

**UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**



**Aplicación móvil basado en reconocimiento de imágenes para apoyar la  
detección de plagas y enfermedades estacionales del cultivo de arándano en el  
distrito de Olmos**

**TESIS PARA OPTAR TÍTULO DE  
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

**AUTOR**

**Maria Esther Suyon Farroñan**

**ASESOR**

**Ricardo David Iman Espinoza**

<https://orcid.org/0000-0003-0409-8773>

**Chiclayo, 2025**

**Aplicación móvil basado en reconocimiento de imágenes para apoyar la  
detección de plagas y enfermedades estacionales del cultivo de arándano en el  
distrito de Olmos**

PRESENTADA POR  
**Maria Esther Suyon Farroñan**

A la Facultad de Ingeniería de la  
Universidad Católica Santo Toribio de Mogrovejo  
para optar el título de

**INGENIERO DE SISTEMAS Y COMPUTACIÓN**

APROBADA POR

**Hector Miguel Zelada Valdivieso**  
PRESIDENTE

**Miguel Ángel Diaz Espino**  
SECRETARIO

**Ricardo David Iman Espinoza**  
VOCAL

## **Dedicatoria**

A Dios principalmente por ser mi fortaleza en cada momento.

A mis padres, hermanos, mi pareja e hija quienes me motivaron a seguir y poder concluir con mi trabajo.

A mi suegra Elena por poner ese granito de arena y su apoyo incondicional, por creer en mi para culminar con éxito mi carrera profesional.

## **Agradecimientos**

Agradezco a mi asesor Ing. Ricardo Imán Espinoza por la paciencia durante todo el proceso de este proyecto.

A la empresa del distrito de Olmos por brindarme la información necesaria para culminar con este proyecto.

Finalmente, a mis amigos por el apoyo mutuo durante todo el camino de la carrera profesional.

## Aplicación móvil basado en reconocimiento de imágenes para apoyar la detección de plagas y enfermedades estacionales del cultivo de arándano en el distrito de Olmos

### INFORME DE ORIGINALIDAD

18%

INDICE DE SIMILITUD

17%

FUENTES DE INTERNET

6%

PUBLICACIONES

8%

TRABAJOS DEL ESTUDIANTE

### FUENTES PRIMARIAS

1

[hdl.handle.net](http://hdl.handle.net)

Fuente de Internet

5%

2

[tesis.usat.edu.pe](http://tesis.usat.edu.pe)

Fuente de Internet

1%

3

Submitted to Universidad Privada del Norte

Trabajo del estudiante

1%

4

[repositorio.utc.edu.ec](http://repositorio.utc.edu.ec)

Fuente de Internet

1%

5

Submitted to Universidad Tecnológica del Peru

Trabajo del estudiante

<1%

6

[www.aulavirtualusmp.pe](http://www.aulavirtualusmp.pe)

Fuente de Internet

<1%

7

[credicorp.gcs-web.com](http://credicorp.gcs-web.com)

Fuente de Internet

<1%

## Índice

RESUMEN .....	6
ABSTRACT .....	7
INTRODUCCIÓN .....	8
REVISIÓN DE LITERATURA .....	10
MATERIALES Y MÉTODOS .....	16
RESULTADOS EN BASE A LA METODOLOGÍA UTILIZADA .....	20
CONCLUSIONES .....	66
RECOMENDACIONES .....	67
REFERENCIAS.....	68
ANEXOS.....	72

## Resumen

El cultivo de arándano es uno de los más relevantes en la región Lambayeque, pero como todo cultivo también es susceptible a plagas y enfermedades afectando al producto final, muchas veces existe una demora en la detección de los patógenos y una deficiencia en el control de su producto generando pérdidas hasta de un 40%. En este contexto, se desarrolló un aplicativo móvil (APP) basado en el reconocimiento de imágenes para la detección de plagas y enfermedades estacionales del cultivo de arándano. La app ofreció una solución tecnológica e innovadora, donde emite el diagnóstico de los patógenos detectados y brinda recomendaciones de agroquímicos a aplicar en su cultivo. Para ello, se utilizó Metodologías existentes como: Machine Learning y Extreme Programming, se cumplió con los objetivos específicos como crear un repositorio de 5.325 imágenes, así mismo desarrollar un modelo de CNN en lenguaje Python para entrenamiento y validación, para la integración del modelo y la App se usó los frameworks Angular y Ionic. El resultado del modelo entrenado arrojó su precisión del 95.3%, detectando y clasificando las plagas y enfermedades, se realizó pruebas usando la matriz de confusión con el 15% de los datos obteniendo la precisión para alternaria de 95.2%, hoja sana 100%, y mosca blanca 91.1%, se interpreta que el modelo tiene una alta precisión en todas las clases entrenadas. Finalmente se concluye que el aplicativo brinda predicciones acertadas, pero se recomienda seguir validando y ajustando según sea necesario para predecir las enfermedades y plagas del cultivo de arándano.

**Palabras clave:** Redes neuronales convolucionales, arándanos, procesamiento de imágenes.

### **Abstract**

The blueberry crop is one of the most important in the Lambayeque region, but as any crop is also susceptible to pests and diseases affecting the final product, there is often a delay in the detection of pathogens and a deficiency in the control of its product, generating losses of up to 40%. In this context, a mobile application (APP) based on image recognition was developed for the detection of seasonal pests and diseases of the blueberry crop. The app offered a technological and innovative solution, where it issues the diagnosis of the pathogens detected and provides recommendations of agrochemicals to be applied in their crop. To do this, existing methodologies such as: Machine Learning and Extreme Programming were used, specific objectives were met such as creating a repository of 5,325 images, as well as developing a CNN model in Python language for training and validation, for the integration of the model and the App the Angular and Ionic frameworks were used. The result of the trained model showed its accuracy of 95.3%, detecting and classifying pests and diseases, tests were carried out using the confusion matrix with 15% of the data obtaining the accuracy for alternaria of 95.2%, healthy leaf 100%, and whitefly 91.1%, it is interpreted that the model has a high accuracy in all trained classes. Finally, it is concluded that the application provides accurate predictions, but it is recommended to continue validating and adjusting as necessary to predict the diseases and pests of the blueberry crop.

**Keywords:** Convolutional neural networks, blueberries, image processing.

## Introducción

Según la organización de las Naciones Unidas para la alimentación y agricultura (FAO)[1], la producción mundial agrícola es afectada desde un 20% hasta un 40% por plagas y enfermedades. En tanto el director general de Agro Perú informa que algunos de los países como, el centro internacional de agricultura tropical (CIAT) de Colombia y el Instituto Nacional de Innovación agraria (INIA) de Perú implementan tecnologías que permitan localizar los diferentes patógenos en los cultivos de arándanos y yuca[2], teniendo en cuenta su clima y su suelo según sea la zona.

El director ejecutivo de la Asociación civil privada de carácter gremial, sin fines de lucro, Cultivada informó que son 12.000 hectáreas cultivadas obteniendo como principales las zonas de producción La Libertad, Olmos en Lambayeque, Áncash, Lima, Ica y en poco tiempo se llegara a 20.000 ha [3]. Por otro lado, también indico que el cultivo de arándanos enfrenta una amplia lista de plagas y enfermedades, por tal motivo es crucial tener un buen conocimiento de las mismas para ser gestionadas adecuadamente, destacó que las evaluaciones de los patógenos son esenciales para determinar los productos agroquímicos necesarios y los momentos óptimos para su aplicación. Cabe resaltar, que las enfermedades notables en arándanos son la Roya del arándano, Oídium, Botrytis cinérea, Lasiodiplodia teobromas, Alternaria y entre las plagas se tiene a escarabajos “run run”, Gusano perforador, Gusano pegador, Moscas blancas, Arañita roja, Caracha. En la Libertad los campos de arándanos se tuvo incidencias de 10 a 50% en la fruta por Botrytis cinérea y con ella también se vio la reflejado el hongo Alternaria ya que ambos afectan en temporadas de invierno-verano cuando existe humedad. Este último se muestra con puntos rojos en las hojas dificultando la realización de la fotosíntesis, generando daño de defoliación y calidad de la fruta por el alto índice de estrés que afecta al peso de la fruta. Por otro lado, las agroindustriales productoras de arándanos tienen problemas con la plaga mosca blanca que ha demostrado ser un problema significativo, afectando las plantas desde su instalación en el campo y propagándose rápidamente, informo el entomólogo Juan Carlos Cabrera [4], esta plaga suele propagarse en grupos de 50 a 100 insectos en hojas tiernas o brotes dejando sus huevos que luego se convierten en ninfas, estas botan una melaza causando debilitamiento de la planta por lo que se debe hacer un correcto control desde que la planta esta pequeña para que no afecte a la fruta. El ingeniero Pablo Saldias Caro, nos indicó que en los campos de FrusanAgro en Olmos, la plaga mosca blanca y la enfermedad alternaria son las que se visualizan en un 40% a 75%, esto hace que realicen evaluaciones calendarizada de 2 a 3

veces por semana manejado por sistema de lotes que son 24 y en cada lote hacen 25 paradas(puntos) evaluando 4 plantas por parada finalmente son 100 plantas evaluadas. El problema es, al momento que se realizan dichas evaluaciones el personal no diferencia las plagas o las enfermedades, entonces hace una evaluación no acertada al momento de enviar el reporte a su jefe inmediato, ya que las imágenes las envían vía WhatsApp del 100% el 30% se pierde, lo que hace que vuelvan a campo a evaluar generando una pérdida de tiempo, al no identificar si las imágenes son de dicho punto evaluado, por consiguiente no se puede ver el nivel de infestación correcto del campo por ende no podrá aplicar los agroquímicos que son necesarios para mitigar dicha plaga o enfermedad generando que estas avancen al no ser controlada correctamente obteniendo pérdidas en cosecha del 20% al 40% y finalmente económicas. Con todas estas dificultades se ve en la necesidad de hacer la siguiente pregunta, ¿De qué manera la implementación de un aplicativo móvil basado en reconocimiento de imágenes puede apoyar la detección de plagas y enfermedades estacionales en el cultivo de arándanos en el distrito de Olmos?

Esta investigación se justifica científicamente por la escasez de aplicaciones móviles en el cultivo de arándanos en la región Lambayeque, así como la falta de práctica en temas específicas de plagas y enfermedades en este cultivo dentro de la empresa en estudio. Es, por consiguiente, que existe la necesidad de desarrollar herramientas que sean innovadoras para abordar este tipo de problemática y mejorar el control agrícola en la región. Por otro lado, en el ámbito social se justifica; la implementación del aplicativo móvil beneficiará la empresa. Permitirá la detección oportuna y precisa de los patógenos en las plantas, entonces la empresa concederá tomar decisiones informadas, aplicar tratamientos adecuados y proteger su producción. Por último, desde la perspectiva tecnológica se justifica que la aplicación móvil representa una solución eficiente e innovadora para abordar la detección de enfermedades en el cultivo de arándanos permitiendo obtener resultados acertados, brindando un valor agregado a la empresa, mejorando así la producción en el ámbito agrícola. Por consiguiente, se planteó el objetivo general; desarrollar una aplicación móvil basado en el reconocimiento de imágenes para apoyar la detección de plagas y enfermedades estacionales del cultivo de arándano. Se logró cumplir con los siguientes objetivos específicos; crear un repositorio de imágenes afectadas del cultivo de arándano para la detección de las plagas y enfermedades estacionales. Así mismo desarrollar un modelo de sistema inteligente basado en redes neuronales convoluciones para la identificación de plagas y enfermedades estacionales del cultivo de

arándano. Al mismo tiempo diseñar e implementar el aplicativo móvil incorporando el modelo de ayuda en la detección de plagas y enfermedades, finalmente, se validó la fiabilidad y desempeño de la aplicación móvil para la detección de plagas y enfermedades estacionales por parte de los usuarios.

### **Revisión de literatura**

Para el sustento de esta tesis se hizo revisión de la literatura de diferentes fuentes, tenemos a: Caballero et al[5], describe la problemática que existió con la enfermedad de moniliasis en cultivos de cacao en Colombia, que afectó en un 40% de la producción y en otras ocasiones pérdidas hasta en un 100% del fruto, al investigar se dieron cuenta que existió una limitación al usar tecnologías de reconocimiento de imágenes, para ello desarrollaron un aplicativo móvil para detectar la enfermedad de moniliasis mediante reconocimiento de imágenes. Usó el 85% de data para entrenamiento y el 15% para validación del modelo, al validar la aplicación se obtuvo un porcentaje de predicción de 88% de cacao infectado por moniliasis y mazorcas buenas, a su vez el aplicativo da recomendaciones a los agricultores para que sepan que tratamiento efectuar frente a dicha enfermedad en el tiempo oportuno. Para Hernández et al[6] abarco la importancia de controlar las plagas y las enfermedades que afectan al cultivo de tomate, eligiendo a cenicilla polvorienta por las características visuales, aplicando tratamiento de imágenes mediante técnicas de visión computacional, que consiste en mejorar la iluminación de imágenes obtenidas en ambientes no controlados mediante las redes neuronales convolucionales. Para su data set, dividió las imágenes en 26 sanas y 26 enfermas, seleccionando el 50% para ser procesadas y el 50% no procesadas. Finalmente, los resultados obtenidos entre el grupo de imágenes que fueron mejoradas en cuando al color, brillo y las que no fueron procesadas, obteniendo del entrenamiento del 97.11%, en contraste con el 92.16% del RNC por lo que se recomendó que las imágenes sean mejoradas para lograr una exactitud del 80.76%.

Castro[7], en su investigación describe la problemática sobre las deficiencias al identificar las enfermedades en el cultivo de palta, la falta de herramientas tecnológicas para ser usadas y detectar en el momento adecuado afecta directamente en la economía de los agricultores. Para ello recolectó un total de 630 imágenes enfermas y sanas para poder extraer las características,

luego realizó el entrenamiento de los modelos haciendo uso de los siguientes algoritmos: Redes Neuronales con 85%, Support Vector Machines 96%, Random Forest 93,65% y Naive Bayes 89,68% de exactitud. Donde dividió su data\_set en 80% para entrenamiento y 20% para la evaluación de desempeño de los modelos uso la matriz de confusión donde obtuvo como resultado el algoritmo más eficaz para clasificar enfermedades en el cultivo de palto fue Support Vector Machines con un 96% de exactitud y el error del 4%. También Narciso et al [8], en su investigación analizaron los defectos comunes en los arándanos, y clasificarlos para exportación, eligieron las empresas de la región de la Libertad para optimizar sus procesos con tecnología de inteligencia artificial, teniendo en cuenta que los productores hacen el proceso de manera manual ocasionando pérdidas de la cubierta del fruto. Utilizaron el modelo de detección Faster R-CNN el cual utiliza redes neuronales convolucionales y propuesta de regiones, crearon su propio repositorio con las imágenes de arándanos de calidades diferentes con sus respectivos defectos logrando obtener un total de 820 imágenes, de las cuales emplearon 500 con los 5 efectos más resaltantes del fruto donde se ve reflejado en las empresas agroindustriales de la región La Libertad como son: picado de pájaro, desgarrado, corola, rojo y pedicelo. Las horas que se involucraron para el entrenamiento fue de cinco horas aproximadamente con 400 pasos, finalmente obtuvieron las precisiones en un rango del 24% al 40.5% para los cinco defectos de dicho fruto. En tanto, Pajares et al[9], abarcaron la problemática que existen en la empresa Damper dedicada a sembrar arándanos, donde indicó que las plagas que afectan dicho cultivo son *Heliothis* y *Argyrotaenia*, la investigación logró mejorar la precisión y exactitud del reconocimiento de plagas que afectan a las plantas de arándanos, mediante una aplicativo móvil basado en técnicas de machine learning, agilizando así el proceso de reconocimiento para los usuarios con experiencia deficiente en el sector agrícola, para este procedimiento se hizo uso del algoritmo de redes neuronales convolucionales apoyado en las librería de Tensorflow y keras desarrollado en lenguaje Python, por consiguiente se recolectaron 215 fotografías de *Heliothis* para entrenamiento y 118 para validación; para la plaga *Argyrotaenia*, obtuvo 165 y 89 imágenes; para plantas sin plaga, 159 y 101 respectivamente. Finalmente, la verosimilitud al momento de reconocer las plagas fue de 5,2535 usando la solución de machine learning en comparación de antes haber hecho el reconocimiento de manera manual que fue de 2.6122. La dimensión de sensibilidad fue 78,04% antes de usar machine learning, luego de implementar la solución en la aplicación el valor subió a 89,88%. Por consiguiente, esta investigación logro con una solución de machine learning

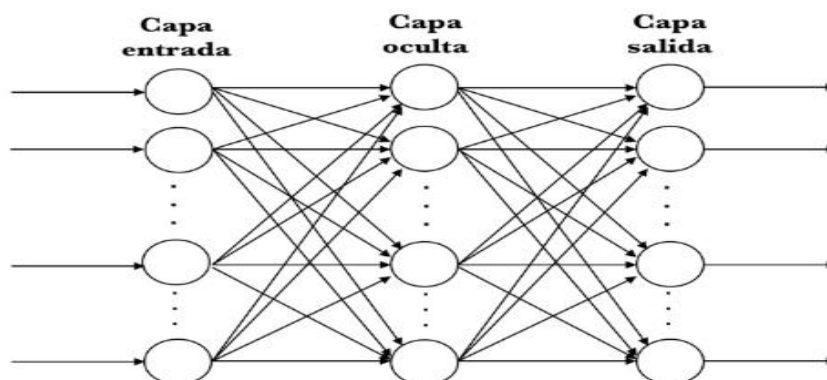
tener una influencia positiva en el reconocimiento de plagas para el cultivo arándano.

Para Zapata[10] la problemática que aborda es la inadecuada y tardía detección de enfermedades y plagas en el cultivo de arroz, ya que a simple vista del ser humano no es detectado, por ende genera un deficiente tratamiento de las mismas. Para dar solución a este problema realizó el desarrollo de un sistema inteligente de reconocimientos de imágenes para el diagnóstico de plagas y enfermedades haciendo uso de las redes neuronales convolucionales, con ello también hizo uso de un data set con imágenes infectadas para entrenamiento y validación del modelo apoyado con las librerías de TensorFlow y Keras. Finalmente, el resultado fue una aplicación móvil que toma fotos de patógenos y proporciona un diagnóstico preciso, indicando la enfermedad o plaga encontrada con el porcentaje de severidad también da recomendaciones de tratamiento y costos. Según Piscoya[11] abordó la problemática de la incorrecta identificación y suministración de agroquímicos para las enfermedades y plagas del cultivo de sandía en el distrito de Ferreñafe, desarrolló una aplicación móvil para identificar y brindar el agroquímico correcto, para ello extrajo las características más relevantes de plagas y enfermedades para brindar una mejor información, también ayudo en la reducción de tiempo promedio al momento de identificarlas en una velocidad de 1.693 segundos, realizado el estudio se concluye que el 82.6% indicó que aumento su nivel de conocimiento con la información proporcionada en el aplicativo, con ello podrá aplicar el agroquímico correcto y bajar los costos en sus cultivos de sandías. Por último, Chaname [12] nos narra la problemática que existe en la agroindustrial Ana Banana de Piura, sobre la deficiencia en el diagnóstico de Enfermedades del cultivo de plátano orgánico, utilizo la metodología COMMONKADS, además empleo NetBeans y Java como plataforma de desarrollo. Valido la seguridad y fiabilidad del sistema mediante lógica difusa durante 5 días de la semana en la Empresa. Teniendo una fiabilidad de un 100% y la seguridad de los datos reales obtenidos durante los 5 días considerados todo ello guardados en base de datos MySQL. Al demostrar la fiabilidad de los datos adquiridos en el primer día fue de 71% aproximado, el segundo día de 62%, vemos en el tercer día de un 59%, en el cuarto día se observa un 50 % de fiabilidad, mientras que el quinto día es 78% aprox. De acuerdo a las pruebas realizadas en los 5 días considerados, arrojaron un 90 % de fiabilidad lo que lo hace seguro el sistema ya que las pruebas realizadas fueron reales. En esta revisión de literatura, se observó que estas investigaciones nos orientaron hacia la elaboración de la investigación en estudio, que incluyó la incorporación de un aplicativo móvil. Este aplicativo tenía como propósito ayudar a los productores a identificar las plagas y

enfermedades que afectaban a sus cultivos, lo que resultaría una mejora de los rendimientos de producción.

Se da pase a las Bases teóricas que sustentan la investigación en cada uno de sus argumentos, Se inició con una revisión de la literatura sobre aprendizaje automático (ML) y aprendizaje profundo (DL). Este análisis se realizó para conocer los principales algoritmos que existen en estas dos áreas, tener conocimiento cómo funcionan y seleccionar el que mejor se adapta al objetivo del proyecto. Hurwitz [13] define el ML como una disciplina de la informática que crea algoritmos que tienen la capacidad de aprender a partir de datos, sin necesidad de instrucciones programadas de manera explícita, estos algoritmos aprenden a medida que se ingresan nuevos datos mejorando su capacidad para predecir resultados. El Deep Learning es un subcampo del ML que usa redes neuronales, las redes neuronales son modelos semejando a un cerebro humano, que están conectadas por una serie de capas de neuronas totalmente interconectadas. Hurwitz [13] afirma que el DL se utiliza a menudo en aplicaciones de visión por computadora, voz y reconocimiento de imágenes. A continuación, se revisan los algoritmos de aprendizaje supervisado, que clasifican los datos en grupos atendiendo a datos previamente etiquetados. Tenemos el **Algoritmo de Regresión**: el modelo de aprendizaje automático tiene una variable dependiente y se comprende con variables que cambian, logrando ser útil pronosticar y predecir fraudes[14]. **Las Redes neuronales Recurrentes**: Una red RNN puede tomar simultáneamente una secuencia de entradas y producir una secuencia de salidas, estas redes han demostrado tener un alto rendimiento en reconocimiento del habla [15]. **Algoritmos de árbol de decisión**; su mismo nombre lo indica sirve para tomar decisiones, el árbol tienen una estructura que inicia con un nodo raíz, y tiene sus ramas con nodos internos y nodos hoja[16], este algoritmo sirve para predecir riesgos de enfermedades y recomendar productos. **Las redes neuronales convolucionales (CNN)**; Según Khan et al. [17] las CNN son semejantes a las redes neuronales estándar, lo diferente lo diferente es que cada unidad en una capa y es un filtro bidimensional convolucional al ingresar la capa permite identificar características de las imágenes para que la red pueda aprender y es útil para detección de objetos, reconocimiento facial y clasificación de imágenes. Por consiguiente, en esta investigación se trabajó con el algoritmo CNN, debido a que su propósito es el reconocimiento y la clasificación de imágenes. Las CNN se componen de las siguientes capas[18]:

- Capa de entrada: Ingresan los datos de entrada.
- Capas convolucionales: Aplican filtros a los datos de entrada para identificar características.
- Capas MaxPooling: Reducen el tamaño de los datos de entrada, manteniendo las características más importantes.
- Capa flatten: Convierte los datos de entrada en un vector.
- Capa softmax: Genera la salida de la red.



**Fig. 01 Red neuronal Deep Learning**

En su investigación [10] señaló que para realizar una evaluación de un modelo de aprendizaje profundo se debe dividir la data en tres conjuntos como es data de entrenamiento, validación y prueba, principalmente se elige la data de entrenamiento para correr el modelo y evaluar si el algoritmo está aprendiendo con los parámetros de entrada asignados, encaso no logre reconocer los datos se vuelve a modificar y entrenar, posteriormente se validan los datos y se realizan las pruebas. Por consiguiente, para desarrollar el modelo CNN existen librerías como Keras y Tensorflow que son de código abierto [11], utilizadas para entrenar redes neuronales con multi-entradas y multi-salidas arrojando resultados en un tiempo prudente. En cuanto a la implementación se utilizó el software Anaconda [12] ya que es el administrador de paquetes para el lenguaje Python en el cual se pueden desarrollar modelos de aprendizaje automático y aprendizaje profundo, también permite realizar pruebas con volúmenes de datos, por otro lado, la comunidad que crea librerías ha fortalecido este lenguaje con uso de inteligencia artificial logrando que haga el reconocimiento, tratado de imágenes y videos a tiempo real [4]. Dentro del proyecto de investigación nos sirvió para implementar el modelo de reconocimiento de las

enfermedades y las plagas del cultivo de arándano.

Para el desarrollo del aplicativo móvil se usó Ionic que es un framework [13], con el propósito de crear aplicaciones híbridas. Además, se integra fácilmente con otros frameworks de frontend como Angular, emplea también Cordova o Capacitor, lo que le confiere un diseño funcional, sencillo y limpio, estructurado por el lenguaje HTML, JavaScript y TypeScript [14]. Se utilizó PostgreSQL que es un gestor libre para procesar datos de un modelo cliente servidor como está estructurado este proyecto.

Para el desarrollo del aplicativo móvil se usó Ionic que es un framework desarrollado en 2013 por Drifty Co[19], con el propósito de crear aplicaciones híbridas. Se trata de un proyecto de código abierto que utiliza HTML, CSS y JavaScript, lo que permite desarrollar aplicaciones móviles y web desde un mismo código base, ya que son compatibles entre sí. Además, se integra fácilmente con otros frameworks de FrontEnd como Angular, React y Ionic emplea también Cordova o Capacitor, lo que le confiere un diseño funcional, sencillo y limpio. Se utilizó el framework angular elaborado por Google, crea aplicaciones web con arquitectura cliente servidor, desarrollado en lenguajes TypeScript ,Html y JavaScript[20].

Para comprender sobre nuestro objeto de estudio tenemos: El arándano o blueberry; planta que mide de 20 a 50 cm de altura, familia de las ericáceas, con hojas aserradas y ovaladas, flores, de color rosado o blanco verdoso sus frutos son azuladas comestible[21]. Por consiguiente, es también propensa a enfermedades y plagas que dificultan su desarrollo desde plantación hasta su cosecha. Las enfermedades más comunes del arándano según France son[22]: **Pudrición gris (botrytis cinérea)** esta enfermedad se aloja en las frutos y flores mostrando lesiones de color café desde el centro hacia los bordes de las hojas produciendo necrosis, y en las flores atizona en su totalidad por ende también el racimo floral. En la fruta madura se presenta como un ablandamiento de la misma, con una tonalidad opaca y se muestra deshidratado. Puede que este hongo hace su aparición desde cuando el fruto está en cuaje y mostrarse cuando la fruta este madura. El control para la prevención debe ser al inicio de floración hasta la pre cosecha. **Alternaria:** [17]Es un hongo que se transporta por el viento y afecta a las hojas de las plantas de arándanos, aparece con frecuencia cuando hay lluvias mayores a 2mm y temperaturas entre 18°C a 28°C mostrando puntos rojos en las hojas, debilitando la planta. Se debe dar una prevención temprana para evitar la defoliación y que la planta realice la fotosíntesis completa. En el control cultural se debe tener en cuenta que las zonas deben ser ventiladas quiere decir

que haya un buen drenaje, evitar el crecimiento excesivo de malezas, cuando se realiza la poda se debe recoger todos los rastrojos de los campos ya que el hongo se reproduce rápidamente en las hojas secas. Entre las plagas que se dan en los arándanos tenemos a **Mosca Blanca** (*Bemisia tabaci* Gennadius) [17] plaga donde los insectos miden de 0.9 a 1 mm, su aparato bucal es picador - chupador atacando a hojas tiernas de color verdosas, lo que hace el insecto excreta melaza y deja sus huevecillos, estos suelen moverse en grupos de 50 a 100 individuos que debilitan la planta, y la miel que bota la mosca blanca sobre la fruta causa daño y no hay forma de eliminarlo ocasionando un daño cosmético.

## **Materiales y métodos**

### **Tipo de investigación**

Según el Manual de Frascatti [23], esta tesis es de tipo Aplicada, con un nivel de investigación preexperimental con un diseño pre test, post test aplicado a un solo grupo para evaluar los cambios antes y después de la aplicación del modelo, tiene como objetivo dar soluciones útiles a problemas específicos como el reconocimiento de imágenes con plagas y enfermedades del cultivo de arándanos.

### **Métodos de investigación**

Los métodos de la investigación empelados fueron:

TABLA I  
MÉTODOS DE INVESTIGACIÓN

<b>Método</b>	<b>Descripción</b>
Analítico	Analizar los problemas que existen en la empresa respecto a las plagas y enfermedades del cultivo de arándano. Revisar la literatura que respalden la investigación.
Deductivo	Deducir la problemática y ayudo a realizar predicciones de plagas y enfermedades
Implementación	Se realizó la ejecución del aplicativo móvil

A continuación, se muestra la siguiente tabla con las técnicas e instrumentos de recolección de datos empleados en esta investigación

TABLA II  
TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Técnicas	Instrumento	Propósito
Observación	Ficha de observación (Anexo01)	Conocer las características de plagas y enfermedades a utilizar
Entrevista	Guía de entrevista (Anexo 02)	Conocer los problemas que existen en la empresa respecto a las plagas y enfermedades, como realizan el proceso de detección y el diagnóstico de las mismas.
Encuesta	Guía de cuestionario (Anexo03)	Conocer el nivel de fiabilidad del aplicativo móvil por parte de los usuarios.

Posteriormente, se definen las variables:

**variable independiente:** Aplicación móvil basado en reconocimiento de imágenes

**Variable dependiente:** Efectividad en la detección de plagas y enfermedades estacionales del cultivo de arándanos.

### Población

- Numero de lotes = 24
- Puntos por lotes =25
- Muestras por punto =4

La población total (N) se calcula:

- $N=24*25*4$
- $N= 2400$

### Muestra

Para calcular el tamaño de la muestra(n), se utilizó la fórmula de muestra para poblaciones finitas.

$$n = \frac{N * z^2 * p * (1 - p)}{E^2 * (N - 1) + z^2 * p * (1 - p)}$$

Donde:

N = Población total (2400)

Z = Valor z para el nivel de confianza (1.96)

P = Proporción esperada de la característica a estudiar (0,5)

E = Margen de error (0,05)

Se calcula:

$$z^2 = (1.96)^2 = 3.8416 \quad , \quad p \cdot (1 - p) = 0.5 * 0.5 = 0.25 \quad , \quad E^2 = (0.05)^2 = 0.0025$$

Se sustituye los valores en la formula:

$$n = \frac{2400 * 3.8416 * .025}{0.0025 * (2400 - 1) + 3.8416 * 0.25}$$

$$n = \frac{23049.6}{6.9579}$$

$$n = 331.47$$

Redondeando el número se necesita una muestra aproximadamente de 331 plantas y proporcionar resultados representativos y confiables sobre la presencia de plagas y enfermedades en las plantas evaluadas.

TABLA III  
MATRIZ DE CONSISTENCIA

FORMULACIÓN DEL PROBLEMA	MÉTODOS DE INVESTIGACIÓN		
¿De qué manera se puede apoyar en la detección de plagas y enfermedades en el cultivo de arándano en el distrito de Olmos?	TIPO DE INVESTIGACIÓN		
	Aplicada		
OBJETIVO GENERAL	MÉTODO	DESCRIPCIÓN	
Desarrollar una aplicación móvil basado en el reconocimiento de imágenes para apoyar la detección de plagas y enfermedades estacionales del cultivo de arándano	Analítico	Analizar los problemas que existen en la empresa respecto a las plagas y enfermedades del cultivo de arándano. Revisar la literatura que respalden a la investigación.	
	Deductivo	Deducir la problemática y ayudo a realizar predicciones de plagas y enfermedades	
	Implementación	Se realizó la ejecución del aplicativo móvil	
	TÉCNICAS	INSTRUMENTO	PROPÓSITO
	Observación	Ficha de observación (Anexo01)	Conocer las características de plagas y enfermedades a utilizar
	Entrevista	Guía de entrevista (Anexo 02)	Conocer los problemas que existen en la empresa sobre las plagas y enfermedades, y como realizan el proceso de detección y el diagnóstico de las mismas.
	Encuesta	Guía de cuestionario (Anexo03)	Conocer el nivel de fiabilidad del aplicativo móvil por parte de los usuarios.
OBJETIVOS ESPECÍFICOS	DESCRIPCIÓN DEL LOGRO DE LOS OBEJTIVOS ESPECÍFICOS		INDICADORES
<b>OE-1:</b> Crear un repositorio de imágenes afectadas del cultivo de arándano para la detección de las plagas y enfermedades estacionales del distrito de olmos	Se creo un repositorio con las imágenes de la plaga y enfermedad seleccionada para entrenamiento		# número de imágenes por plaga y enfermedad Número total de imágenes a entrenar
<b>OE-2:</b> Desarrollar un modelo de sistema inteligente basado en redes neuronales convoluciones para la identificación de plagas y enfermedades estacionales del cultivo de arándano	Se realizo una revisión de la literatura para comprender como funcionan las redes neuronales convolucionales		Porcentaje del nivel de reconocimiento de imagen con plaga o enfermedad mediante la matriz de confusión
<b>OE-3:</b> Diseñar e implementar el aplicativo móvil incorporando el modelo de ayuda en la detección de plagas y enfermedades estacionales del cultivo de arándano	Se incorporo el modelo de CNN con el aplicativo móvil para el reconocimiento de las imágenes con plaga o enfermedad		Porcentaje de predicción de imágenes con plaga o enfermedad
<b>OE-4:</b> Validar la fiabilidad y desempeño de la aplicación móvil para la detección de plagas y enfermedades estacionales por parte de los usuarios	Se realizo la validación por parte de los usuarios usando el aplicativo móvil		Porcentaje de validación relacionado a la aplicación móvil al 85% por los usuarios. Usando escala de Likert con 5 niveles

## **Resultados y discusión**

### **Resultados en base a la metodología utilizada**

Durante el proceso de la investigación se utilizaron las siguientes metodologías, Metodología de Machine Learning usado para el modelo de redes neuronales convolucionales y la Metodología de Extreme Programming (XP) para el desarrollo del software. Para el cumplimiento de esta investigación se inició con las fases de la metodología de machine learning [23].

#### **Fase 1: Definir el objetivo**

Para iniciar el desarrollo de un repositorio de imágenes de plagas y enfermedades del cultivo de arándanos, se realizó un análisis de la realidad problemática. Para ello, se utilizó una herramienta de recolección de datos que se aplicó a un Ingeniero Agrónomo encargado del Área de plagas de la empresa productora de arándanos, de la cual se obtuvo información sobre las plagas y enfermedades que afectan al cultivo de arándano (Anexo 01). El Ingeniero Agrónomo indicó que las plagas y enfermedades más comunes en el cultivo de arándanos son: las plagas como mosca blanca y trips y las enfermedades son alternaria y mildiu (Anexo01 pregunta 1). Además, se observó que la plaga y enfermedad notables en la temporada de julio a diciembre es mosca blanca y alternaria, respectivamente (Anexo 01, pregunta 4). En base a la información, se decidió recolectar imágenes de la plaga mosca blanca, la enfermedad alternaria y hojas sanas, se inició ingresando a campo por cada lote se tomó fotos a las hojas de las plantas en diferentes ángulos desde julio hasta diciembre del 2023 para poder recolectar las imágenes que fueron trabajadas en el proyecto.

Para esta primera fase se plantea definir el objetivo de reconocer las plagas o enfermedades del cultivo de arándanos permitiendo la entrada de imágenes para lograr alcanzar una precisión mayor o igual a 80% [24].

Algunas carencias que podrían afectar al cumplimiento del desarrollo del proyecto serían: obtener una data muy limitada lo que generaría que el entrenamiento del modelo se retrase y no alcance la precisión esperada, otra sería que el modelo elegido no sea el adecuado, para ser entrenado con imágenes debido al tiempo de desarrollo, y finalmente la flexibilidad del diseño no es adaptable a cualquier entorno por desconocimiento del código, para evitar estas deficiencias se elaboró a un cronograma de actividades que refleja las fechas de entregas de

los avances en el desarrollo del proyecto.

## Fase 2: Recolección de la data

Según Murphy [25] en su libro dice que la cantidad de datos necesarios para crear un data set y luego sea entrenado en un modelo de aprendizaje automático depende de la complejidad de la tarea que el modelo debe realizar. Para tareas simples, como la clasificación de imágenes binarias, se puede entrenar un modelo con tan solo unos pocos cientos de datos. Para la detección de objetos o clasificar imágenes, se necesitan miles de datos. Para Pedro Domingo[26] en su artículo nos muestra un ejemplo para clasificar imágenes de perros y gatos, se necesitarían al menos 100 imágenes por clase para ser entrenada. En esta investigación, se recolectaron 1600 imágenes en campo con ayuda de la cámara del móvil, en primera instancia se usaron para entrenar el modelo, pero los resultados obtenidos no eran certeros y se decidió recolectar más imágenes sumando un total de 5.500, para este caso tenemos cuatro clases: plaga “mosca blanca”, enfermedad “alternaria”, “hoja sana” y clase “otros” de las cuales se recolectaron 1.375 imágenes por cada clase. Estas imágenes se almacenaron en una carpeta de nombre data\_set, que fueron clasificadas en la fase3 para obtener un data limpio que posteriormente se utilizó para entrenar el modelo en la fase 5 de la investigación.

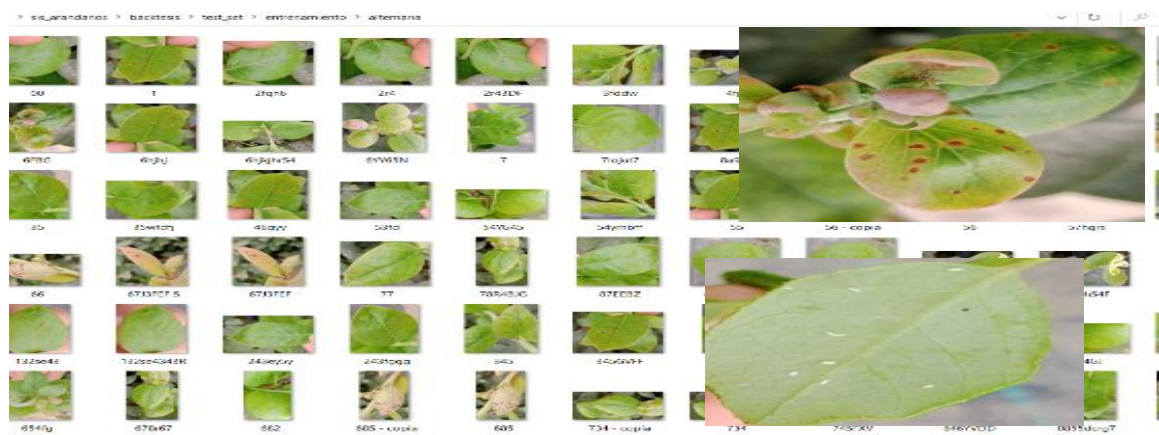


Fig. 02: Recolección de las imágenes

## Fase 3: Preparar la data

Para la fase 3, se dio inicio con la preparación de los datos mediante un proceso de eliminación de imágenes (Anexo 03) que consideró características de calidad de imagen, tales como el

enfoque y la iluminación que dificultan el entrenamiento del modelo [27], al finalizar se seleccionaron un total de 5.325 imágenes aptas para entrenar el modelo en la fase5.

#### **Fase 4: Modelamiento**

La fase 4 de la investigación denominada Modelamiento, Para [28] y[29] realizaron en sus tesis una distribución de sus datos en dos carpetas: entrenamiento el 80% y validación 20%, obteniendo como resultado un 97% y 98% de precisión, mientras para [30] utilizo el 70 % para entrenamiento, 30 % para validación su resultado fue mayor al 80% lo que significa que sus modelos tiene un alto grado de exactitud y son usados para futuros proyectos, es así como se realizó la distribución adecuada para este proyecto. En la carpeta de entrenamiento se crearon cuatro subcarpetas con los nombres, "mosca blanca", "alternaria", "hoja sana" y "otros". Estas subcarpetas contienen todas las imágenes que serán utilizadas durante el proceso de entrenamiento del modelo. Cabe resaltar que se siguió un procedimiento similar en la carpeta de validación, dividiéndola en cuatro subcarpetas para llevar a cabo las validaciones correspondientes en la fase6.

Se realizó el código del modelo de CNN como muestra la siguiente figura 03 donde se inició importando las librerías de Tensor Flow y Keras programadas en lenguaje Python que ayudaron a entrenar el modelo[31], se definieron los parámetros de red, entrenamiento, el tamaño del lote (32), el número de épocas (40), el ancho y la altura de las imágenes (150x150), luego se inició con la función Sequential (), que conecta las capas de manera lineal, se agregó la primera capa convolucional (conv2D ()) que extrae características de las imágenes aplicando un conjunto de filtros de 32 a la imagen de entrada y activa la función relu, los filtros se mueven a través de la imagen y calculan una respuesta para cada píxel. La capa de agrupación máxima (MaxPooling2D ()) redujo el tamaño de la imagen al seleccionar el valor de píxel máximo en cada ventana de 2x2, esta capa reduce el sobreajuste y mejorar el rendimiento general del modelo, lo que hace es transformar la representación de características conjuntas en información valiosa al mantener información útil y eliminando información irrelevante[29]. La segunda y tercera capa convolucional (conv2D ()) aplicaron un filtro de 64 y 128, respectivamente. La capa de aplanamiento (Flatten ()) convirtió la salida de la capa de agrupación máxima en un vector unidimensional, esto permite que la capa siguiente de la red reciba la entrada como un vector. La capa densa (Dense ()) es una capa neuronal totalmente

conectada, esta capa aprende a combinar las características extraídas por las capas anteriores para predecir la clase de la imagen, la salida de la capa densa se activó con la función softmax, que asigna una probabilidad en este caso, las clases son "alternaria", "mosca blanca", "hoja sana" y "otros".

Posteriormente, se corrió el modelo con el optimizador Adam con las métricas de predicción y pérdida que son utilizadas para evaluar el rendimiento del modelo, la pérdida mide la diferencia entre las predicciones del modelo y las etiquetas reales y la precisión mide las predicciones correctas, durante el entrenamiento y la validación en las fases 5 y 6 del proyecto.[32].

```

42 # Crear la arquitectura del modelo CNN
43 model = keras.Sequential([
44     layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)), #EXTRAE
45     layers.MaxPooling2D(2, 2),
46
47     layers.Conv2D(64, (3, 3), activation='relu'),
48     layers.MaxPooling2D(2, 2),
49
50     layers.Conv2D(128, (3, 3), activation='relu'),
51     layers.MaxPooling2D(2, 2),
52
53     layers.Flatten(),
54     layers.Dense(512, activation='relu'),
55     layers.Dropout(0.5), # Regularización Dropout para reducir el sobreajuste <- II
56     layers.Dense(3, activation='softmax') # Capa de salida con 4 clases
57 ])
58
59
60
61 model.compile(optimizer='adam', # Optimizador Adam con tasa de aprendizaje
62              loss='categorical_crossentropy', # Función de pérdida para clasificac
63              metrics=['accuracy']) # Métrica de precisión para evaluar el modelo

```

Fig. 03: Desarrollo del modelo de CNN

### Fase 5: Entrenar el modelo

En esta fase, se llevó a cabo el proceso de entrenamiento del modelo CNN, utilizando el 70% de la data donde el primer escenario se entrenó con 125 imágenes y 80 épocas, pero los resultados no fueron favorables ya que el modelo estaba memorizando dando como precisión un 99,5 % de los datos y no era capaz de diferenciar imágenes nuevas ingresadas al modelo como pruebas, por tal motivo se decidió entrenar nuevamente agregando más imágenes al data

set con hojas infectadas, y se añadió imágenes a la clase “hoja sana” y clase “otros” para que el modelo pueda diferenciar cualquier otra imagen que no pertenece a un patógeno. Finalmente, el modelo se entrenó con un total de 3.727 imágenes, recorriendo todas las imágenes para extraer sus características. Se obtuvo el resultado del entrenamiento en la figura 04 el gráfico de precisión por época, la línea azul representa la precisión de la data entrenada y la naranja la precisión de la data de validación. Los datos de entrenamiento su precisión fue 95.3% en la época 38, quiere decir que está clasificando correctamente las clases ingresadas, extrayendo las características esenciales de cada plaga o enfermedad, mientras que los datos de validación alcanzan la precisión de 94.5% en la época 38, donde se puede observar que el modelo si está alcanzando un buen porcentaje de validación de los datos ingresados, como se ve en la gráfica las dos líneas están encaminadas y no difieren mucho. Estos resultados indican que el modelo está aprendiendo a predecir correctamente los datos de entrenamiento y validación.

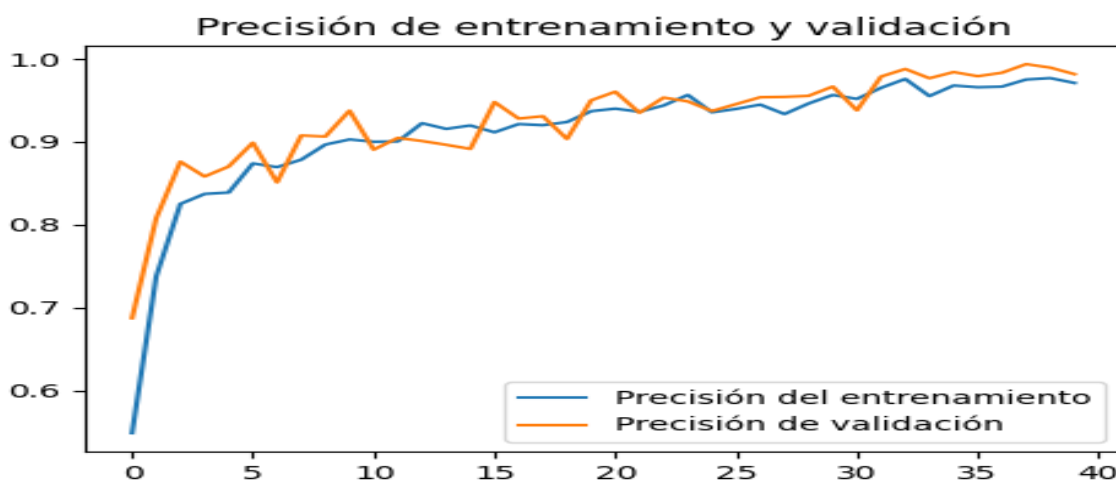


Fig. 04: Gráfico de precisión

Del mismo modo se analizó los valores de la figura 05 muestra el gráfico de pérdida por época, donde la línea azul muestra la pérdida de entrenamiento y línea naranja la pérdida de validación. En la época 39, la pérdida de entrenamiento disminuye 0,07 %, mientras que la pérdida de validación alcanza su mínimo de 0,04 %. En conclusión, los resultados del gráfico de pérdida indican que el modelo ha mejorado el sobreajuste y es capaz de ingresar nuevas imágenes al

modelo.

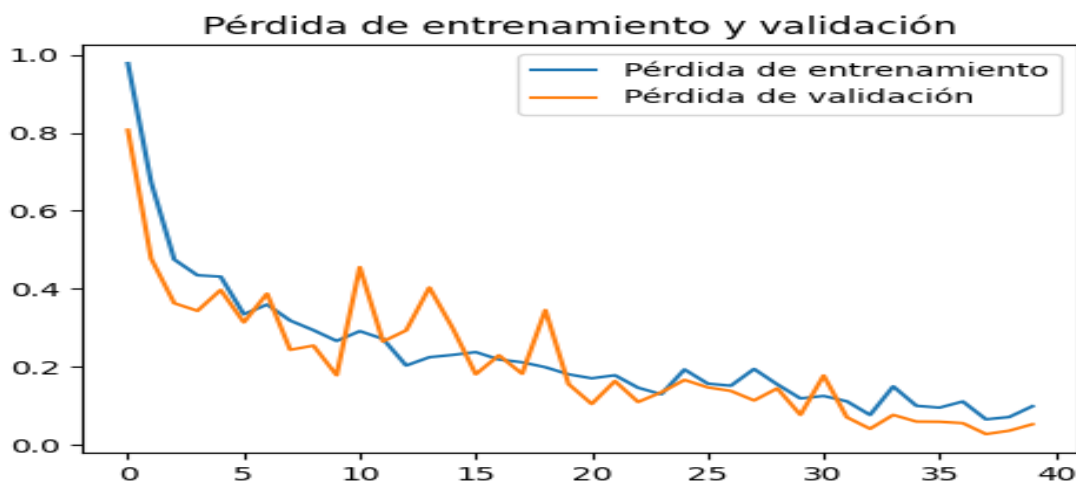


Fig. 06: Gráfico de pérdida

### Fase 6: Validación del modelo

En esta fase, se realizó la validación del modelo con 390 imágenes balanceada entre las cuatro clases. Para ello, se utilizó la matriz de confusión, una métrica utilizada para calcular la precisión por clase del modelo[33]. En este caso, tenemos cuatro clases: “alternaria”, “mosca blanca”, “hoja sana” y clase “otros”. Según muestra el gráfico de matriz de confusión (figura 07), se ingresaron 105 imágenes de alternaria, 105 de mosca blanca, 100 de hoja sana y 80 imágenes para clase otros.

El análisis del gráfico de matriz de confusión muestra la precisión y sensibilidad por cada clase donde:

- La clase alternaria arroja la precisión del 95.2 % y su sensibilidad es de 91% lo que demuestra que el modelo está clasificando bien las instancias de alternaria y confunde con un ligero número de 5 instancias con otras clases con alternaria.
- La clase hoja sana su precisión fue 100 % y sensibilidad de 98% lo que significa que esta clase tiene una alta precisión y rara vez confunde la clase sana con otros.
- Para la clase mosca blanca su precisión fue de 91.1% y su sensibilidad fue de 98% tiene un equilibrio entre la precisión y sensibilidad demostrando que si está clasificando correctamente la clase mosca blanca y muestra un ligero número de 9 instancias que

confunde al clasificar como mosca blanca.

- Para la clase otros se obtuvo la precisión de 91% y sensibilidad de 97% demostró que esta clase también clasifica bien la clase con un ligero número de 2 instancias que fueron erróneas.

A continuación, se detalló los resultados por cada clase ingresada en la matriz de confusión con las siguientes formulas:

- **Precisión de clase**

$$\text{Precisión de clase} = \frac{TP}{TP + FP}$$

- **Sensibilidad (Recall) o tasa de verdaderos positivos**

$$\text{Recall} = \frac{TP}{TP + FN}$$

#### **Clase 01: Alternaria:**

- Verdaderos positivos (TP): 100
- Falsos positivos (FP): 5
- Falsos negativos (FN): 9

$$\text{Precisión de Clase} = \frac{100}{100 + 5} = \frac{100}{105} = 95.23\%$$

$$\text{Recall} = \frac{100}{100 + 9} = \frac{100}{109} = 92.00\%$$

#### **Clase 02: Hoja sana**

- Verdaderos positivos (TP): 98
- Falsos positivos (FP): 0
- Falsos negativos (FN): 2

$$\text{Precisión de Clase} = \frac{98}{98 + 0} = 100\%$$

$$\text{Recall} = \frac{98}{98 + 2} = 98\%$$

**Clase 03: Mosca blanca**

- Verdaderos positivos (TP): 96
- Falsos positivos (FP): 9
- Falsos negativos (FN): 2

$$\text{Precisión de Clase} = \frac{96}{96 + 9} = \frac{96}{105} = 91\%$$

$$\text{Recall} = \frac{96}{96 + 2} = \frac{96}{98} = 98\%$$

**Clase 04: otros**

- Verdaderos positivos (TP): 78
- Falsos positivos (FP): 7
- Falsos negativos (FN): 2

$$\text{Precisión de Clase} = \frac{78}{78 + 7} = \frac{78}{85} = 92\%$$

$$\text{Recall} = \frac{78}{78 + 2} = \frac{78}{80} = 97\%$$

**Precisión global del modelo**

$$\begin{aligned} \text{Exactitud} &= \frac{TP_{alternaria} + TP_{phojasana} + TP_{moscablanca} + TP_{otros}}{\text{total de instancias}} \\ &= \frac{100 + 98 + 96 + 78}{390} = 95.3\% \end{aligned}$$

Finalmente se calculó la precisión (Exactitud) global del modelo donde se realiza una sumatoria de todos los verdaderos positivos de cada clase y el total de todas instancias ingresadas donde se obtuvo como resultado una precisión del 95.3%, esto significa que el modelo clasifica y reconoce las imágenes que son ingresadas a evaluar, aunque hay una cierta confusión entre alternaria y mosca blanca, pero el modelo si ha aprendido correctamente.

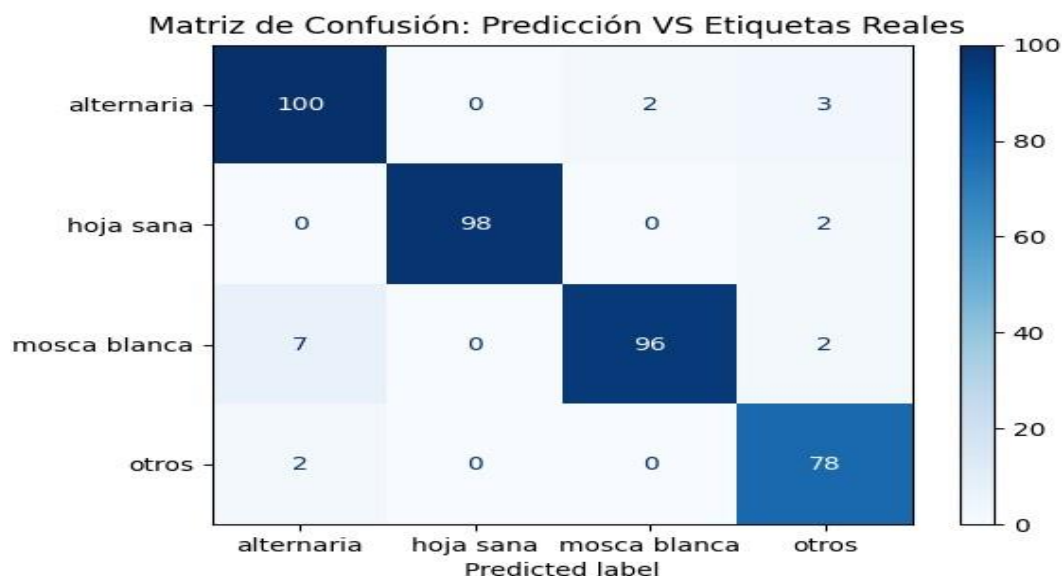


Fig. 07: Matriz de confusión

### Fase 7: Predicción

En la fase predicción, se realizó la ejecución del modelo mediante pruebas del 15% de data se evalúa cada imagen ingresada con su respectivo porcentaje de predicción y el nombre de la clase predicha como se muestra en la Figura 08, se ingresaron 240 imágenes en total dividiéndola entre las cuatro clases se asignó 60 imágenes: “alternaria”, “hoja sana”, “mosca blanca” y “otros”. Se inicio definiendo las clases, luego se llama al modelo entrenado, también la carpeta con las imágenes de pruebas posterior a eso se hace el procesamiento de las imágenes para que el modelo pueda reconocer imágenes con extensiones '.jpg', '.png', '.jpeg', se redimensionan las imágenes a 150\*150 pixeles. Luego, se normaliza el arreglo dividiendo cada valor de píxel por 255.0, lo cual escala los valores de píxel en el rango de 0 a 1, luego se asigna los valores para que muestre el nombre de la clase predicha con su respectivo porcentaje de predicción. Finalmente se analizan los datos de manera general donde la mayoría de las imágenes ingresadas muestra un porcentaje por encima del 95% lo que significa que el modelo tiene un alto desempeño al realizar predicciones con imágenes nuevas.

```

Imagen: IMG-20231115-WA0069.jpg
La clase predicha es: alternaria
El porcentaje de similitud de la clase predicha es: 99.77230429649353%
1/1 [=====] - 0s 36ms/step
Imagen: IMG-20231115-WA0074.jpg
La clase predicha es: alternaria
El porcentaje de similitud de la clase predicha es: 99.99996423721313%
1/1 [=====] - 0s 24ms/step
Imagen: IMG-20231115-WA0076.jpg
La clase predicha es: mosca blan
El porcentaje de similitud de la clase predicha es: 78.7135362625122%
1/1 [=====] - 0s 31ms/step
Imagen: IMG-20231115-WA0077.jpg
La clase predicha es: alternaria
El porcentaje de similitud de la clase predicha es: 81.27774596214294%
1/1 [=====] - 0s 39ms/step
Imagen: loio - copia.jpg
La clase predicha es: mosca blan
El porcentaje de similitud de la clase predicha es: 100.0%
1/1 [=====] - 0s 41ms/step
Imagen: nnt6u6yt - copia.jpg
La clase predicha es: mosca blan
El porcentaje de similitud de la clase predicha es: 100.0%
1/1 [=====] - 0s 31ms/step
Imagen: stevia-natural-des-1.jpg
La clase predicha es: alternaria
El porcentaje de similitud de la clase predicha es: 99.7018575668335%
1/1 [=====] - 0s 40ms/step
Imagen: S_HL 2272.JPG
La clase predicha es: hoja_sana
El porcentaje de similitud de la clase predicha es: 98.30600619316101%
1/1 [=====] - 0s 46ms/step
Imagen: THGF.jpg
La clase predicha es: mosca blan
El porcentaje de similitud de la clase predicha es: 99.99997615814209%
1/1 [=====] - 0s 39ms/step

```

Fig.08: Predicción

Se realizó el desarrollo de la aplicación con la metodología de Extreme Programming (XP).

### **Fase de planificación**

En la fase de planificación, se reunió con el cliente para definir los requerimientos funcionales y no funcionales (TABLA IV) del software para ello se elaboraron las historias de usuario que se utilizaron para describir las funcionalidades del sistema en construcción, de forma detallada y concisa [34]

TABLA IV: REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

FUNCIONALES	NO FUNCIONALES
El sistema permite registrar usuarios e iniciar sesión.	El sistema debe ser fácil de comprender y usar por parte del usuario
El sistema permite tener un usuario administrador.	El sistema debe ser documentado, tener un código limpio.
El sistema permite registrar los lotes con sus respectivos puntos.	Si el sistema tiene algún fallo tendrá un backup de respaldo en la base de datos.
El sistema permite mostrar el resultado del diagnóstico de plaga y enfermedad predichas.	
El sistema permite registrar 4 fotos por puntos para realizar el diagnóstico	
El sistema permite ingresar fotos con extensiones PNG y JPG	
El sistema generara un reporte de los detalles del diagnóstico.	
El sistema generara un reporte con recomendaciones de acuerdo al diagnóstico realizado	

- **Definir historias de usuario**

En la planificación de las historias de usuario, se asignaron los criterios y prioridades de acuerdo con la Tabla V.

TABLA V CRITERIO Y PRIORIDAD[35]

Criterio	Prioridad
Indispensable	1
Necesario	2
Conveniente	3

La Tabla V asigna una prioridad de 1 al criterio de indispensable, una prioridad de 2 al criterio de necesario y una prioridad de 3 al criterio de conveniente. Estas prioridades se asignaron a

las 7 historias de usuario identificadas para su ejecución, como se muestra en la Tabla VI.

TABLA VI HISTORIAS DE USUARIO

Numero de historia	Nombre de historia	Prioridad
H1	Inicio de sesión	1
H2	Gestionar usuarios	2
H3	Registrar lotes	2
H4	Registrar puntos	2
H5	Tomar fotos	1
H6	Gestionar diagnóstico	1
H7	Reporte recomendaciones	3

A continuación, se detalla cada historia de usuario.

TABLA V II H1 INICIO DE SESIÓN

<b>Historia de usuario</b>	
<b>Número: H1</b>	<b>Usuario:</b> Evaluador
<b>Nombre de historia:</b> Inicio de Sesión	<b>Prioridad:</b> 1
<b>Desarrollador responsable:</b> María Esther Suyón Farroñan	<b>Iteración:</b> 1
<b>Descripción:</b> Los usuarios podrán ingresar después de haber registrado su usuario y contraseña.	
<b>Observación:</b> El sistema validara usuario y contraseña sean correctas, sino enviara mensaje.	

TABLA VIII H2 GESTIONAR USUARIO

<b>Historia de usuario</b>	
<b>Numero: H2</b>	<b>Usuario:</b> Evaluador
<b>Nombre de historia:</b> Gestionar usuarios	<b>Prioridad:</b> 2
<b>Desarrollador responsable:</b> María Esther Suyon Farroñan	<b>Iteración:</b> 1
<b>Descripción:</b> los usuarios deben registrar sus datos solicitados (nombre, apellido, id usuario, correo, contraseña, código)	

**Observación:** El sistema validara si el usuario esta registrado, sino enviara un mensaje para que realice el registro.

TABLA IX H3 REGISTRAR LOTES

<b>Historia de usuario</b>	
<b>Numero: H3</b>	<b>Usuario:</b> evaluador
<b>Nombre de historia:</b> Registrar lotes	<b>Prioridad:2</b>
<b>Desarrollador responsable:</b> María Esther Suyon Farroñan	<b>Iteración:1</b>
<b>Descripción:</b> se registrará los lotes con su id_lote, código, y evaluación de dicha producción en la base de datos.	
<b>Observación:</b> el sistema devuelve el registro de lotes de la base de datos.	

TABLA X H4 REGISTRAR PUNTOS

<b>Historia de usuario</b>	
<b>Numero: H4</b>	<b>Usuario:</b> evaluador
<b>Nombre de historia:</b> Registrar puntos	<b>Prioridad:2</b>
<b>Desarrollador responsable:</b> María Esther Suyon Farroñan	<b>Iteración:1</b>
<b>Descripción:</b> se registrará los puntos que corresponden a cada lote en la base de datos.	
<b>Observación:</b> el sistema muestra los puntos registrados de cada lote.	

TABLA XI H5 TOMAR FOTOS

<b>Historia de usuario</b>	
<b>Numero: H5</b>	<b>Usuario:</b> evaluador
<b>Nombre de historia:</b> Tomar fotos	<b>Prioridad:1</b>
<b>Desarrollador responsable:</b> María Esther Suyon Farroñan	<b>Iteración:1</b>

<b>Descripción:</b> Los usuarios podrán tomar las fotos y cargar para que se realice el diagnóstico con el formato de archivo solicitado.
<b>Observación:</b> Si el formato del archivo no es válido el sistema enviara una alerta.

TABLA XII H6 GESTIONAR DIAGNOSTICO

<b>Historia de usuario</b>	
<b>Numero: H6</b>	<b>Usuario:</b> evaluador
<b>Nombre de historia:</b> Gestionar diagnostico	<b>Prioridad:1</b>
<b>Desarrollador responsable:</b> María Esther Suyón Farroñan	<b>Iteración:1</b>
<b>Descripción:</b> el sistema debe mostrar el resultado del reconocimiento de la plaga o enfermedad de la imagen predicha, con su respectiva fecha y hora que fue tomada.	
<b>Observación:</b> se diagnosticó con éxito en el sistema	

TABLA XIII H7 GESTIONAR RECOMENDACIONES

<b>Historia de usuario</b>	
<b>Numero: H7</b>	<b>Usuario:</b> evaluador
<b>Nombre de historia:</b> Gestionar recomendaciones	<b>Prioridad:3</b>
<b>Desarrollador responsable:</b> María Esther Suyon Farroñan	<b>Iteración:1</b>
<b>Descripción:</b> El sistema muestra las recomendaciones previo al diagnóstico.	
<b>Observación:</b> No tiene	

- **Definir tareas:** Para cada historia de usuario, se definieron las tareas necesarias para su implementación. Estas tareas se detallan en la Tabla XII.

TABLA XIV LISTA DE TAREAS POR HISTORIAS DE USUARIOS

<b>Numero de tarea</b>	<b>Numero de historia</b>	<b>Nombre de tarea</b>
<b>T1</b>	H1	Diseñar interfaz de inicio de sesión
<b>T2</b>	H2	Diseñar interfaz de registro de usuario

<b>T3</b>	H2	Registrar y guardar usuarios
<b>T4</b>	H3	Registrar lotes en la base de datos
<b>T5</b>	H3	Diseñar interfaz Generar lotes
<b>T6</b>	H4	Diseñar interfaz generar puntos por lote
<b>T7</b>	H5	Diseñar interfaz de tomar foto
<b>T8</b>	H5	Guardar fotos para diagnostico
<b>T9</b>	H6	Diseñar interfaz de diagnóstico
<b>T10</b>	H6	Gestionar diagnostico
<b>T11</b>	H6	Mostrar detalles de diagnóstico
<b>T12</b>	H7	Diseñar interfaz de reportes recomendaciones
<b>T13</b>	H5	Mostrar recomendaciones de predicción obtenida

- **Iteraciones** La lista de historia de usuario se dividió en 3 iteraciones para las entregas según muestra la Tabla XV de iteraciones

TABLA XV ITERACIONES

<b>Historias de usuario</b>	<b>Iteración 1</b>	<b>Iteración 2</b>	<b>Iteración 3</b>
Inicio de sesión	<b>x</b>		
Gestionar usuarios	<b>x</b>		
Registrar lotes		<b>x</b>	
Registrar puntos		<b>x</b>	
Tomar fotos		<b>x</b>	
Gestionar diagnostico		<b>x</b>	
Reporte recomendaciones			<b>x</b>

- **Plan de entregas:** En este punto, se elaboró un plan de entregas basado en las historias de usuario y sus tareas. Las fechas de entrega se asignaron de acuerdo con la Tabla XVI.

TABLA XIV PLAN DE ENTREGAS

Historia de usuario	Fecha inicio	Fecha fin	Prioridad
Inicio de sesión	02/10/2023	04/10/2023	1
Gestionar usuarios	05/10/2023	10/10/2023	2
Registrar lotes	11/10/2023	12/10/2023	2
Registrar puntos	13/10/2023	14/10/2023	2
Tomar fotos	15/10/2023	20/10/2023	1
Gestionar diagnostico	23/10/2023	03/11/2023	1
Reporte recomendaciones	06/11/2023	10/11/2023	3

### Fase de Diseño

En esta fase de la metodología XP, se define la arquitectura tecnológica, el diagrama de base de datos y se definen las interfases del sistema.

#### Arquitectura tecnológica

La arquitectura del proyecto se basó en un modelo cliente-servidor.

El cliente se desarrolló utilizando los frameworks Angular y Ionic, que permiten crear aplicaciones móviles. El servidor creó el modelo CNN utilizando las librerías TensorFlow y Keras y programado en lenguaje Python. Para la conexión con el cliente, se utilizó la API Flask. La base de datos se desarrolló en PostgreSQL, como se muestra en la figura (09).

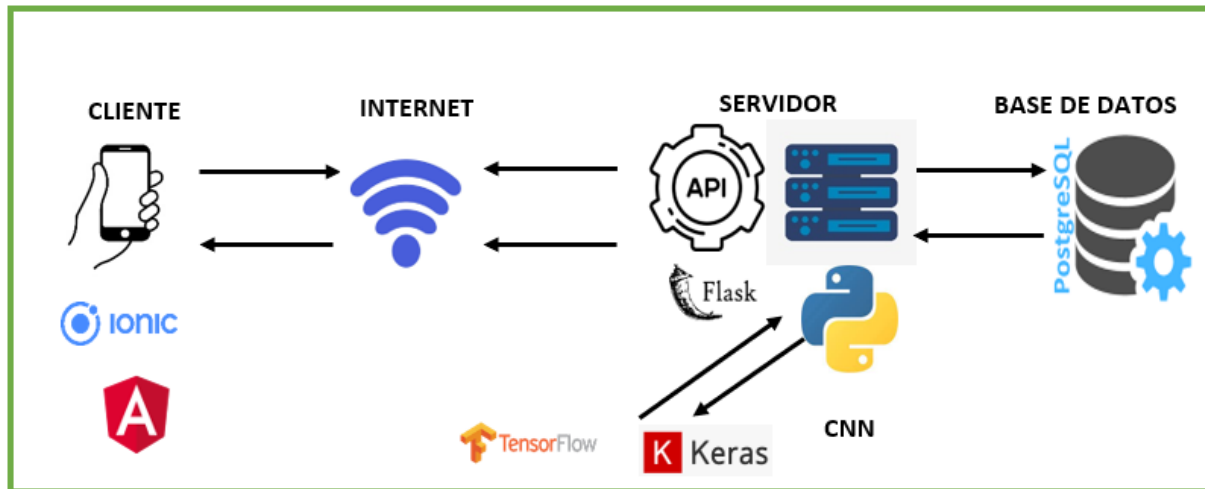


Fig. 09: Arquitectura tecnológica

- **Servidor** = almacena las fotos registradas por el sistema.
- **Base de datos** = PostgreSQL 6.1
- **Internet** = servicio de internet claro.
- **Python:** lenguaje de programación versión 3.11
- **Cliente** = encargado de realizar las evaluaciones tomando las fotos. Usará el sistema.

#### Hardware

- **Teléfono:** Modelo Samsung Galaxy con sistema operativo Android. Espacio de memoria 4GB
- Tablet, espacio de memoria desde 4GB hacia adelante.
- **Pc- Lenovo** con memoria RAM de 4GB hacia adelante, procesador Intel Core i5

- **Diagrama de base de datos.** La base de datos se muestra en la figura10 y tienen la siguiente estructura:

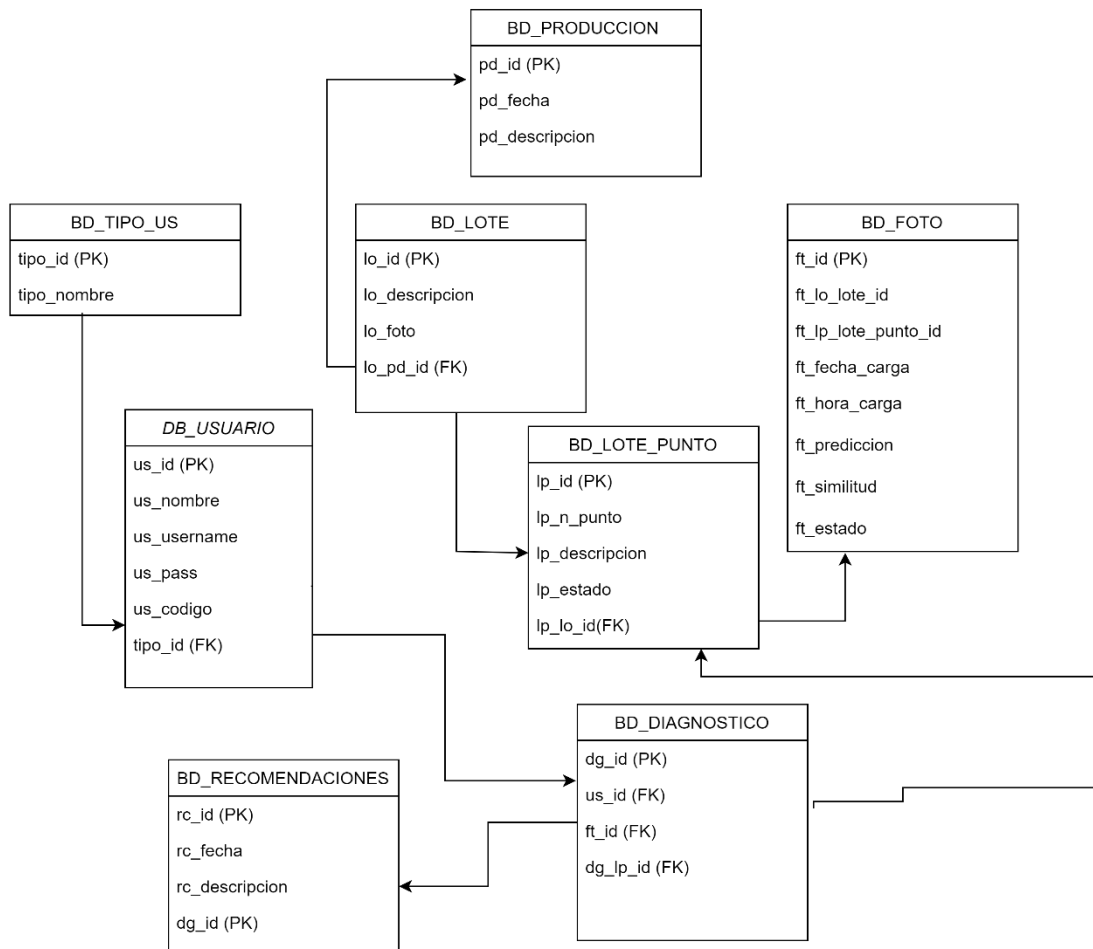


Fig. 10: Diagrama de Base de datos

## Diseño de interfaces

- **Interfaz inicio de sesión**

La siguiente interfaz muestra al usuario que debe ingresar sus datos de usuario y contraseña al dar clic en iniciar sesión accede al aplicativo como muestra la Figura11

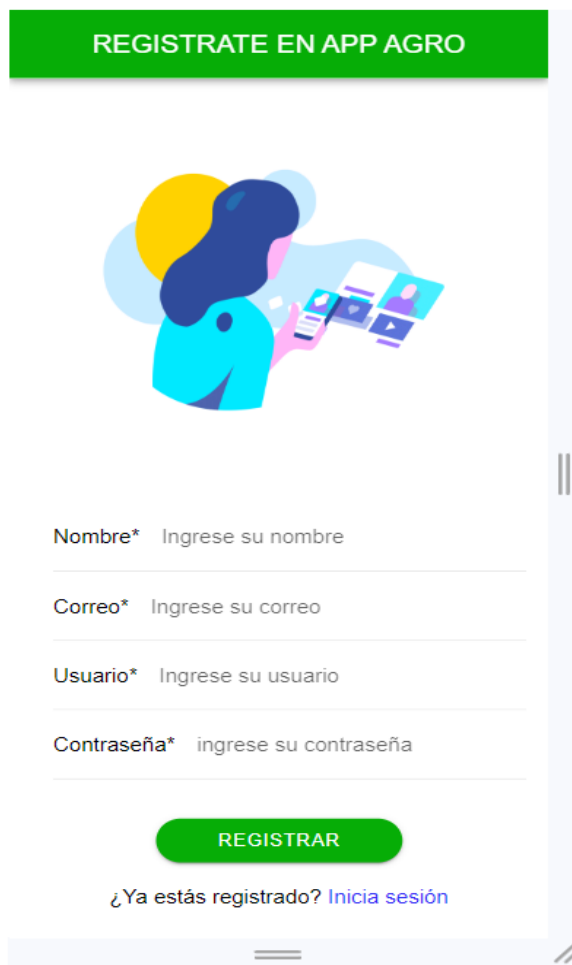
interfaz inicio de sesión.




Fig. 11: Interfaz inicio de sesión

- **Interfaz Registrar usuario**

La interfaz de la figura 12 muestra al usuario los campos que debe llenar para registrarse con su nombre, correo, usuario y contraseña luego clic en Registrar.



REGISTRATE EN APP AGRO



Nombre\* Ingrese su nombre

Correo\* Ingrese su correo

Usuario\* Ingrese su usuario

Contraseña\* ingrese su contraseña

REGISTRAR

¿Ya estás registrado? [Inicia sesión](#)

Fig. 12: Registrar usuario

- **Interfaz Gestionar Diagnostico**

En la siguiente interfaz Figura 13 muestra la fecha de la evaluación que se realiza con su respectiva producción, luego muestra lista lotes, donde el usuario podrá ingresar y generar los puntos del lote que elige figura 14 posteriormente, realiza la toma de 4 muestras figura 15, internamente se genera el diagnóstico y muestra resultados de predicción y similitud figura 16.



Fig.13: Interfaz lista de lotes

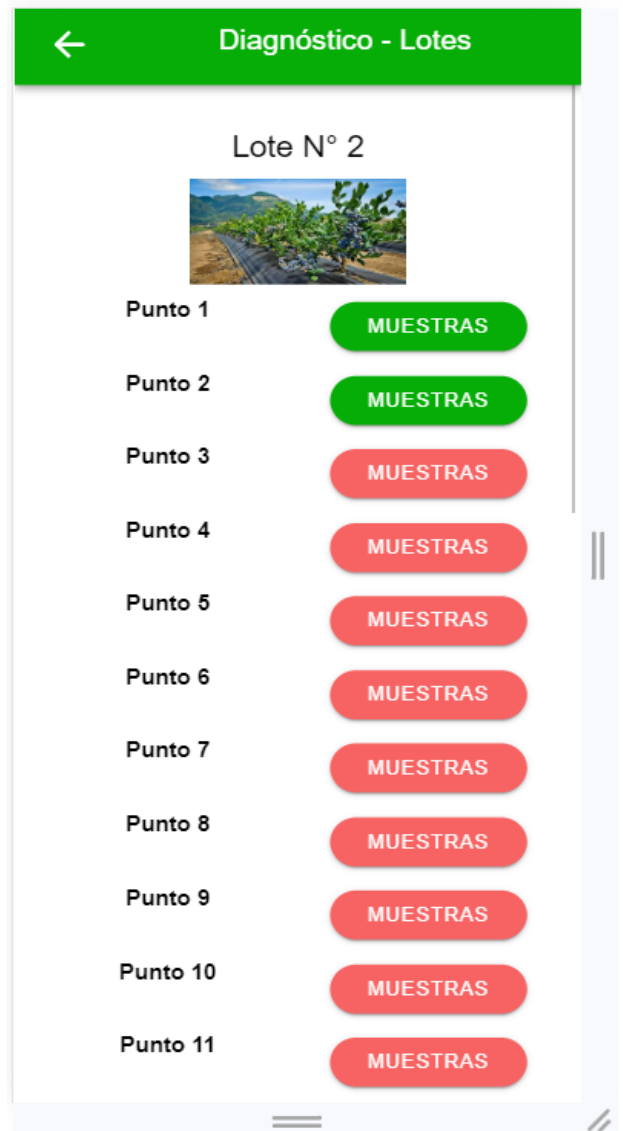


Fig. 14: Interfaz lista de puntos por lote

La interfaz lista de lotes (figura.13), mostrara los lotes registrados, y la interfaz lista de puntosfig.14, muestra los puntos que corresponde al lote que el usuario ha registrado que servirá para ingresar las imágenes a diagnosticar.

La interfaz mostrar fotos(figura.15), muestra las fotos registradas y la interfaz resultados (Figura.15) después de realizar el diagnostico muestra el resultado con fecha y hora de diagnóstico del lote evaluado, donde se visualiza el porcentaje de afectación que le corresponde a cada clase predicha.



Fig.15: Interfaz mostrar fotos



Fig.16: Interfaz resultados

### Fase de Codificación

Se inicio el desarrollo del sistema con la instalación de la CLI de angular, primero se descargó node.js de la página <https://nodejs.org/es/>

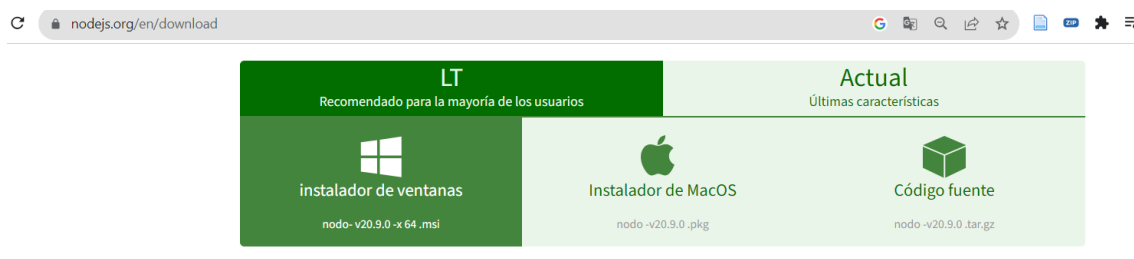


Fig. 17 Descarga de nodej.js

Una vez descargado e instalado node.js se procede a instalar el framework Ionic con el comando **npm install -g @ionic/cli**, el cual permite ejecutar la creación de la aplicación como muestra la figura (19)

```
C:\Users\LENOVO>
C:\Users\LENOVO>npm install -g @ionic/cli
npm WARN deprecated vm2@3.9.19: The library co
```

Fig. 18: Instalación de framework Ionic

Enseguida se crea el proyecto en una carpeta con su respectivo nombre, utilizando el comando **ionic start app-arándanos**, y pide seleccionar el framework con el que se trabajara que es angular, se ejecuta y se esperó que se instale correctamente figura (20).

```
E:\>ionic start app-arandanos

Pick a framework!

Please select the JavaScript framework to use for your new app.
--type option.

? Framework: (Use arrow keys)
> Angular | https://angular.io
  React   | https://reactjs.org
  Vue     | https://vuejs.org
```

Figura 19: crear proyecto app-arándanos

Posteriormente, se abre el proyecto en visual estudio code donde se hizo todo el desarrollo de la aplicación, en la siguiente figura 20 se detalla la estructura de un proyecto en angular

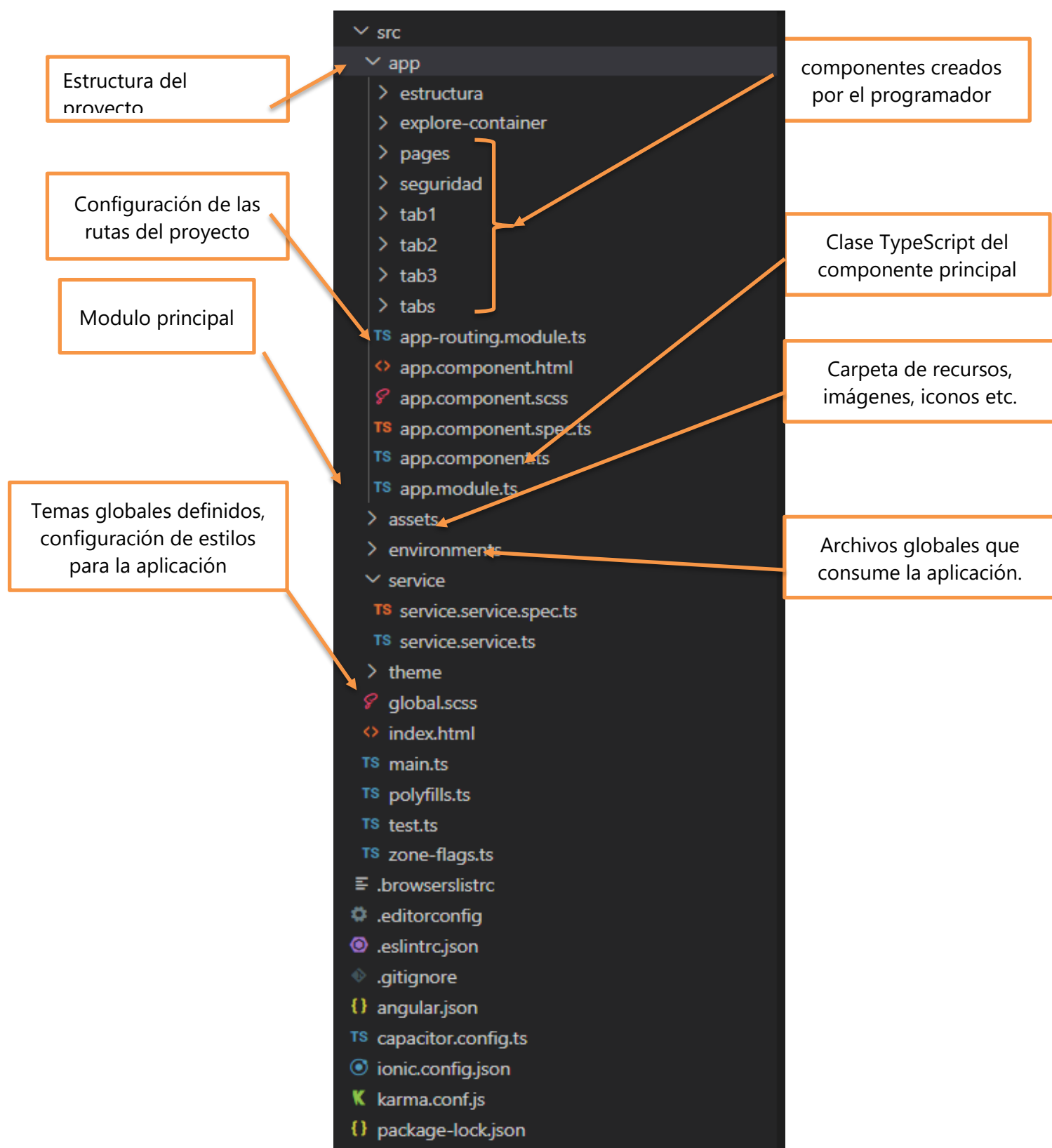


Fig. 20: Estructura de un proyecto en Angular

Se procede con la codificación para iniciar sesión, se ingresa usuario y contraseña, si los datos no son correctos enviara alerta que ingrese los datos correctos. Si esta correcto accede al sistema figura 21.

```
login(usuario: Usuario){
  console.log("usuarios", usuario)
  console.log("usuario =>", this.login_clase);
  this.login_clase.us_username = usuario.usua_usuario.trim();
  this.login_clase.us_password = usuario.usua_password.trim();
  if(this.login_clase.us_username.length == 0 && this.login_clase.us_password.length == 0){
    this.presentToast('danger','Ingrese usuario y contraseña correctamente','alert');
  }else{
    this.showLoading();
    setTimeout(() => {
      this.servicio?.login(this.login_clase).subscribe( resp => {
        this.respuesta = resp;
        this.cerrarLoading();
        console.log("resp => ", resp)
        if(this.respuesta.message == 0){
          this.router?.navigate(["inicio/ diagnostico"]);
          this.presentToast('success','Acceso correcto','checkmark-outline');
        }else{
          this.presentToast('danger','Usuario o contraseña incorrecta','alert');
        }
      },_error => {
        this.servicio!.toastDatos('danger','Ocurrio un error al iniciar sesión'+_error,'alert');
        this.cerrarLoading();
      });
    }, 1000);
  }
}
```

Fig. 21 Código inicio sesión

Para el registro de un usuario se desarrolló como muestra la figura 22 ingresando los siguientes datos: nombre, usuario, contraseña y correo.

```
registrar(){
  console.log("registra 000 => ", this.myFormUsuario);
  if(this.myFormUsuario.status == "INVALID"){
    console.log("sin resultado");
    this.servicio.toastDatos('danger','Complete los datos para continuar','alert');
  }else{
    this.usuario.nombre = this.myFormUsuario.value.nombre;
    this.usuario.username = this.myFormUsuario.value.nombre;
    this.usuario.email = this.myFormUsuario.value.correo;
    this.usuario.password = this.myFormUsuario.value.password;
    this.usuario.rol = 'Admin';
    console.log("usuarios => ", this.usuario)
    this.servicio.loading('Registrando...');
    setTimeout(() => {
      this.servicio.registrarUsuario(this.usuario).subscribe( (respuesta:any) => {
        console.log("registrado", respuesta);
        if(respuesta.est == 0) {
          this.servicio.toastDatos('success',respuesta.message,'checkmark-outline');
          setTimeout(() => {
            this.myFormUsuario.reset();
            this.router.navigate([""]);
          }, 2000);
        }
        this.servicio.cerrarLoading();
      },_error => {
        this.servicio.cerrarLoading();
      });
    });
  }
}
```

Fig. 22 Código inicio sesión

En tanto, por el lado del servidor comienza importando las librerías necesarias, incluidas la Api Flask que conecta el cliente con el servidor, SQLAlchemy y el CORS que sirve para conectar la base de datos e intercambien información ambas tecnologías. La base de datos se crea utilizando PostgreSQL y se llama "bd\_arandanos", se hace la configuración de conexión ingresado una contraseña y el nombre de la base de datos figura23

```

1  from flask import Flask
2  from flask_sqlalchemy import SQLAlchemy
3  from flask_marshmallow import Marshmallow
4  from flask import Flask, request, jsonify
5  from sqlalchemy import Text
6  from flask_cors import CORS
7  from sqlalchemy import ForeignKey
8  from flask import request
9  import os
10 import base64
11 import psycopg2 # Importa la librería psycopg2
12
13 app = Flask(__name__)
14
15 # Inicializa la extensión CORS
16 cors = CORS(app, resources={
17     r"/api/v1/*": {
18         "origins": ["*"],
19         "methods": "GET, POST, OPTIONS",
20         "allow_headers": "Authorization, Content-Type"
21     }
22 })
23 cors = CORS(app) #intercambio de recursos de origen cruzado (FrontEnd al Backend)
24
25 # Configura la conexión a la base de datos PostgreSQL
26 USUARIO = 'postgres'
27 PASS = '76812988'
28 PORT = "5432"
29 DB = 'bd_arandanos'
30 app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://'+USUARIO+':'+PASS+'@localhost:'+PORT+'/'+DB
31
32
33 # Inicializa la extensión SQLAlchemy
34 db = SQLAlchemy(app)
35
36 if __name__ == "__main__":
37     app.run(debug=True)
38     app.run(host="192.168.18.38", port=5400)
39

```

Fig.23 Intercambio de recursos

Posteriormente, se muestra cómo funciona la Api para recibir las fotos y generar el diagnóstico de la plaga y la enfermedad Figura 24. Primero se crea un carpeta llamada UPLOAD\_FOLDER que sirve para almacenar las imágenes que se usaran para el reconocimiento, para ello se recibe el Id del punto y el Id del lote que corresponde a esa evaluación con su respectiva fecha, y las 4 fotos, se verifica que la carpeta exista y pasa a decodificar las imágenes en base 64 para que sea un archivo jpg, seguidamente, se itera el bucle for y carga la imagen usando la función load\_img(). El parámetro target\_size especifica que la imagen debe redimensionarse a 150x150

píxeles y luego se normaliza la matriz de imagen dividiendo cada valor de píxel por 255.0. En resumen, este código clasifica las imágenes en dos categorías: alternaria y mosca blanca. Para ello, utiliza un modelo de aprendizaje automático previamente entrenado. El modelo se entrena en un conjunto de datos de imágenes de alternaria y mosca blanca, una vez que se ejecuta el bucle, la lista registros\_fotos contendrá un registro para cada predicción. Estos registros se almacenan en la base de datos.

```
@app.route('/api/v1/recibir_fotos', methods=['POST'])
def recibir_fotos():
    datos = request.get_json()

    if isinstance(datos, dict) and 'lote_id' in datos:
        lote_id = datos['lote_id']
        punto_id = datos.get('punto_id', 0)
        punto_generado = datos.get('punto_generado', 0)
        fecha = datos.get('fecha', '')
        hora = datos.get('hora', '')
        fotos = datos.get('fotos', [])
        foto_1 = fotos[0].get('foto_1', '')
        foto_2 = fotos[0].get('foto_2', '')
        foto_3 = fotos[0].get('foto_3', '')
        foto_4 = fotos[0].get('foto_4', '')

        horaformato = hora.replace(':', '-')

        # Construye el nombre de la carpeta
        nombre_dir = 'lote-' + str(lote_id) + 'punto-' + str(punto_generado) + '-' + fecha + horaformato
        ruta_completa = os.path.join(UPLOAD_FOLDER, nombre_dir)

        # Verifica si la carpeta ya existe o créala si no existe
        if not os.path.exists(ruta_completa):
            os.makedirs(ruta_completa)

            for i, foto in enumerate([foto_1, foto_2, foto_3, foto_4]):
                if foto:
                    with open(os.path.join(ruta_completa, f'imagen_{i+1}_lote{str(lote_id)}_punto{str(punto_generado)}.jpg'), 'wb') as archivo:
                        archivo.write(base64.b64decode(foto))

            model_path = 'modelo_entrenadoApp.h5'
            model = load_model(model_path)

        # Ruta a la carpeta con las imágenes que deseas predecir
        image_folder_path = ruta_completa
        predictions = []

        image_files = [f for f in os.listdir(image_folder_path) if f.endswith('.jpg')]

        for image_file in image_files:
            image_path = os.path.join(image_folder_path, image_file)
            image = load_img(image_path, target_size=(150, 150))
            image_array = img_to_array(image)
            image_array = np.expand_dims(image_array, axis=0)
            image_array /= 255.0

            prediction = model.predict(image_array)
            predicted_class = np.argmax(prediction)
            class_names = {0: 'alternaria', 1: 'moscablanca'}
            predicted_class_name = class_names[predicted_class]
            probability = prediction[0][predicted_class] * 100

            predictions.append((image_file, predicted_class_name, probability))

        # Crear una lista de registros de fotos con sus predicciones y similitudes
        registros_fotos = []
        for image_file, predicted_class_name, probability in predictions:
            # Reformatea la hora reemplazando guiones bajos con dos puntos
            hora_carga = horaformato.replace('_', ':')
            registro_foto = bd_foto(
                ft_lo_lote_id=lote_id,
                ft_lote_punto_num=punto_id,
                ft_fecha_carga=fecha,
                ft_hora_carga=hora_carga, # Usar la hora reformateada
                ft_prediccion=predicted_class_name,
                ft_similitud=probability,
                ft_lp_lote_punto_id=punto_generado,
                ft_estado=1
            )
            registros_fotos.append(registro_foto)

        # Agregar cada registro a la sesión de la base de datos
        for registro in registros_fotos:
            db.session.add(registro)
```

Fig.24: Código de api recibir foto

Agrega cada registro de foto a la sesión de la base de datos y actualiza el estado del punto de muestreo a "guardado" luego confirma la transacción de la base de datos y finalmente devuelve una lista que contienen las predicciones figura 25.

```
#actualizo el estado de lote_punto
session = db.session

session.query(db_lote_punto).filter_by(lp_n_punto=punto_id, lp_estado=0, lp_lo_id=lote_id).update({'lp_estado': 2}) # guardado

# Confirmar la transacción de la base de datos
db.session.commit()

# Devolver todas las predicciones
response = [{'image_file': pred[0], 'predicted_class_name': pred[1], 'probability': pred[2]} for pred in predictions]

return jsonify({'predictions': response})

else:
return jsonify({'error': 'Error con el JSON'})
```

Fig.25 devolver datos predichas

## Fase de Prueba

### - Pruebas de caja blanca

Prueba N°1 inicio de sesión

TABLA XVIII

CASO DE PRUEBA DE CAJA BLANCA N°1

Numero de prueba	PCB1
<b>Título del caso de prueba</b>	Validar el inicio de sesión.
<b>Resumen de la prueba</b>	El propósito de esta prueba es verificar que el inicio de sesión se dé de forma correcta, para ello se realizarán pasos por medio de técnicas del camino básico para verificar si los datos ingresados son válidos para que el usuario pueda acceder
<b>Pasos</b>	<b>Resultados</b>
1. Acceso	Permitir el acceso al usuario por medio de su autenticación de usuario y contraseña.

## Código para validar inicio de sesión

```
login(usuario: Usuario){
  console.log("usuarios", usuario)
  console.log("usuario =>", this.login_clase);
  this.login_clase.us_username = usuario.usua_usuario.trim();
  this.login_clase.us_password = usuario.usua_password.trim();
  1 if(this.login_clase.us_username.length == 0 && this.login_clase.us_password.length == 0){
  2   this.presentToast('danger', 'Ingrese usuario y contraseña correctamente', 'alert');
  }else{
  3   this.showLoading();
     setTimeout(() => {
     this.servicio?.login(this.login_clase).subscribe( resp => {
       this.respuesta = resp;
       this.cerrarLoading();
       console.log("resp => ", resp)
       if(this.respuesta.message == 0){
         4   this.router?.navigate(["inicio/diagnostico"]);
           this.presentToast('success', 'Acceso correcto', 'checkmark-outline');
         }else{
           this.presentToast('danger', 'Usuario o contraseña incorrecta', 'alert');
         }
       }, _error => {
         5   this.servicio!.toastDatos('danger', 'Ocurrio un error al iniciar sesión'+_error, 'alert');
           this.cerrarLoading();
         });
       });
     }, 1000);
   }
}
```

Fig.23 código iniciar sesión

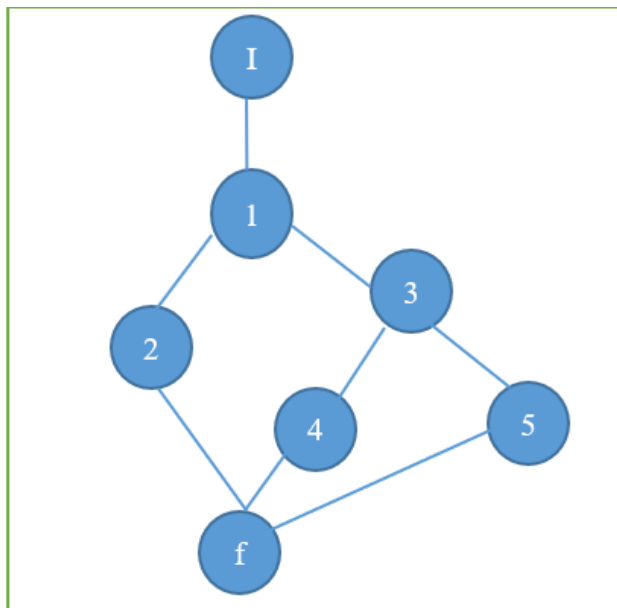
**Grafo de flujo**

TABLA XIX  
EVALUACIÓN DE PRUEBA N°1

Numero	Camino independiente	Resultado esperado
1	I, 1, 2, F	“Ingreso usuario y contraseña correctamente”
2	I, 1, 2,3,4, F	“Acceso correcto”
3	I, 1, 2, 3, 5, F	“Usuario o contraseña incorrecta”

Prueba N° 2 enviar correo

TABLA XX  
CASO DE PRUEBA DE CAJA BLANCA N°2

Numero de prueba	PCB2
<b>Título del caso de prueba</b>	Validar enviar correo
<b>Resumen de la prueba</b>	El objetivo de esta prueba es verificar que se envié un correo con el código que servirá para cambiar contraseña para iniciar sesión
<b>Pasos</b>	<b>Resultados</b>
1. Enviar correo	Se ejecuta correctamente si el correo es valido

	Sino muestra una alerta si el correo es invalido
2.	

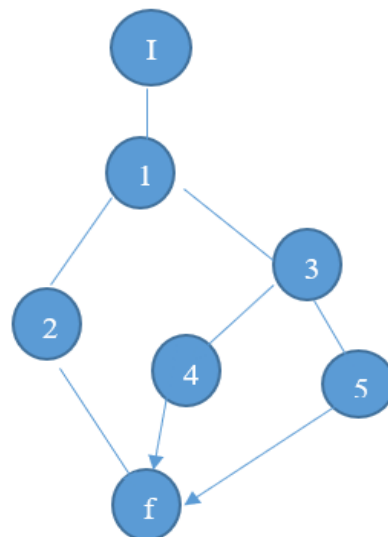
### Código enviar correo

```

envioCorreo(){
  this.correo.correo_destino = this.myFormClave.value.correo;
  this.servicio.loading('Enviando correo...');
  this.servicio.enviarCorreo(this.correo).subscribe( (respuesta:any) => {
    console.log("envio el correo", respuesta);
    if(respuesta.error){
      this.servicio.toastDatos('danger',respuesta.error, 'alert');
      this.servicio.cerrarLoading();
    }else{
      this.cambiar_clave = false;
      this.codigo = true;
      this.envio_correo = false;
      this.servicio.toastDatos('success',respuesta.mensaje, 'checkmark-outline');
      this.servicio.cerrarLoading();
    }
  })
}

```

### Grafo de flujo



TABBLA XXI

### EVALUACIÓN DE PRUEBA N°2

Numero	Camino independiente	Resultado esperado
1	I, 1, 2, F	“envió correo”
2	I, 1,3,4, F	“error al enviar correo”
3	I, 1, 2, 3, 5, F	“Verifica datos”

## Prueba N° 3 cambiar contraseña

TABLA XXII  
CASO DE PRUEBA DE CAJA BLANCA N°3

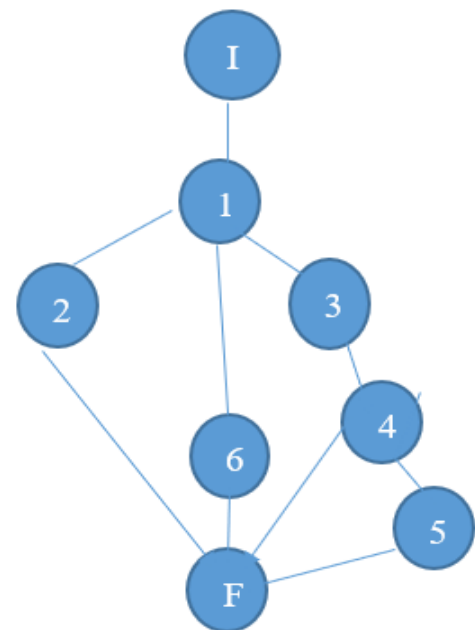
Numero de prueba	PCB3
Título del caso de prueba	Cambiar contraseña
Resumen de la prueba	El objetivo de esta prueba es verificar que el cambio de contraseña es el correcto
Pasos	Resultados
1. Interfaz de cambiar contraseña	Se ejecuta correctamente si la contraseña coincide
2. Ingresar datos de contraseña	Sino muestra una alerta

```

cambiarContra() {
  this.correo.email = this.myFormClave.value.correo;
  this.correo.new_pass = this.myFormClave.value.clave_uno;
  1 if(this.myFormClave.value.clave_uno == this.myFormClave.value.clave_dos){
    this.servicio.loading('Cambiando contraseña...'); 2
    setTimeout(() => {
      this.servicio.cambiar_pass(this.correo).subscribe( (respuesta:any) => {
        console.log("envio el correo", respuesta);
        3 if(respuesta.error){
          this.servicio.toastDatos('danger', respuesta.error, 'alert'); 4
          this.servicio.cerrarLoading();
        }else{ 5
          this.envio_correo = false;
          this.cambiar_clave = true;
          this.codigo = false;
          this.alertaModal(respuesta.success);
          this.servicio.cerrarLoading();
        }
      })
    }, 2000);
  }else{ 6
    this.servicio.toastDatos('danger', 'Contraseñas no coinciden', 'alert');
  }
}

```

Código cambiar contraseña



Grafo de flujo

TABLA XXIII  
EVALUACIÓN DE PRUEBA CAMBIAR CONTRASEÑA

<b>Numero</b>	<b>Camino independiente</b>	<b>Resultado esperado</b>
<b>1</b>	I, 1, 2, F	<b>“Cambiando contraseña”</b>
<b>2</b>	I,1,6, F	<b>“contraseñas no coinciden”</b>
<b>3</b>	I, 1,3,4, F	<b>“envía mensaje de alerta”</b>
<b>4</b>	I, 1, 2, 3, 5, F	<b>“Verifica datos”</b>

#### **Prueba N°4 creación de producción**

<b>Numero de prueba</b>	<b>PCB4</b>
<b>Título del caso de prueba</b>	Validar creación de producción
<b>Resumen de la prueba</b>	El objetivo de esta prueba es validar el registro de producción
<b>Pasos</b>	<b>Resultados</b>
1. Envía fecha y descripción	Nuevo registro de producción Se obtiene el id de producción Se crea los 25 lotes Registro exitoso

## código crear producción

```

@app.route('/api/v1/creo_produccion_lote', methods=['POST'])
def creo_produccion_lote():
    try:
        if request.method == 'POST':
            data = request.json
            pd_fecha = data.get('pd_fecha')
            pd_descripcion = data.get('pd_descripcion')

            nuevo_registro = bd_produccion(
                pd_fecha=pd_fecha,
                pd_descripcion=pd_descripcion
            )
            db.session.add(nuevo_registro)
            db.session.commit()

            # Obtengo el ID recién generado
            pd_id = nuevo_registro.pd_id

            db.session.query(bd_lote).filter_by(lo_estado_prod=0).update({'lo_estado_prod': 1})

            for i in range(1, 25): #creo los lotes
                registro_lote = bd_lote(
                    lo_descripcion='Lote Nº '+str(i),
                    lo_foto='../assets/img/ing-lote02.jpeg',
                    lo_estado=0,
                    lo_estado_prod=0,
                    lo_estado_res=0,
                    lo_pd_id=pd_id
                )

                db.session.add(registro_lote)
                db.session.commit()

            return jsonify({'message': 'Registro exitoso', 'est': 'ok'})

    except Exception as e:
        return jsonify({'message': str(e), 'est': 'error'})

```

## Grafo de flujo

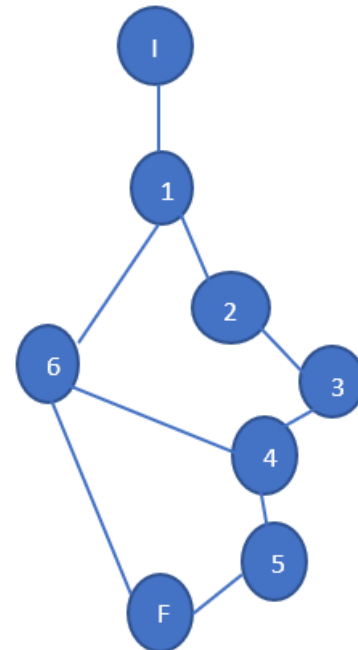


TABLA XXIII

### EVALUACIÓN DE PRUEBA CREACIÓN DE PRODUCCIÓN LOTE

Numero	Camino independiente	Resultado esperado
1	I, 1, 2,3, 4, 5, F	“Registro exitoso”
2	I,1,6, F	“Mensaje de error”
3	I,1, 2, 3, 4, 6, F	“Mensaje de error”

## Prueba de caja negra

Prueba N°1 análisis de predicción de imágenes: se realizó la prueba que corresponde al resultado final del diagnóstico de la plaga o enfermedad ingresada por el usuario

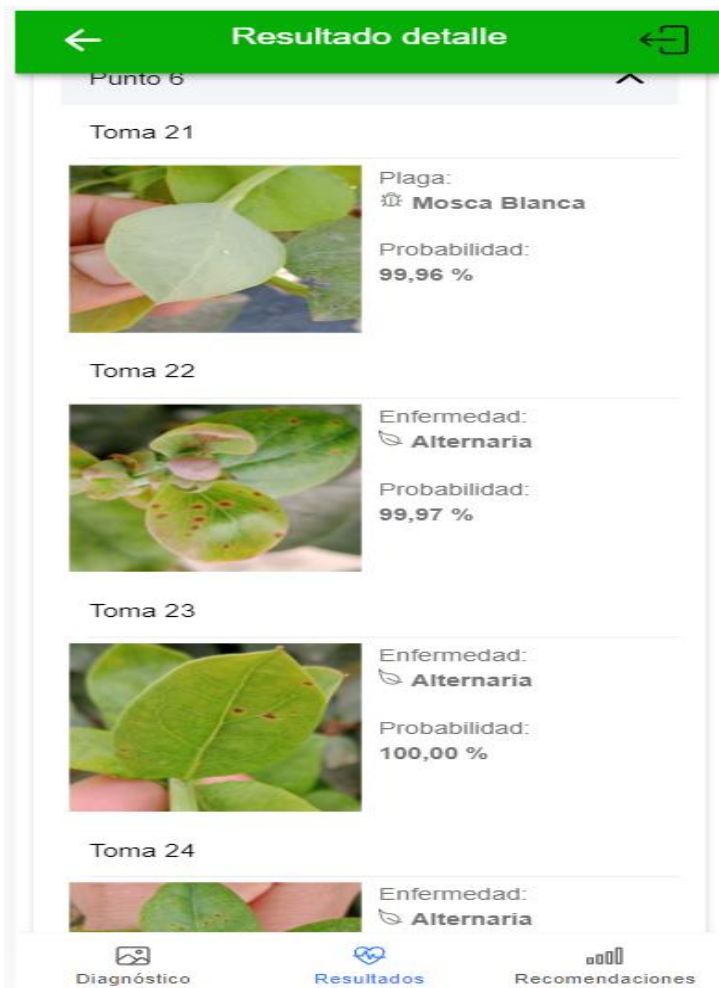


Fig.24 Interfaz diagnóstico con fotos

TABLA XXIV  
CASO DE PRUEBA DE CAJA NEGRA N°1

<b>Realizar análisis de diagnóstico</b>	<b>CPCN01</b>
<b>Escenario:</b> Prueba para evaluar interfaz de la predicción de las imágenes	
<b>Desarrollador:</b> María Esther Suyón Farroñan	Fecha: 15/11/2023
Pre-condiciones:	<ul style="list-style-type: none"> <li>• El usuario inicia sesión</li> <li>• Registra lotes</li> <li>• Registra puntos de cada lote</li> </ul>
Datos de entrada	<ul style="list-style-type: none"> <li>• Ingresa 4 fotos de hojas de arándanos</li> </ul>
Descripción de pasos	<ul style="list-style-type: none"> <li>• El usuario abre interfaz de lotes_puntos</li> <li>• Clic en botón muestras</li> <li>• Ingresa muestras (fotos)</li> <li>• Registra muestras</li> <li>• Guarda diagnóstico</li> </ul>
Resultado esperado	<ul style="list-style-type: none"> <li>• Al realizar el diagnóstico muestra resultados correctos.</li> </ul>
Resultado obtenido	<ul style="list-style-type: none"> <li>• Muestra fotos con su respectivo diagnóstico</li> <li>• Clase predicha y porcentaje % de similitud</li> </ul>

Prueba N°2 interfaz de iniciar sesión



TABLA XXV

## PRUEBA DE CAJA NEGRA N°2 INICIAR SESIÓN

Iniciar sesión	CPCN02
<b>Escenario:</b> Prueba para evaluar la interfaz de iniciar sesión	
<b>Desarrollador:</b> María Esther Suyón Farroñan	Fecha: 15/11/2023
Pre-condiciones:	<ul style="list-style-type: none"> <li>• El usuario se registro</li> </ul>
Datos de entrada	<ul style="list-style-type: none"> <li>• Ingresar usuario y contraseña</li> </ul>
Descripción de pasos	<ul style="list-style-type: none"> <li>• Abre interfaz iniciar sesión</li> <li>• Ingresar usuario</li> <li>• Ingresar contraseña</li> <li>• Clic en botón iniciar sesión</li> <li>• Datos correctos</li> <li>• Usuario incorrecto, contraseña correcta</li> <li>• Usuario correcto, contraseña incorrecta,</li> </ul>
Resultado esperado	<ul style="list-style-type: none"> <li>• inicia sesión</li> <li>• muestra mensaje (<b>usuario y contraseña incorrecta</b>)</li> <li>• mensaje (<b>usuario y contraseña</b></li> </ul>

	<b>incorrecta)</b>
--	--------------------

Prueba N°3 Validar fotos




TABLA XXVI

PRUEBA DE CAJA NEGRA N°3 VALIDAR FOTOS

<b>VALIDAR</b>	<b>CPCN03</b>
<b>Escenario:</b> Prueba para evaluar interfaz de registrar fotos	
<b>Desarrollador:</b> María Esther Suyón Farroñan	Fecha: 15/11/2023
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• Lotes registrados</li> <li>• Puntos registrados</li> </ul>
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Ingresa 4 fotos</li> </ul>
<b>Descripción de pasos</b>	<ul style="list-style-type: none"> <li>• Ingresa fotos en formato JPG, PNG,</li> <li>• Ingresa datos formato pdf, doc.</li> <li>• Clic en botón registrar fotos</li> </ul>
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>• Fotos registradas</li> <li>• mensaje ocurrió error al registrar fotos.</li> </ul>

## Prueba N°4 registrar usuario

**REGISTRATE EN APP AGRO**



Nombre\* Ingrese su nombre

Correo\* Ingrese su correo

Usuario\* Ingrese su usuario

Contraseña\* ingrese su contraseña

**REGISTRAR**

[¿Ya estás registrado? Inicia sesión](#)

TABLA XXVII

## PRUEBA DE CAJA NEGRA N°4 REGISTRAR USUARIO

Registrar usuario	CPCN04
<b>Escenario:</b> Prueba para evaluar interfaz de registrar usuario	
<b>Desarrollador:</b> María Esther Suyón Farroñan	Fecha: 15/11/2023
Pre-condiciones:	Ninguno
Datos de entrada	<ul style="list-style-type: none"> <li>• Ingresa nombre</li> <li>• Ingresa correo</li> <li>• Ingresa usuario</li> <li>• Ingresa contraseña</li> </ul>
Descripción de pasos	1. Ingresa un nombre, correo, nombre de usuario y contraseña válidos.

	<ol style="list-style-type: none"> <li>2. Ingrese un nombre vacío.</li> <li>3. Ingrese un correo electrónico inválido.</li> <li>4. Ingrese un nombre de usuario que ya esté en uso.</li> <li>5. Ingrese una contraseña que no cumpla con los requisitos de longitud o complejidad.</li> </ol>
Resultado esperado	<ol style="list-style-type: none"> <li>1. Usuario registrado con éxito.</li> <li>2. La interfaz muestra un mensaje de error que campo de nombre está vacío.</li> <li>3. La interfaz genera mensaje de alerta que el correo electrónico no es válido.</li> <li>4. La interfaz genera un mensaje de alerta que el nombre de usuario ya se usó.</li> <li>5. La interfaz genera alerta que indica que la contraseña no cumple con los requisitos</li> </ol>

### En base a los objetivos:

#### **Crear un repositorio de imágenes afectadas del cultivo de arándano para la detección de las plagas y enfermedades estacionales del distrito de olmos.**

Para el cumplimiento del primer objetivo de este proyecto, se desarrolló un modelo computacional para la detección de plagas y enfermedades en plantas de arándanos, se realizó la creación de un repositorio de imágenes. En las primeras fases de la metodología de aprendizaje automático (ML), se realizó el análisis de la realidad problemática de la empresa, para conocer cómo se lleva el proceso de control de plagas y enfermedades. Posteriormente, se recolectaron 5500 imágenes afectadas por cada clase como: plaga “mosca blanca”, enfermedad “alternaria”, “hoja sana” y clase “otros” que se almacenaron en una carpeta llamada data\_set. En la fase 3, se analizaron las imágenes para eliminar aquellas que tenían baja calidad, ya que podrían afectar al entrenamiento del modelo computacional. Finalmente, se seleccionaron 5325 imágenes, que se dividieron en 70% para entrenamiento, 15% para validación y 15% para pruebas[36].

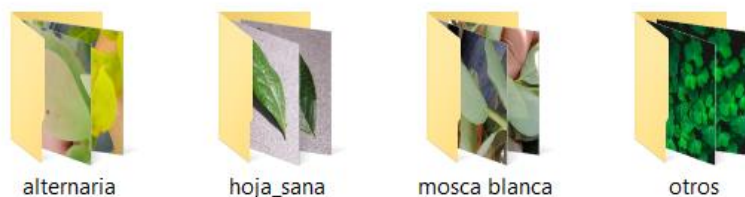


Fig. 27 Data\_set

### Desarrollar un modelo de sistema inteligente basado en redes neuronales convoluciones para la identificación de plagas y enfermedades estacionales del cultivo de arándano.

El segundo objetivo se cumplió, en la fase 4 Modelamiento donde se desarrolló el modelo de redes neuronales convolucionales (CNN), para la identificación de plagas y enfermedades en hojas de arándanos. Para ello, se utilizaron las librerías TensorFlow y Keras. El modelo CNN se desarrolló con capas convolucionales, capas de maxpooling, capas de aplanamiento, softmax y capas totalmente conectadas. Posteriormente, se compiló el modelo con el optimizador Adam y se realizó el entrenamiento con 40 épocas. Finalmente, se guardó el modelo en un archivo con extensión h5. En conclusión, los resultados del entrenamiento mostraron que el modelo tiene un alto nivel de reconocimiento de plagas y enfermedades, con una precisión del 95,3 % en el conjunto de entrenamiento y del 94,5 % en el conjunto de validación (Fase 5. Entrenar el modelo).

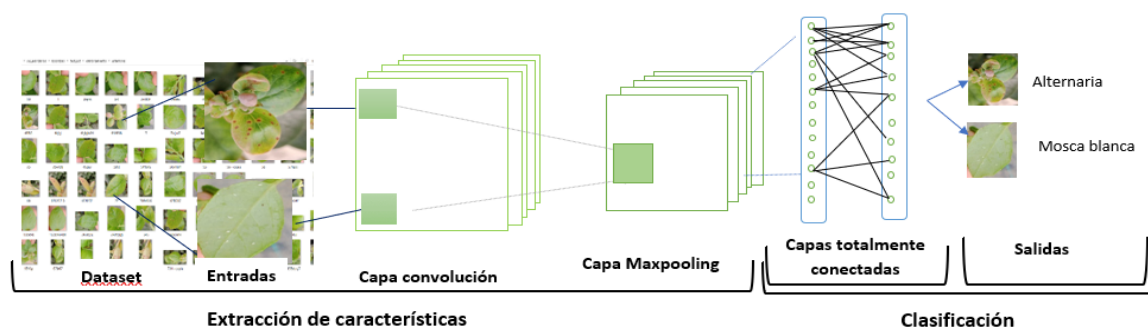


Fig. 28: Modelo CNN

### Diseñar e implementar el aplicativo móvil incorporando el modelo de ayuda en la detección de plagas y enfermedades estacionales del cultivo de arándano

Para el cumplimiento del tercer objetivo se desarrolló la aplicación móvil para la detección de plagas y enfermedades del cultivo de arándano, incorporando el modelo de redes neuronales

convolucionales. Para ello se utilizó las fases de la Metodología XP que ayuda a obtener una aplicación interactiva, el usuario ingresa las fotografías, en los puntos establecidos, de los lotes seleccionados, muestra la fecha de reconocimiento y el sistema realiza el diagnóstico, envía los datos a la base de datos y lista el detalle de la evaluación de los lotes con sus respectivos puntos, mostrando el porcentaje del nivel de infestación de la plaga y enfermedad diagnosticada figura 29 y 30.

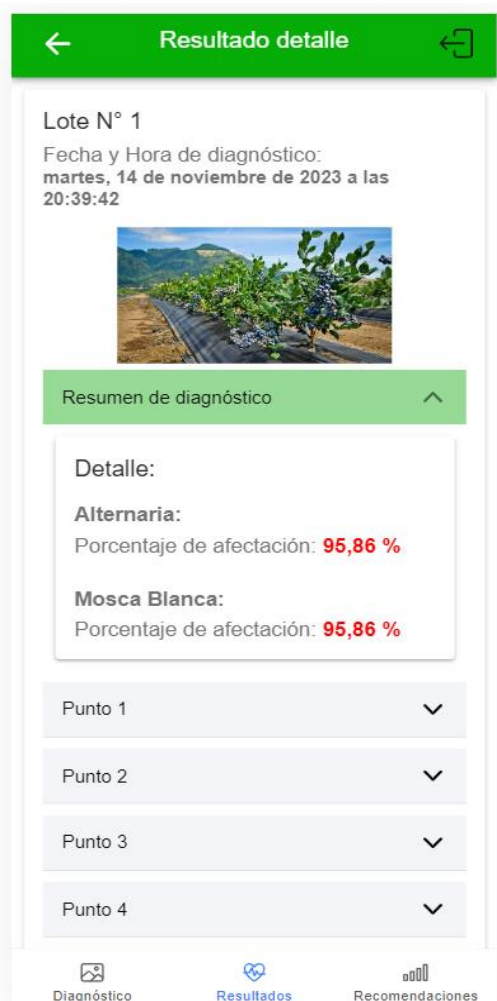


Fig.29 Interfaz de resultado diagnóstico

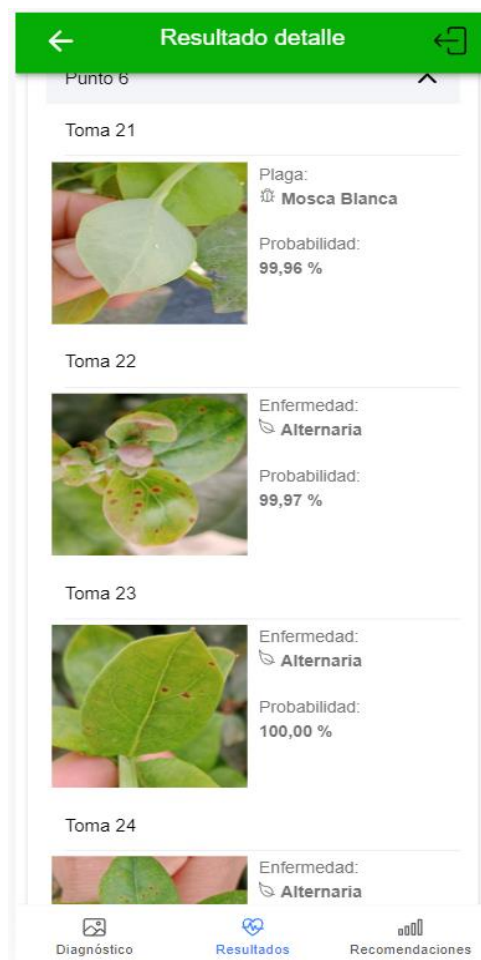


Fig.30 Interfaz diagnóstico con fotos

### Validar la fiabilidad y desempeño de la aplicación móvil para la detección de plagas y enfermedades estacionales por parte de los usuarios

Para el cumplimiento del cuarto objetivo se realizaron pruebas de campo con el técnico agrónomo especialista y 2 evaluadores de plagas y enfermedades del cultivo de arándanos,

ingresando imágenes al sistema, posterior a ello se aplicó el modelo de aceptación tecnológica (TAM) donde el usuario puede evaluar si la aplicación es útil y fiable cuando está usando (ANEXO 03). Para ello se usó 9 preguntas de las cuales se les asignó valor de 1 a 5 con su respectiva valoración.

TABLA XXVII: VALORACION

Quintil	Valoración de utilidad
Q1	Nada útil
Q2	Poco útil
Q3	Indiferente
Q4	útil
Q5	Muy útil

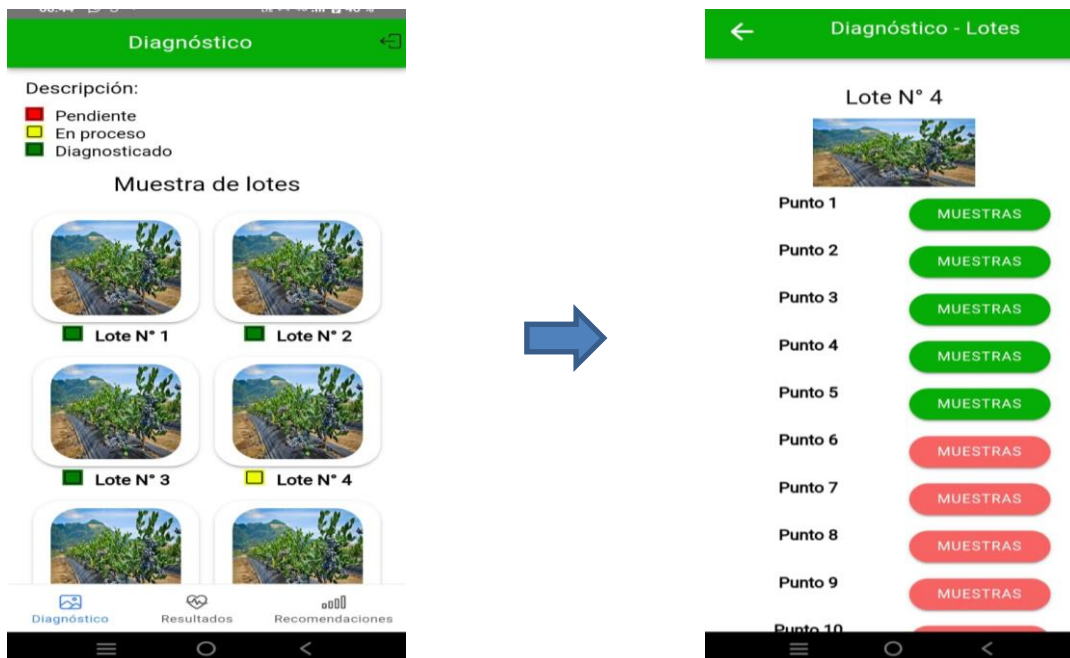
TABLA XXVIII: RESULTADOS DE EVALUACION TAM

Preguntas de percepción de los usuarios para la usabilidad y fiabilidad del sistema	Total, de puntos
1. Usando el sistema APPAGRO permitirá realizar el trabajo más rápido	12
2. Usando el sistema APPAGRO mejoraría mi rendimiento en el trabajo	12
3. Usando el sistema APPAGRO aumentaría mi productividad.	11
4. Usando el sistema APPAGRO mejora mi efectividad en el trabajo.	12
5. Usando APPAGRO se me hace más fácil realizar mi trabajo.	15
Promedio total de usabilidad	62
Porcentaje de usabilidad	83%
6. La interacción con APPAGRO es entendible.	14
7. Encuentro que APPAGRO es fácil de usar	14
8. Usando APPAGRO mis datos están seguros.	12
9. Usando APPAGRO los resultados son fiables.	11
Promedio total de fiabilidad	51
Porcentaje de fiabilidad	85%

finalmente, se analizan los resultados del modelo de aceptación tecnológica mostrando el promedio de usabilidad de las 5 primeras preguntas realizadas a 3 usuarios obteniendo el 83%, lo que sugiere que los usuarios encuentran la aplicación intuitiva y fácil de usar. Este porcentaje

refleja una percepción positiva en términos de rapidez, rendimiento y facilidad de uso en la realización de sus tareas agrícolas. Por otro lado, la fiabilidad de la aplicación, se obtuvo con un promedio del 85%, indica que los usuarios confían en la seguridad de los datos proporcionados por APPAGRO. Los resultados sugieren que los usuarios encuentran la interacción con la aplicación clara y comprensible, y tienen una percepción positiva sobre la seguridad de sus datos y la precisión de los resultados obtenidos sugiere seguir mejorando al ser usado el sistema.

Finalmente se concluye que la aplicación ayuda a detectar plagas y enfermedades del cultivo de arándanos en el distrito de olmos para ello se muestran las pantallas de la APPAGRO (figura 31).



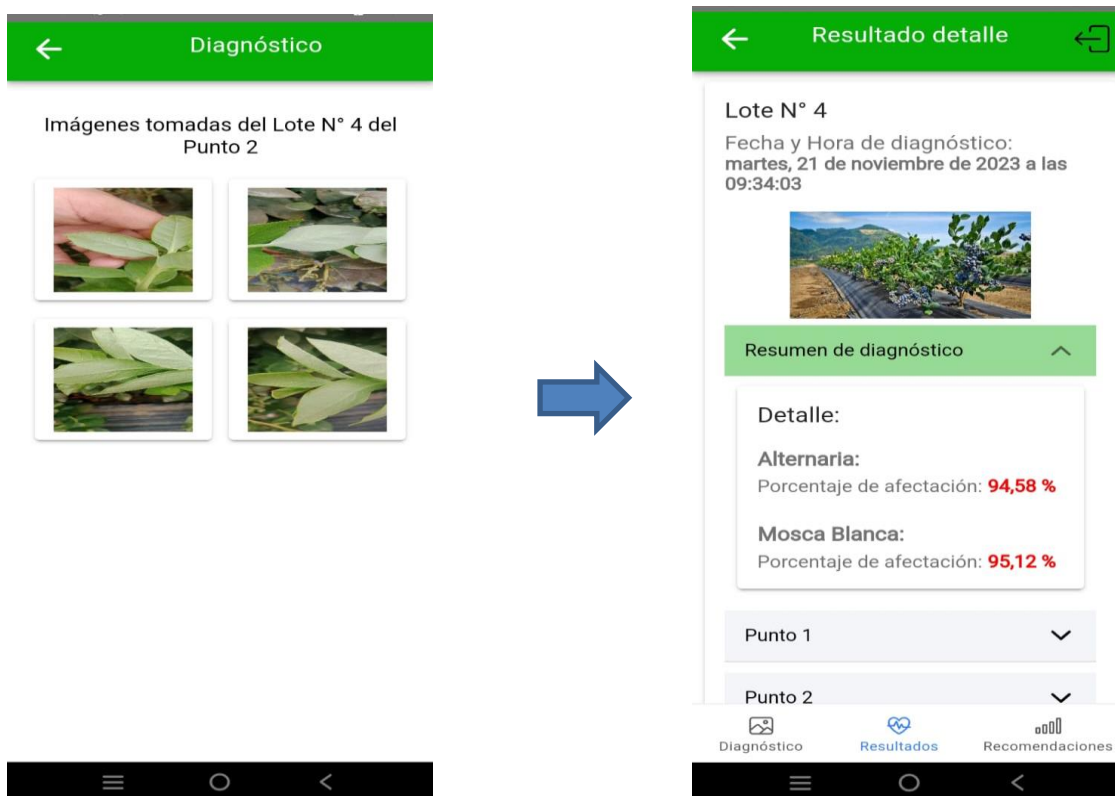


Fig.31 proceso del sistema APPAGRO

## Discusiones

La investigación evidencio la creación de la data\_set con imágenes infectadas con patógenos y además se recolecto imágenes con hojas sanas que sirvieron para entrenamiento, se utilizó el modelo de redes neuronales convolucionales, igual que [5] y [10] quienes trataron el tema de enfermedades y plagas en cultivo de cacao y arroz, utilizaron el modelo de CNN para el reconocimiento de imágenes y obtuvieron resultados de 88,03% y 88% de precisión respectivamente, esta investigación obtuvo 5325 imágenes para entrenar y validar el modelo su resultado fue 95% de precisión, quiere decir que el modelo implementado mejoro el reconocimiento de imágenes con respecto a los otros autores, se tomó en cuenta por que desarrollaron también aplicaciones móviles para apoyar a los agricultores. Por otro lado, para [38] también define su enfoque en enfermedades de palta, pero este entreno varios algoritmos y se quedó con el de Support Vector Machines que arrojó el 96% de exactitud, con un mejor porcentaje para clasificar dichas enfermedades, este trabajo nos sigue entrenar con varios

modelos para saber cuál es el más apropiado en este caso no hemos realizado entrenamiento con otros algoritmos por el tiempo para realizar cada modelo y también recolectar las imágenes pero se logró tomar como referencia porque uso la matriz de confusión para evaluar el modelo, para esta investigación estamos de acuerdo porque con la matriz de confusión hemos logrado obtener resultados para cada clase y también la precisión global del modelo en un 95% al reconocer los patógenos. En esta investigación no se aplicaron técnicas completas de procesamiento de mejora de imágenes por que se adquirieron en el ambiente natural donde se encontraba la planta en este caso se realizó la eliminación de las imágenes que estaban a simple vista borrosas o con baja iluminación para que no confunda el modelo al ser entrenado, pero si se asignó al modelo para que lea imágenes con extensiones '.jpg', '.png', '.jpeg', y se redimensionan las imágenes a 150\*150 pixeles, en cambio para [6] aplico tratamiento de imágenes mediante técnicas de visión computacional, que consiste en mejorar la iluminación de imágenes obtenidas en ambientes no controlados mediante las redes neuronales convolucionales y obtuvo una precisión de 80% que es aceptable pero que recomienda seguir agregando más técnicas de procesamiento de imágenes para futuros proyectos. Finalmente, esta investigación muestra los resultados de la plaga o enfermedad detectada con su respectivo porcentaje, también una recomendación para que sepan que agroquímicos puedan aplicar a su cultivo también el mismo evaluador podrá agregar un comentario sugiriendo que agroquímico se puede usar para las plagas o las enfermedades del cultivo de arándano por lo que nos apoyamos de [11] que también realizo un aplicativo móvil, uso el modelo de redes neuronales convolucionales, logro dar información para que generen conocimiento los agricultores de las plagas del cultivo de sandía posterior a ello puedan aplicar el agroquímico correcto.

## Conclusiones

En esta investigación, con la colaboración de un agrónomo especialista en plagas y enfermedades, se recolectaron un total de 5500 imágenes de plantas de arándanos en su entorno natural, utilizando únicamente la cámara de un teléfono móvil sin la aplicación de filtros adicionales. Posteriormente, se realizó un proceso de depuración eliminando aquellas imágenes que estaban borrosas o tenían una iluminación inadecuada, lo que resultó en un conjunto final de 5325 imágenes aptas a entrenar. Estas imágenes fueron clasificadas en cuatro clases como “alternaria”, “mosca blanca”, “hoja sana” y “otros” utilizadas para el entrenamiento y validación del modelo de redes neuronales convolucionales.

Se desarrolló un modelo de redes neuronales convolucionales (CNN) para la identificación de plagas y enfermedades en plantas de arándanos. Este modelo se entrenó con la totalidad de las 5325 imágenes recolectadas, organizadas en tres conjuntos: 70% para entrenamiento, 15% para validación y 15% para pruebas. Durante el entrenamiento, el modelo alcanzó una precisión del 95.3%. Para la validación del modelo, se utilizó una matriz de confusión que permitió evaluar la precisión por cada clase. Los resultados mostraron una precisión del 91.1% para la clase de mosca blanca y del 95.2% para la clase de alternaria, indicando que el modelo es eficaz en la identificación de estas plagas y enfermedades. La precisión global del modelo fue del 95.3%, confirmando su capacidad para clasificar correctamente las imágenes en general. Finalmente, se realizaron pruebas adicionales con un conjunto de imágenes nuevas no utilizadas en el entrenamiento, donde el modelo mantuvo un rendimiento promedio por encima del 95%. Esto demuestra que el modelo posee un alto desempeño y es capaz de realizar predicciones precisas con imágenes no vistas anteriormente.

Se desarrolló e implementó la aplicación móvil APPAGRO, la cual permite a los usuarios ingresar imágenes de hojas de arándanos directamente desde la cámara de su teléfono o desde la galería para realizar un diagnóstico de plagas y enfermedades. La aplicación utiliza un modelo de redes neuronales convolucionales (CNN) para predecir la clase de la plaga o enfermedad presente, proporcionando resultados con los porcentajes de afectación según el patógeno detectado. Además, APPAGRO permite a los usuarios añadir comentarios y sugerir agroquímicos que podrían ser utilizados en el cultivo, brindando así una herramienta integral para el manejo de plagas y enfermedades en los arándanos.

Finalmente, se concluye que el sistema APPAGRO demostró su fiabilidad y desempeño a través de la aplicación de la encuesta TAM a los usuarios. Los resultados mostraron un promedio de usabilidad del 83%, indicando que los usuarios encuentran la aplicación intuitiva y fácil de usar. En cuanto a la fiabilidad, la aplicación obtuvo un promedio del 85%, lo que refleja la confianza de los usuarios en la seguridad de los datos proporcionados y en la claridad y comprensibilidad de la interacción con APPAGRO.

### **Recomendaciones**

Para investigaciones futuras, se sugiere ampliar el número de plagas y enfermedades detectadas en el cultivo de arándano con el fin de mejorar el modelo y lograr diagnósticos más precisos. Además, se recomienda explorar y comparar diferentes algoritmos de entrenamiento para determinar cuál ofrece la mayor precisión, eligiendo así el más adecuado para el reconocimiento de imágenes en este contexto específico.

Adicionalmente, se aconseja evaluar modelos avanzados como YOLOv4, que ofrece una estructura más completa y está diseñado para la detección de objetos en tiempo real. Esta evaluación podría proporcionar resultados más precisos en la detección de patógenos en los cultivos de arándanos, mejorando así la efectividad del sistema de diagnóstico.

## Referencias

- [1] Agraria.pe, «Agraria.pe», Perú: aumenta la aparición de plagas y enfermedades en cultivos de arándanos. Accedido: 13 de agosto de 2023. [En línea]. Disponible en: <https://chilealimentos.com/peru-aumenta-la-aparicion-de-plagas-y-enfermedades-en-cultivos-de-arandanos/>
- [2] AgroPerú, AGROPERÚ Informa. Accedido: 8 de abril de 2024. [En línea]. Disponible en: <https://www.agroperu.pe/peru-y-colombia-implementaran-tecnologias-para-reducir-plagas-en-arandanos-y-yuca/>
- [3] Chilealimentos, «Perú: Aumenta la aparición de plagas y enfermedades en cultivos de arándanos | Chilealimentos». Accedido: 15 de noviembre de 2023. [En línea]. Disponible en: <https://chilealimentos.com/peru-aumenta-la-aparicion-de-plagas-y-enfermedades-en-cultivos-de-arandanos/>
- [4] RedAgricola, «Enfermedades y plagas ponen a prueba a los arandaneros», Redagrícola. Accedido: 13 de agosto de 2023. [En línea]. Disponible en: <https://redagricola.com/enfermedades-y-plagas-ponen-a-prueba-a-los-arandaneros/>
- [5] A. C. R. Caballero y M. J. R. Calderon, «Iacultivar: aplicación móvil con reconocimiento de imágenes para la detección y prevención de la monilla en el cultivo del cacao».
- [6] D. L. Hernández Rabadán y J. P. Sánchez Hernández, «Redes neuronales convolucionales para el reconocimiento de imágenes con presencia de cenicienta en cultivos de tomate», *Progmata*, vol. 14, n.º 3, pp. 21-28, oct. 2022, doi: 10.30973/progmata/2022.14.3/3.
- [7] G. T. Castro Alvarez, «Aplicación de algoritmos inteligentes para reconocimiento automático de enfermedades foliares de cultivo de palta.», may 2019, Accedido: 5 de junio de 2024. [En línea]. Disponible en: <https://repositorio.unam.edu.pe/handle/UNAM/98>
- [8] Universidad Nacional de Trujillo, Departamento de Ingeniería Mecatrónica. Trujillo, Perú, W. A. Narciso Horna, E. A. Manzano Ramos, y Universidad Nacional de Trujillo, Departamento de Ingeniería Mecatrónica. Trujillo, Perú, «Sistema de visión artificial basado en redes neuronales convolucionales para la selección de arándanos según estándares de exportación», *Campus*, vol. 26, n.º 32, pp. 155-166, dic. 2021, doi:

10.24265/campus.2021.v26n32.01.

[9] R. M. D. Pajares y G. I. O. Li, «“SOLUCIÓN DE MACHINE LEARNING EN EL RECONOCIMIENTO DE PLAGAS PARA PLANTONES DE ARÁNDANO”».

[10] J. L. Galan Zapata, «Sistema inteligente de reconocimiento de imágenes para apoyar el diagnóstico de plagas y enfermedades en el cultivo de arroz en el departamento de Lambayeque en el año 2019», 2021, Accedido: 21 de marzo de 2024. [En línea]. Disponible en: <http://tesis.usat.edu.pe/handle/20.500.12423/3216>

[11] J. E. Piscoya Ferreñan, «Sistema de visión artificial para apoyar en la identificación de plagas y enfermedades del cultivo de sandía en el distrito de Ferreñafe», 2019, Accedido: 21 de marzo de 2024. [En línea]. Disponible en: <http://tesis.usat.edu.pe/handle/20.500.12423/2356>

[12] D. J. C. Chaname, «TESIS PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN».

[13] J. Hurwitz, «Machine Learning For Dummies, IBM Limited Edition», 2018.

[14] «¿Cuáles son los tipos de algoritmos del machine learning?», APD España. Accedido: 25 de septiembre de 2023. [En línea]. Disponible en: <https://www.apd.es/algoritmos-del-machine-learning/>

[15] A. Hellín, V. Alonso, A. Sánchez, y J. C. Jacobo, «Aplicación de algoritmos Machine Learning en ejercicios deportivos realizados en series de repeticiones».

[16] «¿Qué es un árbol de decisión? | IBM». Accedido: 10 de noviembre de 2023. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/decision-trees>

[17] S. Khan, H. Rahmani, S. A. A. Shah, y M. Bennamoun, «Convolutional Neural Network», en *A Guide to Convolutional Neural Networks for Computer Vision*, Cham: Springer International Publishing, 2018, pp. 43-68. doi: 10.1007/978-3-031-01821-3\_4.

[18] J. S. Prieto, *Redes neuronales convolucionales y redes neuronales recurrentes en la transcripción automática*. 2019. doi: 10.13140/RG.2.2.10855.39843.

[19] J. M. A. A. Maluenda Raquel, «Qué es Ionic: ventajas y desventajas de usarlo en apps móviles híbridas», Profile Software Services. Accedido: 31 de agosto de 2023. [En línea]. Disponible en: <https://profile.es/blog/que-es-ionic/>

[20] T. School, «¿Qué es Angular? El framework de JavaScript para crear aplicaciones web dinámicas», Tokio School. Accedido: 31 de agosto de 2023. [En línea]. Disponible en: <https://www.tokioschool.com/noticias/que-es-angular/>

[21] R.- ASALE y RAE, «arándano | Diccionario de la lengua española», «Diccionario de la

lengua española» - Edición del Tricentenario. Accedido: 10 de abril de 2024. [En línea]. Disponible en: <https://dle.rae.es/arándano>

[22] «Manejo de Enfermedades en Arándanos | Intagri S.C.» Accedido: 10 de junio de 2022. [En línea]. Disponible en: <https://www.intagri.com/articulos/fitosanidad/manejo-de-enfermedades-en-arandanos>

[23] «9 - Ciclo de vida de un proyecto de Machine Learning», Codificando Bits. Accedido: 2 de octubre de 2023. [En línea]. Disponible en: <https://www.codificandobits.com/curso/introduccion-machine-learning/9-ciclo-de-vida-proyecto-machine-learning/>

[24] J. R. Coveñas, «SISTEMA INTELIGENTE PARA LA IDENTIFICACIÓN DE DERMATOFITOSIS EN PACIENTES DE UN CENTRO DERMATOLÓGICO DE LA CIUDAD DE CHICLAYO».

[25] K. P. Murphy, *Machine learning: a probabilistic perspective*, 4. print. (fixed many typos). en Adaptive computation and machine learning series. Cambridge, Mass.: MIT Press, 2013.

[26] P. Domingos, «A few useful things to know about machine learning», *Commun. ACM*, vol. 55, n.º 10, pp. 78-87, oct. 2012, doi: 10.1145/2347736.2347755.

[27] D. Zafra, «Ruido en Fotografía: Qué Es y Cómo Controlarlo», Capture the Atlas. Accedido: 3 de noviembre de 2023. [En línea]. Disponible en: <https://capturetheatlas.com/es/ruido-en-fotografia/>

[28] «T-112972 González - Pacheco.pdf». Accedido: 30 de octubre de 2023. [En línea]. Disponible en: <https://www.dspace.espol.edu.ec/bitstream/123456789/57041/1/T-112972%20Gonz%20a11ez%20-%20Pacheco.pdf>

[29] F. L. H. Carmona, «Diagnóstico de neumonía por clasificación de imágenes de rayos X de tórax utilizando el método de redes neuronales convolucionales (CNN)».

[30] M. J. V. Soto, «TESIS PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN».

[31] «TensorFlow Core | Aprendizaje automático para principiantes y expertos». Accedido: 7 de noviembre de 2023. [En línea]. Disponible en: <https://www.tensorflow.org/overview?hl=es-419>

[32] J. J. C. Pacheco, «Clasificación de imágenes médicas con técnicas de “Deep Learning”».

[33] L. Gonzalez, «Matriz de Confusión», Aprende IA. Accedido: 8 de noviembre de 2023.

[En línea]. Disponible en: <https://aprendeia.com/matriz-de-confusion-machine-learning/>

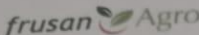
[34] S. G. Rivadeneira Molina, «Metodologías ágiles enfocadas al modelado de requerimientos», *ICTUNPA*, vol. 5, n.º 1, pp. 1-29, jun. 2014, doi: 10.22305/ict-unpa.v5i1.66.

[35] M. J. V. Soto, «TESIS PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN».

[36] E. A. R. Adanaque, «SISTEMA INTELIGENTE PARA IDENTIFICAR ADECUADAMENTE EL MANGO KENT NO EXPORTABLE EN EL ÁREA DE MUESTREO DE UNA EMPRESA AGROINDUSTRIAL DE LA REGIÓN LAMBAYEQUE».

## Anexos

## Anexo 01 CARTA DE ACEPTACION DE PRODUCTO ACREDITABLE



Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho

CONSTANCIA DE APROBACIÓN DEL PRODUCTO ACREDITABLE

Olmos 10 de junio 2024

Carta N° 00124-2024

**Señorita**

**SUYON FARROÑAN MARIA ESTHER**


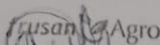
Estudiante de la universidad Católica Santo Toribio de Mogrovejo

Por medio de la presente hago constar la Aprobación del producto acreditable del proyecto titulado "APLICACIÓN MÓVIL BASADO EN RECONOCIMIENTO DE IMÁGENES PARA APOYAR LA DETECCIÓN DE PLAGAS Y ENFERMDADES ESTACIONALES DEL CULTIVO DE ARANDANO EN EL DISTRITO DE OLMOS", el cual fue culminado satisfactoriamente y revisado por el área correspondiente, es por ello que se APRUEBA el sistema APPAGRO.

Sin otro particular aprovechamos la ocasión para expresarle nuestros sentimientos de consideración y estima personal.

Se expide la presente constancia para los fines que crea conveniente.

Atentamente:

  
  
ING. PABLO SALDIAS CARO  
DNI: 005547320  
JEFE DE CONTROL DE CALIDAD

\_\_\_\_\_  
ING. Pablo Saldias Caro  
DNI: 005547320  
Jefe Control de Calidad

<https://web.facebook.com/p/Empresa-Frusan-Agro-SAC->  
Ubicación: Olmos, Lambayeque - Perú

## Anexo 02

**GUIA DE ENTREVISTA 01 CONOCER LA REALIDAD  
PROBLEMÁTICA DE LA EMPRESA EN ESTUDIO****ENTREVISTADO:** Pablo Saldias Caro**ENTREVISTADOR:** María Esther Suyon Farroñan**Objetivo:** Conocer la realidad problemática**Fecha :**01/09/2023**Preguntas:**

1. ¿Cuáles son las plagas y las enfermedades que más atacan los cultivos de arándanos?
2. ¿Cuáles son las características de las plagas y enfermedades que afectan sus cultivos de arándanos?
3. ¿Cómo se propagan estas plagas y enfermedades?
4. ¿En qué tiempo o estaciones se evidencian con mayor visibilidad?
5. ¿Cuáles son las medidas preventivas que toman para evitar las plagas y enfermedades en sus cultivos?
6. ¿Cuáles son los tratamientos químicos o biológicos que utilizan para controlar las plagas y enfermedades?
7. ¿Cómo logran identificar las plagas y enfermedades en sus cultivos?
8. ¿En qué porcentaje llegan afectar las plagas y enfermedades a sus cultivos?
9. ¿Tienen algún repositorio de imágenes con plagas y enfermedades de su cultivo?
10. ¿Cuentan con algún aplicativo para reconocimiento de plagas y enfermedades de sus cultivos?



ING. PABLO SALDIAS CARO  
DNI. 005547320  
JEFE DE CONTROL DE CALIDAD

---

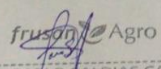
Firma del encuestado

TABLA III  
ELIMINACIÓN DE IMÁGENES

Data set	Mal enfocadas (Borrosas)	Iluminación
Alternaria	65	8
Mosca blanca	58	7
Hoja sana	34	3
Total		175

Anexo 03

ENCUESTA MODELO DE ACEPTACIÓN TECNOLÓGICA(TAM)

TÍTULO	VALIDACIÓN DEL SISTEMA APPAGRO
PARTICIPANTE	ING. PABLO SALDÍAS CARO
OBJETIVO	Conocer la percepción de los usuarios en cuanto a la usabilidad y fiabilidad del sistema
Indicador: escala de Likert de 5 puntos	Índice alto= fácil Índice bajo= difícil
Fuertemente Improbable= 1; poco probable =2; neutral =3; probable =4; Fuertemente probable =5	
<b>Cuestionario:</b> Marque la respuesta con (X)	CALIFICACIÓN
1. Usando el sistema APPAGRO permitirá realizar el trabajo más rápido	1   2   3   4   5 (4)
2. Usando el sistema APPAGRO mejoraría mi rendimiento en el trabajo.	1   2   3   4   5 (4)
3. Usando el sistema APPAGRO aumentaría mi productividad.	1   2   3   4   5 (3)
4. Usando el sistema APPAGRO mejora mi efectividad en el trabajo.	1   2   3   4   5 (4)
5. Usando APPAGRO se me hace más fácil realizar mi trabajo	1   2   3   4   5 (5)
6. La interacción con APPAGRO es entendible.	1   2   3   4   5 (4)
7. Encuentro que APPAGRO es fácil de usar.	1   2   3   4   5 (5)
8. Usando APPAGRO mis datos están seguro.	1   2   3   4   5 (4)
9. Usando APPAGRO los resultados son fiables.	1   2   3   4   5 (3)
Fecha: 20-11-23	
Realizado por: María Esther Suyon Farroñan	
Estudiante de la Universidad Católica Santo Toribio de Mogrovejo	
Firma :	 ING. PABLO SALDÍAS CARO <small>CHAL. 10750497320</small> <small>JEFE DE INSTITUTO DE CALIDAD</small>