

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Prototipo de sistema electrónico para la adquisición de datos que permitan caracterizar el recurso solar aprovechable de una zona determinada

Informe de Proyecto de Graduación para optar por el título de
Ingeniero en Electrónica con el grado académico de Licenciatura

Josué Torres Moreno

Cartago, 12 de junio de 2017

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Proyecto de Graduación
Tribunal Evaluador

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Estudiante: Josué Torres Moreno

Nombre del Proyecto: *Prototipo de sistema electrónico para la adquisición de datos que permitan caracterizar el recurso solar aprovechable de una zona determinada*

Miembros del Tribunal



Ing. Sergio Arriola Valverde

Profesor lector



Dr. Ing. Carlos Meza Benavides

Profesor lector



Dr. Ing. Roberto Pereira Arroyo

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 12 de junio de 2017

Declaratoria de Autenticidad

Bajo esta declaración en la que indico que el presente proyecto de graduación ha sido realizado en su totalidad por mi persona, de forma que, utilizando literatura referente al tema y agregando mis conocimientos propios en cuanto a la materia ingenieril se refiere, contiene la estructura presente.

La bibliografía que se ha utilizado en forma de apoyo a sustentar este trabajo, ha sido indicada de forma adecuada por citas bibliográficas. Por consiguiente, asumo la responsabilidad de este trabajo de graduación realizado y por el contenido del mismo en el presente informe.



Josué Torres Moreno

Cartago, 12 de junio de 2017

Céd: 1-1528-0873

Resumen

En este proyecto se presenta el diseño, implementación y evaluación de un prototipo realizado para obtener una cifra aproximada del recurso solar aprovechable a partir de variables climatológicas de una zona determinada. Para ello se utilizan bibliotecas como OpenCV y K-means Clustering para el análisis de imágenes del cielo, además se hace uso de variables como temperatura, irradiancia global, humedad, entre otras, que permitan caracterizar el ambiente. Dicho prototipo presenta una primera etapa a la obtención de una cifra porcentual que permita la introducción por parte del ITCR a corroborar los cálculos teóricos que se plantean al predecir el recurso solar aprovechable en diferentes zonas.

Palabras clave: Cámara, DSP, Espectro, K-means, OpenCV, Panel Fotovoltaico, Potencia, Recurso Solar.

Abstract

In this project is introduced a design, implementation, and evaluation of a prototype created to obtain data that could be used to obtain approximately the amount of usable solar resource. For this application it's used libraries like OpenCV and K-means Clustering for image analysis of sky, also variables such as global irradiance, temperature, humidity, etc, to characterize the environment. This prototype established a first stage to ITRC to obtain a percentage values about usable solar resource to be corroborated with theoretical data in different zones.

Keywords: Camera,DSP, K-means, OpenCV, Photovoltaic Panel, Power, Spectre, Solar Resource.

a mis queridos padres

Agradecimientos

Primeramente quiero agradecer a quien me da vida todos los días, un Dios clemente y misericordioso que me ha permitido las oportunidades para realizar este proyecto, no es por otra cosa que la voluntad de él, darme dicha faceta de mi vida en la cual si bien han habido muchos errores, percibo más éxitos y crecimientos.

Al Dr.Ing.Carlos Meza Benavides, por brindarme la confianza y la oportunidad de este proyecto, el apoyo y conocimiento a través del mismo y su profesionalismo para manejar las situaciones que se fueron conllevando durante el desarrollo del mismo.

A mi familia, principalmente mis padres que me han enseñado tantas cosas en la vida, principalmente el lograr lo que me propongo y que no existe un límite. A mis queridos hermanos porque sé que siempre me han apoyado y de una u otra forma han brindado un grano de arena a mi estadía en la universidad y finalmente a mi hermana que se volvió mi confidente a través de este largo proceso.

Por última, a todas esas personas que son o fueron amigos, que se quedan o se van y que constantemente forjan quién soy y para donde me dirijo, algunas buenas y otras malas, pero me han enseñado que de todos siempre se aprende y se debe estar anuente a la enseñanza que pueda brindar cualquiera

Josué Torres Moreno

Cartago, 12 de junio de 2017

Índice general

Índice de figuras	iii
Índice de tablas	v
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.2.1 Objetivo General	2
1.2.2 Objetivos específicos	2
1.3 Restricciones	2
1.4 Supuestos	2
1.5 Contribuciones	3
1.6 Glosario y Acrónimos	3
1.7 Estructura del documento	3
2 Teoría Complementaria	4
2.1 Panel Fotovoltaico	4
2.1.1 Curvas Características de Corriente y Tensión para un PV	4
2.1.2 Modelo Matemático	5
2.2 Nubosidad	8
2.2.1 Definición y Procesos de Formación	8
2.2.2 Tipos y Clasificación de Nubes Troposféricas	9
2.3 Cámaras Digitales e Imágenes	11
2.3.1 Funcionamiento y Evolución	11
2.3.2 Tamaño de la Imagen	11
2.3.3 Apertura de Lente (Número F)	12
2.3.4 Sensibilidad a la Luz (ISO)	13
2.3.5 Velocidad de Disparo	13
2.3.6 Digitalización de una Imagen	14
2.3.7 Formato JPEG y RAW	15
2.4 RGB	15
2.4.1 Modelo del Color	15
2.4.2 Otros Modelos	16
3 Prototipo Electrónico	18
3.1 Equipo a utilizar y procesamiento para obtención de variables ambientales	19
3.1.1 Temperatura Ambiente y humedad	19

3.1.2	Irradiancia Global (W/m^2)	20
3.1.3	Temperatura de Panel Fotovoltaico	21
3.1.4	Toma de fotografías del cielo	21
3.2	Acondicionamiento de Señal	22
3.2.1	Analog to Digital Converter (ADC)	22
3.2.2	Cambio de Niveles Lógicos	23
4	Identificación de Nubes	24
4.1	Algoritmos Para Análisis de Nubosidad	24
4.1.1	Primer Algoritmo Propuesto	25
4.1.2	Segundo Algoritmo Propuesto	29
5	Resultados y análisis de Nubosidad	31
5.1	Resultados y análisis de algoritmo propuesto K-means Clustering	31
5.1.1	Cantidad de valores K para algoritmo 1 y 2	31
5.1.2	Porcentaje de nubosidad entre ambos algoritmos	34
5.1.3	Selección de algoritmo para prototipo	37
5.1.4	Comportamiento del algoritmo ante diferentes tipos de nubes	41
5.1.5	Comportamiento del algoritmo ante diferentes variables de la cámara	43
6	Resultados y análisis de Variables Ambientales	48
6.1	Resultados y análisis de Variables Ambientales	48
6.1.1	Irradiancia Global	49
6.1.2	Temperatura Ambiente	50
6.1.3	Temperatura de Celda	51
6.1.4	Humedad	52
7	Conclusiones y recomendaciones	53
7.1	Conclusiones	53
7.2	Recomendaciones	54
8	Bibliografía	55
A	Algoritmos	58
A.1	Algoritmo K-means Clustering	58
A.1.1	Comportamiento del algoritmo	59
B	Diseños Extras	61
B.1	Diseño Módulo Principal para Prototipo	61
B.2	Tensión V_{oc} (circuito abierto) y corriente I_{sc} (corto circuito) del panel fotovoltaico	63
B.3	Irradiancia en una celda fotovoltaica	64

Índice de figuras

2.1	Curva Característica de PV	5
2.2	Modelado de un PV	6
2.3	Clasificación General de Nubes	9
2.4	Clasificación Específica de Nubes	10
2.5	Apertura de Lente	13
2.6	Comportamiento de ISO	13
2.7	Comportamiento de la velocidad de Disparo	14
2.8	Tetraedro RGB	15
2.9	Espacio de color CMYK	16
2.10	Espacio de color HSL	17
2.11	Espacio de color HSV	17
3.1	Diagrama Ejecución Raspberry Pi 3	19
3.2	Divisor de Tensión	21
4.1	Tetraedro RGB	25
4.2	Espacio de Cilindros	26
4.3	Vista 2D del cilindro	27
5.1	Reconstrucción de Imagen a 64 colores con algoritmo K-means Clustering	31
5.2	Reconstrucción de imagen a diferentes valores "K"	32
5.3	Gráfico de Tiempo de ejecución vs Valores de "K"	33
5.4	Detección general de bordes con Algoritmo Canny	34
5.5	Detección de bordes de una nube con Algoritmo Canny	34
5.6	Binarización General de una Imagen	35
5.7	Binarización con algoritmo de Otsu de una Imagen	35
5.8	Binarización con algoritmo 1 propuesto	36
5.9	Binarización con algoritmo 2 propuesto	37
5.10	Análisis de Nube al 100% de nubosidad	38
5.11	Análisis de Nube al 0% de nubosidad	38
5.12	Análisis de Nube con aproximadamente 60% de nubosidad	39
5.13	Análisis de Nube con aproximadamente 5% de nubosidad	40
5.14	Análisis de Nube ante escenario de colores rojizos	40
5.15	Porcentaje de Error de Nubosidad en los algoritmos	41
5.16	Análisis de un Cúmulo	42
5.17	Análisis de un Estrato	42
5.18	Algoritmo Ejecutado en Imagen Tomada por Arducam OV2640	43

5.19	Comportamiento de Algoritmo Propuesto para Prototipo con respecto a la resolución	44
5.20	Comportamiento de Algoritmo Propuesto para Prototipo con respecto al "ISO" .	45
5.21	Comportamiento de Algoritmo Propuesto para Prototipo con respecto al "Shutter Speed"	46
5.22	Comportamiento de Algoritmo Propuesto para Prototipo con respecto al "F Number"	47
5.23	Imagen ante variables de la cámara	47
A.1	Pseudo Código de k-means	59
A.2	Algoritmo K-means Clustering	60
B.1	Diseño en Autocad para Módulo Principal	61
B.2	Diseño Implementado en Acrílico	62
B.3	Diseño de PCB	62

Índice de tablas

2.1	Modelos para un PV ideal, con pérdidas en serie R_s y con pérdidas en paralelo R_p	7
2.2	Abreviaciones para las clasificaciones de nubes.	11
2.3	Valores de tamaños de imagen para diferentes cantidades de Megapíxeles	12
2.4	Valores de tamaños de apertura de lente de forma decreciente	12
3.1	Especificaciones Apogee SP-110	20
3.2	Valores de ganancia para diferentes configuraciones del ADS1115	23
5.1	Relación de Número con ISO	45
5.2	Relación de Número con Velocidad de Disparo	46
5.3	Relación de Número con Número F	47
6.1	Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Irradiancia Global	49
6.2	Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Temperatura Ambiente	50
6.3	Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Temperatura de Celda	51
6.4	Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Humedad	52

Capítulo 1

Introducción

1.1 Motivación

El aprovechamiento del recurso solar por medio de tecnologías fotovoltaicas depende de múltiples variables ambientales tales como temperatura ambiente, humedad, irradiancia, nubosidad, entre otras. Existen diferentes tipos de tecnologías fotovoltaicas en el mercado y en desarrollo; cada una de ellas responde de forma distinta a las variables ambientales antes mencionadas. Para determinar de qué forma afectan éstas variables los distintos tipos de tecnologías fotovoltaicas es necesario cuantificarlas y asociarlas con la producción de energía. La correlación entre variables ambientales y generación fotovoltaica es útil para determinar cuál tecnología es más adecuada para un lugar determinado y para poder estimar de forma más precisa la energía fotovoltaica que se puede generar en un lugar determinado.

El presente proyecto desarrolla un prototipo electrónico que permite cuantificar diferentes variables ambientales necesarias para determinar el aprovechamiento del recurso solar en un lugar determinado con distintas tecnologías fotovoltaicas. Las variables que serán cuantificadas son las siguientes:

- Temperatura en la celda
- Temperatura ambiente
- Humedad
- Irradiancia
- Nubosidad

El principal aporte de este proyecto fue la determinación de un algoritmo que, a partir de imágenes, puede identificar el porcentaje de nubosidad presente en un instante determinado.

De esta forma, en este proyecto se desarrollan los circuitos acondicionadores para la cuantificación de variables ambientales y el desarrollo del sistema de identificación de nubosidad.

1.2 Objetivos

1.2.1 Objetivo General

- Diseñar un sistema electrónico que mida y registre datos que permitan caracterizar el recurso solar aprovechable de una zona establecida para determinada la potencia esperada en la generación de energía fotovoltaica.

1.2.2 Objetivos específicos

- Desarrollar algoritmos que permitan determinar radiación global, temperatura ambiente, humedad y temperatura de panel.
- Desarrollar un algoritmo que permita establecer un índice de nubosidad con base en imágenes del cielo.

1.3 Restricciones

- Trabajar con un sistema operativo de libre uso basado en Debian.
- Disponer de una cámara que sea capaz de obtener imágenes de resoluciones de al menos 1920x1080 píxeles, y que permita controlar la apertura del lente, sensibilidad a la luz y tiempo de disparo.
- Una plataforma embebida que posea GPIO (General Purpose Input Output) y disponga de protocolos de comunicación SPI y I2C.
- El panel fotovoltaico a caracterizar no produce más de 9,5 Amperios de corriente en corto circuito.
- El panel fotovoltaico a caracterizar no produce más de 42 Voltios de tensión en circuito abierto.
- La irradiancia máxima no supera lo 1350 W/m^2 .

1.4 Supuestos

- Las nubes poseen colores solo en la escala de grises.
- El sol es de color blanco puro, es decir, en una escala RGB unitaria tendría el valor de (1,1,1).
- El cielo siempre es de algún color de la escala de azules.
- No existen obstrucciones entre el lente de la cámara y el cielo.

1.5 Contribuciones

- Desarrollo de un algoritmo para la identificación de nubosidad a partir de imágenes.
- Prototipo electrónico que cuantifica y registra todas las variables ambientales necesarias para estimar la producción de energía fotovoltaica.

1.6 Glosario y Acrónimos

RTD: *Resistance Temperature Detector* (Resistencia Detectora de Temperatura)

DSP: *Digital Signal Processing / Digital Signal Processor* (Procesamiento Digital de Señales o Procesador Digital de Señales)

ISO: Sensibilidad a la luz

Número F: Apertura de la cámara

PV: *Photovoltaic Panel* (Panel Fotovoltaico)

V_{oc} : tensión en circuito abierto

I_{sc} : corriente en corto circuito

JPEG: *Joint Photographic Experts Group*

RAW: formato en "cruda" de una imagen

RGB: *Red-Blue-Green* (Rojo-Azul-Verde)

CMYK: *Cyan-Magenta-Yellow-Black* (Cyan-Magenta-Amarillo-Negro)

HSL: *Hue-Saturation-Lightness* (Hue-Saturación-Luminosidad)

GPIO: *General Purpose Input Output*

SPI: *Serial Peripheral Interface* (Protocolo de comunicación)

I2C: *Inter-Integrated Circuit* (Protocolo de comunicación)

PCB *Printed Circuit Board* (Tarjeta de Circuito Impreso)

1.7 Estructura del documento

El capítulo 2 muestra las bases teóricas complementarias al proyecto, de la cual se basan los algoritmos y las soluciones propuestas. En el capítulo 3 se proponen las soluciones para la obtención de las variables ambientales, en el capítulo 4 se proponen los algoritmos que permitan obtener el índice de nubosidad. En el capítulo 5 se validarán las soluciones propuestas implementadas en un código de Python. En el capítulo 6 se validarán las soluciones propuestas para obtener variables ambientales. Finalmente en el capítulo 7 se encuentran las conclusiones y recomendaciones para trabajos futuros.

Capítulo 2

Teoría Complementaria

2.1 Panel Fotovoltaico

Un panel fotovoltaico es un arreglo de celdas solares en serie y paralelo, que consisten en una unión p-n compuesta por materiales semiconductores. Las capas n semiconductoras son formaciones de silicio dopada por impurezas para generar una gran cantidad de electrones libres. Al incidir los fotones en materiales fotovoltaicos se liberan los electrones de las capas más superficiales generando una corriente eléctrica[21].

2.1.1 Curvas Características de Corriente y Tensión para un PV

El panel fotovoltaico produce su máxima corriente cuando la carga conectada no representa ninguna impedancia, es decir cuando se da un corto circuito I_{sc} entre sus terminales positivas y negativas, bajo esta condición la tensión del mismo es nula[16].

Caso contrario la máxima tensión de un panel fotovoltaico se da cuando la carga conectada representa un impedancia infinita, es decir cuando existe un circuito abierto V_{oc} entre sus terminales positivas y negativas, bajo esta condición la corriente del mismo es nula[17].

Un panel fotovoltaico se caracteriza por medio de su curva corriente vs tensión las cuales se llevan a cabo bajo condiciones estándar (STC), en la que establece una irradiancia de 1000 W/m^2 a un valor de 25 grados celsius de temperatura[18].

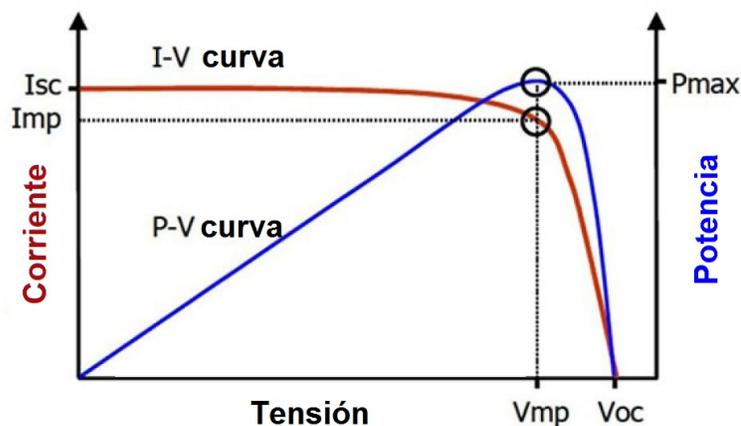


Figura 2.1: Curva Característica de corriente (A)-tensión (V) y potencia(P)-tensión(V) para un panel fotovoltaico.

De la figura 2.1 se extrae que el punto de máxima potencia del panel fotovoltaico se encuentra en determinado valor de corriente y tensión, siendo principalmente sujeto a la carga que este conectado [18]. De manera que la carga debe ser de un valor resistivo alto que se asemeje lo más posible al comportamiento de un circuito abierto, pero sin disminuir la corriente en grandes cantidades (característico para cada panel)[19]. Finalmente tenemos que la potencia es el producto de la corriente y la tensión como se describe en (2.1).

$$P_{max} = I_{max}V_{max} \quad (2.1)$$

2.1.2 Modelo Matemático

El modelo matemático de un panel fotovoltaico se obtiene a partir del circuito planteado como se muestra en la figura 2.2 de: una fuente de corriente en paralelo a un diodo y una resistencia, y en serie otra resistencia [19]. Este modelo incluye las variables características del panel:

- Dependencia de la temperatura, corriente de saturación del diodo I_s y fotocorriente I_g .
- Pérdida debidas al flujo de corriente R_s y pérdidas con referencias a tierra R_p .
- Número de celdas en el análisis n .

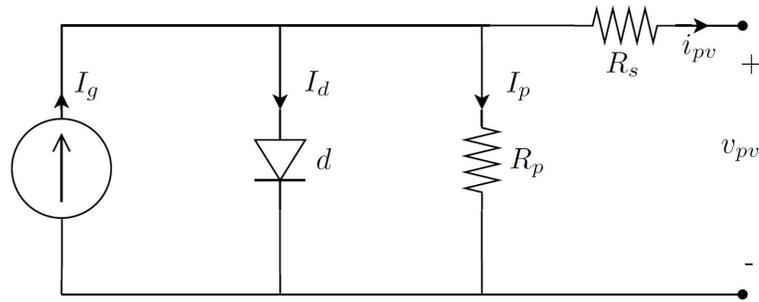


Figura 2.2: Modelo de un panel fotovoltaico, mediante una fuente de corriente, un diodo y dos resistencias.

Del modelo planteado en la figura 2.2 es posible deducir (2.2) y (2.3) las cuales describen las corrientes i_{pv} e I_g , de la siguiente manera [19]:

$$i_{pv} = I_g - i_d + i_p \quad (2.2)$$

$$I_g = 2i_{pv} + \frac{v_{pv} + i_{pv}R_s}{R_p} - I_s + I_s e^{\frac{v_{pv} + i_{pv}v_{pv}}{nv_t}} \quad (2.3)$$

Despejando i_{pv}

$$i_{pv} = \frac{1}{2} \left[I_g - \frac{v_{pv} + i_{pv}R_s}{R_p} + I_s - I_s e^{\frac{v_{pv} + i_{pv}v_{pv}}{nv_t}} \right] \quad (2.4)$$

la corriente que fluye por las terminales de un generador fotovoltaico está determinada por tres corrientes tales como:

- Corriente generada debido al efecto fotoeléctrico I_g
- Corriente de pérdida debido a la juntura p-n i_d
- Corriente de pérdida de naturaleza resistiva i_p

Las siguiente suposiciones se consideran para el comportamiento estático del generador fotovoltaico

- I_g depende de la irradiancia (S), pero no depende de la tensión en las terminales del generador fotovoltaico v_{pv}
- i_p e i_d dependen de la tensión v_{pv}
- i_p depende de la temperatura (T)

Con base a los supuestos anteriores, en (2.5) se describe la i_{pv} de manera que:

$$i_{pv}(v_{pv}, T, S) = I_g(v_{pv}) - i_d(v_{pv}, T) \quad (2.5)$$

Se obtiene un modelo general del comportamiento estático de un panel fotovoltaico:

$$i_{pv}(v_{pv}) = KS - G_p v_{pv} - G_p i_{pv} R_s - i_d \quad (2.6)$$

$$i_{pv} = KS - G_p v_{pv} + I_s(T) - G_p i_{pv} R_s - I_s(T) e^{\frac{v_{pv}}{v_t}} e^{\frac{i_{pv} R_s S}{v_t}} \quad (2.7)$$

Para otros modelos de complejidad y precisión distintas, estarán dados por los valores que se le den a las funciones i_d e i_{pv} , algunos ejemplos de ello se muestran en la tabla 2.1.

Tabla 2.1: Modelos para un PV ideal, con pérdidas en serie R_s y con pérdidas en paralelo R_p

Modelo	i_g	i_p	i_d
1	KS	-	$I_s(T) \left[e^{\frac{v_{pv}}{v_t}} - 1 \right]$
2	KS	$G_p v_{pv}$	$I_s(T) \left[e^{\frac{v_{pv}}{v_t}} - 1 \right]$
3	KS	-	$I_s(T) \left[e^{\frac{v_{pv} + i_{pv} R_s}{v_t}} - 1 \right]$
4	KS	$G_p v_{pv} + G_p i_{pv} R_s$	$I_s(T) \left[e^{\frac{v_{pv} + i_{pv} R_s}{v_t}} - 1 \right]$

2.2 Nubosidad

La nubosidad afecta la intensidad y el espectro de la radiación solar, debido a ello es importante comprender cómo determinar su efecto en la radiación. A continuación se presenta la teoría relevante sobre las nubes.

2.2.1 Definición y Procesos de Formación

De acuerdo con la Organización Meteorológica Mundial(OMM) la nubes se definen como hidrometeoros formados de una masa visible a base de cristales de nieve o gotas de agua microscópicas suspendidas en el aire[1].

Estas poseen sus colores blancos basados en la teoría de Mie que da una solución a la dispersión de la radiación electromagnética en estas partículas con mayor tamaño a la longitud de onda de los colores de la luz, la cual es la razón por la cual se suele observar que estas son blancas[2].

Se estableció por el científico del siglo XVIII Luke Howard (pionero de la meteorología) las cuatro clasificaciones generales de las nubes como se ilustra en la figura 2.3 [3]:

- Cirros: del latín hace referencia a 'rizo', caracterizado por bandas delgadas, finas, acompañadas por copetes.
- Estratos: del latín hace referencia a 'extendido', caracterizada por poseer capas horizontales con una base uniforme.
- Nimbos: del latín hace referencia a 'nube de lluvia', caracterizado por su color gris oscuro y capaz de formar precipitación.
- Cúmulos: del latín hace referencia a 'por acumulación', caracterizado por bordes delimitados y poseer la textura de "algodón".

Las nubes se pueden formar de tres procesos distintos, descritos a continuación [4]:

- Mediante convección térmica: La evaporación superficial de los ríos, lagos y mares, y la transpiración de las plantas acumulan humedad en el aire, cercano al suelo cuando se calienta, asciende creando una corriente térmica. Al ascender se dilata y enfría, condensándose en gotas diminutas y formando nubes. El proceso continúa y más aire caliente y húmedo alimenta la nube, dándole mayor altura, hasta alcanzar una altura máxima limitada por las corrientes frías, de este comportamiento se empieza a ensanchar la nube y a tomar forma de yunque, finalmente mediante las precipitaciones esta se subdivide en algunos tipos de nubes antes mencionados.
- Mediante ascenso orográfico: el aire cálido y húmedo que procede del mar es obligado por la orografía a ascender, durante el ascenso este se enfría y por ende se condensa en pequeñas gotas, formando una nube. De forma que el aire que desciende por las montañas es seco, ya que su humedad yace en la nube creada.
- Mediante convección de un frente: cuando un frente frío choca con una masa de aire caliente y húmedo forma una condensación al bajar su temperatura formando una nube

en altura (cumulonimbo). Además existe el proceso inverso, es decir, un frente cálido y húmedo chocando una masa de aire frío, pero en cambio esta masa asciende sobre el aire frío debido a su baja densidad. Este se enfría y comienza a condensarse formando una nube bajas y oscuras con capacidad de precipitación (nimboestrato).

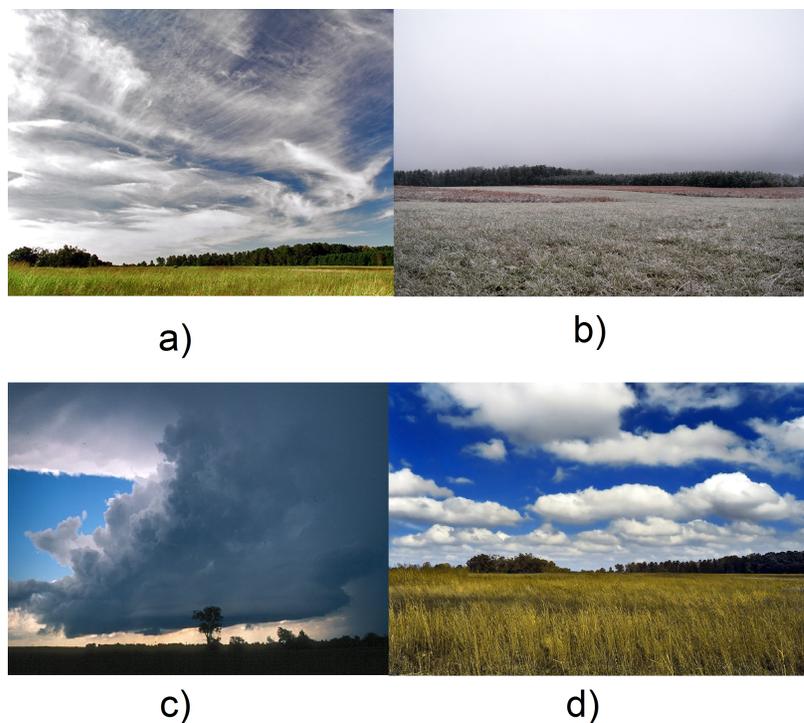


Figura 2.3: a) Nube de clasificación Cirros, b) Nube de clasificación de Estratos, c) Nube de clasificación Nimbos, d) Nube de clasificación Cúmulos

2.2.2 Tipos y Clasificación de Nubes Troposféricas

La clasificación de las nubes está basada en características visuales, proveniente de la OMM y recogida por parte del Atlas Internacional de las Nubes [1].

A continuación se mencionan los géneros en los que se subdivide los 10 tipos de nubes reconocidos por este ente, dentro de los cuales ocho tienen una forma paralela a la superficie terrestre (estratiformes) y dos tienen forma vertical, es decir, perpendicular a la superficie terrestre (cumuliformes):

- Género cúmulos/cumuliformes (Cúmulos): nubes de desarrollo vertical.
- Género Estratos/estratiformes (Estratus, Altoestratus, Cirroestratus, Nimbostratus): son nubes que están compuestas de capas o estratos superpuestos.
- Género Nimbos/cumulonimbiformes (Cumulonimbos): nubes de desarrollo vertical con capacidad de precipitación.
- Género Cirros/cirrififormes (Cirros): nubes blancas muy elevadas y de aspecto fibroso.

- Género Estratocúmulos/stratocumuliformes (Estratocúmulos, Altocúmulos, Cirrocúmulos): estas nubes poseen un desarrollo vertical limitado en forma de roolos u ondulaciones.

Estas nubes no solo son clasificadas por su características físicas, sino también por familias según su altura:

- Familia A: gran altura, por encima de los 5 km.
- Familia B: altura media, de 2 a 5 km.
- Familia C: baja altura, a menos de 2 km.
- Familia D: desarrollo vertical, a menos de 3 km.

En la figura 2.4 podemos observar un resumen de los géneros de nubes con respecto a la familia en que se encuentran (altura) de una forma compacta y fácil de entender:

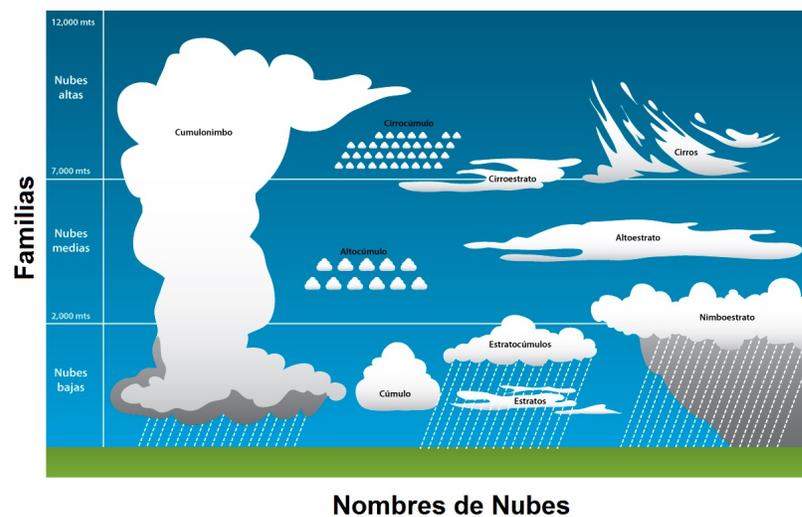


Figura 2.4: Clasificación de las nubes en base a su apariencia, características físicas y familia.

Para mayor facilidad establece la tabla 2.2 con abreviaciones de los diez diferentes tipos de nubes según la clasificación anterior, de forma que permita posteriormente referirse a ellos sin escribir su nombre completo.

Tabla 2.2: Abreviaciones para las clasificaciones de nubes.

Nombre de Nube	Abreviación
Cirrus	Ci
Cirrocúmulos	Cc
Cirroestratos	Cs
Altoestratos	As
Altocúmulus	Ac
Estratos	St
Estratocúmulos	Sc
Nimboestratos	Ns
Cúmulos	Cu
Cumulonimbos	Ac

2.3 Cámaras Digitales e Imágenes

La forma más directa de identificar la nubosidad es por medio de imágenes del cielo. Las imágenes depende de las características y forma en que operan las cámaras, las cuales se discuten a continuación.

2.3.1 Funcionamiento y Evolución

Las cámaras han sido toda una revolución por parte del ser humano en cuanto a la interpretación del espectro electromagnético. Hay cámaras capaces de captar imágenes de diferentes secciones del espectro electromagnético, desde infrarrojo, espectro visible por el ser humano, hasta ultravioleta, permitiendo mejorar la percepción. Es en el año de 1975 la primera cámara digital es introducida por Cromemco llamada Cyclops[6].

Una cámara digital posee generalmente 2 tipos de sensores para captura de imágenes ya sea un CCD (Charged-Couple Device) o un CMOS-APS(Complementary Metal-Oxide Semicondutor Active Pixel Sensor), los cuales delimitan la capacidad de resolución de la cámara. Hoy en día hay empresas dedicadas a la creación de DSP (Digital Signal Processors) que permiten mejorar la recepción de datos analógicos mediante su medida, compresión y filtrado[7].

Una cámara puede ser clasificada en cuanto a la cantidad de megapíxeles, apertura de los lentes, ISO, velocidad de disparo, cantidad de colores percibidos, formatos de escritura, montura de lente y velocidad de procesamiento.

2.3.2 Tamaño de la Imagen

Un píxel es la menor unidad homogénea en color que forma parte de una imagen digital [8]. Estos se encargan de componer las imágenes que son visualizadas en la pantalla de una computadora, de forma que tienen un valor de un byte (256 valores posibles) y por ello se observan cifras

entre los números 0 a 255. Estos a su vez utilizan el modelo RGB para poder representar la proporción de las 3 componentes binarias de cada píxel.

En la tabla 2.3 se observa el comportamiento cuando se menciona a una cantidad X de Megapíxeles.

Tabla 2.3: Valores de tamaños de imagen para diferentes cantidades de Megapíxeles

Megapíxeles	Tamaño Imágen 3:2(Píxeles)	Tamaño Imágen 4:3(Píxeles)
0,3	671x447	632x474
1	1224x816	1155x866
2	1733x1155	1633x1225
3	2121x1414	2000x1500
4	2450x1633	2309x1732
5	2739x1826	2581x1936
6	3000x2000	2828x2121
8	3464x2309	3265x2449
10	3873x2582	3652x2739
12	4242x2828	4000x3000
14	4583x3055	4320x3240
16	4899x3266	4619x3464
18	5196x3464	4899x3674
20	5477x3651	5164x3873
22	5745x3830	5416x4062
24	6000x4000	5657x4243

2.3.3 Apertura de Lente (Número F)

Las cámaras modernas poseen lentes con una apertura variables, esta limitación está determinada por el fabricante y hace referencia al mismo como relación focal o número F siendo el mismo la relación de apertura del diafragma en el objetivo, este en algunos casos no es variable [9]. De una manera más sencilla de explicar la apertura del lente es como la pupila del ojo humano, a mayor apertura mayor luz incide.

La escala para número F se ajusta de modo que cada valor vaya representando la mitad de la intensidad luminosa que el anterior, para ello se divide el área de la pupila de entrada entre 2, y su diámetro entre $\sqrt{2}$ de manera que se obtiene un comportamiento mostrado en la tabla 2.4 y se puede relacionar directamente con un comportamiento de disminución de diámetro como se ilustra en la figura 2.5.

Tabla 2.4: Valores de tamaños de apertura de lente de forma decreciente

f	Número	1	1,4	2	2,8	4	5,6	8	11	16	22	32	45
---	--------	---	-----	---	-----	---	-----	---	----	----	----	----	----

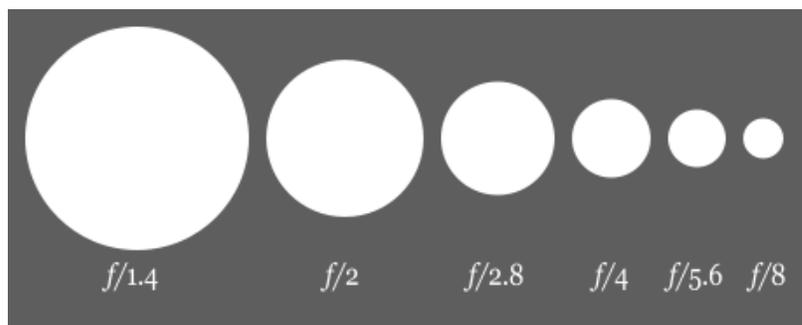


Figura 2.5: Apariencia del lente de forma decreciente en su número f

2.3.4 Sensibilidad a la Luz (ISO)

El ISO es la sensibilidad que tiene la cámara con respecto a la luz incidente. Para esto el sensor de la cámara se ajusta para percibir la luz de otra manera. Este se comporta en valores de doblar la sensibilidad, es decir, 100-200-400-800-1600-3600 y así sucesivamente. Se puede explicar mediante el siguiente ejemplo: se tiene un haz de luz con un valor 125, este se considera un gris intermedio en la escala de grises para un ISO de 1600, si se cambia el ISO a 100 ese mismo haz de luz ahora tendrá un valor de 25 en la escala de grises (acercándose al valor del negro que es 0) o si se cambia el ISO a 12800 ese mismo haz de luz ahora tendrá un valor de 240 en la escala de grises (acercándose al valor del blanco que es 255), de esta manera el ISO cambia los valores dados dentro de la cámara para un mismo haz de luz.

Esta es la característica más cara que se puede encontrar en el cuerpo de una cámara y está directamente relacionada en las fotos con la velocidad de disparo, provocando que cuando este incrementa la velocidad de disparo se disminuye [10]. Pero el precio de la sensibilidad a la luz produce una expresión a la que llamamos ruido, como se ilustra en la figura 2.6.

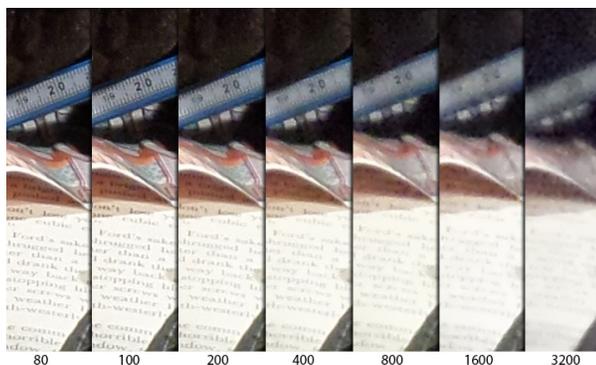


Figura 2.6: Comportamiento del ISO en una imagen bajo las mismas condiciones

2.3.5 Velocidad de Disparo

La velocidad de disparo es la cantidad de tiempo que el sensor de imagen va a ser expuesto a la luz. De esta manera se pueden obtener diferentes efectos sobre la imagen. Ya sea a una velocidad muy alta de 1/4000 segundos que permite captar las gotas de agua en movimiento

como una imagen estática, hasta una velocidad muy baja de 1 min que permite realizar las imágenes de *StarTrails* donde se realizan imágenes con el movimiento de las estrellas. Al estar directamente relacionado con el ISO, se debe compensar para que las imágenes tengan la forma óptima o deseada [10]. El comportamiento de la velocidad de disparo se ilustra en la figura 2.7.



Figura 2.7: Comportamiento de la velocidad de disparo en una imagen bajo las mismas condiciones de ISO y apertura de lente [12]

2.3.6 Digitalización de una Imagen

Para la digitalización de imágenes existen tres procesos que pueden variar dependiendo del sensor de luz que se tiene:

- El primero es por una única exposición: en la cual existen tres sensores para cada color (Rojo, Azul y Verde), de forma que cada uno captura la información necesario y por medio de un DSP se obtiene una cuantización de los datos y se exportan en manera binaria que puede ser interpretada
- El segundo se basa en multi-exposiciones: en este se tienen varias exposiciones pero un único sensor que va recaudando la información para cada color (Rojo, Azul y Verde) y transmite los datos a un DSP que se encarga de su interpretación.
- El tercero es por exploración: se tiene un sistema capaz de explorar el espacio de luz, ya sea de forma lineal o tridimensional, como es el caso de los escáneres o cámaras rotativas respectivamente. De manera que tiene sensores receptores para los colores y así envían los datos a un DSP que se encarga de su interpretación.

De estos tres métodos el más común utilizados por las cámaras es el de una única exposición. Hoy en día existen sensores capaces de obtener imágenes con una cantidad de colores mayor a 256 colores.

La información capturada por los sensores debe ser interpretada y comprimida en un formato que se pueda utilizar. En general para el uso normal se tienen formatos como TIFF, PNG o JPEG, sin embargo, las cámaras suelen aportar un último que se le conoce como *RAW* o *crudo* en español, que aporta no solo colores, sino que también transparencia de los mismos y profundidad.

2.3.7 Formato JPEG y RAW

El formato digital JPEG (Joint Photographic Experts Group) es un estándar de compresión y codificación de archivos e imágenes. Éste utiliza un algoritmo de compresión con pérdida, es decir, intenta representar la mayor cantidad de imagen con la menor cantidad de información. Para el caso de JPEG intenta reducir los espacios de color a colores más comunes o que no se puedan percibir por el ojo humano como una diferencia, de modo que el peso en información de la imagen disminuye.

Estos se pueden dividir en dos fenómenos visuales más exactos, es mucho más sensible en la luminancia que en la crominancia y el brillo se denota con mayor facilidad en zonas homogéneas. Conforme más comprime se obtiene más pérdida de información, degradando cada vez más la imagen. Para la codificación el formato utiliza 24 bits por píxel (ocho para cada color Rojo, Azul y Verde). A pesar de que se utilizan otros modelos de color, no suelen ser muy utilizados en la compresión de dichas imágenes [13].

Para el formato digital RAW cada fabricante de cámaras tiene su propia extensión de archivo. Este formato contiene los valores tal y como los captura el sensor de luz de la cámara. Su compresión de datos no tiene pérdida y es por eso que cada píxel 48 bits.

2.4 RGB

2.4.1 Modelo del Color

El modelo RGB (por sus siglas Red-Blue-Green en inglés) es el sistema más común de imágenes utilizado en la computación. Su característica principal es la *aditiva*, la cual se fundamenta en la mezcla de estos tres colores para formar los demás 15. La cantidad de colores suele estar expresada en una forma unitaria, hexadecimal o de valores en 0 a 255.

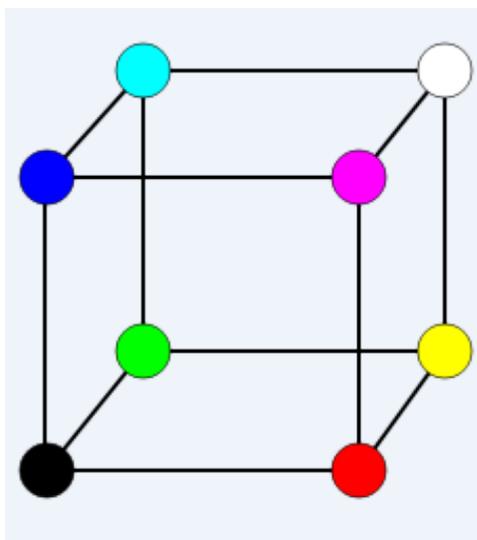


Figura 2.8: Tetraedro Con vértices de colores característicos de un sistema RGB.

La figura 2.8 muestra los valores que se deben tener para poder obtener diferentes colores. El primer vértice es el negro que contiene el mínimo valor posible (0,0,0) y el blanco que posee el mayor valor posible (255,255,255), luego cada distribución de cada vector, rojo (255,0,0), azul (0,0,255), verde (0,255,0) y por último de colores como el cyan (0,255,255), amarillo (255,255,0), magenta (255,0,255). Así cada valor máximo del cubo a sido representado en sus debidos vértices, y de aquí se obtiene otros modelos de color como el CMYK[26].

Para efectos de informática, el sistema RGB puede ser expresada de forma decimal, según descrito por (2.8).

$$R * 65536 + G * 256 + B = ValorDecimal \quad (2.8)$$

También se puede tener su valor hexadecimal el cual es mucho más común, siendo un valor de 6 espacios en donde el rojo se asigna a los primeros espacios (0 y 1), el verde a los segundos espacios (2 y 3) y el azul a los terceros espacios (4 y 5). Por ejemplo 00x000000 representa el negro y 00xFFFFFFFF el blanco. La siguiente expresión indica como puede construir un valor hexadecimal RGB, en donde \oplus significa concatenación):

$$Hex(R) \oplus Hex(G) \oplus Hex(B) = ValorHexadecimal \quad (2.9)$$

2.4.2 Otros Modelos

Modelo CMYK

El modelo CMYK está basado en la substracción de los colores Amarillo, Cyan, Magenta y Negro (Cyan-Magenta-Yellow-Key)[15]. Este modelo permite las técnicas que se utilizan en la impresión, mediante el análisis de cual longitud de onda se refleja y cual se absorbe, se crea un espacio capaz de ello. Siendo el negro la ausencia de los colores, pero en este espacio la contribución de todos, conforme se le va quitando una contribución se obtienen otros colores. Esto se ilustra por medio de la figura 2.11[25].

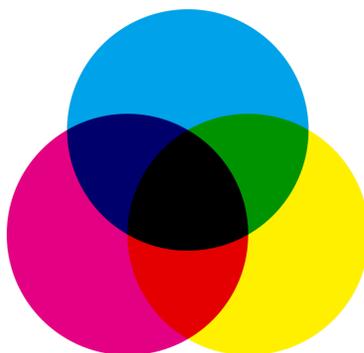


Figura 2.9: Comportamiento de substracción de color en el modelo CMYK

Modelo HSL

Este modelo está basado en el matiz, saturación e intensidad luminosa (Hue, Saturation, Lightness) del color. Representado como un cono doble en donde sus vértices son el blanco y el negro ilustrado en la figura 2.10[25].

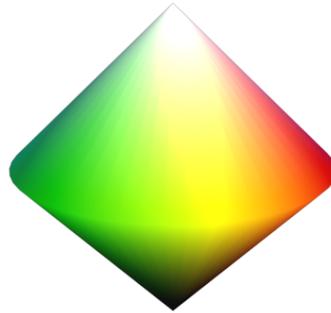


Figura 2.10: Cono doble del modelo HSL

La saturación está determinada por la división del croma(color) por el mayor croma de dicha luminosidad.

La luminosidad es el promedio entre el mayor y el menor componente de RGB.

Modelo HSV

Este modelo está basado en el matiz, saturación y valor (Hue-Saturation-Value), muy parecido en comportamiento al modelo HSL, solo que este contiene la capacidad de poder obtener una versión bidimensional para escoger colores como en la figura 2.13[25].

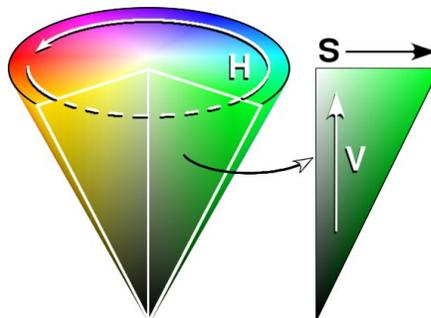


Figura 2.11: Cono del modelo HSL y su espacio bidimensional.

Capítulo 3

Prototipo Electrónico

Se estipula que el prototipo debe ser un módulo capaz de obtener variables ambientales y tomar imágenes al cielo. Debe ser independiente de cualquier otra unidad de procesamiento y además poseer libertad de agregarle complejidad al mismo (agregar módulos nuevos aparte y capacidad de mejorar su programación.)

Para solventar todas las necesidades y obtener datos con una buena precisión se optó por utilizar una Raspberry Pi 3 model B. Esto debido a que posee características que dan facilidad al proyecto, la cuales se enlistan a continuación:

- Se puede operar con un sistema operativo basado en Debian. Esto permite que se realicen algoritmos y pruebas en otras sistemas operativos basados Debian, en este caso en un computador local, así se pueden analizar algoritmos que requieran más tiempo ejecución, mayor memoria y así clasificar qué es viable para las capacidades de las raspberry pi.
- Protocolos de comunicación SPI (Serial Peripheral Interface), I2C (Integrated Integrated Circuit) y SSH.
- Soporte en línea y años de estar en el mercado, lo cual facilita la obtención de información ante problemas y paquetes necesarios para programación.
- Poseer bluetooth y Wi-Fi integrados, además de varios núcleos de procesamiento (cuatro) y 40 pines de GPIO (General Purpose Input Output).
- Dimensiones igual o menores a 10 cm x 8 cm que minimizan el espacio del prototipo final.
- Bajo costo y adaptación a diferentes módulos open hardware que son compatibles con la misma.

En la figura 3.1 se puede observar un diagrama de bloques en dónde se explica el comportamiento del prototipo electrónico, con sus variables de entrada, procesos y datos guardados dentro de la misma.

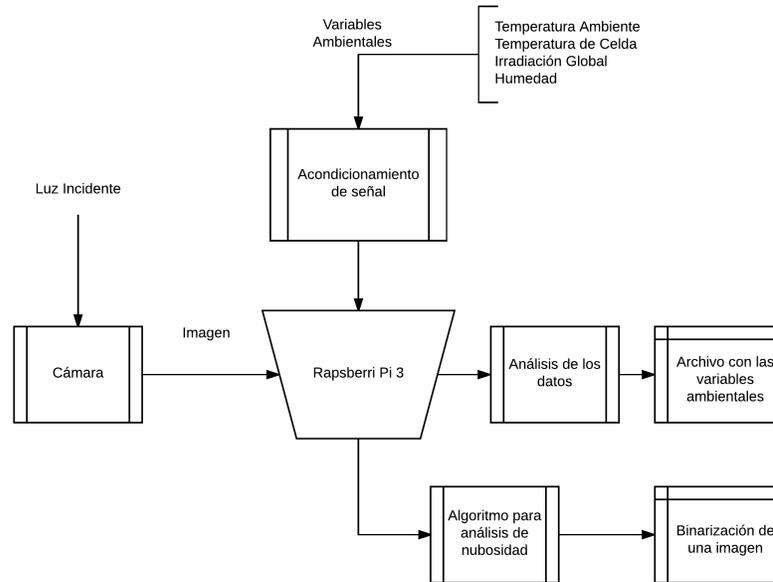


Figura 3.1: Diagrama de ejecución Raspberry Pi 3

3.1 Equipo a utilizar y procesamiento para obtención de variables ambientales

Se procede a describir las variables a medir, como obtenerlas por medio de un hardware determinado a utilizar y el comportamiento de los programas escritos en Python y otros en C, ejecutados en la Raspberry Pi 3.

3.1.1 Temperatura Ambiente y humedad

Para determinar la temperatura ambiente y la humedad se utiliza el sensor SHT-31D de Adafruit, el cuál está en el rango deseado de operación para este proyecto de un porcentaje de error de $\pm 2\%$ en humedad relativa y una diferencia en la temperatura ambiente de $\pm 0.3^\circ\text{C}$. Este se comunica con la plataforma embebida por medio del protocolo de comunicación I2C con la dirección 0X44 [31].

En el algoritmo 1 se describe el funcionamiento del algoritmo de adquisición de datos mediante un pseudocódigo para la obtención de datos:

Datos: Temperatura ambiente y Humedad

Resultado: Variables Promediadas de Temperatura ambiente y humedad
initialization;

```

mientras programa en ejecución hacer
  | si dirección del sensor existe entonces
  |   verificar estado del sensor;
  |   si estado del sensor malo entonces
  |     | mostrar mensaje de error;
  |     | cerrar programa;
  |   en otro caso
  |     | mientras n cantidad de veces hacer
  |       | tomar datos de temperatura y humedad;
  |       | guardarlos en un arreglo;
  |     | fin
  |       | promediar n cantidad de valores del arreglo;
  |       | guardar datos en un archivo;
  |     | fin
  |   en otro caso
  |     | cerrar programa
  |   fin
  |   esperar 5 min;
fin

```

Algoritmo 1: Pseudocódigo para Sensor SHT-31D

3.1.2 Irradiancia Global (W/m^2)

La irradiancia global se puede obtener mediante el sensor Apogee SP-110, que posee una auto-alimentación, fácil conexión de solo 3 cables a utilizar (Tensión positiva, tierra y negativo), y características deseadas como se pueden observar en la tabla 3.2[33]:

Tabla 3.1: Especificaciones Apogee SP-110

Característica	Valor
Sensibilidad	0.2 mV por W/m^2
Factor de calibración	5.0 W/m^2 por mV
Incertidumbre de Calibración	$\pm 5\%$
No-linealidad	< de 1% hasta los 1750 W/m^2
Inestabilidad a largo plazo	menos del 2% por año
Campo de Visión	180 °
Rango espectral	360 a 1120 nm
Reespuesta a temperatura	0.04 \pm 0.04% por °C
Peso	90g

3.1.3 Temperatura de Panel Fotovoltaico

La temperatura del panel se mide utilizando una termo-resistencia (RTD), en este caso específico se utilizará la Thermoprozees Austria RTD Type Pt1000/2L/A/AS5000mm, la cuál consiste de resistencia de platino de 1000Ω a 0°C y descrita por la ecuación (3.1):

$$R(T) = R(0^\circ\text{C})(1 + \alpha T) \quad (3.1)$$

Donde el valor de $R(0^\circ\text{C}) = 1000\Omega$, $\alpha = 0.00392$ y T es de temperatura de la celda. Reacomodando (3.1) se obtiene (3.2):

$$T = \frac{\frac{R(T)}{R(0^\circ\text{C})} - 1}{\alpha} \quad (3.2)$$

Sin embargo para determinar la resistencia característica $R(T)$ se sugiere el circuito de la figura 3.2, de donde se medirá don tensiones, la primer la tensión de entrada y la segunda la tensión en la conexión entre una resistencia conocida (1000Ω por ser el valor a 0°C) y la RTD, aplicando un divisor de tensión tal como describe (3.3):

$$V_{out} = \frac{V_{in}R_1}{R_1 + R_2} \quad (3.3)$$

Donde $R_1 = 1000\Omega$ y $R_2 = RTD$. En la figura 3.2 se muestra el circuito a utilizar, sin embargo la diferencia entre las tensiones V_{in} y V_{out} serían muy bajas entre si y necesita de un amplificador de instrumentación para obtener estos valores de diferencia. La problemática poder cuantificar niveles de tensiones muy bajos (por debajo de los 100 mV) queda resuelta gracias al ADC con amplificador de ganancia que se utiliza, explicado en la sección 3.2.

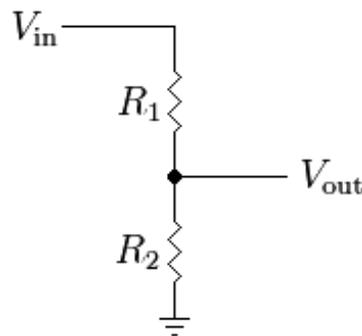


Figura 3.2: Divisor de Tensión

3.1.4 Toma de fotografías del cielo

Para obtener imágenes del cielo se utiliza la cámara OV2640 la cual posee una resolución de 2 megapíxeles, cambios en la resolución, protocolos SPI y I2C para comuniación con la Raspberry Pi 3, dimensiones de $7.62 \times 5.08 \times 2.54\text{ cm}$, con lente intercambiable y un precio inferior a los 30 dólares. A esta cámara se le intercambio su lente original por uno de ojo de pez, el cual permite una visión de 170 grados para obtener una mejor imagen del cielo[35].

Las características antes mencionadas permiten el fácil reemplazo de la cámara, como también su constante intercambio de lentes para corroborar diferentes resultados en la toma de imágenes. Existe la limitación que el enfoque se realiza manualmente, y este es muy sensible, por lo que puede producir problemas para un sistema totalmente aislado sin verificación constante.

La cámara tiene un proceso de comportamiento que se explica en el algoritmo 2 mediante un pseudocódigo.

```

Datos: Solicitud de imagen
Resultado: Imagen capturada por el sensor
initialization;
mientras programa en ejecución hacer
  si dirección del sensor existe entonces
    iniciar protocolo SPI para comunicarse con la cámara;
    si existe comunicación exitosa entonces
      activar sensor de imagen;
      captura de datos y enviarlos por I2C;
      indicar final de comunicación;
      guardar imagen;
    en otro caso
      | indicar error de comunicación y cerrar programa
    fin
  en otro caso
    | cerrar programa
  fin
esperar 15 min;
fin

```

Algoritmo 2: Comportamiento de programación para arducam mini OV2640

3.2 Acondicionamiento de Señal

Existen varios tipos de acondicionamientos de señal, ya sean: amplificación, filtrado, linealización, entre otros. Utilizados para una correcta lectura de los datos que se están recibiendo. Para el prototipo es necesario obtener amplificaciones de señal y cambios de niveles lógicos en los GPIO de la Raspberry Pi 3, para ello se utilizan los siguientes módulos.

3.2.1 Analog to Digital Converter (ADC)

Debido a que las variables bajo análisis están en diferentes rangos de tensión, se utilizará un ADC con un amplificador de ganancia. Se hace uso del ADS-1115 de Adafruit, con una resolución de 15 bits y 1 bit de signo, cuatro canales de lectura A0 A1 A2 A3, además de la posibilidad de ser conectados simultáneamente por medio del protocolo de comunicación I2C dado que existen 4 posibles direcciones y de posee un amplificador de ganancia que permite medir tensiones muy pequeñas. En la tabla 3.2 se muestran los valores que se pueden configurar con este ADC y un algoritmo que detalla el comportamiento de la programación implementada para el mismo[32].

Tabla 3.2: Valores de ganancia para diferentes configuraciones del ADS1115

Ganancia	Rango de Tensión (V)	Valor por bit (μV)
2/3	± 6.144	187.5
1	± 4.096	125
2	± 2.048	62.5
4	± 1.024	31.25
8	± 0.512	15.625
16	± 0.256	7.8125

Datos: Tensión analógica en el canal

Resultado: Valores de tensión digital

initialization;

mientras programa en ejecución **hacer**

si dirección del sensor existe **entonces**

 verificar estado del sensor;

si estado del sensor malo **entonces**

 mostrar mensaje de error;

 cerrar programa;

en otro caso

 verificar ganancia de cada canal;

mientras n cantidad de veces **hacer**

 tomar datos de los canales;

 guardarlos en un arreglo;

fin

 promediar n cantidad de valores del arreglo;

 guardar datos en un archivo;

fin

en otro caso

 cerrar programa

fin

esperar 5min;

fin

Algoritmo 3: Comportamiento de programación para ADS1115

3.2.2 Cambio de Niveles Lógicos

Los niveles de tensión de la Raspberry Pi 3 es de 0V (bajo) o 3.3V (alto), ya que varios módulos se comunican con niveles de lógicos de 0V (bajo) y 5V (alto), se propone utilizar un circuito integrado bidireccional de cambio de niveles lógicos. Para ello el BS138 solventa la necesidad, este convierte el valor de 3.3V a 5V y viceversa para lograr la comunicación con los diferentes módulos a través de 4 canales bidireccionales.

Capítulo 4

Identificación de Nubes

La nubosidad al afectar directamente la intensidad y el espectro de la radiación solar, provoca que el recurso solar aprovechable en una zona cambie. Para determinarla se toman imágenes del cielo y se obtiene por medio de algoritmos un índice de nubosidad que facilite comprender el comportamiento que existe. A continuación se dará una explicación del comportamiento de los algoritmos propuestos para el análisis de nubosidad.

4.1 Algoritmos Para Análisis de Nubosidad

El análisis de nubes es toda una materia en desarrollo, y se han implementado varios algoritmos, en este caso, el más común es utilizar el algoritmo KNN (K Nearest Neighbors), con el cual se obtienen imágenes de alta calidad con respecto a nubosidad se propone. Para el SESLab, son importantes tres variables en la implementación del algoritmo:

- Porcentaje de nubosidad en la imagen
- Color de la nube
- Poder ignorar el sol en las imágenes (esto debido a que algoritmos como el KNN necesitan un bloque entre la posición del sol y la cámara para obtener una imagen sin luz del sol)

Se propone utilizar el algoritmo K-means Clustering debido a que cumple con las características anteriores si se realiza su debida interpretación. Como base para realizar esto se hace uso de las bibliotecas en Python (propiamente *scikit-learn* que está orientada al aprendizaje de máquinas y OpenCV para el análisis de imágenes). De aquí se puede obtener una base directa para la segmentación de imágenes, la obtención de valor de los agrupamientos en una escala RGB que varía entre 0 y 1, además de predecir tiempos de ejecución. De la figura 4.1 se puede extraer los valores vectoriales que posee el RGB y nombres comunes a saturaciones de cada componente (valor unitario en solo 1 de las 3 componentes del vector).

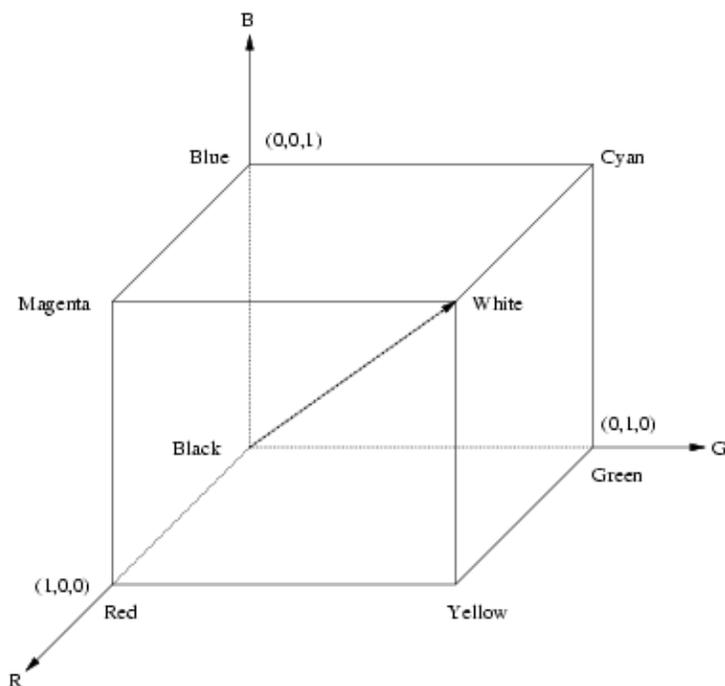


Figura 4.1: Tetraedro de comportamiento en valores RGB de k-means Clustering.

4.1.1 Primer Algoritmo Propuesto

En la primera propuesta de algoritmo se realiza con base al comportamiento de valores unitarios del tetraedro de la figura 4.1. La escala de grises (los colores supuestos para una nube) esta dada por el vector que inicia en $(0,0,0)$ y finaliza en $(1,1,1)$ contemplando todos sus valores intermedios. En un caso ideal cada cluster tendría un valor de (x,x,x) , en dónde x es un valor en 0 y 1, así este siempre sería una tonalidad de gris.

Sin embargo, la nubes no suelen ser perfectas combinaciones de grises y se desea brindar cierta libertad al agrupamiento en cuanto a sus valores. De forma que si un agrupamiento tiene un valor de (x,y,z) , ese punto en específico no pertenece al vector antes planteado, pero puede pertenecer a una tonalidad de gris ante el ojo humano y este debe ser considerado. Para ello se realiza un cilindro que de libertad al punto tridimensional pero que siga comportándose dentro de una uniformidad de color. Si el cilindro es muy delgado, el valor entre sus componentes $x-y-z$ no podrán tener gran diferencia. Conforme el cilindro aumente la diferencia entre esas componentes puede ir aumentando. La figura 4.2 ilustra la teoría de cilindros que se plantea para el algoritmo.

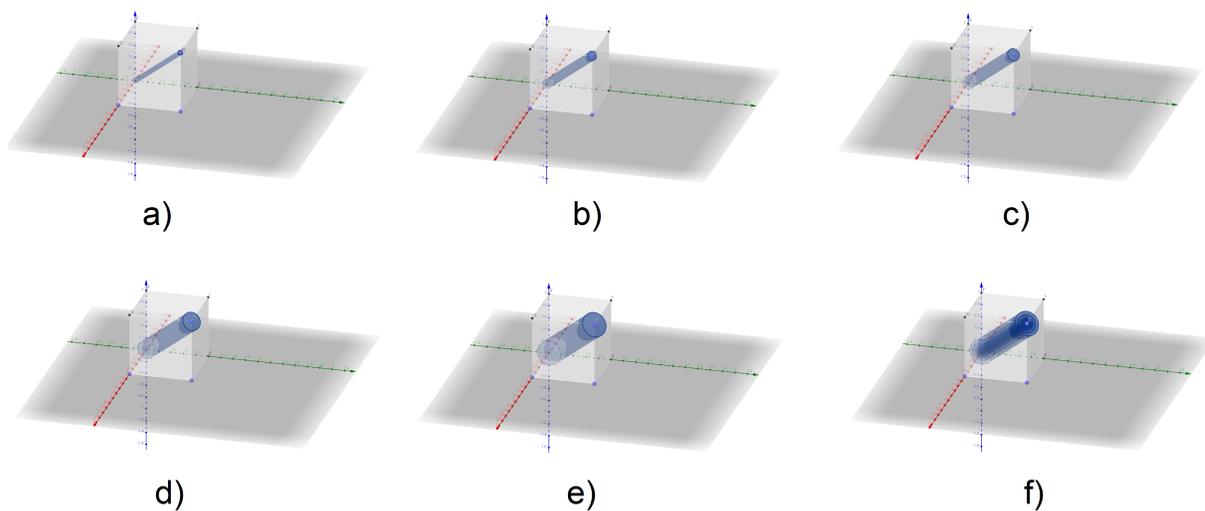


Figura 4.2: a) Cilindro de un 10% del volumen total, b) Cilindro de un 20% del volumen total, c) Cilindro de un 30% del volumen total, d) Cilindro de un 40% del volumen total, e) Cilindro de un 50% del volumen total, e) Visualización de todos los cilindros y sus volúmenes en un mismo plano.

Se plantea obtener un punto en el espacio tridimensional, lo cual haría que se pudiese definir si el agrupamiento se encuentra dentro del cilindro, sin embargo, para hacer más sencillo su análisis a la hora de programación no se buscará un espacio ni límites tridimensionales, ya que cargarían de un fuerte análisis al computador y por medio de una análisis 2D se puede obtener el mismo resultado mucho más sencillo. A continuación se planteará como se obtiene los límites adecuados para garantizar que el punto se encuentra dentro de un cilindro que abarca el 19% del volumen total del cubo RGB (esto para simplificar el cálculo), los demás tubos es simplemente repetir el proceso cambiando los valores que se indicarán en el proceso.

De un cálculo de áreas es posible obtener los valores para las funciones en cuanto a su intersección, los valores se obtendrían mediante la resta del área de 2 triángulos, 1 que contempla toda la área y otro que contempla el área prohibida por el cluster a partir de las ecuaciones (4.1)(4.2) y (4.3).

$$A_{T-completo} = \frac{(1)(1)}{2} = 0.5 \quad (4.1)$$

$$V_{interseccion} = \sqrt{2(A_{prohibida})} \quad (4.2)$$

$$V_{porcentaje} = \frac{A_{prohibida}}{A_{T-completo}} (100) \quad (4.3)$$

Primero se obtiene una visión en 2D para lo que sería el cilindro propuesto del algoritmo. En la figura 4.3 se observa como se visualizaría el cilindro en un plano 2D y que este sería igual para los 3 planos necesarios (X vs Y, X vs Z, Y vs Z).

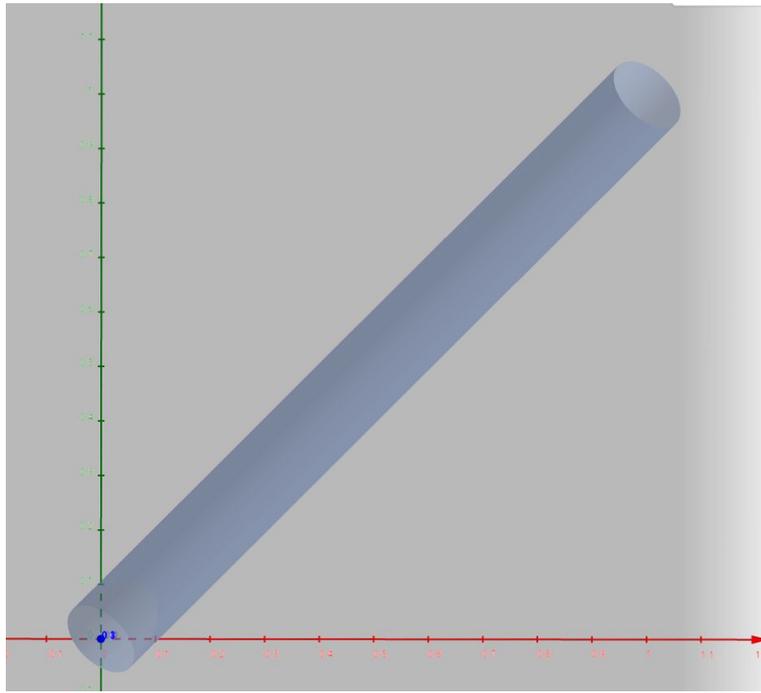


Figura 4.3: Vista del cilindro en un plano 2D, el cuál no tiene ningun cambio en sus diferentes valores de ejes.

De la figura 4.3 se extraen las funciones de los valores de las rectas que contiene el área en dónde se puede encontrar el punto. Estas serían para un plano Y vs X tal como se describe en (4.4) y (4.5):

$$y_{superior} = x + 0.1 \quad (4.4)$$

$$y_{inferior} = x - 0.1 \quad (4.5)$$

Para un plano Z vs X tal como se describe en (4.6) y (4.7):

$$z_{superior} = x + 0.1 \quad (4.6)$$

$$z_{inferior} = x + 0.1 \quad (4.7)$$

Para un plano Z vs Y tal como se describe en (4.8) y (4.9):

$$z_{superior} = y + 0.1 \quad (4.8)$$

$$z_{inferior} = y + 0.1 \quad (4.9)$$

Con estas funciones se puede analizar cada agrupamiento y por ende determinar si este pertenece al cilindro que hemos especificado, esto mediante la asignación siguiente: $R = x$, $G = y$, $B = z$. Sabiendo que el valor máximo de cualquier agrupamiento es de 1, no existirán problemas con los límites del tetraedro RGB de la figura 4.1. Además se contempla para el algoritmo que un valor muy alto de RGB son los rayos solares directamente y deben ser ignorados (considerados como cielo abierto), para esto se utiliza la siguiente premisa para cada valor (R,G,B):

$$V_{sol(RGB)} \geq 0.97 \quad (4.10)$$

El algoritmo como tal le asigna un valor o mejor dicho etiqueta a cada píxel, de forma que esté directamente asociada a algún agrupamiento existente, después del análisis de cada agrupamiento de si es considerado nube o cielo se reasignan estas etiquetas con un valor de 0 para cielo y de 1 para nube, de aquí se obtienen las cantidades asociadas a la nubosidad. El siguiente algoritmo 4 explica el funcionamiento del mismo.

Datos: imagen de Cielo

Resultado: Porcentaje de nubosidad y valor RGB de los clusters

initialization;

mientras programa en ejecución **hacer**

si valores de dirección de imagen correctos y cantidad numérica de clusters **entonces**

 iniciar ejecución de algoritmo k-means clustering;

 verificar posición de cada cluster en el cilindro descrito con sus límites;

 asignar a cada cluster una etiqueta;

 reasignar etiquetas de cada píxel;

 recrear la imagen en base s 2 colores, negro para cielo y blanco para nubes;

 guardar imagen recreada;

 guardar valores de los clusters;

en otro caso

 | indicar error de dirección y cerrar programa

fin

esperar el usuario vea las imágenes;

cerrar programa;

fin

Algoritmo 4: Comportamiento de programación para análisis de nubes mediante cilindro

4.1.2 Segundo Algoritmo Propuesto

En la segunda propuesta del algoritmo se pretende utilizar un método de aproximación conjunto a una pequeña base de datos. Esto debido a que la premisa que de las nubes solo poseen tonalidades de la escala de grises no es posible estimarlo.

A diferencia del algoritmo pasado, ya no se desea trabajar en la escala de 0 a 1 para los valores RGB, ahora se desea obtener su valor de 0 a 255, además se redondea su valor a un número entero y así utilizar la ecuación (4.11) para obtener su valor decimal.

$$R * 65536 + G * 256 + B = ValorDecimal \quad (4.11)$$

De esta manera se puede comparar con respecto a otros valores. A continuación lo que se hace es crear un pequeña base de datos con el valor decimal a partir de un valor hexadecimal conocido de los colores. Es decir se insertará una matriz de n cantidad de posiciones, en base a una determinada cantidad de valores. Como se muestra en la siguiente ecuación:

$$A_{valores} = \left(v_1 \quad v_2 \quad v_3 \quad \cdots \quad v_n \right) \quad (4.12)$$

Tenemos que cada valor es el decimal del color en RGB(a partir de una cantidad hexadecimal). Para este algoritmo se insertaron 350 valores, de los cuales los primeros 113 son solo colores de escala de grises y los 237 valores restantes pertenecen a escala de azules. De esta manera se puede ir evaluando que colores de azules pertenecen a nubes y que colores no, permitiendo una mejor aproximación al porcentaje de nubosidad contemplando algo más que escala de grises. A dicha matriz descrita en (4.12) se le resta este valor decimal del RGB al que pertenece el agrupamiento a analizar.

$$A_{valores} = \left(v_1 \quad v_2 \quad v_3 \quad \cdots \quad v_x \right) - ValorDecimal \quad (4.13)$$

Se continua por buscar el valor más bajo de esta matriz y se aproxima de forma abrupta que el color del agrupamiento al que pertenece está dado por esa posición. Una vez asignado pasa por varias etapas de análisis.

La primera etapa consiste en averiguar si este pertenece a los rayos del sol, para ello se da la siguiente premisa estructurada en la premisa (4.14) en donde se quiere saber si cada componente del RGB son superiores al valor de 235 y por lo menos 2 de estas son superiores al valor de 240, si esto se cumple dicho agrupamiento se considera como los rayos incidentes del sol, debido a que está muy cerca de ser una saturación total del RGB (blanco puro) que solo puede ser dotado por los rayos del sol..

$$V_{R,G,B} \geq 240 \wedge V_{R,G,B} \geq 240 \wedge V_{R-G-B} \geq 235 \quad (4.14)$$

En la segunda etapa se desea saber si es un valor que esté en la escala de grises pero siga poseyendo un rango aceptable entre sus valores RGB (semejante a los límites que existían en el cilindro), solo que esta vez se visualiza como si los valores absolutos entre los valores RGB no superen una diferencia de 20 unidades.

$$|V_R - V_G| \leq 20 \wedge |V_R - V_G| \leq 20 \wedge |V_G - V_B| \leq 20 \quad (4.15)$$

Si la premisa (4.15) se cumple entonces al agrupamiento se le considerará una nube, de lo contrario pasará a ser analizada en la escala de azules. Para ello el valor mínimo de la resta en la matriz tiene que ser un valor superior en cuanto a posición al 113. De igual forma para considerarse nube este debe ser evaluado igual que la forma anterior, solo que con rangos más amplios.

$$|V_R - V_G| \leq 34 \wedge |V_R - V_G| \leq 34 \wedge |V_G - V_B| \leq 34 \quad (4.16)$$

La en la premisa (4.16) permite da un espacio de libertad amplio para la clasificación de color perteneciente a la escala de azules. Si ests se cumple también se le considera nube. Si ninguno de estas clasificaciones se ajusta este será considera cielo por descarte.

En esencia el algoritmo reconoce 3 cosas:

- Cual cluster es el sol
- Cual cluster es una nube de escala de grises
- Cual cluster es una nube de escala de azules

Se podría concluir que lo que se considera cielo se realiza por descarte ya que no es una nube. Después de esta clasificación el proceder de etiquetar y construir imágenes es exactamente igual al algoritmo anterior. En el algoritmo 5 se observa el comportamiento del mismo.

Datos: imagen de Cielo

Resultado: Porcentaje de nubosidad y valor RGB de los clusters

initialization;

mientras *programa en ejecución* **hacer**

si *valores de dirección de imagen correctos y cantidad numérica de clusters* **entonces**

iniciar ejecución de algoritmo k-means clustering;

obtener valor RGB de 0 a 255 de cada cluster;

obtener valor decimal RGB de cada cluster;

verificar color de cluster por aproximación;

asignar a cada cluster una etiqueta;

reasignar etiquetas de cada pixel;

recrear la imagen en base s 2 colores, negro para cielo y blanco para nubes;

guardar imagen recreada;

guardar valores de los clusters;

en otro caso

indicar error de dirección y cerrar programa

fin

esperar el usuario vea las imágenes;

cerrar programa;

fin

Algoritmo 5: Comportamiento de programación para análisis de nubes mediante aproximación de colores de una matriz

Capítulo 5

Resultados y análisis de Nubosidad

5.1 Resultados y análisis de algoritmo propuesto K-means Clustering

Para las pruebas experimentales de este algoritmo se utilizó una cámara Nikon D3200, la cual posee características de control deseadas y explicadas en el capítulo 2 en la sección 2.3, de esta manera se podrán controlar dichas variables y analizar si el algoritmo procede correctamente y de forma óptima a lo deseado.

Gracias a su característica de auto ajuste, las variables ISO, Velocidad de disparo o Apertura de lente van a estar en sus valores óptimos para la fotografía. Por lo tanto se selecciona este modo y se pasan a realizar todas las pruebas, más adelante en otra sub-sección se analizará como estas variables influyen si no se posee el óptimo escenario.

5.1.1 Cantidad de valores K para algoritmo 1 y 2

El algoritmo propuesto se basa en una reconstrucción de imágenes con K cantidad de colores dados, estos son seleccionados aleatoriamente y para el ojo humano basta con tener aproximadamente 64 colores para considerarla prácticamente la misma imagen.

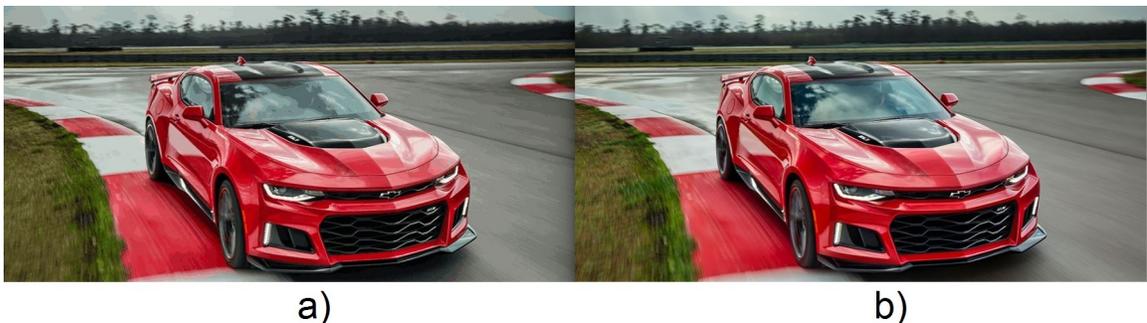


Figura 5.1: Comparación de reconstrucción de imagen a 64 colores por medio del algoritmo K-means Clustering, a)Imagen a 64 colores, b)Imagen a 96 615 colores

De la figura 5.1 se destaca que en la reconstrucción de imágenes no se necesitan miles de colores para considerarla adecuada para el ojo humano. Por lo tanto se intentará minimizar la cantidad de colores (la cual está asociada directamente con el valor K del algoritmo) a un punto en dónde la imagen se considere adecuada y a la vez posea la características necesarias para un análisis adecuado. Además el tiempo de ejecución del algoritmo varía con base a la cantidad de colores con que se desea reconstruir la imagen, este comportamiento se analizará con mayor detalle más adelante.

En la figura 5.2 se observa en el caso de las nubes la forma en que se va reconstruyendo la imagen dependiendo de su valor de K , permitiendo cada vez una visión más clara del comportamiento del algoritmo en cuanto a sus agrupamientos.

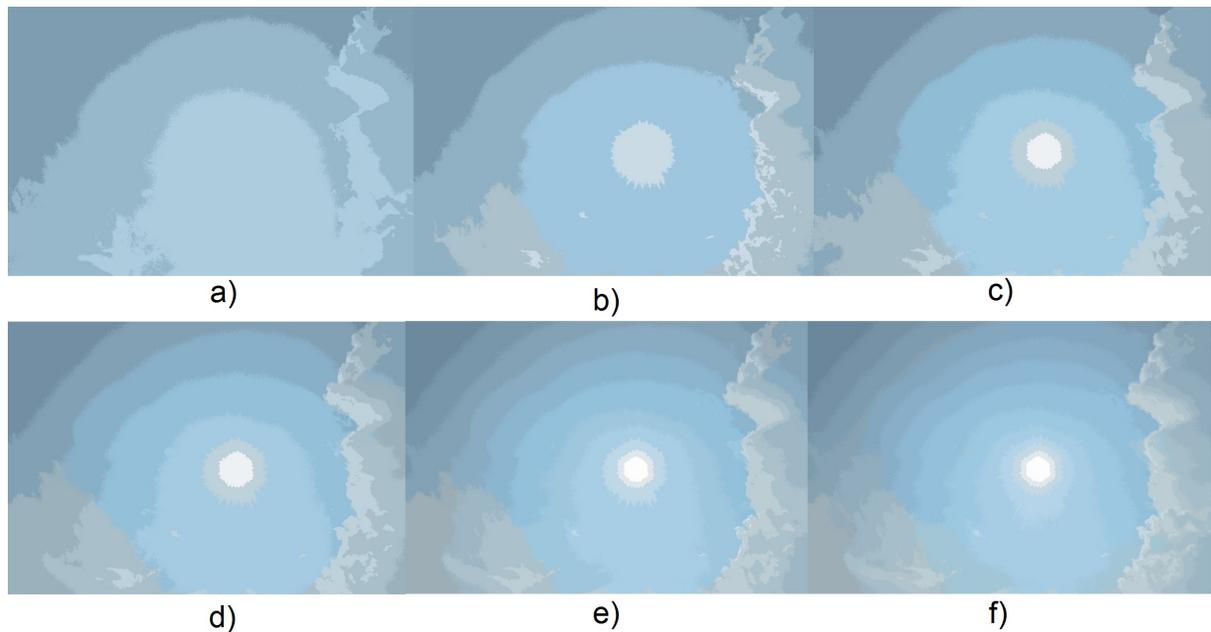


Figura 5.2: Reconstrucción de Imagen a diferentes valores de K , a) $K=4$, b) $K=6$, c) $K=8$, d) $K=10$, e) $K=16$, f) $K=22$

De la figura 5.2 se puede observar como conforme aumenta el valor de K , la reconstrucción de la imagen tiende a ser más precisa, la cantidad de colores permite tener una imagen más suave en la visualización, es decir, elimina los sectores que se consideran límites y van difuminando el color como es debido.

En el análisis que se está efectuando se hace uso de un algoritmo de aprendizaje de máquina no supervisado, por lo tanto se debe garantizar que los agrupamientos realizados por el algoritmo son los adecuados, en la figura 5.2 en la imagen *d*) se aprecia que la información es suficiente para realizar un análisis adecuado ya que el valor RGB del sol está definido cerca de un valor de saturación total (RGB de (255,255,255)) y la cantidad de agrupamientos para las nubes y cielo definen bien las figuras. Para las imágenes *e*) y *f*) el agregar más colores solo afecta en el cielo (agregando más agrupamientos diferentes al cielo), que precisamente se quiere todo lo contrario, se espera siempre que el color del cielo esté considerado lo más uniforme posible (mismo agrupamiento).

De esta manera valores muy altos de K no aportan información útil en las nubes, ya que estas suelen ser de tonalidades grises y algunos azules (caso que más adelante se le dará un solución parcial). Es importante destacar que se necesita obligatoriamente percibir el Sol, de forma que pueda ser discriminado en los algoritmos, esto provocaría que para valores bajos de K como 4, 6, 8, estén descartados ya que, a nivel de cielo se consideran óptimos, pero no pueden saber si se encuentra el Sol o no en la imagen, esto se puede notar mediante su análisis de agrupamientos de los cuales ninguno posee un valor cerca del vector (255,255,255).

Además para seleccionar el valor de K es necesario revisar el tiempo de ejecución del algoritmo y relacionarlo con la información que se necesita, para ello en el siguiente gráfico ilustrado en la figura 5.3, se observa el tiempo de ejecución con respecto a un determinado K dado.

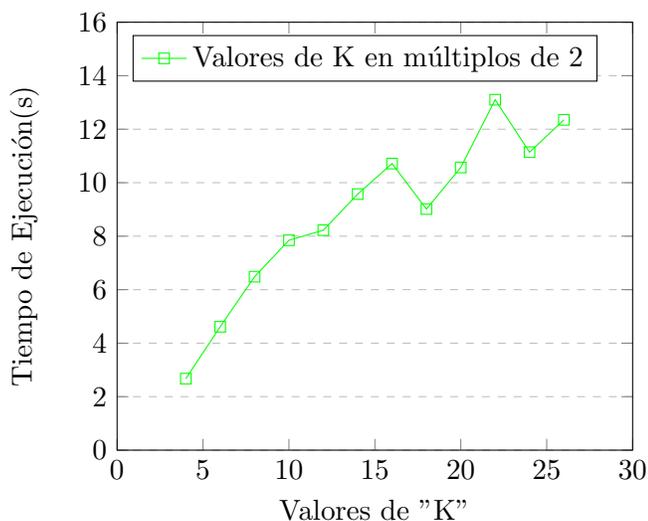


Figura 5.3: Gráfico de Tiempo de ejecución(s) vs Valores de "K" en un computador que posee 10 GB de RAM y dos núcleos de 2.5 GHz

De la figura 5.3 se obtiene que el tiempo de ejecución aumenta cuando el valor de K aumenta. Esto es una clara indicación de que a mayor número de colores a incluir, mayor tiempo de ejecución se ocupa, entonces se tiene un rúbrica importante, ya que se espera que el programa sea lo más eficiente posible, para ello se ha estimado que el tiempo de ejecución entre la Raspberry Pi 3 y el computador de pruebas diferencia en aproximadamente 10 veces y para efectos del proyecto no se desea que este tarde más de 90 s en ejecutarse. En la la figura 5.3 también se extrae que para valores de K entre 14 y 26 tienden a poseer mucha incertidumbre en cuanto al tiempo de ejecución (es decir no se puede suponer que a mayor K siempre habrá mayor tiempo de ejecución), pero igual contiene el comportamiento general de aumentar su tiempo de ejecución con respecto a valores de K menores. Para un valor de $K = 10$ se obtienen los resultados de información mínimos esperados para análisis de la nube y posee un tiempo de ejecución bajo en comparación a los K de mayor valor.

La imagen utilizada para esta prueba está en los valores óptimos de auto-ajuste de la cámara (ISO-100,F-13, Velocidad de Disparo-1/640, Megapíxeles-24.1). Estos valores es importante recordarlos ya que más adelante se dará una explicación de cada valor y su comportamiento, denotando que para cada prueba estos valores pueden cambiar.

5.1.2 Procentaje de nubosidad entre ambos algoritmos

Ahora bien los algoritmos intentan delimitar entre lo que es cielo o nube. En pocas palabras se intenta realizar una binarización de la imagen, que en este caso se espera lograr que el valor de 1 sea otorgado a la nube (color blanco) y el valor de 0 sea otorgado al cielo (color negro). Para ello primero se debe observar el comportamiento de las nubes ante la binarización de la misma. Se estudiarán 3 tipos de algoritmos que justifiquen la creación de uno nuevo para obtener valores congruentes a lo esperado en relación a lo que se puede observar por el ojo humano y lo que es analizado por el computador. Dichos algoritmos de fácil implementación permitirá entender las problemáticas presentes en el análisis de las nubes.

- Algoritmo Canny: en la figura 5.4 se observa lo que el algoritmo es capaz de detectar como bordes [28].

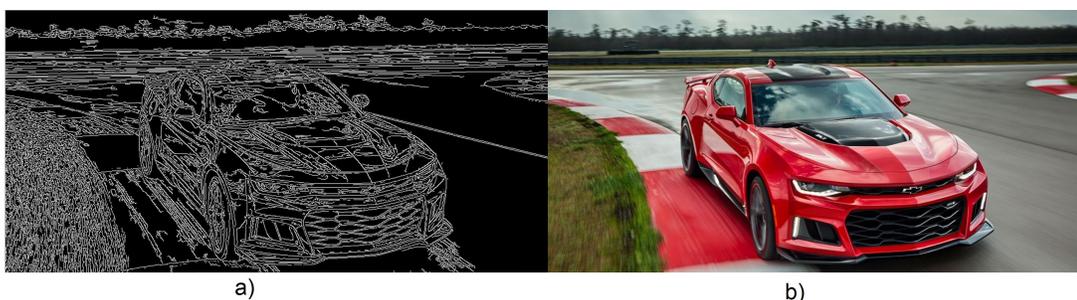


Figura 5.4: Detección de bordes de una imagen de un carro, a)Binarización de sus bordes, b)Imagen original.

Se esperan resultados similares ante una imagen de nubes, ya que poder determinar bordes podría crear áreas determinadas y así poder separar nubes de cielo, sin embargo, el resultado es desastroso comose observa en la figura 5.5.

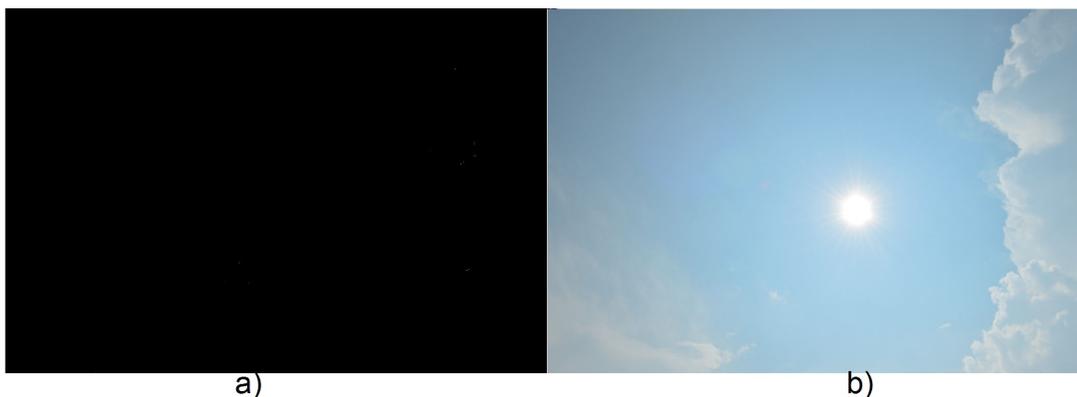


Figura 5.5: a)Binarización de los bordes de una nube, b)Imagen original.

Esto tiene una sencilla explicación, el algoritmo Canny por su comportamiento de buscar bordes basado en la conexión de píxeles entre sí, hace que no encuentre un solo borde. Esto debido a que

la nube visualmente aparenta tener un borde, pero está compuesta por millones de partículas gaseosas y líquidas, que se ven reflejadas en las fotos, no tiene un límite determinado y un algoritmo de esta índole es incapaz de reconocer algo para un análisis posterior.

- Algoritmo de Limitación General: para este caso se utiliza una binarización basado en un valor medio (127 para escala de grises) de forma que pueda realizar una segmentación muy general[29]. Como se puede observar en la figura 5.6.

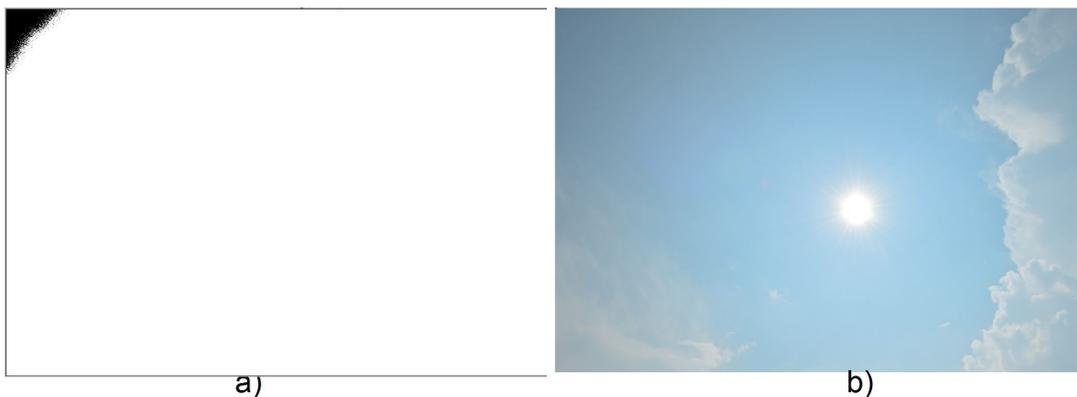


Figura 5.6: a)Binarización general de una nube con un valor de 127 en escala de grises, b)Imagen original.

De la figura 5.6 se puede ver claramente que no se obtiene ninguna información relevante que pueda ser utilizada para análisis del cielo y la nubosidad. El mayor problema recae cuando hay presencia del Sol y satura la escala de grises, dando colores por encima del límite dado, sin embargo, aunque no hubiese presencia del Sol, el comportamiento del algoritmo sigue siendo altamente impreciso de dónde la información se puede considerar irrelevante para un análisis formal.

- Algoritmo de Binarización de Otsu con filtro Gaussiano: se espera una mejor binarización ya que utiliza componentes de probabilidad para obtener el valor de cada píxel, en la figura 5.7 se observa el comportamiento del mismo[27].



Figura 5.7: a)Binarización por medio de algoritmo de Otsu, b)Imagen original.

De la figura 5.7 se aprecia que a pesar de obtener alguna información como lo es la pequeña nube de la izquierda y en la parte superior derecha, y por lo menos ya existe una área de cielo y una de nube, a simple vista se puede notar que posee un error altísimo en cuanto discernimiento de nubes y la información que nos provee de la misma no es útil.

Así se justifica el hecho de implementar un algoritmo más complejo, que permita obtener los valores de color de la nube en RGB y a la vez tenga una buena aproximación en cuanto a al reconocimiento de la nube nos referimos.

- El primer algoritmo propuesto se utilizan funciones matemáticas para obtener los valores de la nube y el resultado esperado es una aproximación bastante congruente en cuanto a la imagen de prueba. En la figura 5.8 se visualiza el comportamiento del mismo.

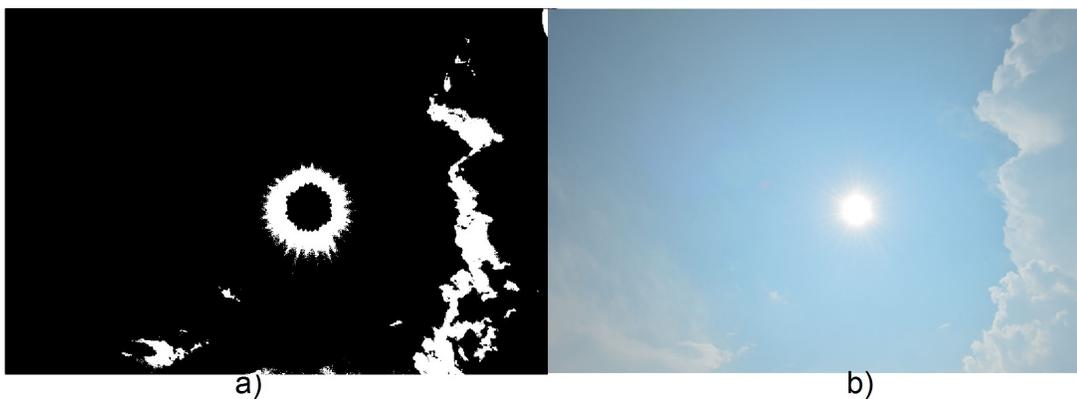


Figura 5.8: a)Binarización por medio de algoritmo 1 propuesto, b)Imagen original.

De la figura 5.8 se puede destacar como el algoritmo tiene un discernimiento primeramente de qué es el Sol y lo denota como espacio abierto, sin embargo, a lo largo de diferentes pruebas se puede ver este comportamiento de la creación de un aro alrededor del mismo, esto debido a que se espera que las nubes solo posean valores de escala de grises, y en ciertas secciones dadas de la imagen (alrededor del sol) algún agrupamiento forma un valor el cual cumple con esta descripción (pertenecer a escala de grises), ya que los rayos de luz se van difuminando y perdiendo su valor cercano al unitario en el RGB.

Se tiene una pequeña sección en la parte inferior izquierda que se ha perdido como reconocimiento de nube, pero no representa un valor alto de la imagen por lo tanto no es de mucha preocupación que el algoritmo falle en el mismo, por otro lado, podemos ver la clara pérdida de mucha información debido a que una sección de la nube es de una tonalidad azul y el algoritmo no está diseñado para discernir estos colores como una nube.

El porcentaje de nubosidad de la imagen asignada por el algoritmo es de 6.588% y se espera que por lo menos se considere un 10% de la imagen como nube. No es un error muy alto a simple vista pero afecta en otras imágenes analizadas, ya que las nubes no solo poseen escalas de grises sino también de azules e incluso rojos en los atardeceres.

- El segundo algoritmo propuesto se utilizan métodos más simples de restas y aproximaciones, a diferencia del primer algoritmo propuestos, el programa se puede depurar, ya que este su aproximación de colores yace en una base de datos guardada en una matriz, se le pueden ir agregando nuevos valores de otros colores (verdes, rojos, púrpuras, etc) al cual una nube puede pertenecer, a diferencia del primer algoritmo que lo único que se puede controlar es la libertad de diferencia entre el valor de un agrupamiento en sus valores RGB (los valores estipulados como x-y-z dentro del cilindro). Se utilizaron 50 imágenes de nubes en esta versión preliminar para lograr resultados congruentes y aceptables en cuanto a las nubes presentes en las imágenes, de manera que, la selección de colores en los arreglos que son considerados nubes son un promedio de cuantas veces dicho color realmente era una nube y cuantas veces el mismo color era cielo. En la figura 5.9 se pueden observar el comportamiento del algoritmo propuesto.



Figura 5.9: a) Binarización por medio de algoritmo 2 propuesto, b) Imagen original.

Es de mucha importancia observar como se ha logrado unificar el cielo con la sección del Sol, obteniendo una sola masa de cielo (que es lo que se deseaba) y ya no posee pequeñas secciones alrededor del sol. Esto indica que a pesar de existir un pequeño aro de difuminación, dicho color fue aproximado como cielo.

Esta vez las secciones que han sido como perdidas son las que el primer algoritmo propuesto había destacado como nubes, sin embargo, aunque parece inducir a un error, al contrario, se ha obtenido un porcentaje de nubosidad más alto. Este con un valor del 12.21% de nubosidad está mucho más cerca del valor de 10% de nubosidad esperado, el cual se reflejará como un porcentaje de error menor en la imagen. Quiere decir que se ha logrado una mayor aproximación en cuanto a nubosidad se espera. Además las pequeñas partes de nubes en la sección de la esquina inferior izquierda han sido reconocidos y tomado su posición como nube, lo cual es un excelente indicio de como se están formando los grupos por parte del algoritmo y que realmente el valor que escogimos para el K aporta suficiente información para este análisis.

5.1.3 Selección de algoritmo para prototipo

Ante los resultados expuestos se necesitan evaluar muchas más fotografías del cielo para determinar cuál algoritmo posee una funcionalidad más cercana a lo esperado. Para ello se va analizar varios escenarios. El primero es dónde se encuentre una foto de una nubosidad del 100%, el

segundo cuando no exista nubosidad del todo 0%, el tercero cuando haya nubes superpuestas de tonalidades diferentes y una nubosidad aproximadamente del 60%, el cuarto cuando exista una nube que esté totalmente definida y ante un fondo totalmente despejado y posea una nubosidad de aproximadamente al 5% y el quinto ante una nube de tonalidades rojizas (este última para poner a prueba le algoritmo bajo una condición no usual).

- Primer caso: se utilizará un imagen que se garantiza que el 100% de la misma es considerada nube, los resultados para ambos algoritmos se pueden observar en la figura 5.10.

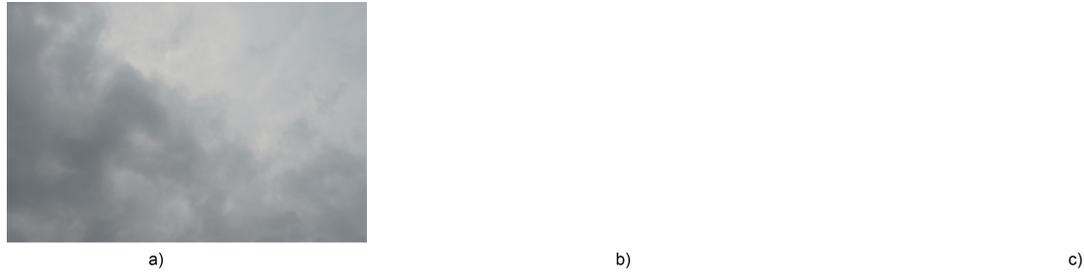


Figura 5.10: a)Imagen original, b)Binarización por medio de algoritmo 1, c)Binarización por medio de algoritmo 2.

De la figura 5.10 se puede destacar con claridad como se obtuvo en ambos casos un resultado totalmente positivo, en dónde el color blanco es considerado nube y tenemos 2 recuadros totalmente blancos(imágenes b) y c)), es decir, 100% de nubosidad para ambos algoritmos. En los dos casos se considera totalmente exitosa la prueba con un 0% de error en el análisis de la imagen. Este caso en específico todavía no nos da un criterio de cuál imagen escoger.

- Segundo caso: se utilizará una imagen que se garantiza que el 0% de la misma es considerada nube, es decir, un cielo despejado, los resultados para ambos algoritmos se pueden observar en la figura 5.11.

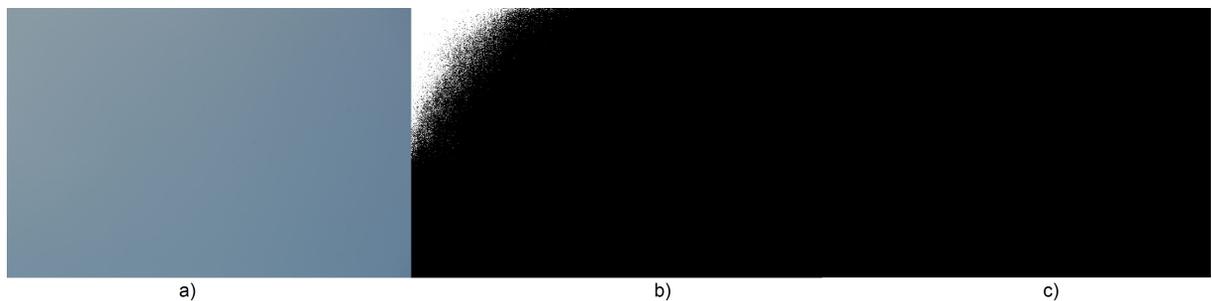


Figura 5.11: a)Imagen original, b)Binarización por medio de algoritmo 1, c)Binarización por medio de algoritmo 2.

De la figura 5.11 se observa que en esta ocasión si se existen cambios en la imagen analizada entre ambos algoritmos, en la imagen b) podemos ver como por el color en la esquina superior

izquierda cumple los requisitos para ser nube (un 5.37% de la imagen), sin embargo, esto no es una nube realmente y simplemente es un cielo de mucha claridad. En cambio en a imagen c) se denota como el algoritmo no ha fallado y toda la imagen es un recuadro negro que denota cielo, ns síntesis, posee un 0% de error en el análisis.

- Tercer caso: se tiene varias nubes superpuestas definidas como si existiesen bordes entre ellas, su calcula que aproximadamente la imagen posee un 60% nubosidad y es quiere saber el detalle con que se puede extraer información de la misma, los resultados para ambos algoritmos se pueden observar en la figura 5.12.

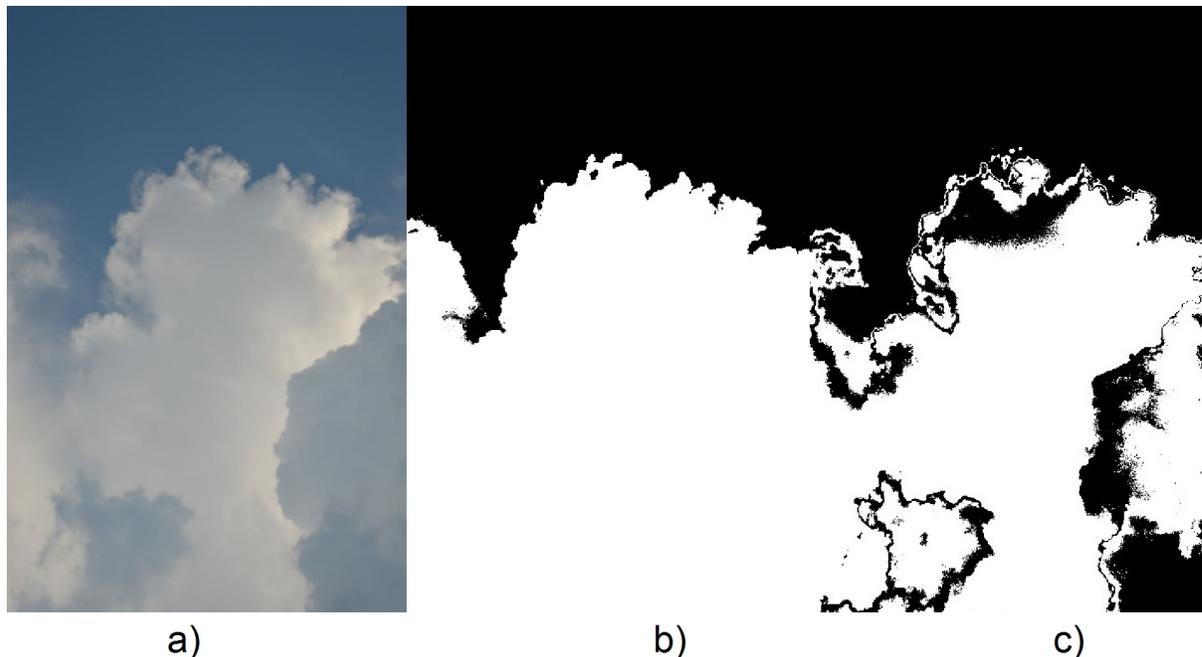


Figura 5.12: a)Imagen original, b)Binarización por medio de algoritmo 1, c)Binarización por medio de algoritmo 2.

De la figura 5.12 se puede observar una imagen reconstruida por parte del primer algoritmo, en donde sin importar las tonalidades ha considerado nube todas las secciones que consideramos correctas, en este caso se debe realizar un énfasis en varios detalles. El primero de ellos es que se ha perdido un poco de información de la parte central izquierda por aproximaciones de la nube, segundo se tiene una forma más uniforme de la nube en cuanto la binarización nos referimos, tercero si aproxima que un 60% de la imagen es nube, y del algoritmo se obtiene un 66.81% es nube, el porcentaje de error sería 11.34%.

En el segundo algoritmo se puede destacar el mayor detalle que existe con respecto a la diferenciación entre nube y cielo, si bien es cierto que existe un error en la esquina inferior derecha, podemos resaltar como el algoritmo es capaz de identificar cada nube por aparte (una característica muy deseada para poder discernir que agrupamientos realmente son una nube o no), además el error que existe en el algoritmo 1 para la parte central se ve corregida. Del algoritmo 2 se obtiene que el porcentaje de nubes de la imagen es de un 51.17%, lo que produce un porcentaje de error de 14.7%.

- Cuarto caso: Se tiene una nube de un tamaño muy pequeño, que se encuentra cerca del Sol y esto complica el algoritmo a nivel de coloración, además de que se podría decir que entre el fondo (cielo azul) y la nube existe una clara diferencia, los resultados para ambos algoritmos se pueden observar en la figura 5.13.

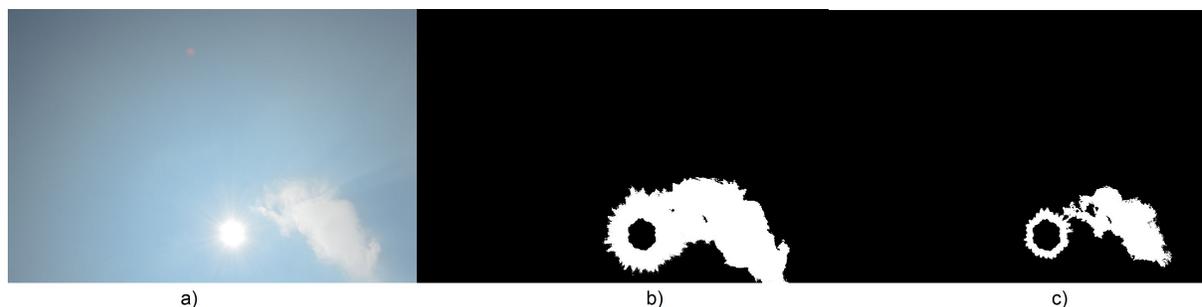


Figura 5.13: a)Imagen original, b)Binarización por medio de algoritmo 1, c)Binarización por medio de algoritmo 2.

De la figura 5.13 se obtiene un claro ejemplo del error constante que posee el algoritmo 1, este es menos sensible a pequeños detalles, por lo tanto las distribuciones existentes de nubes son muy altas, los colores del cielo que estén cerca del sol van a considerarse siempre como nubes y en la figura 5.8 ya se había destacado ese comportamiento y había sido corregido en el algoritmo 2, su porcentaje de error de 99% contra una distribución de aproximadamente 5% de nubes en la imagen, este error a pesar de que es casi total se puede interpretar como que simplemente para pequeñas nubes el algoritmo tiene dificultades en reconocer distribuciones.

Para el segundo algoritmo podemos ver una imagen mucho más detallada de la nube en binarización, bordes mejor definidos y una distribución de nube más pequeña y aproximada a la real, aunque existe este aro alrededor del sol, se puede ver con facilidad que su distribución es muy pequeña por lo tanto lo que aporta a la imagen en nubosidad es despreciable y por último su porcentaje de error sería 15.4%.

- Quinto caso: para el último caso se desea ver el comportamiento de la nube ante tonalidades rojizas de un atardecer, de manera que el comportamiento pueda ser entendido con mayor claridad, los resultados para ambos algoritmos se pueden observar en la figura 5.14.

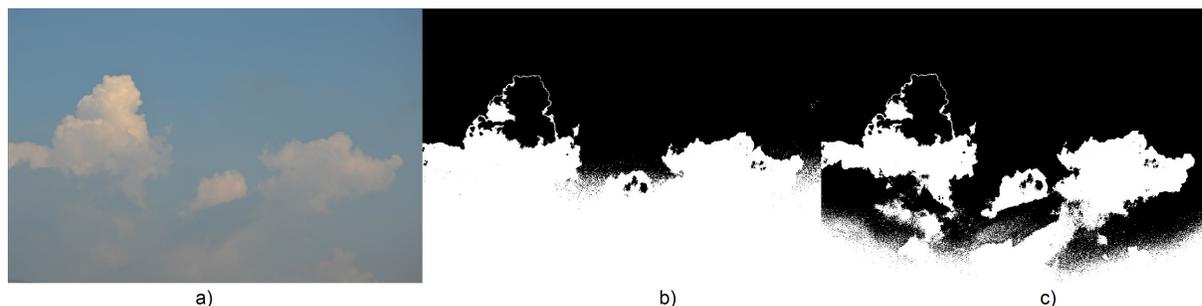


Figura 5.14: a)Imagen original, b)Binarización por medio de algoritmo 1, c)Binarización por medio de algoritmo 2.

En la figura 5.14 se aprecia que del algoritmo 1 se puede extraer una vez más que generaliza porciones que son parte del cielo azul, elevando la cantidad de nubosidad, generando un porcentaje de error mayor y haciendo un análisis más simple y sin detalle de la imagen. En el segundo algoritmo resalta una vez más la cantidad de detalle que se puede extraer de la imagen, las diferentes secciones de cielo y nube bien seccionadas. En ambos algoritmos se ve un comportamiento esperado que es el de esperar que la nube con una tonalidad rojiza sea considerada cielo, si bien en el primer algoritmo será así, en el segundo no, ya que este al hacer aproximaciones posee la característica de incrementar su base de datos de colores y de esta manera lograr que ahora esta sección, gracias a la extracción del color del agrupamiento sea introducida dentro de la base de datos para que sea considerada como nube y deje de interpretarse como cielo.

Ante el análisis realizado en varios escenarios expuestos anteriormente, y muchas más imágenes para entender el comportamiento de los mismos, se procede a escoger el algoritmo que cumpla las siguientes características, porcentaje de errores más bajo en general, mayor detalle de imagen, mínima creación del aro alrededor del sol y finalmente por posibilidad de depuración para obtener mejores resultados con respecto a tonalidad de escalas de otros colores. En el gráfico que se ilustra en la figura 5.15 se puede observar una comparación de los porcentajes de errores entre los algoritmos propuestos.

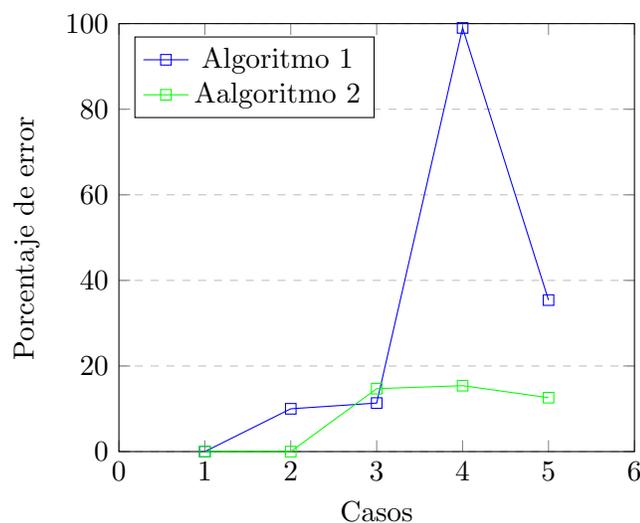


Figura 5.15: Gráfico de Porcentaje de error con respecto a la nubosidad esperada en la imagen vs casos, de ambos algoritmos propuestos para el prototipo.

Finalmente se concluye que el algoritmo que mejor se ajusta a las necesidades es el de la segunda propuesta, esto debido a que cumple todas las características esperadas y su porcentaje de error no fluctúa de forma tan alta como en el caso del primer algoritmo. Además de que permite mayor detalles para futuras referencias de la nube, y la posibilidad de mejorar la adquisición de diferentes nubes de otros colores que no necesariamente sean tonalidades de grises o azules.

5.1.4 Comportamiento del algoritmo ante diferentes tipos de nubes

A pesar de que el algoritmo suele ser una buena aproximación a la nubosidad dada, hay veces que pareciera obtener datos erróneos y esto se debe a que las nubes no siempre son iguales,

es más existen 10 tipos de ellas y cada una tiene un comportamiento diferente para el algoritmo. A continuación se analizarán las nubes que suelen dar mayor cantidad de anomalías en el comportamiento del algoritmo.

De diez tipos de nubes estudiados, se crean dos grandes secciones en dónde estarán clasificados todas las nubes.

- El primero está determinado por las siguientes tipos: Estratocúmulos, Cirrocúmulos, Altocúmulos, Cúmulos, Cumulonimbos y Nimboestratos. A estos le podríamos llamar nubes con *bordes*, por así decirlo sus formas están bien definidas visualmente, y a pesar de que sus bordes realmente no existe, es fácil identificar un grupo de esta índole. En la figura 5.16 se puede observar como el algoritmo obtiene con facilidad un cúmulo y lo diferencia del cielo.

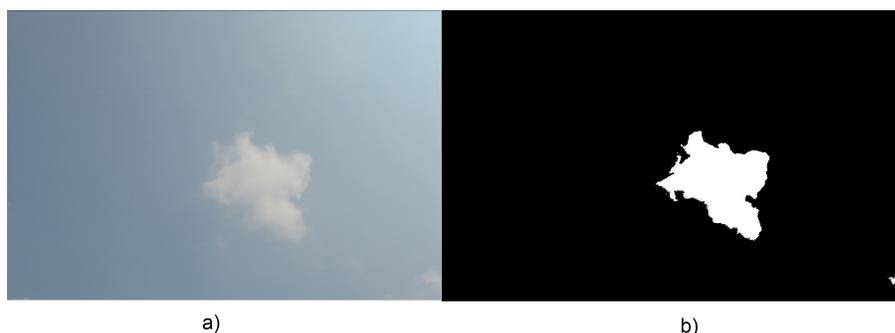


Figura 5.16: a)Imagen original, b)Binarización por medio de algoritmo propuesto para prototipo

- El segundo está determinado por las siguiente nubes: Estratos, Cirros, Cirroestratos y Altoestratos. Estas nubes se caracterizan por tener formas alargadas y sin *bordes* definidos. Este tipo de nubes hacen que el algoritmo identifique grandes masas de nubes donde no las hay. esto debido a la gran distribución que existe en una zona de píxeles separados. En la figura 5.17 se puede observar el comportamiento descrito.

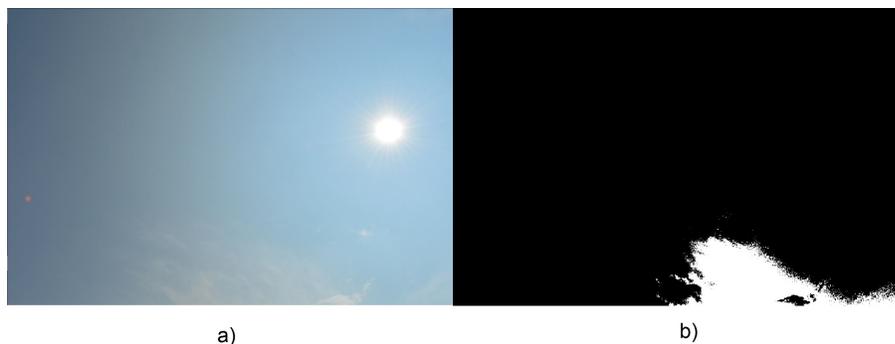


Figura 5.17: a)Imagen original, b)Binarización por medio de algoritmo propuesto para prototipo

Como se puede apreciar en la figura 5.17, para estos casos el agrupamiento de píxeles en K cantidad de colores, produce una anomalía, ya que existen muchos píxeles de nubes y cielo

revueltos. De esta manera el algoritmo aproxima muchos píxeles a un valor que realmente no les pertenece y la confusión produce una cantidad de nubosidad no deseada y en muchos casos que no es para nada despreciable.

5.1.5 Comportamiento del algoritmo ante diferentes variables de la cámara

En esta sección ya se tiene claro el algoritmo que se va a utilizar, por lo tanto se hicieron pruebas con la cámara propuesta (Arducam OV2640) y los resultados obtenidos no eran los esperados, por lo tanto se procedió a analizar el comportamiento de las variables de la cámara y como esta altera el comportamiento del algoritmo, de forma que se pueda dar explicación al comportamiento no deseado. En la figura 5.18 se puede observar el comportamiento de las imágenes tomadas por la cámara propuesta de bajo costo.

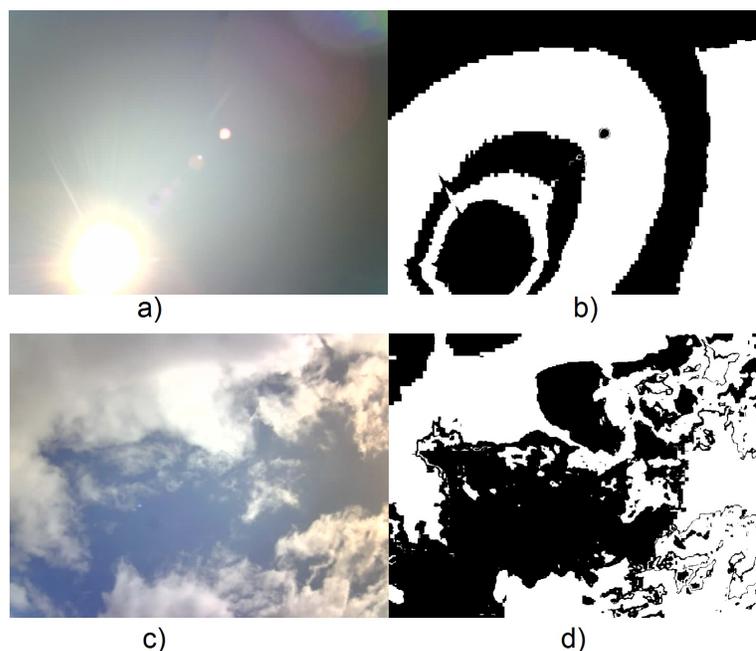


Figura 5.18: a) Imagen original Pi1, b) Binarización de Pi1 por medio de algoritmo 2, c) Imagen original Pi2, d) Binarización de Pi2 por medio de algoritmo 2.

Se observa en la figura 5.18 la imagen de un comportamiento no deseado, y para ello se debe comprender por qué se da, ya sea apertura del lente, velocidad de disparo, ISO o megapíxeles. Para esta prueba no se va a utilizar la cantidad de colores que percibe la cámara, ya que se sabe que es una cantidad inferior a la de una cámara semi-profesional (Nikon D3200), se intenta optar por una vía que no implique directamente la construcción del sensor de luz que viene integrado en las cámaras.

Para poder comprender el comportamiento no deseado del algoritmo en estas fotos, veremos su comportamiento a través de 4 variables vitales en la cámara: Resolución (Megapíxeles), Sensibilidad a la Luz (ISO), Velocidad de Disparo (Shutter Speed) y Apertura de lente (F Number). Cada uno se evaluará dentro de los rangos máximos y mínimos de la cámara marca Nikon modelo D3200 para efectos de validación.

Resolución (Megapíxeles)

En una imagen la cantidad de megapíxeles se puede ver directamente como la cantidad de información que existe sobre la imagen, no a nivel de color, pero si a nivel de nitidez. A mayor cantidad de píxeles, mayor es el análisis que se puede realizar gracias a que los grupos formados poseen menos aproximación en el algoritmo K-means Clustering. Para ello se hacen pruebas en calidad baja, media y alta de la cámara que serían 6, 13.5 y 24.1 megapíxeles respectivamente. En la siguiente figura 5.19 se establece un gráfico para una misma imagen pero con estos valores de píxeles y su tiempo de ejecución.

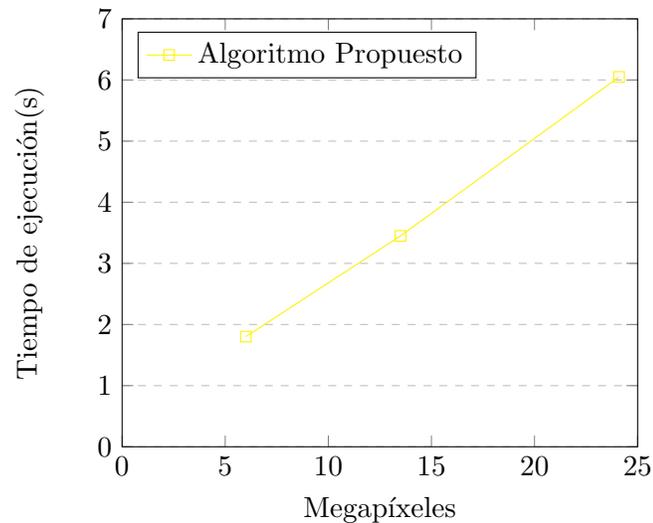


Figura 5.19: Gráfico de tiempo de ejecución vs cantidad de megapíxeles en un computador que posee 10 GB de RAM y dos núcleos de 2.5 GHz

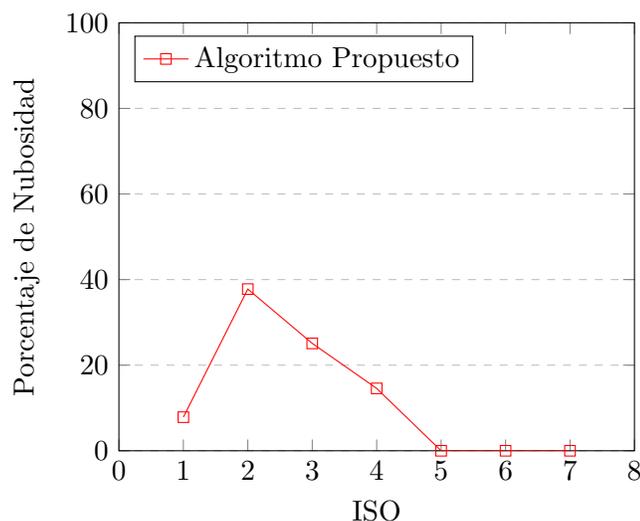
De la figura 5.19 se puede extraer que tiene un comportamiento considerado lineal del tiempo de ejecución con respecto a los megapíxeles de la imagen, siendo proporcionalmente creciente su relación. Esto es importante ya que los agrupamientos realizados por el algoritmo utilizado para el prototipo solo varían para valores bajos de píxeles (por debajo de 6 megapíxeles), pero en valores altos (por encima de los 6 megapíxeles) el cambio es nulo. Quiere decir que entre más bajo es el valor de los píxeles, los agrupamientos tendrán una forma más simple y se perderá detalle en la imagen, y esto se verá reflejado en un análisis inadecuado de nubosidad.

Sensibilidad a la Luz (ISO)

La sensibilidad a la luz es de las características más delicadas al tomar fotos, esto debido a que la percepción de la luz se duplica en cada valor y cambia abruptamente la imagen, se utilizaron todos los niveles de *ISO* que poseía la cámara para explicar este comportamiento. En la tabla 5.1 se puede ver una relación planteada de números del 1 al 7 con una relación de *ISO* de 100 a 6400 utilizada en el eje x de la figura 5.20.

Tabla 5.1: Relación de Número con ISO

1	2	3	4	5	6	7
100	200	400	800	1600	3200	6400

**Figura 5.20:** Gráfico de porcentaje de nubosidad en la imagen vs ISO, para análisis de comportamiento

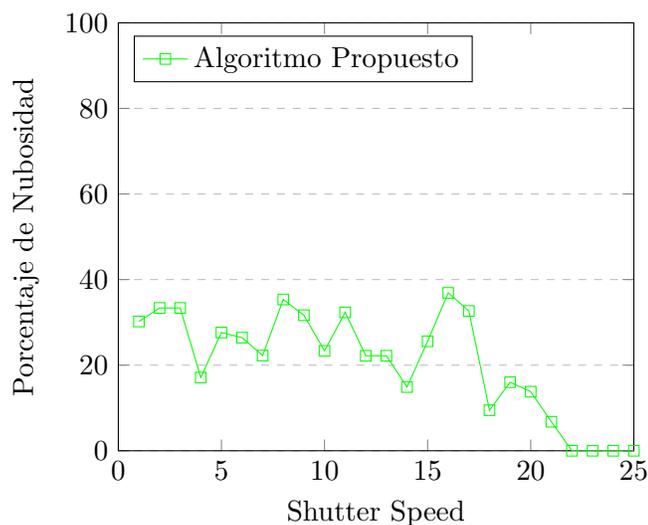
Un ISO muy bajo provocaría que no pueda definir bien los espacios de luz y se emitan errores en el índice de nubosidad, para este caso en específico el mejor valor de ISO es de 200 capta la imagen a la perfección y con los colores más cercanos a lo percibido por el humano, podemos ver que de ahí en adelante hay una pendiente negativa hasta llegar a 0% en el índice de nubosidad, que significa que apesar de que hayan nubes, la luz incide de forma tan directa que solo se ven fracciones blancas en la imagen, simulando así el color que posee el sol y por ende descartado por el algoritmo como nube y supone que toda la imagen es un cielo abierto.

Velocidad de Disparo (Shutter Speed)

Cuando una imagen tiene una velocidad de disparo muy rápida se traduce directamente como una imagen oscura (también en casos de movimiento como un congelamiento de la imagen) y para valores muy lentos se tiene mucho mayor tiempo de exposición y por ende imágenes más claras (también permite captar el movimiento de cuerpo celestes que parecen estáticos). Para esto en cada imagen se analizó el tiempo de disparo y se realizó la gráfica que se observa en la figura 5.21, además la tabla 5.2 indica la relación de los números en el eje x de la figura 5.21 con respecto a la velocidad de disparo utilizada para tomar la imagen.

Tabla 5.2: Relación de Número con Velocidad de Disparo

0	5	10	15	20	25	30
$\frac{1}{4000}$	$\frac{1}{1600}$	$\frac{1}{500}$	$\frac{1}{160}$	$\frac{1}{50}$	$\frac{1}{15}$	1

**Figura 5.21:** Gráfico de porcentaje de nubosidad en la imagen vs velocidad de disparo (Shutter Speed), para análisis de comportamiento

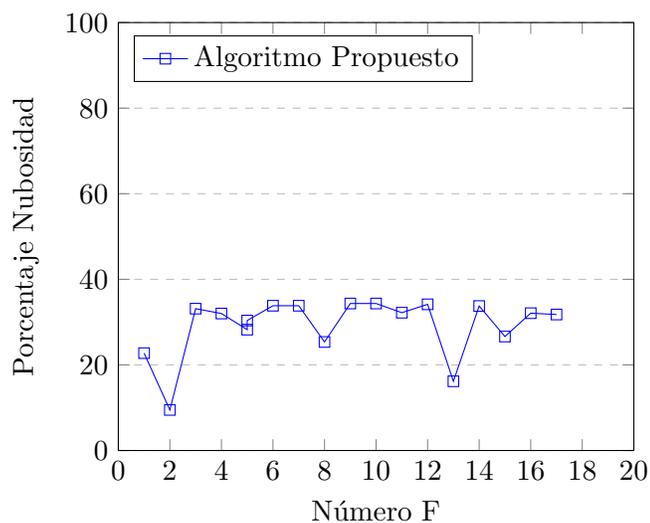
Si se observa en la figura 5.21 tres cuartos del gráfico, el comportamiento del primer valor 0 de disparo más rápido ($\frac{1}{4000}$) hasta el valor 20 de disparo ($\frac{1}{60}$), su comportamiento suele oscilar en una nubosidad entre el 20% y el 40%, es después de este que empieza a haber una pendiente decreciente hasta que la incidencia de la luz es demasiada como para permitirle al algoritmo realizar agrupamientos de colores diferentes, e igual que en el caso anterior con el ISO, tiene el valor del sol y supone que es un cielo abierto.

Apertura de lente (F Number)

La apertura del lente para efectos de fotografía suele determinar el enfoque que existirá entre los diferentes planos de distancia con respecto a la cámara, sin embargo, para efectos del cielo el enfoque apreciado de diferencia entre estos planos va a ser muy bajo debido a la existencia que existe entre el lente y el sujeto de enfoque, por lo tanto se analizará el comportamiento de la luz y el cambio de agrupamientos a través de todos los valores de F. En la tabla 5.3 indica la relación de los números en el eje x de la figura 5.22 con respecto a la velocidad de disparo utilizada para tomar la imagen.

Tabla 5.3: Relación de Número con Número F

1	2	4	6	8	10	12	14	16
3.5	4.5	5.6	7.1	9	11	14	18	22

**Figura 5.22:** Gráfico de porcentaje de nubosidad en la imagen vs número F, para análisis de comportamiento

Del gráfico que se observa en la figura 5.22 la característica más notoria es la de la poca fluctuación entre sus valores F, ante las pruebas se nota que por ser el caso del cielo y la distancia que existe entre el lente y el espacio que se desea obtener la foto, la apertura del lente es indiferente para este algoritmo.

Finalmente en la figura 5.23, se ilustra en la sección *a*) la imagen con sus valores óptimos (ISO-100, F-13, Velocidad de Disparo de $\frac{1}{640}$ segundos) y en la sección *b*) la imagen con sus valores más deficientes (ISO-3600, F-4.5, Velocidad de Disparo de 1 segundo). De esta se puede extraer como la suma de todas las variables en sus valores más deficiente para el análisis de nubes produce la ausencia de una imagen como tal, como se aprecia en la imagen *b*.

**a)****b)****Figura 5.23:** a) Imagen en óptimos valores, b) Imagen en valores más deficientes

Capítulo 6

Resultados y análisis de Variables Ambientales

6.1 Resultados y análisis de Variables Ambientales

Para validación de estos parámetros se propone verificar dichos valores con respecto a una estación meteorológica ubicada en el SESLab. Los valores a identificar principalmente serán Irradiancia Total, Temperatura de la Celda, Temperatura Ambiente y Humedad Ambiente. Estos valores permiten caracterizan el alrededor y algunas variables del panel (como la tensión de salida). Las otras variables propuestas a medir (Tensión en circuito abierto y corriente en corto circuito) son proposiciones extras que complementan el análisis completo del sistema.

La estación meteorológica que se utiliza para validar los datos, solo utiliza valores enteros, por ello se podrán visualizar porcentaje de error en el caso de las temperaturas y humedad que se pueden considerar despreciables practicamente. Además se estima que la estación calcula un promedio de 12 mediciones (1 cada 5 segundos) y refresca cada minuto con base a este promedio, ya que los programas a diferencia de la estación lo que recurren es a una forma promedial de 10 muestras en el instante que se pide la medición para evitar ruido en las medidas, podría reflejarse como diferencias entre las medidas experimentales de los sensores y las extraídas de la estación meteorológica.

6.1.1 Irradiancia Global

Para la irradiancia Global se utilizó el sensor marca Apogee modelo SP110 propuesto en el capítulo anterior. Su fiabilidad y sencillo uso hace que la colección de datos sea posible con un ADC sin necesidad de un circuito acondicionador de señal. En la tabla 6.1 se resumen los datos obtenidos durante una prueba con una frecuencia de recolección un minuto.

Tabla 6.1: Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Irradiancia Global

Irradiancia Apogee(W/m ²)	Irradiancia Estación Meteorológica(W/m ²)	Porcentaje de error(%)
552	575	4.00
527	538	2.04
506	523	3.25
408	420	2.86
370	375	1.34
368	390	5.64
374	392	4.60
405	413	1.94
379	401	5.49
390	398	2.01
400	407	1.72
466	475	1.90
454	474	4.22
407	419	2.86
441	454	2.86
441	456	3.21
420	433	3.00
362	376	3.72
363	376	3.45
365	376	2.93

Con base a los datos mostrados en la tabla 6.1 el porcentaje de error se encuentra generalmente por debajo al 5%, sin embargo, en varias el porcentaje de error está por encima al 5% , a lo que se le atribuye el hecho de que los piranómetros no se encuentran en el mismo lugar, el de la estación meteorológica se encuentra en un espacio totalmente abierto y elevado a aproximadamente 1.50 m en un lugar donde no se produce sombra, en cambio el piranómetro marca Apogee modelo SP110 se encuentra en la superficie del suelo cerca de un poste de luz y una estación de recarga cerca del SESLab (Refleja la irradiancia proveniente del Norte) a la hora de la toma de datos, lo cual puede inducir en la diferencia que se puede notar en varios casos.

6.1.2 Temperatura Ambiente

La temperatura ambiente será medida por medio del sensor SHT-31D mencionado en el capítulo anterior, y la temperatura a la que será comparada está dado por la estación meteorológica ubicada en el SESLab. La tabla 6.2 se muestran los datos obtenidos en una prueba de temperatura tomada cada minuto.

Tabla 6.2: Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Temperatura Ambiente

Temperatura SHT-31D(°C)	Temperatura Estación Meteorológica(°C)	Porcentaje de error(%)
31.369	31	1.19
31.554	31	1.79
31.573	31	1.85
31.203	31	0.65
31.441	31	1.42
30.843	31	0.51
30.798	31	0.65
30.384	30	1.98
31.125	31	0.40
31.203	31	0.65
30.758	31	0.78
30.904	31	0.31
30.7	31	0.97
31.189	31	0.61
31.151	31	0.49
31.680	31	2.19
31.696	31	2.25
31.237	31	0.76
30.818	31	0.59
31.101	31	0.33

Se puede analizar que en general la temperatura ambiente posee un error muy bajo (menor al 2%), sin embargo el comportamiento de la estación meteorológica de donde se obtenían los datos a de temperatura a comparar con bajas cifras significativas (solo 1 valor después del punto), induce a cifras altas de error como los valores de un porcentaje de error mayor al 2%, por lo tanto se necesita un reajuste del algoritmo en dónde solo se cuantifiquen datos con cifras que se encuentre en estos mismo valores, al tener mayor precisión en la temperatura, se aleja al valor comparado que se utiliza para validar.

6.1.3 Temperatura de Celda

La temperatura de la celda se validó en base a la temperatura ambiente, esto debido a que no se disponía de un panel en funcionamiento y la RTD que se utiliza posee un adhesivo de único uso, por lo tanto solo se dejó a la intemperie y se espera que el valor se aproxime al de la estación meteorológica. La tabla 6.3 se muestran los datos obtenidos en una prueba de temperatura tomada cada minuto.

Tabla 6.3: Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Temperatura de Celda

Temperatura Celda(°C)	Temperatura Estación Meteorológica(°C)	Porcentaje de error(%)
31.458	31	1.48
31.986	31	3.18
31.123	31	0.4
31.246	31	0.79
31.845	31	2.73
30.456	31	1.75
30.875	31	0.4
30.496	30	1.63
31.356	31	1.15
31.745	31	2.40
30.652	31	1.12
30.753	31	0.8
30.695	31	0.98
31.368	31	1.19
31.795	31	2.56
31.852	31	2.75
31.635	31	2.05
31.126	31	0.4
30.963	31	0.12
31.198	31	0.64

A pesar de utilizar una RTD de la cual solo se obtiene un valor de tensión, se tiene una muy buena exactitud con respecto a la temperatura ambiente de un porcentaje de error por debajo al 3% en la mayoría de las mediciones, esto nos garantiza que cuando esta sea colocada en un panel, la temperatura esperada será la que realmente es, y para análisis futuros será una fuente viable.

6.1.4 Humedad

La Humedad del ambiente será medida por medio del sensor SHT-31D mencionado en el capítulo anterior, y la humedad a la que será comparada está dada por la estación meteorológica ubicada en el SESLab. La tabla 6.4 se muestran los datos obtenidos en una prueba de temperatura tomada cada minuto.

Tabla 6.4: Mediciones experimentales de Estación Meteorológica, Circuito propuesto y porcentaje de error entre las mismas para Humedad

Humedad SHT-31D(%)	Humedad Estación Meteorológica(%)	Porcentaje de error(%)
46.999	47	0.00
46.431	47	1.21
46.401	47	1.27
46.892	47	0.22
47.881	48	0.25
48.792	49	0.44
48.568	48	1.21
47.532	48	1.00
48.121	48	0.25
47.851	48	0.32
48.121	48	0.25
47.771	48	0.49
46.689	47	0.66
46.697	47	0.64
46.185	46	0.39
45.359	46	1.36
45.882	46	0.25
46.850	46	0.32
46.949	47	0.11
46.475	47	1.12

En la humedad se percibe una muy buena exactitud, los valores se encuentran por debajo del 2%, a esto se le puede atribuir que dicha variable no suele tener cambios abruptos en el ambiente, sin embargo se debe tener cuidado en la instalación del sensor, debido a que si este posee un ambiente encerrado la humedad se concentra produciendo una humedad mayor y serán datos incorrectos lo que se obtiene del mismo, para ello siempre debe tener ventilas en el dispositivo en el que se vaya a acomodar.

Capítulo 7

Conclusiones y recomendaciones

7.1 Conclusiones

- Se demostró que es posible obtener un índice de nubosidad, colores de nubes e ignorar el sol sin necesidad de un obstáculo directo por medio de la implementación de un algoritmo basado en el K-means Clustering.
- Se evidenció que un método de aproximación de colores produce mejores resultados que un método de posicionamiento en el tetraedro RGB con base a un porcentaje de nubosidad en una imagen.
- Se demostró que las nubes no solo poseen colores en escala de grises, sino también en escalas de azules y rojos.
- Se evidenció que para obtener resultados congruentes con base a una imagen del cielo se necesita un equipo con características mínimas controlables para la toma de las mismas.
- Se demostró que con base al tipo de nubes existentes en la atmósfera el comportamiento del algoritmo y el análisis adecuado difiere del propuesto.
- Se comprobó que no se pueden utilizar algoritmos de binarización de imágenes convencionales para obtener índices de nubosidad.
- Se demostró que se obtienen datos fiables con base a un porcentaje de error menor al 2% para la humedad, 3% para la temperatura de la celda, 2% para la temperatura ambiente y 5% para la irradiancia global con el prototipo planteado.

7.2 Recomendaciones

- Realizar análisis para depurar las tonalidades rojizas que puedan tener las nubes.
- Realizar más pruebas a diferentes resoluciones para obtener un algoritmo que se ejecutó más rápido pero sin pérdida de información.
- Crear un algoritmo de identificación de nubes para evitar porciones de nubosidad falsas en el algoritmo ya propuesto.
- Utilizar una cámara con una cantidad de colores muy alta (mayor a los 50 mil colores) para obtener resultados congruentes.
- Colocar el sistema en un lugar que no produzca sombra o se pueda proporcionar una humedad diferente a la del ambiente.
- Utilizar algoritmos más complejos como un *watershed* o las combinaciones de otros (como se puede ver propuesto en la tesis de doctorado del Dr. Pablo Alvarado Moya) que produzcan mejores resultados y determine mejor las superficies deseadas.
- Realizar pruebas con imágenes en formato RAW para analizar si la información extra del mismo produce mejores resultados o si se puede adecuar el algoritmo en base a esta nueva información provista.
- Explorar la distribución de Rayleigh para obtener el color del cielo y de esta manera invertir el algoritmo y discernir que es cielo y que no.

Capítulo 8

Bibliografía

- [1] Organización Meteorológica Mundial (2017). ¿Qué es una nube?. [En línea] Disponible en: http://www.wmo.int/pages/index_es.html
- [2] A. Stratton: Electromagnetic Theory, New York: McGraw-Hill, 1941.
- [3] Hamblyn, Richard (February 28, 2011). The Invention of Clouds: How an Amateur Meteorologist Forged the Language of the Skies
- [4] Gobierno de España Ministerio de Educación. (2017). La Formación de las Nubes. [En línea] Disponible en: http://recursos.cnice.mec.es/biosfera/alumno/3ESO/energia_externa/nubes.htm
- [5] International Cloud Atlas/Atlas internacional de nubes; publicado por la World Meteorological Organization, Ginebra, Suiza. Publicación n.º 407.
- [6] Benchoff, Brian (17 de abril de 2016). Building the First Digital Camera . [En línea] Disponible en: | <http://hackaday.com/2016/04/17/building-the-first-digital-camera>
- [7] Ingrid Verbauwhede; Patrick Schaumont; Christian Pignet; Bart Kienhuis. (2005). Architectures and Design techniques for energy efficient embedded DSP and multimedia processing.
- [8] Real Academia Española. Diccionario panhispánico de Dudas. [En línea] Disponible en: <http://lema.rae.es/dpd/?key=p%C3%ADxel>
- [9] Alan H. Cromer. (2001). Física en la ciencia y en la industria. Pág. 480.
- [10] T. Northrup, C. Northrup .(2016) . Adobe Lightroom6/CC Training for Photographers.
- [11]Kylie Purtell. (2015). [En línea] Disponible en: <http://www.kyliepurtell.com/2015/05/what-is-iso.html>
- [12] T. Northrup. (2016) . How to Create Stunning Digital Photography. Mason Press Inc.
- [13] JPEG Standard (JPEG ISO/IEC 10918-1 ITU-T Recommendation T.81)
- [14] Carrasco, Jorge (2010). Cine y televisión digital. Manual técnico. Barcelona: Edicions de la Universidad de Barcelona.
- [15] Z. Sen, Solar Energy Fundamentals and Modeling Techniques: Atmosphere, Environment, Climate Change and Renewable Energy. (2008). Springer Science & Business Media. google-

Books-ID: 244OdSz4dNQC.

- [16] S. I. Ao and International Association of Engineers, Eds., World Congress on Engineering and Computer Science, WCECS 2008, San Francisco, USA, 22 - 24 October, 2008. Lecture notes in engineering and computer science. Hong Kong: IAENG International Association of Engineers, 2008, oCLC: 551898616.
- [17] F. M. González-Longatt. (2005) . Model of photovoltaic module in Matlab. II CIBELEC.
- [18] A. Cervantes Segura. Unidad de linealización y normalización para un estimador de parámetros de uso en un sistema de optimización de energía en paneles fotovoltaicos, "Tesis de Licenciatura. Escuela de Ingeniería Electrónica. ITCR. Cartago. Costa Rica. Junio 2016.
- [19] J. F. Ferichola. Caracterización de módulos fotovoltaicos con dispositivo portátil," Universidad Carlos III. Madrid. España, 2009. [En línea]. Disponible en: <https://e-archivo.uc3m.es/bitstreamhandle/10016/6037/PFCJulioFernandezFerichola.pdf?sequence=1&isAllowed=y>
- [20]G. Hamerly, C. Elkan. (2002).Alternatives to the k-means algorithm that find better clusterings. Proceedings of the eleventh international conference on Information and knowledge management (CIKM).[En línea] Disponible en: <http://charlotte.ucsd.edu/users/elkan/cikm02.pdf>
- [21] D. Skoog, F.Holler, S.Crouch E. (2008). Principles of Instrumental Analysis. Cengage Learning.
- [22] Oded Maimon and Lior Rokach (2010). Data Mining and Knowledge Discovery Handbook. Springer, New York.
- [23] P Alvarado. (2004). Segmentation of color images for interactive 3D object retrieval.
- [24] D. MacKay .(2003). "Chapter 20. An Example Inference Task: Clustering" (PDF). Information Theory, Inference and Learning Algorithms. Cambridge University Press.
- [25] A. De Los Santos. Fundamentos Visuales 2. La Teoría del Color. [En línea] Disponible en: <https://adelossantos.files.wordpress.com/2010/10/teroria-del-color.pdf>
- [26] C. Clapissa, Netdisseny. Teoría del color. [En línea] Disponible en: <http://www.weblogicnet.com/descargas/teoria-del-color.pdf>
- [27] OpenCV. Image Thresholding. Otsu's Binarization. [En línea] Disponible en: http://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html
- [28] OpenCV. Image Processing. Canny Edge Detector. [En línea] Disponible en: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html
- [29] OpenCV. Image Processing. Simple Thresholding. [En línea] Disponible en: http://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Global_Thresholding_Adaptive_Thresholding_Otsus_Binarization_Segmentations.php
- [30] Raspberrypi. Raspberry Pi 3 Model B. [En línea] Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [31] Adafruit. Sensors. Humidity. [En línea] Disponible en: <https://www.adafruit.com/product/2857>

-
- [32] Adafruit. Breakout Boards. [En línea] Disponible en: <https://www.adafruit.com/product/1085>
- [33] Apogee. Pyranometers. [En línea] Disponible en: <https://www.apogeeinstruments.co.uk/sp-110-self-powered-pyranometer/>
- [34] Sparkfun. Breakout Boards. [En línea] Disponible en: <https://www.sparkfun.com/products/12009>
- [35] Arducam. Camera Modules. [En línea] Disponible en: <http://www.arducam.com/camera-modules/2mp-ov2640/>

Apéndice A

Algoritmos

A.1 Algoritmo K-means Clustering

Existen algoritmos de aprendizaje de computadora subdivididos en 2 grandes categorías: supervisado y no supervisado. Los algoritmos de agrupamientos que no poseen una respuesta determinada por el programador se consideran de la segunda categoría (no supervisado). Ampliamente utilizados en la minería de datos, para crear agrupaciones de información que posean características similares.

Esto con el fin de poder dar análisis posteriores a estos datos, un ejemplo de ello son las características del genoma humano, se utilizan estos algoritmos para agrupar características y posteriormente darles un significado; también se pueden observar en mercadeo en dónde se agrupan las preferencias de las personas y con base a esa información se pueden crear productos.

El algoritmo K-means Clustering permite el análisis adecuado de las nubes mediante a agrupación de píxeles y posteriormente el análisis del resultados de estas agrupaciones. Es un algoritmo perteneciente a la minería de datos, este pertenece al campo de la estadística y las ciencias de la computación [22]. Es una técnica de minería de datos de tipo de agrupamiento o *Clustering*. Utiliza criterios de vectores de distancia, en donde se buscará ir cambiando los vectores de entrada con base en la información de aquellos que tengan características comunes y estén más cerca.

A.1.1 Comportamiento del algoritmo

En síntesis el algoritmo se comporta de la siguiente manera:

Algorithm: *k*-Means Clustering

```

1: initialize  $k$  centroids with  $k$  arbitrary points in the input set
2: assign each point to the cluster with the nearest centroid
3: repeat
4:   assign to each centroid the mean value of all points belonging to its cluster
5:   for each point in the input set do
6:     if nearest centroid belongs to a different cluster then
7:       reassign point to the cluster with the nearest centroid
8:     end if
9:   end for
10: until no cluster reassignments

```

Figura A.1: Pseudocódigo del algoritmo *k*-means descrito en [23]

Se obtienen un conjunto de observaciones (x_1, x_2, \dots, x_n) donde cada observación es un vector real de d dimensiones, el algoritmo *k-meansclustering* pretende una partición de estas observaciones en k conjuntos $S = (S_1, S_2, \dots, S_k)$, esto con el fin de minimizar la suma de los cuadrados existente entre sus grupos [24].

Se puede expresar de forma matemática mediante (A.1).

$$\arg \min \text{ de } S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg \min \text{ de } S \sum_{i=1}^k |S_i| \text{Var} S_i \quad (\text{A.1})$$

donde μ_i es la media de los puntos en S_i . Este es el equivalente a minimizar el par cuadrado de los puntos de desviación del mismo grupo.

$$\sum_{\text{Cluster } C_i} \sum_{\text{Dimension } d} \sum_{x, y \in C_i} (x_d - y_d)^2 \quad (\text{A.2})$$

A razón de que la varianza es constante, es equivalente al cuadrado de las desviaciones entre los puntos de diferentes grupos.

Paso de asignación: se asigna a cada observación un grupo en que la media produzca el mínimo valor de la suma de los cuadrados dentro del grupo [20]. De esta forma se generan los grupos basados en los diagramas de Voronoi (debido a que está basado en su distancia euclidiana). Matemáticamente mediante la ecuación (A.3).

$$S_i^{(t)} = \{X_p : \|x_p - m_i^{(t)}\|^2 \leq \{X_p : \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\} \quad (\text{A.3})$$

Paso de actualización: se calcula la nueva media en donde los centroides de cada agrupamiento debe ser colocado [20].

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (\text{A.4})$$

Estos dos pasos se realizan hasta que se cumpla un número de ciclos deseados o hasta que converga (su asignación de espacio de centroide ya no cambia). Visualmente se puede explicar en la A.2.

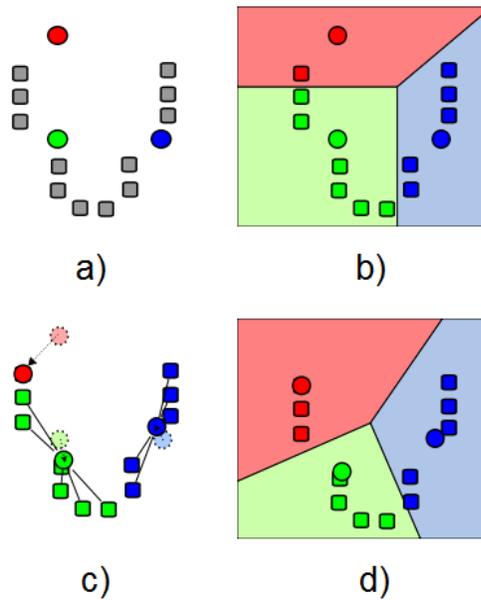


Figura A.2: a) Selección aleatoria de centroides, b) Creación de grupos en base a los centroides, c) Reajuste de los centroides, d) Reajuste de los grupos en base a la nueva posición de los centroides. El paso c y d se repiten hasta llegar a la convergencia

Apéndice B

Diseños Extras

B.1 Diseño Módulo Principal para Prototipo

Para efectos del SESLab se propone que exista un módulo en donde se integren todos estos sensores, circuitos y la Raspberry Pi 3. Para ello utilizó la herramienta de Autocad para conformar un módulo principal que permita conexiones entre otros, es decir, deje la libertad de crear otro módulos e implementarlos por simples conexiones a este que se propone. En la figura B.1 se encuentra lo que va a ser una caja color negro mate para corte láser en acrílico y su construcción. Esta posee dimensiones de 196 mm x 106 mm.

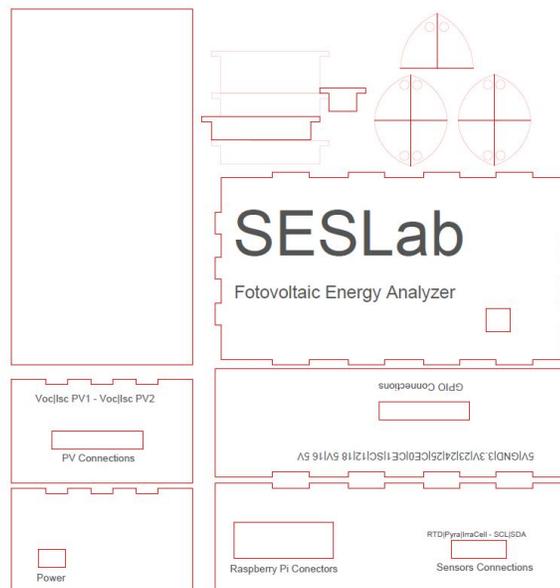


Figura B.1: Diseño en Autocad de Módulo Principal para el prototipo

Una vez realizado el corte láser y pegadas sus partes se obtiene como resultado una caja para el módulo del prototipo como se puede observar en la figura B.2

B.2 Tensión V_{oc} (circuito abierto) y corriente I_{sc} (corto circuito) del panel fotovoltaico

Al analizar las variables de un panel, se espera obtener sus valores máximos, se proponen 2 circuitos, que permiten la lectura de los mismos.

- Para el circuito abierto, se utiliza una resistencia de muy alto valor (igual o mayor a $10M\Omega$), para este caso se toma en cuenta que a una ganancia de $2/3$ en el ADC para poder cuantificar tensiones de hasta $6V$. El ADS1115 tiene una resistencia característica de $10M\Omega$ y puede recibir un máximo de $10mA$, de esta manera se diseña un divisor de tensión que contempla la condición (B.1).

$$V_{out} = \frac{V_{in}R_2}{R_1 + R_2} \quad (B.1)$$

R_2 sería el paralelo entre la resistencia característica del ADS1115 y la que se agregue por efectos de diseño, en este caso otros $10M\Omega$ y el valor es descrito por (B.2):

$$R_T = \frac{R_{ADS}R_e}{R_{ADS} + R_e} \quad (B.2)$$

Sustituyendo los valores de R_{ADS} y R_e

$$R_T = \frac{10M * 10M}{10M + 10M} = 5M\Omega \quad (B.3)$$

Finalmente tenemos que $R_T = R_2$, por lo tanto con un valor de $R_1 = 100M\Omega$ tendríamos los siguientes valores de lectura por el ADS1115:

$$V_{out} = \frac{V_{in} * 5M}{100M + 5M} \quad (B.4)$$

$$V_{out} = \frac{V_{in}}{21} \quad (B.5)$$

Con esto podemos concluir que la tensión en circuito abierto que podemos medir mediante el ADS1115 tiene un amplio rango de $5V$ hasta los $120V$, siempre y cuando su ganancia se mantenga en $2/3$, es decir, ya con el dato extraído se procede a analizar por medio de programación el dato y obtener el valor de V_{oc} del panel.

- Para el corto circuito, se utiliza una resistencia muy baja, en teoría sería conectar los terminales del panel fotovoltaico entre sí, sin embargo para poder obtener un valor de tensión relativo a la corriente que existe entre estos terminales, se conecta una resistencia de una resistencia para corto circuito, que en pocas palabras, es una resistencia de un valor muy bajo (por debajo de $100m\Omega$), que se considera casi como un cable (una fracción de aluminio que se conoce el valor de impedancia para este caso).

Se debe medir la tensión en el extremo positivo de la resistencia con referencia al extremo negativo de la misma, utilizando una resistencia de $50m\Omega$ y mediante la ley de Ohm se conoce el

valor de corriente que circula por la resistencia. La tensión requerida se obtiene de la ecuación (B.6).

$$I = \frac{V_{out}}{R} \quad (\text{B.6})$$

Sustituyendo la R por la resistencia conocida de $50 \text{ m}\Omega$ se obtiene la siguiente ecuación (B.7).

$$I = \frac{V_{out}}{0.05} \quad (\text{B.7})$$

De (B.7) se obtiene que para 1 A de 0.05 V , además se sabe que la corriente de los paneles comunes puede oscilar entre los 7 A a 15 A , colocando estos valores en la ecuación se obtiene una tensión de 0.35 V a 0.75 V respectivamente para los valores, los cuales son cantidades sencillas de medir por el ADS1115 en ajustado en una ganancia 4 que sería directamente una impedancia de $6\text{M}\Omega$ y no existirían ningún problema de corriente incidente en el ADC.

Ante el problema de medir el mismo panel variable diferentes, se establece un sencillo circuito de conmutación mediante un relé (el cual deberá ser dimensionado en base a la cantidad de corriente máxima a medir con un panel). Para ello se hace uso de los GPIO, así se puede controlar el tiempo de conmutación que existe entre un circuito y el otro, ya que los GPIO poseen tener señal en alto (3.3V) o en bajo (0V), para los relés utilizados producen un problema y aunque existen relés que pueden ser conmutados a ese nivel de tensión (3.3V) se hará uso de un convertidor de nivel lógico bidireccional. El BSS138 descrito en el capítulo 3 en la sección 3.2 soluciona el acondicionamiento de señal para los *GPIO*.

B.3 Irradiancia en una celda fotovoltaica

El valor de irradiancia que se obtiene en una celda fotovoltaica difiere en una cantidad considerable con respecto a un piranómetro debido a que esta solo percibe la irradiancia perpendicular al plano en el que se ubica (grado de inclinación.) Para comprobar esta premisa se utiliza el sensor *SpektronIrradiationSensor210*, el cual consiste en una celda monocristalina ya antes estudiada, su conexión es sencilla (positivo y negativo para obtener tensión) y su sensibilidad a la irradiancia está dada por un valor único indicado en cada sensor a $1000\text{W}/\text{m}^2$, en este caso su relación es de 89 mV para $1000 \text{ W}/\text{m}^2$. De igual manera a pesar de tener una tensión muy baja para valores de irradiancia, para el ADS1115 no es ningún problema su lectura al poseer una ganancia de 16 para el canal que se le asigne este sensor.