

# Predicción de Agentes Patógenos en Plantas Ornamentales Utilizando Redes Neuronales

## TESIS

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:  
ING. EMMANUEL ESCOBAR AVALOS

DIRECTOR DE TESIS  
DR. NOEL GARCÍA DÍAZ

CO-DIRECTOR DE TESIS  
DR. JESÚS ALBERTO VERDUZCO RAMÍREZ

VILLA DE ÁLVAREZ, COLIMA, JULIO 2018.





UNIDAD ORGÁNICA: DIVISIÓN DE ESTUDIOS DE POSGRADO  
INVESTIGACIÓN  
ASUNTO: LIBERACIÓN DE TESIS  
NO. OFICIO: DEPI 187/18

Villa de Álvarez, Col. **29/JUNIO/2018**

C. EMMANUEL ESCOBAR AVALOS  
PASANTE DE LA MAESTRÍA EN SISTEMAS COMPUTACIONALES  
PRESENTE.

La División de Estudios de Posgrado e Investigación de acuerdo al procedimiento para la obtención del Título de Maestría de los Institutos Tecnológicos y habiendo cumplido con todas las indicaciones que la comisión revisora hizo a su trabajo profesional denominado **PREDICCIÓN DE AGENTES PATÓGENOS EN PLANTAS ORNAMENTALES UTILIZANDO REDES NEURONALES**, por la opción de Tesis, que para obtener el grado de **Maestro en Sistemas Computacionales** será presentado por Usted, tiene a bien concederle la **autorización** de impresión de la tesis citada.

Sin otro particular por el momento, aprovecho la ocasión para enviarte un cordial y afectuoso saludo.

ATENTAMENTE  
Excelencia en Educación Tecnológica®  
"Estudiar para prever y prever para actuar"

MTRA. RAMONA EVELIA CHÁVEZ VALDEZ  
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



C.p.p. Archivo

RECV/cas





Tu atmósfera de vida

Asunto: Carta de terminación

Colima, Col. 12 de Junio de 2018.

Mtra. Maria Heidi del Pilar Vizcaino Granados  
Jefa del Departamento de Gestión Tecnológica y Vinculación  
Instituto Tecnológico de Colima  
Presente

Por medio de la presente hago constar, que el Ing. Emmanuel Escobar Avalos con número de control G1646016 y alumno de la Maestría de Sistemas Computacionales, terminó satisfactoriamente su estancia de tesis profesional con el proyecto "Predicción de Agentes Patógenos en Plantas Ornamentales Utilizando Redes Neuronales" en las instalaciones propias de la empresa Ornamentales de Colima S.P.R. de R.L. de C.V (ORNACOL), durante el periodo de Enero a Mayo de 2018.

Se extiende la presente a petición de la interesada y para los fines legales que a ella convengan, quedo de usted para cualquier aclaración.

Atentamente

Ing. Rubén Cabrera Silva  
SECRETARIO DEL CONSEJO  
ESTATAL DE PRODUCTORES  
DE PLANTAS DE ORNATO  
DE COLIMA A.C.  
(COEPPLANTS)



km .5 Carretera al Chical Coquimatlan, Colima C.P. 28400



## Agradecimientos

Al Instituto Tecnológico de Colima, el cual me permitió realizar mis estudios y me ofreció a puertas abiertas sus instalaciones para poder concluirla.

Al Consejo Nacional de Ciencia y Tecnología, por el apoyo económico durante toda la carrera, ya que sin ello hubiera sido difícil continuar los estudios.

A los maestros y revisores por su tiempo y dedicación. En especial, al Dr. Noel García Díaz, por ayudarme a despertar el interés por la investigación, guiándome con el aprendizaje en las redes neuronales. Es así como, me hizo encontrar un nuevo mundo en la forma de usar la programación, logrando conocer los sistemas expertos e inteligentes. Su dedicación, didáctica y esa forma tan particular de ofrecer su conocimiento sin restricciones, ayudó a motivarme para concluir esta tesis.

A los compañeros, que, con su entusiasmo y amistad, lograron hacer amena la estancia de posgrado.

A mis hermanos, mi padre y familia política por todo su apoyo.

Finalmente, siendo la pieza fundamental de mi vida; mi esposa, Leonor Suárez, ya que, sin ella, hubiera sido imposible haber iniciado un grado más en mi carrera profesional, agradezco a Dios por haberla puesto en mi camino y ser el complemento de mi vida.

*"Las 'leyes del pensamiento' no solo dependen de las propiedades de las células cerebrales, sino del modo en que están conectadas".*

*Marvin Minsky (1987).*

## Resumen

Una de las principales características para la comercialización de plantas ornamentales es el aspecto visual de la planta. El aspecto visual de la planta facilita dicha comercialización, es decir, el cliente seleccionará aquella planta que cuente con un buen aspecto y una apariencia sana. Sin embargo, el aspecto y la salud de una planta se puede ver comprometido por agentes patógenos (plagas), los cuales pueden además afectar no sólo a la planta en mención, sino también, a todo el lote de plantas ornamentales cercanas a ésta. La detección oportuna de los posibles agentes patógenos en plantas ornamentales es un factor crítico para el sector productivo, ya que permite evitar o controlar la proliferación de éstos. Por lo tanto, en esta tesis de maestría se propuso y se implementó una herramienta llamada "OrnaNet", la cual permite predecir la presencia de agentes patógenos en la producción de plantas ornamentales a través de redes neuronales convolucionales. OrnaNet requiere de imágenes previamente tomadas de plantas dañadas por algún patógeno para ser procesadas por la red neuronal, logrando la extracción de características necesarias para efectuar la predicción del daño ocasionado por el patógeno en la planta ornamental. Los resultados arrojados por OrnaNet son almacenados en una base de datos accesible a través de un cliente web en donde se obtiene la predicción del agente patógeno y la ubicación del daño. OrnaNet se desarrolló en la empresa Ornamentales de Colima S.P.R. de R.L. de C.V (ORNACOL) ubicada en el municipio de Coquimatlán, del estado de Colima. A través de los resultados obtenidos con OrnaNet se concluye en esta tesis de maestría que la utilización de redes neuronales aporta significativamente mejoras en la predicción de agentes patógenos en la producción de plantas ornamentales.

## **Abstract**

One main characteristic for successful commercialization of ornamental plants is the appearance of the plant. Commonly, customers tend to select plants that have good and healthy appearance. Ornamental plants can be compromised by pathogens (pests) that may extend to entire batches of plants. A timely detection of these pathogens is a critical factor to control and avoid proliferation of the plague. In this master thesis, a tool called "OrnaNet" is proposed and implemented. OrnaNet allows predicts the presence of pathogens in ornamental plants through the use of convolutional neural networks. The neural network is trained with images of damaged plants to extract characteristics the pathogen in order to predict if an ornamental plant suffers from the pathogen. OrnaNet was developed and implemented at Ornamentales de Colima S.P.R. de R.L. de C.V. (ORNACOL) in Coquimatlán, Colima. Its results were stored in a database that is accessible through a Web client. We show that the use of neural networks provide signification improvements of pathogen detection of ornamental plants.

# 3ndice

<b>1. Introducci3n</b> .....	10
<b>1.1. La naturaleza del problema</b> .....	10
<b>1.2. El contexto del problema</b> .....	11
<b>1.3. Revisi3n de la literatura</b> .....	12
<b>1.4. Planteamiento de la soluci3n</b> .....	13
<b>1.5. Justificaci3n</b> .....	14
<b>1.6. Motivaci3n</b> .....	14
<b>1.7. Objetivo</b> .....	14
<b>1.7.1. Objetivos particulares</b> .....	14
<b>1.8. Hip3tesis</b> .....	15
<b>1.9. Descripci3n de los m3todos empleados</b> .....	15
<b>1.10. Descripci3n de cap3tulos de tesis</b> .....	16
<b>2. Estado del campo del conocimiento</b> .....	17
<b>2.1. Marco hist3rico</b> .....	17
<b>2.2 Marco contextual</b> .....	23
<b>2.3 Marco te3rico</b> .....	46
<b>3. M3todos empleados</b> .....	52
<b>4. Desarrollo</b> .....	56
<b>4.1. An3lisis</b> .....	56
<b>4.1.1. Modelo conceptual</b> .....	57
<b>4.1.2. Diagrama de actividades</b> .....	59
<b>4.1.3. Diagrama de casos de uso</b> .....	60
<b>4.1.4. Modelo de requisitos</b> .....	61
4.1.4.1 Requisitos funcionales .....	61
4.1.4.2. Requisitos de software y hardware .....	62
<b>4.2. Dise3no</b> .....	63
<b>4.2.1. Modelo de clases.</b> .....	63
<b>4.2.2 Modelo de datos</b> .....	64
<b>4.2.3 Diccionario de datos</b> .....	65

<b>4.2.4 Diagrama de navegación</b> .....	66
<b>4.2.5 Modelo de interfaces</b> .....	67
4.2.5.1 Servidor local con MATLAB .....	67
4.2.5.2 Servidor WEB .....	71
<b>4.2.6 Modelo de componentes</b> .....	73
<b>4.3. Implementación</b> .....	74
<b>4.3.1. Módulo servidor local</b> .....	74
4.3.1.1. Proceso de reentrenamiento de la CNN .....	77
4.3.1.2. Creación de interfaz gráfica .....	78
<b>4.3.2. Creación de la base de datos</b> .....	81
<b>4.3.3. Módulo WEB</b> .....	84
<b>4.4. Verificación y validación</b> .....	89
<b>4.4.1 Prueba de complejidad MacCabe</b> .....	89
<b>4.4.1. Prueba de tiempo de ejecución</b> .....	91
<b>5. Resultados</b> .....	96
<b>6. Conclusiones y trabajo a futuro</b> .....	100
<b>Bibliografía</b> .....	101

## Índice de Tablas y Figuras

<b>Figura 1.</b> Jardines botánicos en México.(Vovides et al., 2010) .....	18
<b>Figura 2.</b> Sistema de visión artificial (Fernández García, 2012). .....	20
<b>Figura 3.</b> Diferentes técnicas para detección y clasificación de objetos, (1) LBP, (2) HOG, (3) SVM, (4) Filtros de HAAR.....	21
<b>Figura 4.</b> Componentes de una neurona artificial y comparativa a una neurona biológica.....	22
<b>Figura 5.</b> Arquitectura de la red neuronal convolucional LeNet-5 (Lecun et al., 1998). .....	22
<b>Figura 6.</b> Arquitectura de LeafNet. ....	25
<b>Figura 7.</b> Imágenes de entrada y salida con la aplicación del algoritmo. ....	26
<b>Figura 8.</b> Diagrama de flujo de la detección de daño en el procesamiento de imágenes. ....	27
<b>Figura 9.</b> Matrices de confusión de las CNN AlexNet y VGGNet respectivamente. ....	28
<b>Figura 10.</b> Tres ejemplos de hojas de conjunto de datos utilizados en (VijayaLakshmi & Mohan, 2016).....	29
<b>Figura 11.</b> Arquitectura del sistema para la detección de daño y clasificación de hojas. ....	30
<b>Figura 12.</b> Estructura de una red neuronal convolucional.....	31
<b>Figura 13.</b> Ejemplo de la aplicación en funcionamiento. ....	31
<b>Figura 14.</b> Precisión de predicción para cada clase por separado. ....	34
<b>Figura 15.</b> Arquitectura propuesta de la red neuronal convolucional. ....	35
<b>Figura 16.</b> Matriz de confusión para la validación de la predicción con 21 especies de plantas. Los números de la diagonal, indican el porcentaje de clasificaciones correctas para cada clase. ....	36
<b>Figura 17.</b> Resultados del modelo de dilatación en diferentes escenarios del entrenamiento. ....	37
<b>Figura 18.</b> Visualización de características de ejemplo de ocho capas de una red neuronal convolucional profunda.....	38
<b>Figura 19.</b> Ejemplo de Leaf Doctor operando.....	39
<b>Figura 20.</b> Izquierda: red de propuesta de región (RPN). Derecha: ejemplos de detecciones usando propuestas de RPN.....	40
<b>Figura 21.</b> Matriz de confusión mostrando la precisión del identificador. ....	41
<b>Figura 22.</b> Diagrama de flujo de la metodología propuesta. ....	42
<b>Figura 23.</b> Función principal de la red neuronal artificial. ....	43
<b>Figura 24.</b> Imagen con la muestra original dañada y aplicación de las diferentes técnicas utilizadas en la investigación. ....	44
<b>Figura 25.</b> Muestra las cinco etiquetas que el modelo considera más probables. La etiqueta correcta se escribe debajo de cada imagen, y la probabilidad asignada a la etiqueta correcta también se muestra con una barra roja. ....	45
<b>Figura 26.</b> Vivero de ORNACOL.....	47

<b>Figura 27.</b> Esquema de visión artificial (Diagrama, 2016). .....	47
<b>Figura 28.</b> Red Neuronal Artificial.....	48
<b>Figura 29.</b> Arquitectura de una red neuronal convolucional.....	49
<b>Figura 30.</b> Arquitectura de la red neuronal artificial AlexNet.....	49
<b>Figura 31.</b> Ejemplo de los procesos de Kanban. ....	50
<b>Figura 32.</b> Logo Angular 5. ....	50
<b>Figura 33.</b> Logo MATLAB. ....	51
<b>Figura 34.</b> Logo XAMPP. ....	51
<b>Figura 35.</b> Arquitectura de la CNN AlexNet (Krizhevsky et al., 2012). ....	52
<b>Figura 36.</b> Muestra de 10 imágenes de entrenamiento. ....	53
<b>Figura 37.</b> Esquema de la metodología Kanban con peticiones de requerimientos.....	55
<b>Figura 38.</b> Sistema para la Predicción de Agentes Patógenos en Plantas Ornamentales Utilizando Redes Neuronales.....	57
<b>Figura 39.</b> Diagrama de actividades. ....	59
<b>Figura 40.</b> Modelo de casos de uso. ....	60
<b>Figura 41.</b> Modelo de clases. ....	64
<b>Figura 42.</b> Modelo de datos.....	64
<b>Figura 43.</b> Diccionario de datos. ....	65
<b>Figura 44.</b> Diagrama de navegación.....	66
<b>Figura 45.</b> Pantalla inicial del sistema local. ....	67
<b>Figura 46.</b> Selección de imagen a procesar. ....	68
<b>Figura 47.</b> Pantalla con datos obtenidos de la imagen y la predicción. ....	69
<b>Figura 48.</b> Pantalla de registros guardados en la base de datos.....	70
<b>Figura 49.</b> Página principal de la aplicación, donde se muestra el mapa y aparecerán las plagas encontradas. ....	71
<b>Figura 50.</b> Pantalla donde se muestra a detalle los registros de datos. ....	72
<b>Figura 51.</b> Modelo de Componentes.....	73
<b>Figura 52.</b> Pantalla principal con icono de complementos de MATLAB. ....	74
<b>Figura 53.</b> Pantalla del explorador de complementos de MATLAB. ....	75
<b>Figura 54.</b> Pantalla del complemento CNN AlexNet. ....	75
<b>Figura 55.</b> Comprobación de funcionamiento de la CNN AlexNet en MATLAB. ....	76
<b>Figura 56.</b> Procesos de reentrenamiento de la CNN. ....	77
<b>Figura 57.</b> Tareas que se están realizando en este capítulo. ....	78
<b>Figura 58.</b> Inicio para la creación de interfaz gráfica de usuario.....	79
<b>Figura 59.</b> Interfaz gráfica de usuario realizada para el servidor generada en MATLAB. ....	80
<b>Figura 60.</b> Tarea requerida de la creación de base de datos. ....	81
<b>Figura 61.</b> Código MySQL para la creación de la base de datos.....	82
<b>Figura 62.</b> Panel de control XAMPP. ....	83
<b>Figura 63.</b> Servidor local MySQL.....	84
<b>Figura 64.</b> Tareas que se realizan en este capítulo. ....	85
<b>Figura 65.</b> Proceso para crear la aplicación WEB. ....	86

<b>Figura 66.</b> Módulos de la aplicación WEB realizada en Angular 5. ....	87
<b>Figura 67.</b> Pantalla principal de la aplicación WEB. ....	88
<b>Figura 68.</b> Gráfica de resultados del primer entrenamiento de la predicción.....	97
<b>Figura 69.</b> Gráfica de resultados del segundo entrenamiento de la predicción.....	98
<b>Figura 70.</b> Gráfica de resultados del tercer entrenamiento de la predicción.....	99
<b>Tabla 1.</b> Detalles del conjunto de datos utilizados para la identificación de plantas.....	23
<b>Tabla 2.</b> Porcentaje de precisión en la clasificación (%) en las diferentes redes neuronales convolucionales. ....	24
<b>Tabla 3.</b> Lista de autores y trabajos realizados con los resultados obtenidos. ....	32
<b>Tabla 4.</b> Resultados de todos los modelos en los conjuntos de datos de imágenes. ....	33
<b>Tabla 5.</b> Capas utilizadas en el modelo de red neuronal reentrenada. ....	53
<b>Tabla 6.</b> Requisitos funcionales del servidor local. ....	61
<b>Tabla 7.</b> Requisitos funcionales de la aplicación WEB. ....	62
<b>Tabla 8.</b> Valores de entrenamiento final de la CNN.....	77
<b>Tabla 9.</b> Riesgo con base a la función de complejidad ciclomática. ....	89
<b>Tabla 10.</b> Resultados de la prueba McCabe.....	90
<b>Tabla 11.</b> Resumen del profiler. ....	91
<b>Tabla 12.</b> Módulos con mayor demanda de tiempo.....	95
<b>Tabla 13.</b> Conjunto de imágenes para entrenamiento. ....	96
<b>Tabla 14.</b> Resultados del primer entrenamiento.....	96
<b>Tabla 15.</b> Resultados del segundo entrenamiento.....	97
<b>Tabla 16.</b> Resultados del tercer entrenamiento.....	98

## Índice de abreviaturas y símbolos

<b>CNN</b>	Convolutional Neural Network
<b>VGG</b>	Visual Geometry Group
<b>LifeCLEF</b>	Laboratorio que contiene base de datos de imágenes de diferentes tipos para entrenamiento de aprendizaje profundo, además de crear concursos para nuevas tecnologías de reconocimiento en imágenes.
<b>MAXPOOLING</b>	Proceso para reducir una matriz de datos para una red neuronal convolucional, ésta toma el dato más grande de los diferentes grupos de matrices para crear una nueva matriz de datos más pequeña con los datos extraídos
<b>SOFTMAX</b>	Función Exponencial Normalizada
<b>SGDM</b>	Stochastic Gradient Descent with Momentum

# 1. Introducción

## 1.1. La naturaleza del problema

El comercio internacional sobre la demanda de flores y plantas se concentra en países de Europa, China, Japón y Estado Unidos. En el 2015, la Unión Europea importó un total de 504 952 toneladas de éstas con un valor de 1.68 billones de euros. Mientras que, exportó el 20.5% de su exportación a Rusia y Suiza; un 11.2% a Estados Unidos; 8.2% a Noruega y 5.9% a China (IPM ESSEN, 2017). La Unión Europea, Estados Unidos y Japón absorben aproximadamente tres cuartas partes del consumo hortícola mundial tanto de plantas como de flores de corte (SAGARPA, 2009).

El territorio de México se considera una extensión fértil para la producción y comercialización de plantas ornamentales (Luis & Carreto, 2014), de éste 12 entidades conforman el “Sistema Producto Ornamentales Nacionales”, siendo éstos los más importantes en la comercialización de plantas ornamentales. El estado de Colima es una de las entidades seleccionadas como productor de planta ornamental (Plan Rector, 2012).

La horticultura está dividida en diferentes ramas ubicando las plantas ornamentales en la floricultura (Felix & Bellido, 1994), ya que se basa en las características estéticas como: flores, hojas, perfume, peculiaridad del follaje y frutos o tallos (Wikipedia, 2017).

En México, una de las empresas dedicadas a la producción y comercialización de plantas ornamentales es “Ornamentales de Colima (ORNACOL)”, la cual por su rápida expansión generó el requerimiento de poder automatizar la monitorización de las plantas, con la finalidad de controlar y prevenir agentes patógenos que puedan afectar a éstas y así seguir manteniendo su calidad, tanto de producción como en comercialización para usuarios finales.

## 1.2. El contexto del problema

Actualmente ORNACOL realiza las operaciones de revisión en plantas ornamentales con personal experto que evalúa y registra algún daño causado en los lotes de plantas para estimar la salud actual de las mismas, con la finalidad de identificar si son comercializables o si alguna de éstas debe ser apartada del lote, por alguna plaga. Esta tarea es tediosa y no facilita contar con un concentrado de los daños que se registra en la bitácora de cada vivero. Por lo tanto, en esta investigación, se realiza la mejora de este proceso, ayudando a realizar la evaluación de manera automática, buscando encontrar el agente patógeno que afecta las plantas ornamentales, para de manera temprana poder mitigar el daño o erradicar la plaga en los viveros, ya que éstas son el principal factor de la proliferación de virus, generando agentes patógenos (Verdin et al., 2017).

Con el objetivo de encontrar agentes patógenos en las plantas ornamentales, se usó el procesamiento digital de imágenes en conjunto con técnicas de redes neuronales, ya que actualmente son técnicas exitosas que se utilizan para el reconocimiento y procesamiento de datos e imágenes (Juan & Chacón, 2011), esto porque dichas técnicas realizan operaciones no lineales que es el ambiente con aproximación a la vida real.

### 1.3. Revisión de la literatura

Ghaiwat y Arora (Ghaiwat & Arora, 2014) realizaron una investigación sobre la detección y clasificación de las enfermedades de las hojas mediante diferentes técnicas de procesamiento de imágenes, como la clasificación de vecinos próximos (Nearest Neighbor), red neuronal probabilística, algoritmos genéticos, máquina de soporte vectorial, lógica difusa y otras más técnicas.

Por otro lado, García y Barbedo (2013) utilizaron técnicas del procesamiento de imágenes digitales para detectar y cuantificar el deterioro de la plantas donde se usó la red neuronal perceptrón multicapa para la estimación del daño en las plantas.

Mientras que García Pablo et. al (2012) realizaron el proyecto para el reconocimiento de imágenes utilizando redes neuronales para dispositivos móviles, En este ámbito, Juan & Chacón (2011), llevaron a cabo una investigación sobre las redes neuronales artificiales para el procesamiento de imágenes, muy detallado y con las diferentes modificaciones que se han presentado a lo largo de los años gracias a la evolución de estas técnicas en las redes neuronales.

Finalmente, el trabajo realizado por Vega, H. et al (2009), describe el reconocimiento de patrones sobre los números naturales, utilizando técnicas de redes neuronales para poder predecir qué número es el muestra la imagen.

Tomando en cuentas las investigaciones anteriores, se observó que cada una de ellas utilizan las técnicas de redes neuronales para reconocimiento de imágenes o patrones en los respectivos trabajos de investigación; sin embargo, ninguno de los trabajos anteriores se enfocó en la predicción de agentes patógenos en plantas ornamentales utilizando redes neuronales convolucionales (CNN).

## **1.4. Planteamiento de la solución**

Con base en las oportunidades detectadas y sustentado por el trabajo de investigación realizado por Gando et al (2016) donde se menciona que las técnicas de CNN se han desempeñado bien para el reconocimiento de características en las imágenes y clasificación de ellas; por lo tanto, se toma en cuenta esta técnica para realizar la creación de una CNN capaz de predecir agentes patógenos en las plantas ornamentales.

Para complementar la solución se propone implantar una herramienta tecnológica que realice la predicción de agentes patógenos, agregando la funcionalidad de registrar los resultados y ubicación de los daños encontrados en cada vivero en una base de datos con el fin de contar con una vista en tiempo real de la presencia de algún agente patógeno en cada uno de los viveros.

El sistema a desarrollar constará con dos módulos importantes; el primero de ellos, es una aplicación central local (servidor), la cual será capaz de predecir el agente patógeno que presenta la imagen y obtener los metadatos de ella, de estos metadatos se obtendrá la ubicación donde se realizó la toma, posteriormente se podrá agregar el registro a una base de datos vía Internet; el segundo módulo es una aplicación vía WEB que en tiempo real mostrará la ubicación de la predicción de agentes patógenos.

## **1.5. Justificación**

La automatización de registros y predicción de agentes patógenos, ayuda a agilizar los muestreos y mantener el control en los diferentes viveros que pertenecen a ORNACOL. Además, ofrecer a los usuarios una forma sencilla de acceder a información que ayuden a prevenir o mantener el control de calidad en la producción de la planta, así como una visión actual en el mapa de algún agente patógeno detectado en los viveros.

## **1.6. Motivación**

El contribuir con la mejora en la producción de plantas ornamentales para aumentar la comercialización y utilización de las mismas en los hogares, ayuda a incrementar la economía en el estado y mejorar el medio ambiente, ya que, en la actualidad, la urbanización ha aumentado el calentamiento global; por lo cual, es necesario contar con plantas en cada uno de los hogares para que ayude a disminuir este efecto dañino. Utilizando técnicas innovadoras y sofisticadas como las CNN se contribuye a reducir el daño que puede causar este tipo de plaga.

## **1.7. Objetivo**

Desarrollar una herramienta tecnológica basada en CNN para predecir la presencia de agentes patógenos en plantas ornamentales.

### **1.7.1. Objetivos particulares**

Para poder llegar al objetivo principal de este trabajo, se plantean los siguientes objetivos particulares:

- Investigar el estado del arte.
- Diseñar y modelar las bases de datos.
- Diseñar y desarrollar la aplicación local.
- Captura de imágenes con agentes patógenos.

- Entrenamiento de la red neuronal con las categorías de agentes patógenos.
- Diseño y desarrollo del portal WEB para la visualización de eventos.
- Pruebas de predicción y operación de las aplicaciones.
- Implantación del sistema.

## **1.8. Hipótesis**

El desarrollo de herramientas de inteligencia artificial basadas en la técnica de redes neuronales para la detección de agentes patógenos en plantas ornamentales promueve la rentabilidad y competitividad de los viveristas.

## **1.9. Descripción de los métodos empleados**

Para el desarrollo de esta investigación, fue necesaria la integración de diferentes tecnologías innovadoras, se utilizó MATLAB como software principal en la modificación de la red neuronal ALEXNET para el reconocimiento de agentes patógenos, a su vez, usando el framework del lenguaje de programación Angular 5, se realizó la obtención de datos almacenados por el software principal para mostrar los datos de las imágenes en un mapa que ilustra la información del daño encontrado, para el control de tareas, se utilizó la metodología Kanban (Ballesteros, Diana et al, 2007), que ayude a llevar una secuencia de tareas que estén requeridas, las tareas que se están realizando y tareas terminadas.

## **1.10. Descripción de capítulos de tesis**

La presente tesis de maestría está estructurada como se describe a continuación. En el presente capítulo, se describen los antecedentes y bases para el inicio del proyecto. El Capítulo 2 describe el estado del campo de conocimiento, el cual contiene antecedentes y trabajos relacionados a esta investigación, que han ayudado a sustentar la investigación y descripción de los conceptos utilizados. En el Capítulo 3, se describen los métodos empleados en la investigación, en el cual se muestra brevemente la forma en que se ha desarrollado este trabajo, así como la metodología que ayudó a llevar un control de las tareas a realizar para lograr el objetivo. De igual manera, en el Capítulo 4 se presenta el desarrollo del proyecto de manera puntual donde se muestra la implementación y conclusión del proyecto, siguiendo la metodología expuesta. El Capítulo 5 muestra los resultados obtenidos en el proyecto, siendo la parte central de la investigación, ya que muestra datos crudos del entrenamiento de la red neuronal convolucional. Finalmente, Capítulo 6 aborda las conclusiones, trabajo a futuro y referencias.

## 2. Estado del campo del conocimiento

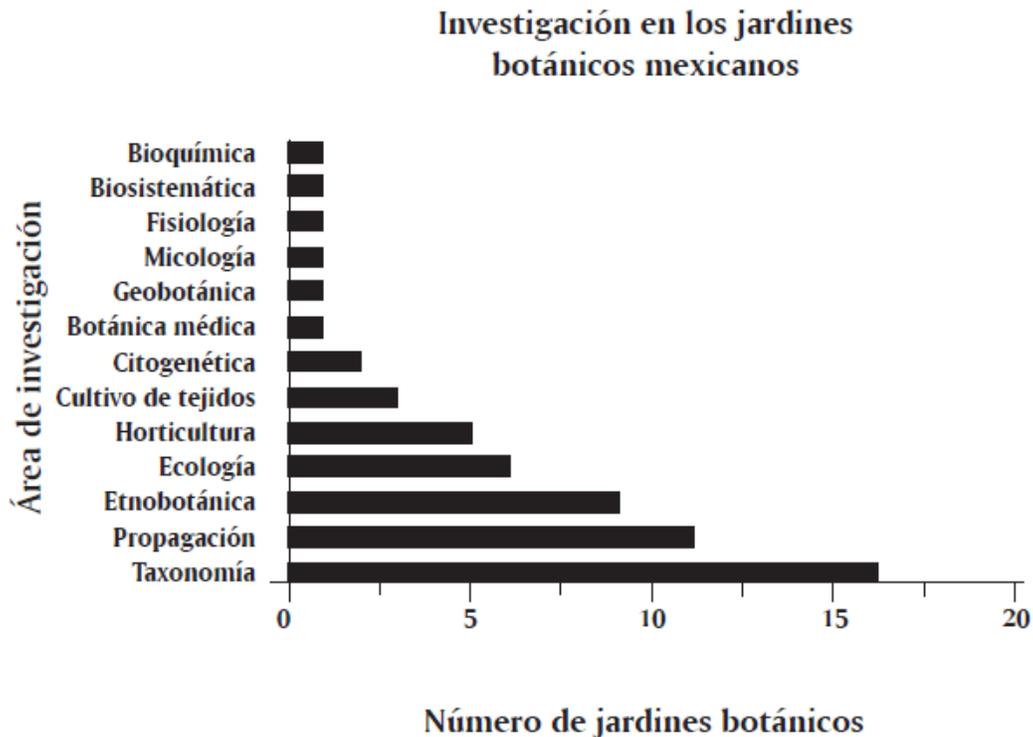
### 2.1. Marco histórico

De las primeras civilizaciones en el cultivo de hierbas medicinales, alimenticias y ornamentales, están consideradas el antiguo Egipto y Mesopotamia, en China, en el siglo XXVIII A.C. se cultivaban plantas medicinales para experimentos, posteriormente en el siglo VIII, en los monasterios cultivaban verduras, frutas y hierbas de distintas clases, que fue el inicio de los jardines físicos. El primer jardín moderno fue establecido en Pisa, Italia, patrocinado por la familia Medici, donde se utilizaban plantas cultivadas para examinar la taxonomía e incluyo ejemplares no medicinales, esta familia también apoyó para el jardín de Padua en el año 1545, siendo el más antiguo de una universidad, y considerado como patrimonio de la humanidad por la UNESCO (Vovides et al, 2010).

Con el paso de los años, se fue extendiendo esta práctica de jardines en su mayoría botánicos por todo Europa, Medio Oriente, Canadá y Norteamérica en donde fue creciendo la tendencia no solamente en plantas medicinales, sino además, agregaron tanto árboles y arbustos, por el año de 1900 se incrementó el uso de invernaderos de vidrio, que aumentaban las posibilidades de contar con ambientes tropicales incrementando la diversidad del cultivo en especies exóticas (Vovides et al, 2010).

En México desde la época prehispánica se tiene entendido el cultivo de flores, los toltecas y aztecas mantuvieron redes de jardines para abastecer de plantas a la clase dominante de Tenochtitlan, tanto plantas medicinales, ornamentales y aromáticas, considerándose así, el inicio de viveros en nuestro país (Vovides et al, 2010), entendiendo por vivero un espacio físico ubicado en un determinado lugar donde se producen y/o comercializan plantas ornamentales (Yanez, 2008). En la Figura 1, se ilustra el número de jardines botánicos mexicanos enfocados a diferentes propósitos al año 2010, se considera que el

número de jardines botánicos se ha incrementado a la fecha; sin embargo, después de una revisión de la literatura, no se encontró una referencia actualizada.



**Figura 1.** Jardines botánicos en México.(Vovides et al., 2010)

A medida que fue creciendo la comercialización de plantas y cultivo en viveros, fue creciendo la necesidad de contar con personal capacitado para poder sostener la producción y calidad en la producción, para esta tarea es importante destacar que la visión artificial juega un importante papel en la automatización del proceso.

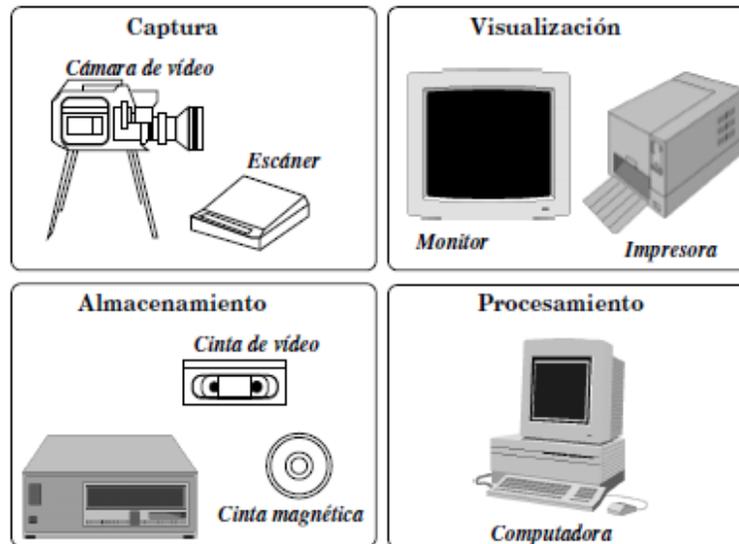
La visión artificial surge en los años 60's, en donde se conectaba una cámara de video a una computadora como modelo básico, Larry Roberts el creador de ARPANET (red de computadoras) creó un programa en el que un robot podía ver una estructura de bloques sobre una mesa, analizaba el contenido y la reproducía de otra perspectiva (René Shun, 2015), así demostró que la información visual se había mandado de la cámara para ser procesada por la computadora, a medida que han avanzado los años fueron creadas

varias técnicas de visión artificial, los primeros sistemas se basaron en imágenes binarias que procesaban bloques y píxeles, con el desarrollo de nuevos algoritmos se fue mejorando el reconocimiento hacia contornos de objetos y posicionamiento de una imagen, posteriormente, se realizó la visión por intensidad de grises operando en diferentes tipos de iluminación, permitiendo mejor reconocimiento en imágenes sin importar la intensidad de los píxeles.

De acuerdo con (René Shun, 2015), los componentes de un sistema de visión artificial, involucra los siguientes elementos:

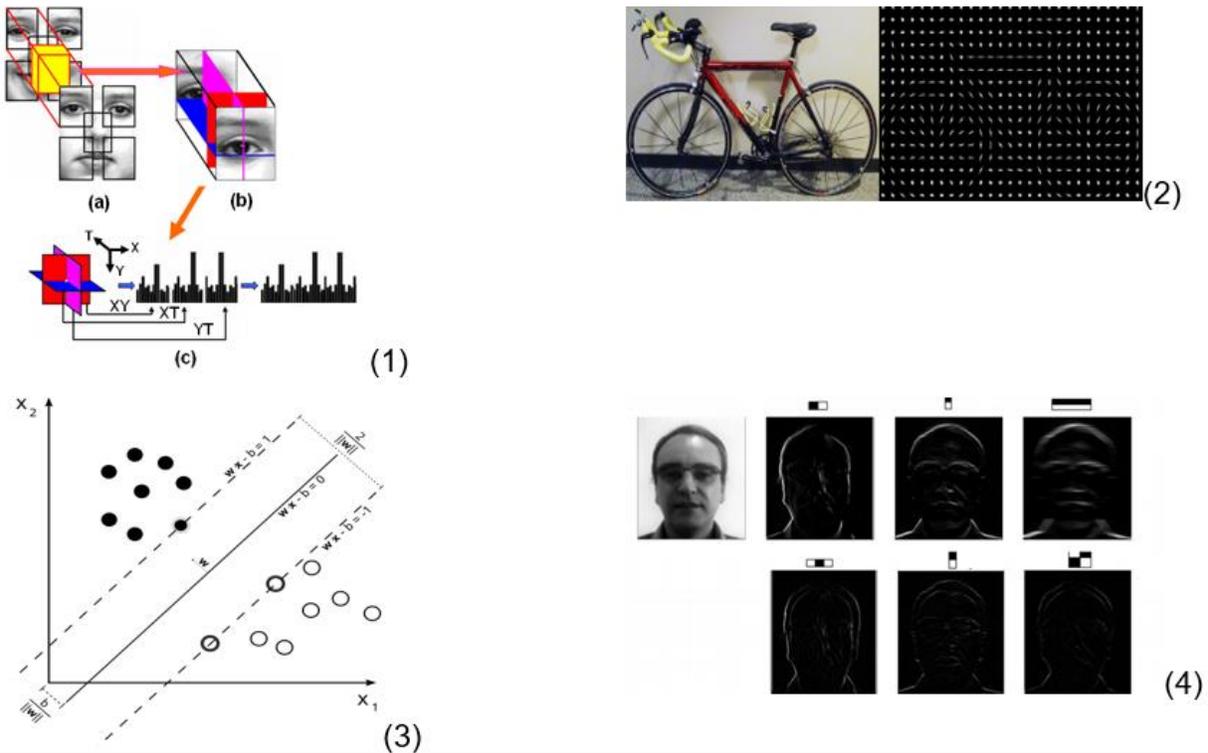
- La iluminación, que juega un papel importante para la detección de objetos.
- Cámaras, que dependen de sensores como los de carga de dispositivos acoplados (CCD) o circuitos con semiconductor complementario de óxido metálico (CMOS), componentes sensibles a la luz que modifican su señal eléctrica en función de la intensidad luminosa que perciben.
- Sistema de procesamiento, generalmente una computadora la cual recibe las imágenes e implementa las funciones para el tratamiento de la imagen.

En la Figura 2, se muestran los componentes de un sistema de visión artificial.



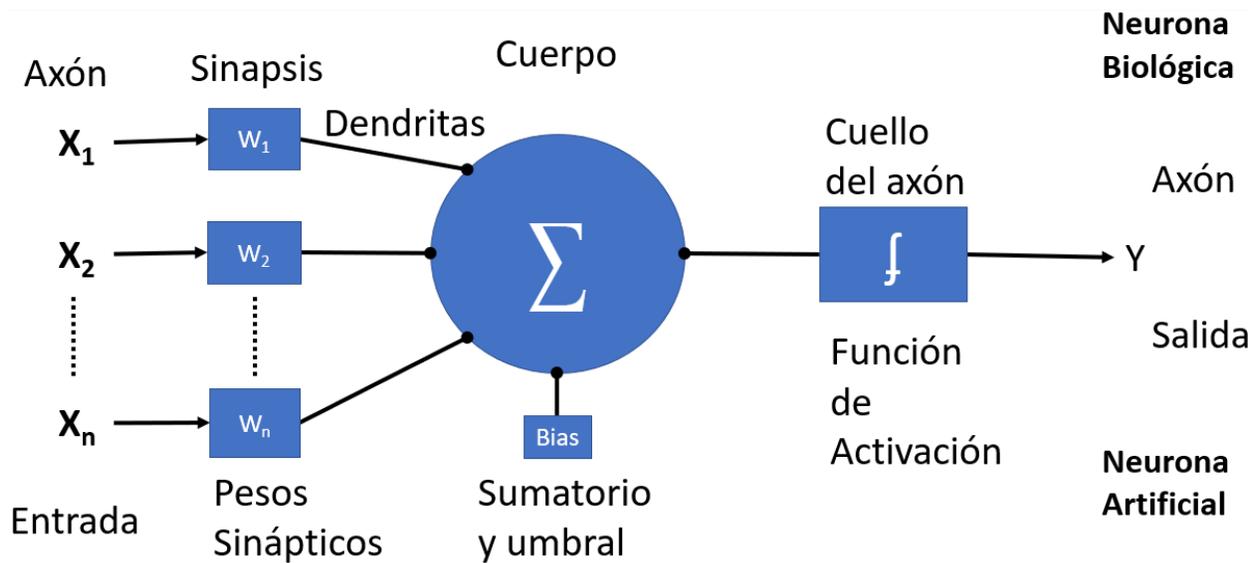
**Figura 2.** Sistema de visión artificial (Fernández García, 2012).

Conforme fueron avanzando los años, se realizaron mejoras tanto en la visión artificial como en la detección y clasificación de objetos o patrones en las imágenes, iniciando con técnicas como Local Binary Patterns (LBP), Histogram of Gradients (HOG), Regresión Logística, Ventana Deslizante, Pirámide, Imagen Integral, Cascada de Clasificadores, Filtros de Haar, Super Vector Machine (SVM), entre otras técnicas; Actualmente, con la evolución de estas técnicas y recursos informáticos, se han integrado las redes neuronales para la detección y/o clasificación de patrones o imágenes, ayudando a la precisión en la detección de objetos. En la Figura 3, se muestran las diferentes técnicas que se han utilizado en la detección de objetos como en la obtención de patrones de imágenes.



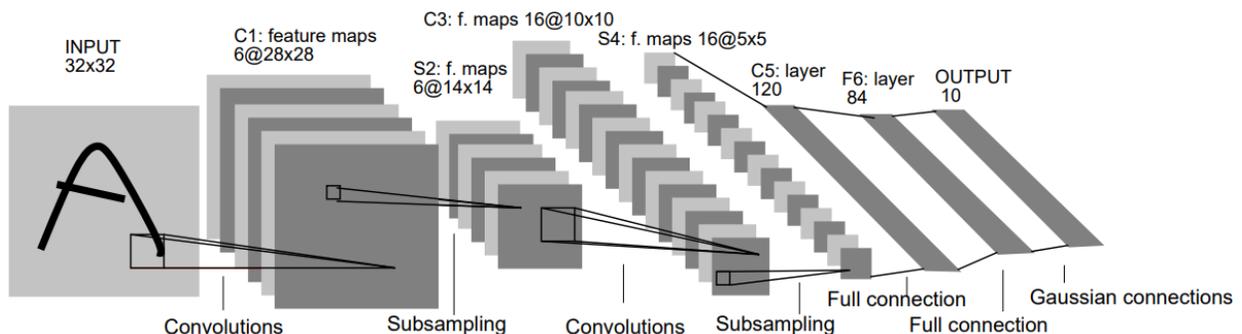
**Figura 3.** Diferentes técnicas para detección y clasificación de objetos, (1) LBP, (2) HOG, (3) SVM, (4) Filtros de HAAR.

De acuerdo con Haykin (2001), una red neuronal es una máquina que está diseñada para modelar la forma en que el cerebro realiza una tarea particular o función de interés. La red suele implementarse mediante el uso de componentes electrónicos o simulación de software. Las redes neuronales han tomado terreno en diferentes aplicaciones de la ciencia y la tecnología, tal es el caso en la visión artificial, las cuales van evolucionando la forma de aplicarse para poder predecir de manera óptima el objeto a detectar. En la Figura 4, se muestra la arquitectura de una neurona artificial, así como el comparativo en las partes que la componen con una neurona biológica.



**Figura 4.** Componentes de una neurona artificial y comparativa a una neurona biológica.

Dentro de las distintas redes neuronales que existen y las que se han probado para este fin, se encuentran las CNN, las cuales son utilizadas en esta investigación. Las CNN se basan en operaciones con matrices bidimensionales, en donde su arquitectura se puede representar con las neuronas convoluciones, neurona de reducción de muestreo y la neurona de clasificación. En la Figura 5, se muestra la arquitectura de una CNN, pionera en el reconocimiento de patrones para reconocimiento de dígitos, propuesta por Yann LeCun.



**Figura 5.** Arquitectura de la red neuronal convolucional LeNet-5 (Lecun et al., 1998).

Las CNN son buenas técnicas para aprender a clasificar datos distribuidos de forma continua en el mapa de entrada y siendo a su vez, estadísticamente similares en cualquier lugar del mapa de entrada, es por ello que son eficaces para la clasificación de imágenes (Wikipedia, 2018).

## 2.2 Marco contextual

En la investigación de Mehdipour Ghazi et al (2017), se usaron redes neuronales convolucionales profundas para identificar las especies de plantas capturadas en una fotografía y evaluar diferentes factores que afectan el rendimiento de estas redes. Se evaluaron tres potentes y populares arquitecturas de CNN's las cuales fueron: GoogLeNet, AlexNet y VGGNet. Además, utilizaron el aprendizaje de transferencia para afinar los modelos pre-entreados utilizando conjuntos de datos de LifeCLEF 2015. En las tabla 1, se muestran los datos utilizados en la investigación, y en la tabla 2, los resultados de la precisión de las CNN's.

**Tabla 1.** Detalles del conjunto de datos utilizados para la identificación de plantas.

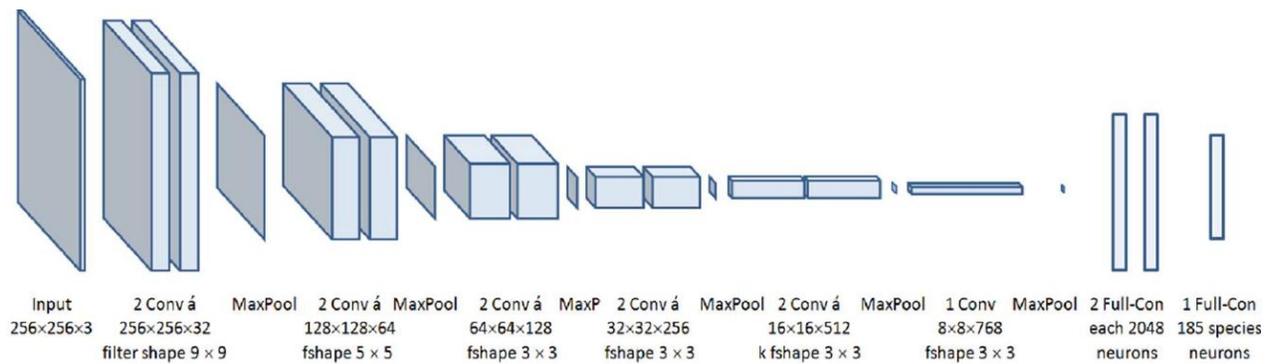
	Branch	Entire	Flower	Fruit	Leaf	LeafScan	Stem	Overall
Number of training samples	6,447	12,567	21,531	6,072	10,367	9,576	4,344	70,904
Number of validation samples	1,683	3,668	6,694	1,648	3,000	3,029	1,132	20,854
Number of testing samples	2,088	6,113	8,327	1,423	2,690	221	584	21,446
Number of species	891	993	967	755	899	351	649	1,000

**Tabla 2.** Porcentaje de precisión en la clasificación (%) en las diferentes redes neuronales convolucionales.

	Branch	Entire	Flower	Fruit	Leaf	LeafScan	Stem	Overall
AlexNet from scratch (200 K, 20, 10x)	23.53	27.32	45.49	35.50	33.87	92.11	33.66	44.19
AlexNet fine-tuning (100 K, 20, 10x)	28.64	30.10	53.45	43.99	45.03	90.99	31.18	49.63
GoogLeNet from scratch (200 K, 20, 10x)	17.35	16.90	36.41	24.82	20.63	83.13	13.16	33.78
GoogLeNet fine-tuning (100 K, 20, 10x)	44.09	38.36	<b>67.93</b>	57.65	60.57	94.16	37.01	61.06
VGGNet from scratch (200 K, 20, 10x)	16.93	16.47	32.12	22.69	20.23	81.91	13.43	31.90
VGGNet fine-tuning (100 K, 20, 10x)	<b>44.68</b>	<b>42.58</b>	66.37	<b>59.59</b>	<b>61.90</b>	<b>95.05</b>	<b>38.87</b>	<b>61.93</b>

Los valores en negrita en la tabla 2, indican los mejores resultados en cada categoría; sin embargo, en esta investigación se menciona que la red AlexNet si se entrena desde cero, supera a las otras dos arquitecturas debido a que está conformada por una arquitectura más simple (Mehdipour Ghazi et al, 2017).

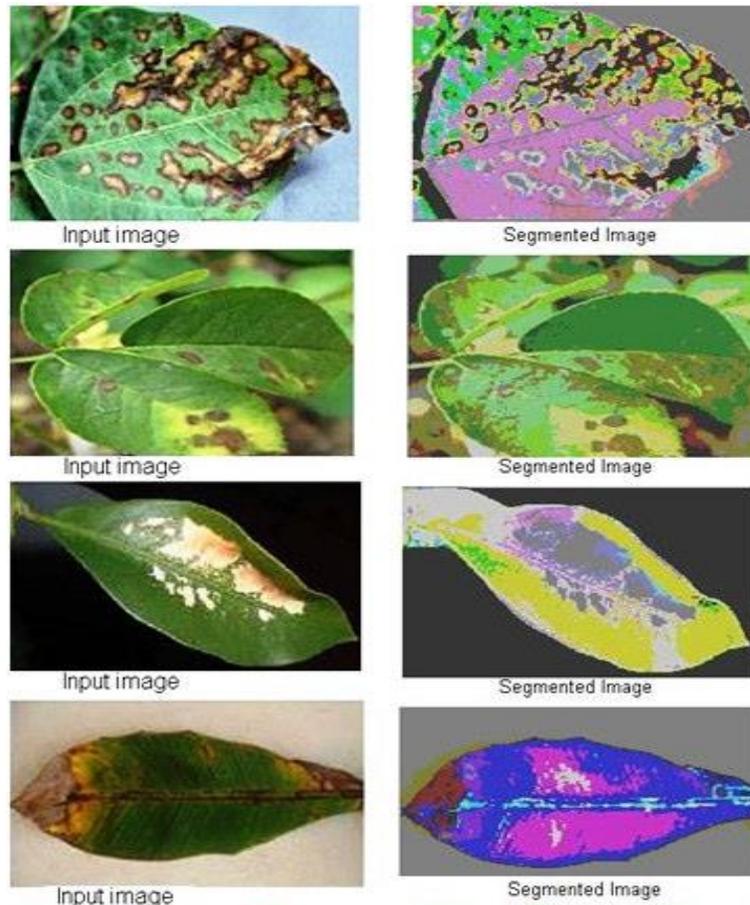
En el trabajo de Barré et al, (2017), sobre la identificación de taxonomía de la planta, se comenta que existen sistemas de identificación basados en imágenes, pero en su mayoría se basan en algoritmos hechos específicamente para resolver u obtener cierta taxonomía extrayendo conjuntos de características elegidas por juicio experto para identificar especies. En consecuencia, dichos sistemas están restringidos a estos datos de taxonomías y además requieren la participación de expertos que brinden conocimientos taxonómicos para desarrollar dichos sistemas personalizados. El objetivo principal de esta investigación fué desarrollar un sistema de aprendizaje profundo para aprender las características taxonómicas de las imágenes de las hojas, junto con un clasificador para la identificación de las especies de las plantas llamado LeafNet. En la Figura 6, se muestra la arquitectura de la CNN utilizada en el clasificador.



**Figura 6.** Arquitectura de LeafNet.

Los enfoques de aprendizaje profundo con las CNN ofrecen una alternativa de autoaprendizaje de vanguardia que permite la adaptación a diferentes clasificaciones, simplemente presentando nuevos datos de capacitación en lugar de desarrollar nuevos sistemas de software (Barré et al, 2017).

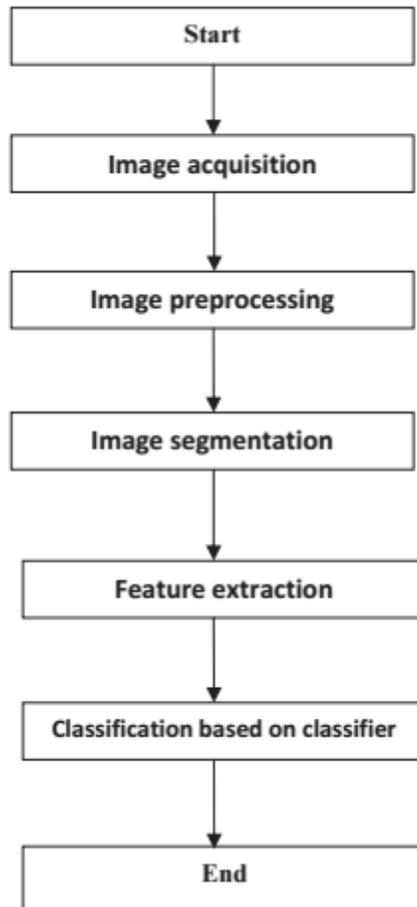
En (Singh & Misra, 2017) se presenta un algoritmo para la técnica de segmentación de imágenes, que se utiliza para la detección y clasificación automática de enfermedades de las hojas de las plantas. Se realizó una encuesta sobre diferentes técnicas de clasificación de enfermedades que ayudó en el desarrollo del algoritmo para detectar estos patrones en las plantas. La segmentación de imágenes es un aspecto importante para la detección de enfermedades de la hoja. En la Figura 7, se muestran los resultados de la aplicación del algoritmo en algunas muestras.



**Figura 7.** Imágenes de entrada y salida con la aplicación del algoritmo.

Gupta et al, (2017), especifican que la identificación de enfermedades en las plantas es clave para prevenir las pérdidas en el rendimiento del producto agrícola. La identificación de estas enfermedades es una tarea difícil de realizar manualmente. Por lo tanto, ellos proponen un proyecto que implica una arquitectura de sistema que permite al usuario lograr todas las actividades anteriores en tiempo real, permitiendo que los agricultores puedan ver los detalles de su granja desde una ubicación remota. Este incluye un módulo colocado en una granja que contiene varios sensores y dispositivos para la conversión y transferencia de datos. Además, incluye otro módulo para el procesamiento de imágenes para la detección de enfermedades de los síntomas visuales de la planta.

En la Figura 8, se muestra el diagrama de flujo utilizado en el proyecto para la identificación de enfermedades.



**Figura 8.** Diagrama de flujo de la detección de daño en el procesamiento de imágenes.

En el trabajo realizado por Robles-Camarillo et al (2016), se muestra una comparación entre dos arquitecturas utilizadas en la competencia de reconocimiento de imágenes llamado ImageNet Large Scale Visual Recognition Competition (ILSVRC). Utilizando el reentrenamiento de las redes neuronales convolucionales AlexNet y VGGNet en la clasificación de imágenes de ojos, y obtener como resultado cual de las dos redes obtiene mejor predicciones. Concluyendo que la red AlexNet, obtuvo la mejor precisión en la predicción de imágenes con un 75% a comparación de la red VGGNet con el 50%

de precisión. En la Figura 9, se presentan las matrices de confusión de las dos CNN involucradas y sus resultados.

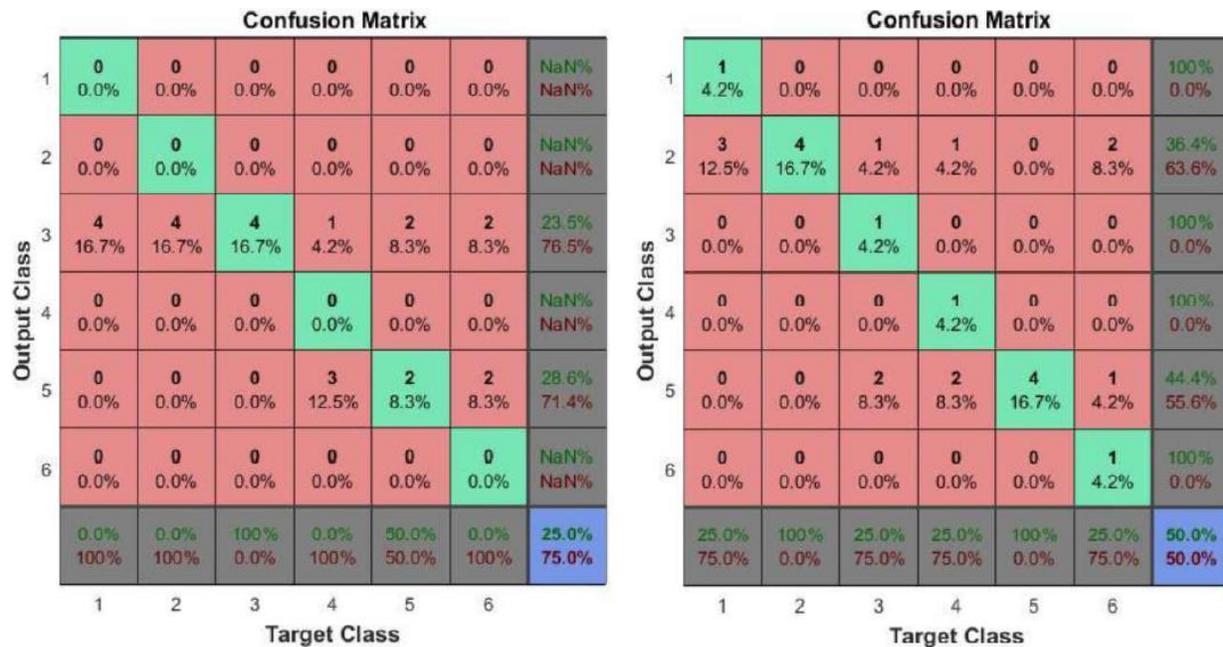
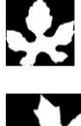
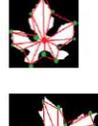
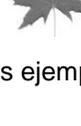
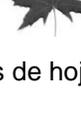
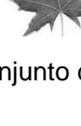
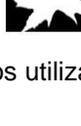
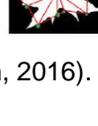


Figura 9. Matrices de confusión de las CNN AlexNet y VGGNet respectivamente.

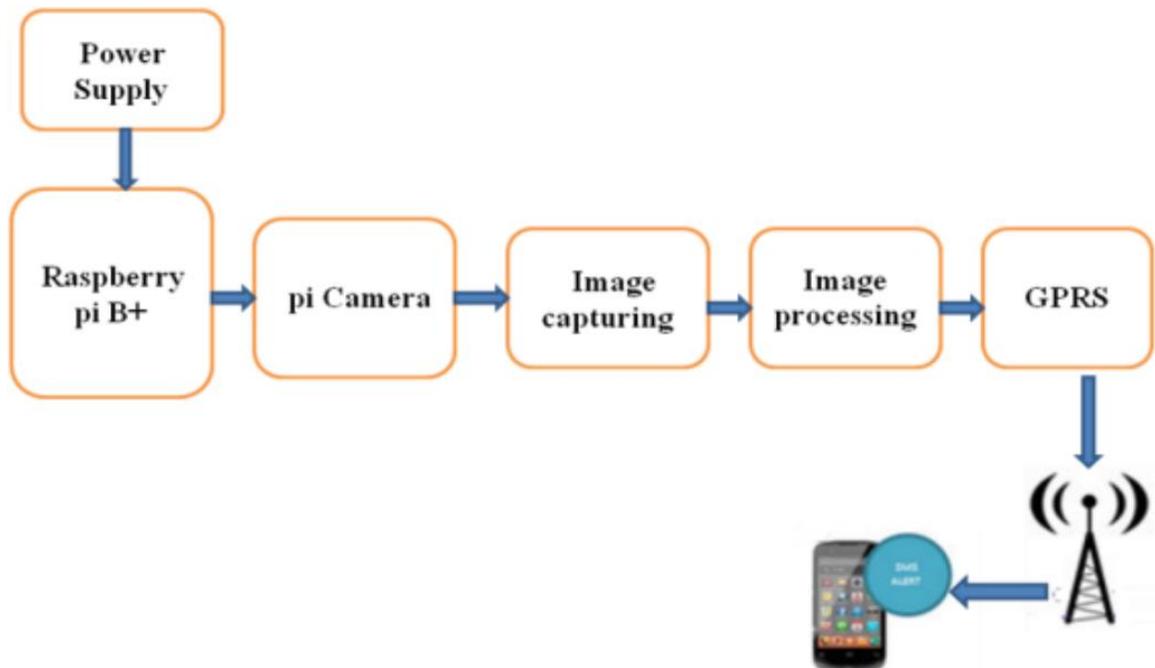
Los autores (VijayaLakshmi & Mohan, 2016), proponen un enfoque para la clasificación de hojas, basado en la caracterización de las propiedades de textura, forma y color. Una hoja original de la planta se pre-procesa inicialmente utilizando el filtro de autómatas celulares (CA) para minimizar el ruido. Para mejorar el contraste y la calidad de la imagen, la ecualización de histograma y la segmentación de las regiones de interés (ROI) se aplican respectivamente. Para analizar la textura en la imagen se usó la técnica matriz de co-ocurrencia de nivel de gris (GLCM) y LBP para la extracción de características. Las características se extraen de cada imagen de hoja, lo que aumenta la complejidad del tiempo. Posteriormente, se presenta el algoritmo Particle Swarm Optimization (PSO) basado en kernel para mejorar el problema anterior de seleccionar las características óptimas. Finalmente, se utiliza la Máquina de Vector de Relevancia Fuzzy (FRVM) para

caracterizar el tipo de hojas. En la Figura 10 se muestra el ángulo de predicción de cada hoja para su clasificación.

S. No	Leaf name	Original image	Cellular automata result	Image enhancement	ROI extraction	Background normalization	Feature extraction	Angle prediction
1.	Gingkgo							
2.	Paper Mulberry							
3.	Mono Maple							

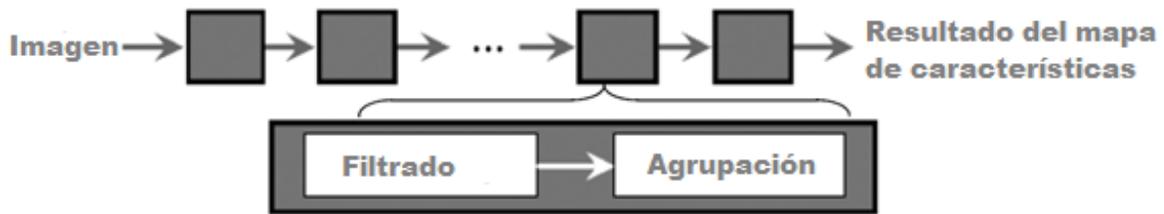
**Figura 10.** Tres ejemplos de hojas de conjunto de datos utilizados en (VijayaLakshmi & Mohan, 2016).

Como en otros trabajos, en la investigación de Chopade & Katkar (2016), también consideran que la detección automática de enfermedades foliares es un tema de investigación muy importante en el campo de la agricultura, ya que puede ser beneficioso para monitorear y controlar grandes campos de cultivos, y de esta manera detectar enfermedades automáticamente tan pronto como aparezcan en las hojas de las plantas. Su trabajo proporciona una descripción de la detección de enfermedades foliares mediante el procesamiento de imágenes que puede reconocer problemas en cultivos a partir de imágenes, en función del color, la textura y la forma para detectar enfermedades automáticamente y brindar soluciones rápidas y precisas al agricultor. La arquitectura propuesta por estos autores, utilizando una Raspberry pi B+ para capturar las imágenes, procesarlas y finalmente, enviar por GPRS los resultados de la enfermedad a un teléfono, como se muestra en la Figura 11.



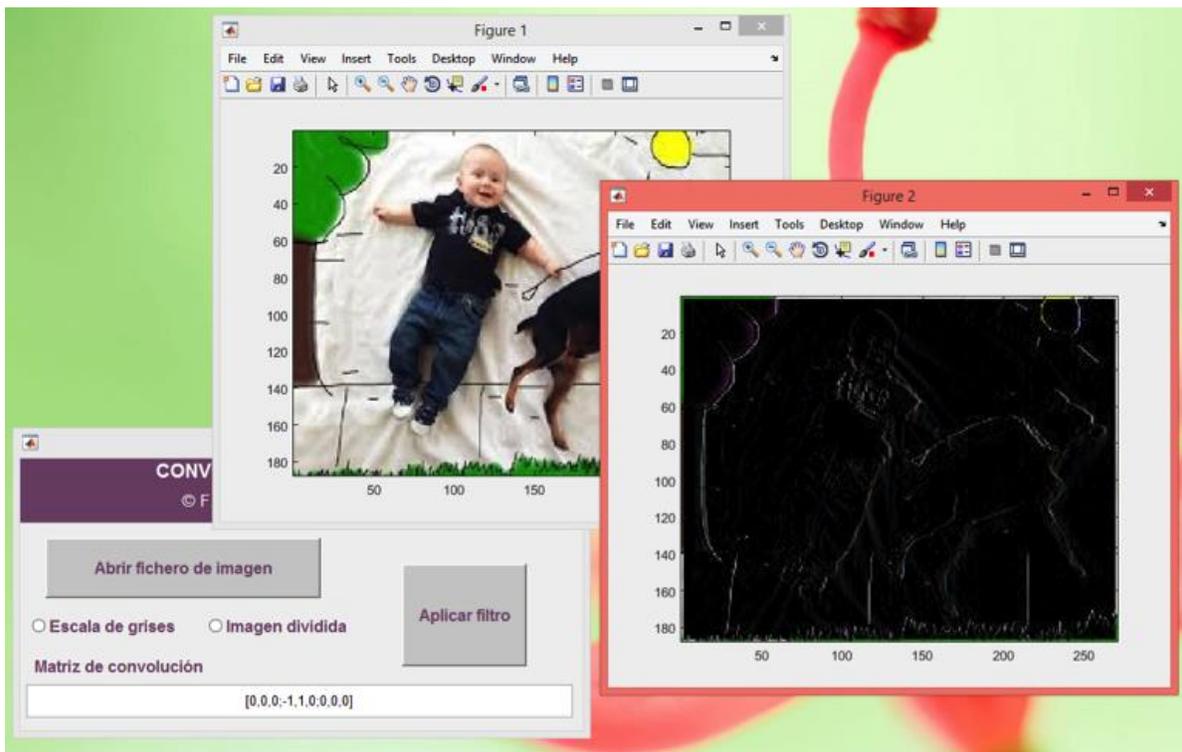
**Figura 11.** Arquitectura del sistema para la detección de daño y clasificación de hojas.

La investigación de (Druzhkov & Kustikova, 2016), se centra en analizar los métodos de aprendizaje profundo para la clasificación de imágenes y la detección de objetos. En particular, consideran modelos de aprendizaje profundos como autoencoders, máquinas Boltzmann restringidas y redes neuronales convolucionales. En la Figura 12, se muestra la estructura principal de una CNN. El proceso inicial es convertir la imagen en matrices de datos (mapas de características) y procesarla en filtros (convolucion de matrices), seguida de una función de activación entre cada filtro; finalmente, la agrupación es una transformación posiblemente no lineal que permite resumir a un valor las descripciones de características más robustas para obtener un resultado del mapa de características.



**Figura 12.** Estructura de una red neuronal convolucional.

En (Palomares et al, 2016), se presentan aplicaciones importantes en donde se utiliza la convolución de matrices, las cuales se aplica a numerosos campos de la ciencia y la tecnología. El filtrado de imágenes que muestran, consiste en la modificación de una imagen donde se mejoran características visibles en ellas para obtener información sobresaliente. En la Figura 13, se muestra la aplicación funcionando creada por estos autores, donde se puede apreciar los resultados de los filtros.



**Figura 13.** Ejemplo de la aplicación en funcionamiento.

En otra investigación sobre las enfermedades de las plantas, se plantea que las enfermedades de éstas pueden afectar a los humanos directa o indirectamente, ya sea en cuestiones de salud o económicamente. Por lo cual, para detectar estas enfermedades de las plantas, se requiere de un proceso automatizado, rápido y eficaz. En este trabajo, las enfermedades se analizan mediante diferentes técnicas de procesamiento de imágenes digitales. Además, se realizó un estudio de las diferentes técnicas de procesamiento de imágenes digitales para detectar las enfermedades de las plantas, las cuales se presentan en la Tabla 3 (Varshney, 2016).

**Tabla 3.** Lista de autores y trabajos realizados con los resultados obtenidos.

<b>Autores</b>	<b>Algoritmos de detección usados</b>	<b>Resultados</b>
S. Phadikar & et al.	Baye's y clasificador SVM	Baye's = 68.1 % y SVM = 79.5%
Ging Yao & et al.	SVM	97.2 % en enfermedades de la planta de arroz.
S. Arivazhagan & et al.	Color co-ocurrencia y SVM	94% en 500 hojas de plantas
Stephen gangWu et al.	Red neuronal probabilística (PNN)	90% en 32 clases de plantas
Smita Naikwadi & et al.	K-mean y método Otsu's	Entre 83% a 94%
Ajay A. Gurjar & et al.	Descomposición normalizada generalizada(RGED) y técnicas de extracción	90% en enfermedades de hongos

En cuanto al proceso de reconocimiento de imágenes, (Gando et al., 2016), realizaron un modelo para implementar la funcionalidad del reconocimiento con técnicas básicas como: maquinas de vector soporte e histogramas, que se consideraron útiles para la clasificación, logrando una precisión del 58%. Sin embargo, al utilizar métodos como el Aprendizaje Profundo, técnica que ha tenido éxito en las tareas de visión por computadora, y las CNN, las cuales obtienen un buen resultado al extraer automáticamente tales características de las imágenes, se obtuvo una mejora en la red neuronal convolucional profunda (DCNN), con una precision del 96,8%, superando a los otros modelos. Concluyen que una DCNN con ajuste fino es el mejor método para para

distinguir automáticamente las ilustraciones de las fotografías. En la tabla 4, se muestran los resultados de los modelos.

**Tabla 4.** Resultados de todos los modelos en los conjuntos de datos de imágenes.

Modelo	Precisión
Color intensity	0.577
Outline detection	0.437
Color histogram (RGB) + SVM	0.842
Color histogram (HSV) + SVM	0.499
Bag of Words (SIFT + SVM)	0.5
1-layer CNN	0.863
2-layer CNN	0.908
3-layer CNN	0.892
4-layer CNN	0.911
5-layer CNN	0.93
AlexNet (without BN)	0.832
AlexNet (with BN)	0.92
AlexNet (fine-tuned)	0.968

Otra investigación sobre las CNN donde se lograron muy buenos resultados en el campo de la clasificación de imágenes es en el trabajo de Sladojevic et al (2016), donde describen un nuevo enfoque para el desarrollo del modelo de reconocimiento de enfermedades de las plantas, el cual está basado en la clasificación de las imágenes de las hojas mediante el uso de redes convolucionales profundas. En éste, se logró reconocer 13 tipos diferentes de enfermedades de plantas a partir de hojas sanas, con la capacidad de distinguir las hojas de las plantas de su entorno. Todos los pasos esenciales requeridos para implementar este modelo de reconocimiento de enfermedades se describen en detalle a lo largo del documento, comenzando con la recopilación de imágenes para crear una base de datos evaluada por expertos agrícolas. En este trabajo utilizaron una red CNN llamada CaffeNet para el reconocimiento, en los

resultados obtenidos que se muestran en la Figura 14 hacen comparación de las imágenes y el resultado de la precisión de predicción.

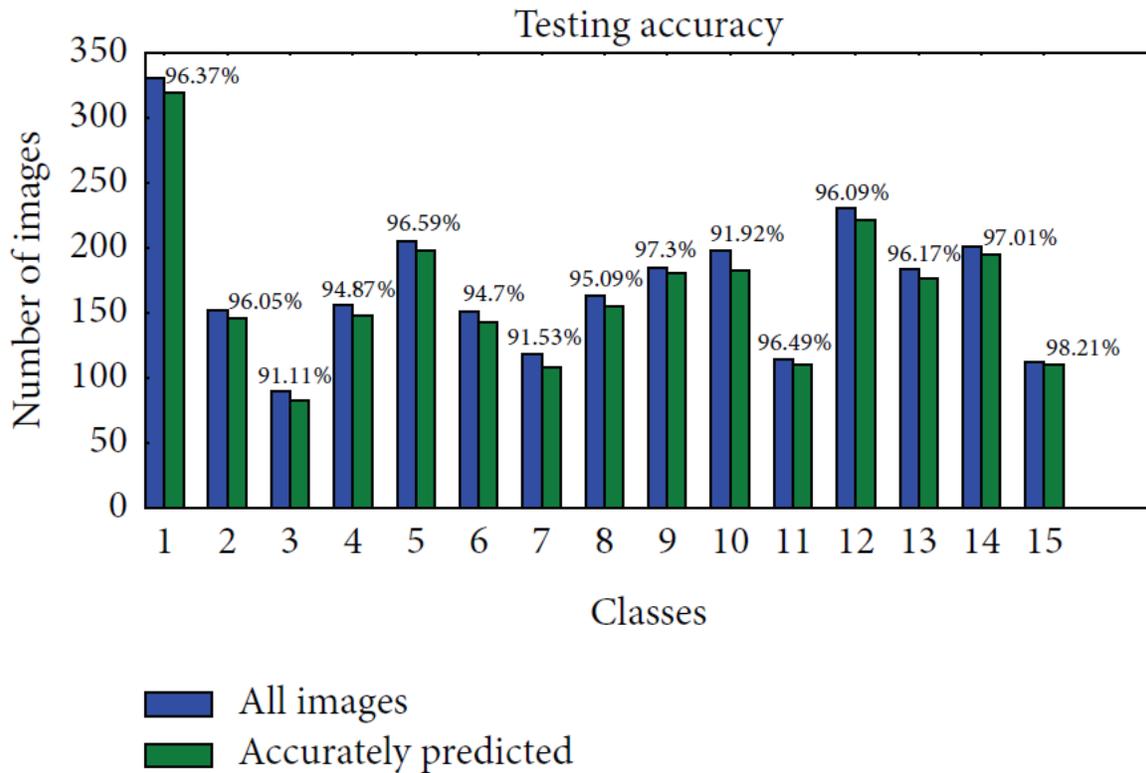


Figura 14. Precisión de predicción para cada clase por separado.

En el trabajo de (Lee et al, 2016), se describe la arquitectura de un sistema de clasificación de plantas creado por ellos mismos para el desafío LifeClef 2016. Donde su objetivo fue el identificar 1000 especies de imágenes de plantas correspondientes a 7 diferentes órganos de la planta (tallos, flores, ramas o frutos), así como detectar automáticamente especies de clases desconocidas. Se propuso un sistema de clasificación de plantas que utiliza una CNN. En la Figura 15 se muestra la arquitectura propuesta en esta investigación. En donde se muestra el proceso de la imagen de entrada, con dimensiones de 224x224, realizando filtros convolucionales de bajo nivel para posteriormente pasar por la extracción de características de estructura orgánica y en que especie se clasifica para obtener el resultado de la predicción.

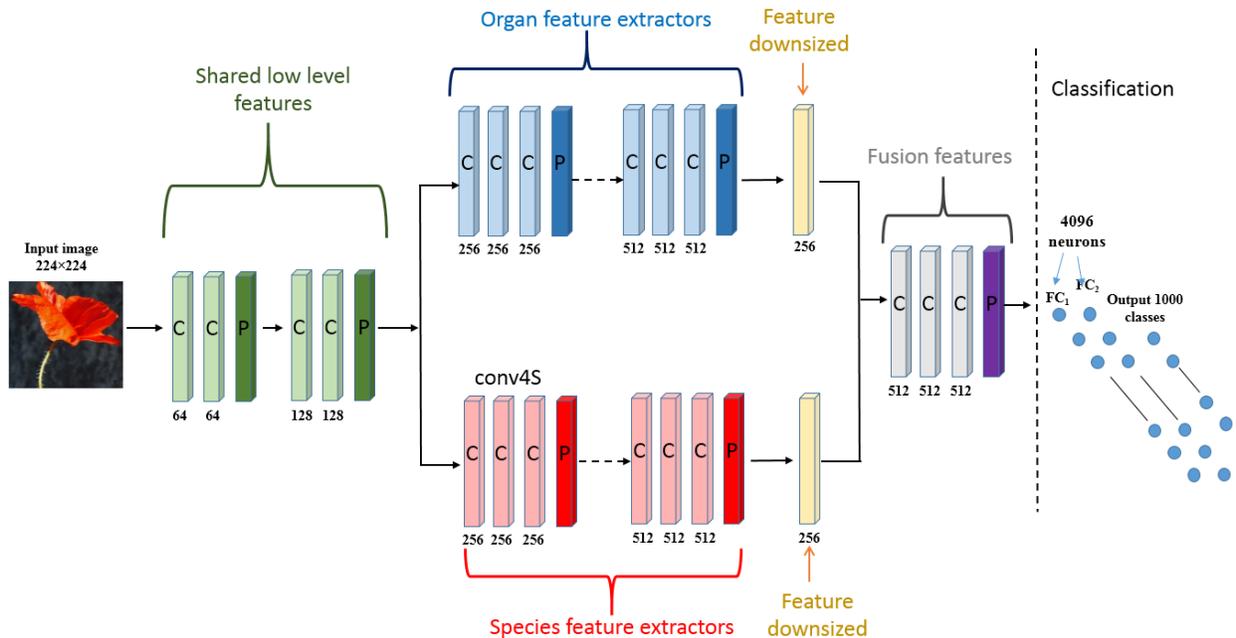
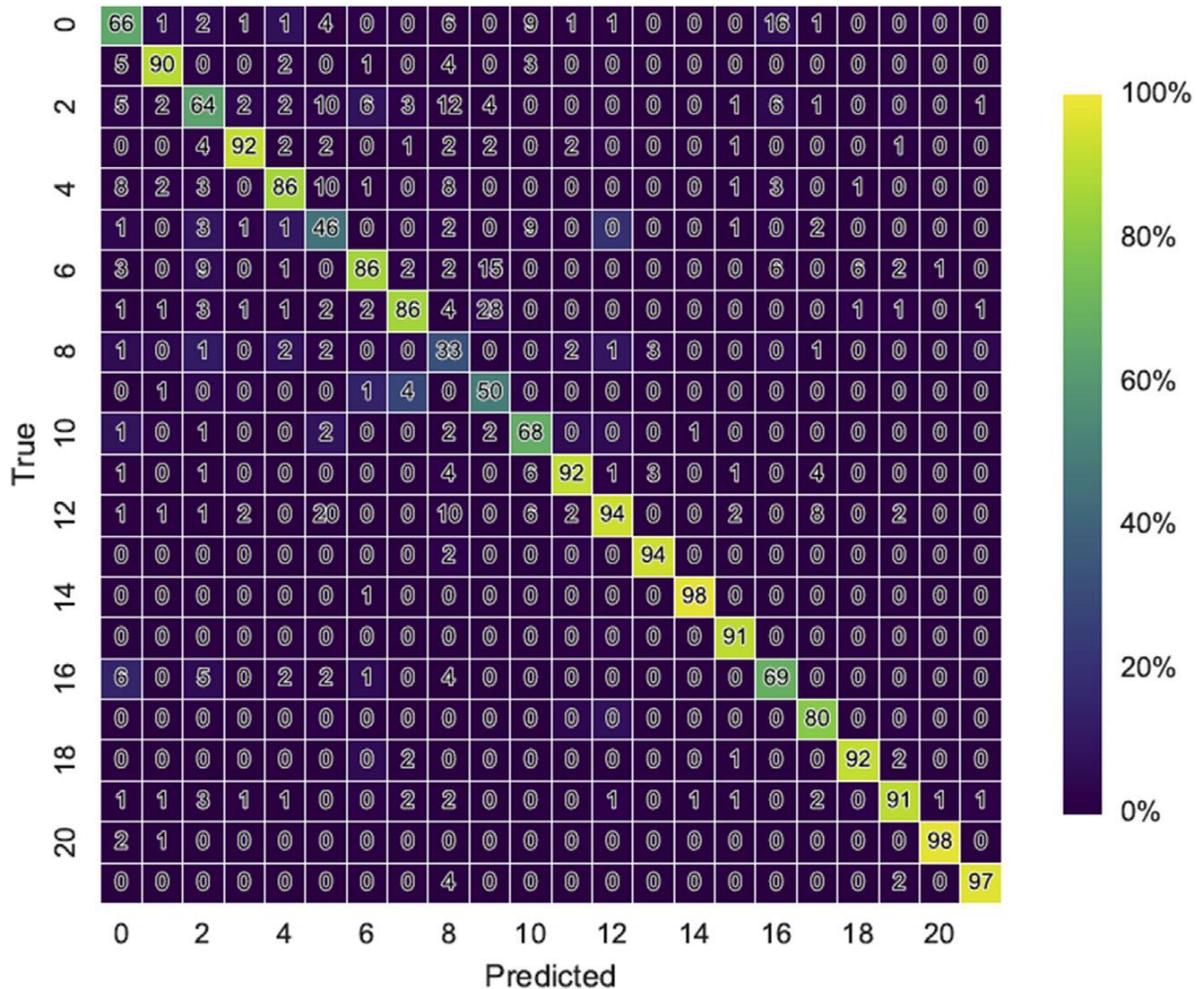


Figura 15. Arquitectura propuesta de la red neuronal convolucional.

Este artículo presentado por (Dyrmann et al, 2016), realiza un método que es capaz de reconocer especies de plantas en imágenes de color mediante el uso de una red neuronal convolucional. La red está construida desde cero, entrenada y probada con un total de 10,413 imágenes que contienen 22 especies de malezas y cultivos en etapas tempranas de crecimiento. Estas imágenes se originan a partir de seis conjuntos de datos diferentes, que tienen variaciones con respecto a la iluminación, la resolución y el tipo de suelo. Esto incluye imágenes tomadas bajo condiciones controladas con respecto a la estabilización e iluminación de la cámara, y las imágenes tomadas con teléfonos móviles de mano en campos con condiciones de luz cambiantes y diferentes tipos de suelo. Para estas 22 especies, la red puede lograr una precisión de clasificación del 86.2%. En la Figura 16,

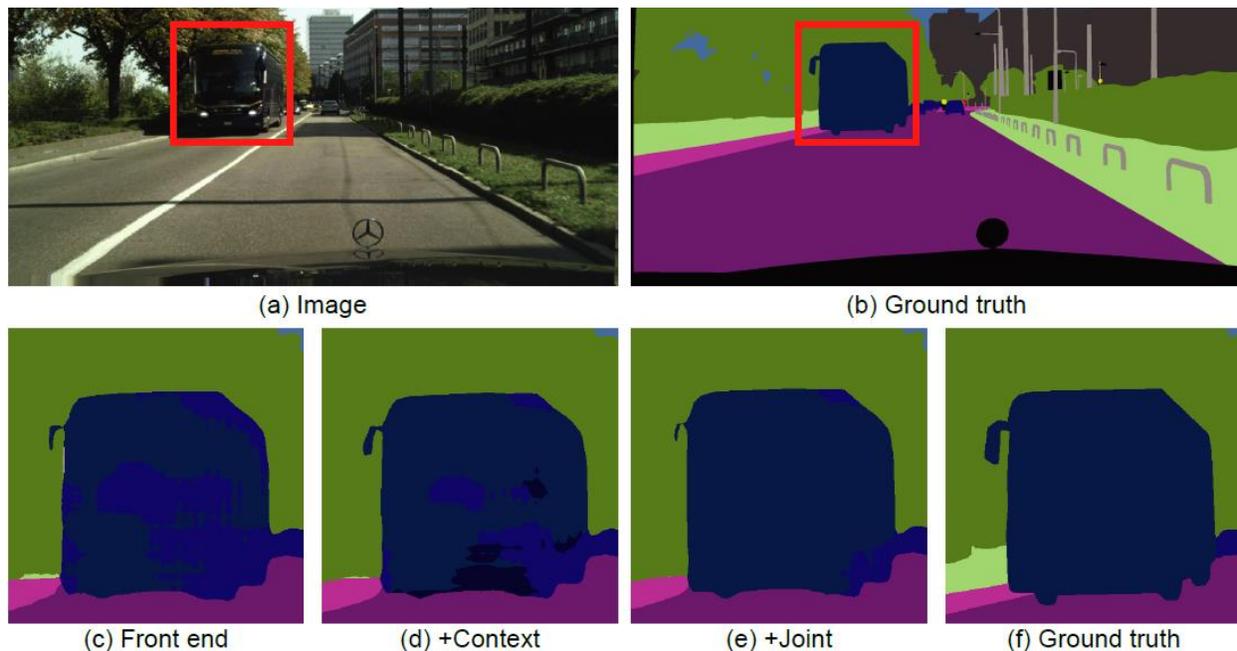
se muestra la matriz de confusión de la clasificación, validando la predicción con 21 especies de plantas y obteniendo resultados óptimos en cada una de ellas.



**Figura 16.** Matriz de confusión para la validación de la predicción con 21 especies de plantas. Los números de la diagonal, indican el porcentaje de clasificaciones correctas para cada clase.

Los autores Yu & Koltun (2016), expresan que los modelos de vanguardia para la segmentación semántica se basan en adaptaciones de redes convolucionales que originalmente se habían diseñado para la clasificación de imágenes. Sin embargo, la predicción densa y la clasificación de imágenes son estructuralmente diferentes. Es por ello, que desarrollaron un nuevo módulo de red convolucional que está específicamente diseñado para la predicción densa. El módulo presentado utiliza convoluciones dilatadas

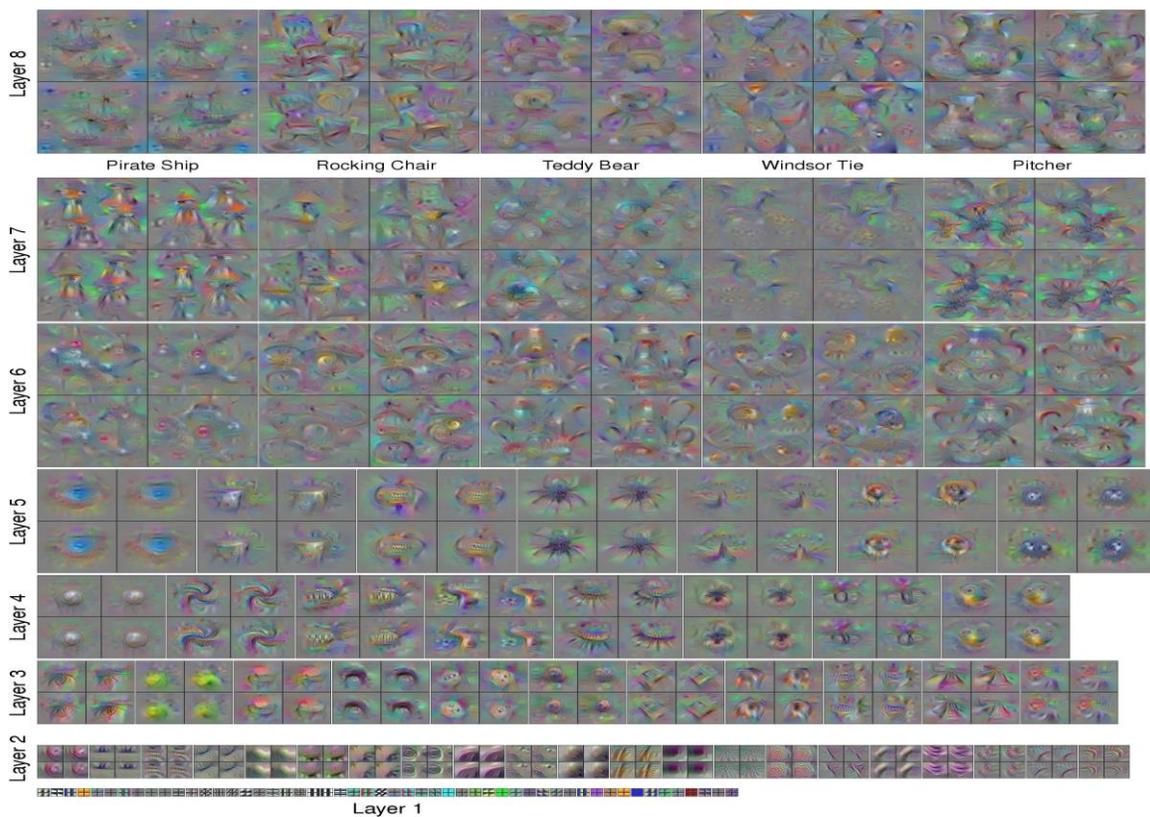
para agregar sistemáticamente información contextual a múltiples escalas sin perder resolución. La arquitectura se basa en el hecho de que las convoluciones dilatadas soportan la expansión exponencial del campo receptivo sin pérdida de resolución o cobertura. Muestran que el módulo presentado aumenta la precisión de los sistemas de segmentación semántica de última generación. Además, examinan la adaptación de las redes de clasificación de imágenes a la predicción densa y muestran que la simplificación de la red adaptada puede aumentar la precisión. Un ejemplo del resultado de esta investigación se muestra en la Figura 17.



**Figura 17.** Resultados del modelo de dilatación en diferentes escenarios del entrenamiento.

En el trabajo de Yosinski et al, (2015), se presentan dos herramientas; la primera es una herramienta que visualiza las activaciones producidas en cada capa de una red neuronal convolucional, la cual se entrena mientras se procesa una imagen o video (ej. La transmisión de una cámara WEB en vivo); la segunda permite visualizar las características en cada capa de un DNN a través de la optimización regularizada en el espacio de la imagen. Con esto, los autores descubrieron que mirar activaciones en vivo

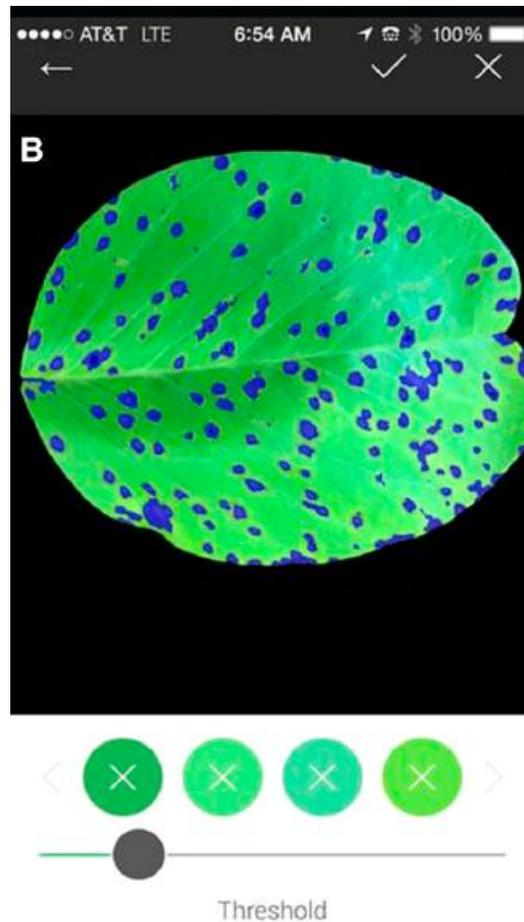
que cambian en respuesta a la entrada del usuario ayuda a construir intuiciones valiosas sobre cómo funcionan las redes convolucionales. Además, ambas herramientas son de código abierto y funcionan en una red convolucional preestablecida con una configuración mínima. En la Figura 18, se muestran resultados de características de las capas de una red neuronal convolucional profunda.



**Figura 18.** Visualización de características de ejemplo de ocho capas de una red neuronal convolucional profunda.

Pethybridge & Nelson (2015), realizaron una aplicación para teléfonos inteligentes la cual interactúa con imágenes en color para distinguir los tejidos de las plantas enfermas de las sanas y calcular el porcentaje de la gravedad de la enfermedad. El usuario toca la pantalla de visualización de la aplicación para seleccionar hasta ocho colores diferentes que representan tejidos sanos. El usuario mueve un control deslizante de umbral hasta que solo los tejidos sintomáticos se hayan transformado en un tono azul. La imagen

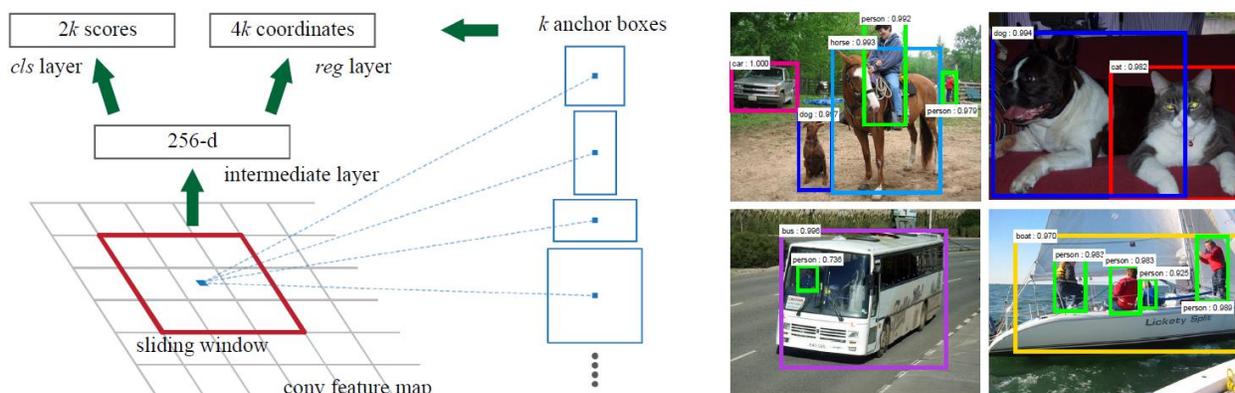
pixelada luego se analiza para calcular el porcentaje de la enfermedad. Este estudio informa la exactitud, precisión y robustez de la aplicación creada llamada “Leaf Doctor” usando seis enfermedades diferentes con lesiones típicas de gravedad variable. El trabajo finalizado se muestra en la Figura 19.



**Figura 19.** Ejemplo de Leaf Doctor operando.

Las redes de detección de objetos abstractas y de última generación, dependen de los algoritmos de regiones de interés (ROI) para las ubicaciones de los objetos. Avances como Spatial Pyramid Pooling (SPPnet) y Fast Region-CNN han reducido el tiempo de ejecución de estas redes de detección, exponiendo el cálculo de propuesta de región como un cuello de botella. En el trabajo de Shaoqing Ren et al, (2015), presenta una red neuronal de regiones de interés (RPN) que comparte características convolucionales de

imagen completa con la red de detección, lo que permite propuestas de regiones cercanas. Un RPN es una red totalmente convolucional que simultáneamente predice los límites de objeto y los puntajes de objetividad en cada posición. El RPN está entrenado de extremo a extremo para generar propuestas de regiones de alta calidad, que son utilizadas por Fast R-CNN para la detección. Además, los autores fusionan RPN y Fast R-CNN en una única red al compartir sus características convolucionales: utilizando la terminología popular de las redes neuronales con mecanismos de "atención", el componente RPN le dice a la red unificada dónde buscar. En la Figura 20 se muestra la red propuesta y el uso de esta técnica en funcionamiento, seleccionando varios objetos en una sola imagen.



**Figura 20.** Izquierda: red de propuesta de región (RPN). Derecha: ejemplos de detecciones usando propuestas de RPN.

En el trabajo realizado por (S. Singh & Bhamrah, 2015), se propone un método para la extracción de las características de las hojas basado en su forma. Un clasificador realizado con una red neuronal artificial (ANN), está entrenado para identificar la clase de hoja exacta. Este trabajo ha sido probado para comprobar la precisión de la red con diferentes combinaciones de características de imagen proporcionando un 98.8% de precisión con un mínimo de siete características de entrada (S. Singh & Bhamrah, 2015).

En la Figura 21, se muestran datos en una matriz de confusión de los resultados en el trabajo realizado.

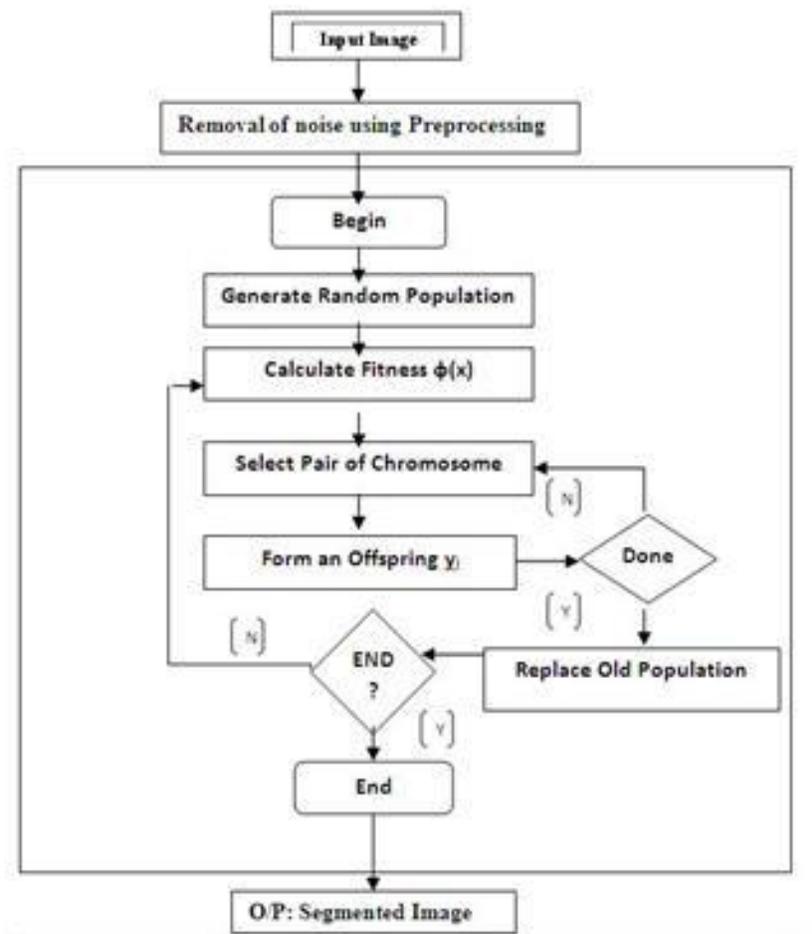
**Confusion Matrix**

<b>Output Class</b>	1	9 11.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	10 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	10 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	1 1.3%	0 0.0%	0 0.0%	10 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.9% 9.1%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 12.5%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 12.5%	0 0.0%	100% 0.0%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 12.5%	100% 0.0%
			90.0% 10.0%	100% 0.0%						
		1	2	3	4	5	6	7	8	
		<b>Target Class</b>								

**Figura 21.** Matriz de confusión mostrando la precisión del identificador.

V. Singh et al, (2015), establecen que la detección de enfermedades de las plantas a través de una técnica automática es beneficiosa, ya que reduce un gran trabajo de monitoreo en grandes granjas de cultivos, y en las primeras etapas detecta los síntomas de las enfermedades cuando aparecen en las hojas de las plantas. Este trabajo presenta

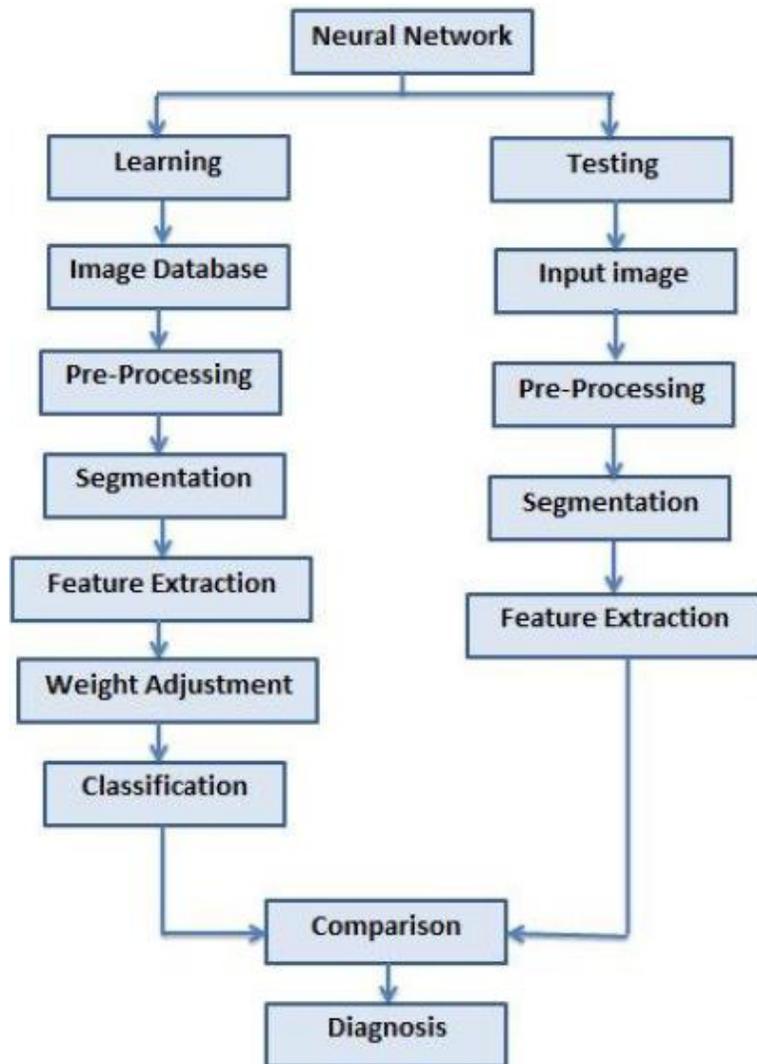
un algoritmo para la técnica de segmentación de imágenes utilizada para la detección automática, así como la clasificación de enfermedades de las hojas de plantas. La segmentación de imágenes, que es un aspecto importante para la detección de enfermedades en las hojas de las plantas, se realiza mediante el uso de algoritmos genéticos, mostrando en la Figura 22 un diagrama de la metodología propuesta.



**Figura 22.** Diagrama de flujo de la metodología propuesta.

La detección de enfermedades implica pasos como: la adquisición de imágenes, el preprocesamiento de imágenes, la segmentación de imágenes, la extracción de características y la clasificación. En esta investigación se discutieron los métodos utilizados para la detección de enfermedades de las plantas utilizando sus hojas. Además, se discutió algunos algoritmos de segmentación y extracción de características

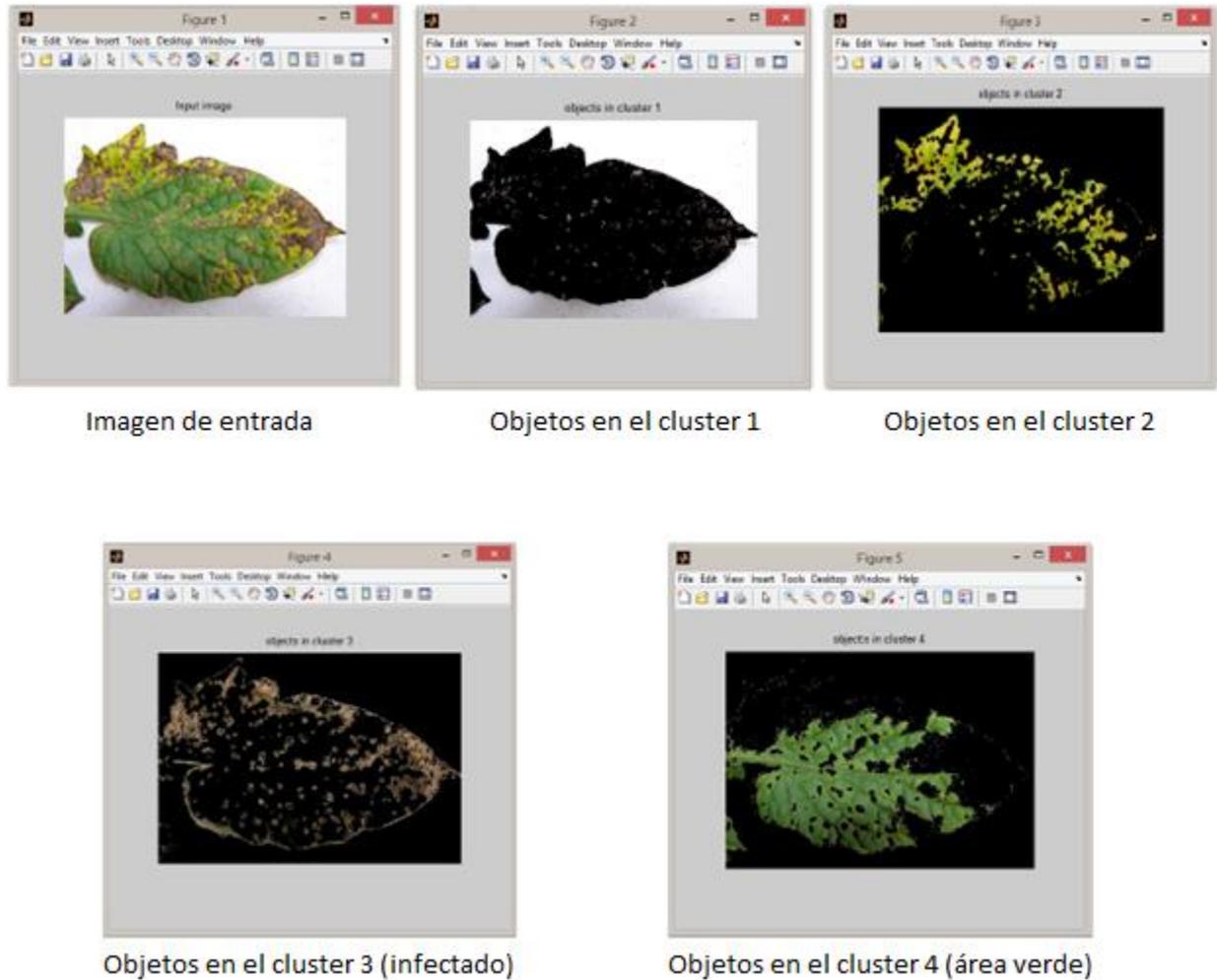
utilizados en la detección de enfermedades de plantas (Khirade & Patil, 2015). En la Figura 23, se muestra el funcionamiento de la red neuronal.



**Figura 23.** Función principal de la red neuronal artificial.

En (Mainkar et al, 2015), se proporciona una solución de software para detectar y clasificar automáticamente enfermedades de hojas de plantas utilizando técnicas de procesamiento de imágenes para clasificar las enfermedades de éstas, proporcionando un enfoque para mejorar la productividad de los cultivos. Para esto, se realizó la adquisición de imágenes, preprocesamiento de imágenes, segmentación, extracción de

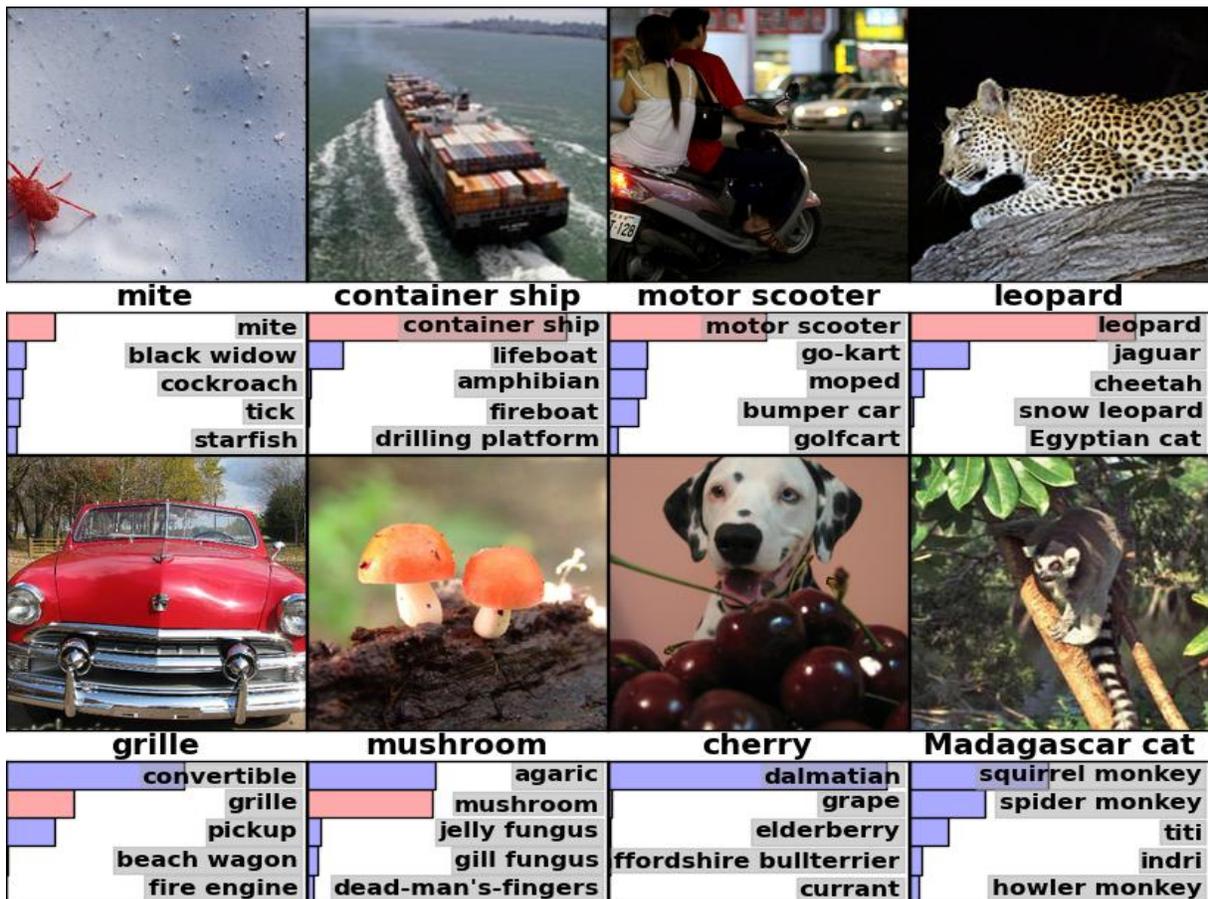
características y clasificación basada en redes neuronales. Algunas técnicas que se utilizaron se muestran en la Figura 24.



**Figura 24.** Imagen con la muestra original dañada y aplicación de las diferentes técnicas utilizadas en la investigación.

Alex Krizhevsky et al (2012), formaron una red neuronal convolucional grande y profunda para clasificar los 1,2 millones de imágenes de alta resolución en el concurso de imágenes LSVRC-2010, generando 1000 clases diferentes de objetos. La red neuronal, que tiene 60 millones de parámetros y 650,000 neuronas, consta de cinco capas convolucionales, algunas de las cuales son seguidas por capas de maxpooling, y tres

capas completamente conectadas con softmax, con un final de 1000 vías (categorías). Utilizaron dos GPU's para el entrenamiento de la red y hacer las operaciones de entrenamiento de la red CNN en el menor tiempo posible. Para reducir el sobreajuste en las capas totalmente conectadas, emplearon un método de regularización recientemente desarrollado llamado "dropout" que demostró ser muy efectivo, como se muestra en la Figura 25.



**Figura 25.** Muestra las cinco etiquetas que el modelo considera más probables. La etiqueta correcta se escribe debajo de cada imagen, y la probabilidad asignada a la etiqueta correcta también se muestra con una barra roja.

De los nuevos trabajos de investigación que se citaron anteriormente, podemos observar que hay infinidad de trabajos relacionado con la técnica de CNN para el reconocimiento y/o clasificación de imágenes.

El trabajo más apegado a esta investigación, es el de Sladojevic et al (2016), donde con una red neuronal llamada CaffeNet, realizan la clasificación de plagas. Sin embargo, la CNN AlexNet es reconocida por ser mejor en clasificación de imágenes y reconocimiento.

Finalmente, Cabe mencionar que en los trabajos relacionados que fueron analizados, no se muestra la creación de un servidor local para procesar la imagen, ni la visualización de los daños en el mapa con la extracción de coordenadas (latitud y longitud), información extraída de los metadatos de la fotografía, funciones que se tienen contempladas en este trabajo.

### **2.3 Marco teórico**

En esta tesis de maestría se presenta una herramienta de software que permitirá a través de una red neuronal convolucional detectar los posibles patógenos que puedan desarrollarse en plantas ornamentales, con la finalidad de prevenir la proliferación y el contagio de éstas. Por consiguiente, en este marco teórico se presentan los conceptos necesarios e involucrados para comprender tanto el contexto donde será utilizada esta herramienta, así como las tecnologías computacionales utilizadas para el desarrollo de ésta. A continuación, describimos cada uno de estos conceptos.

**Viveros**, espacio físico ubicado en un determinado lugar en donde se producen y/o comercializan plantas ornamentales (Yanez, 2008). Un ejemplo de vivero se muestra en la Figura 26.



Figura 26. Vivero de ORNACOL.

La **Visión artificial** es una, “disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un computador” (Visión artificial, 2017). Un esquema de visión artificial se muestra en la Figura 27.

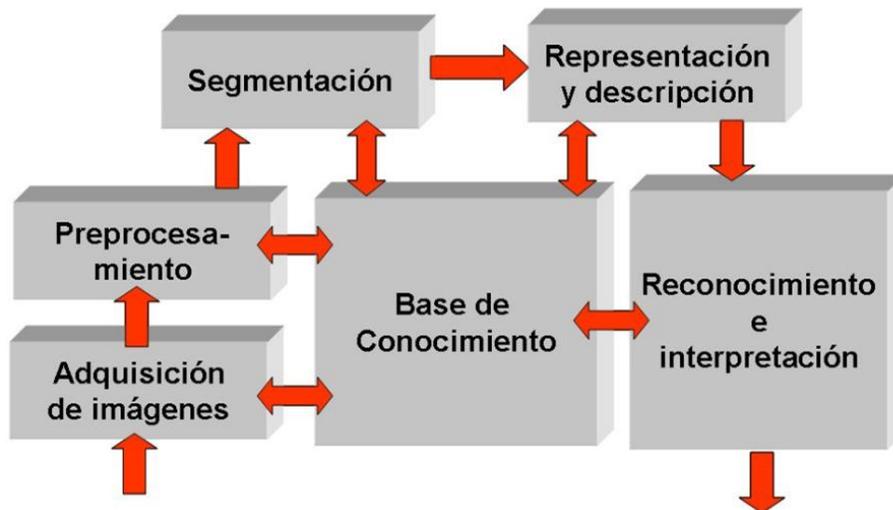


Figura 27. Esquema de visión artificial (Diagrama, 2016).

Una **red neuronal artificial**, es una máquina que está diseñada para modelar la forma en que el cerebro realiza una tarea particular o función de interés (Haykin, 2001). En la Figura 28, se muestra la arquitectura de una red neuronal artificial.

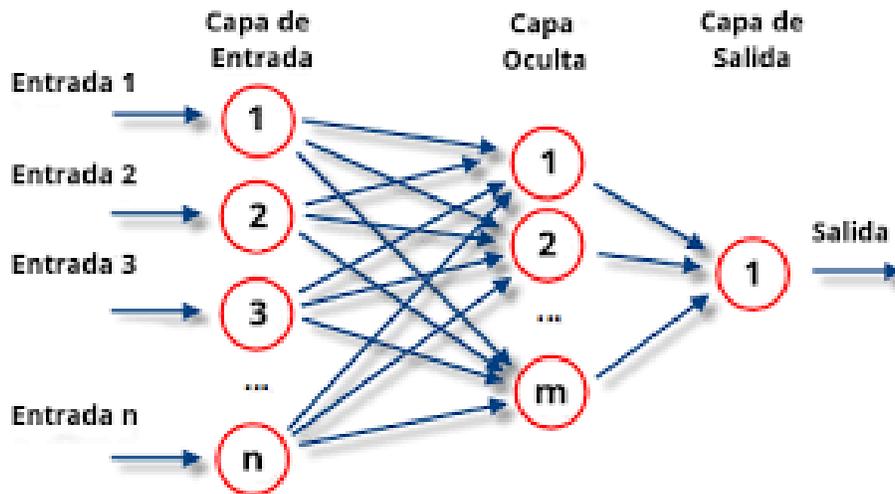


Figura 28. Red Neuronal Artificial.

Una **red neuronal convolucional**, “es una red multicapa que consta de capas convolucionales y de submuestreo alternadas, y al final tiene capas de conexión total como una red perceptrón multicapa” (Diego Calvo, 2018). En la Figura 29, se muestra la arquitectura de una CNN.

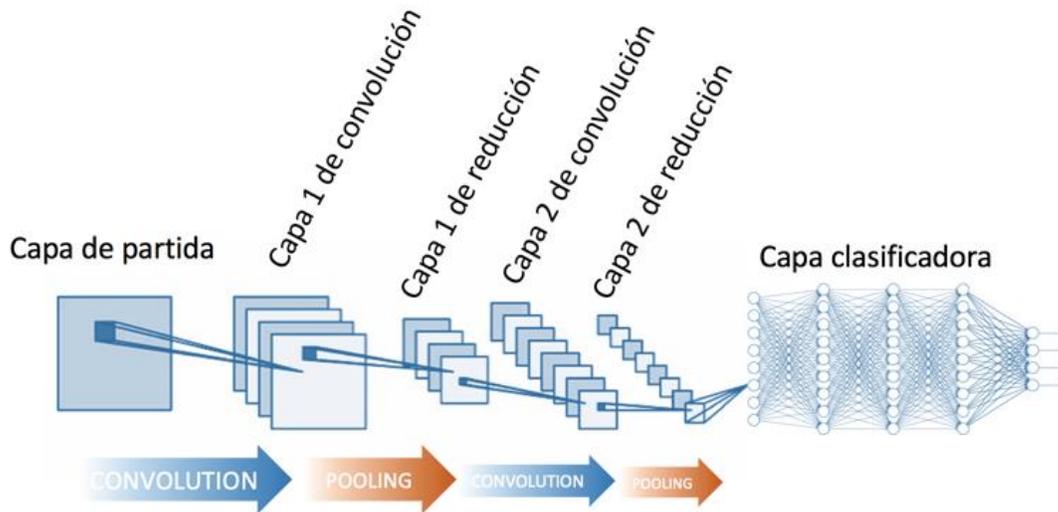


Figura 29. Arquitectura de una red neuronal convolucional.

**AlexNet**, red neuronal convolucional escrita en plataformas de computación paralela (CUDA “Compute Unified Device Architecture”) y soportada en unidades de procesamiento gráfico (GPU). La arquitectura de AlexNet, se muestra en la Figura 30.

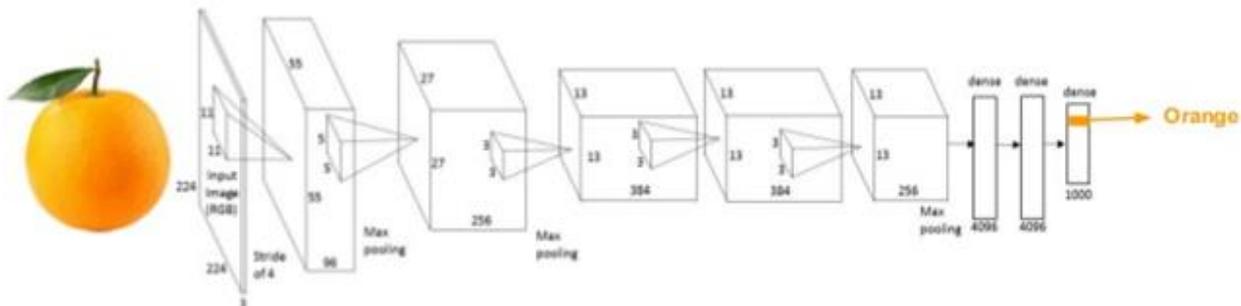


Figura 30. Arquitectura de la red neuronal artificial AlexNet.

**Kanban**, Técnica aplicada para la mejora continua, creada por la empresa Toyota para controlar el avance del trabajo. En la Figura 31, se muestra un ejemplo de esta metodología.

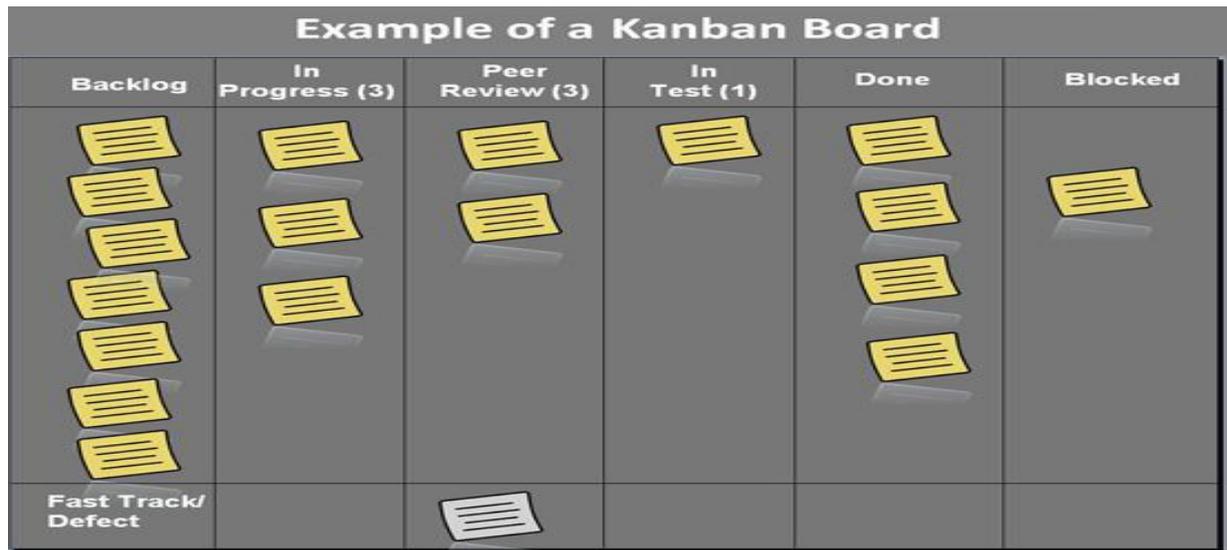


Figura 31. Ejemplo de los procesos de Kanban.

**Angular 5**, es un framework de JavaScript para crear WEBS SPA (Single Page Application). En la Figura 32, se muestra el logo de este framework.



Figura 32. Logo Angular 5.

**Matlab**, (MATrix LABoratory) software matemático que ofrece un entorno de desarrollo (IDE, por sus siglas en inglés) con un lenguaje propio de programación (Lenguaje M). En la Figura 33, se muestra el logo de este software.



**Figura 33.** Logo MATLAB.

**XAMPP**, es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. En la Figura 34 muestra el logo de este software.

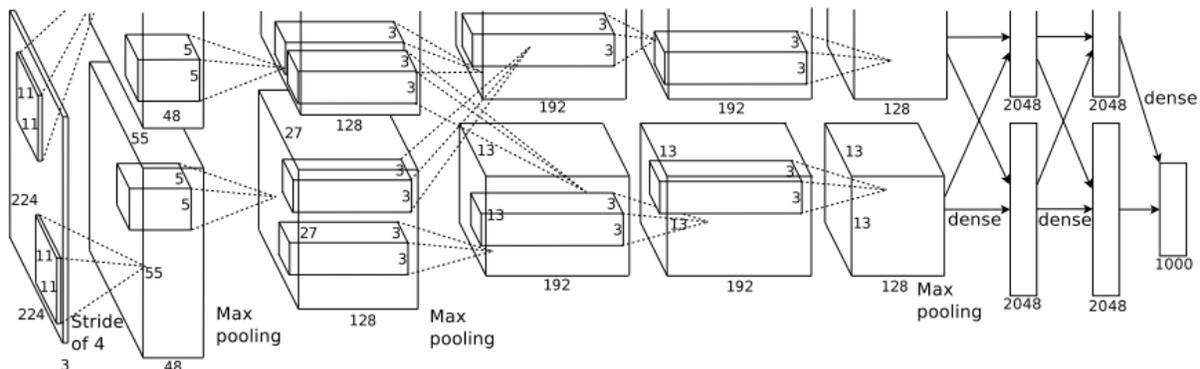


**Figura 34.** Logo XAMPP.

### 3. Métodos empleados

Para el desarrollo de esta tesis de maestría se utilizó y reentrenó AlexNet, una CNN reconocida y utilizada en diversos proyectos. Ésta fue propuesta por Krizhevsky et al (2012), utilizando un modelo de redes neuronales convolucionales y entrenada a través de un conjunto de datos muy extenso conocido como ImageNet. Cuenta con la capacidad de controlar la variación en profundidad y amplitud de la imagen, así como hacer suposiciones correctas sobre la naturaleza de las imágenes de entrada para su clasificación.

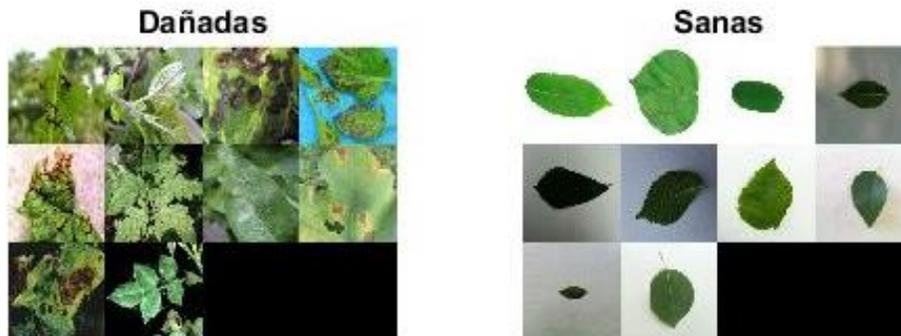
La arquitectura de la red neuronal “AlexNet” contiene ocho capas aprendidas, cinco convolucionales y tres totalmente conectadas. Esta configuración de capas de AlexNet, es considerada una de las mejores, ya que las pruebas que se realizan eliminando alguna de las capas o modificando la estructura actual de las capas permite reducir el rendimiento de la red. En la Figura 35 se muestra la arquitectura utilizada por AlexNet.



**Figura 35.** Arquitectura de la CNN AlexNet (Krizhevsky et al., 2012).

La CNN AlexNet, utilizada en esta tesis, fue reentrenada con la finalidad de clasificar y predecir dos categorías en las hojas: las hojas dañadas y las hojas sanas. Para esto se inició con un conjunto de datos de cinco imágenes por cada categoría, incrementando a diez imágenes en cada conjunto y terminando con 75 imágenes de cada uno de estos

conjuntos, buscando una predicción más precisa. En la Figura 36, se muestra un conjunto de 10 imágenes de cada clasificación de hojas.



**Figura 36.** Muestra de 10 imágenes de entrenamiento.

Estas 75 imágenes de entrenamiento se redimensionaron a 227x227x3 en tamaño, ya que la red AlexNet en su primera capa acepta de entrada este tamaño de dimensiones para cada imagen, para posteriormente proseguir con sus operaciones y sus siguientes 25 capas como se muestra en la tabla 5.

**Tabla 5.** Capas utilizadas en el modelo de red neuronal reentrenada.

1	ImageInputLayer
2	Convolution2DLayer
3	ReLULayer
4	CrossChannelNormalizationLayer
5	MaxPooling2DLayer
6	Convolution2DLayer
7	ReLULayer
8	CrossChannelNormalizationLayer
9	MaxPooling2DLayer
10	Convolution2DLayer
11	ReLULayer
12	Convolution2DLayer
13	ReLULayer
14	Convolution2DLayer
15	ReLULayer
16	MaxPooling2DLayer
17	FullyConnectedLayer

18	ReLU Layer
19	Dropout
20	FullyConnectedLayer
21	ReLU Layer
22	Dropout
23	FullyConnectedLayer
24	SoftmaxLayer
25	ClassificationOutputLayer

Una vez redimensionadas las imágenes, se seccionó en dos carpetas las muestras de imágenes donde se etiquetaron como Dañada y la otra como Sana, para posteriormente crear una variable de tipo almacén de datos en imágenes que nos ayudará a procesarlas internamente en la computadora para el entrenamiento en la CNN.

Contando con la variable de las imágenes categorizadas y procesadas, se realizó el reentrenamiento de la red neuronal para clasificar estas dos categorías y posteriormente someterla a validación con otro conjunto de datos nuevos datos (imágenes), que se usaron nuevamente para conocer la precisión en la predicción de cada hoja.

Contando con el método que se realizaría (Transfer Learning) se utilizó el software de programación MATLAB para poder manipular la red neuronal ALEXNET, modificando las últimas capas de la red y reentrenarla para la clasificación de hojas “Sanas” y “Dañadas”.

Se utilizó a su vez el framework de programación Angular 5, con el cual se desarrolló el cliente WEB que permite visualizar los datos almacenados por la aplicación central de MATLAB en MySQL a través de XAMPP.

Por otro lado, para la realización de tareas y secuencia de actividades, se llevó un registro de actividades con la metodología Kanban donde se visualiza el flujo de trabajo en tarjetas y es fácil poder llevar el control de las tareas a realizar, sobre todo, en qué estado se encuentra la tarea o petición. En la Figura 37, se muestra el inicio de actividades requeridas por la empresa ORNACOL.

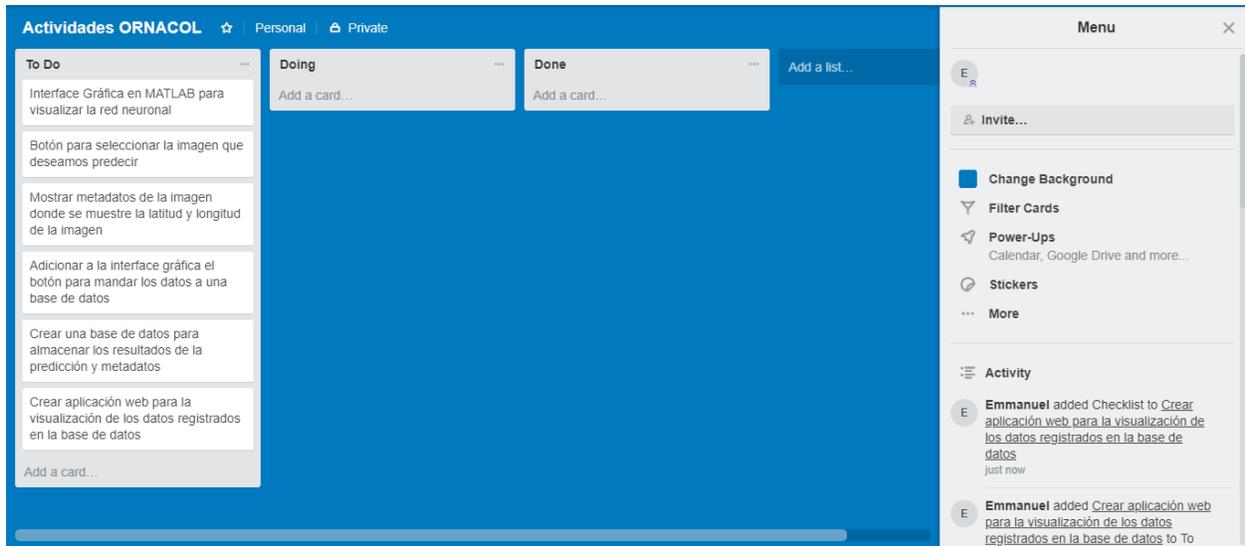


Figura 37. Esquema de la metodología Kanban con peticiones de requerimientos.

## 4. Desarrollo

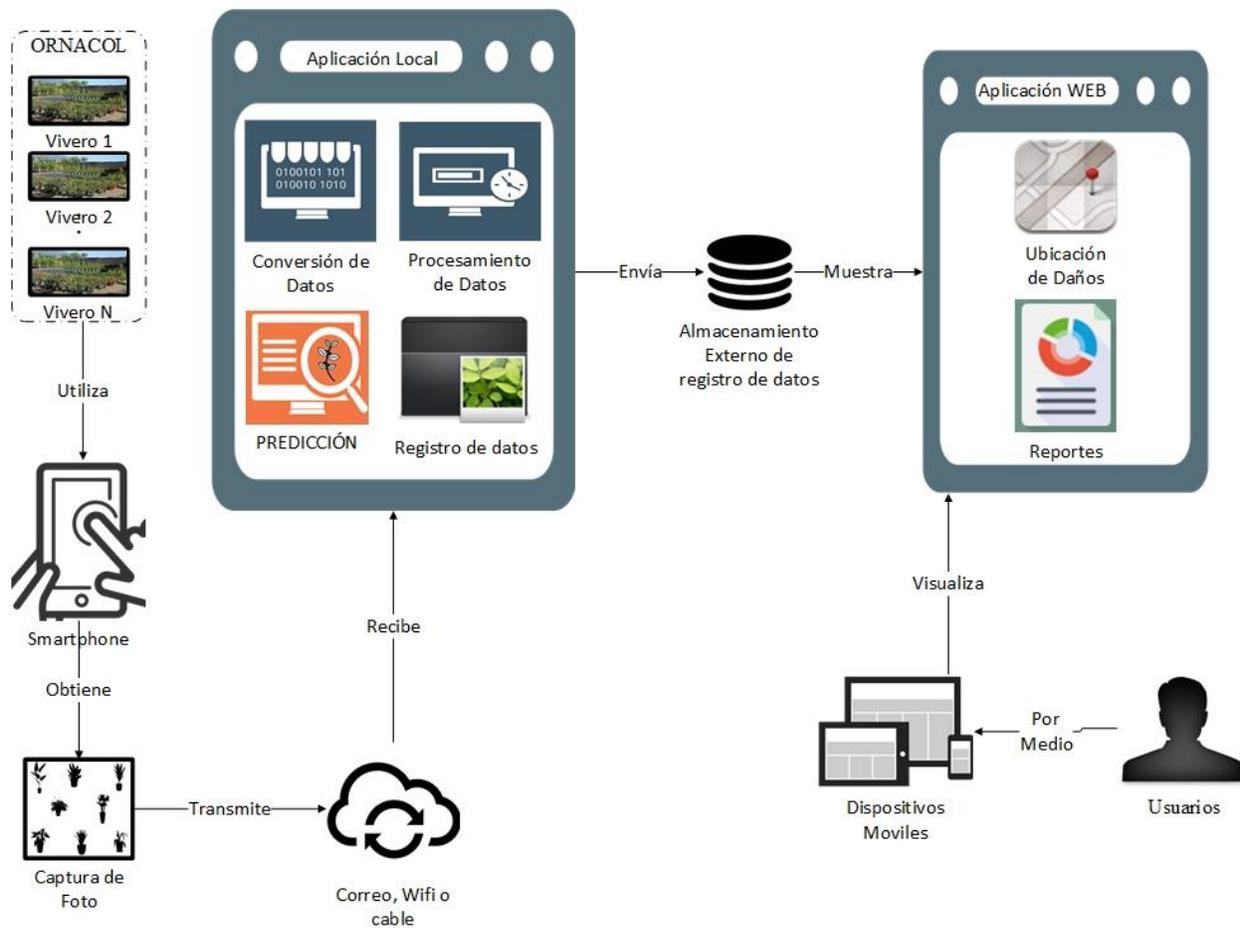
### 4.1. Análisis

A medida que va avanzando la tecnología, las redes neuronales van ganando terreno en los avances tecnológicos, la clasificación de imágenes o reconocimiento de imágenes digitales por medio de un computador, ha dado pasos agigantados que ayudan a resolver problemas que hace unos años fueran únicamente de ciencia ficción. Hoy en día existe infinidad de prototipos tecnológicos que pueden realizar tareas tales como llegar a reconocer a una persona, algún animal, tipos de automóviles, e inclusive algunos patrones que ayudan a la clasificación y/o predicción de objetos.

En esta investigación, se realizó una herramienta que ayuda a la predicción de agentes patógenos utilizando inteligencia artificial en la aplicación principal. En ésta, se implementa de manera sencilla la obtención de datos por medio de un teléfono inteligente (foto) para ser procesada y finalmente visualizarla vía WEB, para obtener la predicción del estado de la planta, como la ubicación en un mapa donde se realizó la toma de datos. Este proceso se visualiza en la Figura 38.

### 4.1.1. Modelo conceptual

Como se puede observar, en la Figura 38 se ilustra el modelo conceptual del sistema para la predicción de agentes patógenos en plantas ornamentales, utilizando redes neuronales convolucionales. Este modelo está conformado por los siguientes elementos:



**Figura 38.** Sistema para la Predicción de Agentes Patógenos en Plantas Ornamentales Utilizando Redes Neuronales

**Aplicación Local.** En este módulo se obtendrá la imagen que envía la aplicación móvil para realizar su procesamiento y predicción del agente patógeno en la imagen, constará de cuatro módulos internos listados a continuación:

**Conversión de Datos.** En este módulo, se convertirán los datos (fotografía) en un formato de matrices de datos (Números), para el posterior procesamiento de los mismos en la red neuronal.

**Procesamiento de datos.** Este segundo módulo, se realizarán las operaciones de los datos, los cuales se procesan por distintas capas en la red neuronal, realizando comparaciones de características en los datos para clasificar el agente patógeno que presenta la imagen y obtener la red entrenada de datos.

**Predicción.** Uno de los módulos más importantes, en el cual se podrá realizar la predicción del agente patógeno que se pueda encontrar en la imagen seleccionada al momento de comparar los valores de la imagen, contra los valores de la red neuronal entrenada de datos.

**Registro de datos.** este módulo realizará el registro de los datos en un almacenamiento externo, el cual se podrá acceder vía WEB para interpretar los datos de forma visual.

**Aplicación WEB.** En este módulo se desarrolla una interfaz gráfica amigable con el usuario, operando en un ambiente WEB, el cual interpretará y mostrará los datos registrados desde la aplicación local.

#### 4.1.2. Diagrama de actividades

En el siguiente diagrama de actividades de la Figura 39, se ilustra el proceso completo para llevar a cabo la predicción del agente patógeno en una planta ornamental, a través de cada uno de los módulos descritos anteriormente

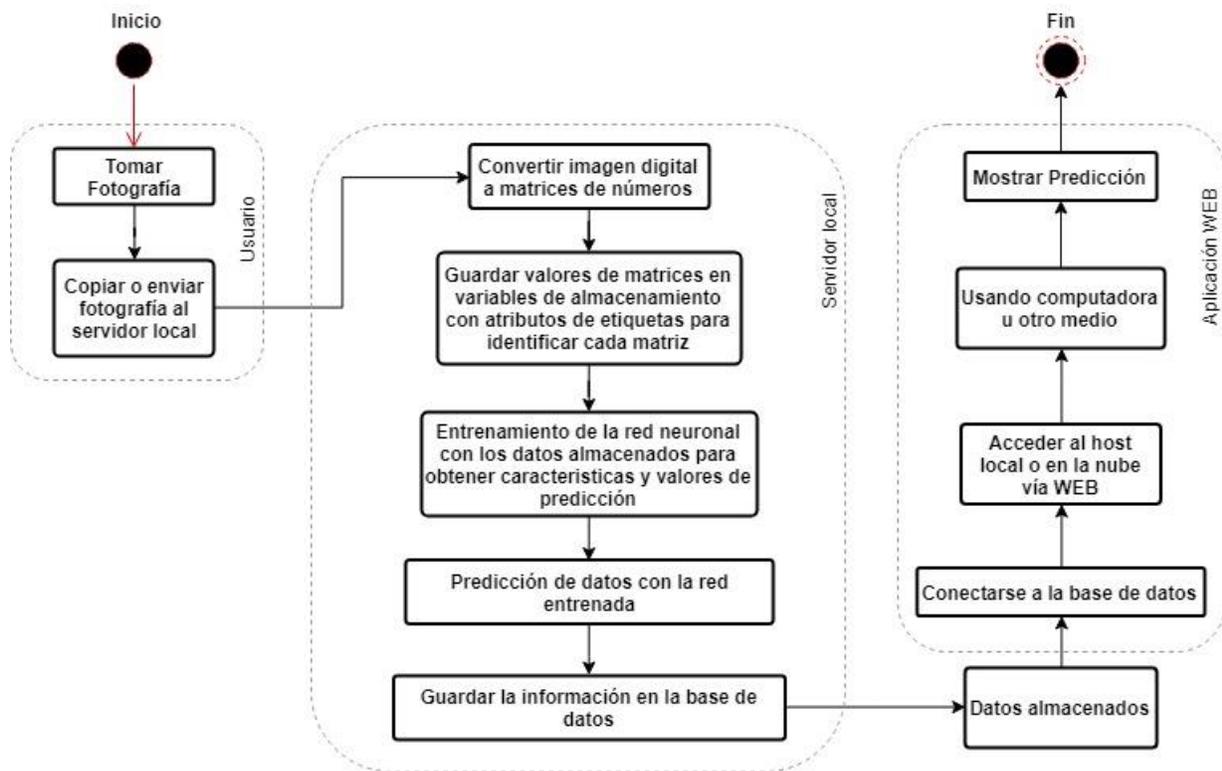


Figura 39. Diagrama de actividades.

### 4.1.3. Diagrama de casos de uso

En la Figura 40 se ilustra el caso de uso, el cual ilustra el conjunto de acciones que se llevan a cabo para poder realizar la predicción de un agente patógeno.

Como se puede observar, el cliente toma una foto la cual se envía al servidor. El servidor realizará la conversión, procesamiento de los datos, predicción y registro. Estos datos serán almacenados en una base de datos. Los usuarios podrán obtener de manera visual, los datos almacenados en la base de datos por medio de un mapa o listado de las imágenes dañadas.

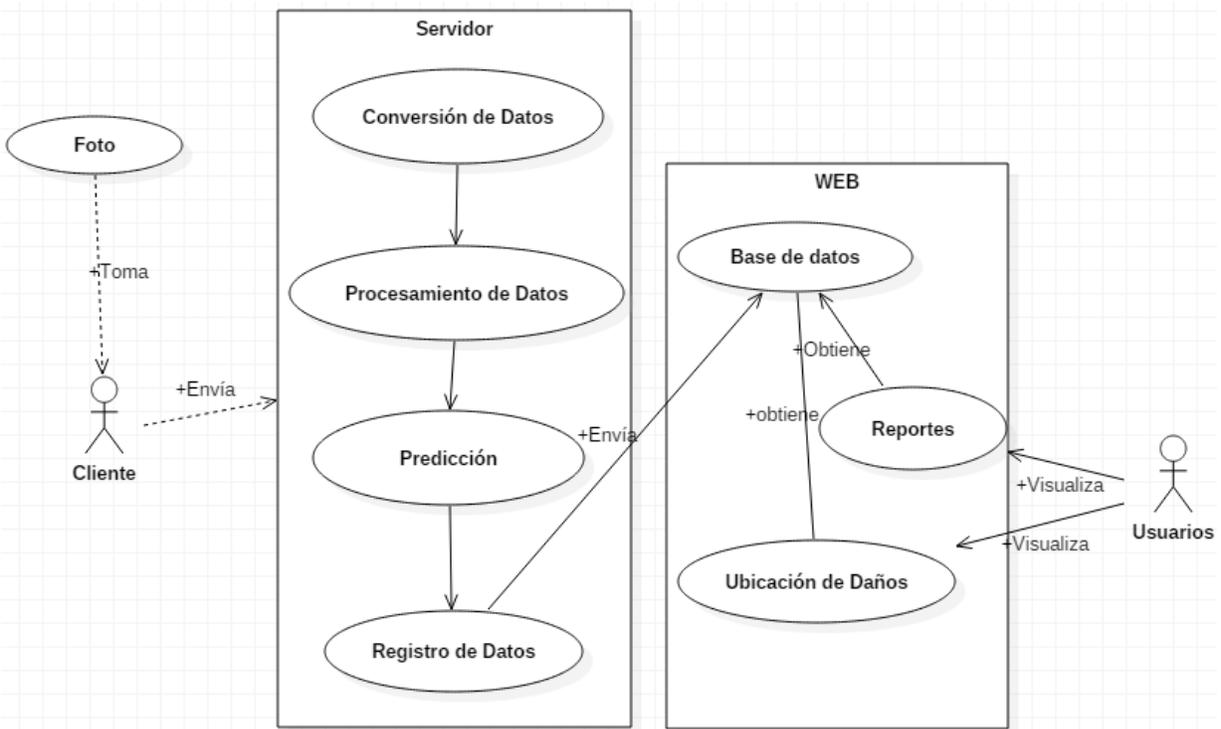


Figura 40. Modelo de casos de uso.

#### 4.1.4. Modelo de requisitos

En este apartado se mostrarán los requisitos necesarios para la operación funcional del sistema, tanto para el sistema local central, como para el módulo WEB de visualización.

##### 4.1.4.1 Requisitos funcionales

Los requisitos necesarios para la operación del sistema se muestran en las tablas 6 y 7.

**Tabla 6.** Requisitos funcionales del servidor local.

Operación del servidor	
01	Se realizará la toma de fotografía de la planta ornamental dañada con un teléfono inteligente, considerando tener activada la función de GPS, ya que esta función permitirá guardar la ubicación en los metadatos de la fotografía.
02	Contando con la imagen en el servidor central, se abrirá la aplicación del sistema, en el cual contará con un botón de búsqueda del archivo para procesarla en la red neuronal artificial y obtener los metadatos de la imagen.
03	Una vez procesada la imagen, al obtener todos los datos necesarios, se tendrá la opción de un botón para envío de datos y registrarlos en una base de datos externa.
04	Se realizará la conexión por FTP, para envío de la fotografía si es requerida.

**Tabla 7.** Requisitos funcionales de la aplicación WEB.

Operación WEB	
01	Se contará con un servidor MySQL el cual alojará nuestra información enviada en una base de datos. Una vez registrados los datos, se podrá consultar los datos vía WEB con la aplicación visual para este propósito.
02	Contar con un dispositivo que contenga una aplicación de navegador WEB (Internet Explorer, Chrome o Firefox)
03	Tener una herramienta para el alojamiento local como XAMPP y acceder a los datos por medio de la dirección "localhost", o bien, contratar un alojamiento externo y acceder al sitio por medio de la dirección WEB o dominio que sea adquirido.
04	Para alojar nuestra página WEB de visualización de datos, se deberá contar con un alojamiento de aplicaciones que soporte JavaScript.

#### 4.1.4.2. Requisitos de software y hardware

Para poder integrar el software es necesario poder contar con:

- MATLAB, al cual se adicionarán los plugins de base de datos, pretrained neural network y la red neuronal convolucional ALEXNET.
- Angular 5, en este framework se adicionará con base al siguiente programa.
- NodeJs, este software nos ayuda a la instalación del framework Angular 5 donde se generará la aplicación WEB para la interacción entre Angular 5 y MySQL.
- XAMPP o hosteo en la nube que cuente con el servicio del gestor de base de datos MySQL, en el cual se almacenarán los registros del sistema.
- Tarjeta gráfica NVIDIA para procesamiento de Inteligencia artificial. Con capacidad de computo de mínimo 6.1, tal como la siguiente tarjeta:

- Nvidia GTX1080TI, 11GB GDDR5X
- Una máquina potente para poder utilizar procesamiento paralelo con las siguientes características mínimas:
  - CPU: Intel® Core™ i7
  - Memoria: RAM 16GB DDR4 3200Mhz
  - Disco Duro: 3TB, 7200RPM

## **4.2. Diseño**

En este apartado se presentan los procesos realizados para el reentrenamiento de la red neuronal, así como la programación del nodo central local y creación de la aplicación WEB.

### **4.2.1. Modelo de clases.**

Se implementó una base de datos sencilla, ya que se requieren pocos datos de la predicción en la foto, los datos que se envían son la ruta, latitud, longitud, vivero y la predicción, los datos de Id y Fecha se calculan automáticamente. El modelo de clases se muestra en la Figura 41.

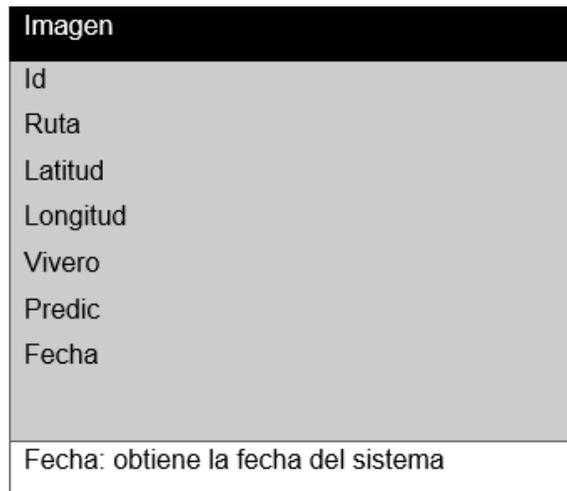


Figura 41. Modelo de clases.

#### 4.2.2 Modelo de datos

La base de datos en MySQL, se muestra en la Figura 42, donde se visualiza como está estructurada y los atributos que contiene cada dato.

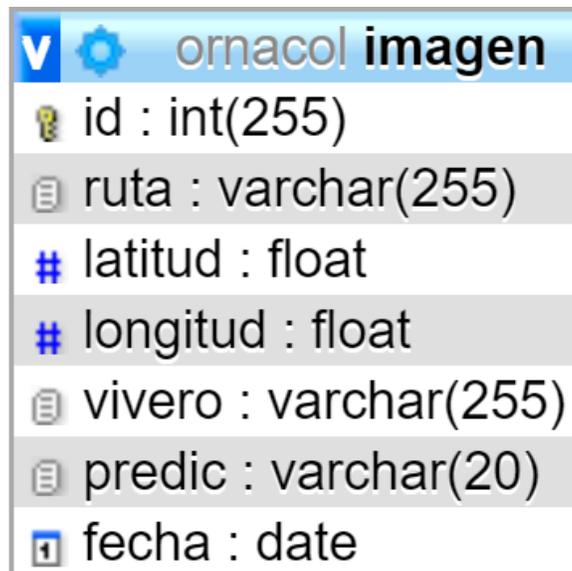


Figura 42. Modelo de datos.

### 4.2.3 Diccionario de datos

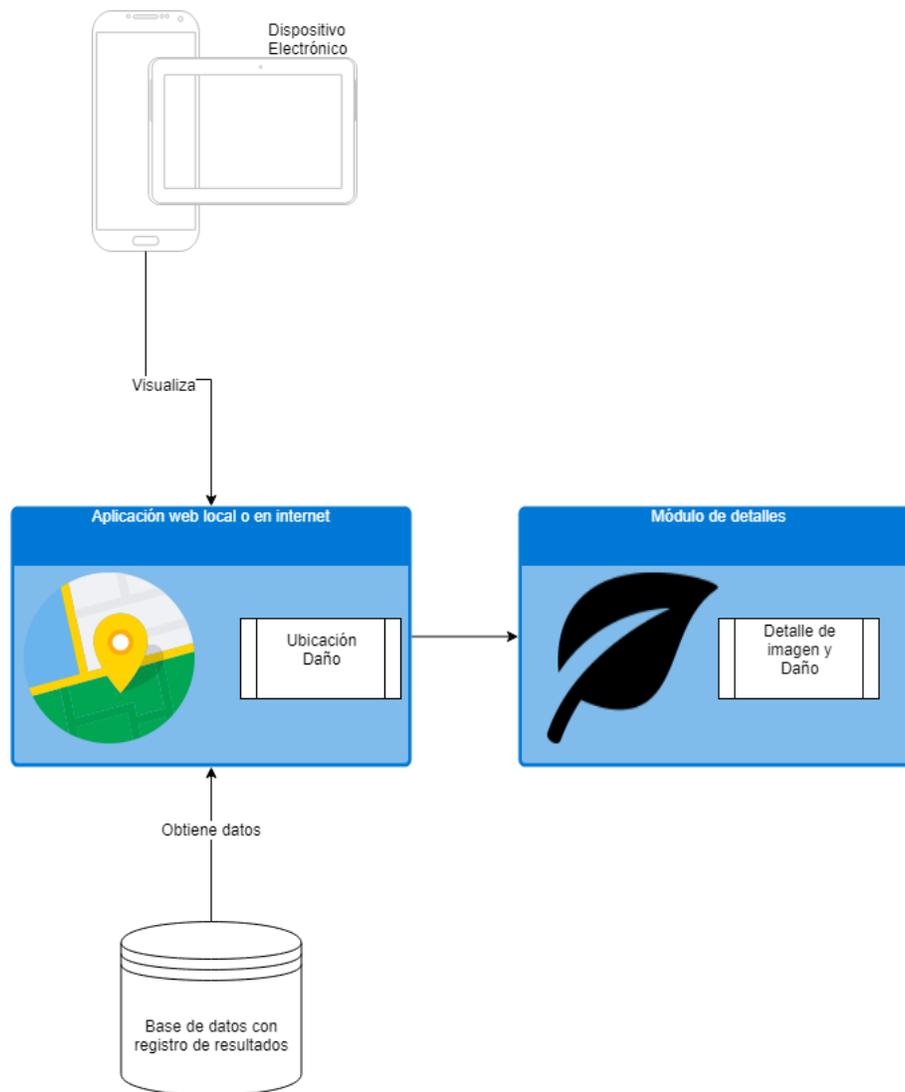
En la Figura 43, se muestra la lista del diccionario de datos con los parámetros que fueron establecidos en MySQL, al momento de crear la base de datos para el registro de resultados.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(255)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  Spatial
<input type="checkbox"/>	2	ruta	varchar(255)	latin1_swedish_ci	Yes	NULL			Change  Drop  Primary  Unique  Index  Spatial
<input type="checkbox"/>	3	latitud	float		Yes	NULL			Change  Drop  Primary  Unique  Index  Spatial
<input type="checkbox"/>	4	longitud	float		Yes	NULL			Change  Drop  Primary  Unique  Index  Spatial
<input type="checkbox"/>	5	vivero	varchar(255)	latin1_swedish_ci	Yes	NULL			Change  Drop  Primary  Unique  Index  Spatial
<input type="checkbox"/>	6	predic	varchar(20)	latin1_swedish_ci	Yes	NULL			Change  Drop  Primary  Unique  Index  Spatial
<input type="checkbox"/>	7	fecha	date		Yes	NULL			Change  Drop  Primary  Unique  Index  Spatial

Figura 43. Diccionario de datos.

#### 4.2.4 Diagrama de navegación

Por ser un módulo pequeño de visualización, únicamente en la parte WEB se muestran dos pantallas, donde su función únicamente es mostrar, en la pantalla principal, los lugares en donde se encontraron dañados, y en una pestaña poder obtener detalles de estos datos registrados como se muestra en la Figura 44.



**Figura 44.** Diagrama de navegación.

#### 4.2.5 Modelo de interfaces

Las interfaces que el usuario tendrá en pantalla se describen posteriormente para tener un bosquejo cercano a la aplicación real.

##### 4.2.5.1 Servidor local con MATLAB

La pantalla principal al iniciar la aplicación aparecerá con dos botones. El botón inicial es el botón “subir fotografía”, el cual nos sirve para buscar en nuestra computadora la imagen que queremos analizar como se muestra en la Figura 45.

The image shows a window titled "OrnaNet" with a white background and a black border. Inside the window, there are several input fields and buttons. On the left side, there is a button labeled "Subir Fotografía". To the right of this button, there are four input fields stacked vertically, each with a label to its left: "Latitud:", "Longitud:", "predicción:", and "Ruta:". At the bottom right of the window, there is another button labeled "Guardar Registros".

**Figura 45.** Pantalla inicial del sistema local.

Al seleccionar el botón “subir fotografía”, se abrirá un navegador para buscar la imagen en la ubicación que se encuentre de nuestra computadora, para poder analizarla y obtener la información necesaria en el sistema, como se muestra en la Figura 46.

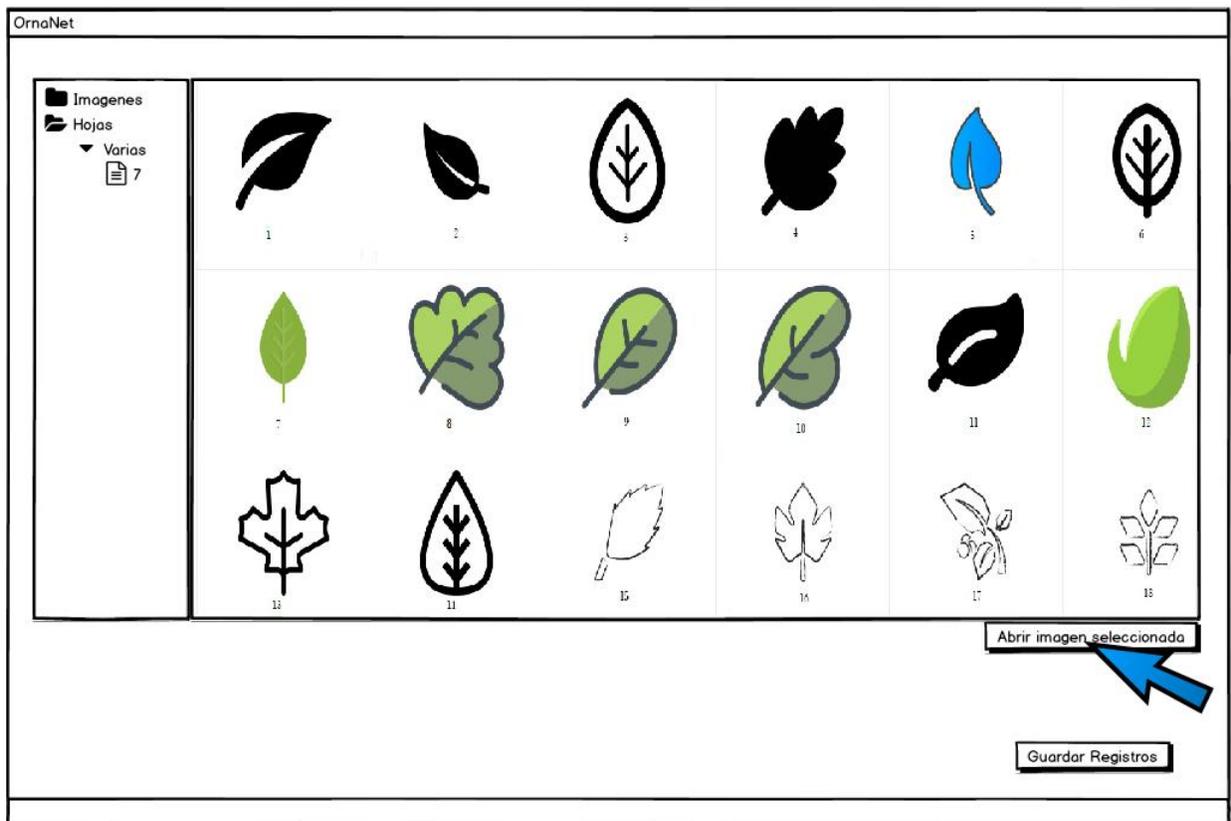


Figura 46. Selección de imagen a procesar.

Al momento de seleccionar la imagen que se desea analizar, el sistema activará la red neuronal que realizará la predicción del estado de la hoja, así como mostrar los datos de ubicación de la imagen en donde fue tomada, tal como se muestra en la Figura 47.

The screenshot displays the OrnaNet web interface. On the left, there is a red leaf image and a button labeled 'Subir Fotografía'. To the right, the system has populated several fields: 'Latitud:' with the value '19.2499700', 'Longitud:' with '-103.7271400', 'predicción:' with 'Dañada', and 'Ruta:' with 'IMG220417.jpg'. A 'Guardar Registros' button is located at the bottom right of the form area.

Figura 47. Pantalla con datos obtenidos de la imagen y la predicción.

Finalizando con el envío de datos mostrados, solo será cuestión de presionar el botón “Guardar Registros” para realizar la conexión a la base de datos y enviar los datos mostrados en pantalla, a los campos de la tabla realizada en MySQL como se muestra en la Figura 48.

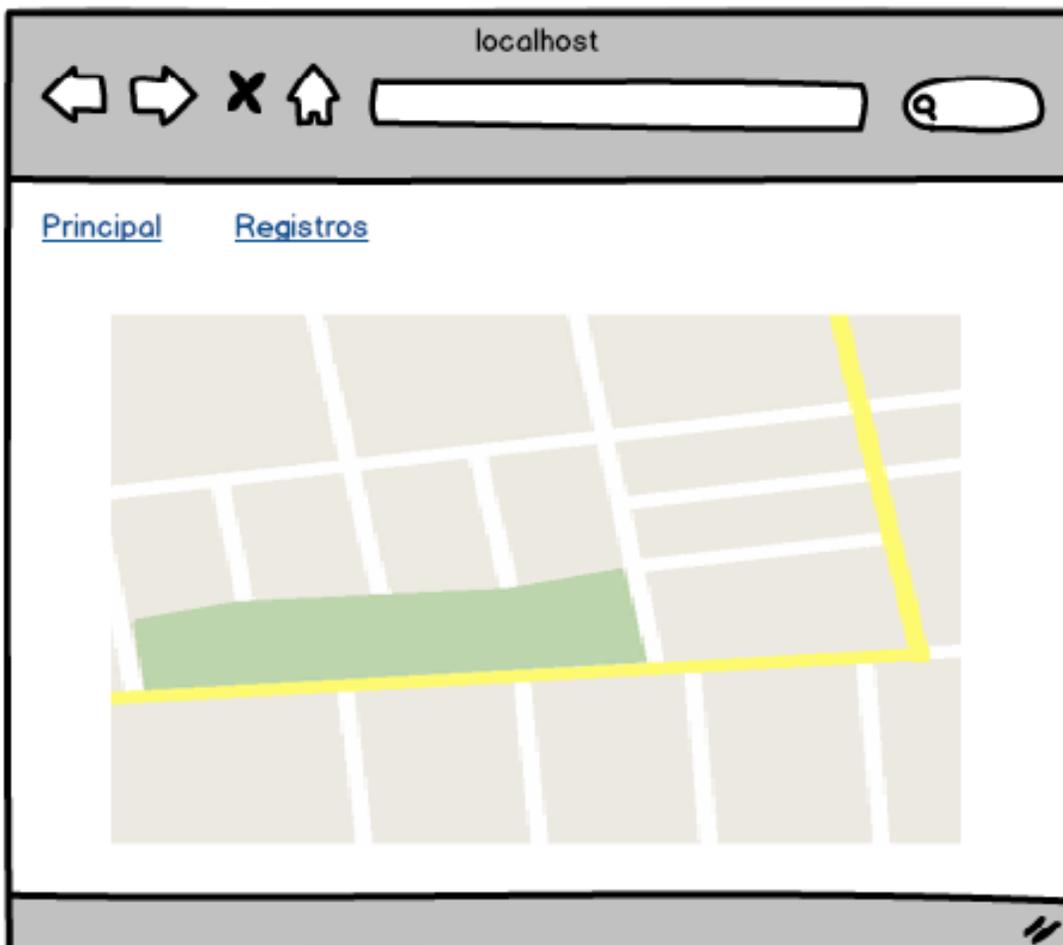
The screenshot displays the OrnaNet web application interface. At the top left, the text "OrnaNet" is visible. Below it, there is a red leaf icon and a text field labeled "Latitud:" containing the value "19.2499700". A large white box with a black border contains the message "Aviso" at the top and "El registro de los datos se realizó con éxito." in the center. Below this message, there is a button labeled "Subir Fotografía". To the right of this button, there is a text field labeled "Ruta:" containing the value "IMG220417.jpg". At the bottom right of the interface, there is a prominent button labeled "Guardar Registros".

**Figura 48.** Pantalla de registros guardados en la base de datos.

#### 4.2.5.2 Servidor WEB

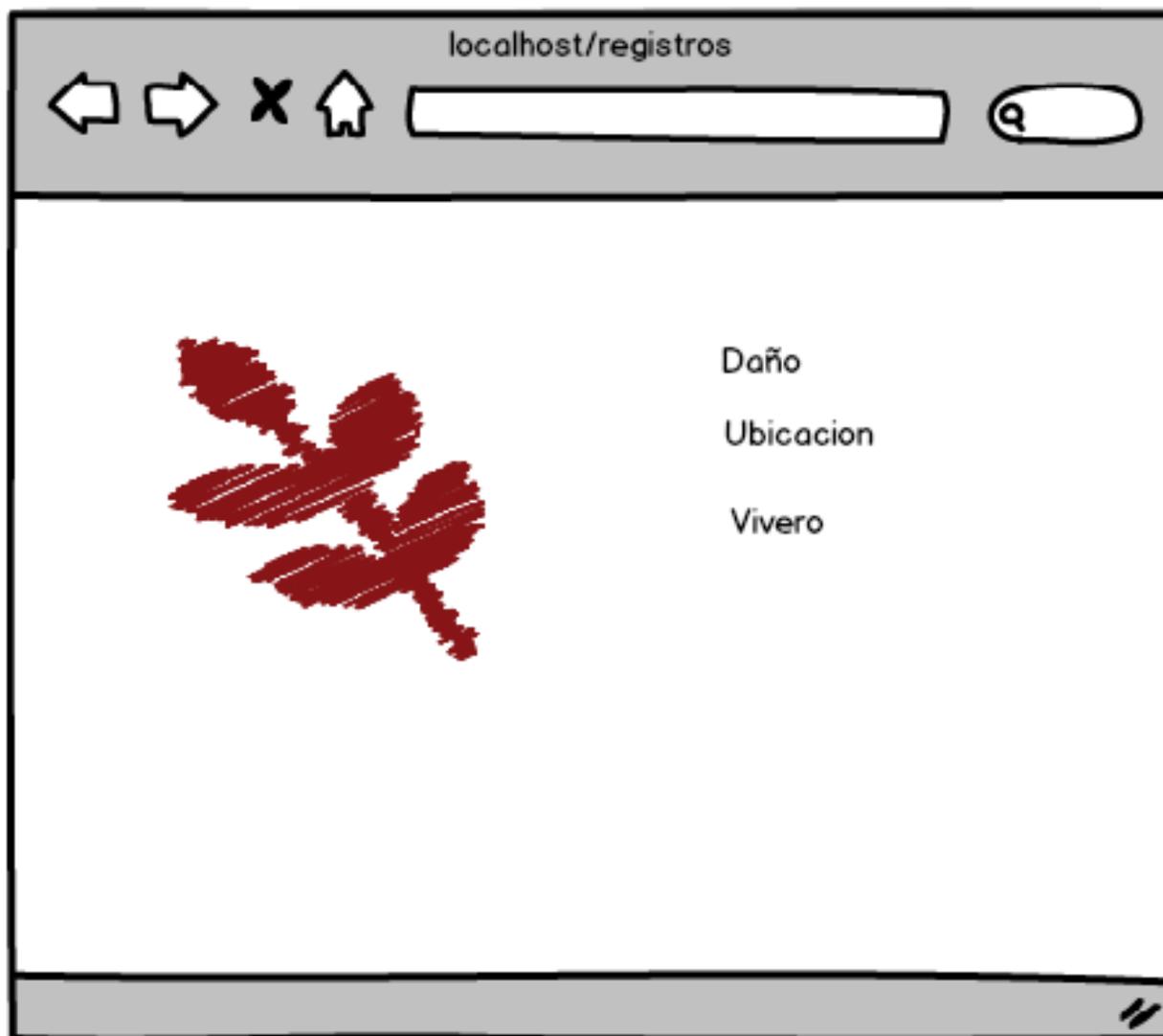
Para poder visualizar los datos almacenados del procesamiento de la imagen en la base de datos, se creó una aplicación WEB que ayude a ver estos datos de manera amigable y llamativa.

La aplicación WEB se realizará sencilla e intuitiva. Al ingresar a la página principal, se mostrará el mapa y grupos de plagas que hayan aparecido o se hayan guardado en la base de datos, como se muestra en la Figura 49.



**Figura 49.** Página principal de la aplicación, donde se muestra el mapa y aparecerán las plagas encontradas.

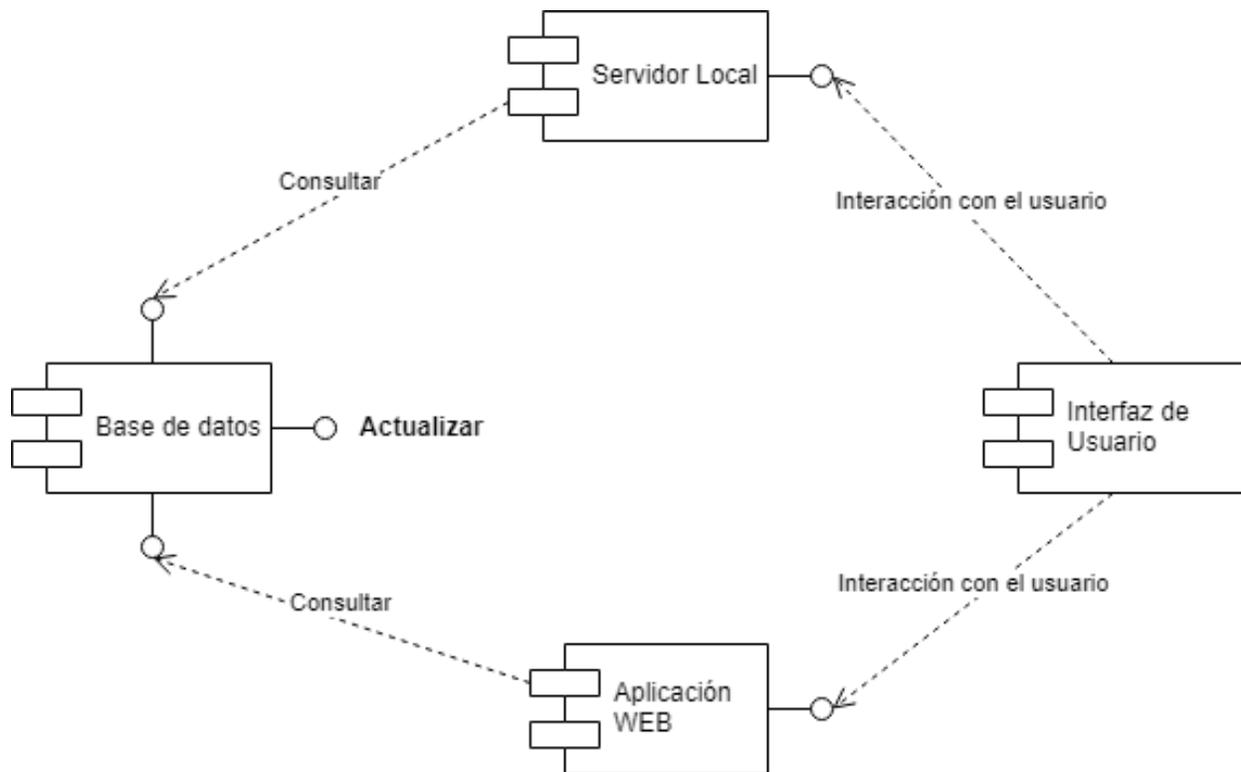
Por otro lado, en la página de registros (ver Figura. 50) se mostrará el conjunto de datos con la información de cada plaga registrada, así como el tipo de daño, la latitud, longitud y vivero en donde fue encontrada.



**Figura 50.** Pantalla donde se muestra a detalle los registros de datos.

#### 4.2.6 Modelo de componentes

Como se puede observar en la Figurara 51, se visualizan los componentes que conforman todo el sistema, tanto el sistema local como el sistema Web y la interacción que realizan entre ellos.



**Figura 51.** Modelo de Componentes.

## 4.3. Implementación

En este capítulo, se muestra el proceso que se realizó para alcanzar el resultado obtenido en el reentrenamiento de la red neuronal AlexNet, así como el desarrollo del cliente WEB para llevar a cabo la visualización de los registros. Esto, permitirá como trabajo futuro realizar mejoras al proyecto de esta investigación.

### 4.3.1. Módulo servidor local

El primer paso que se realizó, fue la instalación del software MATLAB, para poder desarrollar en esta potente herramienta, la programación y reentrenamiento de la red neuronal AlexNet. En la pantalla principal de MATLAB podemos agregar capacidades adicionales, como se muestra en la Figura. 52.

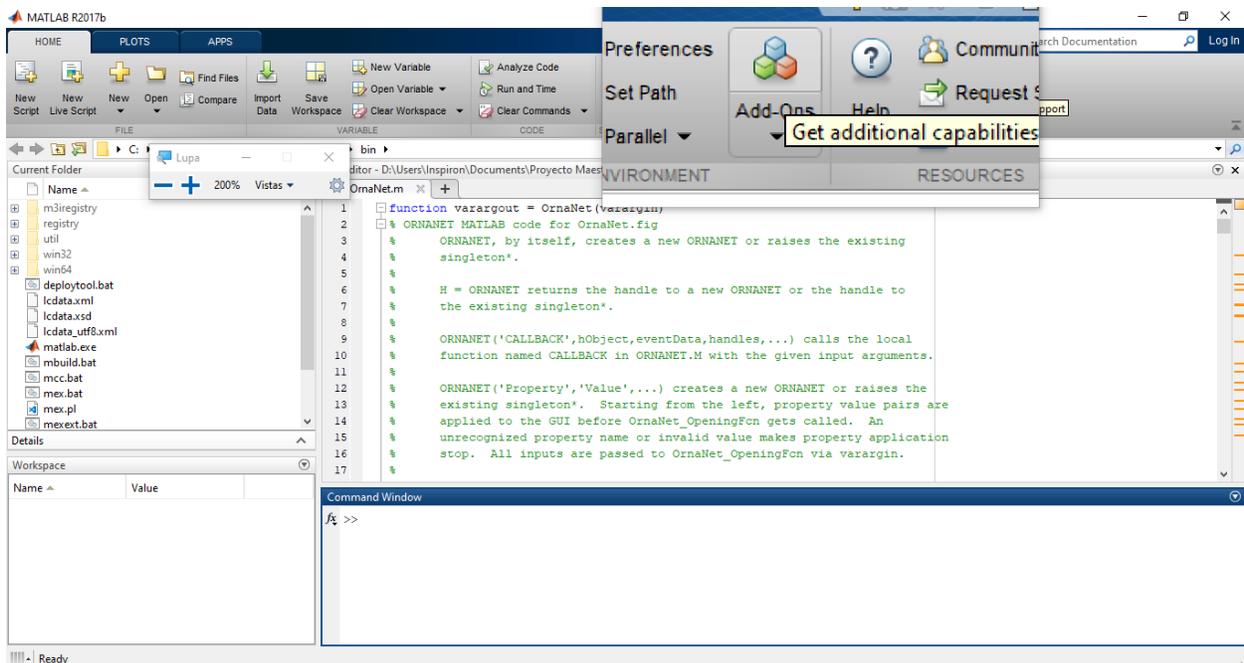


Figura 52. Pantalla principal con icono de complementos de MATLAB.

Lo anterior, permitirá agregar herramientas de inteligencia artificial que se requieran, en este caso se agregó AlexNet como se puede observar en la Figura. 53.

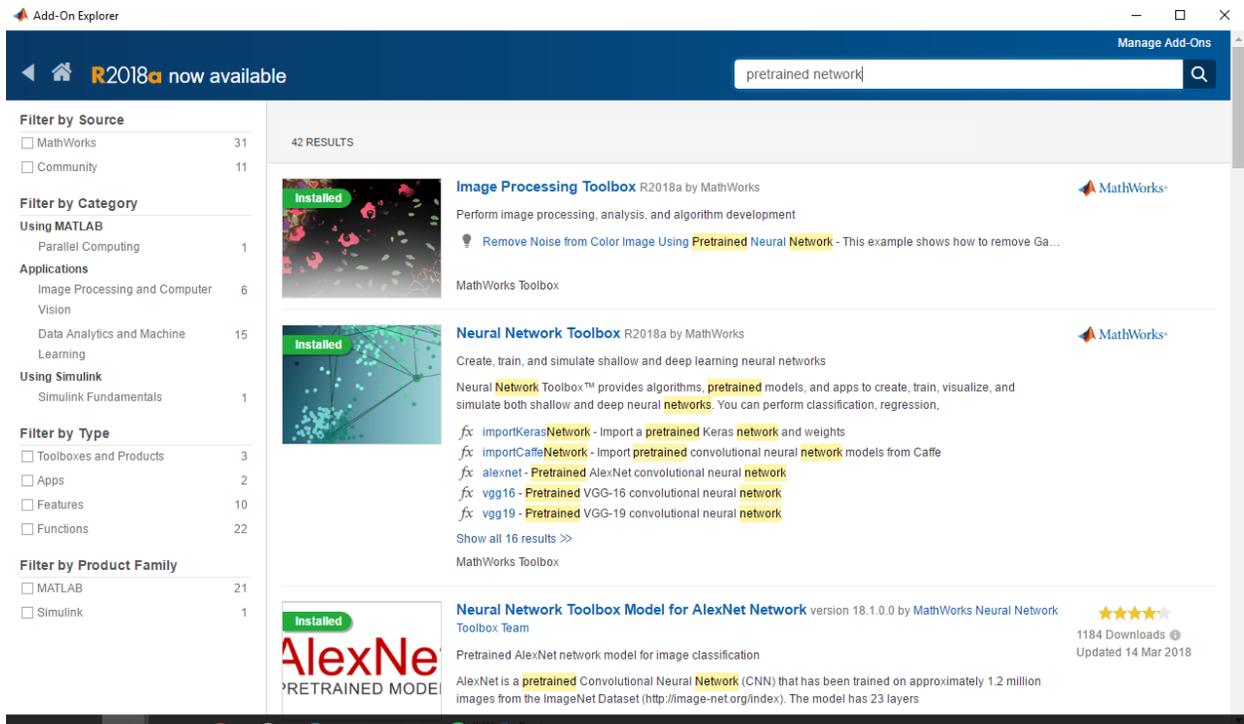


Figura 53. Pantalla del explorador de complementos de MATLAB.

Una vez seleccionada la red neuronal AlexNet, se puede instalar en el sistema, en la Figura 54, aparece la red ya instalada y la opción de administrar.

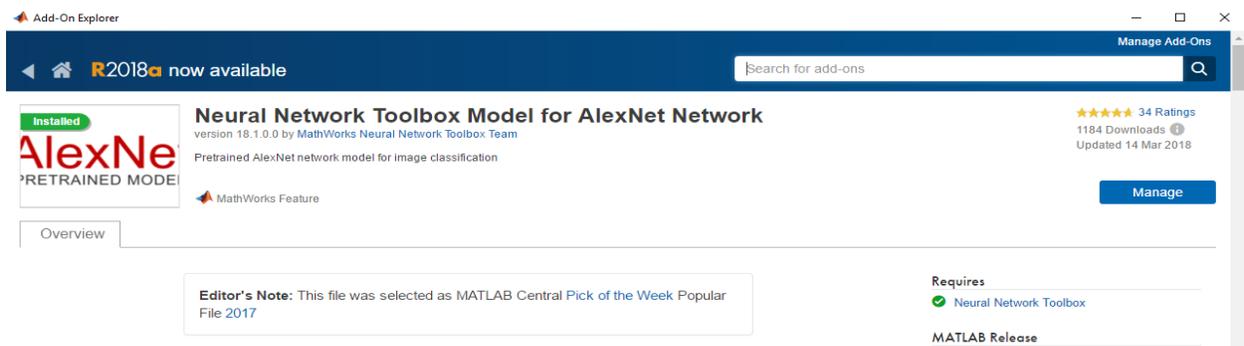
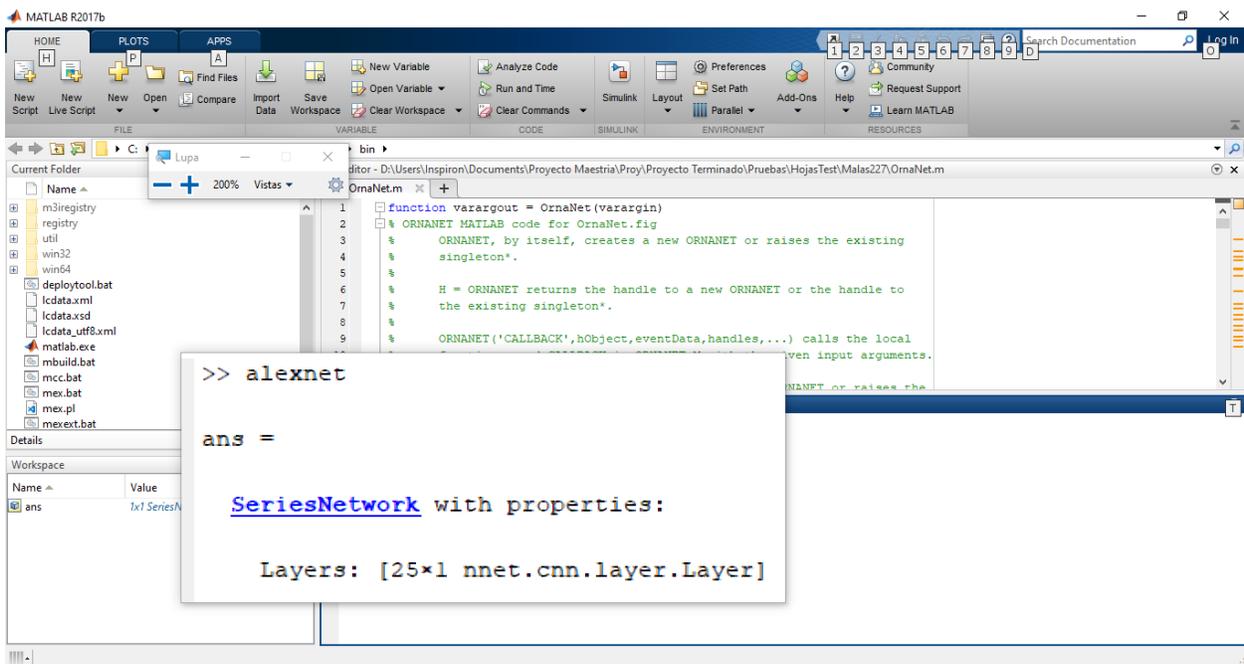


Figura 54. Pantalla del complemento CNN AlexNet.

Una vez instalada la red neuronal, la forma de saber que está funcionando en nuestro sistema, es escribir en la línea de comandos la palabra “alexnet” mostrando datos de la CNN instalada como se muestra en la Figura 55.

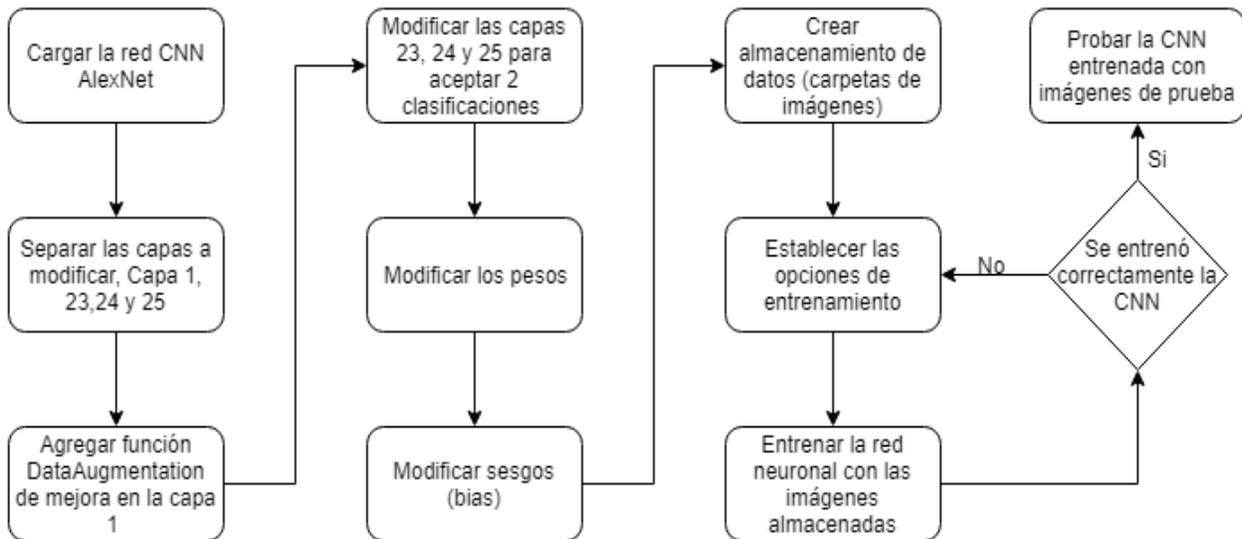


**Figura 55.** Comprobación de funcionamiento de la CNN AlexNet en MATLAB.

Contando con la red instalada, se procede a realizar la modificación de la red neuronal para adaptarla a nuestro objetivo. Para ello, se muestra la secuencia de los procesos realizados en este proyecto para la adaptación y reentrenamiento de la CNN AlexNet.

#### 4.3.1.1. Proceso de reentrenamiento de la CNN

Para explicar de manera concreta el proceso que se realizó en la modificación y reentrenamiento de la CNN, se creó un diagrama el cual explica puntualmente el proceso de cada fase mostrado en la Figura 56.



**Figura 56.** Procesos de reentrenamiento de la CNN.

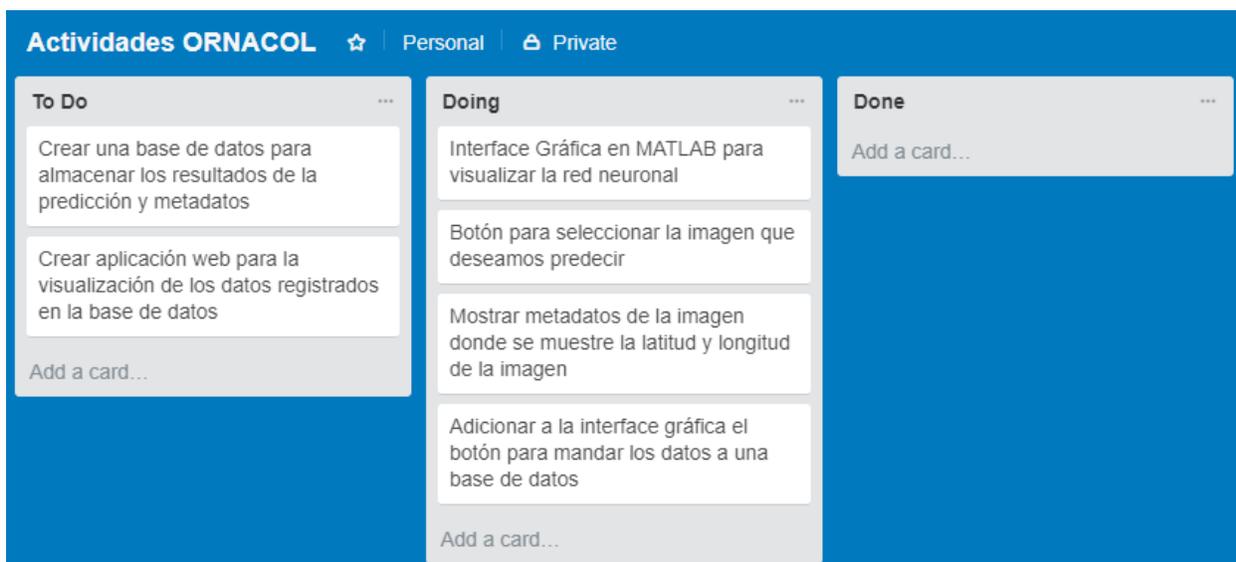
En la última decisión, las opciones de entrenamiento se modifican en la red neuronal para poder acercarnos al resultado deseado. En este caso la tasa de aprendizaje inicial, épocas y el tamaño del mini-lote fueron factores que se modificaron en las opciones de entrenamiento para obtener los resultados esperados. En la Tabla 8 se muestran los valores establecidos.

**Tabla 8.** Valores de entrenamiento final de la CNN.

Opciones de entrenamiento				
Modo de entrenamiento	Tasa inicial de aprendizaje	Factor de regularización	Épocas Máximas	Tamaño de mini-lote
sgdm	0.001	0.0005	100	32

#### 4.3.1.2. Creación de interfaz gráfica

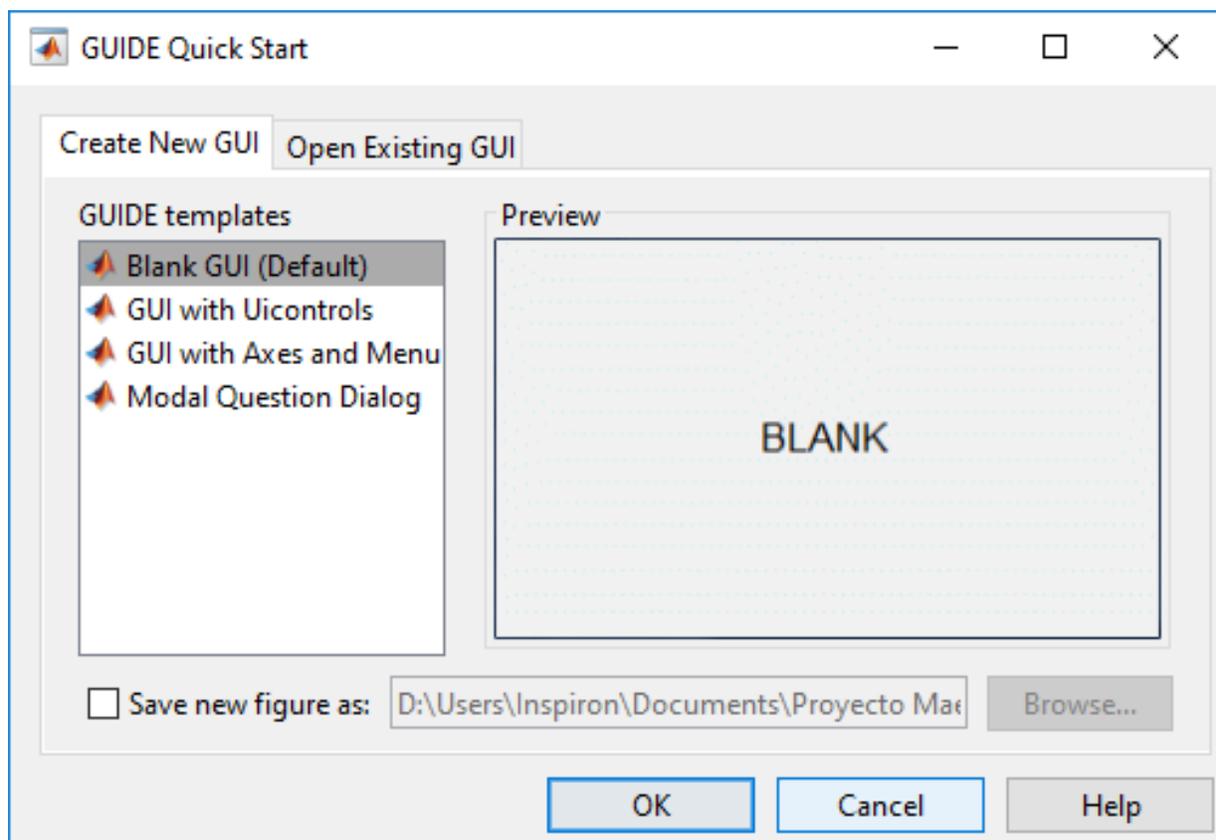
Contado con la CNN reentrenada, se prosigue con la creación de la interfaz gráfica como se tienen en las actividades del modelo Kanban. En la Figura 57, se muestran las actividades que se están realizando en este capítulo.



**Figura 57.** Tareas que se están realizando en este capítulo.

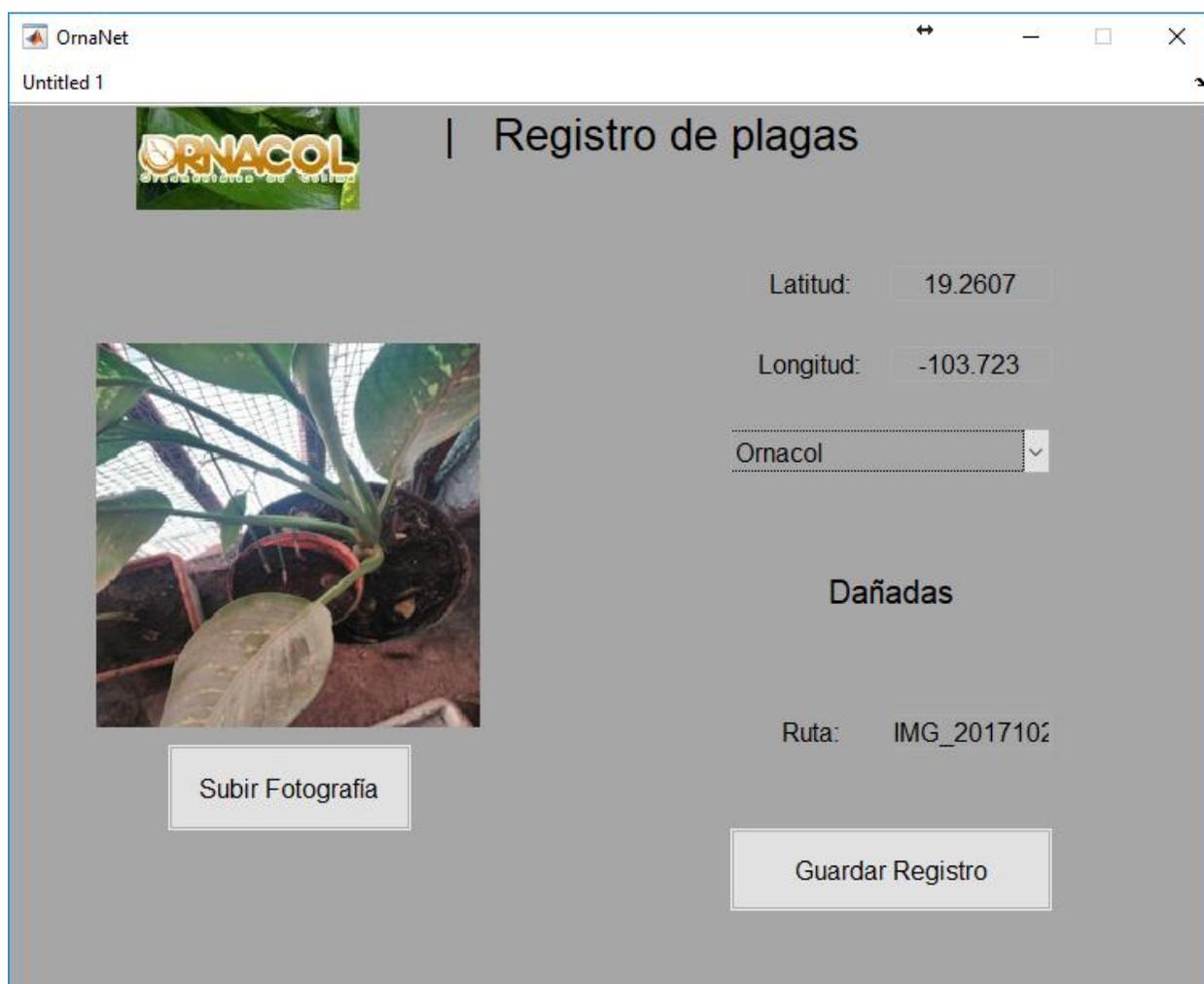
Dentro de los requerimientos solicitados por el cliente, se solicitó realizar una interfaz capaz de agregar una imagen, procesarla con la CNN, la cual permitirá predecir si la planta de la imagen en cuestión cuenta con algún daño en la hoja o bien si se encuentra sana.

En MATLAB, se cuenta con una herramienta para generar aplicaciones visuales que se invoca con el comando “guide”, el cual nos permite realizar interfaces gráficas de usuarios (GUI) para poder crear programas amigables, intuitivos y fáciles de utilizar, sustituyendo el uso de comandos, por botones o cuadros de textos, etc. En la Figura 58, se muestra la pantalla inicial de esta herramienta.



**Figura 58.** Inicio para la creación de interfaz gráfica de usuario.

De acuerdo a los requerimientos presentados sobre la interfaz gráfica, se creó una GUI con todos los componentes externados a cumplir, así como la personalización con el nombre de la empresa externa involucrada en el proyecto, como se muestra en la Figura 59.



**Figura 59.** Interfaz gráfica de usuario realizada para el servidor generada en MATLAB.

### 4.3.2. Creación de la base de datos

El requerimiento posterior fue la creación del almacenamiento para los registros, como se muestra en el modelo de Kanban de la Figura 60.

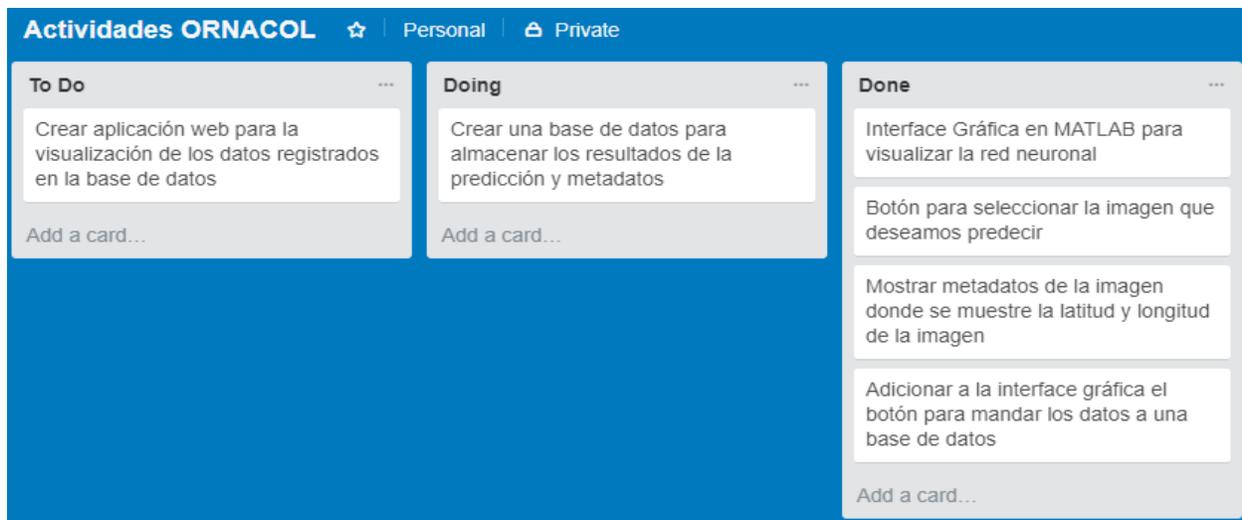


Figura 60. Tarea requerida de la creación de base de datos.

Para la creación de la base de datos, se cuentan con un mínimo de campos, los necesarios para la operación del sistema, ya que únicamente se necesita guardar cinco datos vitales de la imagen, para posteriormente mostrarlos. Para esto, se agrega el código para la creación de la base de datos, la tabla y los atributos, como se puede observar en la Figura. 61.

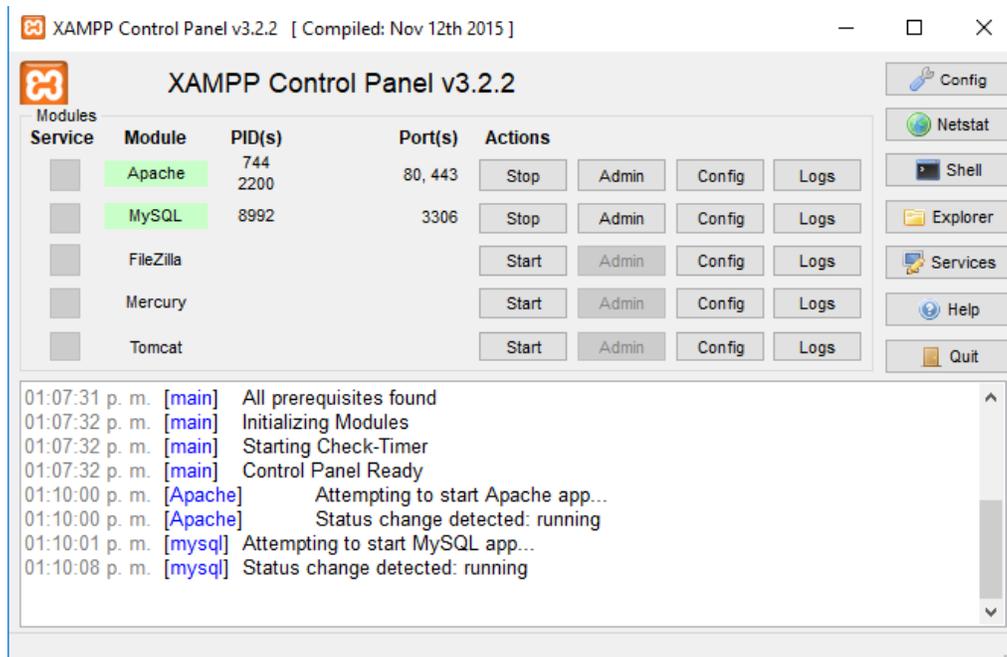
```
CREATE DATABASE IF NOT EXISTS ornacol;

USE ornacol;

CREATE TABLE imagen(
  id          int(255) auto_increment not null,
  ruta       varchar(255),
  latitud    float,
  longitud   float,
  vivero     varchar(255),
  predic     varchar(20),
  fecha      date,
  CONSTRAINT pk_imagen PRIMARY KEY(id)
)ENGINE = InnoDB;
```

**Figura 61.** Código MySQL para la creación de la base de datos.

Este código fue implementado en un servidor local, a través de la utilización de XAMPP (ver Figura. 62), el cual nos permite realizar la conexión del servidor local y la obtención de datos a través de la aplicación WEB.



**Figura 62.** Panel de control XAMPP.

El servidor MySQL que alojó la base de datos, se implementó en la dirección localhost con el gestor de base de datos “phpMyAdmin”, como se puede apreciar en la Figura 63. A través de éste, se puede observar que se encuentra la base de datos llamada “ornacol”, con una tabla llamada “imagen” y los campos requeridos para el almacenamiento de la información necesaria, como son: id, ruta, latitud, longitud, vivero, predicción (predic) y fecha. Estos datos permitirán registrar el agente patógeno.

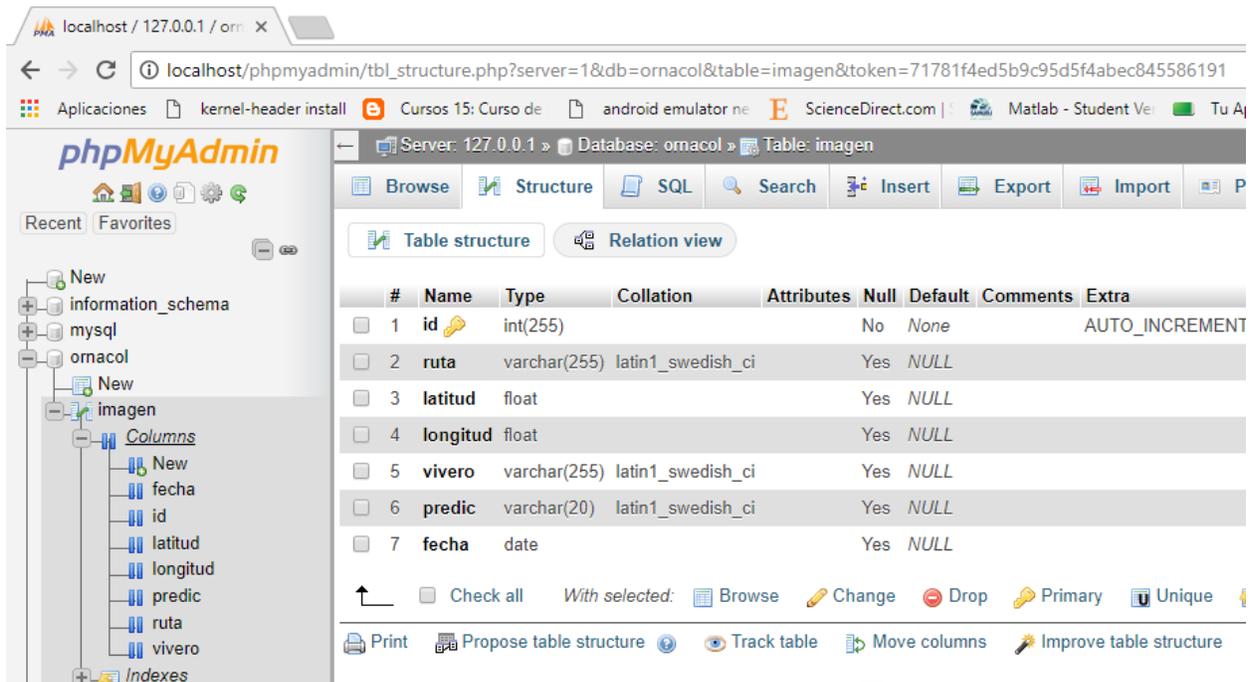
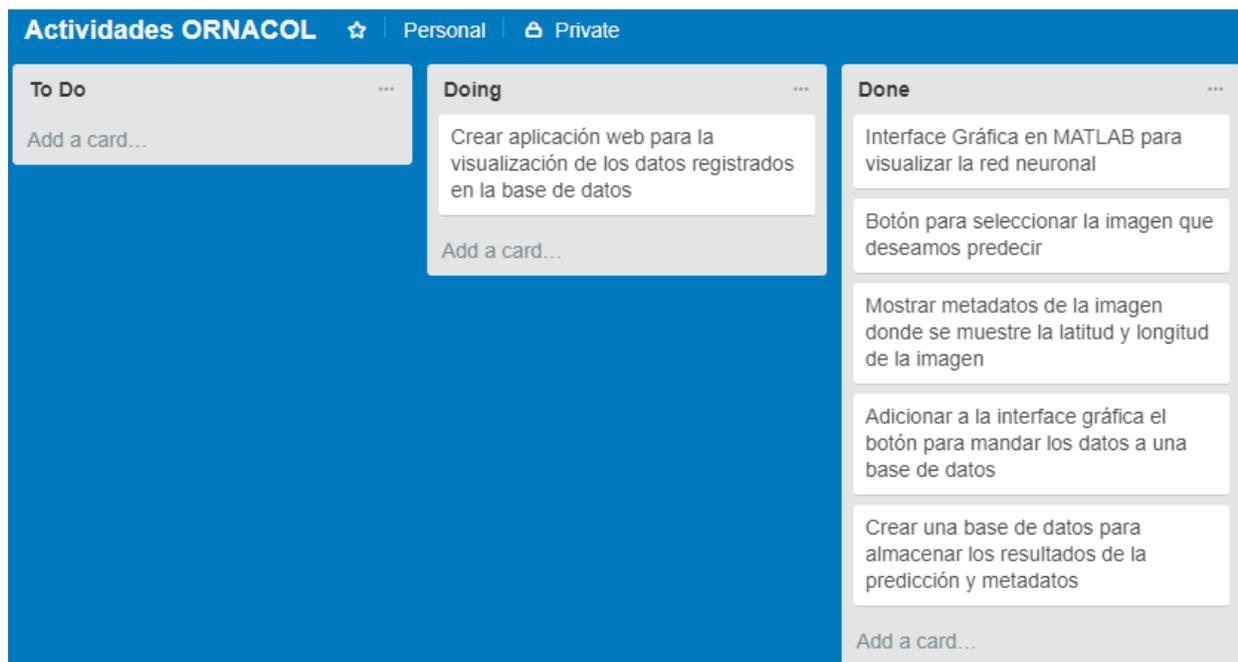


Figura 63. Servidor local MySQL.

### 4.3.3. Módulo WEB

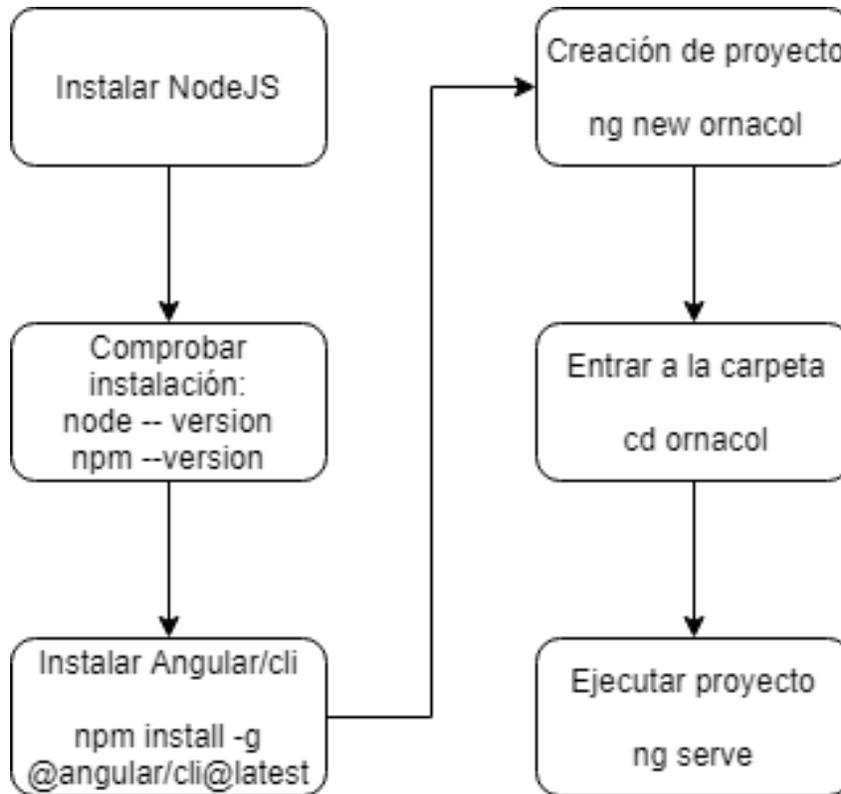
En este apartado del capítulo, se realiza la interfaz de visualización, como se menciona y establece en la metodología Kanban. En la Figura. 64, se muestra la tarea que se está realizando, en este caso, el desarrollo de la aplicación Web, enfatizando que ésta es la última tarea a realizar en la pizarra de Kanban



**Figura 64.** Tareas que se realizan en este capítulo.

Para la creación de la página WEB, se utilizó el framework Angular 5, el cual requiere la instalación de NodeJS para la utilización del lenguaje. A continuación, se muestra el proceso de la puesta a punto de Angular 5.

Se debe tener en cuenta que una vez creado el proyecto en Angular 5 se requiere de la programación y creación de los módulos dependiendo las tareas requeridas. El proceso de instalación se muestra en la Figura 65.



**Figura 65.** Proceso para crear la aplicación WEB.

Estos módulos creados se muestran en la Figura 66, mostrando todos los componentes, modelos, servicios y vistas creadas de la aplicación WEB.

Para la programación de estos módulos se utilizó el editor de código “Visual Studio Code” que ayudó en la modificación y manejo del código fuente en la aplicación WEB.

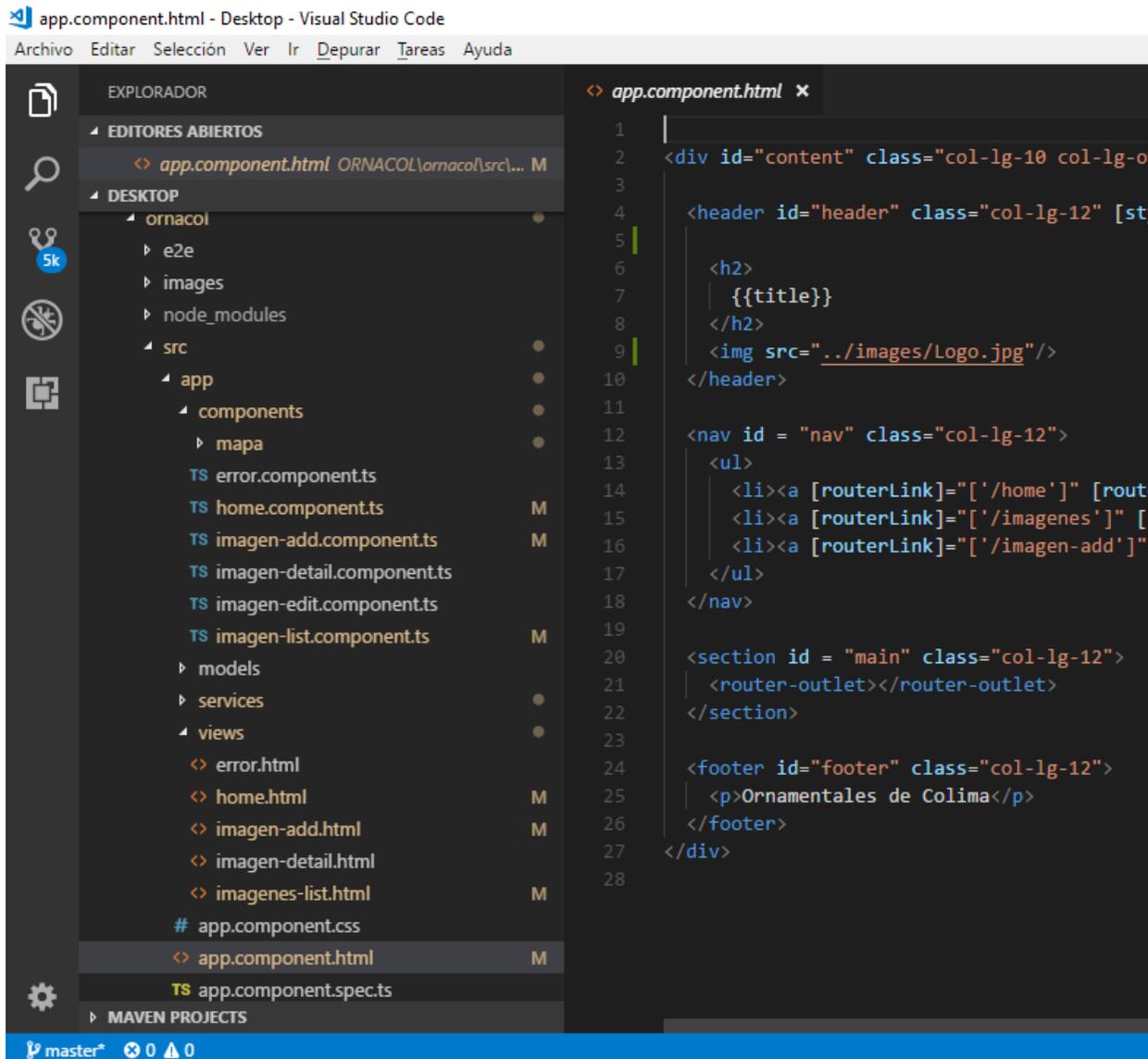
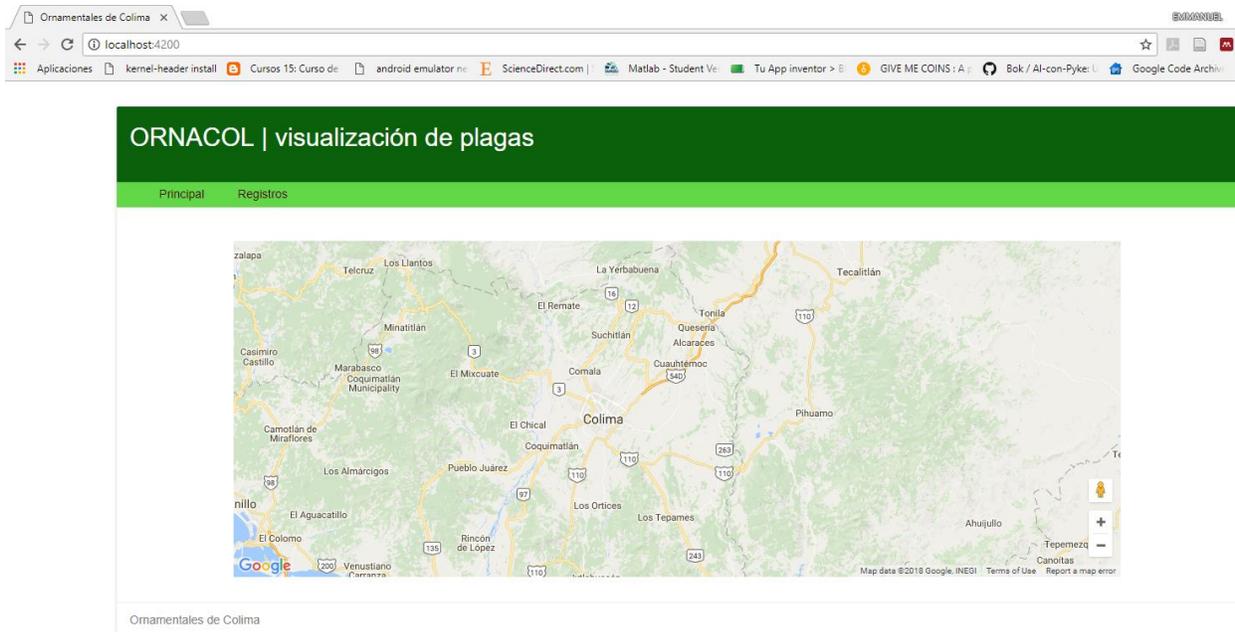


Figura 66. Módulos de la aplicación WEB realizada en Angular 5.

En el trabajo final de este capítulo, se realizó una página con dos enlaces como se puede observar en la Figura. 67. El primer enlace es la página principal en sí, la cual muestra un mapa en donde aparecerán los daños a plantas ornamentales que se registraron en cada uno de los viveros. El segundo enlace, permite visualizar los Registros, mostrando un listado de todas las imágenes procesadas, así como la información de dónde fue encontrado el patógeno, la ubicación del vivero y la predicción con que se clasificó.



**Figura 67.** Pantalla principal de la aplicación WEB.

## 4.4. Verificación y validación

### 4.4.1 Prueba de complejidad McCabe

La validación y verificación del sistema se llevó a cabo a través de la métrica de complejidad McCabe, la cual permite realizar una medición cuantitativa de la complejidad lógica de un programa de software (Saez, 2005).

Se calcula utilizando el flujograma de control de un programa donde el número ciclomático  $V(G)$  de un grafo  $G$  con “ $n$ ” vértices, “ $e$ ” bordes y “ $p$ ” componentes conectados es:  $V(G) = e - n + p$ .

Fue inventado por McCabe (1976). La Tabla 9, muestra cómo se realiza la evaluación de riesgos en función de la complejidad.

**Tabla 9.** Riesgo con base a la función de complejidad ciclomática.

Complejidad ciclomática	Evaluación de riesgo	Probabilidad de mala corrección
<b>1-10</b>	Bajo riesgo, código comprobable	5%
<b>11-20</b>	Riesgo moderado	10%
<b>21-50</b>	Alto riesgo	30%
<b>&gt;50</b>	Riesgo muy alto, código no comprobable	40%

“El número ciclomático puede entenderse como el número mínimo de caminos necesario para, mediante combinaciones, construir cualquier otro camino presente en el grafo. Utilizamos el término camino en el sentido usual, como una sucesión de nodos que puede recorrerse siguiendo arcos presentes en el grafo” (Saez, 2005).

Con base a ello, MATLAB cuenta con la herramienta interna para revisar el código y darle un valor ciclomático de complejidad del programa.

La prueba obtenida en la complejidad ciclomática del software para la red reentrenada y el software compuesto de los módulos para la extracción de metadatos se muestra a continuación:

**Tabla 10.** Resultados de la prueba McCabe.

L 1 (C 22-28): The McCabe complexity of 'OrnaNet' is 4.
L 48 (C 10-27): The McCabe complexity of 'OrnaNet_OpeningFcn' is 1.
L 72 (C 22-38): The McCabe complexity of 'OrnaNet_OutputFcn' is 1.
L 84 (C 10-30): The McCabe complexity of 'btn_buscarim_Callback' is 7.
L 156 (C 10-28): The McCabe complexity of 'btnGuardar_Callback' is 1.
L 183 (C 10-24): The McCabe complexity of 'txtLat_Callback' is 1.
L 193 (C 10-25): The McCabe complexity of 'txtLat_CreateFcn' is 3.
L 206 (C 10-25): The McCabe complexity of 'txtLong_Callback' is 1.
L 216 (C 10-26): The McCabe complexity of 'txtLong_CreateFcn' is 3.
L 229 (C 10-25): The McCabe complexity of 'txtRuta_Callback' is 1.
L 239 (C 10-26): The McCabe complexity of 'txtRuta_CreateFcn' is 3.
L 252 (C 10-28): The McCabe complexity of 'Untitled_1_Callback' is 1.
L 259 (C 10-28): The McCabe complexity of 'lstViveros_Callback' is 1.
L 272 (C 10-29): The McCabe complexity of 'lstViveros_CreateFcn' is 3.

Considerando la Tabla 10, los resultados arrojados de las pruebas de complejidad ciclomática, el software cae en la categoría 1-10, ya que el valor más alto obtenido fue de 7, como se mostró en la Tabla 9, entra en la evaluación de riesgo bajo por no sobrepasar dicha categoría en sus módulos.

#### 4.4.1. Prueba de tiempo de ejecución

Se realizó una prueba de tiempo de ejecución, a través de la herramienta “profiler” de MATLAB, la cual permite medir el tiempo que tarda en cierto módulo el programa, ayudando así a saber que parte de nuestro código puede hacer que éste sea ineficiente. Los resultados obtenidos de esta prueba se muestran en la Tabla 11.

**Tabla 11.** Resumen del profiler.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">OrnaNet</a>	1	0.431 s	0.003 s	
<a href="#">gui_mainfcn</a>	1	0.429 s	0.020 s	
<a href="#">OrnaNet&gt;OrnaNet_OpeningFcn</a>	1	0.244 s	0.107 s	
<a href="#">gui_mainfcn&gt;local_openfig</a>	2	0.124 s	0.001 s	
<a href="#">openfigLegacy</a>	2	0.123 s	0.040 s	
<a href="#">imshow</a>	1	0.101 s	0.017 s	
<a href="#">newplot</a>	2	0.061 s	0.005 s	
<a href="#">movegui</a>	2	0.048 s	0.026 s	
<a href="#">genvarname</a>	5	0.045 s	0.039 s	
<a href="#">workspacefunc</a>	2	0.044 s	0.024 s	
<a href="#">gobjects</a>	4	0.031 s	0.031 s	
<a href="#">imread</a>	1	0.029 s	0.007 s	
<a href="#">newplot&gt;ObserveAxesNextPlot</a>	2	0.024 s	0.002 s	
<a href="#">cla</a>	2	0.023 s	0.002 s	
<a href="#">findall</a>	4	0.017 s	0.012 s	
<a href="#">openfigLegacy&gt;getToken</a>	2	0.015 s	0.001 s	
<a href="#">openfigLegacy&gt;localGetFileAndOptions</a>	2	0.014 s	0.006 s	
<a href="#">imread&gt;call_format_specific_reader</a>	1	0.013 s	0.002 s	

<a href="#">basicImageDisplay</a>	1	0.012 s	0.005 s	
<a href="#">now</a>	2	0.012 s	0.004 s	
<a href="#">graphics\private\claNotify</a>	2	0.011 s	0.011 s	■
<a href="#">imagesci\private\readjpg</a>	1	0.011 s	0.001 s	■
<a href="#">imagesci\private\imjpginfo</a>	1	0.010 s	0.000 s	■
<a href="#">imjpginfo</a>	1	0.010 s	0.002 s	■
<a href="#">graphics\private\clo</a>	2	0.009 s	0.009 s	■
<a href="#">imageDisplayParseInputs</a>	1	0.009 s	0.001 s	■
<a href="#">imjpgbaselineinfo</a>	1	0.008 s	0.004 s	
<a href="#">datenum</a>	2	0.007 s	0.006 s	■
<a href="#">imageDisplayValidateParams</a>	1	0.007 s	0.002 s	
<a href="#">findFigFile</a>	2	0.007 s	0.004 s	
<a href="#">fileparts</a>	5	0.005 s	0.005 s	■
<a href="#">guidata</a>	3	0.005 s	0.004 s	■
<a href="#">fullfile</a>	2	0.005 s	0.004 s	■
<a href="#">axis</a>	1	0.005 s	0.004 s	■
<a href="#">onCleanup&gt;onCleanup.delete</a>	9	0.005 s	0.002 s	■
<a href="#">guidemfile</a>	1	0.005 s	0.001 s	■
<a href="#">isSingleImageDefaultPos</a>	1	0.004 s	0.002 s	■
<a href="#">workspacefunc&gt;getCleanupHandler</a>	2	0.004 s	0.003 s	■
<a href="#">cell.strmatch</a>	2	0.004 s	0.001 s	■
<a href="#">...;restoreToolBarToolPredefinedCallback</a>	1	0.004 s	0.000 s	■
<a href="#">rmfield</a>	1	0.004 s	0.004 s	■
<a href="#">allchild</a>	2	0.004 s	0.002 s	■
<a href="#">guidemfile&gt;getToolBarToolInFigure</a>	1	0.003 s	0.001 s	■
<a href="#">...&gt;showHiddenHandlesToFindAllHandles</a>	4	0.003 s	0.003 s	■
<a href="#">unique</a>	2	0.003 s	0.001 s	■
<a href="#">strmatch</a>	2	0.003 s	0.003 s	■

<a href="#">imjpgbaselineinfo&gt;recover_valid_marker</a>	6	0.003 s	0.003 s	■
<a href="#">imread&gt;parse_inputs</a>	1	0.002 s	0.002 s	
<a href="#">workspacefunc&gt;getWhosInformation</a>	2	0.002 s	0.002 s	
<a href="#">...pace.MatlabWorkspaceListener.swl(sw)</a>	2	0.002 s	0.002 s	
<a href="#">unique&gt;uniqueR2012a</a>	2	0.002 s	0.002 s	
<a href="#">imread&gt;get_full_filename</a>	1	0.002 s	0.002 s	
<a href="#">imagesci\private\getFileFromURL</a>	1	0.002 s	0.000 s	
<a href="#">dec2hex</a>	2	0.002 s	0.002 s	
<a href="#">settings</a>	3	0.002 s	0.002 s	
<a href="#">fullfile&gt;refinePath</a>	2	0.001 s	0.001 s	
<a href="#">contains</a>	1	0.001 s	0.001 s	
<a href="#">onCleanup&gt;onCleanup.onCleanup</a>	9	0.001 s	0.001 s	
<a href="#">intmax</a>	2	0.001 s	0.001 s	
<a href="#">imageDisplayParsePVPairs</a>	1	0.001 s	0.001 s	
<a href="#">openfigLegacy&gt;localCheckOption</a>	4	0.001 s	0.001 s	
<a href="#">...set(rootobj,'ShowHiddenHandles',Temp)</a>	4	0.001 s	0.001 s	
<a href="#">...t;preParseInputsForSpatialReferencing</a>	1	0.001 s	0.001 s	
<a href="#">guidata&gt;getParentFigure</a>	3	0.001 s	0.001 s	
<a href="#">markFigure</a>	1	0.001 s	0.001 s	
<a href="#">...selineinfo&gt;parse_app1_exif_segment</a>	1	0.001 s	0.001 s	
<a href="#">stringToLegacyText</a>	2	0.001 s	0.001 s	
<a href="#">allchild&gt;getchildren</a>	2	0.001 s	0.001 s	
<a href="#">flipud</a>	2	0.001 s	0.000 s	
<a href="#">movegui&gt;get_parent_fig</a>	2	0.001 s	0.001 s	
<a href="#">imshow&gt;getBorder</a>	1	0.001 s	0.001 s	
<a href="#">clearNotify</a>	1	0.001 s	0.001 s	
<a href="#">imagesci\private\rjpg8c</a>	1	0.000 s	0.000 s	
<a href="#">strfun\private\isTextStrict</a>	1	0.000 s	0.000 s	

<a href="#">genvarname&gt;isCharVector</a>	5	0.000 s	0.000 s	
<a href="#">isunix</a>	6	0.000 s	0.000 s	
<a href="#">newplot&gt;ObserveFigureNextPlot</a>	2	0.000 s	0.000 s	
<a href="#">imshow&gt;processReduceSyntax</a>	1	0.000 s	0.000 s	
<a href="#">iskeyword</a>	2	0.000 s	0.000 s	
<a href="#">imageDisplayValidateParams&gt;validateCData</a>	1	0.000 s	0.000 s	
<a href="#">ispc</a>	5	0.000 s	0.000 s	
<a href="#">flip</a>	1	0.000 s	0.000 s	
<a href="#">fliplr</a>	2	0.000 s	0.000 s	
<a href="#">cell.ismember</a>	1	0.000 s	0.000 s	
<a href="#">imshow&gt;preParseInputs</a>	1	0.000 s	0.000 s	
<a href="#">...set(rootobj,'ShowHiddenHandles',Temp)</a>	2	0.000 s	0.000 s	
<a href="#">hasDisplay</a>	2	0.000 s	0.000 s	
<a href="#">uitools\private\allchildRootHelper</a>	2	0.000 s	0.000 s	
<a href="#">axis&gt;allAxes</a>	1	0.000 s	0.000 s	
<a href="#">imageDisplayValidateParams&gt;validateMap</a>	1	0.000 s	0.000 s	
<a href="#">imshow&gt;convertImrefToXDataYData</a>	1	0.000 s	0.000 s	
<a href="#">checkDisplayRange</a>	1	0.000 s	0.000 s	
<a href="#">imageDisplayValidateParams&gt;findImageType</a>	1	0.000 s	0.000 s	
<a href="#">imshow&gt;@(c)isa(c,'imref2d')</a>	1	0.000 s	0.000 s	
<a href="#">...playValidateParams&gt;getCDataMapping</a>	1	0.000 s	0.000 s	
<a href="#">OrnaNet&gt;OrnaNet_OutputFcn</a>	1	0.000 s	0.000 s	
<a href="#">imjpginfo&gt;@( )fclose(fid)</a>	1	0.000 s	0.000 s	

Como se puede observar en la Tabla 11, **Self time**, es el tiempo que tarda una función desde su inicio hasta finalizar el proceso, excluyendo las funciones secundarias que se ejecuten en paralelo.

De estos resultados, se tiene más información específica de cada módulo, sin embargo, se consideró adicionar únicamente los primeros tres módulos que consumen la mayor cantidad de tiempo en ejecución y se descartaron las posteriores para no adicionar información irrelevante a este documento.

Los módulos que demandan el mayor tiempo en su ejecución se muestran en la Tabla 12.

**Tabla 12.** Módulos con mayor demanda de tiempo.

Módulo	Tiempo en segundos	Tarea
<b>OrnaNet</b>	0.431	Carga de la aplicación a memoria
<b>gui_mainfcn</b>	0.429	Carga la interfaz gráfica
<b>OrnaNet_OpeningFcn</b>	0.244	Carga los logos y propiedades de los botones

## 5. Resultados

En el proceso de clasificación las dos categorías de hojas (sanas y dañadas) para el entrenamiento de la red neuronal convolucional AlexNet se agregaron 75 imágenes por cada carpeta como se muestra en la Tabla 11.

**Tabla 13.** Conjunto de imágenes para entrenamiento.

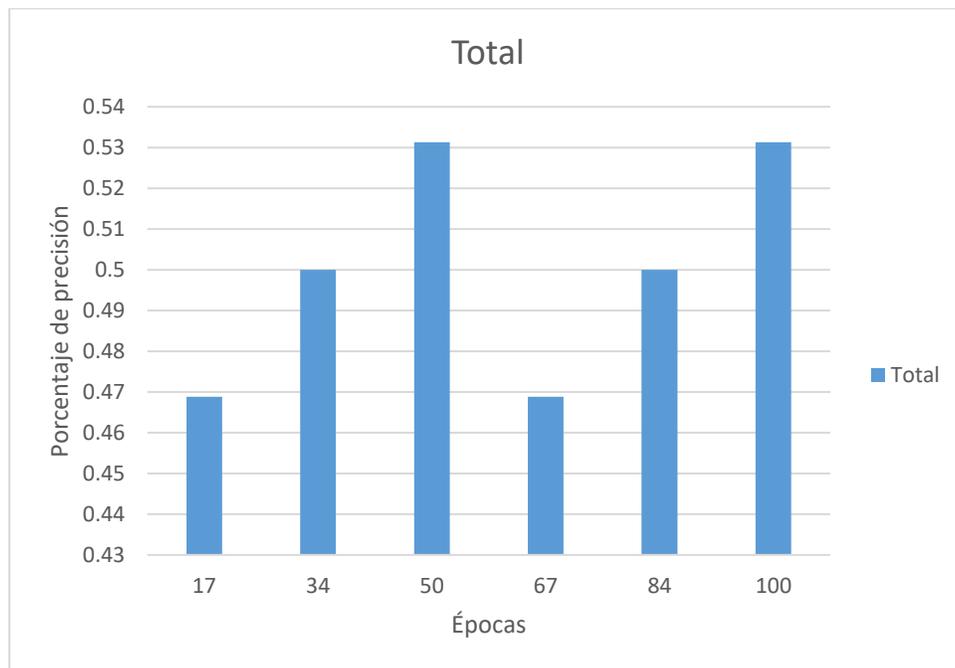
Label	Count
Dañadas	75
Sanas	75

Se realizó el entrenamiento de la red neuronal convolucional a través de 100 épocas (ciclos que pasa cada lote en la red neuronal convolucional para extraer características) y 300 iteraciones por entrenamiento.

En el primer entrenamiento de la red neuronal, los resultados de precisión en las muestras estuvieron entre el 46% y 53% de efectividad para la predicción, como se muestra en la Tabla 14 y la gráfica de estos resultados en la Figura 68.

**Tabla 14.** Resultados del primer entrenamiento.

Épocas	Iteración	Tiempo Transcurrido (segundos)	Perdida por Mini-batch	Precisión de Mini-batch	Tasa de aprendizaje base
17	50	25.1	8.469	46.88%	0.001
34	100	50.06	7.971	50.00%	0.001
50	150	75.26	7.473	53.13%	0.001
67	200	100	8.469	46.88%	0.001
84	250	124.83	7.971	50.00%	0.001
100	300	149.76	7.473	53.13%	0.001

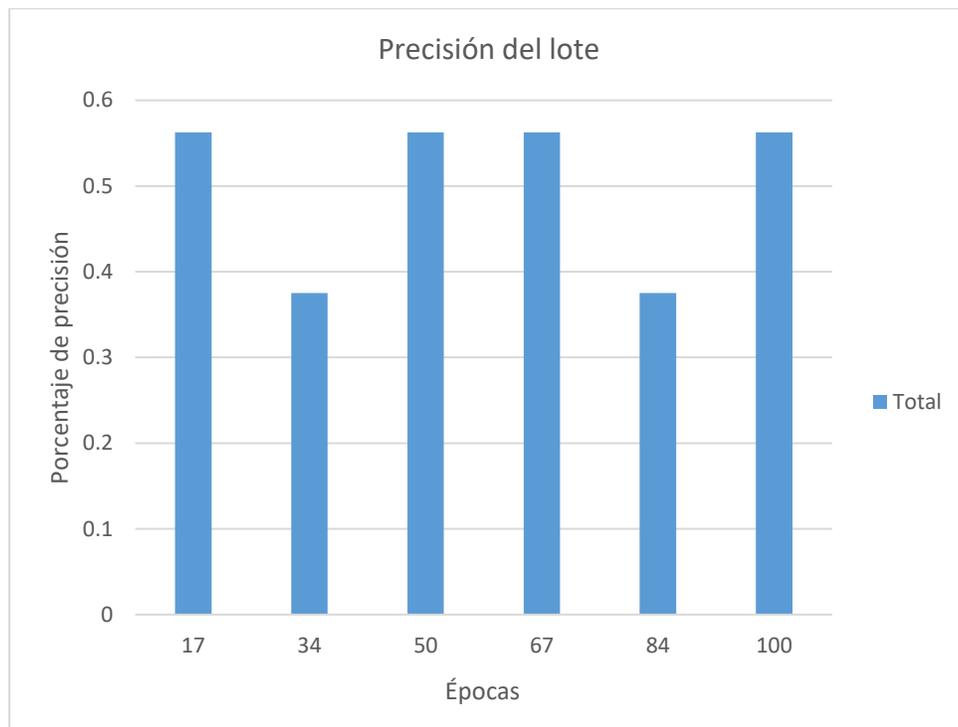


**Figura 68.** Gráfica de resultados del primer entrenamiento de la predicción.

En el segundo entrenamiento la red neuronal, los resultados de precisión en las muestras fueron del 37.50% al 56.25% en efectividad para la predicción, como se muestra en la Tabla 15 y la gráfica de estos resultados en la Figura 69.

**Tabla 15.** Resultados del segundo entrenamiento.

Épocas	Iteración	Tiempo Transcurrido (segundos)	Perdida por Mini-batch	Precisión de Mini-batch	Tasa de aprendizaje base
17	50	60.04	6.975	56.25%	0.001
34	100	119.84	9.964	37.50%	0.001
50	150	179.53	6.975	56.25%	0.001
67	200	239.68	6.975	56.25%	0.001
84	250	299.63	9.964	37.50%	0.001
100	300	359.75	6.975	56.25%	0.001

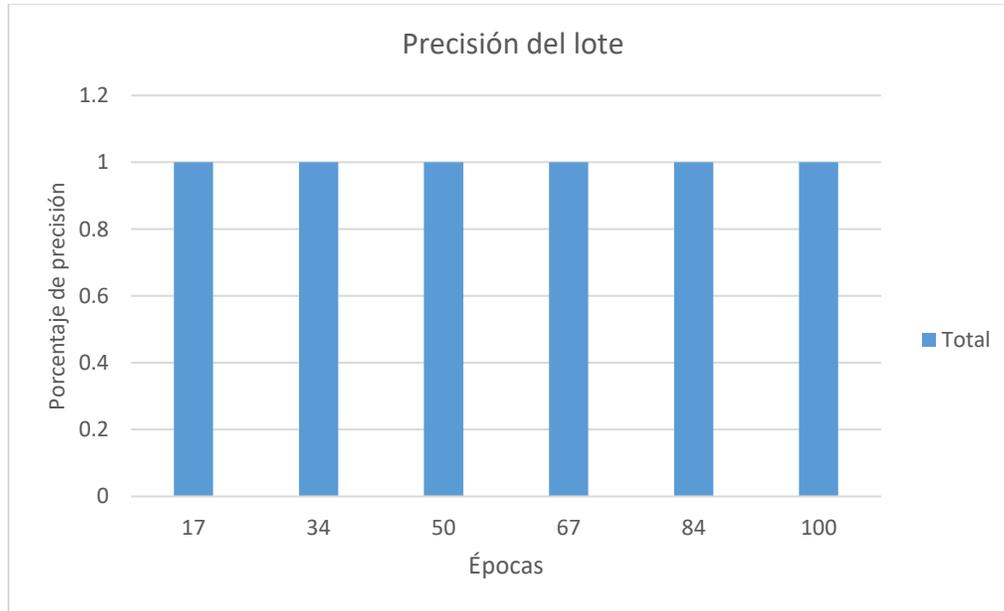


**Figura 69.** Gráfica de resultados del segundo entrenamiento de la predicción.

Para el tercer entrenamiento, la red obtuvo el 100% de efectividad en la predicción como se muestra en la Tabla 16 y la gráfica de estos resultados en la Figura 70.

**Tabla 16.** Resultados del tercer entrenamiento.

Épocas	Iteración	Tiempo Transcurrido (segundos)	Perdida por Mini-batch	Precisión de Mini-batch	Tasa de aprendizaje base
17	50	27.17	0	100.00%	0.001
34	100	54.29	0	100.00%	0.001
50	150	81.49	0	100.00%	0.001
67	200	108.63	0	100.00%	0.001
84	250	135.68	0	100.00%	0.001
100	300	162.86	0	100.00%	0.001



**Figura 70.** Gráfica de resultados del tercer entrenamiento de la predicción.

Al obtener la red entrenada al 100%, está lista para ser aplicada en la plataforma creada para la predicción de agentes patógenos, donde de modo gráfico, al momento de someter la imagen digital del agente a predecir, automáticamente la red neuronal predecirá el daño y lo mostrará como etiqueta para poder almacenar esta información en la base de datos. Como se muestra el capítulo “4.3.1.2. Creación de Interfaz Gráfica” donde se muestra la pantalla del servidor final con la predicción y metadatos extraídos de la imagen presentada.

## 6. Conclusiones y trabajo a futuro

Con los resultados obtenidos en esta investigación, queda claro que las redes neuronales son la mejor herramienta y técnica para la predicción de datos no lineales, adaptándose a cualquier problema en diferentes áreas de conocimiento.

En esta tesis de maestría se logró desarrollar una herramienta tecnológica basada en redes neuronales convolucionales, para predecir la presencia de agentes patógenos en plantas ornamentales.

Se realizó la investigación del estado del arte como se pudo observar en el Capítulo 2, el cual fundamenta que las redes neuronales convolucionales son una tecnología eficaz que puede ser aplicada para la detección de agentes patógenos en plantas ornamentales.

En el Capítulo 4, se presentó el diseño y modelo de la base de datos que permitirá almacenar la predicción y los datos necesarios de la identificación de los agentes patógenos. Además, la predicción que podrá ser visualizada a través de la aplicación WEB.

Como trabajo futuro, se recopilará información sobre diferentes tipos de agentes patógenos para poder obtener un conjunto de datos sobre las plagas que más afectan a las plantas ornamentales en la empresa ORNACOL, las cuales son: piojo harinoso, araña roja, pulgón y gusanos. Ayudando a mejorar la operación de predicción en el reconocimiento de agentes patógenos en las plantas ornamentales.

Como investigación futura, se propone realizar la automatización y procesamiento de imágenes en tiempo real.

## Bibliografía

- BALLESTEROS RIVEROS, DIANA PAOLA; BALLESTEROS SILVA, P. P. (2007). *UNA FORMA PRÁCTICA PARA APLICAR EL SISTEMA KANBAN EN LAS MYPIMES COLOMBIANAS. Scientia et technica* (Vol. 1). Universidad Tecnológica de Pereira. Retrieved from <http://www.redalyc.org/html/849/84920503036/>
- Barré, P., Stöver, B. C., Müller, K. F., & Steinhage, V. (2017). LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics*, *40*, 50–56. <https://doi.org/10.1016/j.ecoinf.2017.05.005>
- Chopade, P. B., & P, K. B. (2016). Image processing Based Detection and classification of leaf disease on fruits crops 1 1. *Special Issue on International Journal of Electrical*, (February), 33–36. Retrieved from [http://www.irdindia.in/journal\\_ijeecs/pdf/vol4\\_iss2/10.pdf](http://www.irdindia.in/journal_ijeecs/pdf/vol4_iss2/10.pdf)
- Diego Calvo. (2018). Red neuronal Convolutacional CNN. Retrieved April 1, 2018, from <http://www.diegocalvo.es/red-neuronal-convolutacional-cnn/>
- Dr. Marianne Altmann. (2016). La situación del mercado de las flores y plantas | IPM ESSEN. Retrieved August 17, 2017, from <https://www.ipm-essen.de/press/press-texts-international/detail/la-situacion-del-mercado-de-las-flores-y-plantas-1241>
- Druzhkov, P. N., & Kustikova, V. D. (2016). A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, *26*(1), 9–15. <https://doi.org/10.1134/S1054661816010065>
- Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*, *151*, 72–80. <https://doi.org/10.1016/j.biosystemseng.2016.08.024>

Español: Diagrama de las Etapas de Vision Artificial. (2016, April). Retrieved from [https://commons.wikimedia.org/wiki/File:Etapas\\_vision\\_artificial.png](https://commons.wikimedia.org/wiki/File:Etapas_vision_artificial.png)

Felix, P. :, & Bellido, S. (1994). LA HORTICULTURA ORNAMEN TAL Una alternativa ante la situación agrícola actual. *ISSN 0002-1334*, 206–213. Retrieved from [http://www.mapama.gob.es/ministerio/pags/biblioteca/revistas/pdf\\_Agri/Agri\\_1994\\_740\\_206\\_213.pdf](http://www.mapama.gob.es/ministerio/pags/biblioteca/revistas/pdf_Agri/Agri_1994_740_206_213.pdf)

Fernández García, N. L. (2012). Introducción a la Visión Artificial Visión Artificial Avanzada, 1–35. Retrieved from <http://www.uco.es/users/ma1fegan/2012-2013/vision/Temas/tema-1.pdf>

Gando, G., Yamada, T., Sato, H., Oyama, S., & Kurihara, M. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. *Expert Systems with Applications*, 66, 295–301. <https://doi.org/10.1016/j.eswa.2016.08.057>

Garcia, J., & Barbedo, A. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases. *A SpringerOpen Journal Barbedo SpringerPlus*, 2. Retrieved from <http://www.springerplus.com/content/2/1/660>

Ghaiwat, S. N., & Arora, P. (2014). Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review. *International Journal of Recent Advances in Engineering & Technology (IJRAET) ISSN (Online)*, (2), 2347–2812. Retrieved from [http://www.irdindia.in/journal\\_ijraet/pdf/vol2\\_iss3/1.pdf](http://www.irdindia.in/journal_ijraet/pdf/vol2_iss3/1.pdf)

Gupta, L., Intwala, K., Khetwani, K., Hanamshet, T., & Somkunwar, R. (2017). Smart Irrigation System and Plant Disease Detection. *International Research Journal of Engineering and Technology*, 4(3), 2395–56. Retrieved from <https://www.irjet.net/archives/V4/i3/IRJET-V4I321.pdf>

Haykin, S. (2001). *Neural Networks and Learning Machines Third Edition*, 5. Retrieved

from [https://cours.etsmtl.ca/sys843/REFS/Books/ebook\\_Haykin09.pdf](https://cours.etsmtl.ca/sys843/REFS/Books/ebook_Haykin09.pdf)

IPM ESSEN. (2017). Market Description of the Flower and Plant Markets. Retrieved from <http://www.florinews.com/index.php/past-featured-articles/40-past-featured-articles/258-market-description-of-the-flower-markets>

Juan, R. Q., & Chacón, A. (2011). Redes neuronales artificiales para el procesamiento de imágenes , una revisión de la última década. *Riee&C*, 9(1), 7–16. Retrieved from [http://www.itson.mx/publicaciones/rieeyc/Documents/v9/art2vf\\_redes\\_neuronales\\_artificiales\\_para\\_el\\_procesamiento\\_de\\_imagenes-una\\_revision\\_de\\_la\\_ultima\\_decada.pdf](http://www.itson.mx/publicaciones/rieeyc/Documents/v9/art2vf_redes_neuronales_artificiales_para_el_procesamiento_de_imagenes-una_revision_de_la_ultima_decada.pdf)

Khirade, S. D., & Patil, A. B. (2015). Plant Disease Detection Using Image Processing. In *2015 International Conference on Computing Communication Control and Automation* (pp. 768–771). IEEE. <https://doi.org/10.1109/ICCUBEA.2015.153>

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012a). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems(NIPS)*, 1097–1105. <https://doi.org/10.1117/12.2176558>

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9. <https://doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007>

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>

Lee, S. H., Chang, Y. L., Chan, C. S., & Remagnino, P. (2016). Plant identification system based on a convolutional neural network for the lifeclef 2016 plant classification task. In *CEUR Workshop Proceedings* (Vol. 1609, pp. 502–510).

Retrieved from <http://ceur-ws.org/Vol-1609/16090502.pdf>

Luis, I., & Carreto, G. (2014). LA IMPORTANCIA DEL SECTOR ORNAMENTAL COMO UN POTENCIAL DE ALTO CONTENIDO DE PARTICIPACIÓN SOCIAL. Retrieved from

[http://snics.sagarpa.gob.mx/Documents/Presentaciones\\_FIRA\\_2014/2\\_La\\_importancia\\_del\\_sector\\_ornamental\\_como\\_un\\_potencial\\_de\\_alto\\_contenido\\_de\\_participación\\_social](http://snics.sagarpa.gob.mx/Documents/Presentaciones_FIRA_2014/2_La_importancia_del_sector_ornamental_como_un_potencial_de_alto_contenido_de_participación_social). Ing.Luis\_Granada\_Carreto.pdf

Mainkar, P. M., Gorpade, S., & Adawadkar, M. (2015). Plant Leaf Disease Detection and Classification Using Image Processing Techniques. *International Journal of Innovative and Emerging Research in Engineering*, 2(4), 139–144. Retrieved from [http://ijiere.com/\(X\(1\)S\(4cvpimbqdznduhye35wsoy\)\)/FinalPaper/FinalPaper20155411310389.pdf](http://ijiere.com/(X(1)S(4cvpimbqdznduhye35wsoy))/FinalPaper/FinalPaper20155411310389.pdf)

Mccabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, (4), 308–320. <https://doi.org/10.1109/TSE.1976.233837>

Mehdipour Ghazi, M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing*, 235, 228–235. <https://doi.org/10.1016/j.neucom.2017.01.018>

Pablo, P., García, G., Antonio, J., & Orozco, L. (2012). RECONOCIMIENTO DE IMÁGENES UTILIZANDO REDES NEURONALES ARTIFICIALES. Retrieved from <http://eprints.ucm.es/23444/1/ProyectoFinMasterPedroPablo.pdf>

Palomares, F. G., Serrá, J. A. M., & Martínez, E. A. (2016). Modelling in Science Education and Learning MSEL. *Modelling in Science Education and Learning*, 9(1), 97–108. <https://doi.org/10.4995/msel.2016.4524>

Pethybridge, S. J., & Nelson, S. C. (2015). Leaf Doctor: A New Portable Application for Quantifying Plant Disease Severity. *Plant Disease*, 99(10), 1310–1316.

<https://doi.org/10.1094/PDIS-03-15-0319-RE>

Plan Rector Comité Sistema Producto Ornamentales Colima 2 Contenido. (2012). *Coepplants*, 34. Retrieved from [http://dev.pue.itesm.mx/sagarpa/estatales/EPT/COMITE SISTEMA PRODUCTO ORNAMENTALES COLIMA/PLAN RECTOR QUE CONTIENE PROGRAMA DE TRABAJO 2012/PR\\_ORNAMENTALES\\_COLIMA\\_2012.pdf](http://dev.pue.itesm.mx/sagarpa/estatales/EPT/COMITE%20SISTEMA%20PRODUCTO%20ORNAMENTALES%20COLIMA/PLAN%20RECTOR%20QUE%20CONTIENE%20PROGRAMA%20DE%20TRABAJO%202012/PR_ORNAMENTALES_COLIMA_2012.pdf)

René Shun. (2015). *Sistemas de Visión Artificial, Historia, Componentes y Procesamiento de Imágenes*. Retrieved May 17, 2018, from <https://www.scribd.com/doc/259914658/Sistemas-de-Vision-Artificial-Historia-Componentes-y-Procesamiento-de-Imagenes>

Robles-Camarillo, D., Paris, S., Nakano-Miyatake, M., & Álvaro Luna-González, J. (2016). Comparación de Arquitecturas de Redes Neuronales Convolucionales para la Clasificación de Imágenes de Ojos. Retrieved from [https://www.researchgate.net/publication/309455781\\_Comparacion\\_de\\_Arquitecturas\\_de\\_Redess\\_Neuronales\\_Convolucionales\\_para\\_la\\_Clasificacion\\_de\\_Imagenes\\_de\\_Ojos](https://www.researchgate.net/publication/309455781_Comparacion_de_Arquitecturas_de_Redess_Neuronales_Convolucionales_para_la_Clasificacion_de_Imagenes_de_Ojos)

Saez. (2005). *Medidas de la complejidad del software*. Retrieved from [http://dit.upm.es/~fsaez/intl/libro\\_complejidad/14-medidas-de-la-complejidad-del-software.pdf](http://dit.upm.es/~fsaez/intl/libro_complejidad/14-medidas-de-la-complejidad-del-software.pdf)

SAGARPA. (2009). *La infraestructura y sistemas requeridos para el desarrollo de clusters de horticultura ornamental orientado a la exportacion*. Retrieved from [http://www.sagarpa.gob.mx/agronegocios/Documents/Estudios\\_promercado/ORNAMENTAL.pdf](http://www.sagarpa.gob.mx/agronegocios/Documents/Estudios_promercado/ORNAMENTAL.pdf)

Shaoqing Ren, Kaiming He, Ross Girshick, and J. S. (2015). 暗小汐的专栏 论文笔记 : Faster R-CNN : Towards Real-Time Object Detection with, 39(1096), 1137–1149. <https://doi.org/10.1007/s11263-013-0620-5>

- Singh, S., & Bhamrah, M. S. (2015). Leaf Identification Using Feature Extraction and Neural Network. *IOSR Journal of Electronics and Communication Engineering*, 5(1), 134–140. <https://doi.org/10.9790/2834-1051134140>
- Singh, V., & Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture*, 4(1), 41–49. <https://doi.org/10.1016/j.inpa.2016.10.005>
- Singh, V., Varsha, & Misra, A. K. (2015). Detection of unhealthy region of plant leaves using image processing and genetic algorithm. In *Conference Proceeding - 2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015* (pp. 1028–1032). IEEE. <https://doi.org/10.1109/ICACEA.2015.7164858>
- Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience*, 2016, 1–11. <https://doi.org/10.1155/2016/3289801>
- Varshney, S. (2016). Plant Disease Prediction using Image Processing Techniques- A Review. *International Journal of Computer Science and Mobile Computing*, 5(5), 394–398. Retrieved from <http://www.ijcsmc.com/docs/papers/May2016/V5I5201682.pdf>
- Vega, H., Cortez, A., Huayna, A., Loayza, L. y Naupari, P. (2009). Reconocimiento de patrones mediante redes neuronales artificiales. *Revista de Ingeniería de Sistemas e Informática*, 6(2), 17–26. Retrieved from [http://sisbib.unmsm.edu.pe/Bibvirtual/publicaciones/risi/2009\\_n2/v6n2/a03v6n2.pdf](http://sisbib.unmsm.edu.pe/Bibvirtual/publicaciones/risi/2009_n2/v6n2/a03v6n2.pdf)
- Verdin, E., Wipf-Scheibel, C., Gognalons, P., Aller, F., Jacquemond, M., & Tepfer, M. (2017, May). Sequencing viral siRNAs to identify previously undescribed viruses and viroids in a panel of ornamental plant samples structured as a matrix of pools.

*Virus Research*. <https://doi.org/10.1016/j.virusres.2017.05.019>

VijayaLakshmi, B., & Mohan, V. (2016). Kernel-based PSO and FRVM: An automatic plant leaf type detection using texture, shape, and color features. *Computers and Electronics in Agriculture*, 125, 99–112.

<https://doi.org/10.1016/j.compag.2016.04.033>

Visión artificial. (2017, May). Retrieved from

[https://es.wikipedia.org/w/index.php?title=Visión\\_artificial&oldid=99252705](https://es.wikipedia.org/w/index.php?title=Visión_artificial&oldid=99252705)

Vovides, A. P., Linares, E., & Bye, R. (2010). Jardines Botánicos de México: historia y perspectivas, 232. Retrieved from

[https://www.sev.gob.mx/servicios/publicaciones/serie\\_hcyt/jardines\\_botanicos.pdf](https://www.sev.gob.mx/servicios/publicaciones/serie_hcyt/jardines_botanicos.pdf)

wikipedia. (2017, July). Planta ornamental. In *Wikipedia, la enciclopedia libre*.

Wikipedia. (2018). Redes neuronales convolucionales - Wikipedia, la enciclopedia libre.

Retrieved May 17, 2018, from

[https://es.wikipedia.org/wiki/Redes\\_neuronales\\_convolucionales](https://es.wikipedia.org/wiki/Redes_neuronales_convolucionales)

Yanez, E. (n.d.). CAPITULO I ASPECTOS GENERALES SOBRE VIVEROS, PRODUCTOS (PLANTAS) ORNAMENTALES Y PEQUEÑA Y MEDIANA EMPRESA. Retrieved from

<http://ri.ufg.edu.sv/jspui/bitstream/11592/8118/2/631.52-F634p-CAPITULO I.pdf>

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding Neural Networks Through Deep Visualization. Retrieved from

<http://yosinski.com/deepvis>

Yu, F., & Koltun, V. (2016). Multi-Scale Context Aggregation by Dilated Convolutions.

<https://doi.org/10.16373/j.cnki.ahr.150049>