



# PROCESAMIENTO DIGITAL DE IMÁGENES UTILIZANDO REDES NEURONALES PARA IDENTIFICAR LA SIGATOKA NEGRA EN EL CULTIVO DE PLÁTANO

## TESIS

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:  
ING. ISIS SIOMARA GARCÍA ALCALÁ

DIRECTOR DE TESIS:  
DR. NOEL GARCÍA DÍAZ

CO-DIRECTOR:  
MTRO. JUAN GARCÍA VIRGEN

VILLA DE ÁLVAREZ, COLIMA. 14 DE AGOSTO DE 2020.





# EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®

Instituto Tecnológico de Colima

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Villa de Álvarez, Colima, **4/Septiembre/2020**

Oficio No. DEPI 1.2.11/112/2020

**ALUMNA GARCÍA ALCALÁ ISIS SIOMARA  
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 **"PROCESAMIENTO DIGITAL DE IMÁGENES UTILIZANDO REDES NEURONALES PARA IDENTIFICAR LA SIGATOKA NEGRA EN EL CULTIVO DE PLÁTANO"**, por la opción de tesis, que para obtener el grado de Maestra en Sistemas Computacionales será presentada por Usted, tiene a bien concederle la **AUTORIZACIÓN** de impresión de la tesis citada.

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

**ATENTAMENTE**  
Excelencia en Educación Tecnológica®

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



SEP - TecNM  
INSTITUTO TECNOLÓGICO  
De Colima  
DIVISIÓN DE ESTUDIOS DE  
POSGRADO E INVESTIGACIÓN

REC/cas

C.p. Archivo.



## Epígrafe

*“Sabemos muy bien que lo que estamos haciendo no es más que una gota en el océano. Pero si esa gota no estuviera allí, al océano le faltaría algo.”*

-Madre Teresa de Calcuta

## **Agradecimientos**

Al Instituto Tecnológico de Colima, por abrirme sus puertas durante mi estancia en el posgrado y convertirse en mi segundo hogar.

Al Consejo Nacional de Ciencia y Tecnología, por el apoyo y sustento recibido en el transcurso de la maestría.

A los maestros y revisores, por su tiempo, dedicación, entrega y enseñanzas, especialmente al doctor Noel García Díaz, por su destacada labor de asesor y guía en el complejo camino del aprendizaje y superación.

A mis seis musas de vida, que sin su presencia en mi existir me vería reducida a nada. A su manera, en su tiempo y desde su plano existencial, pero siempre conmigo.

## Resumen

La producción de banano y plátano a nivel mundial asciende a 145 millones de toneladas año. En México, existen más de 77,000 hectáreas dedicadas al cultivo de bananos y plátanos, posicionándolo en el decimosegundo lugar a nivel mundial, y sexto lugar en la tabla de producción en América Latina. El estado de Colima ocupa el lugar número 4 en producción nacional, con 178,487 toneladas según las estadísticas obtenidas de la SAGARPA en el año 2018, siendo esta una de las actividades más importantes del estado, por lo tanto, garantizar la calidad del producto se vuelve un factor básico para mantener y mejorar las estadísticas de exportación y con esto favorecer la economía estatal y nacional. Los bananos, debido a las condiciones bajo las que se desarrollan, son susceptibles a la aparición de plagas y enfermedades, las cuales, en su mayoría, comienzan su aparición y desarrollo de manera casi imperceptible. Actualmente, la enfermedad conocida como la Sigatoka Negra es el principal problema fitosanitario que afecta la producción de este frutal en el país, así como en América Central, Sudamérica y el Caribe, generando pérdidas que abarcan desde el 50 hasta el 100% del producto. Para ayudar a mantener el control de calidad del banano, se propone la realización de un sistema de procesamiento digital de imágenes utilizando redes neuronales, para la identificación de la Sigatoka Negra en el cultivo de plátano. Este trabajo de investigación se desarrolla en áreas de siembra ubicadas en el estado de Colima. Para su implementación se utilizó una cámara digital de alta tecnología, con una resolución de 16 megapíxeles y zoom óptico de 83x, para posteriormente ser procesadas utilizando técnicas de procesamiento de imágenes y redes neuronales. Los resultados obtenidos mejoran las investigaciones previas al no necesitar Internet para llevar a cabo las tareas de análisis y detección, así como proveer mayor precisión en los resultados debido al entrenamiento de la red, utilizando un número superior de datos a los previamente presentados. Con este trabajo de investigación se puede concluir que el uso de redes neuronales contribuye a la detección oportuna de la Sigatoka Negra, evitando su proliferación.

## Abstract

The Banana worldwide production amounts to 145 million tons per year. In Mexico, there are more than 77,000 hectares dedicated to the cultivation of bananas, ranking in twelfth place worldwide, and sixth place in the production table in Latin America. The state of Colima occupies the number 4 place in national production, with 178,487 tons according to the statistics obtained from SAGARPA in 2018, being one of the most important activities in the state, therefore, the quality of the product returns to produce a basic factor to maintain and improve export statistics and with this favor the state and national economy. Bananas, due to the conditions under which they develop, are susceptible to the appearance of pests and diseases, which, for the most part, begin their appearance and development almost imperceptibly. Currently, the disease known as Black Sigatoka is the main phytosanitary problem that affects the production of this fruit in the country, as well as in Central America, South America and the Caribbean, generating losses ranging from 50 to 100% of the product. To help maintain quality control of bananas, it proposes the realization of a digital image processing system using neural networks, for the identification of Black Sigatoka in banana cultivation. This research work is carried out in planting areas located in the state of Colima. For its implementation, use a high-tech digital camera, with a resolution of 16 megapixels and an optical zoom of 83x, to be further processed using image processing techniques and neural networks. The results obtained improve previous investigations by not needing the Internet to carry out the analysis and detection tasks, as well as providing greater precision in the results due to network training, using a higher number of data than those previously affected. With this research work it can be concluded that the use of neural networks contributes to the timely detection of Black Sigatoka, preventing its proliferation.

## Indice General

Epígrafe .....	II
Agradecimientos .....	III
Resumen .....	IV
Abstract.....	V
I.- INTRODUCCIÓN.....	1
<b>1.1.- Naturaleza del problema .....</b>	<b>1</b>
<b>1.2.- Contexto del problema.....</b>	<b>2</b>
<b>1.3.- Revisión de la literatura .....</b>	<b>3</b>
<b>1.4.- Planteamiento de la solución del problema a investigar .....</b>	<b>5</b>
<b>1.5.- Justificación .....</b>	<b>6</b>
<b>1.6.- Motivación .....</b>	<b>6</b>
<b>1.7.- Objetivos.....</b>	<b>6</b>
1.7.1.- Objetivo general.....	7
1.7.2.- Objetivos específicos .....	7
<b>1.8.- Metas .....</b>	<b>7</b>
<b>1.9.- Hipótesis .....</b>	<b>7</b>
<b>1.10.- Métodos y herramientas .....</b>	<b>7</b>
<b>1.11.- Organización de la tesis.....</b>	<b>8</b>
II. ESTADO DEL CAMPO DEL CONOCIMIENTO.....	9
<b>2.1 El Marco Histórico .....</b>	<b>9</b>
2.1.1 Agentes patógenos del cultivo de plátano .....	9
2.1.2 Sigatoka Negra (SN) .....	10
2.1.3 Procesamiento de imágenes .....	14
2.1.4 Redes neuronales (NN).....	15
<b>2.2 El Marco Contextual .....</b>	<b>16</b>
<b>2.3 El Marco Teórico.....</b>	<b>17</b>
2.3.1 Procesamiento de imágenes .....	17
2.3.2 Inteligencia Artificial .....	17
2.3.3 Red neuronal New Feed-Forward .....	18

2.3.4 Aplicación de escritorio .....	19
2.3.5 MATLAB .....	20
2.3.6 MySQL.....	20
III. MÉTODOS EMPLEADOS .....	21
<b>3.1 Investigación preliminar .....</b>	<b>21</b>
<b>3.2 Análisis de requerimientos .....</b>	<b>21</b>
<b>3.3 Planeación .....</b>	<b>21</b>
<b>3.4 Descripción de metodología.....</b>	<b>21</b>
<b>3.5 Hardware y Software .....</b>	<b>23</b>
3.5.1 Hardware .....	23
3.5.2 Software.....	24
<b>3.6 Diseño y modelado del prototipo .....</b>	<b>24</b>
<b>3.7 Construcción .....</b>	<b>24</b>
<b>3.8 Implementación del prototipo.....</b>	<b>24</b>
<b>3.9 Documentación.....</b>	<b>25</b>
IV. DESARROLLO DE LA TESIS.....	26
<b>4.1 Análisis.....</b>	<b>26</b>
4.1.1 Modelo conceptual (Big Picture).....	26
4.1.2 Diagramas de actividades con responsabilidades .....	27
4.1.3 Modelo de casos de uso .....	28
4.1.4 Modelo de requisitos .....	29
<b>4.2 Diseño .....</b>	<b>31</b>
4.2.2 Modelo de datos.....	31
4.2.3 Diccionario de datos.....	32
4.2.4 Diagrama de navegación .....	34
4.2.5 Modelo de interfaces.....	35
4.2.6 Modelo de componentes .....	37
<b>4.3 Implementación .....</b>	<b>38</b>
4.3.1 Servidor local .....	38
4.3.2 Entrenamiento de la NN NFF .....	39
4.3.3 Creación de interfaz gráfica .....	40
4.3.4 Creación de la base de datos.....	45
<b>4.4 Verificación y Validación .....</b>	<b>49</b>

4.4.1 Pruebas unitarias .....	50
4.4.2 Pruebas de integración .....	50
4.4.3 Pruebas de sistema .....	51
4.4.4 Prueba de aceptación .....	51
4.4.5 Prueba de tiempo de ejecución .....	51
V. RESULTADOS OBTENIDOS .....	61
VI. CONCLUSIONES Y TRABAJO FUTURO.....	87
VII.- REFERENCIAS BIBLIOGRÁFICAS.....	88

## Índice de tablas y figuras

<b>Figura 1:</b> Modelo conceptual .....	6
<b>Figura 2:</b> Foto estado 1 .....	11
<b>Figura 3:</b> Foto estado 2 .....	12
<b>Figura 4:</b> Foto estado 3 .....	12
<b>Figura 5:</b> Foto estado 4 .....	13
<b>Figura 6:</b> Foto estado 5 .....	13
<b>Figura 7:</b> Foto estado 6 .....	13
<b>Figura 8:</b> Localización predio cultivo de plátano .....	16
<b>Figura 9:</b> Cuatro definiciones IA (Stuart J. Rusell, 2004) .....	18
<b>Figura 10:</b> Arquitectura new feed-forward (Haykin, 1 Feedforward neural networks: an introduction, 2004) .....	19
<b>Figura 11:</b> Metodología PUA (Ambles, 2005) .....	23
<b>Figura 12:</b> Big Picture-Modelo conceptual .....	26
<b>Figura 13:</b> Diagrama de Actividades toma de muestras .....	27
<b>Figura 14:</b> Diagrama actividades proceso .....	28
<b>Figura 15:</b> Diagrama caso de uso .....	29
<b>Figura 16:</b> Modelo de clases .....	31
<b>Figura 17:</b> Modelo de datos .....	32
<b>Figura 18:</b> Diagrama de Navegación .....	34
<b>Figura 19:</b> Interfaz principal .....	35
<b>Figura 20:</b> Interfaz consulta de reportes .....	36
<b>Figura 21:</b> Interfaz inspección-reporte .....	37
<b>Figura 22:</b> Modelo de componentes .....	38
<b>Figura 23:</b> Pantalla principal MATLAB .....	39
<b>Figura 24:</b> Entrenamiento NN NFF. Carga de imágenes .....	39
<b>Figura 25:</b> Entrenamiento NFF: creación NN .....	40
<b>Figura 26:</b> Ventana comandos MATLAB .....	40
<b>Figura 27:</b> Ventana GUI MATLAB .....	41
<b>Figura 28:</b> Interfaz menú principal Sigatection .....	42
<b>Figura 29:</b> Interfaz consulta de reportes .....	43
<b>Figura 30:</b> Interfaz inspección .....	44
<b>Figura 31:</b> Interfaz Apache .....	46
<b>Figura 32:</b> Fragmento BD tabla uno .....	46
<b>Figura 33:</b> Fragmento BD tabla 2 .....	47
<b>Figura 34:</b> Fragmento disparador uno .....	47
<b>Figura 35:</b> Database connectivity and reporting .....	48
<b>Figura 36:</b> Conexión ODBC .....	48
<b>Figura 37:</b> Conexión DB con MATLAB .....	49
<b>Figura 38:</b> Modelo de pruebas en V .....	50
<b>Figura 39:</b> Diagnóstico erróneo del sistema: hoja grado 2 .....	75
<b>Figura 40:</b> Diagnóstico acertado del sistema: hoja grado 1 .....	83

<b>Figura 41:</b> Diagnóstico acertado del sistema: hoja grado 0 .....	84
<b>Figura 42:</b> Diagnóstico acertado del sistema: hoja grado 3 .....	85
<b>Tabla 1:</b> Requisitos funcionales .....	29
<b>Tabla 2:</b> Requisitos no funcionales .....	30
<b>Tabla 3:</b> Diccionario de datos inspeccion.....	32
<b>Tabla 4:</b> Diccionario de datos sección .....	32
<b>Tabla 5:</b> Diccionario de datos planta.....	33
<b>Tabla 6:</b> Diccionario de datos hoja.....	33
<b>Tabla 7:</b> Diccionario de datos diagnóstico .....	33
<b>Tabla 8:</b> Prueba de tiempo de ejecución interfaz principal .....	52
<b>Tabla 9:</b> Prueba tiempo de ejecución interfaz reportes .....	55
<b>Tabla 10:</b> Tabla tiempo de ejecución interfaz inspección .....	58
<b>Tabla 11:</b> Conjunto de imágenes primer entrenamiento .....	61
<b>Tabla 12:</b> Conjunto de imágenes segundo entrenamiento .....	68
<b>Tabla 13:</b> Conjunto de imágenes tercer entrenamiento .....	75

## I.- INTRODUCCIÓN

El cultivo de plátano es una actividad agrícola con gran importancia a nivel mundial, debido a que es una buena fuente de ingresos y proporciona un gran aporte comercial. Es considerado el cuarto producto alimenticio a nivel mundial, después del arroz, trigo y maíz (Jesús Cuevas-Gutiérrez, 2015). Durante el 2018, México se posicionó en el duodécimo lugar a nivel mundial de exportaciones de plátano (Workman, 2019), y según información obtenida del presidente de la Unión Regional de Productores de Plátano, Francisco Hueso Alcaraz (Bravo, 2019), en ese mismo año el estado de Colima se posicionó en el tercer lugar como productor de plátano a nivel nacional, registrando una producción de más de 350 mil toneladas.

El cultivo de plátano se ve afectado por diversos factores dentro de las categorías de plagas y enfermedades, siendo totalmente susceptible no solo a los padecimientos, sino también a la pérdida total del producto. La enfermedad más común y que representa mayor amenaza dentro de este campo es la Sigatoka Negra (SN), la cual, si no se logra su prevención o se toman las medidas correspondientes desde su temprana aparición, desglosa la mayor cantidad de mermas a nivel económico, así como de cantidad y calidad en el producto.

El empleo de las redes neuronales (NN) como solución dentro de los problemas de la vida diaria, que conllevan cierta precisión y contienen variables cambiantes es una de las soluciones inteligentes más óptimas del momento. Éstas son utilizadas ampliamente para realizar procesamiento digital de imágenes en diferentes áreas con la finalidad de obtener información oportuna y con un mayor rango de precisión, teniendo como ventaja el enfoque único que se le puede otorgar y constante aprendizaje del que se puede alimentar la red en pro de las estadísticas de predicción y aciertos.

El procesamiento digital de imágenes puede ser aplicado para la identificación de plagas y enfermedades dentro del área de la agricultura como lo es en el cultivo de plátano.

### 1.1.- Naturaleza del problema

Con datos obtenidos de (INEGI, 2017) se menciona que, México, tiene una superficie sembrada de 103,160 hectáreas dedicadas al cultivo de bananos y plátanos, y una producción estimada de 2,229,519 toneladas. Los bananos, debido a las condiciones bajo las que se desarrollan, son susceptibles a la aparición de plagas y enfermedades,

las cuales, en su mayoría, comienzan su aparición y desarrollo de manera casi imperceptible. Actualmente, la enfermedad conocida como la SN es el principal problema fitosanitario que afecta la producción de este frutal en el país, así como en América Central, Sudamérica y el Caribe. Esta enfermedad es generada por un hongo llamado *Mycosphaerella fijiensis*.

En el trabajo desarrollado por (Elizabeth Álvarez, 2013) se indica que el ciclo de vida de *Mycosphaerella fijiensis* comienza con la deposición de las esporas, ya sean ascosporas o conidios, sobre las hojas libres de la enfermedad o sobre las ya afectadas. Los procesos de germinación y penetración del inóculo ocurren únicamente cuando hay condiciones favorables con humedad relativa de 90 a 100%, temperaturas de 26 a 28°C y presencia de agua libre sobre las hojas. Al estar bajo tales condiciones, las esporas germinan en un período de 2 a 6 horas, formando tubos germinativos que se extienden y ramifican en busca de las estomas. De este modo, se inicia el proceso de penetración el cual, si las condiciones de humedad relativa, temperatura y mojadura foliar son las adecuadas, tarda de 2 a 3 días.

Los efectos de la SN repercuten de manera negativa en la industria platanera debido a sus efectos altamente destructivos. Además, el manual de la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) de (Elizabeth Álvarez, 2013) nos señala que afecta las principales variedades de bananos y plátanos; por lo tanto, no se está exento de ella y la necesidad de recurrir a cuidados meticulosos para su prevención y tratamiento es indispensable. Hoy en día, el combate químico es la alternativa más usada para el tratamiento de esta enfermedad.

Debido a las facilidades que otorga el procesamiento de imágenes a través de Inteligencia Artificial (IA) contribuye a la detección de enfermedades como la SN, su detección oportuna, así como el tratamiento temprano de esta, son clave para garantizar el éxito esperado durante el proceso de producción, incluyendo beneficios para el agricultor, tales como incrementos en la economía a nivel local, estatal e inclusive nacional.

## **1.2.- Contexto del problema**

La producción de plátano es una de las principales fuentes agrícolas del estado de Colima, por lo cual garantizar su calidad en el proceso de producción es indispensable para generar productos óptimos para su comercialización. La SN es el principal problema fitosanitario que afecta dicha producción, el principal método de combate se realiza mediante químicos (Elizabeth Álvarez, 2013); sin embargo, esta opción trae consigo

problemas consecuentes, comenzando por la pérdida económica que se genera durante la aplicación innecesaria y descontrolada del producto.

La aplicación de químicos para la prevención y control de la SN se realiza, en su mayoría, mediante avionetas, igualmente es uno de los principales problemas que genera el uso de estos recursos, pues su aplicación conlleva contaminación ambiental y desequilibrio ecológico. Además, el hongo pierde sensibilidad a los fungicidas empleados. Es por esto por lo que la búsqueda de nuevas alternativas para la prevención y control de la enfermedad son prioridad, permitiendo la reducción de químicos aplicados y mermando los gastos innecesarios, generando además mejoras económicas para el productor.

### **1.3.- Revisión de la literatura**

Se ha desarrollado una búsqueda de trabajos relacionados en el área de producción de plátanos afectados por SN, entre los cuales se encuentran los que serán descritos a continuación.

En el trabajo realizado por (Andrade, 2018) se desarrolló un sistema llamado BSNet que procesa imágenes mediante el empleo de NN convolucionales para la identificación de las distintas fases de la SN en el cultivo de plátano. En la implementación del sistema, se usó una cámara digital con una resolución de 16 megapíxeles y zoom óptico de 83x que permite obtener imágenes de alta calidad. Las imágenes son procesadas utilizando técnicas de procesamiento de imágenes y NN convolucionales contenidas en la herramienta Image Processing y Neural Network de MATLAB, incluyendo sus librerías de Windows para crear una interfaz amigable para el usuario, dicha interfaz presenta las fases del agente patógeno.

Los resultados obtenidos mejoran las investigaciones previas al considerar la identificación de las fases de la plaga utilizando procesamiento de imágenes y técnicas de NN; por lo tanto, se pudo concluir que el uso de NN contribuye en la detección de la SN, logrando con ello que no se reproduzca en otros sectores del cultivo; sin embargo, el sistema se torna poco utilizable para el productor al momento de no ser establecido en un solo sitio y ser imposible de otorgar capacitación al usuario, pues el sistema se realizó con plataforma web a diferencia del planteado en esta investigación que será de escritorio sin necesidad de conexiones, para garantizar la comodidad del usuario y utilidad.

La investigación realizada en (al V. K., 2013) diseñó y desarrolló un sistema de procesamiento de imágenes a color de la planta en exteriores bajo iluminación controlada para predecir el contenido de nitrógeno del cultivo en campo. Las herramientas

empleadas constan de una cámara para capturar la imagen de la planta, cuatro luces para controlar la iluminación y una computadora portátil para procesar la señal. Los resultados fueron comparados con el contenido de clorofila del cultivo medido por medidor SPAD y el análisis químico de la hoja de la planta. El procesamiento de la imagen de la planta se realizó en el programa MATLAB. Diferentes características como R, G, B, 'r' normalizada y 'g' normalizada fueron analizadas para ambos procesos. Los modelos de regresión fueron desarrollados y evaluados entre varias características de imagen y contenido de nitrógeno de la planta; esto demostró que el contenido de nitrógeno de la planta se puede estimar con éxito por su característica de imagen en color, sin embargo, cuando el modelo fue probado se encontró con una precisión mínima del 65%, y un promedio de precisión del 75%. Si bien los resultados muestran que se puede medir el nitrógeno en las plantas por medio de imágenes, la precisión obtenida es considerablemente baja, a diferencia del trabajo citado anteriormente donde las herramientas empleadas para el procesamiento de imágenes arrojaron mejores resultados. Por otra parte, la iluminación necesaria para la toma de fotografías es artificial, por lo tanto, el trabajo se ve limitado a ciertos recursos.

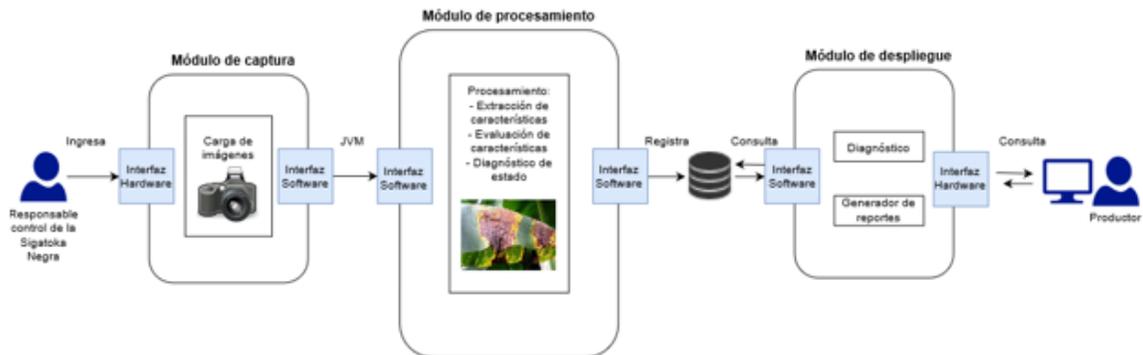
En el trabajo publicado por (al K. G., 2016) fue desarrollado un sistema para la detección de la enfermedad viral llamada Streak virus. El sistema se llevó a cabo utilizando una placa de desarrollo Embedded Linux que se conecta con una cámara, que se utiliza para capturar fotografías de la hoja de la planta de banano. La imagen capturada debe procesarse mediante un algoritmo llamado nivel de umbral económico (ETL). Este algoritmo se utiliza para establecer el valor de umbral de una hoja de banana saludable de la imagen capturada. Al detectar la hoja de banano de la imagen capturada, si el valor de píxel excede el valor de umbral, se muestra el signo de la presencia del virus de la raya del banano y se deben aplicar los métodos de recuperación correspondientes. Aunque el sistema arrojó resultados positivos y útiles dentro de su campo, este trabajo no utiliza técnicas de NN que ayuden a obtener una mejor y mayor predicción.

La investigación elaborada en (Gutiérrez, 2015) presentó un sistema de adquisición y análisis de información, el cual ayuda en la realización de muestreos de SN, de tal manera que se buscó otorgar una herramienta que sirviera de manera informativa para el usuario, siendo de fácil manejo para cualquiera, mostrando los diversos comportamientos de la enfermedad para la toma de decisiones posterior. Los resultados indican que el software desarrollado es confiable y eficiente para conocer el estado fitosanitario de la huerta analizada, sin embargo, si bien el trabajo pretende educar para la detección y manejo de la SN, la responsabilidad queda en manos de la persona a la cual se le haya mostrado, de tal manera que si su aprendizaje fue incorrecto sus decisiones se verán afectadas. El uso de las NN hubiera hecho de este proceso algo más sencillo y, sobre todo, preciso.

En (al P. J., 2010) se presenta un método el cual sirve para la identificación de cuatro estados de maduración en los frutos de café, esto implementando en un circuito electrónico. El método se basa en la representación del color Hue-SaturationValue (HSV), las diferencias de los estados de maduración en el plano de las componentes “hue”(Matiz) y “saturation”(Saturación) (hs), y conlleva un algoritmo de aprendizaje que entrega puntos de corte para limitar las áreas de los cuatro estados de maduración sobre el plano hs. El algoritmo usa un proceso llamado *Optimización de la Eficacia Global* (OEG), el cual sirve para la identificación de las áreas que contienen el máximo número de lecturas de un estado de maduración. El sistema tiene dos funciones, una de calibración que usa el algoritmo para determinar el área correspondiente a cada estado de maduración sobre el plano hs, y la función de clasificación, que cuenta con la capacidad de identificar y separar hasta 50 frutos/s en tiempo real. La eficacia de identificación varía entre 94,8% y 99,6%, lo cual hace que el dispositivo sea confiable para el control de calidad de la materia prima a procesar en el beneficio del café; sin embargo, no utiliza NN, lo cual haría de este sistema algo más estable y efectivo, sin dependencias del funcionamiento electrónico como tal.

#### **1.4.- Planteamiento de la solución del problema a investigar**

En este trabajo se propone la optimización e implementación de una herramienta para la detección de la SN en los cultivos de plátano mediante el procesamiento digitalizado de imágenes, utilizando NN. La toma de las fotografías se realizará con la cámara digital Nikon Coolpix P900, que cuenta con la tecnología necesaria para la toma de fotografías claras, así como enfoque de detalles, factores necesarios para su posterior análisis. Las imágenes de muestreo obtenidas serán almacenadas en una tarjeta SD para posteriormente ser inspeccionadas. Para llevar a cabo el análisis se realizará una aplicación de escritorio, de tal manera que el encargado de la toma de muestreo dentro del cultivo pueda alimentar la red y darle el uso apropiado sin necesidad de conexión a Internet. En la figura 1 se muestra el modelo conceptual del proyecto.



**Figura 1:** Modelo conceptual

## 1.5.- Justificación

La implementación de un sistema para la detección oportuna de la SN dentro de los cultivos de plátano en apoyo a la producción del fruto en cuestión. Otorgando una herramienta factible al usuario, que fomente la productividad y evite los retrasos provocados por la enfermedad en cuestiones de calidad, así como ahorro de productos para su control.

## 1.6.- Motivación

El desarrollo de este trabajo de investigación tiene como motivación otorgar una herramienta de uso cómodo para el usuario en apoyo a los agricultores de plátano, que sirva como un mecanismo de detección temprana de la SN dentro del cultivo, esto con la finalidad de optimizar los procesos, incrementar la producción, las ganancias y mermar las posibles pérdidas durante y después de éste, de tal manera que sea factible para el productor tanto en cuestiones de uso como mejoras en entregas de productos de calidad ideales para su comercialización.

## 1.7.- Objetivos

A continuación, se describe el objetivo general de este trabajo de investigación, así como los objetivos específicos.

### 1.7.1.- Objetivo general

Identificar las fases de la SN en el cultivo de plátano mediante el procesamiento de imágenes digitales utilizando NN.

### 1.7.2.- Objetivos específicos

- Analizar las herramientas para procesamiento digital de imágenes.
- Analizar los distintos tipos de NN.
- Seleccionar la red neuronal pertinente para el desarrollo del proyecto.
- Desarrollar un sistema basado en redes neuronales para el análisis y detección de las fases de la SN.
  - Realizar las pruebas del sistema desarrollado.
  - Analizar los resultados de la identificación de las fases de la SN.
  - Implantar el sistema en las instalaciones del productor.

## 1.8.- Metas

- Identificar las fases de la SN a tiempo para evitar su propagación y contribuir a la toma de decisiones para su disminución en la producción de plátano.
- En Julio del año 2020 quedará implantado el sistema para su uso interno y operación.

## 1.9.- Hipótesis

El desarrollo de un sistema para la identificación de la SN en el cultivo de plátano a través del procesamiento de imágenes digitales incrementa la producción y la calidad del producto.

## 1.10.- Métodos y herramientas

En la búsqueda del cumplimiento efectivo de los requisitos y propósitos de este proyecto, se seleccionó a los usuarios finales como participantes de la investigación, es decir el propietario del predio y el trabajador encargado de la inspección del plantío en la búsqueda de la SN. La selección se realizó con base a la cercanía que tendrían con el sistema, así como la información y experiencia con la que cuentan en torno al problema de investigación, factores útiles para lograr el correcto desarrollo del sistema.

Para la recolección de datos, se cuenta con constantes visitas al predio, con la finalidad de levantar requisitos y proceder a la obtención de imágenes de la enfermedad, las cuales son base como punto de partida y desarrollo del sistema, para ello se utilizó una cámara Nikon Coolpix P900.

Con base a los requisitos necesarios para el desarrollo del software, se usaron las técnicas de procesamiento de imágenes y NN entrenando la New Feed-Forward (NFF) con las imágenes de las fases de la SN, mediante la herramienta MATLAB. Para poder hacer uso de la NFF primero se ingresaron las imágenes previamente seleccionadas en base a su enfoque y claridad, para posteriormente pasar a la aplicación de filtros como son: bordes, color y manchas. Finalmente se realizan cálculos para obtener la predicción de la imagen que se está analizando.

### **1.11.- Organización de la tesis**

Este documento de tesis se encuentra organizado de la siguiente forma:

- Capítulo 1. Se describe la introducción del tema, donde se expone la naturaleza y el contexto del problema, se plantean los motivos, se propone una solución y se proponen los objetivos en los que se basa la realización de este proyecto, así como la hipótesis.
- Capítulo 2. Se realizó la investigación correspondiente al tema de investigación como es el estado del campo de conocimiento; se describe en una forma amplificada el contexto del problema y se lleva a cabo la integración de la teoría con la investigación.
- Capítulo 3. Se presenta la metodología utilizada para el desarrollo del sistema.
- Capítulo 4. Este capítulo describe las fases de análisis, diseño, implementación y validación, profundizando en cada uno de estos y describiendo cada paso que les conforma.
- Capítulo 5. En este último capítulo se describen los resultados obtenidos mediante el empleo del sistema.

En los capítulos posteriores se plasma la conclusión obtenida mediante comparativas de los resultados obtenidos con la hipótesis y objetivos establecidos al inicio de la investigación, así como se citan las referencias empleadas a lo largo del documento.

## II. ESTADO DEL CAMPO DEL CONOCIMIENTO

El Estado del campo del conocimiento contiene los estudios e investigaciones antecedentes, una amplia descripción del contexto del problema y la integración de la teoría con la investigación y sus relaciones mutuas.

Incluye:

- El Marco Histórico
- El Marco Contextual
- El Marco Teórico

### 2.1 El Marco Histórico

A continuación, se describe el surgimiento y evolución del problema de investigación, mencionando las fases y destacando sus hallazgos suscitados.

#### 2.1.1 Agentes patógenos del cultivo de plátano

El cultivo de plátano se ve severamente afectado por diversos agentes patógenos, siendo las enfermedades consecuentes una gran limitante en la producción. De acuerdo con (Gilberto Manzo Sanchez, 2014) estas enfermedades son principalmente causadas por hongos, bacterias y virus, de igual manera menciona que las enfermedades comunes y de mayor impacto en el cultivo de plátano son: Mal de Panamá, Moko bacteriano, marchitez bacteriana, virus del bunchy top, virus del estríado del plátano, virus del mosaico del plátano y mancha foliar eumusae, en esta última categoría se encuentran al menos 20 especies de *Mycosphaerella*, siendo *Mycosphaerella fijiensis*, agente causal de la SN, la más importante a nivel mundial por ser la mayor causante de pérdidas significativas.

En (FAO línea agrícola, 2012) se menciona que, las enfermedades más limitantes en el cultivo de plátano son las que se enlistan y describen a continuación:

Moko o madurabiche: Esta enfermedad es causada por la bacteria *Ralstonia solanacearum*. Su presencia puede generar la pérdida total de las plantas afectadas, sin embargo, los mayores costos de manejo se relacionan con la aplicación de medidas de erradicación de focos y tiempo cesante durante el cual no se pueden sembrar las áreas afectadas con banano. Todos los órganos de la planta, desde las raíces hasta el escapo floral, pueden ser infectados y presentan síntomas internos y externos (FAO línea agrícola, 2012).

Pudrición acuosa del pseudotallo o bacteriosis (*Dickeya chrysanthemi*): El desarrollo comienza en los pseudotallos externos y a partir del punto en donde ocurre la infección se va extendiendo en todos los sentidos hasta afectar totalmente el sitio. Durante los inicios de su desarrollo, en las plantas afectadas se observa una quemazón en el borde de las hojas más viejas que luego avanza a toda la lámina foliar, ocasionando un amarillamiento total de la hoja (FAO línea agrícola, 2012).

Mal de Panamá: Este hongo produce potentes toxinas que ocasionan marchitez vascular, pudrición en semillas, raíces, tallos, cormos y tubérculos. Es un habitante natural del suelo que sobrevive en restos de plantas infectadas en forma de micelio y esporas. Los síntomas externos producidos por *Fusarium oxysporum* f. sp. cubense se caracterizan por un amarillamiento de las hojas más adultas a lo largo del margen foliar que continúa hacia la nervadura central hasta quedar las hojas completamente marchitas y de color café (FAO línea agrícola, 2012).

Sigatoka Negra (*Mycosphaerella fijiensis* Morelet var. *difformis*): La Sigatoka Negra es la enfermedad foliar más destructiva. De manera directa afecta sólo las hojas del banano, de manera más rápida y severa que la Sigatoka amarilla (FAO línea agrícola, 2012).

Los síntomas iniciales de la enfermedad aparecen como pequeñas estrías de color café ligero en las hojas. Cuando la infección aún se encuentra dentro de una categoría baja, las estrías desarrollan lesiones ovales o elipsoidales con el centro gris. Sin embargo, cuando la densidad de infección es alta, las lesiones coalescen y se necrosan, formando grandes áreas de tejido foliar (Gilberto Manzo Sanchez, 2014). El grado de infección de la Sigatoka Negra en base a los síntomas, se clasifica en seis estados que serán descritos en el apartado 2.1.2

El reporte (Elizabeth Álvarez, 2013) menciona que, el desarrollo de la enfermedad se encuentra directamente influenciado por las condiciones climáticas, susceptibilidad de la variedad sembrada y manejo del cultivo. Las zonas más afectadas por la Sigatoka Negra se caracterizan por tener una precipitación mayor a 1.400 mm anuales, humedad relativa mayor al 80% y temperatura promedio entre 23 a 28 °C. La enfermedad es más agresiva en épocas lluviosas, debido a la presencia continua de una lámina de agua sobre las hojas, que favorece los procesos de liberación e infección de las esporas (Elizabeth Álvarez, 2013).

### 2.1.2 Sigatoka Negra (SN)

En (CropLife Latin America, 2018) se menciona que, la SN es la enfermedad del banano más importante a nivel mundial. Su nombre viene del Valle de Sigatoka en las Islas Fiji donde en 1912 fue identificada por primera vez. Durante los siguientes 40 años, la

enfermedad se difundió a todos los países productores de banano. La SN apareció en América Central en 1934, posteriormente fue detectada en México en 1980 y finalmente llegó a Colima en el año 1989, causando severos daños en una superficie de 11,000 ha sembradas de plátano, siendo reducidas a 3,800 ha.

De acuerdo con los investigadores de la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO), la SN es causada por el hongo *Mycosphaerella fijiensis*, y representa la principal enfermedad foliar limitante en la producción de plátano a nivel mundial. La enfermedad afecta el área foliar fotosintética de la planta, lo cual causa que los racimos y los frutos tengan un menor peso en comparación con plantas sanas. Además, las infecciones severas de la SN causan la madurez prematura del fruto, generando grandes pérdidas en el cultivo (Elizabeth Álvarez, 2013).

En el reporte informativo (Elizabeth Alvarez, 2013) también se menciona que la enfermedad evoluciona en la planta a través de los siguientes seis estados, basados en la escala de Fouré:

**Estado 1.** Pequeñas lesiones o puntos de color blanco-amarillento a marrón, de 1 mm de longitud, denominadas pizcas, apenas visibles en el envés de las hojas (Elizabeth Álvarez, 2013). Véase figura 2.



**Figura 2:** Foto estado 1

**Estado 2.** Rayas o estrías cloróticas de 3–4 mm de longitud por 1 mm de ancho, de color marrón (Elizabeth Álvarez, 2013). Véase figura 3.



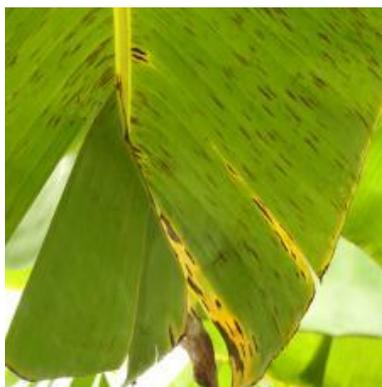
**Figura 3:** Foto estado 2

**Estado 3.** Las rayas o estrías se alargan y amplían dando la impresión de haber sido pintadas con pincel, sin bordes definidos y de color café, que pueden alcanzar hasta 2 cm de longitud (Elizabeth Álvarez, 2013). Véase figura 4.



**Figura 4:** Foto estado 3

**Estado 4.** Manchas ovaladas de color café en el envés y negro en el haz (Elizabeth Álvarez, 2013). Véase figura 5.



*Figura 5: Foto estado 4*

**Estado 5.** Manchas negras rodeadas de un anillo negro y a veces un halo amarillento y centro seco y semihundido (Elizabeth Álvarez, 2013). Véase figura 6.



*Figura 6: Foto estado 5*

**Estado 6.** Manchas con centro seco y hundido, de coloración marrón clara, rodeadas de tejido clorótico (Elizabeth Álvarez, 2013). Véase figura 7.



*Figura 7: Foto estado 6*

Para la diseminación de la enfermedad influyen diferentes factores (FAO línea agrícola, 2012):

- Viento: es el factor que permite la dispersión de las esporas de la sigatoka negra, éstas son dispersadas y depositadas en las hojas más jóvenes de la planta; si las condiciones de humedad son buenas, el hongo penetra en el tejido foliar y se produce el primer

síntoma característico de pizca y posteriormente la mancha necrótica (FAO línea agrícola, 2012).

- Lluvia: juega un papel importante en la liberación del inóculo, la precipitación provee condiciones de humedad que favorecen en desarrollo de las infecciones, permitiendo establecer una época con relativa baja incidencia y otra de alta incidencia (FAO línea agrícola, 2012).
- Movilización incontrolada de hojas enfermas por los humanos (FAO línea agrícola, 2012).

Si bien la SN no tiene remedio, esta puede ser controlada mediante un manejo integrado de la enfermedad (MIE). Los principales métodos para el manejo de la enfermedad son: Control cultural. Está orientado a reducir las fuentes de inóculo del patógeno y a mejorar las condiciones de la planta de plátano o banano para minimizar el impacto de la Sigatoka Negra. Control químico: La aplicación de fungicidas químicos puede disminuir el daño ocasionado por la SN, pero su uso debe ser justificado y supervisado, de tal manera que se eviten sobrecostos, daños a la salud y al medio ambiente (Elizabeth Álvarez, 2013).

### 2.1.3 Procesamiento de imágenes

Las primeras aplicaciones de la imagen digital surgieron en la industria periodística, cuando se envió por primera vez la imagen mediante un cable submarino entre Londres y Nueva York. Durante la década de 1920, se logró la reducción del tiempo requerido para transferir una imagen a través del Atlántico de tal manera que, de demorar más de una semana se redujo el tiempo a tres horas, esto mediante la introducción del sistema de transmisión de imágenes por cable.

Las técnicas de procesamiento digital de imágenes comenzaron a tomar impulso en el campo científico mediante su uso en el Jet Propulsion Laboratory, en 1959, con la finalidad de mejorar las imágenes enviadas por los cohetes. Los resultados obtenidos en un tiempo relativamente corto fueron tan impresionantes que muy pronto se extendieron las aplicaciones del método a otros campos (Malacara, 1997), algunos de ellos comenzaron en los últimos años sesenta y principios de los setenta para usarse en imágenes médicas, observaciones remotas de recursos de la Tierra y astronomía (Bhauasaheb Shivajirao Shinde, 2013).

#### 2.1.4 Redes neuronales (NN)

El trabajo de investigación publicado por (Haykin, Neural Networks and Learning Machines, 2009), describe a una NN como una máquina que está diseñada para modelar la forma en que el cerebro realiza una tarea particular de función de interés; La red generalmente se implementa mediante el uso de componentes electrónicos o se simula en software en una computadora digital. Para lograr un buen rendimiento, las redes neuronales emplean una interconexión masiva de células de computación simples a las que se hace referencia como "neuronas" o "unidades de procesamiento" (Haykin, Neural Networks and Learning Machines, 2009).

El modelo neuronal de McCulloch y Pitts de 1943 (D. Michie, 1994), Linear Threshold Unit o Threshold Logic Unit (TLU), fue el primer modelo moderno neuronal, y ha servido de inspiración para el desarrollo de otros modelos de la misma rama. El modelo McCulloch-Pitts pese a la distorsión y bajo nivel de comprensión que género en sus inicios, actualmente está retomando importancia debido a que es uno de los pocos modelos digitales en tiempo discreto.

En los años de formación de las NN (1943-1958) varios investigadores se destacan por sus contribuciones pioneras (Haykin, Neural Networks and Learning Machines, 2009):

- McCulloch y Pitts (1943) por introducir la idea de NN como máquinas de computación.
- Hebb (1949) por postular la primera regla para el aprendizaje auto organizado.
- Rosenblatt (1958) por su propuesta "Perceptron" como primer modelo para el aprendizaje supervisado.

## 2.2 El Marco Contextual

El sistema desarrollado será implementado en un predio de cultivo de plátano, ubicada casi en los márgenes de la laguna de Cuyutlán, ejido el Colomo municipio de Manzanillo, Colima. La figura 8 muestra la localización de la granja.



**Figura 8:** Localización predio cultivo de plátano

Actualmente, el proceso de detección y control de la SN se realiza de manera personal, es decir, se cuenta con un encargado el cual una o dos veces por semana, dependiendo del clima, realiza inspecciones alrededor de las tres tablas o secciones que conforman el predio. El proceso de inspección consta en seleccionar determinadas plantas de plátano por sección, por cada una se troza un pedazo de la hoja más joven y se observa detenidamente en busca de pizca o estría. En caso de encontrar rastros de infección, se procede a inspeccionar más hojas de la misma planta.

Cada inspección conlleva un registro manual en un formato para la evaluación de SN usando el sistema stover de grado, el cual, al ser finalizado, arroja mediante fórmulas el grado de infección que padece el predio, determinando la gravedad de este.

En (Juan Salvador García Andrade, 2018) se realizó un sistema de procesamiento de imágenes aplicada a la identificación de la Sigatoka Negra en el cultivo de plátano mediante el empleo de NN, en este caso, una red neuronal convolucional AlexNet para llevar a cabo la clasificación de los datos en cuatro grados, así como la categoría “sanas” para aquellas hojas no infectadas. Para la realización del sistema se entrenó la red con

100 datos, los cuales, si bien arrojaron resultados prácticos, debido a que su funcionalidad depende de la conexión a Internet, el desarrollo del proyecto pierde usabilidad en ocasiones puesto a que este servicio no está disponible la mayor parte del tiempo.

## **2.3 El Marco Teórico**

### **2.3.1 Procesamiento de imágenes**

El procesamiento de imágenes es la aplicación de un conjunto de técnicas y algoritmos a una imagen digital para analizar, mejorar u optimizar características de la imagen, como la nitidez y el contraste (Paul S. Ganney, 2014).

El procesamiento de imágenes es un método para realizar algunas operaciones en una imagen, con el fin de obtener una imagen mejorada o extraer información útil de ella. Es un tipo de procesamiento de señal en el que la entrada es una imagen y la salida puede ser una imagen o características / características asociadas con esa imagen. Hoy en día, el procesamiento de imágenes se encuentra entre las tecnologías de rápido crecimiento. También forma un área central de investigación dentro de las disciplinas de ingeniería e informática. (Digital Image Processing, 2014).

### **2.3.2 Inteligencia Artificial**

La IA es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que, si fuera llevado a cabo por una persona, se diría que es inteligente. El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente (Juan Jesús Romero, 2007).

En la figura 9 se muestran las definiciones que manejan en (Stuart J. Rusell, 2004) para IA:

<b>Sistemas que piensan como humanos</b>	<b>Sistemas que piensan racionalmente</b>
<p>«El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985)</p> <p>«[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...» (Bellman, 1978)</p>	<p>«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985)</p> <p>«El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)</p>
<b>Sistemas que actúan como humanos</b>	<b>Sistemas que actúan racionalmente</b>
<p>«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990)</p> <p>«El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)</p>	<p>«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole <i>et al.</i>, 1998)</p> <p>«IA... está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)</p>

**Figura 9:** Cuatro definiciones IA (Stuart J. Russell, 2004)

### 2.3.3 Red neuronal New Feed-Forward

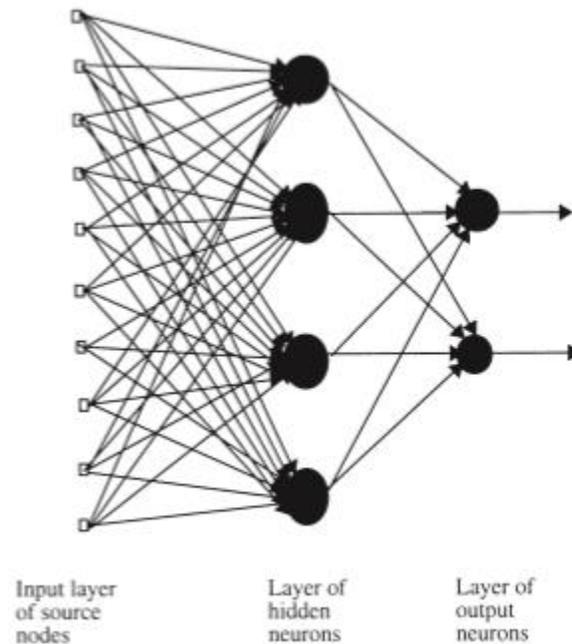
Una NN es un procesador distribuido masivamente paralelo compuesto de unidades de procesamiento simples que tiene una propensión natural para almacenar el conocimiento experiencial y ponerlo a disposición para su uso. Se asemeja al cerebro en dos aspectos (Haykin, *Neural Networks and Learning Machines*, 2009):

1. El conocimiento es adquirido por la red desde su entorno a través de un proceso de aprendizaje (Haykin, *Neural Networks and Learning Machines*, 2009).
2. Las fuerzas de conexión interneuronales, conocidas como pesos sinápticos, se usan para almacenar los conocimientos adquiridos (Haykin, *Neural Networks and Learning Machines*, 2009).

Las NN son sistemas ideados como abstracciones de las estructuras neurobiológicas (cerebros) encontradas en la naturaleza y tienen la característica de ser sistemas desordenados capaces de guardar información (Casarin, 2014).

Las NN feed-forward, o perceptrones multicapa (MLP), son los modelos de aprendizaje profundo por excelencia. El objetivo de una red feed-forward es aproximar alguna función  $f^*$ . Por ejemplo, para un clasificador,  $y = f^*(x)$  asigna una entrada  $x$  a una categoría  $y$ . Una red de avance define una asignación  $y = f(x; \theta)$  y aprende el valor de los parámetros  $\theta$  que resultan en la mejor aproximación de la función (Ian Goodfellow, 2016).

Una NN new feed-forward tiene una capa de entrada de nodos fuente y una capa de salida de neuronas (es decir, nodos de cálculo); Estas dos capas conectan la red al mundo exterior. Además de estas dos capas, generalmente tiene una o más capas de neuronas ocultas, que se llaman así porque estas neuronas no son directamente accesibles. Las neuronas ocultas extraen características importantes contenidas en los datos de entrada, como se muestra en la figura 10 (Haykin, 1 Feedforward neural networks: an introduction, 2004).



**Figura 10:** Arquitectura new feed-forward (Haykin, 1 Feedforward neural networks: an introduction, 2004)

#### 2.3.4 Aplicación de escritorio

Una aplicación Desktop o escritorio es aquella que está instalada en el ordenador del usuario, que es ejecutada directamente por un sistema operativo, y cuyo rendimiento depende de diversas configuraciones de hardware (Buyto, 2011).

Las aplicaciones de escritorio son desarrolladas e instaladas en un sistema operativo particular y son diseñados especialmente para cumplir una función o actuar como herramienta para acciones puntuales del usuario (Gutierrez, 2020).

### 2.3.5 MATLAB

MATLAB es una plataforma de programación diseñada específicamente para ingenieros y científicos. El corazón de MATLAB es el lenguaje MATLAB, un lenguaje basado en matrices que permite la expresión más natural de las matemáticas computacionales. Combina un entorno de escritorio ajustado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matriz y matriz directamente (MATLAB, 2020).

MATLAB es un poderoso lenguaje de programación que incluye los conceptos comunes a la mayoría de los lenguajes de programación. Se trata de un lenguaje con base en scripts (Víctor Campos Olgún, 2007).

### 2.3.6 MySQL

Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB21. MySQL2 incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos (Guillen, 2014).

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web (ORACLE, 2020).

### **III. MÉTODOS EMPLEADOS**

En este capítulo se describen los métodos utilizados durante el desarrollo de este trabajo de investigación.

#### **3.1 Investigación preliminar**

Para la correcta realización del proyecto, se llevó a cabo una investigación respecto a la problemática planteada, se visitó el predio de cultivo de plátano en el que se trabajó, esto con la finalidad de conocer sus características, y propuestas de solución previas mediante la revisión de la literatura. De igual manera se indagó sobre las NN y sus diversos empleos, esto con la finalidad de llevar a cabo una solución óptima.

#### **3.2 Análisis de requerimientos**

En el análisis de requerimientos, se empleó la estrategia de entrevista con las partes involucradas, con la finalidad de recaudar la información necesaria para definir los requerimientos del proyecto.

Una vez obtenidos los requerimientos, se procedió a su análisis y definición, con la finalidad de conocer las características de entorno que involucran el desarrollo del proyecto, tales como: etapas de la SN, recursos disponibles para la solución del problema, costos de producción, etc.

#### **3.3 Planeación**

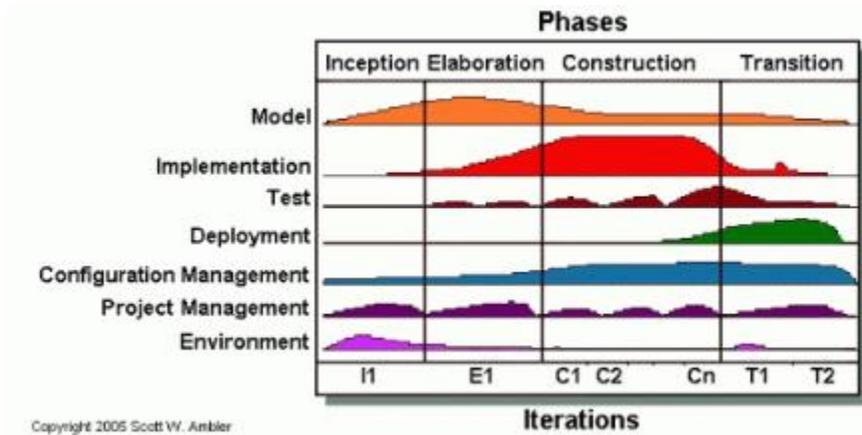
En la etapa de planeación se plantearon las tareas y actividades por realizar, generando una organización de los recursos necesarios, una calendarización de las actividades y los productos que se obtendrían del trabajo. Se estimó el tiempo, costo y esfuerzo para el desarrollo del proyecto.

#### **3.4 Descripción de metodología**

La metodología empleada es la PUA, la cual, Pressman (Pressman, 2010) nos describe como una metodología que brinda un revestimiento en serie, lo cual permite al equipo visualizar el flujo general del proceso del proyecto de software en cuestión; El equipo repite cada actividad con la finalidad de agilizar y entregar tan rápido como sea posible

incrementos de software a los usuarios finales. Cada iteración de la metodología PUA aborda las siguientes actividades (Pressman, 2010):

- Modelado. Se crean representaciones UML de los dominios que conciernen al negocio y el problema. Para conservar la agilidad estos modelos deben ser solo “suficientemente buenos”, y entonces permitir que el equipo avance (Pressman, 2010).
- Implementación. Los modelos del paso anterior se traducen a código fuente (Pressman, 2010).
- Pruebas. El equipo diseña y ejecuta un conjunto de pruebas con la finalidad de detectar errores y asegurar que el código fuente cumple correctamente con los requerimientos (Pressman, 2010).
- Despliegue. El despliegue se centra en la entrega incremental del software y en la obtención de retroalimentación por parte de los usuarios finales (Pressman, 2010).
- Configuración y administración del proyecto. La administración de la configuración incluye la administración del cambio y el riesgo, así como el control de cualesquiera productos del trabajo persistentes que produzca el equipo (Pressman, 2010).
- Administración del ambiente. La administración del ambiente se encarga de coordinar una infraestructura del proceso que incluye estándares, herramientas y otra tecnología de apoyo a disposición del equipo (Pressman, 2010). En la figura 11 se muestra la estructura de la metodología PUA.



**Figura 11:** Metodología PUA (Ambles, 2005)

### 3.5 Hardware y Software

A continuación, serán descritas las herramientas de hardware y software que empleadas para la elaboración de este proyecto.

#### 3.5.1 Hardware

El equipo de software necesario para el desarrollo de este proyecto consta de una computadora y una cámara digital. Enseguida, se describen las características correspondientes de cada uno:

- Computadora Lap top:
  - Marca Hewlett-Packard (HP)
  - Intel Core i7 8th Gen
  - Memoria 4 GB + Intel OPTANE 16 GB
  - 1 TB Disco duro
  
- Cámara digital:
  - Marca Nikon
  - Coolpix P900
  - 16 Megapixeles CMOS
  - Zoom óptico de 83x
  - GPS, Wi-fi y NFC
  - Lente focal de f/2.8-6.5

### 3.5.2 Software

Los programas utilizados en el desarrollo de la aplicación son los que se enlistan a continuación:

- Sistema Operativo Windows 10
- MATLAB R2015a
- XAMPP

### 3.6 Diseño y modelado del prototipo

El proceso para el modelado consistió en hacer un primer análisis y diseño con el fin de entender el funcionamiento en general del sistema y NN a utilizar. Se definió el número de datos de entrada que tendría la NN, con la finalidad de encontrar un estándar mínimo o punto de partida y corroborar su correcto funcionamiento para el desarrollo del proyecto. Posteriormente se fue refinando el análisis y diseño con mayor detalle para optimizar el prototipo mismo hasta llegar al diseño final.

### 3.7 Construcción

En la etapa de construcción se escribió el código necesario para realizar el entrenamiento de la red y comenzar el procesamiento de imágenes para la detección de la Sigatoka Negra. En la primera parte del código se obtienen las imágenes que actuarán como datos de entrada. En la segunda parte, estos datos se leen, se obtiene la información de las capas de la imagen y se almacena en un vector. Posteriormente se crea la red neuronal mediante la función `newff` perteneciente al Neural Network Toolbox de MATLAB, ingresando los parámetros correspondientes. Al finalizar esta iteración, se obtiene un promedio con base al procesamiento de la imagen mediante la red. En el código se programa la condicional para asignar los parámetros con base a los promedios obtenidos, que corresponden a cada clasificación de la hoja. Además, se escribió código para recibir los datos monitoreados y almacenarlos en una base de datos, desde la cual se podrán realizar consultas e imprimir reportes; así como también una interfaz gráfica para manipular y mostrar los datos de entrada.

### 3.8 Implementación del prototipo

En la implementación del prototipo se instaló y configuró el sistema en el equipo a utilizar por el productor, posteriormente se brindó capacitación para el uso del sistema y

finalmente se evaluó el correcto funcionamiento del sistema a través de la retroalimentación del usuario.

### **3.9 Documentación**

En esta fase se redactó la documentación del proyecto.

## IV. DESARROLLO DE LA TESIS

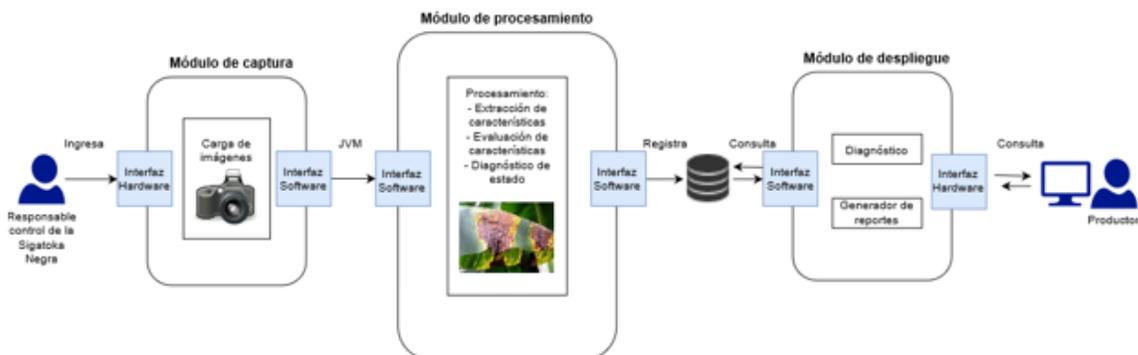
Este apartado describe las fases seguidas en el proceso para instrumentar la investigación.

### 4.1 Análisis

En esta parte del desarrollo se describen las etapas involucradas con el análisis: modelo conceptual del sistema, diagramas de actividades con responsabilidades, modelo de casos de uso y el modelo de requisitos.

#### 4.1.1 Modelo conceptual (Big Picture)

La figura 12 se muestra el modelo conceptual para la solución planteada a la problemática presentada en el capítulo 1, el cual está compuesto por los siguientes elementos:



**Figura 12:** Big Picture-Modelo conceptual

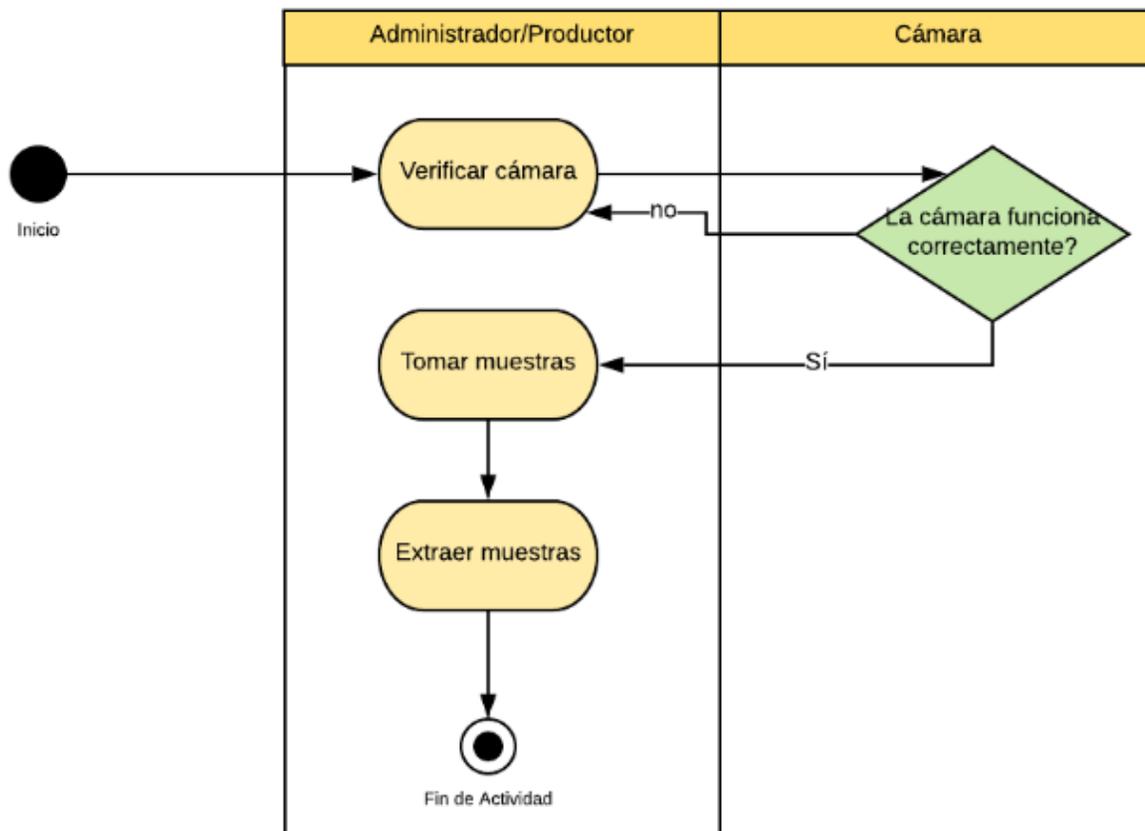
**Módulo de captura.** En este módulo, el usuario responsable del control de la Sigatoka Negra carga las imágenes de las hojas que previamente fueron tomadas con la cámara durante la inspección del predio.

**Módulo de procesamiento.** En este segundo módulo, se realizan los procesos de extracción de características de las imágenes ingresadas, evaluación de estas características y, finalmente, arroja un diagnóstico de estado de las hojas, posteriormente automáticamente se realiza el registro en la base de datos.

**Módulo de despliegue.** Este tercer módulo muestra al usuario final, ya sea el encargado de la inspección o el productor, el diagnóstico de la imagen ingresada, así como permite al finalizar la inspección generar un reporte, el cual también se puede consultar en otra interfaz dedicada a la consulta de reportes creados en fechas específicas.

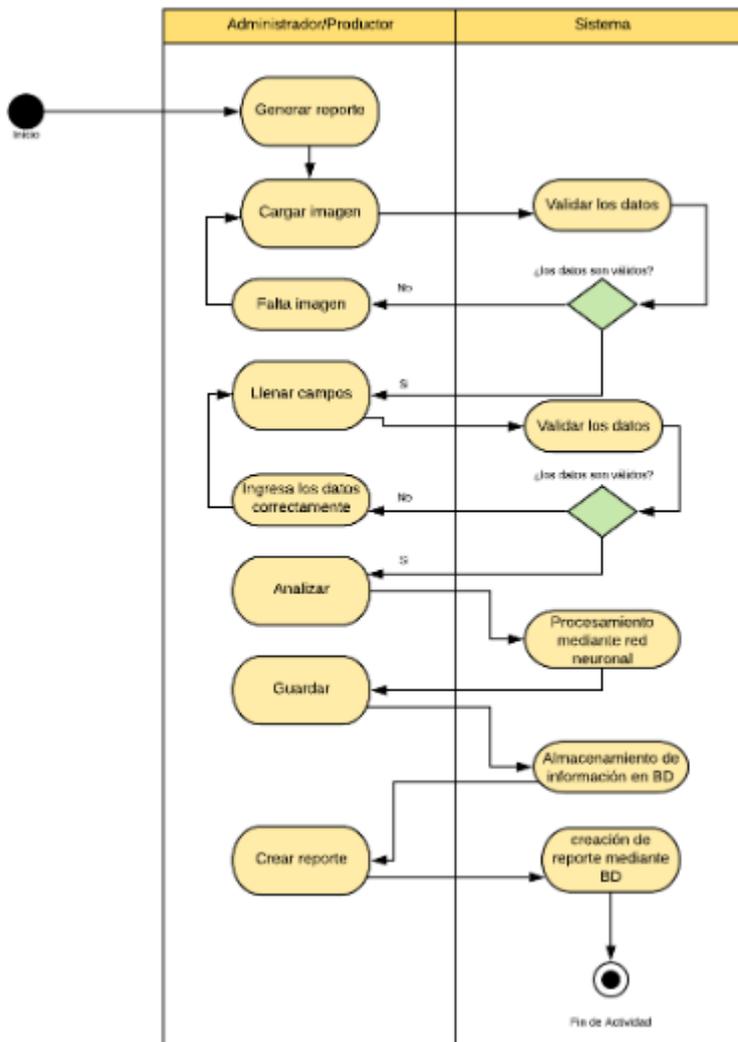
#### 4.1.2 Diagramas de actividades con responsabilidades

El sistema está integrado los diagramas de actividades presentados en la figura 13 y figura 14; En la figura 13, se ilustra el proceso para la obtención y captura de las muestras que serán ingresadas al sistema.



**Figura 13:** Diagrama de Actividades toma de muestras

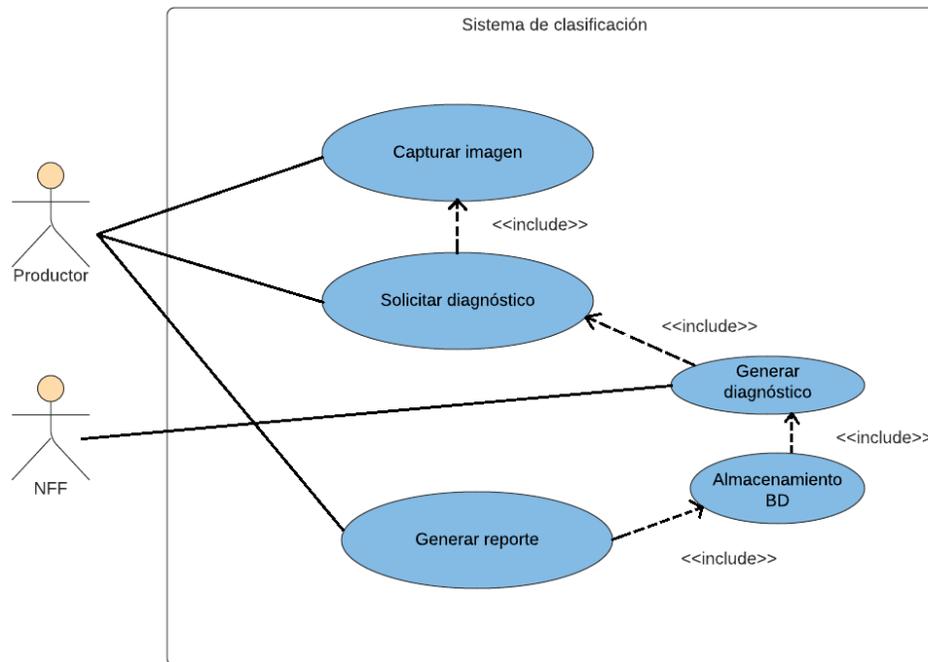
En la figura 14 se muestra el diagrama de actividad del usuario encargado del manejo del sistema.



**Figura 14:** Diagrama actividades proceso

#### 4.1.3 Modelo de casos de uso

A continuación, en la figura 15 se presenta el modelo de casos de uso del sistema.



**Figura 15:** Diagrama caso de uso

#### 4.1.4 Modelo de requisitos

Los requisitos funcionales son descritos en la tabla 1, mediante los cuales se definió el alcance del sistema.

**Tabla 1:** Requisitos funcionales

No. Requisito	Descripción
RF01	El sistema debe cargar la imagen obtenida del ordenador
RF02	El sistema debe mostrar la imagen cargada al usuario
RF03	El sistema debe poder registrar número de planta
RF04	El sistema debe poder registrar número de hoja
RF05	El sistema debe procesar la imagen de hoja ingresada mediante la red neuronal newff
RF06	El sistema debe determinar mediante la red neuronal newff el grado de infección de la hoja
RF07	El sistema debe arrojar un diagnóstico de la imagen ingresada
RF08	El sistema debe sumar el total de hojas registradas por planta
RF09	El sistema debe clasificar el total de hojas por grado de infección
RF10	El sistema debe registrar el número de la hoja más joven registrada
RF11	El sistema debe sumar el número de hoja correspondiente a cada hoja más joven registrada
RF12	El sistema debe promediar el total de hojas jóvenes registradas
RF13	El sistema debe registrar el número de la hoja más joven con estría

RF14	El sistema debe sumar el número de hoja correspondiente a cada hoja más joven con estría registrada
RF15	El sistema debe promediar el total de hojas jóvenes con estría registradas
RF16	El sistema debe sumar el total de hojas registradas por grado de infección
RF17	El sistema debe promediar el total de hojas registradas por grado de infección
RF18	El sistema debe calcular el PPI del cultivo de plátano
RF19	El sistema debe registrar la fecha en que se generó el reporte
RF20	El sistema debe generar un archivo a manera de reporte
RF21	El sistema debe clasificar solo los primeros 4 niveles de Sigatoka Negra

Los requisitos no funcionales son descritos a continuación en la Tabla 2:

**Tabla 2:** Requisitos no funcionales

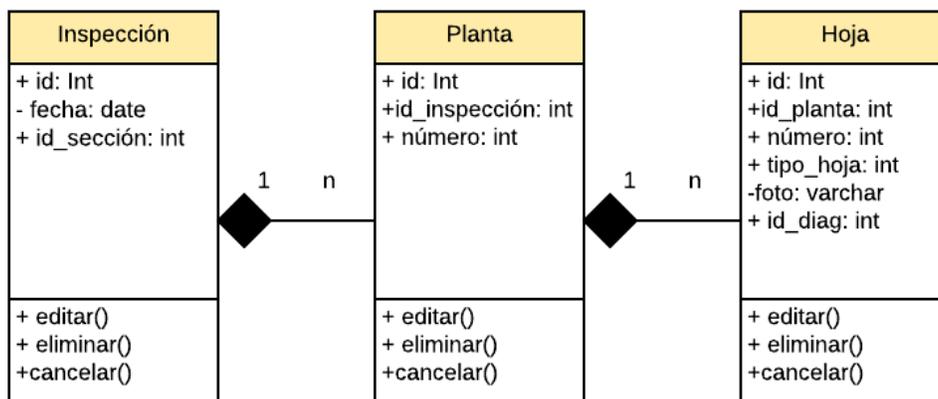
No. Requisito	Descripción
RNF01	El sistema debe ser compatible para las versiones de Windows a partir de la octava
RNF02	El sistema será desarrollado para plataforma PC
RNF03	Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 10 segundos
RNF04	El sistema debe ser de fácil manejo y comprensión
RNF05	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas

## 4.2 Diseño

En este apartado se presentan los diferentes procesos a modo de diagramas, que se utilizaron para el almacenamiento de la información en la base de datos, así como también la programación utilizada para el desarrollo de los módulos que integran el sistema.

### 4.2.1 Modelo de clases

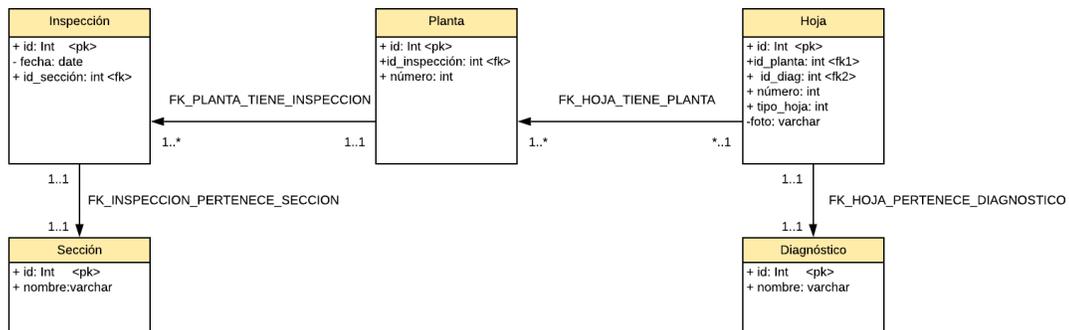
La base de datos que se fue estructuralmente sencilla, ya que se requieren pocos datos para la predicción en la foto y realización del reporte, los datos que se envían son sección a la que pertenece la muestra, número de planta, número de hoja, fecha y diagnóstico. Los datos de Id y Fecha se calculan automáticamente, por lo tanto el llenado de este último campo puede ser opcional. El modelo de clases se muestra en la figura 16.



**Figura 16:** Modelo de clases

### 4.2.2 Modelo de datos

La base de datos realizada en MySQL se muestra en la figura 17, en la cual se puede apreciar la estructura y atributos que la componen, dato por dato.



**Figura 17: Modelo de datos**

#### 4.2.3 Diccionario de datos

En las tablas 3, 4, 5, 6 y 7, se muestra la lista del diccionario de datos con los parámetros que fueron establecidos al momento de crear la base de datos en MySQL.

**Tabla 3: Diccionario de datos inspección**

Nombre del campo	Tipo de dato	Formato	Valor nulo (S/N)	Indexar (S/N)	PK o FK	Clave externa y referencia
id_inspeccion	entero		N	S	PK	
fecha	fecha	AAAA-MM-DD	N	N		
id_sección	entero		N	S	FK	Sección (id_seccion)

**Tabla 4: Diccionario de datos sección**

Nombre del campo	Tipo de dato	Formato	Valor nulo (S/N)	Indexar (S/N)	PK o FK	Clave externa y referencia
id_seccion	entero		N	S	PK	
nombre	cadena(20)		N	N		

**Tabla 5:** Diccionario de datos planta

Nombre del campo	Tipo de dato	Formato	Valor nulo (S/N)	Indexar (S/N)	PK o FK	Clave externa y referencia
id_planta	entero		N	S	PK	
numero	entero		N	N		
id_inspeccion	entero		N	S	FK	Inspeccion (id_inspeccion)

**Tabla 6:** Diccionario de datos hoja

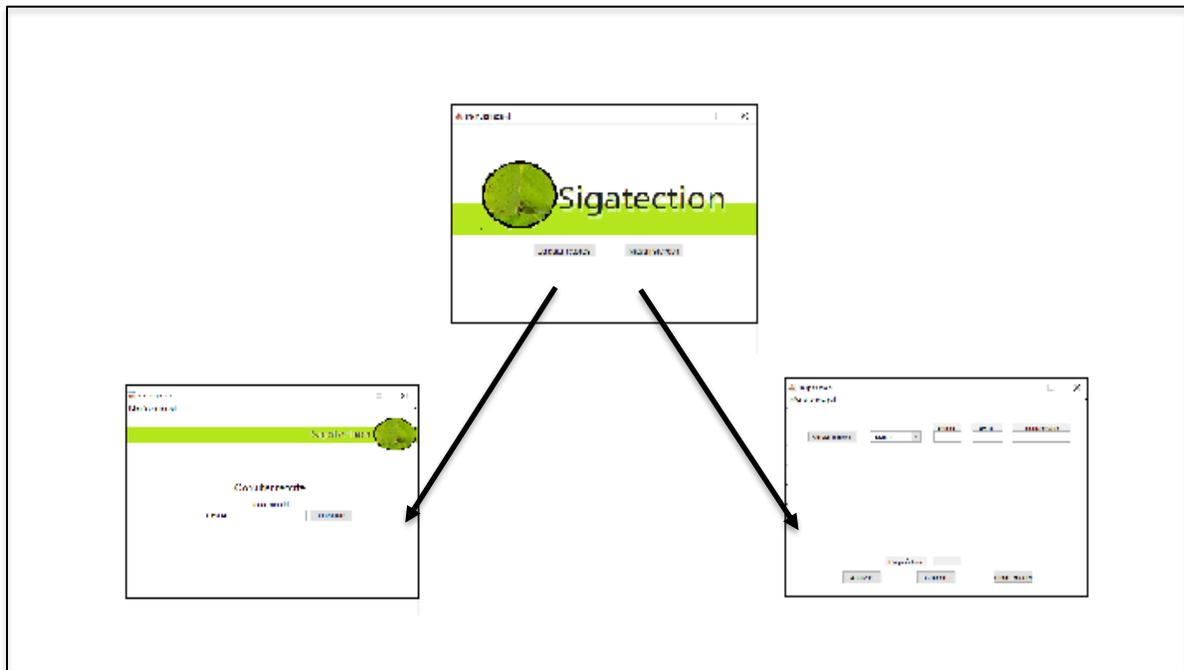
Nombre del campo	Tipo de dato	Formato	Valor nulo (S/N)	Indexar (S/N)	PK o FK	Clave externa y referencia
id_hoja	entero		N	S	PK	
numero	entero		N	N		
id_planta	entero		N	S	FK	Planta(id_planta)
tipo_hoja	entero		N	N		
foto	cadena(200)		N	N		
Id_diagnostico	entero		N	S	FK	Diagnostico (id_diagnostico)

**Tabla 7:** Diccionario de datos diagnóstico

Nombre del campo	Tipo de dato	Formato	Valor nulo (S/N)	Indexar (S/N)	PK, CK o FK	Clave externa y referencia
id_diagnostico	entero		N	S	PK	
nombre	cadena(20)		N	N		

#### 4.2.4 Diagrama de navegación

El diagrama de navegación describe el orden en que se interactúa durante la ejecución de la aplicación. En la figura 18, se aprecia esta representación mediante tres ventanas que componen el sistema y la transición mediante flechas de una ventana a otra. La interacción entre pocas ventanas beneficia en agilizar el proceso de captura de datos, así como facilita el aprendizaje del sistema. Se inicia con la principal del sistema, que funge como menú de selección de operaciones a realizar mediante dos botones, a través de ella se interactúa con el resto de las interfaces que lo conforman.

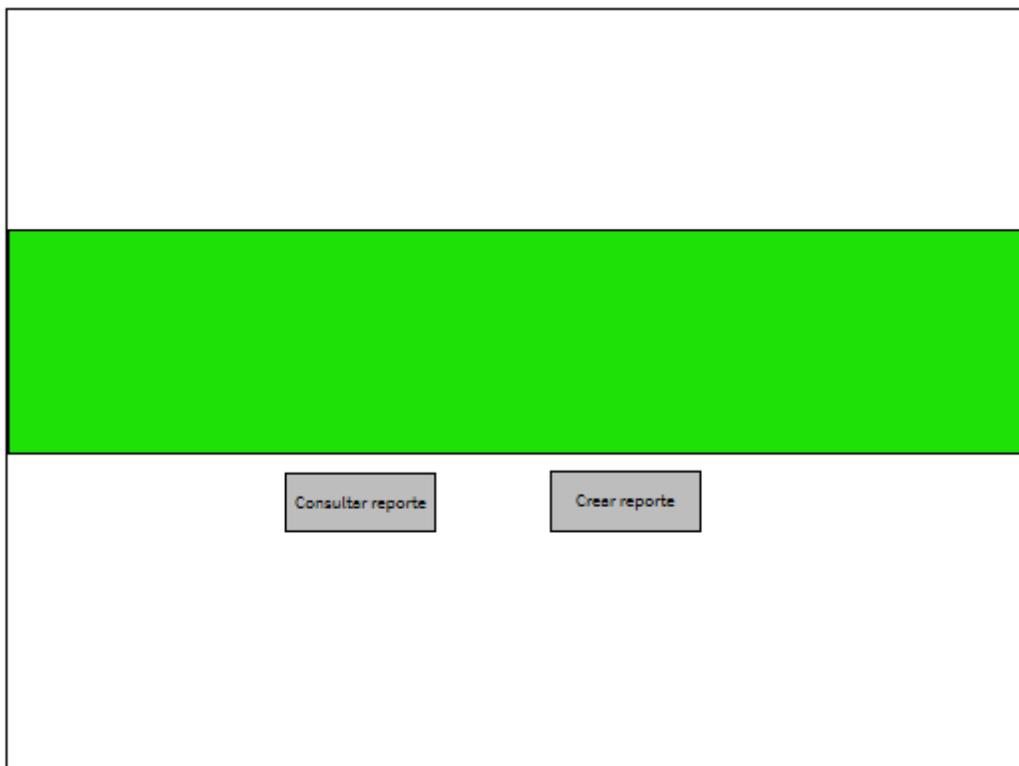


**Figura 18:** Diagrama de Navegación

#### 4.2.5 Modelo de interfaces

En esta sección serán descritas y mostradas a modo de bosquejo, las interfaces que componen el sistema.

En la figura 19 se muestra la pantalla principal que aparecerá al iniciar la aplicación, esta muestra dos botones: El botón “consultar reporte”, que nos llevará a la interfaz mostrada en la figura 20, en la cual se podrán realizar consultas de inspecciones pasadas y el botón “crear reporte”, el cual nos llevará a la interfaz descrita en la figura 21, donde se realizarán los reportes de inspección.



**Figura 19:** Interfaz principal

Al dar clic en el botón “consultar reporte” nos llevará a la interfaz que se muestra en la figura 20, la cual cuenta con un campo para ingresar la fecha que se desea consultar, posteriormente se da clic en el botón “consultar” y automáticamente creará y almacenará en el computador el reporte de la inspección realizada en la fecha solicitada.

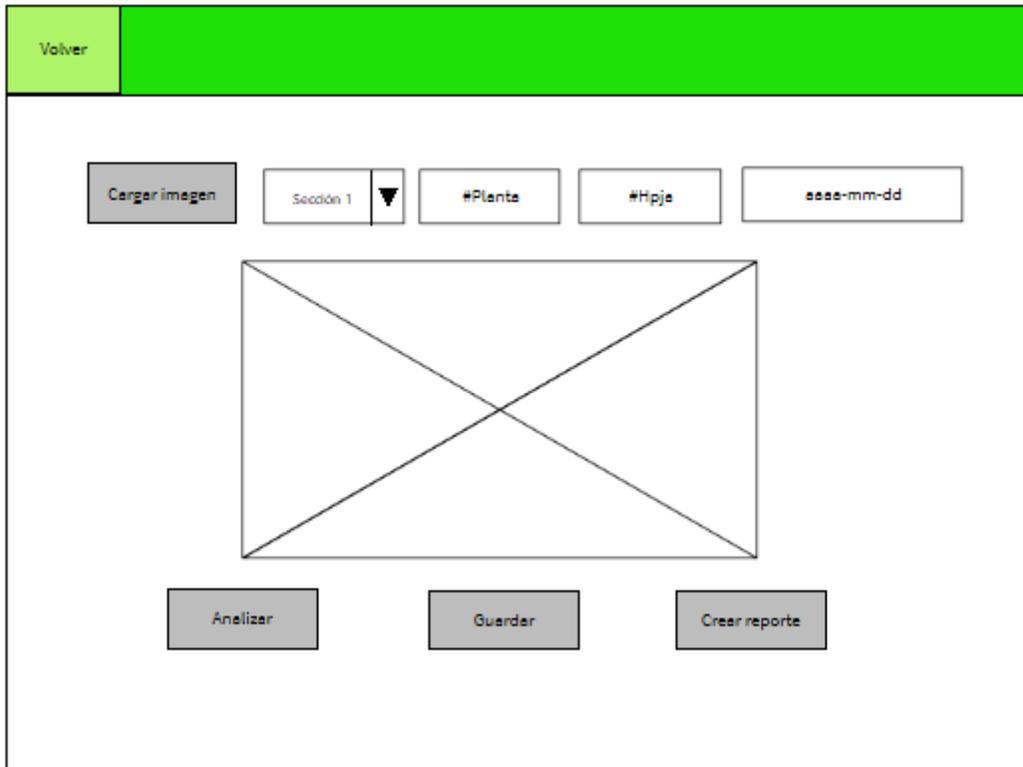
Volver

## Consultar reporte

Fecha:

**Figura 20:** Interfaz consulta de reportes

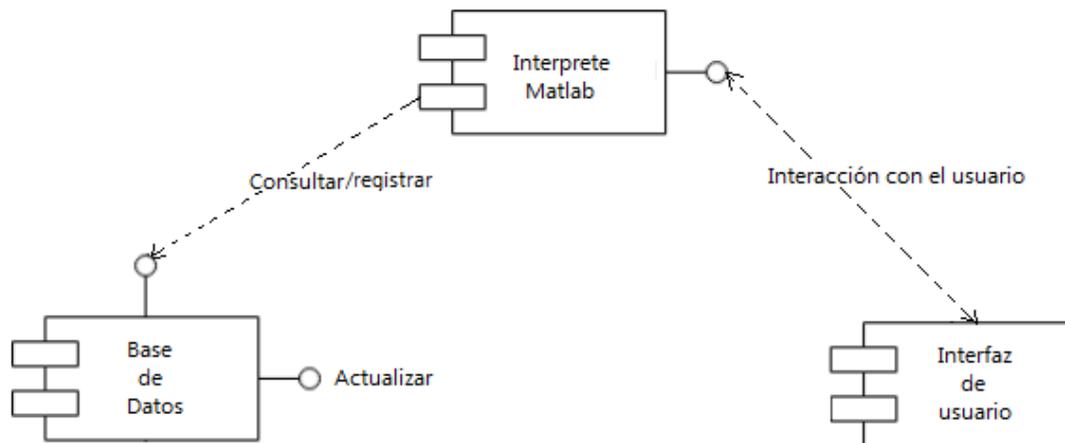
Al dar clic en el botón “Crear reporte” que se muestra en la figura 19, se abrirá la interfaz mostrada en la figura 21, en la cual se selecciona la imagen a inspeccionar en el botón “cargar imagen”, y posteriormente se llenan los campos siguientes. Al dar clic en el botón “analizar” el sistema activará la red neuronal y procesará la imagen previamente ingresada, finalmente al dar clic en el botón “guardar” los datos ingresados en los campos, así como el diagnóstico serán almacenados en la base de datos para, al finalizar el ingreso de datos, desplegar en el reporte que se generará al dar clic en el botón “crear reporte”.



**Figura 21:** Interfaz inspección-reporte

#### 4.2.6 Modelo de componentes

Se puede observar en la figura 22 los componentes que conforman el sistema y las interacciones que se realizan entre ellos.



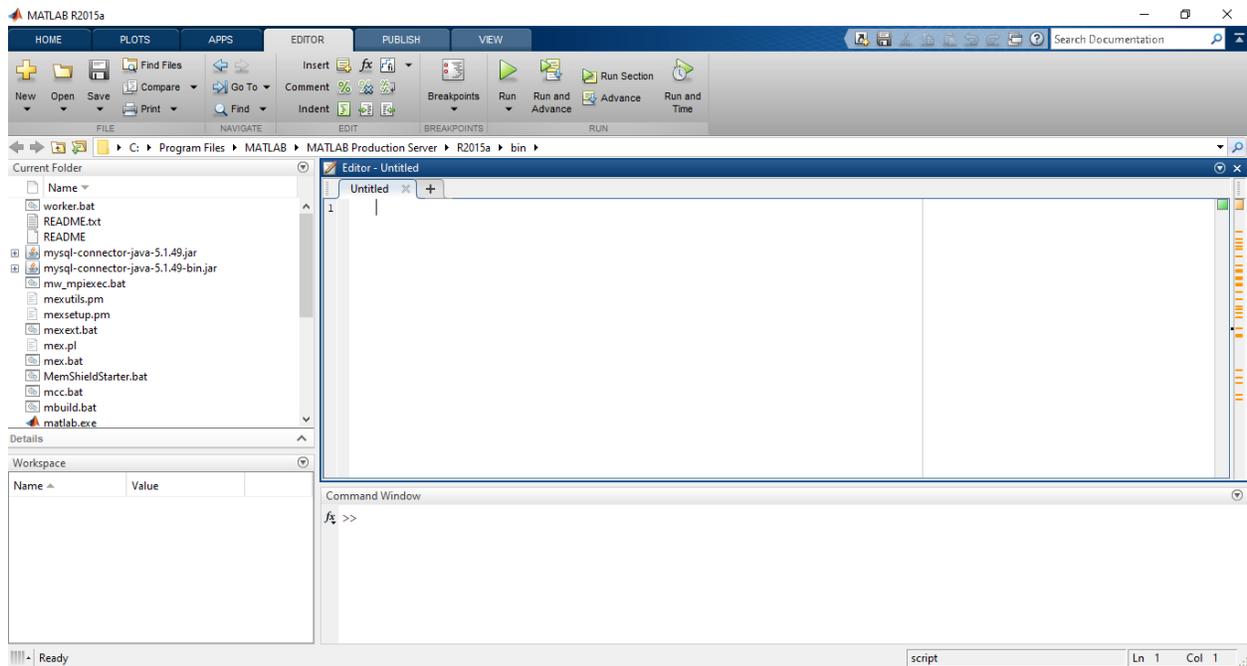
**Figura 22:** Modelo de componentes

### 4.3 Implementación

En este apartado, se muestra el proceso que se llevó a cabo para alcanzar el resultado obtenido durante la investigación, así como el entrenamiento de la red newff, desarrollo del sistema y creación de reportes.

#### 4.3.1 Servidor local

En este módulo se instaló el software MATLAB 2015a con todos sus complementos para generar las pantallas del sistema y poder entrenar la red neuronal new feed-forward. En la figura 23 se muestra la pantalla principal de MATLAB.



**Figura 23:** Pantalla principal MATLAB

#### 4.3.2 Entrenamiento de la NN NFF

Para llevar a cabo el entrenamiento de la red, primero se cargan las imágenes de las diferentes etapas de la SN que servirán como base del entrenamiento, posteriormente por cada imagen, mediante la función `mean` se devuelve la media de los elementos de la imagen a lo largo de la primera dimensión de matriz. A continuación, en la otra variable se almacenan las capas roja y verde, esto último servirá para obtener los promedios y con base a ellos determinar el estado de la hoja, ya sea sana o infectada, y en el caso último determinar los parámetros mediante los promedios que arrojarán el diagnóstico del grado de infección de la hoja. Véase figura 24.

```

ka=num2str(i);
UV = imread(['C:\Users\52331\Desktop\prueba2\sanas\' ,ka, '.jpg']);
PUV = mean(mean(UV));
VUV(:,i) = [PUV(:, :, 1); PUV(:, :, 2)];
i=i+1;

```

**Figura 24:** Entrenamiento NN NFF. Carga de imágenes

Al terminar el ciclo de carga de datos es creada la NN que es utilizada en este trabajo mediante la instrucción `newff`, la cual basa su arquitectura en los siguientes parámetros: la variable que representa la matriz en la que fueron almacenados los vectores de entrada, posteriormente se encuentra el parámetro que representa la matriz de vectores objetivo o salida. Finalmente se encuentra la variable que representa el tamaño de capas ocultas de la NN. Véase figura 25.

```
net=newff(guardar,guardarsauv,400);
```

**Figura 25:** Entrenamiento NFF: creación NN

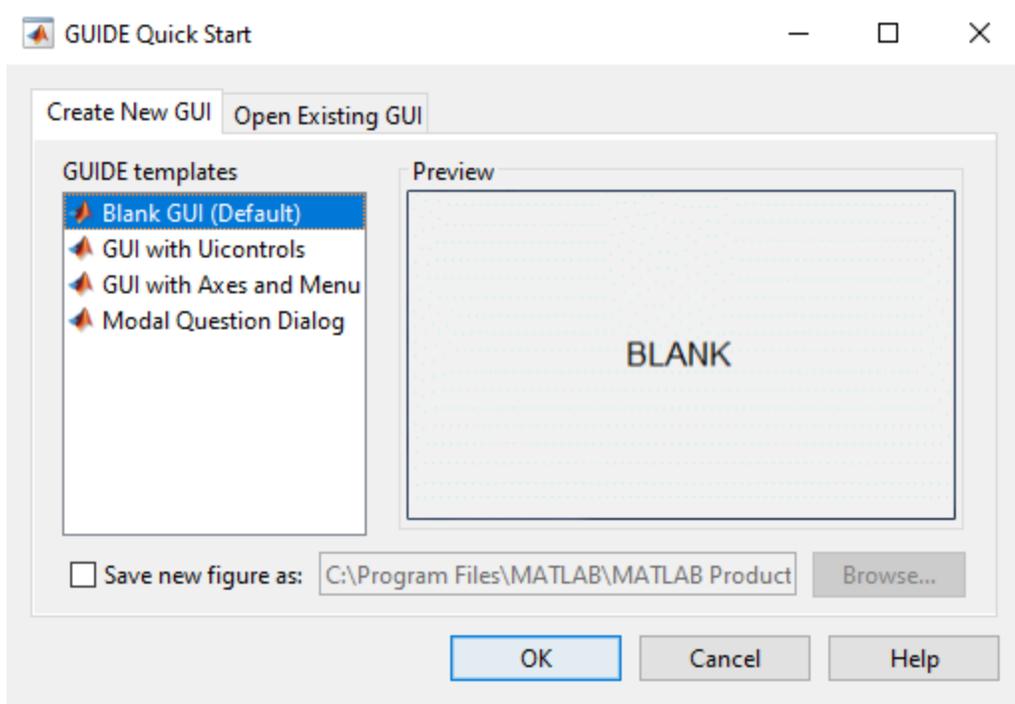
#### 4.3.3 Creación de interfaz gráfica

Una vez creada la NFF NN, se continua con el desarrollo de la aplicación de escritorio, basando las interfaces gráficas en los bosquejos mostrados en el apartado 4.2.4, en las figuras 19, 20 y 21.

Las interfaces se realizaron en MATLAB, mediante la herramienta con la que este software cuenta para crear interfaces gráficas. Para hacer uso de esta herramienta escribimos la palabra “guide” en la consola de MATLAB, la cual al dar enter automáticamente nos abre una ventana para elegir la plantilla de la GUI a crear. Este proceso se muestra en la figura 26 y 27.



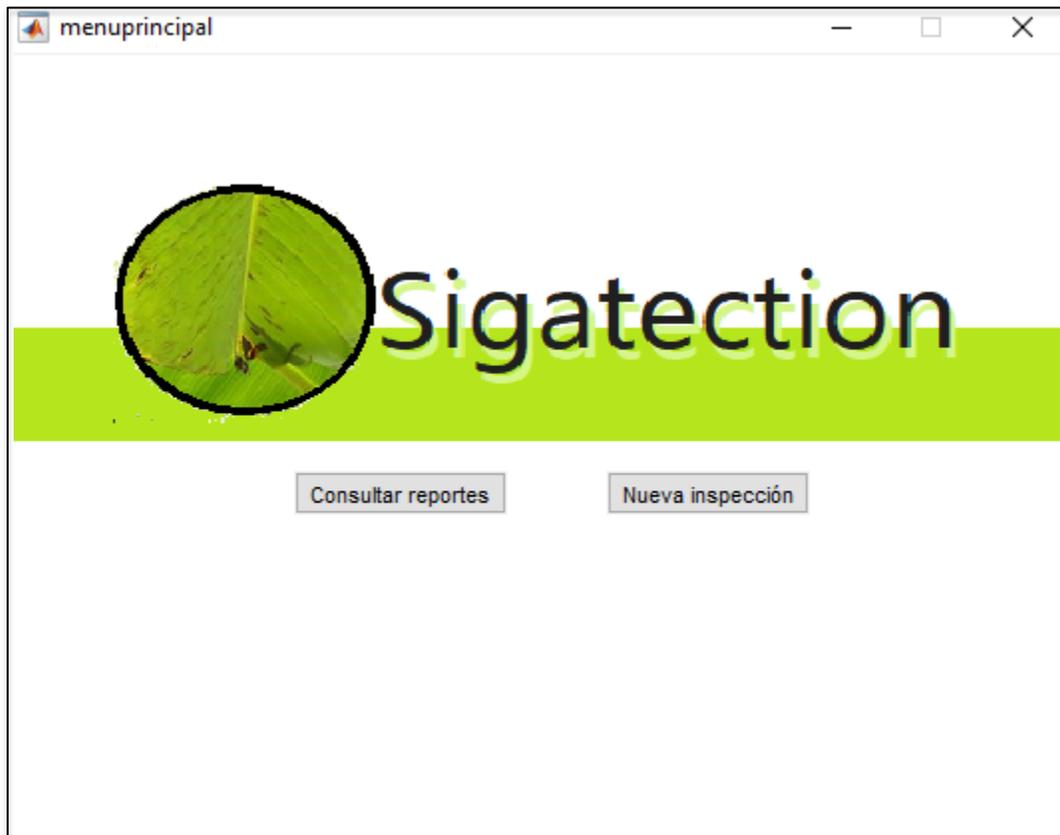
**Figura 26:** Ventana comandos MATLAB



**Figura 27:** Ventana GUI MATLAB

Una vez seleccionada la plantilla Blank GUI(Default) se procedió a replicar los bosquejos prediseñados, mostrando los resultados en las figuras 28, 29 y 30.

#### 4.3.3.1 Interfaz menú principal



**Figura 28:** Interfaz menú principal Sigatection

**Consultar reportes:** Redirige a interfaz “consultar reporte”, dónde podrán generarse reportes de inspecciones realizadas en fechas específicas.

**Nueva inspección:** Redirige a interfaz “inspección”, en la cual se podrá generar el diagnóstico de las imágenes ingresadas y realizar los registros pertinentes en la base de datos.

#### 4.3.3.2 Interfaz consulta de reportes

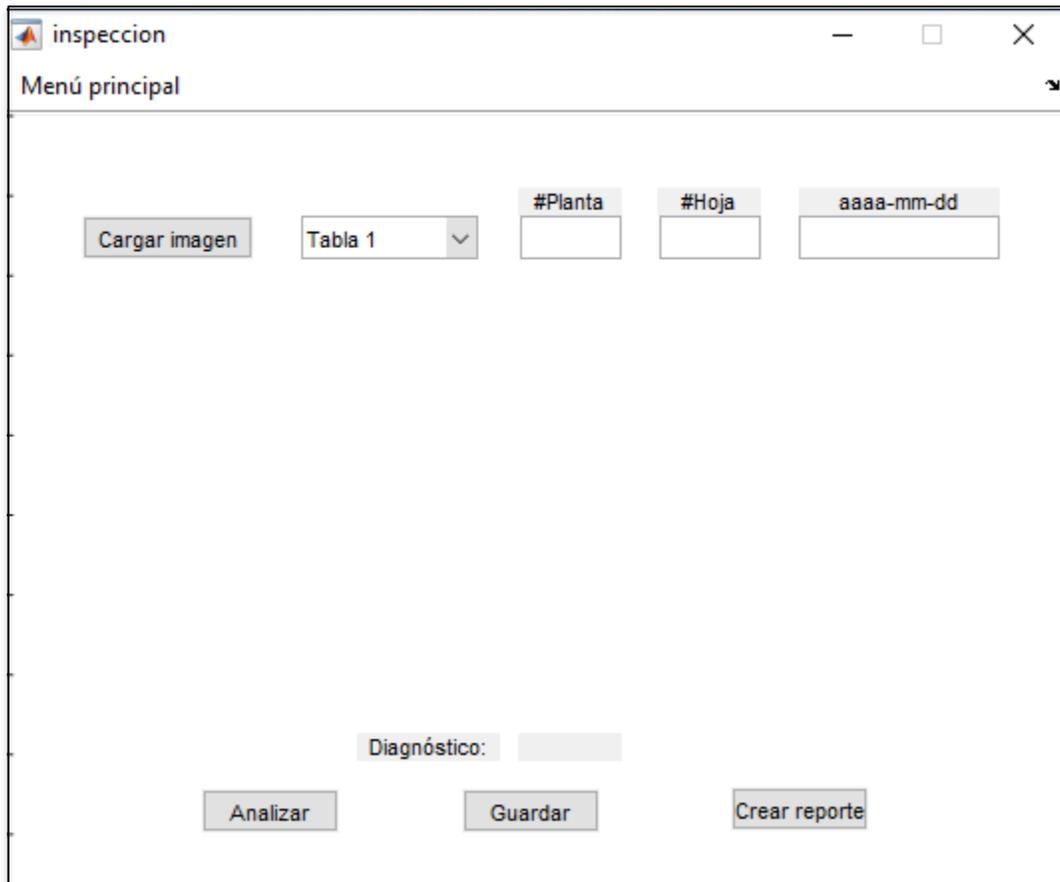


**Figura 29:** Interfaz consulta de reportes

**Fecha:** Se ingresa la fecha de la cual se desea obtener el reporte de inspección, siguiendo la estructura marcada en la parte superior del campo: aaa-mm-dd.

**Consultar:** Botón que genera y guarda en el escritorio del equipo el reporte solicitado.

### 4.3.3.3 Interfaz inspección



**Figura 30:** Interfaz inspección

**Cargar imagen.** Al dar clic se abre una ventana que permite seleccionar la imagen que se desea analizar desde el ordenador.

**Selector tabla.** El usuario selecciona la tabla a la que pertenece la hoja a inspeccionar.

**#Planta.** Se ingresa el número de planta a la que pertenece la hoja a diagnosticar.

**#Hoja.** Se ingresa el número de hoja perteneciente a la planta que se está inspeccionando.

**Aaaa-mm-dd.** Este campo es de llenado opcional ya que se registra automáticamente en la base de datos, en el cual se ingresa la fecha del día en que se está realizando la inspección.

**Analizar.** Realiza el análisis de la imagen ingresada.

**Diagnóstico.** Se muestra el diagnóstico arrojado por la red.

**Guardar.** Registra en la base de datos los datos previamente ingresados: Tabla, número de planta, número de hoja, fecha (en caso de haber sido ingresada) y diagnóstico,

**Crear reporte.** Genera y guarda en el escritorio del equipo el reporte de la inspección recién realizada.

#### 4.3.4 Creación de la base de datos

A la par de la creación de las interfaces se llevó a cabo la creación de la base de datos para el almacenamiento y consulta de los registros. La base de datos se realizó mediante el gestor MySQL, con apoyo de Apache (ver figura 31) y el editor de texto Visual Studio Code. Para el desarrollo de la base de datos primero se definieron las tablas y atributos que la compondrían en base a los requisitos del sistema, posteriormente se llevó a cabo la relación entre las tablas. En la figura 32 y 33 se muestra un pequeño fragmento del código de la base de datos.

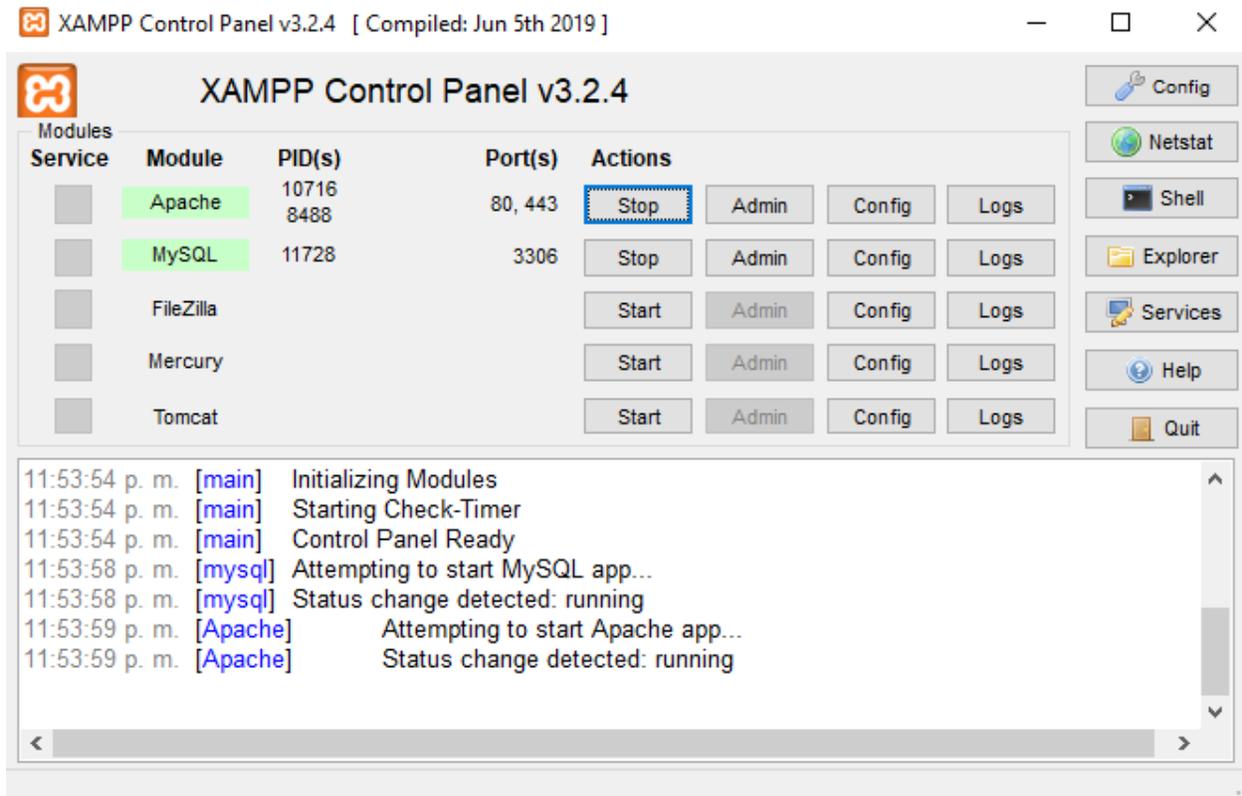


Figura 31: Interfaz Apache

```

DROP TABLE IF EXISTS `diagnostico`;
CREATE TABLE `diagnostico` (
  `id_diag` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `nombre_d` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  PRIMARY KEY (`id_diag`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 7 CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;

-- Table structure for hoja
DROP TABLE IF EXISTS `hoja`;
CREATE TABLE `hoja` (
  `id_hoja` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `planta_id` int(10) UNSIGNED NOT NULL,
  `diag_id` int(10) UNSIGNED NOT NULL,
  `n_hoja` int(10) UNSIGNED NOT NULL,
  `foto` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  PRIMARY KEY (`id_hoja`) USING BTREE,
  INDEX `fk_hoj_pla_idx` (`planta_id`) USING BTREE,
  INDEX `fk_hoj_dia_idx` (`diag_id`) USING BTREE,
  CONSTRAINT `fk_hoj_dia` FOREIGN KEY (`diag_id`) REFERENCES `diagnostico` (`id_diag`) ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT `fk_hoj_pla` FOREIGN KEY (`planta_id`) REFERENCES `planta` (`id_planta`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 30 CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;

```

Figura 32: Fragmento BD tabla uno

```

DROP TABLE IF EXISTS `planta`;
CREATE TABLE `planta` (
  `id_planta` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `numero_p` int(10) UNSIGNED NOT NULL,
  `inspeccion_id` int(10) UNSIGNED NOT NULL,
  `seccion_id` int(10) UNSIGNED NOT NULL,
  PRIMARY KEY (`id_planta`) USING BTREE,
  INDEX `fk_pla_ins_idx` (`inspeccion_id`) USING BTREE,
  INDEX `fk_pla_sec` (`seccion_id`) USING BTREE,
  CONSTRAINT `fk_pla_ins` FOREIGN KEY (`inspeccion_id`) REFERENCES `inspeccion` (`id_inspeccion`) ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT `fk_pla_sec` FOREIGN KEY (`seccion_id`) REFERENCES `seccion` (`id_seccion`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE = InnoDB AUTO_INCREMENT = 31 CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;

```

**Figura 33:** Fragmento BD tabla 2

Uno de los factores base en la codificación de la base de datos fue la utilización de disparadores (véase ejemplo en la figura 34), ya que permitió evitar la duplicidad de valores al momento de ingresar todos los datos que conlleva una inspección, facilitando las consultas e impresión de reportes. Esto dio por resultado una base de datos flexible para su reutilización en proyectos de similar finalidad.

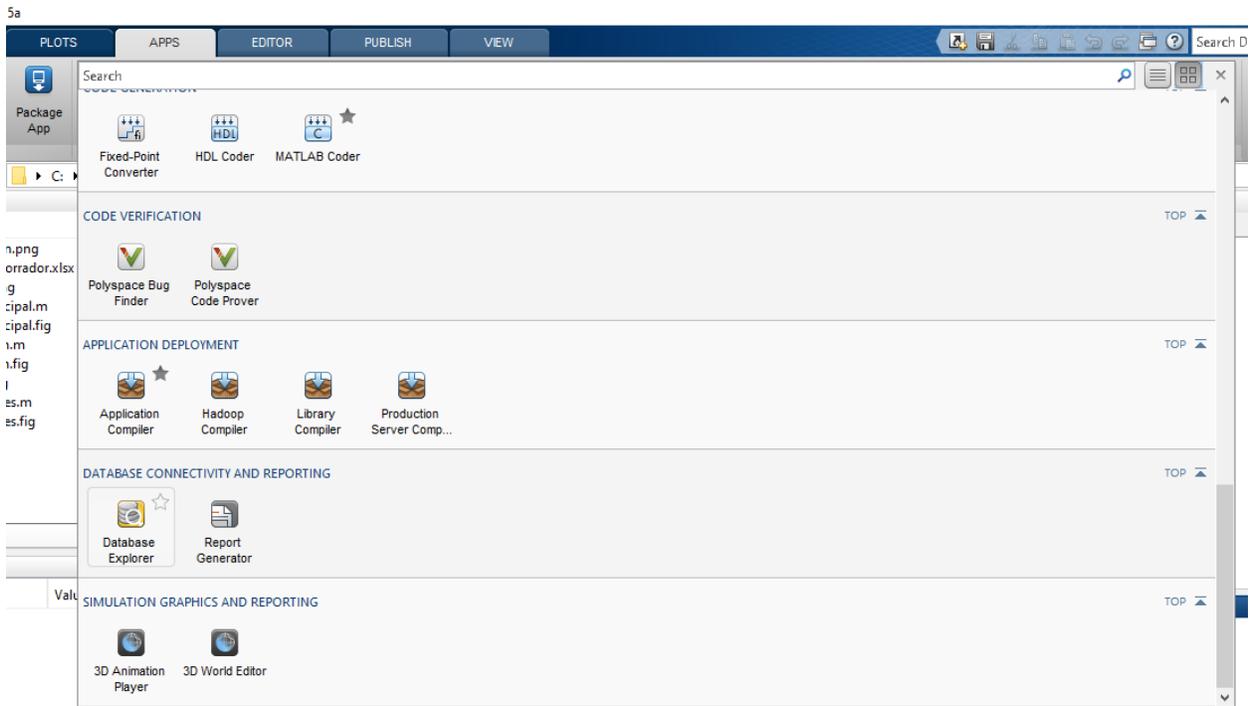
```

DROP TRIGGER IF EXISTS `trValidarHoja`;
delimiter ;;
CREATE TRIGGER `trValidarHoja` BEFORE INSERT ON `hoja` FOR EACH ROW BEGIN

```

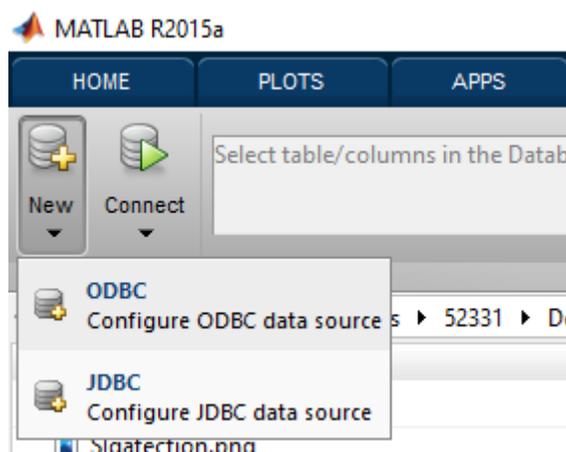
**Figura 34:** Fragmento disparador uno

Una vez codificada la base de datos, se procedió a la conexión de la esta con MATLAB, esto mediante el uso de la aplicación Database Explorer ubicada en la sección “Database connectivity and reporting” del menú “APPS” en MATLAB, como se puede apreciar en la figura 35.



**Figura 35:** Database connectivity and reporting

Dentro de Database Explorer, se configuró la base de datos mediante la opción “New”, opción ODBC (Open DataBase Connectivity), el cual es un estándar de acceso a las bases de datos. Véase figura 36.



**Figura 36:** Conexión ODBC

Una vez realizada la conexión de la base de datos con MATLAB, el ingreso de los registros ella se hizo importando la sección de código correspondiente a la tabla y campo de la base de datos, para posteriormente implementarlo en el código de la función de MATLAB donde se fuese a utilizar, mediante el manejo de variables y SQL, como se muestra en el fragmento de código de la figura 37.

```
conexion = database('mysql','root','');
insert (conexion,'inspeccion',{'basura'},{'1'});
close (conexion);

setdbprefs('DataReturnFormat', 'cellarray');
setdbprefs('NullNumberRead', 'NaN');
setdbprefs('NullStringRead', 'null');

conn = database('mysql', 'root', '');

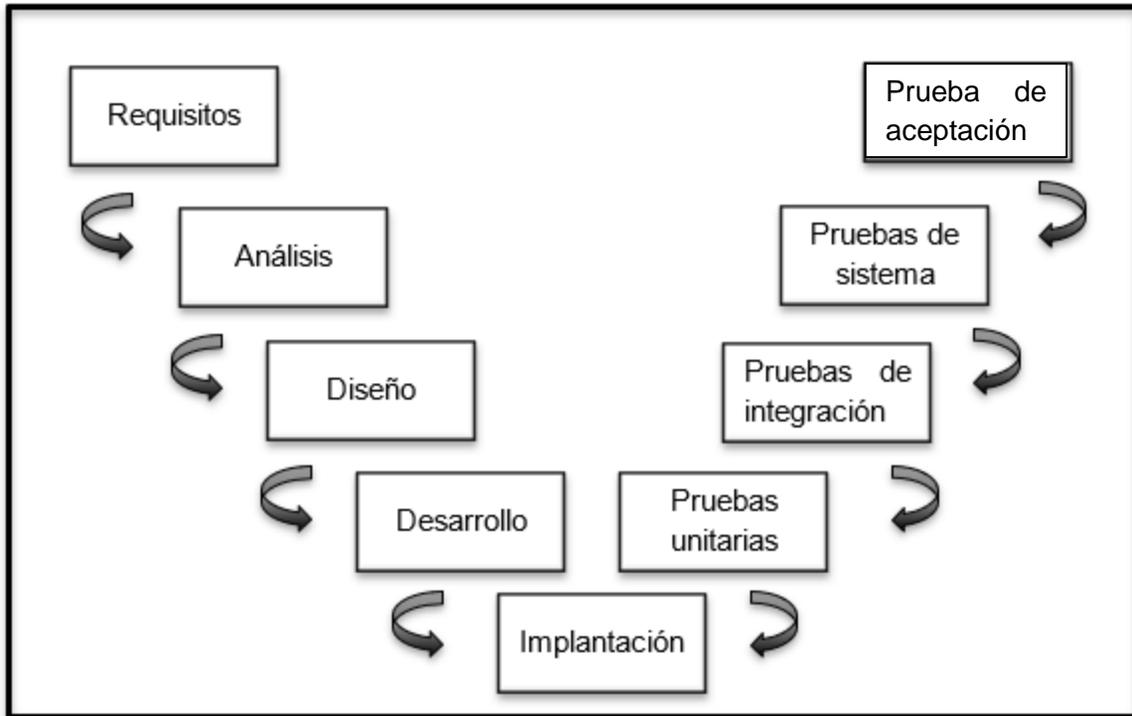
curs = exec(conn, ['SELECT inspeccion.id_inspeccion'...
    ' FROM sigatection.inspeccion ' 'ORDER BY id_inspeccion DESC ' 'LIMIT 1']);

curs = fetch(curs);
```

*Figura 37: Conexión DB con MATLAB*

#### **4.4 Verificación y Validación**

Con el objetivo de comprobar y determinar el nivel de funcionamiento y efectividad del sistema, se realizaron las pruebas que se describirán a continuación, comprobando el nivel de cumplimiento respecto a los requerimientos del sistema, validando cada componente en fin de que funcione correctamente. Para esto se utilizó el modelo V de pruebas, el cual, de acuerdo con (Ramírez, 2010) es una secuencia de pasos para el desarrollo del ciclo de vida de un proyecto, la cual describe todas las actividades que deben seguirse para el desarrollo de dicho proyecto. El lado izquierdo de la V es la representación de la corriente donde se definen los requisitos del sistema, mientras que el lado derecho de la V significa la integración de todos los pasos y su verificación definida en la parte izquierda. Véase figura 38.



**Figura 38:** Modelo de pruebas en V

#### 4.4.1 Pruebas unitarias

Estas pruebas se realizaron con la finalidad de detectar fallas y probar el correcto funcionamiento de cada uno de los módulos que componen el sistema, con enfoque en los componentes. Durante la fase de codificación del sistema se fueron haciendo las pruebas correspondientes a cada nueva función que se fuera desarrollando, con la finalidad de que corroborar que funcionaran adecuadamente y conforme a lo requerido por el usuario.

#### 4.4.2 Pruebas de integración

Durante la programación del sistema, una vez terminado un módulo con sus respectivas pruebas unitarias, se procedió a realizar las pruebas funcionales de integración con el resto de los módulos funcionales, para asegurar de que el sistema trabajara adecuadamente en conjunto con el resto de los componentes. Para esto se realizaron pruebas como: Pasar de una interfaz a otra, revisar que los elementos se desplegaran apropiadamente, ingresar información a la base de datos, revisar que se almacenara en

tiempo y forma y arrojará la información correcta en los reportes mediante solicitudes específicas.

#### 4.4.3 Pruebas de sistema

La finalidad de estas pruebas fue poner a prueba el sistema completo, asegurándonos de haber cumplido con los requisitos solicitados por el usuario, tanto funcionales como no funcionales, siendo nuestra prioridad el correcto funcionamiento de la red neuronal y creación de reportes. Para esto se sometieron diversas imágenes de prueba al sistema, abarcando todos los estados de la SN a detectar, registrando que el diagnóstico fuera acertado. Los reportes se sometieron a prueba haciendo diversas solicitudes en base a registros varios, realizados en diferentes días, corroborando la creación con los datos solicitados y almacenamiento en el equipo.

#### 4.4.4 Prueba de aceptación

Esta prueba fue realizada con el objetivo de que el usuario evaluara el sistema, en base a sus necesidades y los requisitos solicitados. Se aprobó de manera satisfactoria, confirmando la usabilidad y eficacia del sistema.

#### 4.4.5 Prueba de tiempo de ejecución

Para validación y verificación del sistema, en conjunto con las pruebas anteriormente mencionadas, se implementó la prueba de tiempo de ejecución, a través de la herramienta “profiler” de MATLAB, la cual permite medir el tiempo que tarda en ejecutarse determinado módulo del programa. Los resultados obtenidos de la prueba de tiempo de ejecución del primer interfaz se muestran en la tabla 8.

**Tabla 8:** Prueba de tiempo de ejecución interfaz principal

<b>Function Name</b>	<b>Calls</b>	<b>Total Time</b>	<b>Self Time*</b>	<b>Total Time Plot (dark band = self time)</b>
<a href="#">FigFile.FigFile&gt;FigFile.FigFile</a>	1	0.059 s	0.001 s	
<a href="#">FigFile.FigFile&gt;FigFile.set.FigFormat</a>	1	0 s	0.000 s	
<a href="#">FigFile.FigFile&gt;FigFile.set.Format3Data</a>	1	0 s	0.000 s	
<a href="#">...&gt;FigFile.set.RequiredMatlabVersion</a>	1	0.001 s	0.001 s	
<a href="#">FigFile.FigFile&gt;localIsHG</a>	1	0 s	0.000 s	
<a href="#">FigFile.FigFile&gt;localVerifyObject</a>	1	0 s	0.000 s	
<a href="#">FigFile.read</a>	1	0.058 s	0.054 s	
<a href="#">...leanupDockedFigWarnState(wsDockedFig)</a>	1	0 s	0.000 s	
<a href="#">FigFile.read&gt;@()warning(ws)</a>	1	0.001 s	0.001 s	
<a href="#">FigFile.read&gt;cleanupDockedFigWarnState</a>	1	0 s	0.000 s	
<a href="#">...leanupDockedFigWarnState(wsDockedFig)</a>	1	0 s	0.000 s	
<a href="#">FigFile.read&gt;create@()warning(ws)</a>	1	0 s	0.000 s	
<a href="#">FigFile.read&gt;localGetSaveVersion</a>	1	0.001 s	0.000 s	
<a href="#">LoadRecursionGuard&gt;@()obj_remove(file)</a>	1	0.002 s	0.000 s	
<a href="#">...ursionGuard&gt;LoadRecursionGuard.add</a>	1	0.002 s	0.001 s	
<a href="#">...rd&gt;LoadRecursionGuard.addAndRemove</a>	1	0.002 s	0.000 s	
<a href="#">...ard&gt;LoadRecursionGuard.getInstance</a>	1	0 s	0.000 s	
<a href="#">...Guard&gt;LoadRecursionGuard.isInStack</a>	1	0.001 s	0.001 s	

<a href="#">genvarname&gt;isString</a>	6	0.001 s	0.001 s	
<a href="#">graphics\private\claNotify</a>	1	0 s	0.000 s	
<a href="#">graphics\private\clo</a>	1	0.005 s	0.005 s	
<a href="#">gui_mainfcn</a>	1	0.201 s	0.006 s	██████████
<a href="#">gui_mainfcn&gt;local_openfig</a>	2	0.135 s	0.000 s	██████████
<a href="#">guidata</a>	5	0.004 s	0.003 s	
<a href="#">guidata&gt;getParentFigure</a>	5	0.001 s	0.001 s	
<a href="#">guidemfile</a>	1	0 s	0.000 s	
<a href="#">guidemfile&gt;getToolBarToolInFigure</a>	1	0 s	0.000 s	
<a href="#">...;restoreToolBarToolPredefinedCallback</a>	1	0 s	0.000 s	
<a href="#">guihandles</a>	1	0.006 s	0.001 s	
<a href="#">guihandles&gt;createHandles</a>	1	0.005 s	0.002 s	
<a href="#">guihandles&gt;getParentFigure</a>	1	0 s	0.000 s	
<a href="#">hgload</a>	1	0.064 s	0.000 s	██████
<a href="#">...o.e)figureInstanceCreatedForLoad(o.e)</a>	1	0 s	0.000 s	
<a href="#">...o.e)figureInstanceCreatedForLoad(o.e)</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;figureInstanceCreatedForLoad</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;localCheckRecursion</a>	1	0.002 s	0.000 s	
<a href="#">hgload&gt;localCheckRequiredVersion</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;localParseOptions</a>	1	0.001 s	0.001 s	
<a href="#">hgload&gt;localPostClassActions</a>	1	0 s	0.000 s	
<a href="#">imagesci\private\imftype</a>	1	0.003 s	0.001 s	
<a href="#">imagesci\private\ispng</a>	1	0.001 s	0.001 s	
<a href="#">imagesci\private\pngreadc (MEX-file)</a>	1	0.002 s	0.002 s	
<a href="#">imagesci\private\readpng</a>	1	0.018 s	0.004 s	█
<a href="#">imagesci\private\readpng&gt;parse_args</a>	1	0 s	0.000 s	
<a href="#">imformats</a>	1	0.001 s	0.001 s	
<a href="#">imformats&gt;find_in_registry</a>	1	0 s	0.000 s	
<a href="#">imread</a>	1	0.027 s	0.004 s	█
<a href="#">imread&gt;parse_inputs</a>	1	0.002 s	0.002 s	
<a href="#">intmax</a>	3	0.001 s	0.001 s	



<a href="#">...pace.MatlabWorkspaceListener.swl(sw)</a>	2	0.001 s	0.001 s	
<a href="#">...pace.MatlabWorkspaceListener.swl(sw)</a>	2	0 s	0.000 s	
<a href="#">workspacefunc&gt;createComplexScalar</a>	2	0.002 s	0.001 s	
<a href="#">workspacefunc&gt;getAbstractValueSummaryJ</a>	4	0.002 s	0.000 s	
<a href="#">workspacefunc&gt;getCleanupHandler</a>	2	0.001 s	0.001 s	
<a href="#">workspacefunc&gt;getShortValueObjectJ</a>	6	0.011 s	0.006 s	█
<a href="#">workspacefunc&gt;getShortValueObjectsJ</a>	2	0.011 s	0.000 s	█
<a href="#">workspacefunc&gt;getWhosInformation</a>	2	0.002 s	0.002 s	
<a href="#">workspacefunc&gt;getclass</a>	4	0.002 s	0.002 s	
<a href="#">workspacefunc&gt;num2complex</a>	6	0.005 s	0.001 s	

En la tabla 9, se muestran algunos resultados de la prueba de tiempo de ejecución realizada a la segunda interfaz.

**Tabla 9:** Prueba tiempo de ejecución interfaz reportes

<b>Function Name</b>	<b>Calls</b>	<b>Total Time</b>	<b>Self Time*</b>	<b>Total Time Plot (dark band = self time)</b>
<a href="#">...'.hObject.eventdata.guidata(hObject)</a>	1	0 s	0.000 s	
<a href="#">...'.hObject.eventdata.guidata(hObject)</a>	1	0.008 s	0.000 s	
<a href="#">FigFile.FigFile&gt;FigFile.FigFile</a>	1	0.158 s	0.000 s	█
<a href="#">FigFile.FigFile&gt;FigFile.set.FigFormat</a>	1	0 s	0.000 s	
<a href="#">FigFile.FigFile&gt;FigFile.set.Format3Data</a>	1	0 s	0.000 s	
<a href="#">...&gt;FigFile.set.RequiredMatlabVersion</a>	1	0 s	0.000 s	
<a href="#">FigFile.FigFile&gt;localsHG</a>	1	0 s	0.000 s	
<a href="#">FigFile.FigFile&gt;localVerifyObject</a>	1	0 s	0.000 s	
<a href="#">FigFile.read</a>	1	0.158 s	0.157 s	█
<a href="#">...leanupDockedFigWarnState(wsDockedFig)</a>	1	0 s	0.000 s	
<a href="#">FigFile.read&gt;@()warning(ws)</a>	1	0 s	0.000 s	
<a href="#">FigFile.read&gt;cleanupDockedFigWarnState</a>	1	0 s	0.000 s	
<a href="#">...leanupDockedFigWarnState(wsDockedFig)</a>	1	0 s	0.000 s	
<a href="#">FigFile.read&gt;create@()warning(ws)</a>	1	0 s	0.000 s	

<a href="#">...&gt;showHiddenHandlesToFindAllHandles</a>	1	0 s	0.000 s	
<a href="#">flip</a>	1	0 s	0.000 s	
<a href="#">fliplr</a>	3	0 s	0.000 s	
<a href="#">flipud</a>	2	0 s	0.000 s	
<a href="#">fullfile</a>	2	0.001 s	0.001 s	
<a href="#">fullfile&gt;refinePath</a>	2	0 s	0.000 s	
<a href="#">gcbf</a>	2	0.001 s	0.001 s	
<a href="#">genvarname</a>	6	0.030 s	0.012 s	█
<a href="#">genvarname&gt;isCellString</a>	12	0 s	0.000 s	
<a href="#">genvarname&gt;isString</a>	6	0 s	0.000 s	
<a href="#">graphics\private\claNotify</a>	1	0 s	0.000 s	
<a href="#">graphics\private\clo</a>	1	0.009 s	0.009 s	
<a href="#">gui_mainfcn</a>	3	0.407 s	0.019 s	██████████
<a href="#">...nfcn&gt;local_isInvokeActiveXCallback</a>	2	0 s	0.000 s	
<a href="#">gui_mainfcn&gt;local_isInvokeHGCallback</a>	2	0 s	0.000 s	
<a href="#">gui_mainfcn&gt;local_openfig</a>	2	0.299 s	0.013 s	██████████
<a href="#">guidata</a>	7	0.009 s	0.008 s	
<a href="#">guidata&gt;getParentFigure</a>	7	0.001 s	0.001 s	
<a href="#">guidemfile</a>	1	0 s	0.000 s	
<a href="#">guidemfile&gt;getToolBarToolInFigure</a>	1	0 s	0.000 s	
<a href="#">...:restoreToolBarToolPredefinedCallback</a>	1	0 s	0.000 s	
<a href="#">guihandles</a>	1	0.009 s	0.000 s	
<a href="#">guihandles&gt;createHandles</a>	1	0.009 s	0.009 s	
<a href="#">guihandles&gt;getParentFigure</a>	1	0 s	0.000 s	
<a href="#">hgload</a>	1	0.163 s	-0.000 s	██████
<a href="#">...o.e)figureInstanceCreatedForLoad(o.e)</a>	1	0 s	0.000 s	
<a href="#">...o.e)figureInstanceCreatedForLoad(o.e)</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;figureInstanceCreatedForLoad</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;localCheckRecursion</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;localCheckRequiredVersion</a>	1	0 s	0.000 s	
<a href="#">hgload&gt;localParseOptions</a>	1	0 s	0.000 s	

<a href="#">hglload&gt;localPostClassActions</a>	1	0 s	0.000 s	
<a href="#">imagesci\private\imftype</a>	1	0 s	0.000 s	
<a href="#">imagesci\private\ispng</a>	1	0 s	0.000 s	
<a href="#">imagesci\private\pngreadc (MEX-file)</a>	1	0 s	0.000 s	
<a href="#">imagesci\private\readpng</a>	1	0.016 s	0.008 s	
<a href="#">imagesci\private\readpng&gt;parse_args</a>	1	0 s	0.000 s	
<a href="#">imformats</a>	1	0 s	0.000 s	
<a href="#">imformats&gt;find_in_registry</a>	1	0 s	0.000 s	
<a href="#">imread</a>	1	0.025 s	0.009 s	
<a href="#">imread&gt;parse_inputs</a>	1	0 s	0.000 s	
<a href="#">intmax</a>	3	0 s	0.000 s	
<a href="#">iscom</a>	2	0 s	0.000 s	
<a href="#">iskeyword</a>	3	0 s	0.000 s	
<a href="#">ismember</a>	1	0.008 s	0.000 s	
<a href="#">ismember&gt;ismemberBuiltinTypes</a>	1	0.008 s	0.008 s	
<a href="#">ismember&gt;ismemberR2012a</a>	1	0.008 s	0.000 s	
<a href="#">ispc</a>	11	0 s	0.000 s	
<a href="#">isunix</a>	4	0 s	0.000 s	
<a href="#">loadUtil</a>	1	0 s	0.000 s	
<a href="#">loadUtil&gt;localsProp</a>	1	0 s	0.000 s	
<a href="#">movegui</a>	2	0.098 s	0.098 s	█
<a href="#">movegui&gt;get_parent_fig</a>	2	0 s	0.000 s	
<a href="#">newplot</a>	1	0.009 s	0.000 s	
<a href="#">newplot&gt;ObserveAxesNextPlot</a>	1	0.009 s	0.000 s	
<a href="#">newplot&gt;ObserveFigureNextPlot</a>	1	0 s	0.000 s	
<a href="#">onCleanup&gt;onCleanup.delete</a>	7	0.007 s	0.000 s	
<a href="#">onCleanup&gt;onCleanup.onCleanup</a>	7	0.001 s	0.001 s	
<a href="#">openfigLegacy</a>	2	0.286 s	0.002 s	█
<a href="#">...Legacy&gt;@(o,e)cleanUp(o.e.filename)</a>	1	0.002 s	0.000 s	
<a href="#">openfigLegacy&gt;cleanUp</a>	1	0.002 s	0.001 s	
<a href="#">...&gt;create@(o,e)cleanUp(o.e.filename)</a>	1	0 s	0.000 s	

En la tabla 10, se muestra un fragmento de los resultados de la prueba realizada a la tercer interfaz.

**Tabla 10:** Tabla tiempo de ejecución interfaz inspección

<b>Function Name</b>	<b>Calls</b>	<b>Total Time</b>	<b>Self Time*</b>	<b>Total Time Plot (dark band = self time)</b>
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	0.015 s	0.000 s	
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	0 s	0.000 s	
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	0.016 s	0.000 s	
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	0 s	0.000 s	
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	0 s	0.000 s	
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	4.298 s	0.001 s	
<a href="#">...hObject.eventdata.guidata(hObject))</a>	1	5.549 s	0.000 s	
<a href="#">...leChooser.addPeerFileExtensionFilters</a>	1	0.015 s	0.015 s	
<a href="#">...ileChooser&gt;FileChooser.blockMATLAB</a>	1	0.017 s	0.000 s	
<a href="#">...FileChooser&gt;FileChooser.createPeer</a>	1	0.016 s	0.016 s	
<a href="#">...er.FileChooser&gt;FileChooser.dispose</a>	1	0.001 s	0.001 s	
<a href="#">...ser&gt;FileChooser.getPeerFilterIndex</a>	1	0.001 s	0.001 s	
<a href="#">...FileChooser&gt;FileChooser.initialize</a>	2	0.047 s	0.000 s	
<a href="#">...eChooser&gt;FileChooser.prepareDialog</a>	1	0.015 s	0.000 s	
<a href="#">...FileChooser.prepareDialog/localupdate</a>	1	0.017 s	0.001 s	
<a href="#">...Chooser&gt;FileChooser.set.FileFilter</a>	3	0.016 s	0.016 s	
<a href="#">...er&gt;FileChooser.set.InitialFileName</a>	3	0.031 s	0.000 s	
<a href="#">...leChooser&gt;FileChooser.setFileExist</a>	1	0 s	0.000 s	
<a href="#">...ser&gt;FileChooser.setPeerCurrentFile</a>	1	0 s	0.000 s	
<a href="#">...leChooser&gt;FileChooser.setPeerTitle</a>	1	0 s	0.000 s	
<a href="#">FileChooser.FileChooser&gt;FileChooser.show</a>	1	3.796 s	0.030 s	
<a href="#">...gt;FileChooser.showPeerAndBlockMATLAB</a>	1	3.751 s	0.001 s	
<a href="#">...eChooser&gt;FileChooser.unblockMATLAB</a>	1	0 s	0.000 s	
<a href="#">...ooser&gt;FileChooser.updateFromDialog</a>	1	0.013 s	0.012 s	
<a href="#">...leExtensionFilter.FileExtensionFilter</a>	1	0 s	0.000 s	

<a href="#">inspeccion</a>	12	10.471 s	0.001 s	
<a href="#">inspeccion&gt;edit1_Callback</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;edit1_CreateFcn</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;edit2_Callback</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;edit2_CreateFcn</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;edit3_CreateFcn</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;inspeccion_OpeningFcn</a>	1	0.022 s	0.000 s	
<a href="#">inspeccion&gt;inspeccion_OutputFcn</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;pushbutton1_Callback</a>	1	4.293 s	0.060 s	
<a href="#">inspeccion&gt;pushbutton2_Callback</a>	1	5.549 s	0.084 s	
<a href="#">inspeccion&gt;pushbutton3_Callback</a>	1	0.278 s	0.063 s	
<a href="#">inspeccion&gt;pushbutton4_Callback</a>	1	0.131 s	0.037 s	
<a href="#">inspeccion&gt;seccion_Callback</a>	1	0 s	0.000 s	
<a href="#">inspeccion&gt;seccion_CreateFcn</a>	1	0 s	0.000 s	
<a href="#">int2str</a>	127	0.021 s	0.021 s	
<a href="#">intmax</a>	2	0 s	0.000 s	
<a href="#">intmin</a>	128	0.016 s	0.016 s	
<a href="#">isNotParallelData</a>	1	0 s	0.000 s	
<a href="#">isScalar</a>	10	0 s	0.000 s	
<a href="#">isScalar</a>	7	0 s	0.000 s	
<a href="#">isSingleImageDefaultPos</a>	1	0.002 s	0.001 s	
<a href="#">iscom</a>	11	0 s	0.000 s	
<a href="#">isdir</a>	2	0 s	0.000 s	
<a href="#">iskeyword</a>	2	0 s	0.000 s	
<a href="#">ismember</a>	4	0 s	0.000 s	
<a href="#">ismember&gt;ismemberBuiltinTypes</a>	4	0 s	0.000 s	
<a href="#">ismember&gt;ismemberR2012a</a>	4	0 s	0.000 s	
<a href="#">ispc</a>	42	0 s	0.000 s	

<a href="#">trackBestNetwork</a>	8	0 s	0.000 s	
<a href="#">...twork&gt;trackByValidationPerformance</a>	8	0 s	0.000 s	
<a href="#">trainNetwork</a>	1	1.178 s	0.032 s	■
<a href="#">trainNetwork&gt;@(0)warning(ws)</a>	1	0 s	0.000 s	
<a href="#">trainNetwork&gt;create@(0)warning(ws)</a>	1	0 s	0.000 s	
<a href="#">trainNetwork&gt;fcns2struct</a>	1	0 s	0.000 s	
<a href="#">trainNetwork&gt;trainNetworkInMainThread</a>	1	1.094 s	0.017 s	■
<a href="#">trainPerf</a>	15	0.065 s	0.024 s	
<a href="#">training_fcn</a>	1	0 s	0.000 s	
<a href="#">training_fcn&gt;get_info</a>	1	0 s	0.000 s	
<a href="#">trainlm</a>	10	1.332 s	0.000 s	■
<a href="#">trainlm&gt;formatNet</a>	1	0 s	0.000 s	
<a href="#">trainlm&gt;get_info</a>	1	0.154 s	0.018 s	
<a href="#">trainlm&gt;initializeTraining</a>	1	0.331 s	0.015 s	
<a href="#">trainlm&gt;train_network</a>	1	1.178 s	0.000 s	■
<a href="#">trainlm&gt;trainingIteration</a>	8	0.293 s	0.181 s	
<a href="#">trainlm&gt;updateTrainingState</a>	9	0.050 s	0.018 s	
<a href="#">transfer_fcn</a>	17	0 s	0.000 s	
<a href="#">transfer_fcn</a>	15	0.133 s	0.000 s	
<a href="#">transfer_fcn&gt;get_info</a>	1	0 s	0.000 s	
<a href="#">type</a>	4	0 s	0.000 s	
<a href="#">type</a>	4	0 s	0.000 s	
<a href="#">type</a>	10	0 s	0.000 s	
<a href="#">type</a>	15	0 s	0.000 s	
<a href="#">type</a>	1	0 s	0.000 s	
<a href="#">type</a>	2	0 s	0.000 s	
<a href="#">type</a>	4	0 s	0.000 s	
<a href="#">type</a>	7	0 s	0.000 s	
<a href="#">uigetfile</a>	1	3.994 s	0.027 s	■
<a href="#">uitools\private\allchildRootHelper</a>	2	0 s	0.000 s	
<a href="#">uitools\private\uiigetputfile helper</a>	1	3.958 s	0.066 s	■

## V. RESULTADOS OBTENIDOS

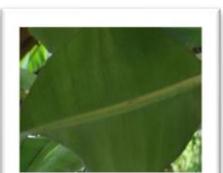
En la parte inicial de proceso de clasificación de la NN, se realizó en dos categorías: hoja sana y hoja infecta. Se entrenó la red con un total de 100 imágenes, las cuales están conformadas como se indica la tabla 11.

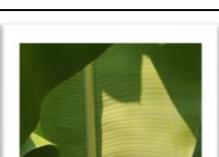
**Tabla 11:** Conjunto de imágenes primer entrenamiento

Etiqueta	Cantidad
Sanas	50
Infectadas	50

Después del primer entrenamiento de la NN, se sometieron a prueba un conjunto de 50 imágenes abarcando de manera proporcional cada uno de los diagnósticos e ingresadas en desorden específico, dando como resultado una efectividad del 100% en la predicción, tal como se representa en la tabla 12.

**Tabla 12:** Pruebas primer entrenamiento

Fotografía	Diagnóstico original	Diagnostico NN NFF
	Grado 1	Grado 1
	Grado 1	Grado 1
	Grado 0	Grado 0

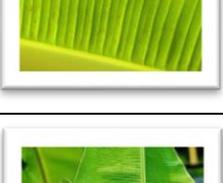
	Grado 4	Grado 4
	Grado 2	Grado 2
	Grado 3	Grado 3
	Grado 1	Grado 1
	Grado 3	Grado 3
	Grado 3	Grado 3
	Grado 0	Grado 0
	Grado 2	Grado 2

	Grado 2	Grado 2
	Grado 1	Grado 1
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 3	Grado 3
	Grado 4	Grado 4

	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 3	Grado 3
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 0	Grado 0
	Grado 4	Grado 4
	Grado 1	Grado 1

	Grado 1	Grado 1
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0

	Grado 3	Grado 3
	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 1	Grado 1
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0

	Grado 3	Grado 3
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 0	Grado 0

En el segundo entrenamiento, se incrementó la cantidad de datos de entrada a 200 imágenes, ahora modificando las clasificaciones, como se indica en la tabla 13.

**Tabla 13:** Conjunto de imágenes segundo entrenamiento

Etiqueta	Cantidad
Sanas	40
Grado 1	40
Grado 2	40
Grado 3	40
Grado 4	40

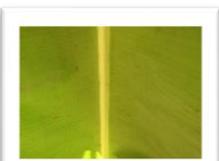
Los resultados de este segundo entrenamiento arrojaron como resultados: 10/10 imágenes de hoja sana, 6/10 imágenes grado 1, 7/10 imágenes grado 2, 10/10 imágenes de grado 3, 10/10 imágenes de grado 4. Quedando como se muestran en la tabla 14.

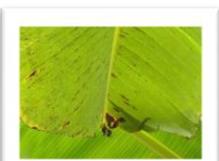
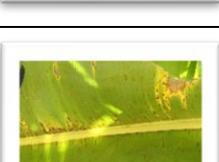
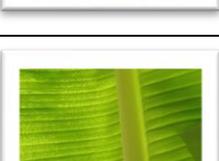
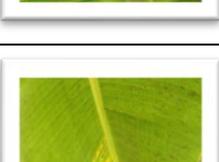
**Tabla 14:** Pruebas segundo entrenamiento

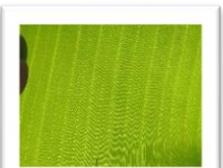
Fotografía	Diagnóstico original	Diagnostico NN NFF
	Grado 1	Grado 1
	Grado 1	Grado 1
	Grado 0	Grado 0
	Grado 4	Grado 4

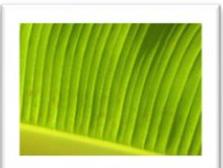
	Grado 2	Grado 1
	Grado 3	Grado 3
	Grado 1	Grado 2
	Grado 3	Grado 3
	Grado 3	Grado 3
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 2	Grado 2

	Grado 1	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 3	Grado 3
	Grado 4	Grado 4
	Grado 4	Grado 4

	Grado 1	Grado 2
	Grado 3	Grado 3
	Grado 3	Grado 3
	Grado 2	Grado 1
	Grado 0	Grado 0
	Grado 4	Grado 4
	Grado 1	Grado 2
	Grado 1	Grado 1

	Grado 0	Grado 0
	Grado 2	Grado 1
	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 3	Grado 3

	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 1	Grado 1
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 3	Grado 3

	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 0	Grado 0

La figura 39, muestra una captura del sistema en ejecución al momento de arrojar un error, diagnosticando como grado 1 a la imagen ingresada, que pertenece a grado 2 de infección.



**Figura 39:** Diagnóstico erróneo del sistema: hoja grado 2

Para el tercer y último entrenamiento de la red, se optó por ingresar una cantidad mayor de imágenes de grado 1 y grado 2, ya que, al ser los más difíciles de diagnosticar se necesitó una cantidad superior, así como ingresar imágenes con características de toma específicas para mejorar la predicción, resultando la nueva clasificación como se describe en la Tabla 15.

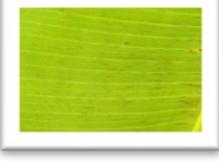
**Tabla 15:** Conjunto de imágenes tercer entrenamiento

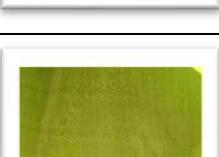
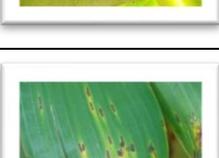
Etiqueta	Cantidad
Sanas	60
Grado 1	80
Grado 2	80
Grado 3	60

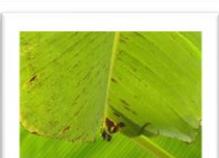
Grado 4	60
---------	----

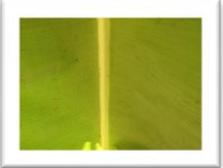
Después de esta modificación en el entrenamiento de la NN NFF, siendo la definitiva, se obtuvo una precisión en el diagnóstico del 100%, tal como se muestra en la tabla 16.

**Tabla 16:** Pruebas tercer entrenamiento

Fotografía	Diagnóstico original	Diagnostico NN NFF
	Grado 1	Grado 1
	Grado 1	Grado 1
	Grado 0	Grado 0
	Grado 4	Grado 4
	Grado 2	Grado 2
	Grado 3	Grado 3

	Grado 1	Grado 1
	Grado 3	Grado 3
	Grado 3	Grado 3
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 2	Grado 2
	Grado 1	Grado 1
	Grado 4	Grado 4

	Grado 0	Grado 0
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 3	Grado 3
	Grado 4	Grado 4
	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 3	Grado 3

	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 0	Grado 0
	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 1	Grado 1
	Grado 0	Grado 0
	Grado 2	Grado 2

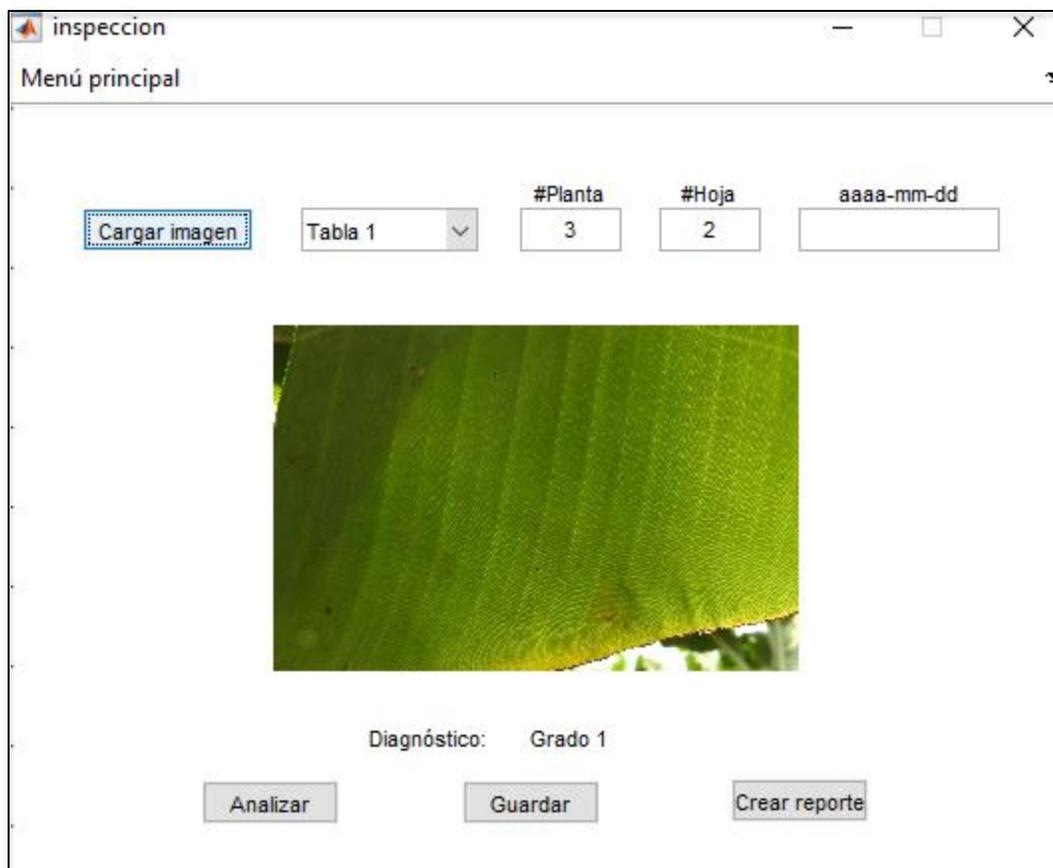
	Grado 4	Grado 4
	Grado 1	Grado 1
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 3	Grado 3
	Grado 4	Grado 4
	Grado 1	Grado 1

	Grado 1	Grado 1
	Grado 3	Grado 3
	Grado 2	Grado 2
	Grado 4	Grado 4
	Grado 0	Grado 0
	Grado 3	Grado 3
	Grado 4	Grado 4
	Grado 0	Grado 0

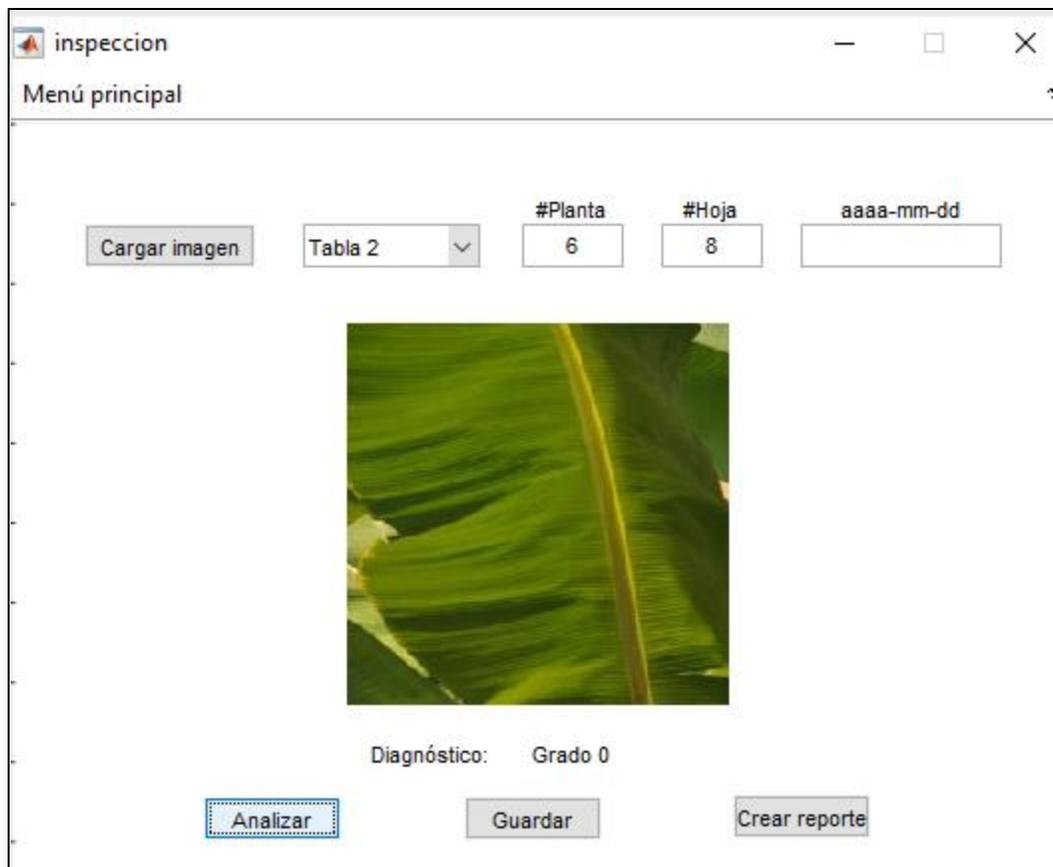
	Grado 2	Grado 2
	Grado 0	Grado 0
	Grado 2	Grado 2
	Grado 0	Grado 0

Al obtener la red completamente entrenada, es adaptada a la interfaz de inspección, donde de manera gráfica al ingresar una imagen esta se proyectará para el usuario y, al presionar el botón “analizar” la red entrará en juego para realizar la inspección y posteriormente, imprimir en la interfaz el diagnóstico obtenido, tal como se explica en el apartado 4.3.3.3 “Interfaz inspección”.

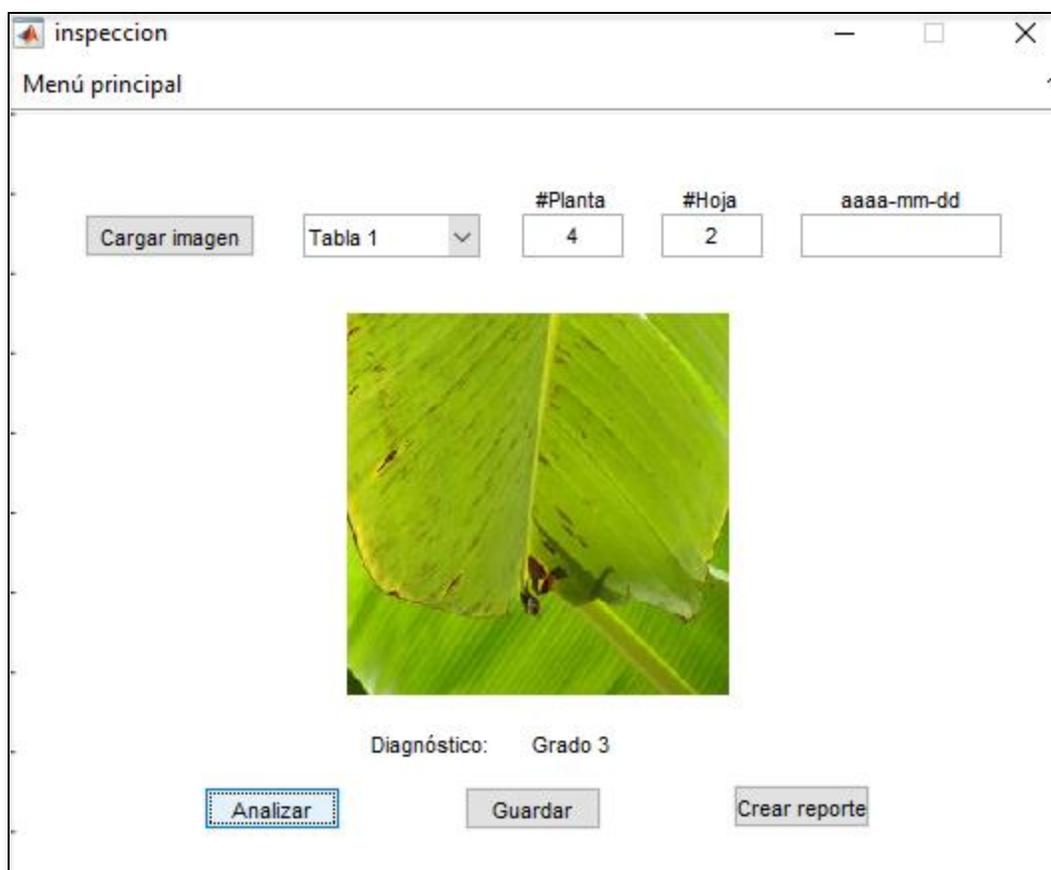
En la figura 40, 41 y 42, se muestra el correcto funcionamiento de la NN, mediante capturas del sistema en ejecución.



**Figura 40:** Diagnóstico acertado del sistema: hoja grado 1



**Figura 41:** Diagnóstico acertado del sistema: hoja grado 0



**Figura 42:** Diagnóstico acertado del sistema: hoja grado 3

Como se describe en la sección 1.3 “Revisión de la literatura”, existen otros trabajos que cumplen un propósito similar al presentado en esta investigación, cómo es el trabajo publicado por (al K. G., 2016) el cual, aunque el sistema arrojó resultados positivos y útiles dentro de su campo, este trabajo no utiliza técnicas de redes neuronales, herramienta que ayudaría a obtener una mejor y mayor predicción.

Por su parte la investigación elaborada en (Gutiérrez, 2015) presentó un sistema de adquisición y análisis de información para realizar muestreos de la Sigatoka Negra, el cual si bien demostró ser una herramienta óptima, no utilizó redes neuronales, las cuales hubieran otorgado mayor precisión en el análisis y, al tratarse de una herramienta orientada a personas sin el mínimo conocimiento de la enfermedad, si el aprendizaje por parte de esta fue incorrecto o las decisiones respecto al predio se verán afectadas.

Finalmente se encuentra el trabajo presentado por (Andrade, 2018) que realizó un sistema web para la detección de la Sigatoka Negra mediante el uso de redes neuronales convolucionales. Sin embargo, el presente trabajo se destaca por el empleo de la red neuronal newff, la disminución en los tiempos de ejecución y su sobresaliente usabilidad,

siendo de fácil comprensión para cualquier usuario sin necesidad de tener conocimientos tecnológicos previos para aprender su uso, así como brinda un proceso de instalación sencillo al mantener todos sus componentes integrados en la misma herramienta y no requerir conexión a Internet para su uso.

## VI. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo de investigación se desarrolló un sistema basado en inteligencia artificial, utilizando como técnica una red neuronal new feed-forward con la finalidad de obtener un diagnóstico oportuno de la presencia de la Sigatoka Negra en el cultivo de plátano.

Con los resultados obtenidos queda claro que el uso de redes neuronales es una de las mejores herramientas para el apoyo en el control de la proliferación de la Sigatoka Negra y cualquier otro factor determinante en la rentabilidad del cultivo de plátano y otros ámbitos, adaptándose a cualquier problema de otras áreas.

En esta tesis se logró desarrollar un sistema como herramienta tecnológica basada en redes neuronales del tipo new feed-forward, para el diagnóstico de la Sigatoka Negra en los cultivos de plátano, dando como resultado una herramienta óptima para la detección oportuna de la SN, de fácil comprensión para el usuario, tanto en su uso como instalación, por contar con todos sus componentes integrados en una sola herramienta y no requerir conexión a Internet.

Como trabajo futuro, se propone agregar las diferentes enfermedades y plagas que pueden acechar al cultivo y afectar su rentabilidad, logrando con ello una red neuronal no solamente limitada a la Sigatoka Negra. Además, se propone el uso de otros tipos de redes neuronales para encontrar una mejor precisión.

## VII.- REFERENCIAS BIBLIOGRÁFICAS

- al, K. G. (2016). *Detection and prevention of Banana leaf diseases from banana plant using Embeeded Linux board*. Coimbatore, India: 2016 Online International Conference on Green Engineering and Technologies.
- al, P. J. (s.f.). *IDENTIFICACIÓN Y CLASIFICACIÓN DE FRUTOS DE CAFÉ EN TIEMPO REAL, A TRAVÉS DE LA MEDICIÓN DE COLOR*. Cenicafé.
- al, V. K. (2013). *Estimation of plant nitrogen content using digital image*. Kharagpur, India: Indian Institute of Technology.
- Ambles, S. W. (2005). Proceso Unificado Ágil.
- Andrade, J. S. (2018). *PROCESAMIENTO DE IMÁGENES APLICADAS A LA IDENTIFICACIÓN DE SIGATOKA NEGRA EN EL CULTIVO DE PLÁTANO*. Villa de Álvarez, Colima: Instituto Tecnológico de Colima.
- Bhausahab Shivajirao Shinde, A. D. (2013). The Origins of Digital Image Processing & Application areas in Digital Image Processing Medical Images. *IOSRJEN*, 1(1), 066-071.
- Buyto. (Marzo de 2011). *¿Web o Desktop?*. Obtenido de *¿Web o Desktop?* : <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones>
- Casarin, H. (2014). *Episentido 5 Último*. lulu.com.
- CropLife Latin America. (2018). *Sigatoka Negra*. Obtenido de Sigatoka Negra (Mycosphaerella Fijiensis): <https://www.croplifela.org/es/plagas/listado-deplagas/sigatoka-negra>
- D. Michie, D. S. (1994). *Machine Learning, Neural and Statical Classification*. Ellis Horwood.
- Digital Image Processing*. (2014 de Septiembre de 29). Obtenido de University of Tartu: <https://sisu.ut.ee/imageprocessing/book/1>
- E. Escobar Avalos, N. G. (2017). Predicción de agentes patógenos en plantas ornamentales utilizando redes neuronales. *Revista Tecnologías de la Información*, 32-40.
- Elizabeth Álvarez, A. P. (2013). La Sigatoka negra en el plátano y banano. Guía para el reconocimiento y manejo de la enfermedad, aplicado a la agricultura familiar. *CIAT*.
- Elizabeth Alvarez, A. P. (2013). *La Sigatoka Negra en plátano y banano*. *Centro Internacional de Agricultura Tropical*. Obtenido de FAO: <http://www.fao.org/3/as089s/as089s.pdf>
- FAO línea agrícola. (2012). *Manejo fitosanitario del cultivo de plátano*. Obtenido de Medidas para la temporada invernal : [http://www.fao.org/fileadmin/templates/banana/documents/Docs\\_Resources\\_2015/TR4/cartilla-platano-ICA-final-BAJA.pdf](http://www.fao.org/fileadmin/templates/banana/documents/Docs_Resources_2015/TR4/cartilla-platano-ICA-final-BAJA.pdf)
- Gilberto Manzo Sanchez, M. O. (2014). Enfermedades de importancia cuarentenaria y económica del cultivo de banano (Musa sp.) en México. *Revista mexicana de fitopatología*, 89-107.
- Guillen. (2014). *La biblia de MySQL*. ANAYA.
- Gutiérrez, J. d. (2015). *SISTEMA DE ADQUISICIÓN Y ANÁLISIS DE INFORMACIÓN PARA EL DIAGNÓSTICO DE LA SIGATOKA NEGRA*. Colima, Colima: Universidad de Colima.
- Gutierrez, M. E. (2020). *Estudio comparativo de la implementación de una aplicación web o de escritorio para el control del personal de la empresa AMESEPREVE en la ciudad de Babahoyo*. Obtenido de <http://dspace.utb.edu.ec/bitstream/handle/49000/7643/DORADO%20GUTIERREZ.pdf?squence=1&isAllowed=y>
- Haykin, S. (2004). 1 Feedforward neural networks: an introduction.

Haykin, S. (2009). *Neural Networks and Learning Machines*. Ontario Canada: Prentice Hall.

Ian Goodfellow, Y. B. (2016). *Deep Learning*. MIT Press.

INEGI. (2017). *Agricultura*. Recuperado el 06 de Diciembre de 2018, de <https://www.inegi.org.mx/temas/agricultura/>

Juan Jesús Romero, C. D. (2007). *Inteligencia artificial y computación avanzada*. Fundación Alfredo Brañas.

Juan Salvador García Andrade, N. G. (2018). Procesamiento de imágenes aplicadas a la identificación de Sigatoka Negra en el cultivo de Plátano. *Instituto Tecnológico de Colima*.

Malacara, D. (1997). PROCESAMIENTO DE IMAGENES. En *ÓPTICA TRADICIONAL Y MODERNA*.

MATLAB. (2020). *What is MATLAB?* Obtenido de What is MATLAB?: <https://www.mathworks.com/discovery/what-is-matlab.html>

ORACLE. (2020). *Oracle MySQL Database Service*. Obtenido de <https://www.oracle.com/mysql/>

Paul S. Ganney, S. P. (2014). *Clinical Engineering*. Academic Press.

Pressman, R. S. (2010). *Ingeniería del software un enfoque práctico. 7ma edición*. México: Mc Graw Hill.

Ramírez, R. (2010). *Modelo de método en V*. Obtenido de Universidad Estatal de Milagro: <https://es.calameo.com/read/000356419f5d318c7eefc>

Saez. (2005). *Medidas de la complejidad del software*. Obtenido de [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)

Stuart J. Rusell, P. N. (2004). *Inteligencia artificial un enfoque moderno*. Madrid: Prentice Hall.

Víctor Campos Olguín, H. M. *MATLAB para ingenieros*. Pearson México.