

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE INGENIERÍA EN INFORMÁTICA

Tema: “Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020.”

Trabajo de titulación previa la obtención del
Título de Ingeniero en Informática

AUTOR(A): Palacios Julio Leandro Leodan

TUTOR(A): Ing. Naranjo Cedeño Jeffery Alex Mgs

Tulcán, 2021

CERTIFICADO JURADO EXAMINADOR

Certificamos que el estudiante Palacios Julio Leandro Leodan con el número de cédula 1004234470 ha elaborado el trabajo de titulación: “Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020.”

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de Titulación, Sustentación e Incorporación de la UPEC, por lo tanto, autorizamos la presentación de la sustentación para la calificación respectiva.

f.....
Ing. Naranjo Cedeño Jeffery Alex Mgs
TUTOR

f.....
Ing. Miranda Realpe Jorge Humberto Mgs
LECTOR

Tulcán, diciembre de 2021

AUTORÍA DE TRABAJO

El presente trabajo de titulación constituye requisito previo para la obtención del título de **Ingeniero** en la Carrera de ingeniería en informática de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Palacios Julio Leandro Leodan con cédula de identidad número 1004234470 declaro: que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

f.....

Palacios Julio Leandro Leodan

AUTOR(A)

Tulcán, diciembre de 2021

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

Yo, Palacios Julio Leandro Leodan declaro ser autor/a de los criterios emitidos en el trabajo de investigación: “Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020” y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes legales de posibles reclamos o acciones legales.

f.....

Palacios Julio Leandro Leodan

AUTOR(A)

Tulcán, diciembre de 2021

AGRADECIMIENTO

Mis agradecimientos, para los técnicos y agricultores de la comunidad de Pusir Grande por la atención brindada en medio de sus actividades diarias, por el excelente trato y cordialidad en sus participaciones.

A la Universidad Politécnica Estatal del Carchi, por permitirme formar parte de su prestigioso campus universitario a lado de excelentes figuras profesionales que transmitieron sus conocimientos y valores sin limitaciones.

A la carrera de Ingeniería en informática, por ayudarme a forjar el conocimiento y cumplir uno de los anhelos de mi proyecto de vida.

A mis docentes académicos, por ser partícipes en mi formación, por compartir su conocimiento, y por sus consejos éticos y morales compartidos dentro y fuera del aula de clases.

A mis amigos y compañeros del salón de clases, por su carisma, por su perseverancia y por formar parte de esta foto del álbum de mi vida.

A mi familia, por creer en mí y ayudarme a consolidar uno de mis objetivos.

DEDICATORIA

Dedicado a la memoria de mi padre, por ser mi figura de éxito, perseverancia y trabajo constante, por sembrar en mí la semilla de humildad y compromiso, por forjar mi carácter y personalidad.

A la memoria de María Marisol Palacios Julio, por ser la pionera en el cumplimiento de este objetivo de mi plan de vida.

A mi madre, por enseñarme a sobreponerme de las adversidades, por orientarme en el camino del bien y forjarme como un hombre guerrero y creyente.

A mi hija Alanis Palacios, por llegar a mi vida en medio del camino académico y recordarme el verdadero significado de la palabra responsabilidad.

A Jhonson García, Angie Ramírez y Juan David Sapuyes, por ser parte de esta etapa y por estar conmigo y ayudarme a enfrentar las adversidades durante el arduo camino universitario.

A mi familia, por su constante apoyo moral.

ÍNDICE

I. PROBLEMA	20
1.1. Planteamiento del problema	20
1.2. Formulación del problema.....	21
1.3. Justificación	21
1.4. Objetivos y preguntas de investigación	23
1.4.1. Objetivo General.....	23
1.4.2. Objetivos Específicos	23
1.4.3. Preguntas de Investigación	23
II. FUNDAMENTACIÓN TEÓRICA	24
2.1. Antecedentes investigativos	24
2.2. Marco teórico.....	26
2.2.1 Introducción a la inteligencia Artificial (IA)	26
2.2.2 Técnicas de inteligencia artificial.	26
2.2.2.1 Machine Learning.....	26
2.2.2.2 Deep Learning	28
2.2.2.3 Contraste entre Machine Learning y Deep Learning	29
2.2.3 Técnicas de procesamiento de imagen tradicional	31
2.2.3.1 Métodos en el dominio espacial.	32
2.2.3.2 Métodos en el dominio de la frecuencia.....	32
2.2.3.3 Métodos de extracción de características.	33
2.2.3.4 Algoritmos de detección de objetos.....	34
2.2.4 Procesamiento de imágenes.....	34
2.2.4.1 Características del procesamiento de imágenes.....	35
2.2.4.2 Componentes del procesamiento de imágenes.....	36
2.2.5 Metodología de gestión de proyectos.	41

2.2.5.1 Comparación entre metodología ágil y tradicional.....	41
2.2.5.2. Metodología de proceso incremental.....	42
2.2.6 Norma Técnica ISO/IEC 15504 SPICE.....	44
2.2.6.1 Características.....	44
2.2.7 Planeación del entrenamiento.....	45
2.2.7.1 Niveles de planeación.....	45
2.2.8 Herramientas físicas para el procesamiento.....	45
2.2.9 Herramientas de software para el procesamiento.....	45
2.2.9.1. Librerías para el preprocesamiento de imágenes.....	46
2.2.9.2. TensorFlow.....	46
2.2.9.3 TensorFlow Lite.....	47
2.2.10 Requerimientos para el aplicativo.....	47
2.2.10.2 Lenguajes de programación métodos y algoritmos de procesamiento.....	48
2.2.11 Enfermedades del pimiento (<i>Capsicum annuum</i>).....	48
2.2.12 Clasificación de las enfermedades.....	48
2.2.12.1 Enfermedades de hoja.....	49
2.2.12.2 Enfermedades de raíz.....	49
2.2.12.3 Enfermedades de tallo.....	49
2.2.12.4. Identificación de las enfermedades.....	49
III. METODOLOGÍA.....	53
3.1. ENFOQUE METODOLÓGICO.....	53
3.1.1. Enfoque.....	53
3.1.2. Tipo de Investigación.....	53
3.1.2.1. Investigación Bibliográfica.....	53
3.1.2.2. Investigación Aplicada.....	53
3.1.2.3. Investigación de campo.....	54
3.2. Idea a defender.....	54

3.3. Definición y operacionalización de variables.....	54
3.3.1. Definición De Variables	54
3.4. MÉTODOS UTILIZADOS	57
3.4.1. Método conceptual deductivo.....	57
3.4.2. Método de la Investigación-Acción Participativa (IAP)	57
3.4.3. Método de reconstrucción de hechos.....	57
3.4.4. Análisis Estadístico	57
3.4.4.1. Población.	57
3.4.4.2. Muestra.	58
3.5. Técnicas	59
3.5.1. Encuesta.....	59
3.5.2. Entrevista estructurada	59
3.5.2. Observación Directa	59
IV. RESULTADOS Y DISCUSIÓN.....	60
4.1. RESULTADOS	60
4.1.1. Análisis E Interpretación De Resultados	60
4.1.1.1. Análisis de resultados de la encuesta.....	60
4.1.1.2. Análisis de la entrevista aplicada a técnicos agrónomos.....	69
4.1.2. Resultados identificación con procesamiento tradicional	71
4.1.3. Resultados identificación con procesamiento a base de inteligencia artificial.....	73
4.1.4. Evaluación del análisis	76
4.1.5. Propuesta	76
4.1.5.1. Situación Actual	76
4.1.5.2. Situación Ideal	77
4.1.6 Introducción.....	77
4.1.7. Metodología de proceso incremental.....	77
4.1.7.1. Fase de análisis	77

4.1.7.2. Fase de diseño.....	78
4.1.7.3. Fase de codificación	91
4.1.7.4. Pruebas funcionales	114
4.1.7.5. Pruebas de aceptación.....	117
4.1.7.6. Pruebas de comparativas	117
4.2. DISCUSIÓN.....	119
V. CONCLUSIONES Y RECOMENDACIONES	120
5.1. CONCLUSIONES.....	120
5.2. RECOMENDACIONES	121
IV. REFERENCIAS BIBLIOGRÁFICAS	122
V. ANEXOS	126

ÍNDICE DE FIGURAS

Figura 1. Principal diferencia del Machine Learning y Deep Learning	30
Figura 2: Secuencia del procesamiento de imágenes.	35
Figura 3: Ejemplo de procesamiento con algoritmo de segmentación.....	35
Figura 4: Aspectos que conforman una imagen digital.....	36
Figura 5: Píxel resultante a partir de la intensidad de sus componentes	37
Figura 6: Representación del píxel de una imagen digital a color transformada.	37
Figura 7: Fórmula del modelo HSV	40
Figura 8: Fórmula del modelo HSI.....	41
Figura 9: Fases del modelo incremental.....	42
Figura 10: Fases de la metodología de proceso incremental.....	43
Figura 11: Ejemplo de enfermedad con afectación en hoja	49
Figura 12: Análisis de resultado pregunta 1	60
Figura 13: Análisis de resultado pregunta 2	61
Figura 14: Análisis de resultado pregunta 3	62
Figura 15: Análisis de resultado pregunta 4	63
Figura 16: Análisis de resultado pregunta 5	64
Figura 17: Análisis de resultado pregunta 6	65
Figura 18: Análisis de resultado pregunta 7	66
Figura 19: Análisis de resultado pregunta 8.....	67
Figura 20: Análisis de resultado pregunta 9.....	68
Figura 21: Análisis de resultado pregunta 10.....	69
Figura 22: Imagen obtenida mostrada en Python OpenCV	71
Figura 23: Normalización de contraste	72
Figura 24: Aplicación de filtro de mediana sobre contraste normalizado.....	72
Figura 25: Segmentado de la imagen	72
Figura 26: Aplicación del algoritmo ORB para obtención de características	73
Figura 27: Dataset para en entrenamiento y validación del modelo	73
Figura 28: Proceso de entrenamiento del modelo	74
Figura 29: Continuación del proceso de entrenamiento del modelo.....	75
Figura 30: Gráficas de entrenamiento y validación del modelo.....	75
Figura 31: Visión global del diseño.....	78
Figura 32: Diseño de la integración del modelo entrenado a Android.....	78
Figura 33: Creación de cuenta en 000webhost y base de datos online	79

Figura 34: Interfaces de usuario	79
Figura 35: Continuación interfaces de usuario.....	80
Figura 36: Diseño interfaces de administración.....	81
Figura 37: Continuación diseño de interfaces de administración	81
Figura 38: Proceso administrador	82
Figura 39: Proceso agricultor	82
Figura 40: Modelo físico base de datos.....	83
Figura 41: Inicio del sistema de detección de enfermedad	85
Figura 42: Creación de cuenta de usuario/agricultor	86
Figura 43: Registro de cultivo al sistema.....	86
Figura 44: Identificación de la enfermedad	87
Figura 45: Recomendación tratamiento / agricultor.....	88
Figura 46: Interfaz de ingreso a la administración del sistema administración	88
Figura 47: Interfaz de creación de un nuevo administrador.....	89
Figura 48: interfaz de registro de una nueva enfermedad	89
Figura 49: Panel de administración, gestión a base de datos	90
Figura 50: Caso de uso gestión de usuarios / administración.....	90
Figura 51: Caso de usos usuarios del sistema/agricultores	91
Figura 52: Subida y extracción del dataset de las imágenes del cultivo	91
Figura 53: Definición de las clases del modelo de transfer learning entrenamiento y validación	92
Figura 54: Importación de paquetes y activación del modelo VGG16 de machine learning..	92
Figura 55: Traspaso de capas entrenadas del modelo VGG16 al modelo a entrenarse	93
Figura 56: Compilación del modelo transferido	93
Figura 57: Preprocesado de las imágenes del dataset en las clases de entrenamiento y validación	93
Figura 58: Reentrenar el modelo transferido con las nuevas clases añadidas.....	94
Figura 59: Conversión de modelo a extensión compatible con Android y guardar.....	94
Figura 60: Codificación clase java login.....	95
Figura 61: Continuación codificación clase java login	95
Figura 62: Continuación codificación clase java login	96
Figura 63: Continuación codificación clase java login	96
Figura 64: Codificación de clase java registro de usuarios/agricultor	97
Figura 65: Continuación codificación clase java registro de usuario/agricultor.....	97

Figura 66: Continuación codificación clase java registro de usuario/agricultor	98
Figura 67: Continuación codificación clase java registro de usuario/agricultor	98
Figura 68: Codificación clase java registro cultivo.....	98
Figura 69: Continuación codificación clase java registro cultivo	99
Figura 70: Codificación clase java registro cultivo.....	99
Figura 71: Codificación clase java registro cultivo.....	99
Figura 72: Codificación clase java tratamiento.....	100
Figura 73: Continuación codificación clase java tratamiento	100
Figura 74: Continuación codificación clase java tratamiento	101
Figura 75: Codificación clase java identificación de enfermedad	101
Figura 76: Continuación codificación clase java identificación de enfermedad.....	101
Figura 77: Continuación codificación clase java identificación de enfermedad.....	102
Figura 78: Continuación codificación clase java identificación de enfermedad.....	102
Figura 79: Codificación vista login usuario/agricultor	103
Figura 80: Continuación codificación vista login usuario/agricultor.....	103
Figura 81: Continuación codificación vista login usuario/agricultor.....	104
Figura 82: Continuación codificación vista login usuario/agricultor.....	104
Figura 83: Continuación codificación vista login usuario/agricultor.....	105
Figura 84: Codificación vista registro de usuario/agricultor.....	105
Figura 85: Continuación codificación vista registro de usuario/agricultor	106
Figura 86: Continuación codificación vista registro de usuario/agricultor	106
Figura 87: Continuación codificación vista registro de usuario/agricultor	107
Figura 88: Continuación codificación vista registro de usuario/agricultor	107
Figura 89: Continuación codificación vista registro de usuario/agricultor	108
Figura 90: Codificación vista registro cultivo.....	108
Figura 91: Continuación codificación vista registro cultivo	109
Figura 92: Continuación codificación vista registro cultivo	109
Figura 93: Continuación codificación vista registro cultivo	110
Figura 94: Codificación vista tratamiento.....	110
Figura 95: Continuación codificación vista tratamiento	111
Figura 96: Continuación codificación vista tratamiento	111
Figura 97: Continuación codificación vista tratamiento	112
Figura 98: Codificación vista identificación de enfermedad	112
Figura 99: Continuación codificación vista identificación de enfermedad.....	113

Figura 100: Continuación codificación vista identificación de enfermedad.....	113
Figura 101: Continuación codificación vista identificación de enfermedad.....	114

ÍNDICE DE TABLAS

Tabla 1. Características de arquitecturas de red neuronal convolucional.....	29
Tabla 2: Cuadro comparativo Árboles de decisión y Redes neuronales	30
Tabla 3. Algoritmos de detección de objetos y sus características.....	34
Tabla 4: Cuadro comparativo entre una imagen digital a color y una imagen digital transformada	38
Tabla 5: Cuadro de características de las metodologías de desarrollo de proyectos.....	41
Tabla 6: Cuadro comparativo librerías utilizadas en el procesamiento de imágenes.....	46
Tabla 7: Comparativa entorno de codificación Python local y el servicio de Google - Google Colab.....	47
Tabla 8: Factores importantes de las enfermedades para usar en el aplicativo	52
Tabla 9. Operacionalización de la variable independiente.	55
Tabla 10. Operacionalización de la variable dependiente	56
Tabla 11: Resultado cálculo de muestra.....	58
Tabla 12: Resultados encuesta pregunta 1	60
Tabla 13: Resultados encuesta pregunta 2	61
Tabla 14: Resultado encuesta pregunta 3	62
Tabla 15: Respuesta encuesta pregunta 4.....	63
Tabla 16: Respuesta encuesta pregunta 5.....	64
Tabla 17: Respuesta encuesta pregunta 6.....	65
Tabla 18: Respuesta encuesta pregunta 7.....	66
Tabla 19: Respuesta encuesta pregunta 8.....	67
Tabla 20: Resultado encuesta pregunta 9	68
Tabla 21: Resultado encuesta pregunta 10	69
Tabla 22: Evaluación del análisis de los modelos de identificación	76
Tabla 23: Diccionario de datos Entidad user-agricultor.....	83
Tabla 24: Diccionario de datos entidad Cultivo	84
Tabla 25: Diccionario de datos entidad enfermedad	84
Tabla 26: Diccionario de datos entidad tratamiento.....	84
Tabla 27: Diccionario de datos entidad super usuario.....	84
Tabla 28: Prueba en dispositivos de nivel de API 23 versiones de Android 6.0 – 6.0.1.....	114

Tabla 29: Prueba en dispositivos de nivel de API 24 - 25 versiones de Android 7.0 – 7.1.2	115
Tabla 30: Prueba en dispositivos de nivel de API 26 - 27 versiones de Android 8.0 – 8.1 ..	115
Tabla 31: Prueba en dispositivos de nivel de API 28, 29 versiones de Android 9.0, 10.0....	116
Tabla 32: Prueba en dispositivos de nivel de API 30, 31 versiones de Android 11.0, 12.0..	116
Tabla 33: Aceptación de los agricultores para el aplicativo	117
Tabla 34: Comparativa entre la app resultante de la investigación y plantNet	117
Tabla 35: Segunda comparativa entre la App resultante de la investigación y PlantVillage	118

ÍNDICE DE ANEXOS

Anexo 1: Certificado o Acta del Perfil de Investigación	126
Anexo 2: Certificado del abstract por parte de idiomas	127
Anexo 3: Informe Turnitin	128
Anexo 4: Ficha técnica Identificación de enfermedad	129
Anexo 5: Código Python preprocesado de las imágenes identificación tradicional	130
Anexo 6: Cuestionario encuesta agricultores	131
Anexo 7: Continuación cuestionario encuesta agricultores	132
Anexo 8: Hoja de respuestas cuestionario encuesta agricultores	
Anexo 9: Continuación hoja de respuestas cuestionario encuesta agricultores	134
Anexo 10: Continuación hoja de respuestas cuestionario encuesta agricultores	135
Anexo 11: Continuación hoja de respuestas cuestionario encuesta agricultores	136
Anexo 12: Cuestionario entrevista a técnicos	137
Anexo 13: Ficha técnica con síntomas y sus características	138
Anexo 14: Matriz de evaluación del software	139
Anexo 15: Aplicación de encuesta agricultor fotografía 1	139
Anexo 16: Aplicación de encuesta agricultor fotografía 2	140
Anexo 17: <i>Aplicación de encuesta agricultor fotografía 3</i>	140
Anexo 18: Aplicación de encuesta agricultor fotografía 4	141

RESUMEN

El proyecto de investigación denominado “Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020”, cuyo énfasis se centró en la actividad de identificación de enfermedades. La meta del proyecto fue la elaboración de una herramienta informática que pueda identificar enfermedades en las hojas a partir de una fotografía de la hoja del cultivo, en su desarrollo se contempló los dos tipos de enfoques de investigación, recopilando información a base de entrevistas a técnicos y encuestas a agricultores de la localidad. A partir de los resultados conseguidos y la facilidad de obtener una imagen digital haciendo uso de un dispositivo móvil, se desarrolló un prototipo de app móvil para Android. El desarrollo de la propuesta estuvo ligada a la metodología de proceso incremental cumpliendo cada una de sus fases. En la fase de análisis, por encuesta se obtuvo como resultado que la población objetiva si usarían el aplicativo y que también disponen de un dispositivo con las especificaciones que el software demanda, por lo tanto para las siguientes fases técnicas se procedió con el alojamiento del aplicativo en el hosting gratuito 000webhost usando una base de datos creada en MySQL, para el desarrollo de las vistas se usó el editor de código de Microsoft Android Studio conjuntamente con las clases de java codificadas en el mismo editor. Además, se utilizó la técnica de Inteligencia Artificial transfer learning para reentrenar un modelo Red Neuronal Convolutiva a base de un modelo ya entrenado VGG16, la integración del modelo a Android se la realizó transformado el modelo a la versión de TensorFlow compatible con Android denominada TensorFlow Lite.

Palabras clave: App Móvil, procesamiento de imágenes, Red neuronal, Transferencia de Aprendizaje

ABSTRACT

The research project named "Digital image processing for the identification of leaf diseases in pepper crops from Pusir Grande community in Bolívar canton in the year 2020", whose emphasis was focused on the activity of disease identification. The goal of the project was the development of a computer tool that can recognize diseases in leaves from a photograph of a crop leaf, in its development the two types of research approaches were considered where information based on interviews with technicians and surveys to local farmers was collected. Starting from the results obtained and how easy it is to obtain a digital image using a mobile device, a prototype of a mobile app for Android was developed. The proposal development was linked to the incremental process methodology fulfilling each of its phases. In the analysis phase, per survey it was obtained as a result that the target population would indeed use the application, they also have a device with the specifications that the software requires, therefore for the following technical phases, the accommodation of the application in the 000webhost free hosting using a database created in MySQL was carried out, and for the development of the views, the Microsoft Android Studio code editor was used along with the java classes coded in the same editor. In addition, the Artificial Intelligence transfer learning technique was used to retrain a Convolutional Neural Network model based on an already trained VGG16 model, the integration of the Model to Android was carried out by transforming the model to the Android-compatible version of TensorFlow called TensorFlow Lite.

Key words: mobile app, image processing, Neural Network, transfer learning

INTRODUCCIÓN

Actualmente el cultivo del pimiento está tomando gran posición en el sector agrícola del Valle del Chota, siendo este y otros productos considerados como los responsables del ingreso económico a las familias, no obstante, las enfermedades atacan a este cultivo en niveles exorbitantes, en vista de ellos los agricultores de la localidad buscan estrategias para identificar la afectación y aplicar un correctivo antes de que esta termine con el sembrío en su totalidad. Pusir Grande es una comunidad que pertenece a este sector y al igual que las otras depende de esta actividad en su mayoría, el 80% de la comunidad se dedica al agro por lo que los índices de familias que cultivan pimiento son moderadamente grandes. Además del fenómeno de las enfermedades la comunidad tiene otros agravantes, a diferencia de las otras comunidades que están próximas a la pana vial esta se encuentra alejada y el ingreso a ella se lo realiza por carreteras de tercer orden lo que ocasiona menos visitas técnicas de los agrónomos. Es por ello que la investigación tiene como objetivo principal la identificación temprana de dos de las tres principales enfermedades que atacan a los cultivos de pimiento en esta comunidad, con fundamento en el marco teórico y la metodología que se aplicó para el desarrollo del sistema de detección de enfermedades en las hojas, realizando la integración de tecnología moderna, conocimientos técnicos y experiencia de los agricultores. La importancia del proyecto está en poder identificar la enfermedad en su etapa de inicio ya que es más fácil su corrección. Como resultados obtuvimos el desarrollo de un aplicativo móvil para la identificación de enfermedades, basados en los análisis de los resultados de las encuestas, entrevistas, marco teórico y metodología, donde se definieron las estrategias, métodos y tecnología a utilizar para cumplir el objetivo de la investigación.

I. PROBLEMA

1.1. Planteamiento del problema

En la actualidad las enfermedades que atacan a la planta de pimiento se han presentado de diferentes formas y colores, su afectación se presenta sin importar el tipo de cultivo, volumen de producción y/o ubicación del sembrío. Agricultores de diversos lugares del mundo de alguna u otra forma tuvieron que tratar con el fenómeno. El pimiento es cultivado en cifras elevadas en los principales países del mundo estos son: China, México, Turquía, Indonesia y España, mismos que no están excluidos de este problema. Como lo menciona Jordá (2016) las enfermedades más conocidas que han atacado a los cultivos de pimiento a nivel mundial son 30, las clasificaciones en enfermedades producidas por hongos, por bacterias, por virus y por daños fisiológicos.

Según el nivel de afectación no todas son relevantes, su incidencia es variable según los años y depende de diferentes factores, las condiciones ambientales es uno de ellos, sus síntomas dependen del tipo de bacteria, hongo o virus que la provoque y son más notables en plantas en estado vegetativo temprano. Jordá (2016) afirma que: “podemos encontrar en invernaderos como más frecuentes el Moteado suave del pimiento conocido antiguamente como una raza de virus del Mosaico del Tabaco, Mosaico del Tomate, virus Y de la patata y el virus del Bronceado del tomate” (p. 67).

En Ecuador, la no detección a tiempo de las enfermedades para tomar medidas correctivas ha ocasionado que el cultivo no pueda producir su totalidad, generando un bajón de producción agrícola y con ellos pérdidas económicas. Una investigación realizada en la Universidad Técnica del Norte (UTN) por Carrera (2014) menciona que:

De todas las enfermedades, criptogámicas o bacterianas, que afectan el cultivo de pimiento, la más específica y grave sin ninguna duda es la causada por *Phytophthora capsici* Leo. (P.c.), conocido como la “Tristeza del pimiento” o “Marchitez por phytophthora”, siendo una de las causas más importantes de pérdidas en cultivos de pimiento (p.18).

En la comunidad Pusir Grande ubicada en la parroquia de San Vicente de Pusir cantón Bolívar, se evidenció en padecimiento de enfermedades en sus pequeñas parcelas. El Sr. Sangucho, I. (comunicación personal, 8 de enero del 2020) agricultor de la zona por más 40 años dice que: “En los años pasados el tipo de pimiento cultivado no era de la variedad que se cultiva en la actualidad, pero el problema de las enfermedades siempre ha estado presente, la lucha por controlar a este fenómeno ha sido constante, anteriormente se hacía uso de remedios comunes sin obtener un total correctivo, dice que no identificar a tiempo lo que pasa con el cultivo hasta

22 la actualidad es una agravante debido a que las enfermedades en su estado de madurez son más difíciles de controlar ”.

Así mismo, el Sr. Bolaños, W. (comunicación personal, 8 de enero del 2020) agricultor de la comunidad en cuestión supo manifestar que el rendimiento del cultivo de pimiento en los últimos años ha tenido una evolución considerable comparada a los años pasados, también expresó molestia por las enfermedades en este caso dio prioridad a la ceniza, dice que es la que más ha afectado sus cultivos

1.2. Formulación del problema

¿Cómo el procesamiento de imágenes digitales permite la identificación de las enfermedades de hoja en los cultivos de pimiento, en la Comunidad Pusir Grande del cantón Bolívar en el año 2020?

1.3. Justificación

El pimiento es un producto de importancia económica mundial, es por eso que el manejo y control de las enfermedades debe ser explícito y en el tiempo establecido.

Según Aenverde (2019) en los últimos 10 años la producción de pimiento ha favorecido a los países europeos. Datos del último periodo muestran un crecimiento del 36% en las importaciones de pimiento intraeuropeas, mientras que en los países productores extracomunitarios la importación de este producto ha disminuido en un 19%.

Por su parte Hortoinfo (2018) reveló un total de 34.497 millones de kilos según la publicación del año 2016 de la FAO (Food and Agriculture Organization), misma que posiciona a China como el principal productor de pimiento con 17.435 millones de kilos, lo que representa el 50,34% del rendimiento mundial, seguido por México con 2.737 millones de kilos, Turquía con 2.457, Indonesia con 1.961 y dejando como última en la escala a España con 1.082 millones de kilos.

Se realizó la investigación para personalizar el proceso de identificación de las enfermedades de hoja en cultivos de pimiento de la comunidad de Pusir Grande, en una solución informática con información característica de diferentes estados de la planta, fusionando tecnologías actuales usadas en la agricultura de precisión.

Es importante porque la herramienta es una alternativa de mejora en el sector agropecuario de la comunidad, se presenta como una propuesta de identificación temprana de enfermedades en la hoja del pimiento, utilizando el procesamiento de imágenes digitales aplicado en hojas de plantas que presenten anomalías. El objetivo es que la herramienta informática sea una ayuda para el agricultor cuando este requiera identificar enfermedades, en la actualidad el uso de la

23 tecnología de inteligencia artificial en las actividades agropecuarias está ocupando un gran campo de investigación con sus algoritmos de detección de objetos y clasificación de imágenes. Por tal motivo la propuesta va dirigida directamente a los agricultores de la comunidad de Pusir Grande, misma que su actividad comercial es netamente agrícola, en sus beneficiarios el desaprovechamiento de los avances que ofrece la tecnología en el ámbito agropecuario es evidente y muy pronunciado. Es relevante mencionar que la importancia económica de la comunidad se ve afectada, pues alrededor de 263 familias se dedican al cultivo de pimiento siendo este su principal producto agrícola. Con esta propuesta, se busca acoger la mayor parte de actores de la comunidad que se dedican a esta actividad, es importante mencionar que de estos cultivos subsiste otra parte de la comunidad que se vería beneficiada indirectamente. La originalidad de tema radica en la combinación de diferentes tecnologías vigentes en la actualidad y en agro, además de utilizar herramientas de código abierto que han permitido desarrollar un software sin generar gasto de licencia, otro factor complementario a la unicidad del proyecto es la aplicación al sector agrícola que en la actualidad es muy versátil. Su factibilidad y fiabilidad reside en que la información se la obtuvo desde la realidad y de los actores implicados, gracias al compromiso de los técnicos y agricultores que prestaron completa disponibilidad para colaborar con el proyecto, obteniendo los datos necesarios para la base de datos que posteriormente sería consumida por el aplicativo.

1.4. Objetivos y preguntas de investigación

1.4.1. Objetivo General

Diseñar una herramienta de procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar.

1.4.2. Objetivos Específicos

- Elaborar un marco metodológico para la investigación del procesamiento de imágenes digitales y su colaboración en la identificación de enfermedades de hoja de cultivos de pimiento en el sector agrícola.
- Determinar características de las enfermedades de hoja para la obtención de patrones digitales en las hojas de los cultivos de pimiento.
- Analizar algoritmos, técnicas y tecnologías de inteligencia artificial usadas en el procesamiento de imágenes digitales para la determinación de características comunes de diseño.
- Establecer una propuesta de procesamiento de las imágenes digitales para la identificación de enfermedades de hoja en cultivos de pimiento.

1.4.3. Preguntas de Investigación

- ¿Permite la elaboración del marco metodológico obtener información sobre el procesamiento de imágenes digitales y la colaboración en el sector agropecuario de la comunidad?
- ¿Cómo ayuda la determinación de características de las enfermedades de hoja en la obtención de patrones digitales en las hojas de los cultivos de pimiento?
- ¿Cómo aporta el análisis de algoritmos, técnicas y tecnologías de inteligencia artificial usadas en el procesamiento de imágenes digitales para la determinación de características comunes de diseño?

II. FUNDAMENTACIÓN TEÓRICA

2.1. Antecedentes investigativos

La investigación realizada por Sánchez (2016) en la ciudad de Ambato – Ecuador denominada: “SISTEMA DE MONITOREO AGRÍCOLA CON TECNOLOGÍA INALÁMBRICA Y GENERACIÓN DE ALERTAS PARA LA PREVENCIÓN TEMPRANA DE PLAGAS Y ENFERMEDADES EN EL CULTIVO DE PAPA EN LA PARROQUIA QUIMIAG DEL CANTÓN RIOBAMBA DE LA PROVINCIA DE CHIMBORAZO.”, presenta como objetivo combinar las tecnologías inalámbricas en un sistema de monitoreo agrícola que entregue advertencias para detección temprana de plagas, para llegar a la meta propuesta el autor realizó el análisis de los parámetros que interfieren en la prevención oportuna de plagas en los cultivos y así mismo la determinación de las tecnologías inalámbricas que se podría hacer uso en cumplimiento del objetivo.

En el trabajo de investigación realizado por Casimba (2017) en las ciudades de Otavalo e Ibarra – Ecuador que adopta como tema: “IDENTIFICACIÓN DE PLAGAS Y ENFERMEDADES (HONGOS Y BACTERIAS) DEL CULTIVO DE JÍCAMA (*Smallanthus sonchifolius*) EN EL CANTÓN COTACACHI, OTAVALO E IBARRA –IMBABURA.” Encontramos como meta propuesta por la investigadora llegar a la identificación de las enfermedades en el cultivo de jícama considerando los factores climáticos que presenta el lugar donde se llevó a cabo el proyecto, para ello realizó el análisis de las condiciones climáticas que se presentan considerando los cambios de tiempo, también se aprecia la determinación de las condiciones de incidencia y severidad que presentan las plagas y enfermedades.

Un artículo científico publicado en la revista digital IEEE Xplore por Francis, Dhas & Anoop (2016) denominado: “Identification of leaf diseases in pepper plants using soft computing techniques” o “Identificación de enfermedades foliares en plantas de pimienta mediante técnicas de computación blanda.” en su traducción al español, la investigación tiene como objetivo hacer uso del procesamiento de imágenes digitales y las técnicas de análisis de imágenes, para lo cual realizaron el estudio de las plantas y la caracterización de sus síntomas llevándolos a patrones que puedan ser visualmente observados, también realizaron el análisis en granjas llevando un monitoreo constante de las particularidades de las enfermedades antes de ser analizadas, concluyen que con un algoritmo de procesamiento de imágenes se puede diferenciar las plantas enfermas de las sanadas dejando cultivos más saludables, con este análisis su puede estimar el porcentaje de afectación de enfermedades en granjas completas.

La investigación de Maeda et al. (2018) con el tema :” Redes neuronales convolucionales para la detección y clasificación de enfermedades de plantas basadas en imágenes digitales”, que presenta como objetivo principal el desarrollo de un autómata inteligente que clasifique e identifique las enfermedades en las plantas, haciendo uso de las metodologías de inteligencia artificial y visión por computadora, para conseguir el cumplimiento de esta meta usaron una conjuntos de datos de PlantVillage – Dataset conformado para el entrenamiento de la red neuronal además de la arquitectura AlexNet con estas tecnologías aplicadas el sistema obtuvo una exactitud de respuesta del 98.9%.

El proyecto de investigación realizado por Cusme & Loor (2019) con el tema: “APLICACIÓN MÓVIL DE DETECCIÓN Y CLASIFICACIÓN DE “LA ROYA” EN HOJAS DE CAFÉ ROBUSTA MEDIANTE APRENDIZAJE AUTOMÁTICO” presenta el análisis de algoritmos de detección y clasificación instintiva para la separación de áreas sanas y áreas afectadas en las hojas de los cultivos de café, previo a este análisis los autores realizaron la obtención de la imágenes de las hojas para sobre estas aplicar el procesamiento con el algoritmo seleccionando, acciones que fueron llevadas a cabo para el cumplimiento de su objetivo principal que es la integración de esta tecnología en una app móvil que cumpla con la función de identificación y clasificación de la enfermedad desde un teléfono celular en tiempo real.

La tesis realizada por Román y Ruiz (2021) denominado: “Detección de macronutrientes y enfermedades en campos de cultivo de banano orgánico con Machine Learning” tiene como propósito desarrollar una solución informática a base de inteligencia artificial, entre las actividades para el cumplimiento de esta meta se presenta el análisis de algoritmos de clasificación e identificación y el estudio de los tipos del suelo, características, diferencias y los principales elementos que intervienen en el desarrollo de los micronutrientes, los autores también analizaron las probabilidades de padecimiento del mismo.

2.2. Marco teórico

2.2.1 Introducción a la inteligencia Artificial (IA)

La inteligencia artificial (IA) desde su aparición ha evolucionado de manera considerada en nuestro diario vivir, desde las formas más comunes y simples hasta las más complicadas y compuestas. La humanidad a medida que pasa el tiempo ha ido haciendo uso de esta tecnología inmersa en dispositivos móviles inteligentes, robots etc. El autor (Da Silva, 2021)) en su publicación, menciona a tecnologías como Machine Learning y Deep Learning. La primera que es hija de la teoría que hace hincapié a los equipos computacionales que pueden realizar aprendizaje autónomo sin necesidad de que se los programe para realizar tareas ordenadas, también se basa en el conocido método de reconocimiento de patrones, por su parte Deep Learning está inmerso en Machine Learning es considerado como la versión delta de este, es decir es una evolución de ML, que está inspirado en la percepción humana.

Este algoritmo automático pretende imitar el funcionamiento del cerebro humano y la conexión neuronal.

2.2.2 Técnicas de inteligencia artificial.

En su forma tradicional el procesamiento digital está limitado en su sensibilidad ante perturbaciones exteriores que pueden cambiar la forma de la imagen, es por eso que se ha optado por incluir nuevas técnicas de inteligencia artificial (IA) para mejorar los algoritmos (Reyes, Mejia, & Useche, 2018).

2.2.2.1 Machine Learning

También conocido como aprendizaje autónomo por su traducción del inglés, esta tecnología no necesita de los humanos para cumplir con tareas que no están especificadas dentro de su código de programación, además está orientada al ámbito empresarial, la encuesta de Deloitte aplicada a los ejecutivos de empresas reveló un crecimiento del 82% en sus ganancias después de aplicar la Inteligencia Artificial (Da Silva, 2021). Ejemplo de aplicaciones.

- Reconocimiento de correos de tipo spam
- Reconocimiento de voz en búsquedas de navegadores
- Recomendaciones de amigos, grupos y videos en redes sociales
- Tu ubicación con el GPS
- Clasificación de imágenes

- Características de Machine Learning

- Orientado al reconocimiento de patrones en los datos
- Usado en la manipulación de big data del ámbito empresarial

- Su función se apega a proceso de datos mas no de imágenes
- Cumple tareas especificadas y no especificadas dentro de su programación
- Aplicado para la predicción de oportunidades rentables y riesgos posible

a) Algoritmos de Machine Learning y sus características

Según (Chavero, 2021) Estos están distribuidos en tres categorías:

- Aprendizaje supervisado

Su aprendizaje está basado en sistema de etiqueta que se asocian con los datos esta permite predecir y con ellos tomar decisiones. Por ejemplo, spam que etiqueta correo.

- Aprendizaje no supervisado

Estos buscan patrones y formas de organización debido a que no tienen conocimiento previo. Por ejemplo, en el campo del marketing.

- Aprendizaje por refuerzo

Aprenden por propia experiencia ofrecen mejores resultados sometiendo a los datos al método de prueba y error. Ejemplo de uso reconocimiento facial, diagnósticos médicos o clasificar secuencias de ADN.

b) Árboles de decisión

Según Charris et al (2018), es un proceso de acercamiento de una función objetiva de valores discretos, donde su propósito es representado mediante un árbol de decisión una respuesta posible gracias a los valores que la función va tomando en cada iteración. Los árboles aprendidos también pueden representarse como un conjunto de reglas Si-entonces. Un algoritmo de generación de árboles de decisión consta de 2 etapas: inducción del árbol y clasificación.

1. Se construye el árbol de decisión a partir del conjunto de entrenamiento; comúnmente cada nodo interno del árbol se compone de un atributo de prueba y la porción del conjunto de entrenamiento presente en el nodo es dividida de acuerdo con los valores que pueda tomar ese atributo.
2. Cada objeto nuevo es clasificado por el árbol construido; después se recorre el árbol desde el nodo raíz hasta una hoja, a partir de la que se determina la membresía del objeto a alguna clase. El camino a seguir en el árbol lo determinan las decisiones tomadas en cada nodo interno, de acuerdo con el atributo de prueba presente en él.

- Características

- Su construcción inicia por un nodo raíz, su atributo de prueba es dividido en dos o más subconjuntos.

- En cada una de las particiones se genera un nuevo nodo.
- Muestran similitud con los sistemas de precisión basados en reglas.
- No muestran complejidad al representar los resultados.
- Son robustos y presentan la invariancia en la estructura de sus árboles de clasificación.

2.2.2.2 Deep Learning

Es el aprendizaje profundo desde su traducción de inglés, Machine Learning sería el pionero y Deep Learning sería la evolución mejorada debido a que sus algoritmos tiene la extracción de características inmersa en su código de entrenamiento y su funcionamiento se debe a millones de neuronas y el valor de entradas equivale a la cantidad de pixeles de la imagen (Zambrano, 2019). Ejemplos de aplicaciones

- Pintado fotografías
- Transcripciones automáticas
- Clasificación cosas, individuos y sustancias dentro de una imagen
- Generado etiquetas que describen una imagen

- Características Deep Learning

- Consta de algoritmos súper potentes
- Más orientado a la identificación de imágenes usando los pixeles
- Extracción de características incorporada en algoritmo
- Evolución de Machine Learning
- El número de entrada corresponde a los numero de pixeles de la imagen

a) Redes neuronales artificiales

Las redes neuronales son de dos tipos, fueron diseñadas con el objetivo de imitar el funcionamiento del cerebro humano a base de computadoras, esta tecnología aparece debido a que en la programación tradicional se han presentado numerosas actividades complejas de resolver, es por eso que están orientadas a resolver tareas con altos estándares de complicidad (Negrete, 2021).

Entre sus dos tipos se decidió utilizar la red convolucional (CNNs) con fundamento en sus características y su pronunciado trabajo con imágenes, esta técnica de inteligencia artificial fue utilizada para realizar la identificación de los estados del cultivo, con fundamento en los síntomas que presentan las hojas enfermas durante la etapa de desarrollo de patógeno.

- Características

- Su funcionamiento busca aparentar al cerebro humano.
- Genera sistemas complejos con una organización de compendios simples.

- Las hay de dos tipos multicapa y convolucionales.
- **Características red multicapa MPL (MultiLayer Perceptrón)**
- Sus neuronas están constituidas y dispersas en capas
- Copia cualquier sistema con entradas y salidas conocidas
- Más utilizada en la medicina
- **Características red convolucionales (CNNs)**
- Son aptas para trabajar con imágenes
- Pueden clasificar y reconocer
- Están inspirado en el sistema de aprendizaje biológico
- Permiten el manejo de grandes estructuras de datos
- **Arquitectura de red neuronal**

No necesita la extracción de características de forma manual, esta aprende directamente de los datos que son ingresados en las capas cuando se realiza el entrenamiento del algoritmo. El número de entradas de cada una de estas capas es igual a la cantidad de píxeles de la imagen, entre las arquitecturas más comunes encontramos: AlexNet, VGG16, ResNet, Inception ver tabla 1.

Tabla 1. Características de arquitecturas de red neuronal convolucional

Arquitectura	Exactitud	Valores de pérdida	Aprendizaje	Baja de pesos
AlexNet	83,6%	Un fallo del 16,4%	0,01	0.0005
VGG16	92,7%	Un fallo del 7.3%	0,001	0.00001
ResNet	96%	Un fallo del 3,6%	0,1	0.00001
Inception	78.1%	Un fallo del 28%	0,01	0.0005

Nota: muestra las características más representativas de las arquitecturas de red neuronal de Deep Learning.

Fuente: (Negrete, 2021).

2.2.2.3 Contraste entre Machine Learning y Deep Learning

Para completar el análisis de las técnicas de procesamiento a base de inteligencia artificial presentaremos las diferencias más significativas.

- **Principal diferencia**

Machine Learning necesita de la extracción de características, es decir son obtenidas de forma manual por el programador.

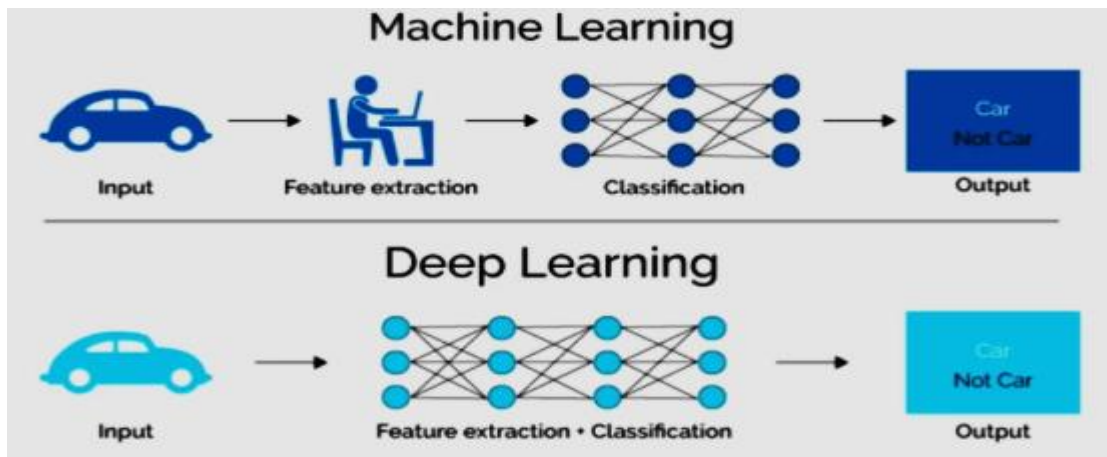


Figura 1. Principal diferencia del Machine Learning y Deep Learning
Fuente: (Chavero, 2021)

- **Otras diferencias**

- Los algoritmos de Deep Learning son más potentes y robustos que los de Machine Learning.
- A Deep Learning únicamente se le pasa como entrada la fotografía, mientras que Machine Learning necesita fotografía y características.
- Deep Learning trabaja con algoritmos de red neuronal.
- Los algoritmos de Machine Learning están basados en árboles de decisión y los de Deep Learning en redes neuronales

Tabla 2: Cuadro comparativo Árboles de decisión y Redes neuronales

Característica	Árboles de decisión	Redes neuronales artificiales
Trabajo con imágenes	Medio	Alto
Nivel de clasificación de objetos	Alto	Alto
Compatibilidad con lenguajes de programación	Medio	Alto
Manejo de estructura de datos	Alto	Alto
Capacidad de aprendizaje	Medio	Alto
Tiempo de respuesta	Alto	Alto
Accesibilidad	Alto	Alto

Nota: muestra los parámetros de evaluación de las redes neuronales de Deep Learning contra los árboles de decisión de Machine Learning, se realizó el test con un parámetro cualitativo siendo Alto la cualidad más significativa. Fuente: (Reyes, Mejia, & Useche, 2018).

2.2.3 Técnicas de procesamiento de imagen tradicional

Son técnicas de procesamiento que permiten realizar mejoras a las imágenes, su funcionalidad está orientada a perpetrar perfeccionamientos en la calidad o facilitar la búsqueda de información, encontramos una secuencia de pasos que cumplen.

- **Adquisición.**

Según Cuzco (2020) esta se encuentra basada en 4 procesos que la componen y es la forma cómo se captura la imagen desde la realidad. }

- **Dispositivos de captura**

- Cámaras digitales
- Escáner
- Teléfono celular.

- **Características**

- Imagen binaria
- Imagen en escala de grises
- Imagen a color

- **Formatos de almacenamiento**

- BMP
- TIFF
- GIF
- JPG
- PNG

- **Realce y mejora**

Se consideró para la reducción del ruido que se genera en las imágenes al ser capturadas desde la realidad, con el objetivo de obtener un procesamiento más nítido y de mejor nivel. Aplicando filtros, realce de contraste e identificación de bordes etc.

- **Segmentación**

Consiste en la división de patrones con iguales características que se encuentran dentro de la imagen, estos pueden estar basados en formas o en colores. La figura 1 muestra un ejemplo de la aplicación del método en una determinada imagen de muestra.

- **Extracción de características**

Es una técnica inmersa en el procesamiento de imágenes digitales, es para obtener las características de las imágenes, para luego ser almacenadas en base de datos externa o local, 33 información con la que se realiza la comparación e identificación. Existen algoritmos dedicados

a esta actividad como por ejemplo SIFT, SURF, ORB por mencionar a los más conocidos (Lima, 2021).

Realizaremos la introducción de tres de las técnicas más comunes, también enumeramos sus características de cada proceso.

2.2.3.1 Métodos en el dominio espacial.

En estas técnicas las operaciones son realizadas sobre el valor del píxel, sin considerar a los píxeles vecinos. Son transformaciones que se encuentran divididas en dos grupos, transformación de intensidad simple y procesada del histograma.

a) Características

- Realce y mejora

- Mejorar condiciones de bajo contraste, baja luminosidad o demasiada claridad
- Explotar el uso del contraste facilita la lectura de una imagen
- Se utiliza cuando una imagen procesada excede ampliamente la capacidad del dispositivo de representación
- Es útil cuando la imagen tiene gran cantidad de ruido y se quiere eliminar.
- Tiene la ventaja de que el valor final del píxel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media

2.2.3.2 Métodos en el dominio de la frecuencia.

Estas técnicas consideran a la imagen digital como una función no periódica, definen los ejes en otro espacio bidimensional cuyos ejes están determinados por la amplitud o frecuencia para cada dirección de imagen.

a) Características

- Realce y mejora

- Elimina ruido y suaviza la imagen para mejorar la calidad.
- Atenúa las frecuencias altas y mantiene sin variaciones las bajas.
- Atenúa las frecuencias bajas manteniendo invariables las frecuencias altas.

- Segmentación

- Aplica la derivada de primer orden y la de segundo orden
- Realiza un rastreo horizontal, vertical y hacia otras direcciones.
- Permite discriminar los valores en que el gradiente no es máximo local.
- Permite encontrar el umbral más bajo entre dos picos
- Permite encontrar el segundo pico más alto

- Son iterativos
- Permite obtener el umbral óptimo de manera automática
- Utiliza distribución de probabilidad, media y varianza

2.2.3.3 Métodos de extracción de características.

Como su nombre lo indica, esta técnica consiste en extraer las características de la imagen tales como tamaño, colores, área, forma, perímetro, entre otras. También permite conocer las características topográficas como la representación de objetos geográficos y el tamaño de la superficie.

a) Características

- Su complejidad computacional es la más sencilla
- Su cálculo se da a través de la ecuación del punto pendiente y la ecuación normal.
- Permite minimizar las superficies de triángulos que se forman
- Establece sumas cuadráticas a través de cuatro direcciones: vertical, horizontal y dos diagonales
- Aplican derivada de primer orden
- Aplican Umbralización
- Es un proceso multifase jerárquico
- Es más útil en imágenes con geometrías regulares
- Aplica el cálculo de la primera derivada
- Se le aplica segmentación con un barrido de arriba hacia abajo y de izquierda a derecha
- Es un método iterativo
- Permite encontrar todo tipo de figuras geométricas diferente a las líneas

Conociendo la definición y las características de las técnicas mencionadas, se las aplicó para diferentes procesos dentro de la investigación. El realce y mejoramiento se realizó con la técnica de dominio espacial debido a que esta presenta como ventaja un alto índice de trabajo con imágenes tomadas de realidad, entonces la consideramos para la reducción de ruido en las imágenes de las hojas de cultivos de pimiento.

La segmentación estuvo a cargo de la técnica de dominio de la frecuencia ya que esta toma la imagen en cualquier dirección y la examina en tamaño, forma y color. Lo que se presenta cuando las imágenes son obtenidas con diferentes dispositivos, la segmentación de las hojas de los cultivos se aplicó considerando su naturaleza, tratando de no afectar su forma.

Y la extracción de características se realizó con la técnica de extracción de características debido a su complejidad computacional más sencilla.

2.2.3.4 Algoritmos de detección de objetos

Según Cuzco (2020) el proceso se realiza mediante un sistema inteligente que tiene la funcionalidad de obtener datos de un medio para luego aplicar los algoritmos necesarios para su análisis e interpretación independientemente de los factores que influyen. La detección de objetos consiste en la identificación de puntos característicos dentro de la imagen (esquinas, bordes, vegetación, etc.) con el propósito de enmarcarlos como referencia.

Tabla 3. Algoritmos de detección de objetos y sus características.

Característica	Sift	Surft	ORB
Tiempo de respuesta	Bajo	Medio	Alto
Afectación de ruido	Alto	Medio	Bajo
Trabajo con sistemas de tiempo real	Medio	Medio	Alto
Trabajo con terminales de baja potencia	Baja	Medio	Alto
Fusión de varias imágenes	Medio	Alto	Alto
Rastreo de parches en teléfonos móviles	Alta	Alta	Alta
Tiempo de detección	Medio	Medio	Alta

Nota: muestra la ponderación de los algoritmos de detección en una escala cualitativa donde Alto es la cualidad más significativa. Fuente Godoy y Ducuara (2019)

- **Identificación de la imagen**

Consiste en la identificación de la imagen a base de otra, pasa el parámetro ingresado a iguales características, se genera una matriz vectorial, se recorren los píxeles considerados como posiciones de la matriz, encontrando similitudes que arrojan el resultado (Lima, 2021).

El análisis de técnicas de procesamiento tradicionales e Inteligencia Artificial, se realizó para obtener resultados de investigación al documentar la comparativa entre las tecnologías, con el objetivo de encontrar características en cada una. Considerando sus resultados la tecnología será aplicada al desarrollo de la solución informática planteada en el objetivo general. Basado en las características, ventajas y desventajas que entregó el análisis se decidió usar la técnica de Inteligencia Artificial Deep Learning, para realizar el módulo de entrenamiento del modelo, haciendo uso del algoritmo de red neuronal utilizado en la clasificación de imágenes ver tabla 2.

2.2.4 Procesamiento de imágenes

El procesamiento de imágenes se encarga del proceso de análisis de figuras, entre sus funcionalidades destaca la obtención de características comunes y no comunes presentes en una imagen digital.



Figura 2: Secuencia del procesamiento de imágenes.
Fuente: Godoy y Ducuara (2019)

El procesamiento de imágenes es una técnica aplicada a fotografías o imágenes de cualquier objeto, recurso y/o individuo en cuestión con el objetivo de encontrar características, realizando un análisis mediante algoritmos de procesamiento (Turmero, 2017). En la figura 3 se muestra la aplicación del algoritmo de segmentación.

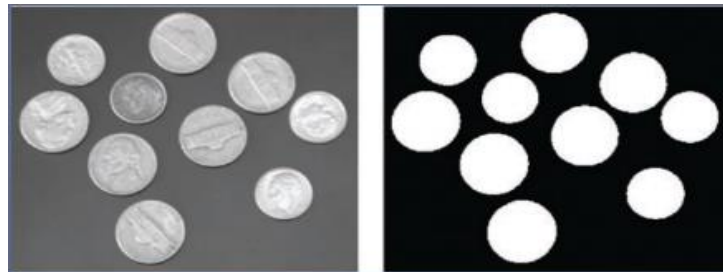


Figura 3: Ejemplo de procesamiento con algoritmo de segmentación
Fuente: (Turmero, 2017)

2.2.4.1 Características del procesamiento de imágenes

En 2017, Cárdenas y Mejía encontraron que la representación global del procesamiento de imágenes presenta dos características que sobresalen: el procesamiento de imágenes a bajo nivel y el entendimiento de imágenes a alto nivel.

En la primera, el análisis de las fotografías está dado por muy poco uso de conocimiento respecto al contenido de estas, por lo general se basa en una sucesión de 4 pasos para su desarrollo: Adquisición, preprocesamiento, segmentación, descripción y clasificación de objetos.

Por su parte, en el entendimiento de imágenes a alto nivel, es el contenido inmerso en las fotografías el que permite tomar decisiones respecto a ellas.

Las características del procesamiento han permitido comparar el resultado de la identificación de imágenes de las hojas de cultivos de pimiento en estado saludable o de enfermedad, basado en la respuesta entregada por cada una de las características del procesamiento.

2.2.4.2 Componentes del procesamiento de imágenes

a) Imagen digital

Una imagen está inmersa en el análisis y procesamiento de las mismas, esta puede ser definida como una expresión binaria o numérica, es una forma de ilustración donde sus principales componentes son los números, considerada un lenguaje expresado en carácter binario (Correa, 2018, p.15).

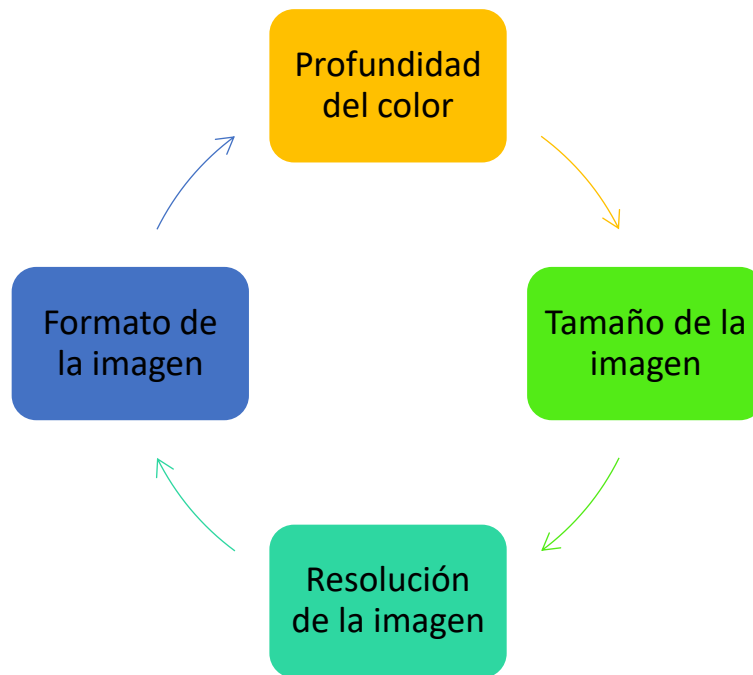


Figura 4: Aspectos que conforman una imagen digital

- Imagen digital a color

La representación de este tipo de imagen está basada en la interpretación de color y su intensidad, donde cada píxel está conformado por un valor que corresponde a cada componente principal. El color resultante del píxel vendrá por tanto definido por la cantidad de intensidad que tenga cada componente, su máximo representa el blanco y su mínimo el negro (Aguirre, 2019, p.44).

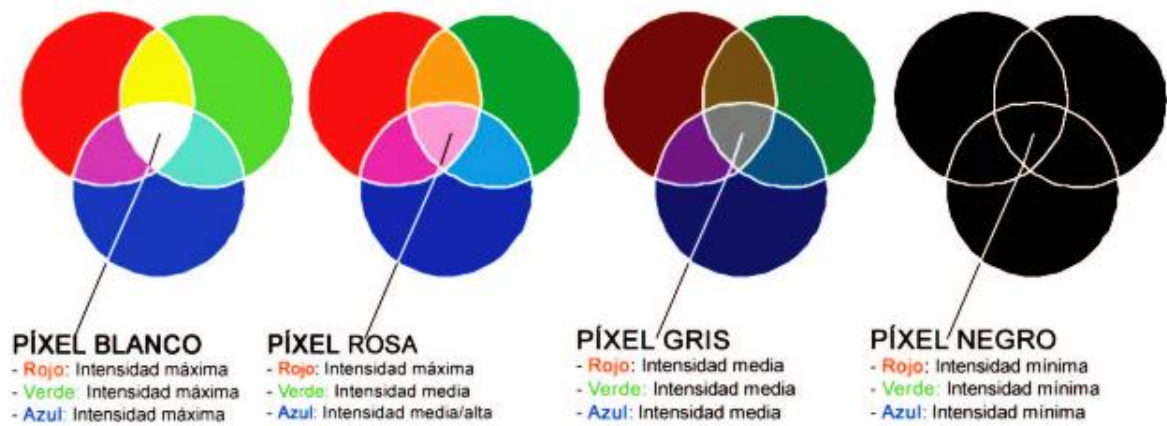


Figura 5: Píxel resultante a partir de la intensidad de sus componentes
Fuente: (Aguirre, 2019, p.44)

- **Imagen digital transformada**

Es una imagen resultante de la representación gráfica original, su transformación varía según el tipo de acción que el autor pretenda realizar, consiste en conformar una matriz para cada píxel, muy recomendable para el trabajo de visión por computador e inteligencia artificial. Su transformación también puede ser de tipo flujo de datos por bus, depende de la tecnología en que se vaya a usar (Aguirre, 2019, p.44).

(Aguirre, 2019)

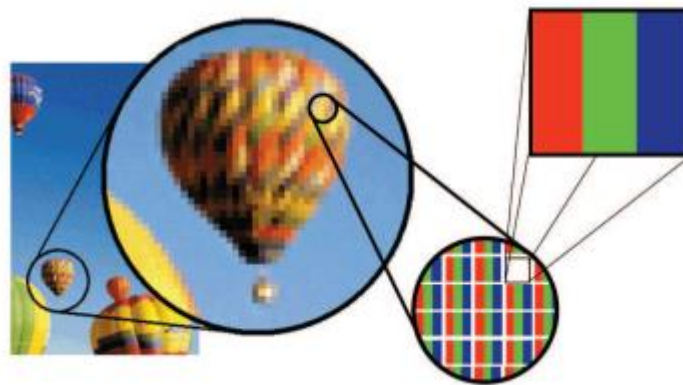


Figura 6: Representación del píxel de una imagen digital a color transformada.
Fuente: (Aguirre, 2019, p.44)

Tabla 4: Cuadro comparativo entre una imagen digital a color y una imagen digital transformada

Imagen digital	Ventajas	Desventajas
Imagen digital a color	<ul style="list-style-type: none">• Originalidad de los colores• Originalidad de profundidad del color.• Características originales.	<ul style="list-style-type: none">• Ruido de la realidad en la adquisición.• Píxeles unificados• Presenta una sola matriz de píxeles.• No recomendable para realizar análisis.• Sus características se encuentran afectadas por efectos de la naturaleza.
Imagen digital transformada	<ul style="list-style-type: none">• Mejor calidad de profundidad de color.• Trabajo con visión por computadora e inteligencia artificial.• Mayor definición de los colores.• Disminución en la afectación producida en la adquisición.• Representación de píxeles en matrices.	<ul style="list-style-type: none">• En ocasiones hay confusión del ruido con característica de la imagen.• Requiere de fuerza computacional para su mejora.

Nota: nuestras ventajas y desventajas comparativas entre la imagen digital a color y la imagen digital transformada.

Fuente: (Correa, 2018, p.15).

La imagen digital transformada se utilizó como representación gráfica de las enfermedades que se encuentran en los cultivos de pimiento de la comunidad, en base a la caracterización previamente obtenida de las hojas se realizó el procesamiento para la identificación del estado de la planta.

- Variables del color

Dentro de la imagen digital encontramos variación del color que comienza desde las propiedades innatas del color, que son el matiz, la luminosidad, el tono y la saturación. Los define así:

- El matiz (Hue). Es el valor cromático de un color, la frecuencia del espectro donde se encuentra. Depende de la longitud de onda dominante, y es la cualidad que permite clasificar a los colores como amarillo, rojo, violeta, etc.
- La luminosidad (Lightness). Es el resultado de la mezcla de los colores con blanco o negro y tiene referencia de matiz. Representa la cantidad de luz presente en un color, más blanco o más negro, según sea el caso. Cuanto mayor es la luminosidad, mayor es la cantidad de luz en un color, es decir, más color blanco posee.
- La saturación (Saturation). Se refiere al grado de pureza de un color y se mide con relación al gris. Los colores con menor saturación se muestran con más tonalidad de gris, con mayor cantidad de impurezas y con menor intensidad luminosa (Correa, 2018, p.56).

Esta variación de colores fue usada debido a la dificultad que se presenta al definir los colores de la afectación en las hojas de cultivos de pimiento teniendo que usar diferentes espacios de color para optimizar los algoritmos de detección.

Los Espacios de Color son una herramienta importante en el procesamiento digital de imágenes, ya que permiten analizar cada píxel desde otro punto de vista.

b) Modelos de color

- RGB

Representa a los colores primarios rojo, verde, azul o en su traducción al inglés red, green, blue de ahí sus siglas RGB. Su interpretación es en forma de un cubo unitario con sus ejes nombrando con sus siglas, donde el origen hace referencia al color negro en sus coordenadas (0,0,0) y al blanco en la posición (1,1,1). La suma de las intensidades de luz relativas de los colores principales de este modelo da como resultado otros colores.

- Características

- Muy utilizado en los sistemas informáticos
- Utilizado en la creación de y reproducción de colores en monitores y pantallas
- Las imágenes están en tres planos de imágenes independientes
- El procesamiento en este modelo es aplicable cuando se trabaja en los tres planos de color.
- Tiene mucha importancia en el procesamiento cuando de la comparación de colores se trata.

- HSV

Las siglas de este modelo representan el Tono, la Saturación y el Valor, también se lo conoce como HSB siendo B la representación del brillo, tiene como subconjunto la conformación de una pirámide de base hexagonal proveniente del sistema de coordenadas en forma cilíndrica. En este modelo los colores con más incidencia de brillo están contenidos en el área hexagonal donde V adopta el valor de 1, para medir la tonalidad se utiliza el ángulo que se encuentra rodeando al eje S. Rojo 0°, Azul 120°, Verde 240° y los colores complementarios se encuentran a 180° de su color primario.

- Características

- Aplica la transformación lineal al modelo RGB
- Es invariante a las condiciones de luz
- Tiene una alta complejidad computacional
- Es un recurso de difícil implementación

$$H = \arccos \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{((R - G)^2 + (R - B)(G - B))}}$$
$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B}$$
$$V = \frac{1}{3}(R + G + B)$$

Figura 7: Fórmula del modelo HSV

- HSI

Este modelo a diferencia del HSV sus dos primeras siglas representa al Tono y Saturación y la diferencia es la letra I que representa la Intensidad, genera una modificación en el subespacio del cubo transformándolo en dos conos esto gracias a la transformada no lineal del espacio de color RGB, la componente de saturación generalmente se corresponde con la distancia radial de dicho cono que corresponde al ángulo respecto al color rojo.

- Características

- El tono varía en un rango de 360° y una separación angular de 120°
- Tiene similitud con la forma de predecir el color de los humanos
- Es posible desarrollar potentes algoritmos de procesamiento de imágenes
- Tiene una alta complejidad computacional
- Es un recurso de difícil implementación

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360 - \theta & \text{si } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{(R - G) + (R - B)}{2[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\}$$

$$S = 1 - \frac{3 \min[R, G, B]}{(R + G + B)}$$

$$I = \frac{1}{3}(R + G + B)$$

Figura 8: Fórmula del modelo HSI

Con base en las características de los modelos de color investigados, se hizo uso del modelo RGB, debido a que en el análisis de algoritmos se contempló la posibilidad de usar un algoritmo tradicional o de inteligencia artificial, en estos últimos según su arquitectura la comparación se la realiza a base de color, por lo que la transformación a escala de grises no será necesaria.

2.2.5 Metodología de gestión de proyectos.

2.2.5.1 Comparación entre metodología ágil y tradicional.

Sin dejar de lado ninguna de estas, las metodologías ágiles como las tradiciones son de pleno uso y válidas para el desarrollo de software, acertadamente son válidas dependiendo de la forma de distribución de actividades del equipo de trabajo, se presentan numerosas discrepancias entre ambos grupos (Gómez, Cervantes y González, 2019).

Tabla 5: Cuadro de características de las metodologías de desarrollo de proyectos

Metodología Tradicional	Metodología Ágil
Iniciado el proyecto no se permite cambiar los requerimientos.	Las fases son flexibles y admiten cambios en cualquiera de ellas.
Está dirigida para equipos de trabajo grandes.	Se limita a equipos de trabajo menores a 10.
Orientada a proyectos con duración extendida.	Por lo general se aplica en proyectos de duración corta.
Se define contrato al inicio del proyecto	El contrato es flexible o no se lo realiza
Para desplegar el software es necesario terminar todo el desarrollo.	Se puede desplegar el software constantemente

Nota: muestra las características de las metodologías de desarrollo de software. Fundamentos de Ingeniería de Software 2019 por Gómez, Cervantes y González

Después de realizar el análisis característico de cada metodología, se pudo deducir que usar una metodología ágil tiene muchas ventajas, pero debido al tiempo de duración del proyecto y la flexibilidad que presenta se ha decidido utilizar una metodología tradicional.

2.2.5.2. Metodología de proceso incremental

Proporciona un grado de flexibilidad que permite considerar versiones de productos con incremento en sus funcionalidades, esto a la vez aprueba la visualización de un prototipo antes del final del desarrollo. Además, nos permite empezar a trabajar sin contar con todos los recursos de hardware, software y personal necesarios. En sus inicios, usada en el desarrollo de software para más tarde ser replicada en otras ramas, se basa en un calendario de tiempos con actividades que cumplir, en cada una de ellas se muestra un avance en el producto (Pérez, 2016).

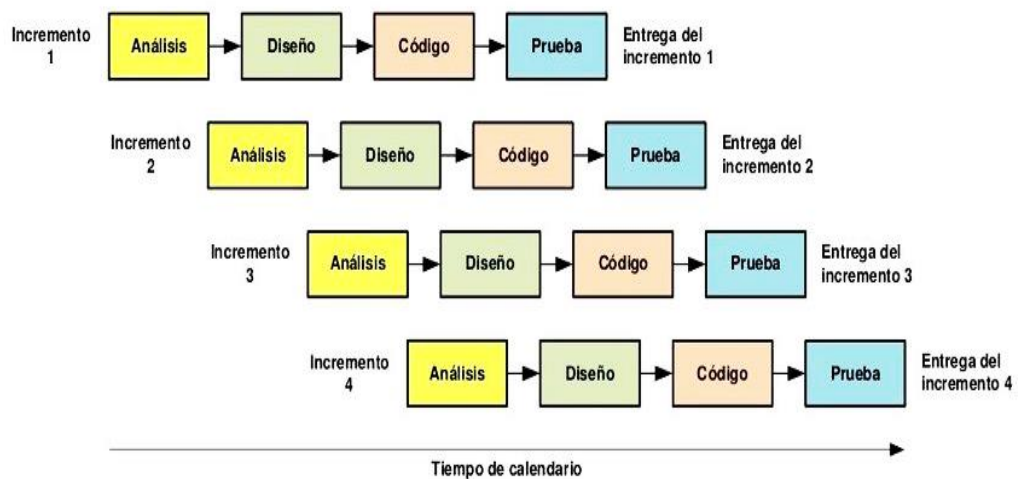


Figura 9: Fases del modelo incremental

Fuente: (Pérez, 2016)

- Características

- Difícil de evaluar el costo total
- El usuario se involucra más con el proyecto.
- Se evita proyectos largos y se entrega algo de valor a los usuarios con cierta frecuencia

a) Objetivo de la metodología de proceso incremental

Reducir los bucles de repetición de las actividades en el proceso de desarrollo.

b) Fases de la metodología de proceso incremental

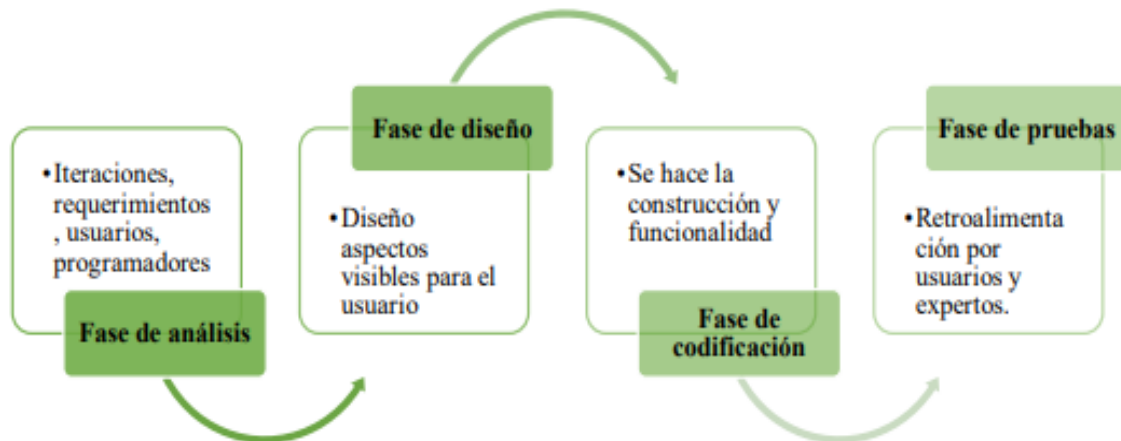


Figura 10: Fases de la metodología de proceso incremental

- Fase de análisis

Esta fase se refiere a todos los objetivos del proyecto, tanto el general o central como los específicos y requerimientos del sistema. Partiendo de estos requisitos se elabora un plan de iteraciones y reuniones de ser el caso de trabajar en grupo.

- Fase de diseño

Se puede realizar diseños simples y generalizados, se define cuál será la evolución del proyecto en las iteraciones. Cada una de ellas debe superar a la que le precede para que se pueda dar el incremento, también se puede realizar un rediseño para mejorar el siguiente incremento.

- Fase de codificación

Los diseños definidos son llevadas a cabo y así los incrementos previstos son desarrollados, los programadores escriben el código necesario para que la prueba acepte la funcionalidad desarrollada, esta forma de programar garantiza una retroalimentación sobre los errores que se está cometiendo, cumpliendo el objetivo del incremento.

- Fase de pruebas

Si el trabajo es en grupo, los encargados de la gestión de proyecto han de verificar que cada iteración culminada del incremento sea la esperada. Si no es así, hay que hacer una revisión de las tareas y buscar las causas de los errores. A diferencia de otras metodologías cada incremento generado en cada iteración es sometido a prueba.

c) Ventajas

- Se genera software operativo de manera rápida.
- Es un modelo flexible.
- Reduce costos en el cambio y alcance de requisitos.

- Es fácil de probar por sus iteraciones pequeñas.
- Es fácil administrar los riesgos.
- Cada iteración es significativa para el resultado final y es fácil de administrar.

d) Desventajas

- Se requiere experiencia para definir los incrementos.
- Cada iteración es rígida y no se cruza con alguna otra.
- Todos los problemas a futuro tienen que ser previamente definidos.
- Pueden surgir problemas de que no todos los requisitos hayan sido establecidos.

La metodología de proceso incremental permite realizar pruebas durante el desarrollo del software, se procedió diseñar diferentes prototipos sin la necesidad de generar un aplicativo final, mismos que fueron puestos en marcha durante el proceso investigativo, obteniendo mejoras en cada uno de ellos.

2.2.6 Norma Técnica ISO/IEC 15504 SPICE

Se trata de un estándar internacional que regula los procesos de calidad en la evaluación del software, con un modelo que valora la capacidad de las fases de producción de software (ISO/IEC 15504 SPICE, 2018).

2.2.6.1 Características

La herramienta cuenta de las siguientes características:

- Proponer y desarrollar un estándar de evaluación de procesos de software.
- Evaluar su desempeño mediante su experimentación en la industria emergente del desarrollo SW
- Promover la transferencia de tecnología de la evaluación de procesos de software a la industria del software a nivel mundial.

Según (ISO/IEC 15504 SPICE, 2018) la norma ISO 15504 se caracteriza por:

- Ser aplicable a cualquier organización o empresa.
- Ser independiente de la organización, el modelo del ciclo de vida, la metodología y la tecnología.
- Ser un marco para métodos de evaluación, no un método o un modelo en sí.
- Cubrir diferentes objetivos para la evaluación de procesos: - Determinación de la capacidad (niveles de capacidad o de madurez); - Mejora de procesos - Evaluar el cumplimiento de determinados requisitos del ciclo de vida de desarrollo de software

2.2.6.2 Objetivo

Proporciona una base para realizar evaluaciones de la capacidad de los procesos de software y permite reflejar los resultados obtenidos sobre una escala común.

2.2.7 Planeación del entrenamiento

La planeación del entrenamiento fue utilizada para listar las actividades que se llevarán a cabo para efectuar la identificación de las enfermedades por medio de la solución informática.

2.2.7.1 Niveles de planeación

- Planeación estratégica

Se consideró para identificar el área de trabajo, extensión y objetivo de la acción entre otras actividades como: las enfermedades más comunes, clasificación de enfermedad que más afecta al cultivo, obteniendo los resultados de la ficha técnica realizada en Excel que dejó como resultado de estudio a dos enfermedades Mancha Gris (*Stemphylium solani*) y Oidiopsis (*Oidiopsis taurica - Leveillula taurica*) ver anexo 3:

- Planeación operativa

Se aplicó para realizar un sondeo de las necesidades como infraestructura de hardware, software, tecnología, personal, patrones de las enfermedades del cultivo y otros que sean necesarios para proceso de identificación.

2.2.8 Herramientas físicas para el procesamiento

Laptop con sistema operativo Windows 10 Home Single Lenguaje de 64 bits, procesador Intel(R) Celeron(R) CPU N3060 @ 1.6GHz y 4GB de RAM, las condiciones de desarrollo con estas características de no fueron completamente óptimas por lo que se recomienda utilizar equipo con características más afines.

Por lo expuesto anteriormente sobre las características del equipo se usó emuladores físicos con diferentes versiones de Android para probar la App en su etapa de desarrollo.

2.2.9 Herramientas de software para el procesamiento

Se utilizó para el procesamiento de las imágenes digitales de las enfermedades en los cultivos dentro de la solución informática, con su ayuda se realizó la parte funcional de la aplicación correspondiente a la obtención de características e identificación de la enfermedad y entrenamiento del modelo de inteligencia artificial.

2.2.9.1. Librerías para el preprocesamiento de imágenes

En la actualidad existen librerías que están orientadas al procesamiento de imágenes digitales a continuación veremos la comparativa entre algunas de las más usadas en la actualidad.

Tabla 6: Cuadro comparativo librerías utilizadas en el procesamiento de imágenes

Característica	TensorFlow	PyTorch	LTI-Lib
Trabajo con aplicaciones en tiempo real	SI	NO	NO
Distribución bajo licencia BSD (libre)	SI	SI	NO
Trabajo con algoritmos de visión artificial	SI	NO	SI
Compatibilidad con Windows, Linux, Android y Mac OS.	SI	NO	NO
Interfaces para varios lenguajes de programación	SI	NO	NO
Incluye transparencias y animaciones	NO	SI	SI
Algoritmos completos y robustos	SI	NO	SI
Degrado en el formato JPEG tras operaciones	SI	NO	NO
Es modular	NO	SI	SI
Tiene orientación a objetos	NO	NO	SI

Nota: muestra las características comparativas de las librerías de procesamiento de imágenes más usadas Fuente: Jiva (2019)

El análisis de las librerías evidencia el resultado de la efectividad de TensorFlow en tratamiento de imágenes, siendo considerada para realizar el proceso de análisis de las imágenes de los cultivos previo a la identificación de las enfermedades, basado en sus características que se ajustan a las necesarias para el desarrollo de la solución informática. Como se mencionó anteriormente se realizó un entrenamiento de un modelo de inteligencia artificial.

2.2.9.2. TensorFlow

Se hizo uso de esta librería de Python y los paquetes de keras para entrenar el modelo de clasificación de imágenes, para ello se recaudaron imágenes de hojas del cultivo del pimiento que presentaban los síntomas y se las reunió en un paquete de datos. El entrenamiento por motivos de fuerza computacional se realizó en Google colab.

Según (Baume, 2021) Google Colab es un servicio de Google diseñado para ejecutar código Python en la nube, surge debido al costo computacional y a la falta de almacenamiento en los pc físicos, también permite ejecutar texto enriquecido, debido a la depuración con GPU y TPU se lo considera una excelente herramienta de codificación Python en la nube.

Tabla 7: Comparativa entorno de codificación Python local y el servicio de Google - Google Colab

Características	Google colab	Entorno de Python en Pc
Funcionamiento	CPU, GPU, TPU	CPU
Tiempo de respuesta	En minutos y segundos	En días y horas
Paquetes de requeridos	Previamente instalados	Realizar instalación de la mayoría
Configuraciones previas	Configuraciones de por defecto	Configurar una a una
Licencia de código	Python libre	Python libre
Distribución de código	Libre	Libre
Lenguaje de programación que usa	Python	Python
Accesibilidad	Ilimitada	Ilimitada
Apariencia	Amigable con usuario	Amigable con usuario
Almacenamiento de datos	Temporal hasta cierre de sesión	Guardado en disco de computador
Almacenamiento de código	Indefinido en Google drive	Guardado en disco de computador

Nota: muestra los parámetros comparativos del entorno local y el servicio de Google.

2.2.9.3 TensorFlow Lite

Se usó para realizar la transformación del modelo de TensorFlow entrenado en Python a un formato compatible de Android mismo que se puede hacer uso dentro de Android Studio.

2.2.10 Requerimientos para el aplicativo

Fueron utilizados para generar un aplicativo funcional que hace uso de las diferentes tecnologías, fusionado en uno solo entre los requerimientos usados listamos los siguientes.

2.2.10.1 Edición de código

- ✓ Android Studio 4.2.2
- ✓ IDLE 3.9 (64-bit)
- **Administración de base de datos MySQL**
- ✓ phpMyAdmin
- **Alojamiento**
- ✓ 000WebHost en su versión gratuita

2.2.10.2 Lenguajes de programación métodos y algoritmos de procesamiento

Se los utilizó para dar funcionamiento al aplicativo, fue con ellos que se consiguió que la solución informática cumpla con su funcionalidad y objetivo de desarrollo.

- **Lenguajes de programación**

- ✓ Java
- ✓ Python 3.9.6

- **Métodos y algoritmos de procesamiento**

Fueron considerados dentro de la solución informática para la obtención de resultado proveniente del preprocesamiento de las imágenes y aplicación de las técnicas mencionadas.

- ✓ Chaquopy 9.1.0: es un complemento que se utilizó para realizar la integración del módulo de Python en Android en su versión gratuita misma que prohíbe su distribución al público de ser ese el objetivo se debe hacer uso de su versión de pago.
- ✓ TensorFlow
- ✓ Algoritmo de Deep Learning
- ✓ Red neuronal convolucional
- ✓ Arquitectura de Red neuronal

2.2.11 Enfermedades del pimiento (*Capsicum annuum*)

Saber sobre las enfermedades en forma general, sirvió en la investigación para conocer los daños que estas producen al hospedarse en la planta, información que será considerada para las características que tendrá el aplicativo (Soto, 2018) menciona que:

Las enfermedades que afectan al cultivo del pimiento pueden provocar importantes pérdidas en el rendimiento y calidad de los frutos. Para que exista una enfermedad, se deben presentar tres factores que permiten el desencadenamiento de ésta: un hospedero susceptible, condiciones medioambientales que favorezcan el desarrollo de una patología y un agente causal virulento. Dentro de estos últimos podemos tener hongos, bacterias, virus y nemátodos (p.69).

La planta de pimiento por su parte se consideró como ejemplo, para llevar a cabo las acciones mencionadas, se consideró las enfermedades que habitualmente se presentan en este cultivo con más incidencia.

2.2.12 Clasificación de las enfermedades

La clasificación de las enfermedades será considerada para la identificación de patrones característicos que puedan ser orientados de manera que ayuden en la selección del más idóneos para llevar a cabo la investigación, revisando la clasificación de Jordá (2016), este las distribuye como enfermedades producidas por hongos, por bacterias, por virus y por daños fisiológicos,

encontramos peculiaridades en dicha categorización. Por ejemplo, algunas enfermedades presentan sus inicios de la afectación en las hojas, otras en las raíces y otras en el tallo. Con fundamento en lo anterior presentamos esta agrupación así: enfermedades de hoja, enfermedades de tallo y enfermedades de raíz.

2.2.12.1 Enfermedades de hoja

Son aquellas que sus principales síntomas presentan en las hojas, mostrando pequeñas deformaciones, cambios de colores, manchas de diferentes colores y por último la caída de la hoja del cultivo. Se ha considerado este apartado para realizar el análisis y documentación de las características y síntomas.



Figura 11: Ejemplo de enfermedad con afectación en hoja
Fuente: (Obregón, 2016)

2.2.12.2 Enfermedades de raíz

Son aquellas que sus principales síntomas reposan en las raíces de la planta, usualmente estas son las más difíciles de identificar. Se presentan como bulbos en las raíces.

2.2.12.3 Enfermedades de tallo

Son aquellas que sus síntomas principales se pueden apreciar en el tallo, se presentan manchas de pudrición o en ocasiones escaldaduras, comúnmente son ocasionadas por el sol o por la lluvia.

2.2.12.4. Identificación de las enfermedades

Se realizó un análisis previo a la revisión de bibliografías, se aplicó una ficha técnica asistida con 4 iteraciones dos dirigidas a técnicos y dos a agricultores. El motivo, realizar un sondeo de las enfermedades que se presentan con más incidencia en los cultivos de pimiento de la comunidad de Pusir Grande ver anexo 3. Basado en la clasificación y los resultados de la ficha se procede a realizar el siguiente análisis bibliográfico de estas afectaciones.

- Ficha:

Tanto agricultores como técnicos coincidieron con estas enfermedades obteniendo el índice promedio de incidencia.

- Mancha bacteriana Gris – incidencia del 95% - grupo, enfermedad de hoja.
- Ceniza/ Oidiopsis - incidencia del 98.75% - grupo, enfermedad de hoja.
- Marchitez *Phytophthora capsici*- incidencia del 100% - grupo, enfermedad de raíz.

a) Mancha bacteriana Gris (*Stemphylium solani*)

Según Franco, (2019) en los años 90 es visualizada por primera vez en Argentina, es causada por 3 diferentes patógenos de la categoría *Stemphylium*. Su alcance de daño se expande a nivel mundial, el índice de destrucción de esta afectación es letal en ocasiones se han presentado estadísticas de daño a cultivos en su totalidad, está dentro del rango de destrucción entre las más letales.

- **Características importantes**

- Afecta en primera instancia a las hojas
- Si las condiciones están a favor del patógeno afecta al tallo y a los pecíolos en su etapa temprana.
- Aún no se han encontrado lesiones en el fruto.

Enfermedad de tipo foliar producida por patógenos del género *Stemphylium* su afectación hace referencia a *Stemphylium botryosum*, *S. lycopersici* y *S. solani*.

- **Caracterización de los síntomas**

Es importante saber identificar los primeros síntomas de esta afectación ya que son estos los esenciales para su control, para ello se debe considerar:

- **Se presenta como manchas**

- De tamaño pequeño.
- De forma circular a oblonga
- Tienen una repartición al azar
- Su color esta entre pardo a negro
- Se las encuentra en las hojas de la parte inferior de la planta

Una vez presentada su madurez media la anomalía foliar sufre cambios en su composición de forma y color, expandiéndose y formando contusiones necróticas de núcleo gris, orilla marrón oscuro y halo clorótico. Algunas veces, las lesiones se unen ocupando un porcentaje de la hoja más expandido (Franco, 2019).

En su desarrollo completo presenta sequedad completa de la hoja y desprendimiento.

- **Período del Padecimiento y Epidemiología**

Los microorganismos causantes de esta enfermedad tienen una nutrición a base de restos de otros cuerpos, pueden estar presentes entre los cambios de temporada de los cultivos.

- El patógeno puede ser asexual y sexual.
- Para llegar a la superficie de los folios las esporas se encaminan por diferentes medios.
- La germinación se hace efectiva pasado las 6 horas de la infección.
- La enfermedad se agrava en las condiciones predisponentes, como el clima húmedo, hospedaje de agua lluvia en las plantas, y en el intervalo de temperatura de 23 °C y 27 °C.

- **Tratamiento de la enfermedad**

El patógeno que produce la enfermedad no se debe tratar de eliminar en su totalidad, sino controlarlo de tal manera que se lo mantenga bajo la línea de daño y afectación. Por lo cual es necesario realizar el manejo integral de enfermedades, aplicando estrategias como el plantado de cultivos fuertes y la aplicación de soluciones químicas y biológicas.

- **Especificaciones**

- Tratar de erradicar el rastrojo durante y después del cultivo.
- Impedir la proximidad en tiempo y espacio de los huéspedes vecinos.

b) Oidiopsis (Oidiopsis taurica - Leveillula taurica)

- **Características importantes**

- La afectación se encuentra en las hojas anverso y reverso
- Si las condiciones están a favor del patógeno también afecta al tallo en su etapa tierna y a los pecíolos.
- Este si presenta lesiones en el fruto.

La afectación es a las hojas, el patógeno que la produce recibe el nombre de *Oidiopsis taurica* -*Leveillula taurica*.

- **Caracterización de los síntomas**

Es importante saber identificar los primeros síntomas de esta afectación ya que son estos los esenciales para su control, para ello se debe considerar:

- **Se presenta como manchas**

- De tamaño pequeño distribuidas por la hoja en el anverso.
- De forma redonda
- Su color esta entre amarillo y verde claro
- Se las encuentra en las hojas de la parte inferior de la planta

- En la cara posterior de la hoja se desarrolla polvillo blanquecino

En su desarrollo completo presenta sequedad completa de la hoja y desprendimiento, también afecta al fruto.

- **Período del Padecimiento y Epidemiología**

Temperatura de 25°C es condiciones óptimas de desarrollo y cuando son superiores a 30°C se acelera el desarrollo de síntomas. Las esporas (conidios) son transportados por el viento y es la principal vía de diseminación. *L. taurica* es un parásito obligado y sobrevive en pimientos silvestres y hospedantes alternativos durante todo el año.

- **Tratamiento de la enfermedad**

- Tratar de erradicar el rastrojo durante y después del cultivo.
- Impedir la proximidad en tiempo y espacio de los huéspedes vecinos.
- Aplicar tratamientos químicos

Tabla 8: Factores importantes de las enfermedades para usar en el aplicativo

Aspecto	Propósito	Resultado a usar
Síntomas	Fueron investigados para identificar qué anomalías presentan estas dos enfermedades, entre estas se consideró, el color y la forma que presentan durante su etapa de desarrollo, misma que fueron obtenidos de la documentación bibliográfica.	Características de los síntomas.
Condiciones predisponentes	Se realizó el análisis para poder entregar una posible predicción de los factores que las producen, y que agente causal es el causante de la afectación al cultivo, información que será entregada a través del aplicativo.	Factores ambientales Patógeno
Manejo de la enfermedad	Se lo investigo para la toma de decisiones y la recomendación de un posible tratamiento que puede ser aplicado en el cultivo, se evidenciará a través de la aplicación.	Preventivos y tratamientos

Nota: muestra especificaciones de los aspectos, el propósito del mismo y el resultado que se va usar en el desarrollo de la solución informática.

Fuente: Franco, (2019)

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

El presente proyecto de investigación en su marco metodológico utilizó un enfoque mixto, logrando indagar en las características de los dos enfoques, en primer lugar, el cuantitativo debido a las características que presentan los datos obtenidos desde su naturaleza por medio de la encuesta. Además de la documentación encadenada establecida para dar cumplimiento al objetivo de la investigación. Asegurando que las actividades del sujeto investigador estén encaminadas a la obtención de un producto rentable.

En cuanto al enfoque cualitativo se usó para analizar el uso de tecnologías de procesamiento de imágenes digitales en el sector agrícola, además del estudio característico y sintomatológico de las enfermedades de hojas del cultivo de pimiento a través de la aplicación de una entrevista la cuál describió los pasos para la identificación del padecimiento.

3.1.2. Tipo de Investigación

3.1.2.1. Investigación Bibliográfica.

Esta investigación es bibliográfica porque se indaga en fuentes de información relacionadas con la identificación de enfermedades en plantas utilizando el procesamiento de imágenes digitales. Es un literal de suma importancia, gracias a esté se pudo encontrar diferentes formas de llevar a cabo dicha actividad, como por ejemplo la aplicación de redes neuronales, el reconocimiento de patrones, la integración de librerías de procesamiento compatibles en diferentes sistemas operativos entre otros.

Campos (2017) afirma que: “Una investigación bibliográfica o documental es aquella que utiliza datos secundarios como fuente de información. No se trata solamente de una recopilación de datos contenidos en libros, sino que se centra, más bien, pretende buscar soluciones a los problemas planteados”. (p.13)

3.1.2.2. Investigación Aplicada

Es aplicada porque durante el desarrollo del proyecto se pudo aplicar el conocimiento científico otorgado por los docentes en las aulas durante los diferentes periodos académicos cursados en la universidad, además de la experiencia en el desarrollo de software adquirido en el transcurso de la carrera, también porque nos permite aplicar las tecnologías que se encuentran vigentes en el desarrollo de soluciones informáticas. Según Schwarz (2019):

La investigación aplicada busca la generación del intelecto con aplicación directa a los problemas del ámbito social. Tiene como objetivo resolver un determinado problema,

ocupándose en la consolidación del proceso para el enlace entre la teoría y el desarrollo de un producto. Aportando con la contribución entre la universidad y la industria en el proceso de entrega de tecnología. (p. 7)

3.1.2.3. Investigación de campo

La investigación es de campo, considerando que el levantamiento de información se realizó por medio de entrevistas, encuestas y fichas de observación directa aplicadas a los agricultores y técnicos agrónomos de la comunidad, de los cuales se pudo obtener como resultado las características y síntomas de las enfermedades, las condiciones climáticas del lugar, y además de información relevante para el desarrollo del aplicativo como por ejemplo, el tiempo de ejecución, el nivel de procesamiento, la exactitud de los resultados sugerida entre otros.

La investigación de campo entrega datos completamente relevantes y reales esto se debe que son recopilados directamente de la realidad y no a través de suposiciones, esta investigación hace uso de las técnicas de recopilación de información, mismas que aportan a la solución del problema planteado anteriormente (Cuenca y Ante, 2019, p.26).

3.2. Idea a defender

El procesamiento de imágenes digitales ayudará a la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande.

3.3. Definición y operacionalización de variables

3.3.1. Definición De Variables

- **Variable dependiente**

Identificación de enfermedades de hoja en cultivos de pimiento

- **Variable independiente**

Procesamiento de imágenes digitales

Tabla 9. Operacionalización de la variable independiente.

Variable	Definición conceptual	Dimensión	Indicadores	Técnica	Instrumento
Procesamiento de imágenes digitales	El procesamiento de imágenes es una técnica aplicada a fotografías o imágenes de cualquier objeto, recurso y/o individuo en cuestión con el objetivo de encontrar características realizando el análisis mediante algoritmos de procesamiento (Turmero, 2017).	Imagen Digital	Profundidad de color Tamaño de la imagen Resolución de la imagen Formato de la imagen Píxeles	Encuesta Documentación Pruebas	Cuestionario Bibliografía Ficha técnica
		Técnicas de procesamiento.	Tiempo de respuesta Procesamiento de datos Análisis de patrones		
		Red Neuronal	Simplicidad Velocidad de ejecución Eficiencia		
		Arquitectura de procesamiento	Exactitud Valores de pérdida Aprendizaje Baja de pesos		

Tabla 10. Operacionalización de la variable dependiente

Variable	Definición conceptual	Dimensión	Indicador	Técnica	Instrumento
Identificación de enfermedades de hoja en cultivos de pimiento	Las enfermedades son un fenómeno que aparece en diversos estatus de la palta, sus efectos pueden terminar con la misma.	Clasificación	Agente Causal Daños fisiológicos		
		Síntomas	Decoloración en hojas Marchitez Manchas de sequedad	Entrevista Observación Directa Documentación	Cuestionario Grabadora Ficha técnica Bibliografía
		Condiciones Predisponentes	Temperatura ambiental Días de lluvia Humedad excesiva Maleza		

3.4. MÉTODOS UTILIZADOS

3.4.1. Método conceptual deductivo

Este método fue utilizado dentro de la investigación, para analizar los datos y obtener como conclusión la teoría del proceso. Siendo un método secuencial que pasada de lo particular a lo general ayudó a organizar la información relevante en este caso fue aplicado principalmente en la recolección teórica para la clasificación de las enfermedades, se estipula síntomas, condiciones predisponentes, agentes causales, tratamientos. Con fundamento en estos resultados se propuso el procesamiento de imágenes para la identificación de enfermedades, mismo que se reflejó en una solución informática.

Carbajal (2019) afirma que:

Está basado en la observación y experimentación de hechos y acciones concretas para obtener una conclusión general sobre el tema, este proceso empieza en la información de los datos y termina llegando a la teoría, por lo que se deduce que va de lo particular a lo general. (Diapositiva 31)

3.4.2. Método de la Investigación-Acción Participativa (IAP)

Fue utilizado para levantar información de los dos tipos de enfermedades consideradas como ejemplo de estudio. El método fue aplicado al realizar conversatorios directamente con los agricultores de la comunidad Pusir Grande.

3.4.3. Método de reconstrucción de hechos

Se utilizó el método de reconstrucción de hechos en el análisis de algoritmos y técnicas de procesamiento de imágenes existentes, con la información que se encontró en libros, portales de internet, artículos científicos, tesis, etc. Se seleccionó el que más apego tiene con el tema, para dar cumplimiento a la propuesta de esta investigación.

3.4.4. Análisis Estadístico

3.4.4.1. Población.

Es el número de individuos que poseen las características comunes, este grupo puede ser finito o infinito es el número de integrantes que lo conforman quién da el calificativo. Además, permite que se obtengan resultados estadísticos de un determinado estudio (Gamboa, 2017, p.8).

Para realizar el proyecto de investigación se tomó como referencia de población a los 263 agricultores de la comunidad de Pusir Grande y a 10 técnicos agrónomos que interactúan en la comunidad.

3.4.4.2. Muestra.

Se usó para obtener el número de individuos que participaran en la encuesta de la investigación, según Ventura (2017) “una muestra es entendida como un subconjunto de la población conformado por unidades de análisis”.

En donde:

n= Valor de la muestra poblacional a determinar

N= Tamaño de la población.

Z= Nivel de confianza. (95% valor de fórmula 1,96)

P= Probabilidad de éxito, o proporción esperada (valor de fórmula 0.5)

Q= Probabilidad de fracaso. (Valor de fórmula 0.5)

e= porcentaje de error de muestreo. (Valor de fórmula 0.05)

$$n = \frac{Z^2 * p * q * N}{e^2 * (N-1) * Z^2 * p * q}$$

$$n = \frac{1.96^2 * 0.5 * 0.5 * 273}{0.05^2 * (273-1) * 1.96^2 * 0.5 * 0.5}$$

$$n = \frac{262.19}{1.64}$$

$$n = 159.87 = 160$$

$$\text{Agricultores} = \frac{263}{273} = 0,9633 * 160 = 154$$

$$\text{Técnicos} = \frac{10}{273} = 0,0366 * 160 = 5.85 = 6$$

Tabla 11: Resultado cálculo de muestra

Actores	Población	Muestra
Agricultores	263	154
Técnicos agrónomos	10	6
Total	273	160

Nota: muestra los resultados de la aplicación de la fórmula de población y muestra.

Para obtener los resultados de acuerdo con la investigación se aplicó el muestreo probabilístico, aplicado al 58,55 % de los agricultores y al 60% de los técnicos agrónomos.

3.5. Técnicas

3.5.1. Encuesta

Fue aplicada a los agricultores de la comunidad en cuestión con objetivo de obtener información relevante para la investigación. Se aplicó un cuestionario de 10 preguntas relacionadas con la propuesta de investigación, este instrumento fue realizado en el editor de texto Microsoft Word, también consta de una hoja de respuestas realizada en Microsoft Excel, con las respuestas obtenidas se pudo constatar la aceptación de los agricultores a la solución propuesta.

Según Cabrera y Ramírez (2017) está dedicada a la extracción de información de utilidad social, tiene como instrumento al cuestionario estructurado, misma que nos deja saber las opiniones de un grupo determinado que son partes de la situación, la diferencia con la entrevista está en el guion que esta sigue.

3.5.2. Entrevista estructurada

Esta técnica fue aplicada para obtener información sobre las indicaciones técnicas, tratamientos y procesos preventivos que deben seguir los agricultores antes y durante la cultivación del sembrío de pimiento. Se la aplicó a técnicos agrónomos que interactúan con la población de estudio en la comunidad, esta entrevista siguió un patrón de preguntas con respuestas abiertas dejando libertad a los entrevistados para que emitan sus criterios.

Esta técnica está orientada a buscar datos informativos que sean de ayuda para la investigación, es un conversatorio entre dos o más partes que interactúan en un tema en específico, puede realizarse de forma estructurada o no. Es la persona que entrevista la encargada de llevar el orden de las preguntas para que el entrevistado otorgue sus respuestas brevemente de manera colaborativa (Folgueiras, 2016, p.3).

3.5.2. Observación Directa

Se aplicó la observación directa para obtener las imágenes de las hojas del cultivo de pimiento que posean síntomas de las dos enfermedades que fueron tomadas como ejemplo, para ello se realizó una ficha técnica con las características que debe presentar la hoja, en base a ello se procedió a digitalizar haciendo uso de dispositivos móvil. Ver anexo 12

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Análisis E Interpretación De Resultados

4.1.1.1. Análisis de resultados de la encuesta

El cuestionario se realizó en base al objetivo general planteado en la investigación, una vez aplicada se procede a realizar el análisis de cada numeral de la encuesta.

a) Cuestionario encuesta Agricultores

1. ¿Es necesaria la investigación y análisis de especificaciones técnicas y características comunes y no comunes de las enfermedades foliares?

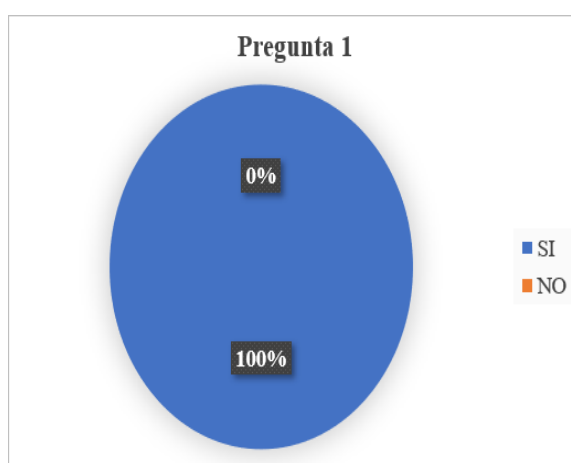


Figura 12: Análisis de resultado pregunta 1

Tabla 12: Resultados encuesta pregunta 1

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	100%	154
No	0%	0
Total	100%	154

- Análisis

De los agricultores encuestados la totalidad de ellos mencionaron que sí sería necesaria la investigación y análisis de especificaciones técnicas y características comunes y no comunes de las enfermedades foliares.

2. ¿Actualmente cuenta con un sistema de detección temprana de enfermedades en hojas por medio de procesamiento de imágenes?

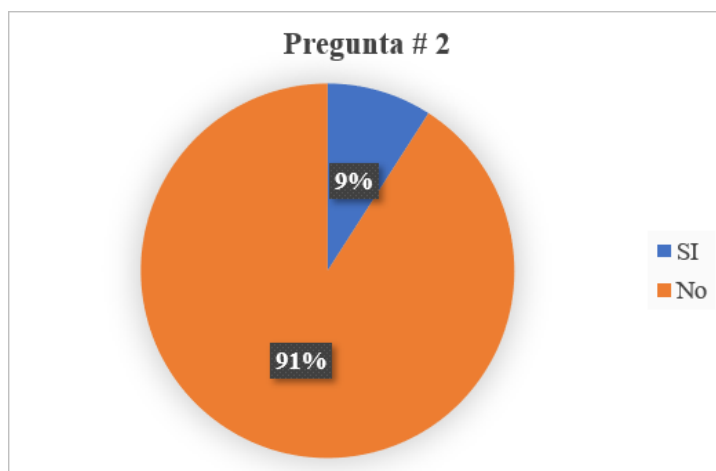


Figura 13: Análisis de resultado pregunta 2

Tabla 13: Resultados encuesta pregunta 2

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	9%	14
No	91%	140
Total	100%	154

- Análisis

Del total de encuestados su mayoría dio a conocer que no dispone de un sistema de detección temprana de enfermedades en hojas por medio de procesamiento de imágenes, dejando así un número reducido de agricultores que afirman si hacer uso de un aplicativo de similares características.

3. ¿La elaboración y seguimiento de un informe actualizado del estado de la planta reduciría el riesgo de desastres en el cultivo?

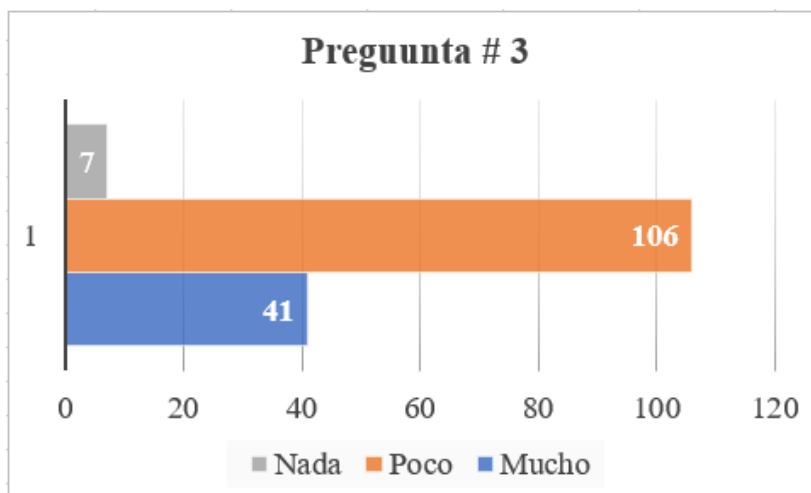


Figura 14: Análisis de resultado pregunta 3

Tabla 14: Resultado encuesta pregunta 3

Respuesta	Ponderación	
	Porcentaje	Agricultor
Mucho	27%	41
Poco	69%	106
Nada	4%	7
Total	100%	154

- Análisis

La mayor parte de los moradores de la comunidad Pusir Grande, que se dedican a la actividad agrícola dijeron que el seguimiento y control de un informe actualizado del estado del cultivo reduciría poco el riesgo de desastres en el cultivo, por su parte un número de individuos menos significativo mencionaron que reduciría mucho dejando a una minoría que afirma no reduciría nada.

4. ¿El cumplimiento de las indicaciones entregadas por los técnicos agrónomos permite frenar la propagación de la enfermedad en el cultivo?

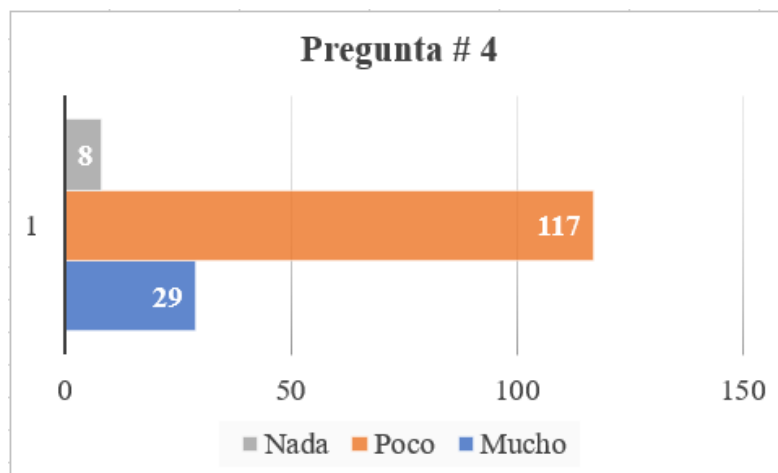


Figura 15: Análisis de resultado pregunta 4

Tabla 15: Respuesta encuesta pregunta 4

Respuesta	Ponderación	
	Porcentaje	Agricultor
Mucho	19%	29
Poco	76%	117
Nada	5%	8
Total	100%	154

- Análisis

De los agricultores de la comunidad la mayoría dejaron saber que el cumplimiento de las indicaciones entregadas por los técnicos agrónomos permite frenar poco la propagación de la enfermedad en el cultivo, de los restantes la opinión estuvo dividida entre mucho y nada siendo la primera opción de las dos la más puntuando.

5. ¿La implementación de medios tecnológicos en la identificación de las enfermedades de hoja aseguraría el desarrollo saludable del cultivo?

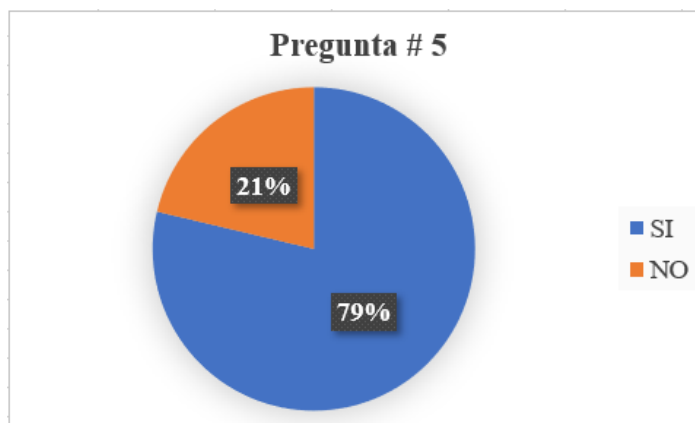


Figura 16: Análisis de resultado pregunta 5

Tabla 16: Respuesta encuesta pregunta 5

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	79%	122
No	21%	33
Total	100%	154

- Análisis

Por medio de la encuesta a 154 agricultores de la comunidad Pusir Grande se pudo obtener como resultado que la mayor parte de estos manifiestan estar de acuerdo que, con la implementación de medios tecnológicos para la identificación de enfermedades de hoja, el desarrollo del cultivo podrá ser más saludable, dejando a una parte pequeña de encuestados dijo que no.

6. ¿El diseño y desarrollo de un aplicativo de detección temprana de enfermedades en las hojas permitiría reducir el impacto negativo en la producción?

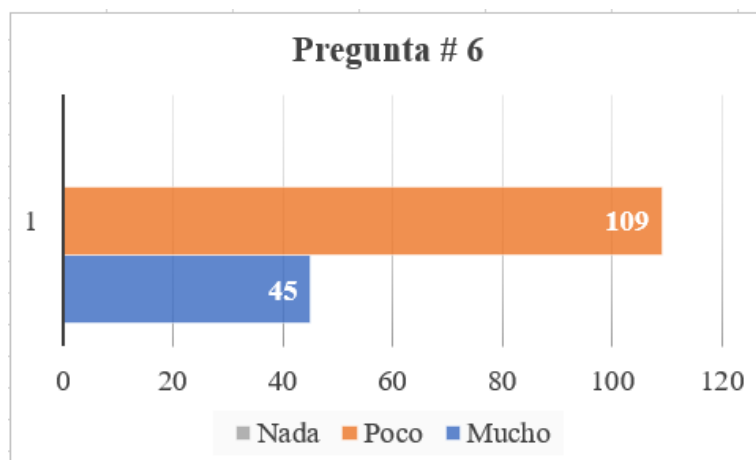


Figura 17: Análisis de resultado pregunta 6

Tabla 17: Respuesta encuesta pregunta 6

Respuesta	Ponderación	
	Porcentaje	Agricultor
Mucho	29%	45
Poco	71%	109
Nada	0%	0
Total	100%	154

- Análisis

Se aplicó una encuesta en la comunidad de Pusir Grande a 154 agricultores, obteniendo como resultados a una mayoría que manifestaron que un aplicativo de procesamiento de imágenes digitales para identificar las enfermedades en las hojas podría reducir poco el impacto negativo en la producción, opuestos a este criterio obtuvimos una minoría que afirma que reduciría mucho, ninguno de ellos dijo que no reduciría nada.

7. ¿Considera usted que la forma actual de identificación de las enfermedades en las hojas es totalmente apropiada

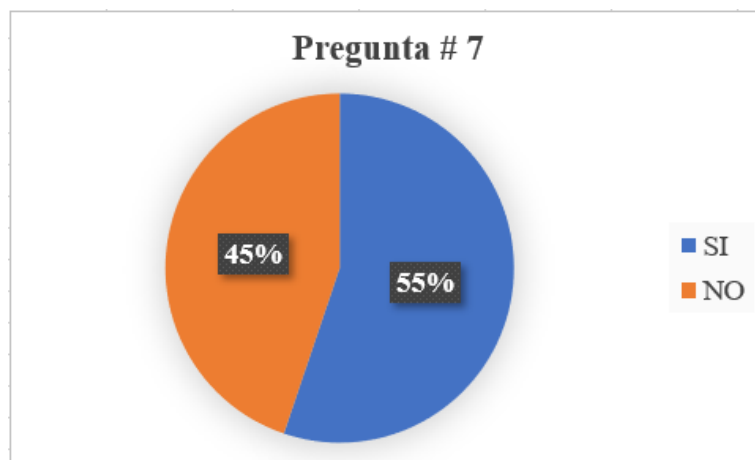


Figura 18: Análisis de resultado pregunta 7

Tabla 18: Respuesta encuesta pregunta 7

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	55%	85
No	45%	69
Total	100%	154

- Análisis

La encuesta a un grupo de agricultores de la comunidad de Pusir Grande arrojó como resultado, una división de criterios un porcentaje menor afirma que el método de identificación de las enfermedades en las hojas no es totalmente apropiado, dejando a un porcentaje mayor de la muestra con una opinión opuesta afirmando que si es el indicado.

8. ¿Cree usted que es importante la creación de un aplicativo de detección de enfermedades en hojas mediante el uso de imágenes?

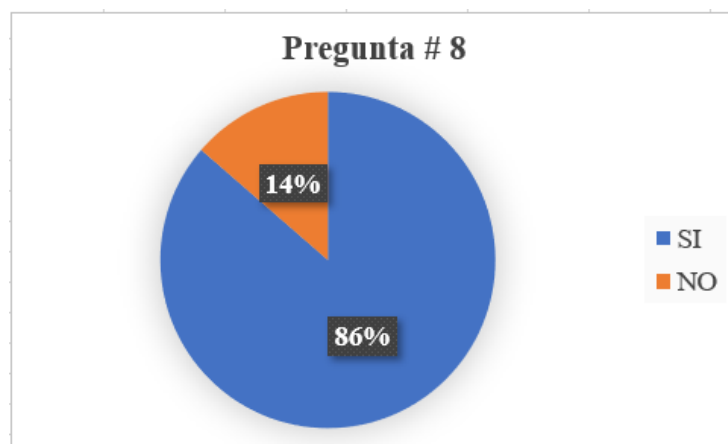


Figura 19: Análisis de resultado pregunta 8

Tabla 19: Respuesta encuesta pregunta 8

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	86%	133
No	14%	21
Total	100%	154

- **Análisis**

Del total de agricultores que se tomó como muestra para llevar a cabo la encuesta la mayoría afirma que proponer un aplicativo para la identificación de enfermedades es importante, el otro grupo restante dice que no es importante.

9. ¿Usted cómo agricultor dispone del equipo adecuado para hacer uso del aplicativo de detección temprana de enfermedades en hojas por medio de procesamiento de imágenes?

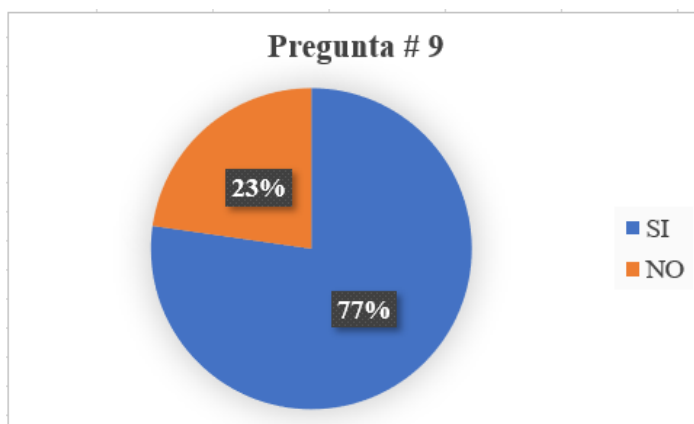


Figura 20: Análisis de resultado pregunta 9

Tabla 20: Resultado encuesta pregunta 9

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	77%	119
No	23%	35
Total	100%	154

- Análisis

Aplicada la encuesta en la comunidad de Pusir Grande se encontró con una parte pequeña del grupo de agricultores que no disponen de un equipo para hacer uso del aplicativo de procesamiento de imágenes digitales, la mayoría afirmó que si disponen de un equipo que cumple con las características que demanda el aplicativo.

10. ¿De existir el aplicativo de detección temprana de enfermedades por medio del procesamiento de imágenes, apoyaría su uso?

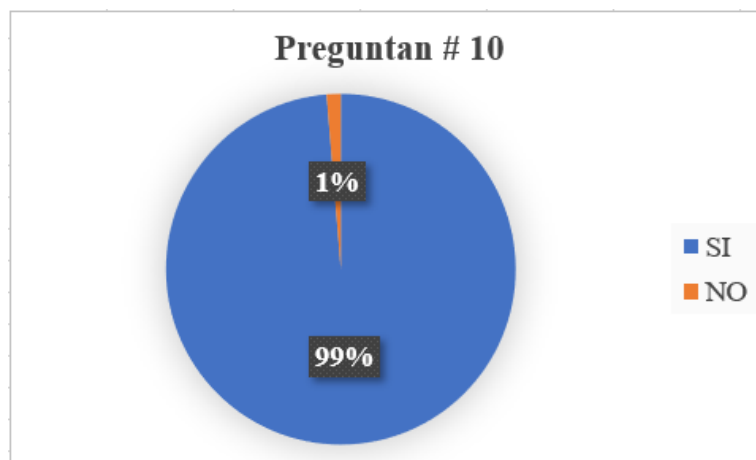


Figura 21: Análisis de resultado pregunta 10

Tabla 21: Resultado encuesta pregunta 10

Respuesta	Ponderación	
	Porcentaje	Agricultor
Si	99%	152
No	1%	2
Total	100%	154

- Análisis

Del total de agricultores encuestados casi su totalidad afirma que de existir el aplicativo, estarían de acuerdo con que se haga uso en la identificación de las enfermedades en los cultivos de pimiento, un grupo mínimo de personas menciona que no apoyaría su uso.

4.1.1.2. Análisis de la entrevista aplicada a técnicos agrónomos

1. ¿Se tienen procedimientos y políticas actualizados para la detección, control y protección de las enfermedades de los cultivos de pimiento, de haberlos cuáles son estos?

- Análisis de respuestas pregunta 1

Adquieren procedimientos mediante instrucción universitarias, gracias a la teoría pueden identificar si la enfermedad es causada por hogo, mismo que hay de diferentes tipos. Las políticas que manejan son mediante visión óptica por experiencia en relación a las características que presenta o la sintomatología que se muestra en la planta.

2. ¿En dónde son almacenados los resultados obtenidos al realizar el análisis para encontrar enfermedades en el cultivo?

- Análisis de respuestas pregunta 2

Llevar una libreta de campo donde registran todo lo que ocurre con el cultivo, lo asemejan a una bitácora del producto del cliente, en la actualidad almacenan en las tablets o en sus celulares, como notas archivadas y fotografías.

3. ¿En caso de que se encuentre anomalías en las hojas de los cultivos cuáles serían los correctivos que se recomiendan aplicar?

- Análisis de respuestas pregunta 3

Cuando identifican que le está atacando al cultivo, proceden a la aplicación de tratamientos de forma química o corrección de vectores, mencionan que estos son los principales transmisores de los hongos. En otros casos también es necesaria la corrección de nutrición, manifiestan que cuando más nutrida está la planta menos se puede enfermar, después de la detección de las anomalías en las hojas recetan tanto correctivos y prevención.

4. ¿Cada que tiempo se realiza análisis de enfermedades y aplicación de correctivos en la planta de pimiento?

- Análisis de respuestas pregunta 4

Tomando como punto de partida las condiciones climáticas considerando el tiempo y la zona en la que se encuentre, realizan comparaciones de los climas que varían en los sectores de Cuambo, Valle del Chota y Urcuquí que son los lugares donde más se cultiva el pimiento, la recomendación varía según la estación del año Invierno o verano, la frecuencia más corta de aplicación de fungicidas en invierno va de 8 a 12 días y en verano de 12 a 15.

5. ¿Existen protocolos de prevención que el agricultor debe realizar previo al plantado del cultivo, si los hubiese especifíquelos?

- Análisis de respuestas pregunta 5

La siembra del mismo cultivo desgasta el terreno y eso lleva a que se lo llene de plagas y enfermedades por lo que ellos recomiendan como protocolo de prevención una desinfección con microorganismos, al momento del plantado recetan realizar la desinfección de hongos e insectos.

6. ¿Qué beneficios se tendría al hacer uso de una aplicativo de detección temprana de enfermedades en las hojas por medio del procesamiento de imágenes?

- Análisis de respuestas pregunta 6

Manifiestan que sería de mucha ayuda, con este aplicativo los agricultores ya podrían tener antecedentes post recomendación, pero mencionan que se debe considerar que en ocasiones las enfermedades no se deben a insectos ni hongos sino a daños de nutrición.

También dicen que con esto los agricultores les ayudarían a recomendar antes de llegar al cultivo hacer la visita.

4.1.2. Resultados identificación con procesamiento tradicional

• Adquisición

Se utilizó la cámara de celular Samsung A01 Core para la adquisición de la imagen el formato jpg, el procesamiento se realizó en entorno de codificación Visual Studio Code con el lenguaje de programación Python instalado en laptop Lenovo con sistema operativo Windows 10 Home Single Lenguaje de 64 bits, procesador Intel(R) Celeron(R) CPU N3060 @ 1.6GHz y 4GB de RAM.



Figura 22: Imagen obtenida mostrada en Python OpenCV

- **Realce y Mejoramiento de la imagen**



Figura 23: Normalización de contraste



Figura 24: Aplicación de filtro de mediana sobre contraste normalizado

- **Segmentación de la imagen**

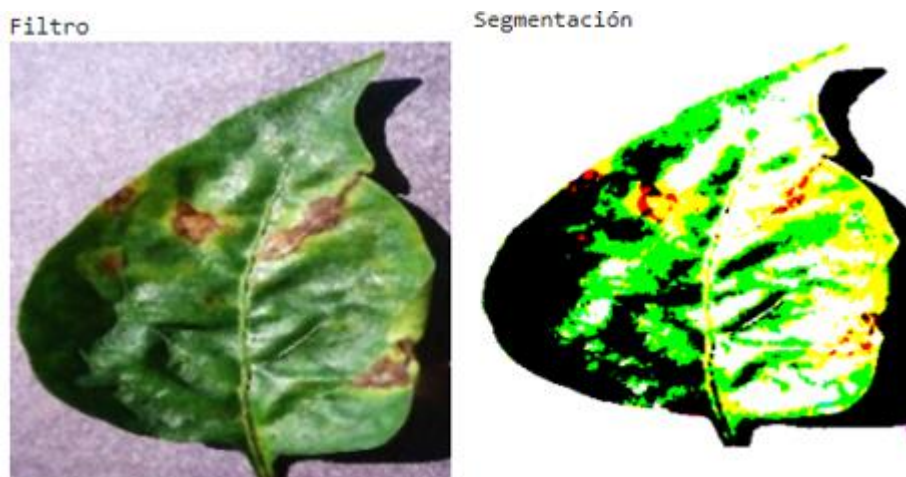


Figura 25: Segmentado de la imagen

- **Obtención de características**

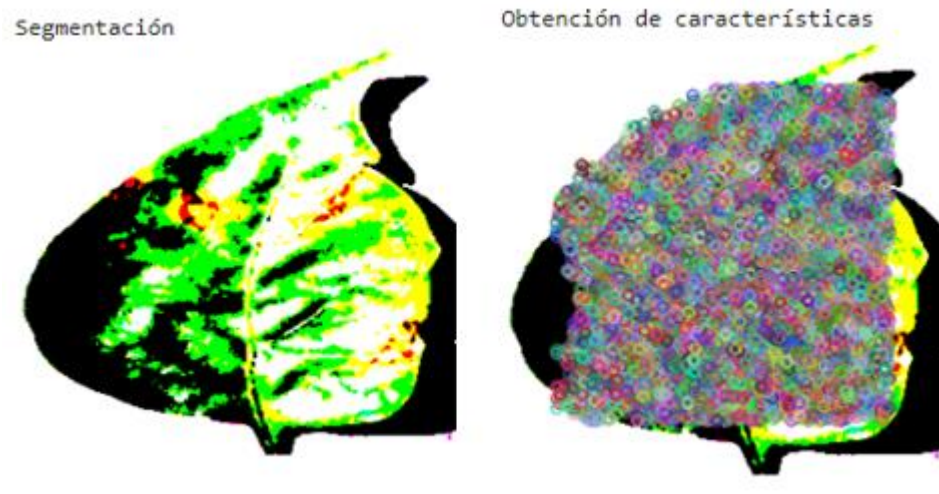


Figura 26: Aplicación del algoritmo ORB para obtención de características

- **Importante**

Una vez obtenidas las características estas pueden ser almacenadas en base de datos, realizado el preprocesamiento de las imágenes, la identificación se la puede realizar con un modelo entrenado de Machine Learning o con la máquina vectorial ver anexo 4.

4.1.3. Resultados identificación con procesamiento a base de inteligencia artificial

- **Adquisición de dataset de imágenes**

Imágenes de hojas de cultivo que poseen la enfermedad y saludables obtenidas de la realidad con diferentes cámaras de dispositivos móviles. También cuenta con una clase ajena para diferenciar de individuos que están fuera de la investigación



Figura 27: Dataset para en entrenamiento y validación del modelo

- **Entrenamiento red neuronal**

Se entrenó un modelo de Red Neuronal Convolutacional para la clasificación de imágenes basado en la técnica de transferencia de aprendizaje o transfer learning en su traducción ha inglés, se utilizó la arquitectura VGG19 que es un modelo de identificación de imágenes ya entrenado que nos brinda los paquetes de TensorFlow y keras, al que se le aplicó Transfer Learning para transferir su aprendizaje a nuestro modelo entrenado para la identificación. Ver figuras de 34 a 43, muestra el código del entrenamiento del modelo.

- **Entradas del proceso de identificación**

Las entradas del procesamiento como se definió en la teoría de Deep Learning son la cantidad de píxeles que presenta cada imagen.

- **Salidas del proceso de identificación**

Entrega una predicción a baja de las cuatro clases que se le dieron al modelo como parámetros de entrenamiento.

```
/usr/local/lib/python3.7/dist-packages/keras/engine/training.py:1972: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version
warnings.warn("`Model.fit_generator` is deprecated and ")
Epoch 1/30
21/21 [=====] - 47s 712ms/step - loss: 1.4835 - accuracy: 0.4474 - val_loss: 0.8982 - val_accuracy: 0.5960
Epoch 2/30
21/21 [=====] - 7s 342ms/step - loss: 0.5012 - accuracy: 0.7778 - val_loss: 0.4818 - val_accuracy: 0.7881
Epoch 3/30
21/21 [=====] - 7s 342ms/step - loss: 0.3255 - accuracy: 0.8799 - val_loss: 0.5519 - val_accuracy: 0.7483
Epoch 4/30
21/21 [=====] - 7s 343ms/step - loss: 0.2383 - accuracy: 0.9309 - val_loss: 0.4384 - val_accuracy: 0.8278
Epoch 5/30
21/21 [=====] - 7s 346ms/step - loss: 0.1567 - accuracy: 0.9760 - val_loss: 0.4788 - val_accuracy: 0.7815
Epoch 6/30
21/21 [=====] - 7s 342ms/step - loss: 0.1284 - accuracy: 0.9850 - val_loss: 0.4578 - val_accuracy: 0.8212
Epoch 7/30
21/21 [=====] - 7s 332ms/step - loss: 0.0942 - accuracy: 0.9940 - val_loss: 0.4509 - val_accuracy: 0.8477
Epoch 8/30
21/21 [=====] - 7s 344ms/step - loss: 0.0996 - accuracy: 0.9820 - val_loss: 0.4805 - val_accuracy: 0.8278
Epoch 9/30
21/21 [=====] - 7s 342ms/step - loss: 0.0747 - accuracy: 0.9970 - val_loss: 0.4236 - val_accuracy: 0.8477
Epoch 10/30
21/21 [=====] - 7s 343ms/step - loss: 0.0753 - accuracy: 0.9880 - val_loss: 0.4824 - val_accuracy: 0.7682
Epoch 11/30
21/21 [=====] - 7s 339ms/step - loss: 0.0520 - accuracy: 1.0000 - val_loss: 0.4661 - val_accuracy: 0.8411
Epoch 12/30
21/21 [=====] - 7s 341ms/step - loss: 0.0417 - accuracy: 0.9970 - val_loss: 0.4592 - val_accuracy: 0.8411
Epoch 13/30
21/21 [=====] - 7s 342ms/step - loss: 0.0406 - accuracy: 1.0000 - val_loss: 0.4615 - val_accuracy: 0.8278
Epoch 14/30
21/21 [=====] - 7s 345ms/step - loss: 0.0379 - accuracy: 0.9970 - val_loss: 0.4260 - val_accuracy: 0.8212
Epoch 15/30
21/21 [=====] - 7s 342ms/step - loss: 0.0298 - accuracy: 1.0000 - val_loss: 0.4239 - val_accuracy: 0.8477
Epoch 16/30
21/21 [=====] - 7s 341ms/step - loss: 0.0310 - accuracy: 1.0000 - val_loss: 0.4310 - val_accuracy: 0.8609
Epoch 17/30
```

Figura 28: Proceso de entrenamiento del modelo

```

21/21 [=====] - 7s 342ms/step - loss: 0.0298 - accuracy: 1.0000 - val_loss: 0.4239 - val_accuracy: 0.8477
Epoch 16/30
21/21 [=====] - 7s 341ms/step - loss: 0.0310 - accuracy: 1.0000 - val_loss: 0.4310 - val_accuracy: 0.8609
Epoch 17/30
21/21 [=====] - 7s 333ms/step - loss: 0.0248 - accuracy: 1.0000 - val_loss: 0.4309 - val_accuracy: 0.8411
Epoch 18/30
21/21 [=====] - 7s 344ms/step - loss: 0.0219 - accuracy: 1.0000 - val_loss: 0.4253 - val_accuracy: 0.8411
Epoch 19/30
21/21 [=====] - 7s 344ms/step - loss: 0.0248 - accuracy: 1.0000 - val_loss: 0.4128 - val_accuracy: 0.8477
Epoch 20/30
21/21 [=====] - 7s 338ms/step - loss: 0.0232 - accuracy: 1.0000 - val_loss: 0.4197 - val_accuracy: 0.8278
Epoch 21/30
21/21 [=====] - 7s 340ms/step - loss: 0.0213 - accuracy: 1.0000 - val_loss: 0.4342 - val_accuracy: 0.8212
Epoch 22/30
21/21 [=====] - 7s 342ms/step - loss: 0.0183 - accuracy: 1.0000 - val_loss: 0.4517 - val_accuracy: 0.8477
Epoch 23/30
21/21 [=====] - 7s 344ms/step - loss: 0.0182 - accuracy: 1.0000 - val_loss: 0.4552 - val_accuracy: 0.8146
Epoch 24/30
21/21 [=====] - 7s 341ms/step - loss: 0.0158 - accuracy: 1.0000 - val_loss: 0.4228 - val_accuracy: 0.8411
Epoch 25/30
21/21 [=====] - 7s 342ms/step - loss: 0.0146 - accuracy: 1.0000 - val_loss: 0.4289 - val_accuracy: 0.8609
Epoch 26/30
21/21 [=====] - 7s 340ms/step - loss: 0.0116 - accuracy: 1.0000 - val_loss: 0.4224 - val_accuracy: 0.8609
Epoch 27/30
21/21 [=====] - 7s 342ms/step - loss: 0.0117 - accuracy: 1.0000 - val_loss: 0.4364 - val_accuracy: 0.8543
Epoch 28/30
21/21 [=====] - 7s 340ms/step - loss: 0.0116 - accuracy: 1.0000 - val_loss: 0.4559 - val_accuracy: 0.8477
Epoch 29/30
21/21 [=====] - 7s 344ms/step - loss: 0.0111 - accuracy: 1.0000 - val_loss: 0.4384 - val_accuracy: 0.8543
Epoch 30/30
21/21 [=====] - 7s 344ms/step - loss: 0.0095 - accuracy: 1.0000 - val_loss: 0.4450 - val_accuracy: 0.8411

```

Figura 29: Continuación del proceso de entrenamiento del modelo

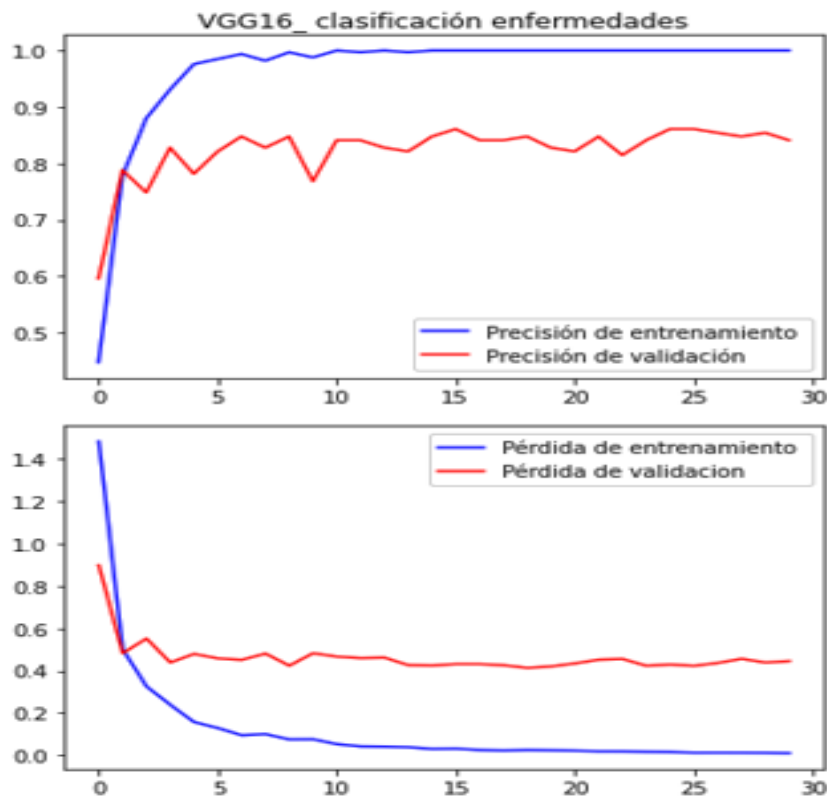


Figura 30: Gráficas de entrenamiento y validación del modelo

4.1.4. Evaluación del análisis

Escala de ponderaciones

Bajo = 5, Intermedio = 4, Moderado= 3, Excesivo = 2, Pésimo = 1

Tabla 22: Evaluación del análisis de los modelos de identificación

Parámetro de evaluación	Procesamiento Tradicional					Procesamiento con inteligencia artificial				
	B	I	M	P	E	B	I	M	P	E
Ponderación										
Valores de pérdida			✓			✓				
Fallos de predicción			✓			✓				
Tiempo de respuesta		✓					✓			
Tiempo de Ejecución			✓					✓		
Tiempo de entrenamiento					✓	✓				
Tiempo empleado en preprocesamiento					✓	✓				
Total	0	4	9	0	4	20	4	3	0	0
Resultado análisis	17 → 56,66 %					27 → 90%				

Nota: muestra el índice porcentual de la exactitud del procesamiento de imágenes haciendo uso de inteligencia artificial.

4.1.5. Propuesta

La propuesta de app de procesamiento de imágenes para identificación de dos tipos de enfermedades en plantas de pimiento, se comenzó a elaborar a partir de los resultados de la investigación y los acercamientos del investigador a los agricultores, si la herramienta está realizada en base a las necesidades del agricultor de la comunidad de Pusir Grande, esta será usada para la función estipulada en el inicio del párrafo, para el desarrollo del aplicativo se usó de la metodología de desarrollo de software tradicional denominada proceso incremental con sus 4 fases de trabajo análisis, diseño, codificación, pruebas. Así como también, se aplicó la normativa técnica ISO/IEC 15504 SPICE para realizar una matriz en Excel encargada de evaluar la calidad del software, con resultados de pruebas funcionales y aceptación de los agricultores ver anexo 13.

4.1.5.1. Situación Actual

La actividad de detección de estas enfermedades en la comunidad de Pusir Grande es de forma tradicional, tomando las hojas de los cultivos para la identificación a la medida del ojo humano,

en la actualidad capturan una fotografía del cultivo enfermo y la envían a un técnico a través de la red social WhatsApp para luego esperar la respuesta y el respectivo tratamiento.

4.1.5.2. Situación Ideal

La app de reconocimiento de imágenes permite al agricultor ver una posible afectación de la planta con un respectivo correctivo, facilitando al técnico las labores de visita a los sembríos.

4.1.6 Introducción

Esta investigación tiene como propósito entregar una herramienta de identificación de enfermedades foliares tomando como cultivo de ejemplo el pimiento, su funcionamiento hace uso de la inteligencia artificial y busca que la tarea de detección de enfermedades sea más pronta a base de tecnología, donde se aplica técnicas de procesamiento de imágenes para mejorar la calidad de las fotografías, aplicar filtros y obtener características que ayudan a pronosticar el padecimiento. Para el ojo humano es dificultoso obtener las características no comunes que presenta la planta con una simple revisión de la hoja afectada, para mejorar esto se entrenó un modelo de red neuronal convolucional a base de un paquete de imágenes que posee hojas con síntomas de dos enfermedades Mancha gris (*Stemphylium solani*) y Oidiopsis (*Oidiopsis taurica* - *Leveillula taurica*). El modelo fue implementado en una aplicación móvil desarrollada en Android Studio usando el lenguaje de programación java. Con fundamento en la investigación de la problemática y los resultados del levantamiento de requisitos a través de las técnicas de investigación, se pudo establecer las directrices para el desarrollo de la solución informática contemplando la experiencia en el desarrollo de software obtenido durante la formación académica.

4.1.7. Metodología de proceso incremental

4.1.7.1. Fase de análisis

Durante esta etapa se procedió a analizar los factores que se presentan en el proyecto, con el propósito de identificar información y requisitos relevantes con relación a la problemática de la investigación, se seleccionó los más fundamentales que engloban el problema y se examinó los componentes posibles del software.

a) Público Objetivo

Se consideró como público objetivo a los individuos de la comunidad de Pusir Grande que se dediquen a la actividad agrícola.

b) Requisitos del aplicativo móvil

El sistema debe contar con acceso a internet debido a que su alojamiento de base de datos se encuentra en un hosting en la nube, dentro de sus requisitos funcionales debemos considerar su

versión que tiene compatibilidad para dispositivos inteligentes con Android a partir de su versión 6. También dispone de autores que se encuentran involucrados con el sistema.

- **Administrador:** Es quien lleva la administración de todo el sistema.
- **Agricultor:** Es quien hará uso del aplicativo

4.1.7.2. Fase de diseño

En esta fase procedemos a entregar una visión global del diseño, se incluyeron las tecnologías que se van a utilizar para hacer posible la solución, en la figura 26 muestra la integración de tecnologías detalladas paso a paso hasta llegar a un resultado que sería nuestra aplicación móvil como vista final al usuario.



Figura 31: Visión global del diseño

a) Detallado del diseño

- **Entrenamiento del modelo e integración con Android Studio.** Es donde preparamos el modelo previo a su integración con Android.

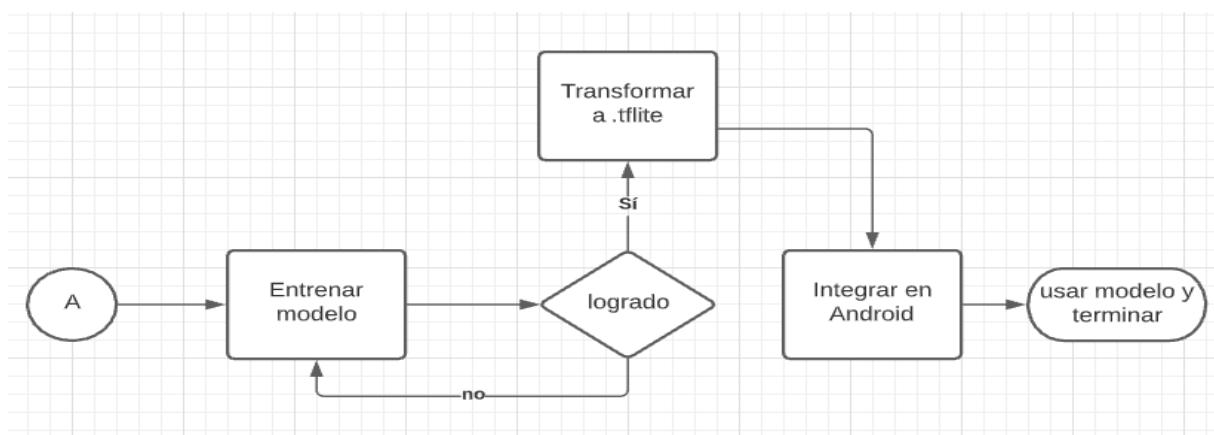


Figura 32: Diseño de la integración del modelo entrenado a Android

- **Alojamiento.** Se consideró hacer uso de la cuenta gratuita de 000webhost ya que en esta versión es uno de los hostings más completos.

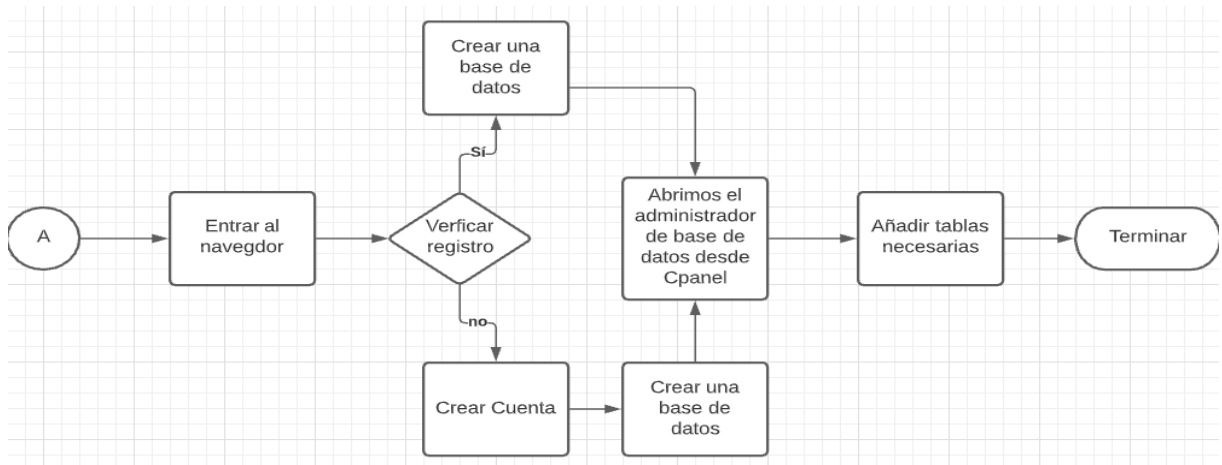


Figura 33: Creación de cuenta en 000webhost y base de datos online

- **Diseño de las interfaces de usuario.** Son las vistas que interactúan con el usuario.
 - ✓ Inicio de sesión
 - ✓ Registro de usuario
 - ✓ Registro de Cultivo
 - ✓ Identificar enfermedad
 - ✓ Recomendar tratamiento



Figura 34: Interfaces de usuario

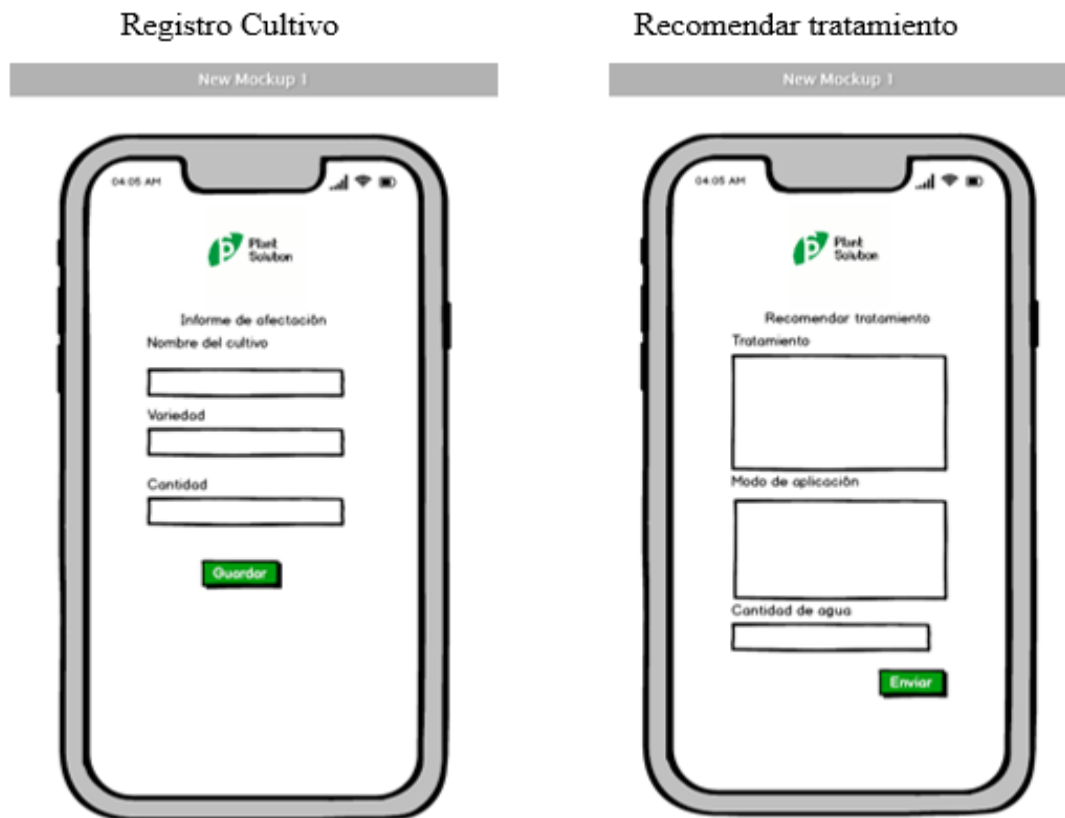
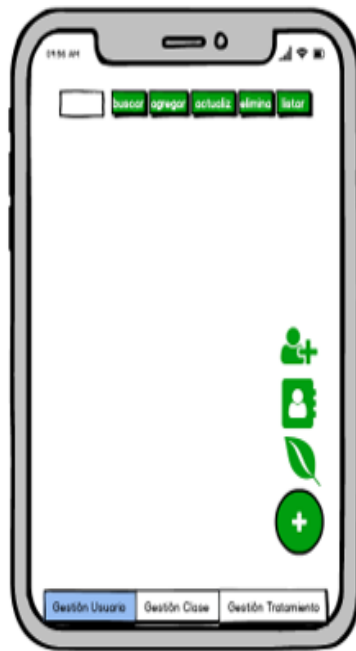


Figura 35: Continuación interfaces de usuario

- **Diseño de las interfaces de administración.** Son las vistas que no interactúan con el usuario a éstas solo tiene acceso el o los administradores del sistema.
 - ✓ Panel de la administración
 - ✓ Login Super Usuario
 - ✓ Registro Super usuario
 - ✓ Creación de clase (nueva enfermedad)

Panel de administración



Registro nueva clase



Figura 36: Diseño interfaces de administración

Login super usuario



Registro super usuario



Figura 37: Continuación diseño de interfaces de administración

b) Diagramas de flujo

✓ **Administrador:**

Aquí se muestra el diagrama de flujo de las funciones del administrador del sistema.

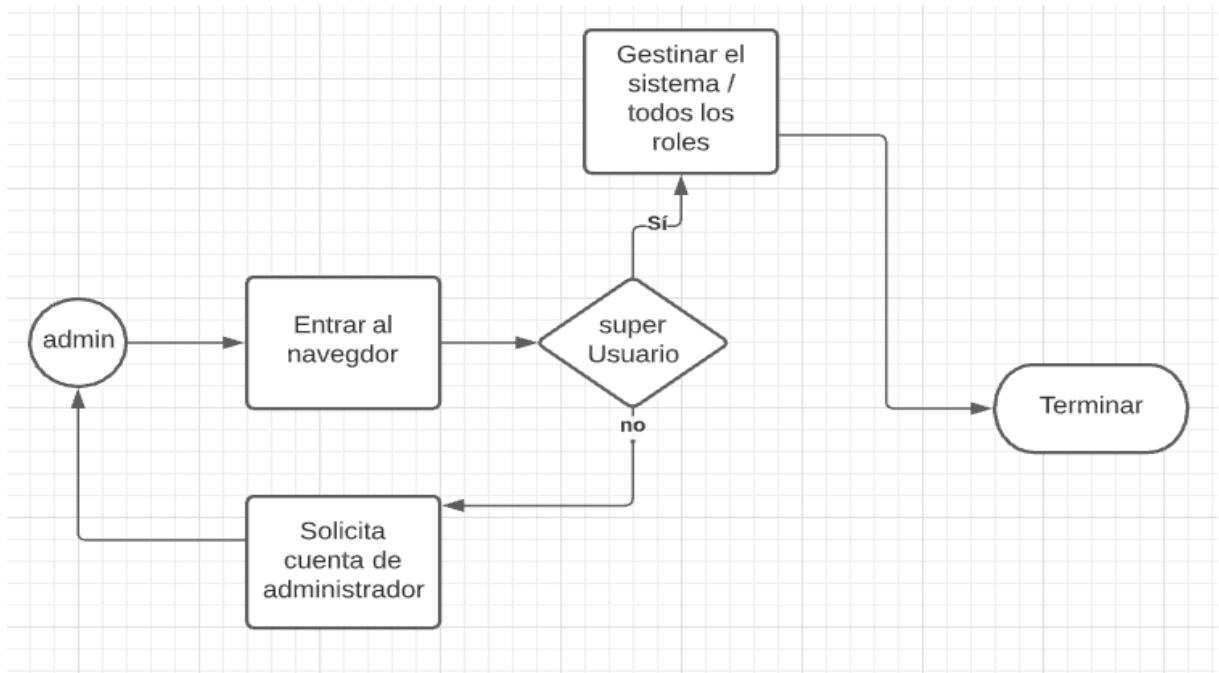


Figura 38: Proceso administrador

✓ **Agricultor**

Aquí se muestra el diagrama de flujo de las acciones que puede realizar en el sistema el usuario/agricultor.

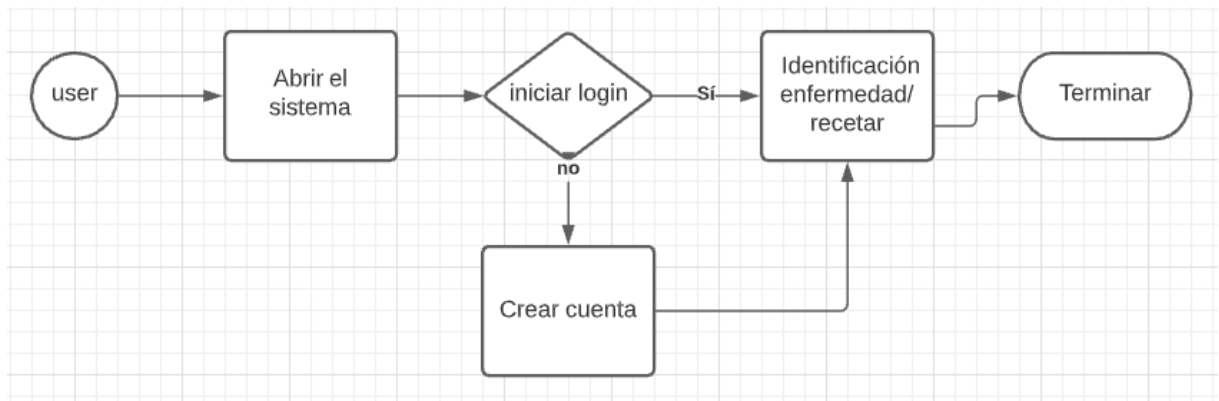


Figura 39: Proceso agricultor

c) **Modelo de la base de datos.**

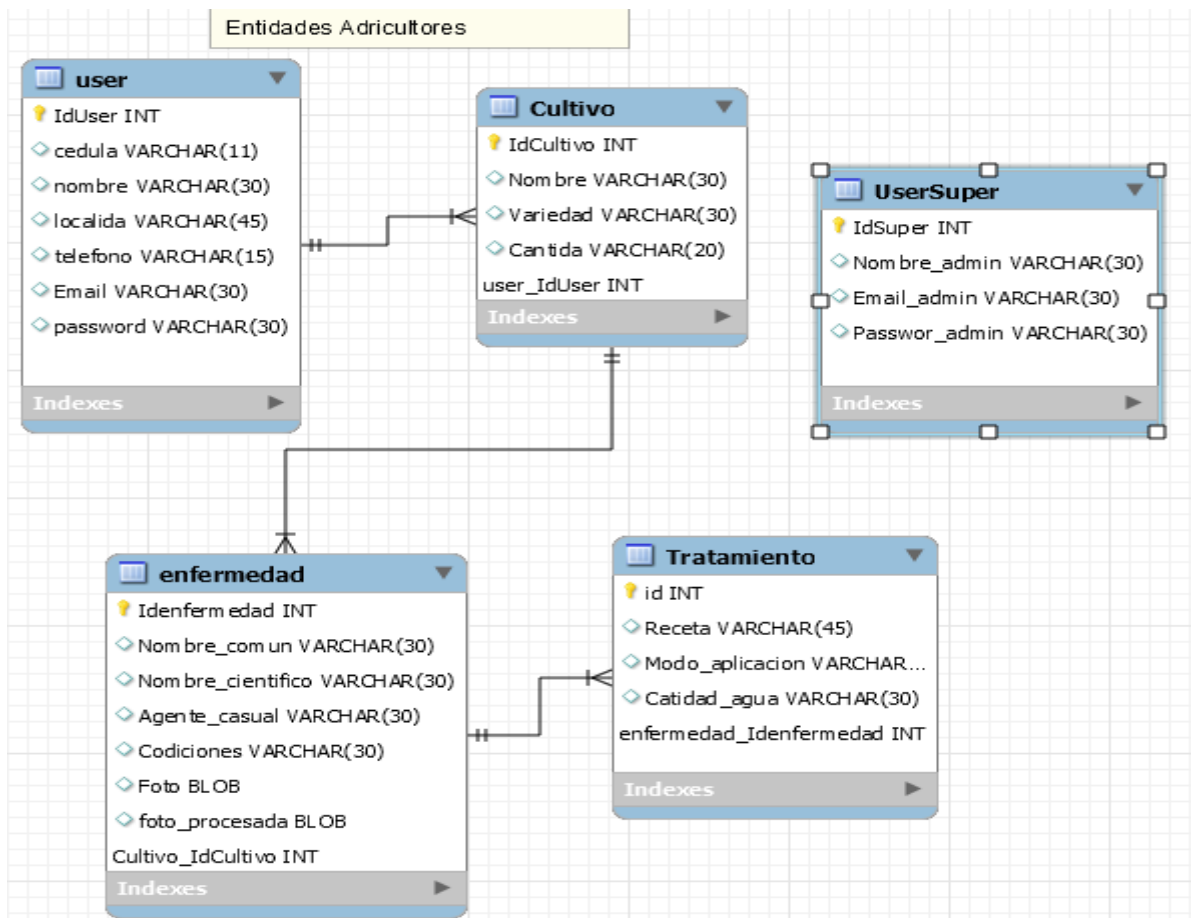


Figura 40: Modelo físico base de datos

d) **Entidades de la base de datos.**

Tabla 23: Diccionario de datos Entidad user-agricultor

Atributo	Tipo	Dimensión	Descripción
IdUser	INT	10	Primary Key
Cédula	VARCHAR	11	NO NULL
Nombre	VARCHAR	30	NO NULL
Localidad	VARCHAR	45	NO NULL
Teléfono	VARCHAR	15	NO NULL
Email	VARCHAR	30	NO NULL
Password	VARCHAR	30	NO NULL

Tabla 24: Diccionario de datos entidad Cultivo

Atributo	Tipo	Dimensión	Descripción
Idcultivo	INT	10	Primary key
Nombre	VARCHAR	30	NO NULL
Variedad	VARCHAR	30	NO NULL
Cantidad	VARCHAR	20	NO NULL
IdUser	INT		CLAVE FORÁNEA

Tabla 25: Diccionario de datos entidad enfermedad

Atributo	Tipo	Dimensión	Descripción
Idenfermedad	INT	10	Primary key
Nombre común	VARCHAR	30	NO NULL
Nombre científico	VARCHAR	30	NO NULL
Agente Causal	VARCHAR	30	NO NULL
Condiciones	VARCHAR	30	NO NULL
Foto	BLOB		NO NULL
Foto Procesada	BLOB		NO NULL
Idcultivo	INT		CLAVE FORÁNEA

Tabla 26: Diccionario de datos entidad tratamiento

Atributo	Tipo	Dimensión	Descripción
Id	INT	10	Primary key
Receta	VARCHAR	45	NO NULL
Modo aplicación	VARCHAR	45	NO NULL
Cantidad Agua	VARCHAR	30	NO NULL
Idenfermedad	INT		CLAVE FORÁNEA

Tabla 27: Diccionario de datos entidad super usuario

Atributo	Tipo	Dimensión	Descripción
Idsuper	INT	10	Primary key
Nombre admin	VARCHAR	30	NO NULL
Email admin	VARCHAR	30	NO NULL
Password admin	VARCHAR	30	NO NULL

e) **Diseño rápido de prototipos**

En esta fase procedemos a ir desarrollando nuestro prototipo que va a ser probado funcionalmente antes de obtener el diseño final de la solución.

- **Interfaz de inicio de sesión / agricultor**

Al abrir la aplicación de detección de enfermedades tendremos un panel de inicio de sesión donde debemos registrar nuestras credenciales de acceso, estas son correo electrónico y contraseña que deben ser previamente definidas al registrarnos en el sistema. En esta misma interfaz encontraremos el apartado crear cuenta que nos dirige al formulario de registro, también se lista la dirección establecida solo para adiestradores del sistema



Figura 41: Inicio del sistema de detección de enfermedad

- **Interfaz de registro de usuario / agricultor**

Esta interfaz se trata de un formulario de registro donde el usuario puede registrar sus datos entre ellos se define las credenciales de acceso al sistema, así como también después de su registro le dirigirá a otro formulario donde el agricultor puede registrar su cultivo.

Plant Solution

Crear Cuenta

Cédula

Nombre y apellido

Localidad

Teléfono

Correo electrónico

Contraseña

Register

Figura 42: Creación de cuenta de usuario/agricultor

- **Interfaz de registro de cultivo / agricultor**

Esta es la continuación a la obtención de cuenta en el sistema, aquí es donde se registrará la cantidad y variedad de cultivo que el agricultor dispone, sobre este se aplicará la identificación de la enfermedad.

Plant Solution

Registre su cultivo

Nombre del cultivo

Cantidad de plantas

Variedad del cultivo

Register

Figura 43: Registro de cultivo al sistema

- **Interfaz de identificación de la enfermedad/ agricultor**

Es en esta interfaz es donde se hará uso del modelo de identificación de enfermedades previamente entrenado, consta de apartados como selección de imagen con carga desde dispositivo y con captura por fotografía haciendo uso de la cámara del dispositivo, además también se evidencia en literal de la predicción que es el encargado de llamar al modelo, para que este identifique la imagen como enfermedad o no y muestre un posible tratamiento que se debe aplicar. También cuenta con un apartado oculto que se activa después de la predicción que le dirige a una interfaz donde el agricultor puede dejar su recomendación como tratamiento si así lo desea.



Figura 44: Identificación de la enfermedad

- **Interfaz de tratamiento/ agricultor**

El agricultor puede dejar un tratamiento dentro del sistema con el objetivo de alimentarlo para que en las nuevas predicciones de la enfermedad pueda presentar nuevos tratamientos.



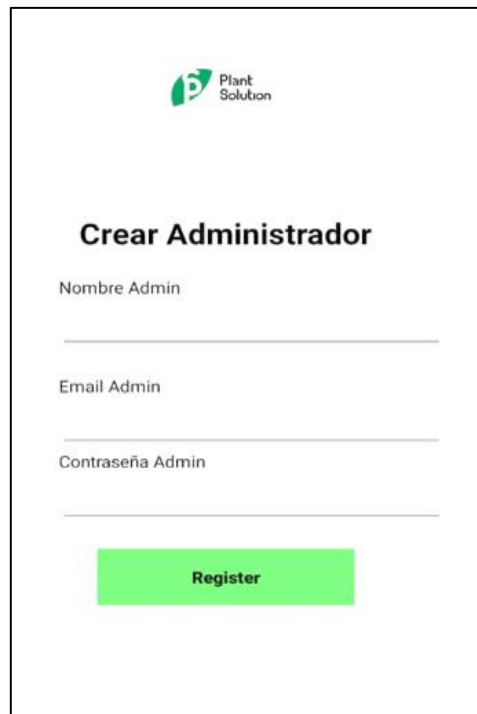
Figura 45: Recomendación tratamiento / agricultor

- **Interfaz de inicio de sesión/ administrador**



Figura 46: Interfaz de ingreso a la administración del sistema administración

- **Interfaz de registro/ administrador**



The screenshot shows a registration form titled "Crear Administrador" for the "Plant Solution" application. At the top left is the "Plant Solution" logo. The form contains three input fields: "Nombre Admin", "Email Admin", and "Contraseña Admin". Below these fields is a prominent green button labeled "Register".

Figura 47: Interfaz de creación de un nuevo administrador

- **Interfaz de registro de nueva clase – enfermedad / administrador**



The screenshot shows a registration form titled "Obtener patrones" for the "Plant Solution" application. It features five input fields: "Nombre Común", "Nombre Científico", "Condiciones", "Tratamiento", and "Agente Casual". Below the fields are two "Plant Solution" logos. At the bottom, there are four green buttons arranged in a 2x2 grid: "Seleccionar imagen", "Cámara", "Procesar", and "Guardar".

Figura 48: interfaz de registro de una nueva enfermedad

- **Interfaz de administración del sistema / administrador**

Figura 49: Panel de administración, gestión a base de datos

c) Diagramas de caso de uso

Administrador del sistema que puede hacer:

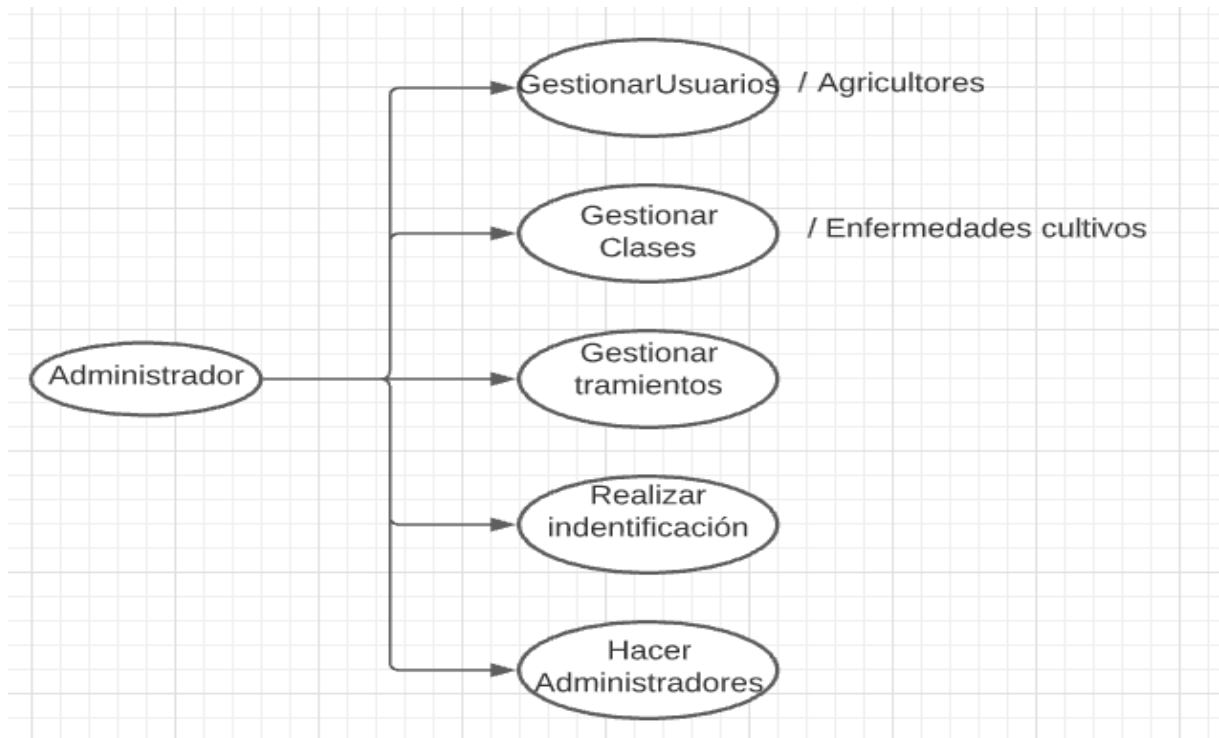


Figura 50: Caso de uso gestión de usuarios / administración

Agricultor usuario del sistema que puede hacer:

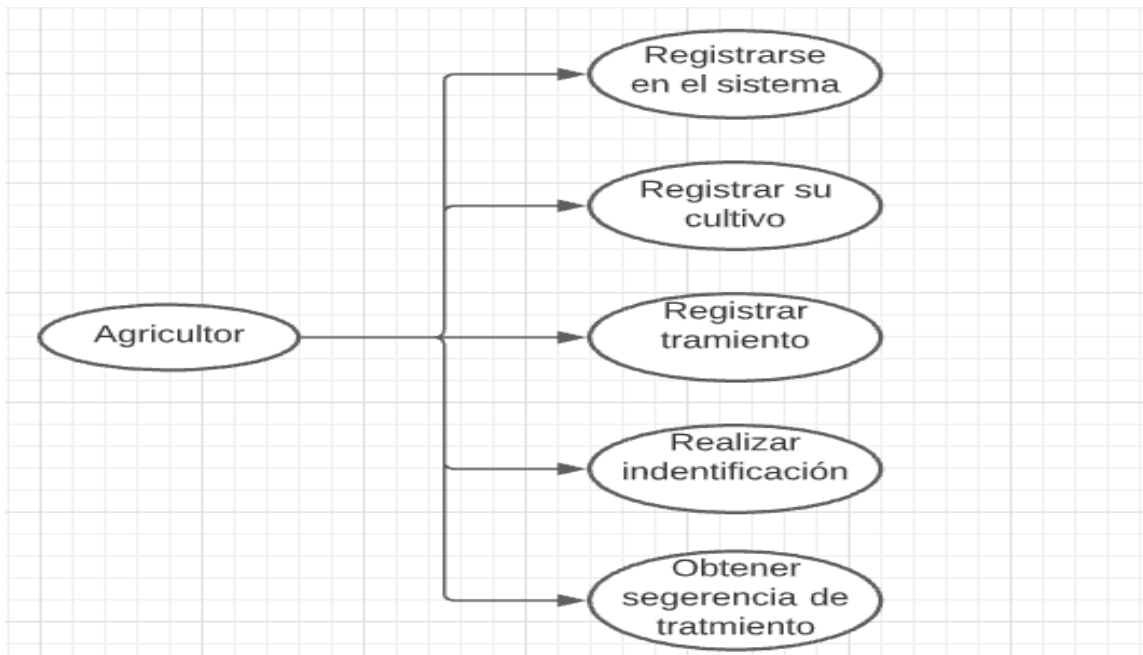


Figura 51: Caso de usos usuarios del sistema/agricultores

4.1.7.3. Fase de codificación

La codificación del aplicativo fue realizada acogiéndose a la programación bajo módulos usados en las actuales apps de Android.

En el módulo de Python se realizó la codificación para la preparación del modelo de Deep Learning de clasificación de imágenes, por conflictos de la fuerza computacional para la programación de este módulo se hizo uso de Google Colab.

```
from google.colab import files
uploaded = files.upload()

[ ] import zipfile
import io
data = zipfile.ZipFile(io.BytesIO(uploaded['hoy.zip']), 'r')
data.extractall()

[ ] # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os

[ ] path = "./dataset"
os.listdir(path)

['Validar', 'Entrenar', 'comparar']
```

Figura 52: Subida y extracción del dataset de las imágenes del cultivo

```
[ ] train_path = os.path.join(path, "Entrenar")
print(os.listdir(train_path))
print("*"*100)
test_path = os.path.join(path, "Validar")
print(os.listdir(test_path))

['Saludable', 'Oidiopsis', 'MachaG', 'Todo']
*****
['Saludable', 'Oidiopsis', 'MachaG', 'Todo']

[ ] from glob import glob
folders = glob("./dataset/Entrenar/*")
folders

['./dataset/Entrenar/Saludable',
 './dataset/Entrenar/Oidiopsis',
 './dataset/Entrenar/MachaG',
 './dataset/Entrenar/Todo']
```

Figura 53: Definición de las clases del modelo de transfer learning entrenamiento y validación

```
[ ] from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential

[ ] SIZE = [224, 224]

[ ] from tensorflow.keras.applications.vgg16 import VGG16

[ ] vgg16 = VGG16(input_shape=SIZE + [3], weights="imagenet", include_top=False)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\_weights\_tf\_dim\_ordering\_tf\_data\_format.h5
58892288/58889256 [=====] - 1s 0us/step
58900480/58889256 [=====] - 1s 0us/step
```

Figura 54: Importación de paquetes y activación del modelo VGG16 de machine learning

```
[ ] for layer in vg16.layers:
    layer.trainable = False

[ ] x = Flatten()(vg16.output)

[ ] prediction = Dense(len(folders), activation="softmax")(x)

modelvg = Model(inputs=vg16.input, outputs=prediction)

[ ] modelvg.summary()
```

```
Model: "model"

Layer (type)                Output Shape                Param #
-----
input_1 (InputLayer)        [(None, 224, 224, 3)]      0
block1_conv1 (Conv2D)        (None, 224, 224, 64)       1792
block1_conv2 (Conv2D)        (None, 224, 224, 64)       36928
block1_pool (MaxPooling2D)   (None, 112, 112, 64)       0
block2_conv1 (Conv2D)        (None, 112, 112, 128)      73856
block2_conv2 (Conv2D)        (None, 112, 112, 128)      147584
block2_pool (MaxPooling2D)   (None, 56, 56, 128)        0
```

Figura 55: Traspaso de capas entrenadas del modelo VGG16 al modelo a entrenarse

```
[ ] modelvg.compile(loss="categorical_crossentropy", metrics=["accuracy"], optimizer="adam")
```

Figura 56: Compilación del modelo transferido

```
[ ] train_datagen_vg16 = ImageDataGenerator(rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2)

test_datagen_vg16 = ImageDataGenerator(rescale=1./255)

[ ] training_set_vg16 = train_datagen_vg16.flow_from_directory(train_path,
    target_size=(224, 224),
    batch_size=16,
    class_mode="categorical", shuffle=True)

Found 333 images belonging to 4 classes.

[ ] testing_set_vg16 = test_datagen_vg16.flow_from_directory(test_path,
    target_size=(224, 224),
    batch_size=16,
    class_mode="categorical", shuffle=False)

Found 151 images belonging to 4 classes.
```

Figura 57: Preprocesado de las imágenes del dataset en las clases de entrenamiento y validación

```
[ ] r_vg16 = modelvg.fit_generator(training_set_vg16,
                                validation_data=testing_set_vg16,
                                epochs=30,
                                steps_per_epoch=len(training_set_vg16),
                                validation_steps=len(testing_set_vg16)
                                )

/usr/local/lib/python3.7/dist-packages/keras/engine/training.py:1972: UserWarning: `Model.fit_generator` is deprecated and
warnings.warn("`Model.fit_generator` is deprecated and ")
Epoch 1/30
21/21 [=====] - 47s 712ms/step - loss: 1.4835 - accuracy: 0.4474 - val_loss: 0.8982
Epoch 2/30
21/21 [=====] - 7s 342ms/step - loss: 0.5012 - accuracy: 0.7778 - val_loss: 0.4818
Epoch 3/30
21/21 [=====] - 7s 342ms/step - loss: 0.3255 - accuracy: 0.8799 - val_loss: 0.5519
Epoch 4/30
21/21 [=====] - 7s 343ms/step - loss: 0.2383 - accuracy: 0.9309 - val_loss: 0.4384
Epoch 5/30
21/21 [=====] - 7s 346ms/step - loss: 0.1567 - accuracy: 0.9760 - val_loss: 0.4788
Epoch 6/30
```

Figura 58: Reentrenar el modelo transferido con las nuevas clases añadidas

```
[ ] labels = '\n'.join(sorted(training_set.class_indices.keys()))
with open('labels.txt', 'w') as f:
    f.write(labels)
```

```
[ ] import tensorflow as tf
# Convert the model.
converter = tf.lite.TFLiteConverter.from_keras_model(modelvg)
tflite_model = converter.convert()
# Save the model.
with open('linear.tflite', 'wb') as f:
    f.write(tflite_model)
```

INFO:tensorflow:Assets written to: /tmp/tmpmk94hh_n/assets

Figura 59: Conversión de modelo a extensión compatible con Android y guardar

Módulo de Android para codificar este módulo se hizo uso del lenguaje de programación java.

- **Codificación de las clases java**

```

package com.example.plan_solutionv2;

import ...

public class LoginActivity extends AppCompatActivity {

    EditText inputEmail, inputPassword;
    Button buttonLogin;
    TextView textCreateAccount,txvSuperUser;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        textCreateAccount = findViewById(R.id.lnkRegister);
        inputEmail = findViewById(R.id.txtloginEmail);
        inputPassword = findViewById(R.id.txtloginPwd);
        buttonLogin = findViewById(R.id.btnlogin);
        txvSuperUser= findViewById(R.id.txvSuperUser);
        // crear cuenta usuario
        textCreateAccount.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(), RegisterActivity.class));
            }
        });
    }
}

```

Figura 60: Codificación clase java login

```

// Iniciar como super usuario
txvSuperUser.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(getApplicationContext(), LoginSuper.class));
    }
});

// inicio login

buttonLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (inputEmail.getText().toString().equals("")) {
            Toast.makeText(LoginActivity.this, "Por favor ingrese su correo electrónico", Toast.LENGTH_SHORT).show();
        } /*else if (!emailValidator(inputEmail.getText().toString())) {
            Toast.makeText(MainActivity.this, "Please Field Valid Email", Toast.LENGTH_SHORT).show();
        }*/ else if (inputPassword.getText().toString().equals("")) {
            Toast.makeText(LoginActivity.this, "Por favor, ingrese contraseña", Toast.LENGTH_SHORT).show();
        } else {
            login();
        }
    }
});
}
}

```

Figura 61: Continuación codificación clase java login

```

public static void hideSoftKeyboard(Activity activity) {
    InputMethodManager inputMethodManager =
        (InputMethodManager) activity.getSystemService(
            Activity.INPUT_METHOD_SERVICE);
    inputMethodManager.hideSoftInputFromWindow(
        activity.getCurrentFocus().getWindowToken(), 0);
}

private void login() {

    final ProgressDialog progressDialog = new ProgressDialog(LoginActivity.this);
    progressDialog.setTitle("Espere por favor");
    progressDialog.setMessage("Iniciando sesión...");
    progressDialog.setCancelable(false);
    progressDialog.show();
    NetworkService networkService = NetworkClient.getClient().create(NetworkService.class);
    Call<LoginResponseModel> login = networkService.login(inputEmail.getText().toString(), inputPassword.getText().toString());
    login.enqueue(new Callback<LoginResponseModel>() {
        @Override
        public void onResponse(@NonNull Call<LoginResponseModel> call, @NonNull Response<LoginResponseModel> response) {
            LoginResponseModel responseBody = response.body();
            if (responseBody != null) {
                if (responseBody.getSuccess().equals("1")) {
                    SharedPreferences preferences = getSharedPreferences(Constants.PREFERENCE_NAME, MODE_PRIVATE);
                    SharedPreferences.Editor editor = preferences.edit();
                    editor.putBoolean(Constants.KEY_ISE_LOGGED_IN, true);
                    editor.putString(Constants.KEY_USERNAME, responseBody.getUserDetailObject().getUserDetails().get(0).getFirst());
                    /*editor.putString(Constants.KEY_LASTNAME, responseBody.getUserDetailObject().getUserDetails().get(0).getLas

```

Figura 62: Continuación codificación clase java login

```

                    editor.putString(Constants.KEY_EMAIL, responseBody.getUserDetailObject().getUserDetails().get(0).getEmail());
                    editor.apply();
                    Toast.makeText(LoginActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                    startActivity(new Intent(getApplicationContext(), IdentificacionActivity.class));
                    finish();
                } else {
                    Toast.makeText(LoginActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
            progressDialog.dismiss();
        }

        @Override
        public void onFailure(@NonNull Call<LoginResponseModel> call, @NonNull Throwable t) {
            progressDialog.dismiss();
        }
    });
}

```

Figura 63: Continuación codificación clase java login

```

package com.example.plan_solutionv2;

import ...

public class RegisterActivity extends AppCompatActivity {

    EditText inputCedula, inputName, inputPassword, inputemail, inputDirection, inputPhone;
    Button buttonRegister;
    TextView linkLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        inputCedula = findViewById(R.id.txtCedula);
        inputName = findViewById(R.id.txtName);
        inputDirection = findViewById(R.id.txtDirection);
        inputPhone = findViewById(R.id.txtPhone);
        inputemail = findViewById(R.id.txtEmail);
        inputPassword = findViewById(R.id.txtPwd);
        linkLogin = findViewById(R.id.lnkLogin);
        linkLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent( packageContext: RegisterActivity.this, IdentificacionActivity.class);
                startActivity(intent);
                finish();
            }
        });
    }
};

```

Figura 64: Codificación de clase java registro de usuarios/agricultor

```

buttonRegister = findViewById(R.id.btnregister);
buttonRegister.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {

        if(inputCedula.getText().toString().equals("")) {
            Toast.makeText( context: RegisterActivity.this, text: "Ingrese la cédula", Toast.LENGTH_SHORT).show();
        } else if (inputName.getText().toString().equals("")) {
            Toast.makeText( context: RegisterActivity.this, text: "Ingrese el nombre", Toast.LENGTH_SHORT).show();
        } else if (inputemail.getText().toString().equals("")) {
            Toast.makeText( context: RegisterActivity.this, text: "Ingrese correo electrónico", Toast.LENGTH_SHORT).show();
        } else if (inputPassword.getText().toString().equals("")) {
            Toast.makeText( context: RegisterActivity.this, text: "Introducir la contraseña", Toast.LENGTH_SHORT).show();
        } else if (inputDirection.getText().toString().equals("")) {
            Toast.makeText( context: RegisterActivity.this, text: "Ingrese la dirección", Toast.LENGTH_SHORT).show();
        } else if (inputPhone.getText().toString().equals("")) {
            Toast.makeText( context: RegisterActivity.this, text: "Ingrese el teléfono", Toast.LENGTH_SHORT).show();
        } else {

            HashMap<String, String> params = new HashMap<>();
            params.put("cedula", inputCedula.getText().toString());
            params.put("name", inputName.getText().toString());
            params.put("home", inputDirection.getText().toString());
            params.put("contact", inputPhone.getText().toString());
            params.put("email", inputemail.getText().toString());
            params.put("password", inputPassword.getText().toString());
            register(params);
        }
    }
};

```

Figura 65: Continuación codificación clase java registro de usuario/agricultor

```

    }
});
}
private void register(HashMap<String, String> params) {

    final ProgressDialog progressDialog = new ProgressDialog( context: RegisterActivity.this);
    progressDialog.setTitle("Espera por favor");
    progressDialog.setMessage("Lo estamos registrando...");
    progressDialog.setCancelable(false);
    progressDialog.show();

    NetworkService networkService = NetworkClient.getClient().create(NetworkService.class);
    Call<RegistrationResponseModel> registerCall = networkService.register(params);
    registerCall.enqueue(new Callback<RegistrationResponseModel>() {
        @Override
        public void onResponse(@NonNull Call<RegistrationResponseModel> call, @NonNull Response<RegistrationResponseModel> response) {
            RegistrationResponseModel responseBody = response.body();
            if (responseBody != null) {
                if (responseBody.getSuccess().equals("1")) {
                    Toast.makeText( context: RegisterActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent( packageContext: RegisterActivity.this, PythonActivity.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText( context: RegisterActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
}
}

```

Figura 66: Continuación codificación clase java registro de usuario/agricultor

```

        progressDialog.dismiss();
    }

    @Override
    public void onFailure(@NonNull Call<RegistrationResponseModel> call, @NonNull Throwable t) {
        progressDialog.dismiss();
    }
});
}
}

```

Figura 67: Continuación codificación clase java registro de usuario/agricultor

```

package com.example.plan_solutionv2;

import androidx.appcompat.app.AppCompatActivity;

public class PythonActivity extends AppCompatActivity {
    //TextView textView;

    EditText inputcultivo, inputplantas, inputvariedad;
    Button buttoncultivo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_python);

        /* textView = (TextView)findViewById(R.id.textView);
        if (!Python.isStarted()) {
            Python.start(new AndroidPlatform(this));
        }
        Python py = Python.getInstance();
        PyObject pyObject = py.getModule("myscript");
        PyObject obj = pyObject.callAttr("main");
        textView.setText(obj.toString());*/

        //instanciar los elementos
        inputcultivo = findViewById(R.id.txtNombreCultivo);
        inputplantas = findViewById(R.id.txtNumPlanta);
        inputvariedad = findViewById(R.id.txtVariedad);
        buttoncultivo = findViewById(R.id.btnCultivo);
    }
}

```

Figura 68: Codificación clase java registro cultivo

```

buttoncultivo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if(inputcultivo.getText().toString().equals("")) {
            Toast.makeText(context: PythonActivity.this, text: "Ingrese en nombre del cultivo", Toast.LENGTH_SHORT).show();
        } else if (inputplantas.getText().toString().equals("")) {
            Toast.makeText(context: PythonActivity.this, text: "Ingrese el numero de plantas", Toast.LENGTH_SHORT).show();
        } else if (inputvariedad.getText().toString().equals("")) {
            Toast.makeText(context: PythonActivity.this, text: "Ingrese la variedad del cultivo", Toast.LENGTH_SHORT).show();
        } else {

            HashMap<String, String> params = new HashMap<>();
            params.put("cultivo", inputcultivo.getText().toString());
            params.put("plantas", inputplantas.getText().toString());
            params.put("variedad", inputvariedad.getText().toString());
            cultivo(params);
        }
    }
});
}
}

```

Figura 69: Continuación codificación clase java registro cultivo

```

private void cultivo(HashMap<String, String> params) {
    final ProgressDialog progressDialog = new ProgressDialog(context: PythonActivity.this);
    progressDialog.setTitle("Espere por favor");
    progressDialog.setMessage("Estamos procesando...");
    progressDialog.setCancelable(false);
    progressDialog.show();

    NetworkService networkService = NetworkClient.getClient().create(NetworkService.class);
    Call<CultivoResponseModel> cultivoCall = networkService.cultivo(params);
    cultivoCall.enqueue(new Callback<CultivoResponseModel>() {
        @Override
        public void onResponse(Call<CultivoResponseModel> call, Response<CultivoResponseModel> response) {
            CultivoResponseModel responseBody = response.body();
            if (responseBody != null) {
                if (responseBody.getSuccess().equals("1")) {
                    Toast.makeText(context: PythonActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(packageContext: PythonActivity.this, LoginActivity.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText(context: PythonActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
            progressDialog.dismiss();
        }
    });
}
}

```

Figura 70: Codificación clase java registro cultivo

```

}

@Override
public void onFailure(Call<CultivoResponseModel> call, Throwable t) {
    progressDialog.dismiss();
}

});
}
}
}

```

Figura 71: Codificación clase java registro cultivo

```

package com.example.plan_solutionv2;

import ...

public class TratamientoActivity extends AppCompatActivity {

    EditText inputtratamiento, inputaplicacion, inputagua;
    Button botontratamiento;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tratamiento);

        inputtratamiento = findViewById(R.id.TextMultiLine);
        inputaplicacion = findViewById(R.id.TextAplicacion);
        inputagua = findViewById(R.id.txtAgua);
        botontratamiento = findViewById(R.id.btnEnviar);

        botontratamiento.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (inputtratamiento.getText().toString().equals("")) {
                    Toast.makeText( context: TratamientoActivity.this, text: "Ingrese su recomendación", Toast.LENGTH_SHORT).show();
                } else if (inputaplicacion.getText().toString().equals("")) {
                    Toast.makeText( context: TratamientoActivity.this, text: "Ingrese el modo de aplicación", Toast.LENGTH_SHORT).show();
                } else if (inputagua.getText().toString().equals("")) {
                    Toast.makeText( context: TratamientoActivity.this, text: "Especifique si la aplicación requiere agua o no", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```

Figura 72: Codificación clase java tratamiento

```

        } else {
            HashMap<String, String> params = new HashMap<>();
            params.put("tratamiento", inputtratamiento.getText().toString());
            params.put("aplicacion", inputaplicacion.getText().toString());
            params.put("agua", inputagua.getText().toString());
            tratamiento(params);
        }
    });
}

public void tratamiento(HashMap<String, String> params){
    final ProgressDialog progressDialog = new ProgressDialog( context: TratamientoActivity.this);
    progressDialog.setTitle("Espere por favor");
    progressDialog.setMessage("Estamos procesando...");
    progressDialog.setCancelable(false);
    progressDialog.show();

    NetworkService networkService = NetworkClient.getClient().create(NetworkService.class);
    Call<TratamientoResponseModel> tratamiento = networkService.tratamiento(params);
    tratamiento.enqueue(new Callback<TratamientoResponseModel>() {
        @Override
        public void onResponse(Call<TratamientoResponseModel> call, Response<TratamientoResponseModel> response) {
            TratamientoResponseModel responseBody = response.body();
            if (responseBody != null) {
                if (responseBody.getSuccess().equals("1")) {
                    Toast.makeText( context: TratamientoActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent( packageContext: TratamientoActivity.this, IdentificacionActivity.class);
                    startActivity(intent);
                    finish();
                }
            }
        }
    });
}

```

Figura 73: Continuación codificación clase java tratamiento

```

    } else {
        Toast.makeText(context, TratamientoActivity.this, responseBody.getMessage(), Toast.LENGTH_SHORT).show();
    }
}
progressDialog.dismiss();
}

@Override
public void onFailure(Call<TratamientoResponseModel> call, Throwable t) {
    progressDialog.dismiss();
}
});
}
}

```

Figura 74: Continuación codificación clase java tratamiento

```

package com.example.plan_solutionv2;

import ...

public class IdentificacionActivity extends AppCompatActivity {
    // Aquí declaraciones de la clase glovales
    final int COD_GALERIA=10;
    final int COD_CATURAR=20;
    String agente0 = "Oidiopsis taurica/Leveillula taurica";
    String condicion0 = "Temperatura óptimas 25°C a 30°C se Acelera ";
    String trato0 = "Aplicar Azoxistrobina+Boscalid+Pyraclostrobin+Difeconazole +Trifloxistrobin";
    String agenteG = "Stemphylium solani";
    String condicionG = "Condiciones de alta humedad/lluvias ";
    String tratoG = "Aplicar Boscalid Pyraclostrobin, Azoxistrobina ";
    Button btnpredecir, btncamara, btngaleria;
    ImageView img;
    // String path;

    //Hacer Bitmap para procesar
    BitmapDrawable drawable;
    Bitmap bitmapProcesada;
    String imageString="";
    private Bitmap imagen;
    private TextView tv, tv2, tv3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_identificacion);
    }
}

```

Figura 75: Codificación clase java identificación de enfermedad

```

img = (ImageView) findViewById(R.id.imgIdentificar);
btnpredecir = (Button) findViewById(R.id.btnPredecir);
btngaleria = (Button) findViewById(R.id.btnSleccion);
btncamara = (Button) findViewById(R.id.btnFoto);
tv = (TextView) findViewById(R.id.txvRInforme);
tv2 = (TextView) findViewById(R.id.txvAgente);
tv3 = (TextView) findViewById(R.id.txvRecomendar);

// Tomar foto
btncamara.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intFoto = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intFoto, COD_CATURAR);
    }
});

// Funcion boton predecir
btnpredecir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        imagen= Bitmap.createScaledBitmap(imagen, dstWidth: 224, dstHeight: 224, filter: true);
        try {
            Modelvg model = Modelvg.newInstance(getApplicationContext());

            // Creates inputs for reference.
            TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 224, 224, 3}, DataType.FLOAT32);
            TensorImage tensorImage = new TensorImage(DataType.FLOAT32);
            tensorImage.load(imagen);
            ByteBuffer byteBuffer = tensorImage.getBuffer();
            inputFeature0.loadBuffer(byteBuffer);

```

Figura 76: Continuación codificación clase java identificación de enfermedad

```

Modelvg.Outputs outputs = model.process(inputFeature0);
TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();

// Releases model resources if no longer used.
model.close();
if(outputFeature0.getFloatArray()[0]==1.0){
    tv.setText("Mancha Bacteriana Gris");
    tv2.setText("Agente Casual:"+ agenteG+ "\n"+ "Condiciones:"+condicionG+"\n"+ "Tratamiento:"+tratoG);
    tv3.setText("Desea Recomendacion tratamiento");
}else if(outputFeature0.getFloatArray()[1]==1.0) {
    tv.setText("Oidiopsis/Ceniza");
    tv2.setText("Agente Casual:"+ agente0+ "\n"+ "Condiciones:"+condicion0+"\n"+ "Tratamiento:"+trato0);
    tv3.setText("Desea Recomendacion tratamiento");
}else if(outputFeature0.getFloatArray()[2]==1.0){
    tv.setText("Cultivo Saludable");
    tv2.setText("Esta fotografia no muestra afectación");
    tv3.setText("");
}else if(outputFeature0.getFloatArray()[3]==1.0){
    tv.setText("Clase no definida");
    tv2.setText("A ingresado un objeto no considerado en la investigacion");
    tv3.setText("");
}
// tv2.setText(outputFeature0.getFloatArray()[0)+"\n"+outputFeature0.getFloatArray()[1)+"\n"+outputFeatu
} catch (IOException e) {
    // TODO Handle the exception
}
}

```

Figura 77: Continuación codificación clase java identificación de enfermedad

```

public void obtenerImagen(View view) { cargarImagen(); }
private void cargarImagen() {
    Intent intentCargar = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    intentCargar.setType("image/*");
    startActivityForResult(intentCargar.createChooser(intentCargar, title "Abrir con"), COD_GALERIA);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode==RESULT_OK) {
        switch (requestCode) {
            case COD_GALERIA: // Aquí cargar imagen
                Uri miPath = data.getData();
                img.setImageURI(miPath);
                try {
                    imagen = MediaStore.Images.Media.getBitmap(this.getContentResolver(), miPath);
                } catch (IOException e) {
                    e.printStackTrace();
                }
                break;
            case COD_CATURAR:
                Bitmap bitmap = (Bitmap) data.getExtras().get("data");
                img.setImageBitmap(bitmap);
                imagen = (Bitmap) data.getExtras().get("data");
                break;
        }
    }
}
}
}

```

Figura 78: Continuación codificación clase java identificación de enfermedad

-
- **Codificación de las vistas**

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    tools:context=".LoginActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:paddingBottom="20dp">

        <ImageView
            android:id="@+id/imgLoginLogo"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:layout_marginTop="80dp"
            app:srcCompat="@drawable/logo" />

```

Figura 79: Codificación vista login usuario/agricultor

```

<TextView
    android:id="@+id/loginscrn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="50dp"
    android:text="Iniciar en el sistema"
    android:textColor="@color/black"
    android:textSize="25dp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/fstTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="20dp"
    android:text="Correo electrónico"
    android:textColor="@color/black" />

<EditText
    android:id="@+id/txtloginEmail"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

```

Figura 80: Continuación codificación vista login usuario/agricultor

```

<TextView
    android:id="@+id/secTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:text="Contraseña"
    android:textColor="@color/black" />

<EditText
    android:id="@+id/txtloginPwd"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:imeOptions="actionNext"
    android:inputType="textPassword"
    android:singleLine="true" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

```

Figura 81: Continuación codificación vista login usuario/agricultor

```

<Button
    android:id="@+id/btnlogin"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:background="@color/verde_logo"
    android:text="Empezar"
    android:textAllCaps="false"
    android:textColor="#000000"
    android:textStyle="bold" />
</LinearLayout>
<TextView
    android:id="@+id/lnkRegister"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp"
    android:gravity="center"
    android:text="Nueva cuenta"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:textStyle="bold" />

```

Figura 82: Continuación codificación vista login usuario/agricultor

```

        <TextView
            android:id="@+id/txvSuperUser"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp"
            android:gravity="center"
            android:text="Super Usuario"
            android:textColor="@color/black"
            android:textSize="20dp"
            android:textStyle="bold" />
    </LinearLayout>
</LinearLayout>

```

Figura 83: Continuación codificación vista login usuario/agricultor

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    tools:context=".RegisterActivity">
    <ImageView
        android:id="@+id/imgRegistroLogo"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        app:srcCompat="@drawable/logo" />
    <TextView
        android:id="@+id/loginscrn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:text="Crear Cuenta"
        android:textColor="@color/black"
        android:textSize="25dp"
        android:textStyle="bold" />

```

Figura 84: Codificación vista registro de usuario/agricultor

```

<TextView
    android:id="@+id/txvCedula"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="130dp"
    android:text="Cédula"
    android:textColor="@color/black" />
<EditText
    android:id="@+id/txtCedula"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="150dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />
<TextView
    android:id="@+id/fstTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="195dp"
    android:text="Nombre y apellido"
    android:textColor="@color/black" />

```

Figura 85: Continuación codificación vista registro de usuario/agricultor

```

<EditText
    android:id="@+id/txtName"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="215dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />
<TextView
    android:id="@+id/txvDirection"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="265dp"
    android:text="Localidad"
    android:textColor="@color/black" />
<EditText
    android:id="@+id/txtDirection"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="285dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

```

Figura 86: Continuación codificación vista registro de usuario/agricultor

```

<TextView
    android:id="@+id/txvPhone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="335dp"
    android:text="Teléfono"
    android:textColor="@color/black" />

<EditText
    android:id="@+id/txtPhone"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="355dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

<TextView
    android:id="@+id/secTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="405dp"
    android:text="Correo electrónico"
    android:textColor="@color/black" />

```

Figura 87: Continuación codificación vista registro de usuario/agricultor

```

<EditText
    android:id="@+id/txtEmail"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="425dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

<TextView
    android:id="@+id/thirdTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="480dp"
    android:text="Contraseña"
    android:textColor="@color/black" />

<EditText
    android:id="@+id/txtPwd"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="500dp"
    android:imeOptions="actionNext"
    android:inputType="textPassword"
    android:singleLine="true" />

```

Figura 88: Continuación codificación vista registro de usuario/agricultor

```

<Button
    android:id="@+id/btnregister"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="560dp"
    android:background="@color/verde_logo"
    android:text="Register"
    android:textAllCaps="false"
    android:layout_centerHorizontal="true"
    android:textColor="#000000"
    android:textStyle="bold" />

<TextView
    android:id="@+id/lnkLogin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="620dp"
    android:gravity="center"
    android:text=""
    android:textColor="@color/black"
    android:textSize="20dp"
    android:textStyle="bold" />
</RelativeLayout>

```

Figura 89: Continuación codificación vista registro de usuario/agricultor

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PythonActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:paddingBottom="20dp">

        <ImageView
            android:id="@+id/imgCultivoLogo"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_marginTop="40dp"
            app:srcCompat="@drawable/logo"
            android:layout_gravity="center" />

```

Figura 90: Codificación vista registro cultivo

```

<TextView
    android:id="@+id/txvCultivo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_marginTop="20dp"
    android:text="Registre su cultivo"
    android:textColor="@color/black"
    android:textSize="24sp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/txvNombreCultivo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="20dp"
    android:textColor="@color/black"
    android:text="Nombre del cultivo"/>

<EditText
    android:id="@+id/txtNombreCultivo"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

```

Figura 91: Continuación codificación vista registro cultivo

```

<TextView
    android:id="@+id/txvNumPlanta"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="20dp"
    android:textColor="@color/black"
    android:text="Cantidad de plantas"/>

<EditText
    android:id="@+id/txtNumPlanta"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

<TextView
    android:id="@+id/txvVariedad"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="20dp"
    android:textColor="@color/black"
    android:text="Variedad del cultivo" />

```

Figura 92: Continuación codificación vista registro cultivo

```

<LinearLayout
    <EditText
        android:id="@+id/txtVariedad"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:imeOptions="actionNext"
        android:singleLine="true"/>

    <Button
        android:id="@+id/btnCultivo"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:background="@color/verde_logo"
        android:text="Register"
        android:textAllCaps="false"
        android:textColor="#000000"
        android:textStyle="bold"/>

</LinearLayout>
</LinearLayout>

```

Figura 93: Continuación codificación vista registro cultivo

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    tools:context=".TratamientoActivity">

    <ImageView
        android:id="@+id/imgTratamiento"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_marginTop="40dp"
        app:srcCompat="@drawable/logo"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center" />

    <TextView
        android:id="@+id/txvtratamiento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="135dp"
        android:text="Recomiende tratamiento"
        android:textColor="@color/black"
        android:textSize="25dp"
        android:textStyle="bold" />

```

Figura 94: Codificación vista tratamiento

```

<TextView
    android:id="@+id/txvReceta"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="180dp"
    android:text="Tratamiento"
    android:textColor="@color/black" />

<EditText
    android:id="@+id/TextMultiLine"
    android:layout_width="300dp"
    android:layout_height="100dp"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="200dp"
    android:imeOptions="actionNext"
    android:singleLine="true"
    android:inputType="textMultiLine" />

<TextView
    android:id="@+id/txvAplicacion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="310dp"
    android:text="Modo de aplicación"
    android:textColor="@color/black" />

```

Figura 95: Continuación codificación vista tratamiento

```

<EditText
    android:id="@+id/TextAplicacion"
    android:layout_width="300dp"
    android:layout_height="100dp"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="335dp"
    android:imeOptions="actionNext"
    android:inputType="textMultiLine"
    android:singleLine="true" />

<TextView
    android:id="@+id/txvAgua"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="440dp"
    android:text="Cantidad de agua"
    android:textColor="@color/black" />

<EditText
    android:id="@+id/txvAgua"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="460dp"
    android:imeOptions="actionNext"
    android:singleLine="true" />

```

Figura 96: Continuación codificación vista tratamiento

```
<Button
    android:id="@+id/btnEnviar"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="560dp"
    android:background="@color/verde_logo"
    android:text="Enviar"
    android:textAllCaps="false"
    android:layout_centerHorizontal="true"
    android:textColor="#000000"
    android:textStyle="bold" />

</RelativeLayout>
```

Figura 97: Continuación codificación vista tratamiento

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".IdentificacionActivity">

    <TextView
        android:id="@+id/txvIdentificar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:layout_marginTop="20dp"
        android:text="Informe afectación"
        android:textColor="@color/black"
        android:textSize="24sp"
        android:textStyle="bold" />

    <ImageView
        android:id="@+id/imgIdentificar"
        android:layout_width="300dp"
        android:layout_height="260dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="60dp"
        app:srcCompat="@drawable/isologo" />
```

Figura 98: Codificación vista identificación de enfermedad

```

<Button
    android:id="@+id/btnSeleccion"
    android:layout_width="100dp"
    android:layout_height="35dp"
    android:layout_centerHorizontal="false"
    android:layout_gravity="center"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="335dp"
    android:background="@color/verde_Logo"
    android:onClick="obtenerImagen"
    android:text="Seleccionar"
    android:textAllCaps="false"
    android:textColor="#000000"
    android:textStyle="bold" />

<Button
    android:id="@+id/btnFoto"
    android:layout_width="100dp"
    android:layout_height="35dp"
    android:layout_marginLeft="145dp"
    android:layout_gravity="center"
    android:layout_marginTop="335dp"
    android:background="@color/verde_Logo"
    android:textAllCaps="false"
    android:textColor="#000000"
    android:textStyle="bold"
    android:text="Cámara" />

```

Figura 99: Continuación codificación vista identificación de enfermedad

```

<Button
    android:id="@+id/btnPredecir"
    android:layout_width="100dp"
    android:layout_height="35dp"
    android:layout_centerHorizontal="false"
    android:layout_gravity="center"
    android:layout_marginLeft="270dp"
    android:layout_marginTop="335dp"
    android:background="@color/verde_Logo"
    android:text="Predecir"
    android:textAllCaps="false"
    android:textColor="#000000"
    android:textStyle="bold" />

<TextView
    android:id="@+id/txvRInforme"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="370dp"
    android:gravity="center"
    android:text="Resultado informe"
    android:textColor="@color/black"
    android:textSize="18sp"
    android:textStyle="bold" />

```

Figura 100: Continuación codificación vista identificación de enfermedad

```

<TextView
    android:id="@+id/txvRecomendar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="400dp"
    android:gravity="center"
    android:text=""
    android:textColor="@color/purple_700"
    android:textSize="14sp" />

<TextView
    android:id="@+id/txvArgente"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:layout_marginTop="425dp"
    android:text="Recomendaciones"
    android:textColor="@color/black"
    android:textSize="14sp"
/>
</RelativeLayout>

```

Figura 101: Continuación codificación vista identificación de enfermedad

4.1.7.4. Pruebas funcionales

Parámetros pruebas de funcionamiento evaluado sobre 5

Excelente → 5, Muy Bueno → 4, Bueno → 3, Regular → 2, Malo → 1

Prueba 1

Tabla 28: Prueba en dispositivos de nivel de API 23 versiones de Android 6.0 – 6.0.1

Embalador	Android 6.0					Android 6.0.1				
	E	MB	B	R	M	E	MB	B	R	M
Dimensiones					x					x
Logo		x					x			
Cuadros de texto			x				x			
Botones			x				x			
Función Asignada	x					x				
Cuadros de entrada			x					x		
Tiempo respuesta			x					x		
Total	5	4	12	0	1	5	12	6	0	1
Resultado de la prueba	22 → 62,86%					24 → 68,57%				

Nota: muestra los resultados de prueba funcional en dispositivos móviles con Android 6.0 y 6.1

Prueba 2

Tabla 29: Prueba en dispositivos de nivel de API 24 - 25 versiones de Android 7.0 – 7.1.2

Evaluador	Android 7.0					Android 7.1.2				
	E	MB	B	R	M	E	MB	B	R	M
Dimensiones			x				x			
Logo		x					x			
Cuadros de texto			x					x		
Botones			x					x		
Función Asignada	x					x				
Cuadros de entrada			x					x		
Tiempo respuesta			x					x		
Total	5	4	15	0	0	5	8	12	0	0
Resultado de la prueba	24 → 68,57%					25 → 71,43%				

La tabla 22 muestra los resultados de prueba funcional en dispositivos móviles con Android 7.0 y 7.1.2

Prueba 3

Tabla 30: Prueba en dispositivos de nivel de API 26 - 27 versiones de Android 8.0 – 8.1

Evaluador	Android 8.0					Android 8.1				
	E	MB	B	R	M	E	MB	B	R	M
Dimensiones		x					x			
Logo		x					x			
Cuadros de texto		x					x			
Botones		x					x			
Función Asignada	X					x				
Cuadros de entrada		x					x			
Tiempo respuesta			x					x		
Total	5	20	3	0	0	5	20	3	0	0
Resultado de la prueba	28 → 80%					28 → 80%				

Nota: muestra los resultados de prueba funcional en dispositivos móviles con Android 8.0 y 8.1

Prueba 4

Tabla 31: Prueba en dispositivos de nivel de API 28, 29 versiones de Android 9.0, 10.0

Evaluador	Android 9.0					Android 10.0				
	E	MB	B	R	M	E	MB	B	R	M
Dimensiones		x					x			
Logo	X					x				
Cuadros de texto	X					x				
Botones		x					x			
Función Asignada	X					x				
Cuadros de entrada		x				x				
Tiempo respuesta			x					x		
Total	15	12	3	0	0	20	8	3	0	0
Resultado de la prueba	30 → 85,71%					31 → 88,57%				

Nota: muestra los resultados de prueba funcional en dispositivos móviles con Android 9.0 y 10.0

Prueba 5

Tabla 32: Prueba en dispositivos de nivel de API 30, 31 versiones de Android 11.0, 12.0

Evaluador	Android 11.0					Android 12.0				
	E	MB	B	R	M	E	MB	B	R	M
Dimensiones		x					x			
Logo	X					x				
Cuadros de texto	X					x				
Botones	X					x				
Función Asignada	X					x				
Cuadros de entrada	X					x				
Tiempo respuesta			x					x		
Total	25	4	3	0	0	25	4	3	0	0
Resultado de la prueba	32 → 91,43%					32 → 91,43%				

Nota: muestra los resultados de prueba funcional en dispositivos móviles con Android 11.0 y 12.0

4.1.7.5. Pruebas de aceptación

Tabla 33: Aceptación de los agricultores para el aplicativo

Aceptación Agricultores			
Nombres	Calificación	Parámetros de evaluación	Resultado Porcentual Total
Santiago Sangucho	7,5		
Brayan Insuasti	8		
Wilmer Bolaños	8,5		
Fabian Usuay	8,5		
Manuel Calixto	7,5	Cumple con la función	
Damian Palacios	8	Nivel de coincidencia con las enfermedades	80%
Klever de Jesús	8	Tiempo de respuesta	
Anderson Espinoza	8,5		
Frank Ogonaga	7,5		
Junior Bolaños	8		

4.1.7.6. Pruebas de comparativas

Tabla 34: Comparativa entre la app resultante de la investigación y plantNet

Nombre del aplicativo	PI@ntNet	Plant Solution
Parámetro		
Funcionalidad	95,3%	82,85%
Compatibilidad	Android e IOS	Android desde vs 6
Accesibilidad	Gratuita	Gratuita
Exactitud	98.8%	78.86%
Aceptación	84%	80%
Valores de pérdida	0, 0032	0, 0310
Tiempo de respuesta de la red neuronal	1, 19 segundos	De 10 a 15 segundos

Nota: muestra los resultados de la comparativa entre App de identificación de imágenes de cultivos enfermos.

Fuente: (Aguirre, 2019)

Tabla 35: Segunda comparativa entre la App resultante de la investigación y PlantVillage

Nombre del aplicativo	PlantVillage	Plant Solution
Parámetro		
Funcionalidad	99,3%	82,85%
Compatibilidad	Vs de Android	Android desde vs 6
Accesibilidad	Gratuita	Gratuita
Exactitud	98.9%	78.86%
Aceptación	100%	80%
Valores de pérdida	0, 0025%	0, 0310%
Tiempo de respuesta de la red neuronal	0, 059 segundos	De 10 a 15 segundos

Nota: muestra los resultados de la comparativa entre App de identificación de imágenes de cultivos enfermos.

Fuente: Maeda et al. (2018)

4.2. DISCUSIÓN

En la presente tesis se investigó el procesamiento de imágenes y la identificación de enfermedades de hoja en cultivos de pimiento de la comunidad de Pusir Grande, contando con un total de 273 individuos que se dedica a esta activa dentro del agro, por lo que se procedió hacer una división entre agricultores y técnicos agrónomos. Con base en esto se planteó la idea a defender que es en la se desarrollara la investigación.

De acuerdo a los resultados obtenidos en esta investigación se puede decir que la tecnología está tomando gran posesión en el sector agrícola en la denominada agricultura de precisión, es por eso que la identificación a base de procesamiento de imágenes y algoritmos de inteligencia artificial está inmersa en ello, proponiéndole como ayuda en la forma de detención de las enfermedades en los cultivos.

En base al objetivo planteado para nuestra investigación que lo hemos denominado elaborar una herramienta de procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar, se comenzó con la estructuración del marco teórico a base de información recopilada. La metodología establecida sería la encargada de referenciar el desarrollo de la herramienta de detección de las enfermedades en plantas de pimiento, donde se procede con la recolección de los datos, análisis de los algoritmos, técnicas y tecnologías de inteligencia artificial que utilizadas en el desarrollo del aplicativo móvil.

Al encontrarnos con una investigación que contempla los dos enfoques tanto cuantitativo como cualitativo, se puede hacer uso de las técnicas de investigación. Se aplicó la entrevista a los técnicos agrónomos que realizan visitas a los cultivos de pimiento de la comunidad Pusir Grande y se llevó a cabo una encuesta a los individuos considerados en la investigación que en este caso serían los agricultores.

Gracias a los resultados que se encontraron en la comparación de los dos grupos de técnicas de procesamiento de imágenes se pudo obtener los principales parámetros para el desarrollo de la herramienta, en la aplicación de los instrumentos de investigación la mayor parte de los individuos considerados objetos de entrevista y encuesta se mostraron contentos con la propuesta de aplicar tecnología en esta actividad agrícola.

La meta de esta investigación fue el desarrollo de la una herramienta para la detección de enfermedades en las hojas de cultivos de pimiento, misma que se la pudo consolidar con la ayuda de la metodología de proceso incremental, obteniendo prototipos en cada una de sus iteraciones mismos que presentaban mejoras en comparación al anterior.

Gracias a él análisis se pudo concluir que la solución informática entrega un aporte al sector agrícola en especial a la zona productiva de la comunidad de Pusir Grande, en la actividad de identificación de estas afectaciones, la tecnología en el agro tiene un sin número de beneficios logrando aplicar la visión por computadora más la observación con el ojo humano.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Para la identificación de enfermedades en los cultivos se encontraron mejores beneficios a través de las técnicas aplicadas por medio de tecnología de inteligencia artificial que en las técnicas usadas para procesamiento de imágenes e identificación de forma tradicional.
- En las etapas de procesamiento de imágenes de forma tradicional, se identificó que era necesario agrupar las técnicas de dominio espacial, dominio de frecuencia y extracción de características, con el objetivo de poder hacer uso de las operaciones matemáticas de este grupo que mejor se apegue a la investigación.
- En el análisis de procesamiento tradicional y a base de inteligencia artificial, nos encontramos con que las técnicas de inteligencia artificial demanda más costo y complejidad computacional.
- El método de prototipado de prueba que ofrece la Metodología de proceso incremental fue de mucha ayuda en el desarrollo del aplicativo, con este método se pudo encontrar problemas de ejecución en el código objetando mejoras en los prototipos funcionales posteriores.
- El funcionamiento del aplicativo móvil, muestra un crecimiento porcentual escalonado en cada una de las pruebas en sus diferentes versiones de Android.
- El funcionamiento del aplicativo móvil, muestra un crecimiento porcentual escalonado en cada una de las pruebas en sus diferentes versiones de Android.

5.2. RECOMENDACIONES

- El proyecto se orientó al desarrollo de un aplicativo para el sector agropecuario por ello se recomienda a la carrera de Ingeniería en Computación de la Universidad Politécnica Estatal de Carchi (UPEC), orientar a sus estudiantes en las aulas a que siguen realizando proyectos de investigación que vayan apegados a sector productivo agrícola.
- Realizar investigaciones posteriores en relación a otras áreas de acción del sector productivo o la búsqueda de soluciones para otros tipos de enfermedades que afectan a los cultivos de los agricultores.
- En la metodología de desarrollo de software de proceso incremental es recomendable prestar atención a los problemas de los prototipos para solventar mejorar en los posteriores.
- La App móvil puede funcionar de buena manera en un alojamiento gratuito en la nube, no obstante, es recomendable que se la actualice a funcionar en un servidor de pago o una virtualización como las que ofrecen Amazon y Microsoft Azure.
- Si bien el aplicativo es compatible a partir de la versión 6.0 de Android se recomienda hacer uso del mismo a partir de la versión 7.0 y posteriores a ésta para evitar posibles complicaciones.

IV. REFERENCIAS BIBLIOGRÁFICAS

- AENVERDE. (25 de Abril de 2019). *Aenverde*. Obtenido de El mercado del pimiento en el mundo: <https://www.aenverde.es/el-mercado-del-pimiento-en-el-mundo/>
- Aguirre, N. (11 de Diciembre de 2019). *Procesamiento de Imágenes*. Recuperado el 5 de octubre de 2021, de [bibing.us.es: http://bibing.us.es/proyectos/abreproy/12112/fichero/Documento_por_capitulos%252F3_Cap%C3%ADtulo_3.pdf](http://bibing.us.es/proyectos/abreproy/12112/fichero/Documento_por_capitulos%252F3_Cap%C3%ADtulo_3.pdf)
- Baume, L. (23 de febrero de 2021). *Breve introducción a Google Colab*. Recuperado el 25 de agosto de 2021, de <http://fcaglp.unlp.edu.ar/~gbaume/grupo/Publicaciones/Apuntes/GoogleColab.pdf>
- Campos, M. (2017). *Métodos de investigación Academia*. (Tesis de Ingeniería). Universidad de Costa Rica, San José, Costa Rica. Recuperado de http://www.icomoscr.org/m/investigacion/%5BMETODOS%5DFolleto_v.1.1.PDF
- Casimba, R. (2017). *IDENTIFICACIÓN DE PLAGAS Y ENFERMEDADES (HONGOS Y BACTERIAS) DEL CULTIVO DE JÍCAMA (Smallanthus sonchifolius) EN EL CANTÓN COTACACHI, OTAVALO E IBARRA – IMBABURA*. (Tesis de Ingeniería). Universidad Técnica del Norte. Ibarra, Ecuador. Recuperando de: <http://repositorio.utn.edu.ec/bitstream/123456789/6510/1/03%20AGP%20215%20TRABAJO%20DE%20GRADO.pdf>
- Carbajal, Y. (2019). *Paradigma, revolución científica y métodos deductivo e inductivo* [Diapositiva de PowerPoint]. Repositorio Material Faculta de economía. Recuperado de http://148.215.1.182/bitstream/handle/20.500.11799/108420/secme22923_1.PDF?sequence=1
- Cárdenas, S & Mejía, M. (2 al 4 de abril de 2017). *Procesamiento de Imágenes Digitales con Variantes de Redes Neuronales Artificiales de Tercera Generación*. XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico. Recuperado el 27 de septiembre de 2021 de: <http://b-dig.iie.org.mxBibDig2P14-0075p359.pdf>
- Cabrera, A., y Ramírez, L. (2017). *PROPUESTA DE UN PLAN DE MEJORAS BASADO EN GESTIÓN POR PROCESOS, PARA INCREMENTAR LA PRODUCTIVIDAD EN LA EMPRESA DISTRIBUCIÓN A & B*. (Tesis de pregrado). Universidad señor de Sipán, Chiclayo, Perú. Recuperado de: <http://repositorio.uss.edu.pe/bitstream/handle/uss/4068/TESIS%20FINAL%2002-08-2017.PDF;jsessionid=3682F2EDA1D703012E0EE4F7D0E9EF06?sequence=1>

- Correa, D. (2018). *La función de la imagen digital en la formación del ideal del yo*. (Tesis de Ingeniería), Pontificia Universidad Católica Del Ecuador, Quito, Ecuador. Recuperado de: <http://repositorio.puce.edu.ec/handle/22000/16368>
- Cuenca, A., y Ante, E. (2019). *PROPUESTA PARA EL MEJORAMIENTO DEL PROCESO PRODUCTIVO EN LA PIZZERIA DI ANGELO APLICANDO LA INGENIERIA DE PROCESOS*. (Tesis de pregrado). Universidad Técnica de Cotopaxi, Cotopaxi, Ecuador. Recuperado de: <http://repositorio.utc.edu.ec/bitstream/27000/5466/1/PI-001393.PDF>
- Cusme, K & Loor, A. (2019). *APLICACIÓN MÓVIL DE DETECCIÓN Y CLASIFICACIÓN DE “LA ROYA” EN HOJAS DE CAFÉ ROBUSTA MEDIANTE APRENDIZAJE AUTOMÁTICO*. Tesis de ingeniería publicada. Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López. Calceta, Ecuador. Recuperado 12 de agosto del 2021 de: <http://repositorio.espam.edu.ec/bitstream/42000/1104/1/TTC21.pdf>
- Cuzco, A. (2020). *SISTEMA DE GESTIÓN Y AUTOMATIZACIÓN DEL ACCESO A UN PARQUEADERO*. Tesis de Ingeniería publicada, Universidad Técnica del Norte, Ibarra, Ecuador. Recuperado 6 de julio del 2021 de: <http://repositorio.utn.edu.ec/bitstream/123456789/11199/2/04%20MEC%20351%20TRABAJO%20GRADO.pdf>
- Chavero, A. (19 de Abril de 2021). *Algoritmos y tipos de machine learning, el futuro es ahora*. Recuperado el 6 de Septiembre de 2021, de Crehana: <https://www.crehana.com/ec/blog/desarrollo-web/tipos-machine-learning/>
- Da Silva, D. (31 de marzo de 2021). *Diferencia entre Machine Learning y Deep Learning*. Recuperado el 9 de septiembre de 2021, de Zendesk: <https://www.zendesk.com.mx/blog/machine-learning-deep-learning-diferencias/>
- Franco, M. (2019). *Mancha Gris de la Hoja del Tomate: Identificación, Biología y Genómica del Agente Etiológico*. (Tesis de doctorado). Universidad Nacional de La Plata. Buenos Aires, Argentina. Recuperado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/73593/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Folgueiras, P. (2016). *La entrevista. Técnica de recogida de información*. Recuperado de <http://diposit.ub.edu/dspace/bitstream/2445/99003/1/entrevista%20pf.PDF>
- Gamboa, M. (2017). *ESTADISTICA APLICADA A LA INVESTIGACIÓN CIENTÍFICA*.

- (Tesis de Ingeniería), Universidad de las Tunas, Cuba. Recuperado de:
<http://roa.ult.edu.cu/bitstream/123456789/3667/1/ESTADISTICA%20APLICADA%200A%20LA%20INVESTIGACION%20CIENTIFICA%20Pag.%2059-76.PDF>
- Gómez, M., Cervantes, J. y González, P. (2019). *Fundamentos de Ingeniería de Software*. Naucalpan, México: Universidad Autónoma Metropolitana. Recuperado de
http://www.cua.uam.mx/pdfs/conoce/libroselec/Fundamentos_Ing_SW-VF.pdf
- Godoy, Y & Ducuara, A (2019). ANÁLISIS COMPARATIVO DE LAS TÉCNICAS SURF Y ORB PARA LA DETECCIÓN DE PUNTOS DE INTERÉS EN FOTOGRAFÍAS AÉREAS. (Tesis de Ingeniería), Universidad de Ibagué, Ibagué, Colombia. Recuperado de:
<https://repositorio.unibague.edu.co/bitstream/20.500.12313/841/1/Trabajo%20de%20grado.%20pdf>
- Hortoinfo. (10 de Enero de 2018). *La producción mundial de pimiento*. Obtenido de
<http://www.hortoinfo.es/index.php/6601-prod-mund-pim-100118>
- Jiva, E (2019). DESARROLLO DE LA TELEOPERACIÓN DE ROBOTS INDUSTRIALES Y COLABORATIVOS MEDIANTE TÉCNICAS AVANZADAS DE VISIÓN ARTIFICIAL. (Tesis de Maestría). Universidad Politécnica de Valencia. Valencia, España. Recuperando de:
<https://riunet.upv.es/bitstream/handle/10251/130010/Jiva%20-%20Desarrollo%20de%20la%20teleoperación%20de%20robots%20industriales%20y%20colaborativos%20mediante%20técnicas%20av...pdf?sequence=1&isAllowed=y>
- Baume, L. (23 de febrero de 2021). *Breve introducción a Google Colab*. Recuperado el 25 de agosto de 2021, de
<http://fcaglp.unlp.edu.ar/~gbaume/grupo/Publicaciones/Apuntes/GoogleColab.pdf>
- Chavero, A. (19 de Abril de 2021). *Algoritmos y tipos de machine learning, el futuro es ahora*. Recuperado el 6 de Septiembre de 2021, de Crehana:
<https://www.crehana.com/ec/blog/desarrollo-web/tipos-machine-learning/>
- Da Silva, D. (31 de marzo de 2021). *Diferencia entre Machine Learning y Deep Learning*. Recuperado el 9 de septiembre de 2021, de Zendesk:
<https://www.zendesk.com.mx/blog/machine-learning-deep-learning-diferencias/>
- ISO/IEC 15504 SPICE. (2018). *EL ESTÁNDAR INTERNACIONAL ISO/IEC 15504*. Recuperado el 25 de Agosto de 2021, de Normas ISO: <https://www.normas-iso.com/iso-iec-15504-spice/>

- Lima, A. (9 de febrero de 2021). *es.acervolima.com*. Obtenido de <https://es.acervolima.com/2021/02/09/coincidencia-de-caracteristicas-usando-el-algoritmo-orb-en-python-opencv/>
- Negrete, A. (21 de Mayo de 2021). *Redes_Neuronales*. Obtenido de ofelia y orquesta: http://ofeliayorquesta.com/articulos/Redes_Neuronales_001.pdf
- Obregón, V. (15/04/2016 de Abril de 2016). *Argentina.gob.ar*. Obtenido de Guía para la Identificación de las enfermedades de pimiento en invernadero: https://inta.gob.ar/sites/default/files/inta-guia_identificacion_de_las_enfermedades_de_pimiento_en_invernadero.pdf
- Pérez, A. (16 de Agosto de 2016). *Características y fases del modelo incremental*. Recuperado el 25 de Agosto de 2021, de OBS Business School: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>
- Reyes, O., Mejia, M., & Useche, J. (19 de Diciembre de 2018). *TÉCNICAS DE INTELIGENCIA ARTIFICIAL UTILIZADAS EN EL PROCESAMIENTO DE IMÁGENES Y SU APLICACIÓN EN EL ANÁLISIS DE PAVIMENTOS*. Obtenido de redalyc.org: <https://www.redalyc.org/journal/1492/149258931014/html/>
- Soto, S. (2018). *Principales enfermedades que afectan al pimiento en Chile*. Recuperado el 4 de agosto de 2021, de Boletín INIA - Instituto de Investigaciones Agropecuarias: <https://biblioteca.inia.cl/handle/123456789/6653>
- Turnero, P. (20 de abril de 2017). *Monografias.com*. Obtenido de <https://www.monografias.com/trabajos107/procesamiento-imagenes-medicas-algoritmos-imagenes/procesamiento-imagenes-medicas-algoritmos-imagenes2.shtml>
- Zambrano, R. (13 de Octubre de 2019). *Diferencias entre Machine Learning y Deep Learning*. Recuperado el 6 de septiembre de 2021, de OpenWebinars: <https://openwebinars.net/blog/diferencias-entre-machine-learning-y-deep-learning/>

V. ANEXOS

Anexo 1: Certificado o Acta del Perfil de Investigación



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES
CARRERA DE INGENIERÍA EN INFORMÁTICA



ACTA

DE LA SUSTENTACIÓN DE PREDEFENSA DEL DEL TRABAJO DE INTEGRACIÓN

NOMBRE: Palacios Julio Leandro
Leodan
NIVEL/PARALELO: 0

CÉDULA DE IDENTID 1004234470
PERIODO ACADÉMICO PAO2021B

TEMA DEL TIC: "Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020."

Tribunal designado por la dirección de esta Carrera, conformado por:

PRESIDENTE: MSC. SAMUEL BENJAMÍN LASCANO RIVERA
DOCENTE TUTOR: MSC. JEFFERY ALEX NARANJO CEDEÑO
DOCENTE: MSC. JORGE HUMBERTO MIRANDA REALPE

De acuerdo al artículo 32: Una vez entregados los documentos; y, cumplidos los requisitos para la realización de la pre-defensa el Director/a de Carrera designará el Tribunal, fijando lugar, fecha y hora para la realización de este acto:

EDIFICIO DE AULAS 0 **AULA:** 0

FECHA: miércoles, 1 de diciembre de 2021

HORA: 15H00

Obteniendo las siguientes notas:

1) Sustentación de la predefensa: 6,30

2) Trabajo escrito 2,90

Nota final de PRE DEFENSA 9,20

Por lo tanto: **APRUEBA CON OBSERVACIONES** ; debiendo acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el miércoles, 1 de diciembre de 2021



Firmado digitalmente por:
SAMUEL BENJAMIN
LASCANO RIVERA

MSC. SAMUEL BENJAMÍN LASCANO RIVERA
PRESIDENTE



Firmado digitalmente por:
JEFFERY ALEX
NARANJO CEDEÑO

MSC. JEFFERY ALEX NARANJO CEDEÑO
DOCENTE TUTOR



Firmado digitalmente por:
JORGE HUMBERTO
MIRANDA REALPE

MSC. JORGE HUMBERTO MIRANDA REALPE
DOCENTE

Anexo 2: Certificado del abstract por parte de idiomas



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL
CARCHI FOREIGN AND NATIVE LANGUAGE
CENTER**

Informe sobre el Abstract de Artículo Científico o Investigación.

Autor: Leandro Leodan Palacios Julio

Fecha de recepción del abstract: 14 de diciembre de 2021

Fecha de entrega del informe: 14 de diciembre de 2021

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según los rubrics de evaluación de la traducción en Inglés, ésta alcanza un valor de 9 por lo cual se validó dicho trabajo.

Atentamente



EDISON BOANERGES
PEÑAFIEL ARCOS

Ing. Edison Peñafiel Arcos MSc
Coordinador del CIDEN

Anexo 3: Informe Turnitin



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Leandro Palacios
Título del ejercicio: Quick Submit
Título de la entrega: Procesamiento de imágenes digitales para la identificación ...
Nombre del archivo: INVESTIGACI_N_PALACIOS_LEANDRO_CORREGIDO_APROVA...
Tamaño del archivo: 6.39M
Total páginas: 140
Total de palabras: 22,058
Total de caracteres: 134,525
Fecha de entrega: 22-nov.-2021 11:32a. m. (UTC-0500)
Identificador de la entre... 1710303965



**JEFFERY
ALEX
NARANJO
CEDEÑO**
Firmado digitalmente por
JEFFERY ALEX
NARANJO CEDEÑO
Fecha: 2021.11.22
14:02:45 -05'00'

Anexo 4: Ficha técnica Identificación de enfermedad

Ficha técnica de identificación enfermedad						
Item 1 / Técnico 1						
Tra tamiento	Enfermedad	Clasificación	Nivel de incidencia	Dato/investigación		
Aplicación foliar	Mariposa	Enfermedad de hoja	80%	Si		
Aplicación foliar	Araña	Enfermedad de tallo y hojas	85%	No		
Aplicación a tierra	Equisetos	Enfermedad de raíz	80%	No		
Aplicación foliar	Oidiospsis	Enfermedad de hoja	95%	Si		
Aplicación foliar	Mancha bacteriana Gris	Enfermedad de hoja	90%	Si		
Aplicación foliar	Viruela	Enfermedad de hoja	90%	Si		
Aplicación a tierra	Tristeza del pimiento	Enfermedad de raíz	100%	No		
Aplicación a tierra	Marchitez Phytophthora capsici	Enfermedad de raíz	100%	No		
Aplicación a tierra	Daños por Nutrición	Enfermedad de hoja	100%	No		
Aplicación foliar	Escaldadur	Enfermedad de tallo	100%	No		
Item 2 / Técnico 2						
Aplicación a tierra	Falta de nutrientes	Enfermedad de hoja	65%	No		
Aplicación foliar	Daños climaticos	Enfermedad de tallo	75%	No		
Aplicación foliar	Manchón Griciano	Enfermedad de hoja	100%	Si		
Aplicación foliar	Mosca Blanca	Enfermedad de hoja	100%	Si		
Aplicación foliar	Araña roja	Enfermedad de tallo	65%	No		
Aplicación foliar	Polvillo / Oidiospsis	Enfermedad de hoja	100%	Si		
Aplicación a tierra	Marchitez Phytophthora capsici	Enfermedad de raíz	100%	No		
Aplicación a tierra	Gusano Tierrero	Enfermedad de raíz	80%	No		
Item 3 / Agricultor 1						
Aplicación a tierra	Yata	Enfermedad de raíz	80%	No		
Aplicación foliar	Arcaros	Enfermedad de raíz	80%	No		
Aplicación foliar	Trix	Enfermedad de hoja	90%	Si		
Aplicación foliar	Ceniza/ Oidiospsis	Enfermedad de hoja	100%	Si		
Aplicación foliar	Palomilla	Enfermedad de hoja	100%	Si		
Aplicación foliar	Mancha Gris	Enfermedad de hoja	100%	Si		
Aplicación a tierra	Marchitez Phytophthora capsici	Enfermedad de hoja	100%	No		
Item 4 / Agricultor 2						
Aplicación foliar	Mancha Gris	Enfermedad de hoja	90%	Si		
Aplicación foliar	Trix	Enfermedad de hoja	90%	Si		
Aplicación foliar	Ceniza/ Oidiospsis	Enfermedad de hoja	100%	Si		
Aplicación a tierra	Yata	Enfermedad de raíz	90%	No		
Aplicación a tierra	Marchitez Phytophthora capsici	Enfermedad de raíz	100%	No		
Resultado						
Enfermedad más mencionada	% Técnico 1	% Técnico 2	% Agricultor 1	% Agricultor 2	%Total promedio	Dato/investigación
Mancha bacteriana Gris	90%	100%	100%	90%	95%	Si
Ceniza/ Oidiospsis	95%	100%	100%	100%	98.75%	Si
Marchitez Phytophthora capsici	100%	100%	100%	100%	100%	No


Anexo 5: Código Python preprocesado de las imágenes identificación tradicional

```
TradicionalPocesamiento.py 2 ●
TradicionalPocesamiento.py > ...
1 import cv2
2 import numpy as np
3 #Adquisición de la imagen
4 img = cv2.imread("gris18.JPG",cv2.IMREAD_COLOR)
5 #Realce y mejora de la imagen
6 norm_img1 = cv2.normalize(img, None, alpha=0, beta=1, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)
7 norm_img2 = cv2.normalize(img, None, alpha=0, beta=1.2, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)
8 norm_img1 = (255*norm_img1).astype(np.uint8)
9 norm_img2 = np.clip(norm_img2, 0, 1)
10 norm_img2 = (255*norm_img2).astype(np.uint8)
11 cv2.imwrite("normalize1.jpg",norm_img1)
12 cv2.imwrite("normalize2.jpg",norm_img2)
13 # segmentación de las imagenes
14 image = cv2.imread('/content/normalize2.jpg', cv2.COLOR_BGR2GRAY)
15 processed_image = cv2.medianBlur(image, 3)
16 _, dst1 = cv2.threshold(processed_image, 96, 255, cv2.THRESH_BINARY)
17 #Obtención de características
18 orb = cv2.ORB_create(nfeatures=5000)
19 keypoints, descriptors= orb.detectAndCompute(dst1, None)
20 dst1 = cv2.drawKeypoints(dst1, keypoints, None)
21 #Impresión de las imagenes del proceso
22 print("Original")
23 cv2.imshow(img)
24 print("Normalización 1")
25 cv2.imshow(norm_img1)
26 print("Normalización 2")
27 cv2.imshow(image)
28 print("Filtro")
29 cv2.imshow(processed_image)
30 print("Segmentación")
31 cv2.imshow(dst1)
32 print("Obtención de características")
33 cv2.imshow(dst1)
34 cv2.waitKey(0)
35 cv2.destroyAllWindows()
```

Anexo 6: Cuestionario encuesta agricultores

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
CARRERA DE INGENIERÍA EN INFORMÁTICA

Cuestionario encuesta Agricultores



Tema: "Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020"

1. ¿Es necesaria la investigación y análisis de especificaciones técnicas y características comunes y no comunes de las enfermedades foliares?
Sí 154 No 0
2. ¿Actualmente cuenta con un sistema de detección temprana de enfermedades en hojas por medio de procesamiento de imágenes?
Sí 14 No 140
3. ¿La elaboración y seguimiento de un informe actualizado del estado de la planta reduciría el riesgo de desastres en el cultivo?
Mucho 41 Poco 106 Nada 7
4. ¿El cumplimiento de las indicaciones entregadas por los técnicos agrónomos permitiría frenar la propagación de la enfermedad en el cultivo?
Mucho 29 Poco 117 Nada 8
5. ¿La implementación de medios tecnológicos en la identificación de las enfermedades de hoja aseguraría el desarrollo saludable del cultivo?
Sí 122 No 38
6. ¿El diseño y desarrollo de un aplicativo de detección temprana de enfermedades en las hojas permitiría reducir el impacto negativo en la producción?
Mucho 45 Poco 109 Nada 0
7. ¿Considera usted que la forma actual de identificación de las enfermedades en las hojas es total mente apropiada

Anexo 7: Continuación cuestionario encuesta agricultores

Sí 85 No 69

8. ¿Cree usted que es importante la creación de un aplicativo de detección de enfermedades en hojas mediante el uso de imágenes?

Sí 133 No 21

9. ¿Usted cómo agricultor dispone del equipo adecuado para hacer uso del aplicativo de detección temprana de enfermedades en hojas por medio de procesamiento de imágenes?

Sí 119 No 35

10. ¿De existir el aplicativo de detección temprana de enfermedades por medio del procesamiento de imágenes, apoyaría su uso?

Sí 152 No 2

Anexo 8: Hoja de respuestas cuestionario encuesta agricultores

Hoja de respuestas		Preguntas																				
N°	Nombre	1		2		3		4		5		6		7		8		9		10		
		S	N	S	N	M	P	N	M	P	N	S	N	M	P	N	S	N	S	N	S	N
1	Wilmar Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
2	Guillermo Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
3	Manuel Calixto	✓				✓				✓		✓		✓		✓		✓		✓		✓
4	Santiago Sangacho	✓		✓						✓		✓		✓		✓		✓		✓		✓
5	Luis Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
6	Dario Salazar	✓		✓						✓		✓		✓		✓		✓		✓		✓
7	Orlando Maneses	✓		✓						✓		✓		✓		✓		✓		✓		✓
8	Andres Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
9	Jhon Gargala	✓		✓						✓		✓		✓		✓		✓		✓		✓
10	Mano De Jesus	✓		✓						✓		✓		✓		✓		✓		✓		✓
11	Andres Saramillo	✓		✓						✓		✓		✓		✓		✓		✓		✓
12	Orlando Chiriboga	✓		✓						✓		✓		✓		✓		✓		✓		✓
13	Andres De Jesus	✓		✓						✓		✓		✓		✓		✓		✓		✓
14	Jairo Sangacho	✓		✓						✓		✓		✓		✓		✓		✓		✓
15	Luis Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
16	Juan Carlos De Jesus	✓		✓						✓		✓		✓		✓		✓		✓		✓
17	Mano De Jesus	✓		✓						✓		✓		✓		✓		✓		✓		✓
18	Alfredo Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
19	Ronald Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
20	Cristian Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
21	Cesar Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
22	Piryan Insuasti	✓		✓						✓		✓		✓		✓		✓		✓		✓
23	Jhon Insuasti	✓		✓						✓		✓		✓		✓		✓		✓		✓
24	Roberto Insuasti	✓		✓						✓		✓		✓		✓		✓		✓		✓
25	Andres Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
26	Sergio Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
27	Thiago Espinoza	✓		✓						✓		✓		✓		✓		✓		✓		✓
28	José Antonio Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
29	Michael Salazar	✓		✓						✓		✓		✓		✓		✓		✓		✓
30	Jhon Gargala	✓		✓						✓		✓		✓		✓		✓		✓		✓
31	José Luis Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
32	Arzon Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
33	Mano De Jesus	✓		✓						✓		✓		✓		✓		✓		✓		✓
34	Stalin Dora	✓		✓						✓		✓		✓		✓		✓		✓		✓
35	Maganda Carabali	✓		✓						✓		✓		✓		✓		✓		✓		✓
36	Rafael Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
37	Eduar Sanchez	✓		✓						✓		✓		✓		✓		✓		✓		✓
38	Thiago Hernandez	✓		✓						✓		✓		✓		✓		✓		✓		✓
39	Walter Garcia	✓		✓						✓		✓		✓		✓		✓		✓		✓
40	Alexander Colorado	✓		✓						✓		✓		✓		✓		✓		✓		✓
41	Damián Salazar	✓		✓						✓		✓		✓		✓		✓		✓		✓
42	Mano Salazar	✓		✓						✓		✓		✓		✓		✓		✓		✓
43	Fernando Lara	✓		✓						✓		✓		✓		✓		✓		✓		✓
44	Walter Calixto	✓		✓						✓		✓		✓		✓		✓		✓		✓
45	Fernando Ramos	✓		✓						✓		✓		✓		✓		✓		✓		✓
46	Fabriceo Boin	✓		✓						✓		✓		✓		✓		✓		✓		✓
47	Braulio Ramos	✓		✓						✓		✓		✓		✓		✓		✓		✓


Anexo 11: Continuación hoja de respuestas cuestionario encuesta agricultores

Tabla 11. Comparación del nivel de conocimiento de identificación de enfermedades foliares (tomando como cultivo de ejemplo el plátano), se fundamenta en el uso de la inteligencia artificial y busca ayudar a que la tarea de detección de enfermedades más pronta y base de una aplicación, entorno que aplica técnicas del procesamiento de imágenes para mejorar la calidad de las fotografías, aplicar filtros y obtener caracteres que ayudan a reconocer el padecimiento.

Número de evaluación	Procesamiento Tradicional	Procesamiento con inteligencia artificial												
		B	I	M	P	E	B	I	M	P	E	B		
148	Danielo Salgado	✓	/	/	/	/	/	/	/	/	/	/	/	/
149	Geovanny Pazillo	/	/	/	/	/	/	/	/	/	/	/	/	/
150	Garman Salgado	/	/	/	/	/	/	/	/	/	/	/	/	/
151	Henry Sanchez	/	/	/	/	/	/	/	/	/	/	/	/	/
152	Ricardo Ojomega	/	/	/	/	/	/	/	/	/	/	/	/	/
153	Carlos Salazar	/	/	/	/	/	/	/	/	/	/	/	/	/
154	Dani Sanchez	/	/	/	/	/	/	/	/	/	/	/	/	/
155														
156														
157														
158														
159														
160														

Anexo 12: Cuestionario entrevista a técnicos

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
CARRERA DE INGENIERÍA EN INFORMÁTICA



Cuestionario encuesta Agricultores

Tema: "Procesamiento de imágenes digitales para la identificación de enfermedades de hoja en los cultivos de pimiento de la comunidad Pusir Grande del cantón Bolívar en el año 2020"

Cuestionario entrevista para ingenieros agrónomos que visitan los sembríos

1. ¿Se tienen procedimientos y políticas actualizados para la detección, control y protección de las enfermedades de los cultivos de pimiento, de haberlos cuales son estos?
2. ¿En dónde son almacenados los resultados obtenidos al realizar el análisis para encontrar enfermedades en el cultivo?
3. ¿En caso de que se encuentre anomalías en las hojas de los cultivos cuales serían los correctivos que se recomiendan aplicar?
4. ¿Cada que tiempo se realiza en análisis de enfermedades y aplicación de correctivos en la planta de pimiento?
5. ¿Existen protocolos de prevención que el agricultor debe realizarlos previo al plantado del cultivo, si los aviese especifíquelos?

Anexo 13: Ficha técnica con síntomas y sus características

Fecha técnica aplicada para obtención de imágenes dataset de entrenamiento			
Enfermedad 1:		Enfermedad 2:	
Mancha bacteriana Gris (Stemphylium solani)		Oidiopsis (Oidiopsis taurica - Leveillula taurica)	
Síntoma	Característica	Síntoma	Característica
Lesiones	Se unen ocupando un porcentaje de la hoja más expandido Afecta en primera estancia a las hojas	Lesiones	La afectación se encuentra en las hojas anverso y reverso Presta lesiones en el fruto.
Machas	De tamaño pequeño. De forma circular a oblonga Tienen una repartición alazar Su color esta entre pardo a negro Se las encuentra en las hojas de la parte	Manchas	De tamaño pequeño distribuidas por la hoja en el anverso. De forma redonda Su color esta entre amarillo y verde claro Se las encuentra en las hojas de la parte inferior de
Manchas en madurez media	Contusiones necróticas de núcleo gris Orilla marrón oscuro y halo clorótico	Polvillo	De color blanquecino en el reverso de la hoja
Sequedad completa de la hoja Desprendimiento de la hoja	Desarrollo completo de la enfermedad	Sequedad completa de la hoja Desprendimiento de la hoja	Desarrollo completo de la enfermedad
Resultados Obtenidos		Resultados Obtenidos	
Se recaudo 400 hojas, 230 cumplen con las características		Se recaudo 400 hojas, 270 cumplen con las características	

Anexo 14: Matriz de evaluación del software

Matriz de Evaluación de a software			
Datos informativos del Software			
Nombre:	Plant Solution	Función :	Identificación enfermedades de hoja en pimiento
Tipo:	App Móvil	Versión:	PlantSolution V 1.0
Compatibilidad y funcionalidad en dispositivos Android			
Versión de Android	Resultado promedio	Parametros de evaluación	Resultado Porcentual Total
API 23 versión de Android 6.0 – 6.0.1	65,71%	Dimensiones	78.86%
API 24 - 25 versión de Android 7.0 – 7.1.2	70%	Logo	
API 26 - 27 versión de Android 8.0 – 8.1	80%	Cuadros de texto	
API 28, 29 versión de Android 9.0, 10.0	87,14%	Botones	
API 30, 31 versión de Android 11.0, 12.0	91,43%	Función Asignada	
		Cuadros de entrada	
		Tiempo respuesta	
Aceptación Agricultores			
Nombres	Calificación	Parametros de evaluación	Resultado Porcentual Total
Santiago Sangucho	7,5	Cumple con la función	80%
Brayan Insuasti	8	Nivel de coincidencia	
Wilmer Bolaños	8,5	Tiempo de respuesta	
Fabian Usuay	8,5		
Manuel Calixto	7,5		
Damian Palacios	8		
Klever de Jesús	8		
Anderson Espinoza	8,5		
Frank Ogonaga	7,5		
Junior Bolaños	8		
Valor de la calidad del Software		Calificación	
Resultado Agricultor	80%		
Resultado Funcionalidad	78.86%		
Resultado de la evaluación	79.43% -----> 8		

Anexo 15: Aplicación de encuesta agricultor fotografía 1



Anexo 16: Aplicación de encuesta agricultor fotografía 2



Anexo 17: Aplicación de encuesta agricultor fotografía 3



Anexo 18: Aplicación de encuesta agricultor fotografía 4

