



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN**

**“DISEÑO DE UN PROTOTIPO ELECTRÓNICO IDENTIFICADOR  
DE MELANOMAS Y LUNARES BENIGNOS, UTILIZANDO  
TÉCNICAS DE MACHINE LEARNING”**

**Trabajo de titulación**

**Tipo:** Dispositivo Tecnológico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN**

**AUTOR:** ROBINSON GEOVANNY CALERO INGA

**DIRECTOR:** ING. EDWIN VINICIO ALTAMIRANO SANTILLÁN

Riobamba-Ecuador

2021

**©2021, Robinson Geovanny Calero Inga**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Yo, Robinson Geovanny Calero Inga, declaro que el presente trabajo de titulación es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 10 de diciembre de 2021



---

**C.I. Robinson Geovanny Calero Inga**

**C.I. 0604140830**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN**

El tribunal del trabajo de titulación certifica que. El trabajo de titulación: Dispositivo Tecnológico “**DISEÑO DE UN PROTOTIPO ELECTRÓNICO IDENTIFICADOR DE MELANOMAS Y LUNARES BENIGNOS, UTILIZANDO TÉCNICAS DE MACHINE LEARNING**”, realizado por el señor **ROBINSON GEOVANNY CALERO INGA** ha sido minuciosamente revisado por los Miembros del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	<b>FIRMA</b>	<b>FECHA</b>
Ing. Pablo Eduardo Lozada Yáñez <b>PRESIDENTE DEL TRIBUNAL</b>	_____	21-12-10
Ing. Edwin Vinicio Altamirano Santillán <b>DIRECTOR DE TRABAJO DE TITULACIÓN</b>	_____	21-12-10
Ing. Paul Patricio Romero Riera <b>MIEMBRO DEL TRIBUNAL</b>	_____	21-12-10

## **AGRADECIMIENTO**

Agradezco de todo corazón a Dios y a todas las personas que dieron su granito de arena en todo el trayecto de mi carrera universitaria, además engrandecer todo lo que me mi madre y tía hicieron por ver a este joven triunfando y no dejándolo caer en ningún momento.

Robinson

## **DEDICATORIA**

Este trabajo de titulación va dedicado a mi padre Geovanny Calero (+) que desde cualquier lugar estará feliz de ver a su hijo escalando académicamente, a mi madre Mariana Inga que en todo momento me acompañó dándome su apoyo y de manera muy especial a mi Tía Anita Inga que fue coautora directa de este logro en mi vida profesional.

Robinson

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS .....	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE ANEXOS .....	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
INTRODUCCIÓN .....	1

### CAPITULO I

<b>1. MARCO TEORICO REFERENCIAL.....</b>	<b>4</b>
<b>1.1 Inteligencia Artificial – Inicios en el campo médico.....</b>	<b>4</b>
<b>1.1.1 Principios de la Inteligencia Artificial.....</b>	<b>4</b>
<b>1.2 Unidades de inteligencia o Agentes Inteligentes .....</b>	<b>6</b>
<b>1.2.1 Definiciones de un agente inteligente. ....</b>	<b>6</b>
<b>1.2.2 Entornos y Características de un Agente Inteligente .....</b>	<b>7</b>
<b>1.3 Técnicas Computacionales.....</b>	<b>8</b>
<b>1.3.1 Lógica Difusa.....</b>	<b>9</b>
<b>1.3.1.1 Conjunto Difusos. ....</b>	<b>9</b>
<b>1.3.1.2 Teoría de Lógica Difusa. ....</b>	<b>11</b>
<b>1.3.1.3 Controlador Difuso. - .....</b>	<b>11</b>
<b>1.3.1.4 Redes Neuronales Artificiales.....</b>	<b>12</b>
<b>1.4 Machine Learning .....</b>	<b>15</b>
<b>1.4.1 Aprendiendo con datos. ....</b>	<b>17</b>
<b>1.5 Tensorflow.....</b>	<b>18</b>
<b>1.6 Cáncer de piel .....</b>	<b>18</b>
<b>1.6.1 Causas .....</b>	<b>19</b>
<b>1.6.2 Melanoma .....</b>	<b>19</b>
<b>1.6.2.1 Señales de advertencia del melanoma.....</b>	<b>20</b>

### CAPITULO II

<b>2. PROPUESTA Y DISEÑO DE PROTOTIPO .....</b>	<b>23</b>
---	-----------

2.1	Requerimientos para el diseño del prototipo del sistema.....	23
2.2	Concepción de la Arquitectura General del Sistema.....	23
2.3	Fase 2, Procesamiento y entrenamiento de la imagen.....	25
2.4	Algoritmo KNN.....	27
2.5	Requerimientos para el Software.....	28
2.5.1	<i>Entornos de desarrollo de programación Python</i> .....	28
2.5.2	<i>Instalación de paquetes frameworks</i> .....	28
2.5.2.1	<i>Numpy</i> .....	28
2.5.2.2	<i>Matplotlib.pyplot</i> .....	29
2.6	Adquisición del conjunto de datos (Dataset).....	29
2.7	Arquitectura del proyecto.....	31
2.8	Preparación del sistema de control.....	32
2.8.1	<i>Preparación del entorno</i> .....	32
2.9	Implementación Algoritmo de Entrenamiento.....	33
2.10	Proceso de Ejecución.....	37
2.10.1	<i>App entorno visual para usuarios finales.</i> .....	39
2.11	Desarrollo prototipo hardware.....	40
2.11.1	<i>Esquemático</i> .....	40
2.11.2	<i>Desarrollo de la estructura del prototipo CAD Case</i> .....	41

### CAPITULO III

3.	VALIDACIÓN DE PROTOTIPO.....	44
3.1	Implementación del sistema.....	44
3.2	Evaluación Variable Lunares.....	46
3.3	Evaluación Variable Melanomas.....	49
3.4	Evaluación Variable Queratosis Seborreica.....	52
3.5	Análisis del consumo energético.....	54
3.5.1	<i>Análisis de temperatura</i> .....	55
3.5.2	<i>Análisis consumo corriente.</i> .....	57

### CAPITULO IV

4.	EVALUACIÓN ECONÓMICA.....	59
4.1	Costos de implementación.....	59
4.1.1	<i>Costos directos</i> .....	59
4.1.1.1	<i>Costos implementación eléctrica</i> .....	60

<b>4.1.1.2</b>	<i>Costos implementación estructural</i> .....	60
<b>4.1.2</b>	<i>Costos directos totales</i> .....	60
<b>4.1.3</b>	<i>Costos Indirectos</i> .....	61
<b>4.1.4</b>	<i>Costo Total</i> .....	61
	<b>CONCLUSIONES</b> .....	62
	<b>RECOMENDACIONES</b> .....	63
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-3:</b>	Tabla de evaluación de la clasificación de imágenes Lunares. ....	46
<b>Tabla 2-3:</b>	Resumen de la evaluación lunares. ....	48
<b>Tabla 3-3:</b>	Tabla de evaluación de la clasificación de imágenes Melanomas. ....	50
<b>Tabla 4-3:</b>	Resumen de la evaluación melanomas.....	51
<b>Tabla 5-3:</b>	Tabla de evaluación de la clasificación de imágenes Queratosis Seborreica.....	53
<b>Tabla 6-3:</b>	Resumen de la evaluación Queratosis Seborreica.....	54
<b>Tabla 7-3:</b>	Análisis de Temperatura. ....	56
<b>Tabla 8-3:</b>	Análisis consumo corriente según obtenidos en las pruebas. ....	57
<b>Tabla 1-4:</b>	Costos de materiales y dispositivos .....	60
<b>Tabla 2-4:</b>	Valores protección y estructura.....	60
<b>Tabla 3-4:</b>	Costos directos totales.....	60
<b>Tabla 4-4:</b>	Costos Indirectos.....	61
<b>Tabla 5-4:</b>	Costos Totales.....	61

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b>	Ilustración Agente inteligente concepto de Russel and Norvid.....	7
<b>Figura 2-1:</b>	Ilustración Fuzzy sets theory.....	9
<b>Figura 3-1:</b>	a) Ajuste de conjunto difuso, Números alrededor del 5. b) conjuntos clásicos, del mismo enunciado.....	10
<b>Figura 4-1:</b>	Proceso para implementar un controlador difuso.....	11
<b>Figura 5-1:</b>	Representación de una red neuronal artificial conectada multicapa.....	12
<b>Figura 6-1:</b>	Representación de una Neurona Artificial.....	13
<b>Figura 7-1:</b>	Representación de una Neurona Artificial.....	14
<b>Figura 8-1:</b>	Ecuación representativa de una RNA.....	15
<b>Figura 9-1:</b>	Técnicas de Machine learning.....	16
<b>Figura 10-1:</b>	Clasificación de Imágenes.....	17
<b>Figura 11-1:</b>	TensorFlow.....	18
<b>Figura 12-1:</b>	Ilustración Cáncer de piel Epidermis y Dermis.....	19
<b>Figura 13-1:</b>	Causas Cáncer de piel.....	19
<b>Figura 14-1:</b>	Causas Cáncer de piel.....	20
<b>Figura 15-1:</b>	A asimetría melanoma.....	21
<b>Figura 16-1:</b>	B bordes melanoma.....	21
<b>Figura 17-1:</b>	C color melanoma.....	21
<b>Figura 18-1:</b>	D Diámetro melanoma.....	22
<b>Figura 19-1:</b>	E evolucionar, antes izquierda y después derecha.....	22
<b>Figura 2-2:</b>	Proceso para el desarrollo del proyecto.....	24
<b>Figura 3-2:</b>	Recursos para la creación del modelo de Aprendizaje Automático.....	25
<b>Figura 4-2:</b>	Recursos para la creación del modelo de Aprendizaje Automático.....	26
<b>Figura 5-2:</b>	Arquitectura del proyecto.....	26
<b>Figura 6-2:</b>	Captura del proceso para la instalación del paquete numpy.....	28
<b>Figura 7-2:</b>	Captura del proceso de la instalación del paquete matplotlib.....	29
<b>Figura 8-2:</b>	Plataforma Kaggle.....	30
<b>Figura 9-2:</b>	Plataforma Kaggle.....	30
<b>Figura 10-2:</b>	Plataforma Kaggle Imagenología.....	31
<b>Figura 11-2:</b>	Arquitectura funcionamiento del proyecto.....	31
<b>Figura 12-2:</b>	Creación Entorno o ambiente de programación TesisRCalero.....	32
<b>Figura 13-2:</b>	Configuración de parámetros.....	34
<b>Figura 14-2:</b>	Configuración de parámetros.....	34
<b>Figura 15-2:</b>	Selección del DataSet.....	34

<b>Figura 16-2:</b>	Entrenamiento y validación.....	35
<b>Figura 17-2:</b>	Creación del modelo VGG16.....	35
<b>Figura 18-2:</b>	Guardar el modelo con extensión .h5.....	36
<b>Figura 19-2:</b>	Graficas.....	36
<b>Figura 20-2:</b>	Evaluación etapa entrenamiento.....	37
<b>Figura 21-2:</b>	Appweb entorno interfaz gráfica.....	39
<b>Figura 22-2:</b>	Appweb entorno interfaz gráfica.....	39
<b>Figura 23-2:</b>	GPIO Raspberry pi 4.....	40
<b>Figura 24-2:</b>	Esquemático.....	40
<b>Figura 25-2:</b>	CAD base del case.....	41
<b>Figura 26-2:</b>	CAD top del case.....	41
<b>Figura 27-2:</b>	CAD case completo.....	42
<b>Figura 28-2:</b>	CAD Raspberry pi 4.....	42
<b>Figura 29-2:</b>	CAD Prototipo.....	43
<b>Figura 30-2:</b>	Draw Prototipo.....	43
<b>Figura 31-2:</b>	Ilustración del dispositivo electrónico fase prototipado.....	43
<b>Figura 1-3:</b>	Interfase para evaluación y carpetas contenedoras de información.....	45
<b>Figura 2-3:</b>	Variables.....	45
<b>Figura 3-3:</b>	Lunares 393 imágenes.....	46
<b>Figura 4-3:</b>	Imagen de ejemplo Lunar.....	48
<b>Figura 5-3:</b>	Predicción Lunar en la plataforma.....	49
<b>Figura 6-3:</b>	Imagen de ejemplo Melanoma.....	51
<b>Figura 7-3:</b>	Predicción Melanoma en la plataforma.....	52
<b>Figura 8-3:</b>	Medición tiempo real Temperatura de operación Raspberry pi 5 minutos.....	55
<b>Figura 9-3:</b>	Medición tiempo real Temperatura de operación Raspberry pi 10 minutos.....	56

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-2:</b>	Proceso para el desarrollo del proyecto.....	23
<b>Gráfico 2-2:</b>	Diagrama de proceso del funcionamiento del algoritmo knn.....	27
<b>Gráfico 3-2:</b>	Diagrama de proceso Entrenamiento para la generación del modelo de aprendizaje.....	33
<b>Gráfico 4-2:</b>	Diagrama de procesos para la ejecución de la clasificación. ....	38
<b>Gráfico 1-3:</b>	Temperatura operación vs Tiempo.....	57
<b>Gráfico 2-3:</b>	Temperatura operación vs Tiempo.....	58

## **ÍNDICE DE ANEXOS**

**ANEXO A:** GRAFICAS RESULTADO DEL ENTRENAMIENTO

**ANEXO B:** PROCESO DE ENTRENAMIENTO 50 ÉPOCAS.

**ANEXO C:** PRUEBA EN TIEMPO REAL

## RESUMEN

En el presente trabajo se evidencia la capacidad de la Inteligencia Artificial y las Redes Neuronales en cuanto a la clasificación e identificación de patrones utilizados en el área de imagenología médica. En este estudio se investigó el beneficio potencial de combinar el conocimiento humano basado en evidencia científica y la recopilación de una base de datos específica, con el objetivo de reducir la brecha del tiempo en lo que corresponde al diagnóstico clínico y la emisión de un resultado de dicho análisis, aportando eficiencia y eficacia para que los tratamientos sean exitosos. La metodología que se utilizó se fundamenta en la recolección de 2750 imágenes correspondientes a la etapa del entrenamiento con 1372 imágenes de lunares, 474 imágenes de melanomas y 254 imágenes de queratosis seborreica, en la etapa de prueba y validación se usó 471 lunares, 147 casos de melanomas y 132 casos de queratosis seborreica. Este conjunto de datos se introdujo en las entradas de las redes neuronales como parte de la técnica más novedosa del “Machine Learning”. En cuanto a la tarea multiclase y el aprendizaje automático supervisado, se obtuvo una precisión del 90,66% (errores 16% Lunares, 4% Melanomas, 8% Queratosis Seborreica, con una media de error del 9,3%). Este desequilibrio varía de acuerdo al número de épocas utilizadas para el entrenamiento, en este caso fueron necesarias 43 horas de entrenamiento para la obtención del modelo idea, considerando que el éxito en la clasificación es directamente proporcional a la calidad de la data set y el número de casos para el aprendizaje

**Palabras claves:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERIA> <INTELIGENCIA ARTIFICIAL> <IMAGENOLOGÍA>, <CANCER DE PIEL> <MELANOMA> <LUNARES> <QUERATOSIS SEBORREICA>

**LUIS  
ALBERTO  
CAMINOS  
VARGAS**

Firmado digitalmente por LUIS  
ALBERTO CAMINOS VARGAS  
Nombre de reconocimiento  
(DN): c=EC, l=RIOBAMBA,  
serialNumber=0602766974,  
cn=LUIS ALBERTO CAMINOS  
VARGAS  
Fecha: 2021.04.14 13:14:57  
-05'00'



1005-DBRAI-UTP-2021

## ABSTRACT

The present research study examines the capacity of Artificial Intelligence and Neural Networks to classify and identify patterns used in the area of medical imaging. The potential benefit of combining human knowledge based on scientific evidence and the compilation of a specific database was investigated. The goal was to reduce the time gap in relation to clinical diagnosis and the results of said analysis being issued, which in turn would strengthen the efficiency and likely effectiveness of treatment. The methodology that was used is based on the collection of 2750 images corresponding to the training stage with 1372 images of moles, 474 images of melanomas and 254 images of seborrheic keratosis. In the test and validation stage, 471 cases of moles were used, 147 cases of melanomas and 132 cases of seborrheic keratosis. This set of data was entered into the inputs of neural networks as part of the newest “Machine Learning” technique. Regarding the multiclass task and supervised machine learning, a precision of 90,66% was obtained (16% Mole errors, 4% Melanoma errors, 8% Seborrheic Keratosis errors, with a mean error of 9,3%). This imbalance varies according to the number of stages used for training, in this case 43 hours of training were necessary to obtain the ideal model. It is important to take into consideration that correct classification is directly proportional to the quality of the data set and the number of cases inputted for learning.

**Keywords:** <ENGINEERING TECHNOLOGY AND SCIENCE>, <ARTIFICIAL INTELLIGENCE>, <IMAGENOLOGY>, <SKIN CANCER>, <MELANOMA>, <MOLES>, <SEBORRHEIC KERATOSES>

LENIN  
IVAN LARA  
OLIVO

Firmado digitalmente por  
LENIN IVAN LARA  
OLIVO  
Fecha: 2021.04.21  
13:02:02 -05'00'

## INTRODUCCIÓN

El cáncer de piel es la afectación más común dentro de las enfermedades que provocan el desorden o mutación de las células en el cuerpo humano. Los posibles factores de riesgo incluyen exposición a la luz solar o radiación ultravioleta (p. Ej., Camas de bronceado), infección viral (p. Ej., VPH), tipo de piel clara, presencia de una gran cantidad o ciertos tipos de nevos (lunares), exposición al arsénico, inmunosupresión, ciertas enfermedades (p. ej., xeroderma pigmentoso) y antecedentes familiares de cáncer de piel (Farmer y Naylor 1996).

El cáncer de piel es una de las problemáticas en que el paciente puede llegar a tener el control del diagnóstico preventivo con las herramientas tecnológicas necesarias. En este caso de estudio el beneficio que brinda la inteligencia artificial y los algoritmos de machine learning para la ejecución de la clasificación de imágenes que se puede montar en un dispositivo móvil, de esta manera todo el conocimiento y experiencia de un experto se recopila en un algoritmo que emitirá un análisis del melanoma realizado en tiempo real para descartar o confirmar la afectación (Farmer y Naylor 1996).

Según la Skin Cancer Foundation, la incidencia del cáncer de piel sigue aumentando de manera exponencial en la última década («Skin Cancer Facts & Statistics», s. f.). Durante el periodo 2011 y 2013 en Ecuador, se estimaron 2138 casos identificados como melanomas y cáncer de piel, cifra que se concentra mayormente hasta la actualidad en las ciudades de Quito, Loja y Cuenca. (Acosta, s. f.).

La incidencia más alta en contraer esta afección son los hombres que viven en la capital con un 35,8%, mientras que las mujeres son del 31,2% (Acosta, s. f.)

El cáncer de piel puede aparecer en forma maligna o benigna. El melanoma benigno es simplemente aparición de lunares en la piel, por otro lado, el Melanoma maligno es la aparición de llagas que provocan sangrado, esta es una alerta preocupante en el paciente, esto se debe al surgimiento canceroso en una lesión cutánea pigmentada. El melanoma maligno es el nombre de la célula de la que se presumiblemente surge, el melanocito. (Jaleel et al., s. f.). Si se identifica esta afectación de manera oportuna puede ser tratable de una manera exitosa hasta una cura definitiva. («Skin Cancer Facts & Statistics», s. f.)

El melanoma puede extenderse a todas las partes del cuerpo a través del sistema linfático o la sangre, es por esta razón que la identificación temprana es vital para precautelar la salud del

paciente. El problema principal del diagnóstico tradicional y toma de muestras ambulatorias, causa la inflamación del área intervenida e incluso estimula la propagación de la lesión. Por lo tanto, es necesario crear otros métodos de diagnóstico breve, menos agresivo y eficaz. (Gilchrest et al., 1999)

## **FORMULACIÓN DEL PROBLEMA**

*¿Es posible diagnosticar, identificar y clasificar imágenes de Melanomas y Lunares Benignos, utilizando técnicas de Machine Learning mediante un dispositivo tecnológico?*

### **Sistematización Del Problema**

- ¿Se evidencia en el mercado la oferta de un dispositivo o servicio que identifique en tiempo real melanomas y lunares benignos?
- ¿Qué necesidad establece el sistema de inteligencia artificial, tanto como software y hardware para su implementación y puesta en marcha?
- ¿Qué se debe considerar, para ejecutar la tarea de clasificación de imágenes utilizando la técnica de redes neuronales convolucionales mediante un entrenamiento supervisado?
- ¿Cuál será la metodología de evaluación del sistema para medir su rendimiento en identificar una afectación de la piel mediante una fotografía como muestra de un diagnóstico clínico?

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Diseñar un Dispositivo Inteligente Identificador de Melanomas y Lunares Benignos, utilizando técnicas de Machine Learning.

### **OBJETIVOS ESPECIFICOS**

- Estudiar las diferentes técnicas de inteligencia artificial y el comportamiento de cada una para medir el rendimiento de clasificación de imágenes en tiempo real.
- Analizar grupos de conjunto de datos en bibliografía médica para el desarrollo del algoritmo de aprendizaje.
- Análisis de diseño del hardware según los requerimientos y exigencias planteadas para la óptima ejecución en el diagnóstico y clasificación de melanomas.

- Realizar las pruebas y resultados de la implementación del prototipo para verificar el cumplimiento de los requerimientos planteados

# CAPITULO I

## 1. MARCO TEORICO REFERENCIAL

En el primer capítulo se detallará los avances del Soft computing y la tendencia de la inteligencia artificial idónea para la clasificación de imágenes de aplicación dermatológica.

### 1.1 Inteligencia Artificial – Inicios en el campo médico

La inteligencia artificial es una parte de la informática que intenta hacer que la computadora sea más inteligente. Uno de los requisitos básicos para cualquier comportamiento inteligente es el aprendizaje. La mayoría de las investigaciones actuales coinciden en que no hay inteligencia sin aprendizaje. Por lo tanto, el aprendizaje automático es una de las principales ramas de la inteligencia artificial y, en general, es uno de los subcampos de la investigación de la IA que se desarrolla con mayor rapidez.(Kononenko 2001)

Los algoritmos de aprendizaje automático se diseñaron y utilizaron desde el principio para analizar conjuntos de datos médicos. Hoy en día, el aprendizaje automático proporciona varias herramientas indispensables para el análisis inteligente de datos. Especialmente en los últimos años, la revolución digital proporcionó medios relativamente económicos y disponibles para recopilar y almacenar los datos. Los hospitales modernos están bien equipados con monitoreo y otros dispositivos de supervisión, y los datos se recopilan y comparten en grandes sistemas de información. La tecnología de aprendizaje automático es actualmente muy adecuada para analizar datos médicos y, en particular, hay mucho trabajo realizado en el diagnóstico médico en pequeños problemas de diagnóstico especializados.(Kononenko 2001)

#### 1.1.1 *Principios de la Inteligencia Artificial*

La inteligencia artificial se empezó a utilizarlo, tan pronto surgieron las computadoras que se utilizaron para cálculos digitales entre las décadas de 1950 y 1960, mediante la utilización de este medio se desarrollaron algoritmos que permitieron modelar y analizar grandes conjuntos de datos.

Desde el principio, surgieron las principales ramas del aprendizaje automático como se detalla a continuación:

El trabajo clásico en el aprendizaje simbólico fue descrito por Hunt (Zelič et al. 1997), donde se describen las técnicas de aprendizaje automático que se pueden utilizar para extraer el conocimiento de los datos almacenados en base a datos previamente medidos, en esta investigación se plantearon la utilización de varios algoritmos para predecir lesiones deportivas. La metodología incluye variantes del algoritmo asistente para la inducción descendente de árboles de decisión y de un clasificador bayesiano. (Zelič et al. 1997)

Según la bibliografía descrita por Nils J. Nilsson desde la Universidad de Standord en el año de 1965, por primera vez establece un concepto interesante acerca del proceso de aprendizaje de una máquina y las relaciones ocultas con la materia prima de un aprendizaje o la base del conocimiento que son los datos, estos mantienen un flujo constante que bombardea nuestros sentidos y millones de canales sensoriales que transportan esta data hasta nuestro cerebro, haciendo esta analogía, (Nilsson 1965), establece que los cerebros también son máquinas de aprendizaje que condicionen, combinan analizan y almacenan datos. Con esta comparativa Nilsson plantea la siguiente hipótesis ¿Es posible aprender algo sobre el aprendizaje observando el estilo de computación que utilizan un cerebro? (Nilsson 1965)

En la rama de las Redes Neuronales, Frank Rosenblatt autor del artículo Principles of Neurodynamics en 1986, establece que el modelo básico de la unidad del procesamiento de una red neuronal es el perceptrón de tres capas, cada capa es un conjunto de elementos de un umbral simple o neuronas. Las capas S, A y R están acopladas en serie por una conexión sináptica  $S \rightarrow A \rightarrow R$ . Si S es un sensor de superficie (ejemplo la retina del ojo humano), la capa A contiene “Unidades Asociativas” o células detectoras, y finalmente, R contiene R-células detectoras (Células de reconocimiento). Presentando un patrón  $S_1$  en S. Después de un retraso sináptico, un patrón  $A_1$  en A está activo, después de otro retraso, se dispara un conjunto de células R. Una R-célula debe activarse precisamente cuando el patrón proyectado en S es de cierto tipo (por ejemplo, cuando se visualiza una imagen de un triángulo). Cada unidad A recibe conexiones de un subconjunto específico de las celdas de S y se dispara por un patrón específico en este soporte. El propósito de la capa A es reducir la superposición entre patrones que deben clasificarse de manera diferente. Los pesos de las conexiones de A a una célula R se establecen por plasticidad sináptica. Cuando se le ha presentado un patrón S, un “maestro” externo decide para que célula R según la respuesta correcta emitida. (Van Der Malsburg 1986).

A lo largo de los años dedicados en la investigación de las ramas anteriormente mencionadas, se logran desarrollar métodos importantes y avanzados, los cuales son:

- K-nearest neighbors.
- Análisis Discriminante.
- Clasificadores Bayesianos.
- Aprendizaje inductivo de reglas simbólicas.
- Árbol de decisiones.
- Redes Neuronales, convolucionales, backpropagation, multicapas.

## **1.2 Unidades de inteligencia o Agentes Inteligentes**

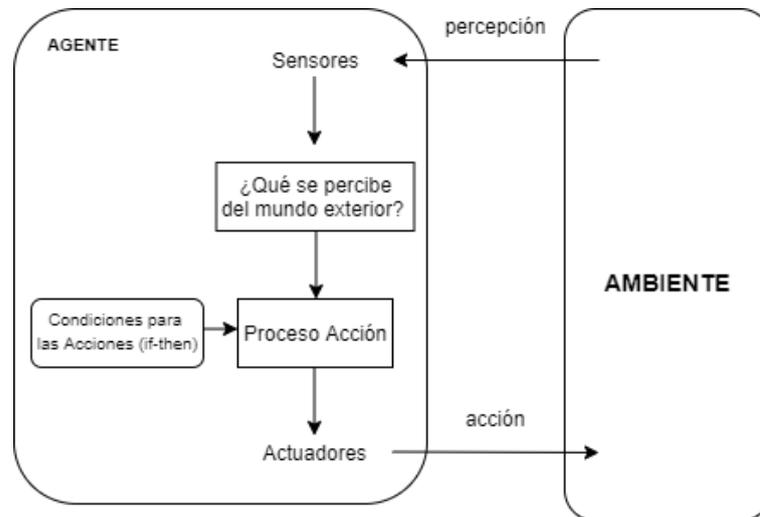
Durante los últimos treinta años, la inteligencia artificial y los sistemas de agentes estuvieron estrechamente relacionados. La Inteligencia Artificial despertó el interés de estudiar a los componentes de la inteligencia (por ejemplo, la habilidad para aprender, para ejecutar un plan o un algoritmo) mientras que el estudio de los agentes, trata de integrar los mismos componentes al desarrollo de una aplicación a una problemática. Esta distinción puede parecer implicar que todos los problemas que se enfrenta la IA es el desarrollo de agentes que se encarguen de tareas en particulares a un universo de soluciones. (Rudowsky 2004). Pero en realidad este no es el caso, Etzioni en su artículo publicado en 1996, describe que “Los agentes inteligentes componen el 99% de la informática y la solución de problemas automáticos, y un 1% es de inteligencia artificial”(Etzioni 1996). Si bien se pueden utilizar técnicas de inteligencia artificial para construir agentes, un agente no requiere todas las capacidades de la inteligencia artificial. Por lo tanto, no es necesario resolver todos los problemas de IA antes de construir un agente. Por ejemplo, la capacidad de aprender puede no ser un rasgo deseable para un agente en algunas situaciones, aunque ciertamente es un componente de la IA. (Rudowsky 2004).

### **1.2.1 Definiciones de un agente inteligente.**

Russel and Norving en su libro Inteligencia Artificial, un enfoque moderno, establecen textualmente que un agente inteligente es “Una entidad que puede percibir el medio ambiente o el entorno mediante sensores e interactuar en el utilizando actuadores” (Russell et al. 2011) Este criterio es el más fundamental y didáctico que ayuda a una comprensión directa de la tarea que realiza un agente inteligente.

Otro criterio fundamental que apoya al esquema de la fundamentación de un agente inteligente es “Un agente inteligente es un sistema informático basado en hardware y/o software que muestra

las propiedades de la anatomía, destreza social, reactividad y proactividad” (Wooldridge y Jennings 1995).



**Figura 1-1:** Ilustración Agente inteligente concepto de Russel and Norvid  
Realizado por: Calero, R, 2021, 2020

Un consenso entre diferentes criterios, establecen que la autonomía, la capacidad de actuar sin intervención humana u otros sistemas, es una característica clave de un agente, en la figura 1-1, se ilustra como un agente recibe información de su entorno, a través de un repertorio de acciones a su disposición, el agente reaccionará según sus actuadores al entorno. Generalmente, en dominios de complejidad razonable, un agente no controlará su entorno por completo. Por lo tanto, la misma acción realizada dos veces en situaciones aparentemente idénticas podría dar como resultados completamente diferentes. La falla también es una posibilidad, es decir, la acción tomada por el agente puede no producir el efecto deseado en su totalidad. (Rudowsky 2004).

### 1.2.2 Entornos y Características de un Agente Inteligente

La decisión crítica que enfrenta un agente es determinar qué acción realizar para satisfacer mejor los objetivos que se plantearon en su diseño. Los entornos de los agentes se clasifican según la funcionalidad de diferentes propiedades que pueden afectar la complejidad del proceso de la toma de decisiones de este. (Russell et al. 2011)

- **Determinista vs No determinista,** lo más razonable es que los sistemas complejos no sean deterministas. El estado que resultará de una acción no está garantizado incluso cuando el sistema se encuentra en un estado similar antes de que se aplique la acción. Esta incertidumbre presenta un desafío mayor para el diseñador de agentes de sistemas deterministas.

- **Episódico frente a No episódico**, en un entorno episódico, las acciones del agente dependen de una serie de episodios discretos sin vínculos entre el desempeño del agente en diferentes escenarios. Este entorno es más sencillo de diseñar debido a la falta de razonamiento sobre las interacciones entre episodios anteriores y futuros, solo es necesario considerar el entorno actual.
- **Accesible frente a Inaccesible**, un entorno accesible es aquel en el que el agente puede obtener información completa, oportuna y precisa sobre el estado del medio ambiente. Cuanto más accesible es un entorno, menos complicado es crear agentes para que operen en él. La mayoría de los entornos de complejidad moderada son inaccesibles.
- **Estática vs Dinámico**, los entornos estáticos permanecen sin cambios a excepción de los resultados producidos por las acciones del agente. Otros procesos operan en un entorno dinámico, cambiando así el entorno fuera del control del agente. Obviamente, un entorno dinámico requiere un diseño de agente más complejo.

### 1.3 Técnicas Computacionales.

El Soft computing es una colección emergente de metodologías que tienen como objetivo explotar la tolerancia a la imprecisión, la incertidumbre y la verdad parcial para lograr robustez, manejabilidad y bajo costo total. La diferencia de la computación convencional o el hard computing, se basa fuertemente en la intuición o la subjetividad. Por lo tanto, el soft computing, brinda una oportunidad atractiva para representar la ambigüedad en el pensamiento humano con la incertidumbre de la vida real. La lógica difusa, redes neuronales, y algoritmos genéticos son las metodologías centrales de esta técnica computacional. (Ovaska, VanLandingham y Kamiya 2002)

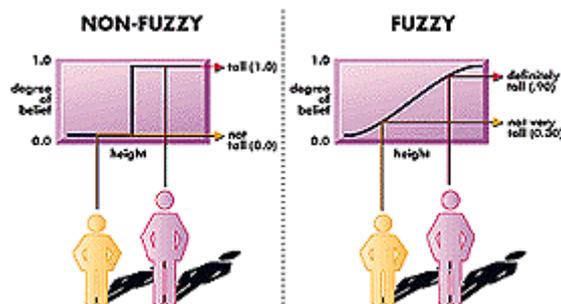
Las técnicas de hard computing suelen ser más sencillas de analizar, su comportamiento y estabilidad son más predecibles, y, la carga computacional de los algoritmos es típicamente baja o moderada. Estas características son particularmente importantes en aplicaciones en tiempo real. Por lo tanto, es natural ver a estas dos técnicas como metodologías potencialmente complementarias. Se necesitan combinaciones según su caso de estudio. (Ovaska, VanLandingham y Kamiya 2002)

### 1.3.1 Lógica Difusa.

#### 1.3.1.1 Conjunto Difusos.

El concepto difuso en las teorías de argumentación matemática fueron propuestos por primera vez por Zadeh en su artículo publicado en 1965 que lleva el mismo nombre (Zadeh 1965), desde la visión de Zadeh, varios investigadores han contribuido de manera importante al desarrollo de la teoría de los conjuntos difusos y sus aplicaciones, resultando en un gran éxito desde el punto de vista teórico y tecnológico, sobre todo en el campo de la inteligencia artificial y la creación de controladores inteligentes.(Massad et al. 2008).

La teoría de conjuntos difusos propone tratar con límites poco claros, representando conceptos ambiguos que establecen variables lingüísticas. En este sentido, los conjuntos difusos surgieron como una forma alternativa transformar lo subjetivo a un modelo objetivo de medida.(Massad et al. 2008). La idea básica de un conjunto difuso se refiere a la flexibilidad sobre el concepto de pertenencia. En la teoría de los conjuntos certeros, la operatividad es capaz de clasificar objetos en conjuntos a través de un proceso binario (se aceptan o rechazan según el criterio de pertenencia), en efecto, un gran número de situaciones es posible decir si el elemento  $x$  pertenece o no al conjunto  $A$  ( $x \in A$  ó  $x \notin A$ ), por ejemplo, en la siguiente afirmación se puede decir que 10 pertenece al conjunto de números naturales y que -10 no pertenece a este conjunto. Sin embargo, ¿qué decir sobre el hecho de que el número 4 pertenece al grupo de números definidos como "alrededor de 5"?, claramente, la respuesta en este caso dependerá del contexto. La principal diferencia entre conjuntos clásicos y difusos es que en el primero existe fragmentación de conceptos que necesariamente debe conservarse. (Massad et al. 2008)

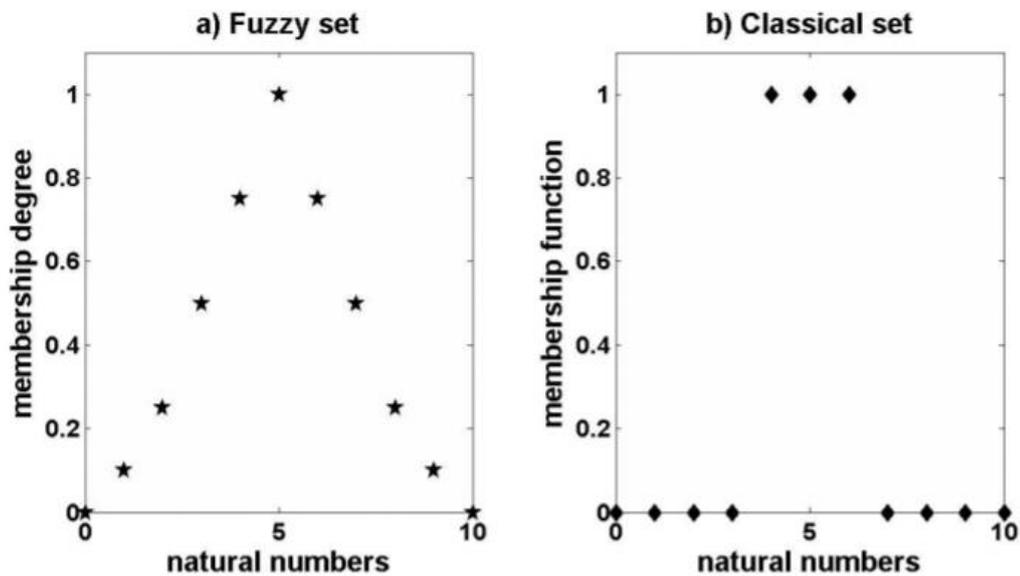


**Figura 2-1:** Ilustración Fuzzy sets theory  
fuente: <http://mbasystemsintegration.blogspot.com/2014/10/fuzzy-logic-introduction.html>

Muchas variables del entorno real presentan límites marcados como, por ejemplo, la edad de una persona, el sexo, estado civil, etc. Dentro de este entorno real, se visualizan variables en las que

la relación de pertenencia no está bien definida, por ejemplo, clasificación de personas delgadas, clasificación de buenos estudiantes, clasificación de alta temperatura, etc. En estos casos no es muy fácil decir si el elemento pertenece o no al un conjunto dado si antes establecer una nueva variable lingüística para cada situación.

Para resolver esta incertidumbre, Zedeh propuso una metodología, que estable un grado de pertenencia según pertenezca el elemento a la clasificación o no, esto prácticamente habla en niveles de parcialidad encontrada en el elemento a clasificar.



**Figura 3-1:** a) Ajuste de conjunto difuso, Números alrededor del 5. b) conjuntos clásicos, del mismo enunciado.  
fuente: (Massad et al. 2008)

a) Definición 1. Sea  $U$  un conjunto clásico no vacío y  $A$  un subconjunto de  $U$ , la función característica de  $A$  está dada por:

$$A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases}$$

En la gráfica 1-3 b, se visualiza que los números que están junto a 5 se marca de una manera binaria, 1 si  $x$  pertenece o no al conjunto de los datos y 0 si no se cumple con este enunciado.

b) Definición 2. Sea  $U$  un conjunto clásico no vacío.  $A$  un subconjunto difuso  $F$  y se describe mediante la siguiente función:

$$F: U \rightarrow [0,1]$$

F, es la función de pertenencia del conjunto difuso.

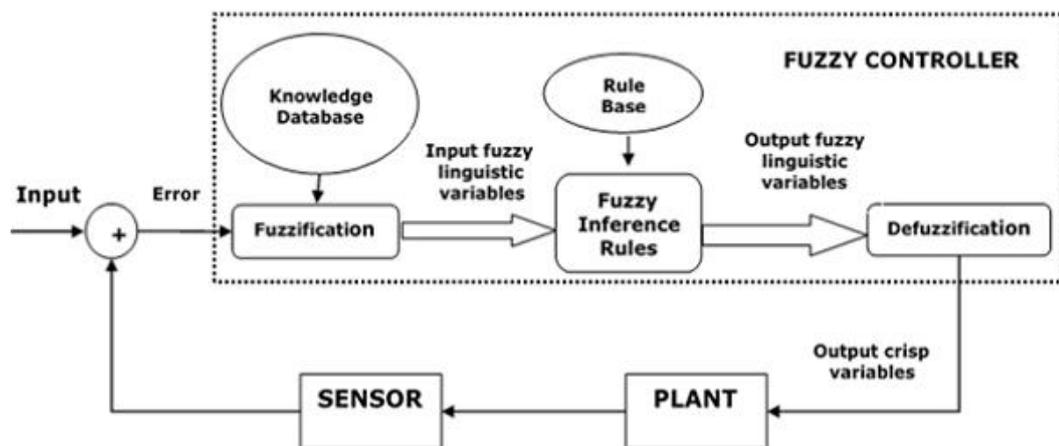
Considerando el enunciado anterior “Alrededor de 5” en el universo de números naturales discutido anteriormente. En algunos contextos se podría proponer los siguientes grados de membresía:

$$F(0) = 0; F(1) = 0.10; F(2) = 0.25; F(3) = 0.50; F(4) = 0.75; F(5) = 1; F(6) = 0.75; F(6) = 0.50; F(7) = 0.50; F(8) = 0.25; F(9) = 0.10$$

### 1.3.1.2 Teoría de Lógica Difusa.

Los sistemas de lógica difusa basan sus decisiones en entradas en forma de variables lingüísticas derivadas de funciones de pertenencia que son fórmulas que se utilizan para determinar el conjunto difuso al que pertenece un valor y el grado de pertenencia a ese conjunto. Luego las variables se emparejan con las condiciones previas de las reglas lingüísticas IF-THEN, y la respuesta de cada regla se obtiene a través de la implicación difusa. Para realizar la regla de inferencia de composición, la respuesta de cada regla se pondera de acuerdo con la confianza o el grado de pertenencia de sus entradas, y se calcula el centroide de las respuestas para generar la salida adecuada. (Elmahi 2015).

### 1.3.1.3 Controlador Difuso. -



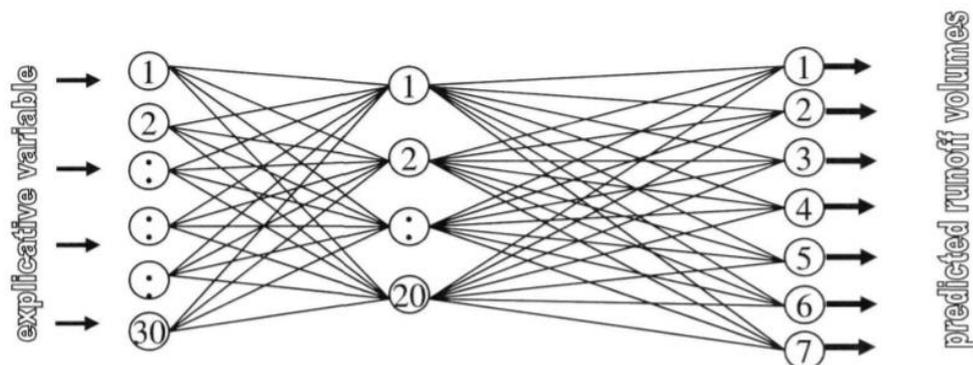
**Figura 4-1:** Proceso para implementar un controlador difuso.  
Fuente: [https://www.tutorialspoint.com/fuzzy\\_logic/fuzzy\\_logic\\_control\\_system.htm](https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_control_system.htm)

Antes de diseñar un controlador difuso se debe tener en consideración las etapas propuestas en la figura 1-4 de la propuesta Elmahi (Elmahi 2015)

1. **Difusor**, La función del difusor es convertir los valores de entrada en valores difusos.
2. **Base del conocimiento difuso**, almacena el conocimiento sobre todas las relaciones difusas de entrada y salida, crea una función de pertenencia que define las variables de entrada a la base de las reglas difusas y las variables de salida a la planta.
3. **Base de reglas difusas**, almacena el conocimiento sobre el funcionamiento del proceso de dominio.
4. **Motor de inferencia**, Actúa como un núcleo de cualquier controlador difuso, básicamente simula decisiones humanas realizando un razonamiento aproximado.
5. **Fusificación**, esta función convierte los valores difusos en valores nítidos que se obtienen del motor de inferencia difuso.

#### 1.3.1.4 Redes Neuronales Artificiales.

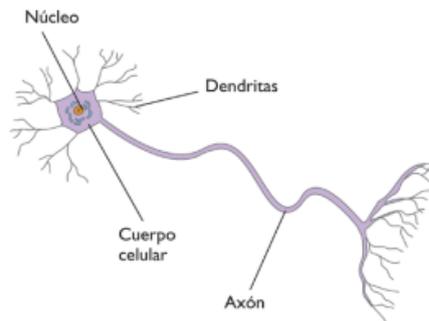
Las RNAs pertenecen a un modelo matemático inspirado en las conexiones sinápticas de las neuronas de un cerebro figura 1-6. Consisten en una serie de neuronas dispuestas en diferentes capas, una capa de entrada, una capa de salida y una o mas capas ocultas que pertenecen al procesamiento interno. (Dolling y Varas 2002)



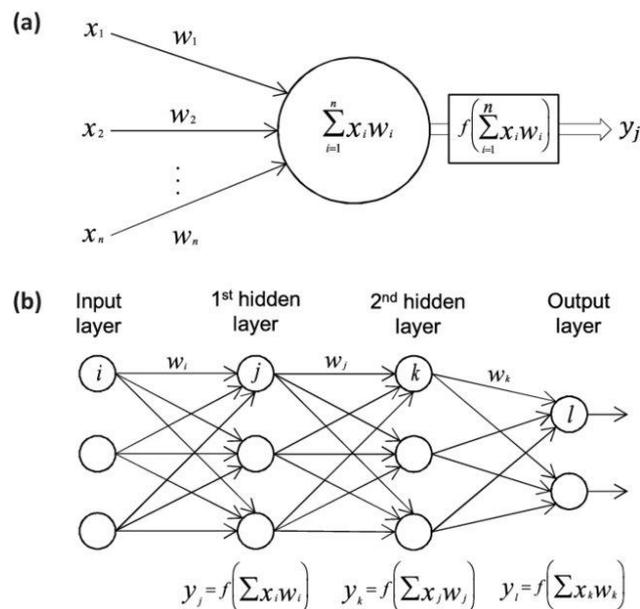
**Figura 5-1:** Representación de una red neuronal artificial conectada multicapa.  
Fuente: (Dolling y Varas 2002)

Las neuronas de entrada reciben y procesan las señales de entrada para posterior enviar una señal de salida a otras neuronas de la red. Cada neurona se puede conectar a las otras neuronas y tienen una función de activación según el umbral y grado de pertenencia, que pueden ser funciones continuas, lineales o no lineales.

La señal que pasa a través de una neurona se transforma mediante pesos que modifican las funciones, por lo tanto, Al modificar los pesos de todas las neuronas de la red, cambia la salida. Una vez definida la arquitectura de la red, se calculan los pesos para representar el resultado deseado a través de un proceso de aprendizaje donde se entrena a la RNA para obtener los resultados esperados. La información disponible se utiliza para definir un conjunto de datos de aprendizaje o entrenamiento y un conjunto de datos de validación (Dolling y Varas 2002).



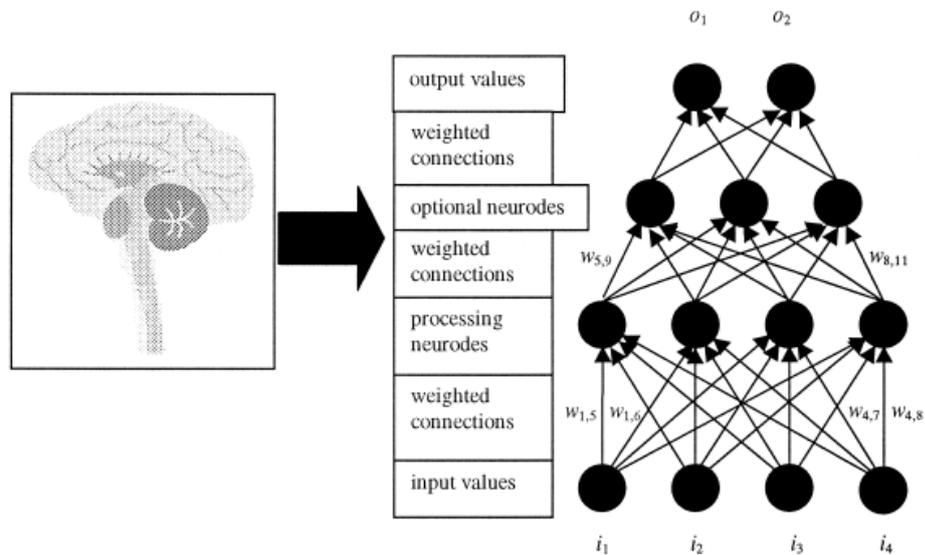
**Figura 1-6:** Representación de una Neurona Biológica.  
Realizado por: Calero, R, 2021.



**Figura 6-1:** Representación de una Neurona Artificial.  
Fuente:(Vieira, Pinaya y Mechelli 2017).

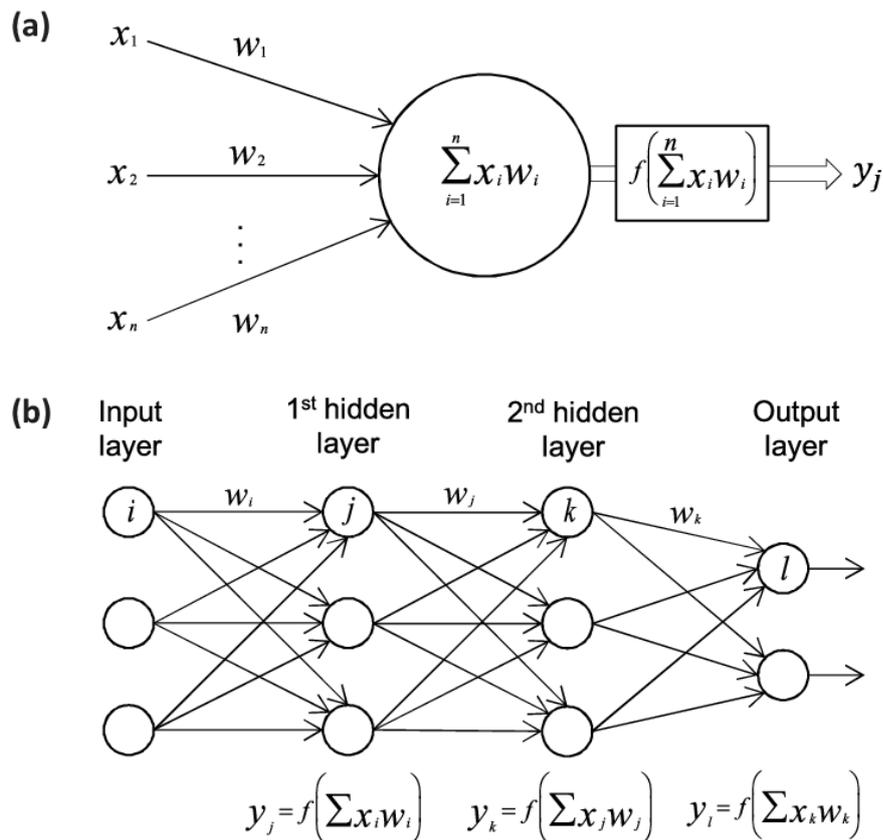
Las redes neuronales artificiales son una tecnología basada en estudios del cerebro y el sistema nervioso como se muestra en la figura 23-1. Los modelos ANN simulan la actividad eléctrica del cerebro y el sistema nervioso. Procesando elementos (también conocidos perceptrón) están conectados a otros elementos de procesamiento. Normalmente los perceptrones están dispuestos en una capa o vector, con la salida de una capa que sirve como entrada a la siguiente y posiblemente otras capas. Se puede conectar un perceptrón a uno o un subconjunto de los

perceptrones en la capa subsiguiente, con estas conexiones simulan las conexiones sinápticas del cerebro. Señales de datos ponderadas que ingresan a un perceptrón simulando la excitación eléctrica de una célula nerviosa y, en consecuencia, la transferencia de información dentro de la red o el cerebro. (Walczak y Cerpa 2003)



**Figura 7-1:** Representación de una Neurona Artificial.  
Fuente: (Walczak y Cerpa 2003)

Los valores de entrada a un elemento de procesamiento  $i_n$ , se multiplican por un peso de conexión  $w_{n,m}$  que simula el fortalecimiento de las vías neurales del cerebro. Es a través del ajuste de las fuerzas de conexión o pesos que el aprendizaje emula las RNA.



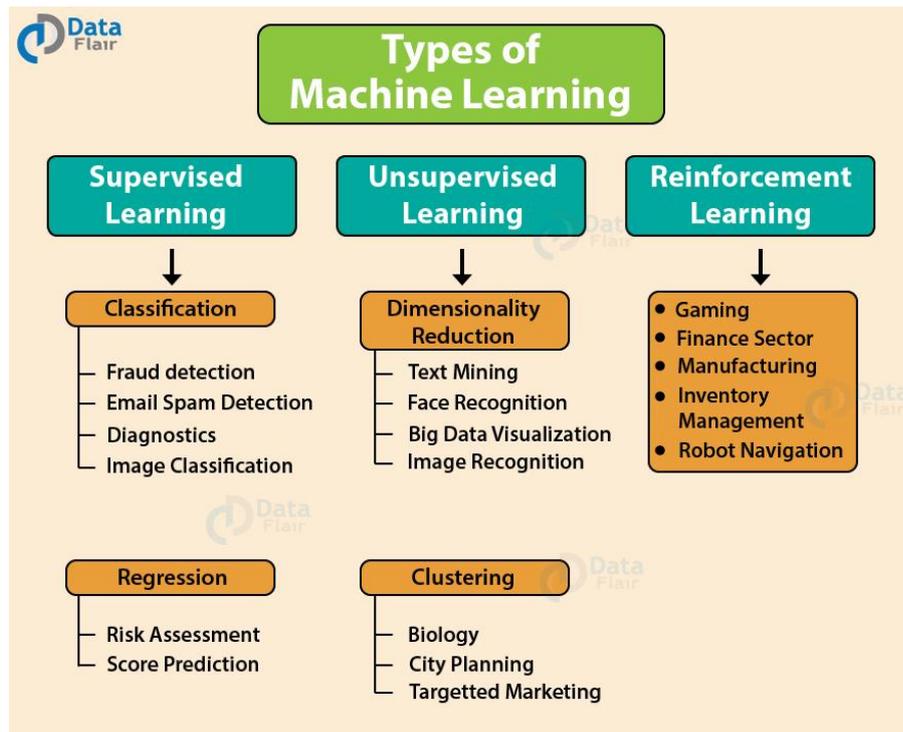
**Figura 8-1:** Ecuación representativa de una RNA.  
Realizado por: Calero, R, 2021.

## 1.4 Machine Learning

El aprendizaje automático es una rama en evolución de los algoritmos computacionales que están diseñados para emular la inteligencia humana aprendiendo del entorno circundante. Se les considera el caballo de batalla en la nueva era del big data. Las técnicas basadas en el aprendizaje automático se han aplicado con éxito en diversos campos que van desde el reconocimiento de patrones, la visión por computadora, la ingeniería de naves espaciales, las finanzas, el entretenimiento y la biología computacional hasta las aplicaciones biomédicas y médicas. (El Naqa y Murphy 2015)

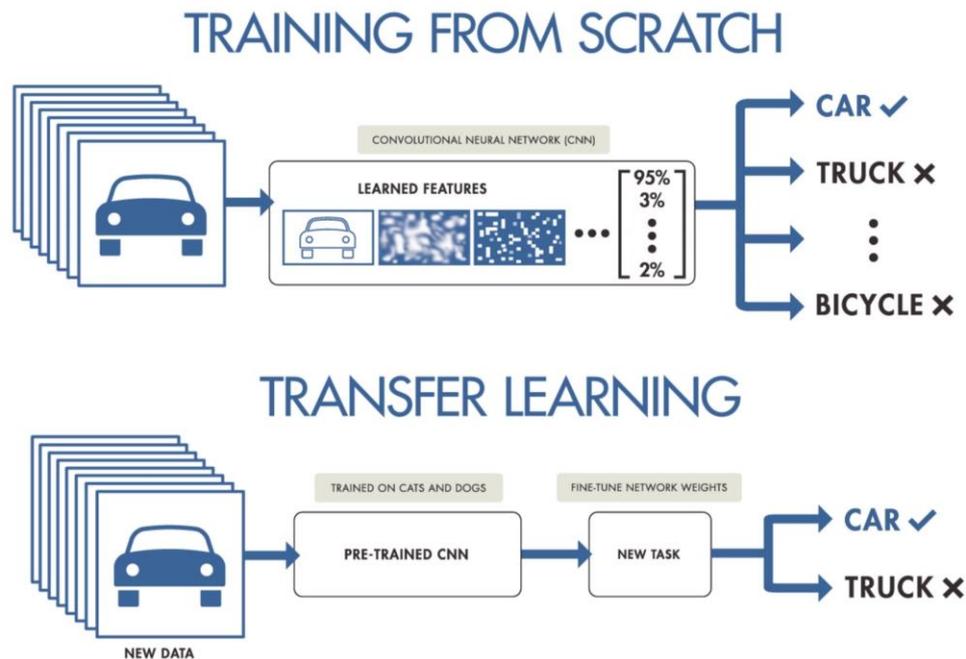
Más de la mitad de los pacientes con cáncer reciben radiación ionizante (radioterapia) como parte de su tratamiento, y es la principal modalidad de tratamiento en estadios avanzados de la enfermedad local. La radioterapia implica un gran conjunto de procesos que no solo abarcan el período desde la consulta hasta el tratamiento, sino que también se extienden más allá para garantizar que los pacientes hayan recibido la dosis de radiación recetada y estén respondiendo bien. Los grados de complejidad de estos procesos pueden variar y pueden involucrar varias etapas de interacciones sofisticadas hombre-máquina y toma de decisiones, lo que naturalmente invitaría al uso de algoritmos de aprendizaje automático para optimizar y automatizar estos

procesos, incluido, entre otros, el control de calidad de la física de la radiación, contorneado y planificación del tratamiento, radioterapia guiada por imágenes, manejo del movimiento respiratorio, modelado de respuesta al tratamiento y predicción de resultados. La capacidad de los algoritmos de aprendizaje automático para aprender del contexto actual y generalizarse en tareas invisibles permitiría mejoras tanto en la seguridad como en la eficacia de la práctica de la radioterapia que conduzcan a mejores resultados. (El Naqa y Murphy 2015)



**Figura 9-1:** Técnicas de Machine learning  
Fuente: <https://data-flair.training/blogs/machine-learning-tutorial/>

### 1.4.1 Aprendiendo con datos.



**Figura 10-1:** Clasificación de Imágenes

Fuente: <https://raminnabati.com/2020/06/adv.-pytorch-modifying-the-last-layer/>

La capacidad de aprender a través de la información del entorno circundante ya sea jugando a las damas o al ajedrez, reconociendo patrones escritos o resolviendo los abrumadores problemas de la oncología de radiación, es la clave principal para desarrollar una aplicación de aprendizaje automático exitosa. El aprendizaje se define en este contexto como la estimación de las dependencias a partir de los datos (El Naqa y Murphy 2015) los campos de la minería de datos y el aprendizaje automático están entrelazados.

La minería de datos utiliza algoritmos de aprendizaje automático para interrogar grandes bases de datos y descubrir conocimientos ocultos en los datos, mientras que muchos algoritmos de aprendizaje automático emplean métodos de minería de datos para preprocesar los datos antes de aprender las tareas deseadas. Sin embargo, debe tenerse en cuenta que el aprendizaje automático no se limita a resolver problemas similares a las bases de datos, sino que también se extiende a la solución de inteligencia artificial compleja. (El Naqa y Murphy 2015)

## 1.5 Tensorflow



# TensorFlow

**Figura 11-1:** TensorFlow  
Fuente: <https://tensorflow.com>

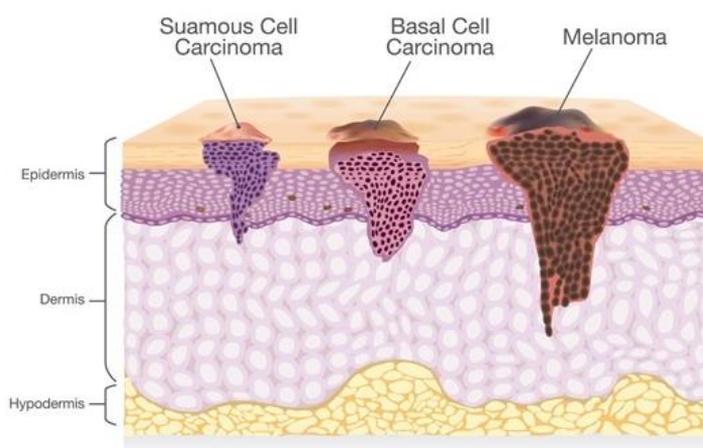
Framework de código abierto muy poderoso para implementar modelos de aprendizaje automático a gran escala. Esto lo hace perfecto para investigación y producción por la versatilidad de implementación, esta librería se puede ejecutar dentro del ambiente de desarrollo de Python, C y JavaScript. («TensorFlow» 2021).

Con esta librería se puede entrenar y ejecutar redes neuronales convolucionales profundas para clasificación de patrones, esto es comúnmente utilizado en identificar patrones dentro de imágenes. («Guía inicial de TensorFlow 2.0 para principiantes | TensorFlow Core» 2021).

## 1.6 Cáncer de piel

El cáncer de piel es el crecimiento descontrolado de células anormales en la epidermis figura 28-1, causado por daños no reparados en el ADN que desencadenan mutaciones. Estas mutaciones hacen que las células de la piel se multipliquen rápidamente y formen tumores malignos. (Bath-Hextall et al. 2007)

## Type of skin cancer

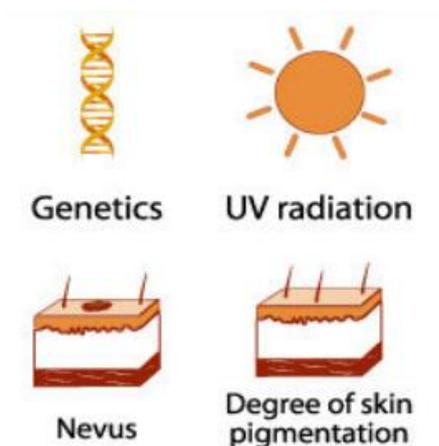


**Figura 12-1:** Ilustración Cáncer de piel Epidermis y Dermis

Fuente: <https://cccnevada.com/skin-cancers-you-may-not-know-about/>

### 1.6.1 Causas

En la actualidad, según la evidencia científica, explica que existen 2 causas principales de cáncer de piel las cuales están directamente relacionadas a la exposición de los rayos solares y el uso UV de camas de bronceado. (Urbach 1993) La buena noticia es que si el cáncer de piel se lo detecta temprano (Etapa precancerosa), el profesional en dermatología puede tratarlo con poca intervención clínica, eliminándolo por completo. (McKean y Amaria 2017).



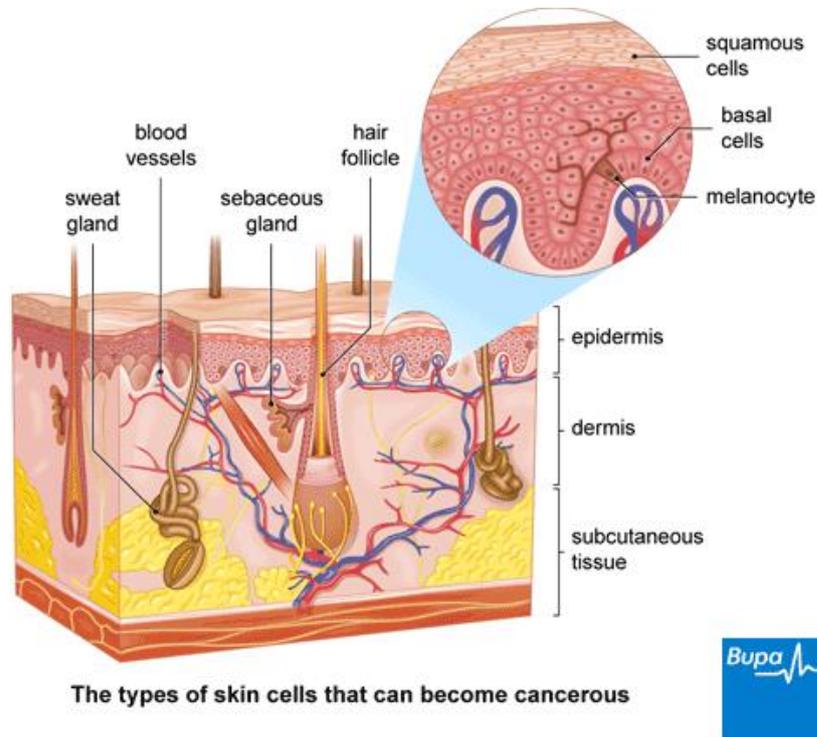
**Figura 13-1:** Causas Cáncer de piel

Fuente: («Skin Cancer Causes & Risk Factors | Reasons & Effects» [2020])

### 1.6.2 Melanoma

El melanoma es un cáncer que se desarrolla a partir de los melanocitos, las células de la piel que producen pigmento de melanina, encargada del color de la piel. (Casariego y Baudo 2004).

Los melanomas a menudo se parecen a los lunares y, a veces pueden surgir de ellos. Se pueden manifestar en cualquier área del cuerpo, incluso en áreas que normalmente no están expuestas al sol. (Radhi 1999). Normalmente se desencadena por el tipo de exposición solar intensa e intermitente que provoca quemaduras solares. El uso de cámaras de bronceado es un factor muy agresivo que favorece a la proliferación de esta manifestación de cáncer. («Skin Cancer Information» [2018]).



**Figura 14-1:** Causas Cáncer de piel

Fuente: <https://www.bupa.co.uk/health-information/cancer/melanoma>

#### 1.6.2.1 Señales de advertencia del melanoma

Encontrar el melanoma en una etapa temprana es crucial, la detección temprana puede aumentar enormemente las posibilidades de un tratamiento exitoso.

La mayoría de los lunares, manchas marrones y crecimientos en la piel son inofensivos, pero no siempre. Existe una metodología que permite diagnosticar de manera clínica que es el ABCDE (las primeras 5 letras del alfabeto) esto es una guía para entender las advertencias del melanoma. (Yarbro et al. 1999)

**A es la asimetría:** La mayoría de los melanomas son asimétricos. Si se dibuja una línea en el medio de la lesión, las dos mitades no coinciden, en comparación a un lunar que es común su forma redonda ovalado y simétrico.



**Figura 15-1:** A asimetría melanoma.

Fuente: <https://www.bupa.co.uk/health-information/cancer/melanoma>

**B es Bordes:** Los bordes del melanoma tienden a ser desiguales y pueden tener bordes festoneados o con muescas, mientras que los lunares comunes tienden a tener bordes más suaves y uniformes.



**Figura 16-1:** B bordes melanoma.

Fuente: <https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/>

**C es el Color:** Los colores múltiples son una señal de advertencia. Si bien los lunares benignos suelen tener un solo tono de marrón, un melanoma puede tener diferentes tonos, negro, café, rojo y blanco.



**Figura 17-1:** C color melanoma.

Fuente: <https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/>

**D es diámetro:** Si bien es ideal para detectar un melanoma cuando es pequeño, es una señal de advertencia si una lesión es del tamaño (6mm o ¼ de pulgada de diámetro). («Skin Cancer Information» [2018])



**Figura 18-1:** D Diámetro melanoma.

Fuente: <https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/>

**E es para evolucionar:** Cualquier cambio en el tamaño, forma, color o elevación de una mancha en la piel, o cualquier síntoma nuevo en ella, como sangrado, picazón o costras, puede ser señal de advertencia de melanoma.



**Figura 19-1:** E evolucionar, antes izquierda y después derecha.

Fuente: <https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/>

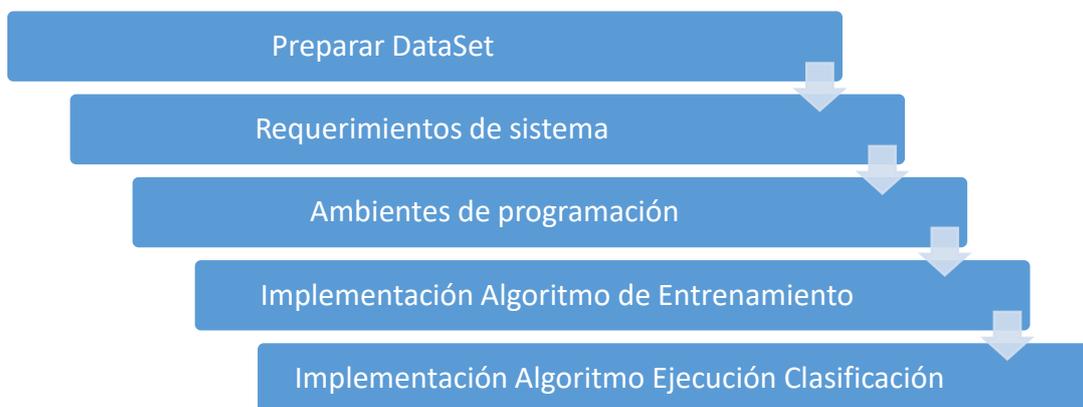
## CAPITULO II

### 2. PROPUESTA Y DISEÑO DE PROTOTIPO

En el presente capítulo se detalla los procedimientos que se realizaron en varias etapas para el diseño y construcción del prototipo, considerando una selección adecuada del hardware a utilizar y el software para la implementación del sistema clasificador.

#### 2.1 Requerimientos para el diseño del prototipo del sistema.

Mediante un previo estudio realizado en el primer capítulo se puede definir algunos de los requerimientos que demanda un sistema inteligente basado en Machine Learning, y también nos permitió aclarar la técnica adecuada de Soft Computing para la clasificación e identificación de patrones en imágenes que son las Redes Neuronales. En el grafico 1-2 se muestran los procesos que se necesitan para el desarrollo del prototipo.



**Gráfico 1-2:** Proceso para el desarrollo del proyecto.  
Realizado por: Calero, R, 2021.

#### 2.2 Concepción de la Arquitectura General del Sistema.

En la figura 1-2 se plasma la arquitectura general del prototipo, mediante el uso de tecnologías de hardware libre se desarrolló el algoritmo de inteligencia artificial para una posterior evaluación y clasificación de imágenes. En esta etapa se captura la imagen del sector donde se encuentra el

melanoma a evaluarse. La fotografía es capturada por la tarjeta raspberry pi y cámara para ser interpretada por el algoritmo que se ejecuta en la misma tarjeta.



**Figura 20-2:** Proceso para el desarrollo del proyecto.  
Realizado por: Calero, R, 2021.

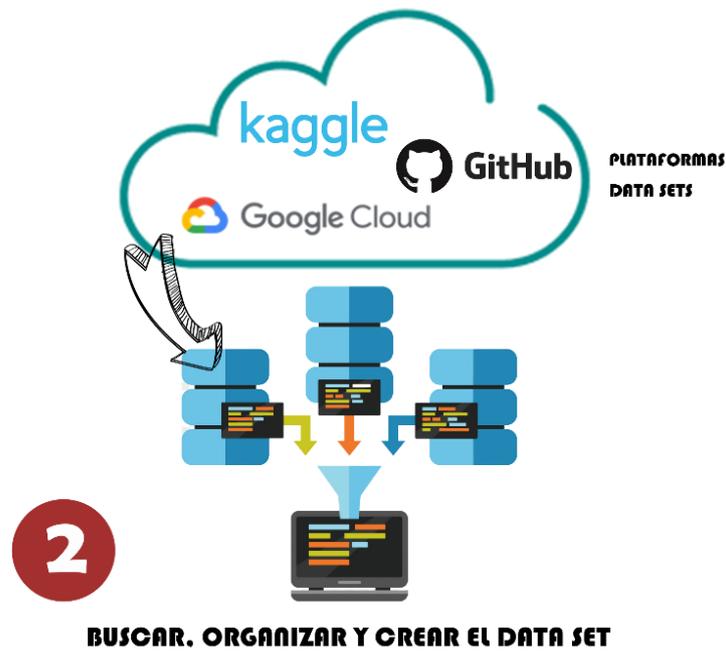
### 2.3 Fase 2, Procesamiento y entrenamiento de la imagen.



**Figura 21-2:** Recursos para la creación del modelo de Aprendizaje Automático.  
Realizado por: Calero, R, 2021.

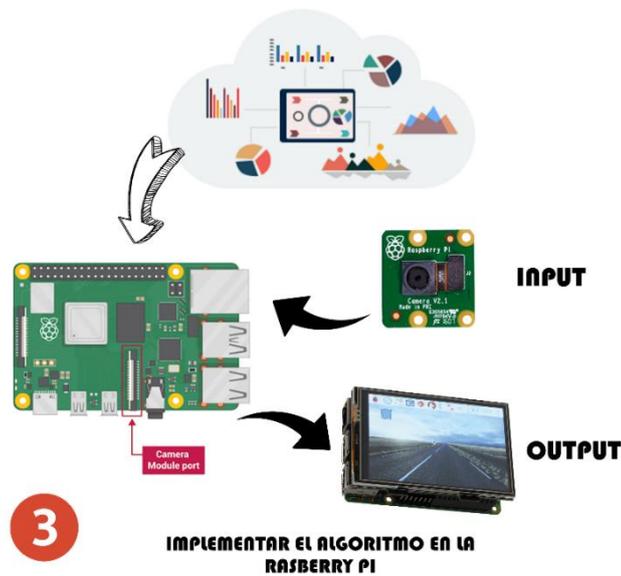
La figura 2-2 representa el funcionamiento general del prototipo, el cual consta de la elaboración de un dispositivo electrónico utilizando tarjetas de desarrollo embebido Raspberry Pi y una cámara para capturar imágenes de la piel en un paciente. El dispositivo tendrá la capacidad de identificar, clasificar y supervisar cada imagen capturada que es procesado píxel a píxel mediante el algoritmo de redes neuronales convolucionales.

En la figura 3-2 se esquematiza los recursos necesarios para la primera parte del prototipo, la creación de un modelo de aprendizaje automático, para ejecutar esta primera etapa es necesario contar con el ambiente de desarrollo basado en Python que es un lenguaje de programación de alto nivel más utilizado en el desarrollo de aplicaciones de inteligencia artificial, se utilizaron librerías de tensorflow y keras que permiten coordinar el algoritmo y tratar a las imágenes del mundo exterior en matrices numéricas para la posterior operación en las redes neuronales.



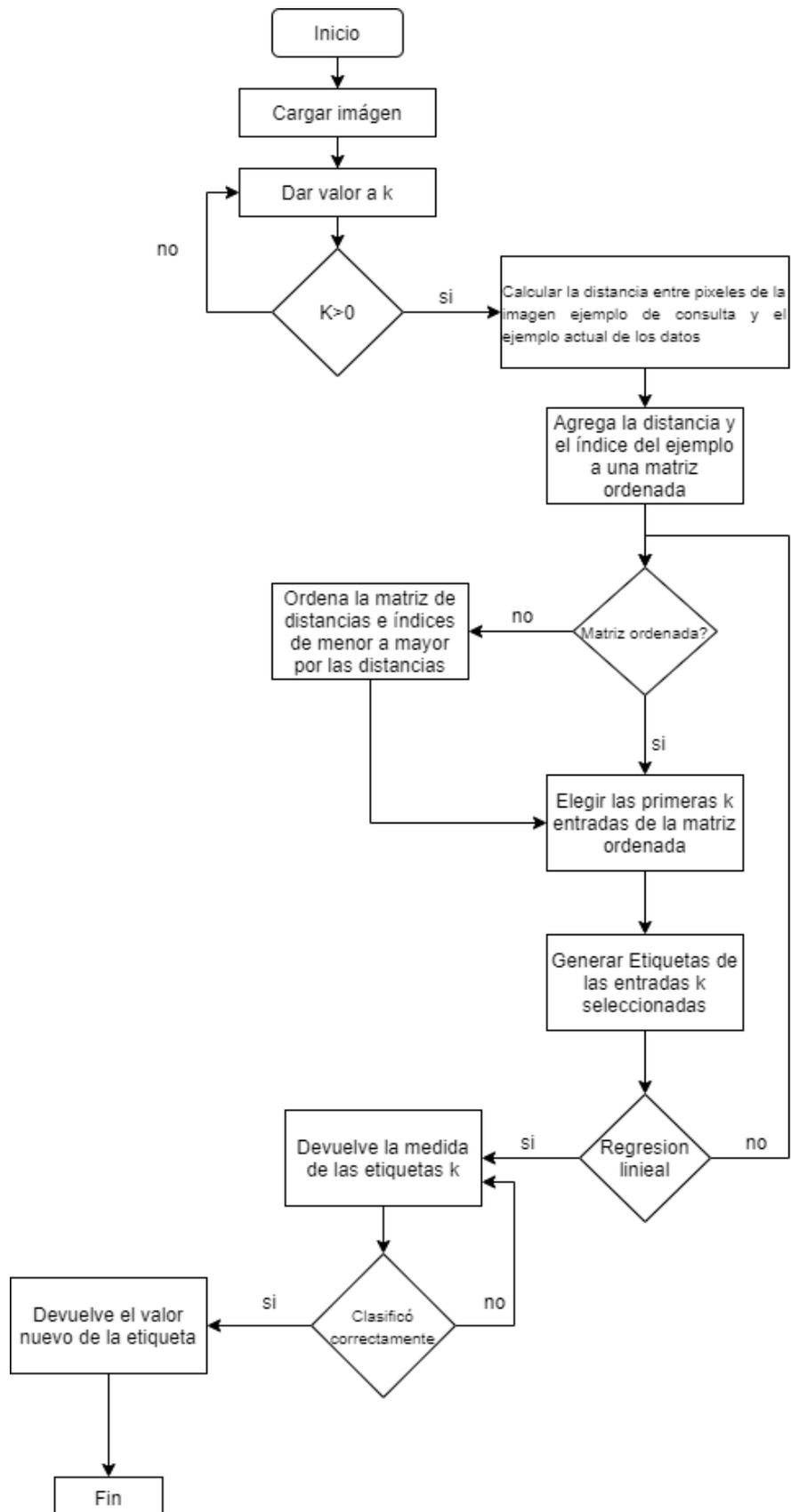
**Figura 22-2:** Recursos para la creación del modelo de Aprendizaje Automático.  
Realizado por: Calero, R, 2021.

Previo a la generación de un modelo de aprendizaje automático es necesario recopilar una base de datos personalizada para la primera etapa del entrenamiento del sistema. En la figura 3-4 se representa los repositorios disponibles en la red de los cuales se puede acudir para organizar las imágenes y etiquetarlas en carpetas.



**Figura 23-2:** Arquitectura del proyecto .  
Realizado por: Calero, R, 2021.

## 2.4 Algoritmo KNN



**Gráfico 2-2:** Diagrama de proceso del funcionamiento del algoritmo knn.  
Realizado por: Calero, R, 2021

## 2.5 Requerimientos para el Software

### 2.5.1 Entornos de desarrollo de programación Python

Para la ejecución y resolución de problemas del presente trabajo de titulación, se eligió el lenguaje de programación Python, ya que es un lenguaje de alto nivel ampliamente utilizado de propósito general. Se utiliza en el desarrollo web, ciencia de datos, creación de prototipos de software, inteligencia artificial como es el presente caso, etc.

Como nuestra formación no está orientada al desarrollo del software y el requerimiento de emparejar el razonamiento lógico de los algoritmos de inteligencia artificial con el hardware, exige utilizar un lenguaje de programación con una sintaxis simple y fácil de usar, lo que es la característica principal de Python.

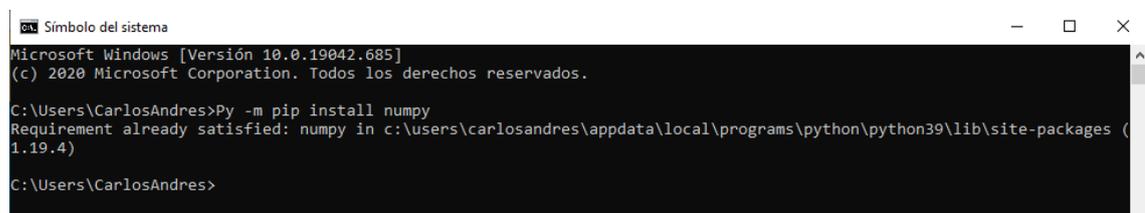
### 2.5.2 Instalación de paquetes frameworks

#### 2.5.2.1 Numpy

Numpy es una herramienta de computación numérica que ofrece funciones matemáticas completas, generadores de núcleos aleatorios, rutinas de álgebra lineal, transformadas de Fourier, etc. Además de sus usos científicos obvios, NumPy también se puede utilizar como un contenedor multidimensional eficiente de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite que NumPy se integre sin problemas y con rapidez con una amplia variedad de bases de datos.

Para la instalación es necesario ingresar al CMD del sistema e ingresar el siguiente código:

*Py -m pip install numpy*



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.685]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\CarlosAndres>Py -m pip install numpy
Requirement already satisfied: numpy in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (1.19.4)

C:\Users\CarlosAndres>
```

**Figura 24-2:** Captura del proceso para la instalación del paquete numpy  
Realizado por: Calero, R, 2021

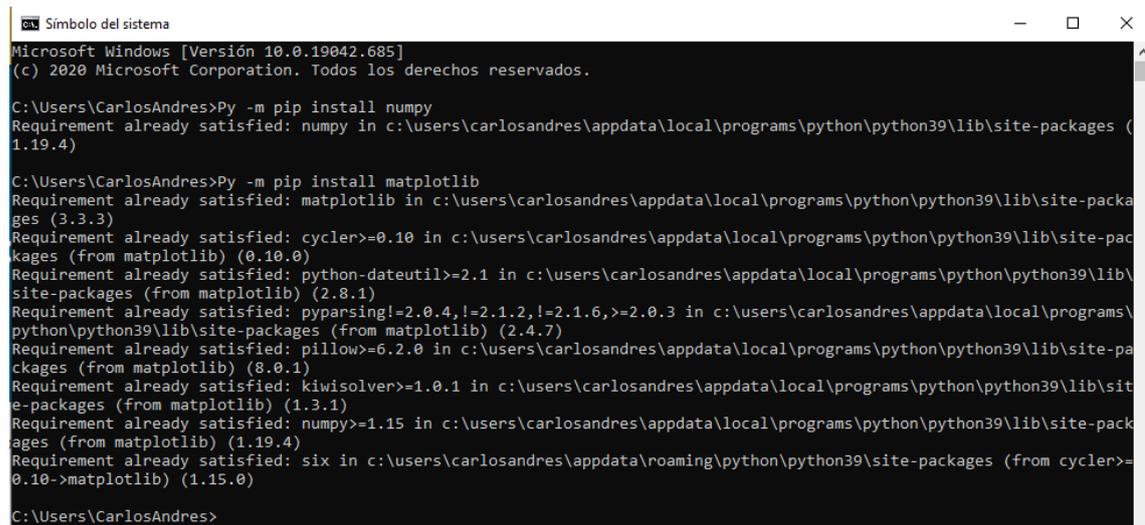
### 2.5.2.2 Matplotlib.pyplot

Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python.

Matplotlib produce figuras con calidad de publicación en una variedad de formatos impresos y entornos interactivos en todas las plataformas. Matplotlib se puede utilizar en scripts de Python, el shell de Python e IPython, servidores de aplicaciones web y varios kits de herramientas de interfaz gráfica de usuario.

Para la instalación es necesario ingresar al CMD del sistema e ingresar el siguiente código:

*Py -m pip install matplotlib*



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.685]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\CarlosAndres>Py -m pip install numpy
Requirement already satisfied: numpy in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (1.19.4)

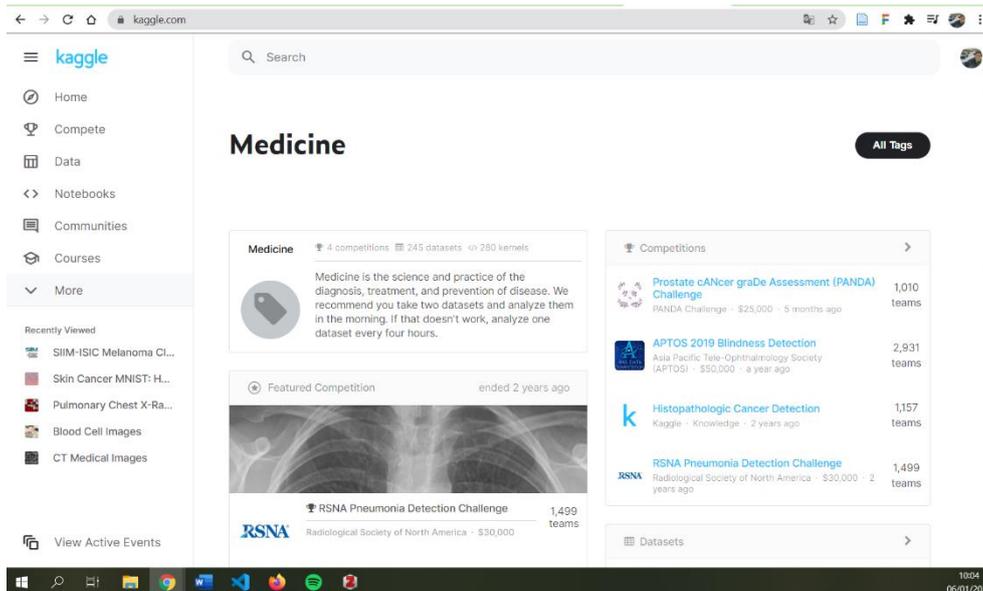
C:\Users\CarlosAndres>Py -m pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (3.3.3)
Requirement already satisfied: cyclor>=0.10 in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!2.1.2,!2.1.6,>=2.0.3 in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (8.0.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: numpy>=1.15 in c:\users\carlosandres\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (1.19.4)
Requirement already satisfied: six in c:\users\carlosandres\appdata\roaming\python\python39\site-packages (from cyclor>=0.10->matplotlib) (1.15.0)

C:\Users\CarlosAndres>
```

**Figura 25-2:** Captura del proceso de la instalación del paquete matplotlib  
Realizado por: Calero, R, 2021

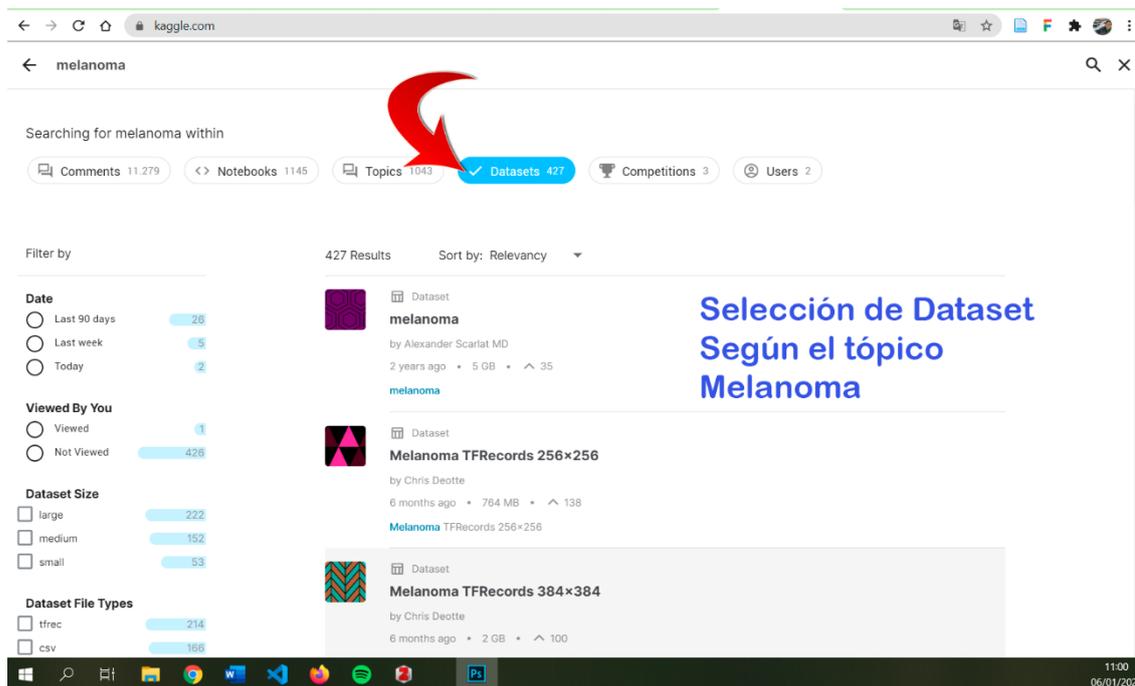
## 2.6 Adquisición del conjunto de datos (Dataset)

Para la adquisición de los conjuntos de datos previamente clasificados y estructurados por expertos en detección de melanomas y otras anomalías presentes en la piel, se utilizó *Kaggle*, esta es una plataforma web donde los científicos de datos pueden adquirir la materia prima para sus análisis. Kaggle es una plataforma de fuentes múltiples para atraer, nutrir, capacitar y desafiar a los científicos de datos de todo el mundo para resolver problemas de big data, aprendizaje automático y el análisis predictivo. Tiene más de 53600 miembros activos en 194 países y recibe cerca de 150000 prestaciones por mes, esta información («What is Kaggle, Why I Participate, What is the Impact?» 2020)

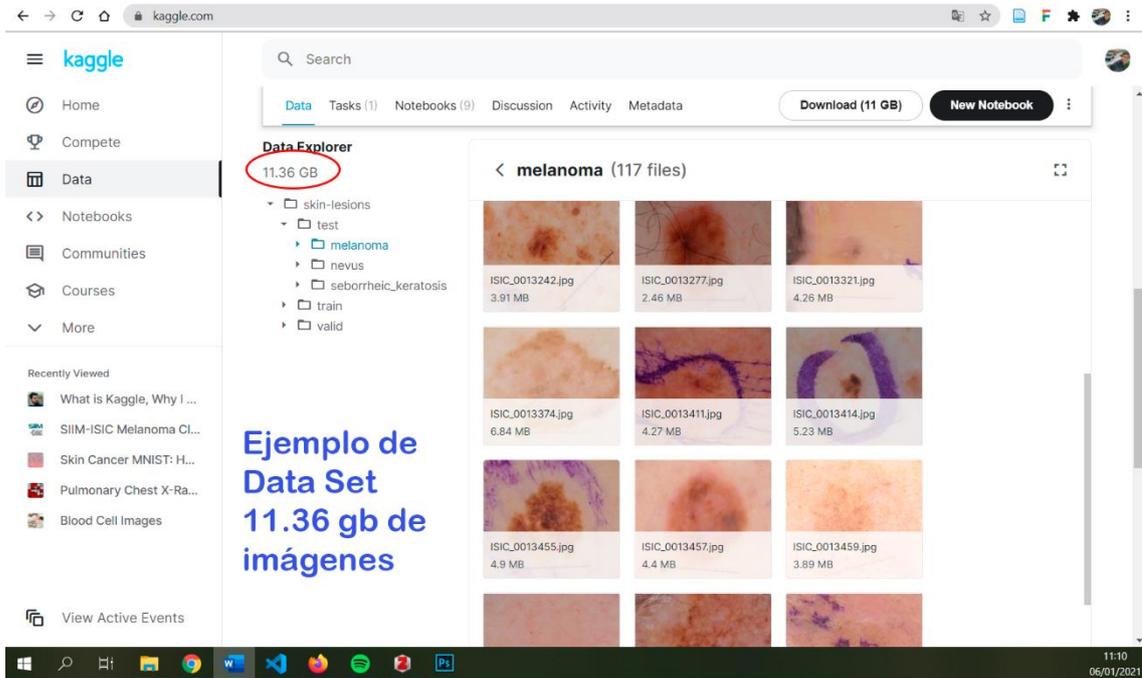


**Figura 26-2:** Plataforma Kaggle.  
 fuente: <https://www.kaggle.com/tags/medicine>

En la plataforma buscamos el Dataset recopilado por científicos de datos, de la siguiente manera:



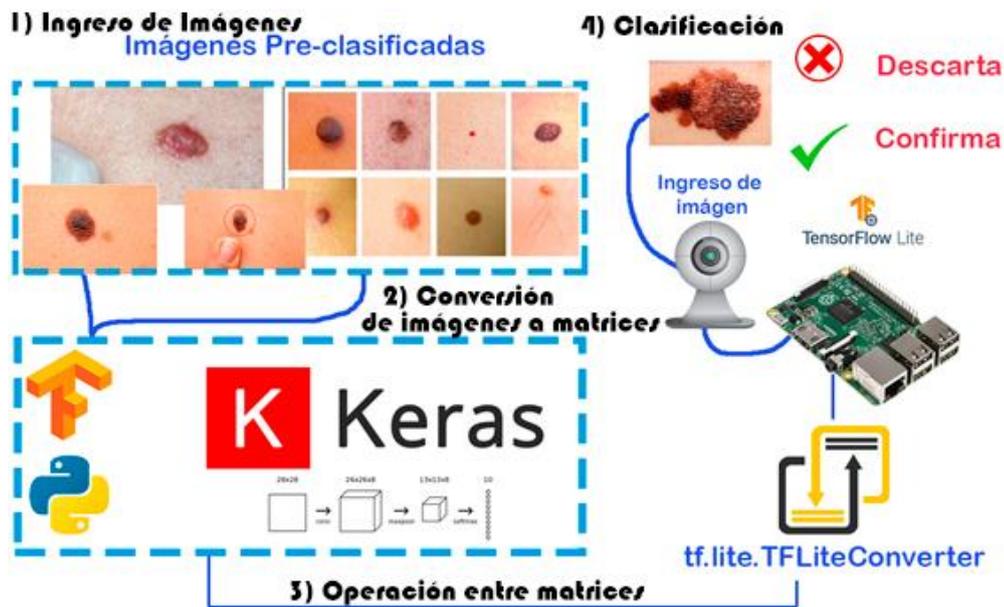
**Figura 27-2:** Plataforma Kaggle.  
 fuente: <https://www.kaggle.com/tags/medicine>



**Figura 28-2:** Plataforma Kaggle Imagenología.  
fuente: <https://www.kaggle.com/tags/medicine>

PONER SIMPLIFICADO...

## 2.7 Arquitectura del proyecto.



**Figura 29-2:** Arquitectura funcionamiento del proyecto.  
Realizado por: Calero, R, 2021

En la figura 12-2 se esquematiza el proceso del análisis a partir del conjunto de datos previamente obtenidos, clasificados y modelados. Se establece la utilización del framework especializado en redes neuronales artificiales convolucionales, Keras que es la api de alto nivel para el análisis

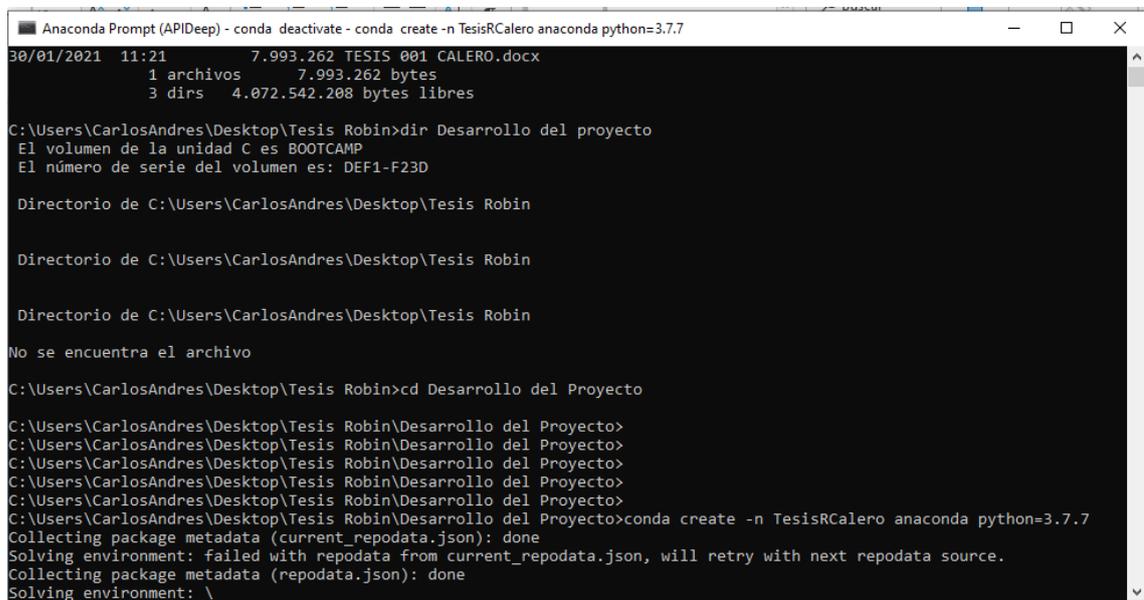
matemático de las imágenes desfragmentadas en píxeles, todo esto se lleva a la puesta en producción y montaje en la placa de Raspberri Pi 4 mediante Tensorflow lite pensada para ejecutarse en dispositivos móviles, mediante el código escrito en el gráfico anterior.

## 2.8 Preparación del sistema de control

Para ejecutar la implementación del sistema, primero se debe considerar que el prototipo está compuesto de 2 etapas, la primera que corresponde al entrenamiento la misma que se recomienda ejecutar desde un computador por la demanda de recursos que conlleva crear un modelo de predicción según las épocas que se declaren en el modelo, esto se lo evaluará en el punto de resultados. La segunda etapa se ejecutará en el Raspberry ya que corresponde a la ejecución del sistema utilizando el framework de Streamlit.

### 2.8.1 Preparación del entorno

Se recomienda utilizar ambientes con Miniconda ya que en las pruebas realizadas el algoritmo no presenta errores en la ejecución con la versión de Python 3.7.7, de esta manera los ambientes ayudan a separar versiones de librerías y trabajarlos por separado, como se muestra en la Figura 13-2.



```
Anaconda Prompt (APIDeep) - conda deactivate - conda create -n TesisRCalero anaconda python=3.7.7
30/01/2021 11:21 7.993.262 TESIS 001 CALERO.docx
1 archivos 7.993.262 bytes
3 dirs 4.072.542.208 bytes libres

C:\Users\CarlosAndres\Desktop\Tesis Robin>dir Desarrollo del proyecto
El volumen de la unidad C es BOOTCAMP
El número de serie del volumen es: DEF1-F23D

Directorio de C:\Users\CarlosAndres\Desktop\Tesis Robin

Directorio de C:\Users\CarlosAndres\Desktop\Tesis Robin

Directorio de C:\Users\CarlosAndres\Desktop\Tesis Robin
No se encuentra el archivo

C:\Users\CarlosAndres\Desktop\Tesis Robin>cd Desarrollo del Proyecto
C:\Users\CarlosAndres\Desktop\Tesis Robin\Desarrollo del Proyecto>
C:\Users\CarlosAndres\Desktop\Tesis Robin\Desarrollo del Proyecto>conda create -n TesisRCalero anaconda python=3.7.7
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: \
```

**Figura 30-2:** Creación Entorno o ambiente de programación TesisRCalero. Realizado por: Calero, R, 2021

En la figura 15-2 se crea el entorno TesisRCalero para trabajar con la versión de Python 3.7.7 por separado y establecer las dependencias necesarios para la etapa del entrenamiento.

Dependencias:

- **Ipykernel.** – instalación del kernel de Python.
- **Tesnsorflow** gpu versión 2.1.0 – Ejecución en la GPU
- **Keras** versión 2.3.1. – biblioteca de Redes Neuronales de código abierto

## 2.9 Implementación Algoritmo de Entrenamiento.

Para la implementación del algoritmo del entrenamiento se utilizó el ambiente de desarrollo Jupyter Notebook con el objetivo de ejecutar cada una de las partes del código secuencial, siendo estas:



**Gráfico 3-2:** Diagrama de proceso Entrenamiento para la generación del modelo de aprendizaje.

Realizado por: Calero, R, 2021

- Configuración de parámetros para el entrenamiento de imágenes:

```
from keras.models import Sequential, Model
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout, BatchNormalization, Input
from keras.optimizers import Adam
from keras.callbacks import TensorBoard, ModelCheckpoint
from keras.utils import np_utils
import os
import numpy as np
from keras.preprocessing import image
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.applications.vgg16 import VGG16
from keras.preprocessing.image import ImageDataGenerator
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
```

**Figura 31-2:** Configuración de parámetros.

Realizado por: Calero, R, 2021

```
width_shape = 224
height_shape = 224
num_classes = 3
epochs = 50
batch_size = 32
```

**Figura 32-2:** Configuración de parámetros.

Realizado por: Calero, R, 2021

- Selección del DataSet

```
train_data_dir = 'dataset/train'
validation_data_dir = 'dataset/valid'
```

**Figura 33-2:** Selección del DataSet.

Realizado por: Calero, R, 2021

- Generador de Imágenes (entrenamiento y validación)

```

train_datagen = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=False,
    preprocessing_function=preprocess_input)

valid_datagen = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=False,
    preprocessing_function=preprocess_input)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(width_shape, height_shape),
    batch_size=batch_size,
    #save_to_dir='',
    class_mode='categorical')

validation_generator = valid_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(width_shape, height_shape),
    batch_size=batch_size,
    #save_to_dir='',
    class_mode='categorical')

```

**Figura 34-2:** Entrenamiento y validación.

Realizado por: Calero, R, 2021

- Entrenamiento del modelo VGG16

```

nb_train_samples = 2000
nb_validation_samples = 150

image_input = Input(shape=(width_shape, height_shape, 3))

model = VGG16(input_tensor=image_input, include_top=True, weights='imagenet')

last_layer = model.get_layer('fc2').output
out = Dense(num_classes, activation='softmax', name='output')(last_layer)
custom_vgg_model = Model(image_input, out)

for layer in custom_vgg_model.layers[:-1]:
    layer.trainable = False

custom_vgg_model.compile(loss='categorical_crossentropy', optimizer='adadelta', metrics=['accuracy'])

custom_vgg_model.summary()

model_history = custom_vgg_model.fit_generator(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator,
    steps_per_epoch=nb_train_samples//batch_size,
    validation_steps=nb_validation_samples//batch_size)

```

**Figura 35-2:** Creación del modelo VGG16.

Realizado por: Calero, R, 2021

- Guardar el modelo entrenado

```
custom_vgg_model.save("models/model_VGG16.h5")
```

**Figura 36-2:** Guardar el modelo con extensión .h5.

Realizado por: Calero, R, 2021

- Gráficas o Reportes dinámicos del Entrenamiento

```
def plotTraining(hist, epochs, typeData):
    if typeData=="loss":
        plt.figure(1,figsize=(10,5))
        yc=hist.history['loss']
        xc=range(epochs)
        plt.ylabel('Loss', fontsize=24)
        plt.plot(xc,yc, '-r',label='Loss Training')
    if typeData=="accuracy":
        plt.figure(2,figsize=(10,5))
        yc=hist.history['accuracy']
        for i in range(0, len(yc)):
            yc[i]=100*yc[i]
        xc=range(epochs)
        plt.ylabel('Accuracy (%)', fontsize=24)
        plt.plot(xc,yc, '-r',label='Accuracy Training')
    if typeData=="val_loss":
        plt.figure(1,figsize=(10,5))
        yc=hist.history['val_loss']
        xc=range(epochs)
        plt.ylabel('Loss', fontsize=24)
        plt.plot(xc,yc, '--b',label='Loss Validate')
    if typeData=="val_accuracy":
        plt.figure(2,figsize=(10,5))
        yc=hist.history['val_accuracy']
        for i in range(0, len(yc)):
            yc[i]=100*yc[i]
        xc=range(epochs)
        plt.ylabel('Accuracy (%)', fontsize=24)
        plt.plot(xc,yc, '--b',label='Training Validate')

    plt.rc('xtick',labelsize=24)
    plt.rc('ytick',labelsize=24)
    plt.rc('legend', fontsize=18)
    plt.legend()
    plt.xlabel('Number of Epochs',fontsize=24)
    plt.grid(True)
```

**Figura 37-2:** Graficas.

Realizado por: Calero, R, 2021

- Evaluación

```

from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.models import load_model

names = ['LUNAR', 'MELANOMA', 'QUERATOSIS SEBORREICA' ]

modelt = load_model("models/model_VGG16.h5")
#modelt = custom_vgg_model

imaget_path = "ImagenPrueba.jpg"
imaget=cv2.resize(cv2.imread(imaget_path), (width_shape, height_shape), interpolation = cv2.INTER_AREA)
xt = np.asarray(imaget)
xt=preprocess_input(xt)
xt = np.expand_dims(xt,axis=0)
preds = modelt.predict(xt)

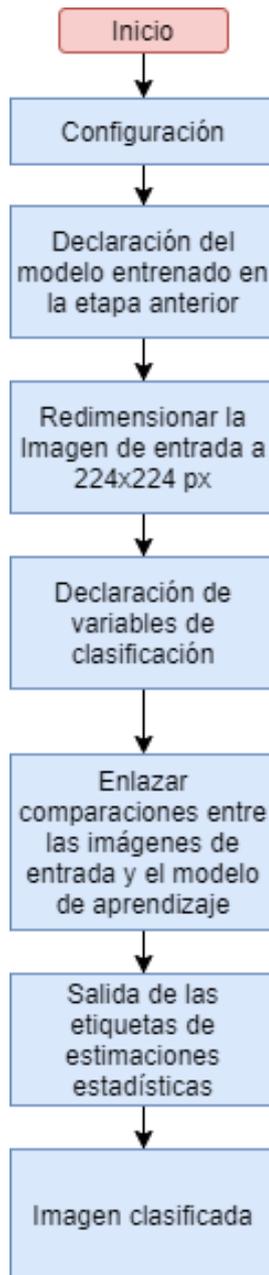
print(names[np.argmax(preds)])
plt.imshow(cv2.cvtColor(np.asarray(imaget),cv2.COLOR_BGR2RGB))
plt.axis('off')
plt.show()

```

**Figura 38-2:** Evaluación etapa entrenamiento.  
Realizado por: Calero, R, 2021

## 2.10 Proceso de Ejecución.

Una vez realizada la etapa del entrenamiento, el cual tuvo como objetivo crear un modelo de conocimiento capaz de identificar y clasificar imágenes según las variables antes expuestas (Lunar, Melanoma y Queratosis Seborreica.)



**Gráfico 4-2:** Diagrama de procesos para la ejecución de la clasificación.

Realizado por: Calero, R, 2021

### 2.10.1 App entorno visual para usuarios finales.

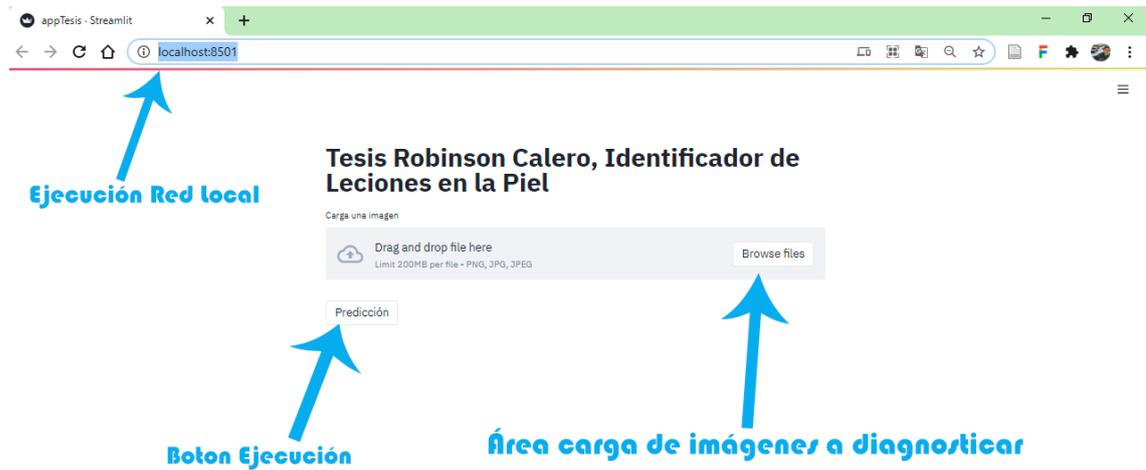


Figura 39-2: Appweb entorno interfaz gráfica.  
Realizado por: Calero, R, 2021

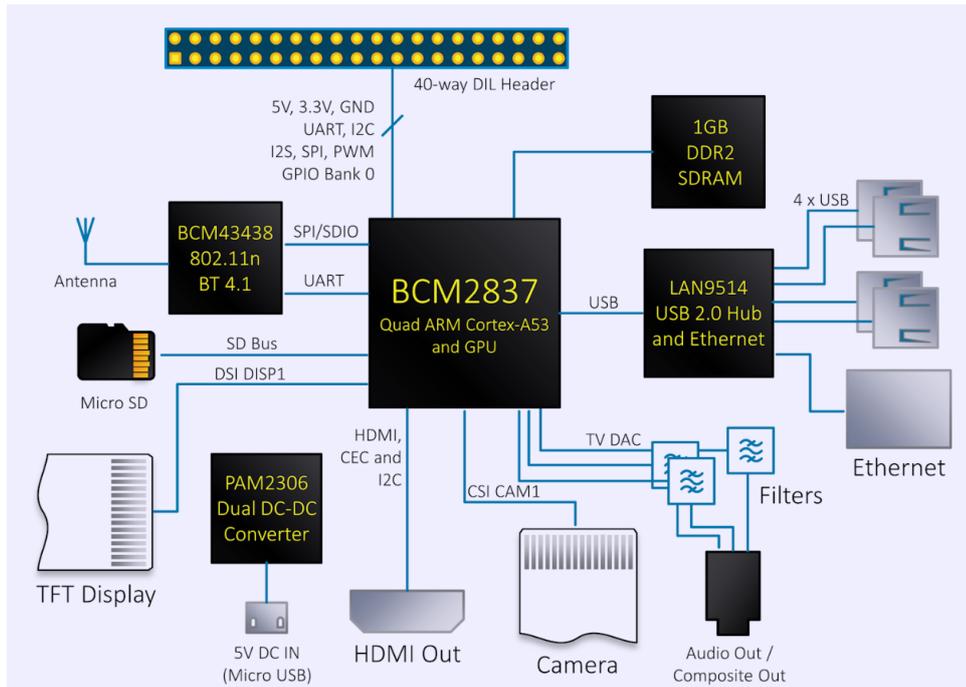


Figura 40-2: Appweb entorno interfaz gráfica.  
Realizado por: Calero, R, 2021

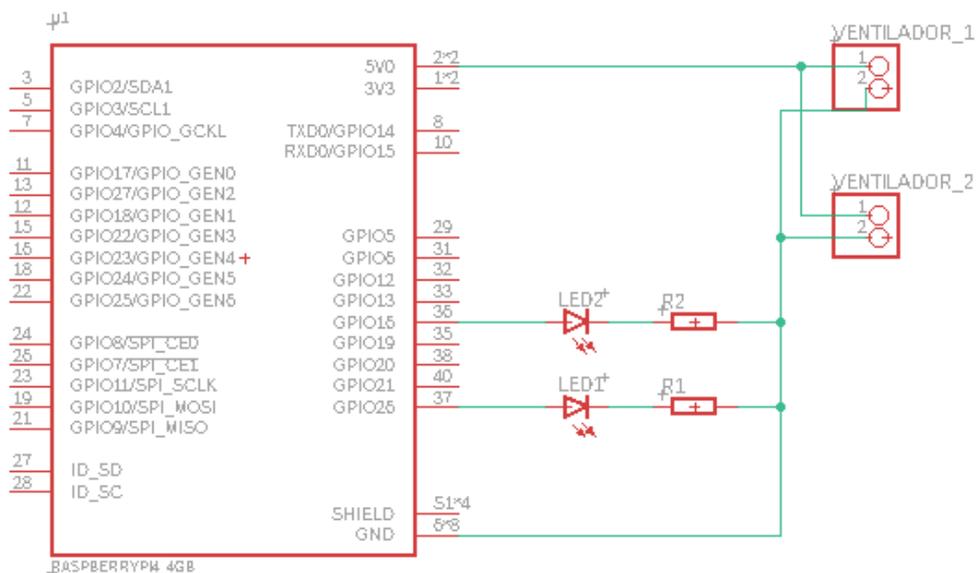
## 2.11 Desarrollo prototipo hardware.

### 2.11.1 Esquemático

En el presente apartado se explica cómo se implementó la ejecución del software en la tarjeta de desarrollo Raspberry pi 4 el mismo que cuenta con 4GB de RAM permitiendo el correcto funcionamiento de la aplicación de clasificación de imágenes de muestra.



**Figura 41-2:** GPIO Raspberry pi 4.  
Realizado por: Calero, R, 2021

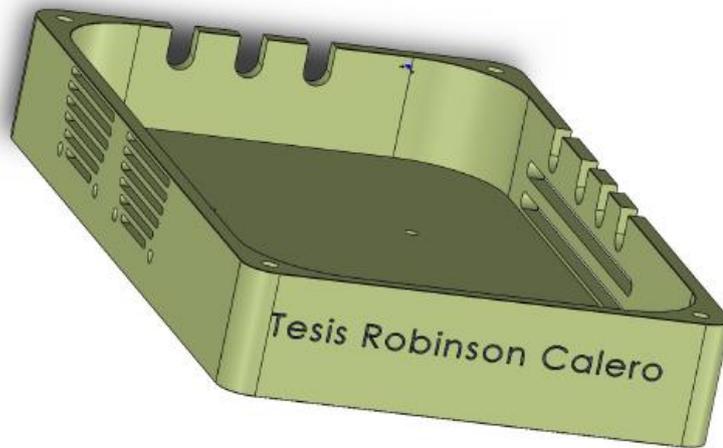


**Figura 42-2:** Esquemático.  
Realizado por: Calero, R, 2021

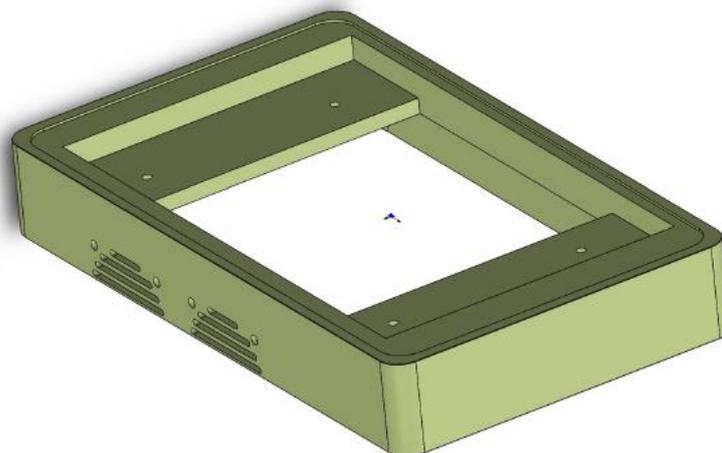
En la figura 32-3 existe la necesidad de proteger el rendimiento de la placa Raspberry Pi, es por este motivo se implementó 2 ventiladores para mantener la disipación de calor controlada.

### 2.11.2 *Desarrollo de la estructura del prototipo CAD Case*

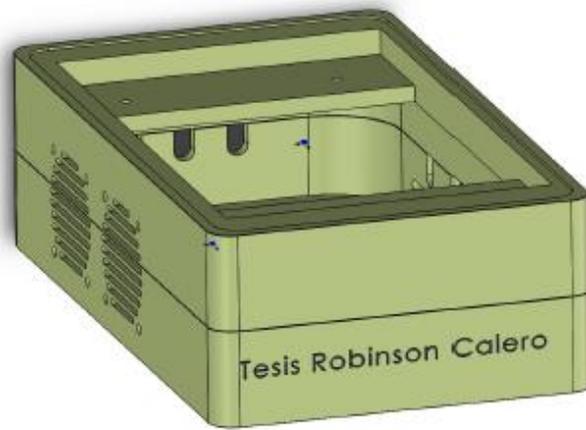
En el presente apartado se ilustra el case que protege al Raspberry Pi, de esta manera el prototipo se encuentra aislado de cualquier manipulación y en un entorno controlado. Para este desarrollo se utilizó el software *Solidworks* como se muestra en las figuras 33-2, 34-2 y 35-2.



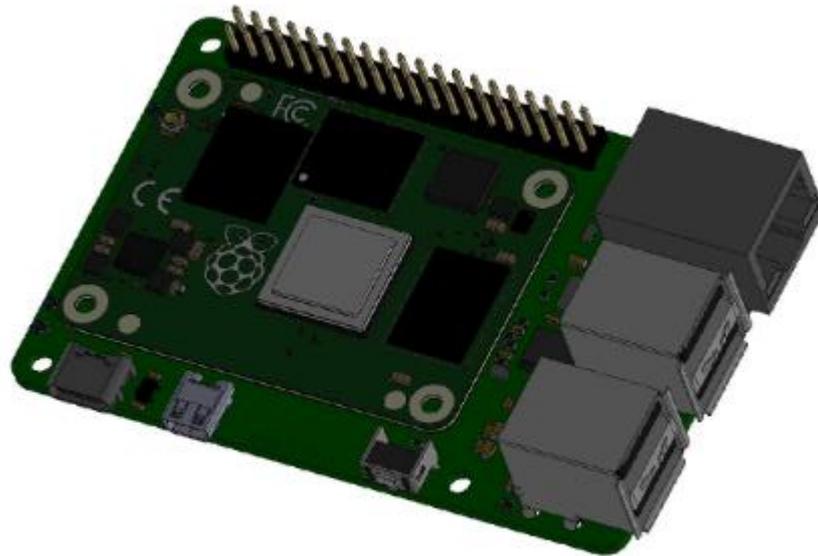
**Figura 43-2:** CAD base del case.  
Realizado por: Calero, R, 2021



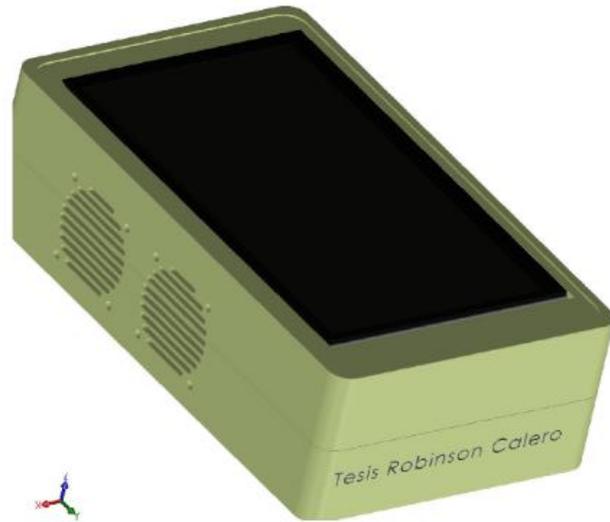
**Figura 44-2:** CAD top del case.  
Realizado por: Calero, R, 2021



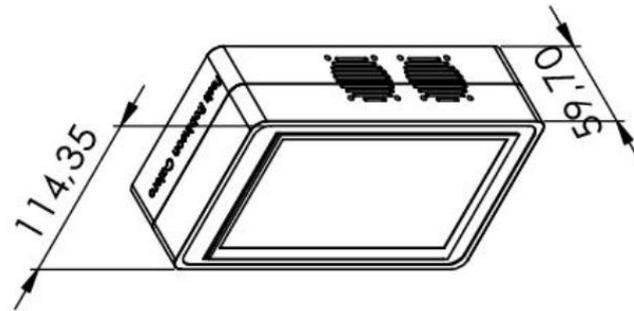
**Figura 45-2:** CAD case completo.  
Realizado por: Calero, R, 2021



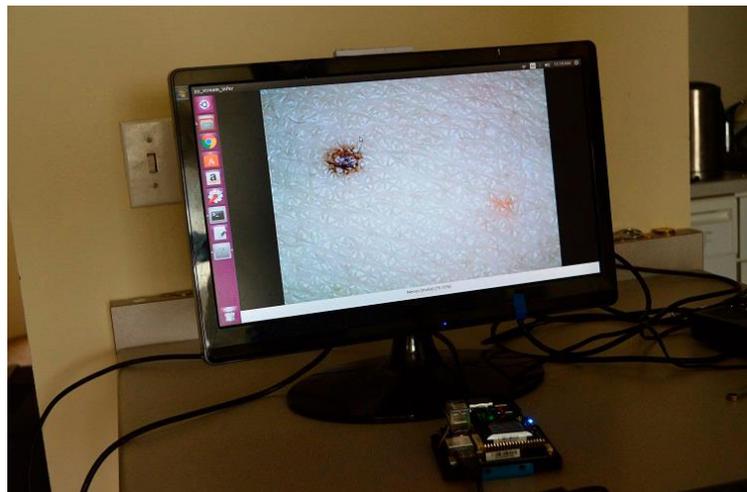
**Figura 46-2:** CAD Raspberry pi 4.  
Realizado por: Calero, R, 2021



**Figura 47-2:** CAD Prototipo.  
Realizado por: Calero, R, 2021



**Figura 48-2:** Draw Prototipo.  
Realizado por: Calero, R, 2021



**Figura 49-2:** Ilustración del dispositivo electrónico fase prototipado.  
Realizado por: Calero, R, 2021

## CAPITULO III

### 3. VALIDACIÓN DE PROTOTIPO

En la validación del prototipo se expondrán las diferentes pruebas realizadas para validar los objetivos planteados al inicio de esta investigación, utilizando metodologías objetivas y analíticas de validación observacional mediante la experimentación.

#### 3.1 Implementación del sistema

Para ejecutar la clasificación y validación de las etiquetas de cada imagen fue necesario preparar un dataset explicado anteriormente, el mismo que cuenta con 3 variables (Lunar, Melanoma y Queratosis Seborreica). Este conjunto de datos fue parte de la materia prima para generar el modelo de aprendizaje adecuado. Para dicho proceso se necesitó alrededor de 42 horas este proceso se lo puede constatar en el anexo 1.

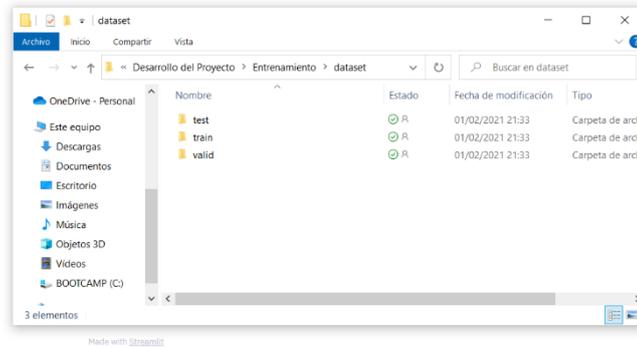
Posterior a este proceso, con el algoritmo de evaluación y ejecución del sistema, se toma el modelo de aprendizaje con las variables de los pesos ya establecidos en la red neuronal y se opta por medir el rendimiento con la siguiente metodología.

Considerando el gran número de imágenes proporcionada por kaggle.com, para el test y verificación del rendimiento de la red neuronal se apartan 393 imágenes de lunares, 117 imágenes de melanomas y 90 imágenes de queratosis seborreica, estas imágenes no forman parte del proceso de aprendizaje, con el objetivo de evaluar a la red neuronal de una no dependencia de imágenes memorizadas y pueda solventar una incertidumbre nueva.

### Tesis Robinson Calero, Identificador de Lesiones en la Piel

Carga una imagen  
Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG  
Browse files

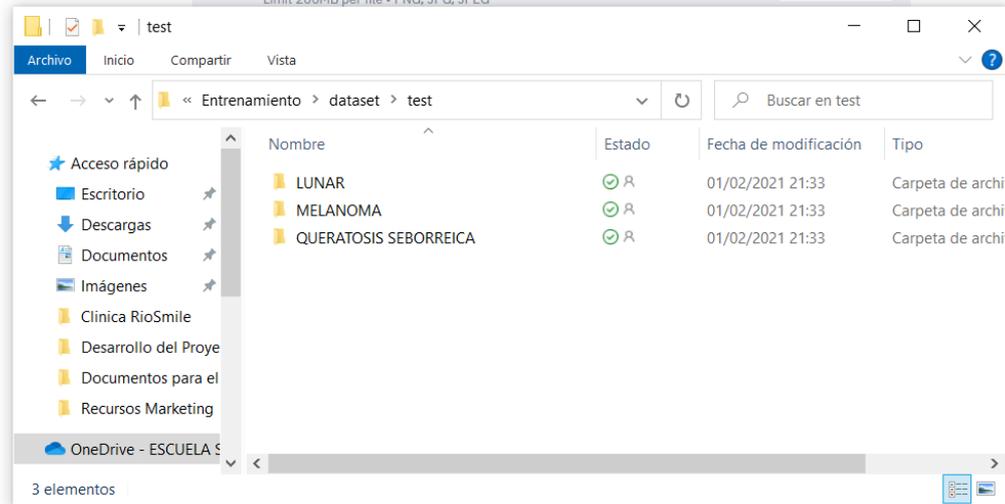
Predicción



**Figura 50-3:** Interfase para evaluación y carpetas contenedoras de información.  
Realizado por: Calero, R, 2021

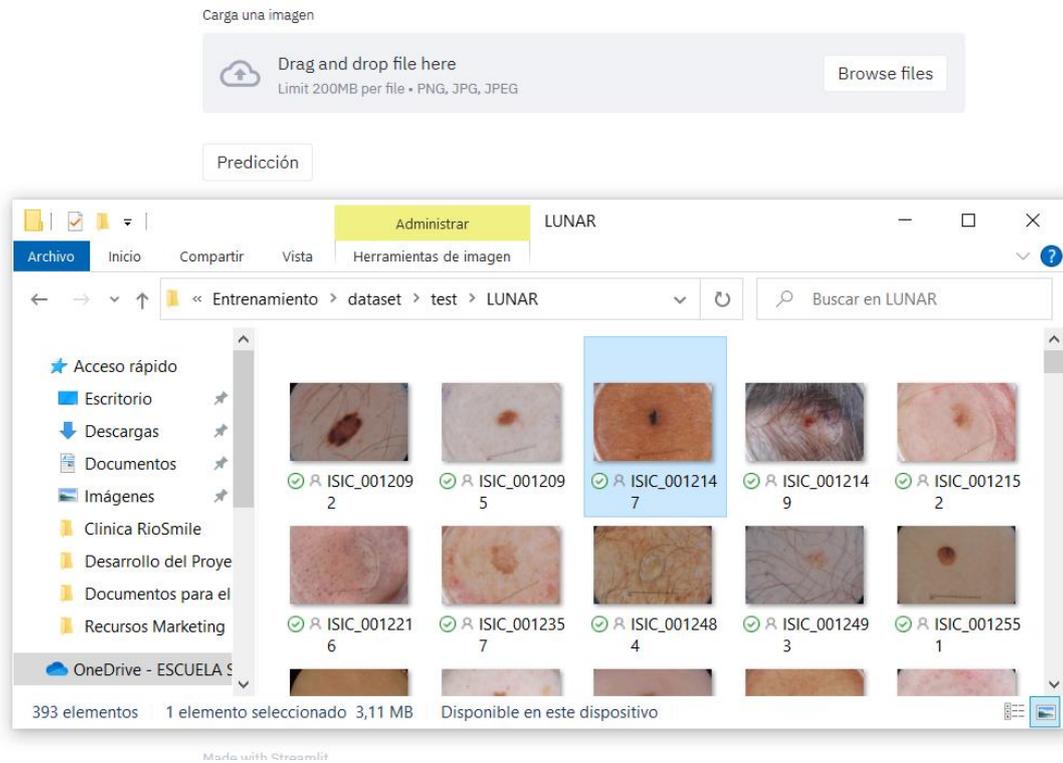
### Tesis Robinson Calero, Identificador de Lesiones en la Piel

Carga una imagen  
Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG  
Browse files



**Figura 51-3:** Variables.  
Realizado por: Calero, R, 2021

# Tesis Robinson Calero, Identificador de Lesiones en la Piel



**Figura 52-3:** Lunares 393 imágenes.  
Realizado por: Calero, R, 2021

## 3.2 Evaluación Variable Lunares

Para esta etapa de evaluación se considera el grupo de imágenes ilustradas en la figura 3-3 se toma al azar 50 imágenes para evaluar y el resultado de esta evaluación se refleja en la tabla 1-3.

**Tabla 1-3:**Tabla de evaluación de la clasificación de imágenes Lunares.

LUNARES		
No. Imagen	Respuesta Correcta	Respuesta Incorrecta
1	Sí	
2	Sí	
3	Sí	
4		No
5	Sí	
6	Sí	
7	Sí	

8	Sí	
9	Sí	
10		No
11	Sí	
12	Sí	
13	Sí	
14	Sí	
15	Sí	
16	Sí	
17		No
18	Sí	
19	Sí	
20	Sí	
21	Sí	
22	Sí	
23		No
24	Sí	
25	Sí	
26	Sí	
27		No
28		No
29	Sí	
30	Sí	
31	Sí	
32		No
33	Sí	
34	Sí	
35	Sí	
36	Sí	
37	Sí	
38	Sí	
39		No
40	Sí	
41	Sí	
42	Sí	
43	Sí	
44	Sí	
45	Sí	
46	Sí	
47	Sí	
48	Sí	
49	Sí	
50	Sí	
<b>Total</b>	<b>42</b>	<b>8</b>

**Tabla 2-3:** Resumen de la evaluación lunares.

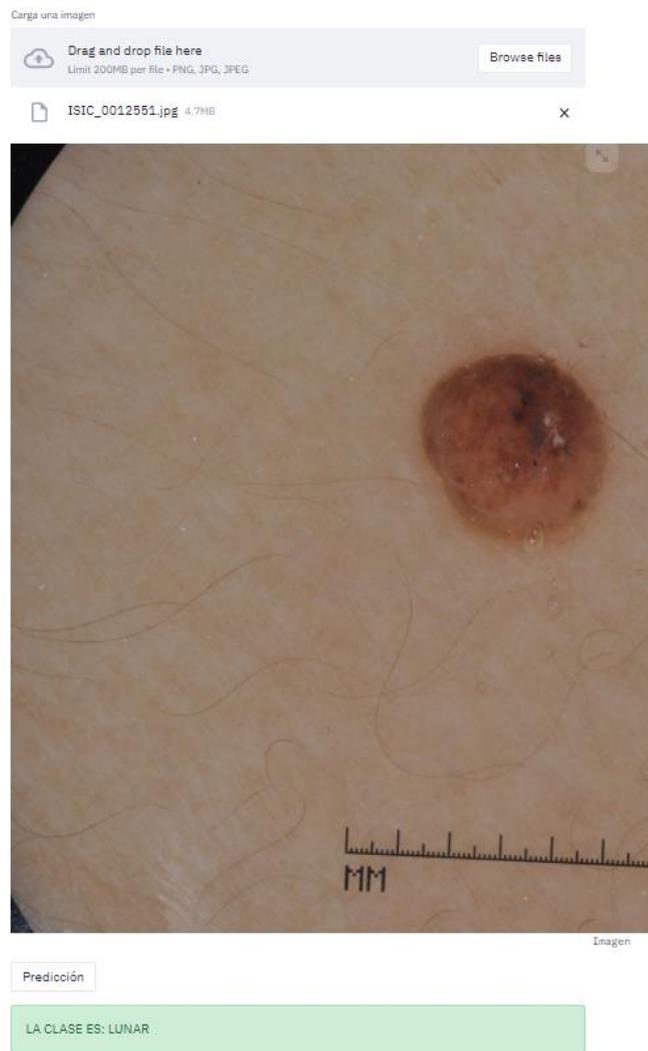
<b>RESUMEN</b>		
<b>Aciertos Positivos</b>	<b>Muestra</b>	<b>Error Absoluto</b>
42	50	8
<b>% Error</b>		
<b>16%</b>		

En la tabla 2-3 de acuerdo al proceso de validación se consideró un error del 16% en reconocer imágenes de lunares, este error se puede mitigar con un refuerzo en la etapa del entrenamiento, aumentando el número de épocas ajustar la red neuronal a una sensibilidad aceptable.



**Figura 53-3:** Imagen de ejemplo Lunar.  
Realizado por: Calero, R, 2021

## Tesis Robinson Calero, Identificador de Lesiones en la Piel



**Figura 54-3:** Predicción Lunar en la plataforma.  
Realizado por: Calero, R, 2021

### 3.3 Evaluación Variable Melanomas

Considerando la metodología de evaluación propuesta, de la base de datos explicada anteriormente de kaggle se toman las 50 primeras imágenes para la evaluación, recalando que estas imágenes no pertenecen al grupo de entrenamiento, con el objetivo de no recaer en un aprendizaje memorizado, el resultado se refleja en la tabla 3-3.

**Tabla 3-3:** Tabla de evaluación de la clasificación de imágenes Melanomas.

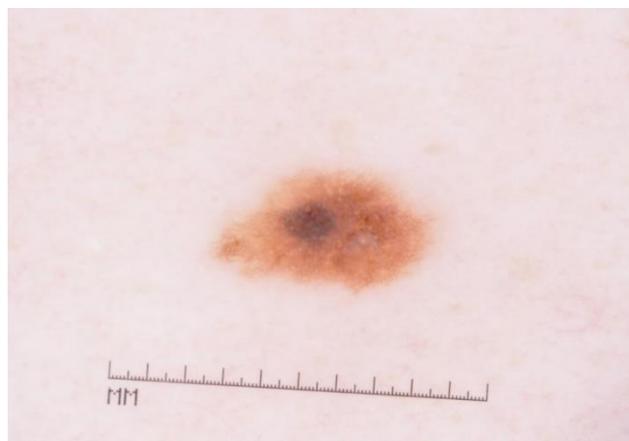
<b>Melanomas</b>		
<b>No. Imagen</b>	<b>Respuesta Correcta</b>	<b>Respuesta Incorrecta</b>
1	Sí	
2	Sí	
3	Sí	
4	Sí	
5	Sí	
6	Sí	
7	Sí	
8	Sí	
9	Sí	
10	Sí	
11	Sí	
12	Sí	
13	Sí	
14	Sí	
15		No
16	Sí	
17	Sí	
18	Sí	
19	Sí	
20		No
21	Sí	
22	Sí	
23	Sí	
24	Sí	
25	Sí	
26	Sí	
27	Sí	
28	Sí	
29	Sí	
30	Sí	
31	Sí	
32	Sí	
33	Sí	
34	Sí	
35	Sí	
36	Sí	
37	Sí	
38	Sí	

39	Sí	
40	Sí	
41	Sí	
42	Sí	
43	Sí	
44	Sí	
45	Sí	
46	Sí	
47	Sí	
48	Sí	
49	Sí	
50	Sí	
<b>Total</b>	<b>48</b>	<b>2</b>

**Tabla 4-3:** Resumen de la evaluación melanomas.

<b>RESUMEN</b>		
<b>Aciertos Positivos</b>	<b>Muestra</b>	<b>Error Absoluto</b>
48	50	2
<b>% Error</b>		
<b>4%</b>		

En la tabla 4-3, se identifican únicamente 2 errores, en la imagen 15 y 20, proporcionando un error del 4%, que es un error aceptable que mejorará con la propuesta del caso anterior (lunares).



**Figura 55-3:** Imagen de ejemplo Melanoma.  
Realizado por: Calero, R, 2021

## Tesis Robinson Calero, Identificador de Lesiones en la Piel



**Figura 56-3:** Predicción Melanoma en la plataforma.  
Realizado por: Calero, R, 2021

### 3.4 Evaluación Variable Queratosis Seborreica.

Para esta evaluación final, siguiendo con la homogeneidad de la muestra, se toman 50 imágenes de la base de datos y se obtienen los siguientes resultados de la tabla 5-3:

**Tabla 5-3:** Tabla de evaluación de la clasificación de imágenes Queratosis Seborreica.

<b>Queratosis Seborreica</b>		
<b>No. Imagen</b>	<b>Respuesta Correcta</b>	<b>Respuesta Incorrecta</b>
1	Sí	
2	Sí	
3		No
4	Sí	
5	Sí	
6	Sí	
7		No
8	Sí	
9	Sí	
10	Sí	
11	Sí	
12	Sí	
13	Sí	
14	Sí	
15	Sí	
16	Sí	
17	Sí	
18	Sí	
19	Sí	
20	Sí	
21	Sí	
22	Sí	
23	Sí	
24	Sí	
25	Sí	
26	Sí	
27		No
28	Sí	
29	Sí	
30	Sí	
31	Sí	
32	Sí	
33	Sí	
34	Sí	
35		No
36	Sí	
37	Sí	
38	Sí	

39	Sí	
40	Sí	
41	Sí	
42	Sí	
43	Sí	
44	Sí	
45	Sí	
46	Sí	
47	Sí	
48	Sí	
49	Sí	
50	Sí	
<b>Total</b>	<b>46</b>	<b>4</b>

**Tabla 6-3:** Resumen de la evaluación Queratosis Seborreica.

<b>RESUMEN</b>		
<b>Aciertos Positivos</b>	<b>Muestra</b>	<b>Error Absoluto</b>
46	50	4
<b>% Error</b>		
<b>8%</b>		

La tabla 5-3 indica la presencia de 4 errores en las imágenes 3, 7, 27 y 35, sumando un error del 8% que puede minimizarse con un entrenamiento superior a 50 épocas.

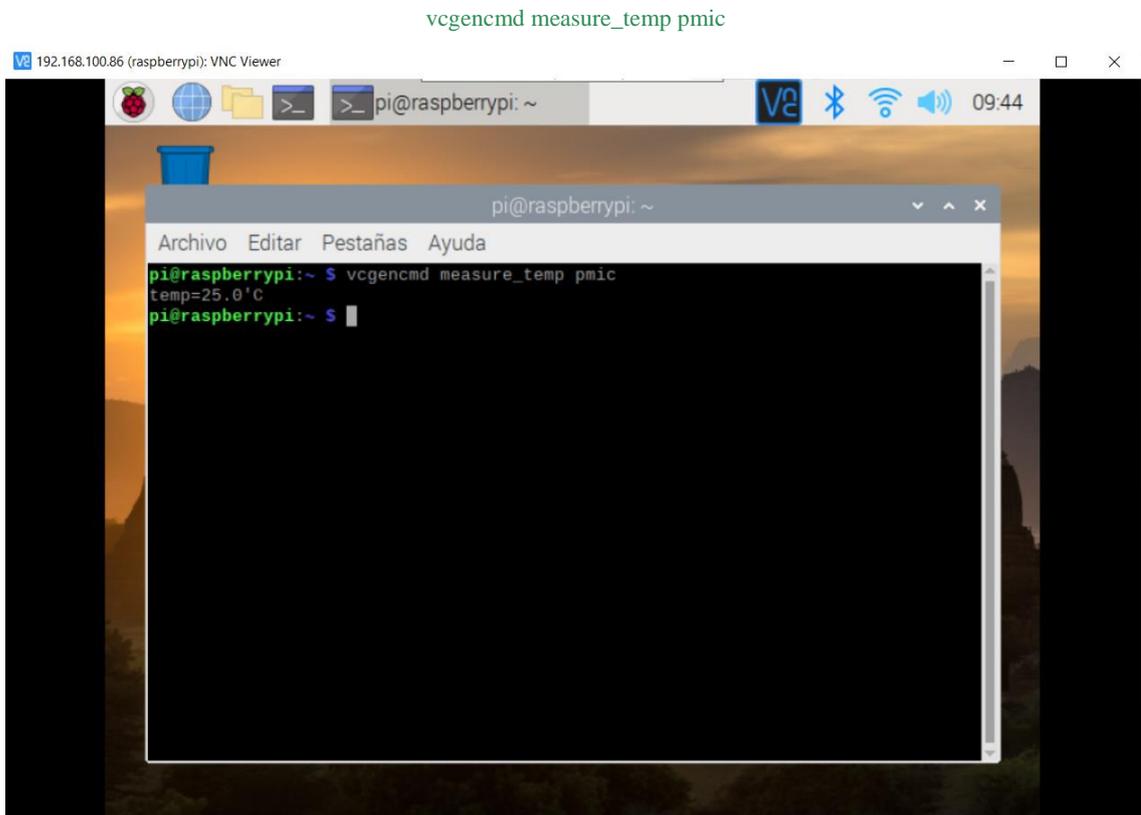
### 3.5 Análisis del consumo energético.

Para la esquematización de un prototipo es muy importante evaluar los niveles de consumo energético que demanda, con el objetivo de medir la escalabilidad y el uso al cual se podría enfocar, estos indicadores pueden ser parte de un análisis de costos para generar una producción exitosamente viable.

### 3.5.1 *Análisis de temperatura*

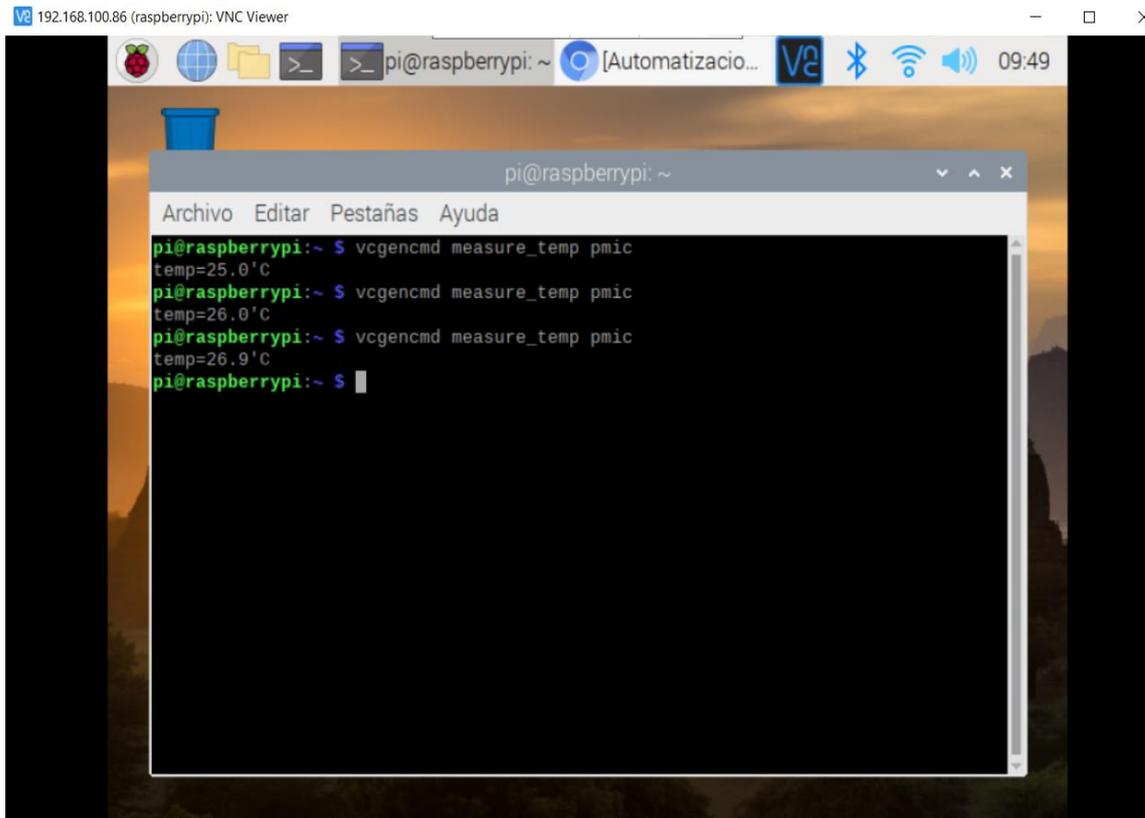
La temperatura de la tarjeta embebida Raspberry pi se la puede medir directamente desde el terminal con el comando:

```
vcgencmd measure_temp pmic
```



The image shows a VNC Viewer window titled '192.168.100.86 (raspberrypi): VNC Viewer'. The desktop background is a sunset landscape. A terminal window is open, displaying the command 'vcgencmd measure\_temp pmic' and the output 'temp=25.0°C'. The terminal window has a menu bar with 'Archivo', 'Editar', 'Pestañas', and 'Ayuda'. The system tray at the top right shows icons for VNC, Bluetooth, Wi-Fi, and a speaker, along with the time '09:44'.

**Figura 57-3:** Medición tiempo real Temperatura de operación Raspberry pi 5 minutos.  
Realizado por: Calero, R, 2021



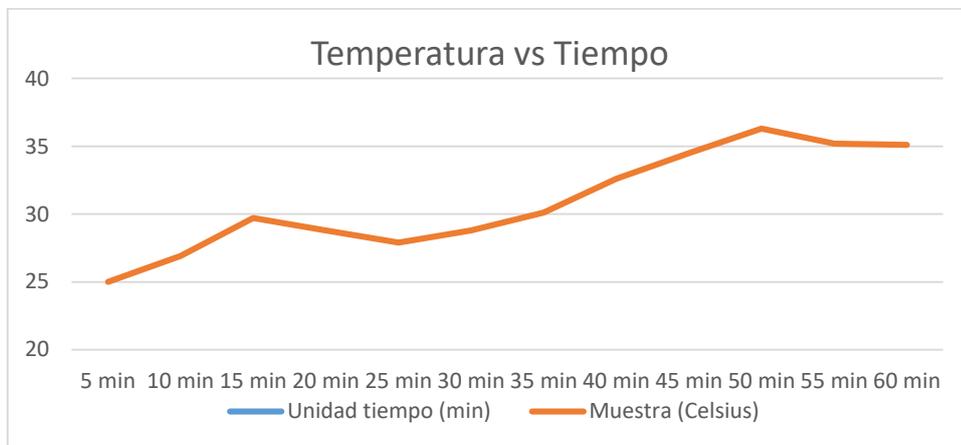
**Figura 58-3:** Medición tiempo real Temperatura de operación Raspberry pi 10 minutos.  
Realizado por: Calero, R, 2021

La metodología para la medición de esta variable, se la tomará 5 minutos después de encender la tarjeta Raspberry pi únicamente ejecutándose el terminal, a los 10 minutos ejecutando la aplicación en el gpu, y tomas de muestras continuas cada 5 minutos hasta tener un intervalo de 60 minutos 12 muestras.

**Tabla 7-3:** Análisis de Temperatura.

Numero de muestra	Unidad tiempo (min)	Muestra (Celsius)
1	5 min	25
2	10 min	26,9
3	15 min	29,7
4	20 min	28,8
5	25 min	27,9
6	30 min	28,8

7	35 min	30,1
8	40 min	32,6
9	45 min	34,5
10	50 min	36,3
11	55 min	35,2
12	60 min	35,1



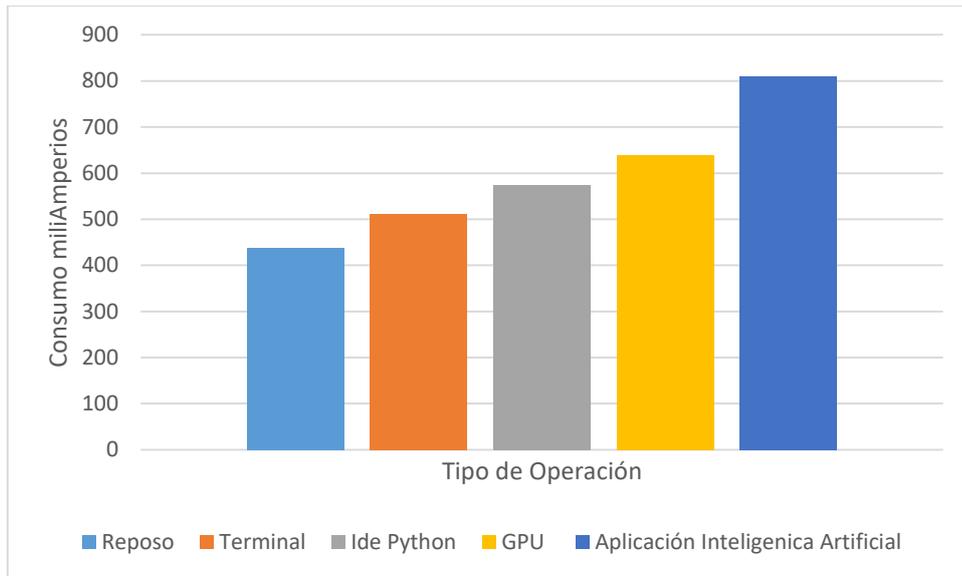
**Gráfico 5-3:** Temperatura operación vs Tiempo

### 3.5.2 Análisis consumo corriente.

En la siguiente tabla se ilustra el consumo en tiempo real de la corriente que demanda el raspberry pi 4 en operación.

**Tabla 8-3:** Análisis consumo corriente según obtenidos en las pruebas.

Tipo de operación	Consumo mA
Reposo	438
Terminal	511
Ide Python	575
GPU	640
Aplicación Inteligenica Artificial	810



**Gráfico 6-3:** Temperatura operación vs Tiempo

## CAPITULO IV

### 4. EVALUACIÓN ECONÓMICA

Hoy en día los diagnósticos médicos virtuales se han popularizado al pasar del tiempo y aún más en la crisis sanitaria actual, donde la brecha del tiempo para diagnosticar una afectación es muy importante, es por este motivo que varias empresas están aportando herramientas auxiliares a varias especialidades médicas, como es el caso de la dermatología en este capítulo se recopiló servicios ya establecidos en el mercado para poder realizar un análisis de costos y estimar una variación real de mercado, estas herramientas son las siguientes:

- **FotoFinder**, trabaja con un algoritmo de aprendizaje profundo que apoya a los dermatólogos en la detección más temprana posible del cáncer de piel. Las redes neuronales artificiales facilitan un aprendizaje automático complejo que es similar a los procesos de aprendizaje biológico. FotoFinder se comercializa directamente a establecimientos de salud, dentro de la empresa ofertante se derivan varios productos y servicios, en el cual se expone un precio de 4 000 USD por el kit consulta automática de cabello y piel. (Fotofinder 2020).
- **SkinVision**, es una aplicación móvil que funciona mediante un algoritmo de inteligencia artificial, que mediante el cual permite analizar manchas y lunares registrados en una fotografía. Al detectar cualquier anomalía emite recomendaciones para que un profesional pueda transmitir el análisis al paciente evaluado. SkinVision tiene un costo de 49,99 euros anuales. («SkinVision Service | Perform self-checks for Skin Cancer» 2017)

#### 4.1 Costos de implementación

Para la realización de este proyecto de investigación, en el siguiente apartado se analizan los estados de costos.

##### 4.1.1 *Costos directos*

Los costos directos, están directamente relacionados a los costos de fabricación o creación física del proyecto de investigación.

Dentro de los costos directos, es indispensable la utilización de un computador donde se realizará el análisis en una plataforma web, es allí donde se realiza una retroalimentación para que el sistema pueda seguir afinando y mitigando los errores. El sistema para que sea mas versátil puede migrar un dispositivo móvil, implementando en un raspberry pi como núcleo procesador de la inteligencia artificial utilizando el GPU de la tarjeta embebida, se consideró utilizar un raspberry pi 4, por la las 4gb en memoria RAM que ofrece, los costos se evidencian en la tabla 1-4 descrita a continuación.

#### 4.1.1.1 Costos implementación eléctrica

**Tabla 9-4:** Costos de materiales y dispositivos

CANTIDAD	ELEMENTO	V. unitario (USD)	V. Total (USD)
1	Computador	800	800
1	Raspberry pi 4 4GB RAM Model B	140	140
1	Fuente de poder 5v 2A	10	10
<b>TOTAL</b>			950

Realizado por: Calero, R, 2021

#### 4.1.1.2 Costos implementación estructural

Para proteger la tarjeta y no se interrumpa el proceso por agentes externos, ya sea por falla humana o ambiental, se decidió protegerlo con un case al raspberry pi.

**Tabla 10-4:** Valores protección y estructura

CANTIDAD	ELEMENTO	V. unitario (USD)	V. Total
1	Case	40	40
<b>TOTAL</b>			40

Realizado por: Calero, R, 2021

#### 4.1.2 Costos directos totales

**Tabla 11-4:** Costos directos totales

Costos	Totalidad de Valores en USD
Valores de costos electrónicos	950
Valores de costos del case	40
<b>Costo directo total</b>	<b>990</b>

Realizado por: Calero, R, 2021

#### 4.1.3 Costos Indirectos

**Tabla 12-4:** Costos Indirectos

N <sup>a</sup>	Descripción	V. unitario (USD)	V. Total (USD)
1	Imprevistos	50	50
		Total	50

Realizado por: Calero, R, 2021

#### 4.1.4 Costo Total

**Tabla 13-4:** Costos Totales

N <sup>a</sup>	Descripción	V. unitario (USD)	V. Total (USD)
1	Costos directos totales	990	990
2	Costos indirectos totales	50	50
		<b>Costo Total</b>	1040

Realizado por: Calero, R, 2021

Analizando las tablas anteriores se resume en un costo de elaboración de 1040 USD, este costo cubre imprevistos como costos de movilidad, electrónicos pequeños como cables, fuentes de poder, ventiladores y disipadores de calor. El costo de elaboración de este prototipo esta previsto todos los costos operativos y de desarrollo por el cual tiene una visión de crecimiento con una inversión mínima, ya que los algoritmos y la programación es lo que hace falta desarrollarla ya conjuntamente con profesionales expertos en la materia, y este es un costo de desarrollo que se tomará en cuenta a futuro.

## CONCLUSIONES

La metodología de análisis de Machine Learning de clasificación de patrones, conjuntamente con las técnicas de procesamiento digital de imágenes y redes neuronales tuvieron un resultado eficiente y exitoso en la complementación de un dispositivo inteligente identificador de Melanomas y Lunares Benignos.

Según el análisis establecido de los métodos y técnicas de inteligencia artificial, se concluyó que dentro de las técnicas del Soft computing, las redes neuronales artificiales convolucionales son las adecuadas para la clasificación de imágenes por su flexibilidad de operación en paralelo y conversión de imágenes de entrada a matrices para la comparación de píxeles parecidos.

Una vez analizado diferentes repositorios y conjunto de datos de plataformas médicas se estableció que kaggle.com fue el adecuado para esta investigación, por su amplia gama de aportaciones de conjunto de datos destinados para los científicos de datos ya que estos se encuentran ordenados, clasificados y etiquetados según análisis clínicos elaborados por especialistas en cada tema.

Una vez establecido el software, y generado el modelo de aprendizaje por medio de un algoritmo, se evaluó que la plataforma embebido ideal fue la raspberry pi 4 que cuenta con un soporte de video OpenGL ES 3.x minimizando los recursos del CPU por lo que la aplicación no presente latencia.

Se ha demostrado que esta metodología y los recursos utilizados para la implementación del prototipo tiene buenos resultados, obteniendo una precisión del 90.66%. Los errores identificados en Lunares tienen un 16% de error, 4% error en Melanomas y 8% de error en Queratosis Seborreica.

## **RECOMENDACIONES**

Para una mejor confiabilidad del veredicto del sistema es necesario seguir alimentando a la base de datos con la mayor información posible y aumentar el tiempo del entrenamiento de la red neuronal a 100 épocas (80 horas), que es el doble de tiempo empleado para este prototipo.

Se recomienda utilizar filtros especiales para eliminar el ruido en las imágenes ya que esto genera falsos positivos en varios casos y la red neuronal tenga una segmentación estadística errónea con muestras poco valiosas.

La media del error es del 9,3% tomando en cuenta que es la primera versión del prototipo y a la gran cantidad de datos y el tiempo empleado en el entrenamiento de la RNA con las cuales se puede mejorar la precisión del sistema.

## BIBLIOGRAFÍA

BATH-HEXTALL, F.J., et al., 2007. Interventions for basal cell carcinoma of the skin. *The Cochrane Database of Systematic Reviews*, no. 1, pp. CD003412. ISSN 1469-493X. DOI 10.1002/14651858.CD003412.pub2.

CASARIEGO, Z.J. y BAUDO, J.E., 2004. Trabajo de revisión: melanoma. *Avances en Periodoncia e Implantología Oral*, vol. 16, no. 3, pp. 157-177. ISSN 1699-6585.

DOLLING, O.R. y VARAS, E.A., 2002. Artificial neural networks for streamflow prediction. *Journal of Hydraulic Research*, vol. 40, no. 5, pp. 547-554. ISSN 0022-1686. DOI 10.1080/00221680209499899.

EL NAQA, I. y MURPHY, M.J., 2015. What Is Machine Learning? En: I. EL NAQA, R. LI y M.J. MURPHY (eds.), *Machine Learning in Radiation Oncology: Theory and Applications* [en línea]. Cham: Springer International Publishing, pp. 3-11. [Consulta: 8 febrero 2021]. ISBN 978-3-319-18305-3. Disponible en: [https://doi.org/10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1).

ELMAHI, B., 2015. A Review: Fuzzy Logic techniques improve the efficiency of the power system stability. [en línea]. [Consulta: 6 marzo 2021]. Disponible en: </paper/A-Review%3A-Fuzzy-Logic-techniques-improve-the-of-the-Elmahi/92f1e5dfd6c74532e7bbb9ea5ee03800c550352f>.

ETZIONI, O., 1996. Moving up the information food chain: deploying softbots on the world wide web. *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2*. Portland, Oregon: AAAI Press, pp. 1322-1326. ISBN 978-0-262-51091-2.

FARMER, K.C. y NAYLOR, M.F., 1996. Sun exposure, sunscreens, and skin cancer prevention: a year-round concern. *The Annals of Pharmacotherapy*, vol. 30, no. 6, pp. 662-673. ISSN 1060-0280. DOI 10.1177/106002809603000615.

FOTOFINDER, 2020. Fotofinder. [en línea]. [Consulta: 10 marzo 2021]. Disponible en: <https://www.fotofinder.de/es/>.

Guia inicial de TensorFlow 2.0 para principiantes | TensorFlow Core. [en línea], 2021. [Consulta: 6 marzo 2021]. Disponible en: <https://www.tensorflow.org/tutorials/quickstart/beginner?hl=es-419>.

KONONENKO, I., 2001. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, vol. 23, no. 1, pp. 89-109. ISSN 09333657. DOI 10.1016/S0933-3657(01)00077-X.

MASSAD, E., et al., 2008. Basic Concepts of Fuzzy Sets Theory. En: E. MASSAD, N.R.S. ORTEGA, L.C. DE BARROS y C.J. STRUCHINER (eds.), *Fuzzy Logic in Action: Applications in Epidemiology and Beyond* [en línea]. Berlin, Heidelberg: Springer, Studies in Fuzziness and Soft Computing, pp. 11-40. [Consulta: 29 marzo 2021]. ISBN 978-3-540-69094-8. Disponible en: [https://doi.org/10.1007/978-3-540-69094-8\\_2](https://doi.org/10.1007/978-3-540-69094-8_2).

MCKEAN, M. y AMARIA, R., 2017. Using Adoptive Cell Therapy in the Modern Era of Metastatic Melanoma Treatment. *Clinical Skin Cancer*, vol. 2, no. 1, pp. 32-38. ISSN 2405-8645. DOI 10.1016/j.clsc.2017.10.001.

NILSSON, N.J., 1965. *Learning machines*. McGrawHill: New York. Learning machines.

OVASKA, S.J., VANLANDINGHAM, H.F. y KAMIYA, A., 2002. Fusion of soft computing and hard computing in industrial applications: an overview. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 2, pp. 72-79. ISSN 1558-2442. DOI 10.1109/TSMCC.2002.801354.

RADHI, J.M., 1999. Malignant melanoma arising from nevi, p53, p16, and Bcl-2: expression in benign versus malignant components. *Journal of Cutaneous Medicine and Surgery*, vol. 3, no. 6, pp. 293-297. ISSN 1203-4754. DOI 10.1177/120347549900300603.

RUDOWSKY, I., 2004. Intelligent Agents. *Communications of the Association for Information Systems* [en línea], vol. 14. [Consulta: 29 marzo 2021]. ISSN 15293181. DOI 10.17705/1CAIS.01414. Disponible en: <https://aisel.aisnet.org/cais/vol14/iss1/14>.

RUSSELL, S., et al., 2011. *Inteligencia artificial: un enfoque moderno*. Madrid: Pearson Educación. ISBN 978-84-205-4003-0.

Skin Cancer Causes & Risk Factors | Reasons & Effects. *Indus Health Plus* [en línea], [sin fecha]. [Consulta: 29 marzo 2021]. Disponible en: <https://www.indushealthplus.com/skin-cancer-causes-risk-factors.html>.

Skin Cancer Information. *The Skin Cancer Foundation* [en línea], [sin fecha]. [Consulta: 20 febrero 2021]. Disponible en: <https://www.skincancer.org/skin-cancer-information/>.

SkinVision Service | Perform self-checks for Skin Cancer. *SkinVision* [en línea], 2017. [Consulta: 19 octubre 2020]. Disponible en: <https://www.skinvision.com/service/>.

TensorFlow. *TensorFlow* [en línea], 2021. [Consulta: 6 marzo 2021]. Disponible en: <https://www.tensorflow.org/?hl=es-419>.

URBACH, F., 1993. Environmental Risk Factors for Skin Cancer. En: E. HECKER, E.G. JUNG, F. MARKS y W. TILGEN (eds.), *Skin Carcinogenesis in Man and in Experimental Models*. Berlin, Heidelberg: Springer, pp. 243-262. ISBN 978-3-642-84881-0. DOI 10.1007/978-3-642-84881-0\_18.

VAN DER MALSBURG, C., 1986. Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. En: G. PALM y A. AERTSEN (eds.), *Brain Theory*. Berlin, Heidelberg: Springer, pp. 245-248. ISBN 978-3-642-70911-1. DOI 10.1007/978-3-642-70911-1\_20.

VIEIRA, S., et al., 2017. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. *Neuroscience & Biobehavioral Reviews*, vol. 74, pp. 58-75. ISSN 01497634. DOI 10.1016/j.neubiorev.2017.01.002.

WALCZAK, S. y CERPA, N., 2003. Artificial Neural Networks. En: R.A. MEYERS (ed.), *Encyclopedia of Physical Science and Technology (Third Edition)* [en línea]. New York: Academic Press, pp. 631-645. [Consulta: 8 febrero 2021]. ISBN 978-0-12-227410-7. Disponible en: <https://www.sciencedirect.com/science/article/pii/B0122274105008371>.

What is Kaggle, Why I Participate, What is the Impact? | Data Science and Machine Learning. [en línea], 2020. [Consulta: 6 enero 2021]. Disponible en: a.

WOOLDRIDGE, M. y JENNINGS, N.R., 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115-152. ISSN 0269-8889, 1469-8005. DOI 10.1017/S0269888900008122.

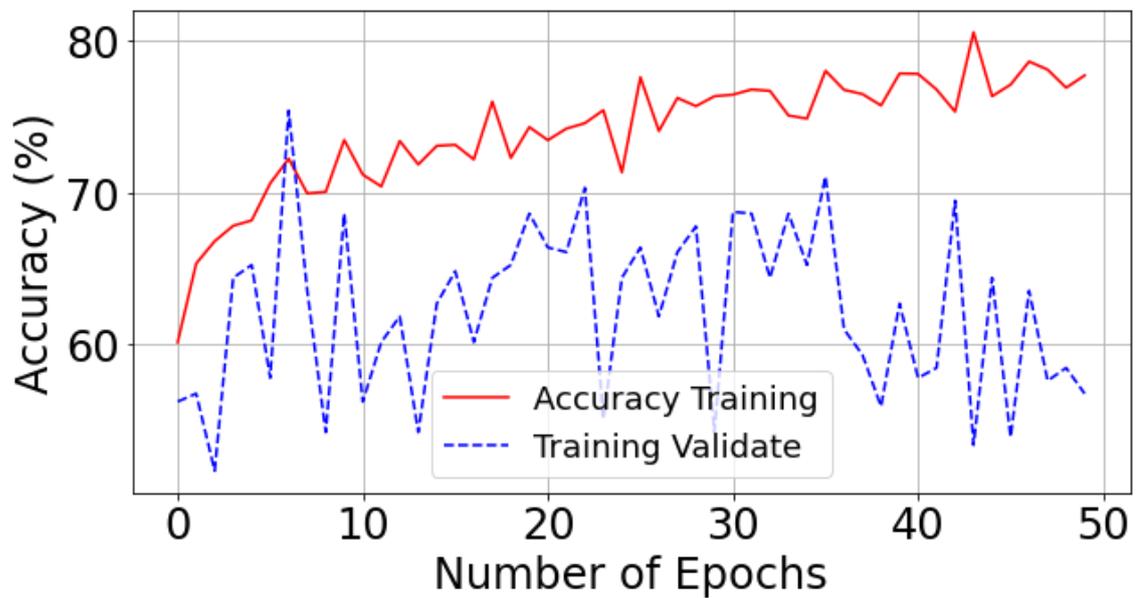
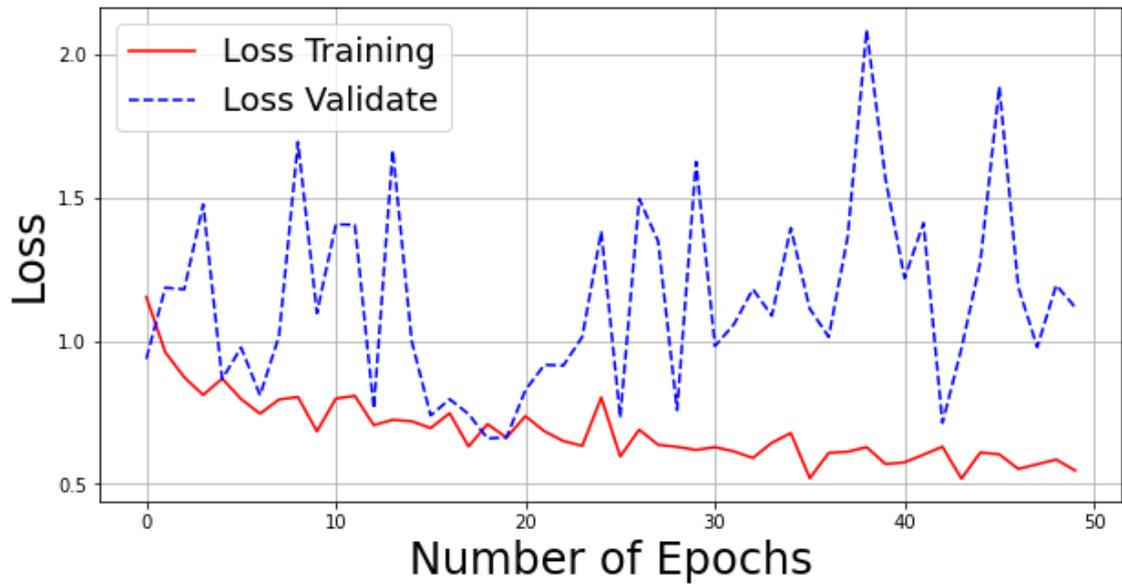
YARBRO, J.W., et al., 1999. American Joint Committee on Cancer prognostic factors consensus conference. *Cancer*, vol. 86, no. 11, pp. 2436-2446. ISSN 0008-543X. DOI 10.1002/(sici)1097-0142(19991201)86:11<2436::aid-cnrc35>3.0.co;2-#.

ZADEH, L.A., 1965. Fuzzy sets. *Information and Control*, vol. 8, no. 3, pp. 338-353. ISSN 0019-9958. DOI 10.1016/S0019-9958(65)90241-X.

ZELIČ, I., et al., 1997. Induction of Decision Trees and Bayesian Classification Applied to Diagnosis of Sport Injuries. *Journal of Medical Systems*, vol. 21, no. 6, pp. 429-444. ISSN 1573-689X. DOI 10.1023/A:1022880431298.

## ANEXOS

### ANEXO A: GRAFICAS RESULTADO DEL ENTRENAMIENTO



## ANEXO B: PROCESO DE ENTRENAMIENTO 50 ÉPOCAS.

Model: "model\_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544

fc2 (Dense) (None, 4096) 16781312

---

output (Dense) (None, 3) 12291

---

=====  
Total params: 134,272,835

Trainable params: 12,291

Non-trainable params: 134,260,544

---

Epoch 1/50

62/62 [=====] - 2197s 35s/step - loss: 1.1512 - accuracy: 0.6016 - val  
\_loss: 0.9364 - val\_accuracy: 0.5625

Epoch 2/50

62/62 [=====] - 1580s 25s/step - loss: 0.9590 - accuracy: 0.6535 - val  
\_loss: 1.1854 - val\_accuracy: 0.5678

Epoch 3/50

62/62 [=====] - 1576s 25s/step - loss: 0.8783 - accuracy: 0.6682 - val  
\_loss: 1.1791 - val\_accuracy: 0.5169

Epoch 4/50

62/62 [=====] - 1580s 25s/step - loss: 0.8114 - accuracy: 0.6784 - val  
\_loss: 1.4755 - val\_accuracy: 0.6441

Epoch 5/50

62/62 [=====] - 1574s 25s/step - loss: 0.8632 - accuracy: 0.6819 - val  
\_loss: 0.8663 - val\_accuracy: 0.6525

Epoch 6/50

62/62 [=====] - 1584s 26s/step - loss: 0.7986 - accuracy: 0.7063 - val  
\_loss: 0.9762 - val\_accuracy: 0.5781

Epoch 7/50

62/62 [=====] - 1626s 26s/step - loss: 0.7564 - accuracy: 0.7226 - val  
\_loss: 0.8093 - val\_accuracy: 0.7542

Epoch 8/50

62/62 [=====] - 1612s 26s/step - loss: 0.7960 - accuracy: 0.6997 - val  
\_loss: 1.0162 - val\_accuracy: 0.6356

Epoch 9/50

62/62 [=====] - 1618s 26s/step - loss: 0.8227 - accuracy: 0.7007 - val  
\_loss: 1.6933 - val\_accuracy: 0.5424

Epoch 10/50

62/62 [=====] - 1592s 26s/step - loss: 0.6837 - accuracy: 0.7349 - val  
\_loss: 1.0964 - val\_accuracy: 0.6864

Epoch 11/50

62/62 [=====] - 1571s 25s/step - loss: 0.7926 - accuracy: 0.7121 - val  
\_loss: 1.4058 - val\_accuracy: 0.5625

Epoch 12/50

62/62 [=====] - 1579s 25s/step - loss: 0.8054 - accuracy: 0.7043 - val  
\_loss: 1.4045 - val\_accuracy: 0.6017

Epoch 13/50

62/62 [=====] - 1580s 25s/step - loss: 0.7024 - accuracy: 0.7342 - val  
\_loss: 0.7620 - val\_accuracy: 0.6186

Epoch 14/50

62/62 [=====] - 1586s 26s/step - loss: 0.7241 - accuracy: 0.7188 - val  
\_loss: 1.6645 - val\_accuracy: 0.5424  
Epoch 15/50  
62/62 [=====] - 1563s 25s/step - loss: 0.7169 - accuracy: 0.7310 - val  
\_loss: 0.9985 - val\_accuracy: 0.6271  
Epoch 16/50  
62/62 [=====] - 1585s 26s/step - loss: 0.6940 - accuracy: 0.7317 - val  
\_loss: 0.7390 - val\_accuracy: 0.6484  
Epoch 17/50  
62/62 [=====] - 1637s 26s/step - loss: 0.7521 - accuracy: 0.7221 - val  
\_loss: 0.7960 - val\_accuracy: 0.6017  
Epoch 18/50  
62/62 [=====] - 1569s 25s/step - loss: 0.6298 - accuracy: 0.7602 - val  
\_loss: 0.7442 - val\_accuracy: 0.6441  
Epoch 19/50  
62/62 [=====] - 1566s 25s/step - loss: 0.7070 - accuracy: 0.7231 - val  
\_loss: 0.6590 - val\_accuracy: 0.6525  
Epoch 20/50  
62/62 [=====] - 1559s 25s/step - loss: 0.6647 - accuracy: 0.7434 - val  
\_loss: 0.6625 - val\_accuracy: 0.6864  
Epoch 21/50  
62/62 [=====] - 1569s 25s/step - loss: 0.7325 - accuracy: 0.7348 - val  
\_loss: 0.8278 - val\_accuracy: 0.6641  
Epoch 22/50  
62/62 [=====] - 1570s 25s/step - loss: 0.6801 - accuracy: 0.7424 - val  
\_loss: 0.9158 - val\_accuracy: 0.6610  
Epoch 23/50  
62/62 [=====] - 1566s 25s/step - loss: 0.6517 - accuracy: 0.7459 - val  
\_loss: 0.9133 - val\_accuracy: 0.7034  
Epoch 24/50  
62/62 [=====] - 1572s 25s/step - loss: 0.6331 - accuracy: 0.7545 - val  
\_loss: 1.0136 - val\_accuracy: 0.5508  
Epoch 25/50  
62/62 [=====] - 1552s 25s/step - loss: 0.7994 - accuracy: 0.7136 - val  
\_loss: 1.3811 - val\_accuracy: 0.6441  
Epoch 26/50  
62/62 [=====] - 1581s 26s/step - loss: 0.5960 - accuracy: 0.7762 - val  
\_loss: 0.7312 - val\_accuracy: 0.6641  
Epoch 27/50  
62/62 [=====] - 1570s 25s/step - loss: 0.6858 - accuracy: 0.7408 - val  
\_loss: 1.4950 - val\_accuracy: 0.6186  
Epoch 28/50  
62/62 [=====] - 1589s 26s/step - loss: 0.6367 - accuracy: 0.7626 - val  
\_loss: 1.3470 - val\_accuracy: 0.6610  
Epoch 29/50  
62/62 [=====] - 1570s 25s/step - loss: 0.6299 - accuracy: 0.7572 - val  
\_loss: 0.7581 - val\_accuracy: 0.6780  
Epoch 30/50

62/62 [=====] - 1592s 26s/step - loss: 0.6204 - accuracy: 0.7637 - val  
\_loss: 1.6241 - val\_accuracy: 0.5424  
Epoch 31/50  
62/62 [=====] - 1593s 26s/step - loss: 0.6292 - accuracy: 0.7647 - val  
\_loss: 0.9813 - val\_accuracy: 0.6875  
Epoch 32/50  
62/62 [=====] - 1592s 26s/step - loss: 0.6137 - accuracy: 0.7681 - val  
\_loss: 1.0572 - val\_accuracy: 0.6864  
Epoch 33/50  
62/62 [=====] - 1581s 25s/step - loss: 0.5929 - accuracy: 0.7673 - val  
\_loss: 1.1800 - val\_accuracy: 0.6441  
Epoch 34/50  
62/62 [=====] - 1577s 25s/step - loss: 0.6438 - accuracy: 0.7510 - val  
\_loss: 1.0870 - val\_accuracy: 0.6864  
Epoch 35/50  
62/62 [=====] - 1587s 26s/step - loss: 0.6767 - accuracy: 0.7490 - val  
\_loss: 1.3933 - val\_accuracy: 0.6525  
Epoch 36/50  
62/62 [=====] - 1578s 25s/step - loss: 0.5188 - accuracy: 0.7805 - val  
\_loss: 1.1115 - val\_accuracy: 0.7109  
Epoch 37/50  
62/62 [=====] - 1566s 25s/step - loss: 0.6024 - accuracy: 0.7679 - val  
\_loss: 1.0134 - val\_accuracy: 0.6102  
Epoch 38/50  
62/62 [=====] - 1587s 26s/step - loss: 0.6128 - accuracy: 0.7651 - val  
\_loss: 1.3626 - val\_accuracy: 0.5932  
Epoch 39/50  
62/62 [=====] - 1568s 25s/step - loss: 0.6253 - accuracy: 0.7576 - val  
\_loss: 2.0848 - val\_accuracy: 0.5593  
Epoch 40/50  
62/62 [=====] - 1560s 25s/step - loss: 0.5703 - accuracy: 0.7787 - val  
\_loss: 1.5623 - val\_accuracy: 0.6271  
Epoch 41/50  
62/62 [=====] - 1588s 26s/step - loss: 0.5735 - accuracy: 0.7785 - val  
\_loss: 1.2180 - val\_accuracy: 0.5781  
Epoch 42/50  
62/62 [=====] - 1597s 26s/step - loss: 0.6022 - accuracy: 0.7681 - val  
\_loss: 1.4103 - val\_accuracy: 0.5847  
Epoch 43/50  
62/62 [=====] - 1572s 25s/step - loss: 0.6314 - accuracy: 0.7536 - val  
\_loss: 0.7132 - val\_accuracy: 0.6949  
Epoch 44/50  
62/62 [=====] - 1570s 25s/step - loss: 0.5163 - accuracy: 0.8059 - val  
\_loss: 0.9721 - val\_accuracy: 0.5339  
Epoch 45/50  
62/62 [=====] - 1570s 25s/step - loss: 0.6086 - accuracy: 0.7637 - val  
\_loss: 1.2752 - val\_accuracy: 0.6441  
Epoch 46/50

62/62 [=====] - 1602s 26s/step - loss: 0.6023 - accuracy: 0.7713 - val  
\_loss: 1.8872 - val\_accuracy: 0.5391  
Epoch 47/50  
62/62 [=====] - 1595s 26s/step - loss: 0.5495 - accuracy: 0.7866 - val  
\_loss: 1.1872 - val\_accuracy: 0.6356  
Epoch 48/50  
62/62 [=====] - 1565s 25s/step - loss: 0.5736 - accuracy: 0.7812 - val  
\_loss: 0.9772 - val\_accuracy: 0.5763  
Epoch 49/50  
62/62 [=====] - 1584s 26s/step - loss: 0.5857 - accuracy: 0.7693 - val  
\_loss: 1.1938 - val\_accuracy: 0.5847  
Epoch 50/50  
62/62 [=====] - 1589s 26s/step - loss: 0.5456 - accuracy: 0.7774 - val  
\_loss: 1.1167 - val\_accuracy: 0.5678

**ANEXO C: PRUEBA EN TIEMPO REAL**

**Tesis Robinson Calero, Identificador de Lesiones en la Piel**

Carga una imagen

Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG

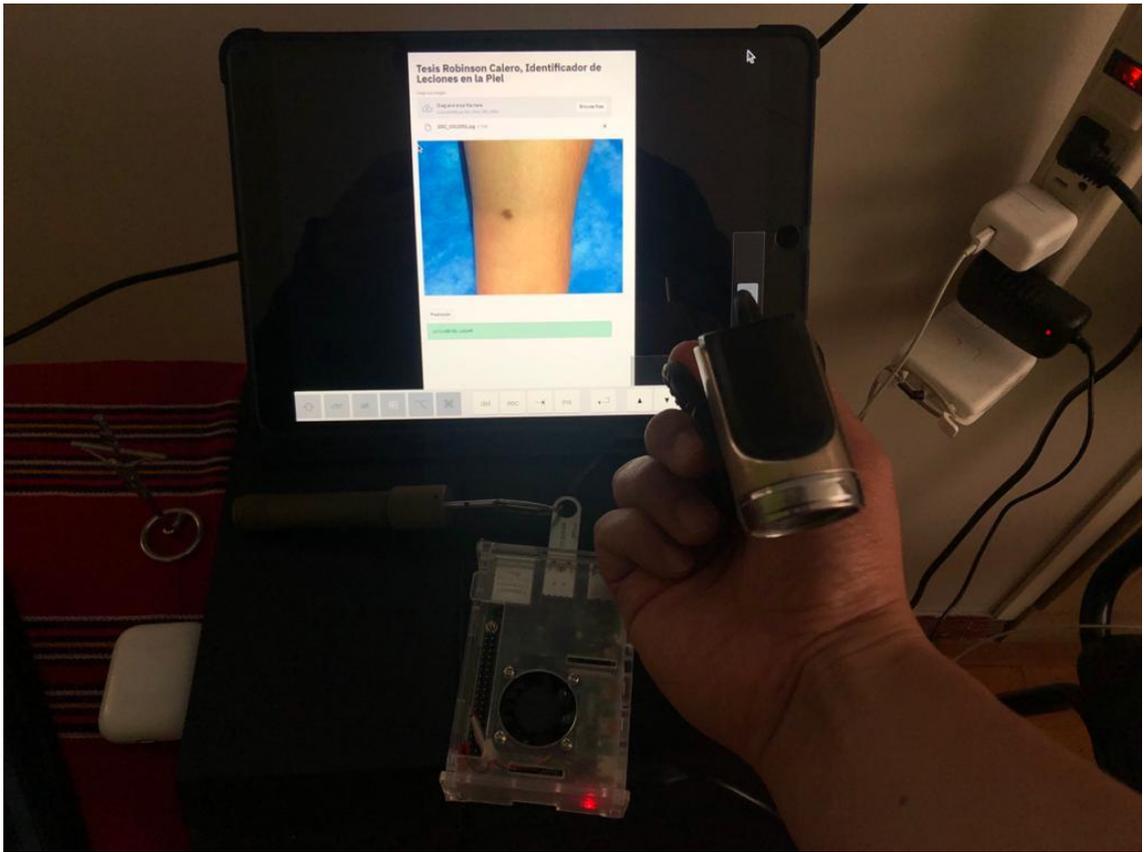
Browse files

ISIC\_0012551.jpg 4.7MB



Predicción

LA CLASE ES: LUNAR





ESCUELA SUPERIOR POLITÉCNICA DE  
CHIMBORAZO  
DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL  
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS  
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 13/12/2021

**INFORMACIÓN DEL AUTOR/A (S)**

**Nombres – Apellidos:** *Robinson Geovanny Calero Inga*

**INFORMACIÓN INSTITUCIONAL**

**Facultad:** *Informática y Electrónica*

**Carrera:** *Electrónica y Automatización*

**Título a optar:** *Ingeniero en Electrónica y Automatización*

**f. Analista de Biblioteca responsable:** *Lic. Luis Caminos Vargas Mgs.*



1005-DBRAI-UTP-2021