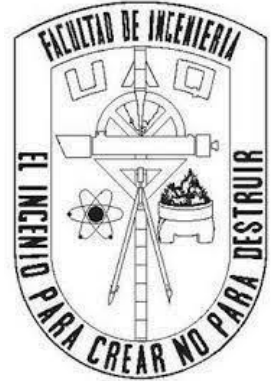


**Universidad Autónoma de Querétaro**  
**Facultad de Ingeniería**



Aplicación móvil para apoyo en detección temprana de melanoma.

**TESIS**

Que como parte de los requisitos para obtener el título de  
**INGENIERO ELECTROMECAÁNICO**

Línea terminal:  
**MECATRÓNICA**

Presenta:  
**Carlos Eugenio Garduño Ramón**

Dirigido por:  
**Dr. Gilberto Alvarado Robles**

San Juan del Río, Querétaro. Octubre 2023



Dirección General de Bibliotecas y Servicios Digitales  
de Información



Aplicación móvil para apoyo en detección temprana  
de melanoma

**por**

Carlos Eugenio Garduño Ramón

se distribuye bajo una [Licencia Creative Commons  
Atribución-NoComercial-SinDerivadas 4.0  
Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

**Clave RI:** IGLIN-250624



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO  
FACULTAD DE INGENIERÍA  
CAMPUS SAN JUAN DEL RÍO



aplicación móvil para apoyo en detección temprana de melanoma

TESIS

Como parte de los requisitos para obtener el título de

**INGENIERO ELECTROMECAÁNICO**

Línea Terminal

**MECATRÓNICA**

Presenta

**Carlos Eugenio Garduño Ramón**

Dirigido por:

**Dr. Gilberto Alvarado Robles**

**SINODALES:**

**Dr. Gilberto Alvarado Robles (19045)** \_\_\_\_\_  
**Presidente**

**Dr. Emmanuel Reséndiz Ochoa (17415)** \_\_\_\_\_  
**Secretario**

**Dr. Irving Armando Cruz Albarrán (15483)** \_\_\_\_\_  
**Vocal**

**Dr. Luis Alberto Morales Hernández (6284)** \_\_\_\_\_  
**Suplente**

San Juan del Río, Qro., octubre 2023.

## RESUMEN

Para la detección de melanomas, hoy en día se cuentan con herramientas de índole ingenieril que pueden ayudar a detectarlos y, sobre todo, que pueden llegar a estar al alcance de muchas personas. Para ello, se sabe que hoy en día se cuentan con dispositivos móviles que permitan tomar fotos y tener conexión a internet, por lo que se abre un gran panorama de posibilidades. Para poder detectar un melanoma, una vez que se tiene una imagen, es necesario hacer uso de un procesamiento de imágenes y Machine Learning para analizar la imagen y poder clasificar si es un posible caso de melanoma. Sin embargo, cuando se habla de imágenes, hay distintos factores que pueden alterar los resultados; entre ellos, se tienen factores como la iluminación, la cual es difícil de controlar, al igual que la calidad de la imagen con la que se cuenta. El objetivo de este trabajo es crear una aplicación móvil para el sistema operativo Android. Esta aplicación permitirá a los usuarios seleccionar imágenes de una base de datos utilizada en el desafío SIIM-ISIC melanoma classification, el cual es de acceso libre, que comprende casos malignos y benignos. Además, se emplea el algoritmo ganador del desafío denominado "Identifying Melanoma Images using EfficientNet Ensemble." (Ha et al., 2020). En la aplicación, se aplican filtros de mejora a las imágenes seleccionadas antes de enviarlas al servicio de Firebase Storage para posteriormente cargarla en el modelo. Este modelo se ejecuta en la plataforma de Colaboratory. Posteriormente, se obtiene los pesos correspondientes a la imagen y se envía el valor resultante a la aplicación móvil para llevar a cabo la clasificación correspondiente. Una vez realizada la clasificación, en la aplicación se muestra el resultado de la clasificación de la imagen como un posible caso maligno o benigno. Siempre se enfatiza la importancia de que el usuario consulte a un especialista médico lo más pronto posible, independientemente del resultado obtenido, con el fin de salvaguardar su salud y cumplir con los lineamientos establecidos por la Norma Oficial Mexicana NOM-241-SSA1-2021. En el presente trabajo se muestra que hoy en día las herramientas ingenieriles tienen un mayor alcance y que pueden ser utilizadas como alternativas en temas relacionados a la salud, de hecho, se puede utilizar como un método alternativo de diagnóstico en donde se prevé que sea accesible, escalable y reconfigurable. en este caso se utilizar herramientas ingenieriles de hardware y software para poder realizar una clasificación de imágenes para diagnosticar un posible caso de melanoma.

## ABSTRACT

For the detection of melanomas, today there are engineering tools that can help detect them and, above all, that can be within the reach of many people. To do this, it is known that today there are mobile devices that allow taking photos and having an internet connection, which opens up a wide range of possibilities. In order to detect melanoma, once you have an image, it is necessary to use image processing and Machine Learning to analyze the image and be able to classify whether it is a possible case of melanoma. However, when talking about images, there are different factors that can alter the results; Among them, there are factors such as lighting, which is difficult to control, as well as the quality of the image available. The objective of this work is to create a mobile application for the Android operating system. This application will allow users to select images from a database used in the SIIM-ISIC melanoma classification challenge which is free access, which comprises malignant and benign cases. In addition, the challenge-winning algorithm called "Identifying Melanoma Images using EfficientNet Ensemble" is used. (Ha et al., 2020). In the app, enhancement filters are applied to selected images before they are sent to the Firebase Storage service for later uploading to the model. This model runs on the Colaboratory platform. Subsequently, the weights corresponding to the image are obtained and the resulting value is sent to the mobile application to carry out the corresponding classification. Once the classification has been carried out, the application shows the result of classifying the image as a possible malignant or benign case. The importance of the user consulting a medical specialist as soon as possible is always emphasized, regardless of the result obtained, in order to safeguard her health and comply with the guidelines established by the Official Mexican Standard NOM-241-SSA1- 2021. This work shows that nowadays engineering tools have a greater scope and that they can be used as alternatives in health-related issues. In fact, they can be used as an alternative diagnostic method where it is expected to be accessible, scalable and reconfigurable. In this case, hardware and software engineering tools will be used to perform image classification to diagnose a possible case of melanoma.

# Dedicatoria

*Dedico este trabajo a mis  
padres, mi hermano,  
amigos y seres queridos.  
Por ser mi mayor motivación en la vida.*

# Agradecimientos

Agradezco a mi mamá Cristina y a mi papá Eugenio por todo el sacrificio que han hecho y el amor incondicional a nuestra familia, siendo mi inspiración en cada momento de mi vida.

Agradezco a mi hermano Marco Antonio, mi mejor amigo, por todo su apoyo y amor incondicional, siendo una gran inspiración en mi crecimiento como persona.

A mi director de tesis, Dr. Gilberto Alvarado Robles por siempre tener la disposición de ayudar a lo largo de este trabajo.

Al Dr. Irving Armando Cruz Albarrán y sinodales por su retroalimentación y apoyo durante el desarrollo de este trabajo.

A los maestros y coordinación de la Facultad de Ingeniería por siempre apoyar a los estudiantes y guiarlos por el camino correcto.

A mis compañeros de generación por dejarme tantas enseñanzas sobre la vida y volviéndose buenos amigos, sin ellos el camino no habría sido el mismo.

# TABLA DE CONTENIDO

RESUMEN.....	i
SUMMARY.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTOS.....	iv
TABLA DE CONTENIDO.....	v
ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS.....	ix
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>1. introducción.....</b>	<b>1</b>
<b>1.1. Estado del arte.....</b>	<b>3</b>
<b>1.2. Normas.....</b>	<b>6</b>
<b>1.3. Descripción del problema.....</b>	<b>7</b>
<b>1.4. Justificación.....</b>	<b>8</b>
<b>1.5. Objetivos.....</b>	<b>9</b>
<b>1.5.1. Objetivo general.....</b>	<b>9</b>
<b>1.5.2. Objetivos particulares.....</b>	<b>9</b>
<b>1.6.Planteamiento general.....</b>	<b>10</b>
<b>CAPÍTULO 2.....</b>	<b>13</b>
<b>2. Revisión de literatura.....</b>	<b>13</b>
<b>2.1. Imagen.....</b>	<b>13</b>
<b>2.1.1. Imagen digital.....</b>	<b>14</b>
<b>2.1.2. Espectro electromagnético.....</b>	<b>15</b>
<b>2.1.3. Color.....</b>	<b>17</b>



<b>2.2. Melanoma</b> .....	18
2.2.1. Melanoma a nivel mundial.....	20
2.2.2. Melanoma a nivel nacional (México).....	22
2.2.3. Métodos de diagnóstico.....	24
2.2.4. Herramientas de detección alternas.....	25
2.2.5. Archivos SIIM-ISIC.....	26
<b>2.3. Android Studio</b> .....	27
<b>2.4. Aprendizaje supervisado y no supervisado</b> .....	28
<b>2.5. Red neuronal convolucional</b> .....	30
2.5.1. Pesos.....	31
2.5.2. Sesgos.....	32
<b>2.6. procesamiento de imágenes</b> .....	33
<b>2.7. Softmax</b> .....	35
<b>2.8. Histograma de ecualización</b> .....	36
<b>2.9. Ajuste de contraste en imágenes</b> .....	42
<b>CAPÍTULO 3</b> .....	44
<b>3. Metodología</b> .....	44
3.1. Descargar base de datos.....	45
3.2. Obtención de "Identifying Melanoma Images using EfficientNet Ensemble.".....	45
3.3. Emplear el modelo en Colaboratory.....	46
3.4. Entrenar el modelo en Colaboratory.....	49
3.5. Creación de la aplicación.....	50
3.6. Adquisición y almacenamiento de imágenes.....	61

<b>3.7. Enlazar Firebase Storage/Realtime Database con la app y Colaboratory</b>	<b>65</b>
3.7.1. Firebase Storage con app	65
3.7.2. Realtime Database con app	69
3.7.3. Colaboratory con Firebase	71
<b>3.8. Procesamiento de imágenes</b>	<b>72</b>
<b>3.9. Clasificador</b>	<b>75</b>
<b>CAPÍTULO 4</b>	<b>78</b>
4. Resultados	78
<b>CAPÍTULO 5</b>	<b>87</b>
5. Conclusiones	87
<b>REFERENCIAS</b>	<b>89</b>

## ÍNDICE DE TABLAS

Tabla 1: Tasa de incidencia a nivel global de melanoma. (Observatorio Mundial del Cáncer, 2022).

Tabla 2: Mortalidad global de melanoma. (Observatorio Mundial del Cáncer, 2022).

Tabla 3: Valores de la sub imagen en escala de grises de 8 bits. (Histogram equalization, 2023).

Tabla 4: Valores ordenados y contados de las veces que se repiten. (Histogram equalization, 2023).

Tabla 5: Distribución acumulada de los píxeles. (Histogram equalization, 2023).

## ÍNDICE DE FIGURAS

Figura 1.1: Diagrama propuesto para el desarrollo del proyecto (autoría propia).

Figura 2.1: Representación visual de una pintura. (Milenio Digital, 2021).

Figura 2.2: Imagen tomada por el Dr. David R. Pickens usando herramientas tecnológicas. (Gonzalez y Woods, 2018).

Figura 2.3: Espectro electromagnético. (Gonzalez y Woods, 2018).

Figura 2.4: Colores primarios y secundarios. a) los colores aditivos. b) Los colores sustractivos. (Szeliski, 2010).

Figura 2.5: Representación de los componentes RGB (Cheonshik et al. 2021).

Figura 2.6: Capas de la piel. (Mitchell et al., 2020).

Figura 2.7: Cronograma de mortalidad por melanoma en México. (Hernández et al., 2018).

Figura 2.8: Biopsia en sacabocados. (Mayo Clinic Family Health Book).

Figura 2.9: Uso de herramientas tecnológicas en el sector de la salud. (Scudellari, 2020).

Figura 2.10: Aprendizaje supervisado y no supervisado (autoría propia).

Figura 2.11: Distribución de pesos en una red neuronal (autoría propia).

Figura 2.12: Diferencias de tener un sesgo alto y bajo con una baja varianza (autoría propia).

Figura 2.13: Imagen con distinta intensidad junto a su histograma respectivamente (Wainschenker et al., 2011).

Figura 2.14: Normalización de los píxeles, usando la función softmax. (Ogun, 2020).

Figura 2.15: Histogramas de una imagen antes y después de la ecualización (Kiliç, 2021).

Figura 2.16: Sub imagen de 8 x 8 que está en escala de grises de 8 bits. (Gonzalez y Woods, 2018).

Figura 2.17: Comparativa de la imagen una vez que se aplica el filtro de histograma de ecualización. (Gonzalez y Woods, 2018).

Figura 2.18: Efecto contraste condicional. (Sonka et al., 2013).

Figura 2.19: En la imagen de alto contraste, las zonas claras se ven más brillantes y las sombras se ven más oscuras. (MathWorks, 2023).

Figura 3.1: Metodología propuesta (autoría propia).

Figura 3.2: Tabla de mejores modelos desarrollados. (Kaggle, 2020).

Figura 3.3: Pesos obtenido con el modelo. (autoría propia).

Figura 3.4: Versión utilizada de Android Studio. (autoría propia).

Figura 3.5: API LEVEL mínimo para la versión y el porcentaje que abarca la versión 5.0. (autoría propia).

Figura 3.6: Template utilizado. (autoría propia).

Figura 3.7: Diseño de la página principal de la aplicación. (autoría propia).

Figura 3.8: Nombre de los componentes utilizados. (autoría propia).

Figura 3.9: Logo creado para la aplicación. (autoría propia).

Figura 3.10: Ubicación de Resource Manager. (autoría propia).

Figura 3.11: Ventana emergente de Resource Manager. (autoría propia).

Figura 3.12: Opciones de personalización para el logo. (autoría propia).

Figura 3.13: Ubicación del archivo strings.xml. (autoría propia).

Figura 3.14: La línea #2 es en donde se cambia el nombre de la app. (autoría propia).

Figura 3.15: Icono de Device Manager. (autoría propia).

Figura 3.16: Ventana para crear un emulador de dispositivo. (autoría propia).

Figura 3.17: Permisos del archivo “AndroidManifest”. (autoría propia).

Figura 3.18: Visualización de los proyectos de Firebase. (autoría propia).

Figura 3.19: Enlazar mediante el navegador. (autoría propia).

Figura 3.20: Enlazar Firebase desde Android Studio. (autoría propia).

Figura 3.21: Datos de la app de Android en Firebase. (autoría propia).

Figura 3.22: Dependencias agregadas al hacer el enlace con Firebase. (autoría propia).

Figura 3.23: Código que se encuentra en la pestaña de reglas dentro de Storage. (autoría propia).

Figura 3.24: Sincronizar una vez agregada en dependencias. (autoría propia).

Figura 3.25: Con estas líneas se tendrá acceso a la base de datos. (autoría propia).

Figura 3.26: Código de Realtime Database. (autoría propia).

Figura 3.27: Secuencia a seguir para crear la carpeta en app. (autoría propia).

Figura 3.28: Línea de código para verificar que OpenCV se pudo cargar. (autoría propia).

Figura 3.29: Son los casos benignos, de arriba abajo tenemos: imágenes normales, con ajuste de contraste y por último histograma de ecualización. (autoría propia).

Figura 3.30: Son los casos malignos, de arriba a abajo tenemos: imágenes normales, con ajuste de contraste y por último histograma de ecualización. (autoría propia).

Figura 4.1: MelaUAQ en el dispositivo móvil. (autoría propia).

Figura 4.2: Interfaz de la aplicación. (autoría propia).

Figura 4.3: Selección de la imagen a procesar. (autoría propia).

Figura 4.4: Imagen desplegada en el ViewImage. (autoría propia).

Figura 4.5: Mensaje mostrado en la consola en Colaboratory. (autoría propia).

Figura 4.6: Ejecución de Colaboratory en el móvil. (autoría propia).

Figura 4.7: Interfaz de Storage, donde se guardan las imágenes enviadas desde la app. (autoría propia).

Figura 4.8: Valor enviado a Realtime Database. (autoría propia).

Figura 4.9: Resultado desplegado en la app. (autoría propia).

Figura 4.10: Gráfico de los pesos solo con el modelo. (autoría propia).

Figura 4.11: Gráfico de los pesos aplicando histograma de ecualización. (autoría propia).

Figura 4.12: Gráfico de los pesos aplicando el filtro de ajuste de contraste (autoría propia).

Figura 4.13: Gráfico de dispersión de valores malignos y benignos. (autoría propia).

# 1. INTRODUCCIÓN

El cáncer es una de las causas principales de muertes en el mundo; dentro de este se conocen distintos tipos, lo que lo convierte en una mutación celular sumamente compleja y diversa, es decir, esto hace que cada uno de los casos presentes entre la población oncológica se presente de manera diferente, lo cual responde a distintos factores.

Gracias a los avances científico-tecnológicos se han logrado identificar aquellos patrones y factores que hacen a una persona más propensa a padecer algún tipo de cáncer; dentro de estos se pueden ver los que corresponden a la genética, el estilo de vida, e incluso cuestiones ambientales del lugar en el que se reside.

En términos generales, al hablar de cáncer se hace referencia a la multiplicación rápida de células de manera anormal que, se extienden más allá de sus límites y pueden dañar a partes adyacentes del cuerpo. (Mitchell et al., 2020).

## *El cáncer de piel.*

Previo a hablar de acerca del cáncer de piel, resulta necesario concretar que la piel es el órgano más grande del cuerpo, la cual se encuentra conformada por distintas capas; siendo las principales la epidermis (capa externa) y dermis (capa interna), donde el cáncer puede hacer aparecer por una exposición prolongada al sol. La gran mayoría de los cánceres de piel ocurren en la epidermis, la cual se encuentra conformada por 3 tipos principales de células: células basales, células escamosas y melanocitos. (Mitchell et al., 2020).

En cuanto a los dos primeros refiere, estos poseen una probabilidad alta de curación siempre y cuando la detección y el tratamiento se realice a tiempo, sin embargo, es preciso dejar claro que se pueden presentar deformaciones en la piel y que el tratamiento resulta costoso. Por otra parte, el cáncer que afecta a los melanocitos o también conocido como melanoma, es tipo de cáncer de piel menos frecuente, pero al mismo tiempo es el más peligroso, puesto que es más probable que afecte a demás partes del cuerpo. El presente proyecto de tesis se encuentra enfocado específicamente en este último tipo de cáncer de piel.



### *El cáncer de piel a nivel mundial.*

El cáncer de piel es uno de los tumores con mayor incidencia en el ser humano, prueba de ello se encuentra que, en 2020, durante ese año, según los registros elaborados, se presentaron alrededor de 10 millones de defunciones a nivel mundial (Ferlay et al., 2021), dónde casi una de cada seis muertes fue a causa de algún tipo de cáncer, de manera concreta el cáncer de melanoma se posicionó en el lugar 17° y en el puesto 22° de defunciones hasta el año del 2020 según la organización internacional de la salud. (Globocan, 2020).

### *El cáncer de piel a nivel nacional.*

Los registros correspondientes al cáncer de piel en el país datan del año 1998 hasta 2016, posicionándolo en el lugar 20° de las causas de muerte a nivel nacional. La información recopilada muestra que la tendencia que se a un aumento de los casos, es decir, a lo largo de los años se han venido duplicando las defunciones por esta causa en México, lo cual al mismo tiempo se traduce en que el cáncer de piel cada vez tiene un mayor impacto y presencia dentro de la población.

Al año en México se diagnostican alrededor de mil casos anuales de cáncer de piel, de los cuales alrededor de 700 casos se convierten en defunciones. En términos estadísticos esos datos corresponden a porcentajes muy altos, sin embargo, al compararlo con países como Estados Unidos el cual es uno de los países con el mayor índice de cáncer de piel se encuentra que se presentan alrededor de 10 veces más muertes por la misma causa que en nuestro país, dicho de otra forma, en dicho país se tienen alrededor de 7000 defunciones anuales.

Aquello que llama la atención sobre el caso mexicano es que con el tiempo las cifras nacionales declaran un aumento considerable en la presencia de casos de cáncer de piel, donde la gran mayoría ya se encuentran en una etapa avanzada, es por ello que este fenómeno debe ser tratado con la importancia que merece, debido a que la mayoría de los casos es prevenible y curable cuando es tratado a tiempo. (Hernández et al., 2018).

### *Métodos de diagnóstico.*

Hay distintos métodos diagnósticos, entre ellos se encuentran tres principales: la biopsia, la tomografía y la dermatoscopia.

La primera consiste en una extracción de células o tejido para posteriormente ser examinada por un patólogo, en cuanto a este método, presenta el problema de que es invasivo y puede dejar marca en la piel. El segundo procedimiento involucra el uso de rayos X, en este se toman imágenes desde diferentes ángulos donde seguido de ello una computadora las combina en una tridimensional detallada que permite ver la presencia de anomalías, este método puede ser peligroso si se usa demasiadas veces. Por último, la dermatoscopia, permite observar áreas de la piel con mayor claridad, para el cual se emplea un aparato que posee lentes con ampliación y una fuente de luz que es colocado cerca de la piel, este método es el menos usado por especialistas en base a su efectividad, por ello el uso de este método es empleado principalmente para evaluar y determinar si es necesario o recomendable recurrir a la biopsia. (Ribas et al., 2019)

Ante el panorama antes señalado, con el avance de la tecnología se han logrado encontrar algunos otros métodos que permiten diagnosticar los melanomas, como herramientas alternas se utilizan cámaras termográficas, con rayos ultravioleta o con cámaras. Es importante resaltar que el objetivo es tener herramientas que puedan ayudar a la detección ante alguna enfermedad, además de que estos sean más accesibles que incluso ir a revisión por un especialista.

## **1.1 ESTADO DEL ARTE**

En torno al trabajo realizado, dentro del procesamiento de imágenes y de herramientas inteligentes hay presencia de una gran variedad de ellos que ayudan en la detección del melanoma. Dentro de las siguientes líneas se menciona algunos de los últimos trabajos desarrollados en torno al tema, en donde tenemos que dejar en claro que muchos trabajos consisten en desarrollar el algoritmo del modelo para la clasificación, mientras que son muy pocos los trabajos en donde se lleva ese algoritmo a una aplicación móvil, pero de igual manera es importante mencionar esos trabajos y resaltar que se hace hincapié en la importancia de la tecnología para resolver problemas del ámbito de la salud, así como el aumento de su uso en la actualidad como herramienta para el diagnóstico o dentro del campo de la investigación.

Como primer antecedente se encuentra el trabajo utilizado para el desarrollo del proyecto de tesis de Qishen Ha et al (2020), el cual supone un punto clave para el desarrollo del presente proyecto puesto que su propuesta formula la solución ganadora al desafío de clasificación de melanoma llevado a cabo por SIIM-ISIC (Sociedad de informática de imágenes en Medicina y la Colaboración internacional de imágenes de la piel) (Zawacki et al., 2020). La propuesta se basa en un conjunto de modelos de red neuronal convolucional (CNN) que posee diferentes redes troncales y tamaños de entrada, en estos la mayoría son modelos de solo imagen, mientras que por otra parte algunos de ellos utilizan metadatos a nivel de imagen y a nivel de paciente. La efectividad lograda en el proyecto fue de un 96.58%, motivo principal por lo que fue tomada la decisión de emplear dicha propuesta de trabajo. (Ha et al., 2020).

Por su parte en el trabajo de Rivera y Alexis (2020), el sistema de detección de melanoma que se empleó funciona utilizando redes neuronales convolucionales, en este se hace hincapié en la importancia de la inteligencia artificial en el área de la tecnología puesto que esta contribuye como una alternativa para la detección de enfermedades sin que el proceso resulte demasiado invasivo para las personas. (Chicaiza y Alexis, 2020).

Meyer et al (2020), trabajaron en el desarrollo de una aplicación móvil usada para la detección automática de melanomas, este trabajo ejemplificar de manera más clara aquello que se busca desarrollar en el presente proyecto. Se obtuvo un prototipo para una aplicación móvil en Android, la cual se basa en técnicas de inteligencia computacional que determina mediante imágenes si una lesión en la piel es benigna o melanoma, para conseguirlo se implementaron clasificadores de imágenes basados en redes neuronales para la detección de lesiones. Resulta posible llevarlo a cabo gracias a que se pose una base de datos con gran cantidad de imágenes que existen en la actualidad, las imágenes suministradas por los médicos sirven para entrenar más modelos y mejorarlos aplicando diferentes técnicas como Data Augmentation. (Meyer et al., 2020).

Un equipo de investigadores, Quintanilla et al. (2022), desarrollaron dentro de su trabajo la propuesta de una aplicación de aprendizaje automático con visión artificial que sirve como herramienta de apoyo para la detección de melanomas benignos y malignos, con su trabajo se puede ir dando cuenta de que cada vez se busca más la implementación de tecnologías y herramientas ingenieriles para la elaboración de diagnósticos en temas relacionados a la salud, por lo que es importante mencionar resaltar como es que se han desarrollado distintos trabajos

para la detección de melanoma, sin embargo muchos de ellos van enfocados en otros medios y con el uso de otras herramientas. (Quintanilla et al., 2022)

En otro estudio, Gómez (2021), trabajaron e implementaron un método de segmentación de imágenes dermatoscópicas para detección de melanomas basado en Watershed, en donde parten de una aplicación existente para la segmentación del área glotal en imágenes tomadas de videos estroboscópicos, la cual fue desarrollada por el grupo de Aplicaciones Multimedia y Acústica (GAMMA) de la Universidad Politécnica de Madrid. Para este trabajo, se hace uso de Visual Studio y del lenguaje de programación C#, además de utilizar la base de datos ISIC. (Gómez, 2021).

Tenemos un claro ejemplo de un trabajo realizado para el Máster Universitario en Inteligencia Artificial, en donde Acero y Paul (2021), desarrollaron una aplicación móvil para detección de melanoma y otras ocho categorías. Hicieron uso de distintas redes neuronales como ResNet50, ResNet101, VGG, InceptionV3 y EfficientNet, para poder realizar experimentos y poder evaluar los parámetros como el tamaño, balanceo de pesos, optimizadores, entre otros. Esto con el fin de obtener la mayor cantidad de falsos negativos y la mejor exactitud. (Acero y Paul, 2021).

El trabajo desarrollado por López (2020), habla sobre una solución desarrollada para el seguimiento de los lunares de manera sencilla y cómoda, en donde se hace uso de visión por computadora. El usuario necesita tomarse fotos de los lunares o manchas en la piel en un periodo de tiempo para posteriormente la aplicación pueda analizar cada una de esas fotografías para extraer de ellas los cambios que se haya producido, este trabajo es desarrollado en la universidad de Málaga, España. (López, 2020).

En el proyecto de tesis de Canchachí y Mendoza (2020), realizaron una aplicación móvil inteligente para mejorar la eficacia de la detección de melanoma en los pacientes del centro de fototerapia en Trujillo, Perú. Utilizaron la base de datos proporcionada por el centro, en donde utilizaron la metodología de trabajo Mobile-D, cuyas fases son: Exploración, iniciación, producción, estabilización y pruebas de software. Una parte para destacar de este trabajo es que llevaron este tipo de herramientas al ámbito de la salud, puesto que realizaron este trabajo en colaboración con el centro de fototerapia, y esto es un claro ejemplo de que estas herramientas pueden ser de mucha utilidad para los especialistas. (Canchachí y Mendoza, 2020).

En el caso de Matas (2018), realizó un diseño de una aplicación para poder hacer una clasificación de imágenes dermatoscópicas para poder verificar si tienen nevus (son lesiones cutáneas benignas muy frecuentes), para ello se realizó primeramente un estudio previo de la enfermedad y de las técnicas de detección y diagnóstico de ésta. (Matas, 2018).

Por último, se encuentra la Evaluación Biopsicosocial del dolor con imágenes térmicas de la expresión facial en sobrevivientes de Cáncer de Mama, dicho trabajo fue realizado por profesores de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro en el Campus San Juan del Río. Se trabajó con procesamiento de imágenes incursionando en el área de la salud, lo cual es importante puesto que muchos trabajos enfocados en el área de la salud como alternativas son realizados por personas externas a esta, además se utilizó el procesamiento de imágenes lo cual hace fundamental que sea retomado en el presente trabajo de tesis puesto que se encuentra enfocada siguiendo esos rubros. (Rodríguez et al., 2018).

La gran diferencia del proyecto que se tiene con los trabajos mencionados, es que se lleva un modelo entrenado con efectividad muy alta a una aplicación móvil, la cual puede ejecutarse en la gran mayoría de dispositivos Android, esto facilita la parte de los recursos computacionales, ya que se usa la plataforma de Colaboratory. Para mejorar los resultados se usan filtros de mejora de procesamiento de imágenes para tener una brecha mayor entre los pesos obtenidos de casos benignos y malignos, logrando así una mejor clasificación.

## **1.2 NORMAS**

Los nuevos software y aplicaciones móviles catalogados como soluciones de Salud digital deben contar con regulaciones para evitar desinformación y promover uso de tecnologías con sustento científico. En México COFEPRIS (Comisión Federal para la Protección contra Riesgos Sanitarios) ha creado un grupo de trabajo en regulación innovadora para SaMD (Software as a Medical Device), donde el objetivo es la evaluación de la NOM-241-SSA1-2021. (Diario oficial de la Federación, 2021).

A pesar de que el proyecto no esté a disposición de cualquier persona, sino que se quede solo para un uso dentro de la Facultad de Ingeniería, al final no cambia el hecho de que la base de datos es de casos confirmados de personas, por lo que una aplicación que se quiera llevar más allá, necesitará pasar por normas al catalogarse como una aplicación referente a la salud.

### **1.3 DESCRIPCIÓN DEL PROBLEMA**

Con la literatura revisada de los distintos trabajos y proyectos realizados, se descubrieron múltiples áreas para aportar, entre ellas el realizar una aplicación móvil para el sistema operativo de Android, la cual nos ayude a diagnosticar un posible caso de melanoma en donde se hará uso de la base de datos proporcionada por ISIC (Colaboración internacional de imágenes de piel) en donde tenemos como principal objetivo el poder hacer una aplicación móvil para la detección de melanomas, en donde se busca que sea accesible, escalable y reconfigurable. En base a la literatura revisa, se tienen 3 vertientes importantes a indagar, las cuales son las siguientes: científico-tecnológica, económico y social.

#### *Científico-tecnológica*

Si bien hay distintos trabajos con melanoma y la detección de este con la ayuda de herramientas ingenieriles, es una realidad que muchas veces estos trabajos se realizan para participar en concursos que se realizan gracias a distintos institutos enfocados en fomentar el uso de las tecnologías para estudiar el área de la medicina, pero es muy rara la ocasión en que se busca llevar esos trabajos a una aplicación móvil para el uso de los distintos usuarios o instituciones, para poder ayudar a diagnosticar un posible caso de melanoma.

#### *Económico*

En las distintas tiendas de aplicaciones de smartphones, se encuentran distintas aplicaciones que tocan el tema del diagnóstico para el cáncer de piel, el problema de muchas de ellas es que no se encuentran disponibles en nuestra región, además de que muchas de ellas terminan siendo caras y por ello muchas personas deciden no considerarlas.

#### *Social*

La situación en México es un tanto especial y esto es principalmente porque no se le da la importancia que merece a las cosas de la salud y mucho menos a la parte de la piel. En muchas ocasiones una persona se enferma y deciden dejarlo pasar, prefieren que el tiempo termine por curarles los síntomas o las enfermedades y por otro lado tenemos el factor de que muchas personas no tienen la oportunidad de ir con un especialista, esto hablando ya en particular en el caso de la piel.

## 1.4 JUSTIFICACIÓN

El proyecto encuentra motivos en la proporción de soluciones a las áreas de oportunidad planteadas en el apartado anterior donde se realiza la descripción del problema. A continuación, se plantean las respuestas y soluciones encontradas para cada una de las categorías.

### *Científico-tecnológica*

Lo que se busca dentro de este sector es realizar una aplicación móvil para el sistema operativo de Android empleando el algoritmo de acceso libre para el desarrollo científico-tecnológico, con el nombre de "Identifying Melanoma Images using EfficientNet Ensemble." (Ha et al., 2020), en donde el usuario pueda procesar una imagen para poder tener un diagnóstico temprano. Además, se busca que la aplicación sea compatible con la mayoría de los dispositivos actuales, es decir, con dispositivos que poseen una versión de Android superior a la 5.0. La aplicación será únicamente una herramienta alterna que contribuye a la realización de un diagnóstico, puesto que esta no buscará remplazar la opinión de un especialista. Por el momento se piensa que la aplicación sea de utilidad en la Facultad de Ingeniería del Campus San Juan del Río, con la finalidad continuar con el proceso de mejora gracias al conocimiento generado y proporcionado en la Línea Terminal de Mecatrónica. Las materias que aportan a la realización del proyecto son programación y sistemas de visión por computadora.

### *Económico*

Una de las principales características que se consideró para el desarrollo de la aplicación ataca algunos de los principales factores que impiden el acceso a sus homologas, es lo referente a la disponibilidad dentro de la región por lo que poder instalarla y usarla no representará problema alguno; al mismo tiempo se ha formulado bajo la premisa del acceso a los servicios de salud inmediatos, dicho de otra forma, ha sido pensada para que pueda llegar a lugares donde no se puede acceder a la atención médica, siguiendo las normas correspondientes.

### *Social*

Tomando en cuenta el escenario al que se enfrenta, resulta necesario que el diseño de la aplicación sea llamativo para el usuario, además de que la aplicación sea sencilla de usar, de igual manera dejar en claro la recomendación de acercarse con un especialista, lo que no resta valor a su función puesto que al final se busca que sea una herramienta alternativa de gran utilidad ya que se puede tener acceso a ella e incluso convertirse en una de las herramientas alternativas para los especialistas.

## **1.5 OBJETIVOS**

### **1.5.1 Objetivo general**

Desarrollar e implementar una aplicación móvil Android, utilizando una base de datos para clasificar imágenes procesadas, la cual es de acceso libre, (Rotemberg et al., 2021), mediante técnicas de mejora, con la finalidad de estudiar casos sospechosos de melanomas que sirva de apoyo en su diagnóstico, basándose en el uso del modelo "Identifying Melanoma Images using EfficientNet Ensemble." (Ha et al., 2020), en donde se hace uso de la base de datos proporcionada por ISIC (Colaboración internacional de imágenes de piel), utilizada en el desafío.

### **1.5.2 Objetivos particulares**

1. Realizar una investigación de proyectos relacionados al estudio de melanomas mediante una revisión de literatura para establecer una comparativa con el proyecto a realizar y buscar las normas relacionadas a este.
2. Desarrollar una aplicación usando Android Studio para adquirir, almacenar, procesar y clasificar imágenes de posibles casos de melanomas
3. Aplicar filtros de mejora a las imágenes adquiridas utilizando OpenCV previo a su clasificación con el algoritmo inteligente ganador del desafío SIIM-ISIC desarrollado por Qishen Ha et al.
4. Clasificar las imágenes adquiridas para determinar el porcentaje de posible caso de melanoma, esto mediante el despliegue en la aplicación Android del modelo inteligente utilizado en la tarea de clasificación de melanomas
5. Comparar los resultados entregados por la aplicación Android con imágenes mejoradas y no mejoradas para identificar áreas de oportunidad del sistema.



## 1.6 PLANTEAMIENTO GENERAL

Como se ilustra en la Figura 1.1, el desarrollo del proyecto consta de una serie de pasos que permiten guiarlo. A continuación, se explican cada uno de ellos.

1. Hacer una revisión en literatura de trabajos relacionados para poder tener un contexto actual de lo que se ha desarrollado respecto al cáncer de piel y la creación de una aplicación móvil para la detección de un melanoma.
2. Posteriormente, teniendo en cuenta el contexto actual y el punto en el que se encuentra el trabajo desarrollado por Qishen Ha et al, se decide plantear el aporte que se va a realizar al modelo ya presentado, además de planificar la creación de una aplicación móvil. Una vez explicado eso, se procede a revisar el modelo y descargar la base de datos, la cual es de acceso libre, proporcionada por ISIC (Colaboración internacional de imágenes de la piel).
3. Llevar el algoritmo inteligente a Colaboratory haciendo los cambios correspondientes de las rutas para tener acceso a modelos y a la base de datos.
4. Entrenar el algoritmo inteligente haciendo uso de la base de datos utilizada y reuniendo distintos resultados de imágenes benignas y malignas. Esto con el fin de poder visualizar los resultados obtenidos y comprender los pesos obtenidos para la clasificación dada.
5. Creación de la aplicación móvil, en donde se deja en claro desde un inicio la versión mínima de Android en la cual va a funcionar será la versión 5.0, además de definir el lenguaje a utilizar, en este caso se usa Java.
6. Ya teniendo la aplicación en un estado base, se procede a programar la misma aplicación para poder acceder a la galería del dispositivo, seleccionar y enviar la imagen al servicio de Firebase y poder cargar la imagen en el modelo empleado en Colaboratory.
7. Una vez que se tiene la aplicación móvil y que se logró enviar las imágenes a Firebase, se realizan pruebas para obtener los pesos de las imágenes de la base de datos con las mismas imágenes, pero enviadas desde la aplicación móvil, con el fin de comparar los pesos obtenidos y ver que no hay diferencia entre estos.

8. Procesar las imágenes adquiridas con la aplicación para poder aplicar técnicas de mejora de imágenes, en este caso se mostrarán los resultados aplicando el histograma de ecualización y filtro de contraste.
9. Una vez que se hagan las suficientes pruebas y se obtengan los resultados, se procederá a comparar los resultados usando el modelo con las imágenes y después con las imágenes una vez aplicado los filtros para observar la diferencia entre estos, para finalizar se realizará un clasificador en base a los pesos obtenidos de las imágenes adquiridas y procesadas.
10. Mostrar la clasificación al usuario de la posible presencia de melanoma e invita al usuario a visitar un especialista.

En cuanto a la línea terminal de Mecatrónica, sin duda fue de gran ayuda ya que desde un principio el proyecto tuvo que ver con la línea terminal, ya que se tocan temas relacionados a programación y procesamiento de imágenes, los cuales fueron conocimientos que se vieron en la línea terminal, además de que siempre se dejó en claro la parte de ser autónomos, lo cual considero fundamental para el desarrollo como estudiante, ya que muchas veces se tuvo que aprender por nuestra cuenta. Asimismo, la universidad cuenta con equipo para seguir trabajando en el área de visión, lo cual fue un gran incentivo para desarrollar el trabajo de tesis enfocado en imágenes, ya que se presta la oportunidad de poder continuar el trabajo, ya sea escalando o reconfigurando, además de que se cuenta con el posgrado en mecatrónica, la cual es una gran oportunidad para seguir desarrollando habilidades en esta área.



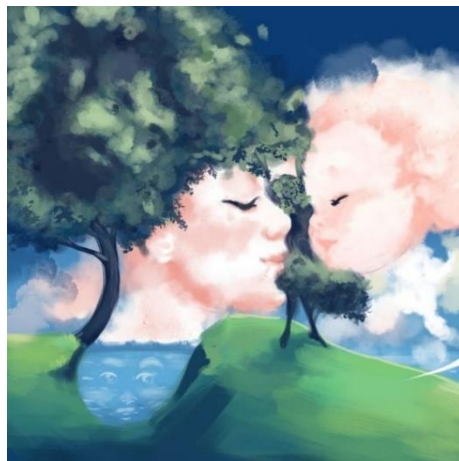
Figura 1.1: Diagrama propuesto para el desarrollo del proyecto (autoría propia).

## 2. REVISIÓN DE LITERATURA

En el presente capítulo se van a analizar los fundamentos teóricos que se necesitan para poder respaldar el presente proyecto de tesis, por lo tanto, esta sección es bastante importante para comprender la metodología. Dentro de los conceptos más importantes que se analizan se encuentran definiciones de melanoma, imagen, redes neuronales, de la plataforma donde se desarrollará la aplicación, además de los métodos de procesamiento de imágenes implementados. También se analizan las normas encaminadas al diagnóstico dentro del área de la salud.

### 2.1 Imagen

Una imagen es la representación visual que manifiesta la apariencia visual de algún objeto real o imaginario. Las imágenes son utilizadas para comunicar o expresar información, hoy en día se cuentan con distintos medios para tener imágenes, entre las cuales está las cámaras que permiten capturar una imagen para imprimirla o tenerla de manera digital, también pueden ser creadas por medio de la pintura, entre otros. Como se puede apreciar en la Figura 2.1, la imagen puede expresarnos distintas ideas, en donde recae en cada uno el interpretar o imaginar algo particular por medio de la imagen.



*Figura 2.1: Representación visual de una pintura. (Milenio Digital, 2021)*

### 2.1.1 Imagen digital

Una imagen se puede definir como una función bidimensional, donde tenemos la función  $f(x,y)$ , donde  $x$  e  $y$  son coordenadas espaciales (planas), y la amplitud de  $f$  en cualquier par de coordenadas  $(x,y)$  se denomina intensidad o nivel de gris de la imagen en ese punto. Cuando  $x,y$  y los valores de intensidad de  $f$  son todas cantidades finitas y discretas, llamamos a la imagen una imagen digital. Cuando hablamos del campo del procesamiento de imágenes digitales, se hace referencia al procesamiento de imágenes digitales por medio de una computadora, hay que tener en cuenta que una imagen digital se compone de un número finito de elementos, cada uno de los cuales tiene una ubicación y un valor en particular. Estos elementos se denominan “elementos de imagen”, “pels” y “píxeles”. El píxel es el término más utilizado para designar los elementos de una imagen digital, en donde no es más que la parte más pequeña, en donde el conjunto de píxeles da como producto una imagen digital.

Una parte fundamental a dejar en claro es que la vista es el más avanzado de nuestros sentidos, por lo que no nos llega a sorprender que las imágenes desempeñen un papel fundamental en la percepción humana, pero, desgraciadamente para los humanos, la banda visual del espectro electromagnético (EM) está limitado, mientras que las máquinas pueden llegar a cubrir casi todo el espectro EM, ya sea desde las ondas gamma hasta las ondas de radio, es por ello que pueden operar imágenes generadas por fuentes que los humanos no están acostumbrados a asociar con imágenes, entre las cuales podemos tener ultrasonido, microscopía electrónica e imágenes generadas por computadoras. Es así que el procesamiento digital de imágenes abarca un amplio y variado campo de aplicaciones. (Gonzalez y Woods, 2018).



*Figura 2.2: Imagen tomada por el Dr. David R. Pickens usando herramientas tecnológicas. (Gonzalez y Woods, 2018).*

### 2.1.2 Espectro electromagnético

El espectro electromagnético fue un fenómeno descubierto por Isaac Newton en 1666 al experimentar con un rayo de luz a través de un prisma de vidrio, en donde se dio cuenta que la luz emergente no fue blanca, sino que consiste en una serie de colores desde el violeta en un extremo hasta el rojo en el otro. Es aquí de donde nace el espectro electromagnético y con el paso del tiempo se continuó estudiando y se encontró que el ser humano no es capaz de visualizar toda la gama de colores, de hecho, es solo una pequeña porción. En la figura 2.3 se puede apreciar el espectro electromagnético y la zona visible es mostrada en un zum para facilitar mucho mejor la gama de colores. (Gonzalez y Woods, 2018).

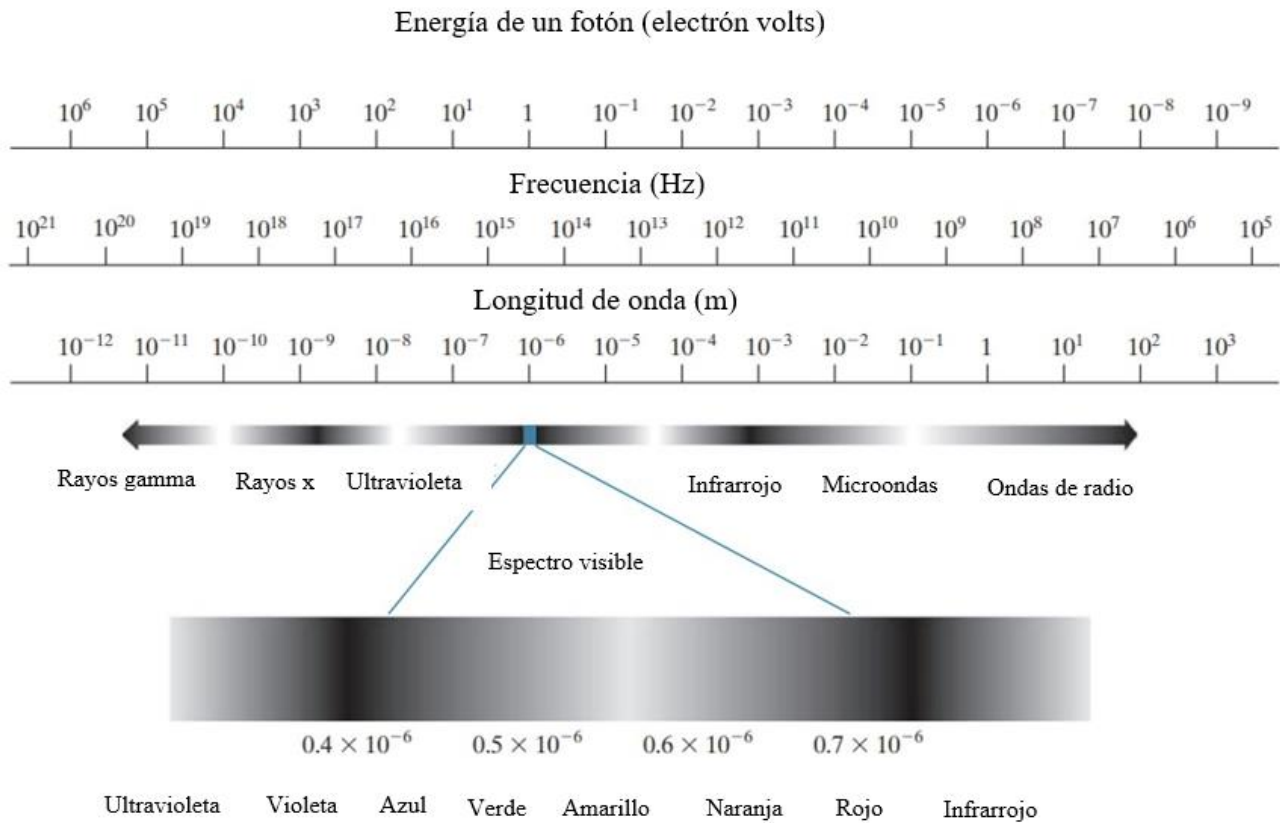


Figura 2.3: Espectro electromagnético. (Gonzalez y Woods, 2018).

En un extremo del espectro se encuentran las ondas de radio con longitudes de onda miles de millones de veces más largas que las de la luz visible. En el otro extremo del espectro están los rayos gamma con longitudes de onda millones de veces más pequeñas que las de la luz visible.

La energía es proporcional a la frecuencia, por lo que fenómenos electromagnéticos de mayor frecuencia (longitud de onda más corta) transportan más energía por fotón. De modo que podemos visualizar en la Figura 2.3, las microondas tienen más energía que las ondas de radio, los infrarrojos aún más, luego los visibles, los ultravioletas, los rayos X y finalmente los rayos gamma, los más energéticos de todos. La parte de radiación electromagnética de alta energía, especialmente rayos gamma y rayos X son dañinos para los organismos vivos.

El espectro de colores se divide en seis amplias regiones: violeta, azul, verde, amarillo, naranja y rojo. Ningún color (u otro componente del espectro electromagnético) termina abruptamente; más bien, cada rango se mezcla suavemente con el siguiente.

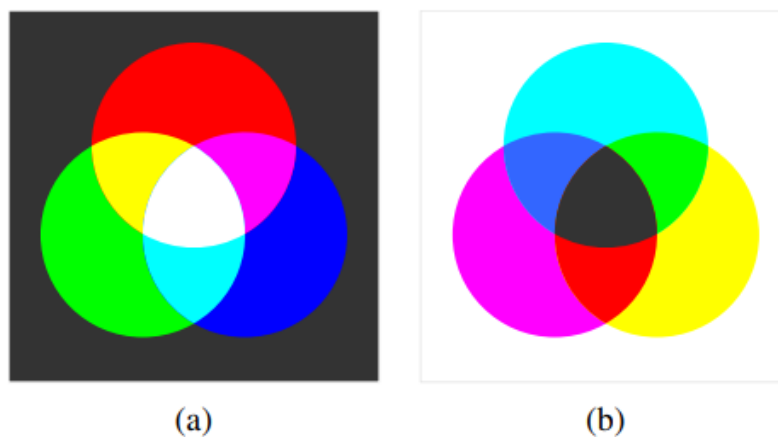
Los colores que pueden ser percibidos en un objeto están determinados por la naturaleza de la luz reflejada por el objeto. Un cuerpo que refleja luz que se considera relativamente equilibrada en todas las longitudes de onda visibles se aparece de color blanco para el observador. Por otro lado, un cuerpo que favorece la reflectancia en un rango limitado del espectro visible exhibe algunos matices de colores. De hecho, un ejemplo muy común a la hora de trabajar con imágenes es el color verde, ya que este refleja la luz con longitudes de onda principalmente en el rango de 500 a 570 nm, mientras que absorben la mayor parte de la energía en otras longitudes de onda.

La luz que no tiene color es llamada monocromática (también conocida como acromática), el atributo con el que cuenta es su intensidad, debido a que se percibe la intensidad de la luz monocromática varía de negro a gris y finalmente al blanco, el término nivel de gris se usa para denotar la intensidad monocromática. El rango de valores de la luz monocromática de negro a blanco se suele denominar escala de grises. La luz cromática (color) abarca el espectro de energía electromagnética desde aproximadamente 0,43 a 0,79  $\mu\text{m}$ , además de la frecuencia, se utilizan otras tres cantidades para describir una fuente de luz cromática: resplandor, luminancia y brillo. (Gonzalez y Woods, 2018).

Es aquí en donde el procesamiento de imágenes, el área de la visión asistido por computadora entra de lleno para poder estudiar imágenes y poder apreciar cosas que no son visibles al ojo humano, además de contar con los recursos para poder entrenar modelos y crear algoritmos que ayuden a detectar áreas de interés.

### 2.1.3 Color

Hablar de color en imágenes es tocar el tema del modelo de color RGB (Red, Green, Blue) el cual es un sistema de representación de colores que se utiliza en imágenes digitales y en lo que son los dispositivos electrónicos. Para ello, hay que tener en claro que los píxeles que vemos en una imagen digital no son más que un pequeño punto los cuales se representan mediante una combinación de valores de componentes rojos, verdes y azules. Por lo tanto, una imagen digital se representa con una matriz de píxeles, en donde los monitores y pantallas están compuestos de componentes electrónicos llamados píxeles y que son capaces de cambiar de color para poder representar un píxel de un color rojo, verde o azul, cambiando la intensidad de estos, dando como resultado los demás colores que conocemos y sobre todo que los humanos podemos distinguir. Los colores aditivos rojo, verde y azul se pueden mezclar para producir los colores cian, magenta y amarillo.



*Figura 2.4: Colores primarios y secundarios. a) los colores aditivos. b) Los colores sustractivos. (Szeliski, 2010).*

Los colores sustractivos se llaman así porque los pigmentos en la pintura absorben ciertas longitudes de onda en el espectro de color, los que corresponden a colores sustractivos son el cian, magenta y amarillo. Ahora bien, la manera en que los colores primarios interactúan para poder producir un nuevo color es debido al resultado de la naturaleza triestímulo (también llamado tricromática) del sistema visual humano, ya que tenemos tres tipos diferentes de cono, cada uno de los cuales responde selectivamente a una porción diferente del espectro de color. (Glassner et al, 1995). Es importante mencionar que para aplicaciones más complejas como puede ser la visión artificial es recomendable utilizar más longitudes de onda. (Szeliski, 2010).



Según el libro “Digital Image Processing” de Rafael C, el modelo RGB puede tomar valores de 0 hasta 255, donde 0 representa la ausencia de color y 255 la máxima intensidad de color, al combinar los modelos podemos obtener una gran variedad de colores. (Gonzalez y Woods, 2018).

Ese sistema permite representar más de 16 millones de colores diferentes ( $2^{32}$ ) a través de la combinación de los 256 niveles de rojo, con 256 de verde y 256 niveles de azul. Para poder tener valores entre 0-255 necesitamos usar 8 bits (corresponde a 1 byte), por lo que el sistema RGB requiere 3 bytes por cada píxel de una imagen.

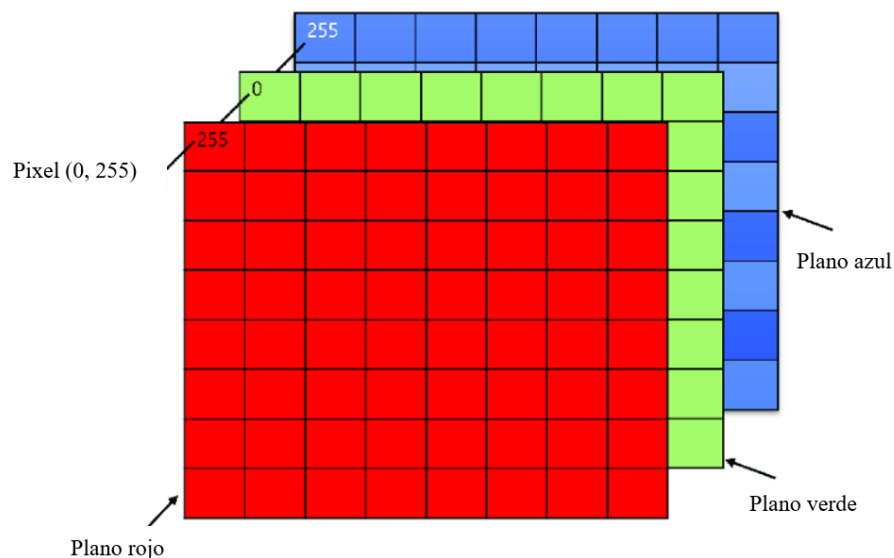
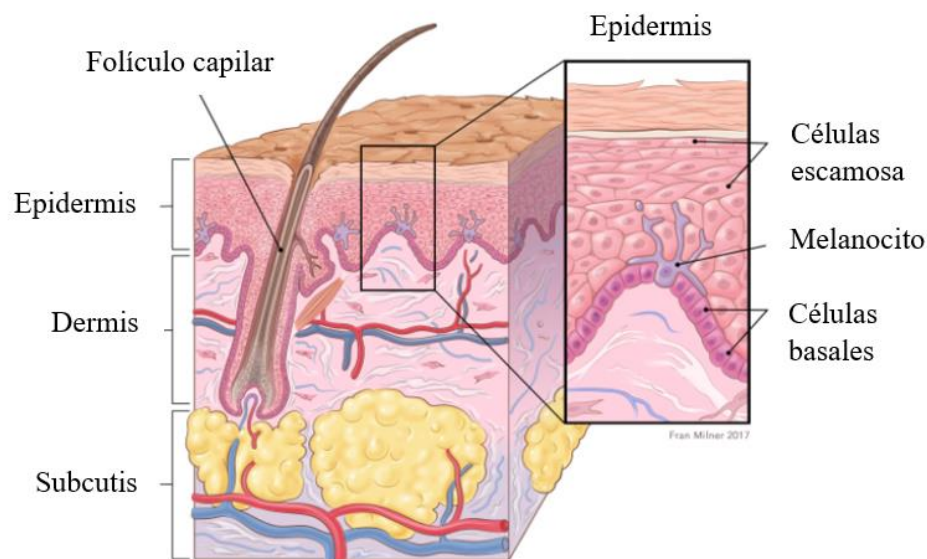


Figura 2.5: Representación de los componentes RGB. (Cheonshik et al. 2021).

## 2.2 Melanoma

El melanoma es uno de los tipos de cáncer de piel que existen, para ello es necesario mencionar que tenemos distintos tipos, además de que cada uno de ellos tiene diferente nivel de peligrosidad. Empecemos definiendo lo que es cáncer, es pocas palabras, el cáncer es una enfermedad el que algunas de las células del cuerpo crecen sin control y se diseminan a otras partes del cuerpo. (Mitchell et al., 2020). La gran mayoría de cánceres de piel comienzas en la capa superior de la piel llamada epidermis, hay 3 tipos principales de células en esta capa, las cuales son las siguientes:

- **Células escamosas:** “Son las células planas en la parte superior (externa) de la epidermis, que se desprende constantemente a medida que se forman otras nuevas.” (Mitchell et al., 2020).
- **Células basales:** “Estas células se encuentran en la parte inferior de la epidermis, estas se dividen constantemente para formar nuevas células que remplazan a las células escamosas que se desgastan en la superficie de la piel. A medida que estas células ascienden en la epidermis, se vuelven más planas y eventualmente se convierten en células escamosas.” (Mitchell et al., 2020).
- **Melanocitos:** “Estas son las células que pueden convertirse en melanoma, por lo general producen una pigmentación de color marrón en la piel llamada Melanina, esta lo que hace es proteger las capas más profundas de la piel de algunos de los efectos nocivos del sol.” (Mitchell et al., 2020).



*Figura 2.6: Capas de la piel. (Mitchell et al., 2020).*

Es importante mencionar que los melanomas aparecen en la capa epidermis, sin embargo, hay registros que nos indican que también hay posibilidad de que aparezcan en los ojos, la boca, los genitales y el área anal, pero son mucho menos comunes. También se tiene que decir que hay ocasiones en los que los melanomas no producen melanina, lo que quiere decir que no aparece una mancha marrón en la piel, sino que algunas veces puede aparecer una mancha rosada, tostada o incluso blanca. (Ribas et al., 2019).

Dentro de los distintos tipos de cáncer de piel, el melanoma es el menos común, tanto es así que en las estadísticas cuando se habla de cáncer de piel, se hace una separación, se pueden encontrar el cáncer melanoma y el cáncer no melanoma, es por ello que en muchos rankings se pueden visualizar estos dos. Dicho lo anterior, el melanoma es el cáncer de piel menos frecuente, pero es el más peligroso puesto que es el que tiene mayor probabilidad de propagarse a otras partes del cuerpo si no se detecta y trata a tiempo. (Mitchell et al., 2020).

### **2.2.1 Melanoma a nivel mundial**

Retomemos que hay dos grandes clasificaciones de cáncer de piel, tenemos el melanoma y el no melanoma, este último reúne los tumores no melanoma más comunes como son el carcinoma de células basales y el carcinoma de células escamosas previamente explicados en la parte de melanoma. Uno de los grandes retos que se tiene a la hora de poder estimar la incidencia del cáncer de piel es debido a que existen múltiples subtipos, por ejemplo, el cáncer de piel no melanoma a menudo no se rastrea en los registros de cáncer ya que en la mayoría de los casos se tratan con éxito mediante una cirugía o ablación. Esto se menciona ya que, debido a estos factores, es probable que la incidencia mundial notificada sea una subestimación. (World Health Organization, 2022).

El melanoma de piel ocupa el puesto 17° de cánceres a nivel mundial, es el decimotercer cáncer más común en los hombres y el decimoquinto en las mujeres. Se tiene registros de que en el año 2020 hubo más de 150.000 nuevos casos de melanoma, a continuación, en la Tabla 1 se puede observar los diez países con las tasas más altas de cáncer de piel registrado en 2020, en la columna de ASR, se refiere a la tasa estandarizada de edad, estas son una medida resumida de la tasa de enfermedad que tendría una población si tuviera una estructura de edad estándar. La estandarización es necesaria cuando se comparan poblaciones que difieren con respecto a la edad porque la edad tiene una poderosa influencia en el riesgo de morir por cáncer.

*Tabla 1: Tasa de incidencia a nivel global de melanoma. (World Health Organization, 2022).*

Rango	País	Número	ASR/100,000
	Mundo	324,635	3.4
1	Australia	16,171	36.6
2	Nueva Zelanda	2,801	31.6
3	Dinamarca	2,886	29.7
4	Los países bajos	8,310	27.0
5	Noruega	2,567	26.4
6	Suecia	4,266	23.3
7	Suiza	3,357	21.6
8	Alemania	31,468	20.5
9	Eslovenia	735	19.7
10	Finlandia	2,090	19.5

En la Tabla 2 se muestran los diez países con la mayor mortalidad global total registrado en 2020, cabe decir que el lugar que ocupan está basado en el ASR (tasa estandarizada de edad).

*Tabla 2: Mortalidad global de melanoma. (World Health Organization, 2022).*

Rango	País	Número	ASR/100,000
	Mundo	57,043	0.6
1	Nueva Zelanda	472	4.7
2	Noruega	375	3.2
3	Montenegro	32	3.0
4	Eslovaquia	317	2.8
5	Eslovenia	127	2.6
6	Australia	1,408	2.4
7	Dinamarca	341	2.4
8	Croacia	236	2.4
9	Los países bajos	906	2.3
10	Serbia	393	2.3

### **2.2.2 Melanoma a nivel nacional (México)**

Es una realidad que los casos de melanoma en México están en constante aumento, pues de entrada se tienen bastantes factores que no ayudan, entre los cuales se conocen la localización, hablamos del hecho de que en la mayoría del país suele estar muy fuerte los rayos del sol, también está el factor de que la gran mayoría de los mexicanos no le dan la importancia que merece a la salud, pues muchas veces se ignoran síntomas y se espera que se cure con el paso del tiempo, lo cual es un fatal error, ahora si hablamos de cosas de la piel es peor ya que muchas veces no somos conscientes de los cambios en nuestro cuerpo e incluso por cuestiones de pena se suele callar y lamentablemente cuando ya es muy tarde es que se decide ir con un especialista.

“Se consultó la base de datos interactiva de mortalidad por cáncer según la Clasificación Internacional de Enfermedades (CIE) disponibles en la página electrónica del Instituto Nacional de estadística y Geografía (INEGI) entre los años 1998-2016. Para tener un mejor panorama se llevó a cabo un estudio por parte del Consejo Nacional de Población (CONAPO), donde se calculó la tasa de mortalidad bruta por cada 100,000 habitantes de acuerdo a las estimaciones y proyecciones poblacionales del CONAPO entre 1998-2016 y no se utilizaron censos poblacionales del INEGI para no combinar metodologías distintas. Para calcular la mortalidad en el Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE), el Instituto Mexicano del Seguro Social (IMSS), el Seguro Popular, Petroleros Mexicanos (PEMEX), la Secretaría de la Defensa Nacional (SEDENA) y la Secretaría de la Marina (SEMAR), se revisó la base de datos de la población derechohabiente disponible en la página electrónica de cada institución y en el INEG.” (Hernández et al., 2018).

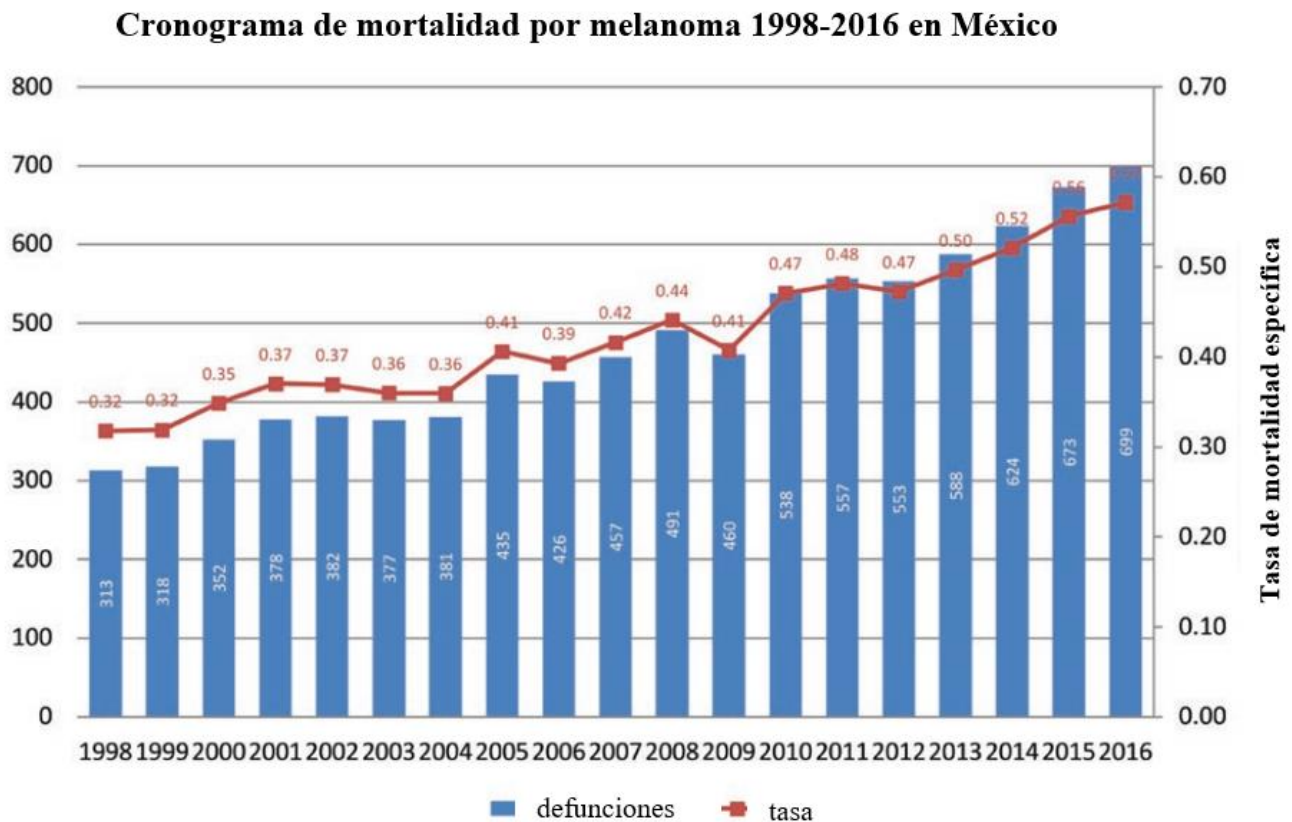


Figura 2.7: Cronograma de mortalidad por melanoma en México. (Hernández et al., 2018).

Una comparativa y dato muy particular, es que en México se tienen registrados cerca de 700 defunciones anuales, lo cual hasta cierto punto no es mucho, pero si lo comparamos con Estados Unidos, este tiene cerca de 7000 defunciones anuales, además podemos visualizar que con el paso de los años ha aumentado el número de defunciones en México.

### **2.2.3 Métodos de diagnóstico**

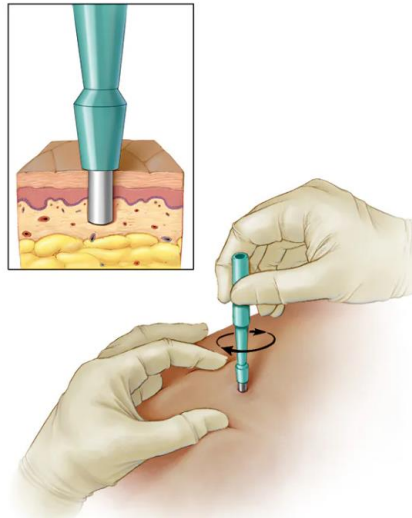
Hoy en día se cuentan con bastantes métodos para poder diagnosticar un melanoma, dentro del proceso se encuentra que el especialista revise la mancha en la piel y pregunte si se ha tenido un crecimiento, además de preguntar si algún familiar ha padecido algún tipo de cáncer de piel entre otras cosas, pero no cabe duda de que a la hora de hacer una mayor revisión usando un método de diagnóstico, estos presentan lados negativos para los seres humanos, pues estos métodos pueden llegar a ser nocivos para la salud.

Empecemos con el método por excelencia, hablamos de la conocida Biopsia, la cual consiste en extraer una parte de la piel para posteriormente someterlo a una revisión detallada de dicho tejido, es importante mencionar que cualquier biopsia es propensa a dejar por lo menos una pequeña cicatriz. Los diferentes métodos pueden dejar diferentes tipos de cicatrices, por otro lado, una biopsia puede hacer un diagnóstico definitivo.

Los estudios por imágenes usan rayos X, campos magnéticos y sustancias radiactivas para crear imágenes del interior del cuerpo. Principalmente se usan para localizar la posible propagación de melanoma en los ganglios linfáticos o en otros órganos. Los estudios por imágenes no son necesarios para la mayoría de las personas con melanomas en etapas muy tempranas que presentan poca probabilidad de propagación. (Ribas et al., 2019).

La tomografía computarizada usa rayos X para producir imágenes transversales detalladas de su cuerpo. A diferencia de una radiografía regular, la tomografía puede mostrar el detalle en tejidos blandos (tal como órganos internos). Este estudio puede indicar si cualquier ganglio linfático está agrandado o si órganos, como los pulmones o el hígado tienen áreas sospechosas que podrían deberse a la propagación del melanoma. La parte negativa de este método es que al usar rayos X se termina sometiendo a la persona a radiación que puede llegar a ser nocivo si es que se somete a muchos estudios bajo rayos X. (Mitchell et al., 2020).

Sin duda el acudir con un especialista es la mejor recomendación, pero aún y cuando es la mejor decisión que uno puede tomar, no deja de existir un posible riesgo para la salud al someterse a un método de diagnóstico, es aquí donde el uso y avance de la tecnología puede entrar para ayudar y encaminar a ver a un especialista si es que el panorama no se ve bien.



*Figura 2.8: Biopsia en sacabocados. (Mayo Clinic Family Health Book)*

#### **2.2.4 Herramientas de detección alternas**

Sin duda el avance de la tecnología está en constante evolución y cada día se aprovechan distintas tecnologías para poder aportar dentro del sector de la salud, uno de los mejores ejemplos corresponde a la parte de procesamiento de imágenes, en donde usar este tipo de tecnologías puede ayudar a diagnosticar, comprender y tratar cánceres de piel. Las tecnologías están ayudando a los dermatólogos a tomar decisiones más informadas para mejorar los resultados de salud y la experiencia del paciente. (Kopec, 2022).

La detección del cáncer de piel es un proceso visual. Y aunque los dermatólogos son expertos en identificar cánceres de piel, los dispositivos de imágenes de alta tecnología como escáneres de cuerpo total puede proporcionar datos de precisión adicionales para ayudar a los médicos a monitorear y manejar a ciertos pacientes. (Kopec, 2022). Aún y con lo anterior mencionado, es cierto que sigue siendo un tema de gran discusión ya que se está poniendo en riesgo la salud de una persona, pero eso no quita que el hecho de poder usar herramientas tecnológicas sea un gran avance y de gran utilidad en el sector de medicina. Hablando de herramientas de detección para melanomas, tenemos hoy en días el uso de cámaras termográficas, cámaras de smartphones e incluso rayos ultravioletas.





*Figura 2.9: Uso de herramientas tecnológicas en el sector de la salud. (Scudellari, 2020).*

### **2.2.5 Archivos SIIM-ISIC**

En cuanto a la base de datos utilizada a lo largo del proyecto, tanto para la parte de entrenamiento como para las pruebas, se hizo uso del conjunto de datos oficial del desafío de clasificación de melanoma SIIM-ISIC. Empecemos especificando la cantidad de imágenes con las que se cuentan y la cantidad de imágenes que representan lesiones benignas y malignas. El conjunto de datos tiene 33,126 imágenes de entrenamiento dermatoscópico de lesiones cutáneas benignas y malignas de más de 2000 pacientes. Cada una de las imágenes están asociadas con una de las personas mediante un identificador de paciente único. Cabe decir que todos los diagnósticos que se mostraron como malignos, fueron confirmados mediante histopatología, lo cual es el estudio de células y del tejido bajo un microscopio, además de que los diagnósticos benignos se confirmaron mediante acuerdos de expertos, seguimiento longitudinal o histopatología. Es importante decir que, a la hora de diagnosticar casos de melanoma, es importante tener una publicación completa que describe las características de ese conjunto de datos, a lo cual se cuenta con un conjunto de datos preimpresos que aún no ha sido revisada por pares. (Rotemberg et al., 2021).

Ahora bien, la base de datos se reunió de distintas partes del mundo, esto es fundamental ya que al ser una base de datos que se va a utilizar con tecnologías que involucren procesamiento de imágenes, un factor importante es que se cuenta con una variación de color de piel, ya que estudiar piel caucásica puede mostrarnos otro tipo de resultados y en caso de procesar una imagen con un tono de piel más oscuro puede que no función correctamente. El

conjunto de datos fue generado por la colaboración internacional de imágenes de la piel (ISIC) y las imágenes provienen de los siguientes lugares: Hospital Clínic de Barcelona, Medical University of Vienna, Memorial Sloan Kettering Cancer Center, melanoma Institute Australia, University of Queensland y University of Athens Escuela de Medicina. (Rotemberg et al., 2021).

## 2.3 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial que se usa para el desarrollo de aplicaciones móviles, este, está basado en el potente editor de código y la herramienta para desarrolladores de IntelliJ IDEA. (Developers, 2023). Android Studio ofrece muchas funciones que mejoran la productividad cuando se compilan apps para Android, como son las siguientes:

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros
- Compatibilidad con C++ y NDK.

Los lenguajes de programación en los que se pueden desarrollar proyectos son principalmente Java y Kotlin, desde el inicio el lenguaje utilizado fue Java, ahora bien, no hay que confundir Android Studio con un lenguaje, ya que como se mencionó anteriormente, este solo es el entorno para desarrollar, usando los lenguajes de programación Kotlin (uno de los principales, Java, NDK y C++). En cuestión de compilación de código se utiliza Gradle, que está especializado para funcionalidades Android.

## 2.4 Aprendizaje supervisado y no supervisado

En el campo del aprendizaje automático (Machine Learning), existen dos enfoques fundamentales para extraer conocimiento y realizar predicciones a partir de los datos: el aprendizaje supervisado y el aprendizaje no supervisado. Estos enfoques desempeñan un papel crucial en el desarrollo de modelos de inteligencia artificial y tienen aplicaciones en una amplia gama de disciplinas, desde el reconocimiento de imágenes hasta el procesamiento del lenguaje natural.

El aprendizaje supervisado es un enfoque en el que se proporciona al modelo un conjunto de ejemplos de entrada junto con sus correspondientes salidas deseadas o etiquetas. Durante el proceso de entrenamiento, el modelo aprende a mapear las entradas a las salidas correctas basándose en estos ejemplos supervisados. En esencia, el modelo es "supervisado" por las respuestas deseadas mientras ajusta sus parámetros internos.

Un ejemplo común de aprendizaje supervisado es la clasificación de imágenes. Al presentar al modelo una gran cantidad de imágenes previamente etiquetadas con categorías como "perro" o "gato", el modelo aprenderá a reconocer patrones visuales que distingan entre las diferentes clases. Luego, cuando se le presenta una imagen nueva sin etiquetar, el modelo puede predecir la clase a la que pertenece.

Por otro lado, el aprendizaje no supervisado es un enfoque en el que no se proporcionan etiquetas o salidas deseadas al modelo durante el entrenamiento. En lugar de eso, el modelo busca descubrir patrones y estructuras inherentes en los datos sin la guía explícita de las respuestas correctas. Es decir, el modelo explora los datos de manera no dirigida.

Un ejemplo común de aprendizaje no supervisado es el agrupamiento. Mediante técnicas de agrupamiento, el modelo puede identificar grupos de datos similares basándose únicamente en las características y similitudes entre ellos. Esto permite explorar y comprender la estructura subyacente de los datos sin tener conocimiento previo de las categorías específicas o etiquetas asociadas a los datos.

Tanto el aprendizaje supervisado como el aprendizaje no supervisado son enfoques esenciales en el campo del aprendizaje automático. Mientras que el aprendizaje supervisado se basa en ejemplos etiquetados para entrenar un modelo capaz de realizar predicciones precisas, el

aprendizaje no supervisado se enfoca en descubrir patrones y estructuras ocultas en los datos.

La elección entre aprendizaje supervisado y no supervisado depende de los datos disponibles y los objetivos del problema en cuestión. Ambos enfoques ofrecen una amplia gama de técnicas y algoritmos que permiten a los investigadores y profesionales del aprendizaje automático abordar una variedad de desafíos y extraer información valiosa de los datos. Al comprender las diferencias y aplicaciones de estos enfoques, se puede aprovechar su potencial para mejorar la toma de decisiones y desarrollar soluciones inteligentes en diversos campos y sectores.

Podemos apreciar en la figura 2.10 un claro ejemplo entre aprendizaje supervisado y no supervisado. Se tiene una base de datos que en este caso son cuadrados y triángulos de color rojo y azul de diferentes tamaños, en donde vemos que en el aprendizaje supervisado se tiene en la entrada la base de datos y etiquetas, las cuales corresponden a especificaciones dadas desde el inicio, en este caso se enseña a que pueda clasificar en base al color y figura, mientras que en el aprendizaje no supervisado solamente se tiene la base de datos a la entrada y solito se entrena la red neuronal para que aprenda a realizar una clasificación, en este caso vemos que termino clasificando en base a la figura. Lo más interesante de todo esto es que uno esperaría que se usará aprendizaje supervisado para trabajos computacionales más complicados, pero es todo lo contrario, ya que muchas veces no se conoce la forma de llegar a un resultado y es aquí donde la red neuronal hace todo el trabajo para encontrar pesos ideales.

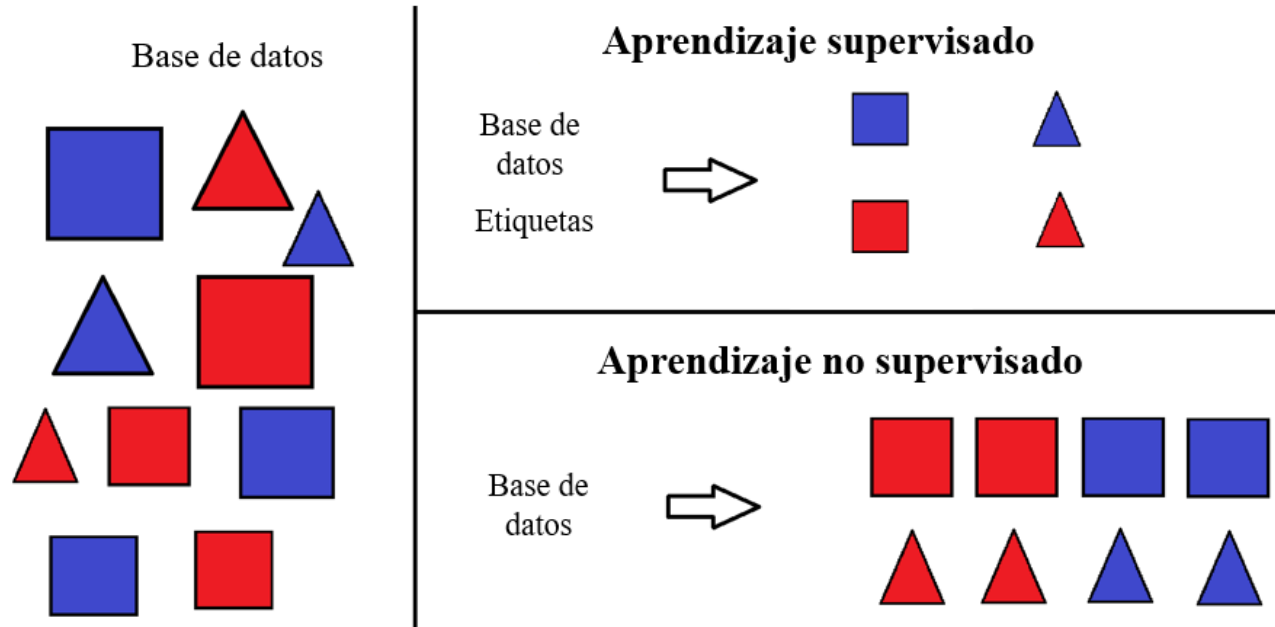


Figura 2.10: Aprendizaje supervisado y no supervisado (autoría propia).

## 2.5 Red neuronal convolucional

Un elemento fundamental empleado en el área de la inteligencia artificial que resulta de mucha ayuda para la parte de clasificaciones y demás, está constituido por las redes neuronales convolucionales, las cuales hoy en día suponen un aspecto a considerar cuando se llevan a cabo trabajos relacionados en la inteligencia artificial. Sin duda alguna en la actualidad la inteligencia artificial está en un completo auge, ya que se ha convertido en una necesidad el poder contar con este tipo de tecnologías para resolver problemas en distintas áreas. Los avances en visión por computadora con aprendizaje profundo se han construido y perfeccionado con el tiempo buscando emular la visión humana, es decir, se quiere conseguir que el campo permita que las máquinas vean el mundo tal y como lo hace el ser humano, además de que sean capaces de percibir de manera similar y usen el conocimiento para una multitud de tareas como reconocimiento por medio de imágenes, videos, análisis, clasificación de imágenes, recreación de medios entre otras.

“Una red neuronal convolucional (ConvNet/CNN) es un algoritmo de aprendizaje profundo que lo que hace es tomar una imagen de entrada, para posteriormente asignar importancia (pesos y sesgos aprendibles) a varios aspectos/objetos en la imagen y ser capaz de diferenciar uno de otro. Una característica muy interesante, es que el preprocesamiento requerido

en ConvNet es mucho menor en comparación con otros algoritmos de clasificación, mientras que en los métodos primitivos los filtros están diseñados a mano, con suficiente capacitación, ConvNet tiene la capacidad de aprender esto filtro/características.” (Saha, 2018).

“Pensemos que tenemos una imagen en 8k (7680 x 4320), no hay duda de que visto desde el punto computacional, es bastante intensivo, pues la dimensión es muy grande, es aquí donde entra el papel de ConvNet, ya que este reduce las imágenes para que sea una forma más fácil de procesar, claro que sin perder características que son críticas para obtener buenas predicciones. Esto es importante a la hora de diseñar una arquitectura que no solo sea buena para aprender funciones, sino que también es escalable a conjuntos de dato masivos.” (Saha, 2018).

### 2.5.1 Pesos

Dentro del contexto de redes neuronales convolucionales, existe lo que se conoce como peso, no es más que un componente fundamental que ayuda a la red a aprender y realizar predicciones precisas. El peso es prácticamente un número que representa la importancia de una conexión sináptica entre do neuronas, prácticamente estos pesos lo que hacen es multiplicarse por la señal de entrada y se suman para calcular la salida de una neurona, a medida que una red neuronal aprende, ajusta los pesos para minimizar el error y mejorar la precisión de las predicciones. (Chollet, 2018). Una cosa importante a mencionar es que el peso es ajustable durante el proceso de entrenamiento de una CNN y se utiliza para adaptar la red o los datos de entrada, permitiendo así realizar clasificaciones o predicciones más precisas.

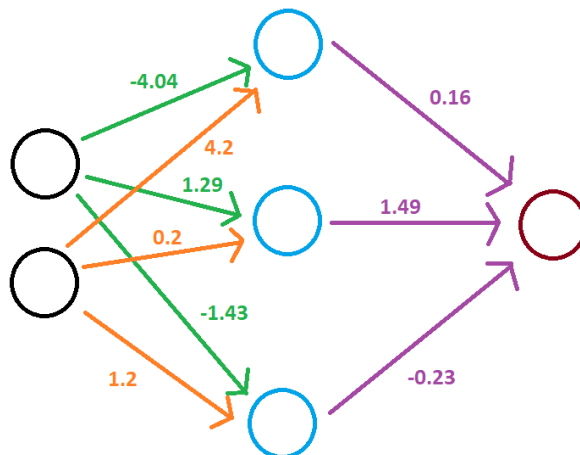


Figura 2.11: Distribución de pesos en una red neuronal. (autoría propia).

### 2.5.2 Sesgos

El termino de sesgo hace referencia a un componente fundamental de las redes neuronales convolucionales, al igual que los pesos. El sesgo es un parámetro que se agregan a la salida de la neurona, en lugar de tener que pasar por el origen, la función de activación puede tener un valor no nulo cuando todas las entradas son cero. Los sesgos desplazan la función de activación y le permiten ajustarse mejor a los datos. (Goodfellow et al., 2015). Al igual que los pesos, esta es ajustable durante el proceso de entrenamiento de la CNN con tal de predecir con una mayor precisión.

El objetivo es siempre construir modelo que puedan predecir datos cercanos a los valores verdaderos, esto no es fácil ya que hay algoritmos muy rígidos para aprender señales complejas del conjunto de datos. Para entrar mejor en contexto pongamos un ejemplo, imaginemos que queremos ajustar una regresión lineal a un conjunto de datos que tiene un patrón no lineal, al final esa regresión no podrá modelar la curva en esos datos, a eso se le conoce como ajuste insuficiente. En general los algoritmos paramétricos como la regresión lineal tienen un alto sesgo que hace que sea rápido de aprender y fácil de entender, pero generalmente son menos flexibles, a su vez tiene un menor rendimiento productivo en problemas complejos. El objetivo de cualquier algoritmo supervisado de Machine Learning es lograr un sesgo bajo y una baja varianza, a su vez, el algoritmo debe lograr un buen rendimiento de predicción, esto se aprecia en la Figura 2.12. (Gonzalez, 2018).

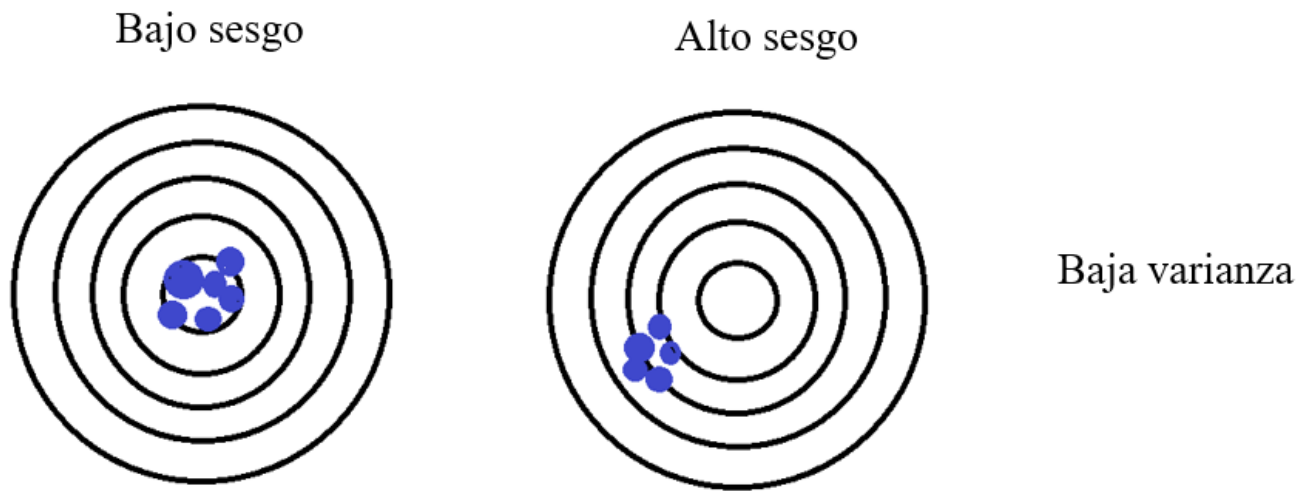


Figura 2.12: Diferencias de tener un sesgo alto y bajo con una baja varianza. (autoría propia).

## 2.6 Procesamiento de imágenes

El procesamiento de imágenes es una disciplina del procesamiento de señales y la visión por computadora que se enfoca en la adquisición, análisis, manipulación y comprensión de imágenes digitales. Consiste en utilizar algoritmos y técnicas computacionales para realizar operaciones en imágenes con el objetivo de mejorar su calidad, extraer información relevante o realizar tareas específicas, como reconocimiento de objetos, detección de rostros, segmentación de imágenes, entre otras.

Primero, los motores de inducción son conectados a uno o más sensores, dependiendo la variable que se requiera medir, por ejemplo, corriente, flujo, vibraciones, etc. Los sistemas de monitoreo y diagnóstico se basan en tratamientos digitales de las señales y con ello, infieren la condición en la que se encuentra operando el motor de inducción, por ello se muestra que a los sistemas de monitoreo y diagnóstico entran las variables que adquieren los sensores y con ayuda de un algoritmo basado en tecnología de detección de fallos pueden presentar un diagnóstico del motor (si se encuentra en su estado de referencia, es decir, en un estado sano o si presenta alguna anomalía contra su estado de referencia, lo cual se traduce como un fallo y que puede ser clasificado en distintos niveles). El impacto del procesamiento de imágenes ha sido significativo en diversos campos y sectores. Aquí hay algunas áreas en las que ha tenido un impacto destacado:

**Medicina:** El procesamiento de imágenes ha revolucionado el campo médico al permitir el diagnóstico más preciso y la visualización de estructuras anatómicas. Ha facilitado la detección temprana de enfermedades, el seguimiento del progreso del tratamiento y la planificación de intervenciones quirúrgicas.

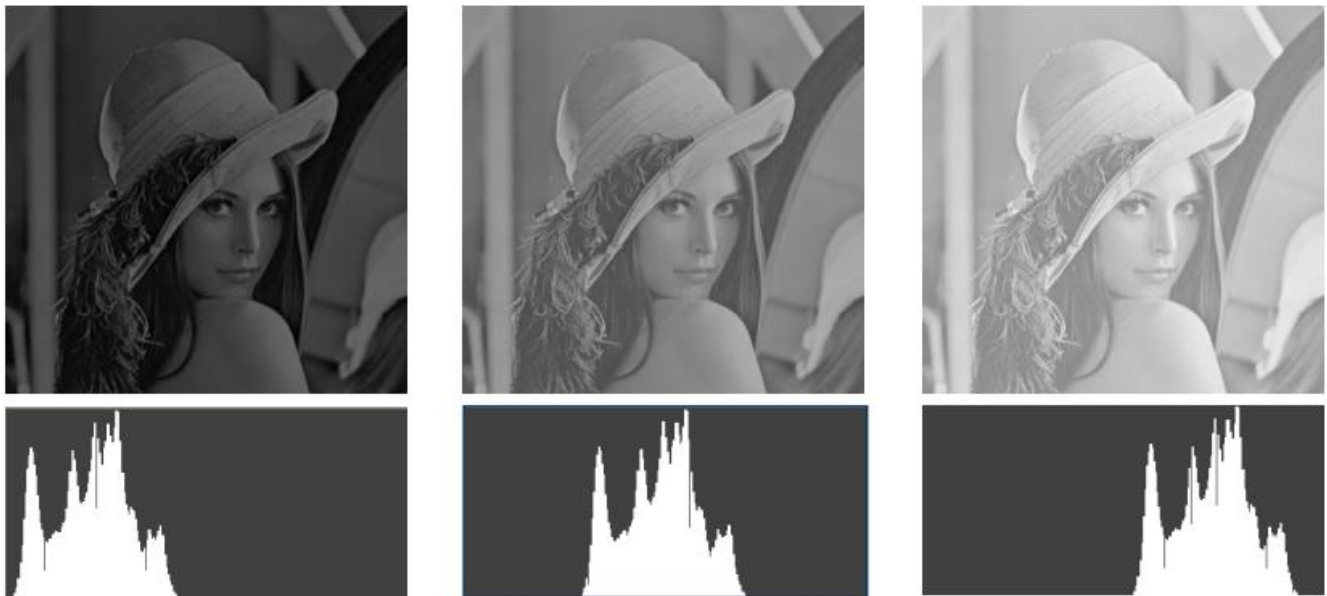
**Industria y manufactura:** En la industria, el procesamiento de imágenes se utiliza para el control de calidad, inspección de productos, detección de defectos, clasificación y reconocimiento de objetos. Ayuda a automatizar procesos y garantizar la consistencia y precisión en la producción.

**Seguridad y vigilancia:** Las técnicas de procesamiento de imágenes desempeñan un papel crucial en la seguridad y vigilancia, permitiendo la detección de intrusiones, seguimiento de objetos, reconocimiento facial, análisis de comportamiento y análisis forense de imágenes y videos.



Automoción: En la industria automotriz, el procesamiento de imágenes es esencial para sistemas de asistencia al conductor, como el reconocimiento de señales de tráfico, detección de peatones y vehículos, y alerta de colisión. Además, juega un papel importante en el desarrollo de vehículos autónomos.

Entretenimiento y realidad virtual: En el ámbito del entretenimiento, el procesamiento de imágenes se utiliza en efectos visuales, animación por computadora, realidad aumentada y realidad virtual para crear experiencias inmersivas y realistas.



*Figura 2.13: Imagen con distinta intensidad junto a su histograma respectivamente. (Wainschenker et al., 2011).*

## 2.7 Softmax

“La función softmax es una de las principales funciones utilizadas en los modelos de clasificación. Por lo general, se presenta al principio de una clase de aprendizaje automático. Toma como entrada un vector de longitud de valor real,  $n$  y lo normaliza en una distribución de probabilidad. Es fácil de entender e interpretar, pero en esencia hay algunos errores de los que uno debe estar al tanto. Esto incluye su implementación en la práctica, estabilidad numérica y aplicaciones.” (Ogun, 2020).

Softmax es una función no lineal, utilizada principalmente en la salida de clasificadores para la clasificación de clases múltiples. Dado un vector:

$$[x_1, x_2, x_3, \dots, x_n] \text{ para } i = 1, 2, \dots, n$$

La función softmax tiene la forma:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (\text{ec. 1})$$

*Donde:*

$\text{softmax}(x_i)$  = Es el valor resultante en la posición  $i$

$x_i$  = Es el valor en la posición del vector de entrada  $x$

$e$  = Es la base del logaritmo natural

$\sum_{j=1}^n e^{x_j}$  = Es la suma de las exponenciales de todos los elementos del vector

La función softmax toma el vector de entrada  $x$  y calcula la probabilidad de cada elemento  $x_i$  sea el valor máximo en función de la exponencial de esos valores. En la ilustración 2.14 se muestra un claro ejemplo de las matrices de unas imágenes al inicio y una vez aplicada la función softmax.

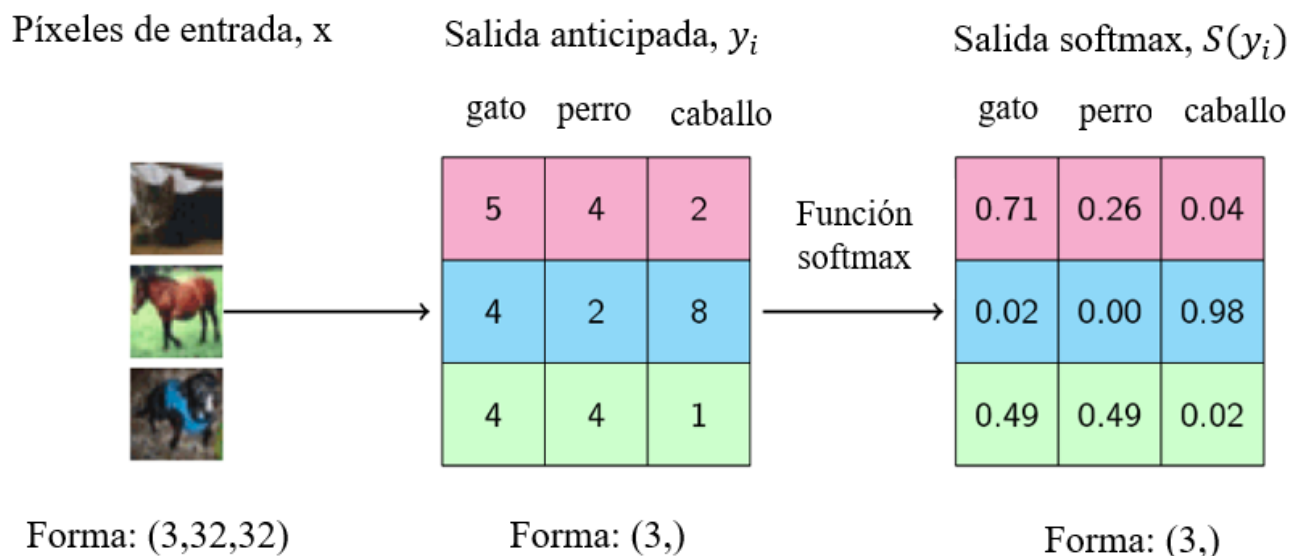


Figura 2.14: Normalización de los píxeles, usando la función softmax. (Ogun, 2020).

## 2.8 Histograma de ecualización

Este método lo que hace es aumentar el contraste global de muchas imágenes, sobre todo las que están representadas por un rango estrecho de valores de intensidad. Con este ajuste las intensidades en el histograma se pueden distribuir de mejor manera para abarcar el rango completo de manera uniforme. Ahora hay que explicar ciertas características que posee este método de procesamiento de imágenes, empezando con que permite que las áreas de menor contraste local obtengan un mayor contraste, esto se logra mediante la distribución efectiva de los valores de intensidad altamente poblados que se utilizan para degradar el contraste de la imagen, pero tenemos que considerar que no a todas las imágenes les viene bien este procesamiento. Si se va a hacer uso de este procesamiento, tenemos que verificar que la imagen cuente con un fondo y primeros planos que son tanto brillantes como oscuros, esto para que sirva mucho mejor esta técnica, ya que puede que se pierda información en caso de usarlo mal. Otro punto es que puede aumentar el contraste del ruido de fondo, mientras disminuye la señal utilizable. Una de las grandes ventajas es que no exige muchos recursos computacionales, lo cual viene de maravilla en caso de trabajar con muchas imágenes y de gran tamaño. (Kiliç, 2021).

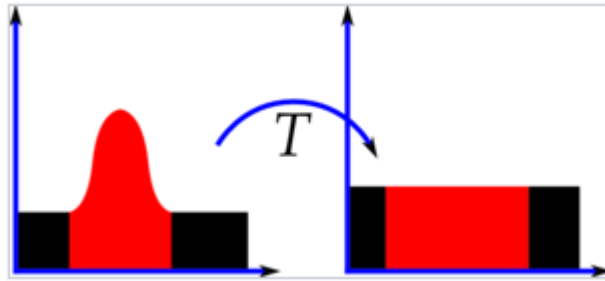


Figura 2.15: Histogramas de una imagen antes y después de la ecualización. (Kiliç, 2021).

Ahora pasamos a explicar la implementación una vez teniendo una imagen discreta en escala de grises  $\{x\}$  y sea  $n_i$  el número de ocurrencia del nivel de gris  $i$ . Bien, para mantener una cierta coherencia en la distribución acumulativa, es necesario tener un control de los valores de la imagen, por valores hablamos de las dimensiones de este. Teniendo una imagen en escala de grises como se muestra a continuación en la figura 2.16.

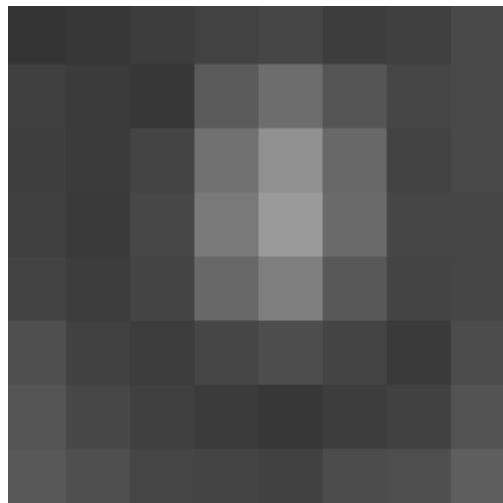


Figura 2.16: Sub imagen de 8 x 8 que está en escala de grises de 8 bits. (Gonzalez y Woods, 2018).

Ahora bien, es de esperarse que al ser una imagen de 8 x 8 se tengan 64 valores, en donde la idea es contar cada uno de los valores obtenidos, para poder hacer una suma de cuantos valores se tienen de un valor repetidos, una vez que se tengan se procede a obtener la función de distribución acumulada *cdf*.

$$cdf_y(i) = \sum_{j=0}^i p_x \quad (ec. 2)$$

Donde:

$cdf_y(i)$  = Función de distribución acumulada

$i$  = El valor del histograma en el nivel de intensidad  $i$ , es decir el número de píxeles

$\sum_{j=0}^i p^x$  = Es la suma de los valores del histograma de la imagen para el valor del píxel

Una vez que se tenga la cuenta de los valores obtenidos en una imagen pixel a pixel, se procede a obtener el valor mínimo de la sub imagen y el valor máximo. Para un mejor entendimiento se retoma una imagen como la mostrada en la figura 2.18 para proceder a la obtención de la distribución acumulada.

Tabla 3: Valores de la sub imagen en escala de grises de 8 bits.

<b>52</b>	<b>55</b>	<b>61</b>	<b>59</b>	<b>79</b>	<b>61</b>	<b>76</b>	<b>61</b>
<b>62</b>	59	55	104	94	85	59	71
<b>63</b>	65	66	113	144	104	6	72
<b>64</b>	70	70	126	154	109	71	69
<b>67</b>	73	68	106	122	88	68	68
<b>68</b>	79	60	70	77	66	58	75
<b>69</b>	85	64	58	55	61	65	83
<b>70</b>	87	69	68	65	73	78	90

Ya que se tienen los valores, se procede a hacer un recuento de los valores que se repiten y de paso se organizan del menor al mayor, así como se marca en la tabla 3 el valor máximo y el valor mínimo.

*Tabla 4: Valores ordenados y contados de las veces que se repiten.*

Valor	Contar	Valor	Contar	Valor	Contar	Valor	Contar	Valor	Contar
<b>52</b>	1	<b>64</b>	2	<b>72</b>	1	<b>85</b>	2	<b>113</b>	1
<b>55</b>	3	<b>65</b>	3	<b>7</b>	2	<b>87</b>	1	<b>122</b>	1
<b>58</b>	2	<b>66</b>	2	<b>75</b>	1	<b>88</b>	1	<b>126</b>	1
<b>59</b>	3	<b>67</b>	1	<b>76</b>	1	<b>90</b>	1	<b>144</b>	1
<b>60</b>	1	<b>68</b>	5	<b>77</b>	1	<b>94</b>	1	<b>154</b>	1
<b>61</b>	4	<b>69</b>	3	<b>78</b>	1	<b>104</b>	2		
<b>62</b>	1	<b>70</b>	4	<b>79</b>	2	<b>106</b>	1		
<b>63</b>	2	<b>71</b>	2	<b>8</b>	1	<b>109</b>	1		

Lo siguiente es la obtención de la distribución acumulada para los valores, en este caso se hace para los distintos valores obtenidos de la imagen, este proceso se va a mostrar para algunos valores como ejemplo y se explicará el procedimiento para la obtención de esta.

*Tabla 5: Distribución acumulada de los píxeles.*

<b><math>v</math>, intensidad de píxeles</b>	<b><math>cdf(v)</math></b>	<b><math>h(v), v</math> igualada</b>
<b>52</b>	1	0
<b>55</b>	4	12
<b>58</b>	6	20
<b>59</b>	9	32
<b>...</b>	<b>...</b>	<b>...</b>
<b>154</b>	64	255

Para la obtención de los valores de  $cdf(v)$  y  $h(v)$  se emplean las siguientes ecuaciones:

$$h(v) = \text{redondeo} \left( \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \right) \times (L - 1) \quad (ec. 3)$$

*Donde:*

$h(v)$  = Es la ecualización del histograma

$cdf_{min}$  = Valor mínimo distinto de cero de la función de distribución acumulativa

$cdf(v)$  = La distribución acumulada para el valor deseado

$M$  = Es el ancho de la imagen

$N$  = Es la altura de la imagen

$L$  = Nivel de gris utilizado (En la gran mayoría es 256)

Ahora mostramos en la *ec. 4* y *ec. 5* el ejemplo empleado para la obtención de los valores 52 y 55.

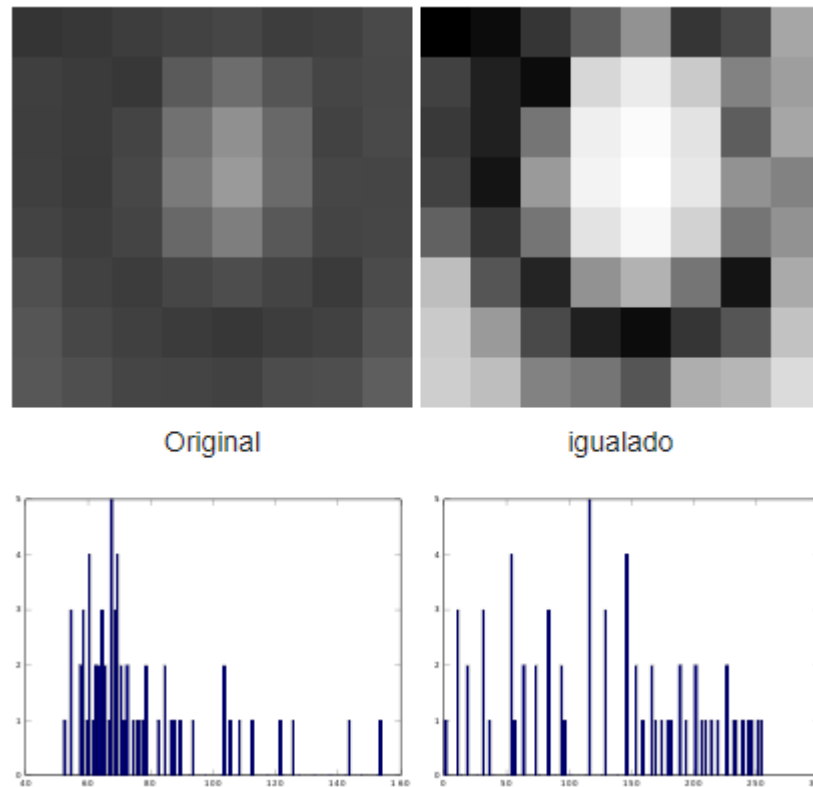
$$h(52) = \text{redondeo} \left( \frac{1 - 1}{(63) - 1} \right) \times (255) = 0 \quad (ec. 4)$$

$$h(55) = \text{redondeo} \left( \frac{4 - 1}{(63) - 1} \right) \times (255) = 12 \quad (ec. 5)$$

Para corroborar que nuestros cálculos fueron correctos, procedemos a hacer el cálculo para el último valor, en este caso es 154, en donde la función de distribución acumulada debe darnos el valor correspondiente al número de píxeles, en este caso es 255.

$$h(154) = \text{redondeo} \left( \frac{64 - 1}{(63) - 1} \right) \times (255) = 255 \quad (ec. 6)$$

Lo siguiente una vez obtenido los valores es reemplazarlos y de esta manera podremos visualizar la imagen una vez aplicada la ecualización de histograma junto a sus respectivos histogramas mediante un proceso en cualquier lenguaje de programación en el que se aplique el histograma de ecualización, donde cabe recalcar que ahora tendremos un valor máximo de 154 (anteriormente era 255) y un valor mínimo de 52 (anteriormente era 0).



*Figura 2.17: Comparativa de la imagen una vez que se aplica el filtro de histograma de ecualización.  
(Gonzalez y Woods, 2018).*

Se puede aplicar este procesamiento de imagen a imágenes de color, sin embargo, aplicar el método en los componentes rojo, verde y azul de una imagen RGB puede cambiar el balance de color de forma drástica debido a las diferencias en las distribuciones de los canales de color. Para evitar este problema, se sugiere convertir la imagen a un espacio de color diferente, como el espacio de color Lab o HSL/HSV, antes de aplicar el algoritmo.



## 2.7 Ajuste de contraste en imágenes

Comencemos hablando sobre que es el contraste en imágenes, es el cambio local de brillo y se define como la relación entre el brillo promedio de un objeto y el fondo, añadido a esto, el ojo humano es logarítmicamente sensible al brillo, lo que implica que, para la misma percepción, un mayor brillo requiere un mayor contraste.

El brillo aparente depende en gran medida del brillo del entorno local, este efecto se llama contraste adicional. En la figura 2.18 se ilustra lo anteriormente mencionado con cinco círculos del mismo tamaño rodeados por cuadrados de diferente brillo, los humanos perciben el brillo de los pequeños círculos como diferentes. (Sonka et al., 2013).



*Figura 2.18: Efecto contraste condicional. (Sonka et al., 2013).*

Algo que se tiene que dejar muy en claro es que el histograma de ecualización y el ajuste de contraste son dos técnicas relacionadas pero diferentes para mejorar la calidad visual de una imagen, sin duda la implementación es muy parecida ya que al final trabajan con imágenes y van trabajando píxel a píxel, pero al final tienen objetivos diferentes y se pueden usar cualquiera de estos métodos de mejora de imagen en base a lo que se busca obtener.

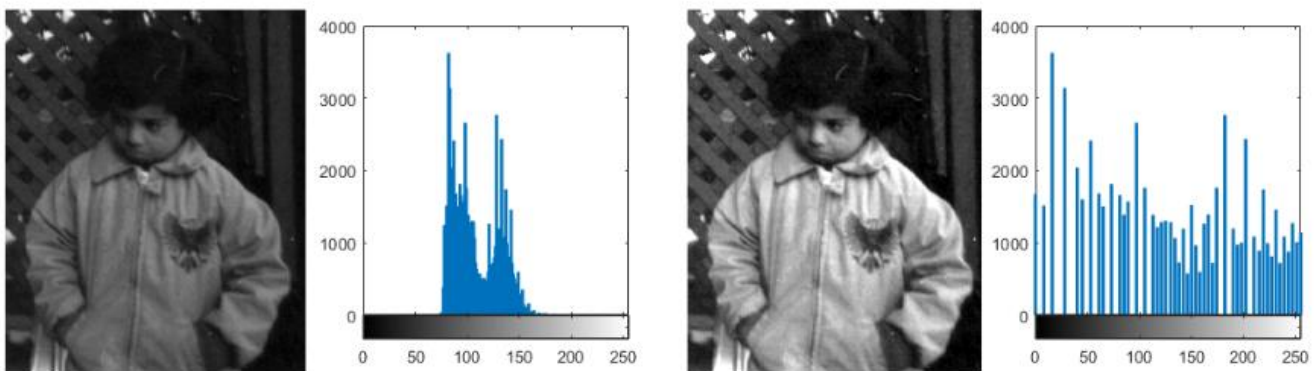
Hablando por el objetivo de estos, cabe mencionar que el histograma de ecualización busca redistribuir la distribución de los niveles de gris en una imagen de manera uniforme, lo que resulta en un aumento global del contraste, esta técnica tiende a hacer que todos los niveles de gris sean igualmente representados en el histograma, lo que es bueno y malo ya que depende el enfoque que se busque, es que puede hacer que los detalles se destaquen en áreas previamente subexpuestas o sobreexpuestas. Por otro lado, el ajuste de contraste se centra en aumentar o disminuir la diferencia entre las áreas claras y oscuras de una imagen sin necesariamente redistribuir todos los niveles de gris de manera uniforme. Puede utilizarse para acentuar regiones específicas de la imagen o para lograr un efecto creativo.

También hay que mencionar que la ecualización del histograma tiene un impacto global en toda la imagen y puede afectar negativamente áreas que ya tenían un buen contraste, mientras

que el ajuste de contraste se puede aplicar de manera local o selectiva en áreas específicas de la imagen. Esto permite un mayor control sobre cómo se ajusta el contraste en diferentes partes de la imagen, lo que lo hace más versátil en situaciones donde se desean resultados específicos

En resumen, el histograma de ecualización es una técnica global que redistribuye uniformemente los niveles de gris en una imagen, mientras que el ajuste de contraste es más versátil y permite un control más específico sobre cómo se ajusta el contraste en las diferentes partes de la imagen. La elección de que técnica utilizar al final queda en el aire ya que dependerá completamente de los objetivos de procesamiento de la imagen y del efecto deseado.

Una imagen con buen contraste tiene diferencias claras entre el negro y el blanco, en la figura 2.19 tenemos un ejemplo en donde la imagen de la izquierda tiene poco contraste y los valores de intensidad en el histograma se limitan al centro del intervalo, mientras que la imagen de la derecha tiene un contraste más alto y sus valores abarcan todo el intervalo de intensidad [0,255]. (MathWorks, ajuste del contraste)



*Figura 2.19: En la imagen de alto contraste, las zonas claras se ven más brillantes y las sombras se ven más oscuras. (MathWorks, 2023).*

## CAPÍTULO 3

### 3 METODOLOGÍA

El desarrollo del proyecto de “aplicación móvil para detección temprana de melanoma”, logró crear una aplicación móvil para el sistema operativo de Android. La interfaz de la aplicación se diseñó de manera sencilla y amigable para el usuario, proporcionando opciones para seleccionar imágenes de la galería y someterlas a un procesamiento de imágenes. Estas imágenes se cargaban en Firebase y se procesaban utilizando el modelo "Identifying Melanoma Images using EfficientNet Ensemble." (Ha et al., 2020), en Colaboratory, lo que permitía clasificarlas como posibles casos de melanoma. Es importante destacar que el proyecto se centró exclusivamente en el aspecto del software, incluyendo la creación de la aplicación móvil, la configuración de la conexión con los servicios de Firebase (Storage y Realtime Database), la transferencia del modelo a Colaboratory para su ejecución, y la aplicación de procesamientos de imágenes para mejorar el rendimiento del modelo. La aplicación se probó específicamente con la base de datos del desafío SIIM-ISIC. Se buscó lograr una eficiencia superior a la del modelo original, y se llevaron a cabo análisis de los pesos obtenidos para mejorar la capacidad de distinguir entre casos malignos y benignos. En la figura 3.1 se muestra el diagrama de proceso que detalla cada etapa del desarrollo del proyecto.

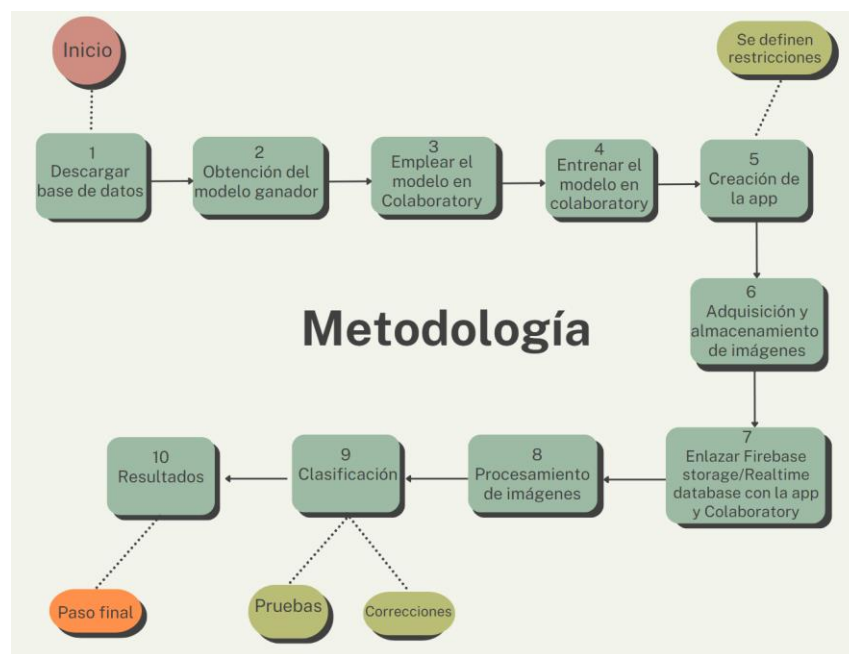


Figura 3.1: Metodología propuesta (autoría propia).

### **3.1 Descargar base de datos**

La Figura 3.1 consta de diez pasos, los cuales comienzan con la obtención de la base de datos, la cual es de acceso libre puesto que fueron las imágenes usadas en el desafío, la cual se obtiene en la página oficial del desafío, la cual lleva por nombre “The ISIC 2020 Challenge Dataset”, cabe mencionar que la base de datos para realizar las pruebas consta de 10,982 imágenes con sus respectivos resultados, en donde se contó con 33,126 imágenes para el entrenamiento de los modelos, para ello se tienen imágenes clasificadas en los siguientes rubros:

- Clasificación
- Sexo del paciente
- Edad aproximada
- Diagnóstico

Cada una de las imágenes se encuentra en el formato JPG, por lo que el entrenamiento se va a basar en el uso de ese formato para las imágenes. En la parte de referencias se encuentra la página para encontrar la base de datos. (Rotemberg et al., 2021)

### **3.2 Obtención de "Identifying Melanoma Images using EfficientNet Ensemble."**

Para acceder al modelo ganador, se visitó la página donde se publicó el anuncio del desafío de SIIM-ISIC melanoma classification. En este sitio web, se encontraba la descripción general del desafío, así como ciertas pautas y la tabla de clasificación de los participantes. Dentro de esta tabla, se destacaba el trabajo realizado por Qishen Ha et al. (SIIM-ISIC, 2020), que previamente se mencionaron en la sección de estado del arte relacionada con este desafío.

Para llevar a cabo este acceso a los diversos proyectos, se utilizó la plataforma "Kaggle". Kaggle es una plataforma de competencias científicas de datos y una comunidad en línea que reúne a científicos de datos y profesionales del aprendizaje automático, respaldada por Google LLC, como se mencionó el modelo es de acceso libre.










#	△	Team	Members	Score	Entries	Last	Solution
1	▲ 880	All Data Are Ext		0.9490	116	3y	
2	▲ 55	aloe		0.9485	61	3y	
3	▲ 262	Deloitte Analytics Spain		0.9484	118	3y	
4	▲ 210	Atagi Yuya		0.9476	23	3y	
5	▲ 723	Wenlu		0.9475	19	3y	

Figura 3.2: Tabla de mejores modelos desarrollados. (SIIM-ISIC, 2020).

### 3.3 Emplear el modelo en Colaboratory

Luego de revisar detenidamente el modelo en la plataforma de Kaggle, se procedió a transferir ese código a Colaboratory. La elección de Colaboratory se debió a su capacidad para ofrecer recursos computacionales adecuados para trabajar con Machine Learning. Además, se optó por utilizar los servicios proporcionados por Google debido a su gratuidad y a la conveniencia de contar con un ecosistema que facilitara la integración de los servicios utilizados.

Un aspecto esencial fue la investigación previa para asegurarse de que Colaboratory permitiera el uso de librerías clave como Tensorflow y PyTorch, que son ampliamente utilizadas en el ámbito del Machine Learning. También se investigó la manera de instalar y utilizar las versiones apropiadas de estas librerías, ya que, dado que están en constante evolución, ciertas actualizaciones pueden ocasionar problemas. Sin embargo, en este caso, no se encontraron obstáculos insuperables, y cualquier problema que surgió se resolvió consultando plataformas como StackOverflow.

Con el objetivo de optimizar la eficiencia y aprovechar las ventajas proporcionadas por las plataformas de Google, se tomó la decisión de conectar Colaboratory con Google Drive. Esta elección se hizo con el propósito de simplificar el proceso de carga de pesos y asegurar que las imágenes destinadas a pruebas se encontraran en una carpeta dentro de Google Drive. Es importante destacar que cada vez que se ejecuta el modelo en Colaboratory por primera vez, es necesario otorgar acceso a la cuenta para que pueda acceder a los archivos mencionados anteriormente. Esto garantiza una interacción fluida entre Colaboratory y Google Drive, facilitando así el flujo de datos necesario para el proyecto.

En cuanto a la explicación del código empleado, después de cargar las librerías necesarias, se procede a definir una serie de variables clave. Esto incluye el tamaño de las

imágenes, establecido en 640 píxeles, así como la especificación de que la CPU de Colaboratory será utilizada para almacenar y ejecutar el modelo. La variable que almacena la dimensión de salida del modelo se establece en 9, lo que probablemente se refiere a la cantidad de clases o categorías diferentes que el modelo puede predecir. Además, se menciona que se utilizará la arquitectura B7 del modelo EfficientNet, según lo empleado en el trabajo de Qishen Ha et al. Los pesos del modelo entrenado se encuentran en Google Drive, y se configura la manera de acceder y utilizar estos pesos en el código. Finalmente, se establece que se realizarán un total de ocho pruebas, y la razón detrás de esta cantidad se explicará posteriormente en el código. Estas variables son esenciales para la correcta ejecución del código y garantizar resultados precisos.

Existe una clase denominada "SIIMISICDataset" que desempeña varias funciones fundamentales. Esta clase utiliza un DataFrame de pandas para almacenar información relacionada con las imágenes y una variable que permite definir si se trata de un conjunto de entrenamiento o de prueba. Además, cuenta con métodos especiales que resultan esenciales en el preprocesamiento de datos. Uno de estos métodos especiales se encarga de calcular y devolver la cantidad de elementos en el conjunto de datos, basándose en el número de filas del DataFrame. Otro método especial proporciona acceso a un elemento específico del conjunto de datos a través de un índice, devolviendo tanto la imagen como sus etiquetas correspondientes. Esta clase se muestra altamente valiosa en el contexto del preprocesamiento de imágenes para su uso en modelos de aprendizaje automático, especialmente en el marco de PyTorch.

En el siguiente segmento, se presenta la clase denominada "enetv2", diseñada especialmente para tareas de clasificación. Esta clase incorpora una arquitectura Backbone utilizada en redes neuronales, respaldada por la librería "geffnet". Además, se define el número de clases de salida y la cantidad de características meta que se pueden incluir en el modelo. También se utiliza un indicador booleano para determinar si se deben cargar los pesos pre-entrenados para el modelo Backbone. Con base en esta descripción, se puede concluir que esta clase define un modelo de red neuronal personalizado que aprovecha un modelo pre-entrenado como Backbone, reemplaza la capa clasificadora original y agrega su propia capa de salida para adaptarse a una tarea de clasificación específica.

Posteriormente, se encuentra una lista de modelos que se utilizan en el contexto del aprendizaje automático. Se implementa un ciclo for que permite el acceso a los preentrenamientos descargados. Esta etapa se considera una de las más demandantes desde el

punto de vista computacional, lo que motiva el uso de la GPU proporcionada por Colaboratory. Esta elección tendrá un impacto significativo en la parte final del proceso, donde se abordará el tiempo total de ejecución. En resumen, esta sección carga los cinco modelos, cada uno con sus pesos pre-entrenados, los traslada al dispositivo especificado y los incorpora a una lista denominada "models" para su uso posterior, como la realización de predicciones en un conjunto de datos.

En el siguiente fragmento de código, se encuentra una función encargada de realizar operaciones de transposición y volteo en la imagen previamente procesada. Cabe destacar que se había definido inicialmente una cantidad de pruebas igual a ocho, debido a que se generan distintas orientaciones de la imagen, incluyendo rotaciones de 90° y sus respectivas versiones reflejadas, obteniendo así un total de ocho posiciones diferentes de la imagen. Posteriormente, se aplica una normalización a la imagen, una operación común que implica restar la media y dividir por la desviación estándar de las intensidades de píxeles en la imagen para estandarizar sus valores.

La siguiente parte del proceso involucra datos CSV (Comma Separated Values) almacenados en un archivo llamado "datasettesting". En este archivo, se escribe una fila de encabezados seguida de una fila de datos, y luego se cargan estos datos en un DataFrame de pandas. El propósito de esta operación es crear y manipular un conjunto de datos en formato CSV.

La última etapa comprende la preparación de un conjunto de datos de prueba que se utilizará en la clase inicial denominada "SIIMISICDataset". Tras cargar la primera imagen del conjunto de datos, se traslada al dispositivo especificado y se le agrega una dimensión adicional para que sea compatible con la mayoría de los modelos de aprendizaje profundo. Esto se logra mediante la función "unsqueeze". Este proceso de preprocesamiento es necesario para alimentar la imagen al modelo y realizar inferencias. Para calcular la probabilidad, se utiliza el modelo denominado "models", que acumula probabilidades de las clases a lo largo de las transformaciones y modelos, para finalmente promediar la probabilidad y obtener un resultado final.

### 3.4 Entrenar el modelo en Colaboratory

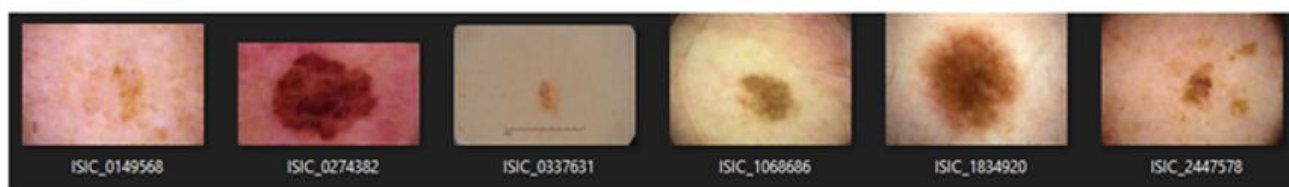
En el contexto del entrenamiento realizado en Colaboratory, se empleó la base de datos de SIIM-ISIC, abordando casos tanto malignos como benignos. Este enfoque se adoptó con el propósito de evaluar los pesos obtenidos y comprender cómo el modelo realiza clasificaciones en diversos escenarios. Además, este proceso sirvió para garantizar que todo estuviera en orden al momento de llevar el código a Colaboratory.

La ejecución de las pruebas se simplificó mediante la carga de las imágenes y un ajuste en la ruta de acceso en el código, lo que permitió llevar a cabo el proceso de manera efectiva. A continuación, en la figura 3.3, se presentan las imágenes utilizadas junto con los pesos obtenidos, tanto para los casos malignos como para los benignos. Este análisis visual resulta fundamental para comprender y validar el desempeño del modelo en situaciones reales.



Pesos con el modelo (casos benignos):

0.00076    0.00346    0.00146    8.68E-05    0.00083    0.01214



Pesos con el modelo (casos malignos):

0.085892    0.731433    0.086202    0.564551    0.001750    0.565801

*Figura 3.3: Pesos obtenido con el modelo. (autoría propia).*

Al observar los pesos obtenidos, se nota que los casos de melanomas benignos muestran valores pequeños, generalmente en el rango de las milésimas, excepto un caso que se encuentra en el rango de las centésimas. Por otro lado, los casos malignos presentan valores más



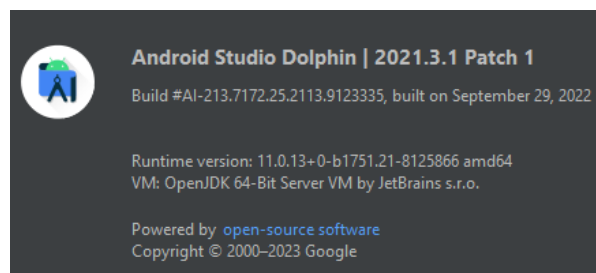
elevados, ubicándose en las décimas y centésimas. Aunque la diferencia es evidente, es importante destacar que existen valores que se solapan, lo que sugiere la necesidad de realizar ajustes específicos para este tipo de casos. También es posible que estos solapamientos se deban al margen de error del 4% mencionado. Se llevaron a cabo pruebas adicionales, y la tendencia general se mantuvo similar, con algunos casos que cumplen con las características mencionadas anteriormente y otros que presentan solapamientos en función de la clasificación propuesta.

Con este punto de partida, se puede avanzar hacia el trabajo con filtros con el objetivo de obtener resultados más distintivos y lograr una mayor diferenciación entre los casos benignos y malignos. Este enfoque en el refinamiento de los filtros puede contribuir a mejorar la precisión de la clasificación.

### 3.5 Creación de la aplicación

En relación a la creación de la aplicación, es fundamental destacar que este proceso abarca varios pasos, desde el diseño de la interfaz hasta la programación y la implementación del procesamiento de imágenes. Por lo tanto, se explicará minuciosamente cada sección correspondiente. En esta etapa inicial, se abordará el trabajo previo a la creación de la aplicación, incluyendo la consideración de permisos y el diseño utilizado.

Comenzando con las especificaciones iniciales de la aplicación, es importante definir ciertos factores antes de iniciar la programación. En primer lugar, se establece que la aplicación estará destinada al sistema operativo Android, lo que implica el uso de Android Studio como entorno de desarrollo para el proyecto. La versión específica empleada a lo largo de todo el trabajo fue la versión 2021.3.1. Esto es relevante mencionarlo debido a las actualizaciones frecuentes en controladores y configuraciones del entorno, lo que podría dar lugar a algunas diferencias si se trabaja con una versión distinta. La elección de la versión se hizo teniendo en cuenta la estabilidad y compatibilidad necesarias para el proyecto.



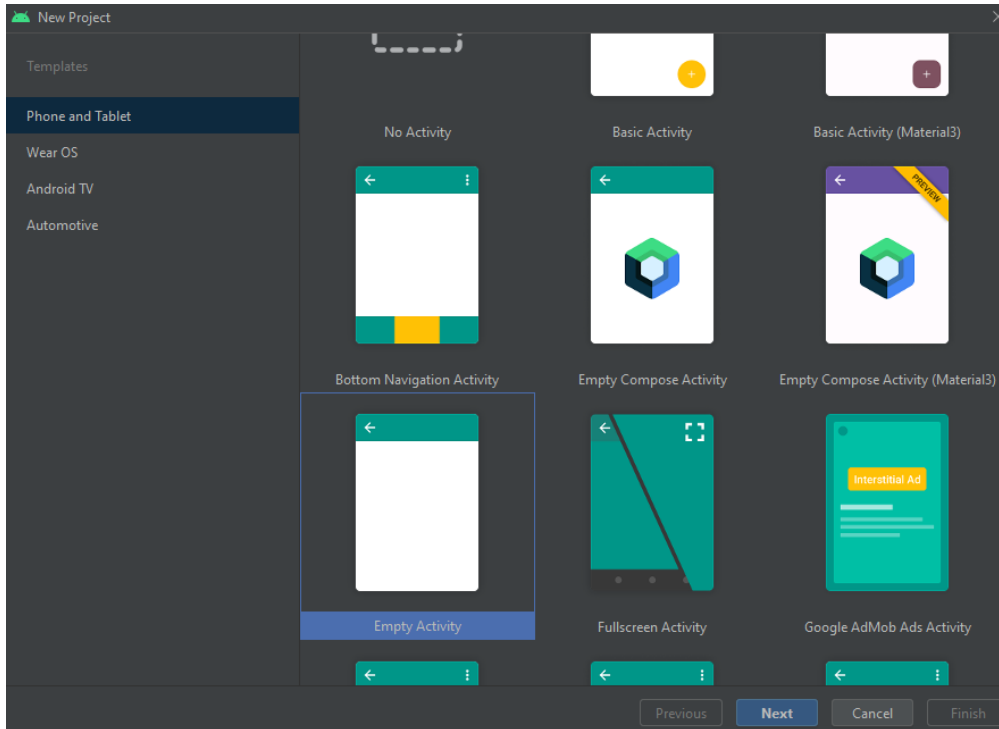
*Figura 3.4: Versión utilizada de Android Studio. (autoría propia).*

La versión del Android SDK utilizada corresponde a la versión 33, un detalle que se aclarará más adelante en el proceso de configuración de la aplicación móvil. Ahora, después de especificar la versión de Android Studio, se procede a definir las configuraciones iniciales y la base de la aplicación. En este caso, el lenguaje de programación seleccionado es Java. Además, se estableció que la aplicación sería compatible con cualquier dispositivo Android que tenga al menos la versión 5.0 (Lollipop) del sistema operativo. Esta elección se hizo debido a que esta versión abarca aproximadamente el 99.5% de los dispositivos móviles en el mercado, lo que garantiza una amplia compatibilidad con los dispositivos actuales, incluso considerando que la versión de Android ha avanzado hasta la versión 13. Además, se destaca que la aplicación será compatible con cualquier capa de personalización, como la que utiliza la marca Xiaomi con MIUI, ya que, en última instancia, siguen siendo dispositivos Android en cuanto a sistema operativo.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.4 KitKat	19	
5.0 Lollipop	21	99.5%

*Figura 3.5: API LEVEL mínimo para la versión y el porcentaje que abarca la versión 5.0. (autoría propia).*

El proyecto se inicia con una ventana en blanco debido a que los elementos se irán agregando según las necesidades de la aplicación. Por esta razón, se selecciona la opción "Empty Activity". Es fundamental destacar que Java es un lenguaje de programación orientado a objetos que posibilita el desarrollo de aplicaciones en diversas áreas. Esta elección permite una mayor flexibilidad para personalizar y construir la aplicación de acuerdo con los requisitos específicos.



*Figura 3.6: Template utilizado. (autoría propia).*

Es esencial tener en mente que dentro del proyecto nos encontramos con varias ventanas, y estas desempeñarán un papel fundamental en el desarrollo de la aplicación. Los elementos que se modificaron en este contexto incluyen los siguientes: MainActivity, AlertDialog, build.gradle(app), build.gradle(NombreProyecto), AndroidManifest y activity\_main. Cada uno de estos apartados se ajustará y personalizará según las necesidades específicas de la aplicación, contribuyendo al desarrollo integral del proyecto.

El primer paso se centra en el diseño de la interfaz, y se optó por algo sencillo. La interfaz se concibió como una ventana que muestra información relevante sobre la universidad y la facultad, además de contar con botones para llevar a cabo las acciones necesarias. Concretamente, se implementaron botones para abrir la galería, procesar la imagen y obtener y visualizar el resultado. Además, se incluyó un botón que permite acceder a información detallada sobre la aplicación. Es importante destacar que se indicó claramente que se trata de un proyecto de tesis y, por lo tanto, la aplicación se limita al uso dentro de la facultad, siguiendo criterios previamente establecidos. También se proporcionó información sobre la clasificación que la aplicación puede proporcionar y se enfatizó que la aplicación no sustituye la opinión de un

especialista. Se hizo hincapié en la recomendación de visitar a un especialista en caso de tener la más mínima duda sobre una mancha en la piel. Además, se consideró valioso mostrar la imagen cargada al modelo para que el usuario tenga confianza en la imagen seleccionada. Esta aproximación en el diseño de la interfaz busca proporcionar una experiencia de usuario clara y útil.

Para la personalización de la pantalla principal, lo necesario fue definir el nombre de la aplicación, el logo y los nombres de los elementos que se utilizarán posteriormente en la programación. Para abordar el diseño de la interfaz, se trabajará en el archivo "activity\_main", que es donde se ubicarán y nombrarán los elementos que se utilizarán en la programación.

En primer lugar, se añadirán dos ImageView, que se utilizarán para mostrar imágenes. Uno de ellos se utilizará para representar la imagen correspondiente a la universidad y la facultad, mientras que el otro se utilizará para mostrar la imagen seleccionada por el usuario. Además, se incorporarán los cuatro botones mencionados anteriormente. Los botones contarán con texto para facilitar su identificación por parte del usuario.

En la parte inferior de la aplicación, se incluirá un elemento de texto que proporcionará información sobre su servidor, en caso de que los usuarios necesiten ayuda o tengan alguna pregunta relacionada con el proyecto. Esta configuración del diseño de la interfaz tiene como objetivo crear una experiencia de usuario intuitiva y eficiente.

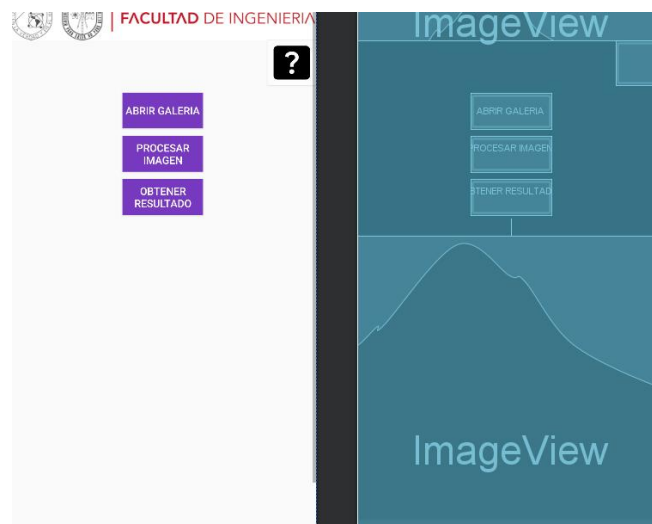


Figura 3.7: Diseño de la página principal de la aplicación. (autoría propia).

En la Figura 3.7, se puede apreciar la interfaz de la aplicación. En el lado derecho de la figura, se muestran los componentes de la interfaz, mientras que en el lado izquierdo se visualiza cómo se presentaría al usuario. A continuación, se mostrará el árbol de componentes en la Figura 3.8, que corresponde a los componentes utilizados en la interfaz. Es importante mencionar que se especificaron propiedades necesarias para cada elemento, lo cual es crucial para la correcta orientación de los elementos y su manipulación en el MainActivity.

En el proceso de desarrollo, se proporcionará una explicación general sobre el uso de estas propiedades para aclarar cualquier duda. Además, el código completo estará disponible en las referencias. La visualización de los componentes y su configuración contribuye a una comprensión clara de la interfaz de usuario y su funcionalidad.

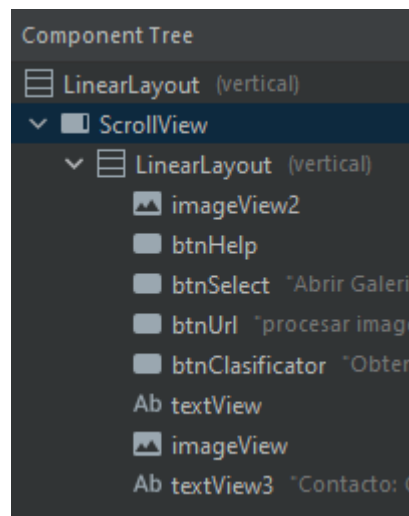


Figura 3.8: Nombre de los componentes utilizados. (autoría propia).

Al adentrarnos en el ámbito de la programación, es crucial mencionar algunas propiedades fundamentales. Comenzando con la propiedad "android:layout", la cual actúa como un contenedor para una o más vistas y controla su comportamiento y posición en la interfaz. En cada componente de la interfaz, esta propiedad se utilizó tanto para definir su longitud como su anchura. Además, la propiedad "android:orientation" se empleó para establecer la orientación de los elementos. En este caso, todos los elementos están dispuestos verticalmente, por lo que se definió la orientación correspondiente para cada uno de ellos.

En cuanto al "ScrollView" en Android, es uno de los componentes que se encuentra dentro de "ViewGroup". En esencia, proporciona una jerarquización que organiza los diferentes elementos de la interfaz. Para lograr esto, es importante comprender los diferentes tipos de "ViewGroup" disponibles, que incluyen "FrameLayout", "LinearLayout", "RelativeLayout" y "ConstrainLayout". Cada uno de estos tipos ofrece ventajas específicas en cuanto al diseño, pero todos ellos se utilizan para distribuir las vistas en el "ViewGroup". Por ejemplo, el "LinearLayout" coloca a sus hijos uno detrás de otro, ya sea en sentido vertical u horizontal, lo que influye en la disposición de los elementos en la interfaz.

Dentro de los ajustes relacionados con los botones, existe una propiedad de gran importancia relacionada con el color de fondo y el color de las letras. Para modificar estos colores, es necesario utilizar la palabra "android:" seguida de la propiedad correspondiente. Android Studio proporciona una gran ayuda en este proceso, ya que ofrece acceso a diversas propiedades y colores de manera interactiva, lo que facilita la personalización de la interfaz.

Se utilizaron las propiedades "android:background" y "android:textColor". Es importante destacar que, para modificar los colores, es necesario utilizar la representación hexadecimal del código de color, que representa el formato RGB mediante la combinación de tres valores. Hay dos formas de acceder a diferentes colores: la primera es buscar el código de color en línea, y la segunda es seleccionar un color de la paleta de colores que Android Studio muestra al modificar el diseño en la parte de "activity\_main".

Por último, es esencial asignar nombres a los diversos elementos, especialmente aquellos que se manipularán o que realizarán acciones específicas. Para lograr esto, se debe cambiar el ID utilizando la línea "android:id". Es importante asegurarse de que no haya elementos con el mismo nombre para evitar confusiones al escribir el código. El valor del atributo "id" indica que la constante llamada "text" se utilizará exclusivamente para identificar esa vista.

En cuanto al nombre y al logo de la aplicación, se optó por un enfoque que reflejara tanto el tema principal como la institución de la universidad, con el objetivo de que el nombre fuera claro y conciso. De esta manera, se creó el nombre "MelaUAQ". El diseño del logo se mantuvo bastante sencillo, utilizando el mismo nombre en el logo y empleando un fondo blanco con el nombre resaltado en rosa. Esta elección de diseño buscó transmitir de manera efectiva la temática de la aplicación y la afiliación a la universidad.



*Figura 3.9: Logo creado para la aplicación. (autoría propia).*

Para realizar la modificación del diseño del icono, en este caso, contando ya con el diseño por parte del desarrollador, se procede a cargar la imagen para utilizarla como logotipo en Android Studio. Este proceso se inicia al acceder a la pestaña "View" y buscar la opción "Tool Windows". Una vez seleccionada, se desplegarán diversas alternativas, y es aquí donde se localiza la función "Resource Manager". Este proceso permite personalizar el logotipo de la aplicación y asegurarse de que se ajuste a las preferencias y necesidades del proyecto en desarrollo.

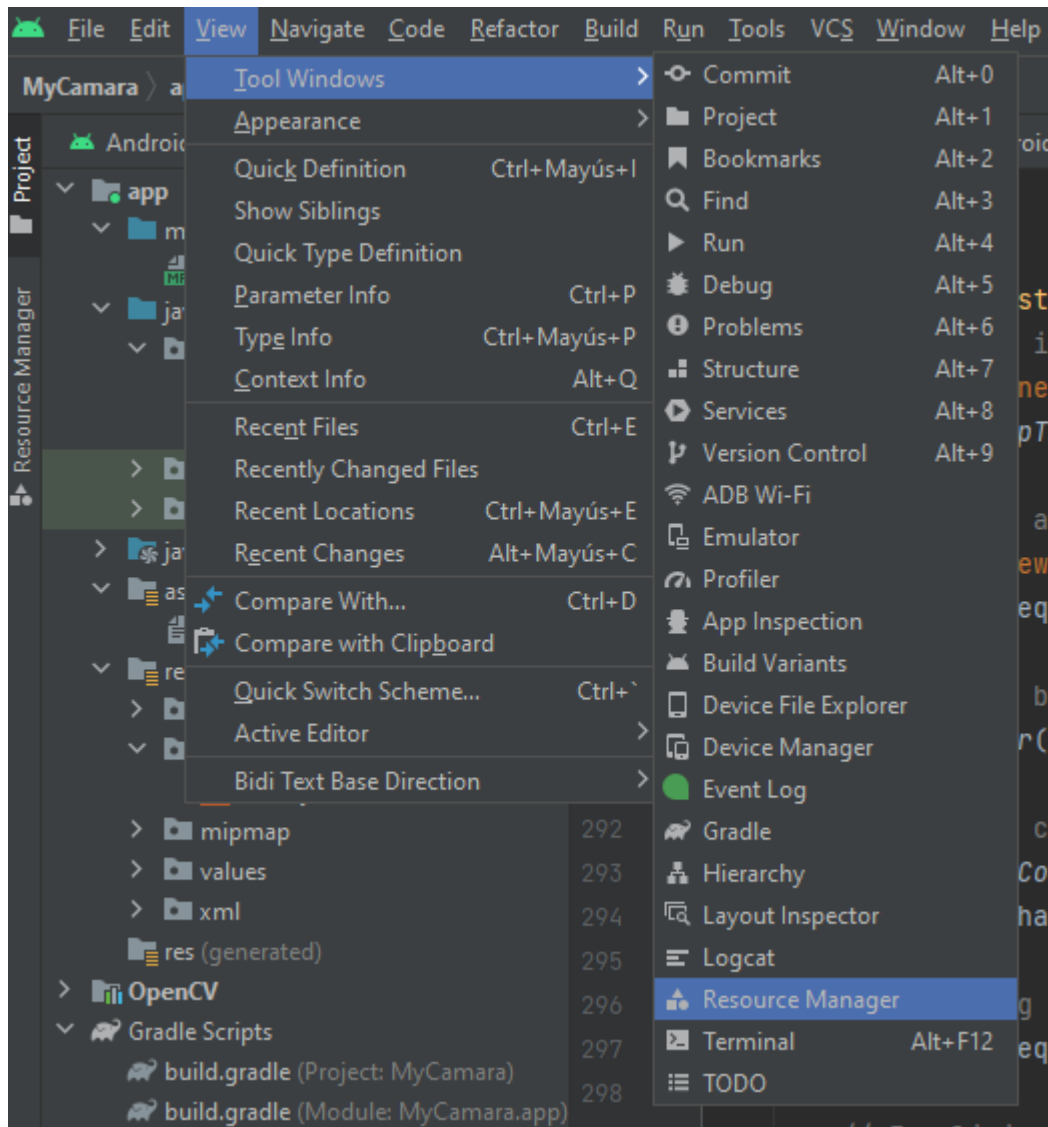


Figura 3.10: Ubicación de Resource Manager. (autoría propia).

Luego de este paso, se abrirá una nueva ventana que permite agregar un nuevo diseño de icono. Tienes la opción de crearlo utilizando las herramientas y opciones proporcionadas por Android Studio o de importar un diseño previamente creado. Para ello, puedes hacer clic en el icono de "+" y luego seleccionar "Image Asset". Esto te permitirá personalizar el diseño del icono de la aplicación de acuerdo a tus preferencias y necesidades, ya sea creándolo desde cero o importando un diseño existente.



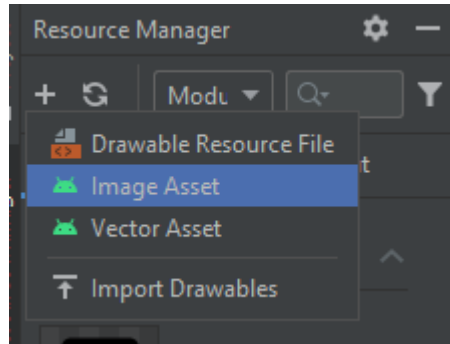


Figura 3.11: Ventana emergente de Resource Manager. (autoría propia).

En este punto, simplemente debes seleccionar la imagen o crear el diseño y definir la forma que tendrá el logo de la aplicación. Gracias a las últimas actualizaciones de Android, tienes la capacidad de dar forma al logo, ya sea circular, redondeado, cuadrado, y también puedes elegir opciones para darle un fondo diferente o centrar la imagen cargada. En este caso, se optó por un diseño redondeado con un fondo de color blanco para que coincida con los colores del logotipo de la aplicación. Esta flexibilidad en la personalización del diseño del logo permite adaptarlo a las preferencias y estilo de la aplicación que estás desarrollando.

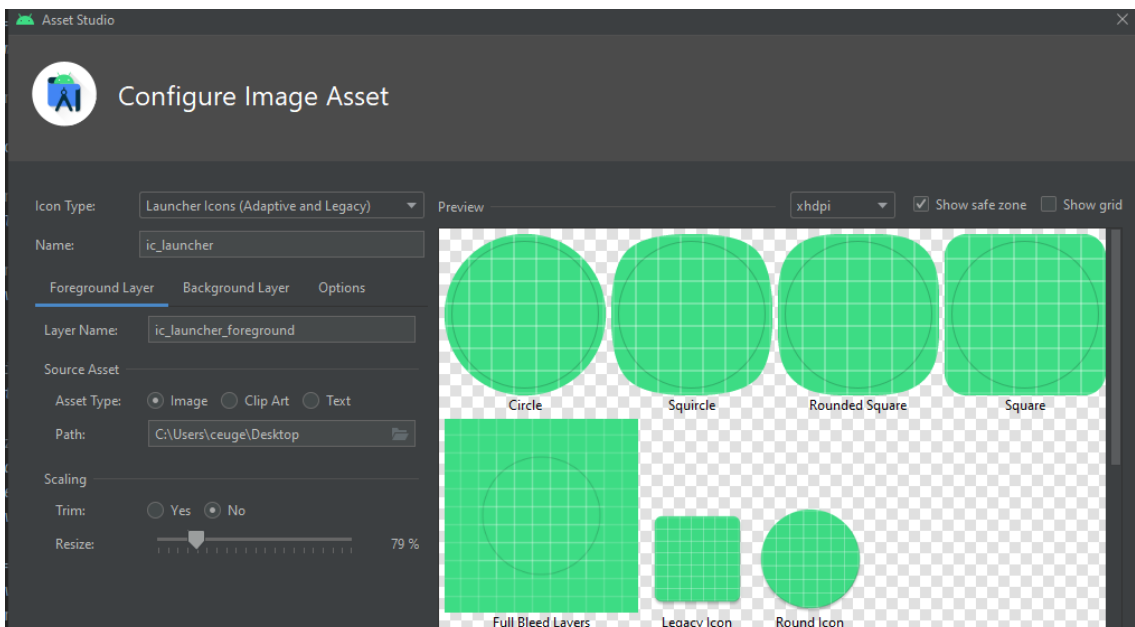
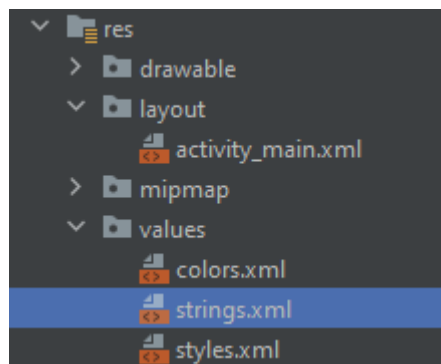


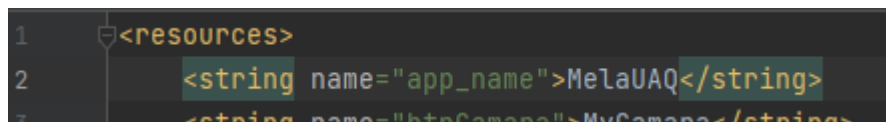
Figura 3.12: Opciones de personalización para el logo. (autoría propia).

Luego de haber abordado la personalización del logo, es momento de ocuparse del nombre de la aplicación. Para ello, se navega a la lista de proyectos ubicada en el lado izquierdo de la interfaz de Android Studio. Dentro de esta lista, se busca la carpeta "res" y, una vez dentro de ella, ubicar la carpeta llamada "values". Allí se encuentra un archivo denominado "strings.xml". Este archivo es fundamental para definir y gestionar los textos y cadenas de caracteres utilizados en la aplicación, incluyendo el nombre de la aplicación.



*Figura 3.13: Ubicación del archivo strings.xml. (autoría propia).*

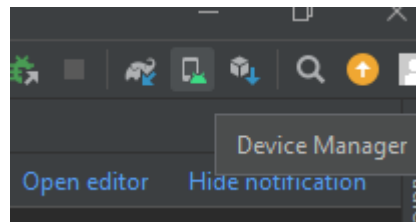
Una vez abierto el archivo "strings.xml", se puede editar el nombre de la aplicación en la primera línea donde se encuentra la etiqueta "app\_name". Es un proceso bastante intuitivo, ya que se observa que el nombre que se asignó al principio del proyecto se encuentra allí. Este paso es útil si en algún momento se decide cambiar el nombre de la aplicación durante el desarrollo del proyecto. Sin embargo, si no se realizó ningún cambio, la aplicación tendrá el nombre que se estableció inicialmente. Es importante destacar que el nombre de la aplicación puede diferir del nombre del proyecto, ya que existen dos archivos "build.gradle": uno para la aplicación y otro para el proyecto en sí, siendo este último el que lleva el nombre del proyecto.



*Figura 3.14: La línea #2 es en donde se cambia el nombre de la app. (autoría propia).*

La siguiente etapa en la creación de la aplicación implica configurar el entorno de emulación que Android Studio proporciona. Este paso resulta fundamental para asegurarse de que la aplicación se compile correctamente. Es especialmente crucial si no se dispone de un dispositivo Android para probar la aplicación. Sin embargo, es importante destacar que realizar pruebas en un dispositivo físico es esencial para verificar aspectos relacionados con el rendimiento y el tamaño de la aplicación.

Se utilizó un emulador con Google Pixel 3ª y Android 10, además de un smartphone Samsung S21 como dispositivo físico para pruebas. Para configurar el emulador, accedimos al "Device Manager" de Android Studio, que se encuentra a través del icono correspondiente en la interfaz.



*Figura 3.15: Icono de Device Manager. (autoría propia).*

Una vez que se abrió se procede a crear un nuevo dispositivo, ahí se cuenta con una gran lista con dispositivos móviles, tablets, tv, wear OS y con las versiones de Android disponibles.

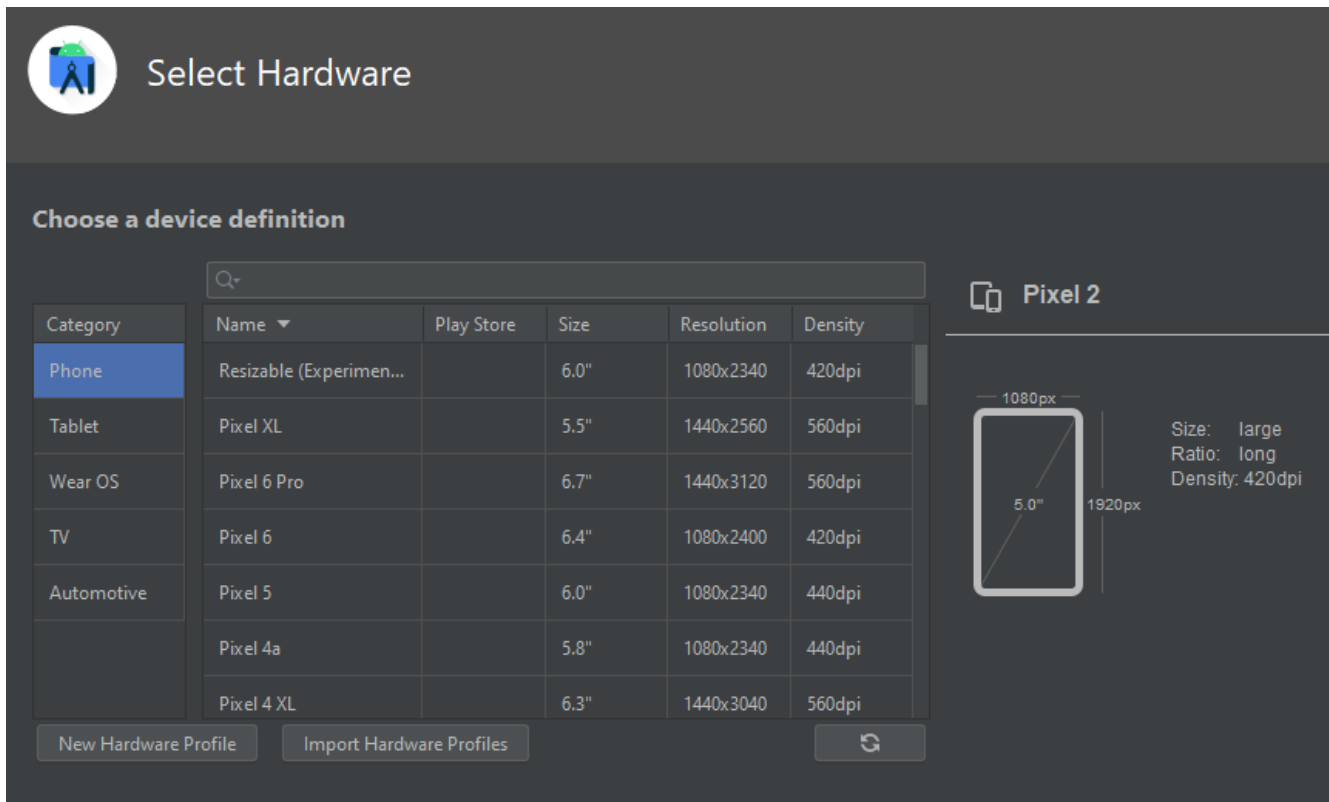


Figura 3.16: Ventana para crear un emulador de dispositivo. (autoría propia).

### 3.6 Adquisición y almacenamiento de imágenes

Para adquirir y almacenar las imágenes capturadas por la aplicación, es esencial declarar los permisos necesarios. Para llevar a cabo este proceso, se debe acceder a la ventana correspondiente en el archivo "AndroidManifest". Aquí, se agregan los permisos necesarios para acceder a la galería de imágenes y para guardar imágenes en el almacenamiento interno de los dispositivos. Cabe destacar que esta sección se enfoca únicamente en la declaración de permisos relacionados con el acceso al almacenamiento.

```
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    android:maxSdkVersion="18" />

<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

Figura 3.17: Permisos del archivo "AndroidManifest". (autoría propia).

Los permisos se deben agregar al principio del archivo "AndroidManifest.xml". Es importante prestar atención a la tabulación adecuada de las líneas de código, ya que un formato incorrecto puede generar errores en el programa. El primer permiso, `WRITE_EXTERNAL_STORAGE`, permite que la aplicación escriba en el almacenamiento externo del dispositivo. El permiso `READ_EXTERNAL_STORAGE` permite a la aplicación leer desde el almacenamiento externo del dispositivo. Por último, el permiso `INTERNET` permite que la aplicación acceda a Internet para establecer conexiones y comunicarse con recursos en línea, como servidores web. Estos permisos son esenciales para garantizar que la aplicación pueda acceder a los recursos necesarios y realizar ciertas acciones en el dispositivo. Es fundamental utilizar los permisos de manera adecuada y conforme a las políticas de privacidad y seguridad de Android. Además, es importante tener en cuenta que la sintaxis de los permisos puede cambiar con las actualizaciones, por lo que es recomendable consultar la documentación de Google para mantenerse actualizado.

En cuanto al código de `MainActivity`, este se divide en tres secciones principales. En la sección superior, encontramos todas las librerías importadas. Estas librerías se agregan desde las dependencias en el archivo `build.gradle` (`NombreProyecto`). Otra forma de agregar una librería es presionar la combinación de teclas `Alt + Enter`, lo que mostrará la opción de importar librerías si es posible hacerlo. Es importante destacar que esta opción puede variar según la versión de Android Studio o la configuración predeterminada, por lo que es necesario verificar cualquier error en la ventana de problemas si no se puede agregar una instancia o función específica.

El `MainActivity` es una clase fundamental en el desarrollo de aplicaciones Android, ya que representa la actividad principal de la aplicación. Esta actividad es la primera pantalla que se muestra al usuario al iniciar la aplicación y suele ser la interfaz de usuario principal de la misma, siendo el punto principal de interacción con el usuario.

Dentro de la clase `MainActivity`, encontramos dos secciones cruciales. En primer lugar, está el método `onCreate`, que desempeña un papel esencial en Android como parte del ciclo de vida de una actividad. Este método se utiliza para realizar la inicialización y configuración necesaria cuando la actividad se crea por primera vez. Por otro lado, tenemos el método `onActivityResult`, el cual se emplea en Android para recibir y gestionar los resultados de actividades secundarias, también conocidas como subactividades, que se inician desde la actividad principal.

Dentro del proceso de acceso a la galería para seleccionar una imagen y el posterior almacenamiento de imágenes, es fundamental entender cómo se gestionan las variables y constantes, así como cómo se asignan eventos a los botones en el MainActivity.

Primero, se declaran las variables y constantes fuera de los métodos onCreate y onActivityResult. En este paso, se especifica el tipo de variable o constante junto con su nombre y se les asigna un valor si es necesario. Estas variables serán utilizadas posteriormente en el código. Además, es importante referenciar las variables con los elementos de la interfaz gráfica, como los botones y las vistas de imagen (imageView y textView), utilizando los IDs correspondientes.

Dentro del método onCreate, se establecen las referencias entre las variables y los elementos de la interfaz gráfica, asegurando que cada elemento esté vinculado a su variable correspondiente. A continuación, se asignan eventos a los botones, lo que significa que cuando un botón se presione (evento de clic), se activará una acción específica. En este caso, se asignan eventos a los cuatro botones para abrir la galería y guardar la imagen.

Se implementa una función llamada abrirGaleria, definida como private void, ya que no devuelve ningún valor. Esta función se llama cuando se produce el evento de clic en el botón correspondiente. Dentro de la función abrirGaleria, se crea una instancia de la clase Intent, que se utiliza para iniciar actividades y servicios. En este contexto, se utiliza para seleccionar una imagen de la galería. Se especifica el tipo de archivo que se puede seleccionar como "image/\*", lo que ayuda al sistema a filtrar las opciones disponibles al usuario, mostrando solo aplicaciones y servicios relacionados con imágenes.

Finalmente, se inicia la actividad correspondiente para que el usuario pueda seleccionar una imagen desde la galería utilizando el Intent creado previamente. El método startActivityForResult se utiliza para iniciar la actividad y esperar un resultado. Se pasa el Intent y el código de solicitud GALLERY\_INTENT como parámetros. Este código de solicitud se utiliza para identificar la respuesta cuando la actividad de selección de imágenes se complete, y GALLERY\_INTENT es una constante previamente definida con el valor 3.

En esta etapa del proceso, ya se puede abrir la galería y seleccionar una imagen. Ahora toca la parte de mostrar esa imagen en el imageView y proceder a guardar una imagen en el dispositivo.

Dentro del método `onActivityResult`, se implementa un condicional para ejecutar ciertas acciones una vez que se cumplan las condiciones necesarias. En este caso, se verifica si `resultCode` es igual a `RESULT_OK`, lo que significa que la actividad de selección de imágenes se completó correctamente y que el usuario seleccionó una imagen. Además, se comprueba si `requestCode` es igual a la constante `GALLERY_INTENT`, que identifica la respuesta de esta actividad.

Una vez que se ingresa al condicional, se crea una variable del tipo `Uri`, que contendrá la ubicación de la imagen seleccionada, obtenida a través de `data.getData()`. A continuación, se crea una variable del tipo `Bitmap` para almacenar la imagen seleccionada. Para manejar posibles excepciones durante el proceso de manipulación de la imagen, se utiliza un bloque `try-catch`.

Para mostrar la imagen seleccionada en el `imageView`, se utiliza la función `setImageBitmap(imgBitmap)`. Ahora, tenemos control sobre la imagen y podemos mostrarla en la aplicación.

Para la función de guardar imágenes, se crea una función llamada `saveImageToGallery`, que recibe como parámetro el `Bitmap` con la imagen. Esta función se define fuera de los métodos `onCreate` y `onActivityResult`, pero dentro de la clase `MainActivity`.

En el interior de `saveImageToGallery`, se crea una variable de tipo `OutputStream` para escribir los datos de la imagen en el almacenamiento del dispositivo. Se utiliza un bloque `try-catch` para manejar posibles excepciones que puedan ocurrir durante la ejecución del código.

Dentro del bloque `try`, se obtiene un `ContentResolver` y se crea un objeto `ContentValues` para configurar los metadatos de la imagen que se guardará en el sistema de archivos. Se genera un nombre de archivo para la imagen con la extensión `".jpg"` y se especifica la ubicación relativa en la que se guardará la imagen, que corresponde a una carpeta llamada `"TestFolder"`. Luego, se inserta un nuevo registro de imagen utilizando los metadatos y se obtiene la URI de la imagen recién creada. A continuación, se abre un `OutputStream` utilizando la URI para escribir los datos de la imagen en el almacenamiento, lo que implica convertir el `bitmap` a formato JPEG y escribirlo en el `OutputStream`. De esta manera, logramos guardar la imagen en el dispositivo móvil.

### 3.7 Enlazar Firebase Storage/Realtime Database con la app y Colaboratory

Para la parte de la conexión entre los distintos servicios, la mejor manera de explicarlo es separando cada uno de ellos. Todo comienza con la vinculación de la aplicación a Firebase para establecer la conexión con el proyecto y obtener las bibliotecas e implementaciones necesarias en Android Studio. Hablando un poco de Firebase, esta plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google ofrece una variedad de servicios, incluyendo autenticación, almacenamiento en la nube, base de datos en tiempo real, mensajería en la nube, entre otros.

#### 3.7.1 Firebase Storage con app

Lo primero que se hace es establecer una conexión entre la plataforma Firebase y nuestra aplicación móvil. Para ello, necesitamos crear una cuenta con una dirección de correo electrónico de Google. Una vez dentro de Firebase, se accede a la consola y encontramos opciones para "agregar un proyecto" o seleccionar uno existente. Las instrucciones para la configuración se presentan una vez que creado el nuevo proyecto. Básicamente, se da un nombre al proyecto y decidir si se permite que se conecte a Google Analytics. No hay ningún problema si se decide no conectarlo a Google Analytics. Firebase nos permite conectar aplicaciones de Android, iOS, Python, sitios web y más. Una vez que se ha creado el proyecto, se verá así en nuestra consola de Firebase.

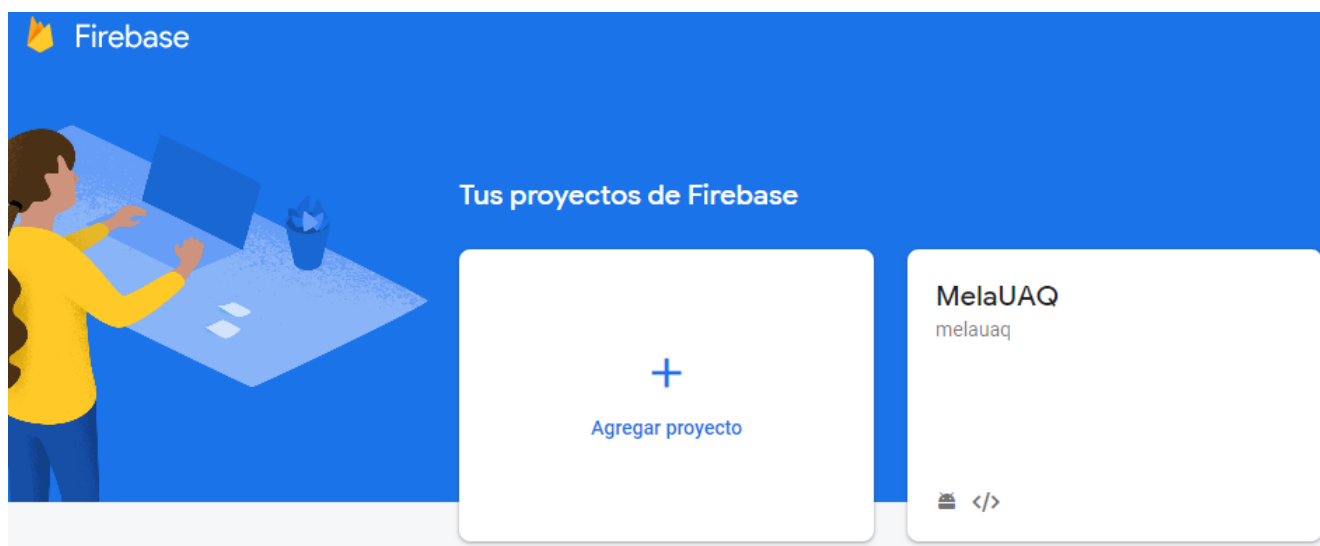


Figura 3.18: Visualización de los proyectos de Firebase. (autoría propia).



Luego, se debe ingresar al proyecto y comenzar el proceso de enlace. Hay dos formas de hacerlo. La primera es seguir los pasos que se indican en el navegador, donde se nos mostrará la siguiente captura de pantalla.



Figura 3.19: Enlazar mediante el navegador. (autoría propia).

La otra manera, que puede resultar más sencilla, especialmente porque en ocasiones el proceso a través de la página web puede fallar y requerir repetición, es hacerlo directamente desde Android Studio. Para hacerlo, debemos dirigirnos al proyecto en Android Studio, buscar "Tools" en la parte superior y, una vez allí, seleccionar "Firebase". Esto abrirá una serie de opciones relacionadas con Firebase que se pueden utilizar en una aplicación Android. El objetivo aquí es vincular nuestra cuenta de Google a Android Studio para acceder a los servicios de Google, como Firebase, y conectar nuestra aplicación al proyecto que creamos anteriormente en la consola de Firebase.

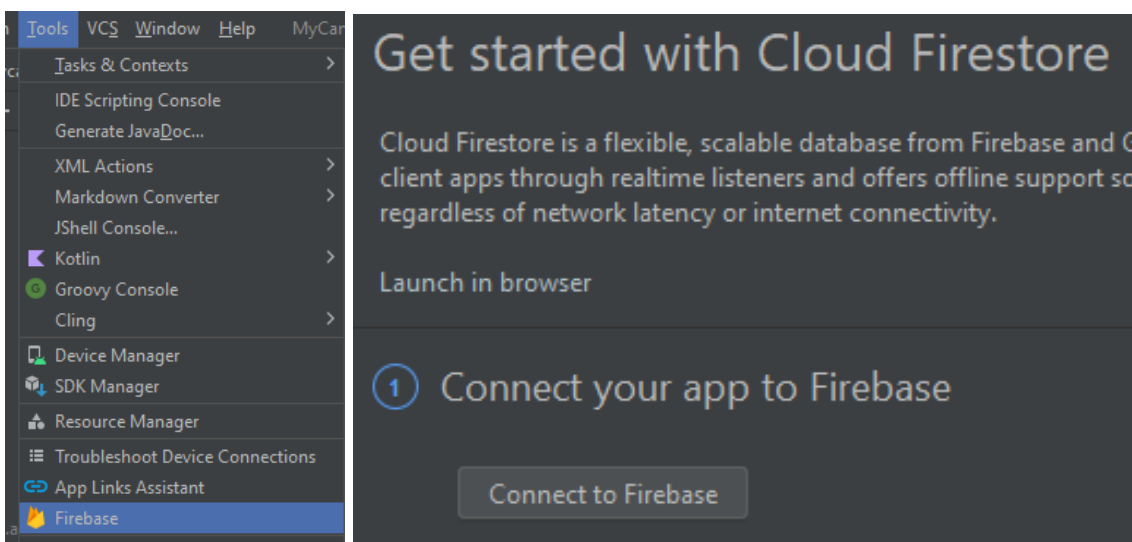


Figura 3.20: Enlazar Firebase desde Android Studio. (autoría propia).

Una vez que se está enlazado al proyecto, se puede entrar y aparecerá el símbolo de Android junto a la información de la aplicación móvil, así como el nombre y la huella digital, esto será prácticamente la verificación de que están enlazados.

The screenshot shows the Firebase console interface. On the left, there are two sections: 'Apps para Android' with a blue header and a card for 'com.example.mycamara' (Android app), and 'Apps web' with a card for 'ConnectionToPython' (Web app). The main content area is titled 'Configuración del SDK' and contains a message: '¿Necesitas volver a configurar los SDK de Firebase en tu app? Revisa las instrucciones de configuración del SDK o descarga el archivo de configuración con las claves y los identificadores de tu app.' Below this message are two buttons: 'Ver las instrucciones del SDK' and 'google-services.json'. A horizontal line separates this section from the app details below. The 'ID de la app' is '1:881252720307:android:11852dfa657b6921a31ac6'. The 'Sobrenombre de la app' is 'Agregar un sobrenombre' with an edit icon. The 'Nombre del paquete' is 'com.example.mycamara'. Below this is a table for 'Huellas digitales del certificado SHA' with two columns: 'Huellas digitales del certificado SHA' and 'Tipo'. The table contains one row with the SHA-1 hash '0e:fd:02:09:b2:9f:f1:75:21:31:05:95:4c:bc:36:70:18:ff:65:09' and the type 'SHA-1'.

Figura 3.21: Datos de la app de Android en Firebase. (autoría propia).

Lo siguiente es descargar el archivo `google-services.json`, el cual será de utilidad para enlazarse con Colaboratory. Con esto, se logra la conexión de la aplicación con Firebase. Sin embargo, el servicio que nos interesa principalmente es Firebase Storage. Este servicio nos permite subir imágenes desde la aplicación móvil a la plataforma de almacenamiento en la nube de Firebase.

Para habilitar esta funcionalidad, se configura el proyecto. En particular, se debe enfocar en el archivo `build.gradle(app)` donde se debe agregar las dependencias necesarias para importar las bibliotecas requeridas al proyecto. Primero, se incluye el plugin `apply plugin: 'com.google.gms.google-services'`. Este plugin, proporcionado por Firebase, es esencial para realizar configuraciones específicas de Firebase en una aplicación Android. Luego, debemos dirigirnos a la sección de dependencias y añadir las siguientes dependencias:

```
implementation platform('com.google.firebase:firebase-bom:32.2.3')
implementation 'com.google.firebase:firebase-analytics'

implementation('com.google.firebase:firebase-core:21.1.1')
implementation('com.google.firebase:firebase-storage:9.6.1')
implementation 'com.google.firebase:firebase-storage-ktx:20.2.1'
```

*Figura 3.22: Dependencias agregadas al hacer el enlace con Firebase. (autoría propia).*

Es importante destacar que una vez que se haya agregado las dependencias, se debe sincronizar esta sección. En la parte superior de Android Studio, se observa la opción "Sync now". Al hacer clic en ella, se descargarán las bibliotecas necesarias y se realizarán los ajustes requeridos para el funcionamiento adecuado de estas dependencias.

A continuación, se agrega la siguiente línea de código en la clase MainActivity para configurar nuestra variable de conexión con Firebase Storage: `mStorage = FirebaseStorage.getInstance().getReference();`

Luego, se procede a enviar las imágenes al Firebase Storage. Esto se logra tomando las imágenes guardadas en el dispositivo, al menos durante las pruebas, con el propósito de visualizar la imagen una vez procesada y comparar la calidad de las imágenes. En otras palabras, tendremos dos imágenes en el Firebase Storage: la imagen original y la imagen procesada.

Para lograrlo, se incorporar código en la función `saveImageToGallery` donde se asignó la imagen seleccionada al objeto `Bitmap`. Ahora, antes de subir imágenes por primera vez al Firebase Storage, es importante abrir Firebase Storage y aceptar las condiciones de uso de Google. Sin embargo, un paso fundamental es configurar las reglas de acceso al Firebase Storage para permitir tanto la escritura como la lectura de datos, ya que, por defecto, estas reglas estarán bloqueadas. Esto se logra configurando las reglas de Firebase Storage de la siguiente manera:

```

1  rules_version = '2';
2
3  // Craft rules based on data in your Firestore database
4  // allow write: if firestore.get(
5  //   /databases/(default)/documents/users/$(request.auth.uid).data.isAdmin;
6  service firebase.storage {
7    match /b/{bucket}/o {
8      match /{allPaths=**} {
9        //allow read, write: if false;
10       allow read, write;
11     }
12   }
13 }

```

Figura 3.23: Código que se encuentra en la pestaña de reglas dentro de Storage (autoría propia).

Lo siguiente es crear una referencia de almacenamiento StorageReference llamada filePathF. Esta referencia se utiliza para definir la ubicación en Firebase Storage donde se guardará la imagen. En este caso, se le da el nombre "imagenConFiltro.jpg". Luego, llamamos al método putFile() en filePathF. Este método inicia el proceso de carga de la imagen que obtuvimos previamente a través del objeto Uri creado.

Después de que se haya completado la carga de la imagen en Firebase Storage, se ejecutará el método addOnSuccessListener(), que se encargará de mostrar un mensaje de texto para indicar que el código se ejecutó correctamente y que la imagen se ha subido exitosamente a Firebase Storage.

### 3.7.2 Realtime Database con app

Ahora para la conexión de Realtime Database con la aplicación, es necesario tener ya enlazado Firebase con la aplicación móvil, después es dirigirse a las dependencias en build.gradle(app) y agregar la siguiente línea:

```
implementation('com.google.firebase:firebase-database:9.6.1')
```

Figura 3.24: Sincronizar una vez agregada en dependencias. (autoría propia).

Lo siguiente es dirigirse al MainActivity y obtener una referencia a la base de datos, las siguientes líneas de código se escriben fuera de onCreate y onActivityResult:

```
DatabaseReference mDatabaseReference = FirebaseDatabase.getInstance().getReference();
DatabaseReference mRootChild = mDatabaseReference.child("valor");
```

Figura 3.25: Con estas líneas se tendrá acceso a la base de datos. (autoría propia).

La siguiente parte de código se encuentra dentro de una función llamada seeClasification, que está dentro del método onCreate. Esta función se llama cuando se hace clic en el botón "obtener resultados", y la asignación del evento de clic al botón se realiza también en el método onCreate. Al hacer clic en el botón, se activa automáticamente el proceso contenido en la función seeClasification.

Dentro de la función seeClasification, se agrega un escuchador de eventos a una ubicación específica en Firebase Realtime Database, que en tu código se asignó como mRootChild y se le dio el nombre "valor". Esto significa que estamos esperando cambios en la base de datos para que se reflejen en nuestra aplicación. La idea es que cuando ocurra un cambio en esa ubicación de la base de datos, se muestre ese cambio en un textView. Además, se espera recibir el peso obtenido en nuestro modelo para realizar una clasificación y mostrar al usuario el resultado, ya sea "benigno" o "maligno".

Es importante mencionar que es necesario ajustar las reglas de Firebase Realtime Database para permitir que todos los usuarios puedan escribir y leer datos. En la Figura 3.26, se muestra cómo debería quedar el código de reglas para que funcione la conexión adecuadamente.

```
1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
```

Figura 3.26: código de Realtime Database. (autoría propia).

### 3.7.3 Colaboratory con Firebase

Para conectar Firebase con Colaboratory, especialmente los servicios de Storage y Realtime Database, es fundamental tener el archivo `Google-service.json` que actúa como una clave para establecer la conexión entre las plataformas. Este archivo es un formato de intercambio de datos entre el cliente (tu aplicación) y los servidores de Firebase.

Como se mencionó en la Figura 3.19, es posible conectarse a Python, el lenguaje de programación utilizado en Colaboratory. Para lograr esta conexión, se debe permitir los accesos que Google solicita y, al final, se generará un código que debes copiar y pegar en Colaboratory. Es importante tener en cuenta que el código no será idéntico, ya que es necesario realizar algunos ajustes para que funcione correctamente en el entorno de Colaboratory. Un problema común es intentar ejecutar el código completo cada vez que se realiza una prueba, lo que puede generar errores, especialmente si Firebase ya se ha inicializado previamente. Para evitar esto, es importante usar bloques `try` para verificar si Firebase ya ha sido inicializado y, si no es así, realizar la inicialización con las credenciales y la configuración adecuadas.

En lo que respecta a descargar imágenes de Storage a Colaboratory, la estrategia es establecer una conexión entre estos servicios. Esto implica acceder a la ruta de Storage utilizando el archivo JSON de configuración y definir la ruta de acceso a los archivos almacenados en Storage. Es esencial llevar un control sobre el nombre de la imagen y el directorio en el que se encuentra para evitar problemas. Por lo general, es más sencillo trabajar en la raíz de Storage.

Para la conexión con Realtime Database, el proceso es similar. Una vez que se ha inicializado el entorno de Firebase previamente, solo es necesario asignar la ruta específica y sobrescribir el valor en la base de datos en tiempo real una vez que el modelo haya analizado la imagen de entrada. En resumen, se trata de una conexión fluida entre tu aplicación, Colaboratory y los servicios de Firebase.

La última parte de la conexión con Colaboratory implica ejecutar el modelo desde la aplicación móvil. Esto se logra al acceder a un enlace que te redirigirá a una página web o URL específica. Una de las grandes ventajas de Colaboratory es que se puede ejecutar desde dispositivos móviles, lo que facilita la integración con la aplicación.

Para hacer esto, se necesita tener permisos para acceder a Internet en la aplicación y asegurarte de que las bibliotecas necesarias para el manejo de páginas web estén configuradas

correctamente. Luego, simplemente se habilita la ejecución en un WebView. El WebView es un componente de Android que permite mostrar contenido web dentro de la aplicación. Para lograr esto, se configura el botón "Procesar imagen" para que, al hacer clic en él, te redirija a la página web donde está cargado Colaboratory.

Además, es importante configurar un WebViewClient, que permite que los enlaces dentro del WebView se abran dentro del propio WebView en lugar de abrirse en el navegador web externo del dispositivo. Esto asegura que la experiencia del usuario se mantenga dentro de la aplicación móvil mientras se ejecuta el modelo en Colaboratory a través del enlace web.

### **3.8 Procesamiento de imágenes**

Para el procesamiento de imágenes en la aplicación, se utilizaron técnicas como el histograma de ecualización y el ajuste de contraste. Estos métodos son importantes para mejorar la calidad y la claridad de las imágenes, y se eligieron debido a sus buenos resultados y la diferencia perceptible que lograron en las imágenes procesadas.

Para comenzar a trabajar con estas técnicas en Android Studio, se integró la librería OpenCV. OpenCV es una librería de código abierto ampliamente utilizada para el procesamiento de imágenes y visión por computadora. Al utilizar OpenCV, se logra una ventaja significativa en términos de eficiencia y velocidad en comparación con implementaciones personalizadas que utilizarían ciclos for para procesar imágenes.

El proceso para integrar OpenCV en Android Studio comenzó descargando OpenCV for Android. Luego, en Android Studio, se creó una carpeta JNI dentro de la carpeta "app" del proyecto. Esta carpeta JNI es donde se alojarán los archivos relacionados con la integración de OpenCV en el proyecto.

La integración de OpenCV permite aprovechar sus funciones y métodos para el procesamiento de imágenes de manera más eficiente y efectiva, lo que mejora la calidad de las imágenes y la experiencia del usuario en la aplicación.

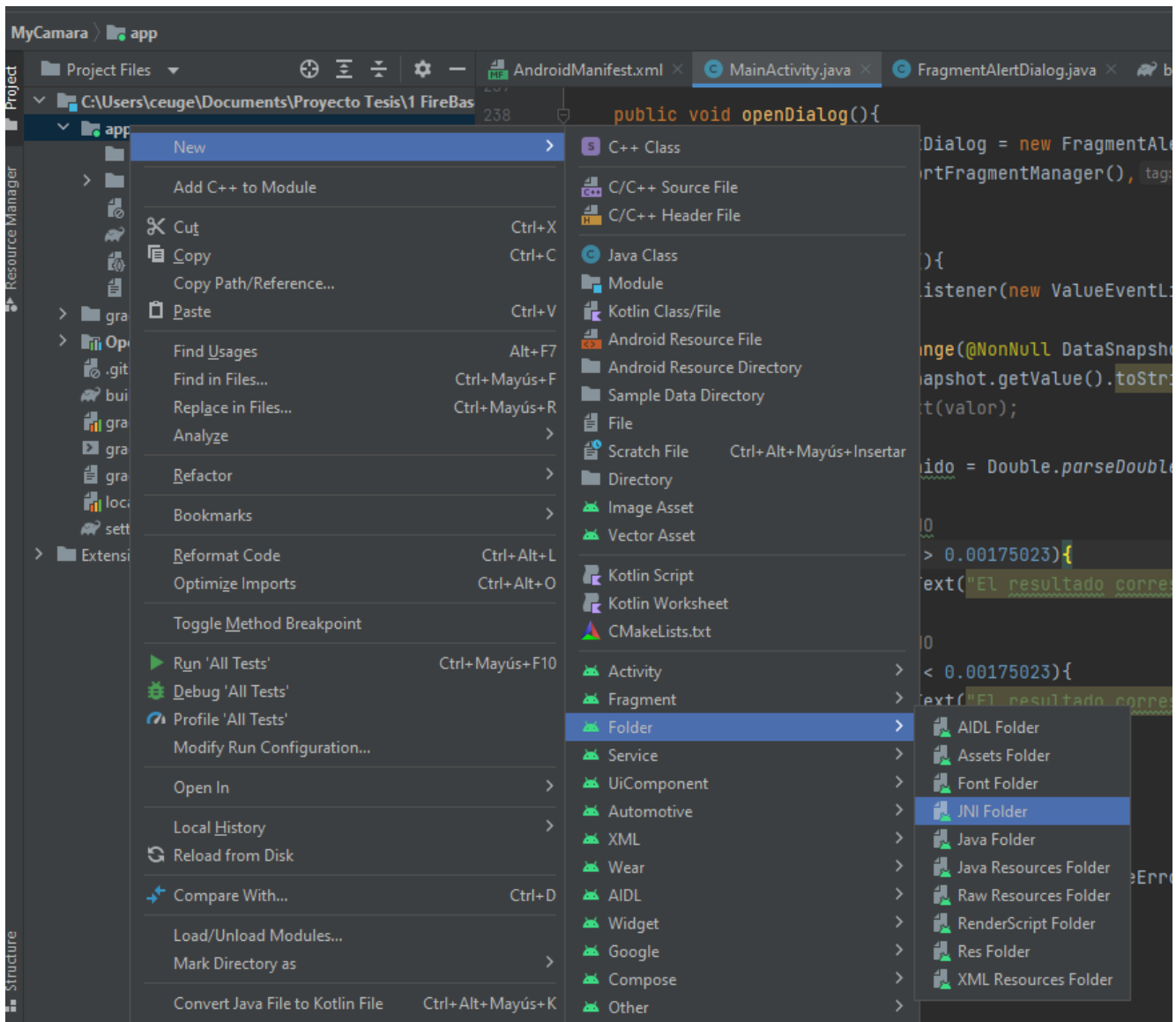


Figura 3.27: Secuencia a seguir para crear la carpeta en app. (autoría propia).

Dentro de la carpeta JNI creada anteriormente, se creará una carpeta adicional llamada "libs". Luego, se copiarán las bibliotecas nativas de OpenCV para Android, que se descargaron previamente, en esta carpeta "libs".

Para verificar que OpenCV se ha cargado correctamente en el MainActivity, dentro del método onCreate se puede agregar un código que realice esta verificación. Esto asegura que la integración de OpenCV se haya realizado con éxito y que las bibliotecas nativas se carguen adecuadamente en la aplicación. Esta verificación es importante para asegurarse de que se pueda



utilizar OpenCV de manera efectiva en la aplicación para el procesamiento de imágenes.

```
if(OpenCVLoader.initDebug()) Log.d( tag: "LOADED", msg: "success");  
else Log.d( tag: "LOADED", msg: "err");
```

*Figura 3.28: Línea de código para verificar que OpenCV se pudo cargar exitosamente. (autoría propia).*

Una vez que se cuenta con OpenCV en Android, se procede a implementar las funciones para probar los filtros de histograma de ecualización y ajuste de contraste.

Para el histograma de ecualización, se crea una función llamada "histogramaDeEcualizacion" que toma como entrada un objeto Bitmap, que corresponde a la imagen seleccionada por el usuario. Dentro de la función, se crea un objeto Mat, que se utiliza para almacenar la representación de la imagen en formato de matriz, que es el formato que utiliza OpenCV. Luego, se convierte el objeto Bitmap en la matriz Mat para procesarlo.

A continuación, se realiza el proceso de ecualización del histograma. Esto implica convertir la imagen RGB a un formato de luminancia-crominancia, realizar la ecualización del histograma en el canal de luminancia y luego fusionar los canales para obtener la imagen ecualizada. La imagen se convierte nuevamente a RGB antes de convertirla a escala de grises y aplicar la ecualización final. Finalmente, se crea un nuevo objeto Bitmap con la imagen ecualizada.

Para el ajuste de contraste, se implementan dos funciones. La función principal se llama "enhanceContrast" y toma como entrada la imagen en formato Bitmap y un factor de contraste. En esta función, se crea un nuevo objeto Bitmap con las mismas dimensiones que la imagen original. Luego, se recorren los píxeles de la imagen original y se aplica el filtro de contraste a cada componente de color utilizando las funciones proporcionadas por OpenCV. El resultado se guarda en el nuevo objeto Bitmap.

La segunda función, llamada "applyContrast", toma un componente de color y un factor de contraste como entrada y devuelve el nuevo valor del componente de color después de aplicar el filtro de contraste. El factor de contraste puede ser menor que 1.0 para reducir el contraste o mayor que 1.0 para aumentarlo.

Estas funciones permiten aplicar los filtros de histograma de ecualización y ajuste de contraste a las imágenes seleccionadas por el usuario en la aplicación móvil, lo que proporciona resultados visuales diferentes y mejorados en función de los parámetros de filtro y ajuste seleccionados.

### **3.9 Clasificación**

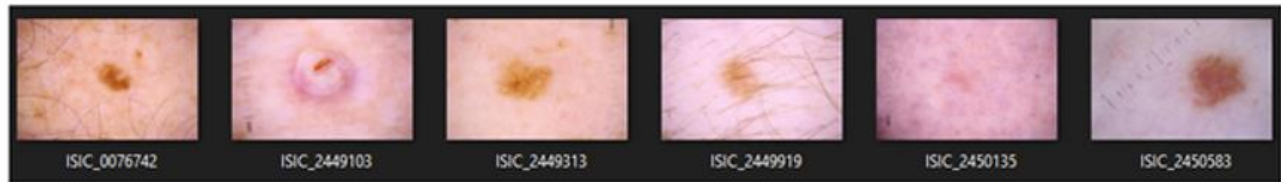
Llegados a este punto, la aplicación está completamente funcional, lo que permite seleccionar una imagen de la galería, aplicar filtros, subir la imagen a Firebase Storage, conectarse con Colaboratory y ejecutar el modelo de clasificación desde la aplicación móvil. El siguiente paso sería realizar pruebas con los filtros y evaluar cómo afectan o mejoran la clasificación entre casos malignos y benignos.

Para agregar un botón de ayuda en la aplicación, se creó una nueva clase llamada "FragmentAlertdialog." En esta clase se define el mensaje que se mostrará cuando se haga clic en el botón de ayuda. La función "openDialog" se utiliza para crear una instancia de un objeto "FragmentAlertDialog" y definir cómo se verá y se comportará la ventana emergente. Luego, se utiliza el método "show" para mostrar el mensaje en la pantalla.

En cuanto a las pruebas con los filtros, se utilizaron las imágenes mencionadas en la sección "Entrenar el modelo en Colaboratory." Estas imágenes se sometieron a los dos filtros implementados: histograma de ecualización y ajuste de contraste. Luego, se cargaron las imágenes procesadas en el modelo para observar cómo afectaban los pesos y si había una mejora en la clasificación.

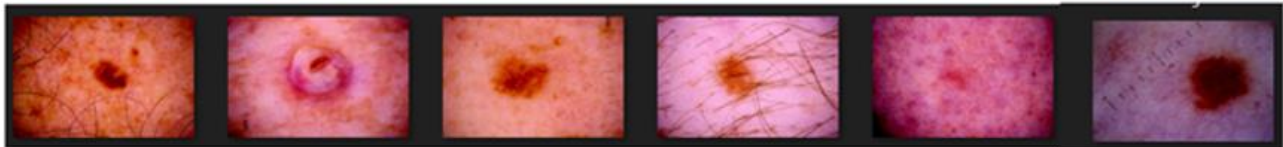
En la Figura 3.29 se muestran ejemplos de las imágenes en estado base, aplicando el filtro de histograma de ecualización y las imágenes con el ajuste de contraste. Estas imágenes se utilizaron para evaluar cómo los filtros afectaban la calidad de las imágenes y si influían en la precisión de la clasificación de casos benignos y malignos.

En resumen, la aplicación está lista para realizar pruebas con los filtros implementados y evaluar su impacto en la clasificación de lesiones cutáneas. Además, se ha proporcionado un botón de ayuda para informar a los usuarios sobre el funcionamiento de la aplicación y la importancia de consultar a un especialista en caso de resultados de clasificación.



Pesos con el modelo:

0.00076    0.00346    0.00146    8.68E-05    0.00083    0.01214



Pesos con ajuste de contraste:

0.009805    0.01857805    0.00340958    0.0001048    0.00342052    0.0087840



Pesos con histograma de ecualización

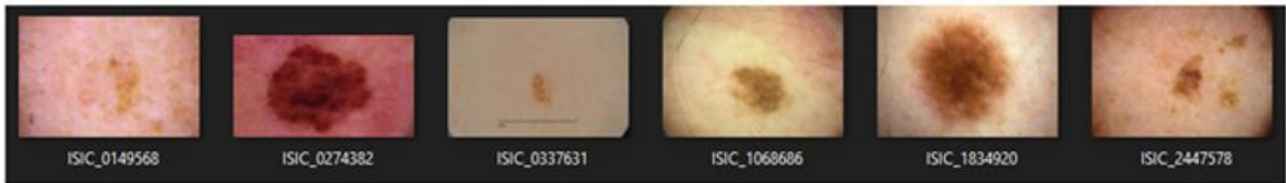
0.01370230    0.0088544    0.0046788    0.00016343    0.01021354    0.01011752

*Figura 3.29: Son los casos benignos, de arriba abajo tenemos: imágenes normales, con ajuste de contraste y por último histograma de ecualización. (autoría propia).*

Es cierto, al observar los pesos marcados en las imágenes después de aplicar los filtros de histograma de ecualización y ajuste de contraste, parece que siguen una tendencia que puede ser beneficiosa para la clasificación. Los valores más pequeños tienden a ser aún más pequeños después de aplicar los filtros, lo que puede indicar una mejora en la distinción entre las características relevantes de las imágenes.

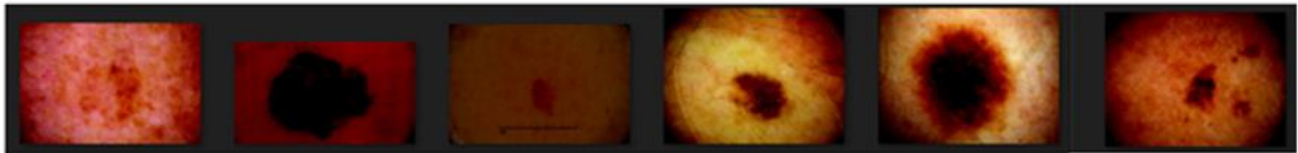
Además, se puede apreciar un contraste más marcado entre las diferentes imágenes después de aplicar los filtros. Esto podría ser útil para resaltar características importantes en las imágenes que podrían haber pasado desapercibidas en las versiones originales.

Estos resultados son prometedores y sugieren que los filtros implementados pueden ser útiles para mejorar la calidad de las imágenes y, potencialmente, la precisión de la clasificación de lesiones cutáneas. Sería interesante realizar pruebas más detalladas y comparativas para evaluar el impacto de estos filtros en la precisión de la clasificación en un conjunto más amplio de datos.



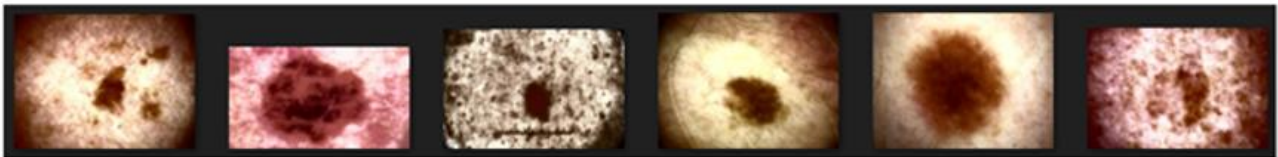
Pesos con el modelo:

0.085892 0.731433 0.086202 0.564551 0.001750 0.565801



Pesos con ajuste de contraste:

0.50496876 0.97960552 0.06273814 0.57948601 0.04619736 0.38582069



Pesos con histograma de ecualización:

0.31117141 0.31534516 0.00543341 0.378365248 0.0117037 0.026211927

*Figura 3.30: Son los casos malignos, de arriba abajo tenemos: imágenes normales, con ajuste de contraste y por último histograma de ecualización. (autoría propia).*

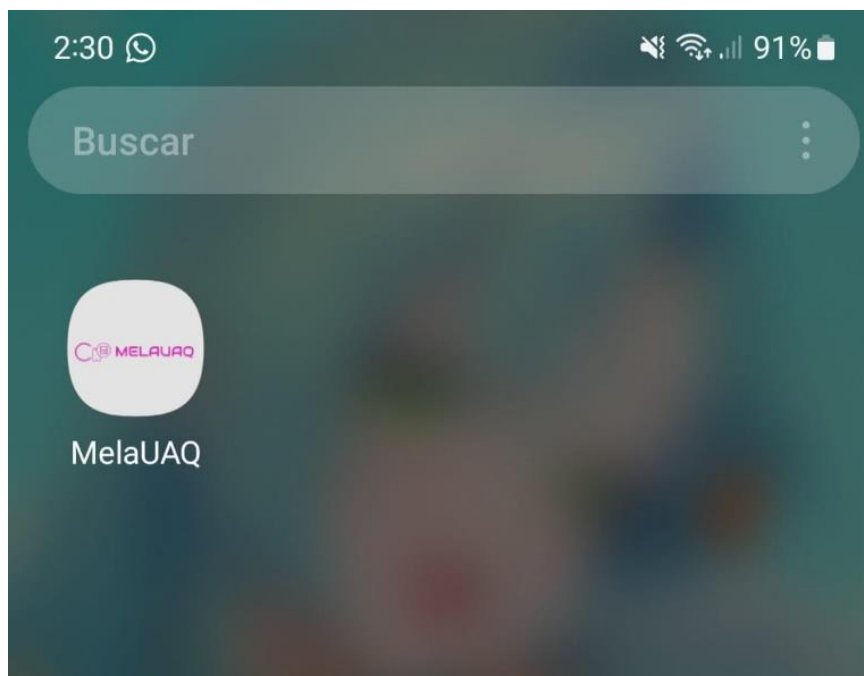
Lo mismo pasa para los casos malignos, solo que aquí se tienen valores que tienden a acercarse a el valor unitario, el contraste está presente y con los resultados obtenidos tanto para benignos como malignos, se puede encontrar una diferencia marcada para hacer una clasificación, estos resultados y clasificación se podrá observar de mejor manera en la parte de resultados.

## CAPÍTULO 4

### 4 RESULTADOS

A continuación, se presentan los resultados obtenidos al seguir la metodología propuesta. Se analizará el proceso necesario para utilizar la aplicación y se mostrarán gráficos que representan los pesos obtenidos. Además, se examinarán los resultados utilizando los dos filtros empleados para definir la clasificación a seguir.

En primer lugar, se instaló la aplicación en un dispositivo móvil. Para ello, se accedió a los ajustes del dispositivo y se habilitó la depuración por medio de USB en la sección de opciones de desarrollador. Una vez otorgado este acceso, solo se tuvo que abrir el proyecto en Android Studio y ejecutarlo en el móvil. La aplicación se identifica fácilmente por su nombre y logotipo, como se muestra en la Figura 4.1.



*Figura 4.1: MelaUAQ en el dispositivo móvil. (autoría propia).*

En la Figura 4.2, se presenta la interfaz de la aplicación, la cual se caracteriza por su sencillez e intuición en su diseño. En primer plano, se encuentra un botón de ayuda ubicado en la parte superior derecha de la interfaz. Al hacer clic en este botón, se despliega información relevante sobre el funcionamiento de la aplicación, proporcionando una comprensión clara de su operación y los posibles resultados esperados.



*Figura 4.2: Interfaz de la aplicación. (autoría propia).*

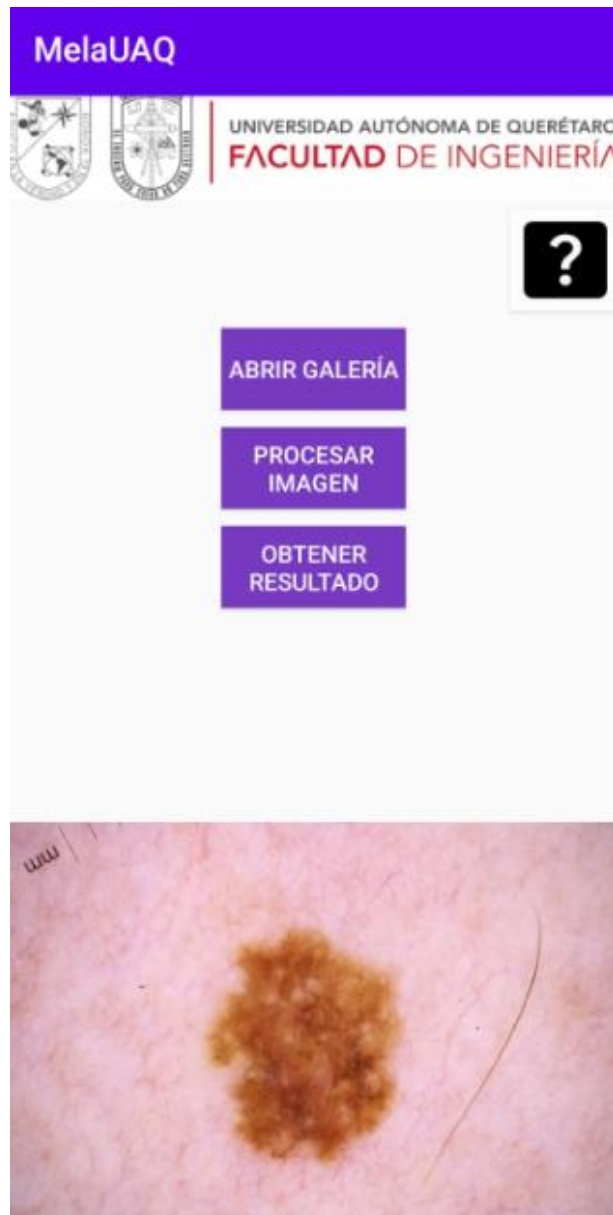
En primer lugar, el usuario debe hacer clic en el botón que indica "Abrir galería", lo que lo llevará a su galería de imágenes. Desde allí, debe seleccionar la imagen que desea analizar en busca de posibles melanomas. Es importante recordar que, tal como se mencionó en todo el proyecto, la aplicación actualmente solo es compatible con la base de datos proporcionada, por lo que la selección debe limitarse a las imágenes disponibles en esa base de datos.



*Figura 4.3: Selección de la imagen a procesar. (autoría propia).*



Para poder visualizar y confirmar la imagen seleccionada, esta se muestra en el imageView, como se puede observar en la Figura 4.4. Una vez que el usuario esté seguro de la imagen seleccionada, debe presionar el botón "Procesar imagen". Esto lo llevará a la página de Colaboratory, donde se muestra el modelo. En este punto, solo es necesario hacer clic en el botón de reproducción (play) en el código para que se inicie el proceso de análisis de la imagen.



*Figura 4.4: Imagen desplegada en el ViewImage. (autoría propia).*

Se puede observar que la página se ejecuta en el navegador predefinido del usuario. La primera vez que se ejecuta, solicitará acceso a Google Drive, ya que es donde se encuentran los modelos que se exportan a Colaboratory.

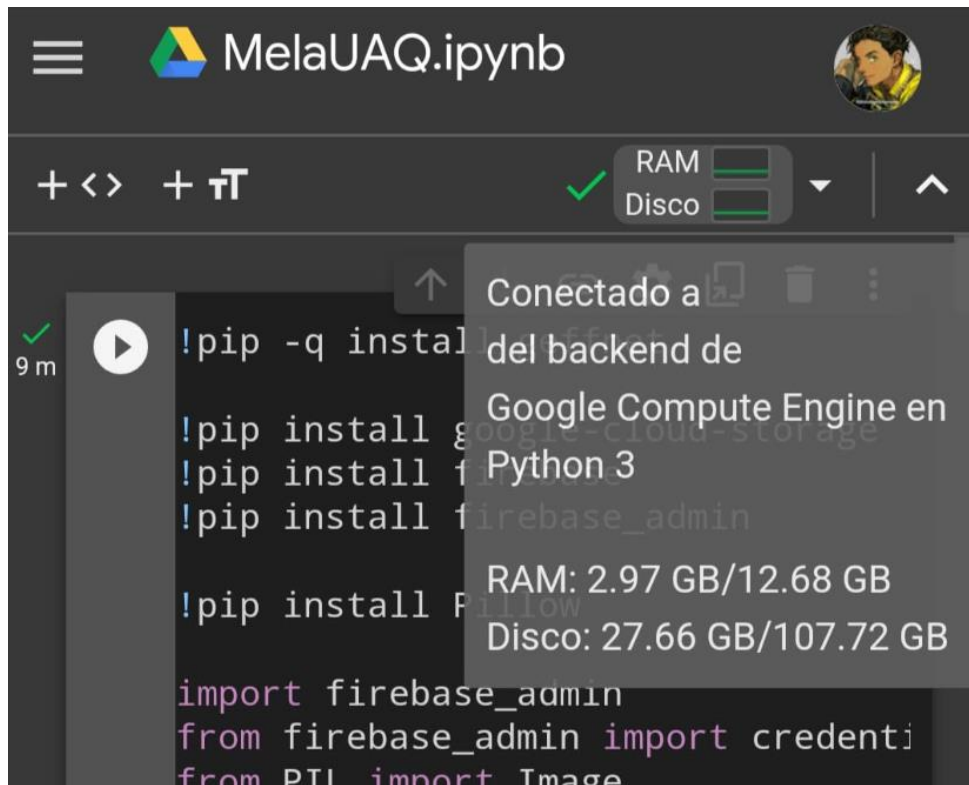


Figura 4.5: Mensaje mostrado en la consola en Colaboratory. (autoría propia).

A continuación, se espera a que se complete la ejecución, lo cual puede llevar entre 2:30 y 3:00 minutos. Esta duración se debe a que la imagen se somete al modelo para obtener un peso, y luego se envía ese valor de regreso a la aplicación móvil para realizar la clasificación y mostrar el resultado al usuario. Una vez finalizado el proceso, se mostrará un mensaje que indica que se puede volver atrás a la aplicación, justo como se ve en la Figura 4.6.

```

0.3385557532310486
LISTO!, regresa a la aplicación móvil para ver tu resultado :)

```

Figura 4.6: Ejecución de Colaboratory en el móvil. (autoría propia).

Antes de regresar a la aplicación, es recomendable verificar en Firebase Storage que la imagen se ha subido correctamente a la plataforma. Una vez confirmada la presencia de la imagen en Firebase Storage, el código utilizado en Colaboratory tiene acceso a esa imagen para someterla al proceso de clasificación.



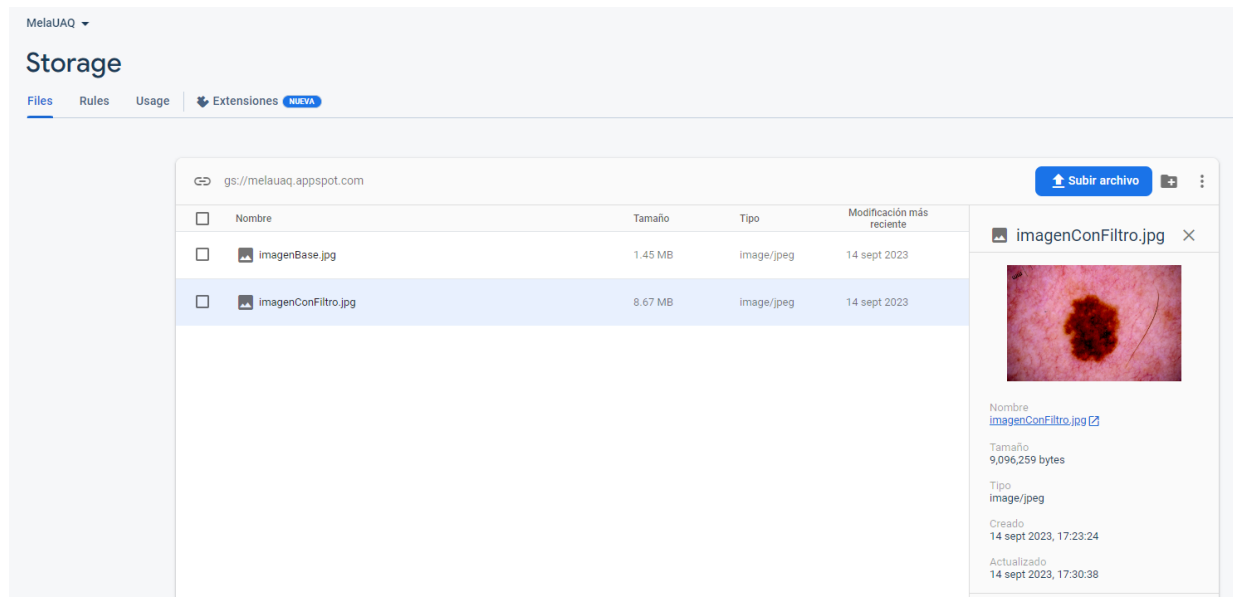


Figura 4.7: Interfaz de Storage, donde se guardan las imágenes enviadas desde la app. (autoría propia).

Una vez que se comprueba que el proceso en Colaboratory ha finalizado, el peso obtenido es enviado a Realtime Database. Una vez recibido en la base de datos, se envía a la aplicación móvil para su comparación. En la Figura 4.8 se muestra el valor obtenido impreso en Colaboratory.



Figura 4.8: Valor enviado a Realtime Database. (autoría propia).

Una vez que se regresó a la aplicación, solo es cuestión de presionar el botón de “Obtener resultado” para que muestre la clasificación de la imagen, en donde tenemos de opción maligno o benigno, la obtención de esta clasificación se logró analizando los valores obtenidos con los filtros de histograma de ecualización y ajuste de contraste.



*Figura 4.9: Resultado desplegado en la app. (autoría propia).*

Una vez que se tiene la aplicación funcional, se descargaron imágenes para realizar pruebas. En este caso, se seleccionaron diez casos de diagnósticos malignos y diez benignos, los cuales fueron sometidos a los filtros, y se recopilaron los resultados para determinar si mejoraban o empeoraban la precisión del modelo.

Lo primero que se hizo fue obtener los pesos correspondientes a las imágenes con el modelo. Los resultados mostraron una efectividad del 96%. Lo interesante de este resultado fue que el margen entre los casos benignos y malignos era muy pequeño, y la clasificación entre uno y otro a menudo dependía de centésimas, como se puede observar en la Figura 4.10.

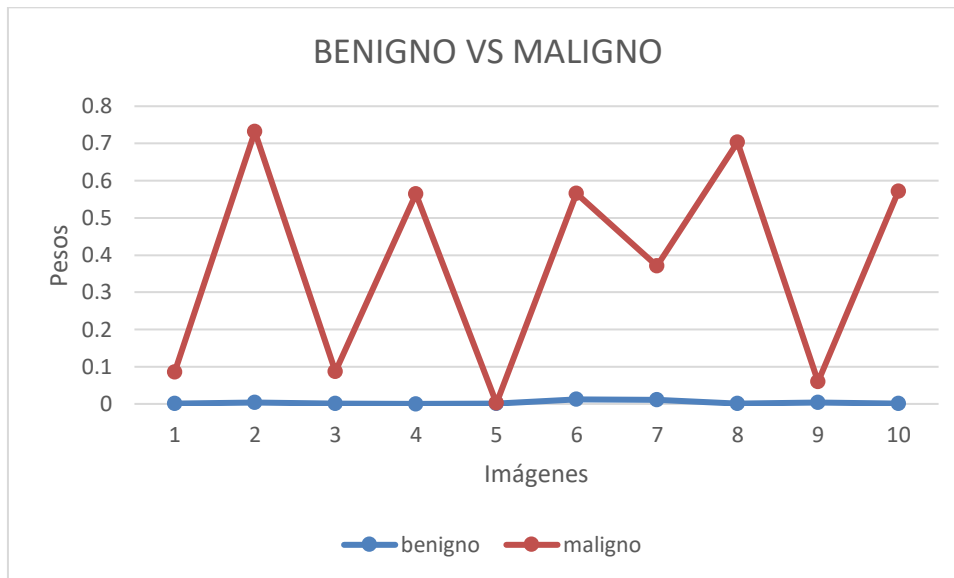


Figura 4.10: Gráfico de los pesos solo con el modelo. (autoría propia).

Lo siguiente que se probó fue poner en comparación los resultados una vez aplicados el filtro del histograma de ecualización y el ajuste de contraste, para visualizar en cual se encontraba una mayor separación.

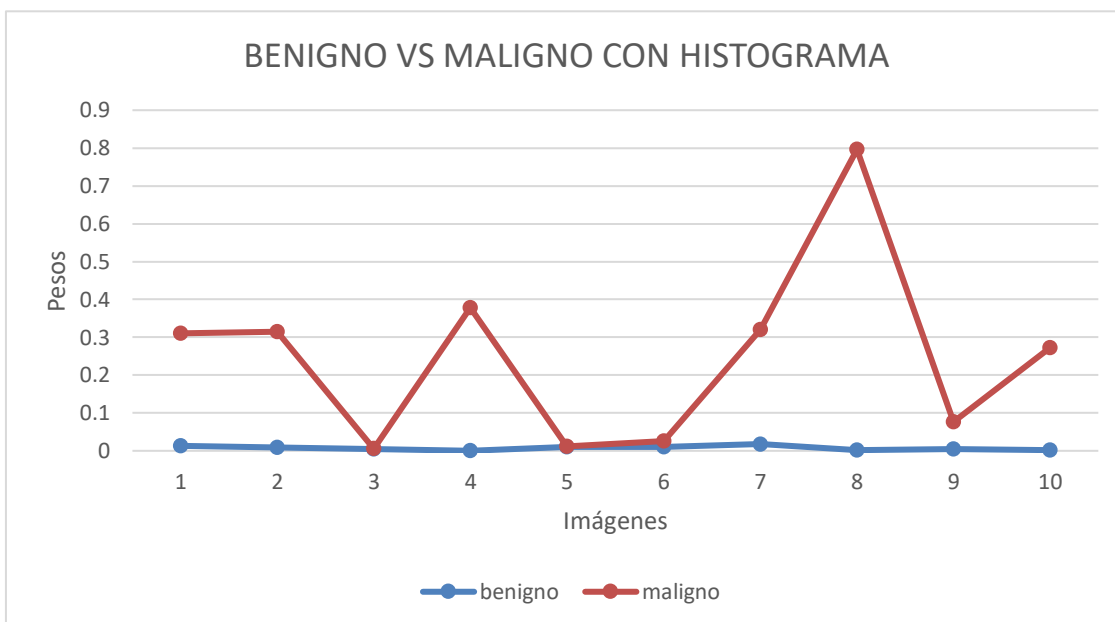


Figura 4.11: Gráfico de los pesos aplicando histograma de ecualización. (autoría propia).

En la Figura 4.11 se puede apreciar que hay valores que se cruzan entre casos benignos y malignos, lo que hace que pueda tener el mismo error que el mismo modelo por su cuenta e incluso que tenga un error aún mayor.

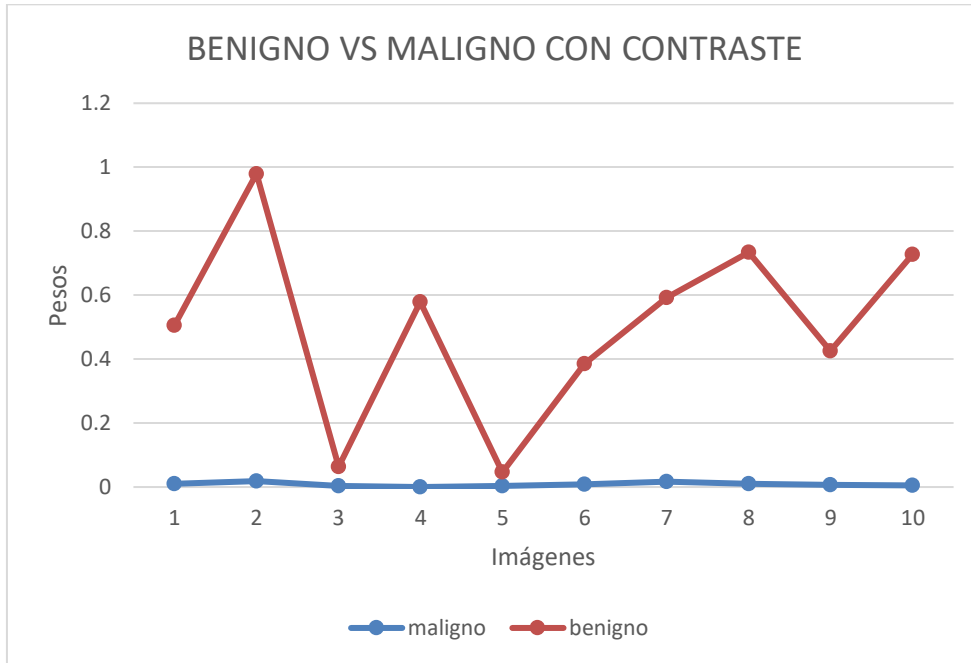


Figura 4.12: Gráfico de los pesos aplicando el filtro de ajuste de contraste. (autoría propia).

En la Figura 4.12 se puede ver claramente una mejora entre los casos malignos y benignos, logrando tener un mejor rango en los pesos.

Dentro de los casos seleccionados, al aplicar los filtros se encontró que, al aplicar el filtro de ajuste de contraste, los pesos se encontraron distribuidos de una mejor manera, logrando que esta fuera la mejor opción para poder clasificar los casos. Lo siguiente fue aplicar logaritmo base diez a los resultados para hacer más grandes los valores y poder visualizar por medio de un gráfico de dispersión la diferencia entre los casos benignos y malignos, además de encontrar como mejor separación el valor de 0.04, esto en base a que los valores obtenidos para los casos malignos son superiores a este valor, mientras que los benignos son inferiores. En la figura 4.13 se puede apreciar de mejor manera esta diferencia al aplicar el logaritmo a los valores.

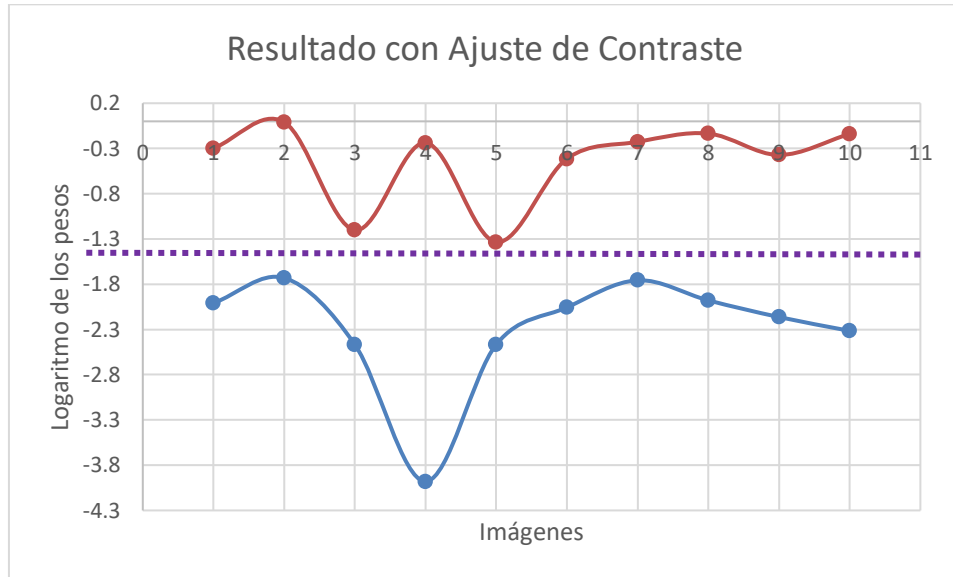


Figura 4.13: Gráfico de dispersión de valores malignos y benignos. (autoría propia).

Con estos resultados es que se logró obtener un clasificador al hacer uso del filtro de ajuste de contraste, se hicieron pruebas con otros filtros, pero los mencionados fueron los que obtuvieron mejores resultados.

## CAPÍTULO 5

### 5 CONCLUSIONES

Después de desarrollar el proyecto de tesis, el cual consistió en realizar una aplicación móvil para detección temprana de melanoma utilizando técnicas de procesamiento de imágenes, una base de datos proporcionada y un modelo empleado con Machine Learning en específico, se logró realizar una aplicación sencilla con una interfaz de usuario amigable para cualquier usuario, el cual solo necesita contar con una red de internet y con la base de datos utilizada para poder hacer uso de esta, la aplicación móvil fue desarrollada para el sistema operativo de Android con el software de Android Studio, en donde se utilizó el lenguaje de Java. Una parte fundamental una vez que se tenía la aplicación, fue el llevar el modelo a Colaboratory para poder ejecutarlo, cabe destacar que se decidió hacer uso de esta plataforma ya que se utilizan bastantes recursos y dicha plataforma nos proporciona un servicio gratuito en la nube el cual cuenta con recursos necesarios para ejecutar trabajos del ámbito de Machine Learning. Otro aspecto a mencionar fue la conexión a este servicio desde la aplicación móvil, en donde se utilizó servicios de Google como fue Firebase para poder enviar imágenes a una nube y poder mandarlas a Colaboratory y viceversa. En cuanto a un aporte al modelo utilizado, fue el emplear filtros de mejora de imagen haciendo uso de OpenCV en Android Studio, ya que todo el proceso de la imagen se realiza mediante la aplicación móvil, para después enviar las imágenes ya procesadas a Firebase. Por último, se tiene el entrenamiento del modelo antes y después de tener imágenes procesadas, ya que el objetivo era tener al menos una eficiencia igual a la del modelo mismo, e incluso mejor si se lograba obtener una mejora en el porcentaje de efectividad, esto se vio de mejor manera al hacer pruebas y al establecer el clasificador para los distintos casos.

La importancia del proyecto recae en decir que el cáncer de piel es una enfermedad que está en constante crecimiento en todo el mundo, en donde muchas veces no se cuenta con los recursos para visitar a un especialista o peor aún, el desconocer o no darle importancia a este tema, es por ello que se vio una oportunidad de emplear herramientas ingenieriles para la posible detección de melanomas haciendo uso de un dispositivo móvil. Sin duda hay áreas de mejora para la aplicación, como es el poder a futuro contar con una base de datos propia y poder hacer uso de la cámara móvil para probar con manchas en la piel que se tengan, otra mejora podría ser el escalar el proyecto, ya sea llevándolo a el sistema operativo de IOS e incluso publicarlo en las

tiendas de aplicaciones de los sistemas operativos para que cualquier persona pueda tener acceso a dicha aplicación. Otra posibilidad es que la aplicación pueda ser de ayuda a especialistas para poder tener una alternativa a los métodos convencionales.

Los resultados obtenidos fueron buenos, ya que se logró de primera mano el tener acceso al modelo, por lo que la eficiencia como mínimo se logró mantener, en cuanto a los filtros de mejora se obtuvo buenos resultados con el ajuste de contraste, en donde se hizo más grande la separación entre los pesos obtenidos para los casos malignos y benignos. Para los casos de malignos, los pesos son muy pequeños, mientras que para los benignos se obtuvieron valores más grandes, teniendo como mínimo un valor de 0.04. Sin duda hay mejores métodos a aplicar a las imágenes, en donde es posible que se requiera un mayor recurso computacional, pero precisamente es un área de oportunidad que se puede indagar y por ello se invita a seguir de cerca los avances en la investigación sobre el diagnóstico de melanoma con herramientas ingenieriles.

## REFERENCIAS

Acero D, Paul L. (2021). Aplicación móvil para detección de melanoma con redes neuronales artificiales. Proyecto para obtención de Máster en Inteligencia Artificial en Universidad Internacional de La Rioja. <https://renati.sunedu.gob.pe/handle/sunedu/3086111>

Canchachí Huamán, D. F., & Mendoza Agurto, J. T. (2020). Aplicación móvil inteligente para mejorar la eficacia de la detección de melanoma en pacientes del Centro de Fototerapia de la Piel en Trujillo, 2020. Universidad César Vallejo.

Cheonshik, K., Dongkyoo, S., Chingnung, Y., & Leng, L. (2021). Data Hiding Method for Color AMBTC Compressed Images Using Color Difference. Applied Sciences.

Chicaiza, R., & Alexis, D. (2020). Implementación de un sistema de detección temprana de melanoma utilizando redes neuronales convolucionales. Quito, 2020.

Chollet, F. (2018). Deep Learning with Python. Manning.

Developers, A. (s/f). Introducción a Android Studio. Android Developers. Recuperado el 8 de octubre de 2023, de <https://developer.android.com/studio/intro?hl=es-419>

DOF - Diario Oficial de la Federación. (2021). Gob.mx. [https://dof.gob.mx/nota\\_detalle.php?codigo=5638793&fecha=20/12/2021](https://dof.gob.mx/nota_detalle.php?codigo=5638793&fecha=20/12/2021)

Ferlay J, Ervik M, Lam F, Colombet M, Mery L, Piñeros M, et al. (2021). Global Cancer Observatory: Cancer Today. Lyon: International Agency for Research on Cancer

Globocan, S. (2020). Melanoma of skin. Iarc.fr. <https://gco.iarc.fr/today/data/factsheets/cancers/16-Melanoma-of-skin-fact-sheet.pdf>

Gómez-Álvarez Domínguez, M. (2021). Segmentación de imágenes dermoscópicas para detección de melanomas. ETSIS\_Telecomunicacion.

Gonzalez C, R., & Woods E, R. (2018). Digital Image Processing. Pearson. <https://dl.icdst.org/pdfs/files4/01c56e081202b62bd7d3b4f8545775fb.pdf>

Gonzalez, L. (2018, noviembre 9). Sesgo y Varianza en Machine Learning. Aprende IA. <https://aprendeia.com/bias-y-varianza-en-machine-learning/>

Goodfellow, I., Bengio, Y., & Courville, A. (2015). Deep Learning.



Ha, Q., Liu, B., & Liu, F. (2020). Identifying melanoma images using EfficientNet ensemble: Winning solution to the SIIM-ISIC Melanoma Classification Challenge. En arXiv [cs.CV]. <http://arxiv.org/abs/2010.05351>

Hernández-Domínguez, D. A., Aldaco-Sarvide, F., Cervantes-Sánchez, G., Erazo-Valle-Solís, A. A., Pérez-Pérez, P., Torrecillas-Torres, L., Cortés-Esteban, P., Juárez-Ramiro, A., & Alvarado, G. D. (2018). La mortalidad por melanoma en México aumenta. Medigraphic.com. <https://www.medigraphic.com/pdfs/quirurgicas/rmq-2018/rmq183b.pdf>

Kılıç, M. N. T. (2021, septiembre 21). Adaptive touch to histograms— CLAHE. Medium. <https://medium.com/@mntalha.kilic/adaptive-touch-to-histograms-clahe-4b0db004e2bd>

Kopec, V. (2022, septiembre 1). Tecnologías emergentes para los cánceres de piel de alto riesgo: herramientas para la toma de decisiones informada. The Skin Cancer Foundation. <https://www.skincancer.org/es/blog/emerging-technologies-for-high-risk-skin-cancers-tools-for-informed-decision-making/>

López Rueda, R. (2020). Aplicación Android para el seguimiento de potenciales melanomas. Universidad de Málaga.

Matas Crespi, A. (2018). Diseño de una aplicación para el tratamiento de imágenes dermatoscópicas. Universitat Politècnica de Catalunya.

MathWorks, E. (s/f). Ajuste del contraste. Mathworks.com. Recuperado el 8 de octubre de 2023, de <https://la.mathworks.com/help/images/contrast-adjustment.html>

Mayo Clinic Family Health Book (Libro de Salud Familiar de Mayo Clinic) 5.<sup>a</sup> edición

Meyer, F., Nicolás, B., Bravo, L., & Mario, N. (2020). Aplicación móvil para la detección automática de melanomas. <https://manglar.uninorte.edu.co/handle/10584/9112>

Milenio Digital. (2021, septiembre 29). Descubre qué significa la primera persona que ves en esta imagen; hay cinco escondidas. Grupo Milenio. <https://www.milenio.com/virales/pasatiempos/reto-visual-test-visual-personalidad-5-opciones-fotos>

Mitchell, T. C., Karakousis, G., & Schuchter, L. (2020). Chapter 66: Melanoma. En J. E. Niederhuber, J. O. Armitage, J. H. Doroshow, M. B. Kastan, & J. E. Tepper (Eds.), *Abeloff's*

Clinical Oncology. Elsevier.

Ogun, S. (2020). You don't really know softmax. Github.io. [https://ogunlao.github.io/2020/04/26/you\\_dont\\_really\\_know\\_softmax.html](https://ogunlao.github.io/2020/04/26/you_dont_really_know_softmax.html)

Quintanilla, M., Tamayo, D., Sunta-xi, D., & Quintanilla, J. L. (2022). Propuesta de una aplicación de aprendizaje automático con visión artificial como herramienta de apoyo para la detección de melanomas benignos y malignos. Edu.ec. <https://journal.espe.edu.ec/ojs/index.php/geeks/article/view/2723/2197>

Ribas, A., Read, P., & Slingluff, C. L. (2019). DeVita, Hellman, and Rosenberg's Cancer: Principles and Practice of Oncology (V. T. DeVita, T. S. Lawrence, & S. A. Rosenberg, Eds.). Lippincott Williams & Wilkins.

Rodríguez Medina, D., Domínguez Trejo, B., Cortés Esteban, P., Cruz Albarrán, I., Morales Hernández, L., & Leija Alva, G. (2018). Biopsychosocial assessment of pain with thermal imaging of emotional facial expression in breast cancer survivors. *Medicines (Basel, Switzerland)*, 5(2), 30. <https://doi.org/10.3390/medicines5020030>

Rotemberg, V., Kurtansky, N., Betz-Stablein, B., Caffery, L., Chousakos, E., Codella, N., Combalia, M., Dusza, S., Guitera, P., Gutman, D., Halpern, A., Helba, B., Kittler, H., Kose, K., Langer, S., Lioprys, K., Malvey, J., Musthaq, S., Nanda, J., Reiter, O., Shih, G., Stratigos, A., Tschandl, P., Weber, J. & Soyer, P. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Sci Data* 8, 34 (2021). <https://doi.org/10.1038/s41597-021-00815-z>

Saha, S. (2018, diciembre 15). A comprehensive guide to convolutional neural networks — the ELI5 way. Towards Data Science. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Scudellari, M. (2020, febrero 12). Skin monitoring apps fail to detect melanomas. IEEE Spectrum. <https://spectrum.ieee.org/skin-monitoring-apps-fail-detect-melanomas>

SIIM-ISIC. (2020). SIIM-ISIC melanoma classification. Kaggle.com. <https://www.kaggle.com/competitions/siim-isic-melanoma-classification>

Sonka, M., Hlavac, V., & Boyle, R. (2013). *Image Processing, Analysis, and Machine*

Vision (4a ed.). Wadsworth Publishing.

Szeliski, R. (2010). Computer Vision: Algorithms and Applications Springer. Springer.

Wainschenker, R., Massa, J. M., & Tristan, P. (2011). Procesamiento Digital de Imágenes. <https://users.exa.unicen.edu.ar/catedras/pdi/FILES/TE/CP1.pdf>

World Health Organization/International Agency for Research on Cancer. (2022, marzo 9). Skin cancer statistics. World Cancer Research Fund International. <https://www.wcrf.org/cancer-trends/skin-cancer-statistics/>

Zawacki, Anna, Helba, Brian, Shih, George, Weber, Jochen, Elliott, Julia, Combalia, Marc, Kurtansky, Nicholas, Codella, Noel, Culliton, Phil, Rotemberg, Veronica. (2020). SIIM-ISIC melanoma classification. Kaggle.com. <https://www.kaggle.com/c/siim-isic-melanoma-classification>