpenCv relacionados con el procesamiento de imágenes, los investigadores han considerado consultar diversas fuentes para lograr conocer cuáles son los más adecuadas y permitan lograr un óptimo resultado.

#### 1.5. Objetivos

##### 1.5.1. Objetivo general

- Implementar una aplicación móvil que permita la detección y sugerencia de tratamientos para enfermedades en cultivos de maíz en la Huerta “Roque Ángulo” de la Provincia de Ascope.

##### 1.5.2. Objetivos específicos

- Desarrollar la aplicación móvil con un nivel de sensibilidad mayor al 81.69% que asegure la identificación correcta de enfermedad en los cultivos de maíz.
- Desarrollar la aplicación móvil con un nivel de especificidad mayor al 50% que asegure la identificación correcta de enfermedad en los cultivos de maíz.
- Desarrollar la aplicación móvil con un nivel de precisión mayor al 83.07% que ayude y asegure la detección correcta de enfermedad en los cultivos de maíz.
- Desarrollar la aplicación móvil con un nivel de desempeño mayor al 80.92% que asegure al usuario el correcto desempeño para la identificación de enfermedad en los cultivos de maíz.

## CAPÍTULO 2. MARCO TEÓRICO

### 2.1. Antecedentes

La investigación realizada por Domínguez Jalili, L (2015), titulada: “Desarrollo de un Sistema de Identificación y Reconocimiento de Plantas Usando Técnicas de Aprendizaje Máquina”, en

la cual se abordó que el reconocimiento de plantas a partir de técnicas de aprendizaje automático es un campo de investigación muy activo en los últimos años y que los algoritmos de detección y reconocimiento de plantas juegan un rol vital en muchas aplicaciones importantes. Los algoritmos pueden ayudar a agilizar procesos y tareas importantes en aduanas, agricultura, investigación, etc. En esta investigación se direcciona dos problemas de investigación importantes en reconocimiento, el primero es la Extracción de características y el segundo los Métodos de clasificación. Los resultados obtenidos fueron analizados en función del conjunto de técnicas de extracción de características y de los métodos de clasificación utilizados. De tal modo que utilizaron tres tipos de extracción de características: Cromáticas, texturales y geométricas. Asimismo, se usaron tres métodos de clasificación: Máquina de Soporte Vectorial (SVM), Bayesianos (Bayes) y Backpropagation (BP). Por lo tanto, se analizaron una de las tantas hojas a las cuales se compararon y dieron como resultado lo siguiente: Cromáticas (Bayes 88.7%, BP 94.2% y SVM 94.9%), Textuales (Bayes 36.2%, BP 47.7% y SVM 56.9%) y Geométricas (Bayes 81.8%, BP 93.5% y SVM 95.4%). Finalmente, este estudio es pertinente porque nos permite conocer distintos métodos de clasificación como son SVM, Bayes y BP, siendo muy importantes para saber cuál de ellos utilizar en nuestro proyecto de investigación.

Por otro lado, en la investigación realizada por Alambiaga Blanco, J (2015), titulada: "Desarrollo de una Aplicación Móvil para la detección y clasificación de hojas de árboles" cuyo objetivo fue desarrollar una aplicación basada en las técnicas propias de la visión artificial, mediante la cual se puedan realizar fotografías a una hoja determinada y, tras la evaluación de distintas características morfológicas de esta, poder clasificar la hoja según la especie de árbol a la que pertenezca. Para ello, se ha realizado e introducido en la aplicación, una base de datos, fácilmente ampliable, con las seis especies evaluadas para este trabajo. Para la implementación del algoritmo de dicha aplicación se usaron tres módulos integrados de OpenCV como son Core, Imgproc y Highgui. En el reconocimiento de la hoja han usado para el tratamiento de la imagen: escala de grises, filtro gaussiano, umbralización, detección del contorno (algoritmo Canny). Para obtener unos resultados favorables, los investigadores realizaron pruebas en varios contextos como son: iluminación, orientación de las hojas, color de fondo, etc., llegando a la conclusión que el llevar a cabo una multitud de pruebas, les ha permitido esclarecer el buen funcionamiento de la aplicación. Finalmente, este estudio es pertinente porque nos da a conocer algunos algoritmos usados en cada una de las etapas del procesamiento de imágenes.

Se investigó el estudio realizado por Meléndez Ruíz, L (2016) titulado: "Desempeño de un Sistema Inmune Artificial en la Identificación de Plantas", en el cual se aborda que el estudio de las plantas provee valiosa información para incrementar nuestro conocimiento acerca de ellas y potenciar el conocimiento; sin embargo, muchas de las especies alrededor del mundo

no han sido identificadas. A la fecha se tienen identificadas poco más de 267 000 especies, pero se estima que una de cada seis especies, es decir, unas 50 000 especies, aun son desconocidas por la ciencia. En esta investigación, se utilizó una librería de Visión Artificial OpenCV 3.1 para detección y reconocimiento de imágenes, soportado para múltiples lenguajes como c, C++ y Java también se implementó el sistema Inmune Artificial de Reconocimiento para WEKA, que es una colección de algoritmos de aprendizaje automática para minería de datos y también se utilizó NetBeans 8.1 como entorno de desarrollo integrado. Las imágenes de las hojas procesadas mediante la interfaz desarrollada por el autor de la investigación en lenguaje Java. Para la consolidación de los resultados y la creación de los gráficos estadísticos se empleó Microsoft Excel 2016. Como resultados, se evaluó el desempeño del Sistema Inmune Artificial de Reconocimiento en conjuntos de imágenes de hojas de Flavia, Leafnap y Mew, dando como valores por cada marco poblacional los siguientes: Exactitud (67.89%, 63.15% y 80.92%), Precisión (72.36%, 65.37% y 83.07%), Sensibilidad (68.29%, 64.21% y 81.69%) y AUC (83.64%, 81.13% y 90.38%). El sistema presento un desempeño significativamente menor al reportado por las redes neuronales para la problemática de identificación de plantas, observando que es susceptible a la presencia de factores como sombra y fondo de color o textura no uniformes en las imágenes, aunque todo esto no significa que sea inadecuado para la identificación y clasificación de plantas. Finalmente, consideramos pertinente la investigación porque nos permite conocer detalles sobre la librería de Visión Artificial OpenCV 3.1, que nos puede servir para la detección y reconocimiento de imágenes. Asimismo, nos permite comparar que tan factible es utilizar dicha librería con otros métodos como son las redes neuronales.

En el estudio realizado por Barrantes, C & Vigo, L (2015), titulado: " Sistema Experto Móvil para el Diagnóstico y Manejo Integral de Plagas en el Arroz", donde se ha desarrollado un sistema experto, haciendo uso de algunos diagramas UML orientado a un enfoque móvil. La programación está realizada en JQuery Mobile con Php. Para el uso del sistema, se tendrá que seleccionar los síntomas uno por uno, y al termino de realizar dichas acciones, el agricultor debe presionar el botón consultar, dando así el resultado del diagnóstico y manejo integral según los síntomas seleccionados. Como resultado final, obtuvieron una certeza del 80% al hallar un diagnóstico y manejo integral de las plagas en el arroz.

Finalmente, en el estudio realizado por Huaccha Castillo, E (2018), titulado: "Desarrollo de un Sistema de Visión Artificial para realizar una clasificación uniforme de limones", se ha diseñado una solución siguiendo las fases de visión artificial (adquisición, pre procesamiento, segmentación, descripción y reconocimiento e interpretación), donde se realizó una comparativa entre los algoritmos dentro de cada fase, para identificar cuáles son los que mejor se adaptan a la problemática planteada permitiendo obtener un resultado adecuado. Los algoritmos empleados en esta investigación fueron implementados en java. El investigador decidió trabajar con 4 indicadores: sensibilidad, especificidad, tiempo y eficacia los cuales

fueron analizados en 385 imágenes obteniendo resultados del 82.8%, 100%, 0.33 segundos y 83.9% respectivamente. Esta tesis es pertinente para nuestra investigación ya que usa el procesamiento digital de imágenes, y nos muestra con que indicadores podemos trabajar para evaluar nuestros resultados y obtener los más óptimos posibles.

## 2.2. Bases teóricas

### 2.2.1. Fundamentos Tecnológicos

#### 2.2.1.1. Android

##### A. Definición

Robledo, C y Robledo, D. (2011) define que “Android es un sistema operativo, inicialmente diseñado para teléfonos móviles como los sistemas operativos iOS (Apple), Symbian (Nokia) y Blackberry OS”. (p.3).

##### B. Proyecto libre (Open Source)

Robledo, C y Robledo, D. (2011) considera que:

Una de las características más importantes de este sistema operativo reside en que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Por esta razón, es muy popular entre los fabricantes de teléfonos y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos.

##### C. Arquitectura

El artículo sobre Android publicado por Ávila, O (2011) nos dice lo siguiente: La arquitectura del SO está compuesta por cuatro capas, la primera de ellas es un kernel basado en Linux, le siguen las bibliotecas entre las que se encuentran las básicas correspondientes a la máquina virtual, a continuación, está el marco de aplicaciones o framework y finalmente las aplicaciones.

#### 2.2.1.2. Aplicación Móvil

##### A. Definición

La Asociación de Marketing Móvil de España (2011) nos dice que “una aplicación móvil consiste en un software que funciona en un dispositivo móvil (teléfonos y tabletas) y ejecuta ciertas tareas para el usuario”. (p.1).

##### B. Categorías de apps

La Asociación de Marketing Móvil de España (2011) nos dice lo siguiente:



Las apps pueden clasificarse en base a distintos criterios, aunque en este documento hemos elegido dos clasificaciones:

- Según el entorno de ejecución, es decir, “donde funciona la aplicación”.
- En base a las funcionalidades que aporta al usuario.

### **C. Tipos de aplicaciones según su desarrollo**

Cuello y Vittone (2013) nos dice que:

A nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico. Aunque a primera vista esto no parezca incumbencia del diseñador, la realidad es que el tipo de aplicación que se elija, condicionará el diseño visual y la interacción. Entre las que tenemos: Aplicaciones nativas, web e híbridas.

#### **2.2.1.3. Librería OpenCV**

##### **A. Definición**

Arévalo, Gonzáles y Ambrosio (2004) nos dicen lo siguiente:

La librería OpenCV es una API de aproximadamente 300 funciones escritas en lenguaje C que se caracterizan por lo siguiente:

- Su uso es libre tanto para su uso comercial como no comercial.
- No utiliza librerías numéricas externas, aunque puede hacer uso de alguna de ellas, si están disponibles, en tiempo de ejecución.
- Es compatible con The Intel Processing Library (IPL) y utiliza The Intel Integrated Performance Primitives (IPP) para mejorar su rendimiento, si están disponibles en el sistema.
- Dispone de interfaces para algunos otros lenguajes y entornos: EiC - intérprete ANSI C escrito por Ed Breen. Hawk y CvEnv son entornos interactivos (escritos en MFC y TCL, respectivamente) que utilizan el intérprete EiC; Ch - intérprete ANSI C/C++ creado y soportado por la compañía SoftIntegration; Matlab - gran entorno para el cálculo numérico y simbólico creado por Mathworks; y muchos más.

##### **B. Estructura y características de la librería OpenCV**

Arévalo, Gonzáles y Ambrosio (2004) nos dicen lo siguiente:

La librería OpenCV está dirigida fundamentalmente a la visión por computador en tiempo real. Entre sus muchas áreas de aplicación

destacarían: interacción hombre-máquina (HCI); segmentación y reconocimiento de objetos; reconocimiento de gestos; seguimiento del movimiento; estructura del movimiento (SFM); y robots móviles. (p. 2).

#### 2.2.1.4. Base de datos

##### A. Definición

Es un conjunto de datos almacenados sin redundancias innecesarias en un soporte informático y accesible simultáneamente por distintos usuarios y aplicaciones. Los datos deben de estar estructurados y almacenados de forma totalmente independiente de las aplicaciones que la utilizan. (Cobo, 2007)

##### B. Gestores

###### i. MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. (Esepe, 2005)

###### ii. SQLite

SQLite es una biblioteca en proceso que implementa un autónomo, sin servidor, configuración cero, transaccional motor de base de datos SQL. El código para SQLite está en el dominio público y, por lo tanto, es de uso gratuito para cualquier propósito, comercial o privado. SQLite es la base de datos más implementada en el mundo con más aplicaciones de las que podemos contar, incluidos varios proyectos de alto perfil. (SQLite, 2018)

###### iii. MongoDB

Es una base de datos de documentos, NoSQL, muy utilizada (algunas estadísticas la consideran la número uno del mundo),

diseñada para reemplazar a las SQL tradicionales de uso general. Es de código abierto y escrita en C++. Actualmente, está disponible para Windows, Linux y otros sistemas operativos (Joyanes, 2013)

Tabla 1: Cuadro comparativo entre gestores de base de datos

Criterios	MySQL	SQLite	Mongo DB
<b>Interfaz</b>	Lenguaje de consulta estructurado	Lenguaje de consulta estructurado	Orientada a documentos
<b>Concesión de Licencias</b>	Dominio con fines de lucro	Dominio público	Dominio público
<b>Espacio de Almacenamiento</b>	4GB	1GB	Menor a 4TB
<b>Acceso</b>	Maneja múltiples operaciones de consulta	Menor número de solicitudes simultáneas de datos	Solo garantiza operaciones atómicas a nivel documento
<b>Seguridad</b>	Sistema de autenticación	No tiene sistema de autenticación	Sistema de autenticación, autorización.

Fuente: Elaboración Propia

#### 2.2.1.5. Ciclos de Vida de Software

##### A. Definición

Según OkHosting (2011) nos dice que el ciclo de vida nos permite entablar una serie de procedimientos mediante las cuales se procede para la realización del mismo. De hecho, gracias a este término, es que se han ido creando los métodos del ciclo del software, que no son otra cosa más que metodologías que indican distintos pasos a seguir para el desarrollo de un producto.

##### B. Tipos de Ciclos de Vida

Según Instituto Nacional de Tecnologías de la Comunicación (2009) los modelos de ciclo de vida son:

##### i. Modelo en V

El modelo en v es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto. La parte izquierda de la v representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la v representa la integración de partes y su validación. V significa "Verificación y Validación". (Instituto Nacional de Tecnologías de la Comunicación, 2009).

## ii. **Modelo Iterativo**

Es un modelo derivado del ciclo de vida en cascada. Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos. Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien, después de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente. (Instituto Nacional de Tecnologías de la Comunicación, 2009).

## iii. **Modelo en cascada**

El modelo toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y, luego, los representa como fases separadas del proceso, tal como especificación de requerimientos, diseño de software, implementación, pruebas, etcétera. (Sommerville, 2011).

## iv. **Modelo de desarrollo incremental**

El desarrollo incremental se basa en la idea de diseñar una implementación inicial, exponer ésta al comentario del usuario, y luego desarrollarla en sus diversas versiones hasta producir un sistema adecuado. Las actividades de especificación, desarrollo y validación están entrelazadas en vez de separadas, con rápida retroalimentación a través de las actividades. (Sommerville, 2011).

**v. Modelo en espiral**

Es un modelo meta del ciclo de vida del software donde el esfuerzo del desarrollo es iterativo, tan pronto culmina un esfuerzo del desarrollo por ahí mismo comienza otro. Un modelo espiral comienza con la determinación de los objetivos tanto funcionales como de rendimiento. Después se enumeran algunas formas posibles de alcanzar estos objetivos identificando las fuentes de riesgos posibles. Luego se resuelven estos riesgos y se llevan a cabo las actividades de desarrollo, para finalizar con la planificación del siguiente ciclo de la espiral. (Fariño, 2011).

**vi. Modelo de prototipos**

La idea principal es hacer un modelo de la aplicación y presentársela al cliente, sobre todo a nivel de interfaces y otras salidas (consultas, reportes). El cliente hará sus observaciones acerca de lo que ve en ese modelo, y el programador modificará ese modelo de acuerdo con dichas observaciones. El proceso se repite hasta alcanzar cubrir todos los requerimientos del producto, para luego construir la aplicación. (Cortés, s.f.)

Tabla 2: Cuadro Comparativo de los Ciclos de Vida del Software

Criterios	Modelo en V	Modelo Iterativo	Modelo en Cascada	Modelo de desarrollo incremental	Modelo en Espiral	Modelo de Prototipos
<b>Descripción</b>	Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto.	Es la iteración de varios ciclos de vida en cascada.	Modelo rígido, poco flexible y con muchas restricciones.	Se basa en construir incrementado las funcionalidades del programa.	Basado en una serie de ciclos repetitivos.	Valida los requerimientos de los usuarios en cualquier ciclo de vida.
<b>Características propias del modelo</b>	Tiene dos subetapas de retroalimentación.	Ideal cuando el usuario necesita entregas rápidas aunque el proyecto no esté terminado.	Cada una de las cascadas se divide en subetapas independientes.	Es más fácil relevar los conocimientos del usuario.	El espiral se repite las veces que sea necesario.	Suaviza la transición entre los requerimientos iniciales y finales
<b>Usos</b>	Aplicaciones simples.	Adecuado en aplicaciones medianas a grandes.	Ideal cuando se cuenta con un plantel de programadores numeroso.	Adecuado para cualquier proyecto.	Adecuado para grandes proyectos internos de una empresa.	Ideal para tecnologías nuevas o poco probadas.

Fuente: Elaboración Propia

### 2.2.1.6. Norma ISO 25010

Según ISO (2018):

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:

Figura 1: Características de Calidad



Fuente: ISO25000. (2018)

### 2.2.1.7. Matriz de Confusión:

Alarcia (2014) afirma que:

En el aprendizaje supervisado, la mayoría de las métricas utilizadas para estimar la bondad de un clasificador pueden estimarse a través de la matriz de confusión. Permitiendo así, comprender la naturaleza y la distribución del error cometido.

Es un problema de dos clases, es decir, clase positiva y clase negativa, los elementos de esta matriz de confusión viene dados por:

- TP: número de instancias positivas que han sido clasificadas de manera correcta.

- FP: número instancias negativas que han sido clasificadas de manera incorrecta.
- TN: número de instancias negativas que han sido clasificadas de manera correcta.
- FN: número de instancias positivas que han sido clasificadas de manera incorrecta

## 2.2.2. Fundamentos Agrónomos

### 2.2.2.1. Agricultura

#### A. Definición

Ruíz, L (2004) nos dice lo siguiente:

Manejo del suelo para producir alimentos y plantas útiles para las industrias de la alimentación y del vestido. Es una de las actividades más importantes para la economía de un país, y el mejorar las técnicas de producción y las condiciones humanas del agricultor constituye una de las constantes preocupaciones de los gobiernos. El mayor rendimiento de las tierras se encuentra íntimamente relacionado con las características del suelo, el nivel de agua freática y el clima. (p. 9).

### 2.2.2.2. Cultivo del maíz

#### A. Definición

Según Franquesa, M (2016) nos dice que el cultivo de maíz procede de hace miles de años. Forma parte de la alimentación básica en muchos países de Sudamérica. Tiene una gran variedad, desde especies para alimento animal, para procesarlo y fabricar sémola, el maíz dulce, y también el conocido para hacer las famosas palomitas.

#### B. Variedades

Según Franquesa, M (2016) nos dice que:



Dentro de las variedades de maíz en el Perú, presentan una grande cantidad de tipos y de usos culinarios, con una variación de tipos, formas, dimensiones y color del grano mayor que en cualquier otra región.

Según Paliwal R.L nos describe las siguientes variedades:

**i. Maíz Duro**

Los granos de este tipo de maíz son redondos, duros y suaves al tacto. El endospermo está constituido sobre todo de almidón duro córneo con solo una pequeña parte de almidón blando en el centro del grano. El maíz duro germina mejor que otros tipos de maíz, particularmente en suelos húmedos y fríos. Es por lo general de madurez temprana y se seca más rápidamente una vez que alcanzó la madurez fisiológica. Está menos sujeto a daño de insectos y mohos en el campo y en el almacenamiento. Sin embargo, los maíces duros rinden por lo general menos que los maíces dentados.

**ii. Maíz Reventón**

El maíz reventón es una planta baja con tallos débiles y de madurez temprana. La planta produce más de dos mazorcas pequeñas -en algunos casos hasta seis- pero de bajo rendimiento en peso, aunque no en número de granos. Este tipo de maíz no es un cultivo comercial común en los trópicos y se siembra en pequeña escala. En varios países de los trópicos los granos de maíces duros son usados como reventones o son tostados en arena caliente y consumidos como bocadillos. (Paliwal, 2001).

**iii. Maíz Dentado**

En términos generales, el maíz dentado es el tipo de maíz cultivado más comúnmente para grano y ensilaje. El endosperma del maíz dentado tiene más almidón blando que los tipos duros y el almidón duro está limitado solo a los lados del grano. Cuando el grano se comienza a secar, el almidón blando en la parte superior del grano se contrae y produce una pequeña depresión. Esto da la apariencia de un diente y de aquí su nombre. (Paliwal, 2001)

**iv. Maíces Cerosos**

Actualmente estos maíces son cultivados en áreas muy limitadas de las zonas tropicales donde las poblaciones locales los prefieren para su

alimentación; su nombre se debe a que su endospermo tiene un aspecto opaco y ceroso. El almidón en los maíces duros y dentados está comúnmente constituido por cerca 70% de amilopectina y 30% de amilosa; en cambio en los maíces cerosos está compuesto exclusivamente por amilopectina. (Dintzis, Bagley y Felker, 1995).

### 2.2.2.3. Enfermedades más comunes en el maíz

#### A. Complejo Mancha de asfalto

Según el Instituto Colombiano Agropecuario (2007) nos dice que:

En el complejo mancha de asfalto o de alquitrán están involucrados tres microorganismos fungosos *Phyllachoramaydis*Maublanc, *Monographellamaydis*Muller&Samuels y *Coniothyriumphyllachorae*Maublanc, el cual es un hiperparásito de los dos anteriores.

Es una enfermedad que ocurre con mayor frecuencia en zonas frescas y húmedas, especialmente en lotes cercanos a las riberas de los ríos, o en suelos con nivel freático alto, pesados o con tendencia al encharcamiento.

#### B. Gusano cogollero

Según Deras (2014) nos dice que:

Es una plaga universal de gran importancia económica que, dependiendo de algunos factores como la edad de la planta, estado de plaga, condición del clima, así es la severidad del ataque. Cuando el clima es caliente y seco, las larvas completamente desarrolladas, que han caído al suelo antes de convertirse en pupas, empiezan a alimentarse en la base de la planta, cercenando el tallo tierno. En períodos de sequía su presencia y acción puede ser fatal.

Asimismo, el cogollero hace raspaduras sobre las partes tiernas de las hojas, que posteriormente aparecen como pequeñas áreas translúcidas; una vez que la larva alcanza cierto desarrollo, empieza a comer follaje perfectamente en el cogollo que, al desplegarse, las hojas muestran una hilera regular de perforaciones a través de la lámina o bien áreas alargadas comidas. (Ortiz, 2010).

#### C. Roya común

Esta enfermedad, está ampliamente distribuida por todo el mundo, en climas subtropicales y templados y en tierras altas donde hay bastante humedad. Esta enfermedad es conspicua cuando se acercan a la floración. Se les reconoce por las pústulas pequeñas y pulverulentas, tanto en el haz como en el envés de las hojas. Las pústulas son de color café claro en las etapas iniciales de infección; más adelante la epidermis se rompe y las lesiones se vuelven negras a medida que la planta madura. Las plantas del hospedante (*oxalis* sp) son infectadas frecuentemente con pústulas de color anaranjado claro. (García, 2013).

#### **2.2.2.4. Complejo Mancha de Asfalto**

##### **A. Definición**

La mancha de asfalto es una enfermedad que afecta al maíz, causada por el complejo de hongos *Phyllacora maydis* Maublanc, *Monographella maydis* Muller y Samuels y *Coniothirium phyllacorae* Maublanc.

Ocurre con mayor frecuencia en zonas frescas y húmedas, especialmente en campos que se encuentran cerca de las riberas o en suelos que acumulan mucha humedad y tienden a inundarse. El CMA fue reportado por primera vez en México en 1904. (Dardon, 2010).

##### **B. Ciclo biológico**

Quiroga, Garrido, Rosales y Salazar (2017) nos dice:

El CMA se inicia con la presencia de *P. maydis*, produciendo una manchita negra, brillante, levantada y dura sobre las hojas de maíz, con apariencia lustrosa semejante al asfalto o chapopote; esta es seguida por la aparición del hongo *M. maydis*, produciendo alrededor de la mancha de asfalto, un área necrótica de color marrón, llegando a causar lo que se llama “ojo de pescado”. Según las condiciones ecológicas, esta sucesión de síntomas puede suceder entre 8 y 15 días en diferentes grados del proceso infectivo, dado que el CMA es típicamente policíclico.

Posteriormente, la sinergia de *P. maydis* y *M. maydis* origina áreas de tejido foliar necrosado cada vez más amplias, generalmente desde las hojas inferiores hacia la parte superior del follaje, donde la hoja de inserción de la mazorca, debido a su importancia en la asignación de fotosintatos que contribuyen al llenado de grano (Hamzi-Alvanagh et al., 2009; Jalilian y Delkhoshi, 2014), es decisiva en el grado de severidad y daño ocasionado por el CMA.

Eventualmente, la necrosis ocasionada por el CMA se extiende a todo el follaje y ocasiona que las mazorcas de las plantas afectadas sean livianas, con granos flojos y algunos granos presentan germinación prematura (CIMMYT, 2004)

### **C. Daños**

Según DuPont Pioneer (2014) si la enfermedad aparece en etapas muy tempranas, antes del llenado, las mazorcas presentan los siguientes daños: pierden peso, los granos se observan chupados, flácidos y flojos, afecta el rendimiento hasta un 60% y disminuye la calidad del grano.

### **D. Condiciones Ambientales**

Según CIMMYT (Centro Internacional de Mejoramiento de Maíz y Trigo), el desarrollo de la enfermedad se asocia a las siguientes condiciones:

- temperaturas promedio mensuales de 17-22°C, con un mínimo de 7 h de humedad foliar cada noche, y
- humedad relativa de > 75%, o
- de 10 a 20 días de niebla al mes, o
- una precipitación mensual mínima de 150 mm, y
- entre 1800 y 1900 horas de luz solar al año (Hock et al. 1989)

La presencia de rocío sobre las hojas por las noches y las mañanas favorece la infección y el establecimiento de estos patógenos. Otros factores que influyen en el desarrollo de la enfermedad son el cultivo continuo de maíz, que proporciona una fuente constante de inóculo; los altos niveles de fertilización, sobre todo la nitrogenada; la siembra de variedades e híbridos de maíz susceptibles; la poca luminosidad y la virulencia de los patógenos causantes de la enfermedad (Pereyda Hernández et al., 2009)

### **E. Detección**

Según Ciesla (1976), existen dos modalidades:

- Vigilancia de campo: Un programa de vigilancia depende de personal conocedor de los problemas de plagas y capaz de reconocer lo inusual o anormal. Otro fundamento de los programas de vigilancia es el

entrenamiento periódico, por profesionales, sobre el reconocimiento de signos y síntomas de los principales problemas de insectos y enfermedades indígenas para una región dada, sus ciclos de vida y procedimientos para el envío de muestras del agente causal o daños.

- Detección sistemática: Como complemento de la vigilancia de campo, se realizan prospecciones de detección «sistemáticas» o «programadas». Estas se efectúan por especialistas entrenados y su único propósito es descubrir y advertir sobre los daños producidos por los insectos y enfermedades.

## **F. Controles**

Según el CIMMYT, para el control del CMA se emplea un método integral que incluye prácticas agronómicas apropiadas y variedades resistentes como son:

Control cultural:

- Sembrar semillas de variedades resistentes al CMA.
- Eliminar los residuos y el rastrojo en aquellos lugares donde se ha observado la enfermedad.
- Practicar la rotación de maíz y otros cultivos en los cuales no se desarrolle el patógeno.
- Utilizar las dosis de fertilizantes recomendadas.

Mecanismos de resistencia:

- Resistencia vertical
- Resistencia horizontal

Control Químico:

Se recomienda utilizar fungicidas a partir de los 35 días después de siembra en lotes con antecedentes de la enfermedad o si se detecta síntoma del patógeno.

Tabla 3: Fungicidas recomendados para el control de mancha de asfalto

INGREDIENTE ACTIVO	DOSIS
<b>Azoxystrobin+ciproconazol</b>	250 a 280 cc/mz (1.5 copas/bomba de 16 litros)
<b>Pyraclostrobin+Expoconazole</b>	750 cc/mz (2 copas/bomba de 16 litros)
<b>Propiconazole</b>	280 cc/mz (1 copa/bomba de 16 litros)
<b>Ciproconazol</b>	150 cc/mz (0.5 copa/bomba de 16 litros)
<b>Difenoconazol</b>	150 cc/mz (0.5 copa/bomba de 16 litros)

Fuente: CENTA (2014).

### 2.2.3. Visión Computacional y Procesamiento Digital de Imágenes

#### 2.2.3.1. Definición

Jiménez (2015) nos dice que:

Visión es la ventana al mundo de muchos organismos. Su función principal es reconocer y localizar objetos en el ambiente mediante el procesamiento de las imágenes. La visión computacional es el estudio de estos procesos, para entenderlos y construir máquinas con capacidades similares. (p.26).

#### 2.2.3.2. Formación de la imagen digital

Mejía, J (2005) citado por Meléndez, L (2016) nos dice lo siguiente:

Aunque se pueden formar imágenes capturando la radiación a lo largo de todo el espectro electromagnético, en esta sección solo se hará referencia a la parte que comprende la luz visible, y que es abordada por los dispositivos de captura de imágenes tradicionales.

La luz detectada por el lente objetivo de un dispositivo de captura óptico, llega hasta un sensor de imagen denominado Dispositivo de Acoplamiento de Carga (CCD, por sus siglas en inglés), que está formado por multitudes de receptores fotosensibles llamados "fotodiodos". Al incidir la luz en cada receptor se genera una pequeña señal eléctrica que posteriormente es transformada de pulsos a una serie de cadenas binarias mediante un conversor analógico digital en un proceso de muestreo y cuantificación. Debido a que el sensor posee un área de captura que

suele ser rectangular, la imagen del mundo real es representada en forma bidimensional como una matriz numérica, donde sus elementos, denominados píxeles, definen la variación del color capturada en determinada posición del sensor codificada en un conjunto de bits. Estos píxeles son los que finalmente conforman la imagen digital, la cual actúa como un almacén de datos que luego puede ser procesado por diferentes técnicas y algoritmos con el objeto de extraer información. (pp. 6-7).

### 2.2.3.3. Procesamiento Digital de Imágenes

Según Sucar & Gómez (2008), El procesamiento digital de imágenes, es una rama de la visión Computacional, la cual tiene como objetivo mejorar la calidad de las imágenes para su posterior utilización o interpretación.

Como lo menciona García (2008), para el Procesamiento Digital de imágenes se consideran las etapas de: Captura/Adquisición de imagen, Pre-procesamiento, Segmentación, Descripción y Reconocimiento; éstas etapas se definen con más detalle en el apartado 2.2.3.4.

### 2.2.3.4. Etapas del Procesamiento Digital de Imágenes

Según García (2008), para el Procesamiento Digital de imágenes se consideran las siguientes etapas:

#### A. Captura y Adquisición de imágenes

Una imagen puede ser capturada mediante el uso de diferentes dispositivos de Adquisición, a continuación se mencionará los más relevantes.

##### i. Cámara Digital:

- **Resolución:** La mayoría de cámaras comerciales generan imágenes de 640x480 píxeles que son adecuadas para muchas aplicaciones.
- **Memoria:** Las cámaras pueden incluir memoria interna fija, así como tarjetas de memoria intercambiables.
- **Sistema de Transferencia:** Para el traspaso de imágenes a la PC, la mayoría cuentan con conexión USB.
- **Velocidad:** Cabe mencionar que depende del Hardware de la cámara digital.

##### ii. Escáner:

Es un dispositivo que le permite realizar una copia digital de una fotografía u otro tipo de documento.

- **Resolución:** Depende la precisión del motor que impulsa pausadamente al CCD (ChargeCoupledDevice - dispositivo de carga acoplado) a lo largo del documento. Algunas resoluciones de escáner son: 300, 600, 1200, 2400, 4800, 9600 dpi (dots per inch – puntos por pulgada).
- **Memoria:** No necesita memoria externa.
- **Sistema de Transferencia:** Cuenta con un CCD con una sola fila de píxeles fotosensibles. Este CCD líneal registra el valor lumínico de cada una de las líneas o filas de una imagen. Al terminar una línea remite la información de inmediato a la PC para poder proceder la lectura de la línea siguiente. Además requiere de una conexión USB.
- **Velocidad:** Depende del motor que impulsa pausadamente al CCD y también del usuario quien coloca los documentos a escanear.

### iii. Cámara de Teléfono Móvil:

- **Resolución:** Numero de pixeles que contienen las imágenes. La mayoría de cámaras móviles comerciales generas imágenes de 640x480 pixeles que son adecuadas para muchas aplicaciones. En la mayoría de cámaras es posible elegir entre dos o más resoluciones.
- **Memoria:** Las cámaras pueden incluir memoria interna fija, así como tarjetas de memoria intercambiables. El formato de las imágenes capturadas serán del tipo JPG, por lo que podrían ser exportadas a la mayoría de los programas de tratamiento.
- **Compresión de archivos:** La mayoría de cámaras móviles permite elegir el grado de compresión de las imágenes para su almacenamiento en la memoria.
- **Sistema de Transferencia:** Para el traspaso de imágenes a la PC, cuenta con conexión USB.
- **Versatilidad y Velocidad:** Las aplicaciones y herramientas de edición te abren un sin fin de posibilidades a la hora de crear tus fotografías.
- **Tamaño:** El tamaño del celular es idóneo para cargarlo de manera cómoda. Por lo tanto, siempre tendrás una cámara en el bolsillo.

Tabla 4: Cuadro Comparativo entre los dispositivos para la captura de imágenes

Características	Cámara Digital	Cámara Móvil	Escáner
-----------------	----------------	--------------	---------



<b>Resolución</b>	16 MP	12 a 16 MP	1200 a 2400 dpi
<b>Memoria</b>	48 MB de memoria interna del dispositivo, expandible a 32GB	16 GB expandible a 32gb	No tiene memoria
<b>Sistema de transferencia</b>	USB 2.0	USB 3.0	USB 2.0
<b>Velocidad de captura</b>	30 fps a 60 fps	30 fps	Depende de la resolución elegida para el escaneo.

Fuente: Elaboración Propia

## B. Pre procesamiento

### i. Técnicas de realce de contraste:

- **Aumento lineal del Contraste**

Según Sucar & Gómez (2008), Utilizando el valor de intensidad mínimo y máximo en una imagen, podemos aumentar su contraste.

La idea básica es llevar el valor mínimo (min) a cero y el máximo (máx.) a 255, pensando en imágenes monocromáticas (0-255). Esta transformación genera que las intensidades se espacien de acuerdo a cierto factor o pendiente; el factor para este aumento lineal de contraste es:

$$(x, y) = ((x, y) - \text{minmax} - \text{min} * 255)$$

Donde  $(x, y)$  es la imagen a procesar y  $(, y)$  es la imagen con aumento lineal del contraste.

Se puede verificar fácilmente que para  $(x, y)$  en  $\text{min}$ ,  $(x, y)$  resulta cero (el numerador es cero); para  $(x, y)$  en  $\text{max}$ ,  $(x, y)$  resulta en 255, obteniendo como cociente 1.

- **Ecuación por Histograma**

Según Cortés, Muriel & Mendoza (2011), el histograma  $hI$  de una imagen  $I$  de escala de grises, es un arreglo vectorial de orden  $1 \times L$  elementos que representa gráficamente los posibles valores de intensidad desde 0 hasta  $L - 1$  en el eje horizontal y con la cuenta  $hI(k)$  de píxeles con igual intensidad  $(r, c) = k$  en el eje vertical mediante barras.

El histograma proporciona información sobre la distribución de la luz en la imagen en forma de brillo o contraste y puede ser utilizado para ajustar dichos parámetros (Cortés, Muriel & Mendoza, 2011).

El principal objetivo de la ecualización de histograma es producir un histograma con una distribución uniforme, para obtener un buen nivel de contraste.

Para Aguilar M. (2007), el proceso de ecualización del histograma consiste en definir una variable continua  $r$  que representa los niveles de gris de una imagen a ser mejorada, asumiendo que  $r$  ha sido normalizada en el intervalo  $[0, 1]$ , con  $r = 0$  representando el negro, y  $r = 1$  representado el blanco. Posteriormente, se considera un planteamiento discreto, en donde los valores de los píxeles estén en el intervalo  $[0, L - 1]$ , dado que  $L$  es el número total de posibles niveles de gris en la imagen. La siguiente es la forma de la función de transformación de interés.

$$S = (r) \quad 0 \leq r \leq 1$$

La ecualización, produce un nivel  $S$  para cada píxel  $r$  en la imagen original.

## ii. Técnicas de Binarización por Umbralización:

Como afirma Sucar & Gómez (2008), la binarización es una operación puntual, la cual consiste en transformar una imagen de entrada a una imagen de salida; de forma que, cada píxel de la imagen de salida sólo depende del correspondiente píxel de la imagen de entrada. Para obtener una imagen binaria se hace una transformación, en la cual cada píxel puede tomar uno de dos valores: 0 ó 1, para otro caso 0 ó 255, etc. La decisión de que cada píxel tome un valor depende de la elección de un umbral  $T$ .

Por otro lado, Cortés, Muriel & Mendoza (2011), define la umbralización como una técnica de segmentación simple y eficiente que permite separar los píxeles de la imagen en escala de grises en dos categorías a partir de un una ventana o borde umbral. El umbral puede ser función de la posición  $(r, c)$ , de la vecindad  $(r, c)$  y de la intensidad  $(r, c)$  actual del píxel

$$T = T(p(r, c), N(r, c), I(r, c))$$

Según Cortés, Muriel & Mendoza (2011), el umbral fijo o global es aquel que es único sobre toda la imagen.

$$b(r, c) = \begin{cases} t_0, & \text{si } (r, c) < T \\ t_1, & \text{si } (r, c) \geq T \end{cases}$$

Cuando el valor  $t_0 = 0$  y el valor  $t_1 = 1$  se dice que la imagen se ha binarizado.

Son 3 técnicas de umbralización global:

- **Método Isodata (Mean Intensity):**

Según Ridler T. & Calvard S. (1978), un objeto puede ser extraído de su fondo en una imagen por la selección de un umbral. El umbral óptimo puede ser elegido de forma automática como resultado de un proceso iterativo, iteraciones sucesivas que proporcionan un aumento de extracciones aclarando la región del objeto.

El proceso iterativo inicia dividiendo el histograma en dos partes, utilizando como umbral inicial el promedio de las intensidades; luego se calcula la media de intensidad de cada segmento del histograma:  $m_1$  y  $m_2$ , utilizando estos valores se calcula un nuevo valor de umbral  $T$  mediante la siguiente ecuación:

$$T = \frac{(m_1 + m_2)}{2}$$

El proceso continúa hasta que en dos pasos consecutivos el valor umbral calculado no cambie. La ecuación genérica que calcula el umbral utilizando el histograma mediante este método es:

$$T_k = \frac{\sum_{i=0}^{T_{k-1}} i * h(i)}{2 \sum_{i=0}^{T_{k-1}} h(i)} + \frac{\sum_{i=T_{k-1}+1}^N j * h(i)}{2 \sum_{i=T_{k-1}+1}^N h(i)}$$

Donde,  $T_k$  es el valor del umbral en la iteración  $k$ .

- **Método del Mínimo:**

Para Cortés J, Muriel A & Mendoza J. (2011), cuando el histograma es de tipo bimodal, hace relativamente fácil la selección del umbral más adecuado. De la inspección visual del histograma, se puede ver que una buena opción para la separación de fondo y los objetos es un lugar el cual corresponde al punto más bajo entre los dos picos principales. Para realizar el algoritmo se recomienda suavizar la curva reiteradas veces hasta lograr que solo existan 2 máximos locales. Si se suaviza el

histograma bimodal, se pueden definir más claramente el umbral adecuado ya que este se encuentra en el valle localizado entre los 2 únicos picos. Para localizar los máximos locales, se recurre a los criterios de la derivada que indican:

Si  $h'(k) = 0$  y  $h''(k) < 0$  entonces es un máximo local

El histograma debe generar dos únicos máximos locales. Seguidamente, se busca el valor mínimo entre los dos máximos locales siendo este punto el que posee el umbral adecuado T (Cortés J, Muriel A & Mendoza J. 2011).

- **Método del Triángulo**

Para determinar el umbral adecuado mediante el método del triángulo, se deben determinar el valor máximo de la probabilidad  $P_{max}$  y mínimo  $P_{min}$  de la gráfica del histograma. El umbral se determina mediante otra línea recta la cual debe cortar perpendicular a la primera y rodear el histograma hasta encontrar la máxima distancia  $d$ . El punto donde la recta toca el histograma corresponde a umbral deseado T. Esta técnica resulta adecuada cuando el histograma es de tipo bimodal y posee solo picos claramente definidos (Cortés J, Muriel A & Mendoza J. 2011).

iii. **Técnicas de Reducción de ruido mediante filtros:**

La mayoría de técnicas de reducción de ruido usan el proceso de Filtrado Espacial, el cual para EDMANS (2006), consiste en recorrer toda la imagen, pixel a pixel, y realizar alguna operación aritmética con un número concreto de píxeles vecinos. Al conjunto de éstos píxeles vecinos se les denomina ventana o máscara, la cual puede ser de 3x3, 5x5,... etc. En la figura 13 se puede ver la representación de una ventana 3x3 y de 5x5, en la que los píxeles vecinos van del 1 al 9 y del 1 al 25 respectivamente, siendo el píxel central sobre el que se aplica un algoritmo. La mayoría de los algoritmos suelen ser combinaciones lineales de los píxeles de la ventana. Para representar un filtro se suele dar el vector o matriz de convolución o núcleo.

A continuación mencionaremos algunas técnicas de Reducción de ruido que usan filtros espaciales.

- **Erosión Binaria:**

Es el resultado de comprobar si el elemento estructurante B está completamente incluido dentro del conjunto X. Cuando no ocurre, el resultado de la erosión es el conjunto vacío:

$$E(X) = X \ominus B = \{x \mid Bx \subseteq X\}$$

Cuando los objetos de la escena sean menores que el elemento estructurante, éstos desaparecerán.

Su utilidad consiste en definir una geometría determinada al elemento estructurante y pasarlo sobre la imagen. Los objetos menores al elemento estructurante no aparecerán en la imagen resultante. Los objetos que queden de la transformación habrán sido degradados. Por tanto, la erosión supone una degradación de la imagen. (Platero Dueñas C., 2014).

- **Dilatación Binaria:**

Es la transformación dual a la erosión. El resultado de la dilatación es el conjunto de elementos tal que al menos algún elemento del conjunto estructurante B está contenido en el conjunto X, cuando B se desplaza sobre el conjunto X:

$$\delta B(X) = X \oplus B = \{x \mid X \cap Bx \neq \emptyset\}$$

Esta operación representa un crecimiento progresivo del conjunto X. Al pasar el elemento estructurante dentro del conjunto, éste no se modificará. Sin embargo, en la frontera del conjunto X, al desplazar a B, el conjunto resultado se expansionará. La aplicación iterada de este operador haría degradar la imagen, haciendo coincidir el conjunto dilatado con la imagen. La dilatación también se interpreta como el valor máximo del entorno de vecindad definido por el elemento estructurante. (Platero Dueñas C., 2014).

Según el Grupo de Investigación EDMANS (2006), la apertura es utilizada para:

- Segmentación de objetos, separando unas formas de otras.
- Descomposición de objetos en elementos más simples.
- Extracción de formas determinadas en un entorno con ruido.
- Eliminar salientes estrechos.
- Separar objetos que no están demasiado pegados.
- Aumentar los agujeros que están dentro de los objetos.
- La apertura puede estar formada por un número de erosiones y dilataciones diferentes, produciendo resultados muy dispares.

- **Filtro de Mediana:**

Es un método de alisamiento que reduce el emborronamiento de las fronteras. La idea es reemplazar el valor de la imagen en un punto por la mediana de los niveles de brillo vecinos. Es obvio que los puntos aislados de ruido son eliminados (Molina R., 1998).

Es muy usado para eliminar el ruido impulsivo de una imagen, La ventaja es que con éste filtro los detalles abruptos y bordes de los objetos (correspondientes a las partes altas de frecuencias) no se difuminan tanto (Grupo de Investigación EDMANS, 2006)

En el filtro de la mediana, la máscara de convolución se emplea así:

- Se toman todos valores de la imagen cubiertos por la máscara.
- Se ordenan de menor a mayor.
- Se toma el valor central.

- **Filtro Gaussiano:**

Para Aguilar M. (2007), el reductor de ruido Gaussiano se basa en una máscara que se convoluciona con una imagen, para efectos de reducir el ruido. Sin embargo, éste tipo de máscara remueve también detalles de la imagen y la difumina.

El interés para procesamiento de imágenes es una máscara en dos dimensiones. En éste sentido, una máscara Gaussiana de 2D se define de la siguiente manera:

$$(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Donde  $\sigma$  es la desviación estándar de la distribución.

En la siguiente figura se da el ejemplo de una máscara de 5x5 para el filtro gaussiano con  $\sigma = 1.0$

Tabla 5: Cuadro comparativo de algoritmos de reducción de ruido

Criterios	Apertura	Filtro de Mediana	Filtro gaussiano
Uso	Modifica el volumen del objeto	Elimina ruido	Emborrona imágenes

<b>Ventajas</b>	<ul style="list-style-type: none"> <li>• Elimina pequeñas protuberancias.</li> <li>• Suaviza contorno de objetos.</li> </ul>	El valor final del pixel es un valor real de la imagen y no un promedio	<ul style="list-style-type: none"> <li>• Produce un suavizado más uniforme.</li> <li>• Es un filtro separable</li> </ul>
<b>Desventajas</b>	Degrada la imagen cuando es de menor tamaño.	<ul style="list-style-type: none"> <li>• Resulta más complejo de calcular.</li> <li>• Pierde detalles</li> </ul>	<ul style="list-style-type: none"> <li>• Disminución de nitidez</li> </ul>

Fuente: Elaboración Propia

### C. Segmentación

Gómez D y Guerrero A (2016) nos dice que:

En el proceso de segmentación, la imagen se fracciona con el objetivo de clasificar las partes que la componen, a estas partes las podemos llamar zonas o regiones de la imagen; cada una de ellas está compuesta por diversas características. Y conforme a estas características es que se seleccionan las zonas de interés para las técnicas que se van aplicar. Los atributos básicos de segmentación de una imagen son: la luminancia en imágenes monocromáticas, los componentes de color en imágenes en color, textura, forma, etc. Podemos decir que la segmentación automática es una de las fases más complicadas del procesamiento de imágenes, en esta se determina si el análisis de la imagen es exitoso o no, y muy pocas veces se logra una solución confortable. Dentro de la etapa de segmentación que hace parte del procesamiento digital de imágenes se encuentran muchas técnicas, las cuales podemos agrupar dentro de estos tres grupos:

- \* Detección de bordes
- \* Umbralización
- \* Basado en regiones

Estas serán explicadas cada una a continuación:

## i. Detección de bordes

La imagen está compuesta por objetos de diversas características y en la detección de bordes se localiza la intersección de estos objetos para poder determinar sus regiones e información. Los bordes pueden clasificarse por su anchura, ángulo de su pendiente de variación, y las coordenadas de su punto medio. Los diferentes modelos de bordes son: línea, tipo escalón, tipo rampa y tipo tejado. Las técnicas más conocidas para la detección de bordes son: Roberts, Prewitt, Sobel y Canny. La mayoría de las técnicas de detección de bordes se basa en la primera y la segunda derivada; en la primera derivada se aplica el operador gradiente y en la segunda derivada se aplica el operador laplaciano.

- **Operador Roberts:** Este operador aplica la derivada de primer orden, es decir, el método del gradiente.

En el operador horizontal se hace un rastreo de filas y en el operador vertical se hace un rastreo de columnas. Es una técnica muy sencilla, ya que consiste en calcular el gradiente en la dirección horizontal y/o vertical para cualquier imagen. También se han propuesto kernels o máscaras para otras direcciones diferentes a la horizontal y vertical.

- **Operador Prewitt:** Este operador al igual que el de Roberts, aplica la derivada de primer orden, es decir, el método del gradiente y está compuesto por dos máscaras o kernels.

El gradiente horizontal permite detectar bordes orientados verticalmente, mientras que para el gradiente vertical se detectan los bordes orientados horizontalmente. El gradiente total de la imagen se puede obtener a través de distintas combinaciones del gradiente horizontal y el gradiente vertical, donde los métodos más utilizados son:

$$\text{Magnitud del gradiente: } G = |G_x|^2 + |G_y|^2 \quad (1)$$

$$\text{Máximo gradiente: } G = \text{Max} \{|G_x|^2 + |G_y|^2\} \quad (2)$$

Después de esto, la imagen del gradiente debe ser binarizada con un umbral adecuado para la detección de los bordes deseados.

- **Operador Sobel:** Este operador se implementa de igual forma que el operador Roberts y Prewitt. El filtro Sobel detecta los bordes horizontales y verticales separadamente sobre una imagen en escala de grises. Las imágenes en color se convierten en RGB en niveles de grises. Como con el filtro Laplace, el resultado es una imagen transparente con líneas negras y algunos restos de color.



- **Operador Canny:** El operador Canny utiliza un filtro basado en la primera derivada de un Kernel G, donde luego permite discriminar los valores en que el gradiente no es máximo local, por lo que se debe a variaciones ruidosas de la intensidad de pixel, a través de este se obtienen mejores resultados que el operador Roberts, Prewitt y Sobel.

Tabla 6: Cuadro comparativo de algoritmos de detección de bordes

Crterios	Roberts	Prewitt	Sobel	Canny
<b>Ventajas</b>	Buena respuesta ante bordes diagonales.	Proporciona mayor inmunidad al ruido.	Desaparece falsos bordes.	Buena detección, localización y precisión
<b>Desventajas</b>	Extremada sensibilidad al ruido y por tanto tiene pobres cualidades de detección.	Detecta valores pixel a pixel, lo que genera muchas zonas inconexas.	Mala respuesta en bordes diagonales, lentitud de cálculo.	Es muy sensible al ruido.

Fuente: Elaboración propia

## ii. Umbralización

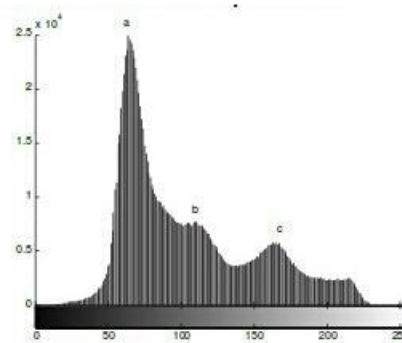
Este es una técnica muy importante dentro de la segmentación de imágenes, el umbral se precisa como una función que convierte una imagen con diferentes tonalidades en una imagen en blanco y negro. Se elige un umbral que permita agrupar los píxeles que hacen parte de los objetos de la imagen, diferenciándolos del fondo. El histograma de una imagen es una función discreta que se plantea como el número de píxeles gris sobre el número de píxeles de la imagen. Por lo tanto, la segmentación por histograma consiste en elegir uno o varios umbrales para poder reunir los puntos de la imagen con respecto a sus características y de acuerdo a sus niveles de gris. El método de la Binarización es muy básico pero sirve para diferenciar el objeto, del fondo de la imagen; este consiste en que los píxeles que están marcados con 1 corresponden al objeto de la imagen y los píxeles que están marcados con 0 corresponden al fondo de la imagen. Si el objeto es oscuro con respecto al fondo, el proceso es inverso. Existen varios tipos

de umbrales; pero hay 3 de ellos que son los más utilizados; estos son: Umbral global, Umbral local y Umbral dinámico.

Estas son las técnicas y/o métodos de la umbralización:

- **Técnica de los dos picos:** Hay veces en que se desea encontrar el umbral más bajo ubicado entre dos picos, en eso se basa esta técnica; descrito en los siguientes dos pasos: - Encontrar los dos picos altos del histograma. - Encontrar el punto más bajo entre ellos. Es sencillo encontrar el primer pico del histograma, pero es menos sencillo cuando se debe encontrar el segundo pico más alto. Veamos el siguiente ejemplo:

Figura 2. Histograma Trimodal



Fuente: Gómez,D. & Guerrero,A. (2016)

En la ilustración se puede observar que el histograma tiene tres picos, a, b y c. El pico más alto corresponde al punto a, el segundo pico más alto es el punto b, pero el umbral debe ser establecido tomando el punto c.

Una solución a este problema es tener en cuenta la distancia al pico máximo; teniendo en cuenta la siguiente ecuación:

Figura 3. Ecuación: Segundo pico máximo.

$$h(j) = \max_{0 \leq j \leq 255} ((j - k)^2 h(j))$$

Fuente: Gómez,D. & Guerrero,A. (2016)

Donde  $h(k)$  viene siendo el pico máximo del histograma y  $h(j)$  es el segundo pico más alto del histograma; la ecuación 1 incluye una restricción de distancia con el fin de poder calcular  $h(j)$  (el segundo pico máximo).

- **Algoritmo ISODATA:** Es un algoritmo iterativo el cual permite obtener el umbral correcto de manera automática. Los pasos son los siguientes:
  - i. Estima un valor umbral inicial  $T$ .
  - ii. Segmenta la imagen usando el valor del umbral  $T$ .
 
$$G1 = \text{píxeles con intensidad} > T$$

$$G2 = \text{píxeles con intensidad} < T.$$
  - iii. Se calcula respectivamente la media de la intensidad  $m1$  y  $m2$  de  $G1$  y  $G2$ .
  - iv. Se calcula un nuevo umbral  $T$ .
 
$$T = \frac{1}{2} (m1 + m2).$$
  - v. Se repiten los pasos del ii al iv hasta que cada iteración de los valores de  $T$  sea más pequeña que un parámetro predefinido Delta de  $T$ .
- **Técnica Otsu:** Otsu es un método de umbralización que utiliza la distribución de probabilidad, la media y la varianza de las intensidades para definir el umbral óptimo. En este método se utiliza un umbral inicial y se definen dos distribuciones de probabilidad, luego las dos medias de las distribuciones respectivamente. Finalmente se calcula la distribución combinada y la dispersión total. El método de Otsu define, para una umbralización de dos niveles, el umbral óptimo como el valor que maximiza la dispersión total obteniendo como final la siguiente ecuación:

Figura 4. Ecuación: umbral optimo como el valor que maximiza la dispersión total.

$$T^* = \max_{0 \leq T \leq 255} (\sigma_c^2(T)).$$

Fuente: Gómez,D. & Guerrero,A. (2016)

El objetivo de este método es calcular el valor umbral de forma que la dispersión dentro de cada clase sea lo más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre clases diferentes.

La umbralización tiene muchas aplicaciones; en el campo de la medicina permite la detección de células en citología; localización de fracturas, análisis de ecografías, CAT's e imágenes de rayos X en traumatología. En la industria permite la supervisión automática de procesos de control de calidad, vigilancia o detección de fallos. Esta técnica también permite el análisis de fotografías aéreas o por satélite para su posterior clasificación.

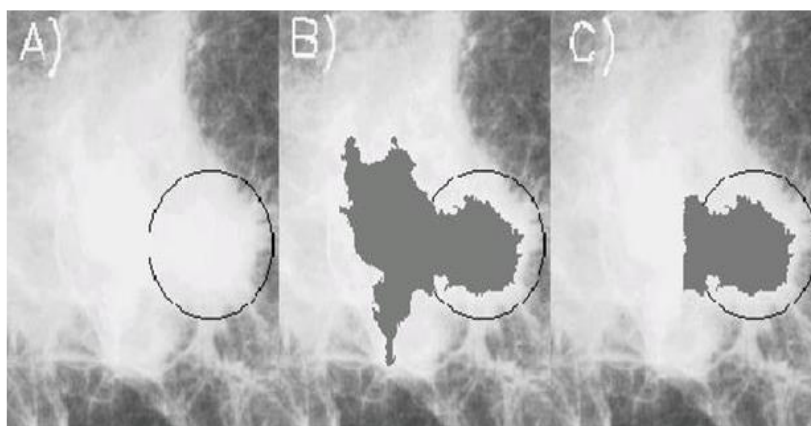
### iii. Basado en regiones

El objetivo principal de esta técnica de segmentación es dirigirse en la búsqueda de las regiones de interés de la imagen. La imagen se divide en regiones cada una con características particulares que la diferencia de las otras regiones; esto se da porque los píxeles de una región poseen características similares. La segmentación basada en regiones es una técnica que verifica las siguientes condiciones:

- La unión de todas las regiones generadas cubren la imagen original a segmentar por completo.
- Las regiones obtenidas son conexas según la distancia seleccionada.
- Las regiones obtenidas son disjuntas.
- Los píxeles pertenecientes a una determinada región cumplen algún criterio de similitud.

La condición de similitud que se cumple en los píxeles de una región, no se cumple para los píxeles de la unión de dos o más regiones.

Figura 5. Imagen segmentada por regiones



Fuente: Gómez,D. & Guerrero,A. (2016)

El algoritmo general que describe paso a paso la segmentación basada en regiones, consta de tres partes las cuales se describirán a continuación:

- i. Iniciar la partición: En esta parte inicial se elige la región o regiones que componen la imagen, es muy importante el conocimiento que se tiene sobre las regiones a extraer; por lo que es conveniente la supervisión de un experto que elija los puntos claves para la segmentación.

ii. Definir un criterio de homogeneidad: Aquí se deben seleccionar los criterios que permiten clasificar las regiones seleccionadas en el punto anterior, por partes o clases. Se clasifican los píxeles en grupos por las semejanzas que comparten; Ej: Un píxel pertenece a una región cuando comparte el mismo color, textura y otras características que posea el grupo de píxeles que se encuentran en esa región.

iii. Optimizar la partición mediante modificaciones (división, unión, etc.). En algunos casos después de esta etapa se regresa de forma iterativa al paso 2: Una vez realizados los dos pasos anteriores se procede al último paso en el cual se modifican y actualizan las particiones con el fin de modificar las condiciones de homogeneidad. Los cambios que se realizan en este paso consisten en la división y unión de regiones.

- **Crecimiento de regiones:** consiste en reunir píxeles o regiones con características similares para formar regiones más grandes que las agrupen. El algoritmo consiste en escoger un conjunto de píxeles, al cual se le llamará píxeles semilla y se le irán agregando píxeles vecinos con características similares a este nuevo grupo de píxeles se le evaluará sus vecinos y se agregaran los que tengan características similares; así se hará hasta que ya no se encuentren píxeles parecidos. El criterio de parada muchas veces representa un problema.
- **División y fusión de regiones:** esta técnica no consiste en la agregación de píxeles como la anterior, por el contrario; consiste en dividir y subdividir la imagen. Sea  $R$  la región correspondiente a la imagen completa y sea  $P$  el predicado de similitud elegido. En este caso para segmentar  $R$  se subdivide recursivamente cada región en cuadrantes más y más pequeños hasta que se cumpla para cada región  $R_i$  que  $P(R_i) = CIERTO$ . Es decir si  $P(R) = FALSO$ , se divide la imagen en cuatro regiones (cuadrantes). Si para alguno de los cuadrantes  $P$  es FALSO, se vuelve a dividir dicho cuadrante en cuatro subcuadrantes y así sucesivamente.

#### D. Descripción

Gómez D y Guerrero A (2016) nos dice que:

**i. Descriptores de frontera según su longitud:**

La longitud de un contorno es uno de los descriptores más sencillos, Si se cuenta el número de píxeles en el contorno se obtendrá una aproximación de su longitud. Si la frontera ha sido codificada usando cadenas de códigos, el número de componentes verticales más el de horizontales más el de diagonales multiplicados por  $\sqrt{2}$  da la longitud exacta (Grupo de Investigación EDMANS, 2006).

El diámetro de una frontera B se define como:

$$(B) = \text{máx} [(p1, p2)]$$

Donde D es una medida de distancia y p1 yp2 son puntos de la frontera.

**ii. Descriptor de región según su Área:**

Para realizar la medida del área de un objeto, se utiliza el vector que define su contorno. EL área se va a definir como el número de píxeles que están dentro del contorno. Uno de los métodos más rápidos, consiste en contar el número de píxeles que hay en cada fila de puntos del objeto.

Figura 6. Ejemplo de segmentación por división agrupamiento en una imagen sintética.

			X	1	1	1	X			=	5
		X	1	1	1	1	X			=	6
	X	1	1	1	1	1	1	X		=	8
X	1	1	1	1	1	1	1	1	X	=	10
X	1	1	1	1	1	1	1	1	X	=	10
X	1	1	1	1	1	1	1	1	X	=	10
X	1	1	1	1	1	1	1	1	X	=	10
	X	1	1	1	1	1	1	1	X	=	9
	X	1	1	1	1	1	1	X		=	8
		X	1	1	1	1	X			=	6
			X	1	1	1	X			=	5

Fuente: Grupo de Investigación EDMANS, 2006.

De esta forma el cálculo del área es muy sencillo:

$$\text{Área} = \sum_{h=\min(\text{filas})}^{\max(\text{filas})} \max(\text{columna}(h)) - \min(\text{columna}(h))$$

Tabla 7: Cuadro comparativo de las técnicas en la etapa de Descripción

Características	Descriptor de frontera según su longitud	Descriptor de región según su área
<b>Ventajas</b>	Facilita la comparación de objetos.	Homogeneidad en el objeto a evaluar.
<b>Desventajas</b>	Cualquier ruido induce a errores.	Ocupa siempre una gran cantidad de memoria independiente del tamaño del objeto.

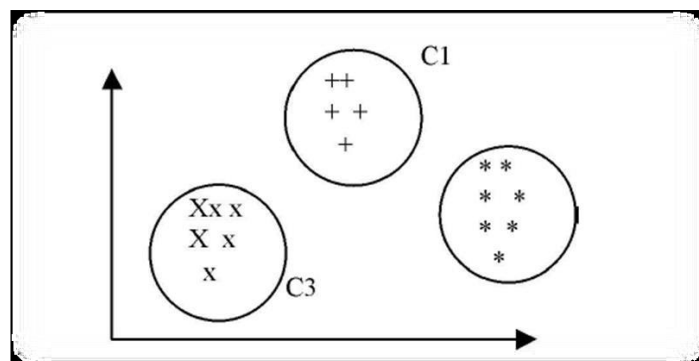
Fuente: Elaboración propia

## E. Reconocimiento

### i. Reconocimiento estadístico de patrones:

Para Sucar, L. & Gómez G. (2008), consiste, básicamente, en buscar, dentro de un espacio paramétrico, la clase (modelo) más "cercana" a la descripción del objeto en la imagen. Si consideramos que se tienen, por ejemplo, dos parámetros y 3 tipos (clases) de objetos, el problema se puede visualizar como se presenta en la siguiente figura:

Figura 7. Espacio paramétrico con 2 parámetros y 3 clases.



Fuente: Sucar, L. & Gómez G. 2008

Cada punto representa un objeto. Los objetos similares se muestran con diferentes símbolos (x, +, \*), los cuales, normalmente, están agrupados en el espacio paramétrico. Dado un objeto desconocido, el problema es encontrar a que grupo (clase) pertenece dados sus parámetros (Sucar, L. & Gómez G. 2008).

## ii. Algoritmo de Hamming:

Es un código que se utiliza en la detección y corrección de errores que se producen en la transmisión de códigos binarios, la palabra de código se conforma por los bits de comprobación y los bits de información. La Distancia entre dos combinaciones binarias, viene dada por el número de bits que hay que cambiar en una de ellas para obtener la otra.

El código Hamming se genera a partir de una distancia mínima de 3, ya que una distancia de 1 o 2 es inconveniente para poder generar un código para detectar un error.

La distancia mínima de Hamming está dada por la siguiente ecuación:

$$D_m = 2X + 1$$

Donde  $D_m$  es la distancia mínima de un código para permitir la corrección de datos y  $X$  es las líneas de datos.

Si se tuviese una distancia de 1 no tendríamos valores para saber si hay o no un error.

Si se tienen una distancia de 2 se podría lograr un sistema de corrección de error con un bit de paridad pero el problema que esto lleva es que un sistema con distancia 2 puede tener datos erróneos que pueden pasar como datos correctos. Con la distancia de 3 se tiene que los códigos no van a tener datos o resultados similares por ende se puede detectar el error y se puede desarrollar un sistema para su corrección como el Hamming (Cisneros & Sepúlveda, 2012).

Ejemplo: Se tiene los vectores:  $h_1 = GRANDE$  y  $h_2 = GRANOS$

Para calcular la distancia de Hamming, Podemos definir la distancia entre ellas como el número de caracteres en las mismas posiciones que son distintos, número de entradas diferentes entre los vectores.

Figura 8. Ejemplo de algoritmo de Hamming



G	R	A	N	D	E
G	R	A	N	O	S
✓	✓	✓	✓	✗	✗

Fuente: Ramírez (2012)

### iii. Decisión teórica para el reconocimiento

Los métodos de decisión teórica para el reconocimiento se basan en la utilización de funciones de decisión (o discriminantes). Supongamos que  $X = (x_1, x_2, \dots, x_n)^T$  representa un patrón vectorial de dimensión  $n$ . Para  $M$  clases de patrones,  $w_1, w_2, \dots, w_m$ , el problema básico en el reconocimiento de patrones por decisión teórica consiste en encontrar  $M$  funciones de decisión  $d_1(X), d_2(X), \dots, d_M(X)$  que tengan la propiedad que, si un patrón  $X$  pertenece a la clase  $w_i$ , entonces,

$$d_i(X) > d_j(X) \quad j = 1, 2, \dots, M \quad i \neq j$$

En otras palabras, un patrón desconocido  $X$  pertenece a la  $i$ -ésima clase de patrones si, al sustituir  $X$  en todas las funciones de decisión,  $d_i(X)$  toma el mayor valor numérico. En caso de igualdad, los conflictos se resuelven arbitrariamente. La frontera de decisión que separa la clase  $w_i$  de la  $w_j$  viene dada por valores de  $X$  para los que  $d_i(X) = d_j(X)$  o, expresado de forma equivalente, por los valores de  $X$  para los que,

$$d_i(X) - d_j(X) = 0$$

La práctica común consiste en hacer que la frontera de decisión entre dos clases sea la función:

$$d_{ij}(X) = d_i(X) - d_j(X) = 0$$

De esta forma,  $d_{ij}(X) > 0$  para los patrones de la clase  $w_i$  y  $d_{ij}(X) < 0$  para los patrones de la clase  $w_j$ . (Departamento de electrónica automática y bioingeniería, 2016)

Tabla 8: Cuadro comparativo de las técnicas en la etapa de Reconocimiento.

<b>Criterios</b>	<b>Reconocimiento estadístico de patrones</b>	<b>Algoritmo de Hamming</b>	<b>Decisión teórica para el reconocimiento</b>
<b>Ventajas</b>	Distribución de probabilidad asociada a las características.	Detecta y corrige con seguridad hasta un bit por cada bloque de información.	Se basa en funciones de decisión.
<b>Desventajas</b>	Problemas al determinar características adecuadas para describir objetos.	La cantidad de bits de paridad empleados en la transmisión de la información le restan eficiencia al proceso.	Se presenta de forma probabilística.

Fuente: Elaboración Propia

## CAPÍTULO 3. HIPÓTESIS

### 3.1. Formulación de la Hipótesis

Una aplicación móvil permite la detección y sugiere el tratamiento de enfermedades en cultivos de maíz en la Huerta Roque Ángulo en Ascope.

### 3.2. Operacionalización de variables

Tabla 9: Operacionalización de variables.

VARIABLE INDEPENDIENTE	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
Aplicación móvil	Es una aplicación informática basada en tomas de imágenes a través de algoritmos capaces de comparar imágenes sujetas a patrones de comparación, diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles.	Sensibilidad	Nivel de sensibilidad
		Especificidad	Nivel de Especificidad
		Corrección Funcional	Nivel de Precisión
VARIABLE DEPENDIENTE	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
Detección y sugerencia de tratamientos para enfermedades en cultivos de maíz	Se determina a aquellas enfermedades en los cultivos de maíz que presentan algún tipo de síntoma basado en enfermedades específicas como el complejo mancha de asfalto, que en su fase preliminar detecta y realiza un tratamiento determinado como sugerencia para el agricultor para su posterior curación.	Capacidad	Nivel de desempeño

Fuente: Elaboración propia.

## CAPÍTULO 4. DESARROLLO

Tabla 10: Comparación características de modelos de software.

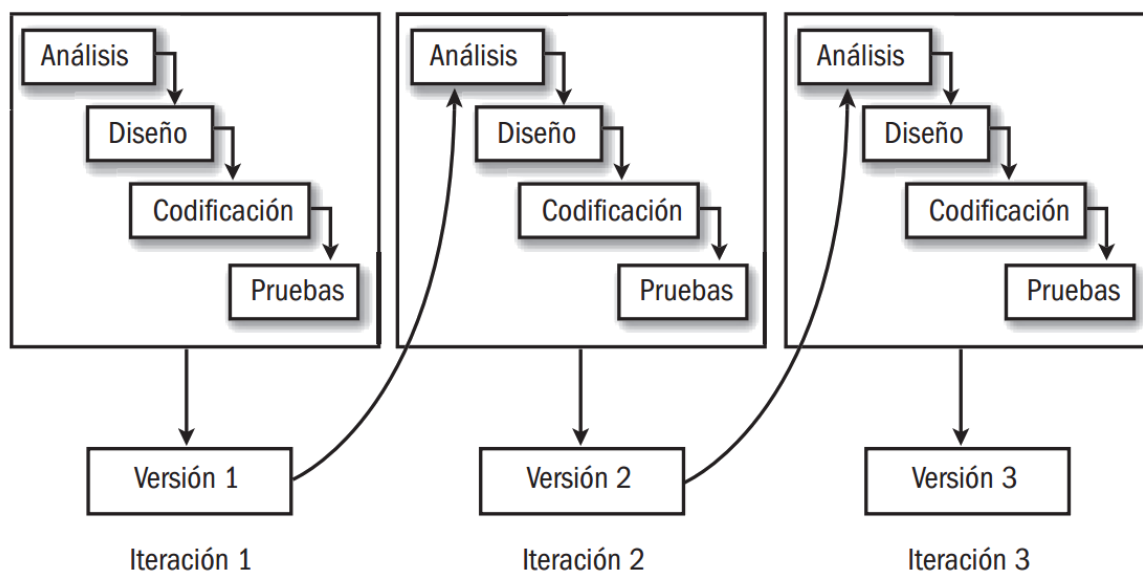
Características	Modelo Espiral	Modelo Incremental	Modelo Iterativo	Modelo Prototipado
-----------------	----------------	--------------------	------------------	--------------------

Está basado en etapas o fases que contemplen la integración completa del proyecto.	✓	✓	✓	✓
Desde el principio hasta el final de cada iteración al cliente se le entrega una versión mejorada del producto	✓	✓	✓	
El producto se va mejorando conforme se vaya refinando el sistema		✓	✓	✓
Por lo general, los requisitos en algún momento van a cambiar			✓	
Los requisitos se determinan de manera prototipada para ser revisada por los usuarios	✓		✓	✓
Se realizan entregas parciales del sistema que permiten incorporar nuevos requisitos o cambios en el sistema	✓	✓	✓	✓
Cada versión es una mejora sobre la predecesora		✓	✓	

Fuente: Elaboración propia.

El presente capítulo trata cada una de las Fases para el Desarrollo de una App Móvil que permite la detección y sugerencia de tratamientos para enfermedades en cultivos de maíz, según el Ciclo de Vida Iterativo como son: La Fase de Análisis en donde se analizará los requerimientos funcionales y no funcionales. La Fase de Diseño donde se realizará el Procesamiento Digital de Imágenes y el Diseño de Interfaces. La Fase de Implementación que es concretamente la fase de programación y por último la Fase de Pruebas donde se realiza las pruebas respectivas con su debida documentación.

Figura 9: Ciclo de Vida Iterativo



Fuente: Puello (2012).

## 4.1. Fase de Análisis

### 4.1.1. Requerimientos Funcionales

Tabla 11: Requerimientos Funcionales.

Id	Descripción
RF-01	La aplicación debe ser capaz de registrar el país con su respectiva región del usuario que desee usar la aplicación.
RF-02	La aplicación debe ser capaz de validar cuando el usuario acepte los términos y condiciones de uso de la aplicación, de lo contrario, no podrá usar la aplicación.
RF-03	La aplicación debe ser capaz de mostrarle sugerencias al usuario de cómo tomar una buena foto a la planta.
RF-04	La aplicación debe ser capaz de mostrar al usuario una pantalla con el botón "Comenzar" para que al presionar active la cámara del teléfono móvil y así pueda capturar una o varias tomas de la planta.
RF-05	La aplicación debe ser capaz de enviar la imagen a su contenedor para que sea evaluada y analizar si está o no enferma.
RF-06	La aplicación debe ser capaz de brindar sugerencia de tratamientos si se encontró la planta enferma.

Fuente: Elaboración propia.

## 4.1.2. Requerimientos No Funcionales

### 4.1.2.1. Rendimiento

- La aplicación ofrecerá respuesta al usuario en tiempo real.
- El tiempo de respuesta promedio de la aplicación no debe superar los 2 segundos.

### 4.1.2.2. Disponibilidad

- La aplicación estará disponible al 100% del tiempo, ya que se trata de una aplicación que se instalará en el dispositivo móvil.
- La aplicación no dependerá de una conexión a internet ni un plan de datos.
- La aplicación estará disponible en Google Play Store.

### 4.1.2.3. Usabilidad

- La aplicación tendrá buena usabilidad, será fácil e intuitiva.

### 4.1.2.4. Mantenibilidad

- La aplicación estará en constante mantenimiento ya que se podría realizar modificaciones o correcciones.

### 4.1.2.5. Portabilidad

- Compatibilidad con plataformas: En el sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.0, ya sea en una Tablet o Smartphone.

## 4.2. Fase de Diseño

### 4.2.1. Diseño de Interfaces

Se realizarán diseños breves que sirven de referencia para la implementación. Se usará el software Balsamiq Mockups 3, ya que la interacción de sus prototipos no va más allá de enlaces entre pantallas.

#### 4.2.1.1. Prototipo de la Pantalla de Registrar País/Región

Nuestro primer prototipo será la pantalla de Registrar País/Región la cual es muy simple ya que solo requiere de una ventana.

Figura 1. Prototipo registro.



Fuente: Elaboración propia.

#### 4.2.1.2. Prototipo de la Pantalla del Botón “Comenzar”

Nuestro segundo prototipo será la pantalla del Botón “Comenzar”, para que cuando el usuario presione dicho botón, se active la cámara del teléfono móvil y empiece a capturar una o varias tomas, según el usuario.

Figura 2. Prototipo bienvenido.



Fuente: Elaboración propia.

#### 4.2.1.3. Prototipo de la Pantalla Principal

Nuestro tercer prototipo será la Pantalla Principal, en donde aparecen dos pestañas llamadas: Inicio y Mis Imágenes. En la primera pestaña. En la segunda pestaña se mostrarán todas las imágenes capturadas por el usuario. Asimismo, aparece el icono de una Cámara, para realizar una nueva captura. Finalmente, aparecen distintos botones como son: Inicio, Comunidad, Estación (Clima) y Librería.

Figura 3. Prototipo pantalla principal.





Fuente: Elaboración propia.

#### 4.2.1.4. Prototipo de la Pantalla de Comunidad

Nuestro cuarto prototipo será la pantalla de Comunidad, donde los usuarios que deseen pertenecer a nuestra comunidad, deben registrarse con su Gmail y crear su contraseña. Finalmente, ingresarlos en los campos correspondientes y ya estarían formando parte. Si es que no desearan formar parte, pueden presionar en el botón "Saltar" y regresaría a la pantalla Principal.

Figura 4. Prototipo registro.



Fuente: Elaboración propia.

#### 4.2.2. Procesamiento Digital de Imágenes

En el presente punto, se realiza la comparación de técnicas en las etapas correspondientes del Procesamiento Digital de Imágenes, que se utiliza en el desarrollo de la presente tesis, para ello se usará una escala de 1 al 5 según la siguiente consideración:

Tabla 12: Escala de calificación para la comparación de técnicas.

1	2	3	4	5
<b>Malo</b>	Regular	Normal	Bueno	Excelente

Fuente: Elaboración propia.

##### 4.2.2.1. Adquisición de la Imagen

El proceso de captura se refiere a la adquisición de la imagen del mundo físico. La imagen puede ser capturada o generada de varias maneras:

fotográficamente, con elementos químicos; o a través de dispositivos electrónicos como el computador, escáner, cámara digital o video-cámara digital.

En la actualidad, una imagen puede ser capturada mediante el uso de diferentes dispositivos de adquisición, los cuales dentro de los fines a los que queremos llegar para la obtención y/o adquisición de una imagen que nos permita realizar una correcta digitalización de la captura obtenida, es que hemos optado por utilizar la cámara móvil.

Como se ha descrito en el apartado 2.2.3.4 parte A, una imagen puede ser capturada mediante el uso de 3 dispositivos de Adquisición. Los criterios de análisis para ésta etapa fueron definidos en la tabla n°2.

Tabla 13: Cuadro comparativo de los dispositivos de captura de una imagen digital basado en puntajes.

Dispositivos	Cámara Digital	Escáner	Cámara Móvil
<b>Criterios</b>			
<b>Resolución</b>	5	4	4
<b>Memoria</b>	5	2	4
<b>Sistema de transferencia</b>	5	3	4
<b>Velocidad de Captura</b>	5	1	5
<b>Total</b>	20	10	17

Fuente: Elaboración propia.

Según los resultados obtenidos, en el cuadro anterior, la cámara digital sería la que debe usarse sin embargo, el dispositivo que se usará para la adquisición de imágenes en el presente proyecto será: **Cámara Móvil**, porque es mucho más accesible y no consume recursos de internet.

#### 4.2.2.2. Pre Procesamiento

La siguiente etapa es el pre procesamiento, que se realiza con el fin de detectar y eliminar las fallas que puedan existir en la imagen para mejorarla. Las técnicas a utilizar son:

**Técnicas de binarización por umbralización:**

En este proceso, se pasa la imagen de escala de grises a una imagen blanca y negra (unos y ceros) mediante una comparación a umbral. Este umbral define el valor límite del pixel siendo negro si está por debajo del umbral y su valor es 0, o blanco si está por encima y su valor es 1.

#### Técnicas de reducción de ruido:

Tabla 14: Cuadro comparativo de las técnicas de reducción de ruido basado en puntajes.

Criterios	Dispositivos Apertura(Erosión, Dilatación)	Filtro de Mediana	Filtro Gaussiano
Preserva bordes	3	3	3
Reduce ruido impulsivo	3	2	4
Suavizado uniforme	2	2	4
<b>Total</b>	<b>8</b>	<b>7</b>	<b>11</b>

Fuente: Elaboración propia

En base a los resultados obtenidos, la técnica de reducción de ruido a emplear es el **Filtro gaussiano**, mediante la cual se buscó eliminar el ruido y difuminar la imagen. Asimismo según el cuadro comparativo descrito en el apartado 2.2.3.4 parte B este sería el mejor algoritmo a emplear.

#### 4.2.2.3. Segmentación

En esta etapa, la imagen se divide en sus partes constituyentes u objetos con el fin de separar las partes necesarias de procesamiento del resto de la imagen que no interesan de acuerdo a la aplicación que se quiera dar.

La técnica seleccionada es la detección de bordes.

Tabla 15: Cuadro basado en características de algoritmos de detección de bordes.

Segmentación – Detección de bordes				
Características	Roberts	Prewitt	Sobel	Canny

Aplica la derivada de primer orden.	✓	✓	✓	✓
Aplica la derivada de primer orden y la de segundo orden.			✓	✓
Realiza un rastreo horizontal y vertical.	✓	✓	✓	✓
Realiza un rastreo horizontal, vertical y hacia otras direcciones.	✓			
Permite discriminar los valores en que el gradiente no es máximo.				✓

Fuente: Elaboración Propia.

Conforme a la tabla 8, el algoritmo más adecuado para la etapa de segmentación es: **Canny**. Asimismo, en el apartado 2.2.3.4 parte C se aprecia que el algoritmo ofrece mejores resultados para bordes de imágenes.

#### 4.2.2.4. Descripción o extracción de características:

Esta etapa, consiste en extraer características con alguna información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otra.

Para elegir el tipo de descripción, se utilizó el siguiente cuadro comparativo:

Tabla 16: Cuadro comparativo de los algoritmos de extracción de características.

Algoritmos	Descriptor de frontera según su longitud	Descriptor de región según su área
Dependencia de cantidad de píxeles en el objeto	3	4
Baja complejidad computacional	3	3
Homogeneidad en el objeto	1	2
<b>Total</b>	<b>7</b>	<b>9</b>

Fuente: Elaboración propia.

De acuerdo al cuadro comparativo anterior, el algoritmo más adecuado para la etapa de descripción es: **Descriptor de región según su área**, asimismo en el apartado 2.2.3.4 parte D se puede confirmar que es uno de los métodos más rápidos y eficaces.

#### 4.2.2.5. Reconocimiento e Interpretación

A continuación, se evalúan los criterios para la etapa de reconocimiento.

Tabla 17: Cuadro comparativo de los algoritmos de reconocimiento e interpretación.

Algoritmos	Decisión teórica para el reconocimiento	Hamming	Reconocimiento estadístico de patrones
Comparación de descriptores de objetos	4	2	3
Búsqueda por proximidad	3	2	2
<b>Total</b>	<b>7</b>	<b>4</b>	<b>5</b>

Fuente: Elaboración propia.

En la etapa de reconocimiento e interpretación, se eligió la técnica: **Decisión Teórica para el reconocimiento.**

## 4.3. Fase de Desarrollo

### 4.3.1. Implementación de Fragments

#### 4.3.1.1. Fragment: Welcome

Este fragment hace referencia a la primera pantalla que aparecerá en la Aplicación. Es una pantalla de Bienvenida.

```
public class WelcomeFragment extends Fragment {
    private Callbacks callbacks;
    private FloatingActionButton button_check;

    public interface Callbacks {
        void showPermCamAlmFragment();
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        callbacks = (Callbacks) context;
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
        container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_welcome, container, false);

        button_check = (FloatingActionButton)
            v.findViewById(R.id.floatingActionButton2);
        button_check.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                callbacks.showPermCamAlmFragment();
            }
        });
        return v;
    }

    @Override
    public void onDetach() {
        super.onDetach();
        callbacks = null;
    }
}
```

#### 4.3.1.2. Fragment: Iniciar Sesión

Este fragment hace referencia a la pantalla de Iniciar Sesión. Es decir, la pantalla donde se tendrá que seleccionar la opción Saltar para que se pueda ingresar a realizar el Procesamiento Digital de Imágenes.

```
public class IniciarSesionFragment extends Fragment {
    private Callbacks callbacks;
    Button button_iniciar_sesion, button_skip;
    TextView textView_registrar;

    public interface Callbacks {
        void showPlantixActivity();
        void showRegistrarFragment();
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        callbacks = (Callbacks) context;
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
        container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_iniciar_sesion, container,
            false);

        button_iniciar_sesion = (Button) v.findViewById(R.id.button3);
        button_skip = (Button) v.findViewById(R.id.button6);
        textView_registrar = (TextView) v.findViewById(R.id.textView6);

        button_iniciar_sesion.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view) {
                callbacks.showPlantixActivity();
            }
        });
        button_skip.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                callbacks.showPlantixActivity();
            }
        });
        textView_registrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                callbacks.showRegistrarFragment();
            }
        });

        return v;
    }
}
```



```
@Override
public void onDetach() {
    super.onDetach();
    callbacks = null;
}
}
```

#### 4.3.1.3. Fragment: Abrir Galería

Este fragment hace referencia a la opción de Abrir galería para Insertar una imagen que posteriormente será evaluada.

```
public void abrirGaleria(View v){
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(
        Intent.createChooser(intent, "Seleccione una imagen"),
        SELECT_FILE);
}
protected void onActivityResult(int requestCode, int resultCode,
    Intent imageReturnedIntent) {
    super.onActivityResult(requestCode, resultCode, imageReturnedIntent);
    Uri selectedImageUri = null;
    Uri selectedImage;

    String filePath = null;
    switch (requestCode) {
        case SELECT_FILE:
            if (resultCode == Activity.RESULT_OK) {
                selectedImage = imageReturnedIntent.getData();
                String selectedPath=selectedImage.getPath();
                if (requestCode == SELECT_FILE) {

                    if (selectedPath != null) {
                        InputStream imageStream = null;
                        try {
                            imageStream = getContentResolver().openInputStream(
                                selectedImage);
                        } catch (FileNotFoundException e) {
                            e.printStackTrace();
                        }
                    }

                    // Transformamos la URI de la imagen a inputStream y este
                    // a un Bitmap
                    bmp = BitmapFactory.decodeStream(imageStream);
                    int nh = (int) ( bmp.getHeight() * (512.0 / bmp.getWidth()) );
                    bmp= Bitmap.createScaledBitmap(bmp, 512, nh, true);

                    Utils.bitmapToMat(bmp, mImagenReal);
                    // Ponemos nuestro bitmap en un ImageView que tengamos
                    // en la vista
                    mImg = findViewById(R.id.img);
                }
            }
        }
    }
}
```

```
mImg.setImageBitmap bmp);
```

```

    }
  }
}
break;
}
}

```

## 4.3.2. Implementación de Funciones

### 4.3.2.1. Binarización

```

private void binarizar()
{
    double ratio=this.bmp.getWidth()/this.bmp.getHeight();
    int valor1=1;
    int valor2=1;
    int valor3=1;
    int valor4=1;
    int valor5=1;
    int valor6=1;
    SeekBar sb1=findViewById(R.id.valor1);
    SeekBar sb2=findViewById(R.id.valor2);
    SeekBar sb3=findViewById(R.id.valor3);
    SeekBar sb4=findViewById(R.id.valor4);
    SeekBar sb5=findViewById(R.id.valor5);
    SeekBar sb6=findViewById(R.id.valor6);
    valor1=sb1.getProgress();
    valor2=sb2.getProgress();
    valor3=sb3.getProgress();
    valor4=sb4.getProgress();
    valor5=sb5.getProgress();
    valor6=sb6.getProgress();
    int area=500000;
    double altura=mImagenReal.size().width;
    Mat imagenRetoque=new Mat();
    Mat imagenRetoque2=new Mat();
    double anchura=mImagenReal.size().height;
    Size size= new Size(3,3);

    Scalar minVerde = new Scalar(valor1, valor2, valor3);
    Scalar maxVerde = new Scalar(valor4/*40*/, valor5, valor6);
    Utils.bitmapToMat bmp, mImagenReal);
    Imgproc.resize(mImagenReal, mImagenRealRedimensionada, size);

    Log.d("valor 1 ",String.valueOf(valor1));
    Log.d("valor 2 ",String.valueOf(valor2));
    Log.d("valor 3 ",String.valueOf(valor3));
    Log.d("valor 4 ",String.valueOf(valor4));
    Log.d("valor 5 ",String.valueOf(valor5));
    Log.d("valor 6 ",String.valueOf(valor6));
    Mat threshold_output= new Mat();

```

```

List<Mat> IstCanalesHSV;
TextView txt=findViewById(R.id.txt);
txt.setText("Minimo color " + String.valueOf(valor1) + ", "
+String.valueOf(valor2)+", "+String.valueOf(valor3) + " Maximo color " +
String.valueOf(valor4) + ", "
+String.valueOf(valor5)+", "+String.valueOf(valor6) );
IstCanalesHSV=new ArrayList<>();

```

```

Imgproc.GaussianBlur(this.mImagenReal, this.mImagenSuavizada, size
,0,0, BORDER_DEFAULT );
Imgproc.cvtColor(this.mImagenSuavizada, this.mImagenHSV,
Imgproc.COLOR_BGR2HLS);
Core.inRange(mImagenHSV, minVerde, maxVerde, imagenRetoque);
showImg(mImagenSuavizada);
Core.split(mImagenHSV,IstCanalesHSV);
showImg2(mImagenHSV);
Mat gris=new Mat();

Imgproc.cvtColor(mImagenHSV,gris,Imgproc.COLOR_RGB2GRAY);

Core.inRange(gris, minVerde, maxVerde, this.mImagenBinarizada);

showImg2(mImagenBinarizada);
}

```

#### 4.3.2.2. Segmentación

```

private void segmentar(){

    edge = new Mat();
    dst = new Mat();

    Imgproc.Canny(this.mImagenBinarizada, edge, 80, 90 );
    Imgproc.cvtColor(edge, dst, Imgproc.COLOR_GRAY2RGBA, 4);

    showImg7(dst);

}

```

#### 4.3.2.3. Descripción

```

private void descripcion(){
    List<MatOfPoint> contours;
    contours = new ArrayList<>();
    Mat mHierarchy = new Mat();
    Imgproc.findContours(mImagenBinarizada, contours, mHierarchy,
    Imgproc.RETR_LIST,Imgproc.CHAIN_APPROX_SIMPLE);

    for ( int i=0; i < contours.size(); i++){
        Scalar color = new Scalar( 0,0,255);
        Imgproc.drawContours( mImagenSuavizada, contours, i, color, 2, 8,
        mHierarchy, 0, new Point() );
    }
}

```

```
        showImg9(mImagenSuavizada);
    }
```

#### 4.3.2.4. Reconocimiento

```
private void reconocimiento(){
    List<MatOfPoint> contours;
    contours = new ArrayList<>();
    Mat mHierarchy = new Mat();
    Imgproc.findContours(mImagenBinarizada, contours, mHierarchy,
    Imgproc.RETR_TREE,Imgproc.CHAIN_APPROX_SIMPLE);

    for ( int i=0; i < contours.size(); i++){

        Imgproc.drawContours(mImagenSuavizada, contours, i, new
        Scalar(255,0,0), -1);
    }

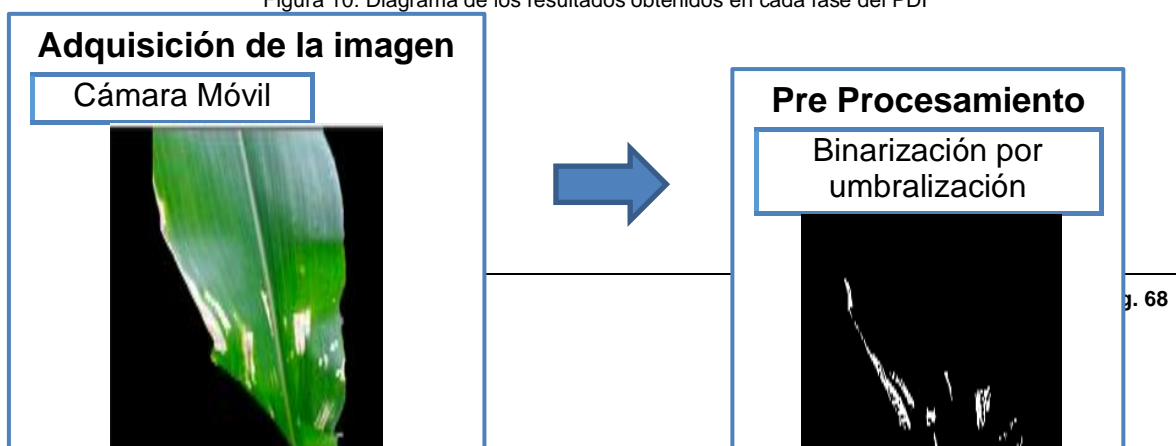
    showImg11(mImagenSuavizada);
}
```

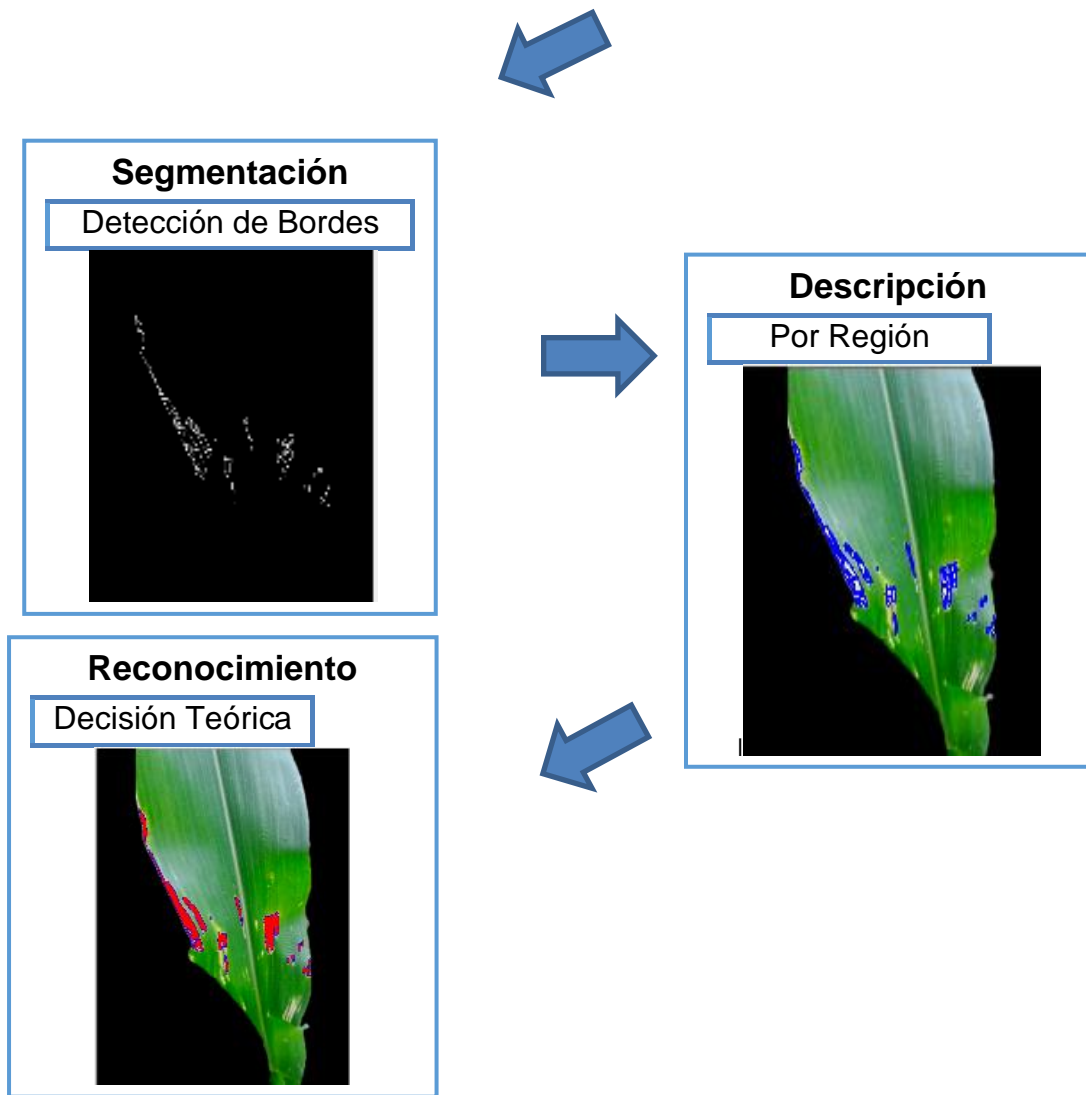
#### 4.3.2.5. Contenedor de Imagen para cada proceso

```
private void showImg(Mat img) {
    final Bitmap bm = Bitmap.createBitmap(img.cols(),
    img.rows(),Bitmap.Config.RGB_565);
    Utils.matToBitmap(img, bm);
    final ImageView imageView = findViewById(R.id.img);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            imageView.setImageBitmap(bm);
        }
    });
}
```

## 4.4. Fase de Pruebas

Figura 10: Diagrama de los resultados obtenidos en cada fase del PDI





Fuente: Elaboración Propia

## CAPÍTULO 5. METODOLOGÍA

### 5.1. Diseño de investigación

Experimental: pre – experimental

Medición	Variable	Medición
$O_1$	X	$O_2$

Dónde:

X: Variable independiente.

$O_1$ : Medición pre-experimental de la variable independiente.

$O_2$ : Medición post-experimental de la variable dependiente.

## 5.2. Unidad de estudio

Cada uno de los cultivos de maíz sembrados en un surco en la Huerta “Roque Ángulo” de la Provincia de Ascope.

## 5.3. Población

Todos los cultivos de maíz sembrados en un surco, lo que corresponde a 105 plantas de maíz amarillo duro.

## 5.4. Muestra

Fórmula para determinar la muestra en una población finita:

$$n = \frac{NZ^2PQ}{(N - 1)E^2 + Z^2PQ}$$

Donde:

N = tamaño de la población. = 105

Z = nivel de confianza 1.96: coeficiente de confianza para el 95% de nivel de confianza.

P = probabilidad a favor: 0.70.

Q = probabilidad en contra: 0.30.

E = error muestral. 5%= 0.05.

$$n = \frac{105 * 1.96^2 * 0.7 * 0.3}{(105 - 1)0.05^2 + 1.96^2 * 0.7 * 0.3}$$

$$n = 79.41$$

Resultado: El tamaño de la muestra es de 79 plantas de maíz amarillo duro.

## 5.5. Técnicas, instrumentos y procedimientos de recolección de datos

Tabla 18: Cuadro de técnicas e instrumentos de recolección de datos.

VARIABLE	DIMENSIONES
DEFINICIÓN CONCEPTUAL	INDICADORES
Observación	<b>Ficha de observación:</b> Se usará la ficha de observación para registrar información acerca de los indicadores como el nivel de sensibilidad, especificidad, precisión y desempeño que puede tener la aplicación al enfrentarse a diferentes entornos.

Fuente: Elaboración propia.

## 5.6. Métodos, instrumentos y procedimientos de análisis de datos

Tabla 19: Cuadro de métodos y procedimientos de análisis de datos.

VARIABLE	INDICADOR	PROCEDIMIENTO	FÓRMULA
Aplicación móvil	Nivel de sensibilidad	Se usa la matriz de confusión para este indicador. Asimismo, se usan los verdaderos positivos y se divide con la suma entre los verdaderos positivos y falsos negativos de la matriz de confusión.	$S = \frac{VP}{VP + FN}$ <p>Dónde: S = Sensibilidad VP = Verdaderos Positivos FN = Falsos Negativos</p>

	Nivel de especificidad	Se usa la matriz de confusión para este indicador. Asimismo, se usan los verdaderos negativos y se divide con la suma entre los verdaderos negativos y falsos positivos de la matriz de confusión	$P = \frac{VN}{VN + FP}$ <p>Dónde: P = Especificidad VN = Verdaderos Negativos FP = Falsos Positivos</p>
	Nivel de precisión	Detalles de precisión	$Precisión = \frac{VP}{VP + VF}$ <p>Dónde: VP = Tasa de verdaderos positivos. VF = Tasa de falsos positivos.</p>
<b>Detección y sugerencia de tratamientos para enfermedades en cultivos de maíz</b>	Nivel de desempeño	Es la suma entre los verdaderos positivos y verdaderos negativos dividido entre el total de la muestra.	$D = \frac{VP + VN}{VP + FN + FP + VN}$ <p>Dónde: VP = Verdaderos Positivos VN = Verdaderos Negativos FN = Falsos Negativos FP = Falsos Positivos</p>

Fuente: Elaboración propia

## CAPÍTULO 6. RESULTADOS

En el presente capítulo, se exponen los resultados obtenidos, al usar la “Aplicación Móvil que permite la detección y sugerencia de tratamientos para enfermedades en cultivos de maíz”, la cual fue implementado en el lenguaje de programación de Android. La Aplicación mencionada, fue ejecutada en un teléfono móvil marca Samsung J5, con un procesador Exynos 7080 1.6GHz y con una memoria RAM de 2GB. (Ver imagen de interfaz de la app en Anexo N° 01).

### 6.1. Análisis de Indicadores

Para verificar la correcta funcionalidad de la “Aplicación Móvil que permite la detección y sugerencia de tratamientos para enfermedades en cultivos de maíz”, se realizaron experimentos con las 79 plantas de maíz amarillo duro, dadas como muestra. El análisis de los indicadores de la presente tesis, dependen de las posibles situaciones que se pueden



obtener después de procesar la detección de enfermedades de las plantas de maíz en la Aplicación, estas situaciones se definen a continuación:

- **Verdadero Positivo (VP):** Evento favorable, en el que la Aplicación detecta y sugiere tratamientos para todas las plantas de maíz enfermas, sin error alguno.
- **Verdadero Negativo (VN):** Evento en que la Aplicación no detecta ni sugiere tratamientos para las plantas de maíz que están sanas. Se le conoce como el Número de Rechazos Correctos.
- **Falso Positivo (FP):** Casos en que la Aplicación detecta enfermedad y sugiere tratamientos erróneamente para las plantas de maíz que están sanas.
- **Falso Negativo (FN):** Corresponde a los casos en que la Aplicación no detecta la enfermedad ni sugiere tratamientos para las plantas de maíz enfermas.

Los resultados de los experimentos realizados con las 79 plantas de maíz amarillo duro, se clasificaron en Verdaderos Positivos, Verdaderos Negativos, Falsos Negativos y Falsos Positivos; obteniendo 67 plantas de maíz amarillo duro como Verdaderos Positivos (84.8%), 8 plantas de maíz amarillo duro fueron Verdaderos Negativos (10.1 %), 0 plantas de maíz amarillo duro fueron Falsos Negativos (0%) y 4 plantas de maíz amarillo duro fueron Falsos Positivos (5.1 %) (Ver Cuadro Anexo n°04). La siguiente gráfica muestra la clasificación en porcentajes con respecto del total de muestra. (Ver gráfico en Anexo n°05)

#### 6.1.1. Índice de Sensibilidad

Es la capacidad de la Aplicación Móvil, para detectar enfermedades y sugerir tratamientos en las plantas de maíz amarillo duro. Se puede obtener en porcentaje, aplicando la siguiente ecuación (Ver gráfico en Anexo n° 05):

$$S = \frac{VP}{VP + FN} = \frac{67}{67 + 0} \times 100 = 100 \%$$

#### 6.1.2. Índice de Especificidad

Es la capacidad de la Aplicación Móvil, para detectar una planta que no está enferma y no le asigne un tratamiento. (Ver gráfico en Anexo n° 05):

$$P = \frac{VN}{VN + FP} = \frac{8}{8 + 4} \times 100 = 66.67 \%$$

#### 6.1.3. Precisión

Es la proporción de casos clasificados correctamente como positivos. (Ver gráfico en Anexo n°05):

$$P = \frac{VP}{VP + FP} = \frac{67}{67 + 4} = 94.37\%$$

#### 6.1.4. Desempeño

Es la proporción del número total de predicciones que fueron correctas, calculada como (Ver gráfico en Anexo n°05):

$$D = \frac{VP + VN}{VP + FN + FP + VN} = \frac{67 + 8}{67 + 0 + 4 + 8} = 94.94\%$$

## CAPÍTULO 7. DISCUSIÓN

En la presente investigación y los resultados alcanzados. En el nivel de Sensibilidad según los valores esperados, es mayor a 81.69%, y se alcanzó un 100% de Sensibilidad en las pruebas experimentales, indicando que aumentó totalmente en comparación a la tesis “Desempeño de un Sistema Inmune Artificial en la Identificación de Plantas”, y no hubo ningún caso donde la aplicación no detecte la enfermedad ni sugiera tratamientos para las plantas de maíz enfermas.

En el nivel de especificidad se alcanzó con la aplicación un 66.67% el cual servirá como un valor base en futuras investigaciones. El valor obtenido es producto que en la aplicación solo hubo 8 casos en los que se detectó la enfermedad en plantas de maíz que estén sanas.

Además, en el nivel de precisión se alcanzó un 94.37%., frente al 83.07% de la tesis de Meléndez lo cual indica un aumento de 11.3%, debido a que la aplicación detecto correctamente 67 plantas enfermas, y sugirió el tratamiento para las mismas.

Finalmente, se midió la exactitud obteniendo un valor de 94.94%; comparándolo con la tesis de Meléndez donde se alcanzó el 80.92%; por lo cual se analizó que existe un aumento favorable de 14.02%. La razón del incremento es debido a que cada valor de positivos y negativos en la matriz de confusión fueron superiores e inferiores respectivamente.

## CONCLUSIONES

- Se verificó la veracidad de la hipótesis planteada y además se cumplieron los objetivos trazados, el cual nos ayudó a desarrollar la Aplicación Móvil para detectar enfermedades en cultivos de maíz y sugerir tratamientos.
- La Aplicación Móvil fue capaz de identificar correctamente la enfermedad en los cultivos de maíz con una sensibilidad del 100%.
- La Aplicación Móvil alcanzó un nivel de especificidad del 66.67%.
- La Aplicación Móvil fue capaz de asegurar la detección correcta de la enfermedad en los cultivos de maíz con una precisión de 94.37%.
- La Aplicación Móvil fue capaz de asegurar al usuario el correcto desempeño para la identificación de enfermedad en los cultivos de maíz, es por ello que se midió a través de la exactitud, obteniendo un valor de 94.94%.

## RECOMENDACIONES

Las recomendaciones que pueden dar lo autores de la presente investigación son:

- Se sugiere incrementar nuevas características al sistema propuesto como la detección de nuevas enfermedades, que conllevaría a incrementar la diversidad de usos de la solución propuesta.
- Se recomienda el uso de redes neuronales, con el fin de identificar con más certeza la plaga y dar un correcto tratamiento.
- Se sugiere investigar nuevos algoritmos que puedan ayudar a detectar enfermedades y sugerir tratamientos en los cultivos de maíz.
- Se recomienda evaluar la influencia que tienen ciertos factores (textura, ambiente) presentes en las imágenes de las hojas, para poder tener un análisis más preciso en las imágenes a evaluar.

## REFERENCIAS

Alambiaga, J (2015). *Desarrollo de una aplicación móvil para la detección y clasificación de hojas de árboles*. (Tesis de Grado). Universidad Politecnica de Valencia, Gandia.

- Alarcia, B (2014). *Detección automática de objetos de interés en imágenes aéreas*. Recuperado de: <https://addi.ehu.es/bitstream/handle/10810/13353/PFC%20SanMartin.pdf?sequence=2>
- Ambrosio, G., Arévalo, V. & Gonzáles, J. (2004). *La Librería de Visión OpenCV*. (Paper). Universidad de Málaga, España.
- Asociación de Marketing Móvil de España. (2011). *Blanco de Apps*. Madrid: MMA.
- CENTA (2014). *Guía para identificar el complejo mancha de asfalto en el cultivo de maíz en el Salvador*. Recuperado de: [http://centa.gob.sv/docs/guias/granos%20basicos/GUIA%20MANCHA%20ASFALTO%20MAIZ.pdf?fbclid=IwAR2arXM4JvH\\_uNa4y7ybBd9souTwp9DjLk1C5YLUzKN1XZ\\_VQ1pRX2p4OY](http://centa.gob.sv/docs/guias/granos%20basicos/GUIA%20MANCHA%20ASFALTO%20MAIZ.pdf?fbclid=IwAR2arXM4JvH_uNa4y7ybBd9souTwp9DjLk1C5YLUzKN1XZ_VQ1pRX2p4OY)
- Ciesla, W., Cadahía, D., Robredo, F. (1976). *La detección de plagas de insectos y enfermedades forestales*. Recuperado de: [https://www.mapama.gob.es/ministerio/pags/Biblioteca/Revistas/pdf\\_plagas%20FBSVP-02-01-037-053.pdf](https://www.mapama.gob.es/ministerio/pags/Biblioteca/Revistas/pdf_plagas%20FBSVP-02-01-037-053.pdf)
- Cobo, A. (2007). *Diseño y Programación de Bases de Datos*. Recuperado de: [https://books.google.com.pe/books?id=anCDr9NkGsC&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.pe/books?id=anCDr9NkGsC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Cortés, R. (s.f.). *Introducción al Análisis de Sistemas y la Ingeniería de Software*. Recuperado de: <https://books.google.com.pe/books?id=Y2CCT0flxYwC&printsec=frontcover&hl=es#v=onepage&q&f=false>
- Cuello, J y Vitone, J. (2013). *Diseñando Apps para Móviles*. [En línea]. Recuperado el 03 de octubre del 2017 en: <http://appdesignbook.com/es/contenidos/presentacion/>
- CYMMYT (s.f.). *Complejo mancha de asfalto del maíz: Hechos y acciones*. [En línea]. Recuperado de: [http://conservacion.cimmyt.org/en/component/docman/doc\\_download/922-tarspot](http://conservacion.cimmyt.org/en/component/docman/doc_download/922-tarspot)
- Del Castillo, E (2018). *Desarrollo de un Sistema de Visión Artificial para realizar una clasificación uniforme de limones*. (Tesis de Grado). Universidad Privada del Norte, Perú.
- Deras, H. (2014). *El cultivo del maíz*. El Salvador: Instituto Interamericano de Cooperación para la Agricultura.
- Domínguez Jalili, L. (2015). *Desarrollo de un Sistema de Identificación y Reconocimiento de Plantas Usando Técnicas de Aprendizaje Máquina*. (Tesis de Grado). Universidad Autónoma del Estado de México, México.
- Dupont Pioner (2014). *Mancha de asfalto*. Recuperado de: [https://www.pioneer.com/CMRoot/International/Mexico\\_Intl/Agronomia/Articulos\\_PDF/SE\\_5B\\_MANCHA\\_ASFALTO\\_2014.pdf](https://www.pioneer.com/CMRoot/International/Mexico_Intl/Agronomia/Articulos_PDF/SE_5B_MANCHA_ASFALTO_2014.pdf)
- Esepe (2005). *¿Qué es my SQL?* Recuperado de: <https://www.espestudio.com/noticias/que-es-mysql>
- Fariño, R. (2011). *Modelo Espiral de un proyecto de desarrollo de software*. Recuperado de: <http://www.ojovisual.net/galofarino/modeloespiral.pdf>

- Franquesa, M. (2016). *El Cultivo del maíz*. Recuperado el 15 de octubre del 2017 en:  
<https://www.agroptima.com/blog/cultivo-del-maiz/>
- García, G. (2013). *Manejo Integrado de Plagas y Enfermedades en el Cultivo de Maíz Amiláceo Blanco*. Recuperado de: <https://www.agrobanco.com.pe/data/uploads/ctecnica/022-d-mab.pdf>
- Gómez, D y Guerrero, A (2016). *Estudio y Análisis de Técnicas para Procesamiento Digital de Imágenes*. (Tesis de Grado). Universidad Tecnológica de Pereira, Colombia.
- Instituto Colombiano Agropecuario (2007). *Enfermedades del maíz y su manejo*. Palmira: Grupo Transferencia de Tecnología.
- Instituto Nacional de Tecnologías de la Comunicación (2009). *Curso de Introducción a la Ingeniería de Software*. Recuperado el 30 de agosto del 2018 en:  
<https://jmpovedar.files.wordpress.com/2014/03/curso-de-introduccion-a-la-ingenieria-del-software.pdf>
- ISO (2018). *ISO/IEC 25010*. Recuperado de: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Joyanes, L. (2013). *Big Data: Análisis de Grandes Volúmenes de Datos en Organizaciones*. Recuperado de:  
<https://books.google.com.pe/books?id=1GywDAAAQBAJ&pg=PT223&dq=mongodb&hl=es&sa=X&ved=0ahUKEwjYhNWCxPzdAhXFwFkKHUYkDzIQ6AEIRDAG#v=onepage&q=mongodb&f=false>
- Meléndez Ruiz, L. (2016). *Desempeño de un Sistema Inmune Artificial en la Identificación de Plantas*. (Tesis de Grado). Universidad Peruana del Oriente, Perú.
- Meneses, J. (2016). *Control químico y comportamiento de híbridos de maíz (Zea mays L.) a la enfermedad denominada mancha de asfalto*. Recuperado de:  
<http://repositorio.lamolina.edu.pe/bitstream/handle/UNALM/2709/H20-M455c-T.pdf?sequence=1&isAllowed=y>
- MINAGRI (2008). *Plan Estratégico Regional del Sector Agrario 2009-2015*. Recuperado de:  
[http://minagri.gob.pe/portal/download/pdf/conocenos/transparencia/planes\\_estrategicos\\_regionales/lalibertad.pdf](http://minagri.gob.pe/portal/download/pdf/conocenos/transparencia/planes_estrategicos_regionales/lalibertad.pdf)
- Organización de las Naciones Unidas para la Alimentación y la Agricultura (2018). *Plagas y enfermedades de las plantas*. [En línea]. Recuperado el 31 de agosto del 2018 en:  
<http://www.fao.org/emergencias/tipos-de-peligros-y-de-emergencias/plagas-y-enfermedades-de-las-plantas/es/>
- Paliwal, R. (2001). *El maíz en los trópicos: mejoramiento y producción*. Recuperado de  
[https://books.google.com.pe/books?id=os79dx6BcmsC&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.pe/books?id=os79dx6BcmsC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Puello, O. (2012). *Ciclo de vida*. Recuperado de:  
<http://singsufps.blogspot.com/search/label/Ingenier%C3%ADa%20de%20Software>

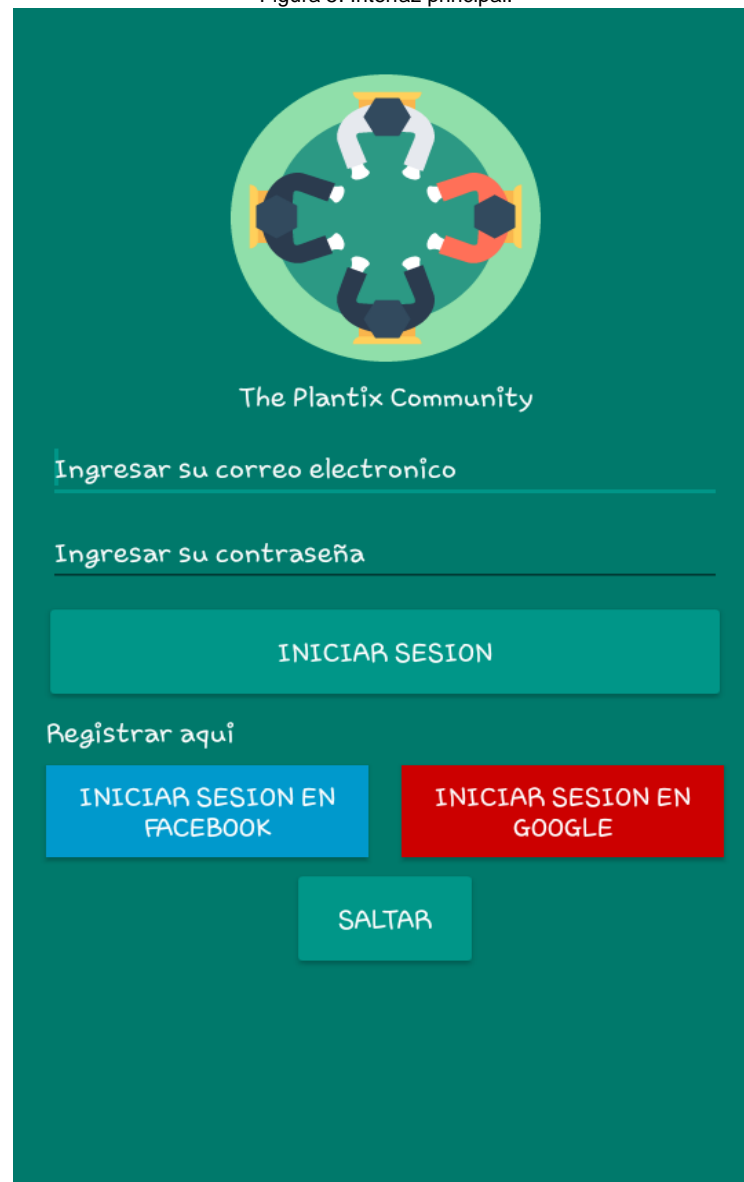
- Quiroga,R. , Garrido, E., Rosales,M, & Salazar,W. (2017). Manejo integrado del complejo mancha de asfalto del maíz en México. Recuperado de: [https://www.researchgate.net/profile/Ricardo\\_Quiroga-Madrigal/publication/317929566\\_Manual\\_tecnico\\_Manejo\\_integrado\\_del\\_complejo\\_mancha\\_de\\_asfalto\\_del\\_maiz\\_en\\_Mexico/links/5951bcb20f7e9b329234d6c0/Manual-tecnico-Manejo-integrado-del-complejo-mancha-de-asfalto-del-maiz-en-Mexico.pdf](https://www.researchgate.net/profile/Ricardo_Quiroga-Madrigal/publication/317929566_Manual_tecnico_Manejo_integrado_del_complejo_mancha_de_asfalto_del_maiz_en_Mexico/links/5951bcb20f7e9b329234d6c0/Manual-tecnico-Manejo-integrado-del-complejo-mancha-de-asfalto-del-maiz-en-Mexico.pdf)
- REDESA (2006). *Manejo integral de plagas: Guía para pequeños productores agrarios*. Recuperado de: [http://www.care.pe/pdfs/cinfo/libro/Manejo\\_de\\_Plagasfinal.pdf](http://www.care.pe/pdfs/cinfo/libro/Manejo_de_Plagasfinal.pdf)
- Robledo, C y Robledo, D. (2011). *Programación en Android*.Madrid: Aula Mentor.
- Ruíz, L. (2004). *Agricultura General*. Paraguay: DEA-MAG.
- SENASA (2015). *Informando sobre el Desarrollo Agrario y la Seguridad Alimentaria*. En Revista Comunicándonos n°6. Perú: Servicio Nacional de Sanidad Agraria. Recuperado de: <https://www.senasa.gob.pe/senasa/descargasarchivos/2016/04/BOLETIN-6.pdf>
- Sommerville, I. (2011). *Ingeniería de Software*. Recuperado de: <https://ingenieriadesoftwareutmachala.files.wordpress.com/2017/01/ingenieria-de-software-somerville.pdf>
- Sucar, L y Gómez, G. (s.f.). *Visión Computacional*. Puebla: Instituto Nacional de Astrofísica, Óptica y Electrónica.
- SQLite (2018). *SQLite*. Recuperado de: <https://www.sqlite.org/about.html>
- Troya García, R (2011). *Evaluación de Cepas de Nucleopoliedrovirus (NPV) Patógenos para el control del cogollero Spodopterafrugiperda en maíz en la zona de Babahoyo*. (Tesis de Grado). Universidad Técnica de Babahoyo, Ecuador.



## ANEXOS

### ANEXO n.º 1. Interfaz de Inicio de la App

Figura 5. Interfaz principal.



Fuente: Elaboración propia.

### ANEXO n.º 2. Ficha de Análisis Mancha de Asfalto

Tabla 20: Análisis Pre Experimental

Captura	Sana	Enferma
C1.		✓
C2.		✓
C3.		✓
C4.		✓
C5.		✓
C6.		✓
C7.		✓
C8.		✓
C9.		✓
C10.		✓
C11.		✓
C12.		✓
C13.	✓	
C14.	✓	
C15.		✓
C16.		✓
C17.		✓
C18.		✓
C19.		✓
C20.		✓
C21.		✓
C22.		✓
C23.		✓
C24.		✓
C25.		✓
C26.		✓
C27.		✓
C28.		✓
C29.		✓

---

C30.	✓
C31.	✓
C32.	✓
C33.	✓
C34.	✓
C35.	✓
C36.	✓
C37.	✓
C38.	✓
C39.	✓
C40.	✓
C41.	✓
C42.	✓
C43.	✓
C44.	✓
C45.	✓
C46.	✓
C47.	✓
C48.	✓
C49.	✓
C50.	✓
C51.	✓
C52.	✓
C53.	✓
C54.	✓
C55.	✓
C56.	✓
C57.	✓
C58.	✓
C59.	✓
C60.	✓
C61.	✓

---

---

C62.	✓
C63.	✓
C64.	✓
C65.	✓
C66.	✓
C67.	✓
C68.	✓
C69.	✓
C70.	✓
C71.	✓
C72.	✓
C73.	✓
C74.	✓
C75.	✓
C76.	✓
C77.	✓
C78.	✓
C79.	✓

---

Fuente: Elaboración Propia

### **ANEXO n.º 3. Distribución de resultados de la Matriz de Confusión aplicada en la App Móvil**

Tabla 21: Cuadro de Distribución de Resultados de la Matriz de Confusión aplicada en la App Móvil

Captura	Verdadero positivo	Verdadero Negativo	Falso Positivo	Falso Negativo
C1.	✓			
C2.	✓			
C3.	✓			
C4.	✓			
C5.	✓			
C6.	✓			
C7.	✓			
C8.	✓			
C9.	✓			
C10.	✓			
C11.	✓			
C12.	✓			
C13.			✓	
C14.		✓		
C15.	✓			
C16.	✓			
C17.	✓			
C18.	✓			
C19.	✓			
C20.	✓			
C21.	✓			
C22.	✓			
C23.	✓			
C24.	✓			
C25.	✓			
C26.	✓			
C27.	✓			
C28.	✓			
C29.	✓			
C30.	✓			
C31.	✓			
C32.	✓			
C33.	✓			
C34.	✓			
C35.	✓			
C36.	✓			
C37.	✓			
C38.	✓			
C39.	✓			
C40.	✓			
C41.	✓			
C42.	✓			
C43.				✓

---

C44.	✓		
C45.	✓		
C46.	✓		
C47.	✓		
C48.	✓		
C49.	✓		
C50.	✓		
C51.			✓
C52.	✓		
C53.	✓		
C54.	✓		
C55.	✓		
C56.	✓		
C57.	✓		
C58.	✓		
C59.	✓		
C60.	✓		
C61.	✓		
C62.	✓		
C63.	✓		
C64.	✓		
C65.	✓		
C66.	✓		
C67.	✓		
C68.	✓		
C69.	✓		
C70.		✓	
C71.			✓
C72.			✓
C73.		✓	
C74.		✓	
C75.			✓
C76.		✓	
C77.		✓	
C78.		✓	
C79.		✓	

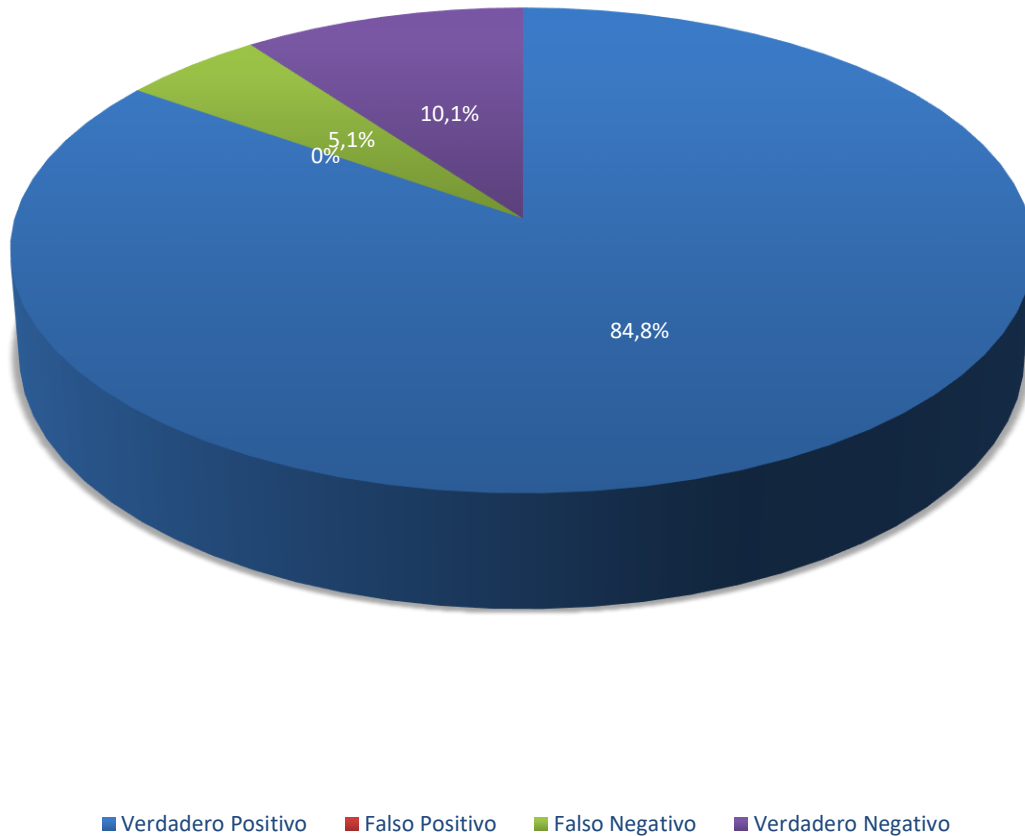
---

Fuente: Elaboración propia

## ANEXO n.º 4.

Gráfico 1. Clasificación de Enfermedades en Plantas de Maíz Amarillo Duro

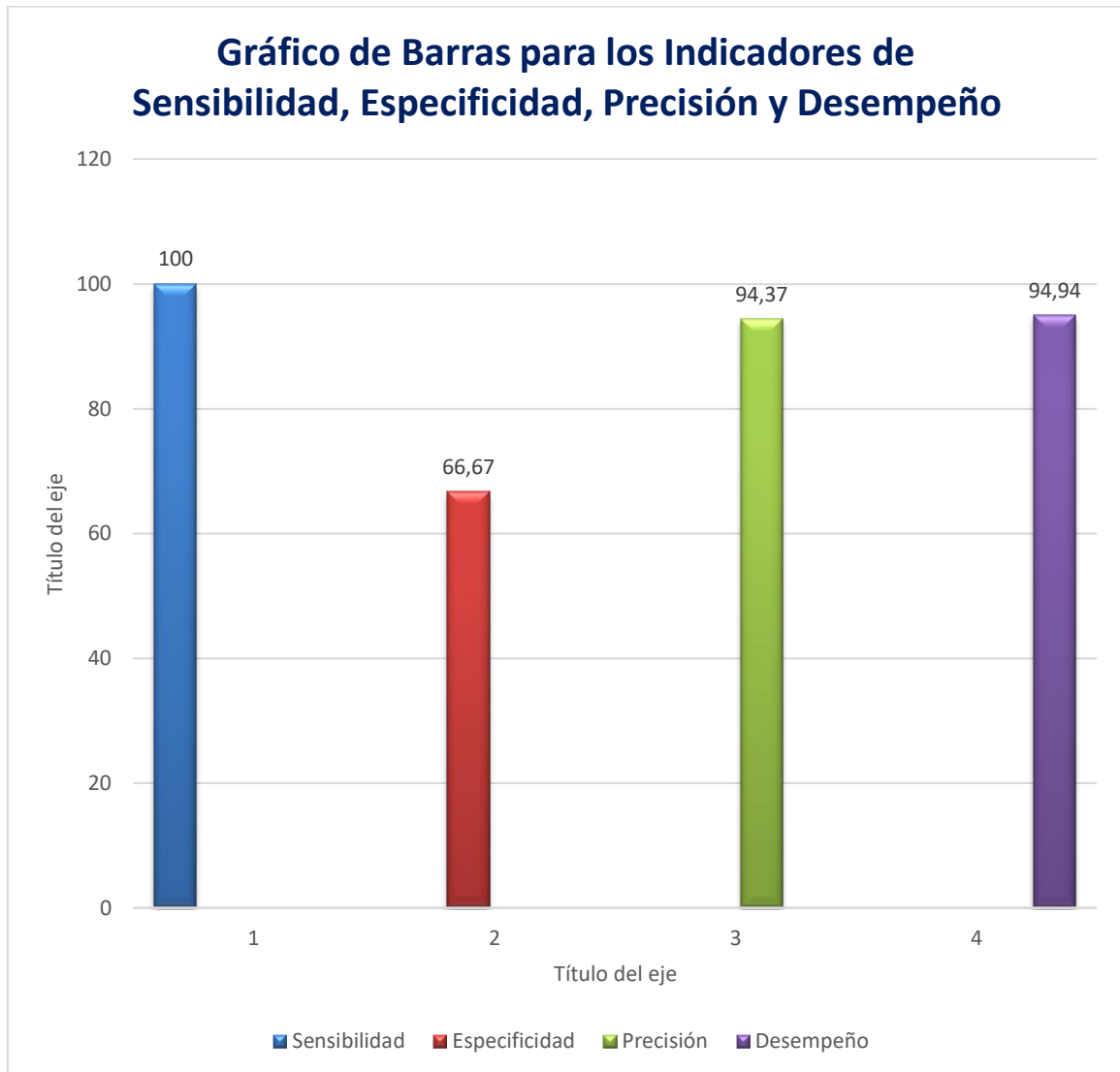
### Clasificación de Enfermedades en Plantas de Maíz Amarillo Duro



Fuente: Elaboración Propia

### ANEXO n.º 5.

Gráfico 2. Indicadores de Sensibilidad, Especificidad Precisión y Desempeño


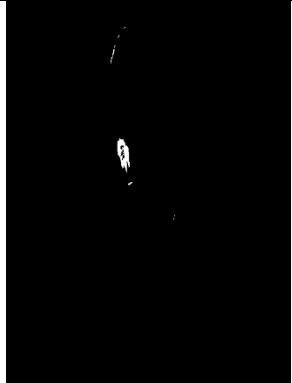


Fuente: Elaboración Propia



## ANEXO n.º 6. Pruebas de la aplicación

Prueba 1:





Imagen Entrada	Valores	Imagen Binarizada
	157 160 223	

Prueba 2:

Imagen Entrada	Valores	Imagen Binarizada
	157 160 223	

Prueba 3:

Imagen Entrada	Valores	Imagen Binarizada
	157 160 223	

## ANEXO n.º 7. ACTA DE CONFORMIDAD

Yo, Omar Colán Garay, en calidad de Biólogo con especialidad en Manejo Integrado de Plagas y Enfermedades Agrícolas, doy conformidad de haber realizado el Análisis Pre Experimental del ANEXO n.º 2. Ficha de Análisis Mancha de Asfalto donde evalué 79 imágenes de hojas de plantas de maíz detectando así la clasificación de estas entre sanas y enfermas.

Este análisis se desarrolló para fines académicos.

  
183/7375