



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**ALGORITMOS DE CLASIFICACIÓN PARA DETERMINAR EL
ESTADO DE MADUREZ DEL FRUTO DE MANGO**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE
SISTEMAS**

Autor:

Bach. Olano Chávez Wilfredo Cristóbal

Asesor:

Mg. Tuesta Monteza Víctor Alexci

Línea de Investigación:

Ciencias de la Computación

Pimentel - Perú

2019



DEDICATORIA

A Dios por darme salud y la dicha de tener una familia maravillosa.

*A mis padres **Eleuterio Olano Zulueta** y **Emilia Chávez Saldaña** por su amor infinito. Gracias por siempre estar pendiente de mi en todo momento, especialmente en los difíciles, así como por apoyar todas las decisiones que he tomado. Los Amo con todo mi ser.*

*A mi hermano **Kelvin Olano Chávez** por demostrarme su grandeza cada día de mi vida por ser mi orgullo y mi bendición, se que vas a cumplir todos tus sueños y algún día serás un gran ingeniero. Te Amo hermano.*

*A mi hermana **Florita Olano Chávez** por ser mi amiga y mi madre a la vez y a mis amigos y hermanos **Ivan** y **Nancy** por sus consejos y apoyo constante gracias.*

*A **Lesly** mi novia y amiga incondicional muchas gracias.*



AGRADECIMIENTOS

*“Si he visto más lejos
es porque estoy sentado
sobre los hombros de gigantes.”*

- Isaac Newton

Agradezco de todo corazón a los ingenieros del laboratorio de ciencia de la computación de la prestigiosa Universidad Libre Seccional Calí, Colombia por todo su apoyo brindado. Así mismo, a mis amigos y maestros del laboratorio multimedia y visión por computador de la Universidad Nacional del Valle Calí, Colombia por su paciencia, su enorme intelecto y ganas de compartir sus conocimientos muchas gracias y finalmente agradecer todos mis docentes Universidad Señor de Sipán por haberme ofrecido todo su apoyo desinteresado para poder salir adelante y tomar buenas decisiones y enfocar grandes objetivos muchísimas gracias.



RESUMEN

La producción de mangos en el Perú constituye una fuente muy importante en la economía. El uso de sistemas de clasificación automática todavía no se aplica ampliamente para apoyar la producción y distribución la cual es realizada por una persona considerada experta. Así que el objetivo de este trabajo es implementar una técnica no destructiva para automatizar el procedimiento de la clasificación de los mangos según su estado de madurez utilizando el sensor de color RGB TCS3200 para la extracción de las características de color. El conjunto de datos se recolectó en fases verde, semi maduro y maduro. A continuación, se realiza el preprocesamiento de los datos el cual consiste en convertir las muestras RGB al modelo de color CIE $L^*a^*b^*$ para trabajar con datos semejantes a como el ser humano percibe el color. La extracción de características de color corresponde a la intensidad dominante del canal cromático (a^*), la saturación (C^*), el tono (h^*) y la diferencia total del color (ΔE) del canal dominante cromático (a^*) del modelo de color CIE $L^*a^*b^*$; las cuales describen la característica del color del mango. La clasificación se realiza utilizando los algoritmos supervisados Árbol de decisión y k-vecinos más cercanos (k-NN). Los resultados obtenidos se dan a partir de la técnica de validación cruzada k-fold la cual especifica que el árbol de decisión generó una exactitud promedio de (94%, 93%, 76%, 60%) y k-NN (100%, 97%, 67%, 48%) respectivamente en un tiempo de rendimiento que varía entre 0.64 y 0.74 milisegundos.

Palabras Clave— Clasificación automática, característica de color, técnica no destructiva, árbol de decisión, k-Vecino más cercano (k-NN), validación cruzada k-fold



ABSTRACT

The production of mangos in Peru is a very important source in the economy. The use of automatic grading systems is not yet widely applied to support production and distribution which is carried out by a person considered an expert. So the aim of this work is to implement a non-destructive technique to automate the procedure of sorting mangoes according to their state of maturity using the RGB TCS3200 color sensor for the extraction of color characteristics. The data set was collected in green, semi-ripe and mature phases. The data is then pre-processed and the RGB samples are converted to the CIE L*a*b* color model to work with data similar to how humans perceive color. The extraction of color characteristics corresponds to the dominant intensity of the chromatic channel (a^*), the saturation (C^*), the hue (h^*) and the total color difference (ΔE) of the dominant chroma channel (a^*) of the CIE L*a*b* color model which describe the color characteristic of the mango. Classification is performed using the Supervised Decision Tree and Nearest K-NN algorithms. The results are given from the cross validation technique k-fold which specifies that the decision tree generated an average accuracy of (94%, 93%, 76%, 60%) and k-NN (100%, 97%, 67%, 48%) respectively in a yield time varying between 0.64 and 0.74 milliseconds.

Keywords— Automatic classification, color characteristic, non-destructive technique, decision tree, k-nearest neighbor (k-NN), cross validation (k-fold)



ÍNDICE

	Pág.
DEDICATORIA	1
AGRADECIMIENTOS	2
RESUMEN	1
ABSTRACT	2
ÍNDICE	3
ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS	9
LISTA DE ABREVIATURAS O SIGLAS	10
1. INTRODUCCIÓN	11
1.1 Realidad Problemática	11
1.2 Antecedentes de Estudio	16
1.3 Teorías Relacionadas al Tema	22
1.3.1 Fruto de Mango	22
1.3.1.1 Variedades de Fruto de Mango	23
1.3.2 Sensor de Color RGB TCS3200	26
1.3.2.1 Especificaciones	26
1.3.3 Microcontrolador Arduino	28
1.3.3.1 Productos Arduino	28
1.3.4 Modelos de Color	29
1.3.4.1 Modelo de Color RGB	30
1.3.4.2 Modelos de Color CIE	31
1.3.5 Algoritmos de Clasificación Supervisada	34
1.3.5.1 Árboles de Decisión	35
1.3.5.2 k-Nearest Neighbors	36
1.3.6 Validación y Evaluación de los Resultados	38
1.4 Formulación del Problema	42
1.5 Justificación e Importancia del Estudio	42
1.6 Hipótesis	43
1.7 Objetivos	43
1.7.1 Objetivo General	43



1.7.2	Objetivos Específicos	43
2.	MATERIAL Y MÉTODO	44
2.1	Tipo y Diseño de Investigación	44
2.1.1	Tipo de Investigación	44
2.1.2	Diseño de Investigación	44
2.2	Población y Muestra	44
2.2.1	Población	44
2.2.2	Muestra	44
2.3	Variables, Operacionalización	45
2.3.1	Variables	45
2.3.1.1	Variable Independiente	45
2.3.1.2	Variable Dependiente	45
2.3.2	Operacionalización	45
2.4	Técnicas e Instrumentos de Recolección de Datos, Validez y Confiabilidad	45
2.4.1	Técnica de Recolección de Datos	45
2.4.1.1	Observación.....	45
2.4.2	Instrumento de Recolección de Datos.....	46
2.4.2.1	Ficha de Observación	46
2.5	Procedimientos de Análisis de Datos	46
2.6	Criterios Éticos	46
2.6.1	Confiabilidad	46
2.6.2	Conformabilidad	46
2.7	Criterios de Rigor Científico.....	47
3.	RESULTADOS	48
3.1	Resultados en tablas y figuras.....	48
3.2	Discusión de resultados	52
3.3	Aporte práctico.....	53
3.3.1	Implementar el protocolo para la extracción de datos	54
3.3.2	Seleccionar los algoritmos de clasificación	58
3.3.3	Caracterizar el color del fruto de mango según su estado de madurez.....	60
3.3.4	Construir la base del conocimiento basándose en el modelo de datos	61
3.3.5	Implementar la propuesta utilizando los algoritmos seleccionados en un lenguaje de programación	72
3.3.5.1	Desarrollo de la propuesta utilizando árbol de decisión y k-NN	72
4.	CONCLUSIONES Y RECOMENDACIONES.....	89
4.1	Conclusiones	89
4.2	Recomendaciones.....	90
	REFERENCIAS	92



5. ANEXOS	97
5.1 Código fuente arduino para interfaz electrónica	97
5.2 Descripción de la interfaz gráfica	99
5.3 Base de datos RGB	100
5.4 Instalaciones necesarias para la ejecución del proyecto	110
5.4.1 Instalar Python 3.7 en Ubuntu 18.04	110
5.4.2 Crear entorno virtual Python.....	111
5.4.2.1 Instalar virtualenv	111
5.4.2.2 Activar / Desactivar entorno virtual (my_tesis_env)	111
5.5 Instalar librerías Python para necesarias para el proyecto.....	112



ÍNDICE DE FIGURAS

FIGURA	Pag.
1.1 Distribución del PBI agrícola por regiones y departamentos	11
1.2 Comportamiento de las exportaciones de mango fresco	13
1.3 Planta de selección de mangos frescos para exportación.....	13
1.4 Componentes del sistema de visión por computadora.	15
1.5 Fruto de mango kent	23
1.6 Fruto de mango haden	23
1.7 Fruto de mango tommy atkins	24
1.8 Fruto de mango keitt.....	24
1.9 Fruto de mango amélie	25
1.10 Fruto de mango ataulfo.....	25
1.11 Fruto de mango manila Súper	25
1.12 Fruto de mango nam doc mai.....	26
1.13 Sensor de color RGB TCS3200.....	26
1.14 Diagrama de componentes sensor TCS3200	27
1.16 Microprocesador arduino mega.....	28
1.15 Listado por categorías de productos arduino	29
1.17 Modelo de color RGB	30
1.18 Representación cúbica de los colores primarios y secundarios.....	31
1.19 Modelo de color CIE XYZ.....	32
1.20 Modelo de color CIE L*a*b*.....	33
1.21 Método de clasificación supervisada	34



1.22	Ejemplo de árbol de decisión	35
1.23	Descripción del algoritmo k-NN	38
1.24	Proceso de creación y validación	38
1.25	Validación cruzada de K iteraciones con $K=4$	39
1.26	Ejemplo de la curva ROC	41
1.27	intervalos para los valores ROC	42
3.1	Rendimiento de la ejecución de los algoritmos	49
3.2	Error promedio resultante	49
3.3	Ejecución crom(a*).....	50
3.4	Ejecución saturación(C*)	51
3.5	Ejecución tono(h*).....	51
3.6	Ejecución diferencia Total(ΔE) de a*	52
3.7	Enfoque propuesto para la investigación	55
3.8	Diagrama de componentes	56
3.9	Esquema del circuito de conexión	56
3.10	Prototipo electrónico implementado	57
3.11	Selección de los algoritmos de clasificación	59
3.12	Análisis de la caracterización del color rgb	61
3.13	Visualización del porcentaje descriptivo de las variables utilizado datos Crom(a*) .	73
3.14	Visualización del porcentaje descriptivo de las variables utilizado datos Saturación(C*)	73
3.15	Visualización del porcentaje descriptivo de las variables utilizado datos Tono(h*) ...	74
3.16	Visualización del porcentaje descriptivo de las variables utilizado datos Diferencia Total(ΔE) de (a*)	74
3.17	Agrupamiento del porcentaje descriptivo de las variables utilizado datos Crom(a*)	75



3.18 Agrupamiento del porcentaje descriptivo de las variables utilizado datos Saturación(C^*)	75
3.19 Agrupamiento del porcentaje descriptivo de las variables utilizado datos Tono(h^*) ..	76
3.20 Agrupamiento del porcentaje descriptivo de las variables utilizado datos Diferencia Total(ΔE) de (a^*)	76
3.21 Interfaz gráfica del desarrollo de la propuesta.....	77
3.22 Segmentación uniforme de la base de datos	79
3.23 Visualización del árbol de decisión base de datos cromas (a^*)	80
3.24 Visualización del árbol de decisión base de datos saturación (C)	81
3.25 Visualización del árbol de decisión base de datos tono (h)	82
3.26 Visualización del árbol de decisión base de datos tono (h)	83
3.27 Análisis de la tasa de error para el valor K	85
3.28 Resultado de la validación cruzada k-fold cromas(a^*).....	87
3.29 Resultado de la validación cruzada k-fold tono(h^*).....	87
3.30 Resultado de la validación cruzada k-fold saturación(C^*)	88
3.31 Resultado de la validación cruzada k-fold (ΔE a^*)	88
5.1 Diseño de la interfaz gráfica para la ejecución de los modelos	100
5.2 Creación del entorno virtual my_tesis_env	111
5.3 Activación del entorno virtual my_tesis_env	112
5.4 Instalación de librerías en el entorno virtual my_tesis_env	112



ÍNDICE DE TABLAS

TABLA	Pag.
1.1 Clasificación de los modelos de color.....	30
1.2 Colores RGB primarios y secundarios.....	31
1.3 Descripción de los componentes L*a*b.....	33
1.4 Pseudocódigo del algoritmo árbol de decisión.....	37
1.5 Pseudocódigo del algoritmo k-NN.....	37
1.6 Matriz de confusión binaria.....	40
1.7 Matriz de confusión con tres clases.....	40
1.8 Métricas de evaluación del rendimiento.....	41
2.1 Indicadores y formulas.....	45
3.1 Resultados de la ejecución de los modelos.....	48
3.2 Análisis de rendimiento de los algoritmos de clasificación.....	59
3.3 Etapas de maduración del fruto de mango.....	60
3.4 Especificaciones del computador.....	63
3.5 Base de datos CIE L*a*b*.....	66
3.6 Resumen de los componentes para extracción de las bases de datos.....	67
3.7 Configuración de los parámetros de los modelos de clasificación supervisada.....	79
5.1 Base de datos con valores RGB.....	100



LISTA DE ABREVIATURAS O SIGLAS

- AUC** Area Under The Curve. 41
- BPNN** Red Neuronal de Propagación hacia Atrás. 20, 44
- CIE** Comisión Internacional de Iluminación. 32
- CIE L*a*b*** Luminosidad, A (verde-rojo), B (azul-amarillo). 6, 9, 15, 32, 33, 52, 54, 62, 64, 66, 67
- GLCM** Matriz de Consistencia de Nivel de Grises. 18, 20
- HSI** Hue, Saturation, Lightness. 15, 20
- HSV** Hue, Saturation, Value. 15, 22
- k-NN** K-Vecino más cercano. 14, 16, 36–38, 44, 48, 50, 54, 58, 72, 84, 86, 89, 90, 99, 100
- NTSC** Comisión Nacional de Sistema de Televisión. 15
- PBI** Producto Bruto Interno. 11
- PLS** Partial Leas Squares. 14
- PNN** Red Neural Probabilistic. 18, 19, 44
- RGB** Read Green Blue. 6, 9, 15, 18–22, 26, 27, 30, 31, 42, 53, 54, 58, 61, 62, 64, 66, 89, 100
- RNA** Red Neuronal Artificial. 19, 44, 58
- ROC** Receiver Operating Characteristic. 41
- SVM** Máquinas de vectores soporte. 14, 20, 44, 58
- USDA** Departamento de Agricultura de los Estados Unidos. 17, 60, 61, 89



1. INTRODUCCIÓN

1.1 Realidad Problemática

El sector agrícola en el Perú constituye una fuente muy significativa para el desarrollo de la economía, generando empleo en gran parte de la población, es por ello que se deben conocer las oportunidades y las carencias que se presenta actualmente, ya que la distribución variable de la población conlleva a no poder llegar de forma relativa a muchas partes del país y esto indirectamente colabora con un progreso desigual del sector agrario de nuestras regiones, tal desigualdad se ve reflejado en el aporte de las tres regiones del Perú al PBI el cual se evidencia en el informe del Banco Mundial (2017) donde especifica que la costa aporta el 44% del PBI poseyendo solo el 23% de las tierras agrícolas del país, mientras tanto la sierra aporta el 42% del PBI agrícola total con tan solo el 39% de tierras agrícolas y la selva aporta solo el 14% del PBI agrícola total a pesar de tener el 38% de tierras agrícolas. Como se muestra en la Figura 1.1. Estos datos evidencian un aporte bajo de la selva en comparación con la costa y la sierra.

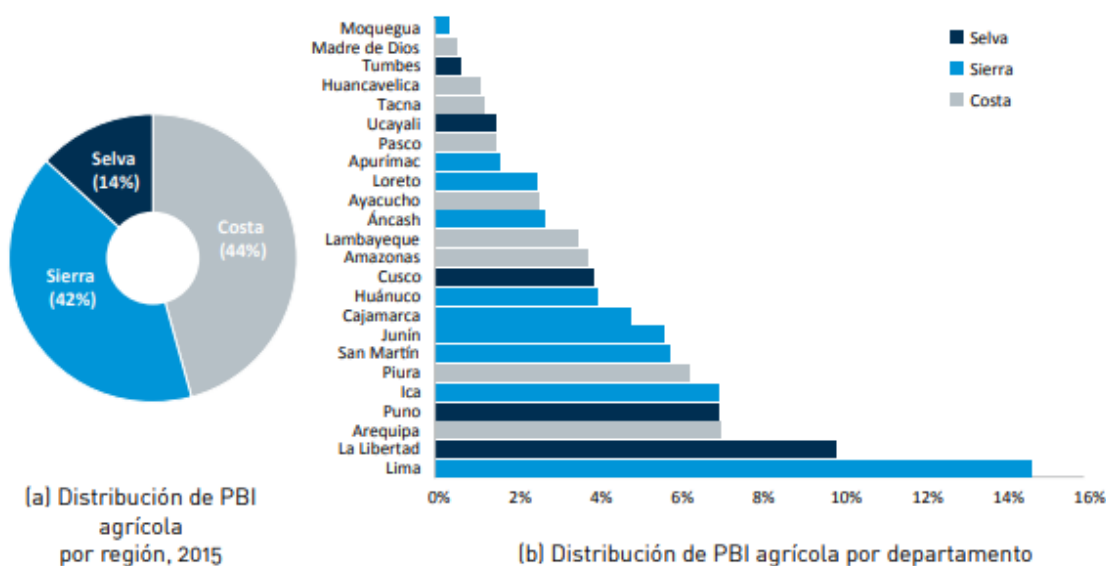


Figura 1.1: Reparto de del PBI agrario por regiones y departamentos, Perú, 2015 Banco Mundial (2017).



Lambayeque es un departamento perteneciente a la región de la costa peruana con una población estimada de 1 270 794 personas en un área de 14 231.3k² cuenta con 3 provincias y 38 distritos y su contribución al PBI está distribuido en los siguientes sectores, 20% del comercio, 10% de la manufactura, 51% de servicios, 8% del sector agropecuario y 11% del resto de actividades MINAGRI (2017).

Estas estadísticas revelan que la agricultura genera un aporte importante a la economía del país, por lo tanto, el desarrollo del mismo está ligado en gran medida a la exportación de los productos agrarios en grandes volúmenes. Como se muestra en el informe del ministerio de agricultura y riego elaborado por Pino (2017) el cual precisa que el aporte de las exportaciones a la economía del país generó que la balanza comercial agraria de enero del 2017 arroje un valor positivo de 151MM de dolares, superior en 579% con relación al mes 2016 en cual se generó un aporte de 97MM de dolares, los productos agrícolas más importantes para estas exportaciones fueron los mangos y las uvas frescas, los cuales significaron el 26% y 12% del total, respectivamente. Asimismo, precisa que al finalizar enero del 2017 las exportaciones peruanas tuvieron como destinos a 98 países, siendo EEUU, Países bajos y España los cuales reunieron el 36%, 11%, (5%) de total.

Como se evidencia el mango peruano cuenta con un alto índice de exportación por ser un producto calificado por su sabor, color y aroma, por consiguiente, el informe de dirección general de políticas agrarias elaborado por Romero (2017) indica que el mango peruano se ha incorporado a las grandes ligas del comercio mundial del mango fresco a partir de la década pasada, en la que destacan países como la India, México, Brasil, Tailandia, Ecuador.

A partir del 2016 el mango peruano ha sobrepasado en porcentaje de exportación al de Brasil, demostrando que las exportaciones del mango peruano están creciendo de forma sostenida y consistente, como se expone en la Figura 1.2.

Observando la importancia del mango peruano para la exportación se deben cumplir los procesos de producción en tiempo y forma para garantizar la calidad, es por ello que uno de los procesos más importantes es la selección del mango de acuerdo a su estado de madurez, el mismo que a



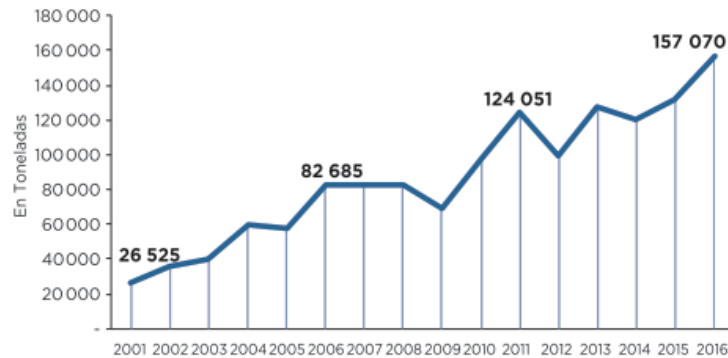


Figura 1.2: Conducta de las exportaciones de mango. Fuente Romero (2017)

la actualidad se sigue clasificando bajo la perspectiva del ojo experto del ser humano para dar su primera valoración del producto y no todos tienen la misma evaluación o punto de vista como se evidencia en la Figura 1.3. Por otro lado, la agrupación manual no es apropiada para ser usada todos los días debido a la distracción o incluso al cansancio por el exceso de trabajo y concentración, de modo que la disminución de la velocidad de trabajo y la calidad de selección también se ve afectada Lal et al. (2017); Savakar (2012).



Figura 1.3: Planta de selección de mangos frescos para exportación. Fuente Romero (2017)

La tecnología ha evolucionado de una forma notable en los distintos ámbitos, surge los sistemas



con aprendizaje automático como alternativa de apoyo al proceso de clasificación de los productos agrarios de acuerdo al estado de madurez aprovechando distintas características como puede ser la forma, el tamaño, el color y la textura. Existe investigaciones propuestas para clasificar los productos de acuerdo a las características mencionadas.

Mustaffa and Khairul (2017) Mencionan que el color y el tamaño son una de las características más importantes para la clasificación precisa de la madurez, se propuso un sistema para determinar la madurez de la fruta de mango utilizando la librería de OpenCV-Python y el ordenador de placa reducida Raspberry Pi para ejecutar el algoritmo de procesamiento de imágenes desarrollado, este sistema propuesto es capaz de determinar el tamaño de la fruta y aplicar un algoritmo de clasificación no supervisada k-Means para determinar el color de la fruta. Así mismo, Pavithra et al. (2015) proponen un algoritmo de estimación de madurez basado en el color para clasificar el tomate en la segunda fase de maduración, siendo la fase I, II buena y III mala utilizando procesamiento de imágenes, se extrajeron características de textura, color y forma. También se ha propuesto extraer solo el fondo de la superficie del tomate y dejar una nueva segmentación basada en color utilizando la distancia euclidiana. A continuación, se realizó la clasificación utilizando el algoritmo de SVM basado en k-NN. Este clasificador combina el clasificador k-NN y SVM para mejorar el rendimiento en términos de precisión y tiempo de cálculo. El rendimiento de SVM basado en k-NN es superador en comparación con el clasificador k-NN y SVM de forma individual. Por otra parte, Malegori et al. (2017) presentan un análisis del rendimiento analítico del espectrómetro NIR el cual es uno de los más pequeños del mercado de última generación, haciendo una comparación crítica con un espectrómetro FT-NIR de sobremesa en la evaluación de la exactitud de predicción, del contenido de ácido ascórbico de la fruta acerola la cual durante su etapa de maduración cambia su color y la fruta puede perder hasta la mitad de su contenido de ácido ascórbico debido a que la variabilidad de los parámetros químicos siguió un perfil estrictamente no lineal, se compararon dos algoritmos de regresión diferentes PLS y SVM para la clasificación. Los modelos de regresión obtenidos con Micro-NIR dan mejores resultados usando el algoritmo SVM, tanto para el cálculo



del ácido ascórbico como de la acidez titulable. Hay que mencionar, además, Kumar and Gill (2015) realizaron un análisis de clasificación de los productos agrícolas haciendo uso del procesamiento de imágenes aplicando sistemas basados en visión por computadora, se utilizó diversos espacios de color tales como NTSC, YCbCr, CIE L*a*b*, HSV y HSI, siendo HSI el mejor espacio de color para poder trabajar la clasificación automatizada de productos utilizando la técnica de procesamiento de imágenes . Se debe agregar que los métodos propuestos utilizan el siguiente esquema (ver Figura 1.4), para la prueba y extracción de muestras.

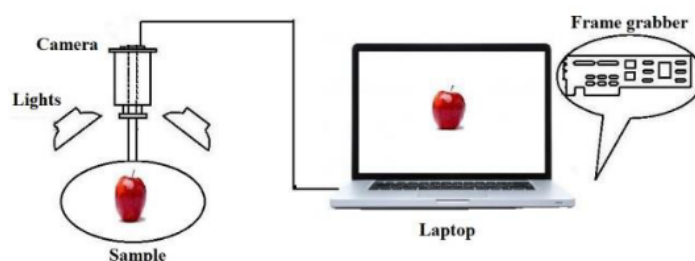


Figura 1.4: Componentes del sistema de visión por computadora. Fuente: Kumar and Gill (2015)

Como se evidencia en estos trabajos la clasificación de las frutas se realiza principalmente por medio del procesamiento de imágenes RGB para la obtención de muestras y posteriormente utilizar algún algoritmo de clasificación supervisada o no supervisada. En este proyecto se plantea una solución optimizada para el proceso de selección del mango de acuerdo al estado de madurez utilizando características descriptivas de color, inicialmente se utilizará una interfaz electrónica la cual utiliza un sensor de color RGB TCS3200 administrado por un microcontrolador arduino Mega, dicha interfaz generará las muestras en el modelo de color RGB las cuales serán convertidas al modelo de color CIE L*a*b* para poder caracterizarlas de acuerdo a índices de grado de luminancia (*luma*) y dos componentes cromáticos (*croma*), en similitud con los estados de maduración del fruto de mango, tales como verde, semi maduro o maduro, finalmente se creará la base de datos del conocimiento que servirá como modelo de datos para los algoritmos de aprendizaje automático,



originalmente se cree que los algoritmos k-NN y Árboles de decisión serán los mejores algoritmos de clasificación por sus altos índices de rendimiento en trabajos previos (ver Figura 3.11), teniendo ya los algoritmos de clasificación implementados en la propuesta es posible separar el 80% de las muestras para el entrenamiento y el 20% para pruebas y poder medir el rendimiento en términos de tiempo de respuesta, exactitud y precisión.

1.2 Antecedentes de Estudio

Hussain Hassan and Nashat (2018) en su investigación titulada como **Nuevas técnicas efectivas para la detección y clasificación automática de defectos de frutos externos del olivo basadas en técnicas de procesamiento de imágenes**". Mencionan que una de las principales preocupaciones de las empresas de venta de fruta, en la actualidad, es encontrar una forma efectiva para la clasificación y detección rápida de defectos de la fruta. El olivo es uno de los productos agrícolas más importantes, que recibe una gran atención de parte de las empresas de venta de frutas y verduras, por su utilización en diversas industrias, como la industria de los aceites y los encurtidos. En tal sentido proponen dos métodos que detectan automáticamente los defectos externos de las aceitunas utilizando el procesamiento de imágenes con un conjunto de 350 frutos de los cuales 200 defectuosos y 150 no defectuosos, estas técnicas pueden distinguir entre las aceitunas defectuosas y las saludables, y luego calcular con precisión y rapidez las áreas válidas de las aceitunas defectuosas. Estos métodos se basan en la homogeneidad entre píxeles vecinos. El primer método se basa en extraer las áreas defectuosas de los objetos seleccionados midiendo la extensión de la diferencia entre los píxeles. El segundo se basa en la aplicación de máscaras especiales a los lotes de fruta de olivo. Los resultados de los métodos propuestos se comparan con los algoritmos de agrupamiento de Canny, Otsu y KMeans y Fuzzy C-means, mostrando un resultado superior en términos de la precisión de clasificación, su capacidad para detectar el tamaño de las regiones defectuosas y la velocidad de procesamiento alcanzando un porcentaje de exactitud de hasta 100%.

Pereira et al. (2018) en su publicación denominada **“Predicción de la maduración de la fruta**



de papaya con imágenes digitales y bosques aleatorios". En la cual refieren que la clasificación de la papaya se realiza manualmente, lo que puede llevar a clasificaciones erróneas, dando como resultado cajas de fruta con diferentes etapas de madurez. Conociendo la importancia de una buena clasificación proponen una técnica para predecir la maduración de la papaya basada en el procesamiento de imágenes utilizando imágenes digitales y el algoritmo random forest para llevar a cabo dicho procedimiento. Se utilizan 114 muestras de 57 frutas para los experimentos y se clasifican en tres etapas de madurez siendo estado de madurez 1 (MS1; 58 imágenes), estado de madurez 2 (MS2; 30 imágenes) y estado de madurez 3 (MS3; 26 imágenes). Después de la adquisición y el análisis de la imagen, se extraen y evalúan 21 características de color, para validar el modelo se emplea dos conjuntos de datos (validación cruzada y conjunto de predicción). Con respecto a todas las características de la imagen; el resultado de la simulación generó un rendimiento del 94.3% en cuanto a la clasificación a través del conjunto de validación cruzada y el conjunto de predicción obtuvo un 94.7% de clasificación errónea de una sola muestra. Para las comparaciones de grupo, la media normalizada del espacio de color RGB logró un mejor rendimiento (78.1%) Pereira et al. (2018).

Mim et al. (2018) en su investigación **“Detección automática de las fases de maduración del mango - Una aplicación de la tecnología de la información a la botánica”**. Mencionan que la madurez es el principio más significativo para determinar el tiempo almacenamiento y la calidad de las frutas como los mangos y esta puede ser reconocida por diferentes atributos y entre ellos el color de la piel el cual es el criterio más importante para juzgar la madurez. Asimismo, detallan que por lo general el procedimiento de la clasificación es realizada por humanos expertos los cuales detectan visualmente el color del fruto para identificar las etapas de madurez y esto conlleva a cometer errores por distintos factores como pueden ser la concentración, exceso de horas de trabajo. Proponen un método de procesamiento de imágenes digitales para clasificar los mangos en seis etapas de madurez de acuerdo con la clasificación estándar del USDA. El experimento considera imágenes de muestra de más de 100 mangos de diferentes etapas. Se extraen un total de 24 características de cada imagen y luego se basan en la correlación y se realiza una evaluación basada



en la ganancia de información para seleccionar los conjuntos de características más informativos. La categorización se realiza utilizando el algoritmo de árbol de decisión el cual proporciona una precisión de hasta el 96%.

Deepa and Geethalakshmi (2013) en su investigación titulada como **“Análisis comparativo de métodos de extracción de características para la clasificación de frutas”**. Mencionan que la india está exportando una gran cantidad de frutas al exterior, una de estas frutas es la naranja mosambi la cual debe ser clasificada de acuerdo a su calidad antes de ser exportadas, conociendo tal importancia se propuso un procedimiento para evaluar las características extraídas utilizadas para la caracterización de frutos defectuosos y no defectuosos, se utilizó un conjunto de datos de 200 imágenes con resolución 640x480 píxeles, estas imágenes se capturaron con una cámara digital la cual se configuró en una posición fija con orientación aleatoria desde vistas perpendiculares, luego se realiza una fase previa de pre procesamiento donde las imágenes se normalizan inicialmente para producir uniformidad en términos de tamaño lo cual logra reducir el tiempo de procesamiento de este modo la imagen original se redimensiono a un tercio de su tamaño normal y la imagen RGB se convirtió a escala de grises quedando listas para el procesamiento. Se evaluaron atributos de textura como GLCM determinando siete características para la discriminación de texturas (auto correlación, contraste, sombra de grupo, entropía, suma de cuadrados y suma de entropía), características basadas en forma o también conocidas como geométricas o morfológicas se considera la forma del defecto como una característica valiosa y se pueden medir como: zona, centroide, diámetro, perímetro y longitud mayor y por último la característica de intensidad la cual es de primer orden y solo depende de valores de píxeles individuales, su variación dentro de las imágenes de la fruta se pueden medir por características como: mediana, moda, desviación estándar y varianza de esta base de datos. El método propuesto se clasificó utilizando el algoritmo PNN con ‘n’ entradas ‘m’ unidades ocultas y una unidad de salida, el resultado mostró que las características de textura, forma e intensidad dieron una tasa de identificación del 96%, 100%, 92%, respectivamente. En resumen, se examinó tres tipos de métodos de extracción de características y el resultado demuestra



que la característica de forma basado en PNN da una mayor tasa de clasificación en comparación con la textura e intensidad.

Lal et al. (2017) en su investigación **“Identificación y conteo de fruta madura de manzana basada en la red neuronal de BP Feed Forward”**. Detallan que la clasificación de las frutas es una tarea onerosa y tediosa debido a la cantidad innumerable de frutas con un enfoque tradicional para la detección, clasificación y nivel de madurez de la fruta este se basa en la observación a simple vista por parte de los expertos, lo cual consume mucho tiempo y causa fatiga ocular. Plantean un método para detectar y clasificar automáticamente el nivel de madurez de la fruta de manzana, ya sea madura o inmadura en función de sus características de color. La primera etapa del método propuesto inicia con el pre procesamiento de las imágenes para el redimensionamiento y reescalado con la finalidad de contribuir al procesamiento. Se utilizan 150 imágenes como entrada que forma una base de datos y cada imagen pertenece a cualquiera de las dos categorías, es decir, madura o inmadura así mismo se realiza el reescalado de la imagen, luego se extraen la matriz de características de color RGB de cada imagen, dicha matriz se normaliza para que los valores se encuentren entre 0 y 1, teniendo el conjunto de datos normalizado se divide en conjuntos de datos de formación y pruebas. (División aleatoria en RNA) a continuación se utiliza el clasificador para entrenar y crear el modelo utilizando el conjunto de datos de la capacitación, por último, se analiza el desempeño del clasificador y se calcula la matriz de confusión que se utiliza como conjunto de datos para la prueba. Si el clasificador genera resultados aceptables se procede con los experimentos, si no se vuelve entrenar el modelo. Los experimentos se llevaron a cabo en un procesador Intel®B960 que se ejecuta bajo el sistema operativo Microsoft Windows 7 con 2GB de memoria. El algoritmo se desarrolló en la plataforma Matlab 2014a (The Mathworks ©), de 32 bits. El modelo propuesto tiene una precisión del 98,1% utilizando el clasificador RNA, con 3 entradas (rojo, verde, azul), 10 nodos ocultos y la salida (madura o inmadura). Este método puede ser utilizado por los agricultores para evaluar el momento exacto de la cosecha de la fruta y las medidas de precaución necesarias pueden ser llevado a proteger la fruta de diversas enfermedades



en el futuro. Esto a su vez aumentará el rendimiento de la fruta de manzana.

Savakar (2012) en su investigación **“Identificación y Clasificación de Imágenes de Frutos a Granel usando Redes Neurales Artificiales”**. Especifica que la capacidad de decisión de los inspectores humanos se ve afectada por influencias externas como la fatiga, cansancio, los prejuicios, etc. Al momento de clasificar una fruta. Se desarrolló un SVM el cual clasificó cinco tipos diferentes de imágenes de frutas (manzana, chikoo, naranja, mango y limón dulce). El procedimiento consistió en registrar 1000 imágenes de cada tipo de fruta dando un total de 5000 imágenes de muestra. El algoritmo se desarrolló mediante la extracción de 18 colores y 27 características de textura, las características de color fueron calculadas por separado para cada componente RGB posteriormente la imagen RGB se convirtió al modelo de color HSI y sus componentes fueron separados, a continuación, se calculó la media, la varianza y el rango para cada componente RGB a HSI por separado. Las características de la textura se calcularon usando las GLCM para finalmente utilizar una BPNN para clasificar y reconocer las muestras. El estudio reveló que la clasificación de chikoo, manzana, limón dulce, naranja y mango fue 94%, 93%, 93%, 92%, 92%, correspondientemente. Cabe mencionar que las imágenes de este estudio son adquiridas de frutas limpias de impurezas. Los resultados de este estudio son útiles para la rápida identificación y clasificación de tipos de fruta a granel mediante el diseño de un elevador que mueve la fruta a través de una cámara.

Naik and Patel (2017) en su investigación **“Imágenes térmicas con clasificador difuso para la madurez y clasificación no destructiva del mango (Mangifera Indica L.) basada en el tamaño”**. Manifiestan que vivimos en la era digital, conociendo la importancia de la demanda del consumidor de mango de buena calidad refieren que la predicción de la madurez se lleva a cabo a través del color, pero para una variedad excepcional de mango como “Langdo”, el cual no cambia de color durante toda su vida y las imágenes normales no sirven para predecir su madurez, se requiere automatizar el proceso de clasificación. Se propone clasificar el fruto de mango en función de su madurez y tamaño utilizando un sistema difuso para la toma de decisiones. Para ello se utiliza



el algoritmo de intensidad media en el espacio de color $L^*a^*b^*$ y una cámara térmica de iPhone 5/5s para la captura de imágenes utilizando la aplicación FLIR ONE, se utilizó un base de datos de 250 imágenes utilizando diferentes vistas de 50 frutos de mango desde una posición superior, la distancia entre la cámara y mango es de 1 metro. Asimismo, hay que mencionar que una imagen contiene dos mangos de los cuales la madurez de uno ya es conocida el cual trabaja como muestra de referencia. Las imágenes capturadas tienen una resolución de 240 x 320 píxeles y todas están en espacio de color RGB. Hay que mencionar, además que los frutos del mango se evalúan utilizando dos características clave basadas en el tamaño y la madurez por medio de inspección visual y se dividen en dos grados, a saber, clase I y clase II, donde clase I = mango no maduro y parcialmente maduro con tamaño pequeño o mediano y clase II = mango maduro o parcialmente maduro con tamaño mediano o grande. Los experimentos se realizan en el software MATLAB r2013b de 64 bits para mac OS en mac book pro con procesador Intel i5 de 2,48 GHz, 4 GB de RAM y 1 GB de tarjeta gráfica, para fines experimentales se ha tomado como referencia el mango parcialmente maduro se obtuvo una precisión de 89% y su tiempo de ejecución es de 2.3 segundos. En conclusión, se realizó una clasificación basada en la madurez y el tamaño utilizando imágenes térmicas recibiendo una precisión del 89% y el tiempo empleado para clasificar un mango en forma individual es de 2.3 segundos, asimismo existen limitaciones en método propuesto y estas son: todo el mango debe almacenarse a la misma temperatura, se necesita un mango de referencia, una cámara térmica de alta calidad para mayor precisión y es preferible una cámara de mantenimiento de temperatura.

Salunkhe and Patil (2015) en su investigación **“Procesamiento de imágenes para la detección de la etapa de maduración del mango: Método RGB y HSV”**. Expresan que el mango es una fruta bastante mercantilizada y producida en el mundo por lo cual deben ser clasificados de acuerdo a su etapa de maduración para su uso comercial y este proceso en la actualidad es llevado a cabo manualmente lo cual no es exacto y es propenso a errores humanos. Proponen dos métodos de clasificación automatizada basados en los cambios de las características visuales. Se considera el fruto de mango “Alfonso”. El procedimiento para el método RGB en la fase de maduración es



leer el archivo de una imagen de color y por cada píxel, detectar la forma del mango, extraer de forma separada los datos de los componentes (rojo, verde y azul), calcular los valores medios de los componentes y finalmente calcular las relaciones entre el rojo a verde y verde a azul. Para método de HSV el procedimiento es exactamente el mismo excepto que los componentes RGB se sustituyen por HSV; se evaluaron tres conjuntos de 24 muestras los cuales contienen mangos en diferentes etapas de maduración. Los algoritmos de proporción RIG (Proporción de rojo a verde) y RIB (Proporción de rojo a azul) son propuestos e implementados usando MATLAB. Los resultados se comparan con los resultados manuales y se encontró que son 90.4% y 84.2% exactos en el caso de RGB y HSV respectivamente. Asimismo, se observó una tasa de falsos positivos del 2,57% en el caso del RGB y del 7,29% en el caso del HSV. Se observó una tasa positiva real de 89,77% y 86,74% en el caso del método RGB y HSV respectivamente. En conclusión, se encuentra que las tres proporciones en el método RGB, es decir, rojo a verde, rojo a azul y verde a azul, aumentan con el proceso de maduración, pero con tasas diferentes. También se demuestra que el color azul tiene el peso más bajo en las imágenes de mango, el color verde tiene el peso más alto en el mango no maduro y el color rojo tiene el peso más alto en el mango completamente maduro. También se estudió el efecto de la intensidad reflectante de luz ambiental y se encontró que el método RGB tiene mejor inmunidad que el método HSV para la detección de la madurez.

1.3 Teorías Relacionadas al Tema

1.3.1 Fruto de Mango

El fruto del mango se a vuelto un elemento primordial de la dieta en muchos países en vías de progreso en los trópicos Saúco (2019). Asimismo, este producto puede consumirse, comercializarse y transportarse en etapa verde o maduro, según las necesidades del mercado y del consumidor final Caribbean (2013).



1.3.1.1 Variedades de Fruto de Mango

El mango actualmente es considerado como una de las frutas mas valiosas del mundo Saúco (2019). Cabe mencionar que existe distintas variedades de las cuales en el peru predominan las de color rojo, verde y amarillo Agrobanco (2007). A continuación, se muestra una breve descripción de cada una de estas.

1. Variedad Roja

(a) Kent

Fruta de un tamaño aproximado de 500 - 800 gramos de color rojizo el cual es de excelente calidad para el consumo Vela (2006).



Figura 1.5: Fruto de mango kent. Fuente: Camino (2018)

(b) Haden

El tamaño de esta fruta varia de entre 380-700 gramos en su etapa de madurez obtiene un color rojo-amarillo Vela (2006).



Figura 1.6: Fruto de mango haden. Fuente: Perú (2018b)



(c) **Tommy Atkins**

Esta fruta es generalmente de un tamaño grande de promedio 600 gramos y se diferencia de los demás por su periodo de conservación Vela (2006).



Figura 1.7: Fruto de mango tommy atkins. Fuente: Perú (2018c)

2. **Variedad Verde**

(a) **Keitt**

Fruta de tamaño de entre 500 a 600 gramos de color verde con una mezcla roja Vela (2006).



Figura 1.8: Fruto de mango keitt. Fuente: Interempresas Media (2018)

(b) **Amélie**

Esta fruta tiene su origen en África Occidental y cuenta con poco contenido de fibra y es de color verde ligeramente anaranjada Vela (2006).





Figura 1.9: Fruto de mango amélie. Fuente: BienManger (2018)

3. Variedad Amarilla

(a) Ataulfo

Esta tipo de mango es muy común, generalmente tiende de un tamaño pequeño a mediano, cuenta con baja fibra pero con muchas vitaminas Vela (2006).



Figura 1.10: Fruto de mango ataulfo. Fuente: Perú (2018a)

(b) Manila Súper

Este tipo de mando es pasado y pequeño con aproximadamente 10 gramos es de forma aplanada y alargada con un sabor fuerte Vela (2006).



Figura 1.11: Fruto de mango manila Súper. Fuente: Walmart (2018)



(c) **Nam Doc Mai**

Es poco fibrosa y su semilla es pequeña y tiene un sabor dulce delicioso Vela (2006).



Figura 1.12: Fruto de mango nam doc mai. Fuente: Pride (2018)

1.3.2 Sensor de Color RGB TCS3200

El sensor de color RGB TCS3200 para arduino es un detector de color completo, que incluye un chip sensor TAOS TCS3200 RGB y 4 LED blancos, puede expresar y computar una variedad casi ilimitada de colores visibles (ver Figura 1.13) DFrobot (2018). Se puede aplicar para lectura, la clasificación por color, entre otros.

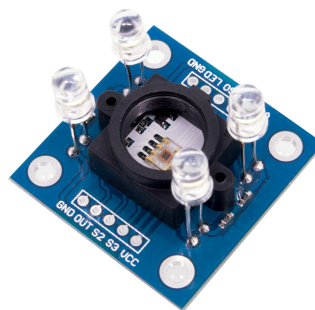


Figura 1.13: Sensor de color RGB TCS3200. Fuente: Electropro (2017)

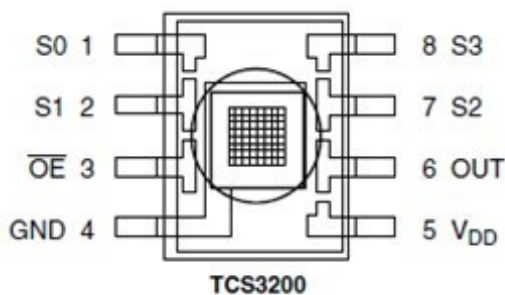
1.3.2.1 Especificaciones

Este módulo utiliza un sensor integrado provisto de 64 fotodiodos. De estos 64 fotodiodos, 16 tienen filtro para el color rojo, 16 para el color verde, 16 para el color azul y 16 para luz directa

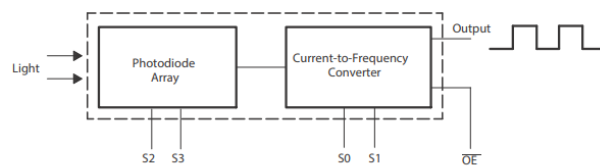


(sin ningún filtro). El diagrama de componentes del sensor de color RGB TCS3200 se evidencia en la Figura 1.14. Asimismo, el sensor cuenta con las siguientes especificaciones.

- Voltaje: 2.7V a 5.5V
- Alta resolución de conversión de la intensidad de luz de frecuencia
- Filtro de color programable (escala completa)
- Característica de apagado
- Comunicación directa con el microcontrolador (arduino, raspberry PI, etc)
- S0 - S1: Entradas de selección de escala de frecuencia de salida
- S2 - S3: Entradas de selección de tipo de fotodiodo
- Pin OUT: frecuencia de salida
- Pin OE: Pin de habilitación de frecuencia de salida (activo bajo)
- Control de suplemento de luz de lámpara LED de apoyo



(a) Diagrama de componentes



(b) Diagrama de bloques funcional

Figura 1.14: Diagrama de componentes sensor TCS3200. Fuente: Electropro (2017)



1.3.3 Microcontrolador Arduino

Arduino es una tablero electrónico de código abierto basado en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón , entre otros) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea Valenzuela (2018).

1.3.3.1 *Productos Arduino*

Existe una gama completa de productos oficiales Arduino, incluidos tableros, módulos (tableros clásicos pequeños), escudos (elementos que se pueden enchufar en un tablero para darle características adicionales) y kits. Estos se evidencian en la Figura 1.15, donde se describen cada categoría Arduino (2018).

El presente proyecto se utiliza un producto arduino mega perteneciente al tablero clásico de nivel de entrada, por ser fácil de utilizar ya que están listos para impulsar proyectos creativos debido a que cuentan con tarjetas y módulos correctamente conectados, esto se evidencia en la (Figura 1.16). Es por ello que se puede comenzar aprender a manipular la electrónica y la codificación.

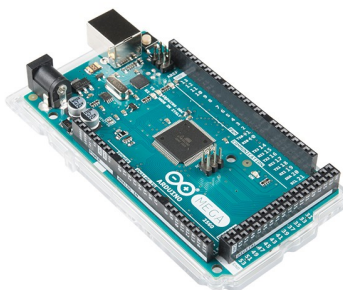


Figura 1.16: Microprocesador arduino mega. Fuente: Sparkfun (2018)



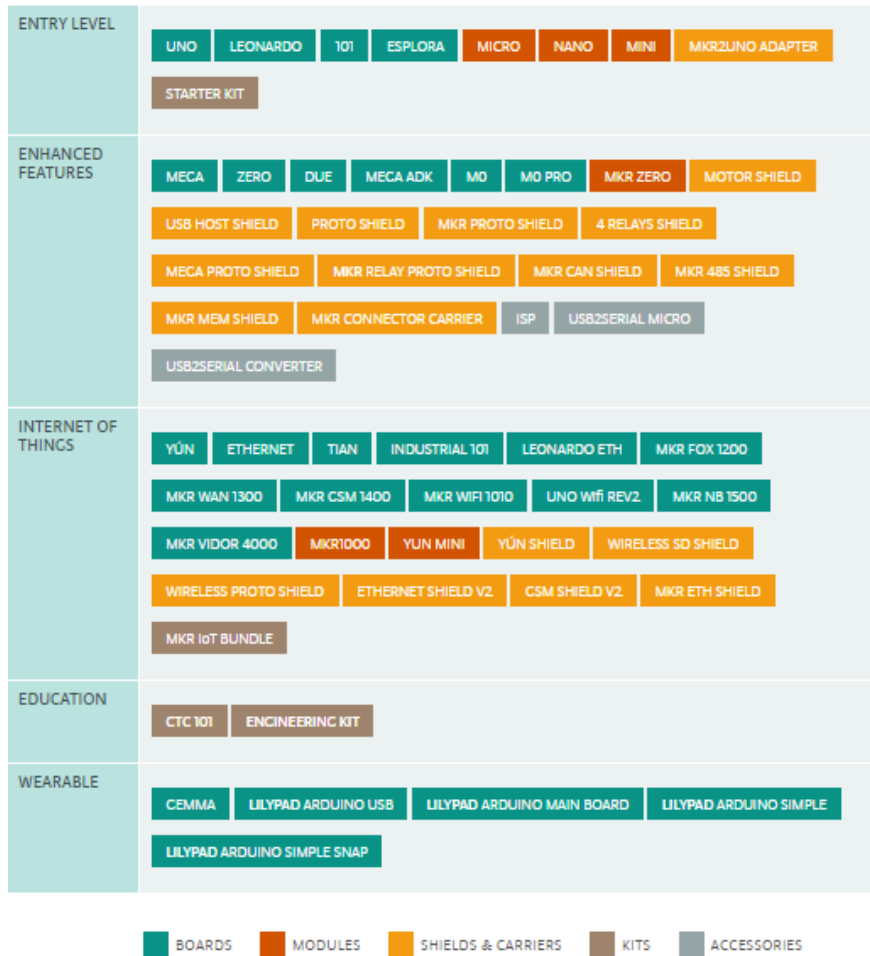


Figura 1.15: Listado por categorías de productos arduino. Fuente: Arduino (2018)

1.3.4 Modelos de Color

Representan modelos matemáticos abstractos los cuales permiten caracterizar los colores en forma numérica, utilizando usualmente 3 o cuatro componente cromáticos Ibraheem et al. (2012). Está basada en un conjunto de coordenadas, con base de N vectores, cuya conjunción lineal genera todo el espacio de color Ibraheem et al. (2012). Así mismo, los modelos de color tratan de representar la mayoría de colores sensibles al ojo humano, aunque existen modelos que no tengan tal finalidad por no ser conveniente para su futura aplicación, Estos modelos se pueden clasificar de acuerdo al número de dimensiones empleadas (ver Tabla 1.1).



Tabla 1.1: Clasificación de los modelos de color. Fuente: Domínguez and Peñalva (2003)

Modelo de Color	Clasificación
Munsell	Dispositivo independiente
RGB, CMY(K)	Dispositivo dependiente
YIQ, YUV, YCbCr	Dispositivo dependiente
HSI, HSV, HSL	Dispositivo orientado al usuario
CIE XYZ, CIE L*U*V*, CIE L*a*b*	Dispositivo independiente, color métrico

1.3.4.1 Modelo de Color RGB

Domínguez and Peñalva (2003) Lo define como un modelo tri dimensional RGB que está definido en un sistema cartesiano donde las bases de referencia son los colores rojos (Red), verde (Green) y azul (Blue), expresados como vectores ortogonales. Se basa en la síntesis aditiva, en el cual, los distintos colores se forman a partir de la mezcla de los tres colores primarios.



Figura 1.17: Modelo de color RGB. Fuente: Ibraheem et al. (2012)

Generalmente, la representación matemática se define con una intensidad comprendida entre 0 y 255 para cada color base, permitiendo crear 256 tonalidades diferentes. La combinación de los tres con sus variantes en intensidad ofrece millones de colores. Como resultado se genera un espacio de color con la forma de un cubo (ver Figura 1.1), donde cada color está definido dentro de este cubo. Según este modelo RGB la ausencia de luz (negro) está situado en las coordenadas (0, 0, 0) y en el vértice opuesto (255, 255, 255) se sitúa el blanco que representa la mezcla de todos los colores según la síntesis aditiva Keith (2005). La recta diagonal que une ambos vértices simboliza la escala de grises. En los vértices restantes del cubo se sitúan los colores primarios y secundarios.



Tabla 1.2: Colores RGB primarios y secundarios. Fuente: Keith (2005)

PRIMARIOS			SECUNDARIOS		
Rojo	Amarillo	Azul	Magenta	Cian	Verde
255, 0, 0	255, 255, 0	0, 0, 255	255, 0, 255	0, 255, 255	0, 255, 0

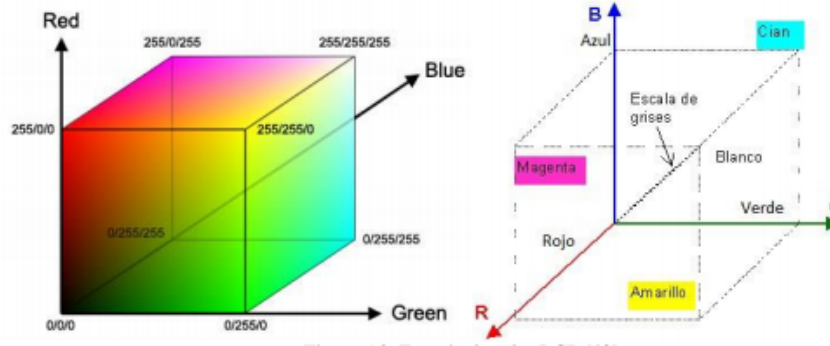


Figura 1.18: Representación cúbica de los colores primarios y secundarios. Fuente: Ibraheem et al. (2012)

Cada componente de color está representado por tres coordenadas RGB y esto permite una fácil cuantificación y codificación cuando se debe realizar cálculos computacionales Keith (2005). Es importante mencionar que cuando se genera un píxel de color este tendrá restricciones debido al propio sistema de producción del color y estas limitaciones son:

- Derivadas por el funcionamiento de las mezclas aditivas. Sólo pueden obtenerse colores interiores al triángulo formado por los tres colores primarios de la luz.
- Los colores primarios no son completamente monocromáticos en la práctica.
- Son derivados por el tipo de pantalla donde se proyecta.

La utilización del modelo RGB se observa en sistemas emisores o receptores de luz. En sus inicios estos dispositivos son los monitores, televisores o proyectores. En cuanto a los receptores estos se evidencian en cámaras digitales fotográficas o de videos Adrian and Roberts (1998).

1.3.4.2 Modelos de Color CIE

Son modelos de color que tienen diferentes combinaciones como se enumeran a continuación:



1.3.4.2.1 *Modelo de color CIE XYZ*

CIE hizo un gran número de mediciones de visión normal y estadísticas, y estableció el observador colorimétrico estándar en 1931, el cual se convirtió en la base del colorimétrico estándar CIE moderno Peifeng et al. (2016). Mientras este era usado para calibrar los espectros podrían ocurrir valores negativos. Debido a esta razón, el espacio de color XYZ se construyó utilizando 3 colores primarios imaginarios el cual evita los valores negativos como se muestra en la Figura 1.19, es un tipo de espacio de color estándar sin el afecto del equipo, que siempre se utiliza para describir el punto de referencia del espacio de color de cromaticidad. A continuación, se muestra la ecuación (1.1) para convertir valores de RGB a XYZ el cual es el puente para lograr los modelos de color LUV Y CIE L*a*b*.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.18 & 0.81 & 0.01 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.1}$$

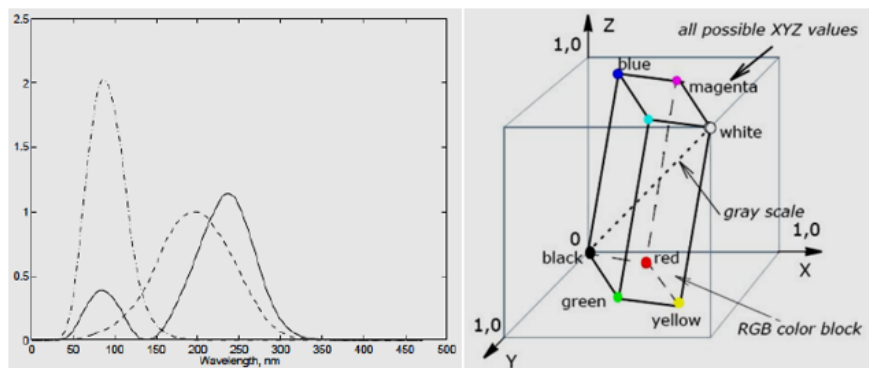


Figura 1.19: Modelo de color CIE XYZ. Fuente: Peifeng et al. (2016)

1.3.4.2.2 *Modelo de Color CIE L*a*b**

El modelo de color CIE L*a*b*, es en este momento uno de los más populares y uniformes usado para evaluar el color de un objeto recomendado por CIE en 1976 Zhu et al. (2012). Por otra parte,



este modelo de color está formado en base a una teoría de color oponente que establece que dos colores no pueden ser (rojo y verde) o (amarillo y azul) al mismo tiempo. Como se evidencia en la (Tabla 1.3), L^* indica la luminosidad y a^* y b^* son las coordenadas cromáticas Minolta (2014).

Tabla 1.3: Descripción de los componentes $L^*a^*b^*$ Fuente: Elaboración propia

Modelo de Color	Rango	Descripción
L^*	0 a 100	(claridad): 0 es negro; 100 es blanco.
a^*	-128 a +127	(rojo-verde): los valores positivos son rojos; los valores negativos son verdes; 0 es neutral
b^*	-128 a +127	(amarillo-azul) - los valores positivos son amarillo; los valores negativos son azules; 0 es neutral.

Martínez (2008) Menciona que estas características perceptuales se relacionan en un formato tridimensional, en el cual la claridad es el eje central (arriba-abajo), el tono es el giro, y el colorido indica la separación respecto al eje central (ver Figura 1.20).

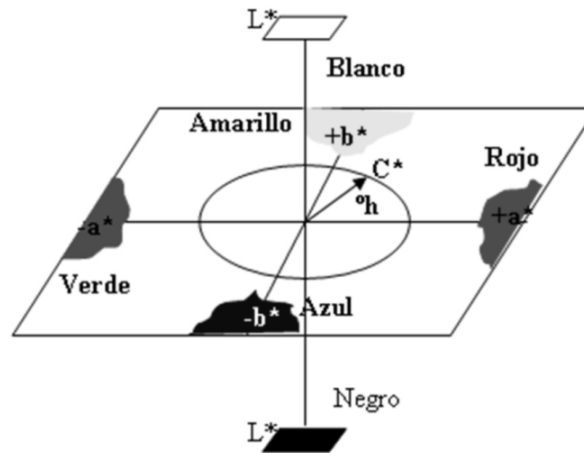


Figura 1.20: Modelo de color CIE $L^*a^*b^*$. Fuente: López and Di Sarli (2016)

A continuación, se describe las ecuaciones lineales para que hacen posible obtener una muestra del modelo XYZ a CIE $L^*a^*b^*$.



$$L^* = 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \tag{1.2}$$

$$a^* = 500 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} \right] \tag{1.3}$$

$$b^* = 500 \left[\left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n} \right)^{\frac{1}{3}} \right] \tag{1.4}$$

Donde $\frac{X}{X_0} > 0.01$, $\frac{Y}{Y_0} > 0.01$ son los valores tri estímulos de CIE XYZ y $\frac{Z}{Z_0} > 0.01$. (X_0, Y_0, Z_0) muestra el punto blanco de referencia Zhu et al. (2012).

1.3.5 Algoritmos de Clasificación Supervisada

Gironés et al. (2017) Indica que estos modelos conocen previamente el conjunto de clases y los patrones de características representantes a cada clase para el entrenamiento la Figura 1.21 evidencia el procedimiento de los datos descriptores y la clase objetivo.

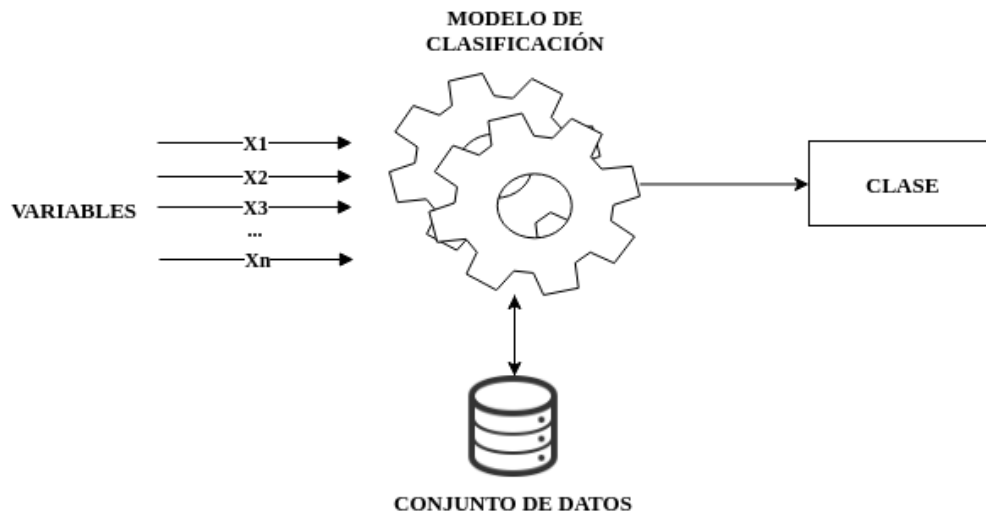


Figura 1.21: Método de clasificación supervisada. Fuente: Elaboración propia



1.3.5.1 Árboles de Decisión

Gironés et al. (2017) Refiere que los árboles de decisión son un modelo de minería de datos que intenta subdividir el espacio de datos de entrada para generar regiones disjuntas, de forma que todos los elementos que pertenezcan a una misma región sean de la misma clase, la cual es utilizada como representante o clase de dicha región.

En la Figura 1.22, se muestra un ejemplo de árbol de decisión así como su representación de partición del espacio que genera, intentando separar elementos de dos clases diferentes en un espacio de datos de entrada de dos dimensiones. En este caso, la profundidad máxima del árbol es dos. La primera condición (C1) crea dos regiones, una de las cuales (R1) solamente contiene elementos de una clase, por lo que no es necesario continuar dividiéndola. La otra región es dividida de acuerdo a una segunda condición (C2) creando dos nuevas regiones (R2 y R3). La región R2 es también pura y no es necesario dividirla de nuevo, mientras que la región R3 aún contiene elementos de las dos clases, por lo que el algoritmo debería ejecutarse hasta conseguir un árbol completo Gironés et al. (2017).

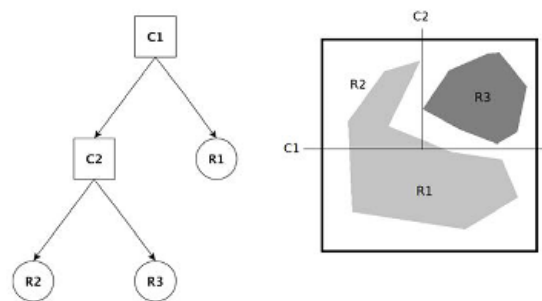


Figura 1.22: Ejemplo de árbol de decisión y la partición del espacio que genera. Fuente: Gironés et al. (2017)

El algoritmo genérico para construir un árbol de decisión utiliza los siguientes criterios:

- C_p o criterio de parada, determina en qué momento se deja de seguir seleccionando nodos



para ser subdivididos. El más habitual es generar un árbol completo, lo cual quiere decir que no quedan nodos que particionar.

- C_s o criterio de selección, determina qué nodo es seleccionado para ser particionado en dos o más sub nodos. Existen diferentes criterios de selección, pero si se generan árboles de decisión completos el criterio utilizado no es relevante.
- C_c o criterio de clasificación, determina qué clase se asigna a un nodo hoja. Normalmente se trata de la clase que minimiza el error de clasificación o el valor numérico que minimiza el error de regresión.
- C_d o criterio de partición, determina cómo se particiona un nodo en dos o más subnodos. Normalmente los árboles de decisión son binarios (es decir, $P = 2$), indicando que cada región se subdivide en dos regiones disjuntas. Esto simplifica el criterio de partición y mejora la intratabilidad del árbol a costa de generar árboles más profundos.

El algoritmo genérico para construir un árbol de decisión o, lo que es equivalente, una partición del espacio de entrada, se muestra en la Figura 1.1.

1.3.5.2 *k-Nearest Neighbors*

Este algoritmo de clasificación parte de un conjunto de datos de entrenamiento y tiene como objetivo clasificar correctamente todas las instancias nuevas. El conjunto de datos típico de este tipo de algoritmos está formado por varios atributos descriptivos y un solo atributo objetivo, también llamado clase (Gironés et al., 2017, p137).

1.3.5.2.1 *Funcionamiento del Método*

El algoritmo k-NN no genera un modelo fruto del aprendizaje con datos de entrenamiento, sino que el aprendizaje sucede en el mismo momento en el que se pide clasificar una nueva instancia.



Tabla 1.4: Pseudocódigo del algoritmo árbol de decisión. Fuente: (Gironés et al., 2017, p2014)

Entrada: conjunto de datos a clasificar D
 $T = D$ // El árbol inicial es un solo nodo hoja
 Etiquetar T de acuerdo a C_c
 $p = T$ // Lista de nodos pendientes (hojas)
mientras no se deba parar según C_p **hacer**
 Seleccionar un nodo q de acuerdo a C_s
 si es posible particionar q según C_d **entonces**
 Particionar q en q_1, \dots, q_P
 Etiquetar q_1, \dots, q_P según C_C
 Añadir q_1, \dots, q_P a p
 Sustituir q en T por un nodo interno
 fin si
 Eliminar q de p
fin mientras
devolver T

Tabla 1.5: Pseudocódigo del algoritmo k-NN. Fuente: (Gironés et al., 2017, p137)

Entrada: k (número de vecinos) y x (nueva instancia a clasificar)
para todo $d_i \in D$ **hacer**
 Calcular la distancia $d(d_i; x)$
fin para
 Inicializar I con las clases c_i de las k instancias de entrenamiento más próximas a x .
devolver La clase mayoritaria en I

A continuación se describe el algoritmo.

1. Fijamos una valor para k , habitualmente pequeño.
2. Calcule la distancia entre los nuevos datos a cada uno de los datos etiquetados
3. Determine k datos etiquetados que tengan la menor distancia.
4. Finalmente, clasifica los nuevos datos en la mayoría de los datos etiquetados.

El cálculo de la distancia se puede buscar utilizando la distancia euclidiana que se muestra en la ecuación 1.5.



$$D_i = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \tag{1.5}$$

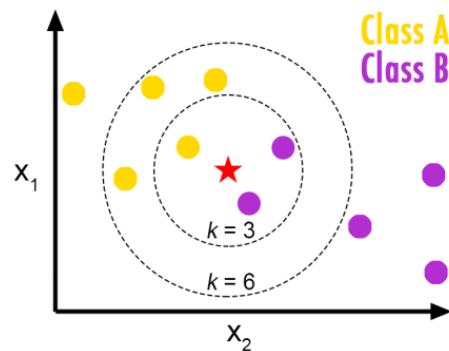


Figura 1.23: k-NN donde k = 3 y k = 6 Fuente: Widiyanto et al. (2019)

1.3.6 Validación y Evaluación de los Resultados

a) Conjunto de Entrenamiento y Test

La validación de un algoritmo de aprendizaje automático se realiza para verificar o validar que un método se fehacientemente factible para futuras pruebas Gironés et al. (2017). Así mismo, se debe conocer el volumen de los datos utilizados para el entrenamiento para evitar el problema conocido como *sobre entrenamiento*.

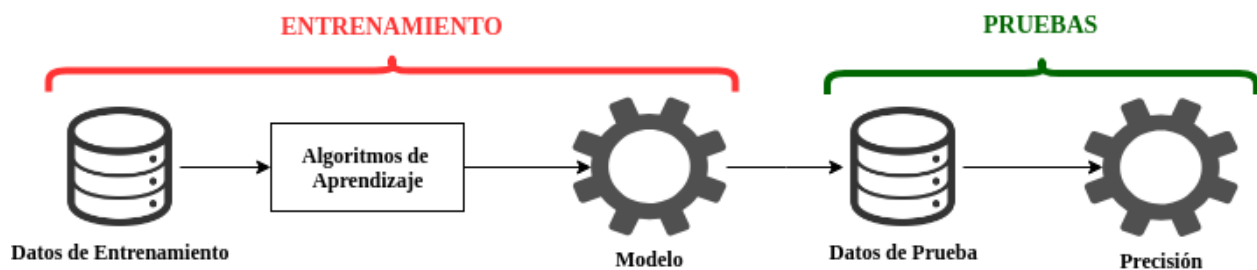


Figura 1.24: Fase de creación y validación de un modelo apoyado en aprendizaje supervisado Fuente: Elaboración propia



b) **Validación cruzada K-Fold**

Si bien es cierto un modelo de aprendizaje genera un resultado que puede considerarse a simple vista como valido pero no es correcto para ello se debe validar los resultados esperados utilizando la técnica conocida como validación cruzada k-fold la cual utiliza los datos para entrenar de forma ascendente de modo de cruzar la información como se evidencia en la Figura 1.25.

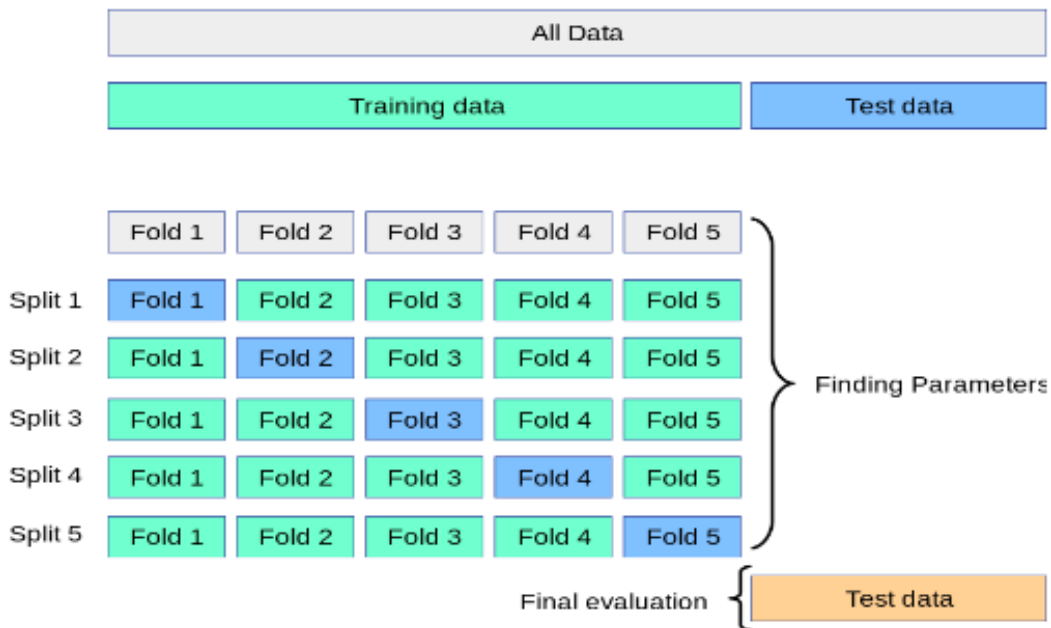


Figura 1.25: Validación cruzada de K iteraciones con K=4

c) **Matriz de Confusión**

Se trata de un modelo gráfico para visualizar el nivel de acierto de un modelo de predicción. También es conocido en la literatura como tabla de contingencia o matriz de errores Gironés et al. (2017).

La Tabla 1.6, presenta la matriz de confusión para el caso básico de clasificación binaria y en la tabla 1.7, se muestra la matriz de confusión para tres clases.



Tabla 1.6: Matriz de confusión binaria Fuente: Elaboración propia

Matriz de Confusión Valor Real	Predicción	
	Positivo	Negativo
Negativo	A	B
Positivo	C	D

Esta matriz indica el número de instancias correcta e incorrectamente clasificadas. Los parámetros que nos indica son:

- Verdadero positivo (A): número de clasificaciones correctas en la clase positiva (P).
- Verdadero negativo (D): número de clasificaciones correctas en la clase negativa (N).
- Falso negativo (C): número de clasificaciones incorrectas de clase positiva clasificada como negativa.
- Falso positivo (B): número de clasificaciones incorrectas de clase negativa clasificada como positiva.

Tabla 1.7: Matriz de confusión con tres clases. Fuente: Elaboración propia

Matriz de Confusión		PREDICCIÓN			Falso Negativo (FN)	Recall		
		Clase 1	Clase 2	Clase 3				
VALOR REAL	Clase 1	A	B	C	B+C	A/(A+B+C)		
	Clase 2	D	E	F			D+F	E/(D+E+F)
	Clase 3	G	H	I				
	Falso Positivo (FP)	D+G	B+H	C+F	Precisión General = A+E+I/(Suma de los cuadros rojos y verdes)			
	Precisión	A/(A+D+G)	E/(B+E+H)	I/(C+F+I)				

■ Verdadero Positivo
 ■ Verdadero Negativo
 ■ Casos Mal Clasificados
 ■ Falso Positivo
 ■ Falso Negativo

A continuación se describe en la tabla 1.8, las métricas derivadas de la matriz de confusión para tres clases.



Tabla 1.8: Métricas de evaluación del rendimiento. Fuente: Elaboración propia

Métrica de Rendimiento	Fórmula	Cálculos
Exactitud	$(TP+TN)/(TP+TN+FP+FN)$	$(A + E+I)/(A + E+I + D+G + B+C)$
Precisión	$TP/(TP+FP)$	$A/(A + D+G)$
Recall	$TP/(TP+FN)$	$A/(A + B+C)$
Especificidad	$TN/(FP+TN)$	$(E+I)/(D+G + E+I)$
F-Score	$2TP/(2TP+FP+FN)$	$(2 * A)/(2 * A + D+G + B+C)$
Coeficiente de Correlación de Matthews	$TP * TN - FP * FN / \sqrt{((TP+FP) * (TP+FN) * (TN+FP) * (TN+FN))}$	$(A * E+I - D+G) / \sqrt{((A + D+G) * (A + B+C) * (E+I + D+G) * (E+I + B+C))}$

MCC: Coeficiente de Correlación de Matthews
■ Verdadero Positivo ■ Verdadero Negativo ■ Casos Mal Clasificados ■ Falso Positivo ■ Falso Negativo

d) **Curvas ROC**

Una curva ROC mide el rendimiento respecto a los falsos positivos y verdaderos positivos . En esta métrica, un clasificador perfecto ocuparía la posición superior izquierda de la gráfica, con una tasa de verdaderos positivos igual a 1 y una tasa de falsos positivos igual a 0. A partir de la curva ROC se calcula el área bajo la curva AUC que permite caracterizar el rendimiento del modelo de clasificación Gironés et al. (2017).

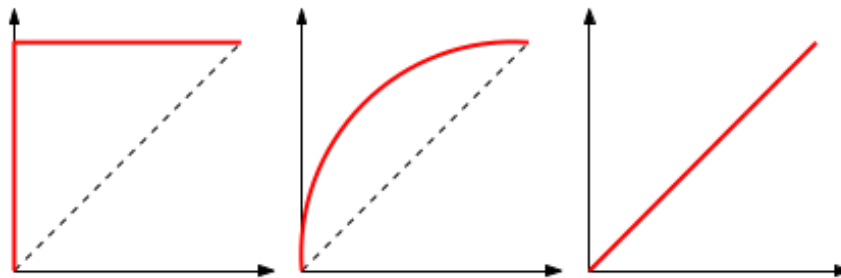


Figura 1.26: Ejemplo de la curva ROC para (AUC = 1), (AUC = 0.8) y (AUC = 0.5) Fuente: Gironés et al. (2017)

A modo de mentor para ilustrar las curvas ROC se han determinado los siguientes intervalos para los valores de AUC:



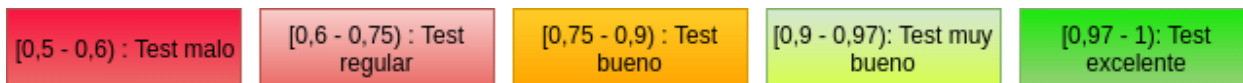


Figura 1.27: intervalos para los valores ROC Gironés et al. (2017)

1.4 Formulación del Problema

¿Qué algoritmo de clasificación será el más eficiente para determinar el estado de madurez del fruto de mango?

1.5 Justificación e Importancia del Estudio

El proyecto surge debido a que en la actualidad la clasificación adecuada de la fruta de mango según su estado de madurez es muy importante para garantizar la calidad de selección en beneficio de la agricultura y las exportaciones, es precisamente en este proceso donde los agricultores no utilizan algún tipo de automatización para realizar la selección y se sigue clasificando según la perspectiva del ojo experto del ser humano. Así mismo, existe propuestas principalmente realizado el procesamiento de imágenes RGB generando buenos resultados; la diferencia de este proyecto es que se utilizará un sensor de color RGB TCS3200 para extraer las muestras de color del fruto de mango y poder caracterizarlas según su estado de madurez. En cuanto a la viabilidad el proyecto no presenta ningún impacto negativo en la agricultura y en lo económico los gastos realizados son mínimos por tratarse de interfaces electrónicas comerciales y software libre.

Hay que mencionar, además que este proyecto pertenece a la línea de investigación **ciencias de la computación** aprobado por la escuela de Ingeniería de Sistemas de la Universidad Señor de Sipán en la directiva N° 002-2018-VRI-USS, la misma ha evolucionado de manera notable en el desarrollo de soluciones automatizadas, así pues el resultado de este trabajo en el aspecto pedagógico, permitirá contar con un análisis de rendimiento en términos de exactitud y tiempo de respuesta de dos algoritmos de aprendizaje clasificación, lo cual permitirá ser una fuente de información para trabajos futuros; en lo social, dar a conocer a la población, que la visión por computadora puede ser utilizada para realizar este tipo de procedimientos experimentando un gran porcentaje de acierto.



1.6 Hipótesis

El algoritmo de clasificación Árbol de Decisión es el más eficiente para determinar el estado de madurez del fruto de mango.

1.7 Objetivos

1.7.1 Objetivo General

Realizar análisis comparativo de los algoritmos de clasificación para determinar el estado de madurez del fruto de mango.

1.7.2 Objetivos Específicos

- Implementar el protocolo para la extracción de datos.
- Seleccionar los algoritmos de clasificación.
- Caracterizar el color del fruto de mango según su estado de madurez.
- Construir la base del conocimiento basándose en el modelo de datos.
- Implementar la propuesta utilizando los algoritmos seleccionados en un lenguaje de programación.
- Realizar análisis comparativo de los resultados obtenidos



2. MATERIAL Y MÉTODO

2.1 Tipo y Diseño de Investigación

2.1.1 Tipo de Investigación

Esta investigación es del tipo **cuantitativa, aplicada y tecnológica**, ya que por su naturaleza se deriva de conocimiento científico apoyando por las ciencias de la computación. Asimismo, se utilizará métodos ya estudiados anteriormente y llevados a la práctica, también pretende medir las cualidades con las que se trabaja con valores como tiempo de respuesta y exactitud. Los cuales permitirán comprobar a través del desarrollo de los estudios el porcentaje de aciertos con respecto al experto.

2.1.2 Diseño de Investigación

Este trabajo pertenece al tipo de diseño **cuasi experimental**, ya que se manipulara la muestra para generar mejores resultados. Se debe aclarar que por naturaleza se van a tomar ciertos procesos, métodos y estrategias existentes para llevarlos a la práctica y comprobar el grado de certeza de un determinado estudio.

2.2 Población y Muestra

2.2.1 Población

Se realizó un estudio teórico de 9 algoritmos de clasificación supervisada los cuales han sido utilizados en diferentes investigaciones previas relacionadas al propósito de investigación las cuales se muestra en la (Tabla 3.2) y estos son RNA, BPNN, PNN, K-Means, SVM, k-NN, Bosques aleatorios, Árboles de decisión, Fuzzy y Bayes óptimo.

2.2.2 Muestra

Por conveniencia se ha considerado dos algoritmos de clasificación supervisados: Árboles de Decisión y k-NN, debido a que, se han realizado estudios de rendimiento en términos de precisión y han generado excelentes porcentajes de aciertos en comparación con los mencionados [2.2.1]



como se muestra en la (Figura 3.11).

2.3 Variables, Operacionalización

2.3.1 Variables

2.3.1.1 Variable Independiente

Algoritmos de clasificación

2.3.1.2 Variable Dependiente

Determinar el estado de madurez del fruto de mango

2.3.2 Operacionalización

Tabla 2.1: Indicadores y métricas para el análisis de las variables detallando los métodos, técnicas e instrumentos de recolección de los datos. El cálculo de de las ecuaciones presentadas se describe en la Tabla 1.8. Fuente: Elaboración propia apoyado en la teoría de Gironés et al. (2017)

Variable	Indicadores	Fórmula	Técnica e Instrumento de Recolección de Datos
Variable Independiente: Algoritmos de clasificación	Tiempo de Respuesta	$T_R = T_A - T_I$	Observación - Ficha de observación
Variable Dependiente: Determinar el estado de madurez del fruto de mango	Error Exactitud	$ERR = \frac{F_P + F_N}{F_P + F_N + T_P + T_N}$ $ACC = 1 - ERR$	Observación - Ficha de observación

2.4 Técnicas e Instrumentos de Recolección de Datos, Validez y Confiabilidad

2.4.1 Técnica de Recolección de Datos

2.4.1.1 Observación

Esta práctica se utiliza a lo largo de todo el proceso de desarrollo de la propuesta de tesis siendo los puntos más importantes comprobar los resultados de la implementación del método planteado [3.7], para la clasificación del fruto de mango Ataulfo según su estado de madurez.



2.4.2 Instrumento de Recolección de Datos

2.4.2.1 Ficha de Observación

A medida que se desarrolle la propuesta se gestionara cada cambio para de este modo salvaguardar y generar un consolidado histórico de los pasos realizados para poder llevar a cabo el objetivo.

2.5 Procedimientos de Análisis de Datos

Se realiza un procesamiento estadístico descriptiva utilizando el lenguaje de programación Python para la orden de la exposición, análisis e interpretación de los resultados finales de la investigación apoyado en los indicadores presentados en la operacionalización 2.3.2.

2.6 Criterios Éticos

Están relacionados con cualquier suceso en el cual se puede ver involucrado esta investigación, los cuales se describen a continuación.

2.6.1 Confiabilidad

La información y la autoría utiliza en la siguiente investigación deberá adquirirse de forma legal evitando así daños en terceros relacionados; como se expresa en la *ley numero 29733 : “Ley de protección de datos personales”, en su título IV: “Obligaciones del titular y del encargado del banco de datos personales” En su artículo 28 señala: Que recopilar datos personales por medios ilícitos, fraudulentos o desleales está penado por el estado peruano.*

2.6.2 Conformabilidad

Los resultados expuestos en la presente investigación serán validados por un experto conocedor en la materia de acuerdo al *El Código Deontológico del Colegio de Ingenieros del Perú en su Capítulo III “Faltas Contra la Ética Profesional y Sanciones”, en su Artículo 105 señala: Los ingenieros serán objetivos y veraces en sus informes y declaraciones, y expresaran opiniones en temas de ingeniería.*



2.7 Criterios de Rigor Científico

- **Originalidad :**

Se especificará las referencias de las fuentes de información consultadas, con el objetivo de poder ser evaluados por el software de antiplagio para demostrar el porcentaje de referencias aceptado.

- **Validez :**

Respecto a los datos resultantes por la propuesta planteada estos van a ser evaluados y evaluados por docentes especializados en tema que se aborda para justificar su veracidad.

- **Consistencia :**

La propuesta planteada expondrá contenido consistente y certificado por los estándares establecidos por la comunidad científica.

- **Fiabilidad :**

La propuesta no se alejara de los objetivos planteados de modo de cumplir con la expectativa planteada basándose en estándares y respetando las políticas internas y externas para su implantación.



3. RESULTADOS

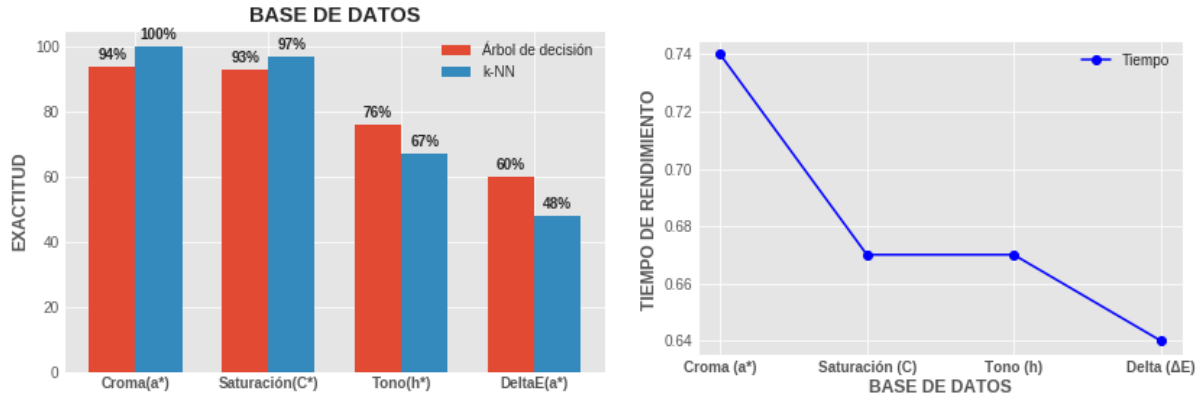
3.1 Resultados en tablas y figuras

El análisis de los resultados alcanzados se fundamenta en la ejecución de los modelos utilizando la técnica de validación cruzado k-fold [3.3.5.1.4] tanto para los algoritmos de árbol de decisión y k-NN la cual evidencia que los modelos utilizando la base de datos con la intensidad dominante del canal croma 'a*' genera un rendimiento del 94%, 100%. Así mismo, la base de datos de Saturación (C) produce un rendimiento de 93%, 97% y la base datos Tono (h) experimenta un resultado de 76%, 67% finalmente la base datos con valores de la diferencia total del color (ΔE) experimenta un resultado de 60%, 48% respectivamente como se expone en la Figura 3.1a. Se debe agregar que, los resultados obtenidos se interpretan según la valoración de las curva ROC [1.27] establecida por Gironés et al. (2017), dicha interpretación se detalla en la Tabla 3.2.

Tabla 3.1: Resultados de la ejecución de los modelos y valoración según Gironés et al. (2017)
Fuente: Propia

Base de datos	Algoritmo	Tiempo de Rendimiento	Resultado	Estado
Croma (a*)	Árbol de decisión	0.74 milisegundos	94%	Test muy bueno
	k-NN		100%	Test excelente
Saturación (C)	Árbol de decisión	0.67 milisegundos	93%	Test muy bueno
	k-NN		97%	Test muy bueno
Tono (h)	Árbol de decisión	0.67 milisegundos	76%	Test bueno
	k-NN		67%	Test regular
Delta E	Árbol de decisión	0.64 milisegundos	60%	Test regular
	k-NN		48%	Test malo





(a) Exactitud promedio

(b) Tiempo de rendimiento

Figura 3.1: Rendimiento de la ejecución de los algoritmos en términos de exactitud y tiempo de ejecución Fuente: Elaboración propia

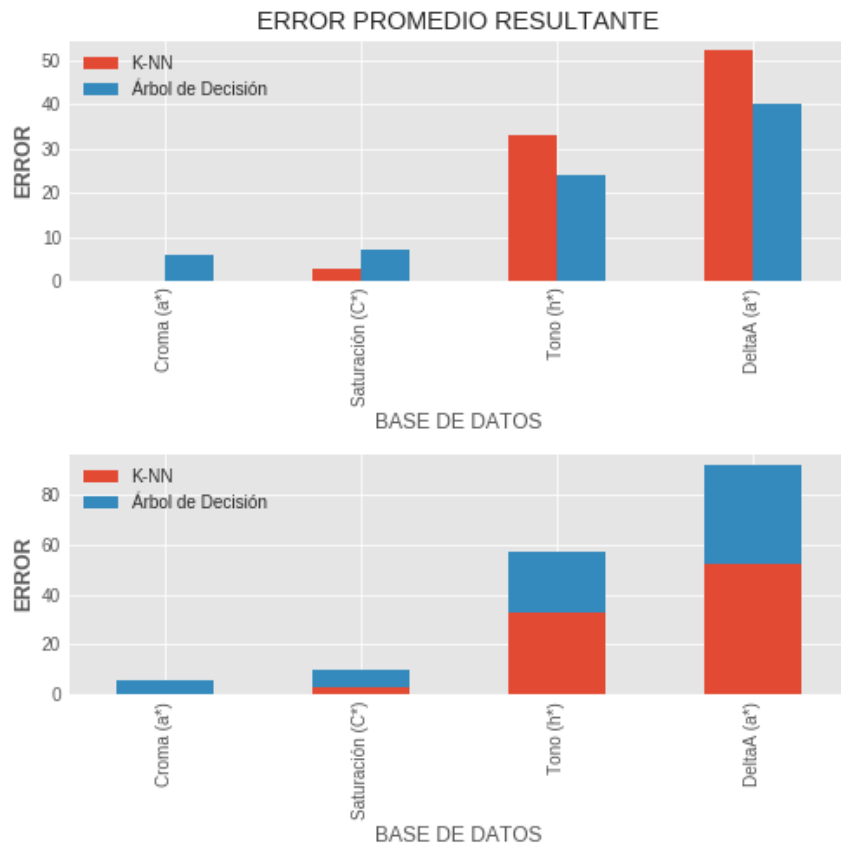


Figura 3.2: Error promedio resultante Fuente: Elaboración propia



En relación con lo antes mencionado se observa que el rendimiento de los algoritmos está estrictamente ligado a la distribución de los datos como se evidencia en la Figuras[3.17][3.18][3.19][3.20], ya que los algoritmos utilizan distancias para clusterizar la información, por lo tanto mientras más agrupado estén los datos mejor será el rendimiento; por otro lado el tiempo de rendimiento de los algoritmos es estable varía entre 0.64 y 0.74 ms como se muestra en la Figura 3.1b, así mismo el error promedio demuestra que el algoritmo k-NN obtiene una mejor performance trabajando con datos de croma (a^*) y saturación (C^*) y el algoritmo Árbol de decisión genera mejores resultados trabajando con datos de tono (h^*) y ΔE de (a^*). Lo dicho hasta aquí supone que los datos cromáticos y los de saturación son mejores en cuanto a la caracterización de la fruta según su estado de madurez, no siendo así para el tono que experimenta un variación aceptable de la caracterización y finalmente los datos extraídos gracias a la diferencia total de color entre dos muestras de croma (a^*) los cuales experimentan una caracterización mala del estado madurez ya que el éxito de esta característica depende de muchas variables como pueden ser la reflectancia, iluminación, etc.

A continuación, se muestra la ejecución de los modelos en la interfaz GUI [3.21].

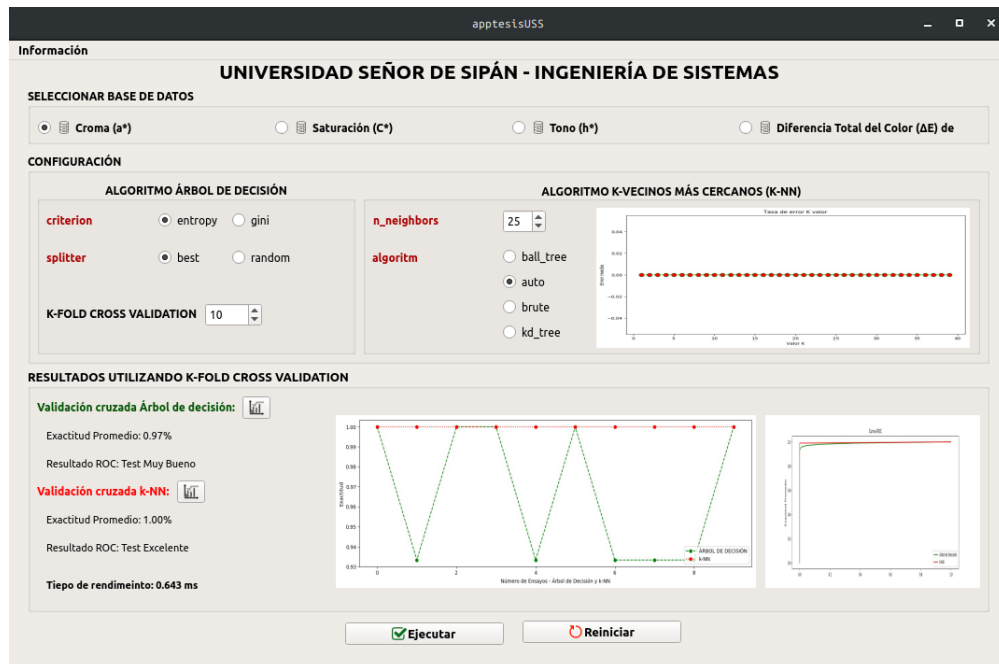


Figura 3.3: Ejecución croma(a^*)



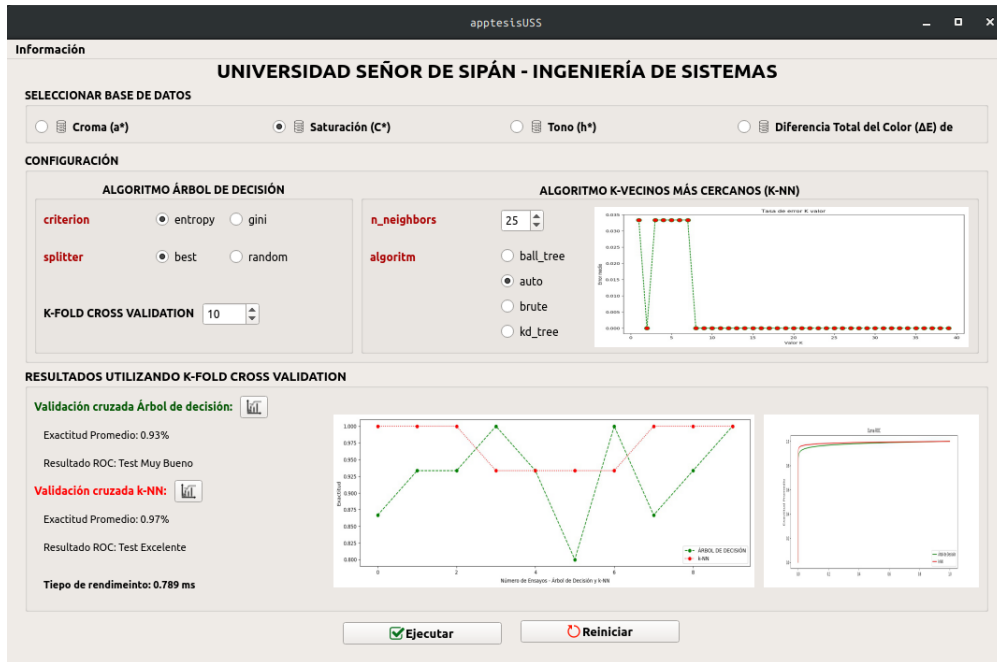


Figura 3.4: Ejecución saturación(C*)

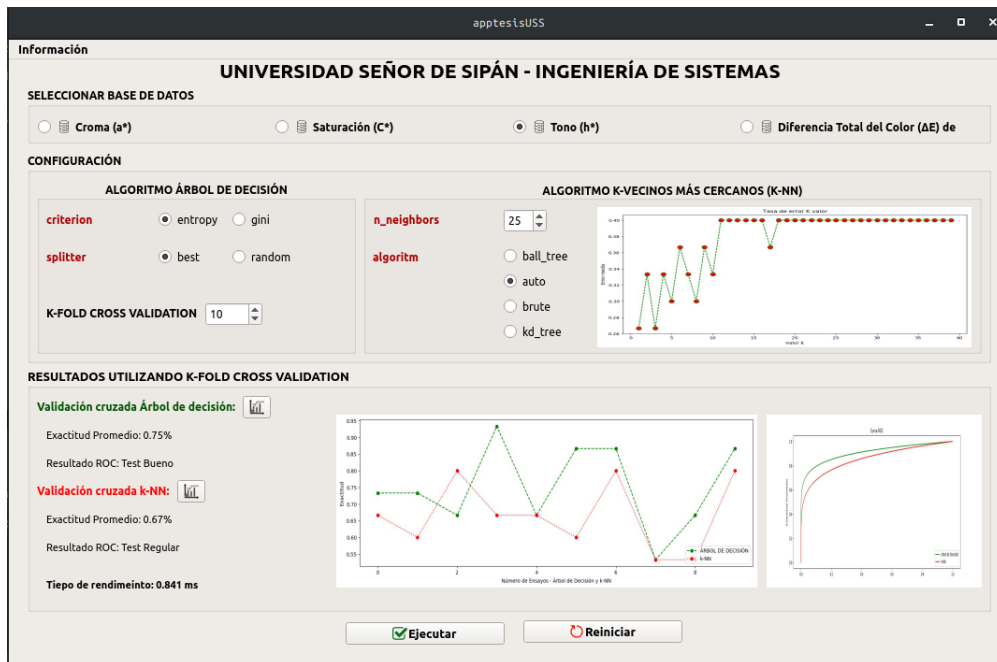


Figura 3.5: Ejecución tono(h*)



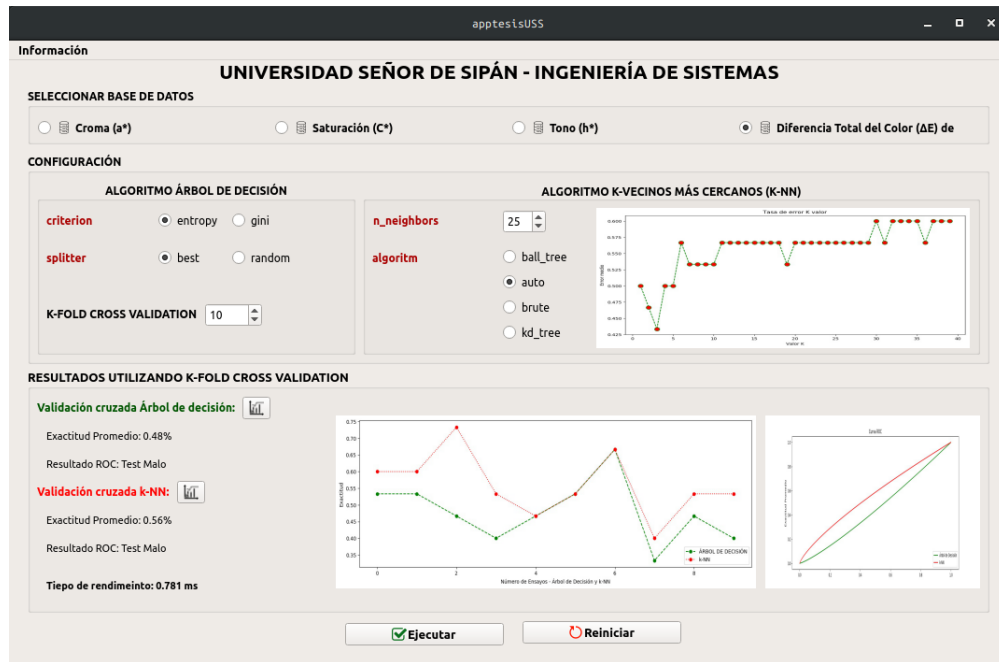


Figura 3.6: Ejecución diferencia Total(ΔE) de a^*

3.2 Discusión de resultados

La Figura 3.1 muestra una tasa de éxito o la exactitud de clasificación de los algoritmos propuestos, estos resultados se pueden comparar con los algoritmos descritos en la población [2.2.1]. Asimismo, los resultados de la ejecución mostraron que los algoritmos propuestos [2.2.2], logran identificar y diferenciar los tres estados de madurez del fruto de mango con éxito. Hay que mencionar, además que se a utilizado los componentes de el modelo de color CIE $L^*a^*b^*$ el cual representa el color desde una percepción similar al ojo de un ser humano aprovechando su componente de luminosidad teniendo en cuenta los elementos de tono, saturación, croma (a^*), la diferencia total del color los cuales demuestran ser índices importantes en cuanto a la caracterización de las muestras a las que representan. Además se evidencia una cercanía en la tasa de clasificación entre los algoritmos propuestos y los algoritmos detallados en la población [2.2.1], los cuales consideran ciertas características tales como el color y la textura generando un alto índice de rendimiento; se debe resaltar que a diferencia de la propuesta realizada a las planteadas [1.2] radica principalmente en la extracción y limpieza de las muestras ingresadas en las máquinas de



aprendizaje automático en lo que refiere al procesamiento de imágenes el cual es el método más importante en cuanto al avance tecnológico para llevar a cabo la clasificación automatizada de los productos agrícolas en donde se debe realizar un preprocesamiento de las imágenes lo cual significa un costo computacional importante dependiendo de la cantidad y el tipo de procesamiento, si bien es cierto que se gana en información a procesar se debería tener en cuenta este indicador; todo lo contrario sucede con la propuesta planteada la cual se basa en información extraída del sensor de color RGB TCS3200 para etiquetar las muestras y generar un conjunto de información fiable para poder entrenar modelos de clasificación con éxito. Es necesario recalcar que si bien es cierto la propuesta planteada no significa una ganancia en cantidad de información a procesar busca demostrar que el uso del sensor mencionado puede ser un medio fiable para llevar a cabo dicho procedimiento. Lo dicho hasta aquí no quiere decir que el método planteado sea mejor o peor que los métodos propuestos si no ser un experimento plenamente controlado y demostrar que el uso del sensor de color puede ser alternativa para llevar a cabo tales trabajos industriales acoplado a otros procedimientos fundamentales como puede ser el manejo de la reflectancia. Por otra parte se debe mencionar que se evidencia un promedio de efectividad superior al los algoritmos Fuzzy Logic, Naive Bayes, Bayes Óptimo y SVM utilizados en los antecedentes.

3.3 Aporte práctico

La propuesta que se plantea en esta tipo de investigación [2.1.1], es emplear algoritmos de clasificación supervisados utilizando muestras extraídas del sensor de color RGB TCS3200 para realizar la clasificación de los frutos de mango según su estado de madurez utilizando únicamente la característica predictora de color.

El primer paso es definir el protocolo para la extracción de los datos de color según los estados de madurez del fruto de mango, para ello se crea una interfaz electrónica la cual utiliza un sensor de color RGB TCS3200 controlado por un microcontrolador arduino Mega. A continuación, se caracteriza el color del fruto y se etiqueta cada muestra considerando los estados de madurez (*verde*, *semi maduro* y *maduro*), luego estas muestras se convierten del modelo de color RGB al modelo



de color CIE $L^*a^*b^*$ para poder trabajar bajo un modelo de color que se asemeje a como el ser humano percibe el color, seguidamente se extrae las bases de datos con los valores de la intensidad dominante del canal cromático a^* , Tono, Saturación y Diferencia total del color (ΔE) de (a^*) , las cuales se estandarizan para poder realizar el entrenamiento y prueba de los modelos de clasificación.

Teniendo en cuenta lo antes expuesto, se plantea utilizar los algoritmos árbol de decisión y k-NN para llevar a cabo la clasificación, separando los datos de forma uniforme tomando el 80% para el entrenamiento y el 20% para prueba. Habría que decir también, que el diagrama de procesos a llevar a cabo la propuesta se evidencia en la Figura 3.7, la cual detalla cada paso del procedimiento. Por último, se realiza el análisis comparativo de los resultados obtenidos.

3.3.1 Implementar el protocolo para la extracción de datos

La implementación del protocolo para la extracción de muestras de color inicia creando una interfaz electrónica la cual se realiza bajo un prototipo específico la misma que se muestra en la Figura 3.8, Así mismo, se ha optado por siguiente diagrama de conexión para realizar el nexo entre el sensor RGB TCS3200 y el microcontrolador arduino Mega el cual se evidencia en la Figura 3.9. Cabe subrayar que la distancia entre el sensor y el fruto de mango es tocando el aislador del mismo a una distancia aproximada de entre 1 centímetro del infrarrojo. Además, se utilizan tubos PVC-SAP de media pulgada a una distancia de 10 centímetros de altura como canales de circuitos para poder realizar la conexión. Hay que mencionar, además que el código Fuente arduino empleado para poder llevar a cabo esta conexión se expone en el algoritmo 5.1 el cual se configuró con frecuencia de salida igual 20% y se emplea 3 filtros de color (rojo, verde, azul) para el óptimo mapeo del color del fruto de mango.



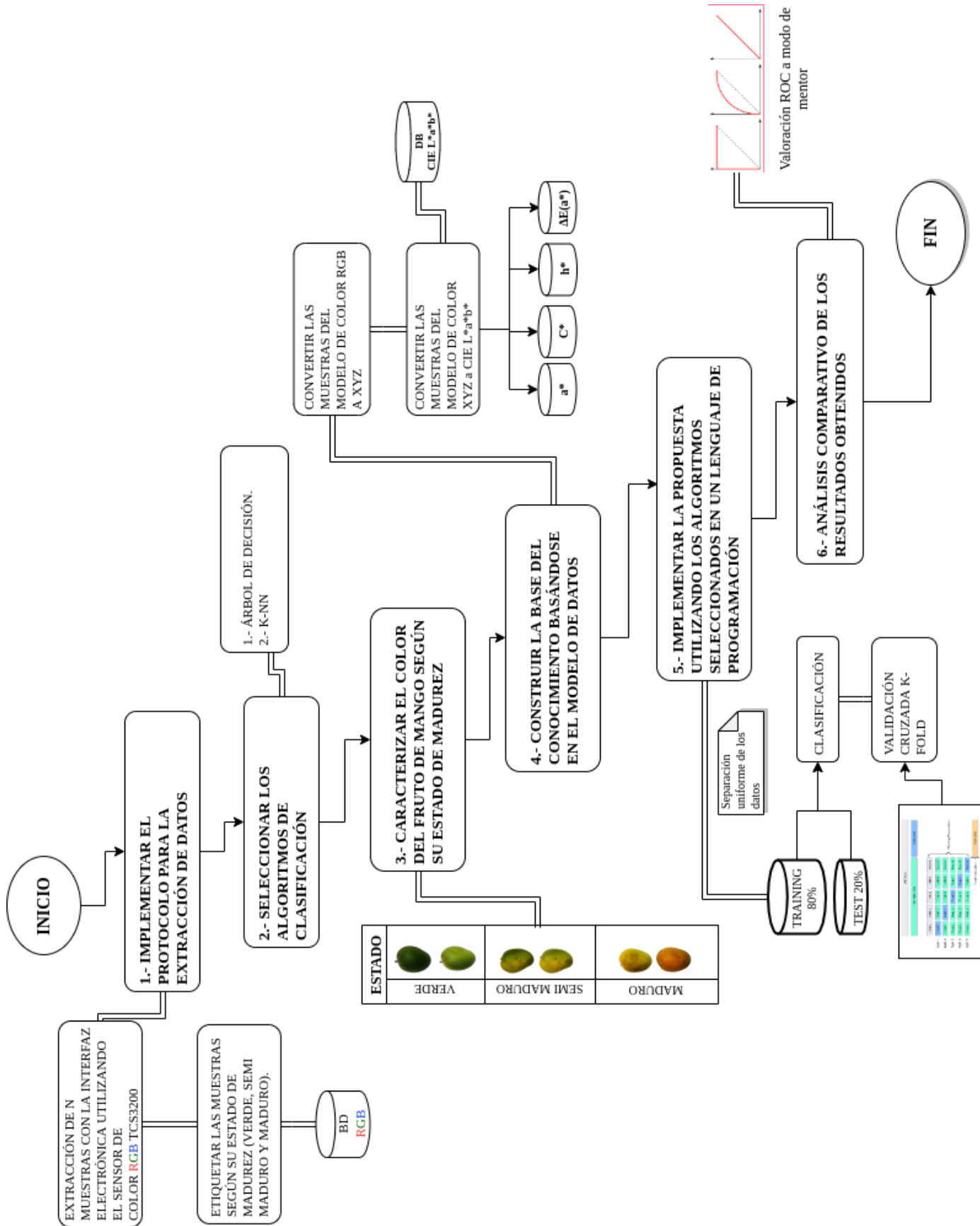


Figura 3.7: Método propuesto para llevar a cabo la investigación planteada Fuente: Elaboración propia



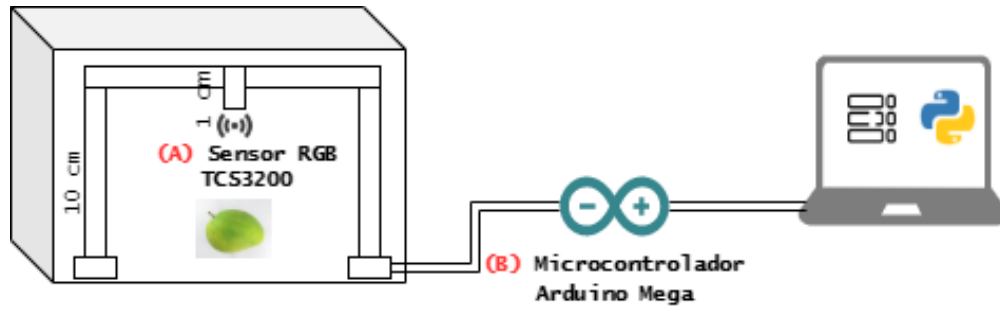


Figura 3.8: Esquema de componentes del protocolo de extracción de datos. Fuente: Elaboración propia

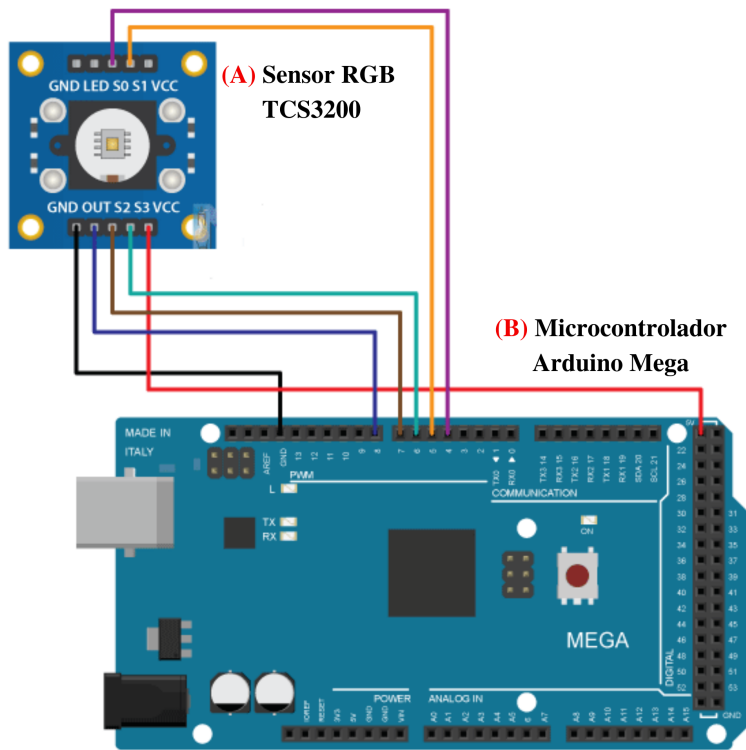


Figura 3.9: Esquema del circuito de conexión entre el sensor RGB TCS3200 y el microcontrolador arduino mema. Fuente: Elaboración propia





Figura 3.10: Prototipo electrónico para la extracción de datos implementado en un ambiente real.
Fuente: Propia

Se debe resaltar que la información cruda expuesta en el serial arduino se transfiere al lenguaje de programación *Python* para su procesamiento haciendo uso de la librería *Pyserial 3.4* la cual permite realizar una conexión al puerto serial 9600 y poder llevar a cabo esta tarea, para ello se emplea el algoritmo 3.1 el cual básicamente escucha todas las entradas que se generen en el puerto serial y las va capturando con el fin de poder almacenarlo en un archivo de texto.

```
# import serial
puerto_serial = '/dev/ttyACM0';
velocidad_transmision = 9600;
ruta_archivo = "datos-crudos.txt";
archivo_salida = open(ruta_archivo, "w+");
ser = serial.Serial(puerto_serial, velocidad_transmision)
while True:
    linea = ser.readline();
    linea = linea.decode("utf-8") #devuelve un binario, convertir a cadena
    print(linea);
    archivo_salida.write(linea);
```

Código Fuente 3.1: Conexión python con arduino Fuente: Marzogh (2019)



3.3.2 Seleccionar los algoritmos de clasificación

La clasificación de frutas es un tema ampliamente explorado teniendo al procesamiento de imágenes RGB como la técnica más utilizada para explotar características significativas de la fruta tales como el color, forma, textura y el tamaño. Los sistemas convencionales de visión artificial se centran en el análisis de la calidad, la identificación de enfermedades y el análisis de daños se encuentran como las principales aplicaciones de la clasificación de frutas basada en la visión artificial. Investigaciones recientes han utilizado una variedad de modelos de aprendizaje automático, por ejemplo, k-NN, SVM, árboles de decisión, bosques aleatorios, fuzzy logic, RNA y sus variantes para realizar este propósito como lo evidencian las investigaciones de Akter and Rahman (2017); Deepa and Geethalakshmi (2013); Habib et al. (2018); Hao et al. (2016); Hussain Hassan and Nashat (2018); Lal et al. (2017); Mim et al. (2018); Naik and Patel (2017); Nandi et al. (2016); Pereira et al. (2018); Savakar (2012).

Teniendo en cuenta lo antes expuesto, la selección objetiva de los algoritmos supervisados de clasificación automatizada se lleva a cabo a partir del revisión del estado del arte en función estrictamente del rendimiento en cuanto al porcentaje de exactitud y el año en que se publicó la Fuente como lo evidencian la (Tabla 3.2 y la Figura 3.11), en tal sentido se concluye que la propuesta de investigación utilizara los algoritmos de clasificación supervisada **árboles de decisión y k-NN** por la naturaleza aritmética de os datos con los que se va a trabajar.



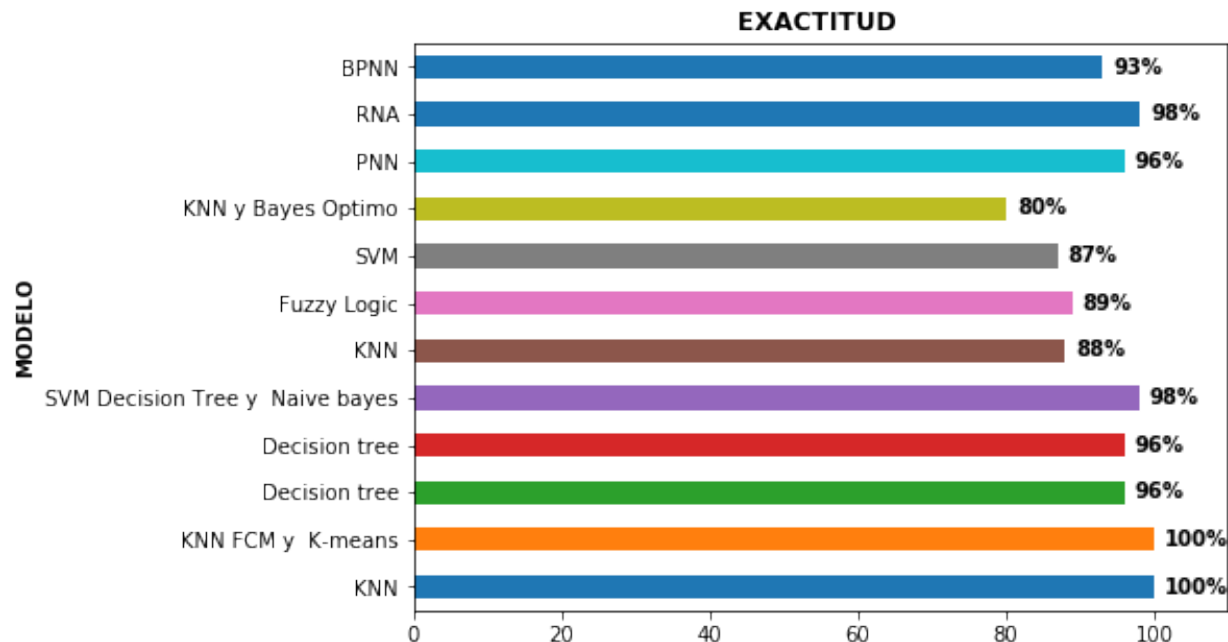


Figura 3.11: Selección de los algoritmos de clasificación. Fuente: Elaboración propia

Tabla 3.2: Análisis de rendimiento de los algoritmos de clasificación. Fuente: Elaboración propia




Año	Fruta	Tamaño DataSet	Característica	Clasificador	Exactitud	Ref.
2018	Fresa	2969	Color	KNN, FCM, K-means	100%	Hussain Hassan and Nashat (2018)
2018	Papaya	114	Color	Decision tree	95,98%	Pereira et al. (2018)
2018	Mango	100	Color	Decision tree	96%	Mim et al. (2018)
2018	Papaya	129	Color	SVM, Decision Tree, Naive bayes	90%	Habib et al. (2018)
2017	Berenjena	50	Color	KNN	88%	Akter and Rahman (2017)
2017	Mango	250	Color	Fuzzy Logic	89%	Naik and Patel (2017)
2016	Mango	200	Color	SVM	87%	Nandi et al. (2016)
2013	Naranja	200	Textura, Forma e Intensidad	PNN	96%	Deepa and Geethalakshmi (2013)
2013	Manzana	150	Color	RNA	98,10%	Lal et al. (2017)
2012	Manzana, Chikoo, Naranja, Mango y Limón Dulce	5000	Color, Textura	BPNN	93%	Savakar (2012)



3.3.3 Caracterizar el color del fruto de mango según su estado de madurez

La clasificación de los estados de madurez del fruto de mango se puede diferenciar en seis etapas de acuerdo a la clasificación estándar USDA y estas son etapa 1 (Dark olive green), etapa 2 (Olive), etapa 3 (Apple green), etapa 4 (Brown), etapa 5 (Saddle brown) y etapa 6 (Dark golden rod) Mim et al. (2018); USDA (2006). Para el presente proyecto se propone agrupar estas seis etapas en 3 clases quedando de la siguiente manera etapa 1 y 2 (Verde), etapa 3 y 4 (Semi Maduro) y etapa 5 y 6 (Maduro) como se evidencia en la (Tabla 3.3), según la teoría del color este se puede diferenciar de acuerdo al círculo cromático del color el cual representa las tonalidades del color de acuerdo a cada ángulo del círculo. Tomando esta referencia se clasifica los valores mínimos y máximos por clase conforme al modelo de color aditivo RGB modelando el color que más se asemeja al estado de madurez del fruto mango.

Tabla 3.3: Caracterización de color de fruto de mango según su estado de madurez. Fuente: Elaboración propia basado en el estándar propuesto por USDA Mim et al. (2018); USDA (2006)

CLASE	ÍNDICE DE COLOR	CLASE	RANGO MODELO DE COLOR RGB						DESCRIPCIÓN
			Rojo		Verde		Azul		
			Min	Max	Min	Max	Min	Max	
	Verde Opaco Verde Claro	Verde	0	30.35	57.39	99.97	43.52	100	Esta etapa de representa el periodo inicial de maduración del fruto mango el cual oscila entre el color verde opaco oscuro hasta un color verde claro. Por otra parte. USDA representa a esta clase fluctúa entre el color verde oliva oscuro y oliva.
	Amarillo Verdoso Verde Amarillento	Semi Maduro	28.77	64.63	27.61	69.03	20.8	58.75	En esta etapa de maduración del fruto representa a los índices de color amarillo verdoso tendiendo a un color verde amarillento. Por otra parte, USDA representa a esta clase oscilando entre el color verde manzana y marrón.
	Amarillo Claro Amarillo Naranja	Maduro	50.97	100	0	35.30	0	34.4	Esta etapa de maduración del fruto ya cuenta con una variación significativa en su color el cual oscila entre el color amarillo claro y amarillo naranja alcanzando el límite de maduración para su consumo. Por otra parte, USDA lo representa con los colores sillín marrón y varilla dorada oscura.



Conociendo la caracterización de las frutas por parte de USDA y el planteamiento del proyecto publicado Mim et al. (2018) en el cual proponen una segmentación del color RGB relacionada al estado de madurez del fruto de mango para generar reglas del algoritmo difuso; tomando esta iniciativa se plantea calcular la diferencia en los valores frontera para normalizar los límites de fusificación y generar rangos con una dispersión del color equilibrada utilizando el valor medio y de este modo reducir el ruido este procedimiento se ve reflejado en la Figura 3.12, en la cual se muestra los valores mínimos y máximos para cada componente del modelo de color RGB.

Rojo			Verde			Azul		
0	30,35		57,39	99,97	11,64	43,52	100	15,23
28,77	64,63	1,58	27,61	69,03	7,69	20,8	58,75	13,6
50,97	100	13,66	0	35,3		0	34,4	
		0,79			5,82			7,615
		6,83			3,845			6,8

	Verde	Semi Maduro	Maduro	RGB
0	31,14	31,15	57,8	Rojo
99,97	63,22	63,21	39,16	Verde
100	51,15	51,14	41,3	Azul

Figura 3.12: Análisis de la caracterización del color rgb. El cálculo de la media en los límites se calcula de la siguiente manera:

Rojo:
 $30.35 - 28.77 = 1.58$ $1.58/2 = 0.79$ $30.35 + 0.79 = 31.14.$
 $64.63 - 50.97 = 13.66$ $13.66/2 = 6.83$ $50.97 + 6.83 = 57.8.$

Verde:
 $69.03 - 57.39 = 11.64$ $11.64/2 = 5.83$ $57.39 + 5.83 = 63.22.$
 $35.3 - 27.61 = 7.69$ $7.69/2 = 3.845$ $35.3 + 3.845 = 39.15.$

Azul:
 $58.75 - 43.52 = 15.23$ $15.23/2 = 7.615$ $43.52 + 7.615 = 51.14.$
 $34.4 - 20.8 = 13.6$ $13.6/2 = 6.8$ $34.4 + 6.8 = 41.2.$

3.3.4 Construir la base del conocimiento basándose en el modelo de datos

El éxito o fracaso de los algoritmos de aprendizaje automático depende pura y exclusivamente de los datos de aprendizaje, es por ello que se debe limpiar la información para generar un



conocimiento puro y mejorar base del mismo para el ello se inicia con el procedimiento de limpieza de los datos el cual se aborda problemas de distinta índole como valores nulos, penalización de variables, datos atípicos, etc. Conociendo claramente la importancia de este procedimiento el proyecto inicialmente extrae 20 valores RGB de un total de 150 muestras de frutos de mango Ataúlfo utilizando sensor de color RGB TCS3200 por lo tanto generan 3 componentes por muestra dada, estas conforman la base de datos cruda o inicial del proceso de clasificación la cual se evidencia en la (Tabla 5.1), si bien es cierto el modelo de color RGB representa una gama de aproximadamente 16 millones de colores esta no es suficiente para asemejarse a la sensibilidad del ojo humana el cual puede reconocer una gama casi ilimitada de colores y está comprobado que *“el sistema visual humano es más sensible a la luz (componente de luminancia (luma)) que al color (componente de crominancia (chroma))”* Gómez and Perea (2015), es por ello que los esfuerzos de la investigación se han dirigido a solucionar este inconveniente partiendo del modelo de color RGB y se han creado modelos de color tales como HSI, HSV, CMY, CMYK cada uno con un propósito específico, para el proyecto se utiliza el modelo de color CIE $L^*a^*b^*$ el cual se asemeja al sistema visión humano gracias a sus componentes luma(L^*) y croma (a^* , b^*) [1.3.4.2.2].

Para lograr lo antes mencionado, se convierte cada muestra de color RGB a XYZ ya que los modelos de color RGB y CIE $L^*a^*b^*$ no son estrictamente lineales [1.3.4.2.1] y finalmente se obtiene el modelo de color CIE $L^*a^*b^*$ utilizando el lenguaje de programación *Python 3.6.5* y la librería de color *colormath 3.0.0*. Las especificaciones del computador utilizado para este proyecto se muestra en la (Figura 3.4).



Tabla 3.4: Especificaciones del computador. Fuente: Elaboración propia

Título	Descripción
Compiler	GCC 4.8.2 20140120 (Red Hat 4.8.2-15)
System	Linux
Release	4.18.0-16-generic
Machine	x86_64
Intel64	Family
Processor	Intel64 Family 6 Model 61 Stepping 4, GenuineIntel
CPU cores	4
Interpreter	64 bit

A continuación, se muestra la fusión *rgbtolab* utilizada por el algoritmo 3.2 para llevar a cabo este procedimiento.

```
# Función rgbtolab que unitliza colormath para pasar del modelo de color
  RGB a L*a*b
# Parametro color es una lista de tres valores color_rgb = [R, G, B]
# Retorna una lista llamada color_lab = [L, a, b]
# is_upscaled escalado interno entre 0-1.
# https://python-colormath.readthedocs.io/en/latest/
# from colormath.color_objects import XYZColor, sRGBColor, LabColor
# from colormath.color_conversions import convert_color

def rgbtolab(color_rgb):
    color_lab = []
    rgb = sRGBColor(rgb_r = color_rgb[0], rgb_g =color_rgb[1], rgb_b =
color_rgb[2], is_upscaled=True)
    xyz = convert_color(rgb, XYZColor)
    lab = convert_color(xyz, LabColor)
    color_lab.append(round(lab.lab_l,2))
    color_lab.append(round(lab.lab_a,2))
    color_lab.append(round(lab.lab_b,2))
```



```

return color_lab

# Prueba de la funcion rgbtolab
color = [3, 84, 45]
lab = rgbtolab(color)
print(lab)
# Result : [30.76, -32.37, 16.71]

```

Código Fuente 3.2: Función rgbtolab Fuente: Elaboración propia

Se debe agregar que para convertir todas las muestras RGB utilizando la función *rgbtolab* [3.2] se utiliza la librería *pandas 0.23.0* para importar la base de datos RGB y la librería *numpy1.14.3* para el manejo de vectores y matrices de forma eficiente en cuanto a la administración de recursos, para llevar a cabo esta tarea se utiliza el Algoritmo 3.3 el cual básicamente recorre todas las filas de la base de datos RGB tomando cada muestra correspondiente a 3 columnas las cuales se envían a la función *rgbtolab* para ser convertidas y posteriormente se almacena los datos CIE L*a*b* en la base de datos *data_lab.csv* resultante.

```

# Importar la librería Pandas con su alias ps
# Importar la librería numpy para manejo de vectores y matrice
data_rgb = pd.read_csv("datasets/data_rgb.csv", sep=',')

# función get_dataset que recibe una dataset y una
# lista de indices a extraer indices[v1,v2,v3]
# Devuelve un dataframe con los indices solicitados
def get_sub_dataset(data,indices):
    return data[indices]

# Función que crea un dataframe a partir de un arreglo y un vector de
indices
def get_df_sub(array, indices):

```



```

df_sub = pd.DataFrame(
    array, columns=[indices[0], indices[1], indices[2]]
)
return df_sub

# Bloque de código para generar la base de datos resultante con los
# valores L*a*b* correspondientes a cada muestra haciendo uso de los
# métodos get_sub_dataset, rgbtolab y get_df_sub
i = 1
frames = []
while i < 61:
    v1 = 'v'+repr(i)
    v2 = 'v'+repr(i+1)
    v3 = 'v'+repr(i+2)
    indices = [v1,v2,v3]
    data_colum = get_sub_dataset(data_rgb, indices)
    array = []
    for indice, fila in data_colum.iterrows():
        color_rgb = fila[:3]
        lab = rgbtolab(color_rgb)
        array.append(lab)

    df = get_df_sub(array, indices)
    frames.append(df)
    i+=3
df_result_lab = pd.concat(frames, axis=1, sort=False)
estados = eval(['[' + "'verde',"*50 + "'semi maduro',"*50 +
"'maduro',"*50 + ']'])
df_resultL['Estado'] = estados
df_result_lab.to_csv('datasets/data_lab.csv',
                    index=False,
                    header=True, sep='\t',
                    decimal='.')

```



`df_result_lab.head()`

Código Fuente 3.3: Algoritmo para la conversión de todas las muestras RGB a CIE L*a*b* Fuente: Elaboración propia

El resultado de la ejecución del (Algoritmo 3.3) genera una base de datos llamada *data_lab.csv* la cual se muestra en la Tabla 3.5 de forma comprimida ya que es una matriz de tamaño 150*61.

Tabla 3.5: Base de datos CIE L*a*b*. Fuente: Elaboración propia

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	...	v52	v53	v54	v55	v56	v57	v58	v59	v60	Estado
0	36.84	-23.31	-6.22	24.74	-17.1	-6.47	31.46	-21.72	0.29	29.28	...	33.54	-21.67	0.18	28.87	-18.7	-3.54	31.71	-19.62	-2.52	verde
1	25.61	-21.27	0.75	24.36	-17.47	-2.39	30.49	-23.23	3.96	37.29	...	30.99	-18.44	-4.87	36.89	-19.81	-8.6	35.84	-34.03	16.86	verde
2	28.68	-13.07	-11.57	32.34	-20.46	-5.41	26.38	-5.28	-17.01	26.12	...	35.65	-31.58	10.58	35.03	-19.07	-5.77	33.5	-26.28	10.11	verde
3	34.81	-23.98	-1.76	31.24	-23.05	-1.35	29.71	-18.05	-7.44	28.37	...	25.48	-14.69	-8.63	27.45	-12.92	-10.19	29.31	-18.71	-2.88	verde
4	32.87	-15.78	-12.19	33.17	-16.41	-13.01	34.08	-19.61	-7.21	29.53	...	24.58	-23.03	3.84	25.9	-22.74	3.79	27.68	-14.12	-10.51	verde

Posteriormente se hace uso de la base de datos *data_lab.csv* generada anteriormente y se crea los modelos de datos para el conocimiento utilizando los valores de tono, saturación, diferencia total del color (ΔE) componentes que han sido utilizados para el monitorio industrial de hormigones coloreados López and Di Sarli (2016) generando una gran distribución de las características del color como descriptores de un determinado producto y componente cromático del canal dominante (a^*) el mismo que ha sido utilizado con éxito en el trabajo presentado por Ankur et al. (2014), el cual utiliza este componente como descriptor de una región segmentada para el conocimiento de un modelo de aprendizaje automático para clasificar imágenes de frutos de mangos, para ello se recorre todas las filas de la base de datos *data_lab.csv* se extrae los valores de tono, saturación, ΔE y croma (a^*) utilizando las ecuaciones que se muestran en la Tabla 3.6. Para llevar a cabo esta tarea se utiliza el Algoritmo 3.4 el mismo que al ser ejecutado genera un vector de 20 elementos característicos y 1 elemento correspondiente a la etiqueta de cada clase respectiva a la fruta de mango por cada fila de la base de datos *data_lab.csv* a diferencia del procesamiento de la base de datos ΔE la cual extrae la diferencia total del color cada 2 muestras generando un vector de 10 valores descriptivos y 1 valor correspondiente a la etiqueta por cada fila.



Tabla 3.6: Resumen de los componentes para extracción de las bases de datos Fuente: Elaboración propia

Nombre BD	Componente	Ecuación	Descripción
data_lab_A.csv	croma (a*)	a*	Tamaño 150*21 con 20 valores correspondientes a la caracterización del color y 1 del estado por muestra.
data_lab_C.csv	Saturación (C)	$C^* = \sqrt{a^{*2} + b^{*2}}$	Tamaño 150*21 con 20 valores correspondientes pureza de cada color y 1 del estado por muestra.
data_lab_h.csv	Tono (h)	$h^* = \text{Arcotg}\left(\frac{b^*}{a^*}\right)$	Tamaño 150*21 con 20 valores correspondientes a la longitud de onda dominante del color o estado puro y 1 del estado por muestra.
data_delta_e_1976.csv	ΔE	$\Delta L^* = L_b^* - L_a^*$ $\Delta a^* = a_b^* - a_a^*$ $\Delta b^* = b_b^* - b_a^*$ $\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$	Tamaño 150*21 con 10 valores correspondientes a la diferencia total del color entre dos muestras CIE L*a*b* y 1 del estado por muestra.

```
# Base de Datos Croma (a*)
data_lab = pd.read_csv('datasets/data_lab.csv', sep='\t')

# Función que define los títulos de las columnas correspondientes a las
# muestras del componente croma(a*)
def definir_titulos(df, lis):
    return df.rename(columns={lis[0]: 'P1', lis[1]: 'P2', lis[2]: 'P3',
        lis[3]: 'P4', lis[4]: 'P5', lis[5]: 'P6', lis[6]: 'P7', lis[7]: 'P8',
        lis[8]: 'P9', lis[9]: 'P10', lis[10]: 'P11', lis[11]: 'P12',
        lis[12]: 'P13', lis[13]: 'P14', lis[14]: 'P15', lis[15]: 'P16',
        lis[16]: 'P17', lis[17]: 'P18', lis[18]: 'P19', lis[19]: 'P20'
    }, inplace=False)

# Valores del componente "a*"
lisA = ['v2', 'v5', 'v8', 'v11', 'v14', 'v17', 'v20', 'v23', 'v26', 'v29',
    'v32', 'v35', 'v38', 'v41', 'v44', 'v47', 'v50', 'v53', 'v56', 'v59']
```



```

df_resultA = pd.DataFrame(data_lab, columns=lisA)
df_resultA = definir_titulos(df_resultA, lisA)
df_resultA['Estado'] = data_lab['Estado']
df_resultA.to_csv('datasets/data_lab_A.csv', index=False, header=True, sep='
\t', decimal='.')
df_resultA.head()

# Base de Datos Tono y Saturación
def get_saturacion_lab(lab):
    ca = lab[1]
    cb = lab[2]
    return round(((ca**2 + cb**2)**0.5), 2)

# Test get_saturacion_lab(lab)
m_lab = [23.77, -10.94, -12.56]
saturacion_test = get_saturacion_lab(m_lab)
saturacion_test
# 16.66

def get_tono_lab(lab):
    import numpy as np
    ta = lab[1]
    tb = lab[2]
    if tb!=0:
        return round((np.arctan(ta/tb)), 2)
    else:
        return 0

# Test get_tono_lab(lab)
m_lab = [25.61, -21.27, 0.75]
tono_test = get_tono_lab(m_lab)
tono_test
# -1.54

```



```

import sys

# Función que calcula un vector de diferencias totales del color por column
as

def get_array_saturacion_or_tono(data_columns, ok):
    array = []
    for indice, fila in data_columns.iterrows():
        m_lab = fila[:3]
        if ok:
            lab_saturacion = get_saturacion_lab(m_lab)
            array.append(lab_saturacion)
        else:
            lab_tono = get_tono_lab(m_lab)
            array.append(lab_tono)

    return array

# Bloque de código para generar el dataset final con
# Delta E = diferencia total de color de todas las muestras
i = 1
k = 1
frames_C = []
frames_h = []
while i < 61:
    v1 = 'v'+repr(i)
    v2 = 'v'+repr(i+1)
    v3 = 'v'+repr(i+2)
    indices = [v1,v2,v3]
    data_columns =data_lab[indices]
    df_C= pd.DataFrame(get_array_saturacion_or_tono(data_columns, True),
    columns=['v'+repr(k)])
    df_h= pd.DataFrame(get_array_saturacion_or_tono(data_columns, False),
    columns=['v'+repr(k)])

```




```

k+=1
frames_C.append(df_C)
frames_h.append(df_h)
i+=3
df_result_C = pd.concat(frames_C, axis=1, sort=False)
df_result_h = pd.concat(frames_h, axis=1, sort=False)

df_result_C['Estado'] = data_lab['Estado']
df_result_h['Estado'] = data_lab['Estado']

df_result_C.to_csv('datasets/data_lab_C.csv', index=False,
                  header=True, sep='\t', decimal='.')
df_result_h.to_csv('datasets/data_lab_h.csv', index=False,
                  header=True, sep='\t', decimal='.')

# Base de Datos Diferencia Total del Color

# Funcion que calcula la diferencia total del color a partir de las listas
lab_a y lab_b
def get_diferencia_color(lab_a, lab_b):
    from colormath.color_objects import LabColor
    from colormath.color_diff import delta_e_cie1976, delta_e_cie2000

    color1 = LabColor(lab_a[0], lab_a[1], lab_a[2])
    color2 = LabColor(lab_b[0], lab_b[1], lab_b[2])
    delta_e_1976 = delta_e_cie1976(color1, color2)
    return round(delta_e_1976, 2)

# Test get_diferencia_color(lab_a, lab_b)
lab_a = [23.77, -10.94, -12.56]
lab_b = [31.68, -15.41, -12.12]

resul_l = get_diferencia_color(lab_a, lab_b)

```



```

print("{0:.2f}".format(resul_1))

# 9.10

# Funcion que calcula un vector de diferencias totales del color por column
as
def get_array_deltaE(data_columns):
    array = []
    for indice, fila in data_columns.iterrows():
        lab_a = fila[:3]
        lab_b = fila[3:]
        deltaE = get_diferencia_color(lab_a, lab_b)
        array.append(deltaE)
    return array

# Bloque de codigo para generar el dataset final con
# Delta E = diferencia total de color de todas las muestras
i = 1
k = 1
frames = []
while i < 61:
    v1 = 'v'+repr(i)
    v2 = 'v'+repr(i+1)
    v3 = 'v'+repr(i+2)
    v4 = 'v'+repr(i+3)
    v5 = 'v'+repr(i+4)
    v6 = 'v'+repr(i+5)
    indices = [v1,v2,v3,v4,v5,v6]
    data_columns =data_lab[indices]
    df= pd.DataFrame(get_array_deltaE(data_columns),columns=['v'+repr(k)])
    k+=1
    frames.append(df)
    i+=6
df_result_dt = pd.concat(frames, axis=1, sort=False)

```



```
df_result_dt['Estado']= data_lab['Estado']
df_result_dt.to_csv('datasets/data_delta_e_cie1976.csv',
                    index=False, header=True,
                    sep='\t', decimal='.')
df_result_dt.head()
```

Código Fuente 3.4: Extracción de las base de datos *Tono, Saturación, Canal dominante (a*) y Diferencia total de color ΔE* . Fuente: Elaboración propia

3.3.5 Implementar la propuesta utilizando los algoritmos seleccionados en un lenguaje de programación

El desarrollo de la propuesta utiliza el lenguaje de programación *Python 3.6.5* el cual es el lenguaje por excelencia para desarrollar modelos de ciencia de datos de la mano con el lenguaje de programación R, así mismo las librerías a utilizar son *sklearn 0.19.1, matplotlib 2.2.2, plotly 3.6, numpy 1.14.3, pandas 0.23.0, graphviz 2.38* y *PyQt5* tanto para el algoritmo supervisado de clasificación árbol de decisión así como para k-NN.

3.3.5.1 Desarrollo de la propuesta utilizando árbol de decisión y k-NN

Es primer paso para llevar a cabo el desarrollo de la propuesta es analizar la distribución de los datos para tener una idea de cuan agrupados se encuentran ya que los modelos de ciencia de datos utilizan la distancia como característica principal para clusterizar la información según ciertas clases y esto se logra realizando la técnica de análisis de componentes principales la cual se refleja en las Figuras [3.13][3.14][3.15][3.16], en donde se expone el resultado de aplicar dicha técnica a los conjuntos de datos mencionados en [3.6] a modo análisis. Asimismo, valiéndose de la librería *plotly* se gráfica el resultado de esta técnica para visualizar la correlación de los datos en términos de agrupamiento (ver Figuras [3.17][3.18][3.19][3.20]), reflejando una distribución centralizada de acuerdo a los estados de madurez bastante acertados para las bases de datos del canal dominante croma (a*) y saturación (C), no siendo así para el conjunto de datos correspondiente al tono (h) y la diferencia total del color (ΔE) las cuales demuestran una baja homogeneidad de los datos.



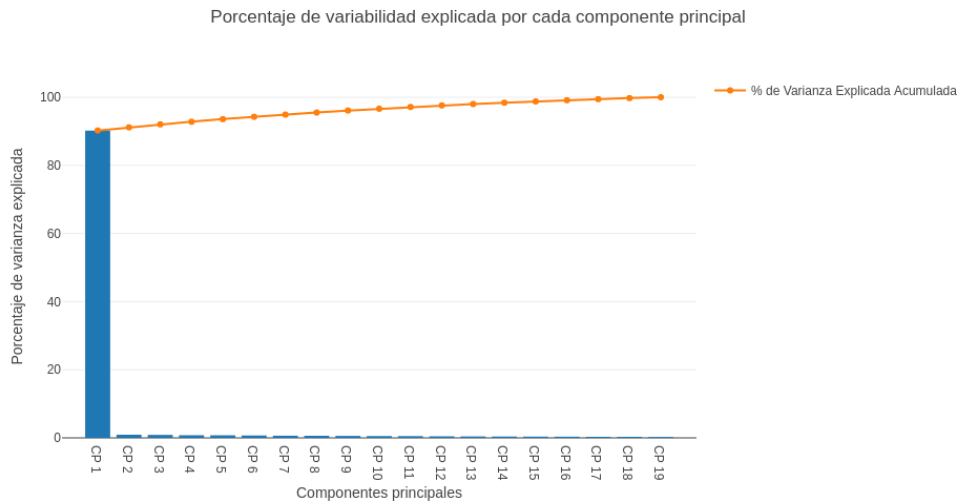


Figura 3.13: Visualización del porcentaje descriptivo de las variables utilizado datos Croma(a*). Fuente: Elaboración propia

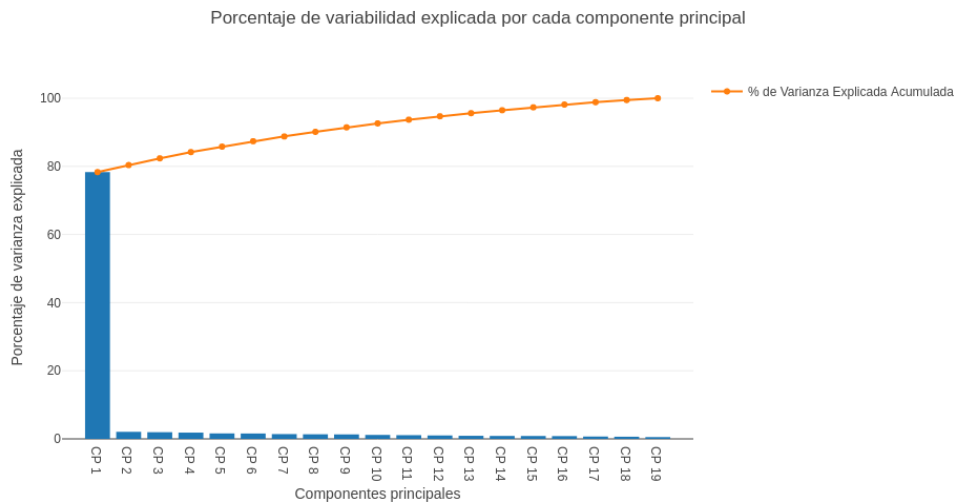


Figura 3.14: Visualización del porcentaje descriptivo de las variables utilizado datos Saturación(C*). Fuente: Elaboración propia



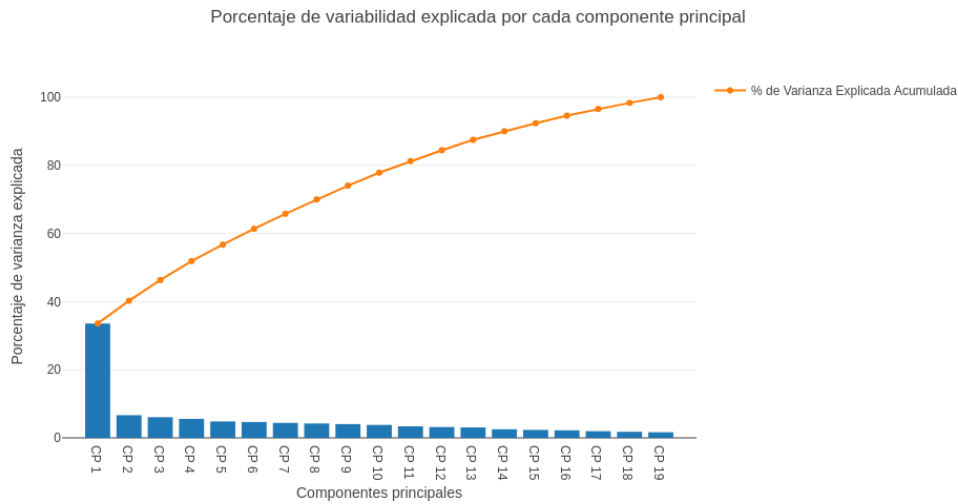


Figura 3.15: Visualización del porcentaje descriptivo de las variables utilizado datos Tono(h*). Fuente: Elaboración propia

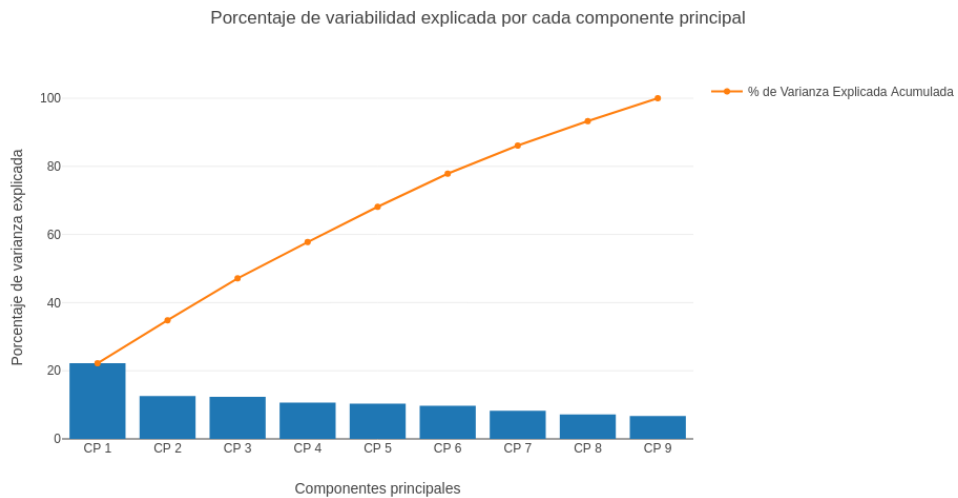


Figura 3.16: Visualización del porcentaje descriptivo de las variables utilizado datos Diferencia Total(ΔE) de (a^*). Fuente: Elaboración propia



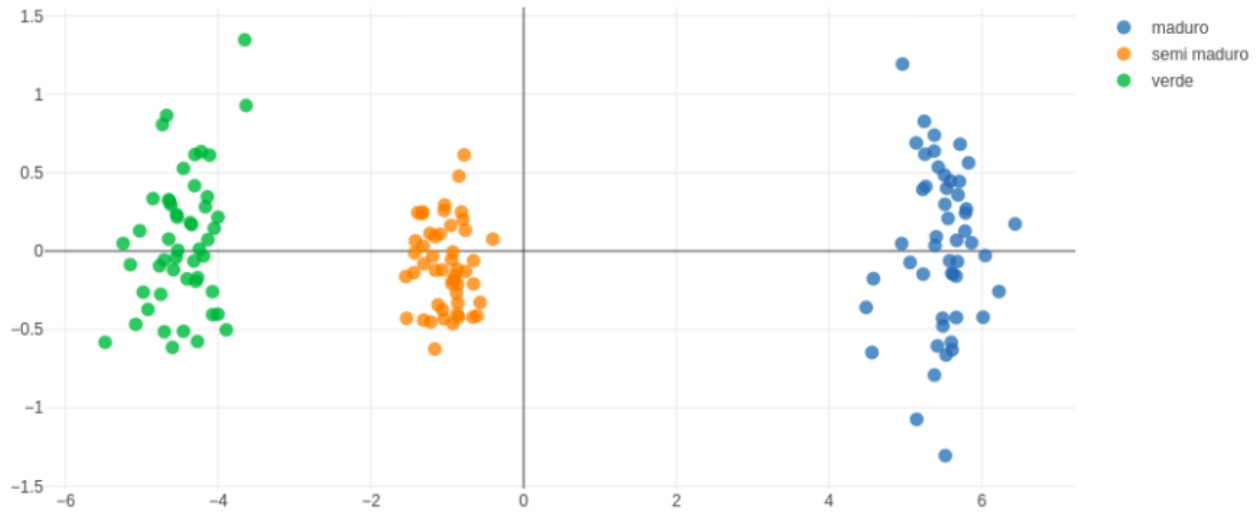


Figura 3.17: Agrupamiento del porcentaje descriptivo de las variables utilizado datos Crom(a*). Fuente: Elaboración propia

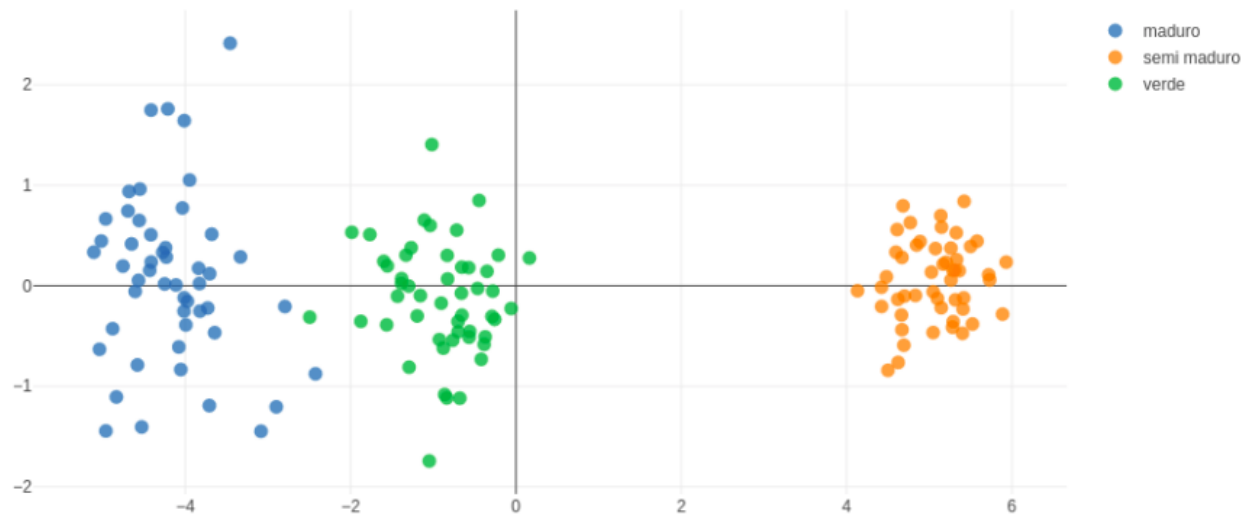


Figura 3.18: Agrupamiento del porcentaje descriptivo de las variables utilizado datos Saturación(C*). Fuente: Elaboración propia



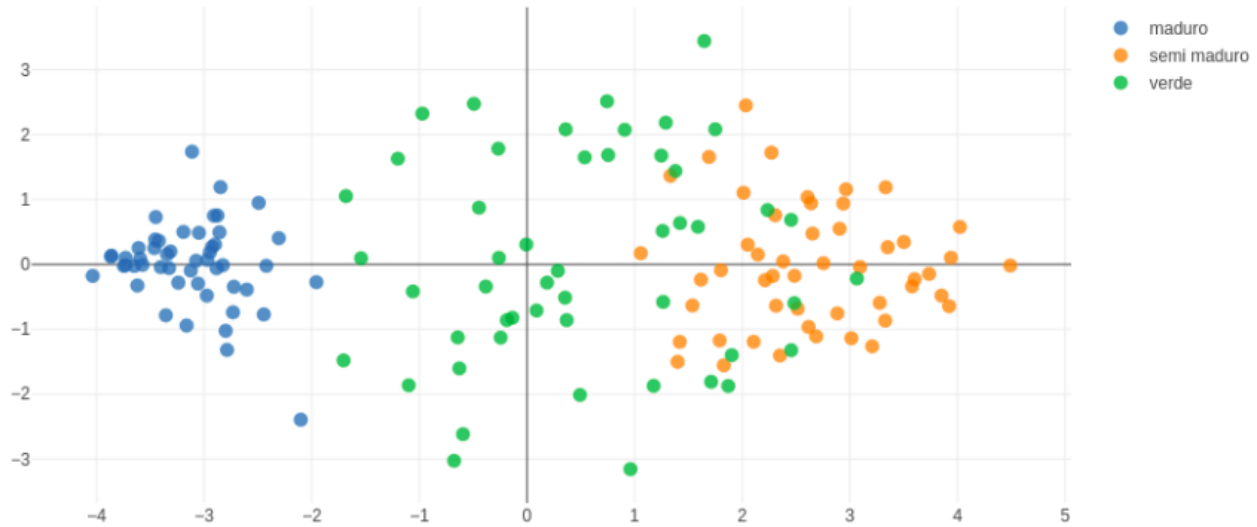


Figura 3.19: Agrupamiento del porcentaje descriptivo de las variables utilizado datos Tono(h*). Fuente: Elaboración propia

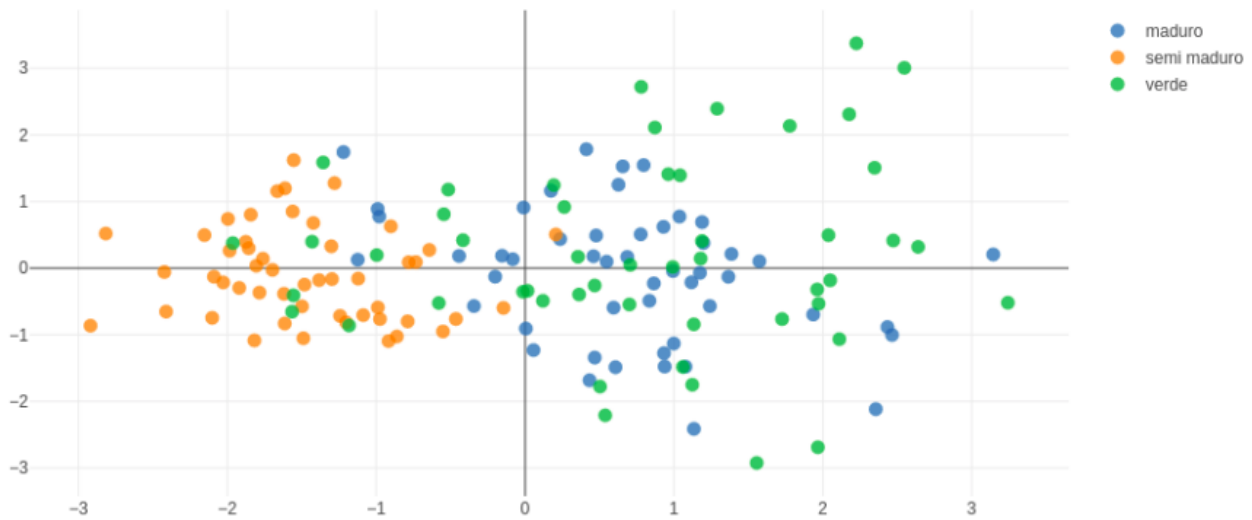


Figura 3.20: Agrupamiento del porcentaje descriptivo de las variables utilizado datos Diferencia Total(ΔE) de (a^*). Fuente: Elaboración propia

Teniendo una en cuenta lo antes expuesto se implementa el modelo para los algoritmos seleccionados [3.3.2]. Así mismo, para tener una mejor interacción se crea el diseño de la interfaz



gráfica utilizando la librería *PyQt5* la cual se muestra en la Figura 3.21 la misma que se describe a cabalidad en el anexo [5.2] en donde se puede configurar los parámetros principales de ambos algoritmos para ajustar los modelos y mejorar el rendimiento de clasificación.

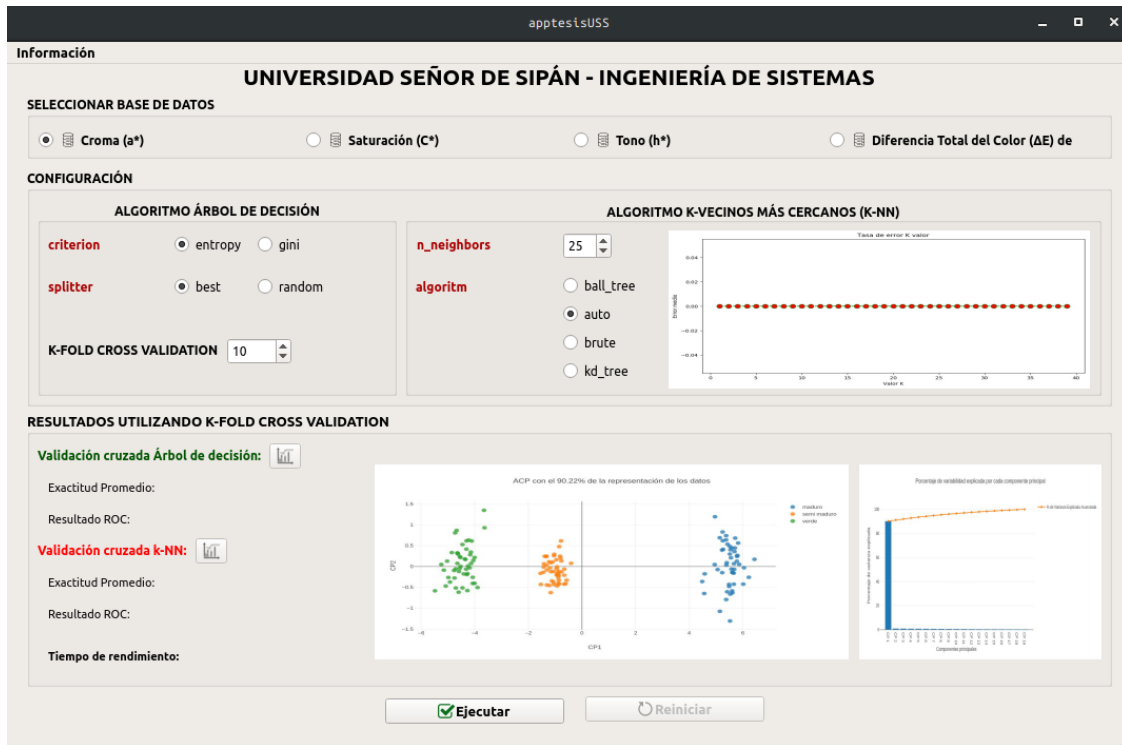


Figura 3.21: Interfaz gráfica del desarrollo de la propuesta. Fuente: Elaboración propia

A continuación, se especifica las secuencia de pasos más relevantes para llevar a cabo la implementación de los modelos planteados:

3.3.5.1.1 *Seleccionar la base de datos*

Haciendo uso de la librería *pandas* se implementó el método que devuelve la base de datos seleccionada en la interfaz gráfica, dicho método se muestra a continuación.

```
def getDB(self):
```




```

data = pd.read_csv('datasets/data_lab_A.csv', sep='\t')
if self.rb_bda.isChecked():
    data = pd.read_csv('datasets/data_lab_A.csv', sep='\t')
if self.rb_bdC.isChecked():
    data = pd.read_csv('datasets/data_lab_C.csv', sep='\t')
if self.rb_bdh.isChecked():
    data = pd.read_csv('datasets/data_lab_h.csv', sep='\t')
if self.rb_bddeltaE.isChecked():
    data = pd.read_csv('datasets/data_delta_e_cie1976.csv', sep='\t')
return data

```

Código Fuente 3.5: Selección de la base de datos. Fuente: Elaboración propia.

3.3.5.1.2 *Distribución uniforme de la información*

La distribución uniforme de los datos es un paso fundamental para implementar un modelo de aprendizaje automático ya que este segmenta la información de tal forma que la variable “X” contiene las columnas predictoras de la información y la variable “y” representa la columna objetivo, seguidamente se fracciona los datos dejando el 80% para el entrenamiento y el 20% para realizar las respectivas pruebas del modelo, a continuación se muestra el algoritmo que lleva a cabo este procedimiento, el mismo que valiéndose del método *rain_test_split* de la librerías *scikit-learn* el cual recibe los datos predictores, objetivo y el porcentaje de datos para la distribución en este caso 20%, el método segmenta la información en 4 partes (ver Figura 3.22) dejando ciertos porcentajes de los datos para el entrenamiento y las pruebas.

```

data = self.getDB()
columns = data.columns.values.tolist()
if self.rb_bddeltaE.isChecked():
    predictores = columns[:9]
    objetivo = columns[10]

```



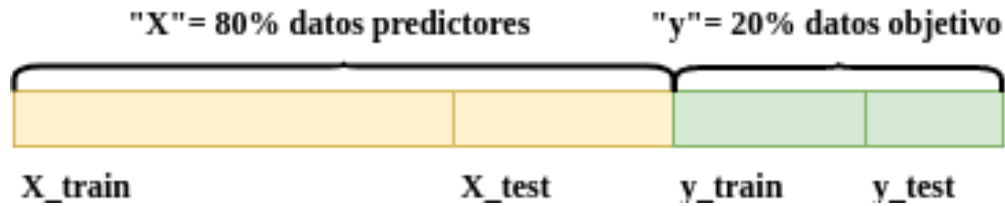


Figura 3.22: Segmentación uniforme de los datos. Fuente: Elaboración propia

```

else:
    predictores = columns[:19]
    objetivo = columns[20]
    X = data[predictores]
    y = data[objetivo]
    # Se separa los datos para entrenamiento 80% y prueba 20%
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
    
```

Código Fuente 3.6: Segmentación uniforme de los datos. Fuente: Elaboración propia

3.3.5.1.3 Creación y entrenamiento de los modelos

La creación de los modelos se da de acuerdo a la configuración establecida en la interfaz gráfica (ver Figura 3.21), la cual se describe a detalle en la Tabla 3.7.

Tabla 3.7: Configuración de los parámetros de los modelos de clasificación supervisada. Fuente: Elaboración propia

Árbol de Decisión		k-NN	
critério	entropy	n_neighbors	25
splitter	best	algorithm	auto

A continuación, se describe cada parámetro de los algoritmos descritos anteriormente para entender de forma teórica lo que representan.

a) Algoritmo árbol de decisión:



- **criterion:** Es la función para medir la calidad de una división. Los criterios admitidos son “gini” para la impureza de Gini y “entropy” para obtener información por defecto gini.
- **splitter:** La estrategia utilizada para elegir la división en cada nodo. Las estrategias admitidas son “best” para elegir la mejor división y “random” para elegir la división al azar.

Visualización de los árboles de decisión

Para realizar la tarea de visualizar los datos distribuidos en un árbol de decisión se utiliza la función de *export_graphviz* de la librería *Graphviz* la cual crea un archivo de extensión *.dot* que luego se convierte a un gráfico para visualizar los arboles de decisión utilizando los datos de las cuatro bases de datos [3.6], el resultado se muestra en las Figuras [3.23][3.24][3.25][3.26] las cuales describen un análisis de la distribución de los datos.

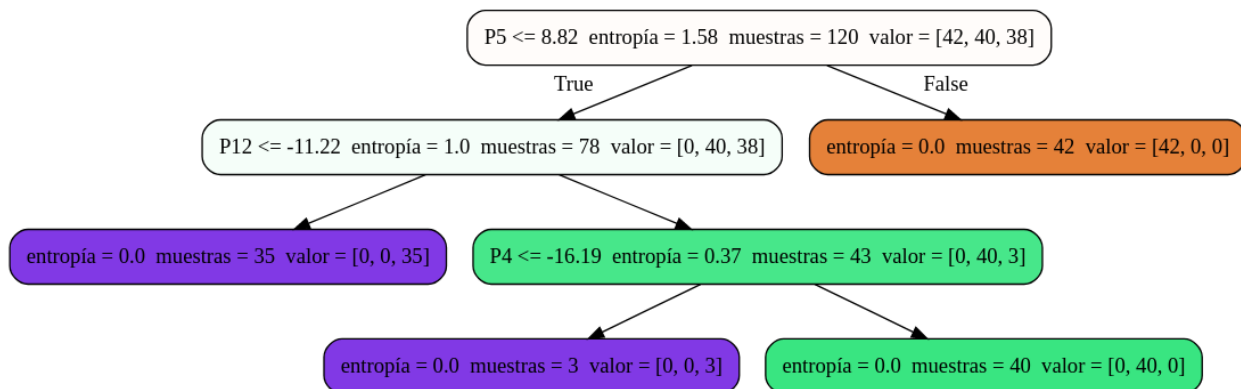


Figura 3.23: Visualización del árbol de decisión para los datos del canal dominante Croma(a*). Fuente: Elaboración propia

El análisis que se desprende de la refleja la Figura 3.23, es el de un árbol balanceado ya que distribuye los caminos dejando como resultados las hojas de color morado representando al estado **maduro**, el color cían representa al estado **semi maduro** y el color naranja representa el estado **verde**. Cabe resaltar que el éxito del balanceo de los datos correspondientes al canal cromático (a*) esta ligado estrictamente al resultado de la clusterización el cual se evidencia en la Figura 3.17.



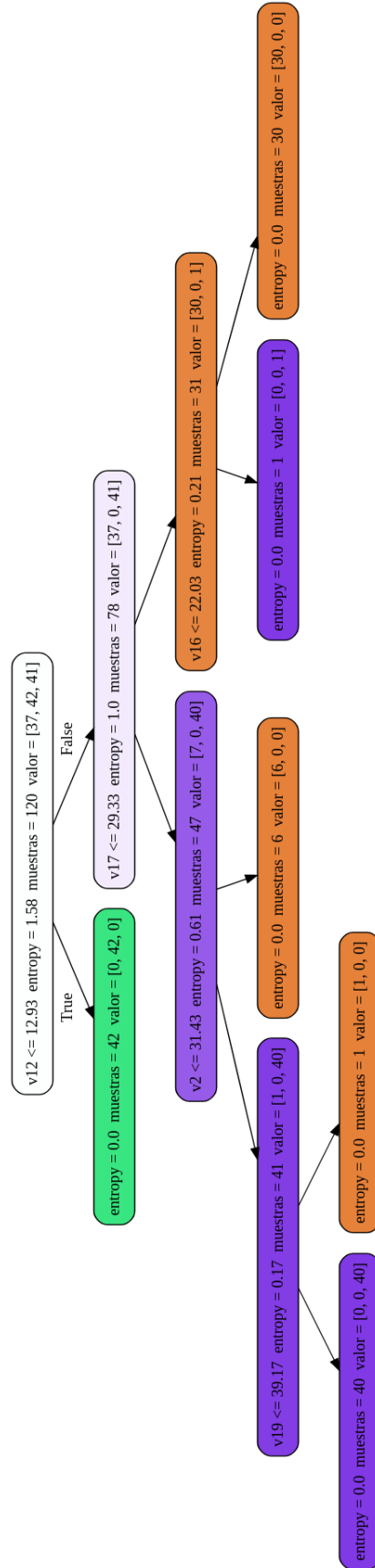


Figura 3.24: Visualización del árbol de decisión para los datos correspondientes a la pureza del color Saturación(C). Fuente: Elaboración propia

Análisis: La Figura 3.24 evidencia un árbol con mayor profundidad que el expuesto en la Figura 3.23 ya que es necesario crear más nodos intermedios para poder distribuir los caminos correctamente balanceados y de esta manera generar resultados fiables. El color morado representa el estado maduro y se puede apreciar que en los nodos intermedios existe ruido respecto a clasificaciones erróneas es por ello que el algoritmo utilizando la entropía para calcular la homogeneidad de los datos ve necesidad de crear un nivel más, lo mismo se observa en la distribución para los datos correspondientes los estados semi maduro y verde. Cabe mencionar que, la fase semi madura se distribuye de forma homogénea ya que solo es necesario un nivel para poder representar las 42 muestras del experimento no siendo así para el estado verde y maduro.



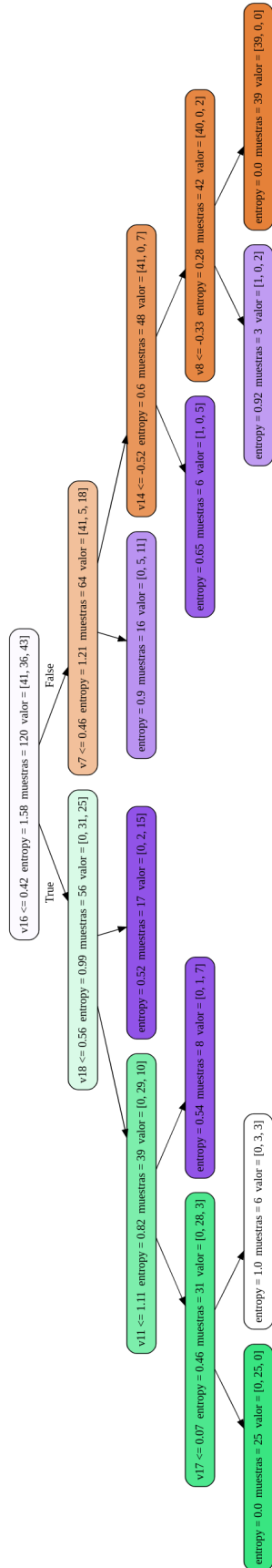


Figura 3.25: Visualización del árbol de decisión para los datos que representan al color dominante actual Tono(h). Fuente: Elaboración propia

Análisis: La Figura 3.25 evidencia un árbol con mayor profundidad que los expuestos en las Figuras [3.23][3.24] ya que es necesario crear mayor cantidad de nodos intermedios para poder distribuir los caminos correctamente balanceados y de esta manera generar resultados mucho más acertados. Lo que refleja este árbol de decisión el cual utiliza los datos propios del tono (h) es que no se puede agrupar los datos de forma 100% efectiva ya que el algoritmo se ve en la posición de resignar exactitud promedio y parar la creación nuevos niveles en pro del rendimiento en términos de eficacia. Se debe aclarar que, el color morado sigue representando el estado maduro y se puede apreciar que en los nodos intermedios existe ruido respecto a clasificaciones erróneas es por ello que se pinta el nodo con un color morado claro y así mientras mejor se ajuste el color va tendiendo al color morado puro, lo mismo se observa en la distribución para los datos correspondientes los estados semi maduro y verde. Se debe aclarar que el error de la clasificación se ve reflejado en el nodo de color blanco el cual representa a 3 muestras de la fase semi madura y 3 del maduro este error se da ya que el tono representa el color dominante actual y estas muestras de color corresponden a los límites de la caracterización



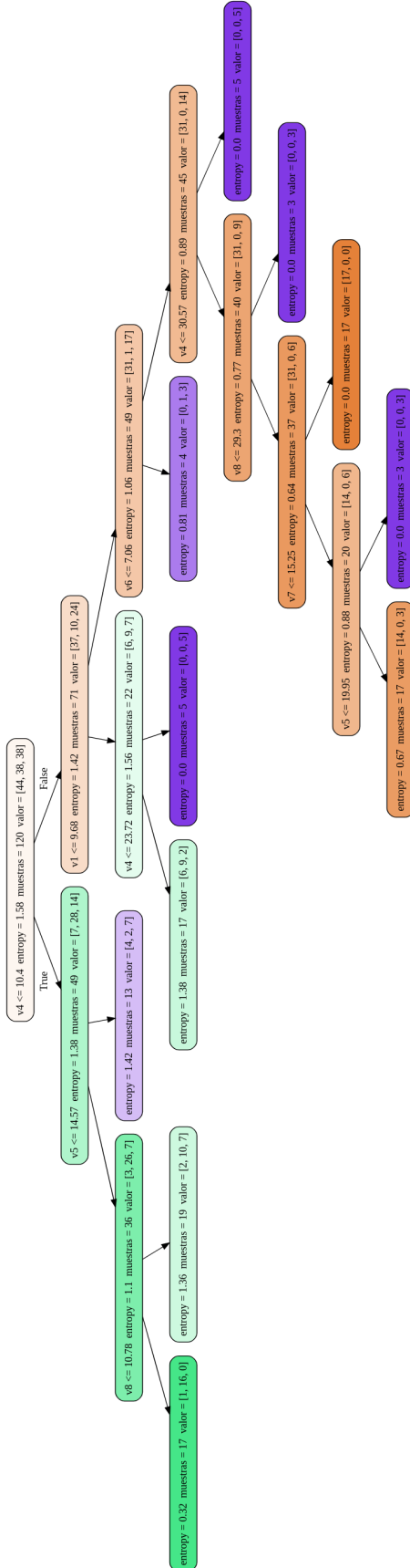


Figura 3.26: Visualización del árbol de decisión para los datos correspondientes a la diferencia total del color utilizando puntos de la base de datos del canal dominante Cromo(a*). Fuente: Elaboración propia

Análisis: La Figura 3.26 ya que es necesario crear mayor cantidad de nodos intermedios para poder distribuir los caminos de forma más balanceada posible y de esta manera generar resultados mucho más acertados. Lo que refleja este árbol de decisión el cual utiliza los datos propios de la diferencia total del color entre los puntos del canal dominante cromina (a*) es que no se puede agrupar los datos de forma 100% efectiva ya que el algoritmo se ve en la posición de resignar exactitud promedio y parar la creación nuevos niveles en pro del rendimiento en términos de eficacia. Se debe aclarar que, el color morado sigue representando el estado maduro y se puede apreciar que en los nodos intermedios existe ruido respecto a clasificaciones erróneas es por ello que se pinta el nodo con un color morado claro y así mientras mejor se ajuste el color va tendiendo al color morado puro, lo mismo se observa en la distribución para los datos correspondientes los estados semi maduro (cíjan) y verde (anaranjado). Se debe aclarar que el error de la clasificación se ve reflejado en los nodos hoja correspondientes al color cíjan claro y morado claro. Así mismo, se observa un árbol semi balanceado por los errores antes mencionados.



b) **Algoritmo k-vecinos más cercanos (k-NN)**

- (a) **n_neighbors:** Número de vecinos a utilizar por defecto para las consultas de los neighbors; por defecto es 5.
- (b) **algorithm:** Algoritmo utilizado para calcular los vecinos más cercanos:
 - * **ball_tree** Búsquedas rápidas de vecinos más cercanos.
 - * **kd_tree** Se utiliza para problemas rápidos generalizados de N puntos.
 - * **brute** Se utiliza para realizar una búsqueda de fuerza bruta.
 - * **auto** intentará decidir el algoritmo más apropiado en función de los valores pasados al método fit.

Finalmente teniendo la configuración establecida para cada modelo se lleva a cabo la implementación y el entrenamiento de la siguiente manera.

```
#Se crea el modelo de Árbol de decisión
tree = DecisionTreeClassifier(criterion="entropy", min_samples_split=20,
random_state=99)
tree.fit(X_train, y_train)
preds_arbol = tree.predict(X_test)

#Se crea el modelo de k-NN
instancia_knn = KNeighborsClassifier(n_neighbors=25)
instancia_knn.fit(X_train, y_train.values.ravel())
pred_knn = instancia_knn.predict(X_test)
```

Código Fuente 3.7: Creación y entrenamiento de los modelos. Fuente: Elaboración propia

Hay que mencionar, además que el parámetro **n_neighbors** del algoritmo de k-NN se configura en 25 de acuerdo análisis de la tasa de error para el valor K de acuerdo a cada ejecución según la



base de datos seleccionada (ver la Figura 3.27).

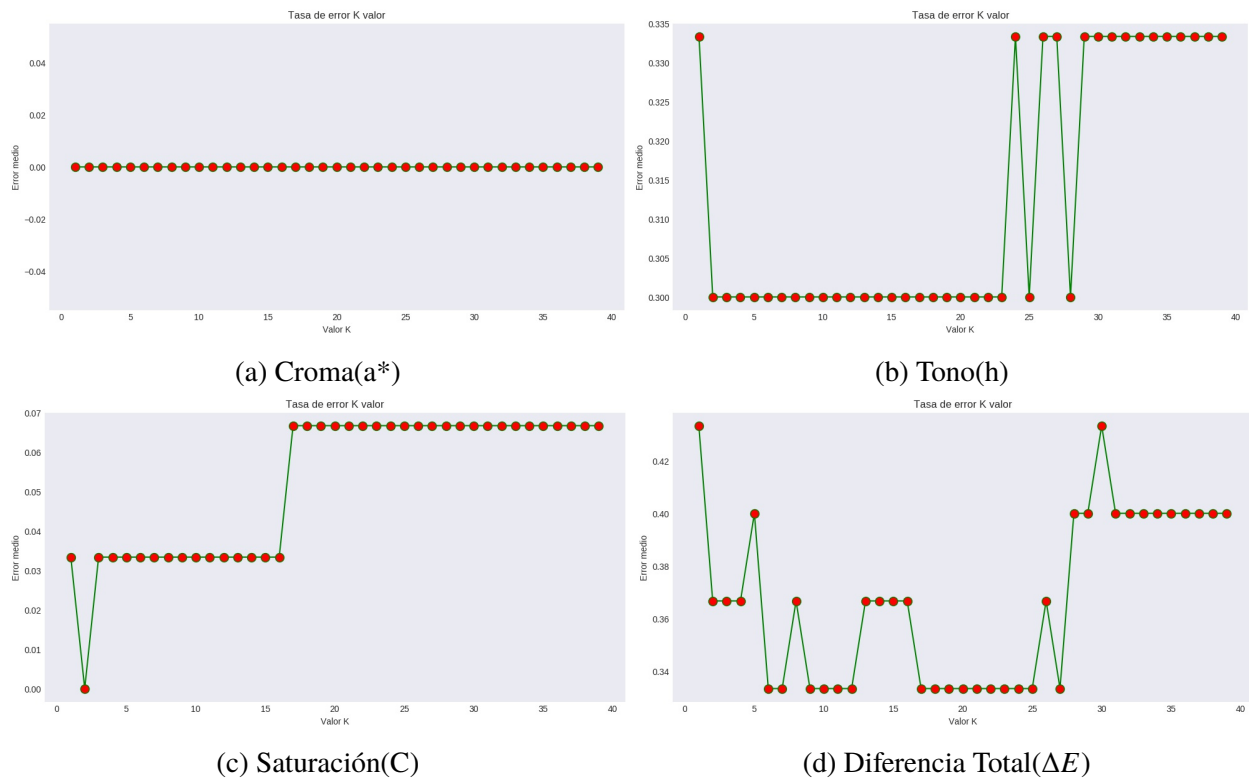


Figura 3.27: Análisis de la tasa de error para el valor K correspondiente k-NN Fuente: Elaboración propia

3.3.5.1.4 Validación cruzada k-fold

La validación de los modelos creados se realiza utilizando la técnica de validación cruzada k-fold [d], la cual se configura con el parámetro constante **k=10** ya que esta comprobado teóricamente que la eficiencia no varía significativamente si se adiciona un valor más. Se debe agregar que esta técnica da una valoración porcentual según los resultados obtenidos de acuerdo a la métrica de **exactitud**, el códigoFuente utilizado para realizar este trabajo se evidencia a continuación el cual hace uso del método *KFold* que recibe el parámetro *n_splits* con un valor extraído de la interfaz



GUI para separar los datos de entrenamiento y pruebas.

```

# Validación cruzada árbol de decisión
vc_arbol = KFold(n_splits=10, shuffle=True, random_state=1)
scores_arbol = cross_val_score(tree, X, y, scoring="accuracy",
cv=vc_arbol, n_jobs=1)

# Cálculo de la media de los 10 resultados obtenidos.
score_ad_F = np.mean(scores_arbol)

# Validación cruzada - k-NN
vc_knn = KFold(n_splits=10, shuffle=True, random_state=1)
scores_knn = cross_val_score(instancia_knn, X, y, scoring="accuracy",
cv=vc_knn, n_jobs=1)

# Calculo de la media de los 10 resultados obtenidos.
score_knn_F = np.mean(scores_knn)

```

Código Fuente 3.8: Validación de los resultados utilizando k-fold cross validation. Fuente: Elaboración propia

A continuación, se muestra los resultados de las ejecuciones para cada componente donde se describe de forma gráfica la exactitud resultante de la ejecución tanto del modelo árbol de decisión (verde) como k-NN (rojo) para las 10 interacciones.



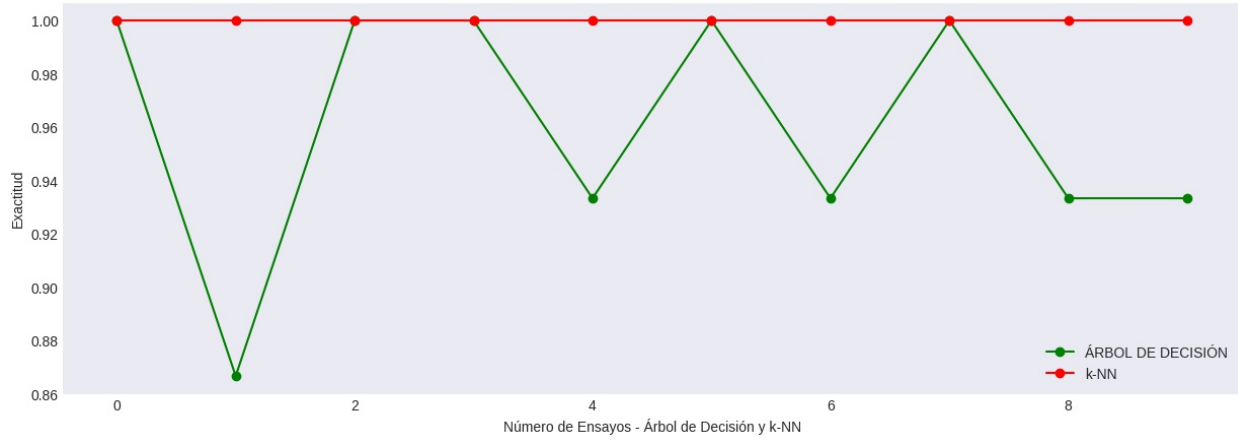


Figura 3.28: Resultado de la validación cruzada k-fold croma(a*) Fuente: Elaboración propia

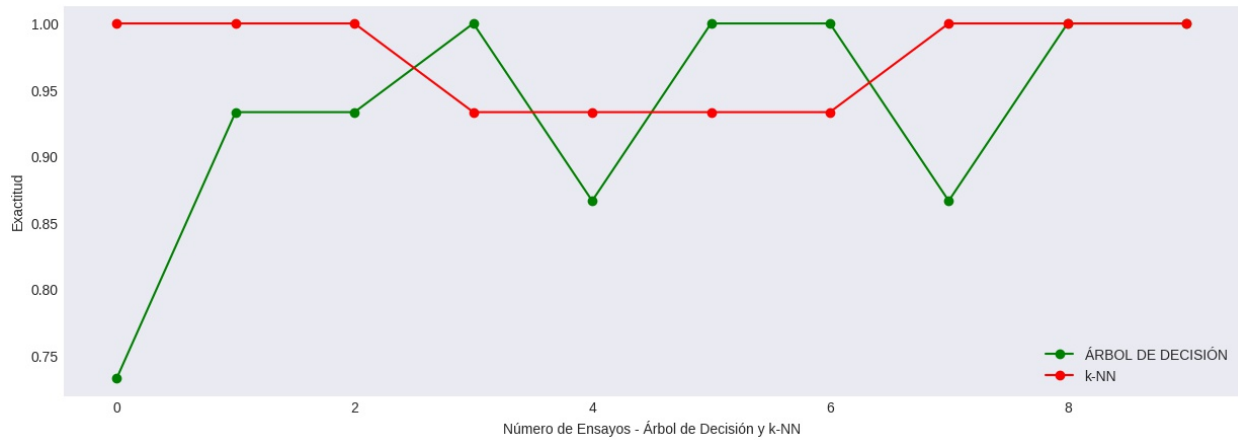


Figura 3.29: Resultado de la validación cruzada k-fold tono(h*) Fuente: Elaboración propia



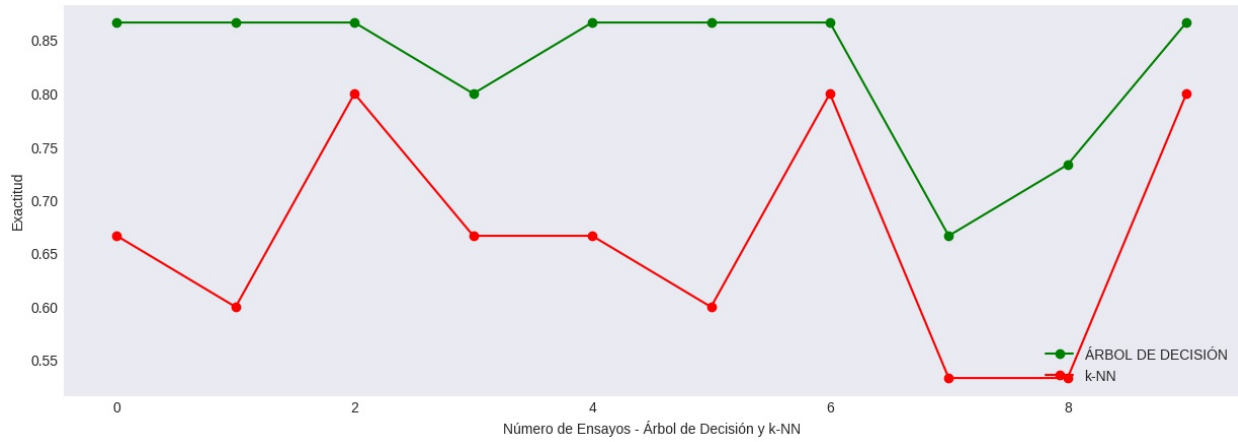


Figura 3.30: Resultado de la validación cruzada k-fold saturación(C*) Fuente: Elaboración propia

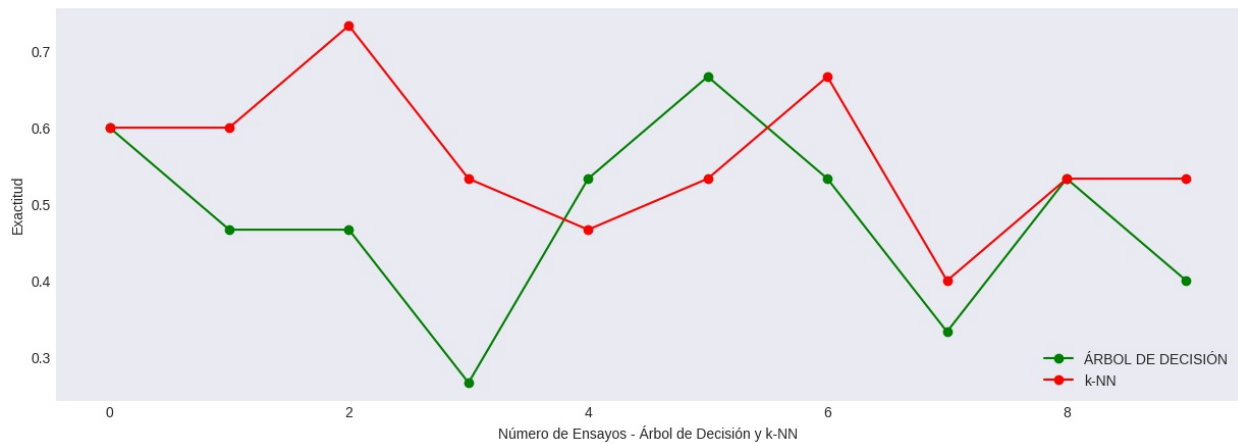


Figura 3.31: Resultado de la validación cruzada k-fold (ΔE a*) Fuente: Elaboración propia



4. CONCLUSIONES Y RECOMENDACIONES

Basado en los resultados expuestos en la propuesta de realizar la clasificación del fruto de mango según su estado se madurez utilizando un sensor de color RGB TCS3200 y los algoritmos supervisados de clasificación árbol de decisión y k-NN, se ha concluido:

4.1 Conclusiones

- a) La propuesta utiliza un sensor de color RGB TCS3200 el cual resultó ser un componente fiable para la extracción de información correspondiente al color de un fruto de mango según su estado de madurez.
- b) Para la selección objetiva de los algoritmos la propuesta hace estricto énfasis en el rendimiento de los resultados en investigaciones previas relacionadas a la clasificación automatizada y por conveniencia se seleccionó los algoritmos Árbol de decisión y k-NN.
- c) La caracterización de color según el estado madurez de la fruta de mango se basa en el estándar propuesto por USDA ajustando el modelo utilizando valores medios que representen la clase a la cual corresponde cada fruta, dicha determinación se ajusta muy bien al propósito antes mencionado.
- d) Para la propuesta se utilizó 4 conjunto de datos correspondientes a componente cromático a^* , saturación (C), tono (h^*) y la diferencia total del color (ΔE) del componente croma (a^*) ya que, al ser una propuesta cuasi experimental, esta debe sostenerse en experimentos previos revisados en los antecedentes de estudio.
- e) La implementación de la propuesta se realizado utilizando el lenguaje de programación Python el cual ha ido ganando terreno en la creación de aplicaciones relacionadas a la ciencia de datos, en cuanto a los resultados obtenidos se debe concluir que este cumple con todos los objetivos planteados haciendo uso de los algoritmos seleccionados.



- f) La ejecución de la propuesta evidencia que de los resultados obtenidos para el algoritmo Árbol de decisión es 94% 93% 76% 60% y para el algoritmo k-NN es 100% 97% 67% 48% utilizando las bases de datos componente cromático (a^*), saturación (C^*), tono (h^*) y la diferencia total del color (ΔE) de (a^*) respectivamente, en un tiempo que varía entre 0.64 y 0.74 milisegundos.
- g) Para evaluar los algoritmos de clasificación supervisada se utilizó los indicadores de exactitud y tiempo de rendimiento los resultados evidencian que utilizando datos correspondientes al componente croma (a^*) y saturación el algoritmo de k-NN obtiene una mejor performance todo lo contrario si se utiliza datos correspondientes al tono (h^*) y ΔE de (a^*) en donde se evidencia que el algoritmo Árbol de decisión obtiene resultados más favorables.

4.2 Recomendaciones

- a) La exactitud de los algoritmos de clasificación supervisada varía dependiendo el contexto y los datos con los que se implementen, por ello se recomienda no asumir que uno es mejor que otro sin antes ejecutar los modelos en un mismo contexto, se debería tomar dichos resultados como referencia para medir el rendimiento de los algoritmos, más no para concluir que uno es mejor o peor que otro.
- b) El sensor TCS 3200 varía dependiendo de la frecuencia con la que se trabaje, es por ello que se recomienda tomar esta investigación como referencia de una prueba netamente experimental. Así mismo, se recomienda utilizar el sensor TCS34725 ya que realiza un tratamiento digital de la medición de color sin utilizar filtros y podría mejorar la precisión de una muestra de color.
- c) Se recomienda utilizar los conjuntos de datos estandarizados con otros algoritmos de arquitecturas diferentes como bosques aleatorios o redes neuronales convolucionales. Cabe resaltar que, para crear modelos utilizando estos algoritmos se debe contar con un mayor volumen de información.
- d) Se recomienda la evaluación descriptiva para representar la clase putrefacta. Así mismo, se debería mejorar la propuesta para representar los valores límites de cada una de las clases de



forma más ecuánime posible.

- e) Se recomienda utilizar alguna tecnología alternativa como Machine Learning de Microsoft Azure para realizar un análisis comparativo de los resultados que se obtengan.
- f) Se recomienda validar los modelos utilizando otras técnicas como Leave One Out Cross Validation o Random Cross Validation para generar un informe alternativo y poder mejorar aún más la exactitud.



REFERENCIAS

- Adrian, F. and Roberts, A. (1998). *Colour space conversions*. Wmin. Accedido el 26-10-2018 a <https://goo.gl/nPB3xG>.
- Agrobanco (2007). *El Cultivo del Mango*. Banco Agropecuario, Área de Desarrollo.
- Akter, Y. A. and Rahman, M. O. (2017). Development of a computer vision based eggplant grading system. In *Proc. 4th Int. Conf. Advances in Electrical Engineering (ICAEE)*, pages 285–290.
- Ankur, M. V., Bijal, T., and Sapan, N. (2014). Quality inspection and classification of mangoes using color and size features. *International Journal of Computer Applications (0975 – 8887)*, 98:1–5.
- Arduino (2018). Arduino products. Accedido el 24-10-2018 a <https://goo.gl/UxVMTk>.
- Banco Mundial (2017). *Tomando impulso en la agricultura peruana: oportunidades para aumentar la productividad y mejorar la competitividad del sector*. Grupo Banco Mundial, Washington, D. C.
- BienManger (2018). Mango amélie. Accedido el 10-02-2018 a <https://goo.gl/ryhRhw>.
- Camino, S. A. (2018). Mango kent. Accedido el 10-02-2018 a <https://goo.gl/dEQ4Jd>.
- Caribbean, N. D. (2013). El mango: sus propiedades nutritivas y los beneficios para la salud. Accedido el 10-10-2018 a <https://goo.gl/DzLBAK>.
- Deepa, P. and Geethalakshmi, S. (2013). A Comparative Analysis of Feature Extraction Methods for Fruit Grading Classifications. *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, pages 221–225.
- DFrobot (2018). Sensor de color tcs3200 (sku: Sen0101). Accedido el 17-02-2018 a <https://goo.gl/uBPFiJ>.



- Domínguez, S. Q. and Peñalva, E. Q. (2003). *Photoshop 7 práctico*. McGraw-Hill España.
Recuperado de <https://ebookcentral.proquest.com>.
- Electropro (2017). Sensor de color tcs3200. Accedido el 17-02-2018 a <https://goo.gl/utW3Q4>.
- Gironés, J., Casas, J., Minguillón, J., and Caihuelas, R. (2017). *Minería de datos: modelos y algoritmos*. Editorial UOC.
- Gómez, A. and Perea, J. (2015). *Estimacion de Movimiento con codificacion Look-ahead en HEVC usando Programacion Paralela*. PhD thesis, Universidad del Valle, Calí, Colombia.
- Habib, M. T., Majumder, A., Jakaria, A., Akter, M., Uddin, M. S., and Ahmed, F. (2018). Machine vision based papaya disease recognition. *Journal of King Saud University-Computer and Information Sciences*.
- Hao, P., Cui, Y., Song, J., and Hu, Z. (2016). Color identification of complex objects based on the Bayes classification. In *Proc. IEEE Int. Conf. Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 304–309.
- Hussain Hassan, N. M. and Nashat, A. A. (2018). New effective techniques for automatic detection and classification of external olive fruits defects based on image processing techniques. *Multidimensional Systems and Signal Processing*, page 1.
- Ibraheem, N. A., Hasan, M. M., Khan, R. Z., and Mishra, P. K. (2012). Understanding color models : A review. *ARPJ Journal of Science and Technology*, pages 265–275.
- Interempresas Media, G. N. A. (2018). Mango keitt. Accedido el 10-02-2018 a <https://goo.gl/aqZ4fG>.
- Keith, J. (2005). Chapter 3: Color spaces. Accedido el 24-10-2018 a <https://goo.gl/LhbSdN>.
- Kumar, A. and Gill, G. S. (2015). Automatic fruit grading and classification system using computer vision: A review. In *Proc. Second Int. Conf. Advances in Computing and Communication Engineering*, pages 598–603.



- Lal, S., Behera, S. K., Sethy, P. K., and Rath, A. K. (2017). Identification and counting of mature apple fruit based on bp feed forward neural network. In *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, pages 361–368.
- López, A. and Di Sarli, A. R. (2016). El modelo CIELAB, las fórmulas de diferencia de color y el uso de la norma europea en 12878 en morteros y hormigones coloreados. *Ciencia y Tecnología de materiales*, 6.
- Malegori, C., Nascimento Marques, E. J., de Freitas, S. T., Pimentel, M. F., Pasquini, C., and Casiraghi, E. (2017). Comparing the analytical performances of Micro-NIR and FT-NIR spectrometers in the evaluation of acerola fruit quality, using PLS and SVM regression algorithms. *Talanta*, 165(November 2016):112–116.
- Martínez, E. L. (2008). Algoritmos para la detección y cuantificación de defectos en manzanas por inspección visual. Accedido el 07-01-2019 a <https://goo.gl/DuiyBS>.
- Marzogh (2019). Python script to read serial data from the arduino. Accedido el 03-04-2019 a <https://gist.github.com/Marzogh/723c137a402be7f06dfc1ba0b8517d09>.
- Mim, S., Smd, G., Fmd, H., and Jerin, A. (2018). Digital measurement of maturity indices of mangoes using selected image features. *cientia Horticulturae*, pages 156–163.
- MINAGRI (2017). *Mapa Interactivo del MINAGRI*, pages 196–208. Ministerio de Agricultura y Riego, Lima Perú.
- Minolta, K. (2014). Entendiendo el espacio de color cie l*a*b*. Accedido el 01-10-2018 a <https://goo.gl/gLn3Cz>.
- Mustaffa, I. B. and Khairul, S. F. B. M. (2017). Identification of fruit size and maturity through fruit images using opencv-python and raspberry pi. In *Proc. Automation and Sciences (ICORAS) 2017 Int. Conf. Robotics*, pages 1–3.



- Naik, S. and Patel, B. (2017). Thermal imaging with fuzzy classifier for maturity and size based non-destructive mango (*mangifera indica l.*) grading. In *Proc. Int. Conf. Emerging Trends Innovation in ICT (ICEI)*, pages 15–20.
- Nandi, C. S., Tudu, B., and Koley, C. (2016). A machine vision technique for grading of harvested mangoes based on maturity and quality. *IEEE sensors Journal*, 16(16):6387–6396.
- Pavithra, V., Pounroja, R., and Bama, B. S. (2015). Machine vision based automatic sorting of cherry tomatoes. In *Proc. 2nd Int. Conf. Electronics and Communication Systems (ICECS)*, pages 271–275.
- Peifeng, H., Yuzhe, C., Jingping, S., and Zhaomu, H. (2016). Color identification of complex objects based on the bayes classification. *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 304–309.
- Pereira, L. F. S., Barbon Jr, S., Valous, N. A., and Barbin, D. F. (2018). Predicting the ripening of papaya fruit with digital imaging and random forests. *Computers and Electronics in Agriculture*, 145:76–82.
- Perú, F. O. (2018a). Mango ataulfo. Accedido el 10-02-2018 a <https://goo.gl/eWH5dG>.
- Perú, F. O. (2018b). Mango haden. Accedido el 10-02-2018 a <https://goo.gl/GNe3Uu>.
- Perú, F. O. (2018c). Mango tommy atkins. Accedido el 10-02-2018 a <https://goo.gl/3zsGyv>.
- Pino, J. M. (2017). *Boletín Estadístico de Comercio Exterior Agrario*, page 33. Ministerio de Agricultura y Riego, Lima Perú.
- Pride, N. (2018). Nam dok mai. Accedido el 17-02-2018 a <https://goo.gl/bokZUx>.
- Romero, C. A. (2017). *Perfil técnico N° 4. Mango: Crecen exportaciones peruanas*, page 6. Ministerio de Agricultura y Riego, Lima Perú.



- Salunkhe, R. P. and Patil, A. A. (2015). Image processing for mango ripening stage detection: Rgb and hsv method. In *Proc. Third Int. Conf. Image Information Processing (ICIIP)*, pages 362–365.
- Savakar, D. (2012). Identification and classification of bulk fruits image using artificial neural networks. *International Journal of Engineering and Innovative Technology*, 1(3):36–40.
- Saúco, V. G. (2019). *El cultivo del mango (2a. ed.)*. Mundi-Prensa. Recuperado de <https://ebookcentral.proquest.com>.
- Sparkfun (2018). Arduino mega 2560 r3. Accedido el 24-10-2018 a <https://goo.gl/MV4XiN>.
- USDA (2006). Shipping Point and Market Inspection Instructions for Mangos. Technical Report July, Departamento de Agricultura de los Estados Unidos.
- Valenzuela, J. S. N. (2018). *Python Aplicaciones prácticas*. Editorial Ra-Ma, España.
- Vela, R. (2006). *Oficina General De Perfil De Mercado Del Mango*, page 21. Ministerio de Agricultura.
- Walmart (2018). Manila súper. Accedido el 10-02-2018 a <https://goo.gl/rydhXH>.
- Widiyanto, W. W., Purwanto, E., and Neighbor, K.-n. (2019). Classification of mango fruit quality based on texture characteristics of glcm (gray level co-occurrence matrices) with algorithm k-nn (k-Nearest Neighbors). *Techno (Jurnal Fakultas Teknik, Universitas Muhammadiyah Purwokerto)*, 20(1):1–10.
- Zhu, A., Yang, L., and Chen, Y. (2012). An FCM-based method to recognize and extract ripe tomato for harvesting robotic system. *IEEE International Conference on Automation and Logistics, ICAL*, pages 533–538.



5. ANEXOS

5.1 Código fuente arduino para interfaz electrónica

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

// Para guardar las frecuencias de los fotodiodos
int frequency, frequency_r, frequency_g, frequency_b = 0;
int count = 0;
String muestra, m = "";
void setup() {

    // Se definen las salidas
    pinMode(S0, OUTPUT);
    pinMode(S1, OUTPUT);
    pinMode(S2, OUTPUT);
    pinMode(S3, OUTPUT);

    // Se define la sensorOut como entrada
    pinMode(sensorOut, INPUT);

    // Se ajusta la escala de frecuencia al 20%
    digitalWrite(S0,HIGH);
    digitalWrite(S1,LOW);

    // Se inicia la comunicación serie
    Serial.begin(9600);
```



```

}
void loop() {

    // Se define la lectura de los fotodiodos con filtro rojo
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);
    // Lectura de la frecuencia de salida
    frequency = pulseIn(sensorOut, LOW);

    // Mapeando el valor de la frecuencia del ROJO (RED = R) de 0 a 255
    // Se coloca los valores mínimos y máximos de frecuencia de acuerdo a la
    distancia
    frequency_r = map(frequency, 0,100,255,0);

    // Se define la lectura de los fotodiodos con filtro verde
    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);
    // Lectura de la frecuencia de salida
    frequency = pulseIn(sensorOut, LOW);

    // Mapeando el valor de la frecuencia del VERDE (GREEN = G) de 0 a 255
    frequency_g = map(frequency, 41,99,255,0);

    // Se define la lectura de los fotodiodos con filtro azul
    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);
    // Lectura de la frecuencia de salida
    frequency = pulseIn(sensorOut, LOW);

    // Mapeando el valor de la frecuencia del BLUE (BLUE = B) de 0 a 255
    frequency_b = map(frequency, 0,100,255,0);

```



```

if(count < 20){
    m = String(String(frequency_r) + " "+String(frequency_g)+" "+String(
frequency_b));
    muestra += m;
    count++;
}else{
    Serial.println(muestra);
    delay(1000);
    count = 0;
    muestra = "";
}
}

```

Código Fuente 5.1: Para el sensor de color RGB TCS32000

5.2 Descripción de la interfaz gráfica

A continuación, se describe cada una de las opciones que nos ofrece la interfaz gráfica correspondiente

- **Seleccionar base de datos** Opción para poder realizar la selección de un conjunto de datos específico ya sea croma “a”, saturación, tono y diferencia total del color. Cabe resaltar que la base de datos seleccionada por defecto es croma “a”.
- **Configuración** Teniendo la base de datos seleccionada se deben configurar los parámetros de los algoritmos para ajustar los modelos teniendo en cuenta lo expuesto en la definición de los modelos [3.3.5.1.3]. Además, se debe configurar el parámetro **n_neighbors** del algoritmo k-NN según lo que muestra la tasa de error para **k** valor. Así mismo, la validación cruzada necesita un parámetro constante **K** el cual por conveniencia o teóricamente se inicializa en 10 cabe resaltar que la dimensión de este parámetro puede variar dependerá de los datos con los que se esté trabajando y los recursos del hardware donde se ejecuten.
- **Visualización de los resultados** El área de la interfaz destinada a la visualización de los



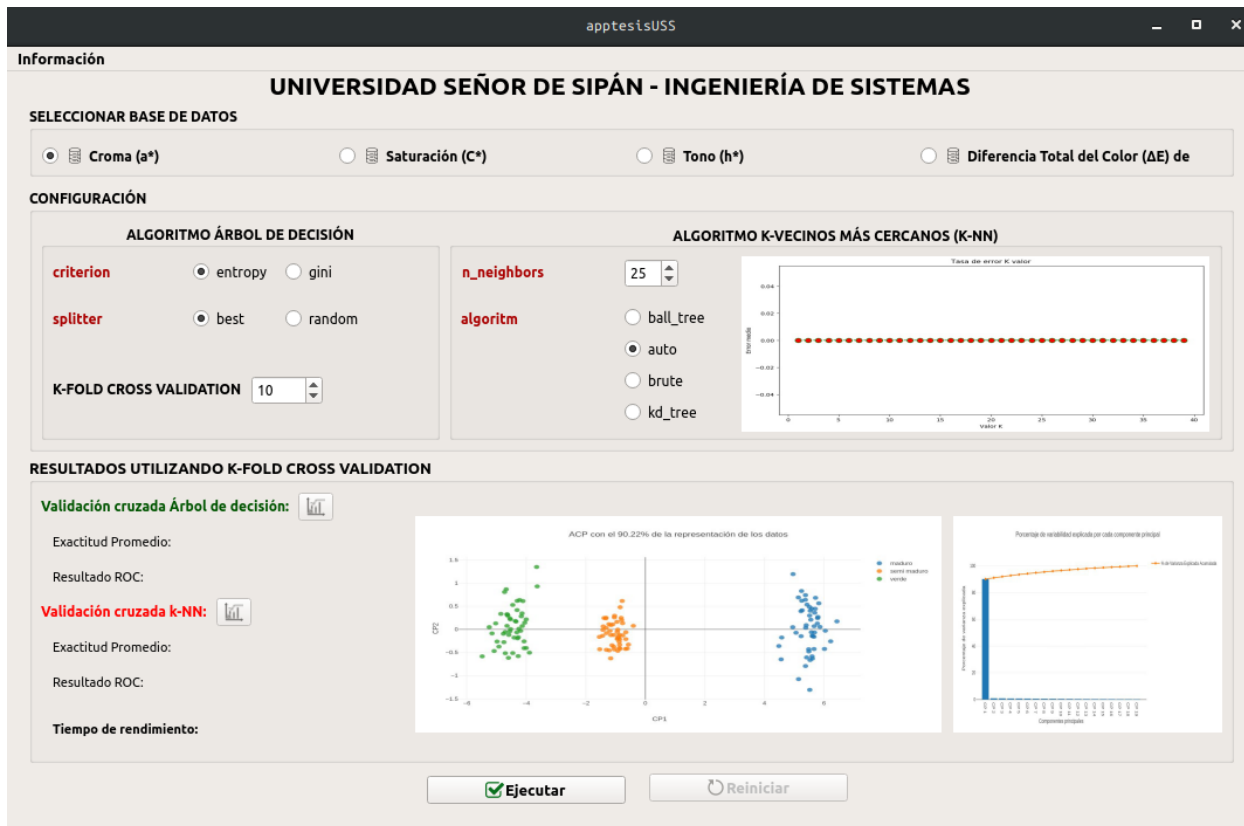



Figura 5.1: Diseño de la interfaz gráfica para la ejecución de los modelos
















resultados cuenta con las etiquetas donde se muestra la exactitud promedio de realizar la ejecución de la validación cruzada tanto para el algoritmo árbol de decisión y k-NN y las etiquetas para el resultado de la curva ROC la cual da un veredicto según el porcentaje resultante de exactitud (ver [d]). Así mismo, el gráfico de líneas donde se muestra los 10 resultados obtenidos por la técnica de la validación cruzada, además se muestra un gráfico correspondiente a la curva ROC y Finalmente se muestra los botones para ejecutar y reiniciar la ejecución de los modelos.

5.3 Base de datos RGB


















Tabla 5.1: Base de datos con valores RGB.

Nº	Valores	Estado	Img.
1	(9,97,96), (3,66,68), (23,83,73), (22,77,72), (6,72,70), (23,99,93), (15,77,53), (2,94,66), (29,80,79), (5,84,88), (8,74,63), (16,67,63), (12,76,78), (17,69,61), (0,77,98), (6,84,73), (29,92,87), (29,88,78), (19,76,73), (26,83,78)	verde	




















2	(7,69,59), (14,65,61), (22,81,65), (20,99,83), (27,87,88), (28,82,66), (29,99,98), (21,87,61), (3,71,78), (1,72,75), (27,78,99), (21,83,86), (20,99,94), (6,69,89), (11,80,92), (18,95,96), (0,69,94), (23,81,80), (24,96,100), (14,97,56)	verde	
3	(21,74,85), (15,85,84), (30,66,88), (15,66,93), (21,83,77), (16,70,69), (19,86,77), (4,92,87), (17,75,98), (30,93,58), (25,87,79), (20,83,52), (1,82,65), (31,86,55), (23,85,96), (3,69,95), (22,72,60), (11,96,66), (30,91,91), (30,89,62)	verde	
4	(16,92,84), (9,83,75), (12,78,81), (16,73,89), (16,84,79), (10,89,62), (24,64,83), (16,84,100), (2,77,74), (29,73,96), (15,84,88), (20,93,54), (0,70,79), (11,96,69), (12,71,82), (12,80,52), (17,81,93), (12,67,73), (22,71,80), (22,77,73)	verde	
5	(19,85,96), (12,86,98), (20,89,91), (22,79,56), (29,87,98), (0,72,100), (2,79,73), (1,96,68), (8,86,53), (18,78,74), (31,78,88), (21,80,62), (8,97,87), (20,75,80), (15,95,79), (14,98,89), (0,68,77), (3,67,52), (9,70,55), (16,72,81)	verde	
6	(20,80,57), (25,91,73), (28,69,80), (16,71,78), (1,74,80), (3,95,92), (2,86,54), (15,92,99), (3,84,57), (14,88,58), (8,71,73), (14,94,92), (9,85,91), (13,85,99), (29,67,76), (28,70,69), (21,78,61), (9,94,87), (11,85,54), (2,93,79)	verde	
7	(14,76,77), (25,70,82), (20,88,84), (27,93,70), (26,96,96), (30,67,88), (4,88,54), (23,69,98), (25,78,61), (12,99,63), (16,69,100), (1,78,73), (4,78,65), (9,83,60), (17,68,94), (28,65,94), (19,66,93), (25,73,95), (20,78,81), (13,72,63)	verde	
8	(24,80,58), (5,65,56), (30,66,75), (13,98,80), (26,93,99), (30,69,100), (19,64,85), (30,88,53), (9,76,87), (9,85,71), (19,64,57), (11,87,84), (7,98,96), (25,65,80), (4,97,70), (16,78,56), (28,70,79), (14,76,61), (23,81,60), (29,82,59)	verde	
9	(21,99,60), (9,64,81), (6,90,65), (27,98,100), (1,72,59), (13,73,95), (25,83,60), (14,86,78), (14,99,87), (10,80,60), (17,91,56), (8,96,69), (12,82,81), (27,75,71), (27,85,60), (25,80,82), (11,69,63), (20,79,55), (27,89,73), (26,92,72)	verde	
10	(3,68,57), (30,89,90), (18,88,99), (6,90,65), (7,91,52), (9,68,86), (25,70,94), (25,71,65), (17,72,62), (23,92,75), (15,91,87), (22,86,89), (4,88,58), (5,70,58), (24,93,78), (6,77,93), (10,75,69), (1,80,62), (24,65,89), (16,86,96)	verde	
11	(31,74,68), (31,68,55), (6,83,56), (20,85,79), (6,67,73), (12,74,62), (27,93,77), (23,93,64), (18,97,80), (13,68,71), (14,81,85), (30,65,73), (25,97,59), (0,71,78), (22,85,86), (14,93,96), (12,81,85), (13,85,94), (13,67,78), (16,87,95)	verde	
12	(20,64,54), (11,91,76), (13,97,53), (28,74,75), (3,64,59), (2,78,64), (30,73,65), (14,78,68), (26,81,75), (25,72,64), (23,80,89), (17,64,55), (26,95,97), (4,90,68), (25,70,78), (1,68,54), (31,90,73), (30,81,81), (27,76,63), (14,65,67)	verde	
13	(12,82,84), (13,92,79), (6,98,71), (15,94,62), (23,83,55), (15,81,86), (15,96,85), (27,78,86), (30,92,61), (21,80,71), (23,77,66), (29,92,61), (0,65,67), (24,88,95), (20,67,87), (12,77,64), (20,94,71), (31,87,89), (29,79,61), (6,98,96)	verde	
14	(12,71,96), (24,77,93), (15,70,67), (26,87,67), (31,88,71), (19,93,85), (26,74,79), (20,93,65), (31,79,84), (2,99,62), (23,76,75), (29,81,79), (19,78,100), (6,80,86), (13,78,98), (8,95,82), (26,93,72), (4,77,91), (6,98,77), (8,78,87)	verde	
15	(22,65,65), (31,88,78), (29,99,74), (9,64,52), (4,68,70), (28,74,86), (21,96,61), (12,95,55), (1,71,57), (4,91,58), (29,77,90), (21,79,63), (21,73,59), (31,99,76), (4,64,75), (28,96,94), (28,81,94), (5,92,59), (19,69,71), (5,89,90)	verde	
16	(14,71,54), (7,75,79), (11,76,87), (14,65,85), (5,75,78), (21,99,63), (4,80,66), (16,67,79), (4,98,53), (1,64,80), (6,68,61), (31,87,53), (5,69,90), (3,69,83), (5,74,99), (15,69,61), (27,83,57), (28,65,77), (14,88,92), (18,89,84)	verde	
















17	(16,84,70), (30,99,99), (20,89,91), (13,87,91), (23,67,56), (9,88,65), (31,95,83), (4,86,55), (18,72,66), (7,88,57), (2,99,94), (9,81,56), (20,83,64), (26,69,95), (22,99,62), (24,68,96), (20,87,54), (27,93,57), (22,97,77), (31,70,78)	verde	
18	(23,87,98), (18,81,85), (11,81,100), (19,85,92), (26,79,76), (7,91,90), (28,65,56), (1,66,71), (30,80,59), (18,83,72), (24,81,90), (17,94,96), (4,86,93), (2,64,53), (23,67,53), (14,80,53), (16,92,98), (12,83,88), (23,69,98), (15,88,82)	verde	
19	(6,76,96), (9,96,90), (2,76,55), (29,90,90), (28,89,87), (31,75,100), (0,79,69), (6,78,70), (19,79,60), (30,66,98), (30,88,92), (4,93,63), (10,90,71), (24,74,64), (12,65,54), (30,65,74), (25,82,78), (3,76,99), (30,84,53), (8,92,89)	verde	
20	(30,95,64), (22,67,83), (19,69,56), (31,64,52), (26,65,52), (3,89,70), (19,71,81), (24,84,100), (18,69,87), (4,68,80), (7,87,100), (16,72,77), (31,77,92), (7,77,72), (18,96,67), (7,98,52), (27,73,92), (18,74,56), (18,77,66), (18,94,92)	verde	
21	(16,70,58), (30,95,71), (19,95,87), (17,64,91), (18,77,60), (7,88,84), (8,83,89), (25,83,64), (13,69,59), (8,68,94), (22,98,72), (4,67,61), (22,85,89), (15,78,71), (2,70,89), (20,90,60), (22,80,58), (23,76,79), (8,65,93), (10,77,61)	verde	
22	(18,70,66), (15,80,85), (31,99,86), (10,99,68), (9,91,89), (27,89,89), (3,91,72), (18,99,98), (24,84,87), (7,84,98), (9,75,87), (27,80,52), (11,94,96), (30,81,75), (6,74,70), (20,80,82), (8,76,75), (12,86,75), (12,83,69), (12,96,70)	verde	
23	(17,96,100), (18,87,98), (16,91,67), (28,94,93), (13,77,69), (17,82,80), (29,77,77), (27,82,95), (0,98,100), (26,84,56), (21,65,77), (2,89,100), (23,98,53), (13,75,65), (3,84,91), (4,68,69), (31,68,56), (15,90,72), (14,66,99), (12,89,54)	verde	
24	(8,93,74), (30,92,74), (13,64,82), (4,81,88), (27,64,70), (4,76,62), (19,82,93), (15,81,79), (18,90,85), (10,83,87), (8,77,64), (0,66,58), (10,67,53), (11,73,58), (7,70,94), (5,66,60), (20,86,98), (5,86,93), (4,68,87), (23,81,64)	verde	
25	(24,90,92), (8,70,100), (10,84,62), (7,84,85), (11,91,81), (22,78,90), (13,97,83), (19,97,99), (26,90,63), (17,73,56), (30,86,54), (0,69,95), (6,72,94), (6,99,66), (12,94,67), (23,67,68), (14,91,83), (29,94,75), (13,69,73), (26,84,79)	verde	
26	(16,99,93), (20,66,57), (20,68,58), (31,72,75), (9,76,88), (13,64,56), (4,84,72), (28,67,82), (2,73,59), (2,68,87), (30,93,65), (24,79,95), (30,80,90), (20,71,87), (25,87,87), (6,98,100), (16,70,67), (14,68,83), (8,74,56), (5,86,81)	verde	
27	(7,73,93), (7,93,83), (12,97,63), (22,98,95), (2,85,75), (25,78,65), (11,87,65), (28,81,87), (10,92,60), (9,67,70), (1,86,72), (29,94,88), (9,91,59), (5,95,55), (31,84,59), (31,86,58), (15,74,64), (10,82,92), (0,85,64), (14,67,70)	verde	
28	(27,84,82), (6,91,89), (9,97,52), (2,86,58), (23,78,54), (0,97,82), (7,89,100), (17,88,52), (15,71,80), (29,81,100), (27,79,75), (23,76,73), (1,64,92), (28,66,76), (7,71,55), (16,79,66), (4,66,72), (1,97,89), (4,77,58), (2,64,55)	verde	
29	(30,96,62), (5,74,72), (7,97,61), (6,78,91), (19,95,89), (24,84,72), (7,96,84), (5,71,91), (8,93,82), (19,74,61), (2,88,77), (9,73,91), (10,87,88), (4,94,92), (14,87,75), (0,94,54), (14,87,86), (8,69,94), (0,82,98), (23,72,61)	verde	
30	(6,95,93), (15,98,73), (10,68,55), (26,64,64), (10,69,74), (10,68,78), (7,94,76), (21,90,78), (17,93,58), (1,83,72), (11,84,84), (2,70,60), (19,72,52), (9,90,80), (17,95,59), (17,92,73), (19,76,59), (19,95,87), (12,88,73), (17,89,97)	verde	
31	(11,70,72), (11,75,78), (11,87,68), (11,71,79), (11,74,98), (10,84,78), (2,97,53), (20,69,76), (14,73,95), (30,77,55), (30,78,72), (15,69,93), (19,88,55), (17,84,84), (13,68,53), (25,87,78), (4,92,83), (31,81,65), (10,87,96), (26,67,96)	verde	
32	(14,86,88), (12,68,90), (13,70,77), (3,87,100), (11,84,91), (9,92,68), (9,85,58), (11,68,97), (22,67,53), (20,92,67), (21,94,62), (3,72,74), (3,70,55), (17,97,85), (9,73,97), (3,96,69), (23,95,93), (8,91,81), (20,96,77), (31,84,70)	verde	
33	(16,95,63), (29,91,83), (18,77,62), (18,92,82), (15,80,52), (8,69,84), (16,75,100), (15,74,64), (4,84,95), (11,66,70), (0,74,59), (14,98,69), (25,79,98), (31,71,69), (10,76,94), (6,77,72), (7,93,56), (11,89,69), (29,81,83), (12,91,83)	verde	
















34	(5,84,97), (1,76,74), (16,88,93), (1,75,71), (1,93,58), (9,77,55), (13,91,54), (3,91,55), (9,90,64), (4,93,67), (24,76,52), (14,85,61), (31,91,60), (1,68,93), (8,96,87), (9,99,85), (10,87,67), (2,69,72), (1,90,57), (5,64,77)	verde	
35	(17,76,64), (13,85,67), (12,64,74), (24,66,59), (3,75,87), (4,93,53), (23,88,68), (10,98,85), (0,96,63), (15,87,86), (7,79,61), (5,71,99), (12,66,71), (3,98,56), (14,70,63), (16,66,64), (12,96,82), (5,75,94), (17,85,93), (21,68,95)	verde	
36	(4,97,72), (31,70,63), (7,99,99), (17,80,98), (13,75,90), (31,80,65), (21,71,52), (27,72,76), (11,91,55), (25,77,69), (14,84,64), (5,83,85), (26,72,80), (24,98,93), (25,77,72), (27,92,55), (26,84,81), (16,89,87), (21,68,99), (30,65,61)	verde	
37	(23,77,74), (8,70,85), (22,81,88), (31,92,62), (9,84,73), (17,90,55), (7,80,97), (20,68,89), (24,66,88), (24,89,82), (21,87,87), (3,78,90), (5,95,99), (8,97,73), (30,90,87), (19,83,76), (27,73,69), (2,80,89), (10,76,75), (4,87,89)	verde	
38	(15,73,79), (12,76,98), (13,85,84), (11,82,82), (14,66,74), (13,68,58), (23,90,94), (19,89,58), (28,80,56), (24,90,66), (19,69,61), (24,85,61), (10,64,97), (20,72,67), (6,74,72), (16,71,100), (0,68,73), (23,92,81), (26,83,83), (25,94,78)	verde	
39	(2,83,79), (14,65,89), (18,64,75), (4,94,71), (16,89,64), (6,75,71), (20,95,71), (10,94,96), (4,99,73), (28,76,54), (19,99,83), (5,90,72), (29,97,75), (4,88,52), (8,66,95), (21,79,65), (4,70,95), (0,66,83), (9,89,92), (4,64,70)	verde	
40	(25,67,83), (23,85,58), (0,90,78), (30,90,86), (23,90,68), (12,87,61), (15,73,69), (12,83,59), (14,89,57), (10,71,84), (17,96,83), (7,91,87), (25,98,74), (20,69,61), (8,88,87), (10,67,94), (12,64,95), (7,88,82), (13,96,55), (27,93,98)	verde	
41	(23,75,79), (13,98,86), (3,73,55), (19,89,62), (6,71,52), (23,82,65), (18,74,84), (23,77,65), (27,88,61), (30,86,87), (8,64,67), (10,81,94), (12,74,53), (0,79,98), (21,65,83), (20,91,69), (1,80,85), (15,85,96), (28,83,93), (7,84,73)	verde	
42	(30,77,95), (3,73,79), (25,92,56), (30,68,99), (9,75,89), (24,90,90), (21,97,68), (18,69,95), (17,95,69), (20,82,52), (14,83,83), (10,73,57), (11,70,64), (23,64,99), (5,95,78), (0,97,75), (14,73,87), (21,98,83), (13,79,72), (27,71,61)	verde	
43	(27,99,53), (16,75,69), (8,99,86), (21,84,69), (13,82,66), (9,85,65), (26,89,52), (13,73,97), (9,82,64), (6,72,80), (26,74,62), (0,89,56), (9,76,56), (30,74,94), (21,95,64), (13,87,63), (23,88,95), (16,73,55), (6,71,83), (20,74,56)	verde	
44	(16,96,91), (15,77,74), (11,91,91), (25,86,70), (17,68,62), (12,95,56), (14,94,80), (0,92,87), (27,67,66), (31,73,87), (6,93,55), (14,73,85), (22,88,74), (24,68,64), (8,71,54), (12,75,76), (17,74,96), (6,75,68), (5,95,59), (26,71,75)	verde	
45	(6,64,85), (17,71,55), (9,65,57), (28,83,72), (17,90,57), (9,70,93), (9,83,80), (0,96,70), (16,80,61), (15,92,91), (2,65,66), (30,76,82), (18,83,81), (6,64,82), (3,95,73), (9,75,56), (15,70,88), (29,83,73), (2,84,59), (18,66,92)	verde	
46	(20,97,73), (16,85,98), (11,80,60), (28,84,90), (16,91,90), (18,87,63), (0,64,82), (20,92,67), (11,94,74), (27,92,85), (28,65,79), (16,94,97), (27,85,70), (17,68,61), (5,90,59), (11,99,69), (26,91,81), (1,99,98), (30,86,69), (30,79,74)	verde	
47	(16,74,68), (8,64,66), (8,75,63), (19,85,78), (28,98,66), (23,90,93), (1,66,90), (31,90,55), (13,66,72), (20,69,98), (5,82,80), (31,74,98), (7,90,79), (28,75,93), (16,73,78), (8,64,62), (24,68,64), (30,81,53), (7,95,96), (3,87,67)	verde	
48	(18,88,69), (11,81,64), (1,87,74), (4,69,76), (11,90,72), (9,97,55), (14,69,61), (28,65,64), (27,67,81), (4,88,66), (0,94,73), (28,99,82), (24,76,65), (20,91,100), (0,67,99), (3,84,85), (8,73,77), (18,68,84), (18,84,60), (10,82,65)	verde	
49	(12,71,55), (30,87,86), (6,68,86), (8,72,52), (31,70,90), (20,74,77), (24,83,91), (10,69,97), (7,73,88), (27,77,83), (31,75,100), (7,85,91), (1,73,62), (3,91,69), (31,76,80), (2,69,89), (10,77,66), (22,97,60), (28,72,99), (4,85,97)	verde	
50	(28,75,79), (25,99,77), (11,83,93), (6,95,75), (21,75,78), (6,68,81), (15,71,66), (6,72,88), (18,64,60), (3,67,91), (9,77,72), (0,69,54), (8,98,62), (6,67,56), (24,75,94), (17,95,61), (0,75,78), (26,78,78), (13,73,59), (23,86,98)	verde	
















51	(35,52,41), (33,40,41), (38,47,51), (33,54,42), (50,48,51), (36,51,42), (55,55,48), (57,43,41), (43,55,51), (37,43,43), (38,51,49), (53,57,50), (48,57,46), (56,41,45), (37,42,48), (44,53,48), (32,46,51), (49,57,43), (55,59,46), (44,59,44)	semi maduro	
52	(51,56,45), (51,54,50), (33,43,47), (51,63,43), (52,59,50), (44,60,51), (41,62,49), (45,50,43), (40,54,41), (40,41,45), (55,45,41), (47,47,45), (47,48,50), (36,42,41), (36,41,51), (54,52,46), (44,61,47), (48,50,50), (47,50,49), (34,50,50)	semi maduro	
53	(54,63,51), (49,44,50), (42,51,45), (56,58,45), (32,40,47), (39,58,48), (34,56,48), (41,51,43), (47,49,44), (38,44,50), (35,46,45), (48,59,41), (42,48,43), (46,63,48), (45,40,49), (32,44,44), (39,62,43), (44,43,50), (52,59,50), (56,60,48)	semi maduro	
54	(57,48,47), (42,44,48), (56,57,50), (53,63,43), (37,54,46), (44,62,48), (53,56,45), (57,52,50), (49,52,41), (43,41,43), (41,63,41), (55,42,45), (56,47,44), (45,60,50), (49,61,47), (46,51,44), (44,51,48), (49,46,42), (42,48,42), (57,49,43)	semi maduro	
55	(45,55,46), (34,42,45), (50,55,43), (51,47,49), (33,46,51), (49,50,41), (45,58,46), (38,45,51), (47,49,41), (42,56,50), (54,45,50), (46,55,48), (35,49,41), (32,41,50), (44,60,42), (43,44,47), (56,59,51), (52,44,48), (44,53,46), (51,61,49)	semi maduro	
56	(43,43,43), (48,48,47), (44,44,45), (54,51,49), (35,46,43), (47,41,42), (42,52,41), (32,51,41), (34,47,51), (57,46,42), (34,42,51), (54,44,50), (48,46,48), (51,48,43), (44,52,49), (38,42,50), (34,60,42), (50,61,47), (42,44,50), (36,60,42)	semi maduro	
57	(52,46,50), (47,47,48), (55,50,51), (46,61,47), (41,44,48), (54,59,49), (52,41,51), (47,52,47), (41,51,51), (40,54,44), (44,44,46), (46,56,47), (50,53,43), (48,44,43), (39,52,46), (34,52,49), (47,50,49), (56,51,51), (41,48,44), (38,62,44)	semi maduro	
58	(45,48,50), (45,59,43), (44,61,48), (32,55,44), (56,48,43), (34,61,50), (54,58,46), (45,57,48), (57,61,42), (32,53,49), (49,58,50), (50,53,50), (54,61,41), (37,55,47), (35,62,51), (46,41,41), (37,54,46), (48,47,50), (40,57,46), (46,55,47)	semi maduro	
59	(38,59,50), (43,58,43), (33,53,44), (46,53,48), (46,48,47), (43,63,48), (55,53,50), (38,57,48), (40,44,42), (54,54,46), (50,58,43), (35,41,48), (55,54,42), (33,43,48), (52,40,46), (32,56,51), (37,60,42), (33,60,47), (52,59,49), (55,57,45)	semi maduro	
60	(44,44,41), (56,52,44), (42,52,44), (42,49,46), (36,57,46), (37,43,41), (50,51,51), (35,42,46), (35,59,50), (38,53,44), (36,57,41), (36,43,46), (43,46,51), (49,52,41), (44,44,46), (34,42,50), (36,55,51), (52,59,41), (39,40,44), (37,62,42)	semi maduro	
61	(55,61,47), (40,60,43), (57,62,49), (42,61,43), (35,62,49), (46,62,41), (45,61,51), (41,59,47), (47,58,43), (53,58,43), (45,50,42), (43,48,46), (53,52,41), (36,40,51), (35,51,44), (46,56,43), (35,57,51), (53,43,51), (41,48,51), (51,55,45)	semi maduro	
62	(33,48,49), (52,59,49), (38,49,45), (52,41,48), (52,58,48), (39,59,48), (51,50,44), (48,42,41), (52,42,41), (56,54,44), (38,52,42), (38,42,51), (46,41,43), (57,54,51), (44,51,42), (44,61,46), (56,58,47), (39,62,50), (34,45,47), (45,47,50)	semi maduro	
63	(36,63,45), (44,60,44), (39,45,51), (55,46,43), (53,56,44), (36,62,46), (50,41,41), (36,46,45), (43,49,47), (37,43,45), (38,56,49), (35,44,43), (32,54,51), (51,54,41), (43,59,42), (48,41,51), (44,48,46), (43,58,44), (47,63,51), (51,51,46)	semi maduro	

















64	(32,44,48), (36,63,44), (33,43,42), (46,63,44), (41,55,41), (37,40,49), (50,58,49), (32,45,44), (38,63,43), (49,63,43), (56,61,46), (47,48,46), (51,62,45), (50,48,51), (43,49,44), (54,60,42), (47,41,43), (57,58,44), (34,40,48), (51,58,45)	semi maduro	
65	(45,46,45), (54,52,48), (50,53,43), (47,43,51), (48,60,46), (34,53,49), (33,58,43), (41,48,50), (44,63,50), (57,54,49), (47,58,49), (41,47,46), (52,47,46), (44,48,47), (57,63,41), (54,48,50), (56,46,42), (33,59,50), (49,46,50), (52,56,44)	semi maduro	
66	(34,45,44), (51,50,49), (41,57,47), (36,58,47), (39,49,42), (41,52,42), (45,60,46), (40,51,41), (35,58,49), (41,56,51), (56,40,47), (51,55,46), (45,55,44), (45,53,51), (32,41,44), (55,52,47), (47,44,45), (55,50,46), (47,46,50), (35,55,44)	semi maduro	
67	(43,43,42), (42,44,47), (57,44,48), (43,57,47), (48,42,48), (53,45,41), (47,42,47), (45,52,50), (32,57,45), (48,47,47), (49,51,45), (44,40,51), (47,43,41), (50,50,46), (41,42,49), (39,42,50), (49,63,46), (49,59,49), (37,43,50), (55,52,46)	semi maduro	
68	(34,44,43), (52,61,50), (32,45,42), (42,50,44), (44,57,43), (48,55,47), (48,51,43), (48,53,49), (43,49,45), (55,50,49), (53,40,45), (49,48,48), (52,51,47), (49,51,46), (50,54,42), (51,54,41), (35,60,46), (36,48,43), (46,46,44), (34,58,48)	semi maduro	
69	(42,47,41), (55,42,50), (57,40,46), (40,54,50), (38,57,45), (39,56,46), (54,61,47), (56,53,51), (40,44,46), (36,40,41), (45,59,48), (56,42,46), (57,43,46), (54,52,50), (56,61,42), (42,51,42), (33,51,49), (38,57,51), (43,43,47), (51,45,50)	semi maduro	
70	(39,40,42), (52,50,44), (41,45,46), (49,60,51), (44,54,46), (41,63,42), (54,52,49), (48,55,51), (35,41,46), (50,42,45), (40,41,51), (51,58,44), (44,41,47), (54,57,50), (45,60,44), (36,48,45), (34,56,51), (49,55,51), (41,41,49), (49,40,45)	semi maduro	
71	(40,50,49), (40,40,48), (46,60,45), (34,63,45), (56,47,50), (57,60,42), (43,40,48), (39,46,51), (57,47,48), (39,41,47), (41,51,47), (48,63,46), (57,48,48), (42,62,43), (45,62,41), (32,50,50), (35,46,49), (49,60,44), (47,40,50), (57,50,47)	semi maduro	
72	(50,62,43), (50,52,41), (35,44,46), (47,48,41), (54,40,44), (36,60,48), (34,56,45), (56,40,48), (50,58,48), (39,46,41), (41,53,45), (54,61,44), (54,53,47), (36,56,42), (56,40,44), (39,41,47), (56,47,46), (41,41,47), (35,50,50), (51,43,46)	semi maduro	
73	(56,54,46), (40,60,45), (45,59,50), (38,48,46), (42,56,51), (45,41,48), (51,52,47), (49,63,49), (52,47,41), (35,52,45), (42,44,43), (56,42,45), (33,58,43), (45,51,46), (36,63,46), (52,51,43), (38,44,45), (40,48,43), (39,49,46), (41,61,41)	semi maduro	
74	(48,60,49), (57,50,41), (48,58,47), (56,41,46), (43,60,45), (53,60,44), (38,41,45), (37,63,41), (51,43,46), (35,63,43), (42,57,44), (56,46,50), (32,54,46), (54,46,43), (38,56,49), (41,53,51), (50,61,46), (37,46,48), (42,49,41), (56,63,46)	semi maduro	
75	(56,57,46), (50,61,43), (53,56,47), (38,41,46), (40,40,46), (57,43,46), (32,40,48), (52,60,49), (55,55,48), (32,56,51), (45,57,41), (51,46,44), (37,57,45), (47,51,47), (46,42,45), (43,47,47), (40,49,48), (39,59,47), (46,44,48), (41,47,48)	semi maduro	
76	(50,43,43), (56,48,42), (38,44,48), (51,53,44), (37,60,41), (44,51,51), (34,47,48), (44,50,51), (36,54,49), (34,53,46), (43,61,49), (52,53,50), (56,48,43), (57,50,48), (35,62,48), (36,40,44), (38,63,49), (51,57,49), (44,56,42), (35,53,43)	semi maduro	





















77	(52,43,51), (53,59,49), (44,43,47), (38,51,45), (57,52,43), (38,50,44), (41,41,43), (39,62,51), (56,59,49), (35,50,48), (49,49,48), (44,43,48), (33,52,49), (57,56,48), (55,61,49), (33,59,44), (35,41,46), (47,45,43), (34,50,46), (54,42,46)	semi maduro	
78	(34,53,46), (54,47,48), (40,49,45), (38,45,51), (36,59,41), (35,53,50), (44,51,41), (42,58,46), (32,46,47), (49,59,44), (37,44,42), (57,46,46), (54,53,46), (36,43,44), (55,63,46), (32,63,51), (57,55,42), (48,43,46), (45,44,44), (46,60,49)	semi maduro	
79	(49,44,50), (51,41,45), (42,62,41), (32,55,44), (53,56,51), (45,55,50), (49,55,50), (49,62,44), (38,57,41), (52,57,41), (53,43,49), (47,59,51), (54,63,44), (39,41,51), (33,52,46), (50,62,45), (43,45,50), (49,54,44), (54,48,48), (45,61,48)	semi maduro	
80	(47,48,43), (36,48,49), (48,56,41), (42,57,44), (56,52,45), (40,52,44), (41,58,45), (52,59,46), (38,41,44), (44,53,44), (32,45,51), (44,61,46), (52,61,42), (39,42,51), (44,48,44), (38,58,43), (44,54,47), (47,63,49), (32,53,49), (45,51,47)	semi maduro	
81	(39,52,45), (33,41,42), (33,43,49), (34,55,43), (40,40,42), (54,54,49), (52,51,46), (49,45,45), (52,50,42), (42,50,47), (41,62,41), (37,40,41), (42,42,43), (52,56,49), (46,51,46), (36,43,46), (54,55,48), (56,49,41), (45,44,45), (55,59,51)	semi maduro	
82	(40,50,50), (57,46,51), (39,59,47), (52,56,51), (51,55,49), (39,62,46), (45,55,42), (54,59,49), (35,42,45), (36,46,46), (44,51,44), (48,55,44), (34,59,47), (49,50,43), (38,62,49), (41,51,46), (36,63,44), (56,55,45), (56,47,50), (53,54,44)	semi maduro	
83	(51,40,46), (56,45,50), (39,44,44), (35,61,49), (42,40,49), (55,44,42), (57,49,44), (48,54,41), (50,42,45), (42,61,42), (34,43,50), (49,61,48), (36,41,50), (53,43,46), (35,54,46), (47,40,47), (32,58,49), (42,60,42), (37,59,44), (39,42,47)	semi maduro	
84	(55,43,51), (52,41,50), (44,61,47), (45,63,46), (46,62,46), (49,60,45), (33,56,41), (46,61,47), (40,44,48), (57,52,48), (34,45,49), (47,57,51), (45,61,48), (50,40,51), (46,45,51), (57,61,48), (54,63,43), (57,43,47), (47,56,51), (36,51,47)	semi maduro	
85	(50,57,47), (35,58,47), (57,45,49), (57,48,51), (55,60,45), (40,59,49), (56,56,42), (36,52,45), (51,55,44), (54,46,51), (53,53,45), (46,45,49), (38,56,45), (52,44,50), (48,52,48), (37,59,41), (56,42,50), (37,52,43), (51,60,43), (32,43,50)	semi maduro	
86	(55,52,46), (50,40,45), (32,46,41), (51,41,45), (34,42,47), (37,43,46), (46,52,49), (43,51,41), (54,41,46), (42,61,51), (41,49,43), (40,53,41), (53,58,42), (51,43,51), (57,61,50), (46,42,47), (51,60,44), (35,42,48), (54,60,41), (32,55,48)	semi maduro	
87	(33,54,46), (32,62,51), (50,53,41), (50,44,44), (49,46,47), (32,50,51), (57,55,46), (32,53,44), (42,46,44), (38,59,49), (37,48,50), (53,48,43), (53,59,51), (39,60,46), (50,54,50), (40,43,50), (48,63,41), (51,59,45), (55,62,48), (45,57,48)	semi maduro	
88	(42,49,49), (32,43,50), (50,60,44), (53,44,49), (44,58,50), (50,60,43), (51,55,42), (54,52,47), (42,47,45), (40,43,50), (46,42,46), (53,40,49), (39,40,44), (57,47,41), (55,53,49), (41,62,47), (35,40,43), (47,54,41), (42,42,46), (37,55,42)	semi maduro	
89	(55,46,44), (51,56,50), (45,52,47), (48,49,46), (37,52,50), (38,51,47), (49,57,44), (45,43,50), (45,41,44), (35,47,50), (50,54,49), (56,41,41), (40,59,50), (35,53,47), (53,48,51), (33,59,46), (38,44,47), (37,51,48), (44,43,50), (32,43,51)	semi maduro	






















90	(45,57,41), (54,63,48), (40,61,49), (50,51,46), (42,45,46), (40,52,43), (39,59,42), (34,54,46), (44,42,51), (47,49,43), (45,43,45), (50,53,46), (42,46,44), (48,42,50), (40,50,45), (35,54,48), (42,55,47), (42,46,51), (54,51,50), (49,44,43)	semi maduro	
91	(35,46,46), (38,40,46), (56,60,50), (33,40,47), (34,42,42), (39,42,47), (54,55,41), (53,57,44), (49,41,44), (50,40,50), (44,53,50), (42,49,43), (53,49,48), (37,60,48), (40,57,43), (53,56,51), (36,57,49), (41,51,48), (55,57,42), (36,60,42)	semi maduro	
92	(43,48,44), (46,42,43), (43,43,50), (47,56,49), (55,60,49), (37,47,50), (38,51,48), (36,55,51), (45,52,51), (57,54,44), (50,62,49), (46,40,43), (36,43,50), (54,60,51), (39,63,44), (56,56,44), (44,48,51), (39,49,41), (38,57,41), (51,55,47)	semi maduro	
93	(33,56,51), (54,43,42), (53,60,46), (50,63,41), (35,54,41), (49,43,43), (40,52,46), (46,49,47), (32,41,50), (33,63,46), (57,53,47), (38,50,45), (33,47,42), (56,54,45), (38,58,51), (38,56,44), (56,63,51), (55,49,45), (35,52,44), (43,54,42)	semi maduro	
94	(34,58,47), (43,46,49), (32,43,46), (55,61,45), (48,46,45), (56,53,41), (42,44,46), (38,45,43), (39,57,48), (39,56,42), (55,42,45), (52,54,45), (40,51,44), (44,41,43), (55,50,42), (40,49,43), (40,42,51), (57,45,42), (35,56,48), (39,49,47)	semi maduro	
95	(57,62,41), (39,51,51), (34,62,46), (43,47,47), (34,46,47), (42,62,44), (57,43,44), (35,52,49), (35,52,48), (44,46,46), (37,47,41), (39,44,49), (36,59,46), (48,60,44), (44,48,44), (49,60,47), (41,58,41), (51,49,44), (45,58,47), (48,58,46)	semi maduro	
96	(37,49,51), (54,49,51), (33,55,44), (54,48,43), (47,53,42), (55,62,46), (40,55,49), (38,41,49), (43,40,46), (41,61,51), (40,46,44), (46,50,49), (36,63,44), (45,47,43), (55,45,50), (40,52,47), (56,47,41), (48,55,46), (38,63,48), (40,59,41)	semi maduro	
97	(49,40,48), (36,42,45), (50,49,46), (46,60,50), (49,50,47), (55,42,47), (48,51,46), (49,62,43), (44,56,45), (57,54,49), (40,48,42), (54,63,46), (49,57,51), (37,42,47), (45,57,43), (40,48,49), (51,59,45), (41,47,48), (42,58,50), (45,45,43)	semi maduro	
98	(50,48,42), (32,49,42), (33,47,46), (44,57,45), (46,59,46), (42,43,48), (40,40,49), (39,46,48), (36,52,44), (43,50,47), (43,52,41), (42,60,50), (55,56,43), (35,52,49), (44,62,45), (53,48,46), (48,50,45), (41,47,43), (32,47,44), (36,62,47)	semi maduro	
99	(57,61,49), (39,52,46), (56,54,43), (42,58,47), (40,58,44), (36,63,51), (37,40,50), (50,59,50), (42,59,48), (54,58,44), (41,45,47), (48,56,42), (32,55,44), (36,60,48), (49,56,51), (44,43,50), (36,57,45), (42,40,42), (47,45,47), (41,48,43)	semi maduro	
100	(44,54,41), (43,55,47), (35,54,42), (36,62,47), (50,42,44), (50,50,41), (50,63,51), (50,48,42), (48,56,45), (50,46,44), (38,43,48), (45,52,41), (35,41,47), (49,52,49), (43,55,44), (56,63,44), (50,59,51), (33,46,41), (39,50,50), (55,48,49)	semi maduro	
101	(90,32,26), (62,4,1), (96,36,16), (64,10,29), (58,11,3), (72,27,4), (59,35,17), (70,13,33), (80,13,0), (60,1,2), (97,28,39), (83,0,26), (58,7,26), (93,1,10), (80,26,40), (76,22,26), (72,1,25), (77,20,11), (79,29,34), (67,31,9)	maduro	
102	(82,39,40), (74,16,8), (73,20,24), (82,0,3), (82,29,33), (79,23,17), (92,22,18), (83,37,2), (60,24,31), (85,2,24), (100,38,38), (65,36,6), (63,26,27), (76,24,20), (76,21,14), (92,17,3), (82,26,21), (78,0,18), (58,28,0), (69,37,34)	maduro	
103	(77,24,20), (97,38,17), (92,36,31), (80,8,33), (70,15,36), (61,18,36), (73,32,17), (64,31,19), (73,25,7), (100,3,23), (94,19,39), (92,30,15), (73,34,20), (95,10,19), (70,38,9), (92,14,14), (66,28,16), (81,8,31), (67,34,30), (77,12,9)	maduro	













104	(90,10,12), (72,9,6), (94,26,16), (86,30,15), (61,32,36), (70,15,13), (81,21,13), (97,31,29), (84,17,10), (83,24,11), (87,11,23), (73,33,21), (85,34,6), (74,7,30), (100,19,9), (90,5,9), (82,4,17), (87,2,21), (64,19,4), (79,15,25)	maduro	
105	(97,35,24), (58,10,10), (72,19,12), (85,22,5), (60,13,7), (72,39,13), (81,24,9), (67,8,21), (62,28,33), (66,12,38), (62,35,25), (60,32,11), (65,15,12), (66,2,12), (69,25,13), (71,14,36), (95,37,27), (70,34,0), (92,33,41), (67,2,19)	maduro	
106	(86,14,29), (82,7,22), (84,24,17), (82,8,5), (82,25,28), (76,6,2), (86,0,29), (85,9,13), (67,3,11), (100,31,3), (83,1,25), (94,29,10), (91,8,37), (88,0,38), (79,8,31), (73,12,17), (60,21,18), (83,35,39), (60,18,31), (74,3,2)	maduro	
107	(92,8,28), (61,22,2), (83,16,31), (99,0,0), (70,37,38), (78,7,29), (95,24,19), (64,36,34), (99,6,29), (97,18,1), (64,7,4), (77,25,17), (66,22,31), (63,10,11), (61,18,37), (63,11,7), (64,39,0), (80,11,33), (87,4,15), (96,8,14)	maduro	
108	(69,35,14), (71,27,6), (74,37,32), (68,26,29), (99,20,1), (64,33,18), (64,31,10), (76,32,12), (72,28,39), (62,4,28), (58,39,16), (76,16,38), (90,28,33), (84,37,32), (59,36,33), (89,8,19), (98,10,35), (94,38,40), (79,29,39), (66,29,29)	maduro	
109	(63,13,33), (89,33,8), (81,39,36), (66,26,2), (95,5,17), (83,6,36), (99,6,31), (87,34,36), (86,28,2), (88,11,14), (100,33,9), (87,27,23), (81,17,22), (78,18,9), (81,15,27), (85,29,24), (77,8,13), (64,34,23), (92,11,16), (99,1,30)	maduro	
110	(73,34,23), (62,23,32), (96,36,37), (59,0,16), (69,26,25), (61,18,0), (71,24,19), (83,39,14), (82,29,10), (77,24,5), (76,33,22), (78,12,29), (72,32,6), (93,19,13), (61,38,38), (84,27,7), (74,24,0), (61,16,38), (75,21,7), (91,0,27)	maduro	
111	(86,38,23), (88,37,18), (61,6,20), (88,9,38), (78,29,4), (60,7,29), (72,2,37), (78,30,39), (83,32,36), (69,3,35), (60,27,30), (69,18,40), (91,32,25), (74,37,22), (89,26,1), (90,14,26), (81,6,28), (65,36,37), (80,18,2), (100,36,28)	maduro	
112	(73,37,27), (82,35,17), (72,3,36), (61,9,23), (87,29,14), (66,12,11), (79,31,38), (80,11,36), (71,20,38), (100,33,27), (69,8,18), (86,0,37), (66,30,38), (77,3,13), (67,2,34), (86,15,20), (87,10,39), (63,29,12), (92,33,7), (89,29,17)	maduro	
113	(68,39,37), (92,12,18), (63,3,13), (61,20,41), (77,14,32), (77,21,9), (99,0,4), (95,38,16), (98,3,5), (97,16,24), (97,16,27), (71,28,7), (99,21,1), (75,30,29), (86,31,37), (68,25,2), (77,26,39), (74,2,9), (85,13,4), (60,36,0)	maduro	
114	(61,6,19), (98,17,6), (94,12,8), (67,14,38), (90,26,11), (89,33,41), (87,35,37), (61,4,1), (63,34,21), (94,35,37), (62,3,14), (90,17,13), (65,18,24), (84,26,29), (85,6,22), (78,21,28), (81,5,5), (81,11,11), (96,23,0), (73,9,29)	maduro	
115	(73,24,11), (74,0,34), (72,26,32), (67,32,34), (71,1,24), (77,36,16), (100,20,28), (78,4,32), (96,26,12), (90,7,27), (79,22,30), (63,17,12), (61,20,37), (95,6,17), (90,2,12), (62,20,16), (64,16,2), (66,1,0), (67,7,13), (86,16,0)	maduro	
116	(81,2,7), (74,31,30), (58,31,0), (98,19,25), (83,23,33), (69,13,28), (92,18,15), (68,12,26), (93,13,20), (86,13,38), (92,27,41), (71,28,39), (92,23,34), (63,25,28), (59,16,4), (92,24,26), (93,26,7), (74,30,33), (100,4,10), (73,19,12)	maduro	
117	(65,21,0), (58,37,29), (71,7,15), (63,18,13), (84,2,30), (86,18,40), (88,28,25), (87,31,2), (100,26,15), (65,31,10), (65,32,18), (69,27,40), (82,23,29), (94,18,19), (78,1,22), (91,28,30), (100,29,34), (92,5,20), (91,22,29), (66,28,23)	maduro	
118	(91,10,30), (62,39,33), (100,2,25), (74,32,40), (60,37,33), (96,4,22), (83,32,19), (63,19,11), (62,31,30), (86,4,22), (75,30,41), (61,15,36), (100,25,1), (82,29,36), (100,26,3), (94,15,14), (59,0,26), (95,16,38), (89,18,29), (67,29,21)	maduro	
119	(93,23,4), (73,28,27), (93,32,22), (77,17,4), (91,8,37), (84,8,2), (69,22,29), (91,14,17), (74,35,18), (96,13,40), (100,38,8), (80,0,4), (89,23,37), (92,1,12), (87,36,10), (58,32,18), (68,38,39), (70,28,26), (63,25,20), (74,1,38)	maduro	
120	(77,19,29), (75,2,38), (63,34,36), (91,16,15), (80,37,11), (73,38,11), (64,13,5), (92,39,0), (79,15,22), (65,33,36), (95,33,41), (65,0,6), (80,24,3), (93,1,23), (62,20,2), (86,4,40), (93,7,6), (95,1,9), (61,22,23), (89,4,33)	maduro	
121	(72,15,35), (74,21,38), (87,31,20), (81,1,14), (76,32,4), (95,4,22), (90,7,9), (66,18,21), (86,28,13), (61,38,37), (99,11,37), (96,34,10), (73,37,15), (78,35,14), (88,24,9), (97,38,40), (83,15,17), (75,33,12), (67,37,5), (96,8,24)	maduro	



122	(85,19,19), (81,16,39), (84,3,41), (97,24,31), (95,4,18), (82,13,8), (99,15,29), (97,37,40), (95,5,30), (63,31,7), (68,39,6), (88,21,16), (84,1,11), (67,24,33), (62,27,18), (90,22,10), (65,27,22), (91,34,8), (74,22,34), (79,36,19)	maduro	
123	(66,35,31), (74,34,40), (97,11,25), (75,27,4), (64,34,22), (89,0,24), (77,23,6), (59,12,31), (81,11,20), (81,7,22), (94,31,18), (76,8,21), (75,31,26), (92,25,17), (93,1,28), (95,23,14), (91,29,15), (87,31,19), (90,33,30), (87,12,30)	maduro	
124	(72,18,8), (95,28,15), (77,26,25), (94,9,41), (92,16,7), (98,37,28), (67,34,9), (69,12,40), (58,28,11), (60,25,17), (91,8,1), (78,26,35), (79,3,24), (87,24,2), (76,29,3), (90,26,25), (76,11,12), (82,3,33), (90,17,37), (80,32,36)	maduro	
125	(73,23,12), (85,33,15), (60,29,16), (100,5,19), (67,1,0), (90,16,30), (81,25,23), (91,29,38), (86,16,0), (59,31,7), (99,26,23), (66,37,21), (94,22,2), (58,39,26), (91,4,35), (85,17,39), (82,28,11), (73,39,27), (72,36,0), (63,4,25)	maduro	
126	(88,2,9), (97,9,40), (82,38,37), (91,24,39), (81,9,35), (69,30,23), (80,31,1), (78,18,39), (64,23,27), (77,20,35), (74,9,13), (83,11,2), (93,17,8), (97,8,40), (60,39,15), (79,30,20), (86,38,23), (68,4,19), (84,34,35), (85,7,13)	maduro	
127	(69,36,36), (91,16,18), (75,8,24), (79,37,0), (70,2,32), (61,3,12), (94,24,14), (83,39,24), (63,13,4), (86,37,34), (98,6,26), (76,7,8), (90,19,3), (90,28,6), (100,31,22), (92,11,33), (77,5,8), (71,24,30), (100,3,9), (92,3,26)	maduro	
128	(95,31,10), (84,8,15), (96,36,13), (84,17,10), (59,24,26), (84,26,31), (98,4,23), (83,29,32), (64,21,3), (69,36,12), (87,31,16), (100,2,15), (88,38,22), (79,0,34), (78,24,24), (69,25,36), (92,21,33), (64,4,24), (73,8,40), (72,30,0)	maduro	
129	(83,15,23), (78,31,32), (82,7,31), (77,10,7), (68,24,33), (91,2,30), (88,14,3), (81,13,24), (64,22,1), (77,13,29), (77,20,11), (75,3,20), (78,9,10), (62,39,8), (79,22,1), (65,11,29), (64,12,26), (100,20,16), (71,8,13), (95,9,33)	maduro	
130	(80,9,34), (99,1,27), (95,35,35), (69,1,35), (78,35,17), (81,25,37), (97,3,38), (75,16,0), (64,13,19), (65,29,24), (85,38,32), (70,6,38), (82,12,38), (85,11,33), (79,3,12), (74,36,12), (97,19,10), (79,6,13), (98,6,25), (68,25,3)	maduro	
131	(84,19,12), (66,26,10), (92,20,39), (84,0,32), (84,37,23), (82,25,40), (62,26,38), (95,18,37), (86,9,16), (81,12,27), (92,29,28), (91,9,15), (62,10,18), (64,5,9), (72,37,29), (80,14,10), (61,36,1), (63,6,35), (98,28,8), (84,5,24)	maduro	
132	(59,8,33), (59,25,23), (86,13,23), (80,26,41), (87,10,37), (66,2,27), (71,6,33), (71,8,29), (95,31,10), (99,34,22), (67,12,23), (78,33,20), (64,15,19), (60,37,18), (84,30,9), (92,5,7), (78,23,23), (70,13,22), (73,22,13), (83,12,30)	maduro	
133	(100,18,17), (67,18,35), (63,4,27), (100,14,9), (63,20,25), (83,26,4), (87,7,1), (73,24,38), (67,27,16), (98,34,6), (72,33,23), (91,4,31), (71,35,9), (64,5,22), (95,6,14), (67,6,8), (66,21,7), (96,11,15), (58,5,12), (81,11,2)	maduro	
134	(66,18,3), (80,32,3), (64,39,13), (77,20,37), (86,20,14), (78,28,14), (78,3,11), (99,13,3), (63,10,27), (83,25,30), (91,23,26), (100,1,1), (71,17,31), (82,39,32), (70,23,27), (85,15,29), (100,12,22), (92,30,32), (66,39,19), (69,20,33)	maduro	
135	(93,11,14), (92,14,38), (76,32,35), (75,37,5), (98,27,13), (65,6,32), (60,11,24), (73,9,11), (93,12,37), (67,20,14), (58,31,12), (62,28,0), (63,12,37), (91,18,16), (82,5,29), (66,35,14), (76,7,28), (63,37,25), (91,12,40), (82,37,23)	maduro	
136	(98,9,34), (67,2,17), (59,22,40), (59,7,29), (90,38,40), (90,18,18), (95,31,38), (70,13,6), (61,21,39), (60,39,22), (93,34,16), (78,27,16), (79,32,38), (59,12,17), (94,13,13), (69,22,10), (90,3,31), (64,29,6), (96,34,32), (97,12,24)	maduro	
137	(78,1,22), (87,24,9), (74,1,12), (92,5,37), (87,18,19), (89,29,27), (92,19,13), (59,21,28), (93,20,7), (85,16,28), (58,37,17), (78,6,0), (77,18,5), (64,7,3), (63,23,25), (100,11,1), (87,28,35), (76,18,2), (59,30,20), (84,0,28)	maduro	
138	(78,24,20), (82,32,1), (85,3,16), (61,4,0), (67,16,12), (68,14,11), (61,31,30), (91,27,30), (69,5,28), (84,32,31), (74,31,23), (81,4,7), (82,17,41), (94,29,29), (93,8,16), (90,20,25), (89,17,34), (65,39,35), (98,7,16), (68,19,27)	maduro	
139	(81,7,23), (66,26,28), (96,31,19), (76,24,16), (77,28,33), (84,39,15), (75,16,27), (99,28,17), (74,4,11), (74,28,13), (99,14,23), (70,8,34), (93,14,26), (63,6,33), (64,9,11), (82,24,30), (98,3,5), (90,14,40), (77,15,3), (99,37,13)	maduro	
140	(81,8,14), (87,1,36), (71,28,36), (89,29,21), (85,1,20), (69,12,36), (81,14,33), (95,10,10), (96,8,40), (76,8,33), (76,2,39), (92,4,22), (75,33,14), (97,11,34), (100,15,35), (82,20,19), (78,17,38), (68,32,21), (89,12,31), (85,2,37)	maduro	



141	(88,11,16), (60,8,22), (77,22,9), (75,28,7), (96,37,3), (78,35,18), (70,4,18), (69,6,17), (61,37,0), (71,22,30), (70,30,9), (67,13,28), (83,0,28), (63,5,24), (74,7,38), (70,25,24), (96,32,25), (92,10,15), (62,21,29), (94,12,10)	maduro	
142	(67,30,8), (88,23,14), (62,32,1), (86,16,19), (68,2,14), (92,9,4), (73,8,37), (91,9,26), (58,15,5), (65,23,35), (89,0,39), (99,17,28), (71,38,12), (58,3,6), (97,9,1), (95,15,14), (79,39,11), (63,39,7), (67,30,25), (81,6,4)	maduro	
143	(80,39,7), (76,13,4), (78,3,27), (63,19,14), (92,30,40), (83,13,5), (86,20,29), (77,38,18), (68,17,21), (62,38,29), (77,16,41), (59,8,34), (63,11,17), (66,11,10), (100,13,37), (76,29,5), (87,12,17), (95,20,29), (95,35,10), (85,26,26)	maduro	
144	(70,12,7), (75,11,1), (83,20,9), (91,23,36), (97,26,15), (74,16,6), (86,39,14), (81,2,8), (80,25,24), (91,30,39), (69,24,3), (68,33,25), (67,6,14), (84,35,6), (80,3,26), (74,22,29), (94,22,21), (82,12,21), (63,14,36), (85,26,3)	maduro	
145	(73,27,4), (58,2,38), (65,19,9), (81,14,34), (82,28,21), (71,21,37), (98,15,22), (73,17,8), (89,28,17), (66,35,17), (92,27,38), (83,26,15), (100,37,19), (76,22,18), (77,1,9), (81,10,17), (95,1,19), (74,9,19), (89,8,1), (74,39,33)	maduro	
146	(86,8,14), (80,3,37), (93,20,23), (92,5,2), (86,3,19), (61,25,18), (89,38,24), (60,9,22), (69,13,36), (75,26,25), (66,34,15), (87,36,21), (85,5,25), (72,36,8), (83,22,37), (82,11,6), (83,27,6), (59,22,41), (70,23,19), (66,25,7)	maduro	
147	(66,7,16), (86,7,31), (82,36,12), (80,5,41), (71,12,11), (94,15,4), (75,8,15), (93,19,30), (66,37,9), (94,37,38), (71,19,24), (93,32,33), (93,18,28), (99,17,6), (70,19,2), (76,34,22), (60,38,10), (95,11,25), (88,12,32), (93,9,13)	maduro	
148	(61,2,40), (90,27,9), (98,21,14), (90,14,21), (100,29,4), (81,20,22), (93,25,14), (93,32,30), (83,5,37), (98,25,12), (77,31,35), (68,14,23), (100,29,4), (93,22,11), (79,26,24), (63,35,9), (72,11,29), (85,33,31), (73,31,13), (62,37,32)	maduro	
149	(63,7,31), (78,7,16), (76,37,36), (96,16,32), (64,4,9), (86,20,15), (91,14,35), (68,11,37), (84,1,2), (83,10,4), (70,17,9), (86,14,13), (67,13,36), (99,37,40), (83,36,14), (93,24,7), (96,34,30), (94,9,7), (71,7,24), (85,13,21)	maduro	
150	(69,30,34), (61,16,14), (93,34,13), (62,35,41), (86,1,23), (70,17,17), (62,23,28), (85,4,36), (59,37,5), (94,3,23), (64,37,22), (59,36,30), (74,31,1), (81,2,2), (67,23,34), (100,28,37), (85,29,20), (72,7,16), (93,8,25), (82,18,17)	maduro	

5.4 Instalaciones necesarias para la ejecución del proyecto

5.4.1 Instalar Python 3.7 en Ubuntu 18.04

Abrir el terminal utilizando CTRL+ALT+T y digitar los siguientes comandos.

- `sudo apt update`
- `sudo apt updatesudo apt install software-properties-common`
- `sudo add-apt-repository ppa:deadsnakes/ppa`
- `sudo apt install python3.7`
- `python3.7 --version`

- Salida: *Python 3.7.3*



5.4.2 Crear entorno virtual Python

Un **virtualenv** o entorno virtual de Python es un ambiente creado con el objetivo de aislar recursos como librerías y entornos de ejecución del sistema principal o de otros entornos virtuales. Esto significa que en el mismo sistema puede tener instaladas múltiples versiones de una misma librería sin crear ningún tipo de conflicto.

5.4.2.1 Instalar virtualenv

Para instalar **virtualenv** es necesario tener instalado el sistema de gestión de paquetes (**PIP**) y luego ejecutar los siguientes comandos.

- `sudo pip3 install virtualenv`
- `virtualenv my_tesis_env -p python3`

```

cristobal@olano-pc: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
cristobal@olano-pc:~$ virtualenv my_tesis_env -p python3
Running virtualenv with interpreter /home/cristobal/anaconda3/bin/python3
Already using interpreter /home/cristobal/anaconda3/bin/python3
Using base prefix '/home/cristobal/anaconda3'
New python executable in /home/cristobal/my_tesis_env/bin/python3
Also creating executable in /home/cristobal/my_tesis_env/bin/python
Installing setuptools, pip, wheel...
done.
cristobal@olano-pc:~$
    
```

Figura 5.2: Creación del entorno virtual my_tesis_env

5.4.2.2 Activar / Desactivar entorno virtual (my_tesis_env)

- `source my_tesis_env/bin/activate`



```
cristobal@olano-pc: ~
Archivo Editar Ver Buscar Terminal Ayuda
cristobal@olano-pc:~$ source my_tesis_env/bin/activate
(my_tesis_env) cristobal@olano-pc:~$
```

Figura 5.3: Activación del entorno virtual my_tesis_env

Para desactivar el entorno virtual:

- deactivate

5.5 Instalar librerías Python para necesarias para el proyecto

Las librerías necesarias para el proyecto se encuentran registradas en un archivo llamado **requirements.txt** y utilizando el gestor de paquetes (PIP)

- **pip install -r /path/to/requirements.txt**

```
cristobal@olano-pc: ~/Escritorio/producto-tesis/apptesisPyQT/app-t...
Archivo Editar Ver Buscar Terminal Ayuda
(my_tesis_env) cristobal@olano-pc:~/Escritorio/producto-tesis/apptesisPyQT/app-t
esis$ pip install -r requirements.txt
```

Figura 5.4: Instalación de librerías en el entorno virtual my_tesis_env

